



## ORIGINAL ARTICLE

# Fault Detection Algorithm for Gaussian Mixture Noises: An Application in LiDAR/IMU Integrated Localization Systems

Penggao Yan | Zhengdao Li | Feng Huang | Weisong Wen | Li-Ta Hsu\*

Department of Aeronautical and Aviation Engineering, Faculty of Engineering, Hong Kong Polytechnic University, Hong Kong

**Correspondence**

\*Li-Ta Hsu, Department of Aeronautical and Aviation Engineering, Faculty of Engineering, Hong Kong Polytechnic University, Hong Kong  
Email: lt.hsu@polyu.edu.hk

**Abstract**

Fault detection is crucial to ensure the reliability of localization systems. However, conventional fault detection methods usually assume that noises in the system are Gaussian distributed, limiting their effectiveness in real-world applications. This study proposes a fault detection algorithm for the extended Kalman filter (EKF) based localization system by modeling non-Gaussian noises as a Gaussian mixture model (GMM). The relationship between GMM-distributed noises and measurement residual is rigorously established through error propagation, which is utilized for constructing the test statistic for a Chi-squared test. The proposed method is applied to an EKF-based 2D light detection and ranging (LiDAR) and inertial measurement units (IMU) integrated localization system. The experimental results in a simulated urban environment show that the proposed method exhibits a 30 % improvement in the detection rate and a 17–23 % reduction in detection delay, compared with the conventional method with Gaussian noise modeling.

**Keywords**

fault detection, non-Gaussian noise, Gaussian mixture model, EKF, Chi-squared test, 2D LiDAR/IMU-based localization

## 1 | INTRODUCTION

*Gaussian noise modeling is not enough in developing fault detection methods:* Fault detection is essential for localization and navigation systems in some safety-critical applications (Joerger & Pervan, 2016; Osechas et al., 2012; Pervan et al., 1998; R. Wang et al., 2016), which is the technology to check the occurrence of faults in the system as well as to determine the time of fault occurs (Z. Gao et al., 2015). Fault detection methods can be mainly classified into model-based, knowledge-based, and signal-based methods (Z. Gao et al., 2015). Among these methods, model-based methods, especially statistical analysis of residuals, appear to be the most popular approach for detecting faults in localization and navigation systems (Angus, 2006; G. Gao et al., 2020; Puchalski & Giernacki, 2022; Yang et al., 2013), since system models are usually well-established and known to designers in these applications (Z. Gao et al., 2015). However, conventional model-based methods often assume that the noises in the system are Gaussian-distributed (Hsu et al., 2017; Joerger et al., 2014; Osechas et al., 2012; Pervan et al., 1998; Walter & Enge, 1995). For example, Walter and Enge (1995) constructed a test statistic by using the sum of the squares of the range residual errors (SSE), which is utilized to do a Chi-squared test to detect potential faults in pseudorange measurements. The measurement noise is assumed to have a multivariate Gaussian distribution in normal conditions where satellites have no malfunctions. R. Wang et al. (2016) designed a fault detection method for the navigation system based on the Chi-squared test and the sequential probability ratio test. In their work, all system process noise and measurement noise are modeled as zero-mean Gaussian distributions. Unfortunately, noises in the real world usually have non-Gaussian properties. Examples can be found in global navigation satellite system (GNSS) (J. Rife & Pervan, 2012), micro-electromechanical systems (MEMS) inertial sensors (Hou & El-Sheimy, 2003; Lethander & Taylor, 2023), and light detection and ranging (LiDAR) sensors (Xu et al., 2018).

These unrealistic Gaussian assumptions can result in increased false alarm rates or degraded fault detection rates in real-world applications, limiting the reliability and effectiveness of preventing systems from faults.

*Gaussian mixture model is promising but remains underexplored in the fault detection research:* The modeling of non-Gaussian noises receives increasing attention, and its application in the localization and navigation field, including multi-sensor fusion, robust localization, and integrity monitoring, has been extensively explored (Davis & Blair, 2015; Langel et al., 2020; J. H. Rife, 2018; Wen et al., 2021). One of the most popular approaches is to model noises as the Gaussian mixture model (GMM), which represents a probability distribution as a weighted combination of multiple Gaussian distributions. Examples include Ali-Loytty and Sirola (2007), which proposed the Gaussian sum filter (GSF) method for hybrid positioning with non-Gaussian noises by approximating the prior density of the state as a Gaussian mixture, Pfeifer and Protzel (2019), which proposed a robust sensor fusion algorithm by adaptively tuning the GMM parameters of the noise distribution, and Blanch et al. (2008), Z. Gao et al. (2022), and Yun et al. (2008), which developed the overbounding algorithms and derived the protection levels for integrity monitoring by modeling measurement noises as a GMM. Nevertheless, very limited research focuses on the fault detection problem considering non-Gaussian noises. One example can be found in Yun et al. (2008), which developed a fault detection method using GSF. In particular, the measurement noise is modeled as a GMM, and several parallel Kalman filters are developed to deal with each Gaussian component. The fault detection process is realized by comparing the one-side tail probability of the residual from the GSF with a predefined threshold. J. Wang et al. (2022) developed a similar algorithm. The difference is that their approach involves summing up the residual of each filter according to the mixture weight and subsequently taking the summation for a Chi-squared test to identify potential faults. Numerical experiments show that these approaches exhibit improved detection performance compared to conventional Gaussian methods. However, the improvement can be attributed to at least two factors: 1) the difference between the GSF-based detection method and the conventional detection method based on the Chi-squared test; 2) the difference between the GMM-based noise modeling and the Gaussian-based noise modeling. It is challenging to differentiate whether these improvements stem from the differences in the detection methods themselves or from the differences in the noise modeling.

*Our contributions in this paper:* Recently, Hashemi and Ruths (2019) proposed a fault detection method for the linear time-invariant control system with non-Gaussian noise. In their work, the residual of the observer in the linear time-invariant (LTI) system is modeled as a GMM and used to construct the test statistics for a Chi-squared test. This architecture is consistent with the conventional Gaussian method based on the Chi-squared test, providing valuable insights on fairly comparing the effects of non-Gaussian noise modeling on fault detection problems. Inspired by their work, this study aims to extend the idea to the fault detection problems in localization systems under non-Gaussian noises. Specifically, we establish the relationship between the GMM-distributed noises and the residual in an extended Kalman filter (EKF) based localization system. Then we transform the residual to a variable whose distribution approaches a standard multivariate normal (MVN) distribution. The Mahalanobis distance from the transformed variable to a standard MVN distribution is taken as the test statistic for a Chi-squared test to detect faults in the measurements. The proposed method is then applied to a localization system constructed by integrating LiDAR measurements and inertial measurement units (IMU) measurements via the EKF. For a fair comparison of different detection algorithms, a simulated urban environment is constructed based on the 3D simulator, CARLA (Dosovitskiy et al., 2017), making it possible to simulate GMM-distributed noises and ensuring the reproducibility of experiments. Finally, we compare the proposed method with the conventional Gaussian method in the simulated environment, and their performance is evaluated regarding two types of measurement failures. Our contributions in this study are two folds:

1. A fault detection algorithm designed for GMM noises. This study presents a fault detection algorithm for localization systems with Gaussian mixture noises, including a comprehensive analysis of the relationship between noises and residuals, a transformation process for the residual based on the law of total covariance, and a Chi-squared test. We prove that the measurement residual is the linear combination of the measurement and process noises and also exhibits a GMM distribution. In addition, the proposed method shares the same methodology as the conventional Gaussian method. The substantial difference in fault detection performance between these two methods is therefore attributed to the difference in noise modeling, making it possible to fairly evaluate the effects of different noise modeling on fault detection tasks in localization systems.
2. A simulated platform for a fair comparison of fault detection algorithms. This study establishes a simulated urban environment based on the 3D simulator, CARLA, which provides a fault-free environment for LiDAR-based localization systems, guaranteeing a fair comparison over different fault detection algorithms. In addition, this platform enables fault injection at specified time periods, providing a handy tool to evaluate the detection performance regarding different types of failure in different scenarios.

The remains of this paper are organized as follows. Section 2 presents the fault detection method for the localization system with GMM noises. We first establish the relationship between residuals of EKF and the multivariate GMM-distributed noises

(Section 2.1), providing the theoretical basis for developing the transformation method based on GMM assumptions (Section 2.2). The fault detection algorithm based on Chi-squared tests is then developed in Section 2.3. In Section 3, we apply the proposed fault detection method to an EKF-based LiDAR/IMU integrated localization system, which mainly includes the construction of the sensor platform, IMU motion model (3.1), and the 2D LiDAR measurement model (Section 3.2). In Section 4, a simulated environment is constructed for fairly comparing the performance of fault detection algorithms, and the fault detection performance of the proposed method is examined in this simulated environment regarding two types of measurement failures. Finally, Section 5 presents a summary.

## 2 | FAULT DETECTION WITH GMM NOISE MODELING

In this section, we present the fault detection algorithm for EKF-based localization systems with noises characterized by GMMs. We first establish the relationship between noises in the system and the measurement residual in EKF through error propagation. We prove that the measurement residual is the linear combination of the measurement and process noises through error propagation, and its distribution is also a GMM. A transformation method is then constructed to transform the residual to a variable whose distribution approaches a standard multivariate normal (MVN) distribution. Finally, we calculate the Mahalanobis distance from the transformed variable to a standard MVN distribution, which is taken as the test statistic for a Chi-squared test to detect faults in the measurements.

### 2.1 | Residual analysis in EKF

#### 2.1.1 | Relationship between the residual and noises

Statistical analysis of residuals is vital in model-based fault detection methods for localization and navigation systems (Angus, 2006; Joerger & Pervan, 2016; Puchalski & Giernacki, 2022; J. H. Rife, 2013). A general measurement model in a localization system can be written as

$$\mathbf{y}_k = h(\mathbf{x}_k) + \boldsymbol{\eta}_k, \quad (1)$$

where  $\mathbf{x}_k$  is the system state at time  $k$ ,  $\mathbf{y}_k$  and  $\boldsymbol{\eta}_k$  are the measurement and noise vector at time  $k$ , respectively, and  $h(\cdot)$  is the measurement function. A general state propagation model can be written as

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{v}_{k-1}), \quad (2)$$

where  $f(\cdot)$  is the state propagation function,  $\mathbf{u}_{k-1}$  is the external input at time  $k-1$ , and  $\mathbf{v}_{k-1}$  is the process noise at time  $k-1$ . In an EKF-based localization system, the propagation equations are given by

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, 0) \quad (3a)$$

$$\mathbf{P}_k^- = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{G}_{k-1} \mathbf{Q}_{k-1} \mathbf{G}_{k-1}^T, \quad (3b)$$

where  $\hat{\mathbf{x}}_{k-1}^+$  is the estimated state at time  $k-1$ ,  $\hat{\mathbf{x}}_k^-$  is the predicted state at time  $k$ ,  $\mathbf{P}_{k-1}$  is the covariance matrix of the estimated state by the EKF at time  $k-1$ ,  $\mathbf{P}_k^-$  is the predicted covariance matrix of the estimated state at time  $k$ ,  $\mathbf{F}_{k-1}$  is the state transition matrix,  $\mathbf{G}_{k-1}$  is the noise Jacobian matrix with respect to  $\mathbf{v}_{k-1}$ , and  $\mathbf{Q}_{k-1}$  is the covariance matrix of  $\mathbf{v}_{k-1}$ . If the EKF receives measurements  $\mathbf{y}_k$  at time  $k$ , the Kalman gain  $\mathbf{K}_k$  is obtained by the following equation (Daum, 2005):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \quad (4)$$

where  $\mathbf{R}_k$  is the covariance matrix of  $\boldsymbol{\eta}_k$ . The estimated state at time  $k$  is obtained by

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - h(\hat{\mathbf{x}}_k^-)), \quad (5)$$

where  $\mathbf{H}_k$  is the Jacobian matrix of  $h(\mathbf{x}_k)$ . The covariance of the estimated state is given by

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \quad (6)$$

The residual corresponding to the measurements at time  $k$  is

$$\mathbf{r}_k = \mathbf{y}_k - h(\hat{\mathbf{x}}_k^-). \quad (7)$$

By substituting Equation (1) into Equation (7) and taking the first-order Taylor expansion at  $\hat{\mathbf{x}}_k^-$ , we can obtain

$$\mathbf{r}_k = \mathbf{H}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \boldsymbol{\eta}_k, \quad (8)$$

where  $\mathbf{H}_k$  is the Jacobian matrix of  $h(\cdot)$  defined with respect to  $\mathbf{x}_k$ . Substitute Equations (2) and (3) into Equation (8) and apply the first-order Taylor expansion at the point  $(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}, 0)$ ,

$$\mathbf{r}_k = \mathbf{H}_k (\mathbf{F}_{k-1} (\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^+) + \mathbf{G}_{k-1} \mathbf{v}_{k-1}) + \boldsymbol{\eta}_k, \quad (9)$$

where  $\mathbf{F}_{k-1}$  is the Jacobian matrix of  $f(\cdot)$  defined with respect to  $\mathbf{x}_{k-1}$ , and  $\mathbf{G}_{k-1}$  is the Jacobian matrix of  $f(\cdot)$  defined with respect to  $\mathbf{v}_{k-1}$ . In Equation (9),  $\hat{\mathbf{x}}_{k-1}^+$  can be obtained by either state propagation or measurement update, both of which are discussed in the following.

(1)  $\hat{\mathbf{x}}_{k-1}^+$  is obtained by state propagation

In this condition, the EKF does not receive external measurements (such as LiDAR measurements). The estimated state is given by the predicted state, i.e.,  $\hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_{k-1}^-$ . Therefore, we have

$$\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^+ = \mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^-. \quad (10)$$

Substitute Equations (2) and (3) into Equation (10) and take the first-order Taylor expansion at  $(\hat{\mathbf{x}}_{k-2}^+, \mathbf{u}_{k-2}, 0)$ ,

$$\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^+ = \mathbf{F}_{k-2} (\mathbf{x}_{k-2} - \hat{\mathbf{x}}_{k-2}^+) + \mathbf{G}_{k-2} \mathbf{v}_{k-2}. \quad (11)$$

Repeat the operation in Equations (9)–(11), we have

$$\mathbf{r}_k = \mathbf{H}_k \left( \prod_{i=1}^m \mathbf{F}_{k-i} (\mathbf{x}_{k-m} - \hat{\mathbf{x}}_{k-m}^+) + \sum_{i=2}^m \prod_{j=1}^{i-1} \mathbf{F}_{k-j} \mathbf{G}_{k-i} \mathbf{v}_{k-i} + \mathbf{G}_{k-1} \mathbf{v}_{k-1} \right) + \boldsymbol{\eta}_k, \quad (12)$$

where  $m$  is the discrete time interval between the last measurement and the current measurement. To simplify the expression, we set the higher-order terms inside the brackets (i.e., terms with the number of matrix multiplications larger than 2) to zero and obtain

$$\mathbf{r}_k = \mathbf{H}_k \mathbf{F}_{k-1} \mathbf{G}_{k-2} \mathbf{v}_{k-2} + \mathbf{H}_k \mathbf{G}_{k-1} \mathbf{v}_{k-1} + \boldsymbol{\eta}_k. \quad (13)$$

(2)  $\hat{\mathbf{x}}_{k-1}^+$  is obtained by measurement update

In this case, we have the following expression by substituting Equation (5) into Equation (9):

$$\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^+ = \mathbf{x}_{k-1} - (\hat{\mathbf{x}}_{k-1}^- + \mathbf{K}_{k-1} (\mathbf{y}_{k-1} - h(\hat{\mathbf{x}}_{k-1}^-))). \quad (14)$$

Substituting Equation (1) into Equation (14) and taking the first-order Taylor expansion at  $\hat{\mathbf{x}}_{k-1}^-$ , we have

$$\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^+ = (\mathbf{I} - \mathbf{K}_{k-1} \mathbf{H}_{k-1}) (\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^-) - \mathbf{K}_{k-1} \boldsymbol{\eta}_{k-1}. \quad (15)$$

Therefore,

$$\begin{aligned} \mathbf{r}_k &= \mathbf{H}_k \mathbf{F}_{k-1} (\mathbf{I} - \mathbf{K}_{k-1} \mathbf{H}_{k-1}) (\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^-) \\ &\quad - \mathbf{H}_k \mathbf{F}_{k-1} \mathbf{K}_{k-1} \boldsymbol{\eta}_{k-1} + \mathbf{H}_k \mathbf{G}_{k-1} \mathbf{v}_{k-1} + \boldsymbol{\eta}_k. \end{aligned} \quad (16)$$

If we initialize the Kalman filter at  $t = k - 1$ , we have  $E[\hat{\mathbf{x}}_{k-1}^-] = \mathbf{x}_{k-1}$ .  $\hat{\mathbf{x}}_{k-1}^-$  is the unbiased estimator of  $\mathbf{x}_{k-1}$  and its distribution can be represented by a Gaussian distribution  $\mathcal{N}(\mathbf{x}_{k-1}, \boldsymbol{\pi}_0)$ , where  $\boldsymbol{\pi}_0$  is the covariance matrix of  $\hat{\mathbf{x}}_{k-1}^-$ . Define

$$\mathbf{e}_{k-1} = \mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^-, \quad (17)$$

and then we have  $\mathbf{e}_{k-1} \sim \mathcal{N}(0, \boldsymbol{\pi}_0)$ . Equation (16) can be written by

$$\begin{aligned} \mathbf{r}_k &= \mathbf{H}_k \mathbf{F}_{k-1} (\mathbf{I} - \mathbf{K}_{k-1} \mathbf{H}_{k-1}) \mathbf{e}_{k-1} \\ &\quad - \mathbf{H}_k \mathbf{F}_{k-1} \mathbf{K}_{k-1} \boldsymbol{\eta}_{k-1} + \mathbf{H}_k \mathbf{G}_{k-1} \mathbf{v}_{k-1} + \boldsymbol{\eta}_k. \end{aligned} \quad (18)$$

As can be seen,  $\mathbf{r}_k$  is the linear combination of  $\boldsymbol{\eta}_k$ ,  $\boldsymbol{\eta}_{k-1}$ ,  $\mathbf{v}_{k-1}$ , and  $\mathbf{e}_{k-1}$ . Assuming that the covariance of the initial state, i.e.,  $\boldsymbol{\pi}_0$ , is considerably small, we have  $\mathbf{e}_{k-1} \approx 0$ . Then,  $\mathbf{H}_k \mathbf{F}_{k-1} (\mathbf{I} - \mathbf{K}_{k-1} \mathbf{H}_{k-1}) \mathbf{e}_{k-1}$  only has limited impacts on Equation (18). If we know the ground truth of  $\mathbf{x}_{k-1}$  (it is possible in a simulation environment, as illustrated in Section 4), we can initialize the Kalman filter at  $t = k - 1$  with  $\hat{\mathbf{x}}_{k-1}^- = \mathbf{x}_{k-1}$ . Then  $\mathbf{r}_k$  approaches to a linear combination of  $\boldsymbol{\eta}_k$ ,  $\boldsymbol{\eta}_{k-1}$ , and  $\mathbf{v}_{k-1}$ :

$$\mathbf{r}_k = \boldsymbol{\eta}_k - \mathbf{H}_k \mathbf{F}_{k-1} \mathbf{K}_{k-1} \boldsymbol{\eta}_{k-1} + \mathbf{H}_k \mathbf{G}_{k-1} \mathbf{v}_{k-1}. \quad (19)$$

This perfect initialization can be approximately regarded as the setting of  $E[\hat{\mathbf{x}}_{k-1}^-] = \mathbf{x}_{k-1}$  with a very small  $\boldsymbol{\pi}_0$ .

If we do not initialize the Kalman filter at  $t = k - 1$ , we can repeat Equations (7)-(19) until the first initialization of the EKF,

$$E[\hat{\mathbf{x}}_1^-] = \mathbf{x}_1. \quad (20)$$

We will find that  $\mathbf{r}_k$  approaches to a linear combination of  $\mathbf{v}_{k-1}, \dots, \mathbf{v}_1$  and  $\boldsymbol{\eta}_k, \boldsymbol{\eta}_{k-1}, \dots, \boldsymbol{\eta}_1$ , and their coefficients are related to  $\mathbf{H}_k, \mathbf{H}_{k-1}, \dots, \mathbf{H}_1, \mathbf{F}_{k-1}, \dots, \mathbf{F}_1, \mathbf{G}_{k-1}, \dots, \mathbf{G}_1$ , and  $\mathbf{K}_{k-1}, \dots, \mathbf{K}_1$ .

### 2.1.2 | Distribution of the residual

To focus on the development of the fault detection algorithm, we simply take the case that  $\hat{\mathbf{x}}_{k-1}^+$  is obtained by the state propagation for further analysis and set the higher-order terms ( $>2$ ) to zero. Then Equation (13) is the final expression of residual at time  $k$ . Let

$$\mathbf{V}_{k-1} = \mathbf{H}_k \mathbf{G}_{k-1}, \quad (21)$$

and

$$\mathbf{N}_{k-1} = \mathbf{H}_k \mathbf{F}_{k-1} \mathbf{G}_{k-2}. \quad (22)$$

Then Equation (13) can be written as

$$\mathbf{r}_k = \mathbf{V}_{k-1} \mathbf{v}_{k-1} + \mathbf{N}_{k-1} \mathbf{v}_{k-2} + \boldsymbol{\eta}_k. \quad (23)$$

In this study,  $\mathbf{v}_{k-1}$ ,  $\mathbf{v}_{k-2}$ , and  $\boldsymbol{\eta}_k$  are assumed to be non-Gaussian noises, and we use the multivariate GMM to model them. Assuming  $\mathbf{v}_{k-1}$  and  $\mathbf{v}_{k-2}$  are independent and identically distributed (i.i.d.), we can obtain the PDF of  $\boldsymbol{\eta}$  and  $\mathbf{v}$  by

$$f_{\boldsymbol{\eta}}(\mathbf{x}) = \sum_{j=1}^{m_1} p_j^{\boldsymbol{\eta}} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j^{\boldsymbol{\eta}}, \boldsymbol{\pi}_j^{\boldsymbol{\eta}}) \quad (24a)$$

$$f_{\mathbf{v}}(\mathbf{x}) = \sum_{j=1}^{m_2} p_j^{\mathbf{v}} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j^{\mathbf{v}}, \boldsymbol{\pi}_j^{\mathbf{v}}), \quad (24b)$$

where  $\sum_{j=1}^{m_1} p_j^{\boldsymbol{\eta}} = 1$ ,  $\sum_{j=1}^{m_2} p_j^{\mathbf{v}} = 1$ ;  $m_1$  and  $m_2$  are the number of Gaussian modes for  $\boldsymbol{\eta}$  and  $\mathbf{v}$ , respectively;  $p_j^{\boldsymbol{\eta}}$  and  $p_j^{\mathbf{v}}$  are the mixture weight (i.e., the prior probability of each Gaussian mode);  $\mathcal{N}(\cdot | \boldsymbol{\mu}_j^{\boldsymbol{\eta}}, \boldsymbol{\pi}_j^{\boldsymbol{\eta}})$  and  $\mathcal{N}(\cdot | \boldsymbol{\mu}_j^{\mathbf{v}}, \boldsymbol{\pi}_j^{\mathbf{v}})$  are probability density functions of each Gaussian mode;  $\boldsymbol{\mu}_j^{\boldsymbol{\eta}} \in \mathbb{R}^{n_1}$  and  $\boldsymbol{\mu}_j^{\mathbf{v}} \in \mathbb{R}^{n_2}$  are means of each Gaussian mode;  $\boldsymbol{\pi}_j^{\boldsymbol{\eta}} \in \mathbb{R}^{n_1 \times n_1}$  and  $\boldsymbol{\pi}_j^{\mathbf{v}} \in \mathbb{R}^{n_2 \times n_2}$  are covariance matrices of each Gaussian mode;  $n_1$  is the dimension of the measurement noise, and  $n_2$  is the dimension of process noise. Inspired by the work of Hashemi and Ruths (2019),  $\mathbf{r}_k$  can be proved to be multivariate GMM distributed by applying the convolution theorem on the characteristic function of each component of the residual (refer to Appendix B for details):

$$f_{\mathbf{r}_k}(\mathbf{x}) = \sum_{a=1}^{m_2} \sum_{b=1}^{m_2} \sum_{c=1}^{m_1} p_{abc} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{abc}, \boldsymbol{\pi}_{abc}), \quad (25)$$

where

$$p_{abc} = p_a^{\mathbf{v}} p_b^{\mathbf{v}} p_c^{\boldsymbol{\eta}} \quad (26a)$$

$$\boldsymbol{\mu}_{abc} = \mathbf{V}_{k-1} \boldsymbol{\mu}_a^{\mathbf{v}} + \mathbf{N}_{k-1} \boldsymbol{\mu}_b^{\mathbf{v}} + \boldsymbol{\mu}_c^{\boldsymbol{\eta}} \quad (26b)$$

$$\boldsymbol{\pi}_{abc} = \mathbf{V}_{k-1} \boldsymbol{\pi}_a^{\mathbf{v}} \mathbf{V}_{k-1}^T + \mathbf{N}_{k-1} \boldsymbol{\pi}_b^{\mathbf{v}} \mathbf{N}_{k-1}^T + \boldsymbol{\pi}_c^{\boldsymbol{\eta}}. \quad (26c)$$

## 2.2 | Transformation methods

### 2.2.1 | Transformation based on GMM assumptions

Section 2.1 illustrates that the residual is a multivariate GMM-distributed random variable. To align with the architecture of the Chi-squared detector, we transform the residual to a new random variable by subtracting its total mean and subsequently pre-multiplying its principal square root matrix of the total covariance (Hashemi & Ruths, 2019)

$$\mathbf{T}_g = \Sigma^{-1/2} (\mathbf{r}_k - \boldsymbol{\mu}), \quad (27)$$

where  $\boldsymbol{\mu}$  is the total mean and  $\Sigma^{-1/2}$  is the principal square root matrix of the total covariance ( $\Sigma$ ) of the residual, respectively. According to the law of total covariance (Weiss et al., 2006),  $\boldsymbol{\mu}$  and  $\Sigma$  are given by

$$\boldsymbol{\mu} = \sum_{a=1}^{m_2} \sum_{b=1}^{m_2} \sum_{c=1}^{m_1} p_{abc} \boldsymbol{\mu}_{abc} \quad (28a)$$

$$\Sigma = \sum_{a=1}^{m_2} \sum_{b=1}^{m_2} \sum_{c=1}^{m_1} p_{abc} \boldsymbol{\pi}_{abc} + (\boldsymbol{\mu} - \boldsymbol{\mu}_{abc})(\boldsymbol{\mu} - \boldsymbol{\mu}_{abc})^T. \quad (28b)$$

In Appendix C, we demonstrate that the transformed variable has a distribution that approaches the standard MVN distribution when each Gaussian component in the GMM-distributed residual has a small difference in the covariance. Therefore, we limit the proposed transformation method to specific applications where the measurement error can be modeled by GMM and the covariance of each component is similar. By setting the number of the Gaussian components of the GMM to one, the above transformation formulation can be easily extended to situations in which the residual is formulated as Gaussian distribution (Da, 1994; Liu et al., 2017).

### 2.2.2 | Transformation based on Gaussian assumptions

To evaluate the impact of GMM noise modeling on the fault detection performance, we also establish a baseline transformation method that takes the Gaussian assumption about the noises. Specifically, the measurement noise  $\boldsymbol{\eta}_k^0$  and the process noise  $\mathbf{v}_k^0$  at time  $k$  are modeled as zero-mean Gaussian noises as follows:

$$\boldsymbol{\eta}_k^0 \sim \mathcal{N}(0, \boldsymbol{\pi}_\eta^0) \quad (29a)$$

$$\mathbf{v}_k^0 \sim \mathcal{N}(0, \boldsymbol{\pi}_v^0), \quad (29b)$$

where  $\boldsymbol{\pi}_\eta^0 \in \mathbb{R}^{n_1 \times n_1}$  and  $\boldsymbol{\pi}_v^0 \in \mathbb{R}^{n_2 \times n_2}$  are the covariance matrices and can be obtained by calculating the variance of samples generated from the noise distribution in Equation (24) through the Monte-Carlo simulation. Following the derivation in Section 2.1, the measurement residual at time  $k$  is given by

$$\mathbf{r}_k^0 = \mathbf{V}_{k-1} \mathbf{v}_{k-1}^0 + \mathbf{N}_{k-1} \mathbf{v}_{k-2}^0 + \boldsymbol{\eta}_k^0. \quad (30)$$

Let  $\Sigma_0 = \mathbf{V}_{k-1} \boldsymbol{\pi}_v^0 \mathbf{V}_{k-1}^T + \mathbf{N}_{k-1} \boldsymbol{\pi}_v^0 \mathbf{N}_{k-1}^T + \boldsymbol{\pi}_\eta^0$ , and then the distribution of the measurement residual is given by

$$f_{\mathbf{r}_k^0}(\mathbf{x}) = \mathcal{N}(\mathbf{x} | 0, \Sigma_0^{-1/2}), \quad (31)$$

which is a zero-mean Gaussian distribution. Similar to Equation (27),  $\mathbf{r}_k^0$  can be transformed to a standard MVN distributed random variable as

$$\mathbf{T}_g^0 = \Sigma_0^{-1/2} \mathbf{r}_k^0. \quad (32)$$

Figure 1 plots the transformation process of the residual based on two different assumptions. Obviously, the substantial difference between the two transformation methods lies in the way of modeling noises. This feature is quite important, which enables us to develop a fault detection method with GMM noise modeling that shares the same methodology as the conventional Gaussian method, providing a way to fairly evaluate the effects of different noise modeling on fault detection tasks in localization systems.

## 2.3 | Fault detection based on Chi-squared test

The two transformations in Section 2.2, though different, yield the standardized measurement residual that is assumed to have a standard MVN distribution. The  $n_1 \times 1$  vector  $\mathbf{T}_g$  can be written as

$$\mathbf{T}_g = [T_{g,1}, T_{g,2}, \dots, T_{g,n_1}], \quad (33)$$

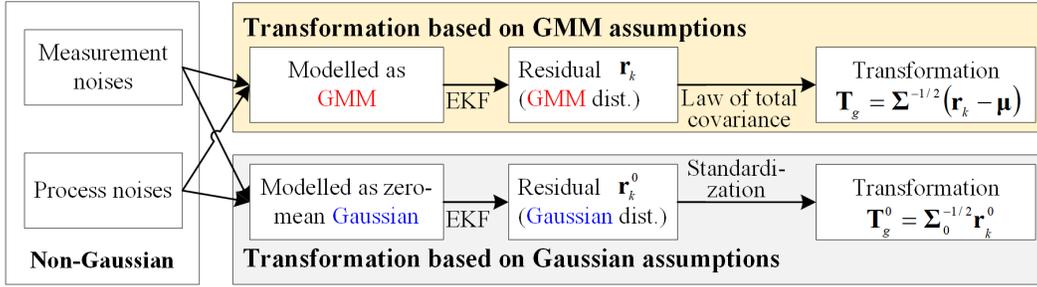


FIGURE 1 The demonstration of transformation methods based on GMM assumptions and Gaussian assumptions, respectively.

where  $T_{g,1}, T_{g,2}, \dots, T_{g,n_1}$  are assumed to be mutually independent standard normal random variables. We can construct the following test statistic  $t_g$ ,

$$t_g = \mathbf{T}_g^T \mathbf{T}_g = T_{g,1}^2 + T_{g,2}^2 + \dots + T_{g,n_1}^2. \quad (34)$$

Since  $T_{g,1}, T_{g,2}, \dots, T_{g,n_1}$  are mutually independent,  $t_g$  follows the Chi-squared distribution with  $n_1$  degrees of freedom (DOF). From another perspective, Equation (34) represents the square of the Mahalanobis distance of  $\mathbf{T}_g$  from the standard MVN distribution, which is known to have the same Chi-squared distribution. Similarly, we can construct the test statistic  $t_g^0$  in the context of the Gaussian assumption as

$$t_g^0 = \mathbf{T}_g^{0T} \mathbf{T}_g^0. \quad (35)$$

Before performing the hypothesis test with the constructed test statistic, we give a brief introduction to the Chi-squared distribution. Its cumulative distribution function (CDF) is given by

$$F(x; k) = \frac{\gamma\left(\frac{k}{2}, \frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}, \quad (36)$$

where  $\gamma(\cdot)$  is the lower incomplete gamma function,  $\Gamma(\cdot)$  is the gamma function, and  $k$  is the degree of freedom. For a given value  $TD \in \mathbb{R}$ , the probability of the Chi-squared statistic less than  $TD$  is given by  $P(x < TD) = F(TD; k)$ , and the associated p-value is given by  $1 - F(TD; k)$ . The value of  $TD$  and the corresponding p-value are interrelated, and their association can be determined through tables of the Chi-squared distribution (Fisher & Yates, 1953). The Chi-squared test for fault detection associated with the test statistic  $t_g$  at a given significance level  $\alpha$  is

$$H_1 : t_g > TD_\alpha \quad (37a)$$

$$H_0 : t_g \leq TD_\alpha, \quad (37b)$$

where  $TD_\alpha$  is determined by

$$P(t_g > TD_\alpha | H_0) = \alpha. \quad (38)$$

A similar hypothesis test can be conducted with the test statistic  $t_g^0$ , which is defined by the similar formulations in Equations (37)–(38) and is omitted for presenting for the sake of clarity. For convenience, we name the fault detection method based on  $t_g$  as the **total Gaussian-GMM** method, and we name the method based on  $t_g^0$  as the Gaussian method.

In Equation (38), the significance level  $\alpha$  can also be interpreted as the desired false alarm rate in the fault detection context when the assumption that  $\mathbf{T}_g$  has a standard MVN distribution is valid. However, the assumption is unrealistic in real-world applications. For example, the GMM cannot perfectly model the non-Gaussian noises in the LiDAR measurement. Another issue is that the elements in the residual vector could be correlated, which violates the assumption and eventually results in the degradation of the fault detection performance. Therefore, criteria are required to examine the real performance of the fault detection results. In this study, two criteria, including the false alarm rate (FAR) and the fault detection rate (FDR) in a period, are formulated as

$$FAR = \frac{n_{FP}}{n_{FP} + n_{TN}} \quad (39a)$$

$$FDR = \frac{n_{TP}}{n_{TP} + n_{FN}}, \quad (39b)$$

where  $n_{TP}$  is the number of faulty events that are classified as faulty (true positive),  $n_{TN}$  is the number of fault-free events that are classified as normal (true negative),  $n_{FP}$  is the number of fault-free events that are classified as faulty (false positive), and  $n_{FN}$  is the number of faulty events that are classified as normal (false negative).

### 3 | THE APPLICATION IN IMU/LIDAR INTEGRATED LOCALIZATION SYSTEM

This section presents the application of the proposed fault detection in an IMU/LiDAR integrated system. The architecture of the fault detection process in the localization system is illustrated in Figure 2. A LiDAR/IMU integrated localization system is constructed by utilizing EKFs (Daum, 2005), where the state propagation equation is constructed by the kinematic model of the IMU motion, and the measurement function is constructed by matching the extracted line features from seven 2D LiDAR points to the plane in the prior map. Notably, 2D LiDARs possess fewer moving parts and components, making them more reliable and easier to maintain in comparison to 3D LiDARs. This characteristic not only helps to prevent hardware faults but also simplifies the development and evaluation of fault detection methods. In addition, 2D LiDARs are low-cost but can only provide limited ranging measurements. As a result, it is of great importance to reliably detect faulty measurements from 2D LiDAR. In this EKF-based localization system, the process and measurement noises are modeled as the multivariate GMM. As illustrated in Section 2, the nominal residual (defined as the residual of a fault-free system) is proved to be multivariate GMM. Therefore, we leverage the law of total covariance to transform the residual into a variable whose distribution approaches a standard MVN distribution. Finally, we calculate the Mahalanobis distance from the transformed variable to a standard MVN distribution, which is taken as the test statistic for a Chi-squared test to detect faults in the measurements.

The sensor platform of the localization system is shown in Figure 2(b), where an IMU is placed at the chassis and seven 2D LiDARs are distributively placed around the vehicle. The center location and the elevation angle of each LiDAR in the IMU frame ( $\{I\}$ ) are given in the embedded table. Instead of defining seven LiDAR frames for each 2D LiDAR, we only define a single LiDAR frame ( $\{L\}$ ) that is fixed at the center of the fourth LiDAR for clarity. As the seven 2D LiDARs have a similar measurement model, such a definition can make the main idea clearer without loss of generality. Following this definition, we denote  ${}^L\mathbf{p}_I$  as the translation from  $\{I\}$  to  $\{L\}$  and  ${}^L\mathbf{q}_I$  as the rotation (the rotation matrix is  ${}^L\mathbf{R}$ ) from  $\{I\}$  to  $\{L\}$ .  ${}^I\mathbf{p}_L$  and  ${}^I\mathbf{R}$  are the extrinsic calibration parameters calibrated in the setup stage of the system. Note that the extrinsic calibration parameters for each 2D LiDAR are different and are affected by the center location and the elevation angle of the 2D LiDAR plane.

The state vector  $\mathbf{x}$  of the system is defined as follows:

$$\mathbf{x} = [{}^G\mathbf{p}_I^T, {}^G\mathbf{v}_I^T, {}^I\mathbf{q}_G^T, \mathbf{b}_a^T, \mathbf{b}_g^T]^T, \quad (40)$$

where  ${}^G\mathbf{p}_I$  and  ${}^G\mathbf{v}_I$  are the position and velocity vectors of IMU frame ( $\{I\}$ ) in the world frame ( $\{G\}$ ),  ${}^I\mathbf{q}_G = [q_w, q_x, q_y, q_z]^T$  denotes the rotation from  $\{G\}$  to  $\{I\}$  in terms of quaternion, and  $\mathbf{b}_a^T$  and  $\mathbf{b}_g^T$  are biases of the accelerometer and gyroscope measurements. In addition, the rotation matrix associated with  ${}^I\mathbf{q}_G$  is denoted by  ${}^I\mathbf{R}$ . The world frame is fixed at the center of the pre-built point cloud map in the East-North-Up (ENU) coordinate system. For clarity, all symbols used in this chapter are listed in Appendix A. The following sections briefly introduce the motion model and the measurement model of the IMU/LiDAR integrated system, and, more importantly, determine the exact form of  $\mathbf{F}$ ,  $\mathbf{G}$ , and  $\mathbf{H}$ , which are used for constructing the test statistic for the fault detection process.

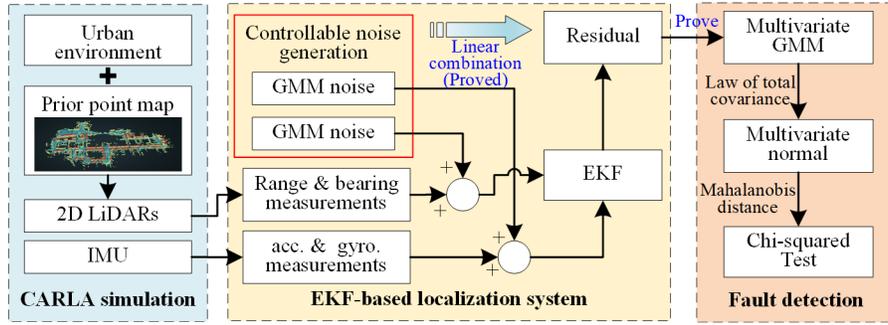
#### 3.1 | Motion model

The kinematic model of the IMU motion in  $\{G\}$  is adopted to propagate the vehicle state (Lefferts et al., 1982). The discrete-time state propagation equation is given by

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{v}_{k-1}) = \begin{bmatrix} {}^G\mathbf{p}_{I,k-1} + {}^G\mathbf{v}_{I,k-1}\Delta t + \frac{1}{2} \left( {}^G\hat{\mathbf{R}}_{k-1} (\mathbf{a}_{m,k-1} - \mathbf{b}_{a,k-1} - \mathbf{n}_a) + {}^G\mathbf{g} \right) \Delta t^2 \\ {}^G\mathbf{v}_{I,k-1} + \left( {}^G\hat{\mathbf{R}}_{k-1} (\mathbf{a}_{m,k-1} - \mathbf{b}_{a,k-1} - \mathbf{n}_a) + {}^G\mathbf{g} \right) \Delta t \\ \exp\left(\frac{\Delta t}{2}\boldsymbol{\Omega}[\mathbf{w}_{m,k-1} - \mathbf{b}_{g,k-1} - \mathbf{n}_g]\right) {}^I\mathbf{q}_{G,k-1} \\ \mathbf{b}_{a,k-1} + \mathbf{n}_{wa}\Delta t \\ \mathbf{b}_{g,k-1} + \mathbf{n}_{wg}\Delta t \end{bmatrix}, \quad (41)$$

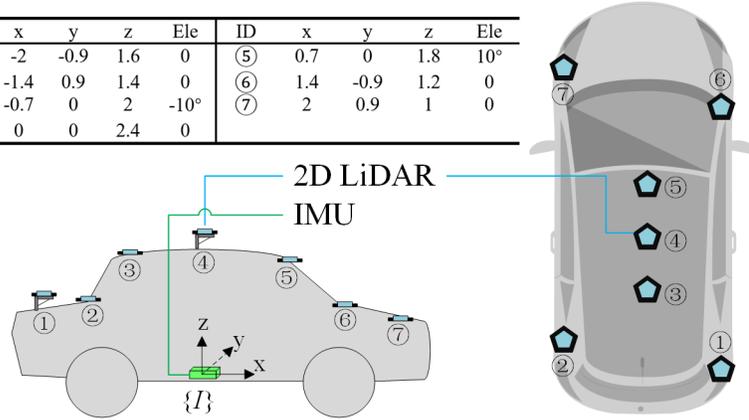
where  $\Delta t$  is a short time period;  ${}^G\mathbf{p}_{I,k-1}$ ,  ${}^G\mathbf{v}_{I,k-1}$ ,  ${}^I\mathbf{q}_{G,k-1}$ ,  $\mathbf{b}_{a,k-1}$ ,  $\mathbf{b}_{g,k-1}$  are system states at the discrete time  $k-1$ ;  $\mathbf{a}_{m,k-1}$  and  $\mathbf{w}_{m,k-1}$  are the acceleration and gyroscope measurements, respectively;  $\mathbf{n}_a$  and  $\mathbf{n}_g$  are the noises of acceleration and gyroscope measurements, respectively;  $\mathbf{n}_{wa}$  and  $\mathbf{n}_{wg}$  are zero-mean Gaussian white noise;  $\mathbf{v}_{k-1} = [\mathbf{n}_a^T, \mathbf{n}_g^T, \mathbf{n}_{wa}^T, \mathbf{n}_{wg}^T]^T$  is the process noise vector at time  $k-1$ ;  ${}^G\hat{\mathbf{R}}_{k-1}$  represents the estimation of the vehicle orientation at time  $k-1$ ; and  $\boldsymbol{\Omega}[\cdot]$  is a  $4 \times 4$  skew symmetric matrix:

$$\boldsymbol{\Omega}[\mathbf{w}] = \begin{bmatrix} -[\mathbf{w}]_{\times} & \mathbf{w} \\ -\mathbf{w}^T & 0 \end{bmatrix}, \quad (42)$$



(a)

ID	x	y	z	Ele	ID	x	y	z	Ele
①	-2	-0.9	1.6	0	⑤	0.7	0	1.8	10°
②	-1.4	0.9	1.4	0	⑥	1.4	-0.9	1.2	0
③	-0.7	0	2	-10°	⑦	2	0.9	1	0
④	0	0	2.4	0					



(b)

**FIGURE 2** (a) The architecture of fault detection in the LiDAR/IMU integrated localization system; (b) The sensor platform of the localization system. “Ele.” is the abbreviation of “Elevation angle”.

where  $[\cdot]_{\times}$  denotes the standard vector cross-product.

By linearizing (41) with respect to  $\mathbf{x}_{k-1}$  and  $\mathbf{v}_{k-1}$  respectively, we can obtain the state transition matrix  $\mathbf{F}_{k-1}$  and the noise Jacobian matrix  $\mathbf{G}_{k-1}$  as follows:

$$\mathbf{F}_{k-1} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 \Delta t & 0 & -\frac{1}{2} \hat{\mathbf{R}}_{k-1} \Delta t^2 & 0 \\ 0 & \mathbf{I}_3 & 0 & -\hat{\mathbf{G}}_{k-1} \Delta t & 0 \\ 0 & 0 & \frac{\partial g({}^I \hat{\mathbf{q}}_{G,k-1}, \mathbf{w}_{m,k-1})}{\partial {}^I \hat{\mathbf{q}}_{G,k-1}} & 0 & \frac{\partial g({}^I \hat{\mathbf{q}}_{G,k-1}, \mathbf{w}_{m,k-1})}{\partial \mathbf{w}_{GI,k-1}} \\ 0 & 0 & 0 & \mathbf{I}_3 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I}_3 \end{bmatrix} \quad (43a)$$

$$\mathbf{G}_{k-1} = \begin{bmatrix} -\frac{1}{2} \hat{\mathbf{R}}_{k-1} \Delta t^2 & 0 & 0 & 0 \\ -\hat{\mathbf{G}}_{k-1} \Delta t & 0 & 0 & 0 \\ 0 & \frac{\partial g({}^I \hat{\mathbf{q}}_{G,k-1}, \mathbf{w}_{m,k-1})}{\partial \mathbf{w}_{GI,k-1}} & 0 & 0 \\ 0 & 0 & \mathbf{I}_3 \Delta t & 0 \\ 0 & 0 & 0 & \mathbf{I}_3 \Delta t \end{bmatrix}, \quad (43b)$$

where  $\frac{\partial g({}^I \hat{\mathbf{q}}_{G,k-1}, \mathbf{w}_{m,k-1})}{\partial {}^I \hat{\mathbf{q}}_{G,k-1}}$  and  $\frac{\partial g({}^I \hat{\mathbf{q}}_{G,k-1}, \mathbf{w}_{m,k-1})}{\partial \mathbf{w}_{GI,k-1}}$  are derived in Appendix E.2.

### 3.2 | Measurement model

The laser scan plane of 2D LiDARs intersects with several planes. The intersection is called as the line segment, which is fitted by a set of 2D LiDAR points, as shown in Figure 3(a). We adopt the method from Pfister et al. (2003) to extract line segments from raw 2D LiDAR measurements and find their associated planes. Details can refer to Appendix D.2. The 2D LiDAR measurement model is constructed by finding the shortest vector in the laser scan plane from the origin of  $\{G\}$  to the plane (Hesch et al., 2010; Zhao & Farrell, 2013), as illustrated in Figure 3(b). Assuming that the laser scan plane is intersected with the plane  $\Pi_i$  at line  ${}^L\xi_i^\perp$  in  $\{L\}$ , we can find the shortest vector  ${}^L\mathbf{x}_i = \rho_i {}^L\xi_i$  in the LiDAR scan plane from the origin of  $\{L\}$  to  $\Pi_i$ , where  ${}^L\xi_i$  is the unit vector,  $\rho_i$  is the length of  ${}^L\mathbf{x}_i$ , and the point  $M$  is the intersection of  ${}^L\mathbf{x}_i$  and  ${}^L\xi_i^\perp$ . The laser beam  ${}^L\mathbf{x}_i$  in the scan laser plane can be represented by ranging and bearing parameters  $(\rho_i, \phi_i)$  in the polar coordinate system, as shown in Figure 3(c). Alternatively,  ${}^L\mathbf{x}_i$  can also be written as  ${}^L\mathbf{x}_i = [\rho_i \cos \phi_i \ \rho_i \sin \phi_i \ 0]^T$  in  $\{L\}$ . The shortest vector from the origin of  $\{G\}$  to the plane  $\Pi_i$  is denoted by  $d_i {}^G\boldsymbol{\pi}_i$  and is intersected with  $\Pi_i$  at point  $N$ , where  ${}^G\boldsymbol{\pi}_i$  is the normal of the plane in  $\{G\}$  and  $d_i$  is the length of  $d_i {}^G\boldsymbol{\pi}_i$ . The vector from  $M$  to  $N$  in  $\{G\}$  is denoted by  ${}^G\mathbf{t}_i$ . The measurement model can be written as

$$\phi_i = h_1^i(\mathbf{x}) + \tilde{\phi}_i = \arctan\left(\text{sgn}({}^L d_i) \frac{a_2^i}{a_1^i}\right) + \tilde{\phi}_i \quad (44a)$$

$$\rho_i = h_2^i(\mathbf{x}) + \tilde{\rho}_i = \frac{|{}^L d_i|}{\sqrt{a_1^{i2} + a_2^{i2}}} + \tilde{\rho}_i \quad (44b)$$

$${}^L d_i = d_i - {}^G\boldsymbol{\pi}_i^T ({}^G\mathbf{p}_I + {}^G\mathbf{R}^I \mathbf{p}_L) \quad (44c)$$

$${}^L \mathbf{a}_i = {}^L_G \mathbf{R}^G \boldsymbol{\pi}_i = [a_1^i, a_2^i, a_3^i]^T, \quad (44d)$$

where the  $i$ th measurement  $\mathbf{y}_i = [\phi_i, \rho_i]^T$  is the ranging and bearing parameters associated with the shortest vector in the laser plane intersected with  $\Pi_i$ ,  $\tilde{\phi}_i$  and  $\tilde{\rho}_i$  are measurement noises, and  $\text{sgn}(\cdot)$  is the sign function. Details can refer to Appendix D.1

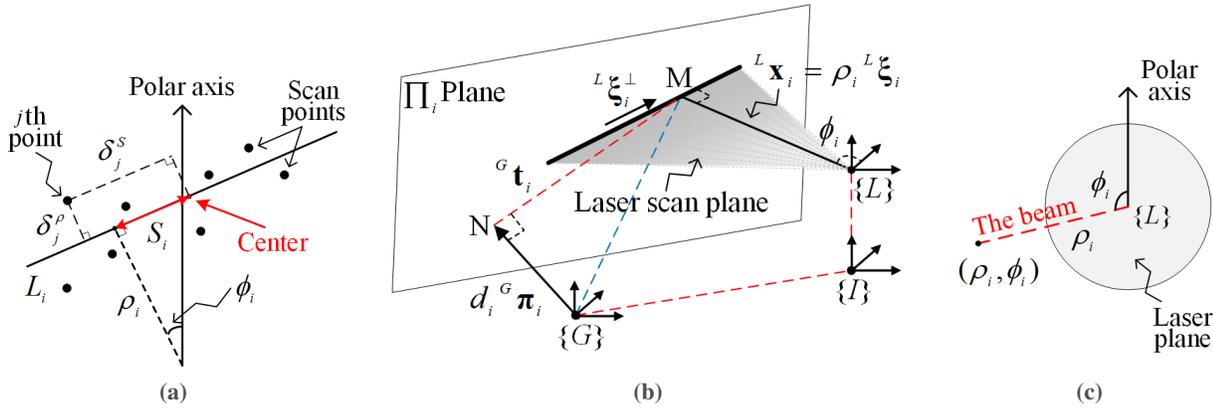


FIGURE 3 (a) A demonstration of extracting line segments from 2D LiDAR points; (b) The 2D LiDAR measurement model is constructed by finding the shortest vector in the laser scan plane from the origin of  $\{L\}$  to the plane  $\Pi_i$ ; (c) The laser plane with the polar coordinate system.

Let  $h^i(\mathbf{x}_k) = [h_1^i(\mathbf{x}_k) \ h_2^i(\mathbf{x}_k)]^T$  be the measurement function of  $i$ th measurement at time  $k$ ,  $\boldsymbol{\eta}_{i,k} = [\tilde{\phi}_{i,k} \ \tilde{\rho}_{i,k}]^T$  be the associated measurement noise,  $\hat{\mathbf{x}}_k^- = [{}^G\hat{\mathbf{p}}_I^T, {}^G\hat{\mathbf{v}}_I^T, I\hat{\mathbf{q}}_G^T, \hat{\mathbf{b}}_a^T, \hat{\mathbf{b}}_g^T]^T$  be the predicted state, and  $\mathbf{H}_k^i$  be the Jacobian matrix of  $h^i(\mathbf{x}_k)$  defined with respect to  $\mathbf{x}_k$  and evaluated at  $\hat{\mathbf{x}}_k^-$ :

$$\mathbf{H}_k^i = \frac{\partial h^i(\hat{\mathbf{x}}_k^-)}{\partial \mathbf{x}_k}. \quad (45)$$

As proved in Appendix E.3,  $\mathbf{H}_k^i$  is given by

$$\mathbf{H}_k^i = \begin{bmatrix} \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \frac{1}{\hat{\mu}_{i,k}^-} \hat{\boldsymbol{\lambda}}_{i,k}^{-T} {}^L \mathbf{R} \mathbf{J}_q^* \left( {}^I \hat{\mathbf{q}}_{G,k}^-, {}^G \boldsymbol{\pi}_{i,k} \right) & \mathbf{0}_{1 \times 6} \\ -\text{sgn} \left( {}^L \hat{d}_{i,k}^- \right) \frac{{}^G \boldsymbol{\pi}_{i,k}^T}{\sqrt{\hat{\mu}_{i,k}^-}} & \mathbf{0}_{1 \times 3} & -\text{sgn} \left( {}^L \hat{d}_{i,k}^- \right) \frac{{}^G \boldsymbol{\pi}_{i,k}^T}{\sqrt{\hat{\mu}_{i,k}^-}} \mathbf{J}_q \left( {}^I \hat{\mathbf{q}}_{G,k}^-, {}^I \mathbf{p}_L \right) + \left| {}^L \hat{d}_{i,k}^- \right| \hat{\boldsymbol{\kappa}}_{i,k}^T {}^L \mathbf{R} \mathbf{J}_q^* \left( {}^I \hat{\mathbf{q}}_{G,k}^-, {}^G \boldsymbol{\pi}_{i,k} \right) & \mathbf{0}_{1 \times 6} \end{bmatrix}, \quad (46)$$

where

$$\begin{aligned} \hat{\mu}_{i,k}^- &= a_1^{i-2} + a_2^{i-2}, \boldsymbol{\lambda}_{i,k}^{-T} = \text{sgn} \left( {}^L \hat{d}_{i,k}^- \right) \left[ -\hat{a}_2^{i-}, \hat{a}_1^{i-}, 0 \right], \\ \hat{\boldsymbol{\kappa}}_{i,k}^T &= -\hat{\mu}_{i,k}^{-\frac{3}{2}} \left[ \hat{a}_1^{i-}, \hat{a}_2^{i-}, 0 \right], {}^I \hat{\mathbf{a}}_{i,k}^- = {}^L \mathbf{R}_G^I \hat{\mathbf{R}}_k^{-G} \boldsymbol{\pi}_{i,k} = \left[ \hat{a}_1^{i-}, \hat{a}_2^{i-}, \hat{a}_3^{i-} \right]^T. \end{aligned} \quad (47)$$

For multiple measurements  $\mathbf{y}_k = \left[ \mathbf{y}_{1,k}^T, \mathbf{y}_{2,k}^T, \dots, \mathbf{y}_{n,k}^T \right]^T$  at time  $k$ , the measurement function is written as

$$\mathbf{y}_k = h(\mathbf{x}_k) + \boldsymbol{\eta}_k, \quad (48)$$

where

$$h(\mathbf{x}_k) = \left[ h^1(\mathbf{x}_k)^T, h^2(\mathbf{x}_k)^T, \dots, h^n(\mathbf{x}_k)^T \right]^T \quad (49a)$$

$$\boldsymbol{\eta}_k = \left[ \boldsymbol{\eta}_{1,k}^T, \boldsymbol{\eta}_{2,k}^T, \dots, \boldsymbol{\eta}_{n,k}^T \right]^T, \quad (49b)$$

and  $n$  is the total number of measurements at time  $k$ . Then the Jacobian matrix of  $h(\mathbf{x}_k)$  can be written as

$$\mathbf{H}_k^T = \left[ \mathbf{H}_k^{1T}, \mathbf{H}_k^{2T}, \dots, \mathbf{H}_k^{nT} \right]^T. \quad (50)$$

It is worth noting that the linearization process presented here is quite different from the work of Hesch et al. (2010) and Zhao and Farrell (2013). In this work, we linearize the measurement function at the nominal state in terms of position vector and quaternion, as shown in Equations (E27) and (E41) in Appendix E.3, which is more direct and intuitive than the approach that performs the linearization at the error state (Hesch et al., 2010; Zhao & Farrell, 2013). In addition, linearization at the nominal state is beneficial to establish a clear relationship between the measurement noise and the state, as well as the relationship between the measurement noise and the residual, both of which are shown in Section 2.1.

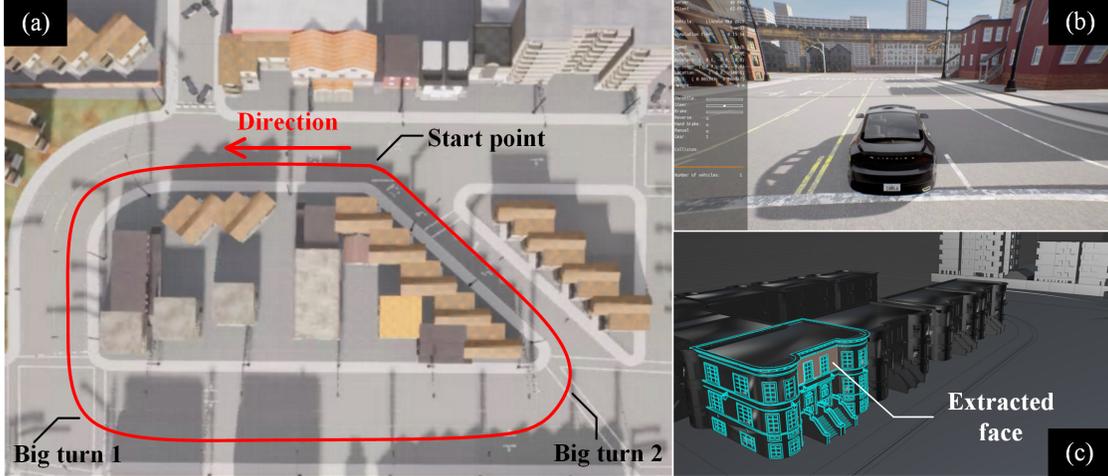
## 4 | NUMERICAL EXPERIMENTS

### 4.1 | Construction of the simulation platform

In this study, we build a simulated urban environment based on the 3D simulator, CARLA (Dosovitskiy et al., 2017). Through automatic and manual checking, we can create a fault-free scenario in the simulated and inject specified faults at a specified time. The main reasons for constructing the simulation platform are listed below:

1. The measurement noise of LiDAR is uncontrollable in the real world. In order to simulate GMM-distributed noises and ensure the reproducibility of experiments, it is utmost to have a fully controllable environment to examine the performance of the proposed method.
2. Faults in real datasets are unpredictable. To our best knowledge, no method can always detect all faults. In order to control the type of faults and the time when faults occur, it is essential to use a simulation tool to generate faults of interest at a given time intentionally.

Some snapshots of the constructed simulation platform are shown in Figure 4. The simulated vehicle is equipped with seven 2D LiDARs and a simulated IMU sensor, as shown in Figure 4(b). The measurements of 2D LiDARs are simulated by the ray-cast technology, which can accurately reflect the real position of the point hit by each laser beam. Users can add extra noises to the 2D LiDAR measurements. The simulated sensors can directly output readings of physical parameters (such as angular velocity and acceleration) of the vehicle through CARLA. Additional noises are incorporated into the readings with customized configurations, eventually simulating the IMU sensor. In the experiment, the noise of 2D LiDAR measurements and IMU measurements are



**FIGURE 4** The simulated platform. (a) The simulated urban environment is constructed based on CARLA, where the designed track of the ground vehicle is marked as the red line; (b) The simulated vehicle moves along the road by manual control or programmed control; (c) All faces of 3D objects in the simulated environment can be extracted and represented correctly by Blender with semi-automatic checking.

configured to be multivariate GMM-distributed. Detailed configuration will be described in Section 4.2. Recall Section 3.2 that the 2D LiDAR measurement model requires pre-stored plane information. In the simulated platform, it is much easier to extract this information without introducing additional errors. We accomplish this by exporting the 3D objects in the simulated environment to Blender (Community, 2018), an open-source 3D modeling and rendering software, to extract the faces of all objects of interest, such as buildings, walls, and roads. As shown in Figure 4(c), the face (plane) information, including normal, center, and vertices, can be extracted correctly by automatic operation and manual checking in Blender.

## 4.2 | Setup of the experiment

Based on the constructed simulation platform, a ground vehicle is programmed to move along the track in Figure 4(a) at a constant speed of 30 km/h. The operation time is 57 s, and the length of the trajectory is 314.3 m. Figure 4(a) plots the simulated trajectory. In the experiment, the output frequency of the LiDARs is 10 Hz, and LiDAR measurement noises are configured to have the multivariate GMM distribution. To set the covariance of the 2D LiDAR measurement noises, we refer to the research conducted by the team from Nagoya University (Carballo et al., 2020), which evaluated the performance of mainstream LiDAR products. Their work found that the root-mean-square error of the ranging measurements varies roughly from 0.01 m to 0.08 m within a measured distance of 100 m. Since the bearing measurements can be directly obtained by referring to the rotation frequency of LiDAR, we assume that the noise of bearing measurements is relatively small. In this study, we use the GMM to set the distribution of the measurement noise  $\eta_i = \begin{bmatrix} \tilde{\phi}_i, \tilde{\rho}_i \end{bmatrix}^T$ . The probability density function of the noise  $\eta_i$  in each 2D LiDAR measurement is given by

$$f_{\eta_i}(\cdot) = p_1 \mathcal{N} \left( \begin{bmatrix} 0 \\ \mu_1 \end{bmatrix}, \begin{bmatrix} 0.0003^2 & 0 \\ 0 & \delta_1^2 \end{bmatrix} \right) + p_2 \mathcal{N} \left( \begin{bmatrix} 0 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} 0.0003^2 & 0 \\ 0 & \delta_2^2 \end{bmatrix} \right), \quad (51)$$

where the bearing noise is set as zero-mean Gaussian with a standard deviation of 0.0003 rad, and the range noise is determined by  $\mathbf{p} = [p_1, p_2]$ ,  $\boldsymbol{\mu} = [\mu_1, \mu_2]$  and  $\boldsymbol{\delta} = [\delta_1, \delta_2]$ . In this experiment, we evaluate four settings of  $\mathbf{p}$ ,  $\boldsymbol{\mu}$  and  $\boldsymbol{\delta}$ , as shown in Table 1. Taking the noise setting N1 as the baseline, we explore scenarios including reducing the difference between mixture weights (N2), diminishing the separation between components (N3), and lessening the disparity between components' variance (N4). The probability density function (PDF) of range noises in four settings is plotted in Figure 5. In addition, a degraded case that the range noise is generated from a Gaussian distribution is also evaluated, as presented in Table 1.

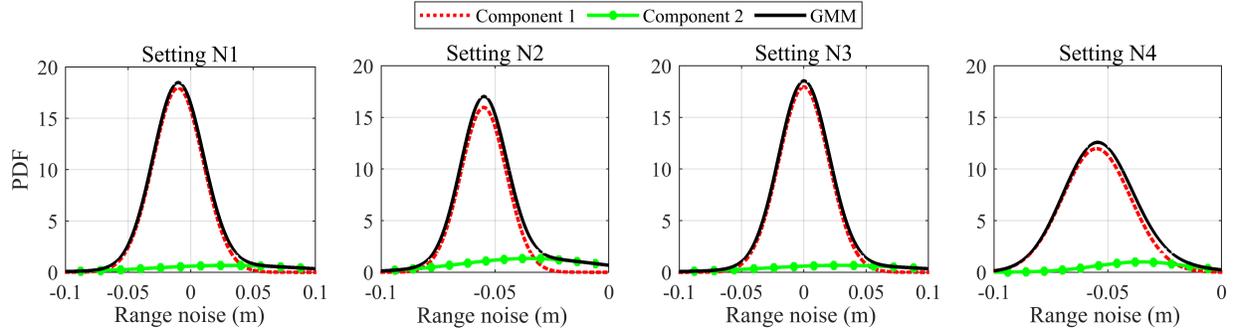


FIGURE 5 The distribution of range noises of 2D LiDAR measurements in four GMM settings.

TABLE 1  
Settings of Range Noises of the LiDAR Measurement

Noise Setting	Mixture $p$	Mean $\mu$ (m)	Std. $\delta$ (m)
N1	[0.9,0.1]	[-0.01,0.03]	[0.02,0.06]
N2	<b>[0.8,0.2]</b>	[-0.01,0.03]	[0.02,0.06]
N3	[0.9,0.1]	<b>[0.00,0.03]</b>	[0.02,0.06]
N4	[0.9,0.1]	[-0.01,0.03]	<b>[0.03,0.04]</b>
Gaussian	-	0	0.03

TABLE 2  
IMU Parameters

Sensor	Parameters	Value
Gyroscope	Bias	2 °/hr
	Noise	0.15 °/√hr
Accelerometer	Bias	3.6 μg
	Noise	0.012 m/sec/√hr

TABLE 3  
LiDAR Range Measurement Failure Coefficients

Failure Type	Group	Coefficient	Value	Failure Time
Step failure	A1	Amplitude	0.1 m	4–20 s
	A2	Amplitude	0.25 m	
Slope failure	B1	Rate	0.05 m/s	34–44 s
	B2	Rate	0.1 m/s	

In this study, we assume that each LiDAR measurements are mutually independent. Therefore, for  $n$  LiDAR measurements, the PDF of the noise can be expressed as follows:

$$f_{\eta}(\cdot) = p_1 \mathcal{N} \left( \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{2n \times 1}, \begin{bmatrix} \mathbf{M}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{M}_1 \end{bmatrix} \right) + p_2 \mathcal{N} \left( \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{2n \times 1}, \begin{bmatrix} \mathbf{M}_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{M}_2 \end{bmatrix} \right), \quad (52)$$

where  $\mathbf{M}_1$  and  $\mathbf{M}_2$  are the covariance matrix of each Gaussian component in the single measurement case, as shown in Equation (51). On the other hand, the probability density function of the process noise  $\mathbf{v} = [\mathbf{n}_a^T, \mathbf{n}_g^T, \mathbf{n}_{wa}^T, \mathbf{n}_{wg}^T]^T \in \mathbb{R}^{12}$  is set as zero-mean multivariate Gaussian distribution with the parameters shown in Table 2, which are simulated based on the MEMS-based IMU (Devices, 2018). The output frequency of the simulated IMU is 100 Hz.

In the EKF setting, we use a perfect initial pose to initialize the EKF. In other words, EKF is assumed to be converged at the initialization and will keep being converged in the following time steps if receiving fault-free measurements. This is a strong assumption, however, applicable in the simulated environment. By making such an assumption, we can focus on the analysis of the residual instead of the convergence of the localization solution. In this study, the standard deviation of the estimated position is set to be 0.05 m, and the estimated orientation represented with quaternion is set as 0.02 at time  $k$ , respectively. To set  $\mathbf{R}$  and  $\mathbf{Q}$ , we calculate the total covariance of  $\boldsymbol{\eta}$  and  $\mathbf{v}$  and set their non-diagonal elements to zero.

In this study, we conducted simulations for two types of single measurement failure, including the step failure and the slope failure, to examine the fault-detection performance across varied scenarios. The coefficients relating to these failures are listed in Table 3. In particular, the slope failure denotes a fault characterized by the escalating magnitude over time (R. Wang et al., 2016).

### 4.3 | Performance of localization

In the fault-free simulated environment, we examine the localization performance of the proposed LiDAR/IMU integrated localization system. Figure 6 plots the absolute translation error of the positioning results under five different settings of measurement noise. It is obvious that the localization performance remains relatively consistent across all noise settings. Throughout most epochs, the absolute translation error is below 0.1 m. In addition, the mean absolute translation error is around 0.06 m during the whole period. However, the absolute translation error shows an increase during the period of 23–30 s and 45–48 s, coinciding with the vehicle's significant turns which are depicted in the thumbnail of Figure 6. This occurrence can be attributed to the lack of features at the turning locations, resulting in an insufficient number of measurements and thus poor observability (Joseph & Murthy, 2019; Sanandaji et al., 2014). Additional sensors, such as cameras and radar, can be employed to capture environmental information to improve the localization performance at these locations. However, such enhancements fall beyond the scope of this work.

### 4.4 | Performance of fault detection

In the experiment, we evaluate the fault detection performance of the proposed method with five settings of measurement noise, four of which are generated from a multivariate GMM distribution, and the remained one is the degraded case that the measurement noise is generated from a Gaussian distribution.

#### 4.4.1 | Step failure detection analysis

Table 4 shows the fault detection results with the step failure when the desired false alarm rate  $\alpha$  is set as 0.05. In group A2, the fault detection rate (FDR) of the proposed total Gaussian-GMM method exceeds 85 % under each noise setting, which is significantly larger than that of the Gaussian method. Particularly within the noise setting N2, the FDR of the total Gaussian-GMM method surpasses that of the Gaussian method by 30 %, showing the most substantial improvement among all noise configurations. As illustrated in Section 4.2, N2 represents the scenario of reducing the difference between mixture weights compared to the baseline setting N1, suggesting that the influence of the Gaussian component with a large variance on the noise distribution is amplified. Compared to the Gaussian method, the total Gaussian-GMM method can effectively capture such change in the characteristics of the noise distribution, thereby exhibiting better detection performance. Nevertheless, the preeminence of the total Gaussian-GMM method in terms of FDR is accompanied by the concession of an ascending false alarm rate (FAR). In all noise settings, the FAR of the total Gaussian-GMM method slightly increases but is still comparable to the desired false alarm rate ( $\alpha = 0.05$ ). Since reducing the missed detection rate is typically prioritized over false positives for multi-sensor navigation systems, the sacrifice in FAR could be accepted to a certain extent.

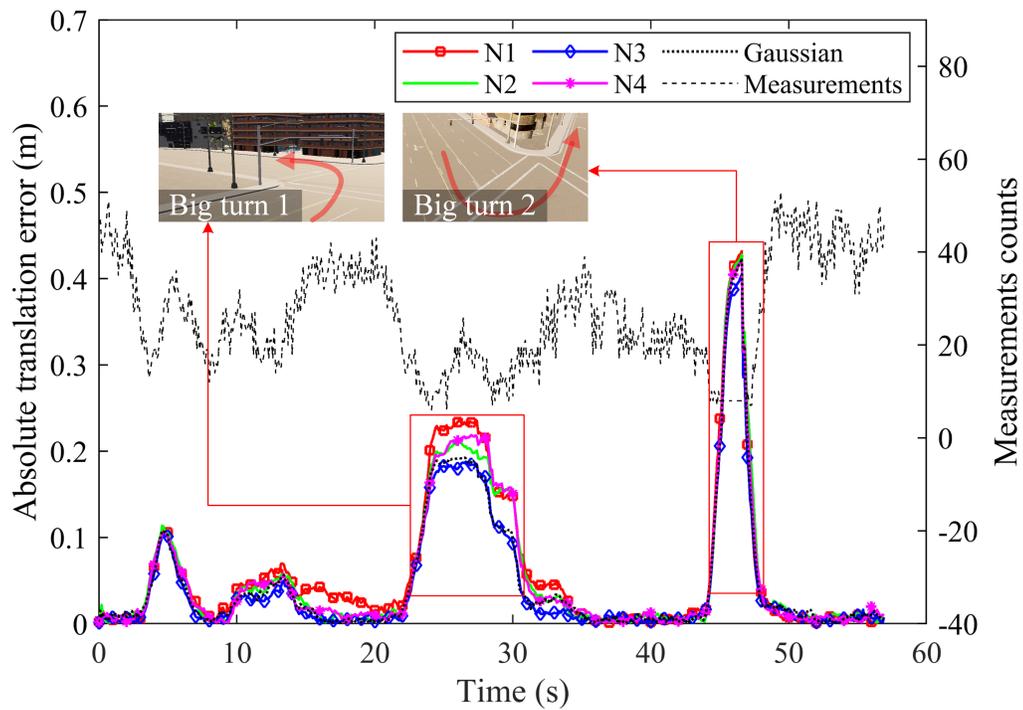


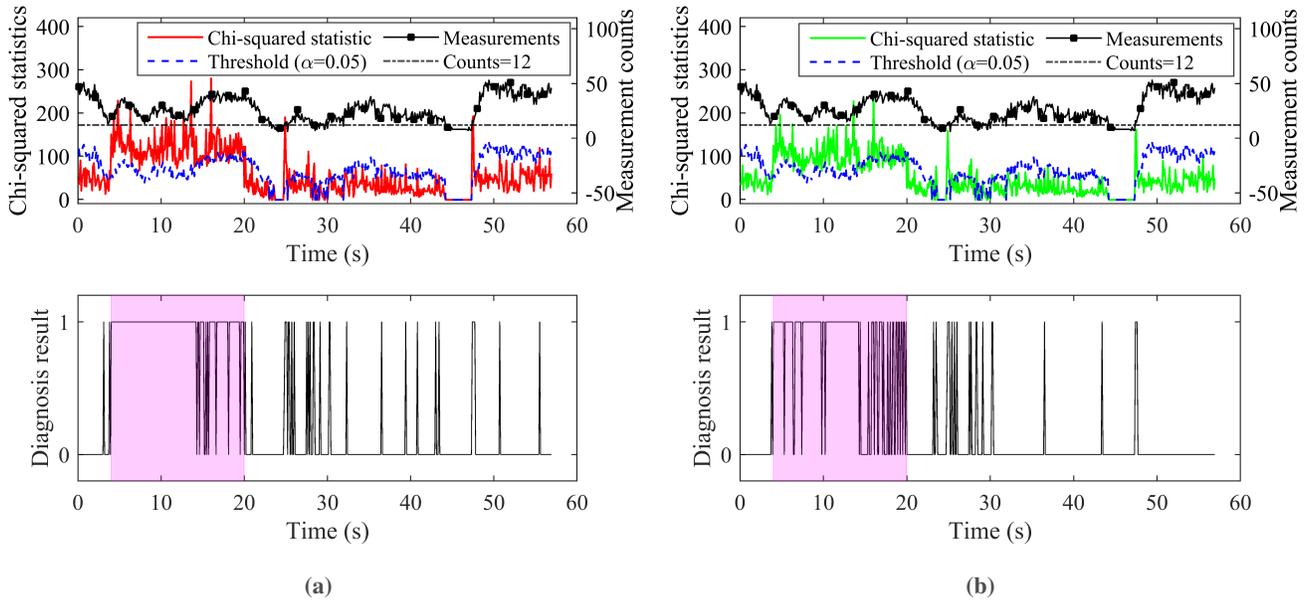
FIGURE 6 The absolute translation error of the positioning results in the fault-free simulated environment.

TABLE 4

Statistical Results of False Alarm Rate (FAR) and Fault Detection Rate (FDR) with Step Failure ( $\alpha = 0.05$ )

Range Noise Setting	Failure Group	Total Gaussian-GMM Method		Gaussian Method	
		FAR (%)	FDR (%)	FAR (%)	FDR (%)
N1	A1	7.26	<b>21.88</b>	5.03	20.00
	A2		<b>94.38</b>		78.13
N2	A1	6.70	<b>23.13</b>	3.63	17.50
	A2		<b>85.00</b>		65.63
N3	A1	5.87	<b>21.25</b>	5.59	20.63
	A2		<b>96.25</b>		92.50
N4	A1	2.79	<b>20.00</b>	2.51	15.00
	A2		<b>85.00</b>		73.75
Gaussian $\mathcal{N}(0, 0.03^2)$	A1	3.07	18.75	3.07	18.75
	A2		93.13		93.13

In group A1, both methods yield a substantially lower FDR compared to that in group A2. This is because the amplitude of the injected faults in group A1 is notably smaller than that in group A2, thereby hindering both methods' ability to detect a fault. Despite this challenge, the total Gaussian-GMM method still exhibits a distinct advantage in detecting faults, showing a



**FIGURE 7** Fault detection results of (a) the total Gaussian-GMM method and (b) the Gaussian method with noise setting N1 for step failure (group A2). In the diagnosis result plot, “1” indicates fault and “0” indicates normal. The failure period is 4–20 s, which is marked by the shaded area.

much higher FDR than the Gaussian method. This result implies that the total Gaussian-GMM method has greater sensitivity to small faults. In the degraded case that measurement noise is generated from a Gaussian distribution, the total Gaussian-GMM method exhibits exactly the same performance as that of the Gaussian method, a result that is anticipated. The detection results for  $\alpha = 0.01$  and  $\alpha = 0.001$  are listed in Appendix F, both of which concur with the findings outlined in Table 4.

Figure 7 plots the statistic curve and the diagnosis result of the two methods with noise setting N1 for step failure A2. The number of valid line measurements from the LiDARs is also plotted in the figures, and the Chi-squared test is only applied when the number of line measurements is no less than 12. As can be seen, the Chi-squared statistics of the total Gaussian-GMM method substantially exceed those of the Gaussian method over the failure period 4–20 s. In addition, the diagnosis results in the figures show that the total Gaussian-GMM method switches less frequently between fault and normal states within the failure period, suggesting that the total Gaussian-GMM method is more stable in the fault detection tasks with step failures.

#### 4.4.2 | Slope failure detection analysis

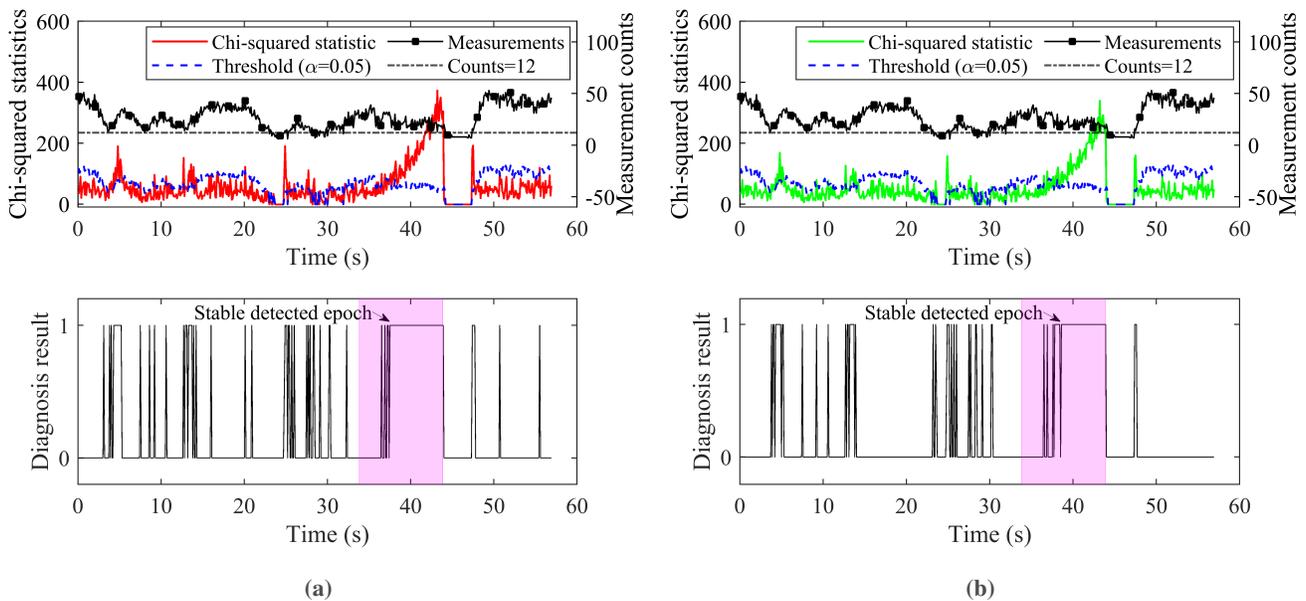
The detection results with the slope failure are listed in Table 5, where the delayed time refers to the interval between the commencement of the fault injection and the first stable detected epoch (i.e., the subsequent diagnosis consistently declares a fault throughout the remaining injection period). In group B1, the total Gaussian-GMM method demonstrates greater sensitivity to slope failure than the Gaussian method. In noise setting N1, N2, and N4, the detection delay for the total Gaussian-GMM method is reduced by 17–23% compared to the Gaussian method. Note that the delayed time of both methods is the same in the noise setting N3, which represents the case of small separation between components. Recall the transformation method based on the law of total covariance in Section 2.2.1, and it is obvious that the second term of the total covariance tends to be zero if all components have a similar mean. In such cases, the total covariance of the GMM distribution approaches the variance of the fitted Gaussian distribution; consequently, the two fault detection methods will exhibit similar performance. This phenomenon is also verified in the step failure experiments under the same noise setting N3, where the total Gaussian-GMM method exhibits only a 3% and 4% increase in FDR compared to the Gaussian method for groups A1 and A2, respectively. Regarding group B2, similar observations are made, with the total Gaussian-GMM method generally exhibiting a shorter delayed time than the Gaussian method. In the Gaussian noise setting, the total Gaussian-GMM method degrades to the Gaussian method, yielding the same delayed time.

The statistic curve and the diagnosis result with noise setting N1 for slope failure B1 are depicted in Figure 8. It is observed that the Chi-squared statistics of both methods consistently rise during the failure period 34–44 s because the magnitude of the fault

increases with time. One distinct difference between the two methods is observed through the diagnosis curve that the diagnosis state of the Total Gaussian-Gaussian method switches more frequently during the unstable period than that of the Gaussian method, indicating that the Total Gaussian-Gaussian method is more sensitive to small faults. However, such sensitivity will lead to an increase in false detection epochs during the normal period.

**TABLE 5**  
Delayed Time of Fault Detection with Slope Failure ( $\alpha = 0.05$ )

Range Noise Setting	Failure Group	Total Gaussian-GMM Method	Gaussian Method
N1	B1	<b>3.5 s</b>	4.6 s
	B2	<b>1.8 s</b>	1.9 s
N2	B1	<b>4.2 s</b>	5.3 s
	B2	<b>2.1 s</b>	2.6 s
N3	B1	3.3 s	3.3 s
	B2	1.7 s	1.7 s
N4	B1	<b>4.3 s</b>	5.2 s
	B2	<b>2.2 s</b>	2.8 s
Gaussian $\mathcal{N}(0, 0.03^2)$	B1	5.3 s	5.3 s
	B2	1.8 s	1.8 s



**FIGURE 8** Fault detection results of (a) the total Gaussian-GMM method and (b) the Gaussian method with noise setting N1 for slope failure (group B1). In the diagnosis result plot, “1” indicates fault and “0” indicates normal. The failure period is 34–44 s, which is marked by the shaded area.

## 5 | CONCLUSION

This study proposes a fault detection method for an EKF-based localization system with GMM noises. The measurement and process noises are then modeled as the multivariate GMM distribution, and the residual of the EKF is derived as the linear combination of these noises, which is shown to be multivariate GMM distributed. Based on the law of total covariance, the residual is transformed into a variable whose distribution approximates a standard MVN distribution. The Mahalanobis distance from the transformed variable to a standard MVN distribution is calculated and taken as the test statistic for a Chi-squared test to detect potential faults. The proposed fault detection method is then applied to a LiDAR/IMU integrated localization system based on the EKF, where the measurement function is constructed by fitting line measurements from seven 2D LiDAR points to the plane, and the state propagation is achieved by the kinematic model of the IMU motion.

In a simulated urban environment constructed based on the 3D simulator, CARLA, we examine the detection performance of the proposed method regarding two types of measurement failures, each of which is examined under four types of GMM noise settings. In the step failure experiment, the fault detection rate of the proposed total Gaussian-GMM surpasses that of the conventional Gaussian method by as large as 30 %, demonstrating the effectiveness of the proposed method in detecting small faults. On the other hand, the detection delay for the total Gaussian-GMM method is reduced by 17–23 % compared to the Gaussian method within three of four noise settings in the slope failure experiment, demonstrating the greater sensitivity of the proposed method to the gradually changing failure. Since the Total-Gaussian GMM method shares the same fault detection methodology with the Gaussian method, the difference in the detection performance between the two methods is consequently attributed to the difference in noise modeling. Therefore, the experiment results also suggest that GMM-based noise modeling is beneficial for fault detection tasks in localization systems with non-Gaussian noises, encouraging researchers to explore this direction.

This study has several limitations, which also point out future research directions. In deriving the relationship between the residual and noises, we ignore high-order terms to reduce the computation load. However, this approximation would underestimate the uncertainty of the residual, thereby increasing the probability of false detection events in the fault detection task. For integrity applications, some integrity and continuity budgets should be allocated to the risks associated with ignoring the high-order terms. A possible solution is to determine the bounding relationship between high-order terms and then scale the low-order terms to compensate for the effects of high-order terms on the uncertainty of the residual. In addition, we assume that the transformation method based on the law of the total variance can transform GMM-distributed residual to a variable whose distribution approaches standard MVN. However, this assumption holds only when each Gaussian component in the GMM-distributed residual has a small difference in the covariance. This constraint is satisfied in the LiDAR/IMU integrated system. However, when the measurement noise exhibits significant heavy tails, the transformation method can not guarantee the similarity, thereby hindering the performance of the proposed fault detection algorithms. Further efforts should be made to develop an advanced transformation method to ensure the similarity between the transformed distribution and the standard MVN distribution, widening the application of non-Gaussian fault detection algorithms in other localization systems, such as GNSS/IMU integrated systems. Moreover, this study focuses on the detection of single measurement failure. Future work can extend the proposed fault detection algorithm to multiple fault detection problems by adopting the exhaustive search or greedy search algorithms as in Blanch et al. (2015). Last but not least, this study gives little concern to the effects of the geometry configuration of 2D LiDARs on state estimation and fault detection performance. For practical purposes, future research could investigate the optimal geometry configuration based on the simulated platform constructed in this study.

## REFERENCES

- Ali-Loytty, S., & Sirola, N. (2007). Gaussian mixture filters and hybrid positioning. *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, 562–570.
- Angus, J. E. (2006). RAIM with multiple faults. *NAVIGATION*, 53(4), 249–257. <https://doi.org/10.1002/j.2161-4296.2006.tb00387.x>
- Blanch, J., Walter, T., & Enge, P. (2008). Position error bound calculation for GNSS using measurement residuals. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3), 977–984. <https://doi.org/10.1109/TAES.2008.4655357>
- Blanch, J., Walker, T., & Enge, P. (2015). Fast multiple fault exclusion with a large number of measurements. *Proceedings of the 2015 International Technical Meeting of The Institute of Navigation*, 696–701.
- Carballo, A., Lambert, J., Monrroy, A., Wong, D., Narksri, P., Kitsukawa, Y., Takeuchi, E., Kato, S., & Takeda, K. (2020). LIBRE: The multiple 3D LiDAR dataset. *2020 IEEE Intelligent Vehicles Symposium (IV)*, 1094–1101.

- Cline, R. E., & Plemmons, R. J. (1976).  $L_2$ -solutions to underdetermined linear systems. *SIAM Review*, 18(1), 92–106. <https://doi.org/10.1137/1018004>
- Community, B. O. (2018). *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam. <http://www.blender.org>
- Da, R. (1994). Failure detection of dynamical systems with the state Chi-square test. *Journal of Guidance, Control, and Dynamics*, 17(2), 271–277. <https://doi.org/10.2514/3.21193>
- Daum, F. (2005). Nonlinear filters: Beyond the Kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8), 57–69. <https://doi.org/10.1109/MAES.2005.1499276>
- Davis, B., & Blair, W. D. (2015). Gaussian mixture approach to long range radar tracking with high range resolution. *2015 IEEE Aerospace Conference*, 1–9. <https://doi.org/10.1109/AERO.2015.7119222>
- Devices, A. (2018). ADIS16465: Precision MEMS IMU module data sheet (Rev. C). <https://www.analog.com/media/en/technical-documentation/data-sheets/adis16465.pdf>
- Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017). CARLA: An open urban driving simulator. *Conference on robot learning*, 1–16.
- Fisher, R. A., & Yates, F. (1953). *Statistical tables for biological, agricultural and medical research*. Hafner Publishing Company.
- Gao, G., Gao, S., Hong, G., Peng, X., & Yu, T. (2020). A robust INS/SRS/CNS integrated navigation system with the Chi-square test-based robust Kalman filter. *Sensors*, 20(20), 5909. <https://doi.org/10.3390/s20205909>
- Gao, Z., Fang, K., Wang, Z., Guo, K., & Liu, Y. (2022). An error overbounding method based on a Gaussian mixture model with uncertainty estimation for a dual-frequency ground-based augmentation system. *Remote Sensing*, 14(5), 1111. <https://doi.org/10.3390/rs14051111>
- Gao, Z., Ding, S. X., & Cecati, C. (2015). Real-time fault diagnosis and fault-tolerant control. *IEEE Transactions on Industrial Electronics*, 62(6), 3752–3756. <https://doi.org/10.1109/TIE.2015.2417511>
- Hashemi, N., & Ruths, J. (2019). Generalized Chi-squared detector for LTI systems with non-Gaussian noise. *2019 American Control Conference (ACC)*, 404–410. <https://doi.org/10.23919/ACC.2019.8815139>
- Hesch, J. A., Mirzaei, F. M., Mariottini, G. L., & Roumeliotis, S. I. (2010). A laser-aided inertial navigation system (L-INS) for human localization in unknown indoor environments. *2010 IEEE International Conference on Robotics and Automation*, 5376–5382. <https://doi.org/10.1109/ROBOT.2010.5509693>
- Hou, H., & El-Sheimy, N. (2003). Inertial sensors errors modeling using Allan variance. *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003)*, 2860–2867.
- Hsu, L. T., Tokura, H., Kubo, N., Gu, Y., & Kamijo, S. (2017). Multiple faulty GNSS measurement exclusion based on consistency check in urban canyons. *IEEE Sensors Journal*, 17(6), 1909–1917.
- Joerger, M., Chan, F.-C., & Pervan, B. (2014). Solution separation versus residual-based RAIM. *NAVIGATION*, 61(4), 273–291. <https://doi.org/10.1002/navi.71>
- Joerger, M., & Pervan, B. (2016). Fault detection and exclusion using solution separation and Chi-squared ARAIM. *IEEE Transactions on Aerospace and Electronic Systems*, 52(2), 726–742. <https://doi.org/10.1109/TAES.2015.140589>
- Joseph, G., & Murthy, C. R. (2019). Measurement bounds for observability of linear dynamical systems under sparsity constraints. *IEEE Transactions on Signal Processing*, 67(8), 1992–2006. <https://doi.org/10.1109/TSP.2019.2899812>
- Langel, S., Crespillo, O. G., & Joerger, M. (2020). Overbounding sequential estimation errors due to non-Gaussian correlated noise. *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 1054–1067.
- Lefferts, E., Markley, F., & Shuster, M. (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5), 417–429. <https://doi.org/10.2514/3.56190>
- Lethander, K. A., & Taylor, C. N. (2023). Conservative estimation of inertial sensor errors using Allan variance data. *NAVIGATION*, 70(3). <https://doi.org/10.33012/navi.563>
- Li, M., & Schwartzman, A. (2018). Standardization of multivariate Gaussian mixture models and background adjustment of PET images in brain oncology. *The Annals of Applied Statistics*, 12(4), 2197–2227.
- Liu, C., Wang, H., Li, N., & Yu, Y. (2017). Sensor fault diagnosis of GPS/INS tightly coupled navigation system based on state Chi-square test and improved simplified fuzzy ARTMAP neural network. *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2527–2532. <https://doi.org/10.1109/ROBIO.2017.8324800>
- MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.
- Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.
- Osechas, O., Misra, P., & Rife, J. (2012). Carrier-phase acceleration RAIM for GNSS satellite clock fault detection. *NAVIGATION*, 59(3), 221–235. <https://doi.org/10.1002/navi.17>

- Pervan, B. S., Pullen, S. P., & Christie, J. R. (1998). A multiple hypothesis approach to satellite navigation integrity. *NAVIGATION*, 45(1), 61–71. <https://doi.org/10.1002/j.2161-4296.1998.tb02372.x>
- Pfeifer, T., & Protzel, P. (2019). Expectation-maximization for adaptive mixture models in graph optimization. *2019 International Conference on Robotics and Automation (ICRA)*, 3151–3157. <https://doi.org/10.1109/ICRA.2019.8793601>
- Pfister, S., Roumeliotis, S., & Burdick, J. (2003). Weighted line fitting algorithms for mobile robot map building and efficient data representation. *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 1, 1304–1311. <https://doi.org/10.1109/ROBOT.2003.1241772>
- Puchalski, R., & Giernacki, W. (2022). UAV fault detection methods, state-of-the-art. *Drones*, 6(11), 330. <https://doi.org/10.3390/drones6110330>
- Rife, J., & Pervan, B. (2012). Overbounding revisited: Discrete error-distribution modeling for safety-critical GPS navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2), 1537–1551.
- Rife, J. H. (2013). The effect of uncertain covariance on a Chi-square integrity monitor. *NAVIGATION*, 60(4), 291–303. <https://doi.org/10.1002/navi.45>
- Rife, J. H. (2018). Overbounding risk for quadratic monitors with arbitrary noise distributions. *IEEE Transactions on Aerospace and Electronic Systems*, 54(4), 1618–1627. <https://doi.org/10.1109/TAES.2018.2798258>
- Sanandaji, B. M., Wakin, M. B., & Vincent, T. L. (2014). Observability with random observations. *IEEE Transactions on Automatic Control*, 59(11), 3002–3007. <https://doi.org/10.1109/TAC.2014.2351693>
- Toschi, I., Rodríguez-González, P., Remondino, F., Minto, S., Orlandini, S., & Fuller, A. (2015). Accuracy evaluation of a mobile mapping system with advanced statistical methods. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40, 245–253.
- Walter, T., & Enge, P. (1995). Weighted RAIM for precision approach. *Proceedings of the 8th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1995)*, 1995–2004.
- Wang, J., Xu, C., Shi, M., & Lu, Z. (2022). Protection level for precise point positioning based on Gaussian mixture model. *China Satellite Navigation Conference*, 45–55.
- Wang, R., Xiong, Z., Liu, J., Xu, J., & Shi, L. (2016). Chi-square and SPRT combined fault detection for multisensor navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 52(3), 1352–1365. <https://doi.org/10.1109/TAES.2016.140860>
- Weiss, N., Holmes, P., & Hardy, M. (2006). *A course in probability*. Pearson Addison Wesley. <https://books.google.com.hk/books?id=Be9fJwAACAAJ>
- Wen, W., Pfeifer, T., Bai, X., & Hsu, L. T. (2021). Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter. *NAVIGATION*, 68(2), 315–331. <https://doi.org/10.1002/navi.421>
- Xu, F., Wang, J., Zhu, D., & Tu, Q. (2018). Speckle noise reduction technique for LiDAR echo signal based on self-adaptive pulse-matching independent component analysis. *Optics and Lasers in Engineering*, 103, 92–99. <https://doi.org/10.1016/j.optlaseng.2017.12.002>
- Yang, L., Knight, N. L., Li, Y., & Rizos, C. (2013). Optimal fault detection and exclusion applied in GNSS positioning. *The Journal of Navigation*, 66(5), 683–700.
- Yun, Y., Yun, H., Kim, D., & Kee, C. (2008). A Gaussian sum filter approach for DGNSS integrity monitoring. *The Journal of Navigation*, 61(4), 687–703. <https://doi.org/10.1017/S0373463308004906>
- Zhao, S., & Farrell, J. A. (2013). 2D LIDAR aided INS for vehicle positioning in urban environments. *2013 IEEE International Conference on Control Applications (CCA)*, 376–381. <https://doi.org/10.1109/CCA.2013.6662778>



## Appendix A | NOTATIONS

**TABLE A1**  
Description of Symbols and Operators

Sym. <sup>1</sup>	Description	Sym.	Description
<b>Symbols in sensor platform and motion model</b>			
${}^G\mathbf{p}_I$	Translation vector from $\{I\}$ to $\{G\}$	${}^G_I\mathbf{R}$	Rotation matrix from $\{I\}$ to $\{G\}$
${}^G\mathbf{v}_I$	Velocity of $\{I\}$ frame in $\{G\}$ frame	${}^L_I\mathbf{p}_I$	Translation vector from $\{I\}$ to $\{L\}$
${}^I\mathbf{q}_G$	Rotation from $\{G\}$ to $\{I\}$ in quaternion	${}^L_I\mathbf{q}_I$	Rotation from $\{I\}$ to $\{L\}$ in quaternion
$\mathbf{b}_a$	Accelerometer measurement bias	${}^L_I\mathbf{R}$	Rotation matrix from $\{I\}$ to $\{L\}$
$\mathbf{b}_g$	Gyroscope measurement bias	$\mathbf{a}_m$	The accelerator measurement
$\mathbf{a}_{GI}$	Acceleration vector of $\{I\}$ w.r.t. <sup>2</sup> $\{G\}$	$\mathbf{w}_m$	The gyroscope measurement
$\mathbf{w}_{GI}$	Angular velocity of $\{G\}$ w.r.t. $\{I\}$	$\Sigma_a$	Covariance of $\mathbf{n}_a$
$\mathbf{n}_a$	Accelerometer measurement noise	$\Sigma_g$	Covariance of $\mathbf{n}_g$
$\mathbf{n}_g$	Gyroscope measurement noise	$\Sigma_{wa}$	Covariance of $\mathbf{n}_{wa}$
$\mathbf{n}_{wa}$	Noise related to $\mathbf{b}_a$	$\Sigma_{wg}$	Covariance of $\mathbf{n}_{wg}$
$\mathbf{n}_{wg}$	Noise related to $\mathbf{b}_g$		
<b>Symbols in LiDAR processing</b>			
$\Pi_i$	$i$ th plane	${}^L\mathbf{x}_i$	The shortest vector in scan plane from $\{L\}$ to $\Pi_i$
${}^G\boldsymbol{\pi}_i$	Normal of the plane $\Pi_i$ in $\{G\}$	$d_i$	Shortest distance from $\{G\}$ to the plane $\Pi_i$
${}^L\mathbf{z}_L$	Normal of the LiDAR scan plane	${}^L\xi_i$	Unit direction vector of ${}^L\mathbf{x}_i$
${}^Ld_i$	${}^Ld_i = d_i - {}^G\boldsymbol{\pi}_i^T ({}^G\mathbf{p}_I + {}^G_I\mathbf{R}^T\mathbf{p}_L)$	$\rho_i$	The length of ${}^L\mathbf{x}_i$
${}^L\mathbf{a}_i$	${}^L\mathbf{a}_i = {}^L_I\mathbf{R}^T {}^G\boldsymbol{\pi}_i = [a_1^i, a_2^i, a_3^i]^T$	$\phi_i$	The angle made by ${}^L\mathbf{x}_i$ and the polar axis
<b>Symbols in EKF based localization</b>			
$\mathbf{x}_k$	State vector at discrete time $k$	$\mathbf{v}_k$	Process noise at time $k$
$\hat{\mathbf{x}}_k^-$	Predicted state at discrete time $k$	$\mathbf{u}_k$	Measurement from IMU at time $k$
$\hat{\mathbf{x}}_k^+$	Estimated state at discrete time $k$	$\mathbf{P}_k^-$	Predicted covariance of estimated state by EKF at time $k$
$n$	Number of measurements	$\mathbf{Q}_k$	Covariance of $\mathbf{v}_k$
$\mathbf{P}_k$	Covariance of estimated state by EKF at time $k$	$\mathbf{K}_k$	The Kalman gain at time $k$
$\mathbf{F}_k$	State transition matrix at time $k$	$\mathbf{G}_k$	Noise Jacobian matrix at time $k$
$\mathbf{y}_{i,k}$	$i$ th measurement at time $k$	$\boldsymbol{\eta}_{i,k}$	Noise of $i$ th measurement at time $k$
$\mathbf{y}_k$	Measurements at time $k$	$\boldsymbol{\eta}_k$	Measurement noise at time $k$
$h^i(\mathbf{x}_k)$	Measurement function associated with $\mathbf{x}_k$ and $\mathbf{y}_{i,k}$	$\mathbf{H}_k^i$	Jacobian matrix of $h^i(\mathbf{x}_k)$
$h(\mathbf{x}_k)$	Measurement function associated with $\mathbf{x}_k$	$\mathbf{H}_k$	Jacobian matrix of $h(\mathbf{x}_k)$
<b>Symbols in residual analysis and fault detection</b>			
$\mathbf{r}_k$	Residual w.r.t. measurements at $k$	$f_\boldsymbol{\eta}(\cdot)$	The PDF of the measurement noise $\boldsymbol{\eta}$
$m_1$	Number of Gaussian modes in $\boldsymbol{\eta}$	$f_\mathbf{v}(\cdot)$	The PDF of the system noise $\mathbf{v}$
$m_2$	Number of Gaussian modes in $\mathbf{v}$	$f_{\mathbf{r}_k}(\cdot)$	The PDF of the residual $\mathbf{r}_k$
$\mathbf{T}_g$	Transformation of $\mathbf{r}_k$ based on GMM assumptions	$t_g$	Test statistic of $\mathbf{r}_k$ based on GMM assumptions
$\mathbf{T}_g^0$	Transformation of $\mathbf{r}_k$ based on Gaussian assumptions	$t_g^0$	Test statistic of $\mathbf{r}_k$ based on Gaussian assumptions
<b>Operators</b>			
$[\cdot]_\times$	Standard vector cross-product	$\times$	Standard vector cross-product
$\cdot$	Vector dot product	$*$	Convolution operator

<sup>1</sup> "Sym." is the abbreviation of "Symbol".

<sup>2</sup> "w.r.t." is the abbreviation of "with respect to".

## Appendix B | DISTRIBUTION OF THE RESIDUAL

Equation (23) shows that  $\mathbf{r}_k$  is the linear combination of independent random variables  $\mathbf{V}_{k-1}\mathbf{v}_{k-1}$ ,  $\mathbf{N}_{k-1}\mathbf{v}_{k-2}$ , and  $\boldsymbol{\eta}_k$ . Therefore, the PDF of  $\mathbf{r}_k$  is the convolution of the PDF of these random variables, which can be written as

$$f_{\mathbf{r}_k} = f_{\mathbf{V}_{k-1}\mathbf{v}} * f_{\mathbf{N}_{k-1}\mathbf{v}} * f_{\boldsymbol{\eta}}, \quad (\text{B1})$$

where  $*$  is the convolution operator. Since  $\mathbf{v}_{k-1}$  and  $\mathbf{v}_{k-2}$  are assumed to be i.i.d., we drop the time index in the above equation. The characteristic functions of  $\mathbf{r}_k$ ,  $\boldsymbol{\eta}$ , and  $\mathbf{v}$  are given as follows:

$$\varphi_{\mathbf{r}_k}(\boldsymbol{\omega}) = \int_{-\infty}^{\infty} f_{\mathbf{r}_k}(\mathbf{x}) e^{i\boldsymbol{\omega}^T \mathbf{x}} d\mathbf{x} \quad (\text{B2a})$$

$$\begin{aligned} \varphi_{\boldsymbol{\eta}}(\boldsymbol{\omega}) &= \int_{-\infty}^{\infty} f_{\boldsymbol{\eta}}(\mathbf{x}) e^{i\boldsymbol{\omega}^T \mathbf{x}} d\mathbf{x} = \sum_{j=1}^{m_1} p_j^\eta \int_{-\infty}^{\infty} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j^\eta, \boldsymbol{\pi}_j^\eta) e^{i\boldsymbol{\omega}^T \mathbf{x}} d\mathbf{x} \\ &= \sum_{j=1}^{m_1} p_j^\eta \varphi_{\mathcal{N}}(\boldsymbol{\omega}) = \sum_{j=1}^{m_1} p_j^\eta e^{i\boldsymbol{\omega}^T \boldsymbol{\mu}_j^\eta - \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{\pi}_j^\eta \boldsymbol{\omega}} \end{aligned} \quad (\text{B2b})$$

$$\varphi_{\mathbf{v}}(\boldsymbol{\omega}) = \int_{-\infty}^{\infty} f_{\mathbf{v}}(\mathbf{x}) e^{i\boldsymbol{\omega}^T \mathbf{x}} d\mathbf{x} = \sum_{j=1}^{m_2} p_j^v e^{i\boldsymbol{\omega}^T \boldsymbol{\mu}_j^v - \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{\pi}_j^v \boldsymbol{\omega}}, \quad (\text{B2c})$$

where  $\varphi_{\mathcal{N}}(\boldsymbol{\omega})$  is the characteristic function of a multinormal variable. Let  $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$  be the linear transformation of the random vector  $\mathbf{x}$ , where  $\mathbf{A}$  is a constant matrix and  $\mathbf{b}$  is a constant vector. The characteristic function of  $\mathbf{y}$  is given by

$$\varphi_{\mathbf{y}}(\boldsymbol{\omega}) = e^{i\boldsymbol{\omega}^T \mathbf{b}} \varphi_{\mathbf{x}}(\mathbf{A}^T \boldsymbol{\omega}). \quad (\text{B3})$$

Therefore, we have

$$\varphi_{\mathbf{N}_{k-1}\mathbf{v}}(\boldsymbol{\omega}) = \varphi_{\mathbf{v}}(\mathbf{N}_{k-1}^T \boldsymbol{\omega}) = \sum_{j=1}^{m_2} p_j^v e^{i\boldsymbol{\omega}^T \mathbf{N}_{k-1} \boldsymbol{\mu}_j^v - \frac{1}{2} \boldsymbol{\omega}^T \mathbf{N}_{k-1} \boldsymbol{\pi}_j^v \mathbf{N}_{k-1}^T \boldsymbol{\omega}} \quad (\text{B4a})$$

$$\varphi_{\mathbf{V}_{k-1}\mathbf{v}}(\boldsymbol{\omega}) = \varphi_{\mathbf{v}}(\mathbf{V}_{k-1}^T \boldsymbol{\omega}) = \sum_{j=1}^{m_2} p_j^v e^{i\boldsymbol{\omega}^T \mathbf{V}_{k-1} \boldsymbol{\mu}_j^v - \frac{1}{2} \boldsymbol{\omega}^T \mathbf{V}_{k-1} \boldsymbol{\pi}_j^v \mathbf{V}_{k-1}^T \boldsymbol{\omega}}. \quad (\text{B4b})$$

According to the convolution theorem,

$$\varphi_{\mathbf{r}_k}(\boldsymbol{\omega}) = \varphi_{\mathbf{V}_{k-1}\mathbf{v}}(\boldsymbol{\omega}) \cdot \varphi_{\mathbf{N}_{k-1}\mathbf{v}}(\boldsymbol{\omega}) \cdot \varphi_{\boldsymbol{\eta}}(\boldsymbol{\omega}), \quad (\text{B5})$$

where  $\cdot$  is point-wise multiplication. Substituting Equations (B2b) and (B4) into Equation (B5), we can obtain

$$\varphi_{\mathbf{r}_k}(\boldsymbol{\omega}) = \sum_{a=1}^{m_2} \sum_{b=1}^{m_2} \sum_{c=1}^{m_1} p_{abc} e^{i\boldsymbol{\omega}^T \boldsymbol{\mu}_{abc} - \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{\pi}_{abc} \boldsymbol{\omega}}, \quad (\text{B6})$$

where

$$p_{abc} = p_a^v p_b^v p_c^\eta \quad (\text{B7a})$$

$$\boldsymbol{\mu}_{abc} = \mathbf{V}_{k-1} \boldsymbol{\mu}_a^v + \mathbf{N}_{k-1} \boldsymbol{\mu}_b^v + \boldsymbol{\mu}_c^\eta \quad (\text{B7b})$$

$$\boldsymbol{\pi}_{abc} = \mathbf{V}_{k-1} \boldsymbol{\pi}_a^v \mathbf{V}_{k-1}^T + \mathbf{N}_{k-1} \boldsymbol{\pi}_b^v \mathbf{N}_{k-1}^T + \boldsymbol{\pi}_c^\eta. \quad (\text{B7c})$$

Do the inverse operation shown in Equation (B2b),

$$\varphi_{\mathbf{r}_k}(\boldsymbol{\omega}) = \sum_{a=1}^{m_2} \sum_{b=1}^{m_2} \sum_{c=1}^{m_1} p_{abc} \int_{-\infty}^{\infty} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{abc}, \boldsymbol{\pi}_{abc}) e^{i\boldsymbol{\omega}^T \mathbf{x}} d\mathbf{x}. \quad (\text{B8})$$

Therefore,

$$f_{\mathbf{r}_k}(\mathbf{x}) = \sum_{a=1}^{m_2} \sum_{b=1}^{m_2} \sum_{c=1}^{m_1} p_{abc} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{abc}, \boldsymbol{\pi}_{abc}) d\mathbf{x}, \quad (\text{B9})$$

which indicates that  $\mathbf{r}_k$  has a multivariate GMM distribution.

## Appendix C | SIMILARITY TO STANDARD MVN DISTRIBUTION

To evaluate the similarity between the transformed distribution and the standard multivariate normal (MVN) distribution, we calculate the discretized Jensen–Shannon divergence between the two distributions (MacKay, 2003). Let  $P(x)$  and  $Q(x)$  be the probability mass function (PMF) of distributions  $P$  and  $Q$ , respectively. The discretized Jensen–Shannon divergence ( $D_{JS}$ ) between  $P$  and  $Q$  defined on the identical sample space is given by

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M) \quad (C1)$$

$$D_{KL}(P||Q) = \sum_{x \in S} P(x) \log \left( \frac{P(x)}{Q(x)} \right),$$

where  $M = \frac{1}{2}(P + Q)$ , and  $D_{KL}$  is the Kullback–Leibler divergence (relative entropy). The minimum value 0 of the  $D_{JS}$  means two distributions are the same, while the maximum value 1 means completely different (MacKay, 2003; Manning & Schutze, 1999). In our work,  $P$  refers to the transformed distribution from a bivariate Gaussian Mixture model (BGMM) and  $Q$  refers to the standard MVN distribution. To comprehensively evaluate the similarity between the transformed distribution and the standard MVN distribution, we start the discussion with the 2-dimensional (2-D) case and then extend the findings to higher-dimensional cases.

### (1) Transformation of 2-dimensional BGMM

The Probability density function (PDF) of a 2-D BGMM can be formalized as

$$f(\mathbf{x}) = p_1 f_N(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1 - p_1) f_N(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), \quad (C2)$$

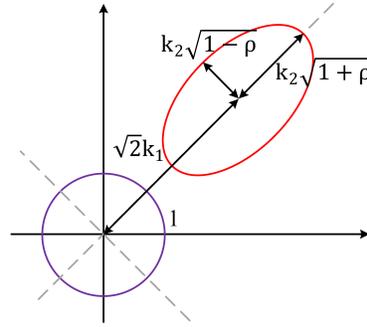
where  $f_N(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$  and  $f_N(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$  are the PDF of the first and the second Gaussian component,  $\boldsymbol{\mu}_1$  and  $\boldsymbol{\mu}_2$  are the corresponding mean value,  $\boldsymbol{\Sigma}_1$  and  $\boldsymbol{\Sigma}_2$  are the corresponding standard deviations, and  $p_1$  and  $1 - p_1$  are the mixing weight of the two Gaussian components, respectively. Equation (C2) is determined by thirteen parameters, i.e.,  $p_1$ , two elements in  $\boldsymbol{\mu}_1$ , two elements in  $\boldsymbol{\mu}_2$ , four elements in  $\boldsymbol{\Sigma}_1$ , and four elements in  $\boldsymbol{\Sigma}_2$ , which span a thirteen-dimension parameter space. It is difficult to determine the parameter combinations in this thirteen-dimension parameter space such that the transformed distribution resembles the standard MVN distribution. Therefore, we use a re-parameterization method to reduce the dimension of the parameter space (Li & Schwartzman, 2018). Specifically, we use four parameters, including  $k_1$ ,  $k_2$ ,  $\rho$ , and  $p_1$ , to define a 2-D BGMM, as shown in Table C1. The 50% density contours of the re-parameterized BGMM with  $\rho = 0.5$  and  $p_1 = 0.5$  are plotted in Figure C1. As can be seen,  $p_1$  determines the weight of two Gaussian components,  $k_1$  determines the distances between the centers of two ellipses,  $k_2$  controls the scale of the ellipse of the 2nd Gaussian component, and  $\rho$  determines the ellipse eccentricity of the 2nd Gaussian component.

TABLE C1  
Re-parameterization of 2-D BGMM

	Mixture Weight	Mean	Variance
Component 1	$p_1$	$\boldsymbol{\mu}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Component 2	$1 - p_1$	$\boldsymbol{\mu}_2 = k_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\boldsymbol{\Sigma}_2 = k_2^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$

Based on the re-parameterized BGMM, We conduct the Monte Carlo simulation to evaluate the similarity between the transformed distribution from a BGMM and the standard MVN distribution. Specifically, for each setting of  $(p_1, k_1, k_2, \rho)$ , we randomly generate  $N$  samples  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  from the 2-D BGMM, where  $N$  is set as  $10^5$  in our experiments. Then, we apply the proposed transformation method to each sample:

$$\mathbf{T}_{i,g} = \boldsymbol{\Sigma}^{-1/2} (\mathbf{x}_i - \boldsymbol{\mu}), \quad (C3)$$



**FIGURE C1** 50% density contours of the re-parameterized BGMM with  $\rho = 0.5$  and  $p_1 = 0.5$ . The purple ellipse stands for the density contour of component 1, while the red ellipse represents the density contour of component 2.

where  $\boldsymbol{\mu}$  is the total mean and  $\boldsymbol{\Sigma}^{-1/2}$  is the principal square root matrix of the total covariance ( $\boldsymbol{\Sigma}$ ) of the BGMM, respectively. According to the law of total covariance (Weiss et al., 2006),  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$  are given by

$$\boldsymbol{\mu} = p_1 \boldsymbol{\mu}_1 + (1 - p_1) \boldsymbol{\mu}_2 \quad (\text{C4a})$$

$$\boldsymbol{\Sigma} = p_1 \boldsymbol{\Sigma}_1 + (1 - p_1) \boldsymbol{\Sigma}_2 + \sum_{j=1}^2 (\boldsymbol{\mu} - \boldsymbol{\mu}_j)(\boldsymbol{\mu} - \boldsymbol{\mu}_j)^T. \quad (\text{C4b})$$

The transformed sample distribution is denoted by  $\{\mathbf{T}_{1,g}, \mathbf{T}_{2,g}, \dots, \mathbf{T}_{N,g}\}$ . Then, we randomly generate  $N$  samples  $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$  from a standard 2-D MVN distribution. Let  $P = \{\mathbf{T}_{1,g}, \mathbf{T}_{2,g}, \dots, \mathbf{T}_{N,g}\}$  and  $Q = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ . We can employ Equation (C1) to calculate the overall similarity between  $P$  and  $Q$ . Since the  $D_{JS}$  is defined on a 1-D distribution, we will show the similarity between  $P$  and  $Q$  in each dimension in the following paragraphs.

Figure C2 shows the  $D_{JS}$  between the 1st dimension of the transformed distribution and the standard 2-D MVN distribution. The results are present in the form of heatmaps with  $\rho \in \{0, 0.5, 1\}$ ,  $k_1 \in \{0.01, 0.1, 1, 10\}$  and  $p_1$  and  $k_2$  vary continuously. It is interesting to find that the lowest  $D_{JS}$  centered in the region of  $k_2 \in [1, 3]$  (except the case of  $\rho = 1, k_1 = 10$ ), where the two Gaussian components have similar covariance. In this region, the total covariance in Equation (C4b) is similar to the covariance of each component, thereby accurately characterizing the uncertainty of the BGMM. Similar results are found in Figure C3, which shows the  $D_{JS}$  between the 2nd dimension of the transformed distribution and the standard 2-D MVN distribution.

## (2) Transformation of higher-dimensional BGMM

We further examine the similarity between the transformed distribution from a  $K$ -dimensional ( $K > 2$ ) BGMM and the standard MVN distribution. Considering it is impossible to conduct experiments with all values of  $K$  (which are infinite), we choose  $K$  to be 27, which is the average of the number of 2D-LiDAR measurements in the simulated experiments in Section 4. We re-parameterize the 27-D BGMM using three parameters, including  $p_1$ ,  $k_1$ , and  $k_2$ , as shown in Table C2. Similar to the 2-D BGMM experiments, we conduct the Monte-Carlo simulation to obtain the transformed sample distribution in different settings of  $p_1$ ,  $k_1$ , and  $k_2$ . Following the analysis procedure in the 2-D BGMM experiments, we need to calculate the  $D_{JS}$  between the transformed distribution and the standard MVN distribution in each dimension, yielding 27 heatmaps. As can be anticipated, this process would complicate the analysis. Therefore, we use an alternative approach to evaluate the similarity between the transformed distribution and the standard MVN distribution.

Let the transform sample distribution be  $P = \{\mathbf{T}_{1,g}, \mathbf{T}_{2,g}, \dots, \mathbf{T}_{N,g}\}$ . For each sample  $\mathbf{T}_{i,g}$  in  $P$ , we calculate its inner product with itself, i.e.,  $\mathbf{T}_{i,g}^T \mathbf{T}_{i,g}$ . Then, we obtain the distribution  $P_1 = \{\mathbf{T}_{1,g}^T \mathbf{T}_{1,g}, \mathbf{T}_{2,g}^T \mathbf{T}_{2,g}, \dots, \mathbf{T}_{N,g}^T \mathbf{T}_{N,g}\}$ . If the distribution  $P$  approaches the standard MVN distribution, the distribution  $P_1$  should approach a chi-squared distribution with 27 degrees of freedom (DOF). Therefore, we can evaluate the similarity between the distribution  $P_1$  and the chi-squared distribution with 27 DOF. Figure C4 shows the  $D_{JS}$  between the distribution  $P_1$  and the chi-squared distribution with 27 DOF in the form of heatmaps with  $k_1 \in \{0.01, 0.1, 1, 10\}$  and  $p_1$  and  $k_2$  vary continuously. Similar to the trends in the 2-D BGMM experiments, the lowest  $D_{JS}$  centered in the region of  $k_2 \in [1, 3]$ . Therefore, the  $P_1$  distribution is similar to the chi-squared distribution with 27 DOF in this region, thus suggesting that the transformed distribution from a 27-D BGMM is similar to the 27-D MVN distribution.

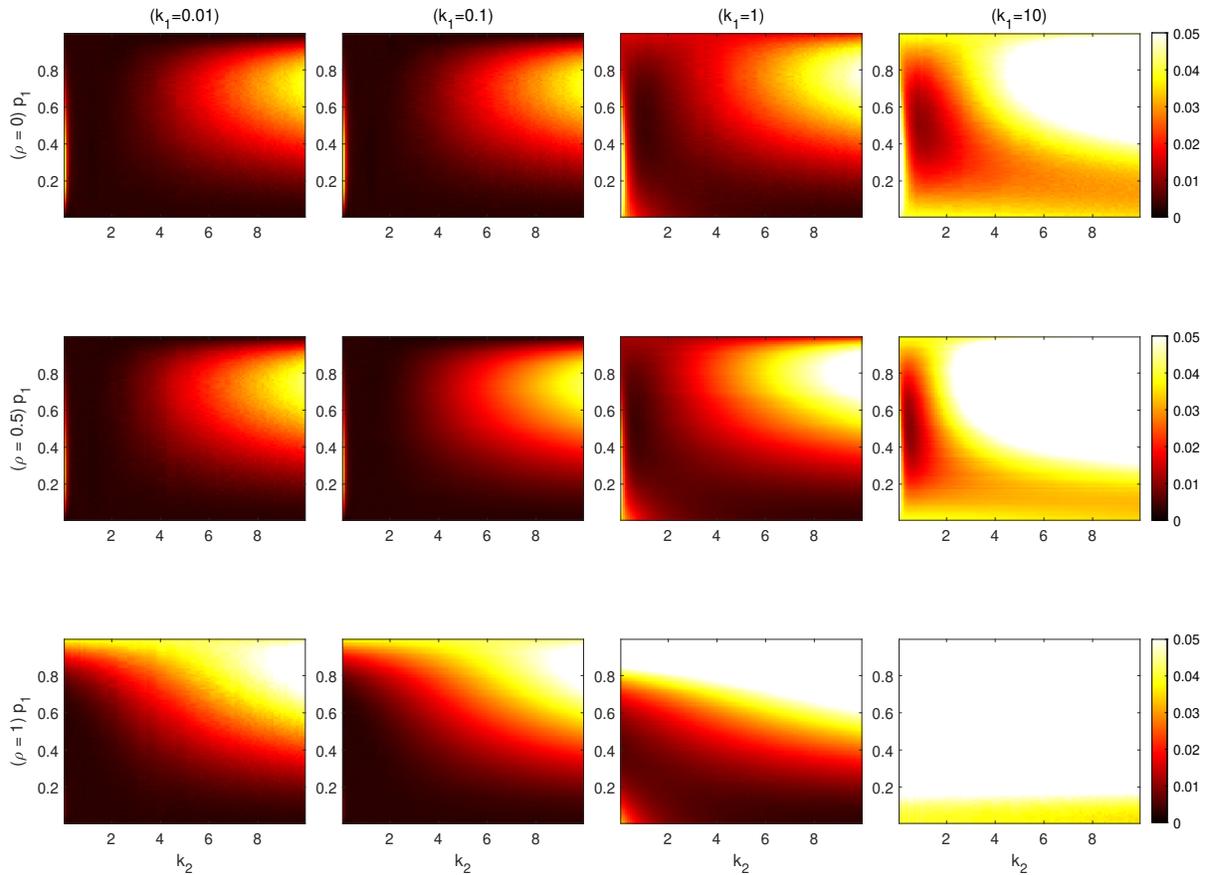
Through the analysis in the 2-D and 27-D BGMM experiments, we find that the similarity assumption holds when each Gaussian component in the GMM-distributed residual has a small difference in the covariance. This constraint is satisfied in the LiDAR/IMU

**TABLE C2**  
Re-parameterization of 27-D BGMM

	Mixture Weight	Mean	Variance
Component 1	$p_1$	$\boldsymbol{\mu}_1 = [0 \dots 0]_{1 \times 27}^T$	$\boldsymbol{\Sigma}_1 = \mathbf{I}_{27}^1$
Component 2	$1 - p_1$	$\boldsymbol{\mu}_2 = k_1 [1 \dots 1]_{1 \times 27}^T$	$\boldsymbol{\Sigma}_2 = k_2^2 \mathbf{I}_{27}$

<sup>1</sup>  $\mathbf{I}_{27}$  is the identity matrix of size 27.

integrated system. This is because the LiDAR measurement error is slightly heavy-tailed (Toschi et al., 2015), indicating that the difference in the covariance between the two Gaussian components is insignificant.



**FIGURE C2** The  $D_{JS}$  between the 1st dimension of the transformed distribution and the standard MVN distribution.

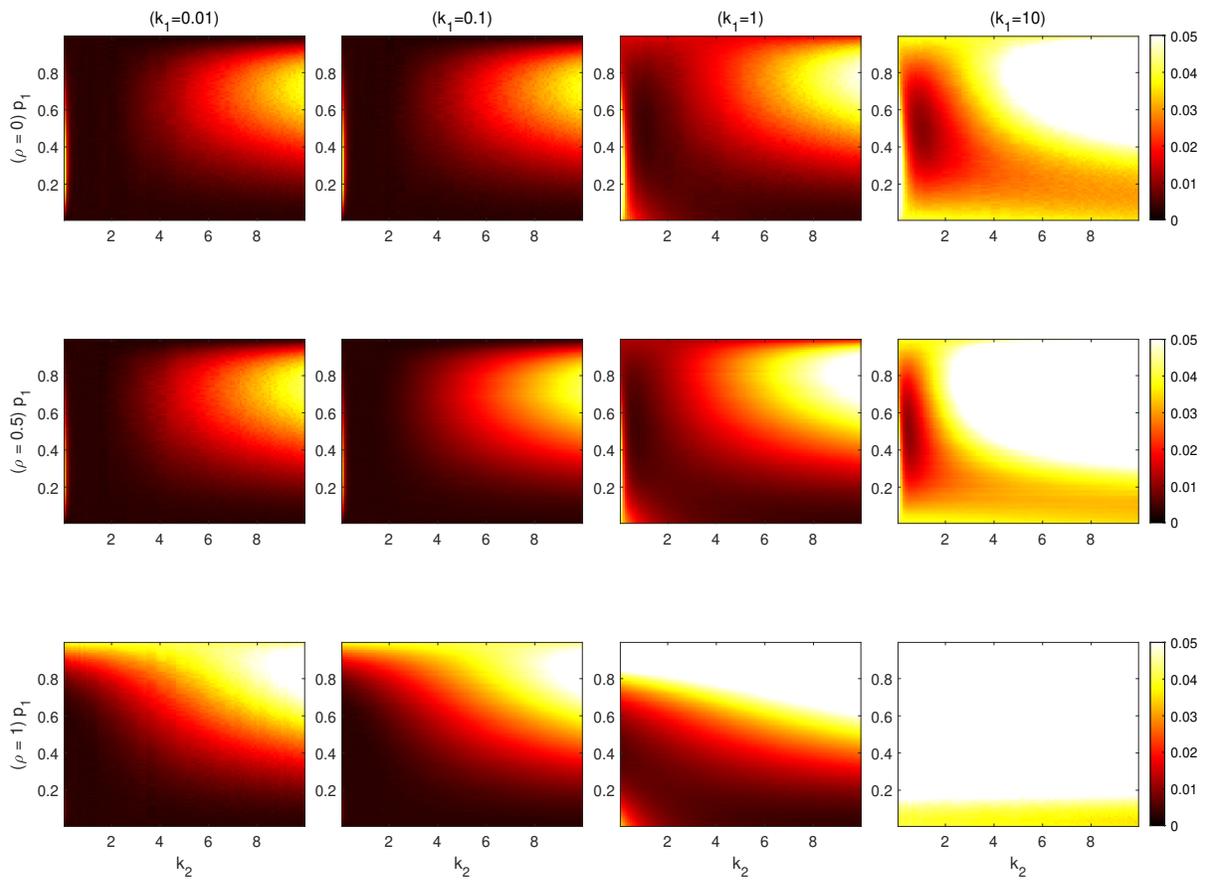


FIGURE C3 The  $D_{JS}$  between the 2nd dimension of the transformed distribution and the standard MVN distribution.

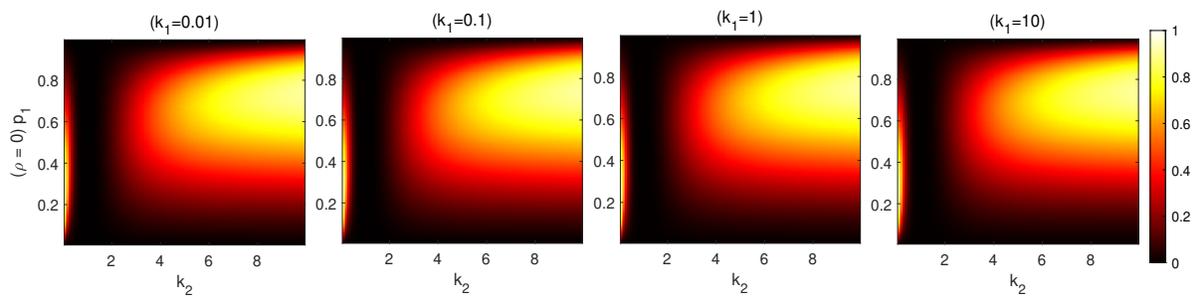


FIGURE C4 The  $D_{JS}$  between the distribution  $P_1$  and the chi-squared distribution with 27 DOF.

## Appendix D | LIDAR PROCESSING

### Appendix D.1 | Least norm solution for LiDAR measurement models

Following notations in Figure 3(b) and their definitions in Section 3.2, we assume that the extrinsic calibration parameters ( ${}^I\mathbf{p}_L, {}^I\mathbf{R}$ ) and the vehicle pose ( ${}^G\mathbf{p}_I, {}^G\mathbf{R}$ ) are known, the plane  $\Pi_i$  associated with the current LiDAR scan plan is found based on the vehicle pose ( ${}^G\mathbf{p}_I, {}^G\mathbf{R}$ ) by a matching algorithm, and  $d_i$  and  ${}^G\boldsymbol{\pi}_i$  are known. We have the following constraints:

(1)  ${}^L\mathbf{x}_i$  is a laser beam, so it must be in the LiDAR scan plane (Zhao & Farrell, 2013)

Let  ${}^L\mathbf{z}_L = [0 \ 0 \ 1]^T$  be the normal of the LiDAR scan plane, we have

$${}^L\mathbf{z}_L^T {}^L\mathbf{x}_i = 0. \quad (\text{D1})$$

(2) Distance constraint (Hesch et al., 2010)

As can be seen in Figure 3(b), the vector from  $\{G\}$  to point  $M$  can be obtained either by

$${}^G\mathbf{p}_I + {}^G\mathbf{R} ({}^I\mathbf{p}_L + {}^I\mathbf{R} {}^L\mathbf{x}_i) \quad (\text{D2})$$

or

$$d_i {}^G\boldsymbol{\pi}_i + {}^G\mathbf{t}_i, \quad (\text{D3})$$

both of which are equivalent. Therefore, we have

$${}^G\mathbf{p}_I + {}^G\mathbf{R} ({}^I\mathbf{p}_L + {}^I\mathbf{R} {}^L\mathbf{x}_i) = d_i {}^G\boldsymbol{\pi}_i + {}^G\mathbf{t}_i. \quad (\text{D4})$$

By projecting Equation (D4) onto  ${}^G\boldsymbol{\pi}_i^T$ , we obtain

$${}^G\boldsymbol{\pi}_i^T ({}^G\mathbf{p}_I + {}^G\mathbf{R} ({}^I\mathbf{p}_L + {}^I\mathbf{R} {}^L\mathbf{x}_i)) = {}^G\boldsymbol{\pi}_i^T d_i {}^G\boldsymbol{\pi}_i + {}^G\boldsymbol{\pi}_i^T {}^G\mathbf{t}_i. \quad (\text{D5})$$

Since  ${}^G\boldsymbol{\pi}_i \perp {}^G\mathbf{t}_i$  and  ${}^G\boldsymbol{\pi}_i$  is a unit vector,

$${}^G\boldsymbol{\pi}_i^T {}^G\mathbf{R} {}^L\mathbf{x}_i = d_i - {}^G\boldsymbol{\pi}_i^T ({}^G\mathbf{p}_I + {}^G\mathbf{R} {}^I\mathbf{p}_L). \quad (\text{D6})$$

By combining Equations (D1) and (D6), we have

$$\mathbf{A} {}^L\mathbf{x}_i = \mathbf{b}, \quad (\text{D7})$$

where

$$\mathbf{A} = \begin{bmatrix} {}^L\mathbf{z}_L^T \\ {}^G\boldsymbol{\pi}_i^T {}^G\mathbf{R} \end{bmatrix}_{2 \times 3}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ {}^Ld_i \end{bmatrix}_{2 \times 1}, \quad (\text{D8})$$

$${}^Ld_i = d_i - {}^G\boldsymbol{\pi}_i^T ({}^G\mathbf{p}_I + {}^G\mathbf{R} {}^I\mathbf{p}_L). \quad (\text{D9})$$

Since  ${}^L\mathbf{x}_i = [\rho_i \cos \phi_i \ \rho_i \sin \phi_i \ 0]^T$  is the shortest vector in the LiDAR scan plane from the origin of  $\{L\}$  to  $\Pi_i$ , we can solve the following optimizing problem to estimate the value of  $\rho_i$  and  $\phi_i$ :

$$\begin{aligned} {}^L\mathbf{x}_i^* &= \arg \min_{{}^L\mathbf{x}_i} \|{}^L\mathbf{x}_i\|^2 \\ \text{s.t. } &\mathbf{A} {}^L\mathbf{x}_i = \mathbf{b}, \end{aligned} \quad (\text{D10})$$

where  ${}^L\mathbf{x}_i^*$  is the least-norm solution of linear equation  $\mathbf{A} {}^L\mathbf{x}_i = \mathbf{b}$  (Cline & Plemmons, 1976). The solution is

$${}^L\mathbf{x}_i^* = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{b}. \quad (\text{D11})$$

Let

$${}^L\mathbf{a}_i = {}^L\mathbf{R} {}^G\boldsymbol{\pi}_i = [a_1^i, a_2^i, a_3^i]^T \quad (\text{D12})$$

which is a unit vector. We can rewrite Equation (D8) as:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 \\ a_1^i & a_2^i & a_3^i \end{bmatrix}. \quad (\text{D13})$$

Therefore,

$$\mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} = \frac{\begin{bmatrix} -a_1^i a_3^i & a_1^i \\ -a_2^i a_3^i & a_2^i \\ 1 - a_3^{i2} & 0 \end{bmatrix}}{1 - a_3^{i2}}. \quad (\text{D14})$$

As  ${}^L\mathbf{a}_i = [a_1^i, a_2^i, a_3^i]^T$  is a unit vector,  $a_1^{i2} + a_2^{i2} + a_3^{i2} = 1$ . We have

$${}^L\mathbf{x}_i^* = \frac{1}{a_1^{i2} + a_2^{i2}} \begin{bmatrix} -a_1^i a_3^i & a_1^i \\ -a_2^i a_3^i & a_2^i \\ 1 - a_3^{i2} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ L d_i \end{bmatrix} = \begin{bmatrix} a_1^i \\ a_2^i \\ 0 \end{bmatrix} \frac{L d_i}{a_1^{i2} + a_2^{i2}} \quad (\text{D15})$$

$$\rho_i^* \begin{bmatrix} \cos \phi_i^* \\ \sin \phi_i^* \\ 0 \end{bmatrix} = \begin{bmatrix} a_1^i \\ a_2^i \\ 0 \end{bmatrix} \frac{L d_i}{a_1^{i2} + a_2^{i2}} = \begin{bmatrix} \frac{a_1}{\sqrt{a_1^{i2} + a_2^{i2}}} \\ \frac{a_2}{\sqrt{a_1^{i2} + a_2^{i2}}} \\ 0 \end{bmatrix} \frac{L d_i}{\sqrt{a_1^{i2} + a_2^{i2}}}. \quad (\text{D16})$$

Therefore, the least norm solution can be obtained as:

$$\rho_i^* = \frac{|L d_i|}{\sqrt{a_1^{i2} + a_2^{i2}}} \quad (\text{D17a})$$

$$\phi_i^* = \arctan \left( \text{sgn}(L d_i) \frac{a_2^i}{a_1^i} \right). \quad (\text{D17b})$$

## Appendix D.2 | Line extraction and feature association

### (1) Extract line segment from 2D LiDAR points

To extract line segments from raw 2D LiDAR measurements, we adopt the method from Pfister et al. (2003), and the main idea is summarized as follows. A line segment  $L_i$  is represented by three parameters  $\rho_i$ ,  $\phi_i$ , and  $S_i$ , where  $(\rho_i, \phi_i)$  is the ranging and bearing parameters associated with the vector perpendicular to  $L_i$  in the polar coordinate system, and  $S_i$  is the distance from the perpendicular to the line segment's center, as shown in Figure 3(a). For  $j$ th 2D LiDAR point, denote its distance from the line segment as  $\delta_j^p$  and its distance from the center of the line segment projected on the line segment as  $\delta_j^s$ . The best-fitted line segment is found by minimizing  $\delta_j^p$  and  $\delta_j^s$  of all LiDAR points by the maximum likelihood estimation.

### (2) Associate line segments with plane

As illustrated in Section 3.2, the laser scan plane intersects with several planes, and the intersection is the line segments. The parameters of these line segments can be predicted according to the current pose of the vehicle, as shown in Equation (44). Therefore, the association of line segments and planes can be achieved by examining the closeness between the predicted line segment and the extracted line segment. For the  $r$ th plane  $\Pi_r$ , the line segment predicted by Equation (44) is parameterized by  $\hat{\mathbf{y}}_r = [\phi_r, \rho_r]^T$ . The extracted line segment can also be parameterized by  $\mathbf{y}_i = [\phi_i, \rho_i]^T$  in the polar coordinate system. The Mahalanobis distance from  $\hat{\mathbf{y}}_r$  to  $\mathbf{y}_i$  is given by  $\|\hat{\mathbf{y}}_r - \mathbf{y}_i\|_{\mathbf{P}_{\mathbf{y}_i}}$ , where  $\mathbf{P}_{\mathbf{y}_i}$  is the covariance matrix of the extracted line segment. The best-fitted plane can be found by minimizing  $\|\hat{\mathbf{y}}_r - \mathbf{y}_i\|_{\mathbf{P}_{\mathbf{y}_i}}$ . Assuming the extracted line segment has the zero-mean Gaussian noise, we can observe that the Mahalanobis distance follows a Chi-squared distribution with 2 DOF. If  $r$ th plane fails to satisfy  $\|\hat{\mathbf{y}}_r - \mathbf{y}_i\|_{\mathbf{P}_{\mathbf{y}_i}} < \delta_a$ , the  $r$ th plane is excluded from the association process. If none of the planes satisfy this condition, the extracted line segment  $\mathbf{y}_i$  is excluded from constructing the final set of measurements. The threshold  $\delta_a$  is determined by a given significance level, which is set as 0.05 (Pfister et al., 2003).

## Appendix E | LINEARIZATION OF EKF-BASED LIDAR/IMU INTEGRATED SYSTEM

### Appendix E.1 | Derivative of the rotation matrix

Let  $\mathbf{g} = [g_x, g_y, g_z]^T$ ,  $\mathbf{q} = [q_w, \mathbf{q}_v^T]^T$ , and  $\mathbf{q}_v = [q_x, q_y, q_z]^T$ , and then the rotation matrix corresponding to  $\mathbf{q}$  is

$$\mathbf{R}\{\mathbf{q}\} = \begin{bmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & 1 - 2(q_x^2 + q_y^2) \end{bmatrix}. \quad (\text{E1})$$

The derivative of  $\mathbf{R}^T\{\mathbf{q}\}\mathbf{g}$  with respect to  $\mathbf{q}$  can be solved by doing the element-wise derivation,

$$\frac{\partial \mathbf{R}^T\{\mathbf{q}\}\mathbf{g}}{\partial \mathbf{q}} = 2 \left[ \mathbf{u}_g \quad [\mathbf{u}_g + q_w \mathbf{g}]_{\times} + (\mathbf{q}_v \cdot \mathbf{g}) \mathbf{I}_3 - \mathbf{g} \mathbf{q}_v^T \right], \quad (\text{E2})$$

where

$$\mathbf{u}_g = \mathbf{g} \times \mathbf{q}_v = [\mathbf{g}]_{\times} \mathbf{q}_v. \quad (\text{E3})$$

Since  $\mathbf{R}\{\mathbf{q}\} = \mathbf{R}^T\{\mathbf{q}^*\}$ , where  $\mathbf{q}^*$  is the conjugate of  $\mathbf{q}$ , we can obtain the derivative of  $\mathbf{R}\{\mathbf{q}\}\mathbf{g}$  with respect to  $\mathbf{q}$ :

$$\frac{\partial \mathbf{R}\{\mathbf{q}\}\mathbf{g}}{\partial \mathbf{q}} = \frac{\partial \mathbf{R}^T\{\mathbf{q}^*\}\mathbf{g}}{\partial \mathbf{q}} = \frac{\partial \mathbf{R}^T\{\mathbf{q}^*\}\mathbf{g}}{\partial \mathbf{q}^*} \frac{\partial \mathbf{q}^*}{\partial \mathbf{q}}, \quad (\text{E4})$$

where

$$\frac{\partial \mathbf{q}^*}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial q_w^*}{\partial q_w} & \frac{\partial q_x^*}{\partial q_x} & \frac{\partial q_y^*}{\partial q_y} & \frac{\partial q_z^*}{\partial q_z} \end{bmatrix} = \text{diag}[1, -1, -1, -1]. \quad (\text{E5})$$

Therefore,

$$\frac{\partial \mathbf{R}\{\mathbf{q}\}\mathbf{g}}{\partial \mathbf{q}} = \frac{\partial \mathbf{R}^T\{\mathbf{q}^*\}\mathbf{g}}{\partial \mathbf{q}^*} \text{diag}[1, -1, -1, -1]. \quad (\text{E6})$$

For convenience, we introduce two symbols,

$$J_q(\mathbf{q}, \mathbf{g}) = \frac{\partial \mathbf{R}^T\{\mathbf{q}\}\mathbf{g}}{\partial \mathbf{q}} \quad (\text{E7a})$$

$$J_q^*(\mathbf{q}, \mathbf{g}) = \frac{\partial \mathbf{R}\{\mathbf{q}\}\mathbf{g}}{\partial \mathbf{q}}. \quad (\text{E7b})$$

### Appendix E.2 | Linearization of the rotation part of the state propagation equation

The rotation part in the state propagation equation in Equation (41) is given by

$${}^I \mathbf{q}_{G,k+1} = g({}^I \mathbf{q}_{G,k}, \mathbf{w}_{GI,k}) = \exp\left(\frac{\Delta t}{2} \boldsymbol{\Omega}[\mathbf{w}_{GI,k}]\right) {}^I \mathbf{q}_{G,k} \quad (\text{E8a})$$

$$\mathbf{w}_{GI,k} = \mathbf{w}_{m,k} - \mathbf{b}_{g,k} - \mathbf{n}_g, \quad (\text{E8b})$$

where  $\exp(\cdot)$  is a function of matrix defined by

$$\exp(\mathbf{M}) = \sum_{i=0}^{\infty} \frac{\mathbf{M}^i}{i!}. \quad (\text{E9})$$

Therefore, Equation (E8a) can be expanded as

$$g({}^I \mathbf{q}_{G,k}, \mathbf{w}_{GI,k}) = \left( \mathbf{I}_4 + \frac{1}{2} \boldsymbol{\Omega}[\mathbf{w}_{GI,k}] \Delta t + \mathcal{O}\left(\left(\boldsymbol{\Omega}[\mathbf{w}_{GI,k}]\right)^2\right) \right) {}^I \mathbf{q}_{G,k}. \quad (\text{E10})$$

By keeping the first-order term and neglecting the high-order terms in the above equations, we have

$$g({}^I \mathbf{q}_{G,k}, \mathbf{w}_{GI,k}) = \left( \mathbf{I}_4 + \frac{1}{2} \boldsymbol{\Omega}[\mathbf{w}_{GI,k}] \Delta t \right) {}^I \mathbf{q}_{G,k} = \begin{bmatrix} \mathbf{I}_3 - \frac{1}{2} \Delta t [\mathbf{w}_{GI,k}]_{\times} & \frac{1}{2} \Delta t \mathbf{w}_{GI,k} \\ -\frac{1}{2} \Delta t \mathbf{w}_{GI,k}^T & 1 \end{bmatrix} {}^I \mathbf{q}_{G,k}. \quad (\text{E11})$$

Let  ${}^I\hat{\mathbf{q}}_{G,k} = [\hat{q}_{w,k}, \hat{q}_{x,k}, \hat{q}_{y,k}, \hat{q}_{z,k}]$  be the estimated vehicle orientation in terms of quaternion at time  $k$ , the Jacobian matrix of  $g({}^I\mathbf{q}_{G,k}, \mathbf{w}_{GI,k})$  with respect to  ${}^I\mathbf{q}_{G,k}$  can be obtained by taking the first-order Taylor expansion at the point  $({}^I\hat{\mathbf{q}}_{G,k}, \mathbf{w}_{m,k})$ ,

$$\frac{\partial g({}^I\hat{\mathbf{q}}_{G,k}, \mathbf{w}_{m,k})}{\partial {}^I\mathbf{q}_{G,k}} = \begin{bmatrix} \mathbf{I}_3 - \frac{1}{2}\Delta t [\mathbf{w}_{GI,k}]_{\times} & \frac{1}{2}\Delta t \mathbf{w}_{GI,k} \\ -\frac{1}{2}\Delta t \mathbf{w}_{GI,k}^T & 1 \end{bmatrix}. \quad (\text{E12})$$

Similarly, the Jacobian matrix of  $g({}^I\mathbf{q}_{G,k}, \mathbf{w}_{GI,k})$  evaluated at  $({}^I\hat{\mathbf{q}}_{G,k}, \mathbf{w}_{m,k})$  with respect to  $\mathbf{w}_{GI,k}$  can be obtained by

$$\frac{\partial g({}^I\hat{\mathbf{q}}_{G,k}, \mathbf{w}_{m,k})}{\partial \mathbf{w}_{GI,k}} = \left[ \frac{\partial g({}^I\hat{\mathbf{q}}_{G,k}, \mathbf{w}_{m,k})}{\partial w_{x,k}}, \frac{\partial g({}^I\hat{\mathbf{q}}_{G,k}, \mathbf{w}_{m,k})}{\partial w_{y,k}}, \frac{\partial g({}^I\hat{\mathbf{q}}_{G,k}, \mathbf{w}_{m,k})}{\partial w_{z,k}} \right] = \frac{\Delta t}{2} \begin{bmatrix} \hat{q}_{z,k} & -\hat{q}_{y,k} & \hat{q}_{x,k} \\ \hat{q}_{y,k} & \hat{q}_{z,k} & -\hat{q}_{w,k} \\ -\hat{q}_{x,k} & \hat{q}_{w,k} & \hat{q}_{z,k} \\ -\hat{q}_{w,k} & -\hat{q}_{x,k} & -\hat{q}_{y,k} \end{bmatrix}. \quad (\text{E13})$$

### Appendix E.3 | Linearization of the measurement function

As defined in Section 3.2,  $h^i(\mathbf{x}_k)$  consists of two parts:  $h_1^i(\mathbf{x})$  and  $h_2^i(\mathbf{x})$ . We are going to derive  $\frac{\partial h_1^i(\hat{\mathbf{x}}_k^-)}{\partial \mathbf{x}_k}$  and  $\frac{\partial h_2^i(\hat{\mathbf{x}}_k^-)}{\partial \mathbf{x}_k}$ , separately. To simplify notations, we omit the indices  $i$  and  $k$  in  $\mathbf{x}_k$ ,  ${}^G\mathbf{p}_{I,k}$ ,  ${}^I\mathbf{q}_{G,k}$ ,  $\hat{\mathbf{x}}_k^-$ ,  ${}^I\hat{\mathbf{p}}_{G,k}^-$ ,  ${}^I\hat{\mathbf{q}}_{G,k}^-$ ,  $a_1^{i,k}$ ,  $a_2^{i,k}$ , and  $a_3^{i,k}$  during the following derivation without loss of generality.

(1) The first part of the Jacobian Matrix

The first part of the Jacobian Matrix  $\mathbf{H}_k^i$  defined in Equation (45) can be written as

$$\frac{\partial h_1^i(\hat{\mathbf{x}}^-)}{\partial \mathbf{x}} = \left[ \frac{\partial h_1^i(\hat{\mathbf{x}}^-)}{\partial {}^G\mathbf{p}_I}, 0_{1 \times 3}, \frac{\partial h_1^i(\hat{\mathbf{x}}^-)}{\partial {}^I\mathbf{q}_G}, 0_{1 \times 6} \right], \quad (\text{E14})$$

where

$$h_1^i(\mathbf{x}) = \arctan \left( \text{sgn}({}^L d_{i,k}) \frac{a_2}{a_1} \right), \quad (\text{E15})$$

$\mathbf{x}$ ,  ${}^G\mathbf{p}_I$ , and  ${}^I\mathbf{q}_G$  are defined in Equation (40),  ${}^L\mathbf{a}_{i,k} = {}^L_G\mathbf{R}^G\boldsymbol{\pi}_{i,k} = [a_1, a_2, a_3]^T$  is the discrete form of  ${}^L\mathbf{a}_i$  defined in Equation (D12), and  ${}^L d_{i,k}$  is the discrete form of  ${}^L d_i$  defined in Equation (D9). According to the chain rule:

$$\frac{\partial h_1^i(\cdot)}{\partial {}^I\mathbf{q}_G} = \frac{\partial h_1^i(\cdot)}{\partial {}^L\mathbf{a}_{i,k}} \cdot \frac{\partial {}^L\mathbf{a}_{i,k}}{\partial {}^I\mathbf{q}_G}. \quad (\text{E16})$$

By substituting Equation (E15) into Equation (E16), the first term of Equation (E16) can be written by

$$\frac{\partial h_1^i(\cdot)}{\partial {}^L\mathbf{a}_{i,k}} = \frac{\partial \arctan \left( \text{sgn}({}^L d_{i,k}) \frac{a_2}{a_1} \right)}{\partial {}^L\mathbf{a}_{i,k}}, \quad (\text{E17})$$

where  $\text{sgn}(\cdot)$  is the sign function and  $\frac{d\text{sgn}(x)}{dx} = 2\delta(x)$ . Now Equation (E17) can be reduced as

$$\frac{\partial h_1^i(\cdot)}{\partial {}^L\mathbf{a}_{i,k}} = \left( 1 + \frac{a_2^2}{a_1^2} \right)^{-1} \left( 2\delta({}^L d_{i,k}) \frac{\partial {}^L d_{i,k}}{\partial {}^L\mathbf{a}_{i,k}} \cdot \frac{a_2}{a_1} + \text{sgn}({}^L d_{i,k}) \frac{\partial \frac{a_2}{a_1}}{\partial {}^L\mathbf{a}_{i,k}} \right). \quad (\text{E18})$$

According to Equations (D6) and (D9),

$${}^L d_{i,k} = d_{i,k} - {}^G\boldsymbol{\pi}_{i,k}^T ({}^G\mathbf{p}_I + {}^G_I\mathbf{R}^I\mathbf{p}_L) = {}^G\boldsymbol{\pi}_{i,k}^T {}^G\mathbf{R}^L\mathbf{x}_{i,k}^*, \quad (\text{E19})$$

where  ${}^G\mathbf{R}^L\mathbf{x}_{i,k}^*$  represents the shortest vector from the origin of  $\{L\}$  to the plane  $\Pi_i$  in  $\{G\}$  at time  $k$ , which cannot be zero. In addition,  ${}^G\boldsymbol{\pi}_{i,k}$  is the normal of the planar surface, which cannot be zero. Therefore,  ${}^L d_{i,k} \neq 0$ , and thus  $\delta({}^L d_{i,k}) = 0$ . Then Equation (E18) is simplified as

$$\frac{\partial h_1^i(\cdot)}{\partial {}^L\mathbf{a}_{i,k}} = \text{sgn}({}^L d_{i,k}) \frac{\partial \frac{a_2}{a_1}}{\partial {}^L\mathbf{a}_{i,k}} \left( 1 + \frac{a_2^2}{a_1^2} \right)^{-1}, \quad (\text{E20})$$

where

$$\frac{\partial \frac{a_2}{a_1}}{\partial^L \mathbf{a}_{i,k}} = \left[ \frac{\partial \frac{a_2}{a_1}}{\partial a_1}, \frac{\partial \frac{a_2}{a_1}}{\partial a_2}, \frac{\partial \frac{a_2}{a_1}}{\partial a_3} \right] = \left[ -\frac{a_2}{a_1^2}, \frac{1}{a_1}, 0 \right]. \quad (\text{E21})$$

Therefore, Equation (E20) has the following expression:

$$\frac{\partial h_1^i(\cdot)}{\partial^L \mathbf{a}_{i,k}} = \frac{1}{\mu_{i,k}} \boldsymbol{\lambda}_{i,k}^T, \quad (\text{E22})$$

where

$$\mu_{i,k} = a_1^2 + a_2^2, \quad (\text{E23a})$$

$$\boldsymbol{\lambda}_{i,k}^T = \text{sgn}({}^L d_{i,k}) [-a_2, a_1, 0]. \quad (\text{E23b})$$

The second term in Equation (E16) is given by

$$\frac{\partial^L \mathbf{a}_{i,k}}{\partial^I \mathbf{q}_G} = \frac{\partial^L \mathbf{R}^G \boldsymbol{\pi}_{i,k}}{\partial^I \mathbf{q}_G} = {}^I \mathbf{R} \frac{\partial^I \mathbf{R}^{TG} \boldsymbol{\pi}_{i,k}}{\partial^I \mathbf{q}_G}. \quad (\text{E24})$$

The above equation contains the partial derivative of the rotation matrix with respect to the quaternion, which can be solved by taking the derivative of the rotation matrix element-wisely, as illustrated in Appendix E.1. Therefore, Equation (E24) can be derived using the conclusion in Appendix E.1:

$$\frac{\partial^L \mathbf{a}_{i,k}}{\partial^I \mathbf{q}_G} = {}^L \mathbf{R} J_q^* ({}^I \mathbf{q}_G, {}^G \boldsymbol{\pi}_{i,k}), \quad (\text{E25})$$

where

$$J_q^* ({}^I \mathbf{q}_G, {}^G \boldsymbol{\pi}_{i,k}) = J_q ({}^I \mathbf{q}_G^*, {}^G \boldsymbol{\pi}_{i,k}) \text{diag} [1, -1, -1, -1] \quad (\text{E26a})$$

$$J_q ({}^I \mathbf{q}_G, {}^G \boldsymbol{\pi}_{i,k}) = 2 \left[ \mathbf{u}_\pi, [\mathbf{u}_\pi + \mathbf{q}_w {}^G \boldsymbol{\pi}_{i,k}]_\times + (\mathbf{q}_w \cdot {}^G \boldsymbol{\pi}_{i,k}) \mathbf{I}_3 - {}^G \boldsymbol{\pi}_{i,k} \mathbf{q}_w^T \right] \quad (\text{E26b})$$

$$\mathbf{u}_\pi = {}^G \boldsymbol{\pi}_{i,k} \times \mathbf{q}_w = [{}^G \boldsymbol{\pi}_{i,k}]_\times \mathbf{q}_w \quad (\text{E26c})$$

$${}^I \mathbf{q}_G = [q_w, \mathbf{q}_v^T]^T = [q_w, q_x, q_y, q_z]^T. \quad (\text{E26d})$$

Substitute Equations (E22) and (E25) into Equation (E16) to obtain

$$\frac{\partial h_1^i(\cdot)}{\partial^I \mathbf{q}_G} = \frac{1}{\mu_{i,k}} \boldsymbol{\lambda}_{i,k}^T {}^L \mathbf{R} J_q^* ({}^I \mathbf{q}_G, {}^G \boldsymbol{\pi}_{i,k}). \quad (\text{E27})$$

Similarly, according to the chain rule,

$$\frac{\partial h_1^i(\cdot)}{\partial^G \mathbf{p}_I} = \frac{\partial h_1^i(\cdot)}{\partial^L \mathbf{a}_{i,k}} \cdot \frac{\partial^L \mathbf{a}_{i,k}}{\partial^G \mathbf{p}_I}. \quad (\text{E28})$$

Since  ${}^L \mathbf{a}_{i,k} = {}^L \mathbf{R}^G \boldsymbol{\pi}_{i,k}$ , which means  ${}^L \mathbf{a}_{i,k}$  is only related to the rotation part of the state, Equation (E28) has the following expression:

$$\frac{\partial h_1^i(\cdot)}{\partial^G \mathbf{p}_I} = 0. \quad (\text{E29})$$

By substituting Equations (E27) and (E29) into Equation (E14), we obtain the final expression of the first part of the Jacobian matrix:

$$\frac{\partial h_1^i(\hat{\mathbf{x}}^-)}{\partial \mathbf{x}} = \left[ 0_{1 \times 6}, \frac{1}{\hat{\mu}_{i,k}^-} \hat{\boldsymbol{\lambda}}_{i,k}^{-T} {}^L \mathbf{R} J_q^* ({}^I \hat{\mathbf{q}}_G^-, {}^G \boldsymbol{\pi}_{i,k}), 0_{1 \times 6} \right], \quad (\text{E30})$$

where  $\hat{\mu}_{i,k}^-$  is defined in Equation (E23a) and  $\hat{\boldsymbol{\lambda}}_{i,k}^-$  is defined in Equation (E23b), both of which are evaluated at  $\hat{\mathbf{x}}^-$ .

(2) The second part of the Jacobian Matrix

The second part of Jacobian Matrix  $\mathbf{H}_k^i$  can be written as

$$\frac{\partial h_2^i(\hat{\mathbf{x}}^-)}{\partial \mathbf{x}} = \left[ \frac{\partial h_2^i(\hat{\mathbf{x}}^-)}{\partial^G \mathbf{p}_I}, 0_{1 \times 3}, \frac{\partial h_2^i(\hat{\mathbf{x}}^-)}{\partial^I \mathbf{q}_G}, 0_{1 \times 6} \right], \quad (\text{E31})$$

where

$$h_2^i(\mathbf{x}) = \frac{|{}^L d_{i,k}|}{\sqrt{a_1^2 + a_2^2}}. \quad (\text{E32})$$

We first consider the derivative with respect to the rotation part,

$$\frac{\partial h_2^i(\cdot)}{\partial^I \mathbf{q}_G} = \frac{1}{\sqrt{a_1^2 + a_2^2}} \frac{\partial |{}^L d_{i,k}|}{\partial^I \mathbf{q}_G} + |{}^L d_{i,k}| \frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial^I \mathbf{q}_G}. \quad (\text{E33})$$

In the first part of Equation (E33),

$$\frac{\partial |{}^L d_{i,k}|}{\partial^I \mathbf{q}_G} = \text{sgn}({}^L d_{i,k}) \frac{\partial {}^L d_{i,k}}{\partial^I \mathbf{q}_G}. \quad (\text{E34})$$

Substitute Equation (D9) into Equation (E34),

$$\frac{\partial |{}^L d_{i,k}|}{\partial^I \mathbf{q}_G} = \text{sgn}({}^L d_{i,k}) \frac{\partial (d_{i,k} - {}^G \boldsymbol{\pi}_{i,k}^T ({}^G \mathbf{p}_I + {}^G \mathbf{R}^I \mathbf{p}_L))}{\partial^I \mathbf{q}_G}. \quad (\text{E35})$$

According to Appendix E.1,

$$\frac{\partial {}^G \mathbf{R}^I \mathbf{p}_L}{\partial^I \mathbf{q}_G} = \frac{\partial {}^I \mathbf{R}^T \mathbf{p}_L}{\partial^I \mathbf{q}_G} = J_q({}^I \mathbf{q}_G, {}^I \mathbf{p}_L). \quad (\text{E36})$$

Substitute Equation (E36) into Equation (E35),

$$\frac{\partial |{}^L d_{i,k}|}{\partial^I \mathbf{q}_G} = -\text{sgn}({}^L d_{i,k}) {}^G \boldsymbol{\pi}_{i,k}^T J_q({}^I \mathbf{q}_G, {}^I \mathbf{p}_L). \quad (\text{E37})$$

In the second part of Equation (E33), the chain rule is applied that

$$\frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial^I \mathbf{q}_G} = \frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial {}^L \mathbf{a}_{i,k}} \cdot \frac{\partial {}^L \mathbf{a}_{i,k}}{\partial^I \mathbf{q}_G}. \quad (\text{E38})$$

By expanding  ${}^L \mathbf{a}_i$  into the elementwise form,

$$\frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial {}^L \mathbf{a}_{i,k}} = \left[ \frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial a_1}, \frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial a_2}, \frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial a_3} \right] = -(a_1^2 + a_2^2)^{-\frac{3}{2}} [a_1, a_2, 0]. \quad (\text{E39})$$

Substitute Equations (E25) and (E39) into Equation (E38),

$$\frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial^I \mathbf{q}_G} = -(a_1^2 + a_2^2)^{-\frac{3}{2}} [a_1, a_2, 0] {}^L \mathbf{R} J_q^*({}^I \mathbf{q}_G, {}^G \boldsymbol{\pi}_{i,k}). \quad (\text{E40})$$

By substituting Equations (E37) and (E40) into Equation (E33), the final expression of Equation (E33) is written as

$$\frac{\partial h_2^i(\cdot)}{\partial^I \mathbf{q}_G} = -\text{sgn}({}^L d_{i,k}) \frac{{}^G \boldsymbol{\pi}_{i,k}^T}{\sqrt{\mu_{i,k}}} J_q({}^I \mathbf{q}_G, {}^I \mathbf{p}_L) + |{}^L d_{i,k}| \boldsymbol{\kappa}_{i,k}^T {}^L \mathbf{R} J_q^*({}^I \mathbf{q}_G, {}^G \boldsymbol{\pi}_{i,k}), \quad (\text{E41})$$

where

$$\boldsymbol{\kappa}_{i,k}^T = -\mu_{i,k}^{-\frac{3}{2}} [a_1, a_2, 0]. \quad (\text{E42})$$

Similarly,

$$\frac{\partial h_2^i(\cdot)}{\partial^G \mathbf{p}_I} = \frac{1}{\sqrt{a_1^2 + a_2^2}} \frac{\partial |{}^L d_{i,k}|}{\partial^G \mathbf{p}_I} + |{}^L d_{i,k}| \frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial^G \mathbf{p}_I}. \quad (\text{E43})$$

In the first part of Equation (E43),

$$\frac{\partial |{}^L d_{i,k}|}{\partial^G \mathbf{p}_I} = \text{sgn}({}^L d_{i,k}) \frac{\partial {}^L d_{i,k}}{\partial^G \mathbf{p}_I}. \quad (\text{E44})$$

Substitute Equation (D9) into Equation (E44),

$$\frac{\partial |{}^L d_{i,k}|}{\partial^G \mathbf{p}_I} = \text{sgn}({}^L d_{i,k}) \frac{\partial \left( d_{i,k} - {}^G \boldsymbol{\pi}_{i,k}^T ({}^G \mathbf{p}_I + {}^G \mathbf{R}^I \mathbf{p}_L) \right)}{\partial^G \mathbf{p}_I} = -\text{sgn}({}^L d_{i,k}) {}^G \boldsymbol{\pi}_{i,k}^T. \quad (\text{E45})$$

In the second part of Equation (E43), since  ${}^L \mathbf{a}_{i,k} = {}^L \mathbf{R}^G \boldsymbol{\pi}_{i,k} = [a_1, a_2, a_3]^T$ ,  $a_1$  and  $a_2$  are only related to the rotation part of the state. Therefore,

$$\frac{\partial (a_1^2 + a_2^2)^{-\frac{1}{2}}}{\partial^G \mathbf{p}_I} = 0. \quad (\text{E46})$$

Substitute Equations (E44) and (E46) into Equation (E43),

$$\frac{\partial h_2^i(\cdot)}{\partial^G \mathbf{p}_I} = -\text{sgn}({}^L d_{i,k}) \frac{{}^G \boldsymbol{\pi}_{i,k}^T}{\sqrt{\hat{\mu}_{i,k}}}. \quad (\text{E47})$$

By substituting Equations (E41) and (E47) into Equation (E31), we obtain the final expression of the second part of the Jacobian matrix,

$$\frac{\partial h_2^i(\hat{\mathbf{x}}^-)}{\partial \mathbf{x}} = \left[ -\text{sgn}({}^L \hat{d}_{i,k}^-) \frac{{}^G \boldsymbol{\pi}_{i,k}^T}{\sqrt{\hat{\mu}_{i,k}^-}}, 0_{1 \times 3}, -\text{sgn}({}^L \hat{d}_{i,k}^-) \frac{{}^G \boldsymbol{\pi}_{i,k}^T}{\sqrt{\hat{\mu}_{i,k}^-}} J_q \left( {}^I \hat{\mathbf{q}}_G^-, {}^I \mathbf{p}_L \right) + |{}^L \hat{d}_{i,k}^-| \hat{\boldsymbol{\kappa}}_{i,k}^T {}^L \mathbf{R} J_q^* \left( {}^I \hat{\mathbf{q}}_G^-, {}^G \boldsymbol{\pi}_{i,k} \right), 0_{1 \times 6} \right], \quad (\text{E48})$$

where  $\hat{\boldsymbol{\kappa}}_{i,k}^T$  is defined in Equation (E42) and evaluated at  $\hat{\mathbf{x}}^-$ .

By re-arranging Equations (E30) and (E48), the Jacobian matrix of  $h^i(\mathbf{x}_k)$  defined with respect to  $\mathbf{x}_k$  and evaluated at  $\hat{\mathbf{x}}_k^-$  can be written as

$$\mathbf{H}_k^i = \begin{bmatrix} 0_{1 \times 3} & 0_{1 \times 3} & \frac{1}{\hat{\mu}_{i,k}^-} \hat{\boldsymbol{\lambda}}_{i,k}^{-T} {}^L \mathbf{R} J_q^* \left( {}^I \hat{\mathbf{q}}_G^-, {}^G \boldsymbol{\pi}_{i,k} \right) & 0_{1 \times 6} \\ -\text{sgn}({}^L \hat{d}_{i,k}^-) \frac{{}^G \boldsymbol{\pi}_{i,k}^T}{\sqrt{\hat{\mu}_{i,k}^-}} & 0_{1 \times 3} & -\text{sgn}({}^L \hat{d}_{i,k}^-) \frac{{}^G \boldsymbol{\pi}_{i,k}^T}{\sqrt{\hat{\mu}_{i,k}^-}} J_q \left( {}^I \hat{\mathbf{q}}_G^-, {}^I \mathbf{p}_L \right) + |{}^L \hat{d}_{i,k}^-| \hat{\boldsymbol{\kappa}}_{i,k}^T {}^L \mathbf{R} J_q^* \left( {}^I \hat{\mathbf{q}}_G^-, {}^G \boldsymbol{\pi}_{i,k} \right) & 0_{1 \times 6} \end{bmatrix}. \quad (\text{E49})$$

## Appendix F | MORE RESULTS OF FAULT DETECTION

**TABLE F1**

Statistical Results of False Alarm Rate (FAR) and Fault Detection Rate (FDR) with Step Failure ( $\alpha = 0.01$ )

Range Noise Setting	Failure Group	Total Gaussian-GMM Method		Gaussian Method	
		FAR (%)	FDR (%)	FAR (%)	FDR (%)
N1	A1	4.47	<b>18.75</b>	2.79	14.38
	A2		<b>85.00</b>		71.88
N2	A1	3.07	<b>18.13</b>	1.40	12.50
	A2		<b>72.50</b>		52.50
N3	A1	4.47	14.38	3.35	15.00
	A2		<b>91.25</b>		88.75
N4	A1	1.68	<b>11.88</b>	1.40	9.38
	A2		<b>71.88</b>		60.63
Gaussian $\mathcal{N}(0, 0.03^2)$	A1	2.23	11.88	2.23	11.88
	A2		68.75		68.75

**TABLE F2**

Statistical Results of False Alarm Rate (FAR) and Fault Detection Rate (FDR) with Step Failure ( $\alpha = 0.001$ )

Range Noise Setting	Failure Group	Total Gaussian-GMM Method		Gaussian Method	
		FAR (%)	FDR (%)	FAR (%)	FDR (%)
N1	A1	2.23	<b>14.38</b>	1.96	8.75
	A2		<b>73.75</b>		62.50
N2	A1	1.11	<b>10.00</b>	0.84	6.88
	A2		<b>58.13</b>		37.50
N3	A1	1.68	<b>9.38</b>	1.68	8.13
	A2		<b>81.88</b>		76.20
N4	A1	1.40	<b>5.63</b>	0.84	3.13
	A2		<b>55.63</b>		47.50
Gaussian $\mathcal{N}(0, 0.03^2)$	A1	0.56	6.88	0.56	6.88
	A2		68.13		68.13