

Received January 8, 2021, accepted January 11, 2021, date of publication January 14, 2021, date of current version January 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051715

A Multi-Sensor Information Fusion Method Based on Factor Graph for Integrated Navigation System

JING XU¹, GONGLIU YANG², YIDING SUN², AND STJEPAN PICEK¹

¹Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2600 AA Delft, The Netherlands

²School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China

Corresponding author: Jing Xu (j.xu-8@tudelft.nl)

This work was funded by the China Scholarship Council (CSC). This work was supported in part by the Inertial Navigation Laboratory, Beihang University, and in part by the Artificial Intelligence and Security Laboratory, Delft University of Technology.

ABSTRACT The current navigation systems used in many autonomous mobile robotic applications, like unmanned vehicles, are always equipped with various sensors to get accurate navigation results. The key point is to fuse the information from different sensors efficiently. However, different sensors provide asynchronous measurements, some of which even appear to be nonlinear. Moreover, some sensors are vulnerable in specific environments, e.g., GPS signal is likely to work poorly in interior space, underground, and tall buildings. We propose a multi-sensor information fusion method based on a factor graph to fuse all available asynchronous sensor information and efficiently and accurately calculate a navigation solution. Assuming the sensor measurements and navigation states in a navigation system as factor nodes and variable nodes in a factor graph, respectively, the update of the states can be implemented in the framework of the factor graph. The proposed method is experimentally validated using two different datasets. A comparison with Federated Filter, which has been widely used in integrated navigation systems, demonstrates the proposed method's effectiveness. Additionally, analyzing the navigation results with data loss verifies that the proposed method could achieve sensor plug and play in software.

INDEX TERMS Integrated navigation, multi-sensor, information fusion, factor graph, plug and play.

I. INTRODUCTION

Autonomous mobile robotic systems have been applied in various fields, e.g., health, transportation, and military [1]. Accurate and reliable navigation is essential in such systems and has become a topic of significant research interest [2]. Compared with the early inertial navigation systems in which the inertial measurement unit (IMU) is the prime sensor, current systems are always equipped with additional sensors to get more accurate and reliable navigation solutions. Although the combination of different sensors can improve navigation accuracy, the asynchronism and nonlinearity of sensor measurements make it difficult to fuse multi-sensor information efficiently [3].

Since its proposal in 1960, Kalman Filter (KF) has been widely used to solve many difficult and complicated information fusion problems [4]. However, the classical Kalman Filter can only be applied in linear systems. Although in

the following decades, Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) were proposed for nonlinear systems [5]–[10], with the increasing number of sensors in the navigation system, these centralized filters [11] can result in an enormous computational burden and poor fault tolerance [12]. Recently, some work has been proposed to improve the performance of KF in the navigation system. Hailiang *et al.* presented a novel robust single GPS navigation algorithm based on dead reckoning and strong tracking filter [13]. More recently, an unconventional multi-sensor integration strategy based on the kinematic trajectory model was proposed in [14]. To solve the dilemma of navigation accuracy and computation cost, Speyer *et al.* proposed the idea of a decentralized filter and developed it for optimal estimation problems [15]–[21]. In 1987, Federated Filter (FF) was presented and became a representative of a decentralized filter because of its flexibility, stability, and good fault tolerance capacity [22]. Yuan *et al.* presented a federated extended finite impulse response filter to obtain accurate position information [23]. However, in practical applications, an integrated navigation

The associate editor coordinating the review of this manuscript and approving it for publication was Salvatore Surdo¹.

system always consists of multi-rate sensors. Thus, Federated Filter needs time registration before fusing the information [24], which complicates the fusion process. Even if the federated filter in [25] improved the time registration process, it still takes some time.

In a multi-sensor integrated navigation system, there are always some cases when some sensors are unavailable. It requires the navigation system to react quickly and recover the navigation accuracy in a short time. Hailiang *et al.* presented a novel robust fault-tolerant federated filter to improve the stability of the navigation system [26]. Kai-wei *et al.* proposed new strategies for the integrated navigation system to achieve Simultaneous Localization and Mapping (SLAM), especially in GNSS challenging environments where GNSS signals are blocked [27]. This work only considers the situation that the GNSS signal is blocked, and the fusion framework needs reconstruction when the sensor signal is lost. Wei *et al.* presented an integrated positioning methodology for train navigation application to solve the problem of positioning during BeiDou navigation satellite system (BDS) outages [28]. This work utilizes a switch module to transfer between two navigation systems with and without BDS signal, which means the navigation system needs to be reconstructed once a specific sensor is unavailable. Although the federated filter is exploited for its flexibility and capability in fault toleration, it needs to reconstruct its framework when sensors are added or removed. Therefore, it is worthwhile to propose a multi-sensor information fusion method to achieve real-time adaptive information fusion.

A recent study in monocular visual-inertial navigation has shown that optimization-based approaches outperform filtering methods in terms of accuracy [29]. As one of the optimization-based methods, the factor graph has recently been used in SLAM and navigation problems. Since 2006, Michael *et al.* have presented a series of Simultaneous Smoothing and Mapping (SAM) algorithms for robot SLAM problem [30]–[33]. Sven *et al.* applied a factor graph to the Unmanned Aerial Vehicle (UAV) navigation system [34]. Inspired by the idea of SAM algorithms, Han-Pang *et al.* proposed an integrated navigation system based on a factor graph for a robot, which includes 57 different sensors [35]. More recently, David *et al.* employed the context of so-called “Dynamic Networks” to fuse the processing of the image, raw inertial, and GNSS observations for UAV applications [36]. Qinghua *et al.* presented a factor graph-based navigation method for Micro Unmanned Aerial Vehicle (MUAV) [37]. Xiao-kai *et al.* proposed an INS/Odometer integrated navigation algorithm based on factor graph for the vehicle integrated navigation system, but there are only two sensors in the system [38]. Although it has been shown that factor graph-based methods have good performance in processing asynchronous data, the current work mostly focuses on small-scenario, low-dynamic SLAM problems and is almost verified through simulation. There is no real-world large-scenario experiment with more than three sensors so far.

This paper proposes a multi-sensor information fusion approach based on a factor graph for integrated navigation systems. Specifically, we make the following contributions.

- We present a method that uses a factor graph for fast computation of the navigation state estimation. The first step towards this goal is constructing the factor graph framework for the multi-sensor navigation system. We analyze the different sensors’ measurement models and formulate corresponding factors. Then we build the factor graph framework for the integrated navigation systems in the experiments.
- We perform two real-world experiments to verify the proposed method’s effectiveness in small and large scenarios. One is based on the KITTI dataset, and the other one is based on a dataset collected by our laboratory.
- We investigate the sensor plug and play experiment to prove that the proposed method can react quickly in different environments. For example, when the GPS signal is poor or unavailable in some cases, the proposed method can continue fusing other available sensor information in a short time.

The rest of the paper is organized as follows. In Section II, we present a brief introduction about the factor graph and Bayesian maximum a posteriori. Section III introduces the multi-sensor information fusion method based on the factor graph. Experimental results are presented and analyzed in Section IV, followed by the conclusions summarized in Section V.

II. BACKGROUND

This section introduces the concepts of factor graph and Bayesian maximum a posteriori used to build the navigation system’s factor graph framework.

A. FACTOR GRAPH

A factor graph is a bipartite graph representing the factorization of a function. Given a factorization of a function $g(\theta_1, \dots, \theta_n)$:

$$g(\theta_1, \dots, \theta_n) = \prod_{i \in I} f_i(\Theta_i), \quad (1)$$

where $\Theta_i \subseteq \{\theta_1, \dots, \theta_n\}$, the corresponding factor graph $G = (\Theta, F, E)$ consists of variable nodes $\Theta = \{\theta_1, \dots, \theta_n\}$, factor nodes $F = \{f_1, \dots, f_m\}$, and edges E . The edges depend on the factorization i.e., there is an undirected edge between factor node f_i and variable node θ_j iff $\theta_j \in \Theta_i$.

For example, consider a function that can be factorized as:

$$g(\theta_1, \theta_2, \theta_3, \theta_4) = f_1(\theta_1)f_2(\theta_2, \theta_3, \theta_4)f_3(\theta_3), \quad (2)$$

with a corresponding factor graph shown in Fig. 1. In the figure, each circle represents a variable node and each black dot represents a factor node. The line between circle and dot indicates the relationship between the corresponding variable and factor node.

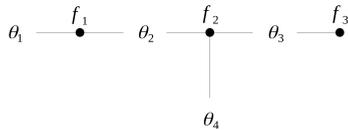


FIGURE 1. An example of a factor graph.

B. BAYESIAN MAXIMUM A POSTERIORI

In Bayesian statistics, a maximum a posteriori (MAP) probability estimate is an estimate of an unknown quantity that equals the mode of the posterior distribution [39]. The MAP can be used to obtain a point estimate of an unobserved quantity based on empirical data. It is closely related to the method of maximum likelihood (ML) estimation. However, it employs an augmented optimization objective that incorporates a prior distribution (that quantifies the additional information available through prior knowledge of a related event) over the quantity we want to estimate. MAP estimation can, therefore, be seen as a regularization of the maximum likelihood estimation.

Let us assume that we want to estimate an unobserved population parameter θ based on observations x . Let f be the sampling distribution of x , so that $f(x|\theta)$ is the probability of x when the underlying population parameter is θ . Then the function $\theta \mapsto f(x|\theta)$ is known as the likelihood function and the estimate $\hat{\theta}_{MLE}(x) = \text{argmax}_{\theta} f(x|\theta)$ is the maximum likelihood estimate of θ .

Next, assume that a prior distribution g over θ exists. So the θ can be treated as a random variable in Bayesian statistics. The posterior distribution of θ can be calculated using Bayes' theorem:

$$\theta \mapsto f(\theta|x) = \frac{f(x|\theta)g(\theta)}{\int_{\Theta} f(x|\vartheta)g(\vartheta)d\vartheta}, \tag{3}$$

where g is density function of θ , Θ is the domain of g .

The method of maximum a posteriori estimation then estimates θ as the mode of the posterior distribution of this random variable:

$$\begin{aligned} \hat{\theta}_{MAP}(x) &= \text{argmax}_{\theta} f(\theta|x) = \text{argmax}_{\theta} \frac{f(x|\theta)g(\theta)}{\int_{\Theta} f(x|\vartheta)g(\vartheta)d\vartheta} \\ &= \text{argmax}_{\theta} f(x|\theta)g(\theta), \end{aligned} \tag{4}$$

where the denominator of the posterior distribution (so-called marginal likelihood) is always positive and does not depend on θ and therefore plays no role in the optimization.

III. INFORMATION FUSION METHOD BASED ON FACTOR GRAPH

Based on the factor graph and maximum a posteriori as presented in Section II, we now propose the multi-sensor information fusion method for an integrated navigation system using the factor graph. This section first defines the optimal navigation solution as a maximum a posteriori estimation of the navigation state. Second, we construct the factor graph framework for an integrated navigation system by analyzing measurement models in various navigation sensors and formulating corresponding sensor factors. Furthermore, we utilize the Gaussian-Newton iteration method to solve

the nonlinear optimization problem. As a result, the proposed method can fuse asynchronous, nonlinear multi-sensor information efficiently.

A. MAXIMUM A POSTERIORI OF NAVIGATION STATE

Based on the idea of factor graph mentioned in Section II-A, we can construct a factor graph for integrated navigation system. In the graph, factor nodes and variable nodes stand for sensor measurements and navigation states, respectively. By calculating the maximum a posteriori estimation of joint probability distribution function for navigation states in a period of time, we can get the optimal estimate of navigation states with all available asynchronous multi-sensor data. The state variable set with the maximum a posteriori probability is considered as follows:

$$\hat{X}_{0:k} = \text{argmax}_{X_{0:k}} p(X_{0:k}|Z_{1:k}), \tag{5}$$

where $X_{0:k}$ is the state variable set from t_0 till t_k , $Z_{1:k}$ is the measurement set from all available sensors till t_k , and $\hat{X}_{0:k}$ is the maximum a posteriori estimation calculated by maximizing the right side of (5). According to the Bayes formula, (5) can be factorized as:

$$\begin{aligned} &p(X_{0:k}|Z_{1:k}) \\ &= \frac{p(Z_k|X_{0:k})p(X_k|X_{k-1})}{p(Z_k)} \cdot p(X_{0:k-1}|Z_{1:k-1}) \\ &= \frac{p(Z_k|X_{0:k})p(X_k|X_{k-1})}{p(Z_k)} \cdot \frac{p(Z_{k-1}|X_{0:k-1})p(X_{k-1}|X_{k-2})}{p(Z_{k-1})} \\ &\quad \cdot p(X_{0:k-2}|Z_{1:k-2}) \\ &= \frac{p(Z_k|X_{0:k})p(X_k|X_{k-1})}{p(Z_k)} \cdot \frac{p(Z_{k-1}|X_{0:k-1})p(X_{k-1}|X_{k-2})}{p(Z_{k-1})} \\ &\quad \cdot \dots \cdot \frac{p(Z_1|X_{0:1})p(X_1|X_0)}{p(Z_1)} \cdot p(X_0) \\ &= p(X_0) \prod_{i=1}^k \frac{p(X_i|X_{i-1})p(Z_i|X_{0:k,i})}{p(Z_i)}, \end{aligned} \tag{6}$$

And because the joint probability distribution function can be factorized in terms of a priori information and individual process and measurement models. Such a factorization can be finally written as:

$$\begin{aligned} p(X_{0:k}|Z_{1:k}) &= p(X_0) \prod_{i=1}^k \left[p(X_i|X_{i-1}, c_{i-1}, Z_i^{IMU})p(c_i|c_{i-1}) \right. \\ &\quad \left. \prod_{Z_i^j \in Z_i \setminus Z_i^{IMU}} p(Z_i^j|X_{0:k,i}^j) \right], \end{aligned} \tag{7}$$

where $p(X_0)$ is the prior information, $p(X_i|X_{i-1}, c_{i-1}, Z_i^{IMU})$, $p(c_i|c_{i-1})$ are the probability distribution of navigation state and calibration parameter in the inertial navigation system, respectively. Finally, $p(Z_i^j|X_{0:k,i}^j)$ is the probability distribution of other navigation sensors' measurements. For example, if Z_i^j is a GPS measurement, $X_{0:k,i}^j$ is simply $\{X_i\}$. This factorization process is similar to that in the factor graph - from a global function to the product of local functions. Therefore, it is reasonable and feasible to apply factor graph to multi-sensor information fusion in integrated navigation system.

In the factor graph, assume that $f(\cdot)$ represents the local probability distribution function, then (6) can be written as:

$$p(X_{0:k}|Z_{1:k}) \propto \prod_i^k f_i(X_{0:k,i}), \quad (8)$$

where $X_{0:k,i}$ is the subset of the state set $X_{0:k}$, local function f_i is related to error function $err(X_{0:k,i}, Z_i)$. So f_i can be formulated as:

$$f_i(X_{0:k,i}) = d(err_i(X_{0:k,i}, Z_i)), \quad (9)$$

where $d(\cdot)$ is the cost function.

Assuming a Gaussian noise model, (8) can be written as:

$$f_i(X_{0:k,i}) = \exp\left(-\frac{1}{2} \left\| err_i(X_{0:k,i}, Z_i) \right\|_{\Sigma_i}^2\right), \quad (10)$$

where $\|\cdot\|_{\Sigma}^2$ denotes the squared Mahalanobis distance, and Σ_i is the measurement noise covariance matrix at time t_i . So, calculating the maximum a posteriori estimation in (5) is equal to minimizing $\sum_i^k \left\| err_i(X_{0:k,i}, Z_i) \right\|_{\Sigma_i}^2$:

$$\begin{aligned} \hat{X}_{0:k} &= \operatorname{argmax}_{X_{0:k}} p(X_{0:k}|Z_{1:k}) \\ &= \operatorname{argmin}_{X_{0:k}} \frac{1}{2} \sum_i^k \left\| err_i(X_{0:k,i}, Z_i) \right\|_{\Sigma_i}^2. \end{aligned} \quad (11)$$

In practical navigation system, the error function $err(X_{0:k,i}, Z_i)$ can be represented as $Z_i - H_i(X_{0:k,i})$, where $H_i(\cdot)$ is the measurement model of the corresponding sensors to predict sensor measurements given a set of states. Then the error function captures the error between the predicted measurements $H_i(X_{0:k,i})$ and the actual measurements Z_i received from sensors.

B. FACTOR GRAPH FRAMEWORK FOR INTEGRATED NAVIGATION SYSTEM

Based on the formulations given in Section III-A, we can construct the factor graph framework for different integrated navigation systems. In the framework, the factor nodes are local functions $f(\cdot)$ in (7) and the variable nodes are navigation states X_0, X_1, \dots, X_k and inertial calibration parameters c_0, c_1, \dots, c_k . Since different sensors have different measurement models, we need to present a distinct factor node formulation for each sensor. Considering the sensors used in our experiments, we analyze measurement models in IMU, Global Positioning System (GPS), BeiDou Navigation Satellite System (BDS), baro-altimeter, visual odometer, and odometer.

1) IMU FACTOR

For inertial measurement unit, the time evolution of the navigation state X can be described by the following continuous nonlinear differential equation:

$$\dot{X} = H_c^{IMU}(X, f_b, \omega_b), \quad (12)$$

where f_b and ω_b are the specific force and the angular acceleration, respectively, measured by the inertial sensors in body frame.

The IMU calibration parameters represented by c are used to compensate for the bias error of IMU measurement f_b, ω_b according to the assumed IMU error model. It is usually estimated at the same time when the navigation state is estimated. Furthermore, the time evolution of c can be described as follows:

$$\dot{c} = g_c(c). \quad (13)$$

A given IMU measurement $Z_i^{IMU} = \{f_i^b, \omega_i^b\}$ relates the navigation states at two consecutive time instances t_{i-1} and t_i . The discrete representation of (12) and (13) is:

$$\begin{aligned} X_i &= H^{IMU}(X_{i-1}, c_{i-1}, Z_i^{IMU}) + N^{IMU} \\ c_i &= g(c_{i-1}) + N^{bias}, \end{aligned} \quad (14)$$

where navigation state X_i is predicted by current inertial measurement Z_i^{IMU} and previous state and calibration parameter X_{i-1}, c_{i-1} . Calibration parameter c_i is updated on c_{i-1} .

Based on the representation of error function and (9), the conventional IMU factor f^{IMU} and bias factor f^{bias} can be given as:

$$\begin{aligned} f^{IMU}(X_{i-1}, c_{i-1}, X_i) &= d(X_i - H^{IMU}(X_{i-1}, c_{i-1}, Z_i^{IMU})) \\ &= \exp\left(-\frac{1}{2} \left\| X_i - H^{IMU}(X_{i-1}, c_{i-1}, Z_i^{IMU}) \right\|_{\Sigma_i}^2\right), \end{aligned} \quad (15)$$

$$\begin{aligned} f^{bias}(c_{i-1}, c_i) &= d(c_i - g(c_{i-1})) \\ &= \exp\left(-\frac{1}{2} \left\| c_i - g(c_{i-1}) \right\|_{\Sigma_i}^2\right), \end{aligned} \quad (16)$$

where we can find that the IMU factor is related with the current state, previous state and bias, while the bias factor is related with the current and previous bias. The corresponding factor graph with IMU and bias factors is shown in Fig. 2 (a). In general, estimating the state variables at IMU frequency that is very high is infeasible. Instead, consecutive IMU measurements can be combined into an equivalent IMU factor [29], [40], relating between two distant navigation and calibration nodes, which can reduce the cost of computational complexity dramatically by decreasing the number of variable nodes in factor graph, as shown in Fig. 2 (b).

2) GPS FACTOR

The GPS measurement equation is given by:

$$Z_i^{GPS} = H^{GPS}(X_i) + N^{GPS}, \quad (17)$$

where N^{GPS} is the measurement noise and H^{GPS} is the measurement function, relating the measurement Z_i^{GPS} to state X_i . The above equation defines the GPS factor f^{GPS} :

$$\begin{aligned} f^{GPS}(X_i) &= d(Z_i^{GPS} - H^{GPS}(X_i)) \\ &= \exp\left(-\frac{1}{2} \left\| Z_i^{GPS} - H^{GPS}(X_i) \right\|_{\Sigma_i}^2\right), \end{aligned} \quad (18)$$

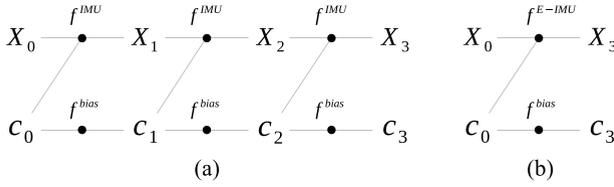


FIGURE 2. Examples of a factor graph in inertial navigation system: (a) Factor graph with conventional IMU and bias factors; (b) An equivalent factor graph to (a) when using one equivalent IMU factor instead of three conventional IMU factors between X_1 and X_3 .

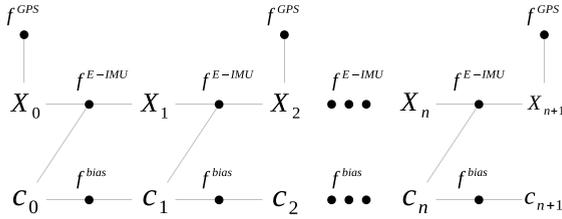


FIGURE 3. Example of a factor graph in INS/GPS integrated navigation system.

which is only related with the current navigation state X_i . Fig. 3 shows a factor graph of an INS/GPS integrated navigation system in which IMU and GPS measurements are added into the graph with different frequencies. Note, BDS factor and baro-altimeter factor equations are similar to the GPS factor equation.

3) VISUAL ODOMETER FACTOR

Visual odometer has become popular in current navigation systems [41] as it allows us to estimate the relative transformation between two stereo frames, assuming a known baseline. Based on this relative transformation information, we can get the current position of the target. The visual odometer measurement model can be given by:

$$Z_{ij}^{VO} = H^{VO}(X_i, X_j) + N^{VO}, \quad (19)$$

where Z_{ij}^{VO} is the measurement based on state set $\{X_i, X_j\}$ and N^{VO} is measurement noise. The corresponding visual odometer factor is:

$$\begin{aligned} f^{VO}(X_i, X_j) &= d(Z_{ij}^{VO} - H^{VO}(X_i, X_j)) \\ &= \exp\left(-\frac{1}{2} \left\| Z_{ij}^{VO} - H^{VO}(X_i, X_j) \right\|_{\Sigma}^2\right). \end{aligned} \quad (20)$$

Fig. 4 is an example of a factor graph in IMU/Visual odometer integrated navigation system. Odometer factor equation is similar to visual odometer factor equation.

C. COST FUNCTION LINEARIZATION

As many sensor measurements in integrated navigation system appear to have a nonlinear character, (10) is a nonlinear least square problem. We use the Gaussian-Newton iteration method to solve this optimization problem. By Taylor-series expansion, we can transform this nonlinear problem into

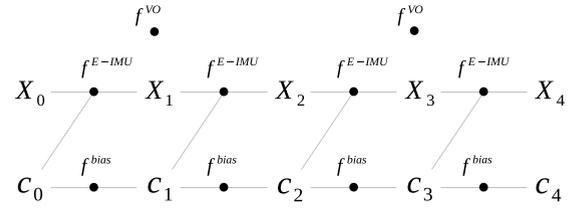


FIGURE 4. Example of a factor graph in INS/Visual Odometer integrated navigation system.

Algorithm 1 Navigation State Update Process Based on Factor Graph

Input: original state variable set $X_{0:k}$, measurements so far $Z_{1:k}$, threshold η

Output: maximum a posteriori estimation of state variable set $\hat{X}_{0:k}$

Initialize: $X_{0:k} \leftarrow \mathbf{0}$

for $t \leftarrow 1$ to k **do**

$$f^{IMU}(X_t, X_{t-1}, c_t) \leftarrow d(X_t -$$

$$H_t^{IMU}(X_{t-1}, c_{t-1}, Z_t^{IMU}))$$

$$f^{bias}(c_t, c_{t-1}) \leftarrow d(c_t - g(c_{t-1}))$$

for $j \leftarrow 1$ to $N_{sensors}$ **do**

$$f^{sensor_j} \leftarrow d(Z_t^j - H_t^j(X_{0:k,t}^j))$$

end for

end for

while $\left\| J(\hat{X}_{0:k})\Delta_{0:k} - b(\hat{X}_{0:k}) \right\|^2 > \eta$ **do**

$$J(\hat{X}_{0:k}), b(\hat{X}_{0:k}) \leftarrow \sum_{t=1}^k \left\| H_t(X_{0:k,t}) - Z_t \right\|_{\Sigma}^2$$

$$\hat{\Delta}_{0:k} \leftarrow \operatorname{argmin} \left\| J(\hat{X}_{0:k})\Delta_{0:k} - b(\hat{X}_{0:k}) \right\|^2$$

$$\hat{X}_{0:k} \leftarrow \hat{X}_{0:k} + \hat{\Delta}_{0:k}$$

end while

linear problem and get the optimal state increment estimation:

$$\hat{\Delta}_{0:k} = \operatorname{argmin}_{\Delta_{0:k}} \left\| J(\hat{X}_{0:k})\Delta_{0:k} - b(\hat{X}_{0:k}) \right\|^2, \quad (21)$$

where $J(\hat{X}_{0:k})$ is the sparse Jacobian matrix, $\Delta_{0:k}$ is the state increment matrix till time t_k , and $b(\hat{X}_{0:k})$ is the residual matrix given the measurements so far. Once $\hat{\Delta}_{0:k}$ is calculated, the new optimal state estimate can be updated by $\hat{X}_{0:k} + \Delta_{0:k}$, which is then used as the linearization point in the next iteration. Alg. 1 presents the pseudocode for the navigation state updating algorithm based on factor graph. In Alg. 1, Z_t^{IMU} is the IMU measurement at time t and Z_t^j is the measurement of sensor j at time t . Z_t is all of the sensors' measurement at time t so $Z_t^{IMU}, Z_t^j \in Z_t$. Similarly, H_t^{IMU} is the function relating between the measurement Z_t^{IMU} and previous state X_{t-1} to the current state X_t . H_t^j is the measurement function of sensor j relating between the measurement Z_t^j to the state X_t and $H_t^{IMU}, H_t^j \in H_t$. The calculating process of navigation state estimation based on factor graph is as follows:

TABLE 1. Structure parameters of the stereo camera in the KITTI dataset.

Parameters	Description	Value
size	The size of image	1241 pixel × 376 pixel
u_0	x coordinate of principal point in camera	607.1928 pixel
v_0	y coordinate of principal point in camera	185.2157 pixel
f	The focus length of camera	718.8560 m
b	The baseline distance of camera	0.5372 m

TABLE 2. Measurement error of navigation sensors in the KITTI dataset.

Sensors	Error Parameter	Value	Frequency
IMU	Gyro constant bias	0.01 °/h	100 Hz
	Gyro angular random walk	0.0105 °/√h	
	Accelerometer constant bias	167 μg	
	Accelerometer random walk	10 μg/√Hz	
GPS	Position error	5 m, 5 m, 5 m	1 Hz
	Velocity error	0.1 m/s, 0.1 m/s, 0.1 m/s	
Stereo camera	Visual measurement error	0.5 m, 0.5 m	10 Hz

1) Formulate the global cost function $\sum_{t=1}^k \|H_t(X_{0:k,t}) - Z_t\|^2$ based on sensors' measurement function H_t^{IMU} and H_t^j . 2) Calculate the sparse Jacobian matrix and state increment matrix. 3) Get the optimal state increment estimation by minimizing $\|J(\hat{X}_{0:k})\Delta_{0:k} - b(\hat{X}_{0:k})\|^2$. 4) Update the state estimate. 5) Repeat step 2) to 4) until $\|J(\hat{X}_{0:k})\Delta_{0:k} - b(\hat{X}_{0:k})\|^2 > \eta$.

IV. EXPERIMENTS

To verify the proposed multi-sensor information fusion method's performance, we conduct two experiments. The first experiment confirms the navigation accuracy of the proposed method in both small and large scenarios. The second experiment's goal is to prove that our method can achieve sensor plug and play in software, which means when some sensors are unavailable in specific environments, the proposed method can still achieve accurate navigation solutions. Our experiments were run on a single core of an Intel i5-4210 processor with a 2.40 GHz clock rate and 6 GiB of RAM.

A. DATASETS

In the first experiment, two real-world datasets are used for small scenario and large scenario, respectively:

- KITTI. KITTI Vision Benchmark Suite [42] is a dataset captured from an autonomous vehicle platform "Anniway" by traveling around different kinds of areas in Karlsruhe, Germany. These areas can be classified into four categories: city, residential, road, and campus. This platform is equipped with a stereo camera, IMU, and a differential GPS/INS system. The differential GPS/INS data provides a highly accurate ground truth position. The structure parameters of the stereo camera are shown in Table. 1. The KITTI dataset provides stereo calibration information, so no additional calibration nodes were required for the stereo factors. The measurement errors of every sensor in this navigation system are shown in Table. 2. To verify the performance



FIGURE 5. Hardware structure of the ground vehicle navigation system.

in the small scenario, we chose a part of a residential dataset that can be downloaded from [43] to carry out the first experiment. The length of the entire trajectory was about 700 m, and the driving time was 100 s.

- SX. To further test the proposed method's performance on the real-world large-scenario environment, we captured real-world data by driving a vehicle in the middle area of Shanxi province, China, and we called this dataset "SX". The vehicle was equipped with IMU, BDS, odometer, baro-altimeter, and a differential GPS/INS system, which provided the ground truth data. The navigation system's hardware architecture to obtain this dataset is shown in Fig. 5, and the parameters of these sensors are shown in Table. 3. The whole trajectory was about 27,383 m long, and the whole driving time was 1359 s. The first half trajectory was nearly a straight line that cannot verify the proposed method's performance properly. Thus, we chose the second half trajectory (13,582 m) containing several turns to analyze the performance.

B. ACCURACY EXPERIMENT

The Federated Filter (FF) has been widely used in integrated navigation systems, which is a near-optimal estimator for

TABLE 3. Measurement error of navigation sensors in the SX dataset.

Sensors	Error Parameter	Value	Frequency
IMU	Gyro constant bias	0.015 °/h	100 Hz
	Gyro angular random walk	0.001 °/√h	
	Accelerometer constant bias	50 μg	
	Accelerometer random walk	5 μg/√Hz	
BDS	Position error	4 m, 4 m, 6 m	1 Hz
Odometer	Measurement noise	8 m	1 Hz
Baro-altimeter	Measurement noise	6 m	5 Hz

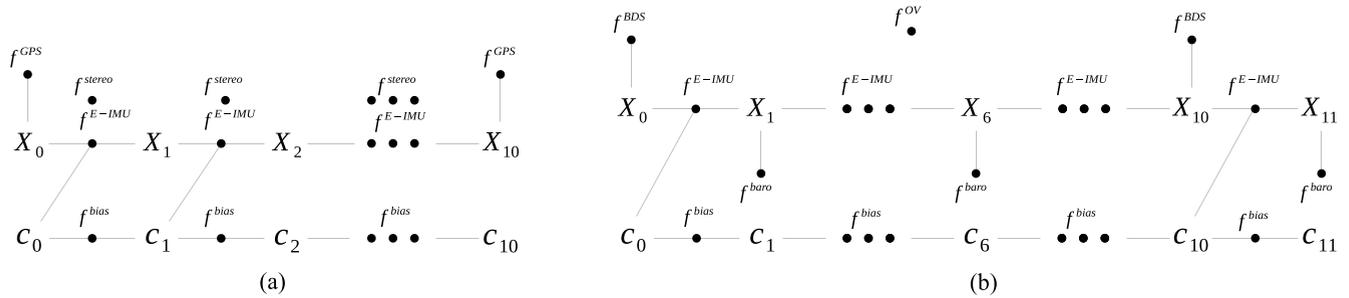


FIGURE 6. The multi-sensor fusion framework based on factor graph for two datasets: (a) The factor graph of the navigation system based on the KITTI dataset. In this framework, based on the update frequency of GPS, stereo cameras, and IMU, between navigation state X_0 and X_{10} , there are two GPS factors, ten visual odometry factors, and ten equivalent IMU factors. Specifically, ten consecutive IMU measurements are combined into an equivalent IMU factor. (b) The factor graph of the navigation system based on the SX dataset. According to the update frequency of sensors in the system, between state X_0 and X_{10} there are two BDS factors, ten equivalent IMU factors, one visual odometry factor, and two baro-altimeter factors. Same as the KITTI dataset, ten consecutive IMU measurements are combined into an equivalent IMU factor in this system.

decentralized, multi-sensor data fusion [44]. To verify the performance of the proposed method from the accuracy perspective, we compared the navigation results of the factor graph method and FF. This experiment is based on two datasets mentioned in Section IV-A, one for small scenarios and the other for large scenarios. According to the sensor information given in Table. 2 and Table. 3, we can construct the factor graph framework for these two navigation systems, as shown in Fig. 6.

1) RESULTS ON KITTI

Fig. 7 presents the results of the proposed method, FF, and the ground truth in Google Road Map on the KITTI dataset. From the results, it is clear that the proposed method had a smaller error in each turning than FF. Especially in the third and fifth turning, the error in FF is increasing for a considerable period. This is caused by the uncertainty of the system model in FF during the turning periods. Furthermore, the vehicle is easily affected by the hybrid of Gaussian noise and non-Gaussian noise in the natural environment, so there are some deviations in both methods.

Fig. 8 illustrates the position and velocity error of these two methods. Recall, the closer the curve is to zero, the better is the effect of the estimation. From the position error curve, we can see that FF has a more considerable fluctuation in x and y directions, while in the vertical direction, FF has a smaller error. Finally, from the velocity error curve, it is clear that FF fluctuates more in all directions.

To quantify the navigation error, we calculate root-mean-square error (RMSE) of position and velocity for these two methods, as shown in Table. 4. The improvement of the factor graph method is obvious: (1) it can be seen that the factor graph method has higher position accuracy than FF in x and

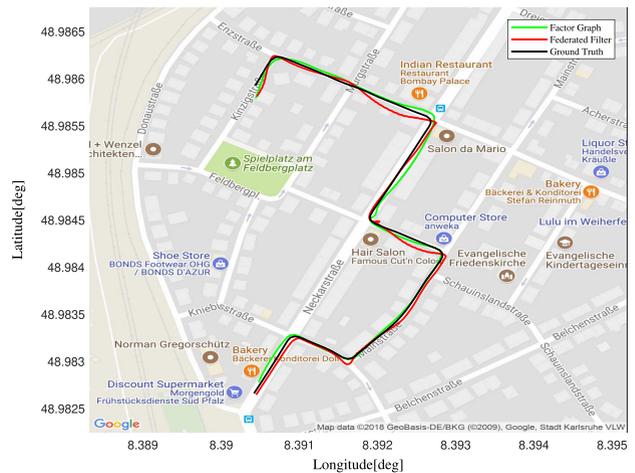


FIGURE 7. Trajectories for the ground truth, factor graph method, and Federated Filter on KITTI dataset.

y directions, reducing to less than 1.5 m in RMSE. Simultaneously, in the vertical direction, FF has a smaller error; (2) as for the velocity error, the factor graph method improves in all directions. More precisely, in horizontal directions, the factor graph method's error is approximately half of FF. Thus, this experiment proves that the factor graph method can reach more precise navigation results in a small scenario than FF.

2) RESULTS ON SX

The navigation results on the SX dataset are shown in Fig. 9. As presented in the enlarged drawing of the first 100s of the trajectory, the proposed method had a considerable drift. We believe this is because, in the proposed algorithm, we initialized the velocity of the experimental vehicle as zero, while the true velocity was not zero (recall, we start fusing

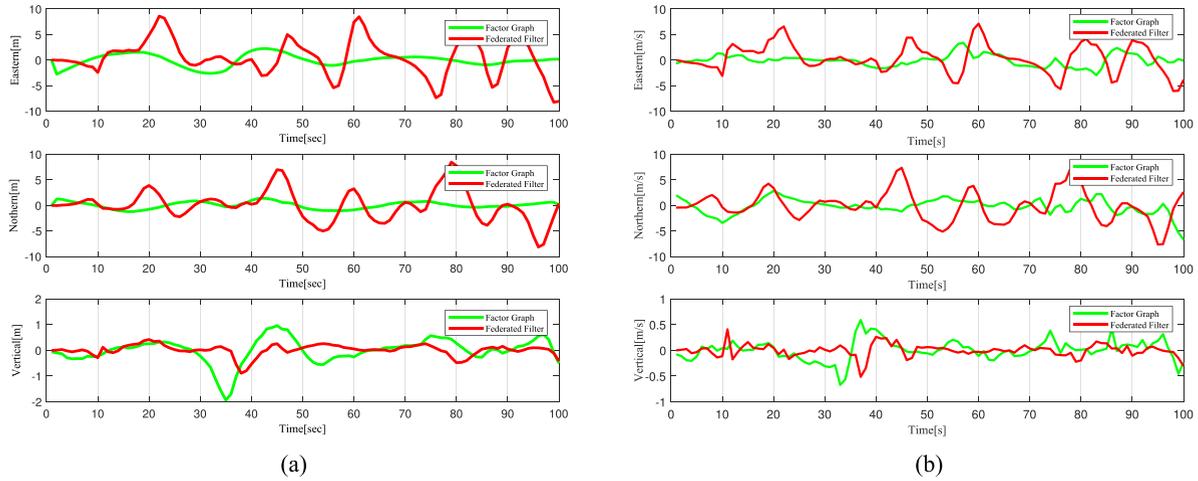


FIGURE 8. Comparison curves of factor graph method and Federated Filter on the KITTI dataset: (a) Position error curves; (b) Velocity error curves.

TABLE 4. Navigation error of factor graph method and Federated Filter on the KITTI dataset.

Error Type	Position RMSE(m)			Velocity RMSE(m/s)		
	Eastern	Northern	Vertical	Eastern	Northern	Vertical
Factor Graph	1.1630	0.6650	0.4798	1.1188	1.5227	0.1966
Federated Filter	3.5060	3.0282	0.2171	2.7614	2.9404	0.1200

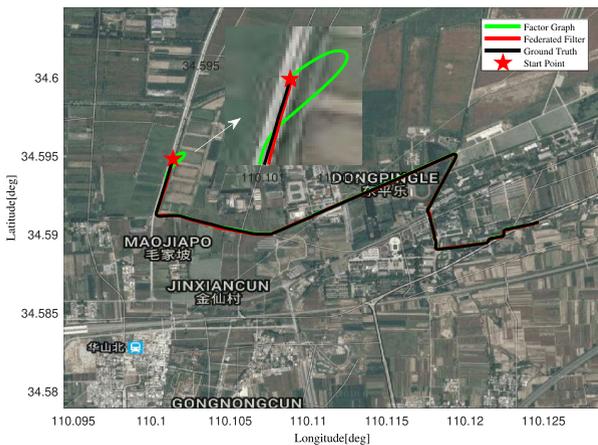


FIGURE 9. Trajectories for the ground truth, factor graph method, and Federated Filter on the SX dataset.

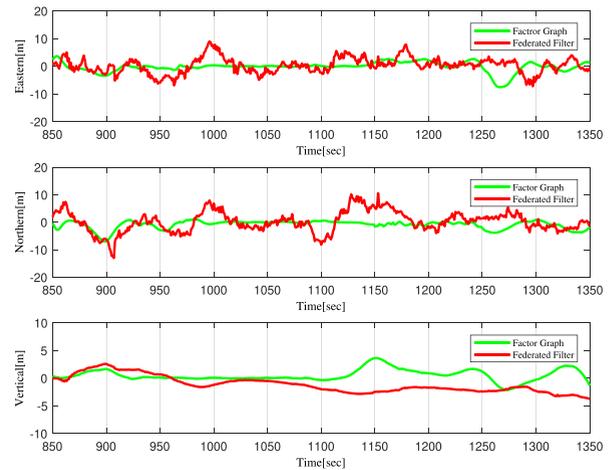


FIGURE 10. Position error curves of two methods on the SX dataset.

the measurements from the middle of the trajectory, not the beginning). In the rest of the trajectory, the proposed method has a slight deviation from the ground truth.

The position error of the two methods after the first 100 s is illustrated in Fig. 10 and Table. 5, where we can find that the proposed method has better performance than FF. Consistent with the KITTI dataset results, in x and y directions, the factor graph method has smaller errors than FF, and its RMSE reduces to less than 2 m. Furthermore, the improvement in horizontal directions is better than the vertical direction.

From the experimental results, we can find that the SX dataset results are somewhat worse than the KITTI dataset because, in this experiment, there is no high-precise visual odometer to be combined to get the navigation result. However, we can still conclude that the proposed method also shows superiority over FF in the large scenario.

TABLE 5. Navigation error of factor graph method and Federated Filter on the SX dataset.

Error Type	Position RMSE(m)		
	Eastern	Northern	Vertical
Factor Graph	1.8364	1.7254	1.1649
Federated Filter	2.6506	3.6609	1.7277

C. SENSOR PLUG AND PLAY EXPERIMENT

Since there are always cases when some particular sensors are unavailable, the navigation system needs to react quickly and recover the required navigation accuracy in a short time. This experiment uses the second dataset - SX to perform a sensor plug and play experiment. We divided the whole trajectory into five sections, and in each section, we assume there is 20 s of BDS signal loss. The exact starting and ending points of these five-time slots are shown in Table. 6. We calculate

TABLE 6. Starting and ending time of five time slots without BDS Signal on the SX dataset.

Time Slot	Starting Time(s)	Ending Time(s)
1	858	878
2	936	956
3	969	989
4	1 205	1 225
5	1 270	1 290

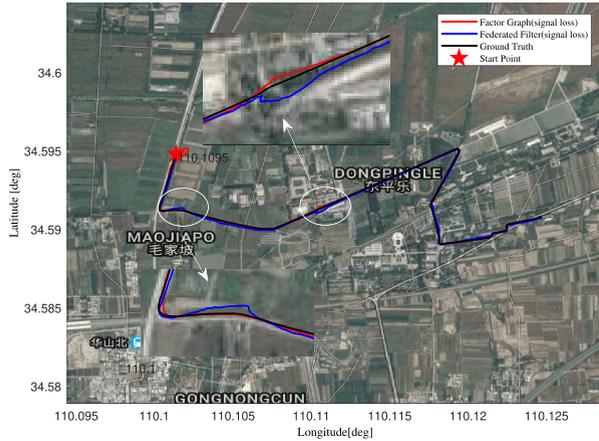


FIGURE 11. Navigation results of factor graph method and Federated Filter with signal loss on the SX dataset.

the navigation solution with BDS signal loss utilizing FF and the factor graph method. The navigation results are shown in Fig. 11.

By enlarging two navigation results in the first and third signal loss time slots (858 s to 878 s and 969 s to 989 s), we can find that the factor graph method can react in a short time and reduce navigation error quickly while FF needs more time to regain a precise result. Especially when the signal is lost, the error of FF increases immediately to a significant degree. The position error with signal loss of two methods is shown in Fig. 12, and the RMSE results are presented in Table. 7. It can be seen that the performance of both methods degrades with the signal loss, but the factor graph method drops less in navigation accuracy, and it can still guarantee accuracy around 3 m. This experiment shows that: (1) for the factor graph-based method, the navigation accuracy is worse without the BDS signal at the beginning, but in general, the navigation results are still precise; (2) FF gets larger position error than the factor graph method when there are some sensors unavailable; (3) factor graph method is more effective for multi-sensor information fusion in various environments.

This experiment can also illustrate the stability of the proposed method. RMSE is a frequently used measure of the differences between values predicted by a model or an estimator, so it is reasonable to use RMSE to compare the stability of the methods. It can be seen from Table. 7 that when signal losing occurs, position RMSE of the factor graph based method increases by 63.84%, 82.41%, 46.77% in eastern, northern, and vertical directions respectively, and for the federated filter, it increases by 160.65%, 172.36%, 21.60% in

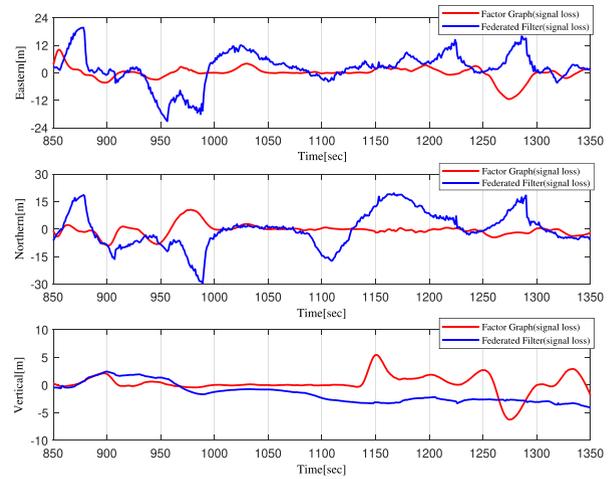


FIGURE 12. Position error curve of two methods with signal loss on the SX dataset.

TABLE 7. Navigation error of two methods with/without BDS signal loss on the SX dataset.

With/without signal loss	Position RMSE(m)		
	Eastern	Northern	Vertical
Factor Graph	1.8364	1.7254	1.1649
Factor Graph-with signal loss	3.0088	3.1473	1.7097
Federated Filter	2.6506	3.6609	1.7277
Federated Filter-with signal loss	6.9088	9.9710	2.1008

the three directions respectively. Therefore, the factor graph method is more stable than FF in a complex environment.

D. TIMING PERFORMANCE

In addition to navigation accuracy, the timing performance of the proposed method is also considered to evaluate and analyze the effectiveness of the proposed method comprehensively.

The experiments of the two methods were carried out 50 times for each dataset. The average computational time is shown in the Table. 8. It can be seen from Table. 8 that the proposed method takes less fusion time for the navigation solution than the federated filter on both two datasets. The calculated fusion time of the two methods is shown in Fig. 13. It is obvious that the computational cost of the proposed method is much smaller than that in the federated filter. Specifically, the average fusion time of the factor graph is 62.05% and 66.16% of the federated filter on KITTI and SX datasets respectively. It can also be seen intuitively from Fig. 13 that under the situation with signal loss, the factor graph method can cost less time to recover, i.e., compared with the fusion time without signal loss, it takes 1.18% more time for factor graph based method to get the navigation solution while federated filter takes 7.27% more time. That is because once there are new navigation measurements added to the integrated navigation system, the proposed method in this paper only needs to add corresponding factors, while the federated filter needs to rebuild the algorithm model. Therefore, the proposed method not only can process data more efficiently but also have better expansibility.

TABLE 8. Time consumptions of two methods based on the two datasets.

	Time(s)		
	KITTI	SX	SX (signal loss)
Factor Graph	3.26	19.87	20.10
Federated Filter	5.25	30.03	32.22

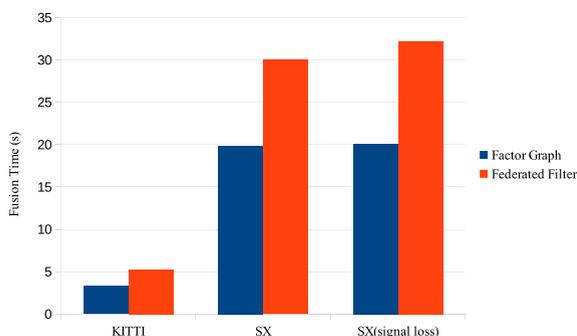


FIGURE 13. Comparison of fusion time of two methods based on KITTI dataset and SX dataset (with/without signal loss).

V. CONCLUSION

In this paper, we study the multi-sensor information fusion method in the integrated navigation systems. To perform efficient information fusion, we proposed an approach based on the factor graph topology. The constructed factor graph framework can clearly illustrate the relationship between the navigation states and sensor measurements, making it easier to calculate the optimal navigation results with asynchronous data. By linearizing the cost function, we can also fuse nonlinear information. Furthermore, when some sensors become unavailable, the algorithm’s only change is to stop adding the corresponding factors to the graph, which is an outstanding advantage compared with conventional filtering methods. Two experiments have been performed to prove the proposed method’s effectiveness, and the results suggest that our proposed method outperforms the popular Federated Filter. Additionally, the sensor plug and play experiment further validates the factor graph method’s good performance in a complex environment. Due to the limitations of experimental conditions, we only fuse four navigation sensors in this paper. In future work, we plan to increase the number of navigation sensors in the integrated navigation system.

REFERENCES

[1] D. Floreano, J. Godjevac, A. Martinoli, F. Mondada, and J. D. Nicoud, “Design, control, and applications of autonomous mobile robots,” in *Advances in Intelligent Autonomous Systems*. Dordrecht, The Netherlands: Springer, 1999, pp. 159–186.

[2] M. B. Alatise and G. P. Hancke, “A review on challenges of autonomous mobile robot and sensor fusion methods,” *IEEE Access*, vol. 8, pp. 39830–39846, 2020, doi: 10.1109/access.2020.2975643.

[3] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, “Information fusion in navigation systems via factor graph based incremental smoothing,” *Robot. Auto. Syst.*, vol. 61, no. 8, pp. 721–738, Aug. 2013, doi: 10.1016/j.robot.2013.05.001.

[4] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960, doi: 10.1115/1.3662552.

[5] H. W. Sorenson and A. R. Stubberud, “Non-linear filtering by approximation of the a posteriori density,” *Int. J. Control*, vol. 8, no. 1, pp. 33–51, Jul. 1968, doi: 10.1080/00207176808905650.

[6] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[7] E. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proc. IEEE Adapt. Syst. Signal Process., Commun., Control Symp.*, Oct. 2000, pp. 153–158.

[8] Y. Sunahara and K. Yamashita, “An approximate method of state estimation for non-linear dynamical systems with state-dependent noise,” *Int. J. Control*, vol. 11, no. 6, pp. 957–972, Jun. 1970.

[9] C. Leondes, J. Peller, and E. Stear, “Nonlinear smoothing theory,” *IEEE Trans. Syst. Sci. Cybern.*, vol. 6, no. 1, pp. 63–71, 1970.

[10] R. Mehra, “On-line identification of linear dynamic systems with applications to Kalman filtering,” *IEEE Trans. Autom. Control*, vol. 16, no. 1, pp. 12–21, Feb. 1971.

[11] D. Willner, C. Chang, and K. Dunn, “Kalman filter algorithms for a multi-sensor system,” in *Proc. IEEE Conf. Decis. Control Including 15th Symp. Adapt. Processes*, Dec. 1976, pp. 570–574.

[12] S.-L. Sun, “Multi-sensor optimal information fusion Kalman filters with applications,” *Aerosp. Sci. Technol.*, vol. 8, no. 1, pp. 57–62, Jan. 2004.

[13] H. Xiong, J. Tang, H. Xu, W. Zhang, and Z. Du, “A robust single GPS navigation and positioning algorithm based on strong tracking filtering,” *IEEE Sensors J.*, vol. 18, no. 1, pp. 290–298, Jan. 2018.

[14] M. Zhu, F. Yu, S. Xiao, S. Fan, and Z. Wang, “An improved posteriori variance-covariance components estimation applied to unconventional GPS and multiple low-cost imus integration strategy,” *IEEE Access*, vol. 7, pp. 136892–136906, 2019.

[15] J. Speyer, “Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem,” *IEEE Trans. Autom. Control*, vol. 24, no. 2, pp. 266–269, Apr. 1979.

[16] T. S. Chang, “Comments on ‘Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control,’” *IEEE Trans. Autom. Control*, vol. 25, no. 3, pp. 609–610, Jun. 1980.

[17] A. Willsky, M. Bello, D. Castanon, B. Levy, and G. Verghese, “Combining and updating of local estimates and regional maps along sets of one-dimensional tracks,” *IEEE Trans. Autom. Control*, vol. 27, no. 4, pp. 799–813, Aug. 1982.

[18] B. C. Levy, D. A. Castañon, G. C. Verghese, and A. S. Willsky, “A scattering framework for decentralized estimation problems,” *Automatica*, vol. 19, no. 4, pp. 373–384, Jul. 1983.

[19] D. Castanon and D. Teneketzis, “Distributed estimation algorithms for nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 30, no. 5, pp. 418–425, May 1985.

[20] T. Kerr, “Decentralized filtering and redundancy management for multi-sensor navigation,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-23, no. 1, pp. 83–119, Jan. 1987.

[21] G. Bierman and M. Belzer, “A decentralized square root information filter/smoothing,” in *Proc. 24th IEEE Conf. Decis. Control*, Dec. 1985, pp. 1902–1905.

[22] N. A. Carlson, “Federated square root filter for decentralized parallel processors,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 3, pp. 517–525, May 1990.

[23] Y. Xu, G. Tian, and X. Chen, “Enhancing INS/UWB integrated position estimation using federated EFIR filtering,” *IEEE Access*, vol. 6, pp. 64461–64469, 2018.

[24] W. Li, H. Leung, and Y. Zhou, “Space-time registration of radar and ESM using unscented Kalman filter,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 3, pp. 824–836, Jul. 2004.

[25] Z. Jiao, “Research on asynchronous registration method based on improved unequal-interval federated filter,” in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Mar. 2019, pp. 2320–2323.

[26] H. Xiong, R. Bian, Y. Li, Z. Du, and Z. Mai, “Fault-tolerant GNSS/SINS/DVL/CNS integrated navigation and positioning mechanism based on adaptive information sharing factors,” *IEEE Syst. J.*, vol. 14, no. 3, pp. 3744–3754, Sep. 2020.

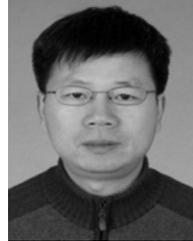
[27] K.-W. Chiang, G.-J. Tsai, H.-J. Chu, and N. El-Sheimy, “Performance enhancement of INS/GNSS/refreshed-SLAM integration for acceptable lane-level navigation accuracy,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 2463–2476, Mar. 2020.

[28] W. Jiang, S. Chen, B. Cai, J. Wang, W. ShangGuan, and C. Rizos, “A multi-sensor positioning method-based train localization system for low density line,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10425–10437, Nov. 2018.

- [29] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proc. Robot. Sci. Syst.*, Jul. 2015.
- [30] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, Dec. 2006.
- [31] M. Kaess, A. Ranganathan, and F. Dellaert, "ISAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.
- [32] O. François and P. Leray, "Learning the tree augmented naive Bayes classifier from incomplete datasets," presented at the 3rd Eur. Workshop Probabilistic Graph. Models, Jan. 2006, pp. 91–98.
- [33] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, Dec. 2011.
- [34] S. Lange, N. Sunderhauf, and P. Protzel, "Incremental smoothing vs. filtering for sensor fusion on an indoor UAV," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 1773–1778.
- [35] H.-P. Chiu, X. S. Zhou, L. Carlone, F. Dellaert, S. Samarasekera, and R. Kumar, "Constrained optimal selection for multi-sensor robot navigation using plug-and-play factor graphs," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 663–670.
- [36] D. A. Cucci, M. Rehak, and J. Skaloud, "Bundle adjustment with raw inertial observations in UAV applications," *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 1–12, Aug. 2017.
- [37] Q. Zeng, W. Chen, J. Liu, and H. Wang, "An improved multi-sensor fusion navigation algorithm based on the factor graph," *Sensors*, vol. 17, no. 3, p. 641, Mar. 2017.
- [38] X.-K. Wei, J. Li, K.-Q. Feng, D.-B. Zhang, Z.-L. Lu, and J.-Q. Li, "Vehicle INS/odometer integrated navigation algorithm based on factor graph," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Oct. 2019, pp. 216–220.
- [39] R. Bassett and J. Deride, "Maximum a posteriori estimators as a limit of Bayes estimators," *Math. Program.*, vol. 174, nos. 1–2, pp. 129–144, Jan. 2018.
- [40] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 61–76, Feb. 2012.
- [41] Y. Yang, Q. Shen, J. Li, Z. Deng, H. Wang, and X. Gao, "Position and attitude estimation method integrating visual odometer and GPS," *Sensors*, vol. 20, no. 7, p. 2121, Apr. 2020.
- [42] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
- [43] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
- [44] H. Xiong, Z. Mai, J. Tang, and F. He, "Robust GPS/INS/DVL navigation and positioning method using adaptive federated strong tracking filter based on weighted least square principle," *IEEE Access*, vol. 7, pp. 26168–26178, 2019.



JING XU received the B.Eng. degree from Shanghai University, China, in 2016, and the M.Eng. degree from Beihang University, China, in 2019. She is currently pursuing the Ph.D. degree in machine learning and cybersecurity with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. Her Ph.D. thesis was on the application of machine learning for cybersecurity and also discusses the security of machine learning methods.



GONGLIU YANG received the Ph.D. degree in precision instrumentation and mechanism from Tsinghua University, Beijing, China, in 2004. He is currently a Professor with the School of Instrumentation and Optoelectronic Engineering, Beihang University. His current research interests include inertial navigation and integrated navigation.



YIDING SUN received the B.E. degree in measurement and control technology and instrumentation from Harbin Engineering University, Harbin, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Instrumentation and Optoelectronics Engineering, Beihang University. His current research interests include inertial navigation and integrated navigation.



STJEPAN PICEK received the Ph.D. degree in computer science from Radboud University, Nijmegen, The Netherlands, and the University of Zagreb, Zagreb, Croatia, in 2015. He was with the COSIC Group, KU Leuven, Leuven, Belgium. He was a Postdoctoral Researcher with the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), Cambridge, MA, USA. He is currently an Assistant Professor with the Cyber Security Research Group, TU Delft, Delft, The Netherlands. His current research interests include cryptography, machine learning, and evolutionary computation.

• • •