electronics

MDPI

Article

An GNSS/INS Integrated Navigation Algorithm Based on PSO-LSTM in Satellite Rejection

Yu Cao [1], Hongyang Bai [1,2,*], Kerui Jin [1] and Guanyu Zou [1]

[1] School of Energy and Power Engineering, Nanjing University of Science and Technology (NJUST), Nanjing 210094, China; caoyu@njust.edu.cn (Y.C.); kr223854@njust.edu.cn (K.J.); gyzounjust@njust.edu.cn (G.Z.)

[2] Key Laboratory of Maritime Intelligent Cyberspace Technology, Nanjing University of Science and Technology, Ministry of Education, Nanjing 210094, China

* Correspondence: hongyang@njust.edu.cn

Abstract: When the satellite signal is lost or interfered with, the traditional GNSS (Global Navigation Satellite System)/INS (Inertial Navigation System) integrated navigation will degenerate into INS, which results in the decrease in navigation accuracy. To solve these problems, this paper mainly established the PSO (particle swarm optimization) -LSTM (Long Short-Term Memory) neural network model to predict the increment of GNSS position under the condition of satellite rejection and accumulation to obtain the pseudo-GNSS signal. The signal is used to compensate for the observed value in the integrated system. The model takes the advantages of LSTM, which is good at processing time series, and uses PSO to obtain the optimal value of important hyperparameters efficiently. Meanwhile, the improved threshold function is used to denoise the IMU (inertial measurement unit) data, which improves the SNR (signal-to-noise ratio) of IMU outputs effectively. Finally, the performance of the algorithm is proved by actual road test. Compared with INS, the method can reduce the maximum errors of latitude and longitude by at least 98.78% and 99.10% while the satellite is lost for 60 s, effectively improving the accuracy of the GNSS/INS system in satellite rejection.

Keywords: GNSS/INS integrated navigation; particle swarm optimization; LSTM neural network; threshold denoising; satellite rejection

check for updates

Citation: Cao, Y.; Bai, H.; Jin, K.; Zou, G. An GNSS/INS Integrated Navigation Algorithm Based on PSO-LSTM in Satellite Rejection. *Electronics* **2023**, *12*, 2905. https://doi.org/10.3390/electronics12132905

Academic Editor: Sergio Busquets-Monge

Received: 30 April 2023
Revised: 25 June 2023
Accepted: 30 June 2023
Published: 2 July 2023

1. Introduction

Navigation technology uses scientific principles and methods such as electricity, magnetism, light and mechanics to locate and guide the motion process of a moving carrier, and is broadly used in important fields such as weapons science, aerospace and transportation, etc. Navigation system refers to the general term of all equipment combinations that can complete certain navigation and positioning tasks. At present, the main navigation systems include radio navigation system [1,2], satellite navigation system [3–5], astronomical navigation system [6], inertial navigation system [7,8] and integrated navigation system [9–11], etc.

Among the mainstream navigation systems, satellite navigation system and INS are most common. The satellite navigation system can realize all-weather global positioning [12], which requires short observation time and has high positioning precision. In addition, the positioning error will not accumulate over time. However, GNSS requires receivers to receive signals from satellites, and in areas where signals are susceptible to interference, such as around tall buildings, indoor spaces and forests, where the need for the location may not be met due to signal loss. Moreover, the frequency of GNSS signals is relatively low. The inertial navigation system is an autonomous navigation system. It uses the IMU's own measurements to calculate location, without the need to receive an external signal. In addition, the output frequency of INS is high and the concealment is stronger. However, the IMU used in general INS belongs to the MEMS (micro-electro-mechanical system), which has the advantages of small size and low price. However, the output of

MEMS is susceptible to vibration and other factors. In addition, since the INS requires integration, its errors will accumulate over time, which may lead to large differences in navigation results over a long period of time [13]. Therefore, scholars have proposed a method composed of GNSS and INS which contains the advantages of GNSS and INS. The navigation result has high precision, fast refresh rate and mighty stability. It has achieved good results in practical applications and become a mainstream integrated navigation method [14]. In fact, GNSS/INS integrated navigation systems inevitably have disadvantages. GNSS is prone to signal loss due to obstruction and electromagnetic interference, especially in complex environments such as cities and battlefields. When GNSS signal are lost, the integrated navigation system can only degenerate into pure inertial navigation and rely on mechanical choreography to solve. Considering the limited performance of IMU used in general GNSS/INS systems, the navigation accuracy of the GNSS/INS system will be greatly compromised when the satellite signal is lost. Therefore, in the case of satellite signal loss, how to compensate the GNSS/INS integrated navigation system through other methods has become an advanced research hotspot.

Now scholars have proposed many methods to compensate for integrated navigation while GNSS is out. Most of them mainly begin with two aspects of an integrated system. One aspect is to improve the system input, such as preprocessing the data, or adding auxiliary sensors to form a multi-source information fusion system and using other auxiliary sensor information to correct the INS [15,16]. The other aspect is to improve the processing methods of INS and GNSS data. Specifically, it can be divided into improving the data fusion algorithm [17] and using artificial intelligence technology [18].

In the aspect of data preprocessing, because of the complicacy of the operating environment of the INS, the IMU signal inevitably makes noise when in use, removing the noise component of the IMU signal before the inertial navigation solution has become a pivotal technology for processing the IMU signal. Wavelet transform method is broadly used in the signal denoising, among which the wavelet threshold denoising method is the most commonly used signal processing method in engineering applications because of its advantages of easy implementation and preserving the original signal characteristics [19,20]. The traditional hard threshold function will produce oscillations in reconstruction and it is not continuous. The soft threshold function will have a certain deviation when the wavelet coefficient is large, so the paper uses a continuous threshold function, which can retain more useful signals, so as to effectively improve the SNR (signal noise ratio) and finally improve the navigation accuracy. In terms of adding sensors to form a multi-source integrated navigation system, sensors such as visual odometry and lidar are increasingly being used. A visual odometer [21–24] can estimate the moving process of a moving carrier by image sequence and its algorithm is relatively mature and simple, but its accuracy is easily affected by image quality and environment. Lidar [25–29] is widely used in integrated navigation because it can detect on a large scale and obtain 3D information. However, it is subjected to environment and price. Of course, there are many other sensors that can be used in this research, such as magnetic sensor [30,31], doppler velocimeter [32] and altitude sensor [33]; however, adding sensors still faces many problems, although it can improve navigation accuracy, such as the increase in calculation and economic costs, extra weight and the failure rate also need to be considered.

In terms of improving capabilities of data fusion, except for the factor graph optimization (FGO) [34], the Kalman filter is more often used in practical projects, which is a stable and reliable method verified by engineering. KF is a method with linear unbiased minimum variance estimation, which is often used to fuse navigation data in engineering [35–37]. Because most of the integrated navigation systems are nonlinear and have colored noise, the deviation of the KF estimate is large. To solve these problems, many improved Kalman filters have been proposed and utilized. R.E. Kalman et al. [38] proposed the extended Kalman filter (EKF) algorithm. However, ignoring higher-order terms of the Taylor expansion will cause the EKF to diverge while the nonlinearity of the system is obvious. In 1988, Carlson [39] proposed a distributed filter, the federated Kalman filter, which solved the disadvantages of the centralized filter and improved the filtering accuracy. Y. Zhao et al. [40] proposed a novel information fusion method based on the complementary filter for the SINS/CNS/GPS integrated navigation system to solve the problems of heavy computational load and poor real time of the FKF. An improved innovation adaptive Kalman filter (IAKF) was proposed by B. Sun et al. [41] in 2022 to solve the vulnerability of KF in challenging environments. Compared with ordinary KF, this method improves the adaptability. To sum up, the traditional fusion method represented by the Kalman filter is to process the raw data from the sensor. It is cheap and scalable, but observations in satellite rejection environment are limited and do not address the underlying problem.

In the aspect of artificial intelligence technology, neural network has good nonlinear modeling ability, which is suitable for error modeling of an integrated navigation system. C. Jiang et al. [42] proposed an AI method to denoise the MEMS IMU output signals by RNN, which improved the precision of inertial navigation system by reducing the error. This method takes advantage of the characteristic that RNN is suitable for dealing with time series problems. J. Zhu et al. [18] used a LSTM/SVR-VBAKF algorithm to aid the integrated navigation system when the Doppler velocity log (DVL) is prone to outliers or even failures during measurement under water. B. Li et al. [43] used LightGBM regression for predicting the position of a vehicle during a GNSS outage. Z. Zhi et al. [44] proposed a wavelet threshold denoising method and a CNN–LSTM model which can extract network input features to assist LSTM. S. Zhao et al. [45] combined CNN and a gated recurrent unit (GRU) and used CKF to enhance the navigation accuracy. This is an effective combination of neural network and filtering.

As can be seen from the above analysis, the improved method based on adding multi-source sensor input can reach good results, but the cost and fault rate will increase accordingly. The improved data processing method is relatively efficient in calculation, but it is easy to be limited in engineering application due to the complexity of calculation. In contrast, AI-based approaches, which balance computational burden and effectiveness, are the most popular among scholars by far. However, when it comes to the value of neural network hyperparameters, usually the empirical method or simple grid search method is used, which cannot give full play to the advantages of artificial intelligence, which not only consumes time but also is difficult to achieve the optimal effect. Aiming at the shortcomings of traditional methods, a performance model of integrated navigation system is proposed in this paper. Firstly, the SNR of IMUs is improved by an improved threshold denoising method which can improve the shortcomings of hard or soft threshold denoising. In addition, to enhance the efficiency of the LSTM network in a hyperparameter setting, a PSO-LSTM model is established. In the case of GNSS signal loss, an LSTM network can connect the position information of the past moment with the output of the current moment, predict the GNSS position increment by training network, generate pseudo-GNSS information and then participate in the integrated navigation solution.

The PSO-LSTM model can give full play to the performance of LSTM to improve prediction accuracy and solve the problem of hyperparameter selection. In addition, sufficient training data are used to prevent the model from overfitting. Training time will increase but most training can be conducted offline. In this paper, the effectiveness of this method is verified by an actual road test. The results show that this method improved the accuracy compared with the pure inertial navigation solution.

The contribution of this article could be summarized as follows.

1. A GNSS/INS integrated navigation method is proposed, which can be used in the case of satellite rejection. The method consists of two parts, which are the improved threshold denoising algorithm and the PSO-LSTM model.
2. The effectiveness and superiority of the method have been verified by practical road tests.

The content of each chapter in this paper is described as follows: Section 2 details the GNSS/INS integrated navigation, the LSTM network unit and the PSO algorithm. Section 3 introduces the architecture of the PSO-LSTM assisted integrated navigation system proposed in the paper. Section 4 introduces the simulation test and real road test. Finally, Section 5 summarizes and discusses the prospects of the research focus for future work.

2. Related Work

2.1. Wavelet Threshold Denoising

An improved threshold denoising method is used in the paper to preprocess IMU data. The principle of wavelet threshold filtering can be summarized as follows: through wavelet decomposition, the energy of a useful signal is concentrated on some wavelet coefficients with a large amplitude in the wavelet domain, while the energy of white noise is distributed in the whole wavelet domain, and the amplitude of the wavelet coefficient of the useful signal after decomposition is greater than that of the noise signal. Therefore, appropriate thresholds should be selected at different scales to process the wavelet coefficient. If the wavelet coefficient is less than the threshold, it will be judged as a noise component, set to zero, and the wavelet coefficient larger than the threshold will be retained, so as to realize noise suppression. Finally, the inverse wavelet transform is carried out to complete the wavelet reconstruction process. The process of denoising is shown in Figure 1.

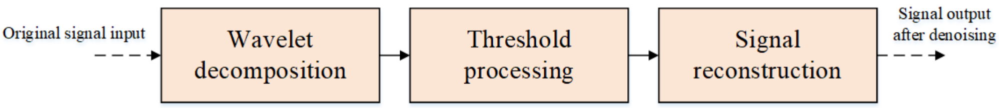


Figure 1. The process of wavelet threshold denoising.

As can be seen from Figure 1, the general steps of the denoising process are as follows. (1) Multi-scale wavelet decomposition of the original noisy signal. (2) According to the signal characteristics, the estimated wavelet threshold values at each scale are calculated, and the wavelet coefficients at each decomposition scale are quantized by threshold. (3) After reconstructing the wavelet coefficients of each layer after threshold processing, the output signal after denoising is obtained.

Among these steps, the most important step is threshold estimation and how to deal with the threshold of wavelet coefficient. The selection of threshold estimation and threshold function is the key to wavelet threshold filtering. The most commonly used threshold functions are hard threshold functions and soft threshold functions.

Hard threshold function:

$$\hat{\omega}_{j,k} = \begin{cases} \omega_{j,k}, \left|\omega_{j,k}\right| > Thr \\ 0, \left|\omega_{j,k}\right| \leq Thr \end{cases} \tag{1}$$

Soft threshold function:

$$\hat{\omega}_{j,k} = \begin{cases} sgn\left(\omega_{j,k}\right) \cdot \left(\left|\omega_{j,k}\right| - Thr\right) & , \left|\omega_{j,k}\right| > Thr \\ 0 & , \left|\omega_{j,k}\right| \leq Thr \end{cases} \tag{2}$$

where $\omega_{j,k}$ represents the wavelet coefficient after wavelet decomposition, and j represents the exponent of the discretized basis and k is the time interval coefficient. $\omega_{j,k}$ is the wavelet coefficient obtained after threshold processing and Thr is the estimated value of the threshold.

Hard and soft threshold function methods both have some inherent shortcomings and deficiencies although they are used widely. Therefore, necessary improvements need to be made. The improved approach is to focus on a core purpose: eliminating the discontinuity of the threshold function while making the function quickly approach the hard threshold function. Based on the investigation, this paper uses a simple and effective improved threshold function, as shown below:

$$\hat{\omega}_{j,k} = \begin{cases} sgn\left(\omega_{j,k}\right) \times \left(\left|\omega_{j,k}\right|^2 - \alpha Thr^2\right)^{\frac{1}{2}}, \left|\omega_{j,k}\right| \geq Thr \\ 0, \left|\omega_{j,k}\right| < Thr \end{cases} \tag{3}$$

where α represents the regulatory factor and is a positive number, which can be adjusted to obtain the appropriate threshold function to make the method more flexible. The diagram of the above threshold function while α = 1 and *Thr* = 1 is shown in Figure 2:

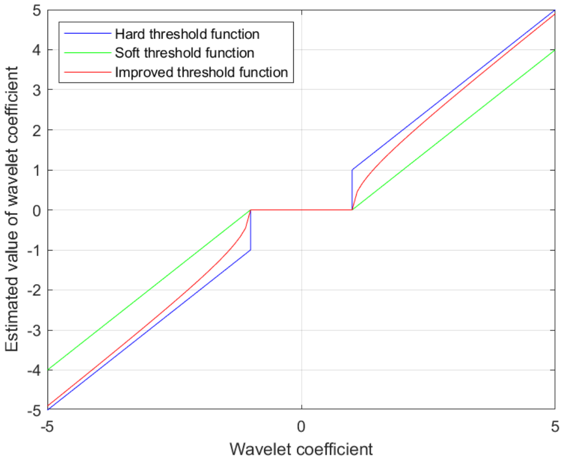


Figure 2. Threshold function curve.

2.2. GNSS/SINS Integrated Navigation Algorithm

INS are generally divided into two types. One is the platform INS, which perform navigation calculations on the gyroscope-stabilized inertial physics platform. Because of its large size and high price, it is only used in some extremely specialized fields. The second is called strapdown INS(SINS) [46–48], in which the three-axis gyroscope and three-axis accelerometer are mounted vertically to achieve angular rate and specific force. The output of IMU is calculated to generate velocity, position and attitude information. In this paper, the inertial navigation system adopts SINS.

According to the different observation information, GNSS/SINS integration can be divided into three categories: loose integration [49,50], tight integration [51,52] and deep integration [53]. Loose integration is a result-level fusion, GNSS positioning results are combined with SINS as observed values. Tight integration belongs to the fusion at the observation level, GNSS pseudo-range, phase and doppler are combined with SINS as observed values. Deep integration belongs to signal level fusion. The position and velocity of SINS recursion assist GNSS signal capture and tracking, and then GNSS provides positioning results or original observation values to integrate with SINS for solving.

Because this paper mainly studies the prediction of pseudo-GNSS information by using a neural network in the case of satellite rejection, compared with the other two combination methods, the loose integration is simple in structure, mature in technology and easy to implement, and its performance is sufficient to verify the prediction method in this paper; so, the loose integration method was chosen. The common 15-dimension state vector is selected as the KF parameter to be estimated. In the loose integrated model, the two systems can calculate the navigation information independently. The position difference and speed difference of the two systems are then fused into the Kalman filter, and then the results of the SINS are minused from the fusion results to realize the error correction. The specific content of the integration method is the same as that in reference [54]. The state vector of the system contains attitude error in three directions, velocity error vector in three directions, error vector of longitude, latitude and height, and the zero bias of the 3-axis gyroscope and accelerometer. The observed value of the filter is the error between the INS position and the GNSS position.

2.3. LSTM Neural Network Algorithm

The LSTM neural network [55] is a deep learning model proposed by Hochreiter S. and Schmidhuber J. in 1997. LSTM uses a memory unit to realize memory function, improves the problem of vanishing gradient in RNN algorithm, and has long-term memory ability for time series data. The internal structure of the unit and the structure of the LSTM network are shown in Figure 3.

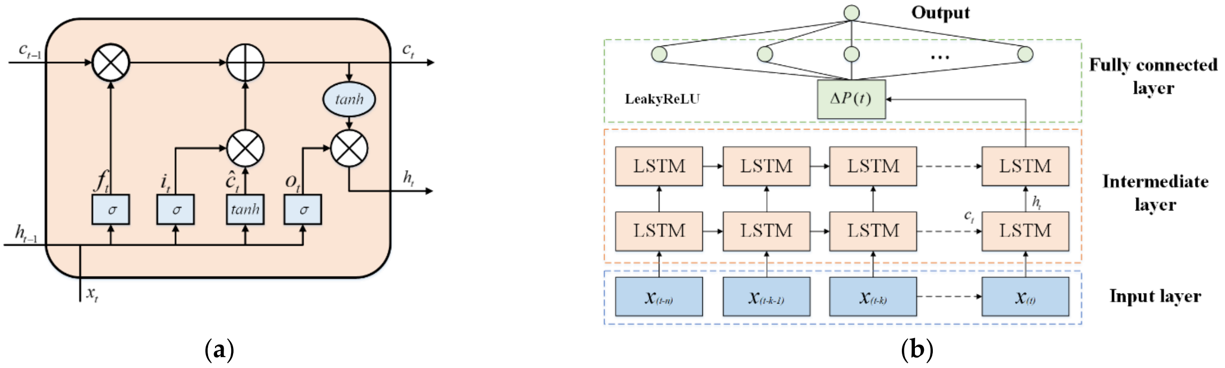


Figure 3. LSTM neural network: (**a**) Internal structure of LSTM unit. (**b**) Structure of the LSTM network.

As shown in Figure 3a, three gate units are forgotten gate (f_t), input gate (i_t) and output gate (o_t), respectively. c_t represents status value of a unit, x_t on behalf of the input vector, h_t is hidden unit state, t is the current moment and $t-1$ is a moment before.

Firstly, calculate f_t output, i_t output and state estimates of the unit $\hat{c}_t$ according to the characteristics vector of the input x_t and h_{t-1} in the $t-1$ moment. New unit state c_t is calculated by f_t, i_t and c_{t-1}, $\hat{c}_t$ weighted separately. Sigmoid (σ) and the tanh are both activation functions. The specific equations are as follows:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{4}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{5}$$

$$\hat{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{6}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t \tag{7}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{9}$$

Multiple LSTM units constitute the LSTM network structure. The structure of the LSTM network is shown in Figure 3b. The structure can be divided into input layer, intermediate layers, and fully connected layers. In fact, if the network is too deep, the network may be overfitting and have a very slow training speed, so only two LSTM layers are selected in this paper. The LSTM output enters the fully connected layer, with the Leaky ReLU function as the activation function. The output after the activation function is used as the final value of the network output, and the difference between it and the training truth value is calculated to test the network performance. The Leaky ReLU activation function is shown as follows:

$$Leaky\ ReLU = \begin{cases} ax, x < 0 \\ x, x \geq 0 \end{cases} \tag{10}$$

The main thought for using the LSTM neural network to assist the GNSS/SINS integrated navigation algorithm is to find the mathematical relationship between navigation information and dynamic data information of the carrier. In recent years, many artificial intelligence-based models have been explored to solve these problems. These models can be divided into three categories by differences in model output: $O_{INS} - \delta P_{INS}$, $O_{INS} - X_K$ and $O_{INS} - \Delta P_{GNSS}$. The first model directly predicts the position error and, the second model predicts the state vector of the KF. However, both of them mix GNSS errors with SINS errors. The third model estimates GNSS position increments at adjacent times and avoids introducing mixing errors from GNSS and SINS effectively.

Next, the input characteristics of the network should be determined. When the GNSS signal is lost, only the inertial navigation system is still working; so, it is necessary to find position-related variables as input feature vectors in the process of SINS position update. As far as vehicle navigation is concerned, the pitch and roll angles are very small. In the end, the method proposed in this paper chooses specific force, angular velocity, the velocity of the INS and yaw angle as the input to the neural network. The relationship between input and output can be expressed as:

$$\Delta P_{GNSS} \propto (f_{ib}^b, \omega_{ib}^b, V_{SINS}, \psi_{SINS}) \tag{11}$$

where ΔP_{GNSS} is position increment, f_{ib}^b is specific force, ω_{ib}^b is angular velocity, V_{SINS} is the velocity of SINS and ψ_{SINS} represents yaw angle of SINS.

For the study of the neural network, the specific architecture of the network, the characteristics of the dataset and the choice of training parameters will affect the convergence of the network. Therefore, in practical applications, it is necessary to evaluate the convergence of the network according to the specific situation, and carry out the necessary adjustment and optimization. It is necessary to use sufficient training data, appropriate network architecture, appropriate loss functions and hyperparameter tuning methods to ensure network convergence. In the experiment of this paper, the construction of our training set takes into account the different motion states and trajectories of vehicle motion, mainly including static, acceleration, deceleration and uniform movement on smooth roads. In terms of track, there are straight lines, left and right turns, U-turns, loops, irregular tracks, etc.

In the design process of the network structure, network hyperparameters need to be selected after actual verification. The number of hidden units and the batch size are two important hyperparameters, and setting them too large will lead to a longer training time to converge the algorithm, and may lead to overfitting problems. Traditionally, cross validation and grid search method are usually used to select hyperparameters. However, these methods rely on experience and have low efficiency, and the final set of network hyperparameters is often not the optimal choice. Therefore, PSO is introduced in this paper to help the LSTM network find the optimal solution of important hyperparameters more efficiently.

2.4. Particle Swarm Optimization Algorithm

Particle swarm optimization [56,57] is a random search algorithm based on group cooperation developed by simulating bird foraging behavior. It can promote each other and generate randomly in different states. Even if the optimized function is not continuous or differentiable, the PSO algorithm can also optimize and converge at a relatively fast speed. The result of any problem in this algorithm can be compared to a bird in the search space, that is, a particle. All alternative solutions in the solution space are determined by all particles, and the matching value of all particles is determined by the optimization function. In addition, they move freely in the full solution space with arbitrary speed, and obtain discovery information by establishing a specific form of information exchange with other particles, so as to guide the movement of the whole group.

The PSO algorithm has the advantages of simple structure, fewer parameters and easy implementation. It is good at efficient optimization in multi-objective parameter optimization to obtain the best parameter results. Therefore, the PSO algorithm is used to optimize the hyperparameters of the LSTM neural network in the paper.

Assuming that $X = [X_1, X_2, \cdots, X_n]$ represents a population of size n and each particle has a search space of d dimensions, then the position and velocity of the ith particle can be expressed by $X = [X_{i1}, X_{i2}, \cdots, X_{id}]$ and $V = [V_{i1}, V_{i2}, \cdots, V_{in}]$. The updated formula of the particle velocity and position is as follows:

$$V_i^{t+1} = \omega V_i^t + c_1 r_{i1}^t (p_{best,i}^t - X_i^t) + c_2 r_{i2}^t (g_{best}^t - X_i^t) \tag{12}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{13}$$

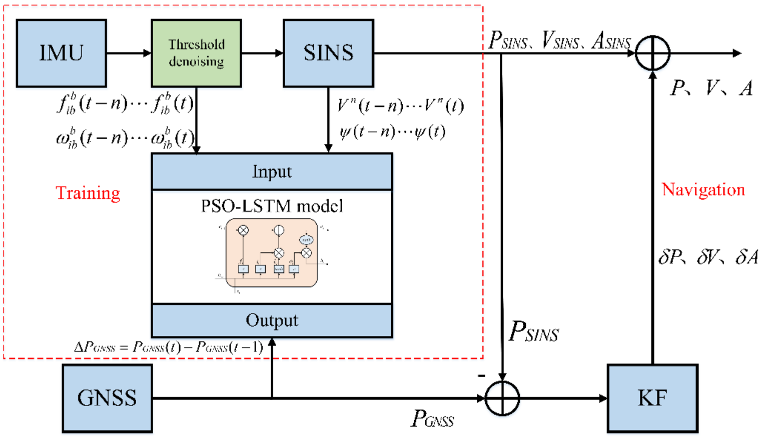
where V_i^{t+1} represents the velocity of the ith particle at iteration $t + 1$, and the initial speed is 0. ω is the inertia weight, which is used to control the global search and optimize locally. c_1 is the cognitive constant used to achieve self-knowledge, and c_2 is the social constant used to simulate interaction with the social population. These two constants usually take the value of 2. r_{i1}^t and r_{i2}^t are random numbers with normal distribution, whose range is [0, 1], used to increase the randomness of particles and avoid falling into local optimum. $p_{best,i}^t$ is the historical optimal position of the ith particle up to the t. g_{best}^t represents the historical optimal position of the group, and x_i^t is the position of the ith particle at iteration t.

3. Proposed Method

3.1. The Architecture of the Method

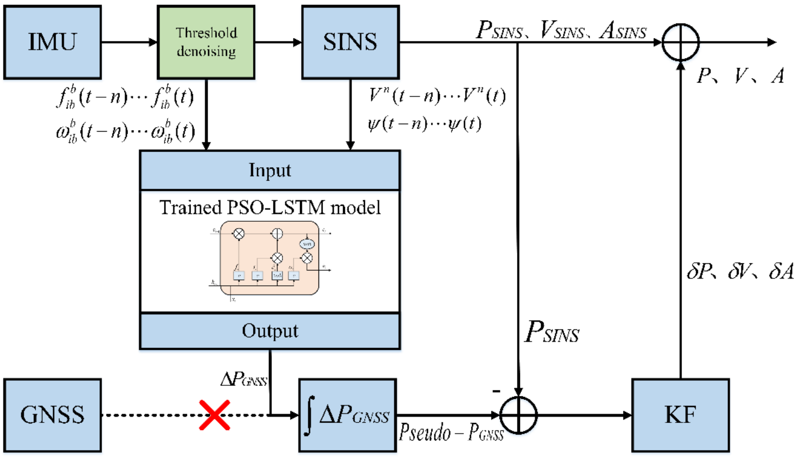
The method proposed in this paper can be divided into two modes: training mode and prediction mode. When the GNSS signal is available, the system works in the training mode. The architecture of the training mode is depicted in Figure 4a.

In training mode, on the one hand, specific force, angular velocity from IMU after denoising and velocity, and yaw angle from SINS form the input to the PSO-LSTM model. On the other hand, the position increment of GNSS becomes the training truth value of the network model. The PSO-LSTM model is responsible for generating loss functions between input and output. Meanwhile, the GNSS signal can enter the loose integrated system to output integrated navigation result normally. As is shown in Figure 4b, when the GNSS signal is interrupted (as shown by red ×), the compensation method converts to prediction mode. The well trained PSO-LSTM model can use its input real-time to predict the corresponding position increment, and cumulate them to obtain the pseudo-GNSS positional information from the neural network. The pseudo-GNSS information is then used to solve the integrated navigation and output the navigation results. In other words, the neural network module plays the role of providing GNSS information to be integrated with SINS when the satellite signal is lost.


IMU
Threshold denoising
SINS
P_{SINS}、V_{SINS}、A_{SINS}
P、V、A
$f_{ib}^b(t-n)\cdots f_{ib}^b(t)$
$\omega_{ib}^b(t-n)\cdots\omega_{ib}^b(t)$
$V^n(t-n)\cdots V^n(t)$
$\psi(t-n)\cdots\psi(t)$
Input
Training
PSO-LSTM model
Navigation
δP、δV、δA
Output
P_{SINS}
$\Delta P_{GNSS} = P_{GNSS}(t) - P_{GNSS}(t-1)$
-
GNSS
KF
P_{GNSS}

(**a**)

Figure 4. *Cont.*


IMU
Threshold denoising
SINS
P_{SINS}、V_{SINS}、A_{SINS}
P、V、A
$f_{ib}^{b}(t-n)\cdots f_{ib}^{b}(t)$
$\omega_{ib}^{b}(t-n)\cdots\omega_{ib}^{b}(t)$
$V^{n}(t-n)\cdots V^{n}(t)$
$\psi(t-n)\cdots\psi(t)$
Input
Trained PSO-LSTM model
Output
δP、δV、δA
P_{SINS}
ΔP_{GNSS}
GNSS
$\int\Delta P_{GNSS}$
$Pseudo-P_{GNSS}$
-
KF

(**b**)

Figure 4. The architecture of the method. (**a**) Training mode; (**b**) Prediction mode.

3.2. The PSO-LSTM Model

The proposed PSO-LSTM model is used to improve the integrated navigation system when the GNSS interrupts. This model makes full use of the advantages of the LSTM neural network, which is good at processing time series, and uses the intelligent optimization method PSO to efficiently obtain the optimal value of important hyperparameters of the network to further improve the prediction accuracy. In this model, the batch size and the number of hidden units of the LSTM neural network are taken as the optimization objects, and the root mean square error of the prediction result is taken as the judgment condition of the prediction effect of the LSTM neural network. PSO keeps searching for optimization according to the judgment conditions until the maximum number of iterations is met. The hyperparameter corresponding to the result with the smallest root mean square error is considered as the optimal hyperparameter.

The flow chart of the model is shown as follows:

As shown in Figure 5, the main training part is shown on the right. After particle swarm initialization, the neural network model starts training and the fitness value is calculated. The particle position is constantly updated until the maximum number of iterations is reached. Then, the optimized parameters are put into the LSTM neural network for GNSS position increment prediction, and the final result of the algorithm is obtained.

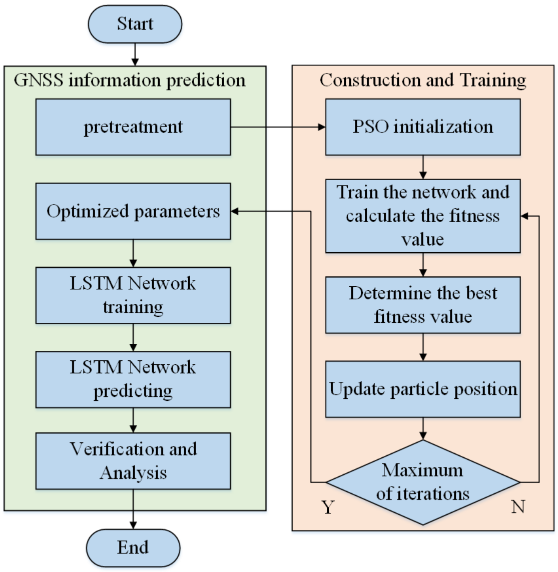

Start
GNSS information prediction
pretreatment
Optimized parameters
LSTM Network training
LSTM Network predicting
Verification and Analysis
End
Construction and Training
PSO initialization
Train the network and calculate the fitness value
Determine the best fitness value
Update particle position
Maximum of iterations
Y
N

Figure 5. Flow chart of PSO-LSTM model.

4. Experiments and Analysis

4.1. Datasets

In the experiment, the algorithm needs to be trained offline by collecting a large amount of road data to build a dataset in advance to preliminarily optimize the LSTM network hyperparameters. The real road test dataset is measured by the unmanned vehicle platform equipped with relevant sensors. Table 1 shows the parameters of each sensor used.

Table 1. Parameters of each sensor.

Sensor Type	Measuring Range	Zero Bias	Random Walk	Precision	Frequency
Gyroscope	±400°/s	0.5°/h	$0.15°/\sqrt{h}$	-	125 Hz
Accelerometer	10 g	0.05 mg	$0.06\ m/s/\sqrt{h}$	-	125 Hz
GNSS receiver	-	-	-	≤0.2 m	100 Hz

The inertial navigation module is STIM300 and GNSS information is from the IPMV vehicle navigation module. The computing platform of the system uses Nvidia TX2 (1.33 TFLOPS), which can meet the algorithm requirements. The installation diagram of the experimental devices on the unmanned vehicle is shown in Figure 6:

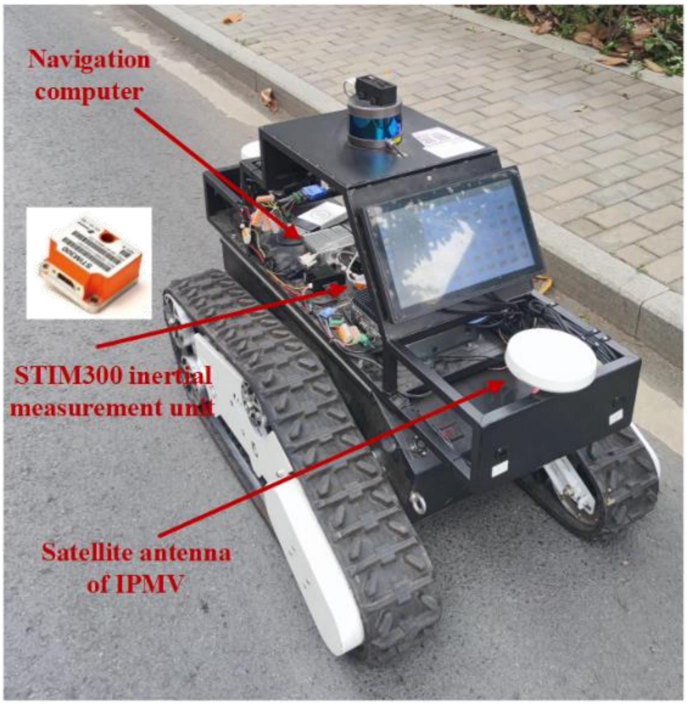


Figure 6. The installation diagram of the experimental devices.

A total of 100 s of static data and 240 s effective dynamic data are collected by the unmanned vehicle to form a real road test dataset. Before forming the test dataset, the original data collected were preprocessed by time synchronization and outlier removal. The vehicle's trajectory in the experiment is shown as follows. The green dot is the starting point, The four tracks selected to test are marked in red and cyan separately. T1, T3 test track takes 60 s and T2, T4 test track takes 30 s. Each test track contains different trajectories such as turns and straight lines. The trajectory is shown in Figure 7:

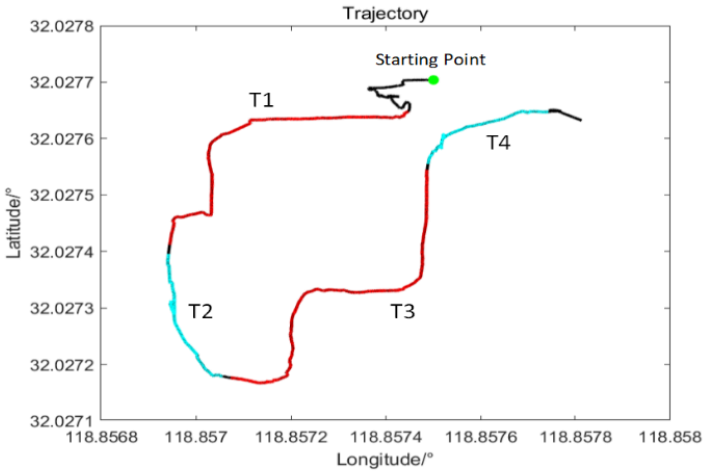


Figure 7. The vehicle's trajectory in the experiment.

4.2. Implementation Details

Firstly, the performance of the hard threshold, soft threshold and the improved threshold denoising methods are verified and compared based on the gyroscope static and dynamic data. Then the four methods, Pure INS, RBF, LSTM and PSO-LSTM, are compared based on dynamic test data. The results calculated by various methods are compared with the output of a high precision dual antenna GNSS receiver. The prediction targets of several neural networks are all location increment information. The reason why we use the RBF comparison is that the RBF neural network is an efficient feedforward neural network. The biggest difference between the RNN neural network studied in this paper and the common feedforward neural network is that the time series is added. We compare RBF with RNN to study the performance of two kinds of networks for satellite information prediction.

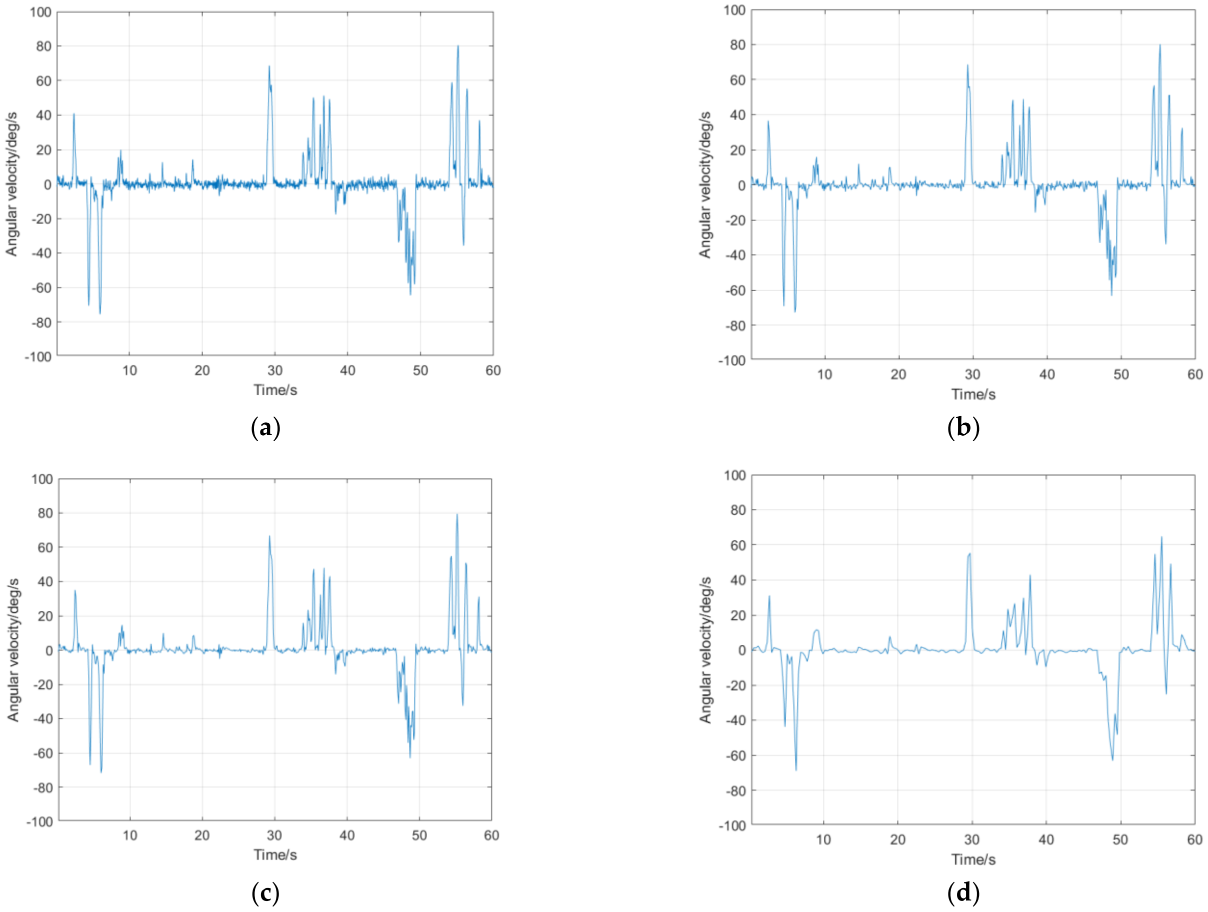
When setting the parameters of the PSO-LSTM model, the grid search method was used for preprocessing to reduce the number of hidden layer elements and the value range of batch size to 30–80, so as to improve the experimental efficiency. The PSO-LSTM model parameters are shown as Table 2:

Table 2. Parameters of PSO-LSTM model.

Parameters	Value
Number of LSTM layer	2
Initial learning rate	0.005
Learning rate drop factor	0.2
Time step	8
Epochs	100
Gradient threshold	1
Number of hidden layer units	30–80
Batch size	30–80
The population size of PSO	10
Iteration number of PSO	15
Learning factor c_1	2
Learning factor c_2	2
Inertia weight	0.5

4.3. Results and Analysis

The results of three threshold denoising methods based on the gyroscope dynamic data in T1 are shown in Figure 8.



(**a**) (**b**) (**c**) (**d**)

Figure 8. Results of gyroscope dynamic data in T1: (**a**) Original data. (**b**) Denoising result of hard threshold function. (**c**) Denoising result of soft threshold function. (**d**) Denoising result of the improved threshold function.

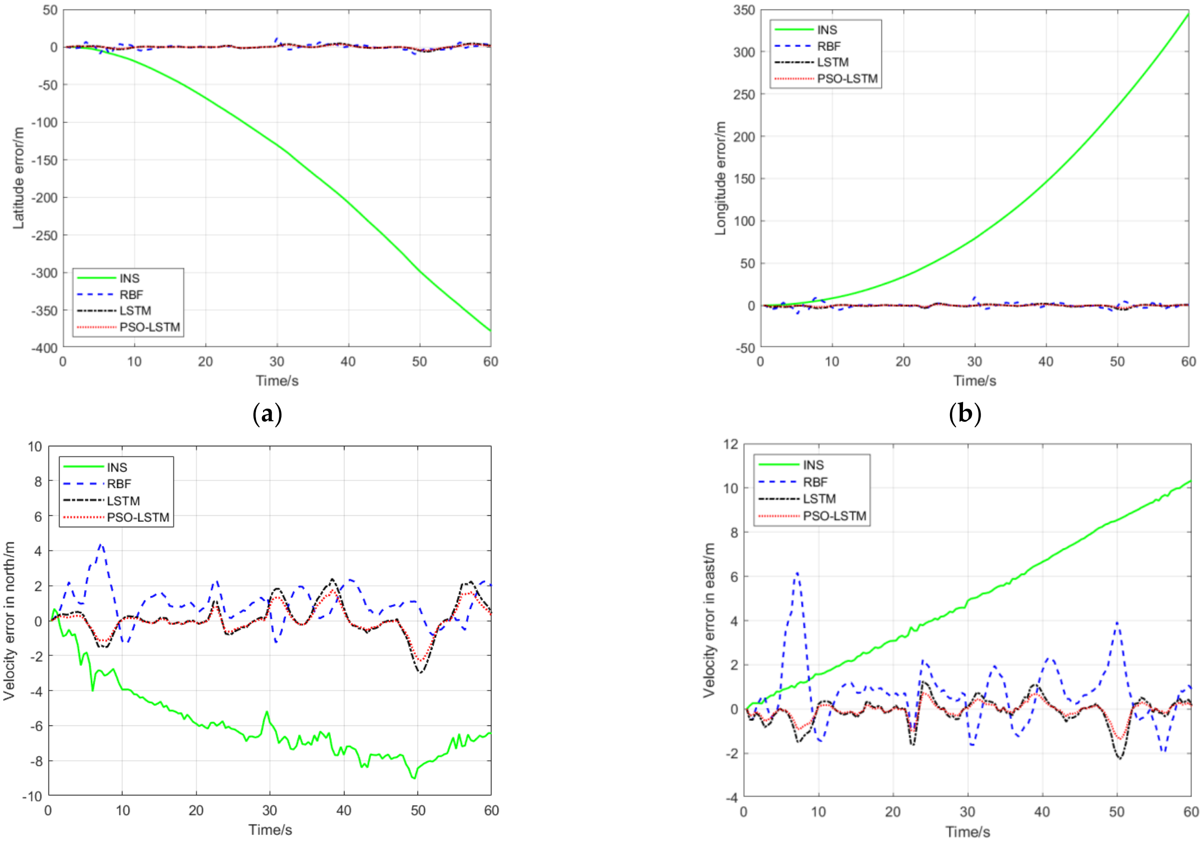
The results of three threshold denoising methods based on 30 s static data and T1 to 4 section of yaw angle are shown in Table 3.

Table 3. Results of gyroscope data denoising experiment.

Data		Hard Threshold	Soft Threshold	Improved Threshold
Static data	SNR/dB RMSE/deg/s	1.947 0.056	4.188 0.043	8.450 0.026
T1	SNR/dB RMSE/deg/s	33.277 0.330	35.994 0.241	39.442 0.162
T2	SNR/dB RMSE/deg/s	28.587 0.361	30.976 0.274	35.249 0.167
T3	SNR/dB RMSE/deg/s	31.411 0.348	33.851 0.263	37.869 0.165
T4	SNR/dB RMSE/deg/s	26.717 0.403	31.647 0.229	35.797 0.142

As can be seen from Table 3, for static data, the proposed method has a very obvious denoising effect compared with the other two methods. The SNR is increased to 8.450 dB and the RMSE (root mean square error) is reduced to 0.026 deg/s. In dynamic data experiments, the improved threshold denoising method can effectively improve the SNR, respectively increasing by 18.53%, 23.30%, 20.56% and 33.99% compared with the hard threshold denoising method and increasing by 9.58%, 13.79%, 11.87% and 13.11% compared with the soft threshold denoising method. In addition, compared with soft threshold and hard threshold functions, the improved method can increase the SNR by 33.99% and reduce the RMSE by 64.76%.

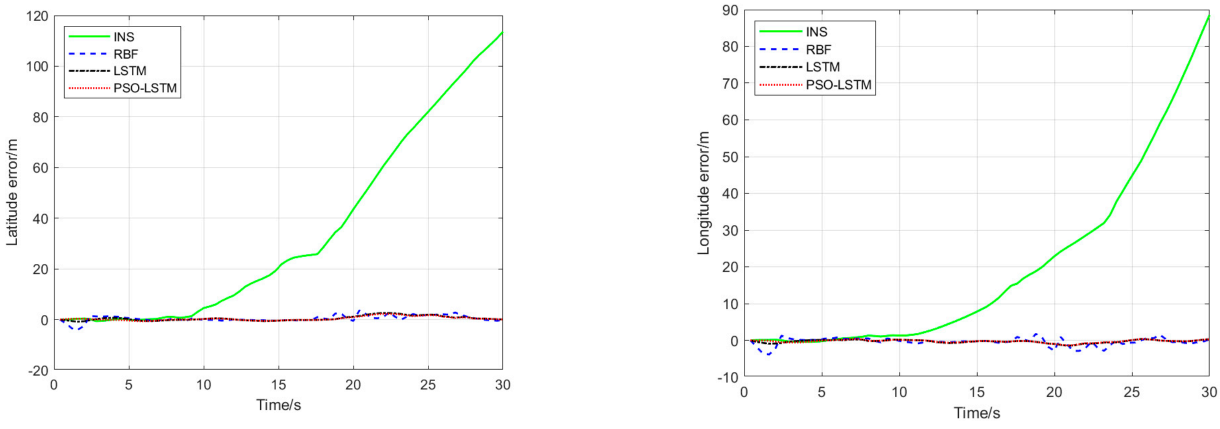
Next, the results of various navigation methods above are shown as Figures 9–12. The statistics of the results are shown in Tables 4–7.



(**a**) (**b**)

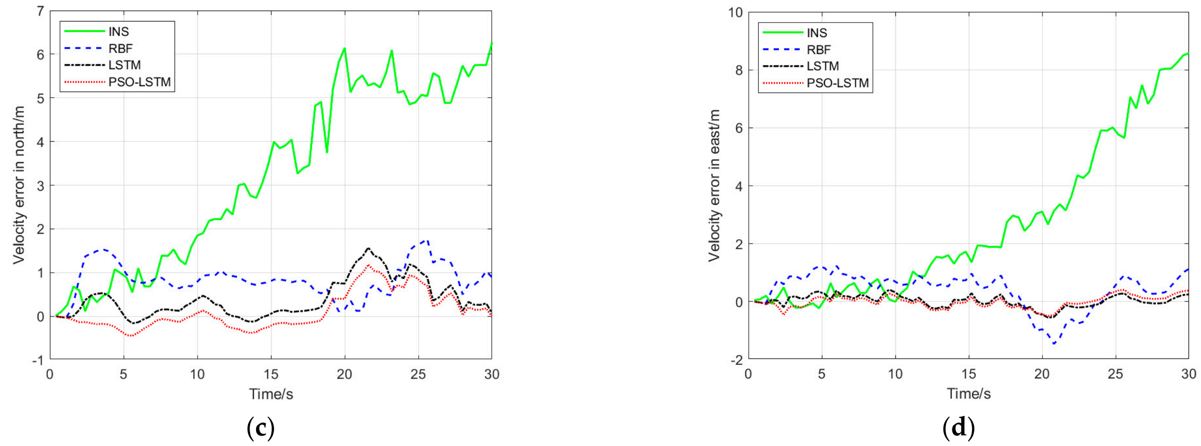
(**c**) (**d**)

Figure 9. The results of T1: (**a**) Error in latitude. (**b**) Error in longitude. (**c**) Error in northbound velocity. (**d**) Error in easterly velocity.



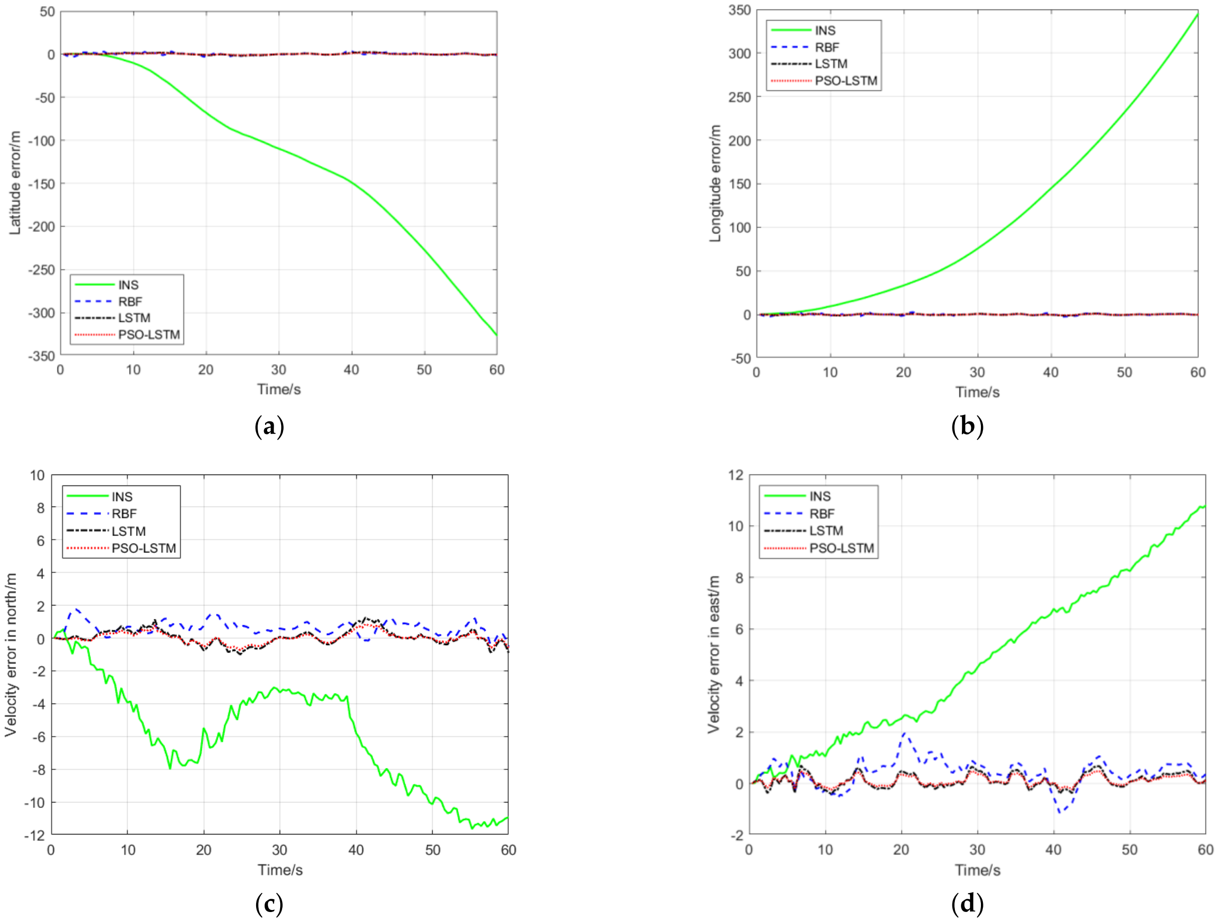
(**a**) (**b**)

Figure 10. *Cont.*



(**c**) (**d**)

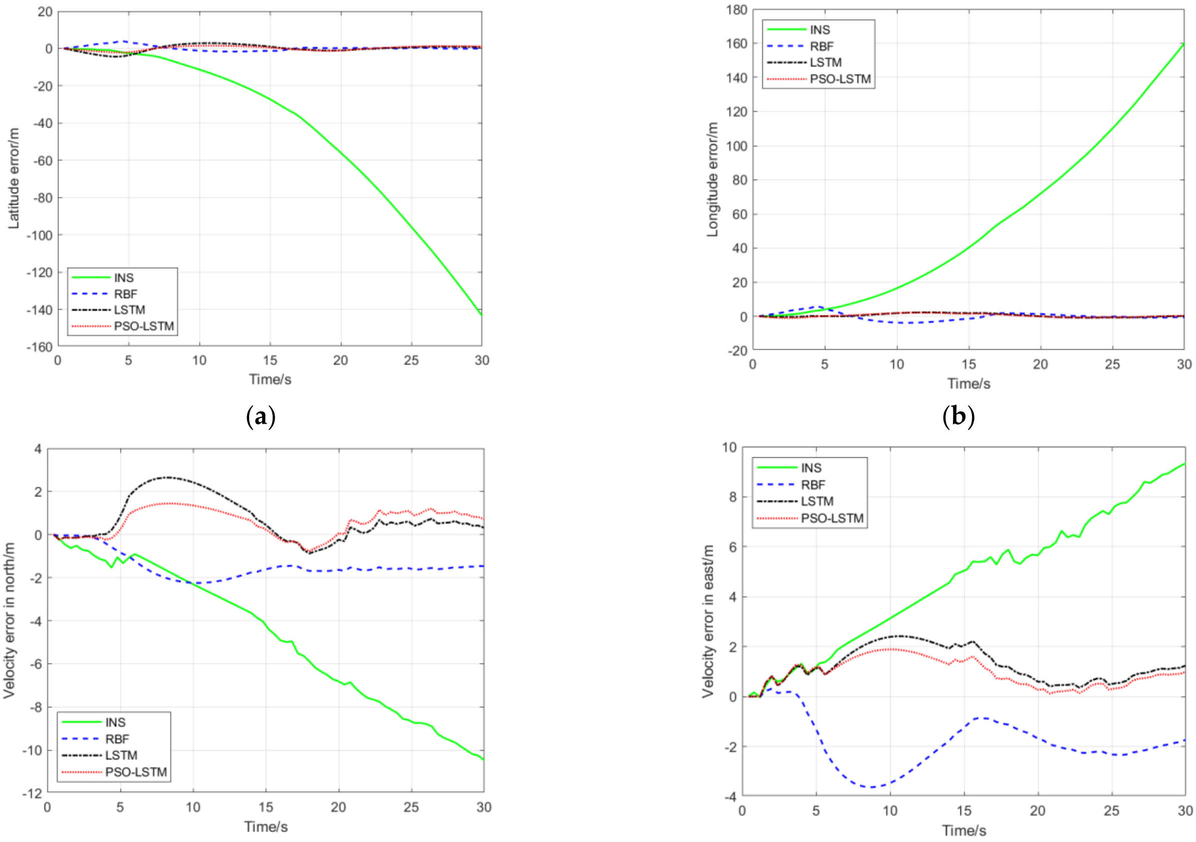
Figure 10. The results of T2: (**a**) Error in latitude. (**b**) Error in longitude. (**c**) Error in northbound velocity. (**d**) Error in easterly velocity.



(**a**) (**b**)

(**c**) (**d**)

Figure 11. The results of T3: (**a**) Error in latitude. (**b**) Error in longitude. (**c**) Error in northbound velocity. (**d**) Error in easterly velocity.



(**a**) (**b**)

(**c**) (**d**)

Figure 12. The results of T4: (**a**) Error in latitude. (**b**) Error in longitude. (**c**) Error in northbound velocity. (**d**) Error in easterly velocity.

Table 4. Results statistics of period T1.

	SINS		RBF		LSTM		PSO-LSTM	
	MAX	RMSE	MAX	RMSE	MAX	RMSE	MAX	RMSE
Error in latitude (m)	380.928	193.989	12.008	3.191	6.178	2.143	4.634	1.606
Error in longitude (m)	350.196	153.735	12.495	2.942	5.232	1.429	3.141	0.855
Error in northbound velocity (m/s)	9.302	6.216	4.491	1.395	3.056	1.010	2.339	0.753
Error in easterly velocity (m/s)	10.421	5.866	6.400	1.589	2.394	0.629	1.441	0.383

Table 5. Results statistics of period T2.

	SINS		RBF		LSTM		PSO-LSTM	
	MAX	RMSE	MAX	RMSE	MAX	RMSE	MAX	RMSE
Error in latitude (m)	116.352	51.647	4.117	1.364	2.585	1.003	2.253	0.870
Error in longitude (m)	91.903	32.527	4.210	1.190	1.448	0.516	1.408	0.475
Error in northbound velocity (m/s)	6.769	3.839	1.892	0.919	1.576	0.571	1.185	0.425
Error in easterly velocity (m/s)	8.884	3.791	1.473	0.748	0.677	0.207	0.610	0.210

Table 6. Results statistics of period T3.

	SINS		RBF		LSTM		PSO-LSTM	
	MAX	RMSE	MAX	RMSE	MAX	RMSE	MAX	RMSE
Error in latitude (m)	330.228	154.648	4.431	1.300	2.333	0.961	1.623	0.675
Error in longitude (m)	350.452	152.096	2.974	0.935	1.236	0.555	0.828	0.373
Error in northbound velocity (m/s)	12.461	6.786	1.815	0.755	1.462	0.473	1.009	0.332
Error in easterly velocity (m/s)	10.934	5.788	2.030	0.697	0.790	0.297	0.555	0.213

Table 7. Results statistics of period T4.

	SINS		**RBF**		**LSTM**		**PSO-LSTM**	
	MAX	**RMSE**	**MAX**	**RMSE**	**MAX**	**RMSE**	**MAX**	**RMSE**
Error in latitude (m)	147.802	63.113	4.071	1.336	4.427	1.858	2.193	1.109
Error in longitude (m)	164.613	73.947	5.921	2.287	2.257	1.120	2.076	1.051
Error in northbound velocity (m/s)	10.623	5.843	3.935	2.769	2.643	1.232	1.439	0.855
Error in easterly velocity (m/s)	9.490	5.425	4.541	2.656	2.416	1.424	1.889	1.010

The performance of various algorithms can be seen from the above results charts, and the specific analysis can be summarized as follows:

Inertial navigation errors gradually diverge with time and increase. Compared with pure INS, the three neural network methods have improved navigation accuracy to some extent. The GNSS position information obtained by network prediction can effectively suppress the error divergence of the inertial navigation solution.

Among RBF, LSTM and PSO-LSTM neural network methods, the method proposed in this paper has the best effect on reducing the maximum error. In the experiment of period T1, which takes 60 s, compared with the pure inertial solution, the proportion of the maximum error reduced by the three methods is, respectively, 96.85%, 98.38% and 98.78% in the latitude direction, 96.43%,98.51% and 99.10% in the longitude direction, 51.72%, 67.15% and 74.85% in north velocity and 38.59%, 77.03% and 86.17% in east velocity. In T3, the proportion of the maximum error reduced by the three methods is, respectively, 98.66%, 99.29% and 99.51% in latitude, 99.15%,99.65% and 99.76% in longitude, 85.43%, 88.27% and 91.90% in north velocity and 81.43%, 92.77% and 94.92% in east velocity. Regarding the 30 s experiments, in T2, the proportion of the maximum error reduced by the three methods is, respectively, 96.46%, 97.78% and 98.06% in the latitude direction, 95.42%,98.42% and 98.47% in the longitude direction, 72.05%,76.72% and 82.49% in north velocity and 83.42%, 92.38% and 93.13% in east velocity. In T4, the proportion of the maximum error reduced by the three methods is, respectively, 97.25%, 97.00% and 98.52% in the latitude direction, 96.40%,98.52% and 98.74% in the longitude direction, 62.96%, 75.12% and 86.45% in north velocity and 52.15%, 74.54% and 80.09% in east velocity. Overall analysis of the maximum error and RMSE in the three groups of experimental results shows that the performance of the LSTM method is better than that of the RBF method and the PSO-LSTM is the best for this experimental dataset.

Compared with the simple LSTM neural network, the PSO algorithm can effectively obtain the optimal important parameters of the LSTM network, and improve the accuracy of the original LSTM neural network method which saves time and improves efficiency. The design of the PSO-LSTM method structure is useful.

To sum up, the method described in this paper is useful for the improvement of the positioning and velocity measurement accuracy of the inertial navigation system with satellite blocking.

5. Conclusions

Aiming at the problem that the accuracy of traditional GNSS/INS integrated navigation degrades to INS when satellite signals are lost or interfered with, in this paper, the PSO -LSTM neural network model is established to assist the integrated system. The network model can predict the GNSS position increment under the condition of satellite blocking and obtain the pseudo-GNSS signal by accumulation. The pseudo-GNSS signal is used to compensate the observed value in the integrated system. By combining the advantages of LSTM and PSO, the model can quickly and accurately obtain the optimal combination of the number of hidden units and the batch size parameters of the neural network, and obtain a good effect of GNSS location increment prediction. At the same time, the improved threshold function is used to denoise the IMU data, which improves the SNR effectively. Finally, the performance of the algorithm is verified by designing a vehicle navigation experiment. Compared with other methods, this method can effectively improve the accuracy in the case of satellite blocking. When the satellite is lost for 30 s, the maximum error of latitude and longitude can be reduced by at least 98.06% and 98.47%, respectively, compared with INS. When the satellite is lost for 60 s, the maximum error of latitude and longitude can be reduced by at least 98.78% and 99.10%, respectively, compared with INS. The experimental results show that this method has the significance of improving navigation accuracy in practice.

There are still some limitations in this paper. Further research can focus on improving the filtering algorithm, adapting to the complex electromagnetic environment and increasing the training dataset to improve the prediction effect and navigation accuracy.

Author Contributions: All authors contributed to this manuscript. Data curation, K.J.; methodology, Y.C.; experimental results analysis, Y.C. and G.Z.; writing original draft, Y.C.; writing review and editing, H.B. and K.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC), grant number U2031138; Defense Science and Technology 173 Program Field Foundation, 2022-JCJQ-JJ-0395.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: The authors would like to thank the anonymous reviewers for the constructive suggestions.

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

References

1. Li, X.; Guo, R.; Chen, J.; Liu, S.; Chang, Z.; Xin, J.; Guo, J.; Tian, Y. New Orbit Determination Method for GEO Satellites Based on BeiDou Short-Message Communication Ranging. *Remote Sens.* **2022**, *14*, 4602. [CrossRef]
2. Alajami, A.A.; Moreno, G.; Pous, R. Design of a UAV for Autonomous RFID-Based Dynamic Inventories Using Stigmergy for Mapless Indoor Environments. *Drones* **2022**, *6*, 208. [CrossRef]
3. Cao, Y.; Chen, J.; Liu, L.; Hu, X.; Liu, Y.; Xin, J.; Zhao, L.; Tian, Q.; Zhou, S.; Wu, B. Development Status and Service Performance Preliminary Analysis for BDSBAS. *Remote Sens.* **2022**, *14*, 4314. [CrossRef]
4. Makar, A. Determination of USV's Direction Using Satellite and Fluxgate Compasses and GNSS-RTK. *Sensors* **2022**, *22*, 7895. [CrossRef] [PubMed]
5. Zhao, L.; Yang, Y.; Xiang, Z.; Zhang, S.; Li, X.; Wang, X.; Ma, X.; Hu, C.; Pan, J.; Zhou, Y.; et al. A Novel Low-Cost GNSS Solution for the Real-Time Deformation Monitoring of Cable Saddle Pushing: A Case Study of Guojiatuo Suspension Bridge. *Remote Sens.* **2022**, *14*, 5174. [CrossRef]
6. Chen, B.; Zheng, Y.; Chen, Z.; Zhang, H.; Liu, X. A review of celestial navigation system on near space hypersonic vehicle. *Acta Aeronaut. Astronaut. Sin.* **2020**, *41*, 623686.
7. Tian, M.; Liang, Z.; Liao, Z.; Yu, R.; Guo, H.; Wang, L. A Polar Robust Kalman Filter Algorithm for DVL-Aided SINSs Based on the Ellipsoidal Earth Model. *Sensors* **2022**, *22*, 7879. [CrossRef]
8. Wei, Q.; Zha, F.; He, H.; Li, B. An Improved System-Level Calibration Scheme for Rotational Inertial Navigation Systems. *Sensors* **2022**, *22*, 7610. [CrossRef]
9. Li, Y.; Yang, S.; Xiu, X.; Miao, Z. A Spatiotemporal Calibration Algorithm for IMU–LiDAR Navigation System Based on Similarity of Motion Trajectories. *Sensors* **2022**, *22*, 7637. [CrossRef]
10. Sun, W.; Sun, P.; Wu, J. An Adaptive Fusion Attitude and Heading Measurement Method of MEMS/GNSS Based on Covariance Matching. *Micromachines* **2022**, *13*, 1787. [CrossRef]
11. Ng, H.-F.; Hsu, L.-T.; Lee, M.J.L.; Feng, J.; Naeimi, T.; Beheshti, M.; Rizzo, J.-R. Real-Time Loosely Coupled 3DMA GNSS/Doppler Measurements Integration Using a Graph Optimization and Its Performance Assessments in Urban Canyons of New York. *Sensors* **2022**, *22*, 6533. [CrossRef] [PubMed]
12. Kim, E.; Kim, S.-K. Global Navigation Satellite System Real-Time Kinematic Positioning Framework for Precise Operation of a Swarm of Moving Vehicles. *Sensors* **2022**, *22*, 7939. [CrossRef]
13. Li, Z.; Deng, Y.; Liu, W. Identification of INS Sensor Errors from Navigation Data Based on Improved Pigeon-Inspired Optimization. *Drones* **2022**, *6*, 287. [CrossRef]
14. Wang, H.; Pan, S.; Gao, W.; Xia, Y.; Ma, C. Multipath/NLOS Detection Based on K-Means Clustering for GNSS/INS Tightly Coupled System in Urban Areas. *Micromachines* **2022**, *13*, 1128. [CrossRef]
15. Chen, B.; Zhao, H.; Zhu, R.; Hu, Y. Marked-LIEO: Visual Marker-Aided LiDAR/IMU/Encoder Integrated Odometry. *Sensors* **2022**, *22*, 4749. [CrossRef]
16. Antonopoulos, A.; Lagoudakis, M.G.; Partsinevelos, P. A ROS Multi-Tier UAV Localization Module Based on GNSS, Inertial and Visual-Depth Data. *Drones* **2022**, *6*, 135. [CrossRef]

17. Zhang, F.; Wu, X.; Ma, P. Consistent Extended Kalman Filter-Based Cooperative Localization of Multiple Autonomous Underwater Vehicles. *Sensors* **2022**, *22*, 4563. [CrossRef]
18. Zhu, J.; Li, A.; Qin, F.; Che, H.; Wang, J. A Novel Hybrid Method Based on Deep Learning for an Integrated Navigation System during DVL Signal Failure. *Electronics* **2022**, *11*, 2980. [CrossRef]
19. Li, Z.; Zhang, Y.; Shi, Y.; Yuan, S.; Zhu, S. Performance Enhancement of INS and UWB Fusion Positioning Method Based on Two-Level Error Model. *Sensors* **2023**, *23*, 557. [CrossRef] [PubMed]
20. Fan, Q.; Zhang, H.; Pan, P.; Zhuang, X.; Jia, J.; Zhang, P.; Zhao, Z.; Zhu, G.; Tang, Y. Improved Pedestrian Dead Reckoning Based on a Robust Adaptive Kalman Filter for Indoor Inertial Location System. *Sensors* **2019**, *19*, 294. [CrossRef]
21. Liu, X.; Guo, X.; Zhao, D.; Shen, C.; Wang, C.; Li, J.; Tang, J.; Liu, J. INS/Vision Integrated Navigation System Based on a Navigation Cell Model of the Hippocampus. *Appl. Sci.* **2019**, *9*, 234. [CrossRef]
22. Li, S.; Li, Z.; Liu, X.; Shan, C.; Zhao, Y.; Cheng, H. Research on Map-SLAM Fusion Localization Algorithm for Unmanned Vehicle. *Appl. Sci.* **2022**, *12*, 8670. [CrossRef]
23. Vetrella, A.R.; Fasano, G.; Accardo, D.; Moccia, A. Differential GNSS and Vision-Based Tracking to Improve Navigation Performance in Cooperative Multi-UAV Systems. *Sensors* **2016**, *16*, 2164. [CrossRef] [PubMed]
24. Nam, D.V.; Gon-Woo, K. Robust Stereo Visual Inertial Navigation System Based on Multi-Stage Outlier Removal in Dynamic Environments. *Sensors* **2020**, *20*, 2922. [CrossRef]
25. Qiu, Z.; Lin, D.; Jin, R.; Lv, J.; Zheng, Z. A Global ArUco-Based Lidar Navigation System for UAV Navigation in GNSS-Denied Environments. *Aerospace* **2022**, *9*, 456. [CrossRef]
26. Hensel, S.; Marinov, M.B.; Obert, M. 3D LiDAR Based SLAM System Evaluation with Low-Cost Real-Time Kinematics GPS Solution. *Computation* **2022**, *10*, 154. [CrossRef]
27. Duong, H.T.; Suh, Y.S. A Human Gait Tracking System Using Dual Foot-Mounted IMU and Multiple 2D LiDARs. *Sensors* **2022**, *22*, 6368. [CrossRef]
28. Liu, W.; Quijano, K.; Crawford, M.M. YOLOv5-Tassel: Detecting Tassels in RGB UAV Imagery with Improved YOLOv5 Based on Transfer Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 8085–8094. [CrossRef]
29. Xia, X.; Meng, Z.; Han, X.; Li, H.; Tsukiji, T.; Xu, R.; Zheng, Z.; Ma, J. An Automated Driving Systems Data Acquisition and Analytics Platform. *Transp. Res. Part C Emerg. Technol.* **2023**, *151*, 104120. [CrossRef]
30. Zhang, J.; Wang, S.; Li, W.; Qiu, Z. A Multi-Mode Switching Variational Bayesian Adaptive Kalman Filter Algorithm for the SINS/PNS/GMNS Navigation System of Pelagic Ships. *Sensors* **2022**, *22*, 3372. [CrossRef]
31. Cao, S.; Gao, H.; You, J. In-Flight Alignment of Integrated SINS/GPS/Polarization/Geomagnetic Navigation System Based on Federal UKF. *Sensors* **2022**, *22*, 5985. [CrossRef]
32. Chang, L.; Li, Y.; Xue, B. Initial alignment for a doppler velocity log-aided strapdown inertial navigation system with limited information. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 329–338. [CrossRef]
33. Silvestrini, S.; Piccinin, M.; Zanotti, G.; Brandonisio, A.; Lunghi, P.; Lavagna, M. Implicit Extended Kalman Filter for Optical Terrain Relative Navigation Using Delayed Measurements. *Aerospace* **2022**, *9*, 503. [CrossRef]
34. Zhu, K.; Yu, Y.; Wu, B.; Jiang, C. GO-INO: Graph Optimization MEMS-IMU/NHC/Odometer Integration for Ground Vehicle Positioning. *Micromachines* **2022**, *13*, 1400. [CrossRef] [PubMed]
35. Xia, X.; Xiong, L.; Huang, Y.; Lu, Y.; Gao, L.; Xu, N.; Yu, Z. Estimation on IMU yaw misalignment by fusing information of automotive onboard sensors. *Mech. Syst. Signal Process.* **2022**, *162*, 107993. [CrossRef]
36. Liu, W.; Xia, X.; Xiong, L.; Lu, Y.; Gao, L.; Yu, Z. Automated Vehicle Sideslip Angle Estimation Considering Signal Measurement Characteristic. *IEEE Sens. J.* **2021**, *21*, 21675–21687. [CrossRef]
37. Xia, X.; Hashemi, E.; Xiong, L.; Khajepour, A. Autonomous Vehicle Kinematics and Dynamics Synthesis for Sideslip Angle Estimation Based on Consensus Kalman Filter. *IEEE Trans. Control Syst. Technol.* **2023**, *31*, 179–192. [CrossRef]
38. Kalman, R.E.; Bucy, R.S. New results in linear prediction filtering theory. *J. Basic Eng.* **1961**, *83*, 95–108. [CrossRef]
39. Carlson, N.A. Federated filter for fault-tolerant integrated navigation systems. In Proceedings of the IEEE PLANS88, Orlando, FL, USA, 29 November–2 December 1988.
40. Zhao, Y.; Yan, G.; Qin, Y.; Fu, Q. Information Fusion Based on Complementary Filter for SINS/CNS/GPS Integrated Navigation System of Aerospace Plane. *Sensors* **2020**, *20*, 7193. [CrossRef]
41. Sun, B.; Zhang, Z.; Qiao, D.; Mu, X.; Hu, X. An Improved Innovation Adaptive Kalman Filter for Integrated INS/GPS Navigation. *Sustainability* **2022**, *14*, 11230. [CrossRef]
42. Jiang, C.; Chen, S.; Chen, Y.; Zhang, B.; Feng, Z.; Zhou, H.; Bo, Y. A MEMS IMU De-Noising Method Using Long Short Term Memory Recurrent Neural Networks (LSTM-RNN). *Sensors* **2018**, *18*, 3470. [CrossRef]
43. Li, B.; Chen, G.; Si, Y.; Zhou, X.; Li, P.; Li, P.; Fadiji, T. GNSS/INS Integration Based on Machine Learning LightGBM Model for Vehicle Navigation. *Appl. Sci.* **2022**, *12*, 5565. [CrossRef]
44. Zhi, Z.; Liu, D.; Liu, L. A performance compensation method for GPS/INS integrated navigation system based on CNN–LSTM during GPS outages. *Measurement* **2022**, *188*, 110516. [CrossRef]
45. Zhao, S.; Zhou, Y.; Huang, T. A Novel Method for AI-Assisted INS/GNSS Navigation System Based on CNN-GRU and CKF during GNSS Outage. *Remote Sens.* **2022**, *14*, 4494. [CrossRef]
46. Li, J.; Jing, Z.; Zhang, X.; Zhang, J.; Li, J.; Gao, S.; Zheng, T. Optimization Design Method of a New Stabilized Platform Based on Missile-borne Semi-Strap-down Inertial Navigation System. *Sensors* **2018**, *18*, 4412. [CrossRef] [PubMed]

47. Li, Z.-T.; Wu, T.-J.; Lin, C.-L.; Ma, L.-H. Field Programmable Gate Array Based Parallel Strapdown Algorithm Design for Strapdown Inertial Navigation Systems. *Sensors* **2011**, *11*, 7993–8017. [CrossRef]
48. Yang, B.; Xi, J.; Yang, J.; Xue, L. An Alignment Method for Strapdown Inertial Navigation Systems Assisted by Doppler Radar on a Vehicle-Borne Moving Base. *Sensors* **2019**, *19*, 4577. [CrossRef]
49. Xu, X.; Nie, Z.; Wang, Z.; Wang, B.; Du, Q. Performance Assessment of BDS-3 PPP-B2b/INS Loosely Coupled Integration. *Remote Sens.* **2022**, *14*, 2957. [CrossRef]
50. Di Pietra, V.; Dabove, P.; Piras, M. Loosely Coupled GNSS and UWB with INS Integration for Indoor/Outdoor Pedestrian Navigation. *Sensors* **2020**, *20*, 6292. [CrossRef]
51. Falco, G.; Pini, M.; Marucco, G. Loose and Tight GNSS/INS Integrations: Comparison of Performance Assessed in Real Urban Scenarios. *Sensors* **2017**, *17*, 255. [CrossRef]
52. Lv, J.; Gao, Z.; Xu, Q.; Lan, R.; Yang, C.; Peng, J. Assessment of Real-Time GPS/BDS-2/BDS-3 Single-Frequency PPP and INS Tight Integration Using Different RTS Products. *Remote Sens.* **2022**, *14*, 4367. [CrossRef]
53. Ban, Y.; Niu, X.; Zhang, T.; Zhang, Q.; Liu, J. Modeling and Quantitative Analysis of GNSS/INS Deep Integration Tracking Loops in High Dynamics. *Micromachines* **2017**, *8*, 272. [CrossRef] [PubMed]
54. Cao, Y.; Bai, H.; Liang, H.; Zou, G. An Integrated Navigation Algorithm Based on LSTM Neural Network. In Proceedings of the International Conference on Guidance, Navigation and Control, Harbin, China, 5–7 August 2022.
55. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
56. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995.
57. Eberhart, R.C.; Shi, Y. Particle swarm optimization: Developments, applications and resources. In Proceedings of the IEEE Conference on Evolutionary Computation, Soul, Republic of Korea, 27–30 May 2001.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.