

Review

Not peer-reviewed version

A Comprehensive Review of Multi-Source Data Fusion Processing Methods

[Xiaping Ma](#), [Peimin Zhou](#)^{*}, [Xiaoxing He](#), Sheng Zhang

Posted Date: 17 March 2025

doi: 10.20944/preprints202503.1108.v1

Keywords: Multi-source navigation; Fusion processing; LSE; KF; PF; MME



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Review

A Comprehensive Review of Multi-Source Data Fusion Processing Methods

Xiaping Ma ¹, Peimin Zhou ^{1,*}, Xiaoxing He ² and Zhang Sheng ³

¹ a School of Geomatics, Xi'an University of Science and Technology, Xi'an 710054, China

² School of Civil Engineering and Surveying and Mapping Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China

³ Gansu coal geology bureau comprehensive survey team, Gansu 741000, China

* Correspondence: 15336205301@163.com

Abstract: In recent years, significant progress has been made in multi-source navigation data fusion methods, driven by rapid advancements in multi-sensor technology, artificial intelligence (AI) algorithms, and computational capabilities. On one hand, fusion methods based on filtering theory, such as Kalman Filtering (KF), Particle Filtering (PF), and Federated Filtering (FF), have been continuously optimized, enabling effective handling of non-linear and non-Gaussian noise issues. On the other hand, the introduction of AI technologies like deep learning and reinforcement learning has provided new solutions for multi-source data fusion, particularly enhancing adaptive capabilities in complex and dynamic environments. Additionally, methods based on Factor Graph Optimization (FGO) have also demonstrated advantages in multi-source data fusion, offering better handling of global consistency problems. In the future, with the widespread adoption of technologies such as 5G, the Internet of Things, and edge computing, multi-source navigation data fusion is expected to evolve towards real-time processing, intelligence, and distributed systems. So far, Fusion methods mainly include optimal estimation methods, filtering methods, uncertain reasoning methods, Multiple Model Estimation (MME), AI, and so on. To analyze the performance of these methods and provide a reliable theoretical reference and basis for the design and development of a multi-source data fusion system. This paper summarizes the characteristics of these fusion methods and the corresponding adaptation scenarios. These results can provide references for theoretical research, system development, and application in the fields of autonomous driving, unmanned vehicle navigation, and intelligent navigation.

Keywords: multi-source navigation; fusion processing; LSE; KF; PF; MME

1. Introduction

Based on Global Navigation Satellite Systems (GNSS), Inertial Navigation Systems (INS), Ultra-Wideband (UWB) technology, Bluetooth, Wireless Local Area Networks (WLAN), visual sensors, pseudolites (PL), and various other sensors, it is difficult to meet navigation performance requirements when used individually. In particular, GNSS, as a non-autonomous navigation system, has limitations in specific complex environments (such as urban canyons, tunnels, etc.), because its signals are prone to being blocked, interfered with, or shielded. An effective way to enhance the overall performance of a navigation system is to employ integrated navigation technology. This involves using two or more different types of navigation systems to measure and calculate the same navigation information, forming quantitative measurements. From these measurements, the errors of each navigation system are calculated and corrected. By employing a variety of technical means and methods, high accuracy and reliability of navigation and positioning services can be achieved in various scenarios, including seamless indoor and outdoor positioning, environments with electromagnetic interference, and underwater and underground environments. Therefore, multi-

source data fusion positioning is based on the collaboration of multiple technology sources and adopts certain optimization criteria to achieve the best fusion positioning. The fusion method is the prerequisite and foundation for all-source navigation, and also serves as the key and core of integrated navigation systems.

Data fusion was initially introduced in an article titled *Extension of the Probabilistic Data Association Filter in Multi-target Tracking* published by the American system scientist Bar-Shalom. The article proposed the probabilistic data interconnection filter as the symbol of the formation of multi-source information fusion technology. The multi-source data fusion methods mainly adopt the switching method, the average weighted fusion method, and the adaptive weighted fusion method.

Based on the performance of different positioning sources, the optimal single positioning source is selected as the positioning means (Zou et al., 2013). However, ignoring other positioning sources is a waste and not the best choice. The average weighted fusion method does not take into account the different performances of different positioning sources but assigns the same weight to all positioning sources for fusion localization (Bhujle et al., 2016), which cannot achieve the optimal fusion effect. The adaptive weighted fusion method assigns different weights according to the characteristics of different fusion sources to achieve the best fusion positioning (Guo et al., 2018). The algorithms corresponding to these three methods mainly include optimal estimation algorithms, weighted fusion algorithms or adaptive weighted fusion algorithms (Yang et al., 2001; Yang et al., 2006; Zhang et al., 2016), Bayesian filters (BF), and variable-decibel Bayesian adaptive estimation (Sun et al., 2016; Li et al., 2023), Particle Filters (PF), Statistical decision theory (Berger et al., 2002), evidence theory (Zhang et al., 2017), fuzzy logic (Arikumar et al., 2016), etc. However, these algorithms all have specific preconditions and application scenarios, and it is necessary to establish a mathematical model between the observation information of the navigation source and the system state parameters.

In the field of dynamic positioning such as autonomous driving and vehicle navigation, the Kalman Filter (KF) has been widely used due to the introduction of physical motion models. However, KF is primarily designed for linear systems. For nonlinear systems, such as inertial navigation, the Extended KF (EKF) is suitable for weakly nonlinear objects because higher-order terms above the second order are discarded in the linearization process. To address the issue that the batch processing of the EKF random model requires storing a large amount of data, a recursive method for the random model has been proposed (Zhang et al., 2021), and time-domain non-local filtering data fusion algorithms have also been included (Cheng et al., 2017). The Unscented KF (UKF) retains the accuracy achieved by the third-order term of the Taylor series, making it suitable for nonlinear object estimation, although it involves relatively high computational demands. When both the system state and measurement noise are nonlinear, the PF can be used for nonlinear systems and systems with uncertain error models. However, the PF requires a probability density that closely approximates the real density. Otherwise, the filtering effect may be poor or even divergent. To address this, the Unscented Kalman Particle Filter (UPF) algorithm has been developed (Qin, 2020). However, both the PF and UPF methods face the issue of rapidly increasing computational load as the number of particles grows.

With the increasing demand from users for more comprehensive and intelligent navigation and positioning performance, filtering methods such as Factor Graphs (FG) and neural networks have been introduced. For example, FG algorithms have been extensively applied in single GNSS positioning, GNSS/INS integrated positioning, ambiguity resolution, and robust estimation (Watson et al., 2018; Wen et al., 2021; Gao et al., 2022; Zhang et al., 2024). To enhance positioning accuracy in urban environments, the FG algorithms have been optimized and improved (Lyu et al., 2021; Bai et al., 2022; Liao et al., 2023; Zhang et al., 2023). These studies have demonstrated that under certain conditions, FG algorithms exhibit higher computational accuracy and robustness compared to EKF. In 1965, Magill proposed the Multi-Model Estimation (MME) method, which enhances the adaptability of system models to real systems and changes in external environments under complex conditions, thereby improving the accuracy and stability of filtering estimates. The design of the model set, the selection of filters, estimation fusion, and the reset of filter inputs are all very important

aspects. To enhance the high fault-tolerance capability of integrated navigation systems, Carlson introduced the Federated Filtering (FF) theory in 1988. This theory has been applied in indoor navigation, robotic navigation, and vision-language tasks (Carlson, 1988; Alsanhi et al., 2023). Existing Artificial Intelligence (AI) algorithms mainly include fuzzy control adaptive algorithms and neural network adaptive algorithms (Bian et al., 2004; Gao et al., 2007). For example, to address the impact of random disturbances on systems in underwater environments, RBF neural network-assisted FF has been employed for information fusion (Li et al., 2011). By establishing a black-box model with sufficiently accurate samples through offline training, the positioning accuracy and adaptability of the algorithm have been improved. To tackle the issues of high cost and susceptibility to weather conditions in existing high-precision satellite navigation for agricultural machinery, Yu et al. (2021) proposed a multi-sensor fusion automatic navigation method for farmland based on D-S-CNN. However, these AI algorithms require extensive training data, comprehensive pre-training of the system, and significant computational resources, and often struggle to ensure real-time performance, typically being used for post-processing.

Currently, scholars from various countries have conducted extensive research on integrated navigation systems. For instance, researchers from Linköping University in Sweden proposed a combined navigation system that integrates GPS, INS, and visible light vision assistance (Conte et al., 2009). This system utilizes the vision system and INS for positioning when GPS fails. Locata Corporation in Australia has integrated the Locata system with GPS, INS, vision systems, and Simultaneous Localization and Mapping (SLAM), achieving high-precision applications of the Locata system in both indoor and outdoor environments (Montillet et al., 2014). A communication and navigation fusion system has been applied for seamless positioning across wide-area indoor and outdoor spaces (Deng et al., 2013). A multi-frequency ground-penetrating radar data fusion system is used for working antennas in different frequency ranges (Coster et al., 2018), while multi-sensor data fusion is employed for analyzing airspeed measurement fault detection in drones (Guo et al., 2018). Additionally, an indoor mobile robot based on dead reckoning data fusion and fast response code detection (Nazemzadeh et al., 2017), and an IoT-based multi-sensor fusion strategy for analyzing occupancy sensing systems in smart buildings have been developed (Nesa et al., 2017). Systems that integrate vision, inertial navigation, and asynchronous optical tracking with Inertial Measurement Units (IMU) have also been implemented. Furthermore, several research teams have successfully developed open-source integrated navigation systems for use by academic or industrial technical personnel (Chen et al., 2021; Chi et al., 2023).

Although the aforementioned literature has conducted extensive research and testing on multi-source data fusion methods, fusion systems, and their applications. The theories and models of these methods have their specific applicable scenarios and conditions. Therefore, this paper summarizes the fundamental principles and mathematical models of multi-source data fusion methods, analyzes the advantages and disadvantages of different fusion approaches, and provides theoretical support and reference for the design, development, and application of fusion systems.

2. Multi-Source Data Fusion Processing Method

Fusion methods are primarily categorized into optimal estimation methods, filtering methods, MME, FG, FF, and other fusion approaches. The following sections will focus on elaborating the basic principles of these methods, their corresponding algorithmic models, and applicable scenarios.

2.1. Optimal Estimation Method

We estimate the parameter through randomly distributed observation vectors, specifically by finding a mapping function to compute the estimated value. Here, the system state is represented by the vector \mathbf{x} , and the measurements corresponding to different navigation sources are denoted by \mathbf{y} . The nonlinear and linear observation equations can be formulated as:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\varepsilon} \quad (1)$$

where \mathbf{A} is the observation matrix. $\boldsymbol{\varepsilon}$ includes the error term caused by random observation noise and linearization. The state of the system can be estimated by observation, and the estimated result is a vector $\hat{\mathbf{x}}$.

By employing different estimation criteria to address the problem of estimating unknown parameters, various estimation methods can be derived. Based on Equation (1) and the optimization of the criterion function, methods such as Least Squares Estimation (LSE), Minimum Variance Estimation (MVE), Maximum Likelihood Estimation (MLE), and Maximum A Posteriori Estimation (MAPE) can be formulated.

2.1.1. General LSE and Weighted LSE (WLSE)

The LSE is a parameter estimation algorithm proposed by the German mathematician Carl Friedrich Gauss in 1795, initially developed to determine planetary orbits (Koch, 1977). Regardless of whether the variable x in the linear model (2) possesses prior statistical information or the random distribution followed by \mathbf{y} , the LSE criterion employs a quadratic minimum.

$$J(\hat{\mathbf{x}}) = (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}})^T \mathbf{W} (\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}) = \min \quad (2)$$

where \mathbf{W} is the appropriate positive definite weighted matrix, the above formula is the general LSE criterion when $\mathbf{W} = \mathbf{I}$. Then the estimation error and its variance are,

$$\begin{cases} \hat{\mathbf{x}}_{LS} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{y} \\ \boldsymbol{\Sigma}_{\delta \hat{\mathbf{x}}_{LS}} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \boldsymbol{\Sigma}_{\varepsilon} \mathbf{W} \mathbf{A} (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \end{cases} \quad (3)$$

It is easy to prove that when $\mathbf{W} = \boldsymbol{\Sigma}_{\varepsilon}^{-1}$, LSE has a minimum variance (Cui et al., 2001; Koch 1988a).

The most significant feature of this method lies in its algorithmic simplicity and independence from statistical information related to estimators and measurements. Previous researchers have implemented this algorithm in multi-source data fusion processing (Zhong et al., 2003; Jiao et al., 2010). However, the LSE exclusively utilizes measurement information for current state estimation, a characteristic that paradoxically constrains its applicability. Furthermore, while LSE optimality criterion guarantees the minimization of the total mean squared error in measurement estimates, it fails to ensure optimal estimation error for the estimator itself, consequently leading to suboptimal estimation accuracy.

2.1.2. MLE

Let the conditional probability density of \mathbf{y} with respect to \mathbf{x} be $f(\mathbf{y}/\mathbf{x})$ and the MLE criterion be,

$$f(\mathbf{y}/\mathbf{x}) \Big|_{\mathbf{x}=\hat{\mathbf{x}}_{ML}} = \max \quad (4)$$

This means that the \mathbf{y} has its maximum value at $f(\mathbf{y}/\mathbf{x}) \Big|_{\mathbf{x}=\hat{\mathbf{x}}_{ML}} = \max$, which is equivalent to,

$$\partial f(\mathbf{y}/\mathbf{x}) / \partial \mathbf{x} \Big|_{\mathbf{x}=\hat{\mathbf{x}}_{ML}} = 0 \quad (5)$$

Obviously, the solution of equation (5) is related to the $f(\mathbf{y}/\mathbf{x})$. Different conditional probability density functions lead to different valuation formulas, therefore a universal formula cannot be derived. When the $f(\mathbf{y}/\mathbf{x})$ follows a normal distribution, that is,

$$f(\mathbf{y}/\mathbf{x}) = \frac{1}{(2\pi)^{n/2} |\Sigma_{\mathbf{y}/\mathbf{x}}|^{1/2}} \exp \left\{ -\frac{(\mathbf{y} - E(\mathbf{y}/\mathbf{x}))^T \Sigma_{\mathbf{y}/\mathbf{x}}^{-1} (\mathbf{y} - E(\mathbf{y}/\mathbf{x}))}{2} \right\} \quad (6)$$

According to the normal distribution condition expectation and variance formula (Koch 1988a; Cui et al., 2001),

$$\begin{cases} E(\mathbf{y}/\mathbf{x}) = E(\mathbf{y}) + \Sigma_{\mathbf{yx}} \Sigma_{\mathbf{x}}^{-1} (\mathbf{x} - E(\mathbf{x})) \\ \Sigma_{\mathbf{y}/\mathbf{x}} = \Sigma_{\mathbf{y}} - \Sigma_{\mathbf{yx}} \Sigma_{\mathbf{x}}^{-1} \Sigma_{\mathbf{xy}} \end{cases} \quad (7)$$

From the above derivation, it can be concluded that MLE does not require prior distribution information about \mathbf{x} . However, when its prior distribution information is known, the $f(\mathbf{y}/\mathbf{x})$ should be derived strictly based on Equation (5).

2.1.3. MAPE

The estimation $\hat{\mathbf{x}}_{MA}$ of MAPE makes the $f(\mathbf{x}/\mathbf{y})$ reach the maximum value under the condition $\hat{\mathbf{x}} = \hat{\mathbf{x}}_{MA}$. The equivalent MAPE criteria are as follows after similar derivation (Cui et al., 2001):

$$f(\mathbf{x}/\mathbf{y}) \Big|_{\hat{\mathbf{x}}=\hat{\mathbf{x}}_{MA}} = \max \quad (8)$$

According to the conditional probability formula, it is easy to prove that (8) is equivalent to,

$$f(\mathbf{x}/\mathbf{y}) \Big|_{\hat{\mathbf{x}}=\hat{\mathbf{x}}_{MA}} = f(\mathbf{x}/\mathbf{y})/f_2(\mathbf{y}) \Big|_{\hat{\mathbf{x}}=\hat{\mathbf{x}}_{MA}} = \max \Leftrightarrow f(\mathbf{x}, \mathbf{y}) \Big|_{\hat{\mathbf{x}}=\hat{\mathbf{x}}_{MA}} = \max \quad (9)$$

That is, the MAPE uses the joint probability density function of the parameters and the observed data as the maximization criterion.

When both \mathbf{y} and \mathbf{x} are normally distributed, the conditional probability density function is,

$$f(\mathbf{x}/\mathbf{y}) = \frac{1}{(2\pi)^{t/2} |\Sigma_{\mathbf{x}/\mathbf{y}}|} \exp \left\{ -\frac{(\mathbf{x} - E(\mathbf{x}/\mathbf{y}))^T \Sigma_{\mathbf{x}/\mathbf{y}}^{-1} (\mathbf{x} - E(\mathbf{x}/\mathbf{y}))}{2} \right\} \quad (10)$$

Differentiate Equation (9) and set the derivative to zero to obtain the variance and estimation error of the MAPE:

$$\begin{cases} \hat{\mathbf{x}}_{MA} = E(\mathbf{x}/\mathbf{y}) \\ \Sigma_{\delta \hat{\mathbf{x}}_{MA}} = \Sigma_{\mathbf{x}/\mathbf{y}} \end{cases} \quad (11)$$

When \mathbf{x} is uncorrelated with \mathbf{y} , the estimated result is equal to the prior information of the parameter.

Compared with MLE, MAPE places greater emphasis on the prior information of parameters. Therefore, MAPE becomes meaningless when parameters are non-random quantities. In practice, MAPE can be understood as a modification of parameters through observational data, where the extent of this modification is determined by the variance of the observations and their correlation with the parameters.

2.1.4. MVE

The MVE is an optimization criterion that minimizes the variance of the estimation error to estimate $\hat{\mathbf{x}}_{MV}$.

$$J(\hat{\mathbf{x}}) = E \left((\mathbf{x} - \hat{\mathbf{x}}_{MV})^T (\mathbf{x} - \hat{\mathbf{x}}_{MV}) \right) \Big|_{\hat{\mathbf{x}}=\hat{\mathbf{x}}_{MV}} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\mathbf{x} - \hat{\mathbf{x}}_{MV})^T (\mathbf{x} - \hat{\mathbf{x}}_{MV}) f(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = \min \quad (12)$$

According to the conditional and edge probability density functions, equation (12) is equivalent to

$$h(\hat{\mathbf{x}}_{MV}) = \int_{-\infty}^{\infty} (\mathbf{x} - \hat{\mathbf{x}}_{MV})^T (\mathbf{x} - \hat{\mathbf{x}}_{MV}) d\mathbf{x} = \min \quad (13)$$

The MVE and its estimation error are:

$$\begin{cases} \hat{\mathbf{x}}_{MV} = E(\mathbf{x}/\mathbf{y}) \\ \Sigma_{\delta\hat{\mathbf{x}}_{MV}} = \int_{-\infty}^{\infty} \Sigma_{\mathbf{x}/\mathbf{y}} f_2(\mathbf{y}) d\mathbf{y} \end{cases} \quad (14)$$

where $f_2(\mathbf{y})$ is the edge probability density function of the observation. When both the \mathbf{x} and the \mathbf{y} are normally distributed, the MVE is completely equivalent to the MAPE.

2.1.5. LMVE

MLE, MAPE, and MVE all require the joint probability density or conditional probability density function of \mathbf{y} and \mathbf{x} . Their estimation formulas are related to distribution information and are not necessarily linear combinations of the observed values. However, LSE does not require the distribution information of the parameters and observed values, the parameter estimation is a linear combination of the observed values and linear estimation. As the name suggests, The LMVE is also a linear estimation method, which does not require the distribution information of the observed values and parameters, but only their statistics.

Let LMVE be estimated as

$$\hat{\mathbf{x}}_{LMV} = \mathbf{a} + \boldsymbol{\beta}\mathbf{y} \quad (15)$$

where \mathbf{a} is a constant vector and $\boldsymbol{\beta}$ is a constant matrix, then the estimation error and its variance are:

$$\begin{cases} \delta\hat{\mathbf{x}}_{LMV} = \mathbf{x} - \hat{\mathbf{x}}_{LMV} = \mathbf{x} - \mathbf{a} - \boldsymbol{\beta}\mathbf{y} \\ \Sigma_{\delta\hat{\mathbf{x}}_{LMV}} = \Sigma_{\mathbf{x}} + \boldsymbol{\beta}\Sigma_{\mathbf{y}}\boldsymbol{\beta}^T - \Sigma_{\mathbf{xy}}\boldsymbol{\beta}^T - \boldsymbol{\beta}\Sigma_{\mathbf{yx}} \end{cases} \quad (16)$$

It is worth noting that LMVE takes the minimum mean square error $\Sigma_{\delta\hat{\mathbf{x}}_{LMV}}$ as the estimation criterion.

$$\begin{aligned} \mathbf{M}_{\delta\hat{\mathbf{x}}_{LMV}} &= E(\delta\hat{\mathbf{x}}_{LMV}^T \delta\hat{\mathbf{x}}_{LMV}) = E(\delta\hat{\mathbf{x}}_{LMV}) E(\delta\hat{\mathbf{x}}_{LMV})^T + \Sigma_{\delta\hat{\mathbf{x}}_{LMV}} \\ &= E(\delta\hat{\mathbf{x}}_{LMV}) E(\delta\hat{\mathbf{x}}_{LMV})^T + (\boldsymbol{\beta} - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1})\Sigma_{\mathbf{y}}(\boldsymbol{\beta} - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}) + \Sigma_{\mathbf{x}} - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}\Sigma_{\mathbf{yx}} \end{aligned} \quad (17)$$

Then,

$$\mathbf{M}_{\delta\hat{\mathbf{x}}_{LMV}} = \min \Leftrightarrow \begin{cases} E(\delta\hat{\mathbf{x}}_{LMV}) = 0 \\ \boldsymbol{\beta} = \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1} \end{cases} \Leftrightarrow \begin{cases} E(\delta\hat{\mathbf{x}}_{LMV}) = 0 \\ \text{var}(\delta\hat{\mathbf{x}}_{LMV}) = \min \end{cases} \Leftrightarrow \text{var}(\delta\hat{\mathbf{x}}_{LMV}) = \min \quad (18)$$

Therefore, LMVE is unbiased estimation, and the variance of estimation error is minimal under the premise of unbiased estimation. However, the minimum variance of the estimation error is only a necessary condition for LMVE. By inserting equation (18) into equation (15) and (16), The estimation error and variance of LMVE are:

$$\begin{cases} \hat{\mathbf{x}}_{LMV} = E(\mathbf{x}) + \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}(\mathbf{y} - E(\mathbf{y})) = \boldsymbol{\mu}_{\mathbf{x}} + \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}(\mathbf{y} - E(\mathbf{y})) \\ \Sigma_{\hat{\mathbf{x}}_{LMV}} = \Sigma_{\mathbf{x}} - \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{y}}^{-1}\Sigma_{\mathbf{yx}} \end{cases} \quad (19)$$

It is noteworthy that equation (19) is derived under the premise of knowing the statistical properties (namely, the mathematical expectation and variance) of \mathbf{y} and \mathbf{x} , and is independent of

their distribution information. Therefore, when both \boldsymbol{x} and \boldsymbol{y} follow a normal distribution, the LMVE is equivalent to the MVE and MAPE.

2.1.6. Comparison of Several Different Optimal Estimation Methods

When \boldsymbol{x} and \boldsymbol{y} are normally distributed, Table 1 presents a table of parameter estimates based on different criteria.

Table 1. Parameter estimation under different criteria ($E(\boldsymbol{x}) = \mu_x$, $E(\boldsymbol{y}) = \mu_y$, $\text{var}(\boldsymbol{x}) = \boldsymbol{\Sigma}_x$, $\text{var}(\boldsymbol{y}) = \boldsymbol{\Sigma}_y$, $\text{cov}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\Sigma}_{xy}$).

Type	Estimation method				
	LS	MLE	MAP	MVE	LMVE
Estimation criterion	Formula (2)	Formula (4)	Formula (8)	Formula (12)	Formula (17)
Estimation formula	The first formula in formula (3)	The first formula in formula (7)	See formulas (10), (13) and (18)		
Estimation error variance	The second formula in formula (3)	The second formula in formula (7)			
Unbiasing	Unbiasedness				

Table 2 presents the advantages and disadvantages of several optimal estimation methods.

Table 2. Comparison of several optimal estimation methods.

Method	Advantages	Disadvantages
LSE	Simple and easy to use, no distribution assumptions, geometrically intuitive, and not dependent on the specific distribution of the data.	Sensitive to outliers, lacks statistical features, and is limited to linear models.
MLE	Gradual progression is beneficial, with strong versatility and clear statistical properties.	Computationally complex, sensitive to initial values, and dependent on distribution assumptions.
MAPE	Utilizing the prior distribution. Suitable for situations with small sample sizes or insufficient data and provides a complete posterior distribution for further analysis.	Relies on prior selection, with high computational complexity requiring integral computation. Sensitive to prior selection, where different prior choices may lead to distinct results.
MVE	The optimal unbiased estimator possesses well-defined statistical properties, achieves the minimum mean squared error among all estimators, and exhibits the best performance across all estimation methods. When both the estimated quantity and the measurements follow a normal distribution, the LMVE becomes equivalent to the MVE.	Dependent on model assumptions, computationally complex, and limited to unbiased estimation. Requires determining the conditional mean of measurements and estimated values under measurement conditions, which is computationally intensive. For non-stationary processes, it necessitates knowledge of first and second-order moments at each time instant, resulting in high computational demands.
LMVE	Best Linear Unbiased Estimator, simple to compute with clear statistical properties.	Limited to linear models. Dependent on model construction. Sensitive to outliers.

2.2. Filtering Algorithm

The optimization method utilizes the observations from the current epoch to estimate the system state, hence the localization results are significantly influenced by the current observation errors. In 1960, R. E. Kalman first proposed the KF, which employs a recursive approach to avoid the accumulation of large amounts of data and designs the filter using the state-space method in the time domain, making KF suitable for the estimation of multi-dimensional random processes. Depending on the differences in system equations and state equations, Filtering algorithm is primarily divided into the following algorithms.

2.2.1. Standard KF

Assuming that both the state motion and observation models are linear, the estimated state \mathbf{x}_k at epoch t_k is driven by the system noise sequence \mathbf{w}_{k-1} , and the driving mechanism is described by the following equation of state:

$$\mathbf{x}_k = \Phi_{k,k-1} \mathbf{x}_{k-1} + \Gamma_{k-1} \mathbf{w}_{k-1} \quad (20)$$

where $\Phi_{k,k-1}$ is a one-step transfer matrix from time t_{k-1} to time t_k . Γ_{k-1} is the system noise-driving term. Let the observation of \mathbf{x}_k be z_k , z_k and \mathbf{x}_k satisfy the linear relationship, and the observation equation is:

$$z_k = H_k \mathbf{x}_k + v_k \quad (21)$$

where H_k is the observation matrix, v_k is the observation noise, \mathbf{w}_k and v_k meet at the same time:

$$E(\mathbf{w}_k) = 0, \text{cov}(\mathbf{w}_k, \mathbf{w}_j) = E(\mathbf{w}_k \mathbf{w}_j^T) = \mathbf{Q}_k \delta_{kj} \quad (22)$$

$$E(v_k) = 0, \text{cov}(v_k, v_j) = E(v_k v_j^T) = \mathbf{R}_k \delta_{kj} \quad (23)$$

$$\text{cov}(\mathbf{w}_k, v_j) = E(\mathbf{w}_k v_j^T) = 0 \quad (24)$$

where \mathbf{Q}_k is the variance matrix of the state noise sequence. \mathbf{R}_k is the variance matrix of the observation noise sequence.

If the \mathbf{Q}_k is not negative definite, the \mathbf{R}_k is positive definite, then the estimate $\hat{\mathbf{x}}_k$ is solved according to the KF equation. The calculation steps of KF include a filter gain calculation loop (time update process) and a filter calculation process (measurement update process) two calculation loops. As shown in Figure 1. Areas with green shading represent the estimates and their covariance matrix.

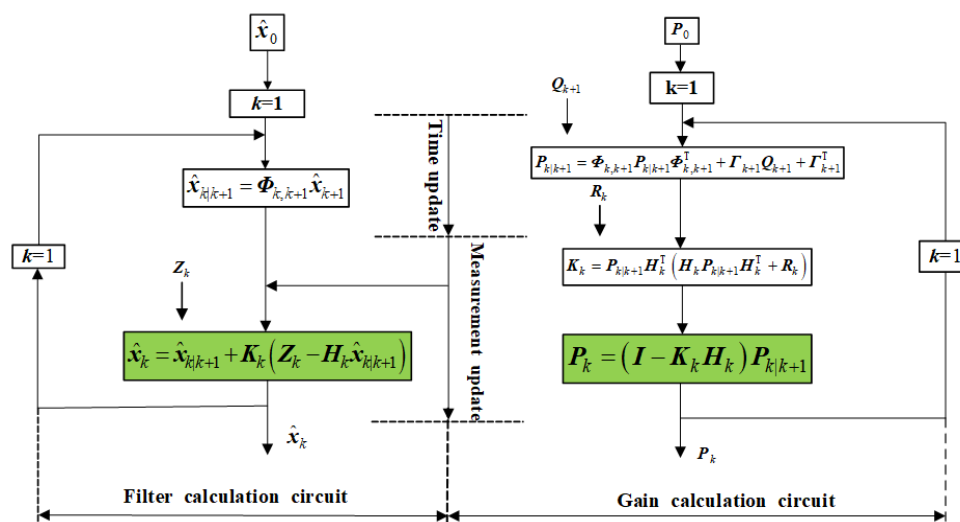


Figure 1. Calculation process of KF.

As can be seen from Figure 1, the KF must provide initial values and. Practice has proven that the recursive computation method is the greatest advantage of KF. This algorithm can be executed by a computer and does not require the storage of large amounts of measurement data over time, which is why the KF has been widely adopted in engineering. KF clearly states that the system driving noise and measurement noise must be white noise. However, in reality, these two types of noise in some systems are colored noise. Consequently, the KF under colored noise conditions has been proposed, mainly including the KF when the system noise is colored and the measurement noise is white, and the KF when the system noise is white and the measurement noise is colored (Jazwinski, 1970).

The estimated $\mathbf{P}_{k|k+1}$ calculated according to the filtering equations tends to zero or to a steady-state value as the number of observation epochs k increases. When the deviation between the estimated value and the true value becomes increasingly large, the filter gradually loses its estimation capability. This phenomenon is known as filter divergence. To mitigate filter divergence, numerous scholars have proposed various methods, including information filtering (Dan, 2006), square root filtering (Bierman et al., 1987), UUDT decomposition filtering (Li et al., 2009), adaptive filtering (Mechra et al., 1972), suboptimal filtering (Athans et al., 1968), and H^∞ filtering (Dan, 2006). Each of these methods has its own advantages, disadvantages, and applicable scopes, and further details can be found in the relevant literature.

2.2.2. Extended Kalman Filter (EKF)

The mathematical model of the system is linear in the KF, which means it is only applicable when both the system equations and observation equations are linear. However, in engineering applications, the mathematical models of systems are often nonlinear, such as in the INS of aircraft and ships, satellite navigation, and industrial control systems, making KF unsuitable for direct application. In such cases, the EKF can be used to linearize the nonlinear system. EKF utilizes Taylor series expansion to linearize the nonlinear system and transforms the original nonlinear system model into state and observation equations.

The nonlinear equation of state and observation equation corresponding to EKF are:

$$\begin{cases} \dot{\mathbf{x}}(k) = \mathbf{f}(\mathbf{x}(k)) + \mathbf{g}(\mathbf{w}(k)) \\ \mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{v}(k) \end{cases} \quad (25)$$

where $\mathbf{f}(\mathbf{x}(k))$ and $\mathbf{g}(\mathbf{w}(k))$ are nonlinear functions of the state vector $\mathbf{x}(k)$ and system noise vector $\mathbf{w}(k)$ respectively. $\mathbf{h}(\mathbf{x}(k))$ is the nonlinear function of the design matrix.

The two equations in formula (25) are linearized as follows:

$$\begin{cases} \delta \mathbf{x}_k = \mathbf{F}_{k-1} \delta \mathbf{x}_{k-1} + \mathbf{G}_{k-1} \mathbf{w}_{k-1} \\ \delta \mathbf{z}_k = \mathbf{H}_k \delta \mathbf{x}_k + \mathbf{v}_k \end{cases} \quad (26)$$

After linearization, the solution can be solved by using KF.

The EKF is suitable for weakly nonlinear systems because its linearization process retains only the first-order terms while discarding higher-order terms above the second-order. To address the issue of the EKF stochastic model batch processing requiring the storage of large amounts of data, the algorithm proposes a recursive approach for the stochastic model (Zhang et al., 2021). Additionally, it includes a time-domain non-local filtering data fusion algorithm (Cheng et al., 2017).

2.2.3. Unscented Kalman Filter (UKF)

The EKF discards higher-order terms above the second order and retains only the linear terms, thus the EKF algorithm is only suitable for the estimation of weakly nonlinear objects. When the nonlinearity of the estimated object is stronger, the resulting estimation error will also be larger, and

may even cause the filter to diverge. The UKF is an effective method for solving nonlinear system problems. In 1995, S.J. Julier and J.K. Uhlmann proposed the UKF algorithm to address the filtering problem of strongly nonlinear objects, which was subsequently further refined by E.A. Weiss and R.V. Merwe (Simon, 2001; Dan, 2014).

The core of UKF is to use Untrace Transformation (UT) to determine the mapping relationship between variables, which is equivalent to retaining the accuracy achieved by the third-order term of the Taylor series, so it is suitable for nonlinear object estimation.

Using discrete nonlinear systems, i.e.:

$$\begin{cases} \mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_k \\ \mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \end{cases} \quad (27)$$

where \mathbf{x}_k is the state vector. \mathbf{w}_k is the system noise vector. \mathbf{z}_k is the observed vector. \mathbf{v}_k is the observed noise vector.

A series of sampling points are selected near \mathbf{x}_{k-1} , The mean and covariance of these sample points are $\hat{\mathbf{x}}_{k-1}$ and \mathbf{P}_{k-1} , respectively. The corresponding transform sampling points are generated through the nonlinear system. The predicted mean and covariance can be obtained by calculating these transform sampling points.

Let the state variable be n-dimensional, then 2n+1 sampling points and their weights are:

$$\begin{cases} \xi_0 = \hat{\mathbf{x}}_{k-1} \\ \xi_i = \hat{\mathbf{x}}_{k-1} + \sqrt{n + \tau} (\mathbf{P}_{k-1})_i & i = 1, 2, \dots, n \\ \xi_{i+n} = \hat{\mathbf{x}}_{k-1} - \sqrt{n + \tau} (\mathbf{P}_{k-1})_i \end{cases} \quad (28)$$

and the weight corresponding to $\xi_{i,k-1}$ ($i=0,1,\dots,2n$) is:

$$W_i^m = W_i^c = \begin{cases} \kappa / (n + \kappa) & i = 0 \\ 1 / 2(n + \kappa) & i \neq 0 \end{cases} \quad (29)$$

where κ is the proportional coefficient, which can be used to adjust the distance between sigma points and $\hat{\mathbf{x}}_{k-1}$, and this coefficient only affects the higher-order matrix deviations above the second order. $(\sqrt{\mathbf{P}_{k-1}})_i$ is the i-th row or column of the square root matrix. There are different ways to determine W_i^m , and some literature has made some improvements, such as the UKF algorithm under additive noise cases and non-additive noise conditions (Liu et al., 2010). The specific process of the UKF algorithm can be referred to in the relevant literature. Practice proves that UKF is suitable for nonlinear object estimation, but the computation is relatively large.

Due to the strong advantages of the UKF in handling nonlinear systems, it has also been applied in multi-source data fusion. Examples include autonomous multi-level positioning based on smartphone-integrated sensors and pedestrian indoor networks (Shi et al., 2023), as well as the use of UKF in mobile mapping applications based on low-cost GNSS/IMU under demanding conditions (Cahyadi et al., 2024).

2.2.4. Particle Filter (PF)

The PF was initially proposed by Metropolis and Wiener as early as 1940 (Wiener, 1956). PF is a method that approximates the probability density function by propagating a set of random samples in the state space. It replaces the integral operation with the sample mean, thereby obtaining the minimum variance estimate of the system state. These samples are referred to as particles in the image, hence the method is known as PF. Here, the samples are the particles, and when the number of samples $N \rightarrow \infty$, it can approximate any form of probability density distribution.

The PF directly calculates the conditional mean, which is the minimum variance estimate, based on the probability density. This probability density is approximated by the EKF or the UKF. The

estimate is determined by the weighted average of sample values (particles) from multiple different distributions. Each particle computation values (particles) from multiple different distributions. Each particle computation requires a complete EKF or UKF calculation. Therefore, PF is suitable for estimation under nonlinear system and measurement conditions, offering higher estimation accuracy than using EKF or UKF alone, but with a significantly higher computational load compared to EKF and UKF.

PF has many different names, For example, Sequential Importance Sampling (SIS) (Doucet et al., 2001), Bootstrap Filtering (Gray et al., 1993), Condensation Algorithm (Isaard et al., 1996; McCormick et al., 1999), Interacting Particle Approximations (Moral, 1998), Monte-Carlo Filtering (Kitagawa, 1996), Sequential Monte-Carlo filtering (Crisan et al., 2002; Andrieu et al., 2004).

According to relevant literature, the general procedure for executing PF is as follows:

Step 1: Initial value determination

The initial particle value $\chi_0^{(i)} (i=1,2,\dots,N)$ is generated according to the prior probability density $p(\mathbf{x}_0)$ of the initial state.

For $k=1, 2, 3, \dots$ Execute,

Step 2: Select the recommended probability density $q[\mathbf{x}_k / (\mathbf{x}_0^{(i)}, \mathbf{z}_0^k)]$. According to this recommended density, a particle $\chi_k^{(i)}$ at time $k, i=1,2,\dots, N$ is generated, as a secondary sample of the original particle.

Calculated weight coefficient,

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p[\mathbf{z}_k / \chi_k^{(i)}] p[\chi_k^{(i)} / \chi_{k-1}^{(i)}]}{q[\chi_k^{(i)} / (\chi_0^{(i)}, \mathbf{z}_0^k)]} \quad (30)$$

$$w_0^{(i)} = p[\chi_0^{(i)}] \quad (31)$$

$$\tilde{w}_k^{(i)} = w_k^{(i)} \left[\sum_{j=1}^N w_k^{(j)} \right], \quad i=1,2,\dots,n \quad (32)$$

Step 3: the PF algorithm, such as the SIR method or residual secondary sampling method, is used to perform secondary sampling on the original particle $\chi_k^{(i)} (i=1,2,\dots,n)$ and generate secondary sampling to update the particle $\chi_k^{+(i)} (i=1,2,\dots,n)$.

Step 4: Calculating the filtering value based on the secondary sampling particles,

$$\hat{\mathbf{x}}_k = \frac{1}{n} \sum_{i=1}^n \chi_k^{+(i)} \quad (33)$$

The PF is mainly used for data fusion and integrity monitoring (Wang et al., 2014 & 2016; Yun et al., 2017; Bi et al., 2019). Therefore, when both system and measurement noises are nonlinear, PF can be applied to nonlinear systems as well as systems with uncertain error models.

2.2.5. UKF-Based Particle Filter (UPF)

The core of PF is to select the proposal probability density. The closer the proposal density is to the true density, the better the filtering effect will be. Conversely, if there is a significant difference between the proposal density and the true density, the filtering effect will deteriorate, and divergence may even occur. If PF is combined with UKF, and the proposal density is determined by UKF, the problem of particle degeneracy can be resolved. When updating the particles, the latest posterior information can be obtained, which is beneficial for regions with a high particle likelihood ratio. The method that combines PF with UKF is called UPF. However, both PF and UPF have a common problem, which is that as the number of particles increases, the computational complexity will sharply increase.

2.2.6. Federated Filtering (FF)

The filtering methods introduced above belong to Centralized KF (CKF). CKF has problems such as high state dimension, heavy computational burden, and poor fault-tolerance. Another filtering method is Decentralized KF (DKF). DKF has been developed for over 20 years. As early as 1971, Pearson (1971) proposed the concept of dynamic decomposition and a two-level structure for state estimation. Subsequently, Speyer (1979), Willsky (1982), Kerr (1987), and Carlson (1988) made contributions to DKF techniques. Among the many DKF methods, the FF proposed by Carlson has been valued for its design flexibility, low computational load, and good fault-tolerance. Now, the FF has been selected as the basic algorithm for the U.S. Air Force's fault-tolerant navigation system, the Common Kalman Filter program (Loomis et al., 1988).

The FF proposed by Carlson is designed to address the following issues:

(1) The filter should have good fault-tolerance. When one or several navigation subsystems fail, it should be able to easily detect and isolate the faults and quickly recombine the remaining normal navigation subsystems (reconfiguration) to continue providing the required filtering solution.

(2) The filtering accuracy should be high.

(3) The fusion algorithm from local filtering to global filtering should be simple, with low computational load and minimal data communication, to facilitate real-time implementation of the algorithm.

The FF is a two-level filter, as shown in Figure 2. The Carlson FF introduces a master filter. Since the master filter does not accept measurement inputs, it only has time updates and no measurement updates. Additionally, a feedback control switch from the master filter to the sub-filters is added. The entire filtering system consists of $\bar{N} = (N+1)$ filters, with N sub-filters providing local estimates ($\hat{X}_k^{(ci)}$ and $P_k^{(ci)}$) to the master filter. These local estimates are optimally combined with the master filter estimates ($\hat{X}_k^{(m)}$ and $P_k^{(m)}$) to obtain the global estimates (\hat{X}_k^g and P_k^g).

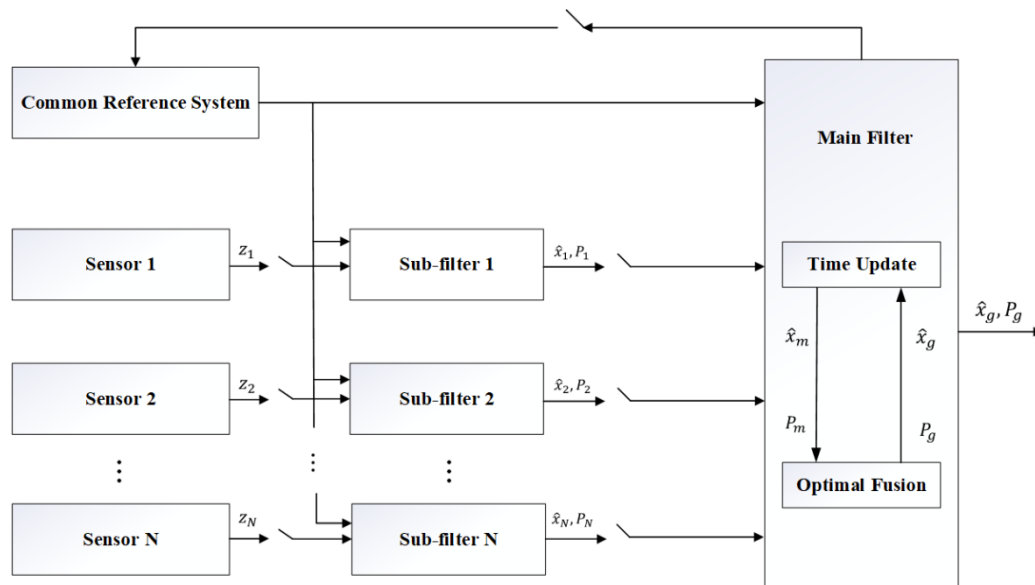


Figure 2. The General Structure of FF.

If there are N local state estimates $\hat{X}_1, \hat{X}_2, \dots, \hat{X}_N$ and their corresponding covariance matrices $P_{11}, P_{22}, \dots, P_{NN}$, and the local estimates are uncorrelated with each other, i.e., $P_{ij} = 0 (i \neq j)$, then the global optimal estimate can be expressed as:

$$\hat{X}_g = P_g \sum_{i=1}^N P_{ii}^{-1} \hat{X}_i \quad (34)$$

$$\mathbf{P}_g = \left(\sum_{i=1}^N \mathbf{P}_{ii}^{-1} \right)^{-1} \quad (35)$$

Let the FF have N ($N > 2$) sub-filters, and the output of sub-filter i is,

$$\hat{\mathbf{X}}_i = \mathbf{X} + \tilde{\mathbf{X}}_i \quad i=1,2,\dots,N \quad (36)$$

where \mathbf{X} represents the common state of all sub-filters, with a dimension of n . $\tilde{\mathbf{X}}_i$ denotes the estimation error of the i -th sub-filter. If the sub-filter operates normally, $\tilde{\mathbf{X}}_i$ is white noise.

The measurement equation for \mathbf{X} can be formulated based on the outputs of the N sub-filters.

$$\mathbf{Z} = \mathbf{H}\mathbf{X} + \mathbf{V} \quad (37)$$

Thus,

$$\mathbf{Z} = \begin{bmatrix} \hat{\mathbf{X}}_1 \\ \hat{\mathbf{X}}_2 \\ \vdots \\ \hat{\mathbf{X}}_N \end{bmatrix}, \mathbf{H} = \begin{bmatrix} \mathbf{I}_{n \times n} \\ \mathbf{I}_{n \times n} \\ \vdots \\ \mathbf{I}_{n \times n} \end{bmatrix}, \mathbf{V} = \begin{bmatrix} \tilde{\mathbf{X}}_1 \\ \tilde{\mathbf{X}}_2 \\ \vdots \\ \tilde{\mathbf{X}}_N \end{bmatrix} \quad (38)$$

Assuming that each sub-filter operates normally and the estimation errors are mutually uncorrelated, we have,

$$\begin{cases} E(\mathbf{V}) = 0 \\ \mathbf{R} = E(\mathbf{V}\mathbf{V}^T) = \text{diag}(\mathbf{P}_{11} \quad \mathbf{P}_{22} \quad \dots \quad \mathbf{P}_{NN}) \end{cases} \quad (39)$$

Among them, $\mathbf{P}_{ii} = E(\tilde{\mathbf{X}}_i \tilde{\mathbf{X}}_i^T)$ is the covariance matrix of the estimation error of sub-filter i .

According to the literature, the Markov estimate of the common state \mathbf{X} is,

$$\begin{cases} \mathbf{P}_g = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} = \left(\sum_{i=1}^N \mathbf{P}_{ii}^{-1} \right)^{-1} \\ \hat{\mathbf{X}}_g = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{Z} = \left(\sum_{i=1}^N \mathbf{P}_{ii}^{-1} \right)^{-1} \sum_{i=1}^N \mathbf{P}_{ii}^{-1} \hat{\mathbf{X}}_i = \mathbf{P}_g \sum_{i=1}^N \mathbf{P}_{ii}^{-1} \hat{\mathbf{X}}_i \end{cases} \quad (40)$$

The physical meaning of the above result is quite evident. If the estimation accuracy of $\tilde{\mathbf{X}}_i$ is poor, that is, \mathbf{P}_{ii} is large, then its contribution to the global estimate is relatively small. The above discussion pertains to the fusion algorithm when the estimates of each sub-filter are uncorrelated. For the fusion algorithm when the estimates of each sub-filter are correlated, one can refer to relevant literature.

2.2.7. Comparison of Different Filtering Method

To more clearly illustrate the performance of the various filtering methods introduced above, Table 3 provides a summary of the information.

Table 3. Performance Comparison of Different Filtering Methods.

Method	Filter Type	Advantages	Disadvantages
KF	Centralized KF	(1) Low computational load and strong real-time performance. (2) it can provide optimal state estimation under linear Gaussian systems. (3) the algorithm is relatively simple and easy to implement.	(1) It can only handle linear systems and is not capable of dealing with nonlinear systems. (2) it has strict requirements for noise distribution, which must be Gaussian. (3) it is sensitive to model errors, and inaccurate initial state

			estimation may lead to a slower convergence rate of the filter.
EKF	Centralized KF	(1) The EKF approximates the state and measurement equations of the real system by linearizing the nonlinear functions, retaining only the first-order terms and discarding the second-order and higher-order terms. (2) In systems with low nonlinearity, the EKF can maintain a high estimation accuracy. (3) Compared with other nonlinear filtering methods (such as the UKF), the EKF has a lower computational complexity and is more suitable for real-time applications.	The EKF approximates the state and measurement equations of the real system by linearizing the nonlinear functions, thereby enabling it to handle nonlinear systems. In systems with low nonlinearity, the EKF can maintain a high estimation accuracy. Compared with other nonlinear filtering methods (such as the UKF), the EKF has a lower computational complexity and is more suitable for real-time applications.
UKF	Centralized KF	(1) It is capable of better handling non-Gaussian noise and nonlinear systems. (2) Compared to the EKF, it retains terms up to the second order in the linearization process, and it performs better in high-dimensional state spaces.	(1) High computational complexity, as it requires the calculation of a large number of sigma points. (2) Sensitivity to noise, which may introduce additional noise. (3) Manual parameter tuning is required, such as the number and weights of sigma points.
PF	Centralized KF	(1) It can handle nonlinear systems and non-Gaussian noise. (2) There are no strict requirements for the noise distribution. (3) The algorithm can be computed in parallel, improving computational efficiency.	(1) It has a large computational load, especially in high-dimensional state spaces. (2) The choice of particle number can affect the filtering performance, and particle degeneracy may occur
UPF	Centralized KF	(1) More accurate than particle filters, especially in high-dimensional state spaces. (2) It can effectively reduce the phenomenon of particle degeneracy.	(1) Slightly higher computational load compared to particle filters. (2) Somewhat dependent on the choice of the UT.
FF	Decentralized KF	(1) It reduces the communication burden of centralized filtering and decreases the dependence on the central processor. (2) Compared with Decentralized filtering, it improves the estimation accuracy by adding some central coordination.	(1) The implementation complexity is relatively high, and an effective information exchange mechanism needs to be designed. (2) There may be data imbalance issues that can affect the accuracy and stability of the model.

2.3. Multiple Model Estimation (MME)

MME can estimate the system state through weighted first-order filter estimates with different parameter values, thereby achieving the goal of adapting to unknown or uncertain system parameters (Magill et al., 1965). In 1970, Ackerson first applied MME to jump environments, arguing that the system pattern is a finite-state Markov chain that can switch between different patterns. Since then, MME methods have been widely used in many fields under various names (Li, 1996), such as

multi-model adaptive estimation, parallel processing algorithm, filter bank method, segmented filter, and improved Gaussian sum filter.

Any real system has different degrees of uncertainty, which is sometimes manifested inside the system and sometimes manifested outside the system. From the inside of the system, the mathematical model structure and parameters describing the controlled object cannot be fully known in advance by the designer. As for the influence of the external environment on the system, it can be expressed as many disturbances. These disturbances are often unpredictable and can be either deterministic or random. In addition, some measurement noise enters the system from different measurement feedbacks in the same way as disturbances, and the statistical properties of these random disturbance noises are usually unknown. Therefore, under the condition that the mathematical models of the controlled object and the disturbances are not fully determined, the control sequence is designed to make the specified performance index as close to and as optimal as possible. Essentially, an adaptive control system is an intrinsically nonlinear system, which is very difficult to analyze. MME theory employs N linear stochastic control systems to solve the nonlinear problem of adaptive control, which will improve the adaptability of the system models to real systems and external environmental changes, and enhance the accuracy and reliability of filter estimation.

Assume the linear stochastic system model is:

$$\begin{cases} \mathbf{x}(k+1) = \Phi(\theta)\mathbf{x}(k) + \mathbf{B}(\theta)\mathbf{u}(k) + \Gamma(\theta)\mathbf{w}(k) \\ \mathbf{z}(k) = \mathbf{H}(\theta)\mathbf{x}(k) + \mathbf{v}(k) \end{cases} \quad (41)$$

where $\mathbf{x}(k)$ is the state vector of the system, $\mathbf{z}(k)$ is the output vector of the system, and $\Phi(\theta), \mathbf{B}(\theta), \Gamma(\theta), \mathbf{H}(\theta)$ are the system matrices of appropriate dimensions. $\mathbf{u}(k)$ and $\mathbf{w}(k)$ are sequences of zero-mean white noise vectors with dimensions n and m, respectively, and their covariance matrices are $\mathbf{Q}(\theta)$ and $\mathbf{R}(\theta)$.

The nonlinear system is linearized at N operating points, allowing the original nonlinear system to be approximated by N sets of linear equations. The parameter $\theta \in \{\theta_1, \theta_2, \dots, \theta_N\}$ can take N discrete values, thereby forming a combination of N linear systems that serve as an approximation of the nonlinear system model.

By representing any value of the parameter θ as θ_i , the system matrix in equation (41) is redefined as follows:

$$\begin{cases} \Phi(\theta_i) = \Phi_i, \quad \mathbf{B}(\theta_i) = \mathbf{B}_i \\ \Gamma(\theta_i) = \Gamma_i, \quad \mathbf{H}(\theta_i) = \mathbf{H}_i \quad (i=1, 2, \dots, N) \end{cases} \quad (42)$$

Based on the aforementioned notation, N discrete random linear systems can be characterized as,

$$\begin{cases} \mathbf{x}_i(k+1) = \Phi_i \mathbf{x}(k) + \mathbf{B}_i \mathbf{u}(k) + \Gamma_i \mathbf{w}(k) \\ \mathbf{z}_i(k) = \mathbf{H}_i \mathbf{x}(k) + \mathbf{v}_i(k) \end{cases} \quad (43)$$

Figure 3 illustrates the fundamental principle of the MME method, wherein N discrete values constitute N distinct systems.

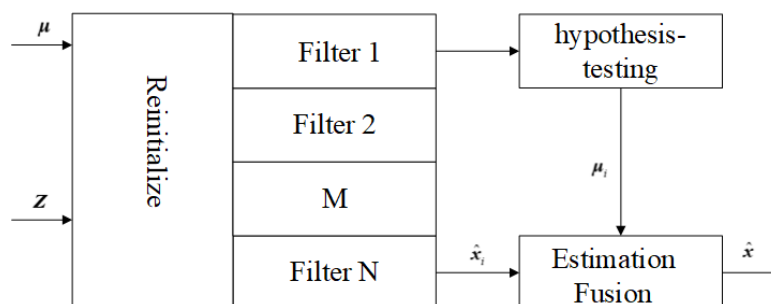


Figure 3. Principle of the MME method.

As illustrated in Figure 3, a parallel array of filters is implemented to accommodate different operational modes of the stochastic hybrid system. Each filter processes both the control input and measurement data from the system, generating output residuals and state estimates based on individual models. The system incorporates model probability design for each corresponding filter, and the comprehensive state estimation is derived from the weighted average of all filter state estimates. Table 4 summarizes the performance characteristics of the MME method.

Table 4. Performance Characteristics of the MME.

Method	Advantages	Disadvantages
MME	<p>(1) The MME method describes different states or behaviors of a system by combining multiple models, enabling more comprehensive coverage of complex system characteristics.</p> <p>(2) Through the integration of multiple models, the system can better address model errors, noise, and uncertainties. Even if one model deviates, others can still provide reliable estimates, thereby enhancing the overall robustness of the system.</p> <p>(3) MME dynamically adjust model weights or switch between models based on real-time data, allowing better adaptation to changes in system states. This adaptive capability makes them particularly effective in dynamic environments.</p>	<p>(1) The MME method requires simultaneous computation and updating of multiple models, resulting in significantly higher computational complexity compared to single-model approaches. This may impose demanding requirements on computational resources for real-time applications.</p> <p>(2) The MME necessitates careful design and management of parameters, weights, and switching logic across multiple models. Coordinating and optimizing interactions between models constitutes a complex process requiring meticulous design and debugging.</p> <p>(3) Accurate parameter estimation for multiple models and effective model switching typically require substantial data support. In scenarios with data scarcity, the performance of multi-model methods may be constrained.</p>

To date, the MME method has been extensively utilized in diverse fields, including integrated navigation data fusion, self-calibration, and target tracking (Luan et al., 2021; Tian et al., 2022; Yang et al., 2024).

2.4. Factor Graph (FG) Methods

In the filtering method mentioned earlier, the state of the system at the current moment is only related to the observations at the current moment and the navigation state at the previous moment. However, in practical applications, some observations may be delayed, and certain position solutions need to be realized over a period before and after the observations. This cannot be described solely by the state information of the current and previous moments. For example, the EKF converts historical observations into prior information of the current state through the state equation for propagation, and the state linearization points corresponding to the historical observations after propagation are fixed. When there is an undetected blunder in the historical observations, the linearization point error is large, which can easily lead to the prior information being contaminated, thereby affecting positioning accuracy. In contrast, the FG can fully utilize historical observations by iteratively updating the linearization points, mining the constraint information of the observations in the temporal dimension, and suppressing the influence of blunders.

FG decomposes a complex global function involving multiple variables into the product of several simpler local functions, thereby constructing a bidirectional graph structure. When dealing

with global functions that involve numerous variables, the conventional approach is to break down the given function into its constituent factors, which serve as local functions. These local functions are then combined multiplicatively to represent the original global function. The FG is a bipartite graph model that elegantly captures this factorization process. It typically consists of variable nodes, factor nodes, and edges connecting them. By leveraging this structure, FG can effectively decompose multivariable functions into products of local functions, facilitating more efficient computation and analysis.

$$g(x_1, x_2, \dots, x_n) = \prod_{j \in J} f_j(X_j) \quad (44)$$

where x_n denotes a variable node, the local function f_j denotes a factor node, denotes the independent variable associated with the local function $f_j(\cdot)$, and subsequently, the factor node is connected to the corresponding variable node via an edge that represents their mutual relationship.

Let $g(x_1, x_2, x_3, x_4, x_5)$ be a global function involving five variables. It can be decomposed into the product of four factors, which can be expressed as follows:

$$g(x_1, x_2, x_3, x_4, x_5) = f_1(x_1, x_3) f_2(x_1, x_2, x_4) f_3(x_3, x_4) f_4(x_4, x_5) \quad (45)$$

The corresponding FG structure is shown in Figure 4.

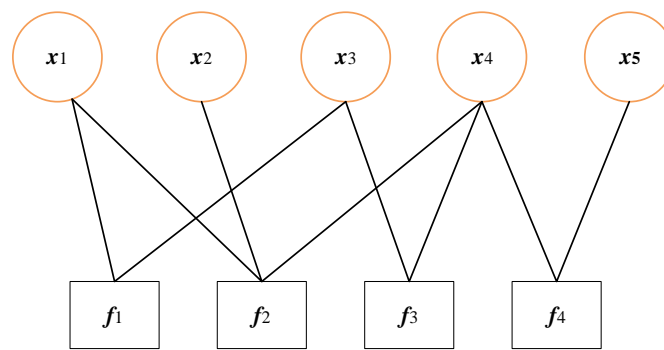


Figure 4. Structure of FG.

Figure 5 establishes a factor graph-based multi-sensor fusion framework, incorporating sensors including IMU, GPS, barometric altimeter, optical flow sensor, magnetic heading sensor, and star tracker.

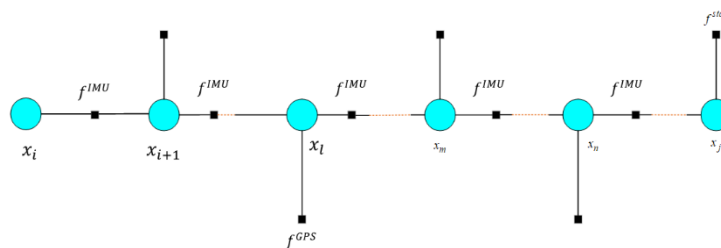


Figure 5. Multi-sensor fusion framework based on FG.

In Figure 5, the large circles represent variable nodes, and the black solid small squares represent factor nodes. The variables of the functions associated with the factor nodes include the variable nodes connected to them. Each factor has a corresponding error function. By adjusting the \mathbf{x} to minimize the error of the FG, the optimal estimate can be obtained. The formula is as follows:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \left(\prod_i g_i(x_i) \right) \quad (46)$$

In the integrated system, the factor node constructs the corresponding function to calculate the difference between the predicted measurement and the actual measurement, thereby obtaining the estimate of the state variable and the cost function, that is:

$$f_i(x_i)=L[g_i(x_i)-z_i] \tag{47}$$

where z_i is the actual measurement value obtained by the sensor, and $L(\cdot)$ is the cost function.

FGs model the optimal estimation problem in a graphical form and solve for state estimates based on the MAPE criterion. During the optimization process, as the system operates over time, the scale of the graph inevitably increases, leading to a decline in the real-time performance of graph optimization. Therefore, a sliding window mechanism is introduced to balance accuracy and efficiency, thereby enhancing the system real-time capabilities. Table 5 summarizes the performance characteristics of the FG method.

Table 5. Performance Characteristics of the FG.

Method	Advantages	Disadvantages
FG	<p>(1) High Flexibility: FG can more generally represent the decomposition of probability distributions and apply to a variety of complex probabilistic models. It can flexibly integrate data from different types of sensors (such as IMU, GPS, LiDAR, etc.) and introduce multiple constraints.</p> <p>(2) Effective Optimization: FG is suitable for large-scale sparse structure optimization and offers better numerical stability, especially in SLAM problems. It optimizes through Bayesian inference and can effectively handle multi-source heterogeneous data.</p> <p>(3) Strong Dynamic Adaptability: FG can dynamically adjust optimization strategies to adapt to dynamic changes in data, especially when dealing with inconsistent sensor information frequencies and dynamically changing validity.</p> <p>(4) Plug-and-Play: FG has a plug-and-play feature, allowing easy integration of new sensor data and strong scalability.</p>	<p>(1) High computational complexity: FG has high computational complexity when dealing with large-scale data, which may limit its real-time applications.</p> <p>(2) High data preprocessing requirements: Multisource data usually needs to be preprocessed, such as data cleaning and normalization, to ensure the quality and consistency of the data.</p> <p>(3) Difficulty in model construction: Constructing an effective FG model requires a deep understanding of the uncertainty and correlation of the data, which increases the difficulty of model construction.</p> <p>(4) Real-time challenges: In some scenarios that require real-time decision-making, the computational complexity may limit its application.</p>

The FG algorithm has been widely applied in various domains, including single GNSS positioning, GNSS and INS integrated positioning, ambiguity resolution, and robust estimation (Watson et al., 2018; Wen et al., 2021; Gao et al., 2022; Zhang et al., 2024). The FG method has been optimized and improved for urban environments to enhance positioning accuracy (Lyu et al., 2021; Bai et al., 2022; Liao et al., 2023; Zhang et al., 2023), and it has been demonstrated that under certain conditions, the FG algorithm achieves higher solution accuracy and robustness compared to the EKF.

2.5. Artificial Intelligence (AI) Method

In addition to the aforementioned fusion methods, machine learning and deep learning have found typical applications in practice, demonstrating excellent performance in fields such as pattern recognition and image processing. Their learning and training concepts have also been applied to position estimation in multi-source integrated navigation, particularly in scenarios where explicit mathematical models cannot be established between navigation source observations and system

states. Such as in WLAN positioning, carrier positions cannot be directly calculated from signal strength, and in visual cooperative target localization, carrier positions cannot be directly derived from acquired image tags. However, through offline learning of signal strength or image tags, a learning model can be established between system states and observational information. Thus, in real-time positioning, carrier positions can be directly computed via image tags. By inputting acquired observational data into the trained learning model, system states can be output, enabling multi-source information fusion processing. Table 6 summarizes the advantages and disadvantages of AI methods.

Table 6. Performance Characteristics of AI Method.

Method	Advantages	Disadvantages
AI	<p>(1) Strong Dynamic Adaptability: AI methods (such as machine learning, deep learning, and reinforcement learning) can dynamically learn and adapt to environmental changes, automatically adjust fusion strategies, and overcome the limitations of traditional methods in handling nonlinearity, time-varying characteristics, and uncertainties.</p> <p>(2) Enhanced Capability in Handling Complex Data Relationships: Through model training, AI algorithms can process complex multi-source heterogeneous data relationships, improving fusion accuracy and system robustness.</p> <p>(3) Automatic Feature Extraction: Neural networks and similar algorithms can automatically learn data features, reducing the need for manual feature engineering, especially for large-scale, high-dimensional data.</p> <p>(4) Improved Decision-Making Efficiency: AI algorithms can rapidly integrate information from diverse sensors or data sources, providing support for real-time decision-making.</p>	<p>(1) High computational costs: Multimodal data fusion models require processing information from multiple data streams, demanding significant computational resources (e.g., GPUs) and energy consumption, which may constrain real-time applications.</p> <p>(2) Demanding data preprocessing requirements: Multi-source data often exhibit discrepancies in formats, scales, semantics, and quality, necessitating preprocessing steps such as data cleaning and standardization, thereby increasing implementation complexity.</p> <p>(3) High model complexity: AI models (e.g., deep learning models) are typically intricate, requiring substantial time for training and optimization, while also demanding high data volume and quality.</p> <p>(4) Data privacy and security concerns: Data sources are diverse and may contain sensitive information, making privacy protection and security safeguarding critical challenges.</p>

The AI-based algorithms in multi-source data fusion mainly include Neural Networks (NN), Support Vector Machines (SVM), Hidden Markov Models (HMM), Decision Trees, etc. (McClelland et al., 1987; Vapnik, 1999; Zhou, 2016).

NN is one of the most classical AI algorithms. They can be classified based on network topology and information flow direction. When categorized by topological structure, NN models can be divided into hierarchical structures, interconnected structures, and sparsely connected structures according to the connection methods between neurons. Based on the direction of internal information flow, they can be classified as feedforward or feedback neural networks. The learning methods of artificial NN mainly fall into three categories: supervised learning, unsupervised learning, and rote learning. Among them, the classic Backpropagation (BP) neural network belongs to the purely hierarchical type with a feedforward information flow direction and employs supervised learning. Its typical three-layer network model is shown in Figure 7.

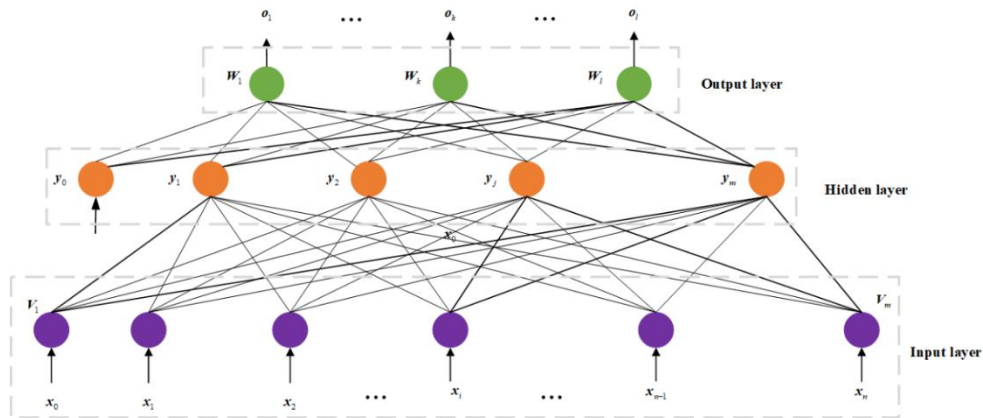


Figure 7. Three-layer network structure of the BP neural network.

The vectors for each layer are denoted as X , Y , and O , respectively. In a multi-source fusion navigation system, the data vector $X = [x_0 \ x_1 \ \dots \ x_{n-1} \ x_n]$ refers to the observation information from different sources. The hidden layer vector is denoted as $Y = [y_0 \ y_1 \ \dots \ y_{n-1} \ y_n]$, and the output layer vectors are denoted as $O = [o_1 \ \dots \ o_k \ \dots \ o_l]$. The output can represent the system state, such as position, or large-scale location identifiers. The connection observation noise and weight matrix are denoted as $V = [v_1 \ \dots \ v_k \ \dots \ v_m]$ and $W = [w_1 \ \dots \ w_k \ \dots \ w_l]$, respectively.

BP neural network algorithm consists of a training process and a learning process, as shown in Figure 8.

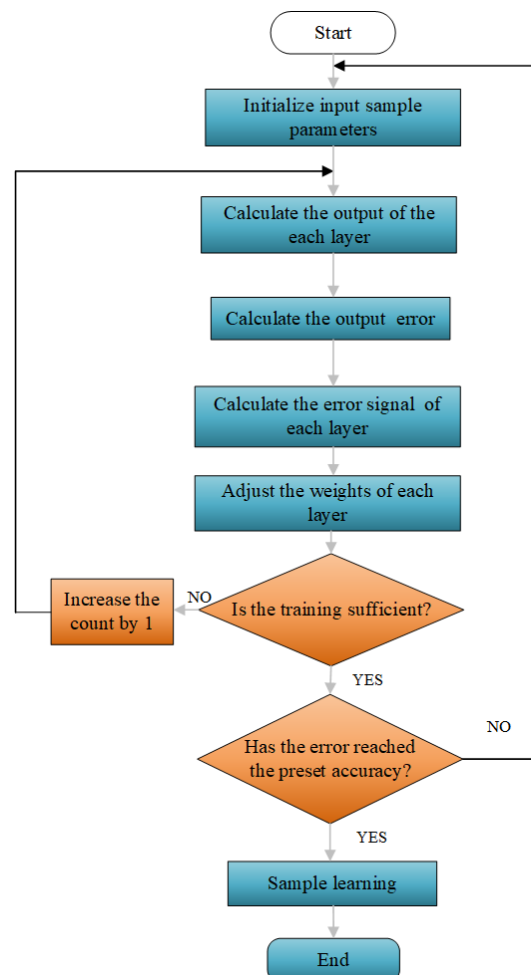


Figure 8. Flowchart of the BP Neural Network Algorithm.

Scholars have conducted research on AI-based adaptive algorithms, including fuzzy control adaptive algorithms and neural network adaptive algorithms (Bian et al., 2004). For instance, to address the impact of random disturbances in underwater environments on systems, a Radial Basis Function (RBF) neural network was employed to assist FF for information fusion (Li et al., 2011). In response to the challenges of high costs and significant susceptibility to meteorological conditions in existing high-precision satellite navigation systems for agricultural machinery in farmlands, Yu et al. (2021) proposed a D-S-CNN-based multi-sensor fusion automatic navigation method for farmland applications. These existing AI algorithms require a large amount of training data and pre-scaled training of the system, which makes it difficult to ensure computational capacity and real-time performance. Therefore, they are mostly used for post-computation.

2.6. Methods Based on Uncertain Reasoning

Some fusion methods are not suitable for establishing a mathematical model between measurements and system states, nor can they directly estimate the state. However, they can be used to assess the reliability of navigation sources and determine large-scale locations. For example, the identification of specific large-scale locations such as classrooms, cafes, conference rooms, offices, and residences can be achieved by combining observations from GNSS, WLAN sources, and inertial sources to perform uncertainty reasoning and obtain large-scale location information. Therefore, a multi-source information fusion method based on uncertainty reasoning is proposed. This method requires extracting evidence from measurement data and using relevant knowledge (mainly expert knowledge) to gradually derive conclusions from the evidence or to verify the credibility of specific information. The currently mainstream uncertainty reasoning methods mainly include subjective Bayesian estimation, evidence reasoning, and fuzzy reasoning (Zhang, 1996). This method can also adjust the weights of each measurement data in the final fusion result to achieve precise identification of locations.

Uncertainty reasoning includes symbolic reasoning and numerical reasoning. The former, such as Endorsement Theory, is characterized by minimal information loss during the reasoning process but involves a large computational burden. The latter, such as Bayesian reasoning and Dempster-Shafer theory of evidence, is characterized by ease of implementation but involves some degree of information loss during the reasoning process. As uncertainty reasoning methods are fundamental tools for target recognition and attribute fusion, the subjective Bayes method and the Dempster-Shafer (D-S) evidence theory are two commonly used uncertainty reasoning approaches. For specific applications, please refer to related literature. To facilitate future use, Table 7 summarizes the advantages and disadvantages of uncertainty reasoning methods.

Table 7. Advantages and Disadvantages of Uncertainty reasoning.

Method	Advantages	Disadvantages
Uncertainty reasoning	(1) Strong capability in handling uncertainty: Uncertainty reasoning methods (such as Dempster-Shafer theory of evidence) can effectively deal with uncertainty in data, including imprecision, inconsistency of data, and sensor errors.	(1) High computational complexity: Uncertainty reasoning methods typically require complex computational processes, especially when dealing with large-scale data, resulting in high computational costs.
	(2) High flexibility: These methods can flexibly handle different types of data sources and meet the needs of fusing multi-source heterogeneous data.	(2) High requirements for data preprocessing: Multi-source data often needs to be preprocessed, such as data cleaning and normalization, to ensure the quality and consistency of the data.
	(3) Enhanced decision-making reliability: By properly dealing with uncertainty, the	

reliability of the fusion results can be improved, providing support for decision-making in complex environments. (4) Support for dynamic updates: In scenarios where data is constantly changing, uncertainty reasoning methods can dynamically adjust the fusion strategy to adapt to new data inputs.	(3) Difficulty in model construction: Building effective uncertainty models requires a deep understanding of the uncertainty and correlation of the data, which increases the difficulty of model construction. (4) Real-time challenges: In scenarios that require real-time decision-making, the computational complexity may limit their application.
--	---

3. Representative Applications of Partial Fusion Methodology in Navigation and Positioning

As the author primarily focuses on research in the field of navigation and location services, only typical applications of some fusion methods in this domain are listed. Contemporary research has demonstrated significant advancements in rapid high-precision navigation for complex urban environments through sensor fusion strategies. Scholars have systematically explored three principal approaches: multi-sensor integration, functional model optimization, and stochastic modeling enhancement. Noteworthy implementations include:

(1) Multi-constellation Hybridization: Wang et al. (2022) developed a train-integrated navigation system leveraging full-state fusion of multi-constellation GNSS and INS. This architecture demonstrates enhanced robustness against signal degradation in metropolitan corridors.

(2) Advanced Tight Coupling Methods: Zhu et al. (2023) proposed a MEMS IMU error mitigation framework combining GNSS RTK carrier phase, TDCP observations, and INS data through adaptive KF. Li et al. (2017) augmented this paradigm with carrier motion constraints, significantly improving solution availability during prolonged GNSS outages.

(3) Precision Positioning Enhancements: Gao (2016) established that multi-GNSS PPP/INS tight coupling reduces PPP convergence time by 40-60% while maintaining centimeter-level accuracy. Urban navigation breakthroughs by Li et al. (2018) revealed that single-frequency RTK/INS integration achieves dual-frequency RTK performance (2-3 cm horizontal RMS) under typical urban multipath conditions.

(4) Multi-Sensor Constraint Integration: Liu (2021) incorporated odometric measurements and zero-velocity constraints into GNSS/INS fusion, demonstrating 75% improvement in INS drift suppression during 60-second GNSS outages. Indoor-outdoor continuity solutions by Yu et al. (2020), Cao (2021), and Yuan et al. (2023) achieved seamless decimeter-level positioning through UWB/GNSS/INS tight coupling.

(5) Resilient Navigation Architectures: Li et al. (2023) engineered a vision-INS-UWB hybrid system where UWB supplants GNSS in denied environments, reducing visual-inertial odometry drift by 32% in extended indoor operations. Emerging multi-sensor SLAM frameworks (Wang et al., 2022) integrate lidar, vision, and inertial data with crowd-sourced mapping, showing particular promise for dynamic urban canyon navigation.

This methodological evolution highlights the critical role of partial fusion techniques in balancing computational complexity with positioning integrity across heterogeneous operational environments.

4. Summary of Features and Application Scenarios for Multi-Source Fusion Methods

The introduced methods for multi-source data fusion processing show differences in terms of optimization criteria, fundamental principles, mathematical models, prior information, number of observations, and application scenarios. To enable users to make targeted selections of different

fusion methods when developing integrated systems, we have summarized the main characteristics and applicable scenarios of the fusion methods introduced, as shown in Table 8.

Table 8. Main Characteristics and Applicable Scenarios of Fusion Method.

Type	Method	Main Characteristics	Applicable Scenarios
Optimal estimation	LSE, WLSE	(1) Parameters are estimated by minimizing the sum of squared errors. (2) In linear regression models, the LSE is unbiased and achieves the minimum variance among all linear unbiased estimators (i.e., it is BLUE, the Best Linear Unbiased Estimator). (3) It is sensitive to outliers because its optimization relies on minimizing the sum of squared errors. (4) It is suitable for data with linear relationships, offering computational simplicity and ease of implementation.	Linear regression and curve fitting. If the data distribution is unknown and the model is simple, it is preferable to prioritize LSE.
	MLE	(1) Parameters are estimated by maximizing the likelihood function, relying solely on observed data without considering prior information. (2) For large sample sizes, MLE typically exhibits consistency (estimates converge to the true values as the sample size increases). (3) The computational complexity is low, and solutions can generally be obtained through analytical methods or numerical optimization. (4) MLE is asymptotically efficient when model assumptions hold but may fail when these assumptions are violated.	Highly versatile and suitable for large samples, but computationally intensive. If the dataset is large and the distribution is known, MLE should be prioritized.
	Optimal estimationMAPE	(1) Combines the likelihood of observed data with prior information about parameters, representing a Bayesian estimation method. (2) MAPE generally outperforms MLE when prior information is reliable and sample sizes are small. (3) MAPE Can be interpreted as a regularized version of MLE, where the prior distribution acts as a regularization term. (4) Computational complexity may be high, particularly with complex prior distributions or when numerical optimization is required.	Combines prior information, suitable for small samples, but relies on prior selection. If the sample size is small and there is a need to incorporate prior information, the MAPE is chosen.
	MVE	(1) Among all unbiased estimators, it has the minimum variance, making it the optimal unbiased estimator. (2) It typically requires assumptions about the data distribution (e.g., normal distribution), and its optimality is guaranteed under these assumptions. (3) It exhibits higher computational complexity, particularly with high-dimensional data.	
	LMVE	(1) It is the estimation method with the minimum variance among all linear estimators. (2) It only requires knowledge of the first and second-order moments of the estimated quantity and measured quantity, making it suitable for stationary processes. (3) For non-stationary processes, precise knowledge of the first and second-order moments at each time instant is required, which significantly constrains its applicability. (4) The computational complexity is moderate, but the estimation accuracy critically depends on the accuracy of the assumed moments	In the case where the model is known and unbiased estimation is required, select either theMVE or the LMVE.
Filtering methods	KF	(1) Linear System Assumption: The KF assumes both the system model and observation model are linear, with additive Gaussian noise. This linear-Gaussian assumption theoretically guarantees a globally optimal solution. (2) High Computational Efficiency: The KF exhibits low computational complexity ($O(n^2)$ for state dimension n), making it well-suited for real-time applications with stringent timing requirements. (3) Optimal Estimation Accuracy: Under strict adherence to linear-Gaussian assumptions, the KF provides statistically optimal estimates in the minimum mean-square error sense.	It is suitable for scenarios where the system is linear and the noise follows a Gaussian distribution, such as in simple navigation and signal processing.

	<p>(4) Limited Applicability: The KF's strict linearity assumptions lead to significant estimation errors when applied to nonlinear systems, severely constraining its practical application scope.</p>	
EKF	<p>(1) Nonlinear system processing: The EKF approximates nonlinear problems by linearizing the nonlinear system model.</p> <p>(2) Moderate computational complexity: The computational complexity of the EKF is higher than the KF but lower than the UKF.</p> <p>(3) Limited accuracy: Due to the linearization process, the EKF may experience larger estimation errors in strongly nonlinear systems.</p> <p>(4) Jacobian matrix calculation: The EKF requires computing the Jacobian matrices of the system model and observation model, which increases the algorithm complexity.</p> <p>(5) Broad applicability: The EKF is a widely used method for handling nonlinear systems and is suitable for moderate nonlinearity.</p>	<p>It is suitable for scenarios where the nonlinearity is not high and fast real-time processing is required.</p> <p>Due to its lower computational complexity, the EKF is well-suited for use in embedded systems with limited computational resources.</p> <p>When the initial state estimation is relatively accurate, the EKF can converge quickly and provide better estimation results.</p>
UKF	<p>(1) Strong nonlinear system processing capability: The UKF selects a set of deterministic sampling points (Sigma points) through the UT, enabling a more accurate approximation of the statistical properties of nonlinear systems.</p> <p>(2) No Jacobian matrices required: The UKF avoids the complex Jacobian matrix calculations required in the EKF, enhancing the algorithm stability and precision.</p> <p>(3) Higher computational complexity: The computational complexity of the UKF is higher than that of the EKF, but it generally remains within acceptable limits.</p> <p>(4) High accuracy: The UKF outperforms the EKF in strongly nonlinear systems, providing more accurate estimation results.</p>	<p>Suitable for scenarios involving nonlinear systems with Gaussian noise, such as complex target tracking and robotic navigation</p>
PF	<p>(1) Strong nonlinear and non-Gaussian adaptability: PF is a recursive Bayesian estimation technique based on Monte Carlo methods, capable of handling complex nonlinear systems and non-Gaussian noise environments. It approximates the posterior probability distribution through a large number of random samples (particles), thus avoiding the linearity and Gaussian assumptions required by traditional filtering methods (e.g., KF).</p> <p>(2) Flexibility and robustness: PF exhibits high flexibility, adapting to diverse systems and observation models. It demonstrates strong robustness against noise and outliers, making it particularly suitable for dynamic systems in complex environments.</p> <p>(3) Parallel computing capability: The computational process of PF can be parallelized, as the prediction and update operations for each particle are executed independently, significantly improving computational efficiency.</p> <p>(4) Handling complex probability distributions: PF can manage multimodal probability distributions, making it ideal for state estimation in complex scenarios. Through its particle weight updating mechanism, it effectively integrates information from multiple sensors.</p> <p>(5) Challenges and optimizations: Despite its adaptability, PF faces challenges such as particle degeneracy and high computational complexity. To address these issues, researchers have proposed improvements like adaptive PF, combining the UKF to generate proposal distributions, and optimizing resampling strategies.</p>	<p>Suitable for nonlinear and non-Gaussian systems, such as target tracking, robotic navigation, and signal processing.</p>
UPF	<p>(1) Adaptability to non-Gaussian noise: When combined with PF, the UKF further enhances its adaptability to non-Gaussian and multimodal probability distributions.</p> <p>(2) Balanced computational efficiency and accuracy: The UKF achieves a good balance between computational efficiency and accuracy. While its computational complexity is higher than that of the traditional KF, the UKF outperforms the EKF in handling high-dimensional state spaces and has lower computational complexity compared to PF. By integrating PF, particles are generated and updated via the UT, further improving particle effectiveness and sampling efficiency.</p> <p>(3) Mitigation of particle degeneracy: In PF, particle degeneracy—a common issue where most particle weights approach zero,</p>	<p>Suitable for nonlinear and non-Gaussian systems where high estimation accuracy is required.</p>

	<p>reducing the number of effective particles—is alleviated when combined with the UKF. Sigma points generated through the UT provide a more accurate description of the state distribution, thereby reducing particle degeneracy.</p> <p>(4) Further optimization via PF integration: The UPF combines the strengths of the UKF and PF. By generating and updating particles through the UKF UT, UPF retains PF strong adaptability to nonlinear and non-Gaussian problems while improving sampling efficiency.</p>	
FF	<p>(1) Distributed Structure and Flexibility: FF adopts a distributed architecture, decomposing data fusion tasks into multiple sub-filters and a master filter. This structural design offers flexibility, allowing the selection of appropriate filtering algorithms (e.g., EKF, UKF) based on the characteristics of different sensors. It also supports dynamic adaptation to sensors joining or leaving the network.</p> <p>(2) Fault Tolerance and Fault Isolation Capability: FF can detect and isolate faulty sensors in real time, preventing erroneous data from degrading global estimation accuracy. This ensures high precision and reliability even when partial sensor failures occur.</p> <p>(3) Diversity of Information Allocation Strategies: FF supports multiple information allocation strategies, including zero-reset mode, variable proportion mode, feedback-free mode, and fusion-feedback mode. These strategies exhibit trade-offs in computational complexity, fusion accuracy, and fault tolerance, enabling flexible selection tailored to specific application scenarios.</p> <p>(4) High Computational Efficiency: By distributing data processing among multiple sub-filters, each of which handles only local data, FF reduces the computational load. This distributed computing approach not only enhances the system real-time performance but also lowers the demand for hardware resources.</p> <p>(5) Adaptive and Dynamic Adjustment Capability: Integrated with adaptive filtering theory, FF dynamically adjusts information allocation coefficients based on sensor performance and environmental changes.</p> <p>(6) Plug-and-Play Functionality: FF supports plug-and-play operation for sensors, enabling rapid adaptation to dynamic sensor addition or removal. This feature enhances flexibility and adaptability in complex environments or evolving mission requirements.</p> <p>(7) Globally Optimal Estimation: The master filter synthesizes local estimates from sub-filters to achieve a globally optimal estimate. This two-tier architecture preserves local filtering precision while improving overall system performance through global fusion.</p> <p>(8) Suitability for Complex Environments: FF excels in multi-source heterogeneous data fusion scenarios, such as integrating satellite navigation, inertial navigation, and visual navigation in navigation systems. It effectively addresses sensor noise, model biases, and environmental interference challenges.</p>	<p>Suitable for multi-sensor networks, especially in scenarios where sensors are widely distributed and communication resources are limited.</p>
MME	<p>(1) Multimodal Data Processing Capability: MME methods can handle data from different modalities, such as images, text, audio, and video. By mapping multimodal data into a unified feature space or integrating information through fusion techniques, these methods leverage the complementary nature of different modalities to enhance model performance.</p> <p>(2) Diversity of Fusion Strategies: MME methods support various fusion strategies, including early fusion (integration at the data level), mid-level fusion (integration at the feature level), late fusion (integration at the decision level), and hybrid fusion (combining multiple strategies). Different strategies are suited to different application scenarios, enabling flexible adaptation to complex data fusion requirements.</p> <p>(3) Enhanced Model Robustness: By incorporating multimodal data and multi-model estimation, uncertainties arising from single-</p>	<p>(1) In scenarios such as multi-object tracking and complex trajectory prediction, multi-model estimation methods can effectively deal with the various motion patterns and uncertainties of targets.</p> <p>(2) Suitable for integrated navigation systems, such as the fusion of inertial navigation and satellite navigation, multi-model estimation can improve positioning accuracy and reliability.</p> <p>(3) In complex industrial scenarios such as power systems and chemical processes, MME methods can be used for modeling and control to</p>

	<p>modality data can be effectively reduced, improving the model's robustness against noise and outliers.</p> <p>(4) Improved Performance and Generalization Capability: MME methods comprehensively capture target information by fusing multimodal data, thereby boosting both model performance and generalization ability.</p> <p>(5) Flexibility in Model Architecture: MME methods typically offer high flexibility, allowing the selection of appropriate model architectures based on specific task requirements. For example, unified embedding-decoder architectures and cross-modal attention architectures can be combined to fully exploit the advantages of different structural designs.</p>	<p>address the dynamic changes in system parameters.</p>
FG	<p>(1) Intuitiveness and Flexibility: FG visually represents the relationships between variables and factors, making complex probabilistic models more intuitive and easier to understand. Compared to Bayesian networks, FG can more generally express the decomposition of probability distributions, making them suitable for a wide range of complex probabilistic models.</p> <p>(2) Efficient Inference and Optimization Capabilities: FGs excel in algorithms such as variable elimination and message passing, significantly improving inference efficiency. By jointly optimizing the relationships between data from different sensors, FG can efficiently fuse multi-sensor data.</p> <p>(3) Strong Multi-Source Data Fusion Capability: FG can effectively integrate data from various sensors (such as cameras, radars, and IMUs). By constructing an FG model, they can jointly optimize heterogeneous multi-source data, thereby enhancing the accuracy and reliability of data fusion.</p> <p>(4) Support for Dynamic Data Processing: FG algorithms support techniques such as sliding-window optimization (Sliding-Window FGO), enabling dynamic processing of real-time data streams. This makes them suitable for scenarios with high computational complexity and stringent real-time requirements.</p> <p>(5) Robustness and Outlier Resistance: In complex environments (such as urban canyons), FGO demonstrates better robustness and positioning accuracy compared to traditional methods like the EKF.</p>	<p>(1) Autonomous Driving and Robot Navigation: FG methods are widely used in autonomous driving and robot navigation to integrate data from multiple sources such as cameras, radar, LiDAR, and IMU, thereby enhancing the accuracy of localization and mapping.</p> <p>(2) Underwater Robot Localization: In complex underwater environments, FG methods can effectively handle abnormal sensor observations, thereby improving the positioning accuracy of Autonomous Underwater Vehicles.</p> <p>(3) Industrial Fault Detection and Diagnosis: FG methods can be used to integrate monitoring data from different sensors to achieve early warning and diagnosis of equipment faults.</p> <p>(4) Medical Diagnosis: In the medical field, FG methods can integrate patients' medical records, imaging data, and laboratory test results to improve the accuracy of disease diagnosis.</p> <p>(5) Environmental Monitoring: FG methods can integrate data from satellites, ground sensors, and meteorological data to monitor environmental changes in real time, supporting environmental management and decision-making.</p>
AI	<p>(1) Heterogeneous Data Processing Capability: The AI method can handle data from diverse sources, formats, and structures (such as text, images, and sensor data) through multimodal models (e.g., Transformers, hybrid neural networks) to achieve unified processing.</p> <p>(2) Automatic Feature Extraction and Representation Learning: Deep learning models (e.g., autoencoders, BERT) autonomously learn high-level abstract features without manual feature engineering, adapting to the complexity of multi-source data. This reduces reliance on domain-specific knowledge while improving feature expression efficiency and generalization.</p> <p>(3) Robustness and Fault Tolerance: AI maintains stability amid data noise, missing values, or conflicts by leveraging adversarial training, data augmentation, and attention mechanisms to filter redundant information.</p> <p>(4) Additional capabilities include cross-modal association and reasoning, real-time performance and scalability, and adaptability and dynamic updating.</p>	<p>(1)Autonomous Driving: Integrating data from cameras, radar, and LiDAR helps vehicles understand their surroundings from multiple perspectives, enabling safe navigation.</p> <p>(2)Medical Diagnosis: Fusing patient medical records with medical imaging data allows for the rapid and accurate identification of diseases, thereby improving diagnostic efficiency.</p> <p>(3) Smart Cities: In real estate management, combining sensor data, social media feedback, and Geographic Information System (GIS) data optimizes urban spatial layout and resource allocation.</p> <p>(4)Industrial Internet of Things (IIoT): Analyzing sensor data</p>

		enables predictive maintenance and optimization of equipment, thereby increasing production efficiency.
		(5) Environmental Monitoring: Integrating satellite data, ground sensor data, and meteorological data allows for real-time monitoring of environmental changes, supporting environmental management and decision-making.
Uncertain reasoning	(1) Strong capability in handling uncertainty: Uncertainty reasoning methods can effectively deal with uncertainty in data, including imprecision, inconsistency, and sensor errors. For example, Dempster-Shafer theory of evidence can model and fuse uncertain information, while Bayesian methods handle uncertainty through probabilistic updates.	(1) Target Recognition and Tracking in Complex Environments: In target recognition and tracking tasks, uncertainty reasoning methods can effectively handle uncertain data from multiple sensors (such as cameras and radars), thereby improving the accuracy and reliability of target detection. (2) Medical Diagnosis: In the medical field, by integrating patients' medical records, imaging data, and laboratory test results, uncertainty reasoning methods can more accurately identify diseases and enhance the accuracy of diagnosis.
	(2) High flexibility: Uncertainty reasoning methods can flexibly handle different types of data sources and meet the needs of fusing multi-source heterogeneous data. They are capable of integrating symbolic reasoning and numerical reasoning to process various types of information, ranging from qualitative to quantitative.	
	(3) Enhanced decision-making reliability: By properly handling uncertainty, uncertainty reasoning methods can improve the reliability of the fusion results and provide support for decision-making in complex environments.	
	(4) High computational complexity, high requirements for data preprocessing, difficulty in model construction, and challenges in real-time performance.	

5. Conclusions

This paper provides a detailed overview of various algorithms corresponding to multi-source fusion processing methods. It summarizes the fundamental principles of these algorithms and briefly introduces their mathematical models, key characteristics, and application scenarios, offering significant theoretical and technical support for intelligent navigation, unmanned driving, autonomous navigation, and related fields. Due to limitations in theoretical understanding and technical conditions, all existing fusion algorithms exhibit shortcomings. Currently, no single fusion algorithm can fully meet the requirements of multi-source integrated navigation systems. Therefore, appropriate fusion algorithms must be selected based on practical needs and application contexts. The historical development of these fusion algorithms reveals their interdisciplinary nature, combining theories and methodologies from integrated navigation, GNSS data processing, satellite geodesy, probability theory and mathematical statistics, computer science, statistics, and artificial intelligence. Consequently, multi-source integrated navigation algorithms should not be confined to traditional positioning and navigation approaches. Instead, they should continuously incorporate insights from other disciplines, foster mutual learning and advancement across fields, and generate innovative theories and methods through interdisciplinary integration. This evolution aims to deliver high-precision, high-reliability positioning, navigation, and timing services across all temporal and spatial domains - representing the future development trend of multi-source integrated navigation systems.

6. Acknowledgment

This research was funded by the National Natural Science Foundation of China (42364002, 42274039), the Major Discipline Academic and Technical Leaders Training Program of Jiangxi Province (20225BCJ23014), the Key Research and Development Program Project of Jiangxi Province (20243BBI91033), Xi 'an Science and Technology plan Project (24ZDCYJSGG0015), and State Key Laboratory of Satellite Navigation System and Equipment Technology (CEPNT2023B02), and Chongqing Municipal Education Commission Science and Technology Research Project (KJQN202403241).

References

1. Ackerson G, Fu K. On-state estimation in switching environments. *IEEE Trans. AC*, 1970,15(1): 10-17
2. Alsamhi, Saeed Hamood, et al. Survey on Federated Learning enabling indoor navigation for industry 4.0 in B5G. *Future Generation Computer Systems*, 2023,148: 250-265
3. Andrieu C, Doucet A, Singh S, Tadic V. Particle methods for change detection, system identification and control. *Proceedings of the IEEE*, 2004, 92(3): 428-438
4. Athans M, Wisher R, Bertolini A. Suboptimal state estimation algorithm for continuous-time nonlinear systems from discrete measurements. *IEEE Transactions on Automatic Control*, 1968, AC-13(5): 504-515
5. Bi X, HU S. Firefly Algorithm with high Precision Mixed Strategy optimized particle Filter. *Journal of Shanghai Jiaotong University*, 2019, 53(2): 232-238
6. Athans M, Wisher R, Bertolini A. Suboptimal state estimation algorithm for continuous-time nonlinear systems from discrete measurements. *IEEE Transactions on Automatic Control*, 1968, AC-13(5): 504-515
7. Bi X, HU S. Firefly Algorithm with high Precision Mixed Strategy optimized particle Filter. *Journal of Shanghai Jiaotong University*, 2019, 53(2): 232-238
8. Bian H, Jin Z, Tian W. Analysis of adaptive Kalman filter based on intelligent information fusion techniques in integrated navigation system. 2004(10): 1449-1452
9. Bierman G, Belzer M. A decentralized square root information filter/Smoothers. *Proc. NAECON*, Dayton, OH, 1987. 1448-1456
10. Cahyadi N, Asfihani T, Suhandri F, et al. Unscented Kalman filter for a low-cost GNSS/IMU-based mobile mapping application under demanding conditions. *Geodesy and Geodynamics*, 2024, 15(02): 166-176
11. Carlson N A. Federated filter for fault-tolerant integrated navigation systems. *Proc. Of IEEE PLANS 88*, Orlando, FL 1988, 110-119
12. Cao Z. Indoor and outdoor seamless positioning based on the Integration of GNSS/INS/UWB. Master's Thesis from Beijing Jiaotong University, 2021
13. Chen K, Chang G. & Chen C. GINav: a MATLAB-based software for the data processing and analysis of a GNSS/INS integrated navigation system. 2021, *GPS Solut* 25, 108
14. Cheng Q, Liu H, Shen H, et al. A spatial and temporal nonlocal filter-based data fusion method. *IEEE Transactions on Geoscience and Remote Sensing*, 2017, 55(8): 4476-4488
15. Chi C, Zhang X, Liu J, Sun Y, Zhang Z, and Zhan X. GICI-LIB: A GNSS/INS/Camera Integrated Navigation Library. *arXiv preprint*, arXiv: 2306.13268
16. Chiu H, Zhou X, Carlone L, et al. Constrained optimal selection for multi-sensor robot navigation using plug-and-play factor graphs. *Proceedings of the IEE International Conference on Robotics and Automation*, 2014: 663-670
17. Crisan D, Ducet A. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. on signal processing*, 2002, 50 (3): 736-746
18. Cui X, Yu Z, Tao B, et al. Generalized surveying adjustment. Wuhan: Wuhan University Press, 2001
19. Dan Simon. Optimal state estimation-Kalman, H^∞ and nonlinear Approaches. John Wiley & Sons, 2006
20. Dellaert F, Kaess M. Square root SAM: simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 2006, 25(12): 1181-1203
21. Doucet A, De Freitas N, Gordon N. Sequential Monte Carlo Methods in practice. Springer- Verlag, New York, 2001
22. Gao H, Li H, Huo H, et al. Robust GNSS real-time kinematic with ambiguity resolution in factor graph optimization//*Proceedings of the 2022 International Technical Meeting of The Institute of Navigation*. 2022: 835-843
23. Gao Z. Research on the methodology and application of the integration between the multi-constellation GNSS PPP and inertial navigation system. Wuhan University Doctoral Dissertation, 2016
24. Gray J, Muray W. A derivation of an analytical expression for the tracking index for the alpha-beta-gamma filtering. *IEEE Trans. on Aerospace and Electronic Systems*, 1993, 29(3): 1064-1065
25. Indelman V, Williams S, Kaess M, et al. Factor graph-based incremental smoothing in inertial navigation systems. *Proceedings of the 15th International Conference on Information Fusion*, 2012:2154-2161

26. Isaard M, Blake A. Contour tracking by stochastic propagation of conditional density. European Conference on Computer Vision, 1996, 343-356
27. Jazwinski A. Stochastic processes and filtering theory. New York, Academic Press, 1970
28. Kerr T. Decentralized filtering and redundancy management for multisensor navigation. IEEE Trans. on Aerospace and Electronic Systems, 1987, AES-23(1): 83-119
29. Kitagawa G. Monte Carlo filter and smother for non-Gaussian nonlinear state space models. Journal of Computational and Graphical Statistics, 1996, 5(1): 1-25
30. Koch K. Least Squares Adjustment and Collocation. Bulletin Geodesique, 1977, 51(2)
31. Lange S, Sünderhauf N, Protzel P. Incremental smoothing vs. filtering for sensor fusion on an indoor UAV. Proceedings of the IEEE International Conference on Robotics and Automation, 2013: 1773-1778
32. Li J, Hao S, Huang G. Modified strong tracking filter based on UD decomposition. Systems Engineering and Electronics, 2009, 31(8): 1953-1957
33. Li P, Xu X, Zhang X. Application of intelligent Kalman filter to underwater terrain integrated navigation system. Journal of Chinese Inertial Technology, 2011,19(5): 579-589
34. Li W, Li W, Cui X, et al. A tightly coupled RTK/INS algorithm with ambiguity resolution in the position domain for ground vehicles in harsh urban environments. Sensors, 2018, 18, 2160
35. Li X. Hybrid estimation techniques in control and dynamic systems: Advances In Theory And Applications. New York: Academic Press, 1996, 76: 213-287
36. Li X, Li J, Wang A, et al. A review of integrated navigation technology based on visual/inertial/UWB fusion. Science of Surveying and Mapping, 2023, 48(6): 49-58
37. Li Y, Yang Y, He H. Effects analysis of constraints on GNSS/INS Integrated Navigation. Geomatics And Information Science of Wuhan University, 2017, 42(9): 1249-1255
38. Liao J, Li X, Feng S. GVIL: Tightly-Coupled GNSS PPP/Visual/INS/LiDAR SLAM Based on Graph Optimization. Geomatics and Information Science of Wuhan University, 2023, 48(7):1204-1215
39. Liu F. Research on high-precision seamless positioning model and method based on multi-sensor fusion. Acta Geodaetica et Cartographica Sinica, 2021, 50(12): 1780
40. Liu Y, Yu A, Zhu J, et al. Unscented Kalman filtering in the additive noise case. Sci China Tech Sci, 2010, 53: 929-941
41. Loomis P, Carlson N, Berarducci M. Common Kalman filter: fault-tolerant navigation for next generation aircraft. Proc. Of the Inst. Of navigation Conf, Santa Barbara, CA, 1988, 38-45
42. Luan Z, Yu C, Gu B, Zhao X. A time-varying IMM fusion target tracking method. Radar and Navigation, 2021, 47(9): 111-116
43. MacCormick J, Blake A. A probabilistic exclusion principle for tracking multiple objects. International Conference on Computer Vision, 1999, 572-578
44. Magill D. Optimal Adaptive estimation of sampled stochastic processes. IEEE Trans. AC, 1965,10(10): 434-439
45. McClelland J, Rumelhart D, PDP research group. Parallel distributed processing. Cambridge: MIT Press, 1987
46. Mehra R K. Approaches to adaptive filtering. IEEE Trans. On Automatic Control, 1972, AC-17(5): 693-698
47. Moral P. Measure valued process and interacting particle systems: Application to non-linear filtering problems. Annals of Applied Probability, 1998, 8(2): 438-495
48. Pearson J. Dynamic decomposition techniques in optimization methods for large-scale system. McGraw-Hill, 1971
49. Shi C, Teng W, Zhang Y, Yu Y, Chen L, Chen R, Li Q. Autonomous Multi-Floor Localization Based on Smartphone-Integrated Sensors and Pedestrian Indoor Network. Remote Sensing. 2023, 15(11): 2933
50. Speyer J L. Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem. IEEE Trans. On Automatic Control, 1979, AC-24(2): 266-269
51. Tian Y, Yan Y, Zhong Y, Li J, Meng Z. Data fusion method based on IMM-Kalman for an integrated navigation system. Journal of Harbin Engineering University, 2022, 43(7): 973-978
52. Vapnik V. An overview of statistical learning theory. IEEE transactions on neural networks, 1999, 10(5): 988-999

53. Wang E, Pang T, Qu P, et al. GPS receiver Autonomous Integrity Monitoring Algorithm Based on Improved Particle Filter. *Tele-communication Engineering*, 2014, 54(4): 437-441
54. Wang E, Qu P, Pang T, et al. Receiver autonomous integrity monitoring based on particle swarm optimization particle filter. *Journal of Beijing University of Aeronautics and Astronautics*, 2016, 42(12): 2572-2578
55. Wang J, Ling H, Jiang W, Cai B. Integrated Train Navigation system based on Full state fusion of Multi-constellation satellite positioning and inertial navigation. *Journal of the China Railway Society*, 2022, 44(11): 45-52
56. Wang X, Li X, Liao J, et al. Tightly coupled stereo visual-inertial-LiDAR SLAM based on graph optimization. *Acta Geodaetica et Cartographica Sinica*, 2022, 51(8): 1744-1756
57. Watson R, Gross J. Robust navigation in GNSS degraded environment using graph optimization//Proceedings of the 30th international technical meeting of the satellite division of the Institute of Navigation (ION GNSS+ 2017). 2017: 2906-2918
58. Wen W, Pfeifer T, Bai X, et al. It is time for Factor Graph Optimization for GNSS/INS Integration: Comparison between FGO and EKF. *arXiv preprint arXiv: 2004.10572*, 2020
59. Willsky A S, Bello M G, Castanon D A, Levy B C, Verghese G C. Combining and updating of local estimates and regional maps along sets of one-dimensional tracks. *IEEE Trans. On Automatic Control*, 1982, AC-27(4): 799-813
60. Yang H, Wang M, Wang Y, Wu Y. Multiple-mode self-calibration unscented Kalman filter method. *Journal of Aerospace Power*, 2024: 1-7
61. Yu H, Li Z, Wang J, Han H. Data fusion for GPS/INS tightly-coupled positioning system with equality and inequality constraints using an aggregate constraint unscented Kalman filter. *Journal of Spatial Science*, 2020, 65(3): 377-399
62. Yu J, Lu W, Zeng M, Zhao S. Low-cost agricultural machinery intelligent navigation method based on multi-sensor information fusion. *China Measurement & Test*, 2021, 47(12): 106-119
63. Yuan L, Zhang S, Wang J, et al. GNSS/INS integrated navigation aided by sky images in urban occlusion environment. *Science of Surveying and Mapping*, 2023, 48(9): 1-8
64. Yun L, Shu S, Gang H. A Weighted Measurement Fusion Particle Filter for Nonlinear Multisensory Systems Based on Gauss-Hermite Approximation. *Sensors*, 2017, 17(10): 2222
65. Zhang T, Wang G, Chen Q, Tang H, Wang L, Niu X. Influence Analysis of IMU Scale Factor Error in GNSS/MEMS IMU vehicle integrated navigation. *Journal of Geodesy and Geodynamics*, 2024, 44(2): 134-137
66. Zhang X, Lu X. Recursive estimation of the stochastic model based on the Kalman filter formulation. *GPS Solutions*, 2021, 25(1): 24
67. Zhang X, Zhang Y, Zhu F. Factor Graph Optimization for Urban Environment GNSS Positioning and Robust Performance Analysis. *Geomatics and Information Science of Wuhan University*, 2023, 48(7): 1050-1057
68. Zhou Zhihua. *Machine Learning*. Tsinghua University Press, 2016
69. Zhu H, Wang F, Zhang W, et al. Real-Time Precise Positioning Method for Vehicle-Borne GNSS/MEMS IMU Integration in Urban Environment. *Geomatics and Information Science of Wuhan University*, 2023, 48(7): 1232-1240

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