



A multi-sensor fusion positioning approach for indoor mobile robot using factor graph

Liyang Zhang, Xingyu Wu, Rui Gao, Lei Pan^{*}, Qian Zhang

School of Control and Mechanical Engineering, Tianjin Chengjian University, Tianjin 300384, China

ARTICLE INFO

Keywords:

Navigation and positioning
Indoor robot
Multi-sensor fusion
Factor graph

ABSTRACT

Navigation and positioning based on sensor fusion has received great attention in low computational complexity, high positioning accuracy and good robustness for indoor mobile robots. This paper presents a multi-sensor fusion factor graph (MSF-FG) positioning method applying IMU, Odometer and LiDAR sensors. By taking advantage of the sensor characteristics and using factor graph theory, a MSF-FG positioning model is constructed to improve positioning accuracy and reduce computational complexity. In addition, an adaptive function is designed to improve the robustness of the system by dynamically adjusting the weight of each factor. Meanwhile, the proposed algorithm is derived by Gauss-Newton and Levenberg-Marquardt methods. Simulation and experimental results show that compared with the conventional inertial navigation system (INS) and extended Kalman filter (EKF) algorithms, the proposed MSF-FG positioning method not only reduces the mean location error by about 40%, but also reduces the computational complexity and enhances the stability of the system.

1. Introduction

Computer control and sensor technique promote the wide application of mobile robots in large indoor scenarios such as supermarkets, airports, hotels, museums and exhibition halls [1–3]. Environment perception is a prerequisite for efficient interaction between mobile robots and the scenario, and accurate acquisition of the robot's location is the key to ensuring the smooth completion of various tasks [4]. Therefore, the accuracy and reliability of mobile robot positioning navigation system have been widely concerned.

Generally, GPS and base station positioning are widely known and used in outdoor positioning [5]. However, outdoor satellite positioning techniques cannot be applied for indoor scenarios due to the diversity of structure and layout as well as occlusion and obstacles. Thus, the robot needs lower positioning delay and higher positioning accuracy to complete tasks under indoor scenario [6]. According to different application demands, indoor positioning technique is rich and diverse, including inertial navigation, WIFI, Bluetooth, RFID, UWB, Odometer, Ultrasonic and LiDAR positioning, etc. Among them, WIFI, UWB, Bluetooth and RFID all need to deploy positioning base stations, resulting in increased cost burden. Therefore, the Odometer, inertial measurement units (IMU) and LiDAR were selected for the actual project requirements to reduce setup work and cost. Odometer positioning is used in indoor robots

because of its high positioning accuracy, low maintenance cost and high reliability [7]. However, the robot cannot eliminate tire slipping and other phenomena. LiDAR navigation can get the map information of the surrounding environment, but required to arrange laser reflective markers for navigation, otherwise the positioning accuracy is low, the reliability is low and the complexity is high. The INS has developed with the advantages of small size, long life and strong anti-interference ability. However, INS positioning errors not only accumulate over time, but also INS takes a long initial alignment time [8]. Since the error characteristics of various sensors are difficult to change, a method of multi-sensor fusion based on different characteristics of IMU and other sensors is developed [9–11], complementing redundant information to calibrate the robot positioning errors in real-time. Although the positioning accuracy is improved to a certain extent, the measurement information from different sensors is difficult to fuse effectively. Besides, real-time positioning suffers from delay and poor positioning stability.

In recent years, multi-sensor fusion positioning technique applied to indoor mobile robots has been developed rapidly. In order to accurately locate the robot, the authors in [12] proposed an indoor positioning method for mobile robots based on radio RFID and Ultrasonic sensors. However, this method is limited to the robot application scenario, only suitable for robot pose calibration. To meet the demand for indoor service robot positioning accuracy and stability, the authors in [13]

^{*} Corresponding author.

E-mail address: panlei4089@163.com (L. Pan).

proposed an alternative indoor positioning method for mobile robot based on the fusion of secondary radar data, Ultrasonic and Odometer sensors. An auxiliary device with absolute position is used to improve the precision and stability of the robot positioning, but the method is complicated and the robot moving range is limited. The authors in [14] applied Odometer, IMU and LiDAR for real-time positioning based on Kalman filtering algorithm, but the location error is relatively large and the robustness is low. The authors in [15] designed a positioning system integrating vision and Odometer to simplify the tedious work at the early stage of robot operation and facilitate task execution in a dynamic environment. Although the system can satisfy the simple positioning accuracy task, the sensor fusion efficiency is very low and the stability is insufficient. Considering high computational complexity due to asynchronous and non-linearity between different sensors, federated filter and EKF are mostly adopted at present [16,17]. The authors in [18] applied Doppler velocity log (DVL) as reference information to propose a new robust adaptive federal strong tracking Kalman filter to make GPS module correct the divergence error of INS and improve the reliability of the whole system. The authors in [19] fused three sensors of different frequencies (GPS, IMU and Odometer) based on EKF and a smoothing device to solve the problem of asynchronous data fusion and improve the noise anti-jamming ability. Interpolation or extrapolation can be used to fuse nonlinear information of different frequencies to minimize information loss. Although the stability of the system and the positioning accuracy at any time are improved, the whole state vector must be re-estimated every time the sensor information arrives, resulting in a large amount of calculation. In addition, the fusion framework needs to be rebuilt when sensors are not available, and increasing the number of sensors also reduces the fault tolerance rate.

The methods on the basis of graph optimization have been widely applied in unmanned mapping and navigation [20]. The authors in [21] studied the probabilistic fusion of multiple sensors under the framework of Hidden Markov model (HMM) of mobile device user positioning. Nevertheless, the proposed graph structure fused numerous sensors without considering the computational complexity and sensor stability. Factor graph is used as a new optimization method to deal with sensor asynchronous measurement information. The variable nodes in the factor graph are associated with the system state, and the correlation between factors and measured values enables the factor graph to have the “plug and play” capability of heterogeneous and asynchronous navigation information [22]. Factor graph encodes the posterior probabilities of states over time for all available measurements given and has simple universality [23–24]. In this way, the problems related to multi-sensor fusion can be easily solved. The authors in [25] designed a factor graph positioning algorithm based on the fusion of IMU and visual sensor by using Kalman filter to ensure smooth solver can work normally under the condition of asynchronous navigation information. Although the positioning methods based on factor graph have good performance in processing asynchronous data, sensor faults can fuse the wrong location information [26]. In order to enable the factor graph method to deal with unstable sensors, the authors in [27] proposed a new factor graph positioning scheme by taking the prediction information as a prior consideration. Significant performance improvements and better tracking performance can be observed when severe sensor errors are suddenly encountered. However, once the sensor has large or small measurement information deviation due to its own reasons or environmental interference, the effective information cannot be distinguished and extracted, and the degradation of sensor performance is not taken into account. Then, to compensate for the extra bias, the authors in [28] introduced the validity recognition algorithm considering the degradation and loss of sensor measurements. Possible sensor faults are detected according to different sensor characteristics, and the effect of sensor measurement deviation is alleviated by covariance adjustment. This method improves the robustness of the positioning system, but it is difficult to meet the needs of the environment with high positioning accuracy. The factor graph positioning algorithms proposed above are

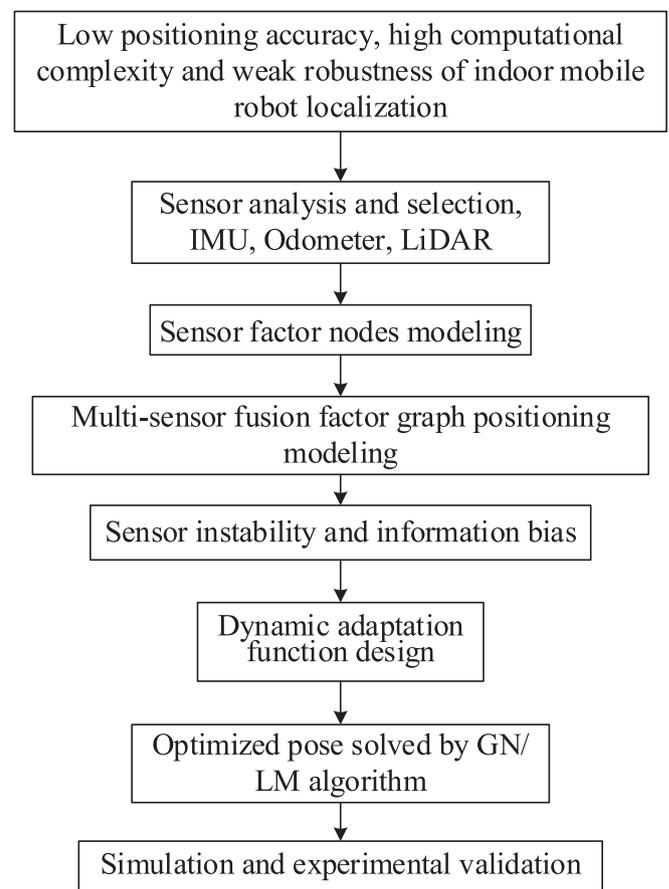


Fig. 1. Block diagram of the proposed research.

generally applied to the occasion where the positioning accuracy is not very high.

In order to solve the problems of low positioning accuracy, high computational complexity and poor robustness of mobile robot indoor positioning system, a multi-sensor fusion positioning method based on factor graph is proposed in this paper. The main advantage of the sensors chosen for this method is that no additional infrastructure is required, resulting in accurate positioning and mapping at a lower cost. The block diagram of the proposed research is shown at Fig. 1. The main contribution of this paper can be summarized as follows:

1. Considering the sensor characteristics, three sensor factor node models, namely IMU, Odometer and LiDAR, were constructed.
2. Combining three sensor factor nodes and variable nodes, a multi-sensor fusion factor graph positioning model was established. Besides, the state estimation was optimized by using the information difference between the sensors.
3. A dynamic adaptive function is designed to improve the robustness of multi-sensor fusion positioning system.
4. The effectiveness of the proposed algorithm in sensor normal and abnormal working conditions was verified by simulation and experiment.

The remainder of this paper is organized as follows. Section II presents the theory related to the factor graph algorithm and the designed positioning system. Section III introduces the sensors modeling for the multi-sensor fusion positioning system. Section IV describes the proposed multi-sensor fusion positioning method using factor graph. Section V provides simulation validation and field experimental test results. Conclusions are given in Section VI.

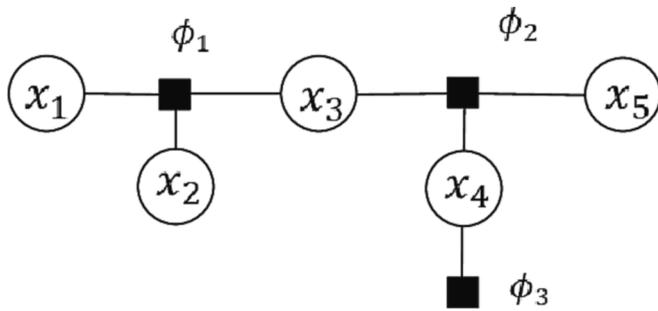


Fig. 2. Factor graph example.

2. The principle and positioning system

This section introduces the methods of factor graph probabilistic modeling commonly used in the field of robotics, and the corresponding indoor mobile robot positioning system is described in this paper.

2.1. Factor graph

Probabilistic graph model can usually describe the conditional independence relationship between multivariate random variables [29]. Generally, the confidence of multivariate random variable \mathbf{X} can be modeled by the probability density function (PDF) $p(\mathbf{X})$ of variable \mathbf{X} . When an observation set Z is given, the simple conditional probability density can be expressed as.

$$p(\mathbf{X}|Z) \quad (1)$$

where the unknown variable \mathbf{X} is the pose of the robot. The process of obtaining conditional probabilities is called probabilistic inference.

Unlike Bayesian network or Markov random field, factor graph can specify not only the probability density [30], but also any factor function $\phi(\theta)$ on the set of variable θ , and can be represented by a binary graph consisting of variable X and factor node ϕ . The variable nodes in the factor graph are related to the system state, and the factor nodes are related to the measured value. For a given factor graph relation formula $F = (\Theta, \Phi, \varepsilon)$, where $\phi_i \in \Phi$ is the factor node, $\theta_i \in \Theta$ is the variable node, and the edge between the variable nodes of the factor nodes is $e_{ij} \in \varepsilon$. In this way, the factor graph $F = (\Theta, \Phi, \varepsilon)$ can be defined as the function factorization associated with all variables X , expressed as.

$$\phi(\mathbf{X}) = \prod_i \phi_i(X_i) \quad (2)$$

For example, a function that can be factored as.

$$\phi(\mathbf{X}) = \phi_1(x_1, x_2, x_3)\phi_2(x_3, x_4, x_5)\phi_3(x_4). \quad (3)$$

The factor graph corresponding to (3) can be described in Fig. 2, where each hollow circle represents a variable node and each solid square represents a factor node. The lines between circles and squares represent the relationship between the corresponding variables and the factor node.

2.2. Positioning system

The indoor positioning system of mobile robot is integrated with IMU, Odometer and LiDAR sensors. The structure of positioning scenario and positioning system is shown in Fig. 3. The left part of Fig. 3 is second floor plan of the school office building. The mobile robot is in the lower left corner of the picture, and the orientation of the robot under the world coordinate system is indicated in the picture. The autonomous mobile robot can move in the indoor scenario according to the command and complete the corresponding marching tasks. The running area is marked in the red area. On the right side of Fig. 3 is the main principle frame of the robot positioning system. Firstly, the robot collects prior measurement information through three kinds of sensors (IMU, Odometer and LiDAR) when building the map. Then, the original sensor data will be input into different measurement models while the robot is navigating and positioning. IMU gets information about the robot's acceleration and angle. Odometer sensor obtains robot velocity and orientation information. LiDAR receives robot location information from temporal and spatial correlation respectively. According to the prior information and sensor measurement model, the initial state estimation and error constraint can be obtained. The navigation state information output by different sensor measurement models is formulated into sensor factors. Secondly, the sensor factors are input to the preset factor dynamic adaptation function as factor nodes, and the robot navigation state is modeled as variable nodes. In this way, variable nodes and factor nodes are used to construct the factor graph model of positioning system. Finally, the information of each node is nonlinear fused, and the optimal state increment estimation of the robot navigation is obtained by Gauss-Newton (G-N) method and Gauss-newton and Levenberg-Marquardt (L-M) algorithm. Then the optimal pose of the robot under navigation is obtained. The multi-sensor fusion factor graph positioning model and the derivation of location estimation algorithm proposed in this paper will be introduced in the next part.

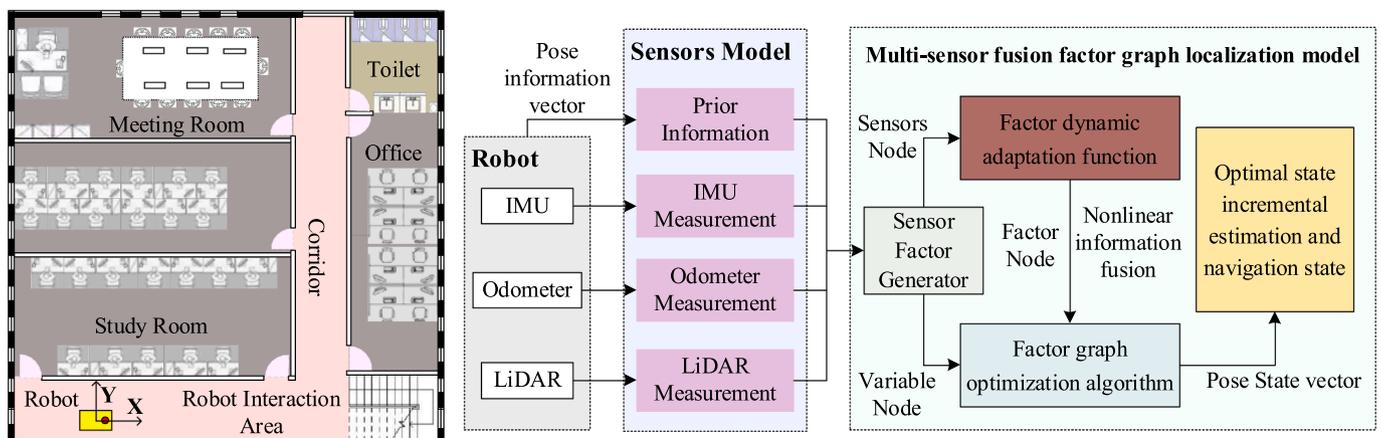


Fig. 3. Indoor positioning system.

3. Proposed Multi-sensor fusion factor graph positioning algorithm

3.1. Sensor modeling

This section mainly constructs sensor factors for IMU, Odometer and LiDAR based on the content in the previous section. Since different sensors have different principles and characteristics, each sensor needs to design a different measurement model and factor node formula.

3.1.1 The system state model

The state of the mobile robot positioning system is a set of physical quantities and all the variables of the sensor that completely describe the robot moving with time, such as location, orientation, velocity and sensor-related parameters. In general, the state at time k is represented as X_k . Assuming that the state changes through the discrete time dynamics model, the following expression is given by.

$$X_{k+1} = f(X_k, V_k, n_k) \quad (4)$$

where n_k is the random noise based on the characteristics of different sensors, and V_k contains all raw measurements from time 0 to k for all sensors such as the wheel encoder or IMU.

A set of measurements Z_1, Z_2, \dots, Z_K are assumed to be of the form $Z_i = H_i(X_{i1}, \dots, X_{ik}, n_{ik})$, where H_i is a known observation prediction function, and the subscript i represents different sensors. Z_K contains the predicted value of the robot state at time k for only one of the sensors. The state of each sensor is in X_1, X_2, \dots, X_k . In reality, the measured values of sensors are uncertain. Thus, each sensor has a random variable noise n_k . In this paper, the physical quantity measured by sensors is taken as the center, and three types of measured values are considered, as follows:

a) The measured value of the robot at time k can be expressed as.

$$Z_k = H_k(X_k, n_k) \quad (5)$$

b) Robot states involving two moments, and the measured values between the two states can be described as.

$$Z_{ok} = H_k(X_o, X_k, n_k), o < k \quad (6)$$

c) The robot's observation of the known landmark information can be expressed as.

$$Z_k = H_k(X_k, l, n_k) \quad (7)$$

For the second measurement method (6), the measured value is given by the motion estimation obtained after sensor observation at two moments O and K , just as the relative motion estimation between two pose nodes in graph optimization. For the third measurement method (7), only the relationship between LiDAR and pose is concerned after the initial value of landmark points is optimized. In this way, the measured values of all time sensors on the robot state can be represented.

3.1.2 Prior information

Prior information is the estimated value before judging the state variable. State $p(X_0)$ is then continuously modified according to the sensor information observed. Prior information can be divided into individual prior information of different sensors and different state variables. For LiDAR, the prior information is represented as the landmark information of a group of state variables. For IMU device, prior information is represented as prior factor nodes constructed by navigation state variables $x \in X_0$ and deviation variables. The probability density function of each prior factor can be expressed separately as.

$$f^{Prior}(x) = d(x) \quad (8)$$

By introducing (8) into Bayes' theorem, prior information can be

changed into $\exp\left(-\frac{1}{2}\|err_t(x - \mu_x)\|_{\Sigma_x}^2\right)$ according to the Gaussian distribution, where μ_x is the average value of X_0 and Σ_x is the covariance of X_0 .

3.1.3 IMU measurement

For the IMU, the discrete form of the equation of state can be described by continuous nonlinear differential equations and calculated by.

$$\dot{X} = H^{IMU}(X, c, f^b, \omega^b) \quad (9)$$

where $H^{IMU}(\bullet)$ is the observation prediction function of the IMU state variable, f^b is the specific force measured by the accelerometer, ω^b is the angular acceleration measured by the gyroscope, and c is the calibration parameter used to compensate the IMU measurement by assuming the IMU error model. The IMU error model is usually estimated together with the navigation state estimation. The process noise used to linearize (9) is a random walk error that follows a Gaussian distribution of zero mean values. Calibration parameter c is the deterministic error of IMU error model. According to the nonlinear model (such as random walk and temperature nonlinear compensation), the time propagation of c can be described by.

$$\dot{c} = g_c(c) \quad (10)$$

where $g(\bullet)$ is the observation prediction function of IMU calibration parameters. The factor graph framework uses a simple Euler integral predictive function to convey state information. The IMU measurement $Z_i = \{f_i^b, \omega_i^b\}$ is related to the navigation state of two continuous time (t_i and t_{i+1}). The nonlinear optimization of IMU state equation only estimates the state of the robot at a certain moment. The continuous formula of IMU measurement model is expressed as.

$$X_{1+i} = H(X_i, c_i, Z_i^{IMU}) + N^{IMU}, c_{1+i} = g(c_i) + N^{bias} \quad (11)$$

Each navigation state X_{1+i} in (11) is obtained from the current inertial measurement Z_i^{IMU} , the previous state X_i , and the calibration parameter c . Calibration parameter c_{1+i} is updated on c_i , and N^{IMU} is the measurement noise of IMU.

3.1.4. Odometer measurement

The wheel speed Odometer mainly uses the pulse count of the encoder to measure the distance increment of the robot within the sampling interval and provide the velocity information of the robot. Therefore, the Odometer is calculated by integrating velocity. According to the geometric kinematics model, the Odometer measurement model can be calculated by.

$$z_i^{Od} = H^{Od}(X_i) + N^{Od} \quad (12)$$

where $H^{Od}(X_i)$ and N^{Od} are the observation prediction function and measurement noise of Odometer respectively. If the robot has location drift, N^{Od} will contain Gaussian noise and calibration parameters.

3.1.5 LiDAR measurement

The LiDAR measurement model is a Range-Azimuth-Elevation (RAE) model. According to the actual measurement equation and prior landmark points, the LiDAR measurement information can be integrated by time and space. For the measured values derived from the LiDAR, the factor graph model can illustrate the correlation between the data. In terms of time data association, each measurement information describes the changes of robot location and orientation on the ground plane between two moments. The LiDAR measurement model is expressed as the relative pose between the pose at moment $i-1$ and the pose at moment i , with no relationship to the roll, pitch and yaw. Hence, LiDAR can be regarded as a binary factor, and the measurement model can be defined

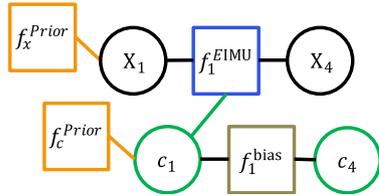
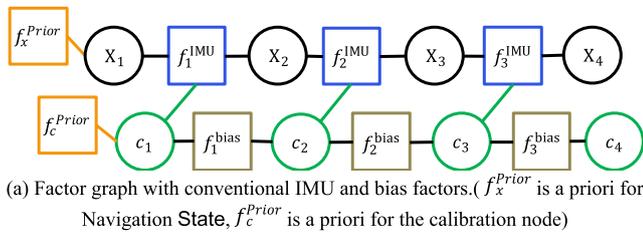


Fig. 4. Factor graph in inertial navigation: (a) and (b) use different symbols to distinguish IMU rate nodes and usage. An equivalent IMU factor accommodates all consecutive IMU measurements between T_1 and T_3 . Navigation and calibration nodes are introduced only when a new factor f^{EIMU} is added at a rate much lower than the IMU rate.

as.

$$Z = H(X_{i-1}, X_i) + N^{lid} \quad (13)$$

where X_{i-1} is the navigation state at the $i-1$ moment, X_i is the navigation state at the i moment, and $H(\bullet)$ is the LiDAR relative pose observation prediction function (representing the relationship between the carrier's pose at two moments). The IMU motion factor can also be regarded as a binary factor. In addition, from the perspective of spatial data association, each LiDAR observation quantity Z_i has a corresponding landmark point l . Bearing measurements can be obtained by observing l at a given X_i . In this way, external entities involving navigation states and unknowns are called external factors, and another measurement model of LiDAR can be expressed as.

$$Z = H(X_i, l) + N^{lid} \quad (14)$$

where $H(\bullet)$ is the observation prediction function, and noise N^{lid} is the LiDAR measurement error described by the zero-mean distribution of the covariance matrix Σ . The errors of LiDAR may arise from the reflection characteristics and motion distortion of the target material.

3.2. Multi-sensor fusion factor graph model

The purpose of the multi-sensor fusion factor graph positioning model established in this section is to provide a mobile robot indoor positioning solution by flexibly utilizing effective sensor measurement information. In addition to the configuration of sensor factor nodes, this section also designed the factor dynamic adaptation function. The factor dynamic adaptive function can detect the sensor state in real time and assign weight values of different factors to ensure the effective fusion of accurate pose information. This section primarily builds IMU nodes, Odometer nodes, and LiDAR nodes based on the measurements of three sensors, respectively. The designed factor dynamic adaptation function is applied to modify the sensor nodes into factor nodes. Then the factor nodes and the state variable nodes are used to build the factor graph positioning model.

3.2.1 IMU factor

Conceptually, the IMU measurement model (11) defines the factors of the relevant nodes in the factor graph. For example, IMU factor f^{IMU} connects navigation nodes X_i and X_{1+i} to bias node c_i , and bias factor f^{bias}

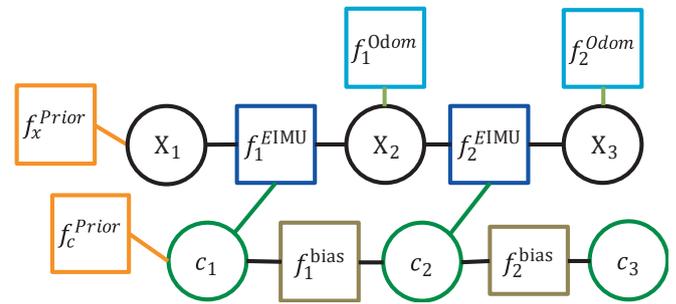


Fig. 5. Factor graph of wheel speed Odometer and IMU sensor measurements. (f^{Odom} denotes the factor node of the Odometer).

connects bias nodes c_i and c_{1+i} . Based on the expression of error function, the conventional IMU factor and bias factor of IMU measurement can be obtained as.

$$f^{IMU}(X_{1+i}, X_i, c_i) = d(X_{1+i} - H(X_i, c, Z_i^{IMU})), f^{bias}(c_{1+i}, c_i) = d(c_{1+i} - g(c_i)) \quad (15)$$

The IMU factor is related to the current state, previous state and bias. Adding a new variable node X_{1+i} to the factor graph requires reasonable initial values that can be obtained by predicting function $H(\bullet)$. c_{1+i} and c_i are represented as variable nodes in the factor graph. The corresponding factor graph model with IMU and deviation factor is shown in Fig. 4 (a). The frequency of the IMU is usually very high, and direct estimation of the navigation state may result in considerable computation. The continuous IMU measurements are combined into an equivalent IMU factor [31], as is the case with the bias node, as shown in Fig. 4 (b).

3.2.2 Odometer factor

According to the Odometer measurement model and (12), the Odometer factor can be defined as.

$$f^{Odom}(X_i) = d(Z_i^{Odom} - H^{Odom}(X_i)) \quad (16)$$

which only connects the variable node X_i of navigation state, and the Odometer factor f^{Odom} can be added to the factor graph together with the equivalent IMU factor f^{EIMU} using the current pre-integral ΔX_i , as shown in Fig. 5.

3.2.3 LiDAR factor

LiDAR estimates robot state changes by observing changes under the surrounding environment [32]. According to the measurement model of temporal data association of LiDAR, (13) representing the pose relationship of the robot at two moments can be obtained. Therefore, the LiDAR binary factor can be expressed as.

$$f^{Lidar}(x_{i-1}, x_i) = d(z_i - H(x_{i-1}, x_i)) \quad (17)$$

On the basis of the spatial data correlation measurement model, external factors can be obtained from the known landmark point l . (14) illustrates the relationship between states and landmarks. Assuming the conditional probability density function $p(z|x, l)$ of the external factor, the external factor can be expressed as.

$$f^{Lidar}(z_i, h(x_i, l)) = d(z_i - H(x_i, l)) \quad (18)$$

Unknown landmarks are included in the factor graph as variable nodes and are added to the factor graph for optimization. Because LiDAR measurements are typically obtained at low frequencies, there may be some lag in state estimation. The interaction between LiDAR and other factors together constructs the improved factor graph model. Fig. 6 shows an example of a factor diagram in LiDAR and other sensor

3.3. Derivation of the proposed positioning algorithm

Based on Bayesian networks [33], factor nodes and variable nodes of factor graph model represent sensor measurement and navigation state respectively. By calculating the maximum posterior estimate of the joint probability distribution function of navigation states over a period of time, the optimal estimate of navigation states for all available asynchronous multi-sensor data can be obtained. A set of state variables with maximum posterior probability can be described as.

$$\hat{X}_{0:k} = \underset{X_{0:k}}{\operatorname{argmax}} p(X_{0:k}|Z_{1:k}) \quad (23)$$

where $X_{0:k}$ is the state variable at time t_0 to t_k , $Z_{1:k}$ is the measurement of all available sensors, and $\hat{X}_{0:k}$ is the maximum posterior probability estimate calculated by maximizing the probability value on the right. According to Bayes' formula, (23) can be transformed into.

$$\begin{aligned} p(X_{0:k}|Z_{1:k}) &= \frac{p(Z_k|X_{0:k})p(X_k|X_{k-1})}{p(Z_k)} \bullet p(X_{0:k-1}|Z_{1:k-1}) \\ &= p(X_0) \prod_{i=1}^k \frac{p(X_i|X_{i-1})p(Z_i|X_{1:k,i})}{p(Z_i)} \end{aligned} \quad (24)$$

Because joint probability distribution functions can be decomposed based on prior information and individual process and measurement models. Taking the state variable with the maximum posteriori probability density as the estimator, the factorization can finally be written as.

$$p(X_{0:k}|Z_{1:k}) = \prod_{i=1}^k [p(X_i|X_{i-1})p(Z_i|X_{1:k,i})] \quad (25)$$

The formula (25) is similar to the factorization process from global function to local function in factor graph. Therefore, applying factor graph to integrated navigation system is reasonable and feasible. In the factor graph, assuming that $f(\bullet)$ represents the local probability distribution function [34], the probability formula of Bayesian transformation can be written as.

$$p(X_{0:k}|Z_{1:k}) \propto \prod_{i=1}^k f_i(X_{1:k,i}) \quad (26)$$

where $X_{1:k,i}$ is a subset of the set of states, and the local function f_i is related to the error function err_i . Thus, f_i can be expressed as.

$$f_i(X_{1:k,i}) = d(err_i(X_{1:k,i}, Z_i)) \quad (27)$$

where $d(\cdot)$ represents the cost function. According to Density function of multivariate Gaussian distribution and applying the Gaussian noise model, (27) can be written as.

$$f_i(X_{1:k,i}) = \exp\left(-\frac{1}{2}\|err_i(X_{1:k,i}, Z_i)\|_{\Sigma_i}^2\right) \quad (28)$$

where $\|\cdot\|_{\Sigma}^2$ represents Mahalanobis distance, and Σ_i represents the covariance matrix of measured noise at time t_i . Therefore, calculating the maximum posterior probability estimate is equivalent to minimizing $\sum_i^k \|err_i(X_{1:k,i}, Z_i)\|_{\Sigma_i}^2$.

Combining the Gaussian noise model with the factor dynamic adaptation function, the maximum posterior estimate of the state variable can be described as.

$$\hat{X}_{0:k} = \underset{X_{0:k}}{\operatorname{argmin}} \frac{1}{2} \sum_i^k \|\Psi(eer_j)err_i(X_{1:k,i}, Z_i)\|_{\Sigma_i}^2 \quad (29)$$

In a practical multi-sensor fusion positioning system, the error function can be expressed as the state estimate minus the actual measured value. Since many sensor measurements in multi-sensor fusion positioning system have nonlinear characteristics, the function

Table 1
Parameters of navigation sensors.

Sensor	Parameter	Value	Frequency
IMU	Gyro(Accel) range	± 500 /sec; ± 16 g	200 ~ 800 Hz
	Gyro bias	1.6 de /h	
	Gyro random walk	0.3 de /h	
	Accel bias	0.6 mg	
	Accel random walk	0.09 m	
	Resolution	0-500:0.0175 /	
LiDAR	Scanning angle	360	10 ~ 50 Hz
	Measurement range	0.2 to 60 m (reflector)	
	Measuring speed	11,520	
	Repetition accuracy	< 12 mm	
	Maximum angular resolution	0.014 / 25,200	
	Measuring method	Pulse Ranging Technology (PRT)	
Odometer	Encoder	6000p/R	50 Hz
	Position accuracy	0.02 m	

of state variable estimation becomes nonlinear least square problem. Next, Gauss-Newton iterative method is used to solve the optimization problem in this paper. To ensure that the calculation method is more universal, the LM method is usually chosen. Thus, nonlinear problems can be transformed into linear problems by Taylor series expansion. The optimal state increment estimate can be expressed as.

$$\hat{\Delta}_{0:k} = \underset{\Delta_{0:k}}{\operatorname{argmin}} \|J(\hat{X}_{0:k})\Delta_{0:k} - b(\hat{X}_{0:k})\|^2 \quad (30)$$

where $\Delta_{0:k}$ is the state increment matrix from t_0 to t_k , $b(\hat{X}_{0:k})$ is the residual matrix measured by the sensor, and $J(\hat{X}_{0:k})$ is the discrete Jacobian matrix that can be decomposed into the equivalent upper triangular matrix by QR or Cholesky [35]. When $\hat{\Delta}_{0:k}$ computes the result by iteration, the new state estimate can be updated by $\hat{X}_{0:k} + \Delta_{0:k}$. Then, the updated state is used as the linearization point in the next iteration. Table A1 of Appendix shows the pseudo-codes of the navigation state update algorithm based on factor graph.

The above is the fusion of factor information using Gaussian Newton iterative method, and the LM algorithm is also available when the Jacobi matrix is a singular matrix. The computational procedure of the factor graph-based navigation state estimation is as shown in Table A2 of Appendix.

4. Simulation and analysis

In this section, the error analysis of the navigation positioning results is presented in a simulated environment with normal and abnormal sensor work respectively. INS and EKF are experimentally compared with the proposed method for evaluating the MSF-FG method. The whole simulation was performed on a PC with i5 CPU @ 2.30 GHz processor and 8 GB RAM, and the development environment was MATLAB version 2019b. The three sensors (IMU, Odometer, and LiDAR) used in the three positioning methods were operated at different output frequencies. The parameters of the three navigation sensors are shown in Table 1.

4.1. Sensors operating in normal condition

For the proposed method, the priority of the IMU factor is set higher than other factors in order to ensure real-time performance, and the equivalent IMU factor is used to join into the factor graph. During the simulation program running, the dynamic adaptation function is made to adjust the robot state variables. Just for example: The IMU factor has a high reliability for mobile robot orientation change, and the Odometer factor has a high reliability for robot velocity change. In addition, the LiDAR binary factor is more reliable in correcting robot pose changes. The confidence degree of external factors of LiDAR is determined according to the observation function of environmental conditions, and

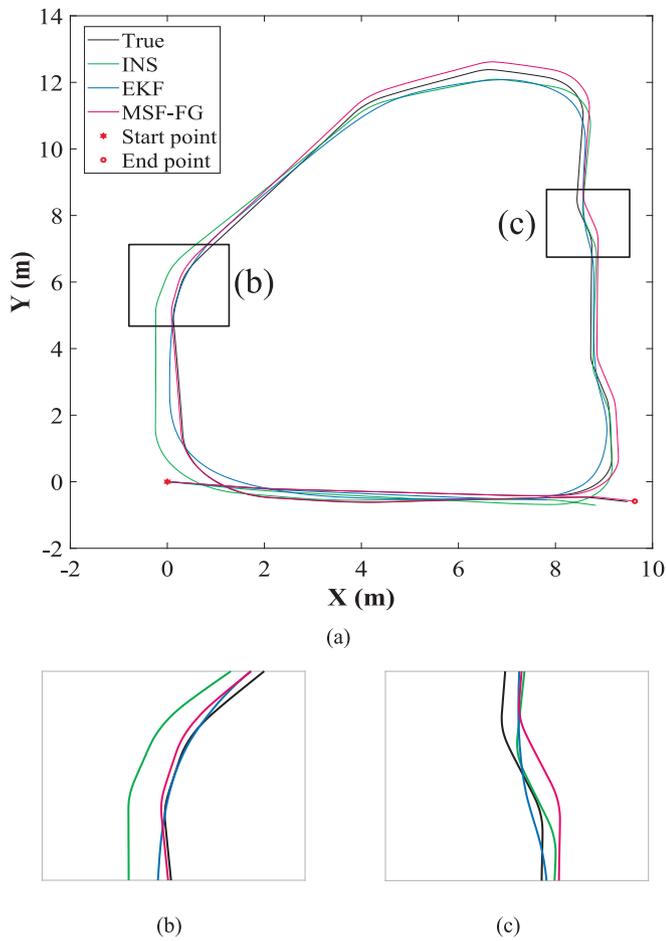


Fig. 8. Ground truth (black), and the estimated navigation trajectory using three methods.

the pose resetting weight is the highest.

Based on the proposed method and the sensor parameters in Table 1, a real typical robot motion trajectory (True) is designed in this paper shown in Fig. 8. The INS method in the trajectory comparison plot is an autonomous navigation system that uses inertial sensitive devices, reference orientation, and initial location information to determine the location, orientation, and velocity of the carrier in inertial space. Magnetometers were not selected in INS because robot in indoor activities is strongly influenced by magnetic fields. EKF is a positioning algorithm that combines the motion model of a mobile robot with an observation model. EKF uses a combination of relative and absolute positioning to achieve more accurate positioning [36]. Nowadays EKF is usually used in unmanned engineering projects. To confirm the performance of the proposed method, three methods are compared mainly in terms of positioning accuracy and computational complexity. The positioning accuracy in this paper includes the root mean square error (RMSE) and the mean location error.

In the comparative plot of the navigation results shown in Fig. 8. The INS through the pre-integration principle of positioning [8], the longer the robot operates the larger the pose error is due to gyroscope drift, accelerometer error, and noise uncertainty. The EKF trajectory curve can follow the real curve well, but turn earlier than the real trajectory. Therefore, the EKF method has unstable positioning accuracy. The proposed method is much smaller than the error of EKF at the turn, as shown in Fig. 8(b) and (c), and the relative error is relatively stable all along.

To further test the positioning accuracy of the proposed method, RMSE was made for all three methods in terms of location and velocity

Table 2
Comparison of RMSE among three positioning methods.

Type	RMSE in the Location		RMSE in the Velocity	
	X (m)	Y (m)	X (m/s)	Y (m/s)
INS	0.313	0.216	0.117	0.050
EKF	0.519	0.50	0.053	0.066
MSF-FG	0.094	0.189	0.019	0.028

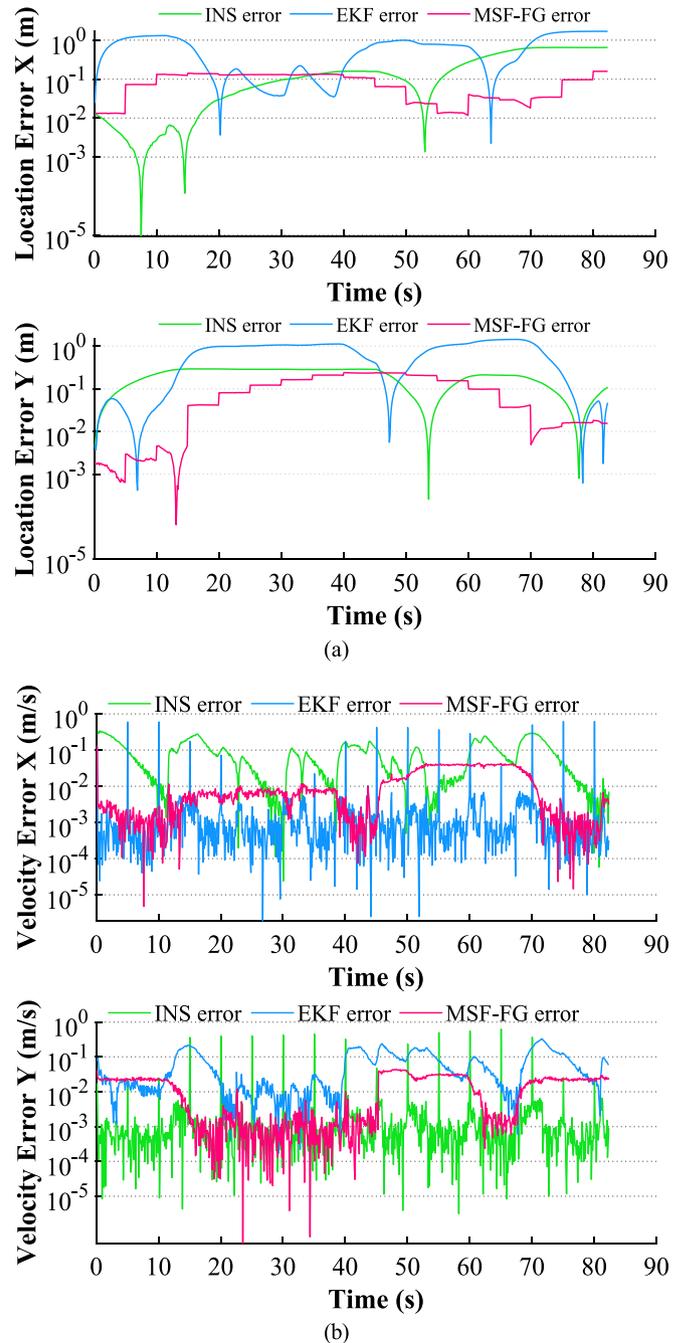


Fig. 9. Comparison of RMSE among different methods under normal operation of various navigation sensors: (a) location RMSE, (b) velocity RMSE.

as shown in Table 2. From Table 2, it can be seen that the proposed method is more accurate than the other two methods in every item. The RMSE change of the three methods with time can be intuitively seen in Fig. 9. The error curve plot better illustrates the performance of different

Table 3
Computation costs of the two methods for different data set sizes.

Data size	5.7 GB	2.6 GB
Methods	Computing Time (s)	
EKF	0.874	0.326
MSF-FG	0.328	0.215

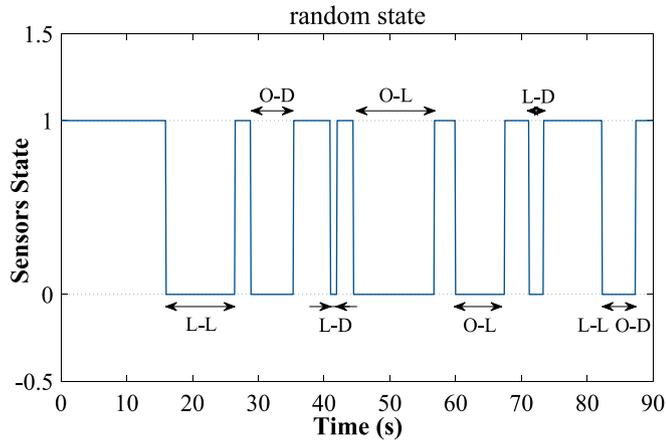


Fig. 10. Sensor positioning random state.

methods for sensors information fusion under the normal operation of various navigation sensors.

The results of the error analysis show that the INS method error gradually becomes larger when each navigation sensor has good performance. The EKF method has unstable positioning accuracy. However, the proposed method has small and stable positioning error. Therefore, the MSF-FG method outperforms the EKF and FG methods. Overall, the location and velocity errors in X-direction are smaller than those in Y-direction. The proposed method effectively corrects the offset errors caused by IMU/Odometer or errors caused by other reasons. Thus, the positioning accuracy of the integrated navigation system is improved.

Due to the simplicity of the algorithm and the small amount of data, the computing time for INS positioning is almost negligible. The difference between EKF and MSF-FG mainly lies in the efficiency of nonlinear fusion for different sensors and the computational complexity of both methods is $O(n)$. 'n' is the number of steps to update the algorithm state. Therefore, the computational complexity of the two methods is mainly reflected in the computing time of state estimation. This paper compares the computing time of different data set sizes, as shown in Table 3. The times in Table 3 are the average computation times for 100 robot state estimation experiments. Compared to the proposed method, the EKF the computational cost is greater. Although the EKF trajectory is smooth and hopefully better, the entire state vector is estimated each time making the delay larger. When applying iterative EKF, large augmented state vectors get more and more time-consuming and do not achieve the desired real-time performance. Compared with EKF, MSF-FG saves more than 30% of the computing time. The proposed method is proven to process data more efficiently and has good scalability.

4.2. Sensors operating in abnormal condition

In this section, the performance of the proposed method is tested in the case of abnormal sensor operation. When the experiment is carried out on complex ground, the robot may have wheel slippage or equipment failure caused by turning or variable velocity, and the OD output is abnormal. The LiDAR positioning data is abnormal when the robot encounters a glass obstacle or moves in a similar environment. The

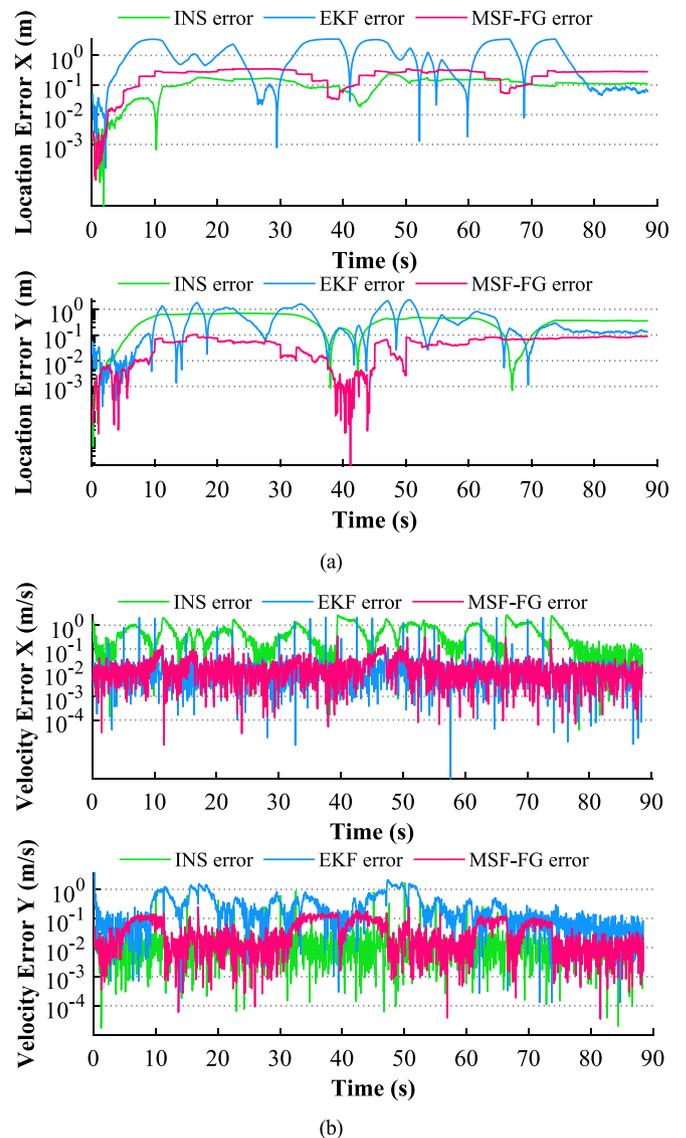


Fig. 11. Comparison of RMSE among different methods under abnormal operation of various navigation sensors: (a) location RMSE, (b) velocity RMSE.

Table 4
Comparison of RMSE among three positioning methods.

Type	RMSE in the Location		RMSE in the Velocity	
	X (m)	Y (m)	X (m/s)	Y (m/s)
INS	0.14	0.425	0.789	0.063
EKF	1.244	0.456	0.125	0.327
MSF-FG	0.23	0.101	0.072	0.061

following experimental scheme aims to verify the effectiveness and robustness of the proposed MSF-FG method.

The following experimental scheme was designed to verify the effectiveness and robustness of the MSF-FG method. The positioning state of the sensors during the experiment is shown in Fig. 10. The number "0" indicates that the sensor performance is degraded or interrupted. The number "1" indicates that the sensor is working normally. A state of "0" in Fig. 10 means that the Odometer or LiDAR sensor is abnormal. "L-L" indicates the loss of LiDAR signal. "L-D" indicates the degradation of LiDAR performance. "O-L" indicates the loss of Odometer signal. "O-D" indicates the degradation of Odometer performance. During the experiment, the sensor noise was increased or the sensor was

Table 5
Computation costs of different methods under the abnormal.

Methods	EKF	MSF-FG
Computing Time (s)	0.851	0.386

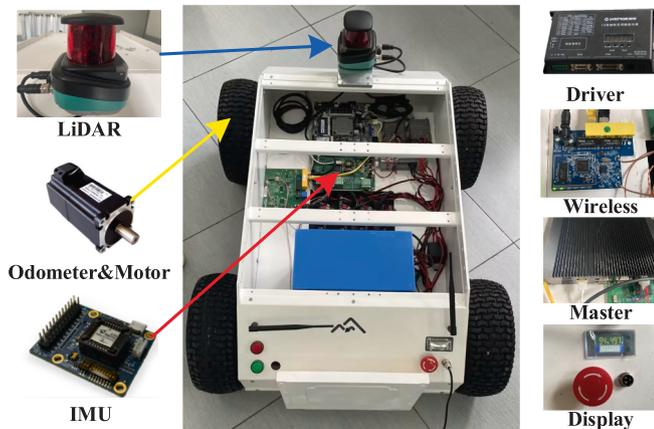


Fig. 12. Experimental robot platform.

set to fault when the state of the sensor was '0'. Fig. 11 compares the location and velocity errors of the different methods in case of sensors performance degradation or even interruption. In addition, Table 4 summarizes the RMSE of the velocity and location errors in both orientations. the computational cost of each method is shown in Table 5.

The error analysis results show that the overall error increases for all methods in the case of degraded or even interrupted sensor performance. The RMSE for EKF positioning is up to 3.5 m. the RMSE for INS positioning is up to 1.0 m. The RMSE for MSF-FG positioning is up to 0.4 m. The MSF-FG method still has a higher accuracy than INS and EKF. Compared with the sensor under normal operation, it can be found that the RMSE of location and velocity of the remaining two methods have increased due to the degradation of sensor performance, but it does not have a great impact on the MSF-FG method proposed in this paper.

The proposed method offers better navigation accuracy and greater robustness than the other two methods due to the introduction of a dynamic adaptation function before fusing the sensor input information and the anomaly information of the performance variation. Comparing Table 3 and Table 5, the EKF runtime increases when there are significant outliers in the measurement information, or even when some sensors fail. The reason for this problem is that the nonlinear information increases significantly in this experiment and the processing of invalid data is more complicated. MSF-FG extracts the valid information in advance, so the running time is almost unaffected by the outliers.

5. Experiments for validation

The proposed method was tested with a robotic experimental platform as shown in Fig. 12. The platform consists of 2D LiDAR, IMU and four-wheel drive chassis including Odometer. The parameters of the navigation sensors are shown in Table 1. The model type of Odometer is 60LCB040C-J00000 (Hollysys Electric Technology Co., Ltd.). And the model type of IMU is SABER CLASSIC (Atom Robotics Co., Ltd). The model type of the LiDAR is R2000 (Pepperl + Fuchs Ltd). The experiments were conducted on an IPC having an Intel i7-7500 single-core processor. The processor has a clock frequency of 2.40 GHz and a memory capacity of 16 GB. The IMU and Odometer data provide highly accurate ground truth location and orientation data. In addition, the sampling frequency of Odometer and IMU is 50 Hz, and LiDAR operates at 20 Hz.

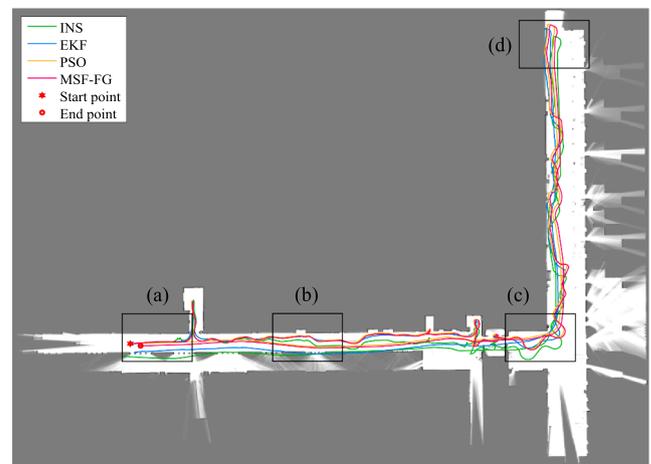


Fig. 13. Real trajectory comparison of the three methods in the experimental field.

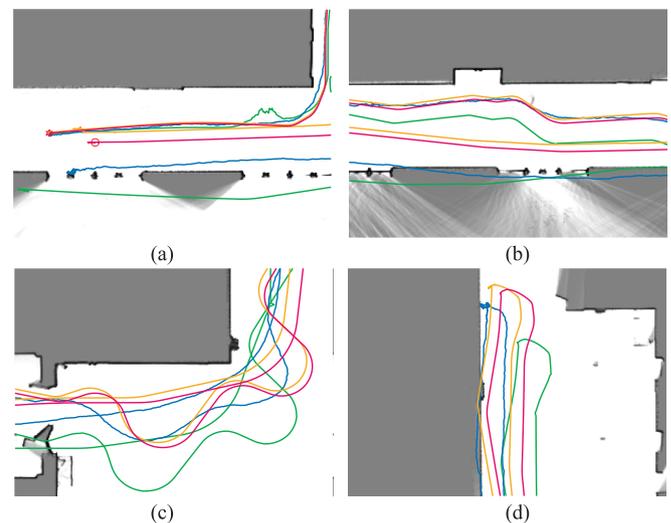


Fig. 14. Comparison of real trajectories with local magnification in Fig. 13. (a) is a zoomed-in plot of the start and end points. (b) shows the robot moving normally. (c) shows the robot turning left or right. (d) shows the robot turning around.

The surfaces selected for robot operation include smooth ground, many different obstacles, and indoor environments with similar landmarks. The environment may make the performance of Odometer and LiDAR may be reduced or even fail. And when sensors are not available in complex environments, the proposed method still achieves accurate navigation solutions. The gyroscope drifts only 0.1 degree in one hour during the motion of the robot. The level of uncertainty in the relative rotation between LiDAR scans calculated by Iterative Closest Point (ICP) is much higher than the uncertainty of the gyroscope. Hence, the positioning credibility of LiDAR is reduced when the robot is rotating.

In order to show the advantages of the proposed positioning algorithm, the experiments were conducted under the same sensor condition. The selected sensors are also in line with the actual needs of logistics company. The experiment was made under the ROS system, the cartographer algorithm was used for slam and navigation. The experimental dataset was used to obtain a comparison plot of the trajectory of the proposed method with the other three methods, as shown in Fig. 13. To distinguish the performance of the MSF-FG method in real experiment further, the four places with obvious location errors in Fig. 13 are labeled and shown in Fig. 14. Areas (a) and (b) in Fig. 14 refer to the

Table 6
Comparison of location error among three positioning methods.

Type	Mean location error		Starting point error	
	X (m)	Y (m)	X (m)	Y (m)
INS	0.37	1.04	0.82	1.97
EKF	0.32	0.56	0.18	0.91
POS	0.26	0.35	0.37	0.44
MSF-FG	0.07	0.18	0.06	0.13



Fig. 15. Experimental scenario specified by a logistics company.

starting point of robot motion, the end point, and the point of robot round-trip error comparison. The turns are responsible for a large portion of the Odometer and LiDAR errors, as shown in Fig. 14(c) and (d). The arrows in the trajectory plot indicate the direction of the robot's route. According to the enlarged part, it is obvious that the starting trajectories of the three methods are almost similar, but the final trajectories are separated. The EKF method gradually shifts with the robot's trajectory, and the INS method shifts more, while the method proposed in this paper basically matches with the real trajectory. The robot navigation positioning applies the MSF-FG method. Since the real trajectory curve cannot be obtained in the experiment, twelve error reference points are set to verify the performance of the proposed method. The mean location error in robot navigation is shown in Table 6. After the robot has run the trajectory several times, the starting point is used as a comparison point to test the position error of the method. Since EKF and

Table A1
Gaussian Newton Iteration Algorithm.

Input: set of original state variables $X_{0:k}$, all measurements $Z_{1:k}$, Set the initial value of the state increment matrix stopping threshold η Maximum iterations m_0
Init: $X_{0:k} \leftarrow 0$ for $t = 1 : k$ do
$f^{IMU}(X_t, X_{t-1}, c_t) \leftarrow d(X_t - H_t^{IMU}(X_{t-1}, c_t, Z_t^{IMU})) \quad f^{bias}(c_t,$
$c_{t-1}) \leftarrow d(c_t - g(c_{t-1})) \quad$ for $j = 1$ to $N_{sensors}$ do
$f^{sensors} \leftarrow d(Z_t^j - H_t^j(X_{0:k,t})) \quad$ end for
$\Psi(eer_j) \leftarrow Z_t^j, H_t^j(X_{0:k,t}) \quad$ end for
Then, the nonlinear multivariate function is linearized.
$J(\hat{X}_{0:k}, b(\hat{X}_{0:k})) \leftarrow \sum_{t=1}^k \ H_t(X_{0:k,t}) - Z_t\ _{\Sigma}^2$ Then determine whether $J(\hat{X}_{0:k})$ is too large or a singular matrix.
while $\ J(\hat{X}_{0:k})\Delta_{0:k} - b(\hat{X}_{0:k})\ ^2 > \eta$ or $m < m_0$ do
$\hat{\Delta}_{0:k} \leftarrow \text{argmin} \ J(\hat{X}_{0:k})\Delta_{0:k} - b(\hat{X}_{0:k})\ ^2$
$m \leftarrow m + 1 \quad$ end while
$\hat{X}_{0:k} \leftarrow \hat{X}_{0:k} + \hat{\Delta}_{0:k}$ Output: Optimal set of state variables $\hat{X}_{0:k}$

INS have a large error during the application in practice, sometimes even run out of the map. Large errors can cause problems with robot colliding with walls. So we compare the trajectory of MSF-FG to calculate the relative mean error of the other two methods, as well as the error of returning to the origin. According to the analysis of experimental results, the proposed MSF-FG method can reduce the mean location error by twice.

In addition, the particle swarm optimization (PSO) [37] method is selected for comparison to better show the superiority of the proposed method. As shown in Fig. 13, The PSO pose estimation is not consistently stable, a little better than EKF but not as accurate as the proposed method. As shown in Table 6, the mean location error of POS is twice as large as that of the proposed method, especially the starting point error is larger.

Therefore, the fusion information positioning results obtained by using the proposed method are better than the EKF and POS fusion results. The experimental trajectory curves and average error data are basically consistent with the simulation results. The proposed method is more adaptable to more complex indoor environments, and not only does the positioning accuracy meet the tasks such as normal robot work, but also has stronger robustness.

As shown in Fig. 12 and Fig. 15, the selected sensors and the mobile robot used in the experimental test are developed for the requirements of a logistics company. The experimental scenario is specified by the logistics company as required. The method proposed in this paper not only meets the needs of the logistics company, but also can provide technical support for indoor robot navigation and positioning.

6. Conclusion

To summarize, this paper presented a MSF-FG indoor robot positioning approach based on IMU\Odometer\LiDAR sensor fusion to improve positioning accuracy, reduce computational complexity and enhance the robustness of the system. According to the characteristics of three different sensors, the sensor factor node models are established respectively. Combining with the factor graph method, a MSF-FG positioning model was constructed to achieve fusing the asynchronous or disordered nonlinear measurement information from multiple sensors and the plug and play function. The relationship between navigation state and sensor measurement was explained in our proposed positioning model. Besides, to make the system more robust, a dynamic adaptive function was designed by dynamically adjusting the weight of each factor. Finally, the optimal pose of the robot was estimated by using Gauss-Newton and Levenberg-Marquardt algorithms. The results reveals that the MSF-FG method can control RMSE below 0.25 m regardless of whether the sensor is normal or not. Compared with INS and EKF, the proposed method reduces the RMSE by at least 40%. Furthermore, the proposed methodology improves the computing time by more than 30% compared to EKF. The proposed positioning technique can not only make the application of multi-sensor fusion more mature, but also satisfy the diversified demands of indoor navigation and positioning services. In our future work, we will improve the proposed framework to explore the multi-floor space positioning for different indoor application scenarios.

CRedit authorship contribution statement

Liyang Zhang: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **Xingyu Wu:** Software, Validation, Writing – review & editing. **Rui Gao:** Validation. **Lei Pan:** Supervision. **Qian Zhang:** Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence

Table A2
Levenberg-Marquardt iterative optimization Algorithm.

Input: initial estimate $X_{0,k}$,
quadratic polynomial cost function $g()$,
stopping threshold η

Init: $t \leftarrow 0$, $\lambda \leftarrow 10^{-3}$
while $\lambda > \eta$ do
 $A, B \leftarrow$ Linearization of $g(X)$ at X_t
 $\Delta \leftarrow$ Solve for $(A^T A + \lambda \text{diag}(A^T A))\Delta = A^T b$
 if $g(X_t + \Delta) < g(X_t)$ then
 $X_{t+1} = X_t + \Delta$
 $\lambda \leftarrow \lambda/10$ else
 $X_t = X_t$ $\lambda \leftarrow \lambda \times 10$ $t \leftarrow t + 1$
 end while

Output: the latest estimate of the set of state variables $\hat{X}_{0,k}$

the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62101377, Tianjin Natural Science Foundation under Grant 22JCQNJC00840 and the research project of Tianjin Municipal Education Commission under Grant 2020KJ057.

Appendix

References

- [1] J. Zhang, X. Yang, W. Wang, et al., Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering, *Autom. Constr.* 146 (2023).
- [2] A.C.A.R. Osman, H. Sağlam, Ş.A.K.A. Ziya, Measuring curvature of trajectory traced by coupler of an optimal four-link spherical mechanism, *Mech. Mach. Theory* 176 (2021) 1–15.
- [3] A.C.A.R. Osman, H. Sağlam, Ş.A.K.A. Ziya, Evaluation of grasp capability of a gripper driven by optimal spherical mechanism, *Mech. Mach. Theory* 166 (2021) 1–19.
- [4] M. Sderlund, When service robots look at themselves in the mirror: An examination of the effects of perceptions of robotic self-recognition, *J. Retail. Consum. Serv.* 64 (2022).
- [5] X. Lei, R. Wang, F. Fu, An adaptive method of attitude and position estimation during GPS outages, *Measurement* 199 (2022).
- [6] W. Qian, F. Lauri, F. Gechter, Supervised and semi-supervised deep probabilistic models for indoor positioning problems, *Neurocomputing* 435 (2021) 228–238.
- [7] C. Kammel, T. Kögel, M. Gareis, M. Vossiek, A cost-efficient hybrid UHF RFID and Odometry-based mobile robot self-positioning technique with centimeter precision, *IEEE J. Radio Frequency Identification* 6 (2022) 467–480.
- [8] B. Barshan, H.F. Durrant-Whyte, Inertial navigation systems for mobile robots, *IEEE Trans Rob Autom* 11 (3) (1995) 328–342.
- [9] Q. Wang, M. Cai, Z. Guo, An enhanced positioning technique for underground pipeline robot based on inertial Sensor/Wheel odometer, *Measurement* 206 (2023).
- [10] T.G.S. Amorim, L.A. Souto, T.P. Do Nascimento, M. Saska, Multi-robot sensor fusion target tracking with observation constraints, *IEEE Access* 9 (2021) 52557–52568.
- [11] Y. Zhang, X. Tan, C. Zhao, UWB/INS Integrated Pedestrian Positioning for Robust Indoor Environments, *IEEE Sens. J.* 20 (23) (2020) 14401–14409.
- [12] B. Choi, J. Lee, J. Lee, K. Park, A Hierarchical Algorithm for Indoor Mobile Robot Positioning Using RFID Sensor Fusion, *IEEE Trans. Ind. Electron.* 58 (6) (2011) 2226–2235.
- [13] Y. Dobrev, P. Gulden, M. Vossiek, An Indoor Positioning System Based on Wireless Range and Angle Measurements Assisted by Multi-Modal Sensor Fusion for Service Robot Applications, *IEEE Access* 6 (2018) 69036–69052.
- [14] Z. Li, Z. Su, T. Yang, Design of Intelligent Mobile Robot Positioning Algorithm Based on IMU/Odometer/Lidar, in: *International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, 2019.
- [15] M. Ouyang, Z. Cao, P. Guan, Z. Li, C. Zhou, J. Yu, Visual-Gyroscope-Wheel Odometry With Ground Plane Constraint for Indoor Robots in Dynamic Environment, *IEEE Sensors Letters* 5 (3) (2021) 1–4.
- [16] H.X. Li, L.H. Ao, H. Guo, et al., Indoor multi-sensor fusion positioning based on federated filtering, *Measurement* 154 (5) (2020) pp.
- [17] E. Wan, R. Merwe, The unscented Kalman filter for nonlinear estimation, in: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, 2000.
- [18] H. Xiong, Z. Mai, J. Tang, F. He, Robust GPS/INS/DVL Navigation and Positioning Method Using Adaptive Federated Strong Tracking Filter Based on Weighted Least Square Principle, *IEEE Access* 7 (2019) 26168–26178.
- [19] V. Girbés-Juan, L. Armesto, D. Hernández-Ferrándiz, J.F. Dols, A. Sala, Asynchronous Sensor Fusion of GPS, IMU and CAN-Based Odometry for Heavy-Duty Vehicles, *IEEE Trans. Veh. Technol.* 70 (9) (2021) 8617–8626.
- [20] X. Tao, et al., A Multi-Sensor Fusion Positioning Strategy for Intelligent Vehicles Using Global Pose Graph Optimization, *IEEE Trans. Veh. Technol.* 71 (3) (2022) 2614–2627.
- [21] X. He, D.N. Aloï, J. Li, Probabilistic Multi-Sensor Fusion Based Indoor Positioning System on a Mobile Device, *Sensors (Basel, Switzerland)* 15 (12) (2015) 31464–31481.
- [22] Y. Ben, Y. Sun, Q. Li, et al., A novel cooperative navigation algorithm based on factor graph with cycles for AUVs, *Ocean Eng.* 241 (2021).
- [23] H.W. Sorenson, A.R. Stubberud, Nonlinear filtering by approximation of the a posteriori density, *Int. J. Contr.* 8 (1) (1968) 33–51.
- [24] H.-A. Loeliger, J. Dauwels, J. Hu, S. Kori, L.i. Ping, F.R. Kschischang, The Factor Graph Approach to Model-Based Signal Processing, in: *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.
- [25] P. Chauchat, A. Barrau, S. Bonnabel, Factor Graph-Based Smoothing Without Matrix Inversion for Highly Precise Positioning, *IEEE Trans. Control Syst. Technol.* 29 (3) (2021) 1219–1232.
- [26] J. Xu, G. Yang, Y. Sun, S. Picek, A Multi-Sensor Information Fusion Method Based on Factor Graph for Integrated Navigation System, *IEEE Access* 9 (2021) 12044–12054.
- [27] M. Cheng, M.R.K. Aziz, T. Matsumoto, Integrated Factor Graph Algorithm for DOA-Based Geolocation and Tracking, *IEEE Access* 8 (2020) 49989–49998.
- [28] M. Cheng, M.R.K. Aziz, DOA-Based 3D Tracking With Factor Graph Technique for a Multi-Sensor System, *IEEE Sens. J.* 21 (22) (15 Nov. 15, 2021.) 25853–25861.
- [29] L. Hambardzumyan, H. Hatami, P. Hatami, A counter-example to the probabilistic universal graph conjecture via randomized communication complexity, *Discret. Appl. Math.* 322 (2022) 117–122.
- [30] D. Frank, K. Michael, Factor Graphs for Robot Perception, *Foundat. Trends Robotics* 6 (2017) 1–139.
- [31] S. Karam, V. Lehtola, G. Vosselman, Simple loop closing for continuous 6DOF LiDAR&IMU graph SLAM with planar features for indoor environments, *ISPRS J. Photogramm. Remote Sens.* 181 (2021) 413–426.
- [32] W. Liu, Z. Li, R. Malekian, M.A. Sotelo, Z. Ma, W. Li, A Novel Multifeature Based On-Site Calibration Method for LiDAR-IMU System, *IEEE Trans. Ind. Electron.* 67 (11) (2020) 9851–9861.
- [33] K.-B. Hwang, B.-T. Zhang, Bayesian model averaging of Bayesian network classifiers over multiple node-orders: application to sparse datasets, *IEEE Transactions on Systems* 35 (6) (2005) 1302–1310.
- [34] Y. Song, L.T. Hsu, Tightly coupled integrated navigation system via factor graph for UAV indoor positioning, *Aerosp. Sci. Technol.* 108 (2021).
- [35] C.J. Demeure, L.L. Scharf, Sliding windows and lattice algorithms for computing QR factors in the least squares theory of linear prediction, *IEEE Trans. Acoust. Speech Signal Process.* 38 (4) (1990) 721–725.
- [36] F. Hu, G. Wu, Distributed Error Correction of EKF Algorithm in Multi-Sensor Fusion Positioning Model, *IEEE Access* 8 (2020) 93211–93218.
- [37] Q. Zhao, C. Li, Two-stage multi-swarm particle swarm optimizer for unconstrained and constrained global optimization, *IEEE Access* 8 (2020) 124905–124927.