Received: 28 April 2020 | Revised: 25 February 2021

DOI: 10.1002/navi.421

ORIGINAL ARTICLE

Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter

Weisong Wen[1] | Tim Pfeifer[2] | Xiwei Bai[1] | Li-Ta Hsu[1]

[1] Hong Kong Polytechnic University, Hung Hom, Hong Kong

[2] Chemnitz University of Technology, Chemnitz, Germany

Correspondence
Li-Ta Hsu, Hong Kong Polytechnic University, Hung Hom, Hong Kong
Email: lt.hus@polyu.edu.hk
Tim Pfeifer, Chemnitz University of Technology, Chemnitz, Germany
Email: tim.pfeifer@etit.tu-chemnitz.de

Funding information
This work is supported by the Emerging Frontier Areas (EFA) scheme by Research Institute for Sustainable Urban Development, The Hong Kong Polytechnic University–BBWK "Resilient Urban PNT Infrastructure to Support Safety of UAV Remote Sensing in Urban Region."

Abstract
Factor graph optimization (FGO) recently has attracted attention as an alternative to the extended Kalman filter (EKF) for GNSS-INS integration. This study evaluates both loosely and tightly coupled integrations of GNSS code pseudorange and INS measurements for real-time positioning, using both conventional EKF and FGO with a dataset collected in an urban canyon in Hong Kong. The FGO strength is analyzed by degenerating the FGO-based estimator into an "EKF-like estimator." In addition, the effects of window size on FGO performance are evaluated by considering both the GNSS pseudorange error models and environmental conditions. We conclude that the conventional FGO outperforms the EKF because of the following two factors: (1) FGO uses multiple iterations during the estimation to achieve a robust estimation; and (2) FGO better explores the time correlation between the measurements and states, based on a batch of historical data, when the measurements do not follow the Gaussian noise assumption.

KEYWORDS
extended Kalman filter, factor graph optimization, GNSS, INS, integration, navigation, positioning, urban canyons, window size

1 | INTRODUCTION

The Global Navigation Satellite System (GNSS) provides all-weather and globally referenced positioning in outdoor environments. However, the accuracy of GNSS positioning may be severely degraded in urban canyons with tall buildings, due to the multipath effect and non-line-of-sight (NLOS) (Groves, 2013) reception caused by reflections from and blockage by buildings. In contrast, an inertial navigation system (INS) (Zhuang & El-Sheimy, 2015; Zhuang et al., 2016) is less dependent on environmental conditions and provides relatively linear acceleration and angular velocity measurements at a high output frequency. However, INS suffers from error accumulation over time. Therefore, the GNSS and INS are complementary, and their integration to form GNSS-INS is a promising solution for vehicular positioning.

Various GNSS-INS integration frameworks were reviewed in Angrisano (2010). Thus far, the most integration solutions available are loosely coupled (LC) (Deng et al., 2017; Hsu et al., 2015; Solimeno, 2007), tightly coupled (TC) (Petovello, 2003), and ultra-tightly coupled (UTC) (Gao & Lachapelle, 2008) integrations. UTC integration requires the baseband signal processing of the GNSS receiver to be changed, which cannot be done by most GNSS-INS integrated system developers. The key difference between LC and TC integrations is in the measurement domain that is used. In LC GNSS-INS integration, the position and velocity estimated by the GNSS receiver are directly incorporated into the INS navigation solution. In contrast, TC integration uses raw GNSS measurements, such as pseudorange, Doppler frequency, and carrier phase measurements. TC GNSS-INS integration based on the extended Kalman filter (EKF) performs better than LC integration, as shown in Petovello (2003), largely because the estimator is optimized over a wider range of possibilities. In summary, TC GNSS-INS integration using the EKF is a common solution in current applications.

The recently proposed factor graph optimization (FGO) (Dellaert & Kaess, 2017) method is akin to a well-known approach in the robotics field for the development of visual (Qin et al., 2018) or light detection and ranging (LiDAR) simultaneous localization and mapping (SLAM) (Wen et al., 2018) to integrate diverse sensor measurements through nonlinear optimization. FGO also exhibits huge potential for GNSS-INS integration and therefore has attracted considerable attention (Li et al., 2018; Wen, Bai et al., 2019; Zhao et al., 2014, 2016). However, no previous study has used urban datasets to compare the performances of FGO and EKF algorithms for the development of low-cost LC and TC GNSS-INS integrations. The present paper aims to fill this gap, by showing how FGO can help mitigate the effects of GNSS outlier measurements on GNSS-INS integration and comparing the results with those obtained with the EKF estimator. In addition, we analyze the effects of window size on FGO performance and discuss the computational efficiencies of EKF and FGO.

The remainder of this paper is structured as follows: The studies related to the application of the EKF and FGO for GNSS-INS integration are reviewed in Section 2. The methodologies used to evaluate the four GNSS-INS integrations are presented in Section 3, and their experimental evaluations are presented in Section 4. Finally, conclusions and future work are presented in Section 5.

2 | RELATED WORK

2.1 | Role of EKF in GNSS-INS integration

The Bayesian filter (Thrun, 2000) has dominated GNSS-INS integration since the early 2000s. The Kalman filter (Welch & Bishop, 1995), the EKF (Julier & Uhlmann, 1997; Roysdon & Farrell, 2017), and the unscented Kalman filter (Wan & Van Der Merwe, 2000) are extremely popular, due to their maturity and computational efficiency in implementations. Several studies (Crassidis et al., 2007; Groves, 2015; Liu et al., 2010) integrated GNSS-INS sensors using an EKF estimator. Numerous practical applications demonstrated the robustness and effectiveness of the EKF in GNSS when the measurement quality is reasonable and the error noise is appropriately modeled. For example, EKF-based GNSS-INS integration is effective in both sparse and open areas with good sky visibility. The conventional EKF efficiently achieves the optimal estimation if: 1. the first-order Markov chain is used, and 2. the random noise is modeled as Gaussian distribution (Barfoot, 2017). However, this is seldom the case when a GNSS receiver is located in urban areas. According to the findings obtained in (Zhao et al., 2016) and (Wen, Kan et al., 2019), GNSS measurements are non-Gaussian-distributed and highly time-correlated in urban canyons. Therefore, EKF-based sensor fusion fails to achieve optimal performance in such areas. From a mathematical perspective, under the Markov assumption, the EKF only evaluates the Jacobians at a single time-step (i.e., performs a single iteration) to achieve its recursive form. Thus, if an outlier is misjudged as a healthy measurement and the corresponding uncertainty is not modeled appropriately, the EKF is highly likely to be misled. This is unacceptable in applications that require accurate positioning services, such as unmanned aerial vehicles (Saripalli et al., 2003) and autonomous driving vehicles (Litman, 2015). In contrast, outlier measurement is frequently performed for GNSS positioning in urban canyons. To improve the performance of outlier measurement, the previous epochs of the state in the state vector may be considered. However, this substantially increases the size and decreases the convergence of the EKF estimator (Valiente et al., 2014).

Recently, the multistate constrained Kalman filter (MSCKF) (Li & Mourikis, 2013) was proposed for use in a visual SLAM field to integrate the information obtained from an INS and a camera. The MSCKF updates the states based on the geometry constraints of the feature measurements conducted inside a sliding window. However, to reduce the size of the features, their states are eliminated from the MSCKF by using a nullspace matrix. In addition, the EKF performance (Julier & Uhlmann, 1997) relies on accurate linearization, due to the nonlinearity of the observation function, and only a single linearization is performed in the EKF. Thus, the linearization accuracy relies heavily on the initial guess of the state. To address this issue, an iterated Kalman filter (Bell & Cathey, 1993) was used to perform multiple iterations during the update step, via the Gauss-Newton method, which helped to mitigate the error generated from the linearization steps.

2.2 | FGO in GNSS-INS integration

The recently proposed FGO formulation (Indelman et al., 2013) has opened a new avenue for multisensor integration (Chen & Gao, 2019; Pfeifer & Protzel, 2019a, 2019b). It is represented as a probabilistic graphical model containing various nodes associated with the system states and factors representing the measurements. The factor graph encodes the posterior probability of the states over time, and unlike the conventional EKF-based integration, considers both historical measurements and system updates to optimize the complete state set. In this case, the historical information is used in FGO, and after all measurements and states are encoded into a factor graph, the sensor fusion problem is iteratively solved through optimization via the Gauss-Newton method. Therefore, the error resulting from the linearization steps is mitigated. Moreover, FGO handles delayed measurements, as these are simply additional sources of factors that are added to the factor graph as they are received. Thus, FGO is used in various challenging GNSS scenarios (Bhamidipati, 2018; Huang, 2016; Watson & Gross, 2018; Zheng Gong, 2018), and the authors in (Pfeifer & Protzel, 2018) showed the strong potential of FGO in sensor fusion, even when the sensor noise is modeled with a non-Gaussian distribution.

Researchers at the Georgia Institute of Technology (Indelman et al., 2012) used simulated data to demonstrate that the use of FGO in LC GNSS-INS integration led to better performance than the use of the EKF estimator. However, there was only a limited improvement in LC integration. A team from Tsinghua University (Li et al., 2018) extended this line of research by studying TC GNSS-INS integration and obtained substantially better performance than that achieved with the EKF estimator. Another study analyzed the performance of TC GNSS-INS–fisheye camera integration (Wen, Bai et al., 2019) using FGO based on a challenging dataset collected in urban canyons. In this study, a fisheye camera was innovatively used to model the uncertainty of GNSS pseudorange measurements before its integration with INS using FGO. The results indicated that TC FGO performed better than the TC EKF estimator. However, the LC integration performance of the EKF and FGO was not evaluated (Wen, Bai et al., 2019).

In another approach, researchers at the University of California, Riverside (Zhao et al., 2016), proposed an optimization-based sliding window for the integration of differential GNSS (DGNSS) and an INS, which they named the contemplative real-time (CRT) method. Based on sensor measurements, a Gauss-Newton method was then applied to optimize the states inside the window. CRT shares the same batch-processing theoretical basis with the factor graph approach. Moreover, its window is similar in size to the fixed-lag size considered in Indelman et al. (2013). The authors in Zhao et al. (2014) presented a CRT method for DGNSS-INS integration with a window size of 10 s and demonstrated the advantage of CRT over the TC EKF estimator. The error sources for the pseudorange measurements were mitigated through double-difference analysis before the CRT was integrated with the INS, leading to decimeter-level positioning accuracy. Therefore, the applicability of CRT in challenging urban canyons using a low-cost GNSS receiver, which may introduce a large pseudorange error, is yet to be explored. Continuous works are presented in Zhao et al. (2016) by adding more measurements into the CRT. However, CRT performance relies heavily on the size of the sliding window. Our published conference paper showed that different window sizes led to different performance improvements. The same phenomenon also was observed in our previous study (Wen, Kan, et al., 2019). As an extension to that study, the present paper makes two additional contributions, as below:

- An extensive literature review is conducted, and a detailed analysis of existing sensor fusion schemes, including iterative EKF-, MSCKF-, and CRT-based fusion methods, is performed.
- Additional experimental analyses are conducted. A sky-view image is provided, and a satellite is projected onto this image to show the environmental conditions. In addition, more detailed error modeling of the pseudorange measurements based on the Gaussian mixture model (GMM) is conducted to analyze the effect of the window size on FGO performance.

3 | METHODOLOGY

This study evaluates the following four GNSS-INS integrations: 1. EKF-based LC GNSS-INS, 2. EKF-based TC GNSS-INS, 3. FGO-based LC GNSS-INS, and 4. FGO-based TC GNSS-INS integrations. We follow the approaches described in Falco et al. (2017) and Li et al. (2018) for implementing EKF- and FGO-based integrations, respectively. Note that we only evaluate positional performance, and the estimated position state lies in the Earth-centered Earth-fixed (ECEF) frame. We also only use the linear acceleration measurement and attitude provided by the attitude and heading reference system (AHRS), which is a commercial solution of the INS used in this study. By propagating the acceleration measurements, we determine the difference in the velocities between the two epochs. Similar to Li et al. (2018), we use a constant-velocity motion model to derive the positional difference between the two epochs. These two steps constitute the implementation of the INS mechanism in FGO. However, to determine GNSS positioning, we use raw pseudorange measurements. The methodologies of the abovementioned four integrations are introduced as follows.

3.1 | LC GNSS-INS integration using the EKF

The flowchart of the implemented EKF-based GNSS-INS integration is shown in Figure 1. The state-space of the

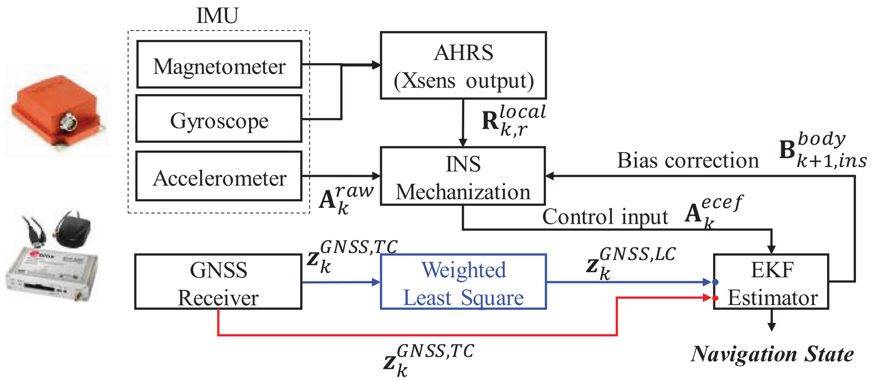


FIGURE 1 Flowchart of the loosely coupled (blue line) and tightly coupled (red line) GNSS-INS integrations implemented using the EKF [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

system ($\mathbf{x}_k$) is represented as

$$\mathbf{x}_k = \left(\mathbf{X}_{k,r}^{ecef}, \mathbf{V}_{k,r}^{ecef}, \mathbf{B}_{k,r}^{body}\right)^T, \tag{1}$$

where $\mathbf{X}_{k,\,r}^{ecef} = (x_{k,r}^{ecef}, y_{k,r}^{ecef}, z_{k,r}^{ecef})$ represents the position of the GNSS receiver in the ECEF (denoted by the subscript r) (Groves, 2013) at a given epoch k, $\mathbf{V}_{k,r}^{ecef} = (v_{k,r,x}^{ecef}, v_{k,r,y}^{ecef}, v_{k,r,z}^{ecef})$ denotes the velocity of the GNSS receiver, $\mathbf{B}_{k,ins}^{body} = (b_{k,x}^{body}, b_{k,y}^{body}, b_{k,z}^{body})$ denotes the bias of the accelerometer in the body (INS) frame, and $\mathbf{R}_{k,r}^{local}$ denotes the attitude in the local frame [the east, north, and up (ENU) frame] provided by the AHRS.

The inertial measurement unit (IMU) measurements are expressed as follows:

$$\mathbf{A}_k^{raw} = \left(a_{k,x}^{body}, a_{k,y}^{body} a_{k,z}^{body}\right)^T, \tag{2}$$

where $a_{k,x}^{body}, a_{k,y}^{body} a_{k,z}^{body}$ represent the acceleration measurements performed in the INS frame. As the estimated state $\mathbf{x}_k$ lies in the global frame (ECEF), the acceleration measurements must be transformed from the INS frame to the global frame, based on the attitude obtained from the AHRS. The transformed acceleration measurement in the global frame, $\mathbf{A}_k^{ecef} = (a_{k,x}^{ecef}, a_{k,y}^{ecef}, a_{k,z}^{ecef})^T$, can be expressed as follows (Groves, 2013):

$$\mathbf{A}_k^{ecef} = \mathbf{R}_{GL}\,\mathbf{R}_{LB}\left(\mathbf{A}_k^{raw} - \mathbf{B}_{k,ins}^{body}\right), \tag{3}$$

where $\mathbf{R}_{GL}$ is the transformation matrix used to transform the acceleration measurement from the local frame to the global frame based on $\mathbf{x}_k$ (see the Appendix for details). A generic dynamic model of EKF-based LC GNSS-INS integration can be written as

$$\mathbf{x}_k = f\,(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \tag{4}$$

where $f(\mathbf{x}_{k-1}, \mathbf{u}_k)$ denotes the state transition function, and can be expressed as

$$f\,(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} x_{k-1,r}^{ecef} + v_{k-1,r,x}^{ecef} \cdot \Delta t \\ y_{k-1,r}^{ecef} + v_{k-1,r,y}^{ecef} \cdot \Delta t \\ z_{k-1,r}^{ecef} + v_{k-1,r,\,z}^{ecef} \cdot \Delta t \\ v_{k-1,r,x}^{ecef} + a_{k-1,x}^{ecef} \cdot \Delta t \\ v_{k-1,r,y}^{ecef} + a_{k-1,y}^{ecef} \cdot \Delta t \\ v_{k-1,r,z}^{ecef} + a_{k-1,z}^{ecef} \cdot \Delta t \\ b_{k-1,x}^{body} \\ b_{k-1,y}^{body} \\ b_{k-1,z}^{body} \end{bmatrix}, \tag{5}$$

where Δt denotes the time difference between two epochs, and the function $f(\mathbf{x}_{k-1}, \mathbf{u}_k)$ is based on the constant-velocity model.

Besides, the measurement model of EKF-based LC GNSS-INS integration can be expressed as

$$\mathbf{z}_k^{GNSS,LC} = h^{GNSS,LC}\,(\mathbf{x}_k) + \mathbf{v}_k^{LC}, \tag{6}$$

where $\mathbf{z}_k^{GNSS,LC}$ are the position measurements in the ECEF frame estimated by the GNSS receiver, and can be expressed as

$$\mathbf{z}_k^{GNSS,LC} = \left(x_k^{GNSS}, y_k^{GNSS}, z_k^{GNSS}\right)^T. \tag{7}$$

Note that $\mathbf{z}_k^{GNSS,LC}$ is calculated using the weighted least squares method (Groves, 2013), based on the pseudorange code measurements obtained from the GNSS receiver. The observation function $h^{GNSS,LC}(*)$ shows the relationship between the observation and the state at the k^{th} epoch:

$$h^{GNSS,LC}\,(\mathbf{x}_k) = \begin{bmatrix} x_{k,r}^{ecef} \\ y_{k,r}^{ecef} \\ z_{k,r}^{ecef} \end{bmatrix}, \tag{8}$$

where $\mathbf{v}_k^{LC}$ is the Gaussian noise associated with the measurements, which is described with a covariance matrix equaling $\mathbf{R}_k$. To calculate $\mathbf{R}_k$, we follow the method used in Maier and Kleiner (2010), as follows:

$$\mathbf{R}_k = \left(h_{pdop} \cdot s_{UERE}\right)^2 \mathbf{I}_{3x3} = \begin{bmatrix} \left(h_{pdop} \cdot s_{UERE}\right)^2 & 0 & 0 \\ 0 & \left(h_{pdop} \cdot s_{UERE}\right)^2 & 0 \\ 0 & 0 & \left(h_{pdop} \cdot s_{UERE}\right)^2 \end{bmatrix}, \tag{9}$$

where s_{UERE} represents the user-equivalent range error (Maier & Kleiner, 2010), which we set as 10 m; h_{pdop} denotes the position dilution of precision (Groves, 2013), which is calculated based on the GNSS measurements; and $\mathbf{I}$ is a 3×3 identity matrix.

3.2 | TC GNSS-INS integration using the EKF

The main difference between LC and TC integrations lies in the domain of the GNSS observations that are applied. TC integration uses raw GNSS pseudorange, while LC uses position measurements. The state-space of the system ($\mathbf{x}_k$) is represented as

$$\mathbf{x}_k = \left(\mathbf{X}_{k,r}^{ecef}, \mathbf{V}_{k,r}^{ecef}, \mathbf{B}_{k,r}^{body}, \delta_{k,r}^{clock}\right)^T. \tag{10}$$

The state is similar to that used in LC integration, except that in TC integration the receiver clock bias $\delta_{k,r}^{clock}$ must also be estimated. The measurements obtained from the INS and dynamic models are identical to those obtained in the LC integration. The measurement model of EKF-based TC GNSS-INS integration can be expressed as

$$\mathbf{z}_k^{GNSS,TC} = h^{GNSS,TC}\left(\mathbf{x}_k\right) + \mathbf{v}_k^{TC} \tag{11}$$

where $\mathbf{z}_k^{GNSS,TC}$ indicates the GNSS pseudorange measurements (N satellites in total) performed in the ECEF frame using the GNSS receiver, and can be represented as

$$\mathbf{z}_k^{GNSS,TC} = \left(\rho_{k,1}^{GNSS}, \rho_{k,2}^{GNSS}, \ldots, \rho_{k,i}^{GNSS}, \ldots, \rho_{k,N}^{GNSS}\right)^T. \tag{12}$$

The variable $h^{GNSS,LC}(*)$ is an observation function that depicts the relationship between the observation and the state at the k-th time instant, $\rho_{k,i}^{GNSS}$ represents the ith pseudorange measurement at epoch k, N denotes the total number of satellites at epoch k, and $\mathbf{v}_k^{TC}$ is the Gaussian noise associated with the measurements. The position of a satellite $\mathbf{SV}_{k,i}$ is represented as $\mathbf{SV}_{k,i} = (x_{SV}^{ecef}, y_{SV}^{ecef}, z_{SV}^{ecef})^T$. Therefore, the predicted GNSS pseudorange measurement for satellite $\mathbf{SV}_{k,i}$ is obtained as

$$h^p\left(\mathbf{SV}_{k,i}, \mathbf{X}_{k,r}^{ecef}, \delta_{k,r}^{clock}\right) = \left\|\mathbf{SV}_{k,i} - \mathbf{X}_{k,r}^{ecef}\right\| + \delta_{k,r}^{clock}. \tag{13}$$

The ionospheric and tropospheric errors are modeled based on (Herrera et al., 2016), and removed from the pseudorange measurements before its integration with the INS. Therefore, the observation function $h^{GNSS,TC}(*)$ is formulated as follows:

$$h^{GNSS,TC}(*) = \begin{bmatrix} h^p\left(\mathbf{SV}_{k,1}, \mathbf{X}_{k,r}^{ecef}, \delta_{k,r}^{clock}\right) \\ h^p\left(\mathbf{SV}_{k,2}, \mathbf{X}_{k,r}^{ecef}, \delta_{k,r}^{clock}\right) \\ \ldots \\ h^p\left(\mathbf{SV}_{k,i}, \mathbf{X}_{k,r}^{ecef}, \delta_{k,r}^{clock}\right) \\ \ldots \\ h^p\left(\mathbf{SV}_{k,N}, \mathbf{X}_{k,r}^{ecef}, \delta_{k,r}^{clock}\right) \end{bmatrix}. \tag{14}$$

To determine the covariance matrix of $\mathbf{R}_k$ corresponding to the measurement vector $\mathbf{z}_k^{GNSS,TC}$, we follow the method proposed in Herrera et al. (2016). Here, each pseudorange measurement is expressed with a different uncertainty, based on its signal-to-noise ratio (SNR) and satellite elevation angle. Given a satellite with the SNR and elevation angle indicated as SNR_i and el_i, respectively, its weighting can be calculated as follows (Herrera et al., 2016):

$$W_i\left(el_i, SNR_i\right) = \frac{1}{sin^2 el_i} \times \left(10^{-\frac{(SNR_i - T)}{a}}\left(\left(\frac{A}{10^{-\frac{(F-T)}{a}}} - 1\right)\frac{(SNR_i - T)}{F - T} + 1\right)\right), \tag{15}$$

where T indicates the SNR threshold, and the parameters a, A, and F are selected based on (Herrera et al., 2016). Therefore, the covariance matrix $\mathbf{R}_k$ is a diagonal matrix constituted by the weighting ${\sigma_i}^2$:

$$\mathbf{R}_k = \begin{bmatrix} {\sigma_1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & {\sigma_N}^2 \end{bmatrix} \tag{16}$$

with $\sigma_i{}^2 = 1/W_i(el_i, SNR_i)$.

The subscript N indicates the number of satellites and the matrix $\mathbf{R}_k$ is an $N \times N$ matrix.

3.3 | LC GNSS-INS integration using FGO

In general, multisensor integration aims to determine the optimal posterior state, based on sensor measurements. Therefore, the sensor integration problem can be formulated as a typical MAP problem (Barfoot, 2017). In this study, the measurements are performed in two parts, namely GNSS measurements and INS measurements. Assuming that these measurements are independent of each other, GNSS-INS integration can be formulated as the following MAP problem:

$$\hat{\mathbf{x}} = \arg\max \prod_{k,i} P(\mathbf{z}_{k,i}|\mathbf{x}_k) \prod_{k} P(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k), \quad (17)$$

where $\mathbf{z}_{k,i}$ represents the GNSS raw measurements performed at epoch k, $\mathbf{x}_k$ represents the system state at k, i denotes the index of the measurements performed at k (e.g., one epoch may have multiple pseudorange measurements), $\mathbf{u}_k$ denotes the control input (e.g., INS measurements), and $\hat{\mathbf{x}}$ is the optimal system state set (Barfoot, 2017). The Bayes filter-based sensor fusion method recursively estimates the current epoch based on 1. the previous state, and 2. the control input and observation measurements at the current epoch. However, it fails to take full advantage of the historical information. Consequently, FGO-based sensor integration (Indelman et al., 2012) is applied to transform the MAP problem into a nonlinear optimization problem.

In FGO-based integration, all sensor measurements are treated as factors (ζ_j) associated with specific states ($\mathbf{x}_j$). According to (Indelman et al., 2012), the MAP problem can be expressed as

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}} \left(\prod_{j} \zeta_j\left(\mathbf{x}_j\right) \right) \quad (18)$$

$$\text{with } \zeta_j(\mathbf{x}_j) \propto \exp\left(-||h_j(\mathbf{x}_j) - \mathbf{z}_j||^2_{\Sigma_j}\right),$$

where $\zeta_j(\mathbf{x}_j)$ is a factor associated with a given measurement $\mathbf{z}_j$, which can be derived from both GNSS and INS measurements; $\mathbf{x}_j$ is the state associated with the given measurements $\mathbf{z}_j$; $h_j(*)$ is the observation function associated with $\mathbf{z}_j$; and the set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_k, \dots\}$ denotes the states that must be estimated. Assuming that all of the sensor noise fits a Gaussian distribution, the negative logarithm of $\zeta_j(\mathbf{x}_j)$ is proportional to the error function (Indelman et al., 2012) associated with the measurements. Therefore, Equation (18) can be transformed as follows:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \left(\sum_{j} \left|\left| h_j\left(\mathbf{x}_j\right) - \mathbf{z}_j \right|\right|^2_{\Sigma_j} \right). \quad (19)$$

Thus, FGO transforms Equation (17) into a standard nonlinear least-squares problem, expressed as Equation (19), and obtains the optimal state set $\mathbf{X}$ by minimizing the derived error function.

Figure 2 shows the graph structure of LC GNSS-INS integration using FGO, and the state space of the system is represented as Equation (1). The graph includes all historical observation measurements and shows a major difference between the conventional Kalman filter-based (Wan et al., 2018) and factor graph-based sensor integrations. The error functions of each listed factor are presented as follows.

3.3.1 | Motion model factor

We use a constant-velocity model to constrain the two consecutive states. Based on this model, the motion model can be expressed as

$$\mathbf{x}_k = h^{MM}\left(\mathbf{x}_{k-1}\right) + \mathrm{N}\left(0, \boldsymbol{\Sigma}^{MM}_k\right), \quad (20)$$

where $\mathbf{x}_k$ denotes the state at the given epoch k and $h^{MM}(*)$ represents the motion model function, and can be expressed as follows:

$$h^{MM}\left(\mathbf{x}_{k-1}\right) = \begin{bmatrix} x^{ecef}_{k-1,r} + v^{ecef}_{k-1,r,x} \cdot \Delta t \\ y^{ecef}_{k-1,r} + v^{ecef}_{k-1,r,y} \cdot \Delta t \\ z^{ecef}_{k-1,r} + v^{ecef}_{k-1,r,z} \cdot \Delta t \\ \mathbf{B}^{body\ T}_{k-1,r} \end{bmatrix}. \quad (21)$$

The variable $\boldsymbol{\Sigma}^{MM}_k$ is a covariance matrix associated with the motion model. This covariance matrix is constant and is formulated based on the specifications of the applied INS in this paper as follows:

$$\boldsymbol{\Sigma}^{MM}_k = \begin{bmatrix} 0.3^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01^2 \end{bmatrix}. \quad (22)$$

The units for the covariance matrix parts of $\mathbf{X}^{ecef}_{k,r}$ and $\mathbf{B}^{body}_{k,r}$ are m and m/s^2, respectively. Therefore, the

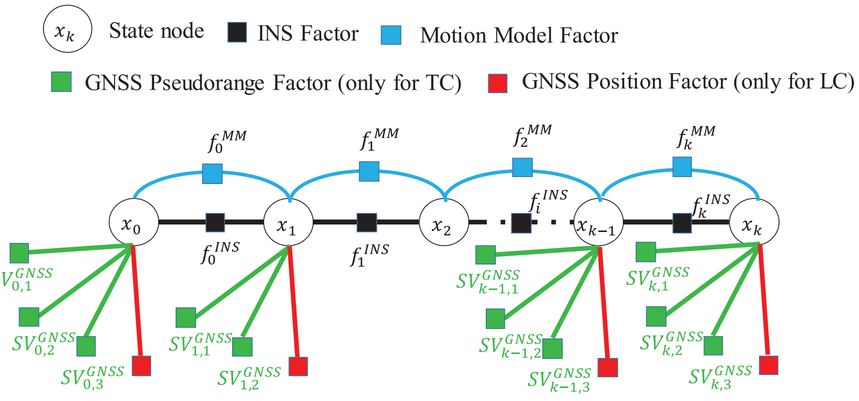


FIGURE 2 Illustration of the graph structure of the implemented loosely coupled and tightly coupled Global Navigation Satellite System–inertial navigation system integrations using factor graph optimization [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

error function ($\mathbf{e}_k^{MM}$) of the motion model factor can be expressed as

$$\left\|\mathbf{e}_k^{MM}\right\|^2_{\boldsymbol{\Sigma}_k^{MM}} = \left\|\mathbf{x}_k - h^{MM}\left(\mathbf{x}_{k-1}\right)\right\|^2_{\boldsymbol{\Sigma}_k^{MM}} \tag{23}$$

3.3.2 | INS factor

In LC FGO, an INS provides linear accelerations that directly correlate with velocities between two epochs. The acceleration measurements performed in the global frame are denoted as $\mathbf{A}_k^{ecef}$ ($\mathbf{u}_k$), based on Equation (3). The measurement model for linear acceleration is expressed as

$$\mathbf{x}_k = h^{INS}\left(\mathbf{x}_{k-1}, \mathbf{A}_k^{ecef}\right) + \mathrm{N}\left(0, \boldsymbol{\Sigma}_k^{INS}\right), \tag{24}$$

where the measurement function $h^{INS}(\mathbf{x}_{k-1}, \mathbf{u}_k)$ is expressed as

$$h^{INS}\left(\mathbf{x}_{k-1}, \mathbf{A}_k^{ecef}\right) = \begin{bmatrix} v_{k-1,r,x}^{ecef} + a_{k,x}^{ecef} \cdot \Delta t \\ v_{k-1,r,y}^{ecef} + a_{k,y}^{ecef} \cdot \Delta t \\ v_{k-1,r,z}^{ecef} + a_{k,z}^{ecef} \cdot \Delta t \end{bmatrix}. \tag{25}$$

Here, the covariance matrix for the INS factor is expressed as $\boldsymbol{\Sigma}_{k,acc}^{INS}$. Therefore, the error function for INS acceleration measurements can be formulated as

$$\left\|\mathbf{e}_k^{INS}\right\|^2_{\boldsymbol{\Sigma}_k^{INS}} = \left\|\mathbf{x}_k - h^{INS}\left(\mathbf{x}_{k-1}, \mathbf{A}_k^{ecef}\right)\right\|^2_{\boldsymbol{\Sigma}_k^{INS}}, \tag{26}$$

where $\boldsymbol{\Sigma}_k^{INS}$ is a constant based on the INS specifications applied in this paper, as is defined as

$$\boldsymbol{\Sigma}_k^{INS} = \begin{bmatrix} 0.15^2 & 0 & 0 \\ 0 & 0.15^2 & 0 \\ 0 & 0 & 0.15^2 \end{bmatrix}. \tag{27}$$

The unit for the covariance matrix is m/s.

3.3.3 | GNSS factor

The error function for a given GNSS measurement for LC FGO is obtained as follows:

$$\left\|\mathbf{e}_k^{GNSS}\right\|^2_{\boldsymbol{\Sigma}_k^{GNSS}} = \left\|\mathbf{z}_k^{GNSS} - h^{GNSS,LC}\left(\mathbf{X}_{k,r}^{ecef}\right)\right\|^2_{\boldsymbol{\Sigma}_k^{GNSS}} \tag{28}$$

where $\boldsymbol{\Sigma}_k^{GNSS}$ denotes the covariance matrix, which is calculated using Equation (9).

3.4 | TC GNSS-INS integration using FGO

Figure 2 shows the graph structure of TC GNSS-INS integration using FGO, and the state space of the system is represented as Equation (10). The motion and INS factors remain the same as those used for LC FGO.

3.4.1 | GNSS pseudorange factor

The GNSS receiver receives signals from multiple satellites at a given epoch k, which is expressed as

$$\mathbf{SV}_k = \left\{\mathbf{SV}_{k,1},\ \mathbf{SV}_{k,2}, \ldots, \mathbf{SV}_{k,i}, \ldots\ \mathbf{SV}_{k,N}\right\}. \tag{29}$$

Therefore, we obtain the error function for a given satellite measurement $\rho_{SV,i}$ as follows:

$$\left\|\mathbf{e}_{k,i}^{P}\right\|^2_{\boldsymbol{\Sigma}_{k,i}^{SV}} = \left\|\rho_{SV,i} - h^{GNSS,TC}\left(\mathbf{SV}_{k,i}, \mathbf{X}_{k,r}^{ecef}, \boldsymbol{\delta}_{k,r}^{clock}\right)\right\|^2_{\boldsymbol{\Sigma}_{k,i}^{SV}}, \tag{30}$$

where $\boldsymbol{\Sigma}_{k,i}^{SV}$ denotes the covariance matrix, which is calculated using Equation (16).

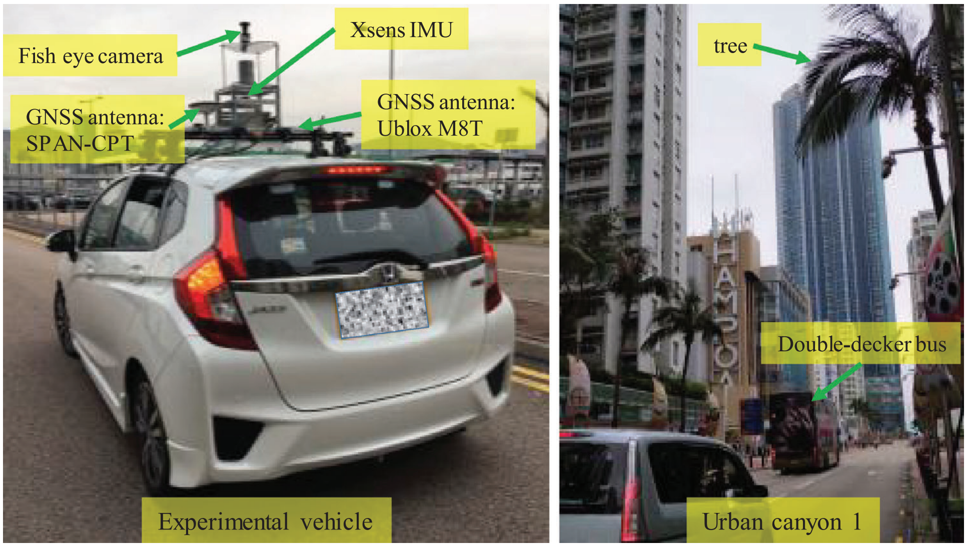


FIGURE 3 The left image shows the experimental vehicle and sensor setup, and the right image illustrates the tested urban canyon in Hong Kong [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

3.5 | LC GNSS-INS integration using FGO

We formulate three types of factors for use in LC GNSS-INS integration with FGO: motion model, INS, and GNSS factors. Therefore, the optimal state set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_k, \ldots\}$ is solved as follows:

$$\mathbf{X}^* = \operatorname{argmin} \sum_k \left\|\mathbf{e}_k^{GNSS}\right\|^2_{\Sigma_k^{GNSS}} + \left\|\mathbf{e}_k^{MM}\right\|^2_{\Sigma_k^{MM}} + \left\|\mathbf{e}_k^{INS}\right\|^2_{\Sigma_k^{INS}}. \tag{31}$$

We also formulate three types of factors for use in TC GNSS-INS integration with FGO: motion model, INS, and pseudorange factors. Therefore, the optimal state set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_k, \ldots\}$ is solved as follows:

$$X^* = \operatorname{argmin} \sum_{i,k} \left\|\mathbf{e}_{k,i}^{P}\right\|^2_{\Sigma_{k,i}^{SV}} + \left\|\mathbf{e}_k^{MM}\right\|^2_{\Sigma_k^{MM}} + \left\|\mathbf{e}_k^{INS}\right\|^2_{\Sigma_k^{INS}}. \tag{32}$$

During the optimization, the Levenberg-Marquardt method is used to solve Equations (31) and (32).

4 | EXPERIMENT EVALUATION

4.1 | Experiment setup

TABLE 1 Parameter values used in this paper

Parameter	F	T	A
Value	10	45	30
Parameter	s_{UERE} (m)	σ_{acc_n} $(\frac{rad}{s^2}\sqrt{Hz})$	σ_{acc_w} $(\frac{rad}{s^2}\sqrt{Hz})$
Value	10	8.17e-3	1.31e-4

To evaluate the performance of the four abovementioned integration schemes, we conducted experiments in an urban canyon in Hong Kong. The experimental vehicle and test scene are shown in Figure 3. The image on the left of Figure 3 shows the experimental vehicle, in which all sensors were installed in a compact sensor kit. The image on the right of Figure 3 shows the tested urban canyon in Hong Kong, which contained tall buildings, a challenging scenario for GNSS positioning.

During the test, a low-cost u-blox M8T GNSS receiver was used to collect raw single-frequency global positioning system (GPS) and BeiDou measurement data at a frequency of 1 Hz. The Xsens Ti-10 IMU was used to collect data at a frequency of 100 Hz, and a fisheye camera was used to capture a sky-view image to show the environmental conditions, for reference only. In addition, the NovAtel SPAN-CPT, which is a GNSS real-time kinematic-INS (fiber-optic gyroscope) integrated navigation system, was used to obtain the ground truth. The gyro-bias in-run stability of the FGO was 1°/h and its random walk was 0.067°/h. The baseline between the vehicle and the GNSS base-station was approximately 7 km. Besides, a positioning accuracy of approximately 10 cm was obtained using SPAN-CPT in the evaluated scene. All data were collected and synchronized using a robotic operating system (Quigley et al., 2009). Before conducting the experiments, the coordinate systems between all sensors were calibrated. We ran the EKF and FGO on a high-performance desktop computer equipped with an Intel i7-9700K at 4.20 GHz and 64GB RAM, in a postprocessing manner. The

TABLE 2 Positioning performance and computational load (time) of the four methods

All data	Loosely coupled EKF	Tightly coupled EKF	Loosely coupled FGO	Tightly coupled FGO
Mean error	9.14 m	8.03 m	7.01 m	3.64 m
Std	7.60 m	7.15 m	6.41 m	2.84 m
Used Time	0.053 s	0.071 s	15.41 s	75.30 s

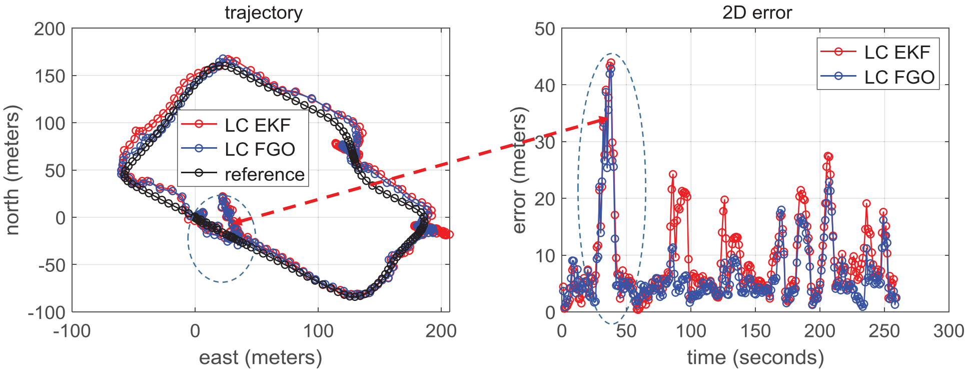


FIGURE 4 Trajectories of the LC GNSS/INS integrations using EKF and FGO in the east, north, and up (ENU) frame. The black curve denotes the reference trajectory. The red and blue curves in the left-hand side figure represent the trajectories from LC integrations using EKF and FGO, respectively. The right figure shows the 2D error [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

INS specifications that were applied are listed in Table 1. Because the choice of covariance parameters is known to substantially affect GNSS-INS accuracy, we used the same parameters for the EKF and FGO based on Equations (9) and (16), respectively. We focused on analyzing the differences between the EKF and FGO. The estimated state was in the ECEF (Groves, 2013). We transformed the positioning results obtained from ECEF into an ENU frame (Groves, 2013). As the orientation was directly obtained from the AHRS, only the two-dimensional (2D) positioning (north and east directions) accuracy was evaluated.

4.2 | Comparison of positioning accuracy

The positioning performances of the integrations conducted in the tested urban canyon are shown in Table 2, where 9.14 m of 2D mean error was obtained using LC EKF with a standard deviation of 7.60 m. The mean error decreased to 8.03 m when the TC EKF method was used. Moreover, the time efficiencies of LC and TC EKF were found to be similar, with less than 0.1 s required to process all data. After applying LC FGO for GNSS-INS integration, the 2D error decreased to 7.01 m with a standard deviation of 6.41 m, which was better than the value obtained for TC EKF. Moreover, the standard deviation decreased from 7.15 to 6.41 m. However, 15.41 s were required to process all data. Notably, the 2D mean error decreased to 3.64 m when TC FGO was used. The standard deviation also decreased dramatically, from 6.41 to 2.84 m, compared to when LC FGO was used. However, the computational time increased substantially (to 75.30 s), due to the increased number of factors in TC FGO compared to LC FGO. In summary, the optimal performance was obtained using TC FGO. As all historical data were considered in FGO, this led to a greater computational load.

The trajectories and positioning errors of the two LC and two TC integrations generated during the test are shown in Figures 4 and 5, respectively. The right side of Figure 4 shows that the mean error of LC FGO is slightly less in most epochs compared to that of LC EKF. However, the mean error is substantially less when TC FGO is used, as shown on the right side of Figure 5.

The residual term evaluates the difference between the measurements and final state estimation. If all applied measurements and their associated error model are correct, the residual should be zero. However, the residuals are usually nonzero, due to the noise generated by signal blockage or reflection, leading to NLOS reception in the GNSS pseudorange measurements. Both the EKF- and FGO-based methods aim to minimize the residuals of all considered measurements based on the associated covariance matrix. Smaller residuals usually indicate that the estimated state is closer to its optimal estimation. Thus, the value of a residual is a good indicator of the quality

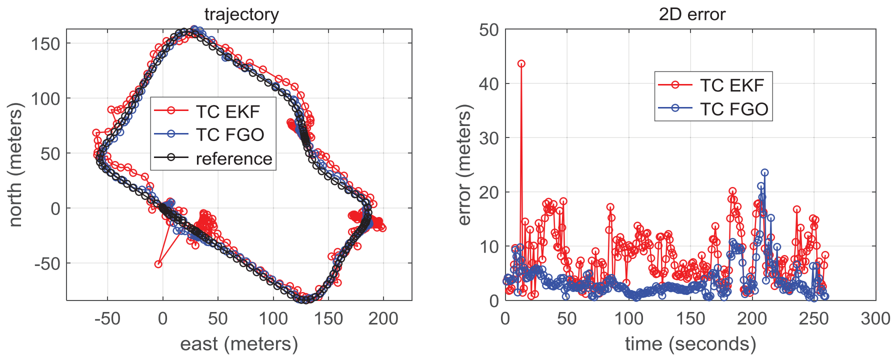


FIGURE 5 Trajectories of TC GNSS/INS integrations using EKF and FGO in the east, north, and up (ENU) frame. The black curve denotes the reference trajectory. The red and blue curves in the left-hand side figure represent the trajectories from TC integrations using EKF and FGO, respectively. The right figure shows the 2D error [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

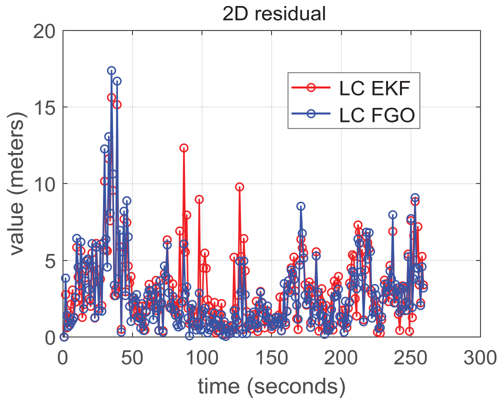


FIGURE 6 2D residuals of the loosely coupled GNSS/INS integrations using EKF and FGO [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

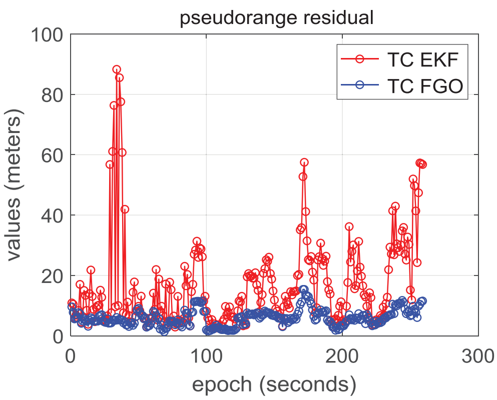


FIGURE 7 2D residuals of the tightly coupled GNSS/INS integrations using EKF and FGO [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

of GNSS-INS integration. Therefore, we also present the residual results of the four integrations used in this study.

The residuals of the LC and TC integrations are shown in Figures 6 and 7, respectively. In LC integration, the observation is the measurement of GNSS positioning solutions. We calculate the L2 norm of the residual ($p_{r,LC}$), which denotes the difference between the GNSS positioning measurements and the final GNSS-INS integration result, as follows:

$$p_{r,LC} = \left\| \left(\mathbf{z}_k^{GNSS,LC} \right)_{xy} - \left(h^{GNSS,LC} \left(\mathbf{x}_k{}^* \right) \right)_{xy} \right\|_2 \qquad (33)$$

where $p_{r,LC}$ denotes the GNSS residual in LC integration and the operator $()_{xy}$ is used to obtain the 2D part (horizontal positioning). Thus, we evaluate 2D residuals for LC integration. Therefore, the 2D positioning error (right side of Figure 4) and residual (Figure 6) are the same metrics and may be compared, where larger residuals usually indicate a larger potential positioning error. The variable $\mathbf{x}_k{}^*$ denotes the estimated state at a given epoch k. Figure 6 shows that, throughout the test, the FGO-based method has a similar or smaller residual than the EKF-based integration.

Figure 7 shows the residual of the pseudorange measurements. To simplify the comparison of the pseudorange residual and mean positioning error of GNSS-INS integration, we calculate the mean of the L1 norm of the pseudorange residual in TC integration, as follows:

$$\rho_{r,TC} = 1/N \sum_{i=1}^{N} \left\| \left(\rho_{k,i}^{GNSS} - h^{GNSS,TC} \left(\mathbf{SV}_{k,i}, \mathbf{X}_{k,r}^{ecef}, \delta_{k,r}^{clock} \right) \right) \right\|_1 \qquad (34)$$

where $\rho_{r,TC}$ denotes the GNSS residual in TC integration at a given epoch k. As shown in the figure, the FGO-based method yields a much smaller residual. Note that the residual for TC integration lies in the GNSS pseudorange domain. A small residual indicates that the final positioning result is closer to the given measurement. The

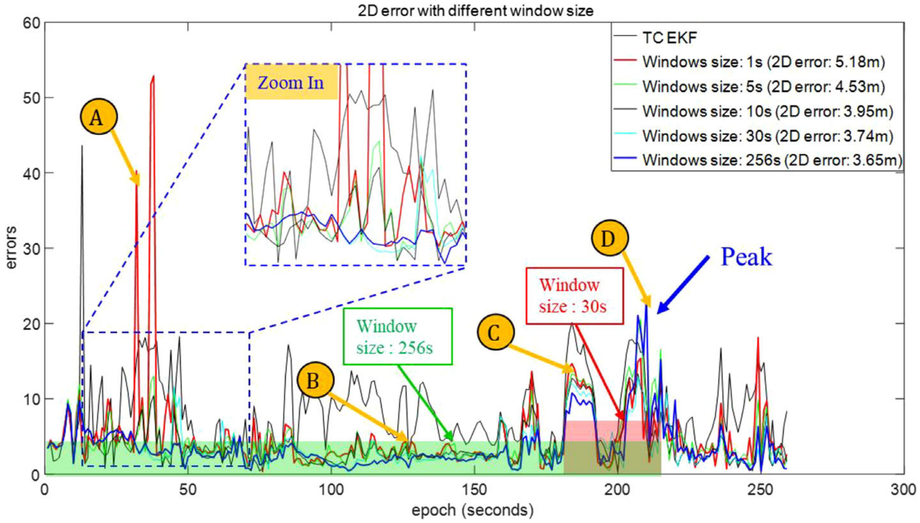


FIGURE 8 2D positioning errors under different window sizes used in TC GNSS/INS integrations using FGO. The x-axis denotes the epochs, and the y-axis represents the value of 2D positioning errors. The red and green rectangles denote the sliding windows of sizes 30 and 256 s, respectively [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

figure also shows that the TC FGO residual is substantially smoother than the TC EKF residual. In sum, FGO-based GNSS-INS integration exhibits better accuracy than EKF-based integration. Moreover, FGO-based TC integration exhibits the best performance of the four listed integrations.

4.3 | Analysis of improvement in FGO vs. window size

According to (Dellaert & Kaess, 2017), a major difference between the EKF and FGO approaches is the number of "iterations" applied. The EKF estimator only iterates once based on the given observation measurements, whereas FGO involves several iterations, which consider all historical and current measurements simultaneously, to approach the optimal state. If the window size of the optimization is set to 1 s [$K = 1$ in (32)], which means that FGO-based TC integration only considers the measurements at the current and last epochs, the major difference between the FGO and EKF approaches lies in the number of iterations applied during GNSS-INS integration. The window size denotes the epoch of historical states considered in FGO. We refer to the FGO with a window size of 1 s as the "EKF-like estimator." Note that this estimator differs from the EKF, which recursively maintains the historical information present in the covariance matrix. Figure 8 shows the 2D positioning error of TC GNSS-INS integration obtained using FGO under different window sizes. The 2D mean positioning error is 5.18 m (black curve in Figure 8) with a window size of 1 s, which is better than that obtained from EKF-based TC integration (8.03 m). This reduction in the mean positioning error is largely due to the number of iterations used in the "EKF-like estimator" compared to the number used in the EKF, which is a key difference between these two approaches. The other key difference between the EKF and FGO approaches is that more historical data are considered simultaneously during FGO. To analyze the effects of window size on the performance of FGO-based GNSS-INS integration, the positioning results obtained using different window sizes are also shown in Figure 8. Note that a window size of 256 s equals the batch optimization, which considers all historical information in FGO. With a window size of 1 s, there is a limited improvement in the FGO. With the increased window sizes (5, 10, 30, and 256 s), more historical information is considered during the optimization, and the overall FGO performance gradually improves. Besides, the error curve (blue) arising from batch optimization (window size of 256 s) is substantially smoother than the other curves. Overall, it can be seen that the use of more historical data tends to increase resilience to outliers in GNSS measurements, such as NLOS receptions and multipath effects. We find that when the window size is approximately 30 s, the accuracy (3.74 m) is close to that (3.65 m) obtained for batch optimization (window size is 256 s).

The improvements obtained from FGO-based GNSS-INS integration can be summarized as follows.

1. **Multiple iterations**: As FGO is a process of determining the optimal estimation based on the gradient (Dellaert & Kaess, 2017), multiple iterations help to estimate the optimal state. Thus, these iterations relax the requirement for the accuracy of the initial guess of the estimators. In addition, linearization is conducted

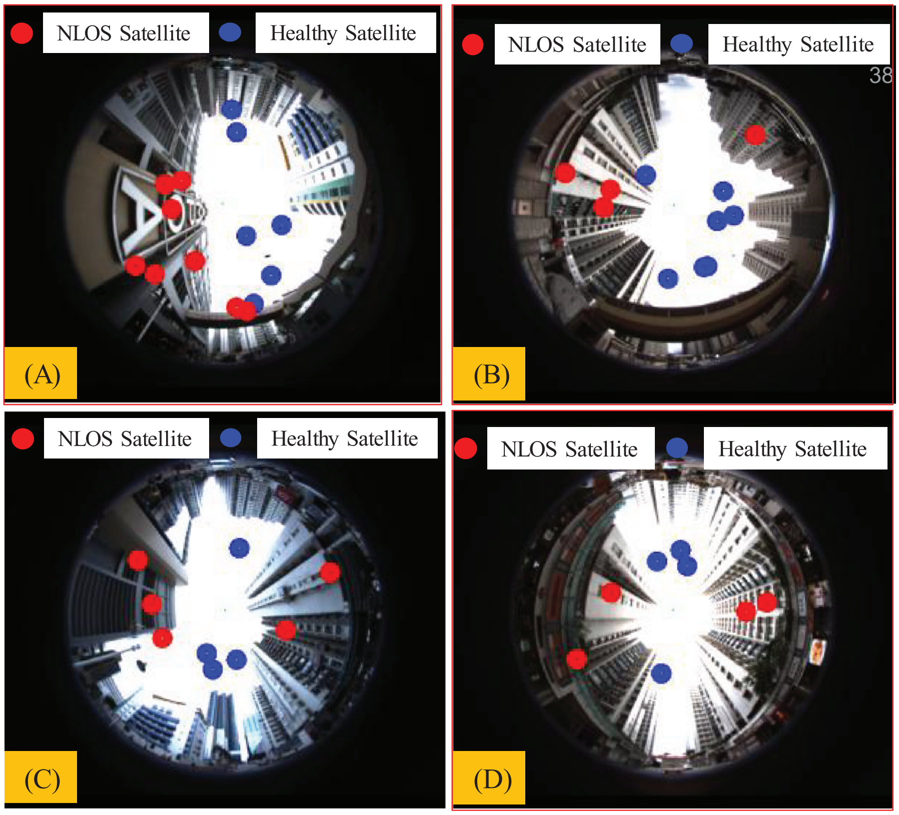


FIGURE 9 Sky-view and satellite visibilities of the four selected epochs in Figure 8. The blue and red circles denote the line-of-sight (LOS) and non-line-of-sight (NLOS) satellites, respectively [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

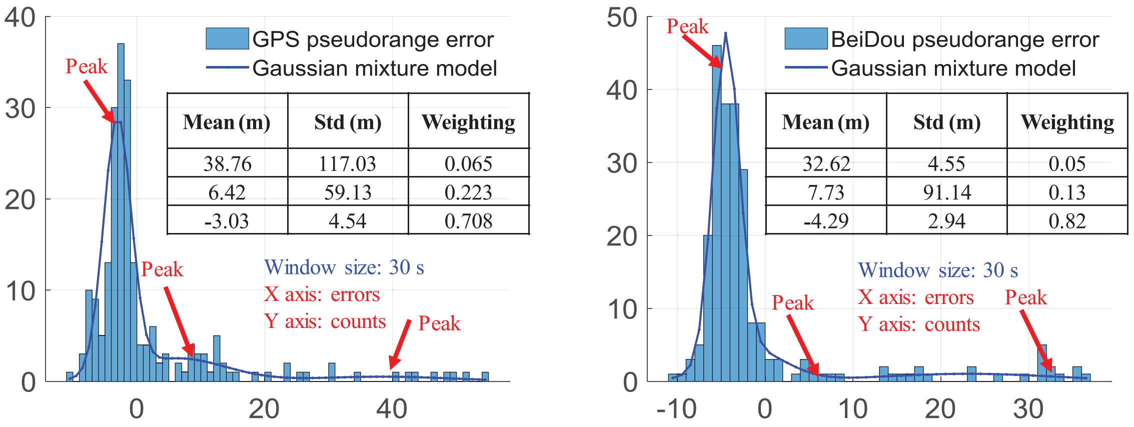
in each FGO iteration, which helps to linearize the nonlinear observation model. However, the nonlinearity of the model is trivial for TC GNSS-INS integration: The line-of-sight (LOS) connecting the GNSS receiver and satellite is accurate, even when the initial guess has an error of 100 m. This has also been mentioned in (Zhao et al, 2014).

2. **Time correlation**: FGO effectively considers historical information, all of which are connected by the INS factor. Thus, the time correlation between the historical epochs is simultaneously explored and used to resist the outliers. Thus, an appropriate FGO window size improves the performance. The same finding has also been discussed in (Zhao et al., 2016).

4.4 | Case study on the relationship between the measurement uncertainty and the selection of window size for FGO

As shown in Figure 8, the error (blue curve) of batch optimization at epoch D is larger than those resulting from smaller window sizes (e.g., 20 s, 10 s, and 5 s). This means that in some epochs a larger window size does not necessarily lead to better FGO performance. Figure 9 shows the sky-view images captured by a fisheye camera and the satellite visibilities for the four selected epochs (A, B, C, and D) in Figure 8. The blue and red circles denote the LOS and NLOS satellites, respectively, which are classified according to our previous study (Wen, Bai, et al., 2019). The errors peak at epochs A, C, and D due to the severe NLOS receptions (see red circles). The period near epoch B introduces a similar positioning error (3–6 m) under different window sizes. The actual sensor noise (error magnitude) model near epoch B is substantially different from that near epoch D. Thus, the historical measurements fail to reflect the measurement noise at the current epoch D. A larger window size (see blue curve in Figure 8) in FGO leads to a worse result at epoch D.

If the error model and relative weightings [corresponding to Equations (9), (15), and (16)] of the pseudorange measurements are provided, the use of more historical data should improve the accuracy of GNSS-INS integration. However, if the error modeling is incorrect, the time correlation will not be accurate, and a larger window size will lead to only limited improvements or even worse accuracy (epoch D in Figure 8). Thus, the limited improvements obtained under the increased window size



Mean (m)	Std (m)	Weighting
38.76	117.03	0.065
6.42	59.13	0.223
-3.03	4.54	0.708

Mean (m)	Std (m)	Weighting
32.62	4.55	0.05
7.73	91.14	0.13
-4.29	2.94	0.82

FIGURE 10 Histogram of GPS (left) and BeiDou (right) pseudorange errors and the corresponding fitted Gaussian mixture model (GMM) inside a sliding window of epoch D with a size of 30 s (as indicated by the red rectangle in Figure 8). The *x*-axis denotes the value of pseudorange errors acquired by the ray-tracing technique. The *y*-axis denotes the counts of errors within the histogram. The blue curve represents the GMM fitted by the histogram using three Gaussian components [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

(from 30 to 256 s) are partially caused by the imperfect weightings (modeling of noise covariance). Both EKF- and FGO-based GNSS-INS integrations rely on the assumption that the sensor noise can be fitted by a Gaussian model. However, this assumption is usually violated due to the satellite signal reflection and blockage caused by buildings. This is a major factor limiting the performance of GNSS-INS integration in urban canyons. Instead of using Gaussian distribution, a team from the Chemnitz University of Technology (Pfeifer & Protzel, 2019a) recently has proposed the use of GMM for modeling the pseudorange measurement noise. They found that if the residuals of all measurements performed inside a window effectively model the error distribution of the measurements at the current epoch, FGO can be substantially improved based on the GMM estimated using the historical pseudorange residuals. The authors (Pfeifer & Protzel, 2019a) argued that the noise of pseudorange measurements in urban canyons does not have a Gaussian distribution. Inspired by their work, we show the pseudorange error and residual distributions near epoch D under different window sizes.

Figure 10 shows the measured GPS/BeiDou pseudorange errors, which are labeled based on the ground-truth trajectory and three-dimensional building modeling using the double-difference technique (Xu et al., 2019). Thus, the figure shows the true error distribution of the pseudorange measurement. Note that the pseudorange errors are within a 30-s window of epoch D, as indicated by the red rectangle in Figure 8. According to our fitting analysis, three Gaussian components are sufficient to fit the histograms of Figure 10 with three major peaks. This agrees with the finding presented in Pfeifer & Protzel (2018), in which three Gaussian components were usually enough to model the pseudorange error distribution in urban canyons. The histogram exhibits a long tail on the right side, which is largely due to NLOS receptions caused by building reflections. This is one of the main reasons for the non-Gaussian property of pseudorange measurements. A similar long-tailed phenomenon is also witnessed in the BeiDou pseudorange errors on the right side of Figure 10. We use a GMM with three Gaussian components, which is shown by blue curves in the figure, to quantitatively fit the GPS/BeiDou error distributions. The mean, standard deviation, and weighting of each Gaussian component are also shown in the figure. The first component of the GMM, for GPS pseudorange measurements with a mean of 38.76 m, represents the NLOS signals, which generate the long-tail phenomenon. The first component of GMM, for BeiDou pseudorange measurements with a mean of 32.62 m, also represents the NLOS signals. In sum, the numerous NLOS receptions lead to a pronounced long-tail phenomenon in the error distribution.

Similar to Figure 10, Figure 11 also shows the residuals of GPS/BeiDou pseudorange measurements, which are calculated as follows:

$$\rho_{r,TC} = \left(\rho_{k,i}^{GNSS} - h^{GNSS,TC}\left(\mathbf{SV}_{k,i}, \mathbf{X}_{k,r}^{ecef}, \boldsymbol{\delta}_{k,r}^{clock}\right)\right), \tag{35}$$

where the actual values (positive and negative) are used. Similar histograms and GMM parameters (see the table inside Figure 11) may be obtained using the pseudorange residuals. This means that the fitted GMM in Figure 11 comprising the pseudorange residuals inside a 30 s sliding window effectively represents the actual error distribution (the GMM shown in Figure 10). As Figure 8 shows, a window size of 30 s leads to similar accuracy to that obtained from the batch optimization. This means that the historical measurements performed within the 30-s win-

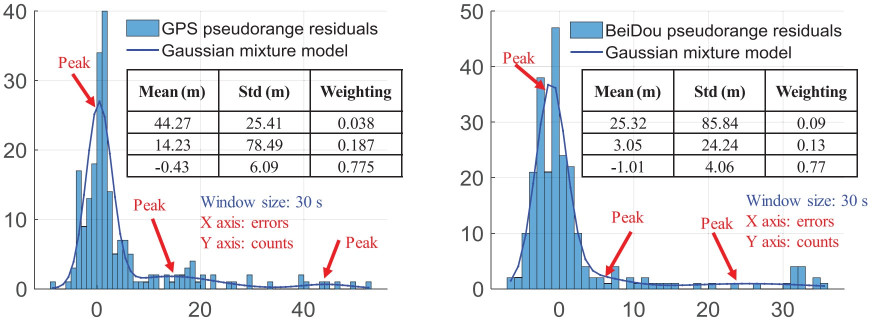


FIGURE 11 Histogram and Gaussian mixture models (GMMs) of pseudorange residuals near epoch D with a sliding window of 30 s. These are similar to Figure 10, with the major difference being that the histogram and GMMs are based on the residuals [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

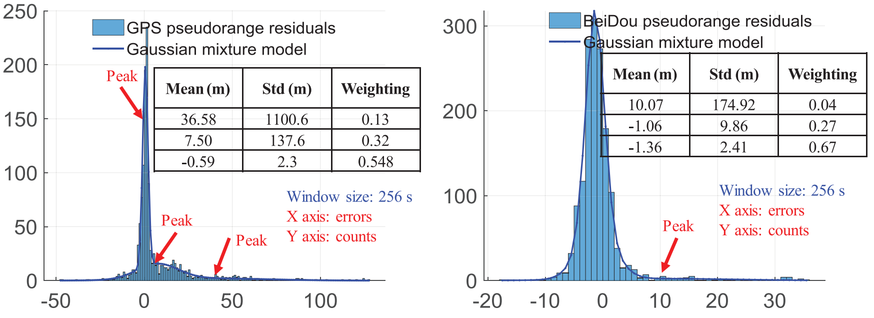


FIGURE 12 Histogram and Gaussian mixture models (GMMs) of pseudorange residuals similar to Figure 11, with the major difference being that the histogram and GMMs are based on the residuals near epoch D with a sliding window of 256 s [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

dow describe the error distribution of the measurement noise near epoch D. Figure 12 shows the distributions of the residuals with substantially larger window size (256 s) than that in Figure 10. The left figure shows that the GPS residuals introduce a long-tail phenomenon, with the maximum pseudorange residual reaching 100 m, which is substantially different from that observed in Figure 11. Thus, the Gaussian component with a mean of 36.58 m introduces a large standard deviation of 1,100.6 m. The mean values (10.07 m) of the GMM for the BeiDou pseudorange measurements are substantially less than the 32.62 m value shown on the right side of Figure 10. In sum, the residuals obtained within the window size of 256 s at epoch D cannot effectively describe the actual noise at this epoch (Figure 10). Thus, this is a significant factor that defines the optimal window size in FGO.

4.5 | Discussion on the computational load of FGO

Another problem faced in FGO applications is the increased computational load compared to that generated by the use of the EKF estimator. To maintain real-time performance, a sliding window may be used, with only the measurements within this sliding window being considered for FGO. However, this technique does not use all of the historical data. To address this issue, the incremental smoothing and mapping (iSAM) algorithm (Kaess et al., 2008) has been proposed and is built on top of the factor graph. A key advantage of iSAM is that it guarantees a low computational load even when large amounts of data are used in the optimization. This is achieved by storing the connections between the historical data in a novel Bayes tree structure. This means that iSAM does not need to recalculate the Jacobians for the states that are unaffected by the newly performed measurements. Figure 13 shows a time comparison of the two TC GNSS-INS integrations solved by FGO and by iSAM. The computational time

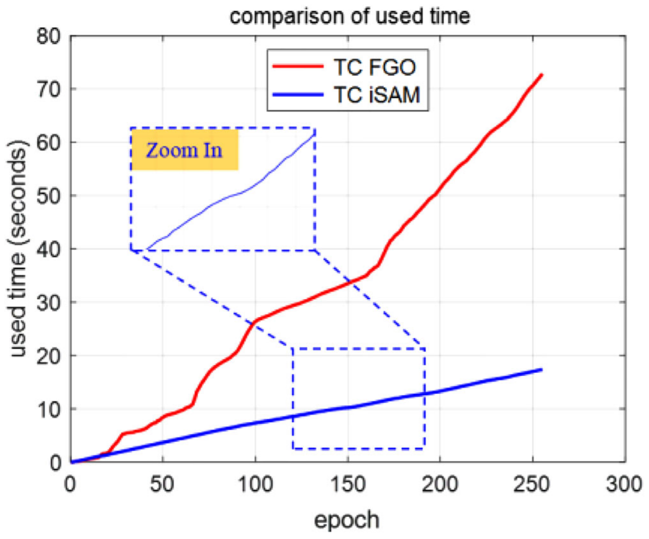


FIGURE 13 Comparison of the computational time used in tightly coupled (TC) integration using factor graph optimization (FGO) and the TC incremental smoothing and mapping algorithm (iSAM) (Kaess et al., 2008) [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com and www.ion.org]

used by FGO increases dramatically when more historical data are used, with 75.30 s required to process all of the data. However, iSAM has a substantially lower computational load when incorporating more historical data, with only 15.41 s required to process all the data. Meanwhile, the slope of the curve shown in Figure 13 is almost constant during the GNSS-INS integration. Thus, although the amount of historical data increases, the time used to process each epoch remains almost constant. Moreover, the positioning performances of both methods are almost the same. Thus, we believe that iSAM is a suitable alternative for solving FGO-based sensor fusion in real-time.

5 | CONCLUSIONS AND FUTURE WORK

This paper comprehensively compares the performance of four GNSS-INS integrations using both the EKF and FGO, using a real dataset collected in urban canyons. The experimental results show that TC GNSS-INS integration performs better than LC GNSS-INS integration. Besides, TC integration using FGO performs the best of the four integrations. According to the analysis results, the superior performance of FGO compared to that of EKF is attributable to the 1. multiple iterations, and 2. greater amount of data applied in the FGO. Therefore, we believe that FGO-based sensor fusion may be a promising replacement for EKF-based methods in the coming decades.

Besides, an analysis of the effect of window size on FGO performance shows that appropriate window size is highly correlated with environmental conditions. Recently, context awareness (Gao & Groves, 2020) has been used to identify the potential sensor noise generated in GNSS positioning. The use of environmental context-awareness to identify the period during which sensor noise characteristics are similar or constant is a promising way to adaptively tune the window size. For example, by limiting the window size to align with the time over which the error models are roughly constant or similar, correct relative weightings are preserved, which likely contribute to the better performance obtained for adaptive window size.

As this paper analyzed only one dataset collected in a typical urban canyon of Hong Kong, it will be interesting and necessary to determine how FGO will handle multiple datasets from diverse urban scenarios. In the future, we will examine FGO performance using more sensors (e.g., LiDAR) with our recently published UrbanLoco dataset (Wen et al., 2020), which comprises a complete set of vehicular navigation sensor data collected in both Hong Kong and downtown San Francisco.

ORCID

Weisong Wen https://orcid.org/0000-0003-4158-0913
Tim Pfeifer https://orcid.org/0000-0001-8974-2278
Li-Ta Hsu https://orcid.org/0000-0002-0352-741X

REFERENCES

Angrisano, A. (2010). *GNSS/INS integration methods*. (Dottorato di ricerca (PhD) in Scienze Geodetiche e Topografiche Thesis). Universita'degli Studi di Napoli PARTHENOPE, Naples.

Barfoot, T. D. (2017). *State Estimation for Robotics*: Cambridge University Press.

Bell, B. M & Cathey, F.W. (1993). The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, *38*(2), 294–297. https://doi.org/10.1109/9.250476

Bhamidipati, S. & Gao, Grace Xingxin. (2018). Multiple GPS fault detection and isolation using a graph-SLAM framework. *Proc. of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*, Miami, FL. https://doi.org/10.33012/2018.16030.

Chen, D. & Gao, G. X. (2019). Probabilistic graphical fusion of LiDAR, GPS, and 3D building maps for urban UAV navigation. *NAVIGATION*, *66*, 151–168. https://doi.org/10.1002/navi.298

Crassidis, J. L., Markley, F. L., & Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, *30*(1), 12–28. https://doi.org/10.2514/1.22452

Dellaert, F. & Kaess, M. (2017). Factor graphs for robot perception. *Foundations and Trends® in Robotics*, *6*(1-2), 1–139. https://www.nowpublishers.com/article/Details/ROB-043

Deng, Z. J., Liu, Y. W., Liu, J. X., Zhou, X, & Ci, S. (2017). QoE-oriented rate allocation for multipath high-definition video streaming over heterogeneous wireless access networks. *Ieee Systems Journal*, *11*(4), 2524–2535. https://doi.org/10.1109/JSYST.2015.2430893

Falco, G., Pini, M, & Marucco, G. (2017). Loose and tight GNSS/INS integrations: Comparison of performance assessed in real urban scenarios. *Sensors*, *17*(2), 255. https://doi.org/10.3390/s17020255

Gao, G. & Lachapelle, G. (2008). A novel architecture for ultra-tight HSGPS-INS integration. *Positioning*, *1*(13). https://www.scirp.org/html/381.html

Gao, H. & Groves, P. D. (2020). Context detection for advanced self-aware navigation using smartphone sensors. *arXiv preprint arXiv:2005.07539*.

Groves, P. D. (2013). Multipath vs. NLOS signals. *Inside GNSS 8*(6), 40–42. https://insidegnss.com/multipath-vs-nlos-signals/

Groves, P. D. (2013). *Principles of GNSS, inertial, and multisensor integrated navigation systems (2nd ed.)*. Artech.

Groves, P. D. (2015). Principles of GNSS, inertial, and multisensor integrated navigation systems (2nd ed.) [Book review]. *IEEE Aerospace and Electronic Systems Magazine*, *30*(2), 26–27. https://doi.org/10.1109/MAES.2014.14110

Herrera, A. M, Suhandri, H.F., Realini, E., Reguzzoni, M., & de Lacy, M.C. (2016). goGPS: open-source MATLAB software. *Gps Solutions*, *20*(3), 595–603. https://geodesy.noaa.gov/gps-toolbox/goGPS.htm

Hsu, L. T., Jan, S. S., Groves, P. D, & Kubo, N. (2015). Multipath mitigation and NLOS detection using vector tracking in urban environments. *GPS Solutions*, *19*(2), 249–262. https://doi.org/10.1007/s10291-014-0384-6

Huang, J. (2016). A study on the use of graph signal processing techniques for satellite-based navigation systems. *Proc. of the 2016 International Technical Meeting of the Institute of Navigation*, Monterey, CA. https://doi.org/10.33012/2016.13444

Indelman, V., Williams, S., Kaess, M., & Dellaert, F. (2012). Factor graph based incremental smoothing in inertial navigation systems. *2012 15th International Conference on Information Fusion*, Singapore, 2154–2161. https://ieeexplore.ieee.org/document/6290565

Indelman, V., Williams, S., Kaess, M., & Dellaert, F. (2013). Information fusion in navigation systems via factor graph based incremental smoothing. *Robotics and Autonomous Systems*, *61*(8), 721–738. https://doi.org/10.1016/j.robot.2013.05.001

Julier, S. J. & Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI, (28 July 1997)*, Orlando, FL. https://doi.org/10.1117/12.280797

Kaess, M., Ranganathan, A., & Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, *24*(6), 1365–1378. https://doi.org/10.1109/TRO.2008.2006706

Li, M. & Mourikis, A. I. (2013). Optimization-based estimator design for vision-aided inertial navigation. *Robotics: Science and Systems*. http://roboticsproceedings.org/rss08/p31.pdf

Li, W., Cui, X., & Lu, M. (2018). A robust graph optimization realization of tightly coupled GNSS/INS integrated navigation system for urban vehicles. *Tsinghua Science and Technology*, *23*(6), 724–732. https://doi.org/10.26599/TST.2018.9010078

Litman, T. (2015). Autonomous vehicle implementation predictions: Implications for transport planning. *TRB 94th Annual Meeting Compendium of Papers,* Washington, DC. https://trid.trb.org/view.aspx?id=1338043

Liu, H., Nassar, S., & El-Sheimy, N. (2010). Two-filter smoothing for accurate INS/GPS land-vehicle navigation in urban centers. *Ieee Transactions on Vehicular Technology*, *59*(9), 4256–4267. https://doi.org/10.1109/TVT.2010.2070850

Maier, D., & Kleiner, A. (2010). Improved GPS sensor model for mobile robots in urban terrain. *2010 IEEE International Conference on Robotics and Automation,* Anchorage, AK, 4385-4390. https://doi.org/10.1109/ROBOT.2010.5509895

Petovello, M. G. (2003). *Real-time integration of a tactical-grade IMU and GPS for high-accuracy positioning and navigation* (Unpublished doctoral thesis). University of Calgary, Calgary, AB. https://doi.org/10.11575/PRISM/23031

Pfeifer, T., & Protzel, P. (2018). Robust sensor fusion with self-tuning mixture models. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Madrid, Spain, 3678-3685. https://doi.org/10.1109/IROS.2018.8594459

Pfeifer, T., & Protzel, P. (2019a). Expectation-maximization for adaptive mixture models in graph optimization. *2019 International Conference on Robotics and Automation (ICRA),* Montreal, QC, Canada. https://doi.org/10.1109/ICRA.2019.8793601

Pfeifer, T., & Protzel, P. (2019b). Incrementally learned mixture models for GNSS localization. *2019 IEEE Intelligent Vehicles Symposium (IV)*, Paris, France, 1131-1138. https://doi.org/10.1109/IVS.2019.8813847

Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, *34*(4), 1004–1020. https://doi.org/10.1109/TRO.2018.2853729

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: an open-source Robot Operating System. *ICRA Workshop on open source software*. http://robotics.stanford.edu/~ang/papers/icraoss09-ROS.pdf

Roysdon, P. F., & Farrell, J. A. (2017). GPS-INS outlier detection & elimination using a sliding window filter. *2017 American Control Conference (ACC)*, Seattle, WA, 1244-1249. https://doi.org/10.23919/ACC.2017.7963123

Saripalli, S., Montgomery, J. F., & Sukhatme, G. S. (2003). Visually guided landing of an unmanned aerial vehicle. *IEEE transactions on robotics and automation*, *19*(3), 371–380. https://doi.org/10.1109/TRA.2003.810239

Solimeno, A. (2007). *Low-cost INS/GPS data fusion with extended Kalman filter for airborne applications*. (Masters of Science thesis, Universidade Technica de Lisboa, Lisbon, Portugal). https://www.semanticscholar.org/paper/Low-Cost-INS%2FGPS-Data-Fusion-with-Extended-Kalman-Solimeno/1dd8f11cb300b9dd134e0de1bb46eefb4b12f075

Thrun, S. (2000). Probabilistic algorithms in robotics. *AI Magazine*, *21*(4), 93. https://doi.org/10.1609/aimag.v21i4.1534

Valiente, D., Gil, A., Fernández, L., & Reinoso, Ó. (2014). A comparison of EKF and SGD applied to a view-based SLAM approach with omnidirectional images. *Robotics and Autonomous Systems*, *62*(2), 108–119. https://doi.org/10.1016/j.robot.2013.11.009

Wan, E. A. & Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. *Proc. of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, Lake Louise, AB, Canada, 153-158. https://doi.org/10.1109/ASSPCC.2000.882463

Wan, G., Yang, X., Cai, R., Li, H., Zhou, Y., Wang, H., & Song, S. (2018). Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia, 4670-4677. https://doi.org/10.1109/ICRA.2018.8461224

Watson, R. M., & Gross, J. N. (2018). Evaluation of kinematic precise point positioning convergence with an incremental graph optimizer. *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Monterey, CA. https://arxiv.org/pdf/1804.04197.pdf

Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter. https://perso.crans.org/club-krobot/doc/kalman.pdf

Wen, W., Bai, X., Kan, Y.-C., & Hsu, L.-T. (2019). Tightly coupled GNSS/INS integration via factor graph and aided by fish-eye camera. *IEEE Transactions on Vehicular Technology*. *68*(11):10651–10652. https://doi.org/10.1109/TVT.2019.2944680

Wen, W., Hsu, L.-T., & Zhang, G. (2018). Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong. *Sensors*, *18*(11), 3928. https://doi.org/10.3390/s18113928

Wen, W., Kan, Y. C., & Hsu, L.-T. (2019). Performance comparison of GNSS/INS integrations based on EKF and factor graph optimization. *Proc. of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019)*, Miami, FL. https://doi.org/10.33012/2019.17129

Wen, W., Zhou, Y., Zhang, G., Fahandezh-Saadi, S., Bai, X., Zhan, W., Tomizuka, M., & Hsu, L.-T. (2020). Urbanloco: a full sensor suite dataset for mapping and localization in urban scenes. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2310-2316. https://doi.org/10.1109/ICRA40945.2020.9196526

Xu, B., Jia, Q., Luo, Y., & Hsu, L.-T. (2019). Intelligent GPS L1 LOS/multipath/NLOS classifiers based on correlator-, RINEX-and

NMEA-level measurements. *Remote Sensing*, *11*(16), 1851. https://doi.org/10.3390/rs11161851

Zhao, S., Chen, Y., & Farrell, J. A. (2016). High-precision vehicle navigation in urban environments using an MEM's IMU and single-frequency GPS receiver. *IEEE Transactions on Intelligent Transportation Systems*, *17*(10), 2854–2867. https://doi.org/10.1109/TITS.2016.2529000

Zhao, S., Chen, Y., Zhang, H., & Farrell, J. A. (2014). Differential GPS aided inertial navigation: A contemplative realtime approach. *IFAC Proceedings Volumes*, *47*(3), 8959–8964. https://folk.ntnu.no/skoge/prost/proceedings/ifac2014/media/files/1658.pdf

Zheng Gong, P. L., Qiang Liu, Ruihang Miao, and Rendong Ying. (2018). Tightly coupled GNSS with stereo camera navigation using graph optimization. *Proc. of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2018)*, Miami, FL, 3070 - 3077. https://doi.org/10.33012/2018.15959

Zhuang, Y., & El-Sheimy, N. (2015). Tightly-coupled integration of WiFi and MEMS sensors on handheld devices for indoor pedestrian navigation. *IEEE Sensors Journal*, *16*(1), 224–234. https://doi.org/10.1109/JSEN.2015.2477444

Zhuang, Y., Li, Y., Qi, L., Lan, H., Yang, J., & El-Sheimy, N. (2016). A two-filter integration of MEMS sensors and WiFi fingerprinting for indoor positioning. *IEEE Sensors Journal*, *16*(13), 5125–5126. https://doi.org/10.1109/JSEN.2016.2567224

How to cite this article: Wen W, Pfeifer T, Bai X, Hsu L-T. Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter. *NAVIGATION*, 2021;*68*(2):315–331. https://doi.org/10.1002/navi.421

APPENDIX A: TRANSFORMATION MATRIX

The variable $\mathbf{R}_{GL}$ is a transformation matrix that transforms the acceleration measurement obtained from the local frame to the global frame based on $\mathbf{x}_k$, which can be expressed as

$$\mathbf{R}_{GL} = \begin{bmatrix} -\sin(\emptyset_{lon}) & -\sin(\emptyset_{lat})\cos(\emptyset_{lon}) & \cos(\emptyset_{lat})\cos(\emptyset_{lon}) \\ \cos(\emptyset_{lon}) & -\sin(\emptyset_{lat})\sin(\emptyset_{lon}) & \cos(\emptyset_{lat})\sin(\emptyset_{lon}) \\ 0 & \cos(\emptyset_{lat}) & \sin(\emptyset_{lat}) \end{bmatrix} \tag{A1}$$

where $\emptyset_{lon}$ and $\emptyset_{lat}$ represent the longitude and latitude, respectively, based on the WGS84 geodetic system (Groves, 2013), which can be derived from $\mathbf{x}_k$. The variable $\mathbf{R}_{LB}$ is the transformation matrix that transforms the acceleration measurements obtained from the body to the local frames based on $\mathbf{R}_{k,r}^{local}$, which can be expressed as follows:

$$\mathbf{R}_{LB} = \mathbf{R}_{LB}^{z}(\alpha)\,\mathbf{R}_{LB}^{y}(\beta)\,\mathbf{R}_{LB}^{x}(\gamma) \tag{A2}$$

$$\text{with } \mathbf{R}^{z}{}_{LB}(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{LB}^{y}(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$\mathbf{R}_{LB}^{x}(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix}$$

where α, β, and γ denote the yaw, pitch, and roll angles, respectively, and $\mathbf{R}_{LB}^{z}(\alpha)$, $\mathbf{R}_{LB}^{y}(\beta)$, and $\mathbf{R}_{LB}^{x}(\gamma)$ denote their rotation matrices.