CNN-GRU-Assisted Factor Graph Integrated Navigation System

Ru Pang, Xishuo Wang

Hubei University, Wuhan 430000, China.

Abstract

To address the degradation of positioning accuracy due to GNSS signal loss and INS error accumulation in denied environments, this paper proposesan integrated navigation algorithm that combines neural network-assisted Factor Graph Optimization (FGO) with an Extended Kalman Filter (EKF).This approach enhances navigation accuracy and real-time performance in complex scenarios by fusing multi-source sensor data and applying nonlinear optimization. A Convolutional Neural Network (CNN) is designed to extract local spatio-temporal features from IMU acceleration and angular velocity data. Integrated with a Gated Recurrent Unit (GRU) to model long-term temporal dependencies, the network generates pseudo-GNSS signals to compensate for positioning errors during outages. The EKF is then employed to preprocess IMU/GNSS data, suppressing sensor noise and refining the initial state estimate. Subsequently, a sliding window-constrained factor graph optimization model performs global nonlinear fusion of the processed IMU and GNSS factors, balancing dynamic error correction with computational efficiency. Simulation results demonstrate that the proposed algorithm achieves lower maximum northward and eastward position errors during GNSS interruptions compared to traditional EKF-LSTM and GRU-onlymethods. Furthermore, the sliding window strategy reduces data processingtime by 57.0% relative to conventional FGO. By leveraging multi-modal data fusion and a dynamic optimization mechanism, the method significantlyimproves the robustness of integrated navigation systems in challenging environments, offering a valuable technical reference for high-precision autonomous navigation.

Keywords

Integrated Navigation, Factor Graph Optimization, Neural Network, Sliding Window OptimizationDeep Learning.

1. Introduction

The Inertial Navigation System (INS) offers advantages such as high autonomy, strong anti-interference capabilities, and high output frequency, making it widely used in various navigation fields. However, due to its fundamental principle of using dead reckoning technology for solution, it suffers from the problem of accumulating errors that continuously increase, failing to meet the precision requirements for long-term navigation. The Global Navigation Satellite System (GNSS) can provide users with position information that is wide-ranging and highly accurate in all weather conditions. GNSS signals are easily disrupted by electromagnetic interference in environments such as high-rise buildings, canyons, and high-voltage towers. [1] Therefore, the integrated navigation system combining Inertial Navigation System (INS) and Global Navigation Satellite System (GNSS) has become the mainstream solution for vehicle and drone navigation. [2] A single navigation system and sensor cannot meet the navigation requirements in different scenarios, and the navigation method integrating multi-source information is the current development trend.

In the current research on integrated navigation information fusion algorithms, the more common ones are the Kalman Filter (KF), Extended Kalman Filter (EKF), and Unscented Kalman

Filter (UKF). [3-5] These filters can quickly update state estimates after receiving new observation data and are widely applied in navigation systems. However, algorithms like KF and EKF are all based on the linearization assumption of the system model, making them difficult to meet the robustness requirements of navigation in actual environments. Factor Graph Optimization (FGO) uses nonlinear optimization methods to process data and can reprocess the signal loss interval when GNSS signals recover after losing lock. Therefore, compared with the traditional EKF algorithm, FGO can yield more reliable navigation and positioning results. [6] The GNSS positioning model that won first place in Google's smartphone decimeter-level positioning competition utilizes the FGO model, demonstrating its effectiveness. [7] However, due to the FGO optimization process relying on multiple iterative calculations to obtain the optimal solution, its real-time precision in navigation and positioning is poor.

When GNSS signals lose lock due to an occluded environment, the GNSS/INS integrated navigation system will degenerate into navigation and positioning relying solely on the INS. As the GNSS signal loss time extends, positioning accuracy will drop rapidly. To address this issue, a neural network module can be designed to assist the GNSS/INS integrated navigation system to suppress error divergence during INS-only positioning. [8] Some scholars have used Radial Basis Function (RBF) and Support Vector Machines to model the INS error system and correct the INS error. However, these neural networks are prone to local optimum and cannot fully handle nonlinear noise and drift issues in the INS system. [9-10] To overcome the limitations of the RBF scheme, references [11] use Recurrent Neural Networks (RNNs), which can be trained based on current and past position and velocity samples, effectively processing time-series data and avoiding the problem of local optimum in RBF neural networks during training. Meanwhile, references [12] use Long Short-Term Memory Recurrent Neural Networks (LSTM) to construct a relationship model between INS solution results and GNSS position increments, thereby predicting position increments when GNSS signals lose lock and achieving accurate of position data. Tang [13] proposed a prediction compensation method for integrated navigation systems based on GRU neural networks. Compared with LSTM neural networks, GRU neural networks have a simpler structure and higher operational efficiency. However, in constructing integrated navigation systems, the aforementioned documents all adopt a traditional single-channel structure, meaning the same weight coefficients are used for fitting and prediction of different position information such as longitude and latitude. In practical application scenarios, there are significant differences between the longitude error and latitude error of the inertial navigation system, leading to different characteristics in the nonlinear correlation between the input information of the neural network model. Therefore, the traditional single-channel neural network structure is usually unable to simultaneously achieve the optimal fitting effect for different position information of the INS.

To address the above issues, this paper takes the INS/GNSS integrated navigation system in GNSS signal loss environments as the research object and proposes a factor graph fusion Kalman filter optimization algorithm assisted by a CNN-GRU neural network. The CNN-GRU neural network combines the advantages of CNN and GRU, being able to not only extract key features from multi-variable input data but also better handle the shortcomings of gradient disappearance and long-term dependence in other models. First, low-pass filtering is applied to the IMU data to remove noise, and a pseudo GNSS position data is predicted using the CNN-GRU model to improve navigation accuracy when GNSS signals lose lock. Extended Kalman filtering is used to preprocess current navigation information and correct errors, effectively enhancing the accuracy and robustness of IMU and GNSS data fusion. Finally, the fused data is used to construct a factor graph optimization model, improving the overall performance and real-time performance of the system. Finally, the proposed algorithm is compared with algorithms based on GRU neural networks and factor graph optimization methods in terms of navigation accuracy and positioning real-time performance, verifying the effectiveness of the proposed algorithm.

2. Mathematical Model

This section mainly describes the algorithm part. First, the INS/GNSS loosely coupled navigation model is introduced, followed by the CNN-GRU neural network model, and finally the factor graph optimization theory and Kalman filter algorithm are detailed.

2.1. Loosely Coupled Model

According to the depth and method of information fusion, GNSS/INS integrated navigation systems usually have three combinations: coupled, tightly coupled, and deeply coupled. Among them, the loosely coupled model has the advantages of simple structure, strong robustness, and low implementation cost [14]. It uses an integrated navigation filter to fuse the navigation solution output by the INS and the position and velocity information output by the GNSS to solve the navigation parameters of the integrated navigation system. Compared with tightly coupled and deeply coupled models, it is more suitable for complex practical application scenarios. Since the focus of this paper is to solve the accuracy problem in GNSS signal loss environments, the loosely coupled model is selected as the combination method, and its structure is shown in Figure 1.

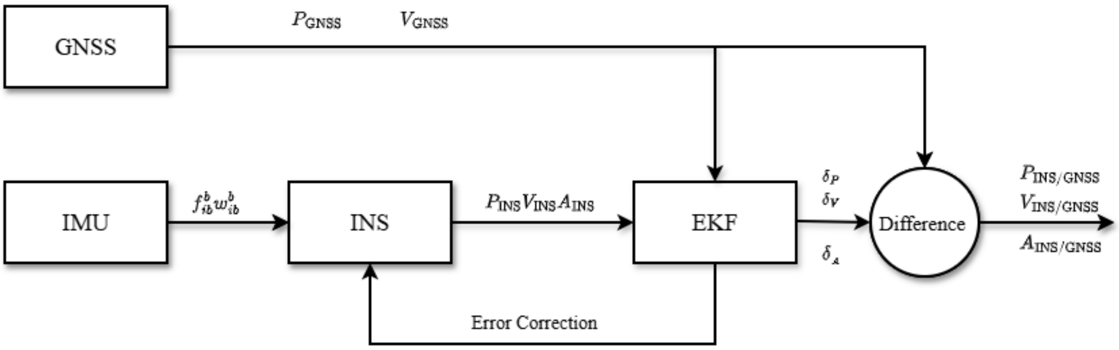

Fig.1 GNSS/INS Tightly Coupled Integration Block Diagram

2.2. CNN-GRU Model

In neural network prediction models, the GRU network has the ability to capture long-term dependencies in time series. Compared with traditional RNN models, it effectively overcomes the problem of gradient disappearance through a gating mechanism and has a simple structure. The CNN network [15] extracts local features from data through convolutional layers. Therefore, this paper proposes combining CNN and GRU networks to achieve navigation trajectory prediction under GNSS signal loss. Its network architecture is shown in Figure 1.

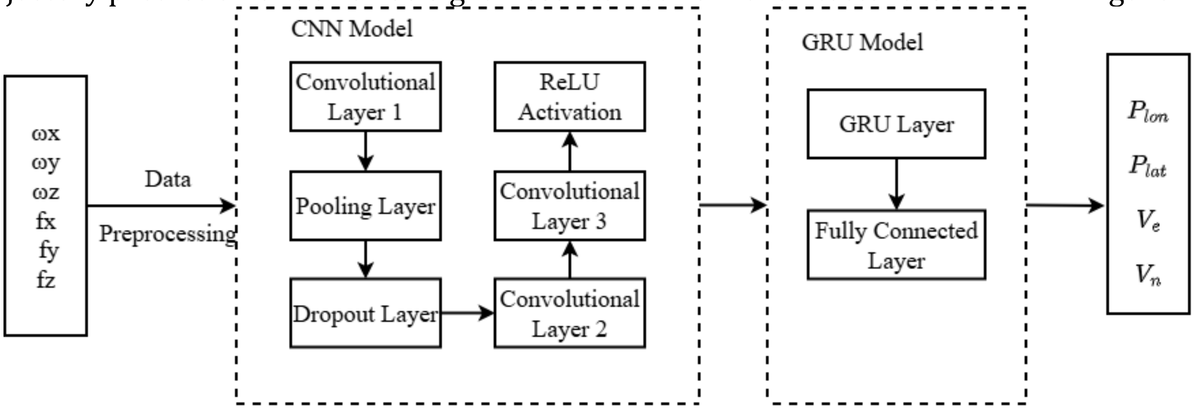

Fig.2 Architecture of a Hybrid CNN-GRU Deep Learning Model

As shown in Figure 2, when GNSS signals lose lock, the CNN neural network first uses convolutional layers to extract local features from the three-axis accelerations fx, fy, fz and angular velocities input by the INS. Pooling layers are then used to reduce the dimensionality of these features. To improve the model's generalization ability and prevent overfitting, a Dropout layer is introduced. Subsequently, the multi-dimensional features are flattened into a one-dimensional vector by a Flatten layer, and finally, the Rectified Linear Unit (ReLU) activation function is applied. The extracted features are then passed to the GRU model, which can model long and short-term dependencies in time series data and further learn the temporal features. The GRU consists of a reset gate and an update gate. It uses a gating mechanism to selectively update information, thereby addressing the gradient vanishing problem that occurs in traditional recurrent neural networks for long sequences. Compared to LSTMs, GRU has a simpler structure and higher computational efficiency. Finally, a fully connected layer outputs the pseudo-GNSS information predicted by the CNN-GRU model to compensate for the GNSS signal missing in the integrated navigation system.

2.3. Factor Graph Algorithm

A factor graph [16] is a probabilistic graph modeling method. Its advantage lies in expressing a complex multi-variable global function in a factored form as a product of local functions. This framework not only overcomes the limitation of the Extended Kalman Filter (EKF) that may introduce approximation errors when handling high-dimensional state spaces, but also supports the flexible addition of new factor nodes to achieve plug-and-play functionality for multiple sensors. Different constraint types are represented by different factor nodes. Navigation state variables are defined as optimization variable nodes to fuse multi-source sensor information. The factor graph model structure mainly consists of a variable set X, a factor set F, and an edge set E. Each factor node can be connected to multiple variable nodes to form a factor graph G=(X,F,E), which contains two types of nodes: factor nodes f and variable nodes x [17]. Its basic structure is shown in Figure 3.

In Figure 3, the variable nodes (x1, x2, ..., xk) represent the system state at time k (k=1, 2, ..., k), and the factor nodes (f1, f2, ..., fk) represent the measurement information at time k. αk represents the error state of the INS. The mathematical model of the factor graph f(x) is the product of factors:

$$\mathrm{f}(\mathrm{X}) = \prod_{\mathrm{k}} \ \mathrm{f}_{\mathrm{k}}\left(\mathrm{x}_{\mathrm{k}}\right) \tag{1}$$

In the equation, f(•) represents the local function corresponding to each factor node, and $\mathrm{f}_{\mathrm{k}}(\mathrm{x}_{\mathrm{k}})$ represents the set of measurement information.When constructing the factor graph for integrated navigation, the state variables in the time series are usually modeled as variable factors:

$$\begin{cases} \mathrm{X} = \{\mathrm{x}_0 \quad \mathrm{x}_1 \quad \cdots \quad \mathrm{x}_\mathrm{k}\} \\ \mathrm{x}_k = \left[p_k \quad v_k \quad q_k \quad \varepsilon_k \quad \nabla_k\right], \ k \in \{0,1,\cdots k\} \end{cases} \tag{2}$$

Herein, x represents the set of node variables associated with x_k, where x_k includes information such as velocity, position, attitude, and the bias parameters of the accelerometer and gyroscope. According to the factor graph theory [14], factor graphs are used to model probabilities and probabilistic relationships. By doing so, the optimal estimation problem for navigation information fusion can be converted into a Maximum A Posteriori (MAP) problem.

$$X^{map} = \underset{\mathrm{X}}{\operatorname{argmax}} \prod_{i} \ f_i\left(X_i\right) \tag{3}$$

To convert the multi-source information fusion problem into a nonlinear optimization problem, the negative logarithm of equation (3) is taken and minimized to solve for the MAP, transforming the optimization estimation into a weighted least squares problem:

$$\mathrm{X_{map}} = \operatorname{argmin}\left\{-\log \prod_i \; f_i\,(X_i)\right\} = \operatorname{argmin}\left\{\sum_i \; \| \,(h_i(X_i) - Z_i)\|^2_{\Sigma_i}\right\} \tag{4}$$

In the equation: represents the predicted value of the sensor data at time i. The maximum a posteriori probability density solution is obtained by calculating the difference between the actual measured value and the predicted value obtained from the factor nodes, thereby achieving multi-source information fusion.

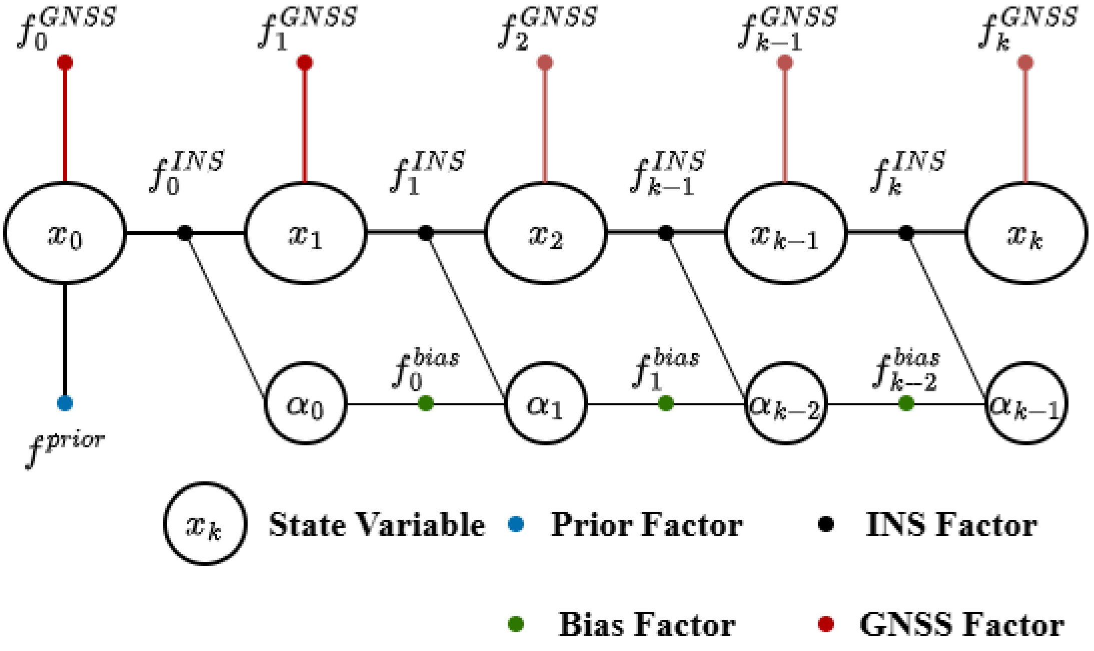

Fig.3 Factor Graph for GNSS/INS Integrated Localization

2.4. Extended Kalman Filter Algorithm

The Extended Kalman Filter (EKF) algorithm overcomes the limitation of the Kalman Filter (KF), which is only applicable to systems that follow a Gaussian distribution. Its core idea is to use the Jacobian Matrix derived from the current state estimation to locally linearize the state transition equation and the observation equation of the nonlinear system. The linearized model is then applied within the Kalman Filter framework to yield updated state estimates and covariance matrices [18]. Although the computation of the Jacobian matrix is added, EKF can effectively handle nonlinear system research, and thus it has found extensive applications in complex real-world navigation scenarios. The principle is illustrated in Figure 3.1. The forms of the state equation and observation equation for a nonlinear system are as follows:

$$\begin{cases} \mathrm{x_k} = f_{\mathrm{k-1}}(\mathrm{x_{k-1}}, \mathrm{u_{k-1}}, \mathrm{w_{k-1}}) \\ \mathrm{z_k} = h_{\mathrm{k}}(\mathrm{x}_k) + \mathrm{v_k} \end{cases} \tag{5}$$

In the equation:f_{t-1} is the state transition function;u_{t-1} is the control input;w_{t-1} is the Gaussian white noise;h_k is the measurement function;v_k is the measurement Gaussian white noise.The state transition function is expanded using the Taylor series to obtain the process Jacobian matrices A and W, and thus the time update equation of the EKF is given by Equation (8).

$$A_{\mathrm{k-1}} = \frac{\partial f_{x-1}}{\partial x}\Big|_{x=\hat{x}^+_{\mathrm{k-1}}} \tag{6}$$

$$W_{t-1} = \frac{\partial f_{x-1}}{\partial w}\Big|_{x=\hat{x}_{t-1}^{+}} \tag{7}$$

$$\begin{cases} \hat{x}_k^- = f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}) \\ P_k^- = A_{k-1}P_{k-1}^+A_{k-1}^T + W_{k-1}Q_{k-1}W_{k-1}^T \end{cases} \tag{8}$$

Based on the time update equation, the system state at time k can be predicted from the state and covariance estimates at time k-1. Then, the state and covariance estimates are corrected using the measurement zt, resulting in the measurement update equation of the EKF [19]:

$$\overline{K}_k = P_k^- H_k^T \left[H_k P_k^- H_k^T + V_k R_k V_k^T\right]^{-1} \tag{9}$$

$$\hat{x}_k^+ = \hat{x}_k^- + \overline{K}_k[z_k - h_k(\hat{x}_k^-, 0)] \tag{10}$$

$$P_k^+ = [I - \overline{K}_k H_k]P_k^- \tag{11}$$

In the equation: H and V are the measurement Jacobian matrices at time k, expressed as:

$$H_k = \frac{\partial h_x}{\partial x}|x = \hat{x}_k^- \tag{12}$$

$$V_k = \frac{\partial h_x}{\partial v}|x = \hat{x}_k^- \tag{13}$$

3. Model and Parameter Settings

3.1. Data Preprocessing

Since the factor graph integrates all observation data and constraints of the system and solves the system state through global optimization, it focuses on optimizing the entire system state to improve accuracy. Therefore, the factor graph optimization method does not have the advantage of real-time performance. In contrast, the Kalman filter has high computational efficiency in real-time processing. To enable the factor graph to be better applied in practical navigation environments, this paper processes GNSS data and IMU data through a Kalman filter before inputting them into the factor graph. This provides the navigation system with initial state estimates and error covariance matrices, allowing the system to more accurately model dynamic characteristics and improve the real-time performance of the factor graph method. The process is illustrated in Figure 4.

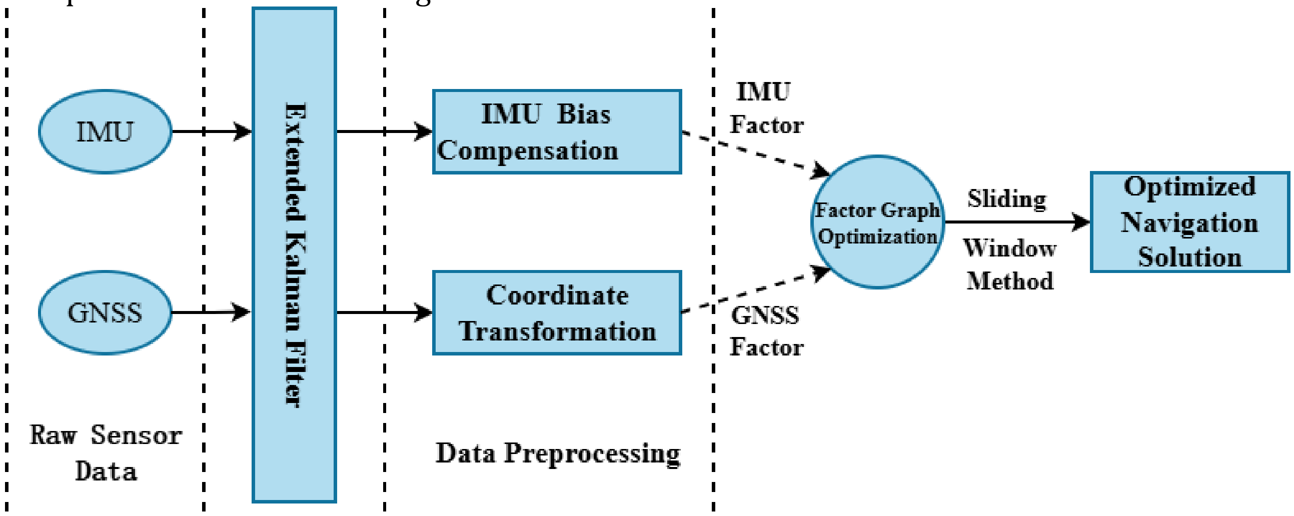

Figure 4 Processing Flowchart of a Sliding-Window Factor Graph Based GNSS/INS System

First, raw data is obtained from IMU and GNSS sensors. The IMU provides acceleration and angular velocity information, while the GNSS provides position and velocity information. Dynamic biases of the gyroscopes and accelerometers are extracted from GNSS data using interpolation methods and applied to the IMU data to eliminate biases caused by sensor errors. The IMU data is preprocessed [19] using the Extended Kalman Filter method to estimate and correct IMU errors. At the same time, coordinate transformation is performed to ensure that GNSS data and IMU data are fused in a unified coordinate system. GNSS factors are constructed using preprocessed GNSS measurements to provide observation constraints for navigation solution. Meanwhile, IMU factors are constructed using corrected IMU measurements to describe prior information of the motion state. In each iteration of the factor graph, the EKF performs error compensation through IMU data and then updates the state through GNSS data. When the amount of data reaches a certain number, a sliding window strategy is adopted to remove expired factors and add new factors. This ensures that the factor graph method performs constraint optimization on historical trajectories during real-time operation, reduces error accumulation, and ultimately improves the robustness and navigation accuracy of the integrated navigation system in GNSS signal-denied environments. Among them, the measurement models of IMU and GNSS are Z_{IMU} and Z_{GNSS}, respectively [20]:

$$Z_{IMU}=\widehat{\omega}_k+\widehat{\alpha}_k+w_{IMU} \tag{14}$$

$$Z_{GNSS}=h_{GNSS}(x_k)+w_{GNSS} \tag{15}$$

In the equation: $\widehat{\omega}_k$和 $\widehat{\alpha}_k$ are the IMU measurement, $h_{GNSS}(x_k)$ is the GNSS input data, and w_{IMU} and w_{GNSS} are the navigation errors during vehicle motion. The EKF performs state prediction based on the system's state equation, and its predicted state is:

$$x_k^- = f(x_{k-1}, u_k) \tag{16}$$

In the equation: f $(\cdot)$denotes the system's state transition function, and u_k represents the acceleration and angular velocity information input by the IMU. Meanwhile, the EKF is used to update the covariance matrix:

$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + Q_k \tag{17}$$

$F_{k-1}$is the Jacobian matrix of the state transition matrix, and Q_k is the process noise covariance matrix. Next, the EKF updates the state estimate by fusing the observation information from the IMU and GNSS. x_k is the updated state estimate, z_k is the actual measurement value, and $h(x_k^-)$ is the observation value derived from the predicted state. The difference between the updated state estimate and the predicted state estimate is fed back into the state estimation as the IMU error, thereby reducing the impact of IMU cumulative errors on the construction of the factor graph optimization.

$$x_k = x_k^- + K_k\big(z_k - h(x_k^-)\big) \tag{18}$$

$$P_k = (I - K_kH_k)P_k^- \tag{19}$$

$$e_{IMU} = x_k - x_k^- \tag{20}$$

3.2. Factor Nodes

(1) IMU Preintegration Factor

The IMU contains triaxial accelerometer and triaxial gyroscope information, with its measured values denoted as

α and ω and true values as $\widehat{\alpha}_k$、 $\widehat{\omega}_k$:

$$\widehat{\omega}_k = \omega_k + b_k^{\omega} + n_k^{w} \tag{21}$$

$$\widehat{\alpha}_k = R_k^{BW}(\alpha_k - g) + b_k^{\alpha} + n_k^{\alpha} \tag{22}$$

Let u_k = ($\widehat{\omega}_k$ $\widehat{\alpha}_k$) represent the IMU input; the state estimate $\hat{x}_k$ at time k is:

$$\hat{x}_k = l_{imu}(x_{k-1}, u_k) \tag{23}$$

where l_{imu} is the IMU state model. To reduce the number of factor nodes, this paper uses the IMU preintegration method shown in Figure 5 to construct an IMU factor within an update cycle. Its nonlinear optimization residual expression is:

$$f_{pre,imu}(x_j, x_i) = \exp\left(-\| \; x_j - l_{pre,imu}(x_i, u_{i,j})\|^2_{\Sigma_{ij}}/2\right) \quad （24）$$

where $u_{i,j}$ denotes the IMU measured values from time i to j-1, and $x_j - l_{pre,imu}(x_i, u_{i,j})$ represents the difference between the preintegrated value and the updated measured value.

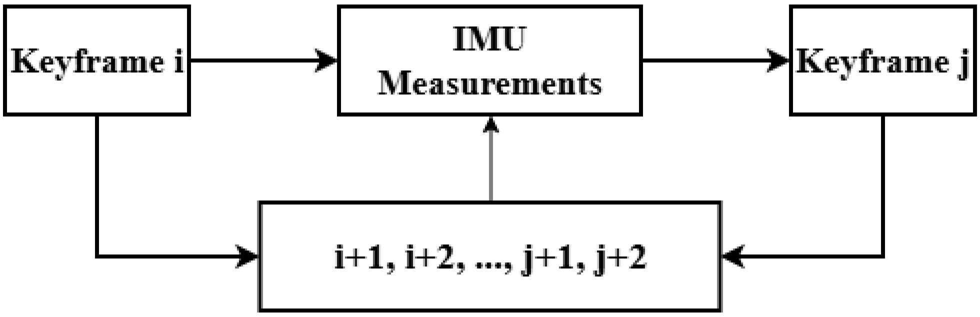

Fig.5 IMU Pre-integration Between Keyframes

(2) GNSS Factor

Typically, the GNSS measurement equation is expressed as:

$$z_k^{gnss} = h^{\mathrm{gnss}}(x_k) + n^{gnss} \tag{25}$$

where h_{gnss}（x_k） denotes the measurement function of the GNSS measurement equation, and n_{gnss} represents the GNSS measurement noise. For loosely coupled GNSS factors, the difference between the measured value and the predicted value is usually treated as the nonlinear optimization residual. Thus, the expression for the GNSS factor node is given by Equation (23), where $\Sigma gnss$ is the covariance matrix of GNSS measurement errors [21].

$$f_{\mathrm{gnss}}(x_k) = \exp\left(-\| \; z_{\mathrm{gnss}} - h_{\mathrm{gnss}}(x_k)\|^2_{\Sigma_{\mathrm{gnss}}}/2\right) \tag{26}$$

3.3. Overall Framework of the CNN-GRU Model

To mitigate the errors of the INS/GNSS integrated navigation system during GNSS signal outages, this paper proposes applying a CNN-GRU neural network to the system. By denoising the acceleration and angular velocity data input from the IMU and compensating the integrated navigation system through the combination of CNN and GRU networks, the accuracy of the navigation system can be improved during GNSS signal outages. Its structure is shown in Figure 5. When GNSS signals are denied, the low-pass filter layer denoises the IMU measurement data. The CNN layer extracts features from the denoised data using convolutional and pooling layers, and the processed feature values are input into the GRU neural network. To avoid overfitting, a Dropout layer is introduced [22]. The input of the CNN-GRU includes thirteen-dimensional time-series vectors: accelerometer data, gyroscope data, time parameter T, and the position P_{ins} and velocity V_{ins} calculated by the INS, totaling thirteen-dimensional time-series vectors. T time-series vectors at consecutive moments are input into the model, and the GNSS position parameters are used as the output to train the CNN-GRU model. The navigation state is corrected using GNSS factors. When the GNSS signal fails, the trained CNN-GRU generates a pseudo-GNSS signal, and the KF-FGO method is used for navigation data fusion, ensuring the accuracy and robustness of real-time navigation.

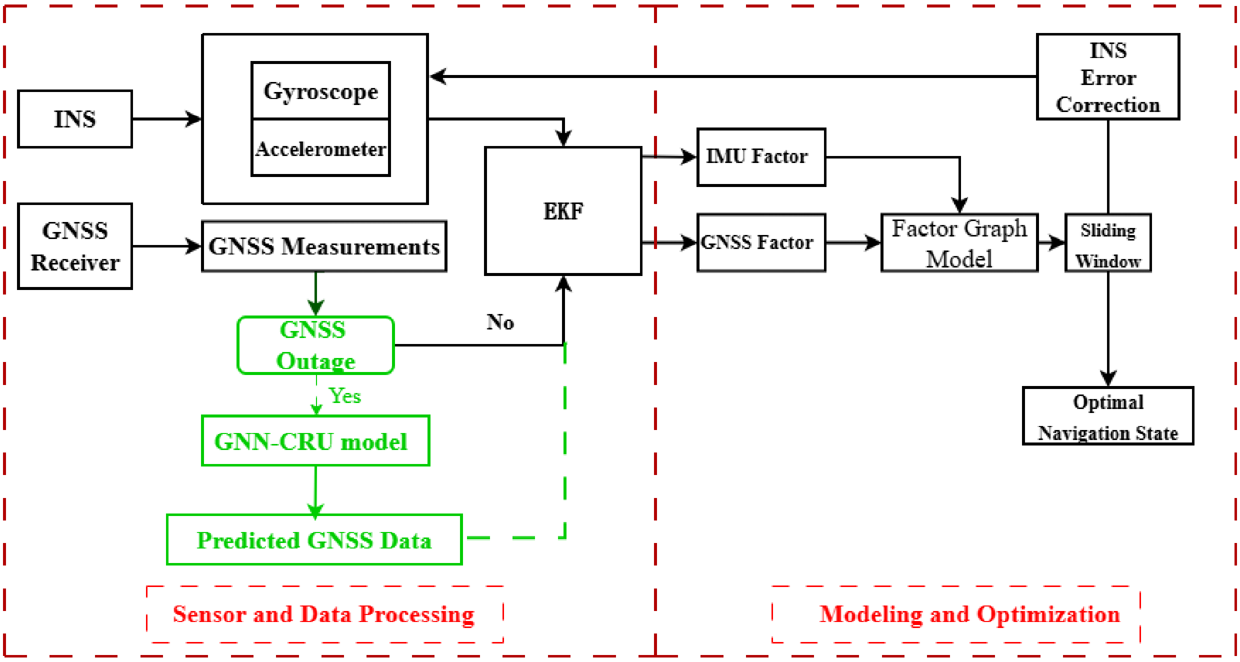

Fig. 6 GNSS/INS Tightly Coupled Integration Diagram

4. Simulation Results and Analysis

To verify the effectiveness of the proposed algorithm in GNSS signal-denied environments, this study uses the NaveGo navigation toolbox to process the integrated navigation system, including INS solution, inertial sensor analysis, and error simulation of GNSS and IMU data. In addition, this paper uses the dataset provided by NaveGo, which is derived from EKINOX-GNSS and EKINOX-IMU devices and collected when the vehicle travels on the streets of Turin, Italy. Since GNSS provides low-frequency absolute position information while IMU provides high-frequency relative motion data, a 5-second delay is introduced during the simulation to achieve data synchronization. The number of samples used for training is 214,029, and the number of verification samples is 220,000, which are randomly selected. The NaveGo dataset is randomly divided into a training set and a validation set, with 80% used for model training under normal GNSS signal conditions and 20% for model validation. The parameter information of EKINOX-IMU is shown in Table 1.

Table1 IMU specific parameters

Parameter	Value
Angle Random Walk (rad/s/√Hz)	[1.5752×10^{-4} 1.6037×10^{-4} 1.8055×10^{-4}]
Velocity Random Walk (m/s^2/√Hz)	[1.9037×10^{-4} 1.8802×10^{-4} 1.9388×10^{-4}]
Gyroscope Standard Deviation (rad/s)	[2.2×10^{-3} 2.3×10^{-3} 2.6×10^{-3}]
Accelerometer Standard Deviation (m/s^2)	[2.7×10^{-3} 2.7×10^{-3} 2.7×10^{-3}]
Gyroscope Static Bias (rad/s)	[-2.4511×10^{-4} 1.2301×10^{-4} -2.2161×10^{-5}]
Accelerometer Static Bias (m/s^2)	[1.6108×10^{-4} -2.4×10^{-2} 3.8×10^{-3}]
Gyroscope Dynamic Bias (rad/s)	[8.0988×10^{-6} 7.1022×10^{-6} 9.8470×10^{-6}]
Accelerometer Dynamic Bias (m/s^2)	[8.3726×10^{-5} 6.7348×10^{-5} 7.3788×10^{-5}]
Gyroscope Dynamic PSD (rad/s/√Hz)	[2.5611×10^{-4} 2.2459×10^{-4} 3.1139×10^{-4}]
Accelerometer Dynamic PSD (m/s^2/√Hz)	[5.9203×10^{-4} 9.5244×10^{-4} 7.3788×10^{-4}]

This study constructs a hybrid deep learning model based on Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) to achieve navigation data prediction in GNSS signal loss environments. The neural network model adopts a hierarchical structure, including two convolutional neural network layers, one gated recurrent unit layer, and a final fully connected output layer. To improve the model's training efficiency and convergence performance, Adam optimizer is used for gradient updates. In addition, dropout layers are introduced to prevent overfitting during model training. To further optimize the model's prediction effect, a large number of experiments are conducted in hyperparameter tuning, and the model performance is evaluated under different parameter configurations. Finally, the optimal parameter settings are determined through multiple rounds of experiments, and the specific parameters are shown in Table 2.

Table 2 GRU neural network specific parameters

Parameter	Value
Input Dimension	13
GRU Hidden Layer Size	128
Number of GRU Layers	2
Fully-Connected Layer Dimension	3
Initial Learning Rate	0.01
Learning Rate Decay Factor	0.5
Learning Rate Decay Epochs	400
Batch Size	128
Maximum Training Epochs	100
L2 Regularization	0．001
Dropout Rate	0.1
Activation Function	ReLu

4.1. Experimental Results and Analysis of the Prediction Model

To verify the superiority of the proposed neural network prediction compensation method, a GNSS outage environment is simulated during vehicle motion. This study uses CNN-LSTM, GRU, and LSTM neural networks to predict pseudo-GNSS data when GNSS signals are lost, and compares the predicted data with the true reference trajectory. The GNSS outage time is set to 25 seconds in the experiment, and the experimental results of different compensation methods are shown in Figure 8. To comprehensively evaluate the positioning and attitude accuracy of the navigation system, the GNSS output values that can provide RTK services are used as the positioning reference, and two error metrics [19-20] are selected for quantitative analysis:

1) Root Mean Square Error (RMSE) $\mathrm{err}_{\mathrm{RMSE}}(\mathrm{l}) = \sqrt{\frac{1}{\mathrm{l}}\sum_{\mathrm{k}=1}^{\mathrm{l}} (\mathrm{e}_{\mathrm{k}} - \hat{\mathrm{e}}_{\mathrm{k}})^2}$

2) mean-square error $\mathrm{R}^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}}$

where $\hat{\mathrm{e}}_{\mathrm{k}}$ is the estimated value of the proposed algorithm, $\mathrm{e}_{\mathrm{k}}$is the RTK measurement value, l is the serial number of the sample points, SSres represents the difference between the model prediction value and the actual value, and SStot represents the difference between the actual value and its mean. The former can reflect the overall magnitude of the error, while the latter can measure the model's ability to explain the variation of the data. To avoid the large number of verification samples from destroying the intuitiveness of the comparison chart, the average value is taken for every ten thousand data points. The metrics of the four experimental results are shown in Table 2.

Table 3 Comparison of Model Errors

Compensation Method	CNN-GRU		CNN-LSTM		GRU		LSTM	
Error Type	Evaluation Metrics							
	RMSE	R2	RMSE	R2	RMSE	R2	RMSE	R2
Longitude Error (m)	0.0509	0.9828	0.0984	0.9654	0.1047	0.9718	0.1645	0.9267
Latitude Error (m)	0.1326	0.8725	0.1680	0.8349	0.2010	0.7976	0.2013	0.7445
Height Error (m)	0.1951	0.7537	0.2612	0.7051	0.4084	0.7705	0.2835	0.4357

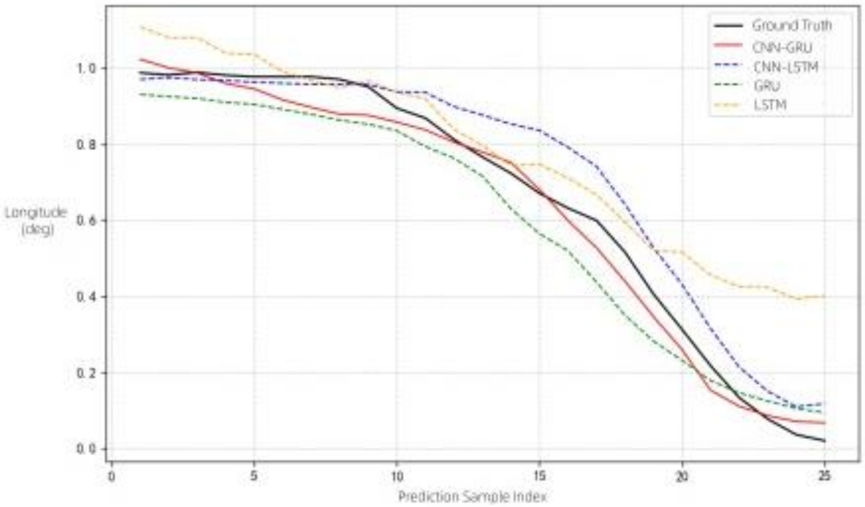

Fig. 7 Longitude comparison for CNN-GRU.

In the GNSS signal outage scenario, this study compares the predicted trajectories using LSTM, GRU, CNN-LSTM, and CNN-GRU neural networks. The specific results are shown in Figures 7, 8, and 9. The experimental results indicate that in the initial signal loss phase, the GRU network exhibits superior initial fitting capability due to its simplified gating mechanism, with its latitude prediction error reduced by 12.3% compared to LSTM. The predicted trajectories of the traditional LSTM and CNN-LSTM models show minimal differences, and both can maintain good spatial consistency with the reference trajectory under normal GNSS signal conditions. As the GNSS outage duration extends to 12 seconds, the interference of temporal cumulative errors becomes more significant for all models. The navigation positioning deviations of the GRU, LSTM, and CNN-LSTM models exhibit a nonlinear growth trend. Among them, the CNN-GRU model effectively suppresses error propagation by integrating the local feature extraction capability of convolutional neural networks and the temporal modeling advantages of GRU. Under the condition of a 25-second long-term outage, its accuracy RMSE is reduced by 69.1% compared to the traditional LSTM. The pseudo-GNSS data predicted by the CNN-GRU neural network has the smallest error, and its predicted trajectory is closest to the reference trajectory.

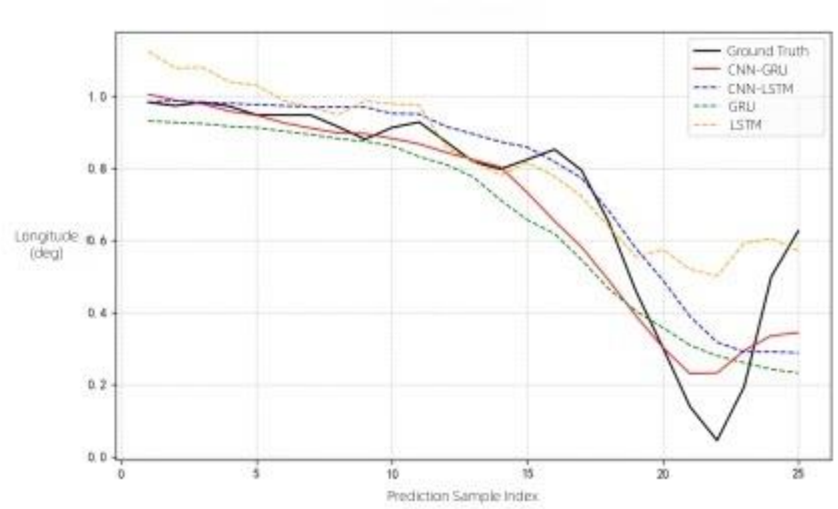

Fig. 8 Latitude comparison for CNN-GRU

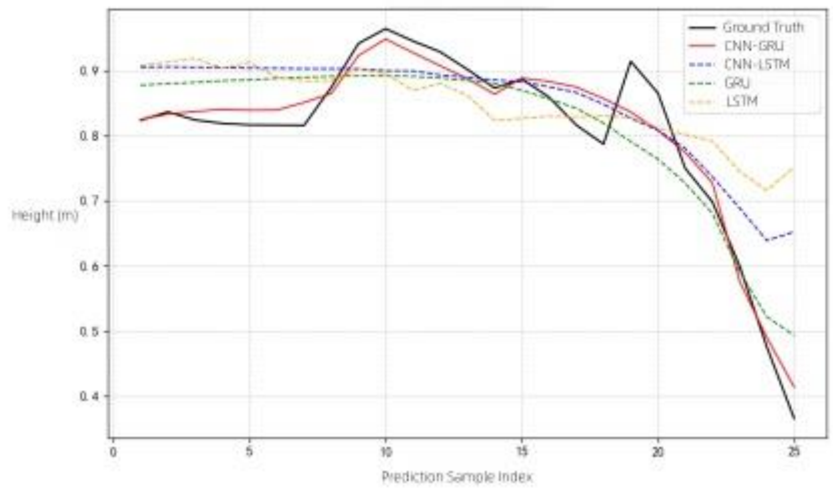

Fig. 9 Height comparison for CNN-GRU

To further verify the effectiveness of the proposed KF-FGO algorithm, this study compares the factor graph optimization results, extended Kalman filter (EKF) processing results, and factor graph optimization results fused with the extended Kalman filter. A comparative analysis of the 3D trajectory comparisons using different methods is shown in Figure 10. Due to the fact that the errors calculated by pure inertial navigation are much larger than those of the FGO and EKF algorithms, they are not discussed here. The results indicate that due to the divergence problem in filtering accuracy when EKF processes nonlinear systems, there are significant deviations between its predicted trajectory and the reference trajectory. In contrast, the FGO output trajectory is closer to the reference path than EKF, especially at the road bend approximately

200 m north, -300 m east, and -10 m height. The root mean square error (RMSE) of EKF-FGO is significantly lower than that of FGO and EKF. This advantage is mainly due to the EKF-FGO fusion model being able to perform local optimization and error correction on IMU data through EKF when the vehicle is in a turning state, thereby effectively reducing cumulative errors at bends and improving navigation accuracy. In addition, although the FGO method performs global optimization by repeatedly iterating observed data to correct trajectory errors, sensor noise can exacerbate position estimation deviations at path bends. FGO cannot dynamically adjust the trajectory in real-time, leading to a reduction in the local accuracy of the navigation system. In contrast, EKF-FGO performs local error compensation through the EKF method, enabling the system to better adapt to trajectory changes in complex dynamic environments and thus enhancing overall navigation performance.

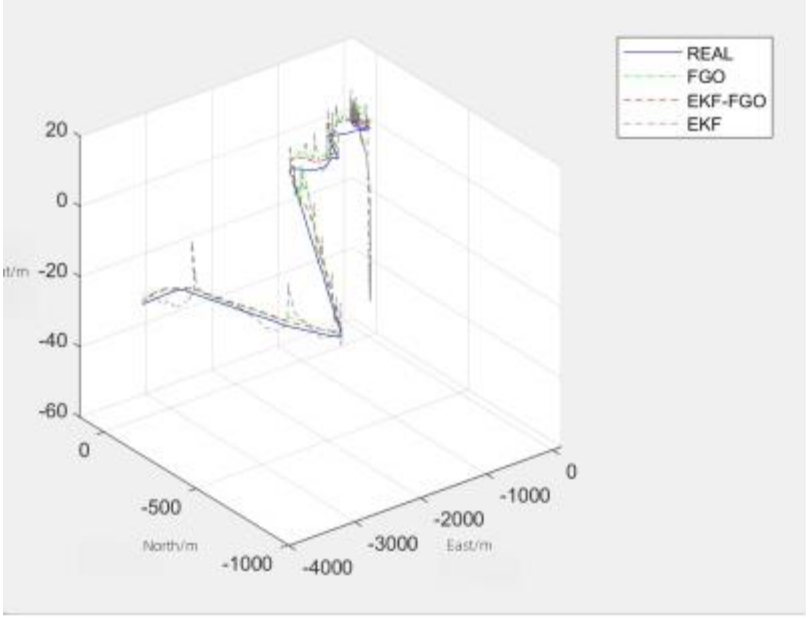

Fig. 10 Comparison of 3D navigation trajectories

To further optimize the data processing efficiency of the FGO-EKF method, this paper uses the sliding window method to limit the size of factor nodes. The root mean square errors of the 3D position estimation under different constraint methods and the processing time are shown in Table 3. It can be seen that: compared with the classic EKF algorithm, the use of the EKF-FGO method reduces the root mean square error values in the east and north directions by 61.15% and 59.16%, respectively, and the average position root mean square error value also decreases accordingly by 14.72%. The EKF-FGO without the sliding window only reduces the error in the vertical direction and the average position by 19.08% and 14.41%, respectively, while there is no improvement in the root mean square error in the east and north directions. In contrast, the EKF-FGO method with the sliding window optimization strategy reduces the root mean square error values in the east and vertical directions by 25.00% and 55.92%, respectively, and its processing time is reduced to 26.39 s, a 57.0% reduction compared to FGO. Therefore, the EKF-FGO method with the sliding window strategy not only improves position accuracy but also significantly reduces processing time, making it applicable to integrated navigation systems where GNSS signals are lost under complex road conditions.

Table 4 RMSE of 3D Position Estimation and Processing Time

Metric		EKF	FGO	EKF-FGO (without sliding window)	EKF-FGO (with sliding window)
Position Error (m)	East	2.78	1.08	1.08	0.81

	North	2.67	1.09	1.09	1.07
	Down	4.74	3.04	2.44	1.34
	Average	3.39	3.4	2.89	1.89
Processing Time (s)		490.15	61.3	75.61	26.39

5. Conclusion

To address the problem of cumulative errors in inertial navigation systems under GNSS signal-denied environments, this paper proposes a combined navigation algorithm based on CNN and GRU-assisted FGO fused with EKF. Through simulation experiments, it is verified that the proposed algorithm achieves lower maximum errors in the north and east directions compared to the comparative algorithms during GNSS signal outages. Meanwhile, combined with the sliding window strategy, the processing time of the data is reduced by 57.0% compared to the traditional factor graph optimization method. The proposed method significantly improves the robustness of the integrated navigation system in complex environments through multi-modal data fusion and dynamic optimization mechanisms, providing a technical reference for high-precision autonomous navigation.

References

[1] S. Chen et al., "Error Compensation Method of GNSS/INS Integrated Navigation System Based on AT-LSTM During GNSS Outages," in IEEE Sensors Journal, vol. 24, no. 12, pp. 20188-20199, 15 June15, 2024, doi: 10.1109/JSEN.2024.3395009.

[2] Jwo, D.,& Weng, T.An adaptive sensor fusion method with applications in integrated navigation. Journal of Navigation,2008,61(4),705-721.

[3] FARUOI F A, TURNER KJ. Extended Kalman filtersynthesis for integrated global positioning/inertial navigation systems.Applied Mathematics and Computation, 2000,115(2/3):213-227

[4] JULIER S J, UHLMANN JK, DURRANT_WHYTE H F.A new approach for filtering nonlinear systems Proceedings of 1995 American Control Conference-ACC95.Piscataway, NJ: IEEE Press, 1995:1628-1632.

[5] ARULAMPALAM M S, MASKELL S, GORDON N, etal. A tutorial on particle filters for online nonlinear/nonGaussian Bayesian tracking.IEEE Transactions on Signal Processing, 2002, 50(2):174-188.

[6] Xiaoni Zheng, Yanpeng Dong, Yufei Zhao, Baosheng Zhang, Mengfan Li,TSF-GINS: Based on time-fixed sliding window with factor graph a global navigation satellite system and inertial measurement unit tightly coupled localization system,Measurement,Volume 239,2025, 115421, ISSN 0263- 224.

[7] SUZUKI T. Precise position estimation using smartphone raw GNSS data based on two-step optimization [J] .Sensors,2023,23(3): 1205.

[8] Bitar, N.A, Gavrilov, A Khalaf, W. Artificial intelligence based methods for accuracy improvement of integ rated navigation systems during GNSS signal outages: An analytical overview-gyroscopy and Navigation,2020,11(1):41-58

[9] El-Sheimy, N., Chiang, K., & Noureldin, A. (2006). The utilization of artificial neural networks for mul- isensor system integration in navigation and positioning instruments. IEEE Transactions on Instrumentation and Measurement, 55(5), 1606–1615.

[10] SHARAF R, NOURELDIN A. (2007). Sensor integration for satellite-based vehicular navigation using neural networks.

[11] Dai, H., Bian, H., Wang, R., & Ma, H. (2019). An INS/GNSS integrated navigation in GNSS denied environment using recurrent neural network. Defence Technology, 16(2), 334–340. https://doi.org/10.1016/j.dt. 2019.08.011

[12] Song, L., Xu, P., He, X., Li, Y., Hou, J., & Feng, H. (2023). Improved LSTM Neural Network-Assisted Combined Vehicle-Mounted GNSS/SINS navigation and positioning algorithm. Electronics, 12(17), 3726. https://doi.org/10.3390/electronics12173726

[13] Tang, Y., Jiang, J., Liu, J., Yan, P., Tao, Y., & Liu, J. (2022). A GRU and AKF-Based Hybrid Algorithm for Improving INS/GNSS Navigation Accuracy during GNSS Outage. Remote Sensing, 14(3), 752. https://doi.org/10.3390/rs14030752

[14] Kim, Y., & Hwang, D. (2017). Loosely-Coupled Vision/INS Integrated Navigation System. Journal of Positioning Navigation and Timing, 6(2), 59–70.

[15] Zou, X., Lian, B., & Wu, P. (2019). Fault identification ability of a robust deeply integrated GNSS/INS system assisted by convolutional neural networks. Sensors, 19(12), 2734. https:// doi.org/ 10.3390/s19122734

[16] Taylor, C., & Gross, J. (2024). Factor Graphs for Navigation Applications: A tutorial. NAVIGATION Journal of the Institute of Navigation, 71(3), navi.653.

[17] Wei, X., Li, J., Zhang, D., & Feng, K. (2021). An improved integrated navigation method with enhanced robustness based on factor graph. Mechanical Systems and Signal Processing, 155, 107565. https://doi.org/10.1016/j.ymssp.2020.107565

[18] Katriniok, A., & Abel, D. (2015). Adaptive EKF-Based vehicle state estimation with online assessment of local observability. IEEE Transactions on Control Systems Technology, 24(4), 1368–1381. https://doi.org/10.1109/tcst.2015.2488597

[19] Qin, H., Wang, X., Wang, G., Hu, M., Bian, Y., Qin, X., & Ding, R. (2023). A novel INS/USBL/DVL integrated navigation scheme against complex underwater environment. Ocean Engineering, 286, 115485. https://doi.org/10.1016/j.oceaneng.2023.115485

[20] Tang, H., Zhang, T., Niu, X., Fan, J., & Liu, J. (2022). Impact of the Earth rotation compensation on MEMS-IMU preintegration of factor graph optimization. IEEE Sensors Journal, 22(17), 17194–17204. https://doi.org/10.1109/jsen.2022.3192552

[21] Li, Q., Zhang, L., & Wang, X. (2021). Loosely coupled GNSS/INS integration based on factor graph and aided by ARIMA model. IEEE Sensors Journal, 21(21), 24379–24387. https:// doi. org/10.1109/jsen.2021.3112490

[22] Naveen, P. (2024). Advancements in underwater imaging through machine learning: Techniques, challen- ges, and applications. Multimedia Tools and Applications, 84(22), 24839–24858. https://doi.org/10.1007/s11042-024-20091-4