

Quantifying Robot Localization Safety: A New Integrity Monitoring Method for Fixed-Lag Smoothing

Osama Abdul Hafez¹, Student Member, IEEE, Guillermo Duenas Arana², Student Member, IEEE, Mathieu Joerger³, Member, IEEE, and Matthew Spenko⁴, Senior Member, IEEE

Abstract—Localization safety, or integrity risk, is the probability of undetected localization failures and a common aviation performance metric used to verify a minimum accuracy requirement. As autonomous robots become more common, applying integrity risk metrics will be necessary to verify localization performance. This letter introduces a new method, solution separation, to quantify landmark-based mobile robot localization safety for fixed-lag smoothing estimators and compares its computation time and fault detection capabilities to a chi-squared integrity monitoring method. Results show that solution separation is more computationally efficient and results in a tighter upper-bound on integrity risk when few measurements are included, which makes it the method of choice for lightweight, safety-critical applications such as UAVs. Conversely, chi-squared requires more computing resources but performs better when more measurements are included, making the method more appropriate for high performance computing platforms such as autonomous vehicles.

Index Terms—Localization, autonomous vehicle navigation, probability and statistical methods, robot safety, performance evaluation and benchmarking.

I. INTRODUCTION

MOBILE ROBOTS have matured enough for wide deployment. Well-known examples include Unmanned Aerial Vehicles, robot vacuum cleaners, and Autonomous Vehicles (AVs). While some applications, such as cleaning, have public acceptance and pose little to no safety threat, others, like AVs, are inherently dangerous [1]. Currently, AV safety rules are adapted from related applications, such as ISO26262, ARP4754, and SOTIF, where the AV system is divided into components that are separately certified to prove an overall safety level [2]–[4]. This letter tackles one of those modules, the localization subsystem, by introducing and evaluating a new method to quantify landmark-based robot localization safety for *fixed-lag smoothing estimators* in the presence of undetected faults: solution separation integrity monitoring.

Manuscript received September 10, 2019; accepted January 27, 2020. Date of publication February 21, 2020; date of current version March 5, 2020. This letter was recommended for publication by Associate Editor N. Ständerhauf and Editor Sven Behnke upon evaluation of the reviewers' comments. This work was supported by NSF under Grant #1637899. (Corresponding author: Osama Abdul Hafez.)

Osama Abdul Hafez, Guillermo Duenas Arana, and Matthew Spenko are with the Mechanical, Materials, and Aerospace Engineering Department, Illinois Tech, Chicago, IL 60616 USA (e-mail: oabdulhafez@hawk.iit.edu; gduenasar@hawk.iit.edu; mspenko@iit.edu).

Mathieu Joerger is with the Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: joerger@vt.edu).

Digital Object Identifier 10.1109/LRA.2020.2975769

A robot's localizer estimates pose and warns the system when a minimum level of localization performance cannot be guaranteed. The majority of robotics publications assess localization safety using state estimate variance, which is insufficient when an unmodeled fault occurs [5]. Faults are rarely occurring events not modeled by the common zero mean Gaussian white noise assumption. Examples include GPS clock errors, incorrect associations among mapped objects, unmapped static objects mistaken as parts of the map, and measurements due to dynamic objects.

Instead, this work utilizes a more comprehensive safety measure: localization integrity risk. Integrity is a quantifiable performance metric used to set certifiable requirements on an individual subsystem to ensure a level of safety for the overall system [6]. Localization integrity risk is the probability of robot's pose estimate error exceeding predefined acceptable limits without triggering an alarm. Integrity risk has been the primary safety measure in open-sky Guidance Navigation Satellite Systems (GNSS)-based aviation applications for decades, and prior work has applied similar methods to robots operating in GNSS-denied environments [7]–[10].

Several publications have dealt with enhancing localization and mapping performance [11]–[16], but relatively few focus on localization safety. Those that did concentrated on the empirical evaluation of different fault detection mechanisms [17]–[19]. What distinguishes integrity monitoring from these is that it upper-bounds the risk of undetected faults while using an optimal filter [20], [21], commonly with a chi-squared test for fault detection [22].

Recent work has brought integrity monitoring from open-sky aviation applications to GNSS-denied environments such that other sensors, like lidars or cameras, must be employed to provide the necessary sub-meter position accuracy [23]–[26]. This letter builds upon that work by introducing a solution separation integrity monitoring method tailored for landmark-based localization via fixed-lag smoothing, a popular robotics localization technique that often outperforms Kalman filter accuracy. The letter also compares the technique to the previously investigated chi-squared technique.

The remainder of the letter begins with a review of fixed-lag smoothing localization where measurements may be affected by faults, a point that differentiates this work. Section III defines integrity risk and the fault detector for solution-separation and chi-squared methods. Section IV derives the estimate error and fault detector distributions for each method as a function of measurement faults. Each method's integrity risk is statistically upper-bounded in Section V. Section VI compares the

performance of each method, and simulation and experimental results are presented in Section VII. Finally, Section VIII presents the conclusion and future work.

II. FIXED-LAG SMOOTHING

This section illustrates the main components of fixed-lag smoothing localization. First, the general nonlinear optimization problem is illustrated. Then, several potential measurement models are reformulated into a generalized model that allows us to leverage prior work in integrity monitoring for GNSS applications. Finally, the robot's estimated pose is expressed as a function of the measurements.

A. Optimization Problem

Fixed-lag smoothing is an optimization problem that estimates robot pose at each epoch within a preceding time window of size M (including the current epoch) by minimizing the squared norm of the weighted measurements residual:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{j=1}^n \|\mathbf{z}_j - \mathbf{h}_j(\mathbf{x})\|_{\mathbf{V}_j}^2 \text{ such that } \|\mathbf{d}\|_{\mathbf{D}}^2 \equiv \mathbf{d}^T \mathbf{D}^{-1} \mathbf{d} \quad (1)$$

where the robot's states within the preceding time window are concatenated in the state vector, $\mathbf{x} = [\mathbf{x}_{k-M}^T \dots \mathbf{x}_{k-1}^T \mathbf{x}_k^T]^T$, such that $\mathbf{x}_k \in \mathbb{R}^m$ is the state at current time and $\mathbf{z}_j \in \mathbb{R}^{n_j}$ is the j^{th} measurement in the preceding time window. Each of the n measurements can be represented as a non-linear function of the states with additive noise and a fault:

$$\mathbf{z}_j = \mathbf{h}_j(\mathbf{x}) + \mathbf{v}_j + \mathbf{f}_j \quad (2)$$

where $\mathbf{h}_j(\cdot)$ is the j^{th} measurement observation function (known), $\mathbf{v}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_j)$ is the Gaussian white noise in the j^{th} measurement with $\mathbf{V}_j$ as the measurement noise covariance matrix, and $\mathbf{f}_j$ is the fault in the j^{th} measurement, such that $\mathbf{f}_j = \mathbf{0}$ means that the j^{th} measurement is non-faulted.

A measurement fault is a rarely occurring unknown deterministic error that can not be modeled by the Gaussian white noise assumption, which might lead to a non-zero estimate error mean. Localization algorithms *must* account for these faults as robots become more prevalent in life or mission-critical applications. In the next section, both absolute and relative measurements are written to match the format in (2).

B. Measurement Models

Relative measurements, such as the ones provided by IMU or wheel encoders, are usually modeled as:

$$\mathbf{x}_{k+1} = \mathbf{g}_j(\mathbf{x}_k, \mathbf{u}_j) - \mathbf{w}_j - \mathbf{f}_j \quad (3)$$

where $\mathbf{u}_j$ is the j^{th} relative measurement, $\mathbf{f}_j$ is the projection of the j^{th} relative measurement fault on the state space, $\mathbf{g}_j(\cdot, \cdot)$ is the state evolution model of the j^{th} relative measurement (known), and $\mathbf{w}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_j)$ is the Gaussian white process noise in the j^{th} relative measurement with known covariance matrix $\mathbf{W}_j$ (in this case, $\mathbf{w}_j$ is the projection of the j^{th} relative measurement noise on the state space). (3) is reorganized to fit

the format provided by (2) as follows:

$$\underbrace{\mathbf{0}}_{\mathbf{z}_j} = \underbrace{\mathbf{x}_{k+1} - \mathbf{g}_j(\mathbf{x}_k, \mathbf{u}_j)}_{\mathbf{h}_j(\mathbf{x})} + \underbrace{\mathbf{w}_j}_{\mathbf{v}_j} + \underbrace{\mathbf{f}_j}_{\mathbf{f}_j} \quad (4)$$

Absolute measurements, such as GNSS or features extracted from lidar measurements, naturally follow the form in (2). For example, in landmark-based navigation, $\mathbf{z}_j$ represents a feature's measurements extracted from the detected landmark, $\mathbf{h}_j(\cdot)$ relates robot states to a landmark's feature, and $\mathbf{f}_j$ is nonzero whenever a measurement extracted from the detected landmark is faulted. Examples of a landmark's feature measurement fault include data association and moving landmark faults [8], [9]. A special case of absolute measurements is the *prior measurement*, $\hat{\mathbf{x}}_{k-M}$, the state estimate at the last epoch in the preceding time window:

$$\underbrace{\hat{\mathbf{x}}_{k-M}}_{\mathbf{z}_j} = \underbrace{\mathbf{x}_{k-M}}_{\mathbf{h}_j(\mathbf{x})} + \underbrace{\boldsymbol{\delta}_{k-M}}_{\mathbf{v}_j} + \underbrace{\mathbf{f}_{k-M}}_{\mathbf{f}_j} \quad (5)$$

where $\boldsymbol{\delta}_{k-M} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}_{k-M}^{-1})$ is the Gaussian uncertainty in the prior state estimate with $\boldsymbol{\Lambda}_{k-M}$ as its information matrix.

C. Estimation

The robot's pose estimate as a function of the measurements after the optimization converges can be found by first expressing the optimization problem in (1) using batch notation:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{h}(\mathbf{x})\|_{\mathbf{V}}^2 \quad (6)$$

where $\mathbf{z} \in \mathbb{R}^N$ is the measurement vector, $N = \sum_{i=1}^n n_i$ is the number of independent measurements received during the preceding time window, and $\mathbf{V} \in \mathbb{R}^{N \times N}$ is a block matrix with measurement noise covariance matrices along its diagonal. Thereafter, the optimization problem is solved by successively linearizing the observation function, $\mathbf{h}(\mathbf{x})$, e.g. using the Gauss-Newton method. After convergence, the measurement function, $\mathbf{h}(\mathbf{x})$, is linearized around the best estimate $\mathbf{x}^*$ (obtained in the optimization's last iteration):

$$\hat{\boldsymbol{\delta}} = \underset{\boldsymbol{\delta}^*}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{h}(\mathbf{x}^*) - \mathbf{H}\boldsymbol{\delta}^*\|_{\mathbf{V}}^2 \quad (7)$$

where $\mathbf{H} = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}^*}$ is the Jacobian matrix of the observation function and $\boldsymbol{\delta}^* = \mathbf{x} - \mathbf{x}^*$. By defining $\mathbf{A} = \mathbf{V}^{-1/2} \mathbf{H}$, as the standardized observation matrix, and $\mathbf{b} = \mathbf{V}^{-1/2}(\mathbf{z} - \mathbf{h}(\mathbf{x}^*))$ as the residual vector, (7) can be expressed in the general quadratic form:

$$\hat{\boldsymbol{\delta}} = \underset{\boldsymbol{\delta}^*}{\operatorname{argmin}} \|\mathbf{A}\boldsymbol{\delta}^* - \mathbf{b}\|^2 \quad (8)$$

There are efficient methods to solve (8) by exploiting the sparseness of the standardized observation matrix, $\mathbf{A}$, [11], [13]. The solution of the least squares problem in (8) is:

$$\hat{\boldsymbol{\delta}} = \mathbf{A}^{-1} \mathbf{A}^T \mathbf{b} \quad (9)$$

where $\hat{\boldsymbol{\delta}} = \hat{\mathbf{x}} - \mathbf{x}$ is the estimate error, $\mathbf{A} = \mathbf{A}^T \mathbf{A}$ is the information matrix, and $\mathbf{x}$ is the actual robot pose. Thus, the fixed-lag smoothing state estimate can be expressed as:

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{A}^{-1} \mathbf{A}^T \mathbf{b} \quad (10)$$

III. HAZARDOUS MISLEADING INFORMATION

Localization safety is quantified using integrity risk, or the probability of Hazardous Misleading Information (HMI) [8]. HMI occurs when the estimate error in the state-of-interest (e.g. AV's lateral error) exceeds a predefined threshold or *alert limit*, and the fault detector does not trigger an alarm [24]:

$$HMI = \begin{cases} \left| \alpha^T \hat{\delta} \right| > l \cap \bigcap_{i=1}^{n_H} |\Delta_i| \leq T_{\Delta_i} & \text{for } SS \\ \left| \alpha^T \hat{\delta} \right| > l \cap q \leq T & \text{for } CS \end{cases} \quad (11)$$

where $\alpha \in \mathbb{R}^{(M+1)m}$ is the state-of-interest extraction vector; l is the alert limit; n_H is the number of fault hypotheses; q is the chi-squared fault detector such that it triggers an alarm whenever it exceeds a predefined threshold, T , $q > T$; and $\Delta_i, \forall i = 1, \dots, n_H$ are a set of statistics that comprise the solution separation fault detector such that it triggers an alarm whenever at least one of the statistics' magnitudes exceeds its corresponding predefined threshold, $T_{\Delta_i}, \bigcup_{i=1}^{n_H} |\Delta_i| > T_{\Delta_i}$.

The fault detector, a statistical measure of localization's measurement inconsistency, checks whether the mission can continue and is expressed as:

$$\begin{cases} \Delta_i = \alpha^T (\hat{x} - \hat{x}_i), & \forall i \in \{1, \dots, n_H\} & \text{for } SS \\ q = \|\mathbf{b}\|^2 & & \text{for } CS \end{cases} \quad (12)$$

where $\mathbf{b}$ is the residual vector, $\hat{x}$ is the robot pose estimated using all measurements in the time window, and $\hat{x}_i$ is the robot pose estimated using only the non-faulted measurements (specified by the i^{th} hypothesis) in the time window.

After defining the fault detectors, the probability of HMI, $P(HMI)$, is quantified under each fault hypothesis, $H_i, \forall i \in \{0, \dots, n_H\}$, where the fault hypothesis indicates which measurements are faulted and H_0 is the fault-free hypothesis. Since both the state-of-interest estimate error and the fault detector are affected by measurement faults within the time window, the hypotheses must include faults occurring in the window.

Fixed-lag smoothing is typically used in robotics as a sequential filter by including the prior estimate, $\hat{x}_{k-M}$ as a measurement. Consequently, the impact of faults occurring before the time window are accounted for by including the possibility of faults in the prior estimate among the hypotheses. Then, given a set of mutually exclusive, collectively exhaustive fault hypotheses, $\{H_0, \dots, H_{n_H}\}$, the $P(HMI)$ is evaluated as:

$$P(HMI) = \sum_{i=0}^{n_H} P(HMI|H_i) P(H_i) \quad (13)$$

[24] presents a method to evaluate fault hypotheses probabilities, $P(H_i)$, given the probability of each measurement being faulted. Since monitoring integrity for all possible hypotheses can be computationally intractable, the number of hypotheses can be limited by only considering those in which the number of simultaneous faults is less than a given number, $n_{\max}$, while accounting for the risk of having more than $n_{\max}$ simultaneous faults (see Appendix C of [27]), implementing hypotheses grouping [21], or improving the computational efficiency [7].

The next section derives the distributions of the state-of-interest estimate error and the fault detector in the presence of measurement faults for each integrity monitoring method as a preliminary step to compute $P(HMI|H_i)$.

IV. ESTIMATE AND FAULT DETECTOR DISTRIBUTIONS

As shown in (11), HMI is a function of both the estimate error and the fault detector. Therefore, quantifying $P(HMI)$ requires deriving their statistical distributions. From (9), the state estimate error is defined as the difference between the estimated and true robot poses:

$$\hat{\delta} = \hat{x} - \mathbf{x} = \mathbf{A}^{-1} \mathbf{A}^T \mathbf{V}^{-1/2} (\mathbf{v} + \mathbf{f}) \quad (14)$$

Note that the estimate error, $\hat{\delta}$, is a function of both measurement's Gaussian noise and faults. Accordingly, the distribution of the state-of-interest estimate error, $\alpha^T \hat{\delta}$, is:

$$\alpha^T \hat{\delta} \sim \mathcal{N} \left(\alpha^T \mathbf{A}^{-1} \mathbf{A}^T \mathbf{V}^{-1/2} \mathbf{f}, \alpha^T \mathbf{A}^{-1} \alpha \right) \quad (15)$$

Next, the distribution of the fault detector for the two integrity monitoring methods will be derived.

A. Solution Separation Fault Detector Distribution

After convergence for both the state estimate obtained using all measurements ($\hat{x}$) and the state estimate obtained by excluding the faulted measurements in the i^{th} fault hypothesis ($\hat{x}_i$), the solution separation fault detector ($\Delta_i, \forall i = 1, \dots, n_H$) can be expressed, by substituting (10) in (12), as follows:

$$\Delta_i = \alpha^T (\mathbf{A}^{-1} \mathbf{A}^T - \mathbf{A}_i^{-1} \mathbf{A}^T \mathbf{B}_i^T \mathbf{B}_i) \mathbf{b} \quad (16)$$

where $\mathbf{A}_i = \mathbf{A}^T \mathbf{B}_i^T \mathbf{B}_i \mathbf{A}$ is the information matrix for the i^{th} solution, $\hat{x}_i$, and $\mathbf{B}_i$ is the non-faulted measurements extraction matrix for the i^{th} hypothesis. Thus, the distribution of Δ_i is:

$$\Delta_i \sim \mathcal{N} \left(\alpha^T (\mathbf{A}^{-1} \mathbf{A}^T - \mathbf{A}_i^{-1} \mathbf{A}^T \mathbf{B}_i^T \mathbf{B}_i) \mathbf{V}^{-\frac{1}{2}} \mathbf{f}, \alpha^T \mathbf{A}_i^{-1} \alpha \right) \quad (17)$$

where $\mathbf{A}_i^{-1} = \mathbf{A}_i^{-1} - \mathbf{A}^{-1}$ (see Appendix B of [28] for proof).

Given the distribution of Δ_i , the i^{th} detector threshold (T_{Δ_i}) can be defined such that the frequency of false alarms (triggering the alarm when there are not faults) is limited by a predefined value, I_{FA} :

$$P \left(\bigcup_{i=1}^{n_H} |\Delta_i| > T_{\Delta_i} | H_0 \right) \leq \sum_{i=1}^{n_H} P(|\Delta_i| > T_{\Delta_i} | H_0) \leq I_{FA} \quad (18)$$

Note that the left hand side of (18) has been upper-bounded, by replacing the union by the sum, to simplify the evaluation of T_{Δ_i} . Thus, by giving each fault hypothesis equal allocation I_{FA}/n_H , the i^{th} detector threshold, T_{Δ_i} , can be expressed as:

$$T_{\Delta_i} = \Phi^{-1} \left[1 - \frac{I_{FA}}{2n_H} \right] \sqrt{\alpha^T \mathbf{A}_i^{-1} \alpha} \quad (19)$$

where $\Phi^{-1}[\cdot]$ is the inverse of the standard normal Cumulative Distribution Function (CDF).

B. Chi-Squared Fault Detector Distribution

After convergence, the fault detector, q , can be written as:

$$q = \left\| \mathbf{V}^{-1/2} (\mathbf{z} - \mathbf{h}(\hat{x})) \right\|^2 = \left\| \mathbf{V}^{-1/2} (\mathbf{v} + \mathbf{f}) - \mathbf{A} \hat{\delta} \right\|^2 \quad (20)$$

Substituting (14), the definition of $\hat{\delta}$, in (20) results in:

$$q = \left\| (\mathbf{I} - \mathbf{A} \mathbf{A}^{-1} \mathbf{A}^T) \mathbf{V}^{-1/2} (\mathbf{v} + \mathbf{f}) \right\|^2 \quad (21)$$

Accordingly, the distribution of q is defined as follows:

$$q \sim \chi^2_{N-(M+1)m, \lambda} \quad (22)$$

where $\chi^2_{a,b}$ denotes non-central chi-squared distribution with a degrees of freedom and non-centrality parameter b . The non-centrality parameter for q is:

$$\lambda = \mathbf{f}^T \mathbf{V}^{-1/2} (\mathbf{I} - \mathbf{A} \mathbf{\Lambda}^{-1} \mathbf{A}^T) \mathbf{V}^{-1/2} \mathbf{f} \quad (23)$$

Note that the number of degrees of freedom is less than N because robot pose estimate, $\hat{\mathbf{x}}$, is linearly dependant on the measurement vector, $\mathbf{z}$.

Given q 's distribution, the detector threshold, T , can be defined so that the probability of false alarms is upper-bounded by a predefined value, I_{FA} :

$$P(q > T | H_0) = I_{FA} \quad (24)$$

Therefore, the detector threshold, T , can be expressed as:

$$T = X_{N-(M+1)m}^{-2} [1 - I_{FA}] \quad (25)$$

where $X_a^{-2}[\cdot]$ is the inverse of the central chi-squared CDF with a degrees of freedom.

This section presented the distribution of the estimate error and the fault detectors, which will be used in the next section to quantify the conditional probability of HMI, $P(HMI|H_i)$, for each integrity monitoring method.

V. INTEGRITY MONITORING

Measurement faults are low frequency events not accounted for by the Gaussian white noise assumption. In addition, these events do not follow any clear pattern; thus, they are difficult to statistically model [23]. Therefore, this work models faults as unknown deterministic variables, which in turn makes quantifying $P(HMI|H_i)$ challenging. This section will upper-bound $P(HMI|H_i)$ for each integrity monitoring method.

A. Solution Separation Integrity Monitoring

$P(HMI|H_i)$, the probability of HMI given the i^{th} fault hypothesis, for the solution separation method is:

$$P(HMI|H_i) = P\left(|\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}| > l \bigcap_{j=1}^{n_H} |\Delta_j| \leq T_{\Delta_j} | H_i\right) \quad (26)$$

Note that in the i^{th} fault hypothesis the i^{th} solution, $\hat{\mathbf{x}}_i$, is fault-free (unlike $\hat{\mathbf{x}}$ and the rest $n_H - 1$ solutions), and so $|\Delta_i|$ is the most likely statistic to exceed its threshold, T_{Δ_i} . Accordingly, the right hand side of (26) will be upper-bounded by neglecting the non-detection events of all statistics except the i^{th} one [28]:

$$P(HMI|H_i) \leq P\left(|\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}| > l \cap |\Delta_i| \leq T_{\Delta_i} | H_i\right) \quad (27)$$

Using the conditional probability theorem and upper bounding $P(|\Delta_i| \leq T_{\Delta_i} | H_i)$ by one [28]:

$$P(HMI|H_i) \leq P\left(|\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}| > l \mid |\Delta_i| \leq T_{\Delta_i}, H_i\right) \quad (28)$$

Knowing that $\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}} = \boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}_i + \Delta_i$ and that if $a = b + c$ then $|a| \leq |b| + |c|$, the magnitude of the state-of-interest estimate error,

$|\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}|$, and the magnitude of the i^{th} statistic, $|\Delta_i|$, can be related as follows [28]:

$$|\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}| \leq |\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}_i| + |\Delta_i| \quad (29)$$

which will upper-bound the right hand side of (28):

$$P(HMI|H_i) \leq P\left(|\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}_i| + |\Delta_i| > l \mid H_i, |\Delta_i| \leq T_{\Delta_i}\right) \quad (30)$$

Now, the condition $|\Delta_i| \leq T_{\Delta_i}$ will upper-bound $|\Delta_i|$ by T_{Δ_i} :

$$P(HMI|H_i) \leq P\left(|\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}_i| > l - T_{\Delta_i} | H_i\right) \quad (31)$$

Recall that the i^{th} solution, $\hat{\mathbf{x}}_i$, is obtained by excluding the faulted measurements in the i^{th} hypothesis, H_i . Thus, $P(HMI|H_i)$'s upper-bound in (31) is remarkable because the distribution of $\hat{\mathbf{x}}_i$ given H_i is not a function of faults [28]:

$$\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}_i | H_i \sim \mathbb{N}(0, \boldsymbol{\alpha}^T \mathbf{\Lambda}_i^{-1} \boldsymbol{\alpha}) \quad (32)$$

Accordingly, the probability of HMI given the i^{th} hypothesis, $P(HMI|H_i)$, reduces to:

$$P(HMI|H_i) \leq 2\Phi\left[\frac{T_{\Delta_i} - l}{\sqrt{\boldsymbol{\alpha}^T \mathbf{\Lambda}_i^{-1} \boldsymbol{\alpha}}}\right] \quad (33)$$

where $\Phi[\cdot]$ is the standard normal CDF.

B. Chi-Squared Integrity Monitoring

The probability of HMI given the i^{th} fault hypothesis, $P(HMI|H_i)$, for chi-squared integrity monitoring can be expressed as follows:

$$P(HMI|H_i) = P\left(|\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}| > l \cap q \leq T | H_i\right) \quad (34)$$

Recall, the fault detector, q , follows a chi-squared distribution, whereas the estimate error, $\hat{\boldsymbol{\delta}}$, follows a Gaussian distribution, which makes the evaluation of their joint probability complex. Luckily, they are statistically independent (see Appendix C of [28] for proof) so the events can be evaluated individually:

$$P(HMI|H_i) = P\left(|\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}| > l | H_i\right) P(q \leq T | H_i) \quad (35)$$

Given the distributions of the state-of-interest estimate error, $\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}$, and fault detector, q , $P(HMI|H_i)$ is expanded as:

$$P(HMI|H_i) = \left(\Phi\left[\frac{l - \mu_{\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}}}{\sqrt{\boldsymbol{\alpha}^T \mathbf{\Lambda}^{-1} \boldsymbol{\alpha}}}\right] - \Phi\left[\frac{-l - \mu_{\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}}}{\sqrt{\boldsymbol{\alpha}^T \mathbf{\Lambda}^{-1} \boldsymbol{\alpha}}}\right]\right) X_{N-(M+1)m, \lambda}^2[T] \quad (36)$$

where $\mu_{\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}}$ is the state-of-interest estimate error mean defined in (15), and λ is the non-centrality parameter defined in (23). Note that $\mu_{\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}}$ and λ are both affected by measurement faults, $\mathbf{f}$. Unlike the solution separation method where $P(HMI|H_i)$ can be manipulated to get a formula that is not a function of measurement faults, $P(HMI|H_i)$ for the chi-squared method is upper-bounded by searching for the fault vector that maximizes it (worst-case fault vector) given the set of faulted measurements specified by the i^{th} hypothesis, H_i :

$$\mathbf{f}_i^{worst} = \mathbf{f}_i^{dir} \Gamma_i^{mag} \quad (37)$$

Algorithm 1: Summary of Solution Separation Method Bounds.

- 1: Estimate robot pose, $\hat{\mathbf{x}}$, using (1).
 - 2: **for** every hypothesis $H_i \forall i \in 1, \dots, n_H$ **do**
 - 3: - Estimate the i^{th} solution, $\hat{\mathbf{x}}_i$, using (1).
 - 4: - Quantify the i^{th} detector, Δ_i , using (12).
 - 5: - Evaluate T_{Δ_i} using (19), and check the alarm.
 - 6: - Quantify $P(HMI|H_i)$ using (33) and $P(H_i)$ as in [24].
 - 7: Evaluate the integrity risk, $P(HMI)$, using (13).
-

Algorithm 2: Summary of Chi-Squared Method Bounds.

- 1: Estimate robot pose, $\hat{\mathbf{x}}$, using (1).
 - 2: Quantify detector q , and threshold, T , using (12) and (25).
 - 3: **for** every hypothesis $H_i \forall i \in 1, \dots, n_H$
 - 4: - Evaluate i^{th} worst-case fault direction, $\mathbf{f}_i^{dir}$, using (38).
 - 5: - Quantify $P(HMI|H_i)$ using (39) and $P(H_i)$ as in [24].
 - 6: Evaluate the integrity risk, $P(HMI)$, using (13).
-

where $\mathbf{f}_i^{dir}$ and $\mathbf{f}_i^{mag}$ is the direction and magnitude of the worst-case fault vector, respectively. [28] proved that the worst-case fault vector direction, $\mathbf{f}_i^{dir}$, can be obtained analytically as:

$$\mathbf{f}_i^{dir} = \mathbf{E}_i^T [\mathbf{E}_i (\mathbf{I} - \mathbf{A}\mathbf{A}^{-1}\mathbf{A}^T) \mathbf{E}_i^T]^{-1} \mathbf{E}_i \mathbf{A}\mathbf{A}^{-1} \boldsymbol{\alpha} \quad (38)$$

where $\mathbf{E}_i$ is the faulted measurements extraction matrix for the i^{th} hypothesis. After substituting $\mathbf{f}_i^{dir}$ in (36), the upper-bound on $P(HMI|H_i)$ can be quantified by searching for $\mathbf{f}_i^{mag}$ that maximizes (36).

$$P(HMI|H_i) \leq \max_{\mathbf{f}_i^{mag}} \left[\left(\Phi \left[\frac{l - \mu_{\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}}}{\sqrt{\boldsymbol{\alpha}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\alpha}}} \right] - \Phi \left[\frac{-l - \mu_{\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}}}{\sqrt{\boldsymbol{\alpha}^T \boldsymbol{\Lambda}^{-1} \boldsymbol{\alpha}}} \right] \right) X_{N-(M+1)m, \lambda}^2 [T] \right] \quad (39)$$

This section derived an upper-bound on $P(HMI|H_i)$ for both the solution separation and chi-squared methods. Summaries of the two procedures are shown in algorithms 1 and 2. The next compares the performance of the two methods.

VI. TECHNICAL COMPARISON

This section compares the solution separation and chi-squared integrity monitoring algorithms based on: 1) the computational load needed to evaluate the fault detector, 2) the computational load needed to evaluate the integrity risk bound, and 3) the tightness of the integrity risk bound.

A. Computational Efficiency of Fault Detector Evaluation

Evaluating the solution separation fault detector, (12), requires implementing the fixed-lag smoothing algorithm n_H times at each epoch to estimate the n_H solutions evaluated after removing the faulted measurements in each hypothesis. In contrast, the chi-squared fault detector, (12), can be evaluated directly from the pose estimate without optimization, making the computation of the chi-squared detector more efficient.

B. Computational Efficiency of Integrity Risk Evaluation

Quantifying the chi-squared integrity risk bound (39) requires solving n_H optimization problems to find the worst-case fault magnitude that maximizes the conditional integrity risk, $P(HMI|H_i)$, $\forall i = 1, \dots, n_H$. This type of optimization problem is usually solved using non-gradient optimizers, which are significantly slower than those used to solve the non-linear least squares problem in fixed-lag smoothing.

In contrast, the integrity risk bound in solution separation (33) does not require numerical optimization since the full set information matrix, $\boldsymbol{\Lambda}$, and the n_H solutions' information matrices, $\boldsymbol{\Lambda}_i$, $\forall i = 1, \dots, n_H$, become available when the state has been estimated and the fault detector has been evaluated, respectively. Thus, solution separation is more efficient from the perspective of integrity risk bound evaluation.

C. Tightness of the Integrity Risk Upper-Bound

For solution separation, the first bound in (27) only considers the i^{th} statistic not triggering an alarm (i.e. it ignores all others; Δ_j , $\{\forall j = 1, \dots, n_H, \text{ such that } i \neq j\}$). This bound is tight because the full set solution, $\hat{\mathbf{x}}$, and all of the n_H solutions except the i^{th} one, ($\hat{\mathbf{x}}_j$, $\{\forall j = 1, \dots, n_H, \text{ such that } i \neq j\}$), will be affected by measurement faults. In other words, all of the statistics involve a subtraction between two random variables, both of which are affected by faults (except for the i^{th} statistic, which involves a subtraction between a faulted and fault-free random variable); thus, the ignored events are expected to occur with high confidence.

The second bound, (28), assumes the i^{th} statistic's nondetection probability, Δ_i , is one. The bound might be loose because the higher the fault magnitude, the lower the probability that Δ_i will not trigger an alarm. The bound becomes tighter as the number of measurements increases because the impact of measurement faults on the mean of Δ_i reduces.

The third bound, (29) and (31), bounds the full set estimate error magnitude, $\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}$, by the sum of the i^{th} solution error's magnitude, $\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}_i$, and the i^{th} threshold, T_{Δ_i} . Of all three bounds, it has the highest impact on the integrity risk bound's tightness. If the number of measurements is low, then the probability of the faulted estimate's magnitude, $\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}$, being greater than the fault-free estimate's magnitude, $\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}_i$, becomes higher, $|\Delta_i|$ approaches T_{Δ_i} , and the bound tightens. Conversely, if the number of measurements is high, then the probability of the faulted estimate's magnitude being greater than the fault-free estimate's magnitude becomes lower, $|\Delta_i|$ becomes much smaller than T_{Δ_i} , and the bound loosens.

Unlike solution separation, chi-squared integrity risk has a single bound that uses the worst-case fault instead of the actual (unknown) fault; thus, tightness depends on the number of measurements. If low, then the impact of measurement faults on the probability of the estimate error's magnitude, $\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}$, being greater than the alert limit, l , becomes higher, and the probability of no detection becomes more sensitive to measurement faults. Thus, the difference between the worst-case fault and the actual (unknown) fault will have a high impact on $P(HMI|H_i)$, and the bound will be loose. Conversely, if the number of measurements is high, then the measurement fault's impact on the probability of the estimate error's magnitude, $\boldsymbol{\alpha}^T \hat{\boldsymbol{\delta}}$, being greater than the alert limit, l , becomes lower, and the probability

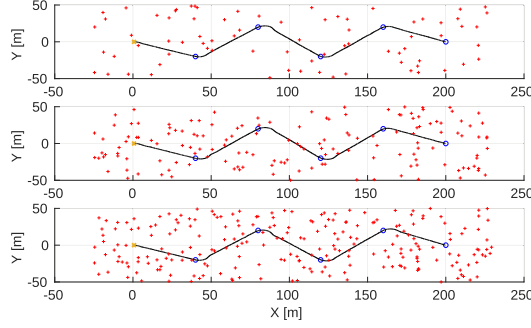


Fig. 1. Estimated robot trajectory while navigating through each of the three randomly generated maps (3e-3 (top), 6e-3 (middle), and 9e-3 (bottom) landmarks/ m^2). Blue circles refer to way-points, red plus-signs represent landmarks, and the yellow cross-sign is the robots' starting location.

TABLE I
SIMULATION PARAMETERS

Velocity	25 km h ⁻¹	$\sigma_{velocity}$	1 m s ⁻¹
Time step	0.1 s	σ_{gyro}	2 ° s ⁻¹
Sensor range	25 m	σ_{lidar}	0.2 m
Alert limit	0.5 m	$\sigma_{steering\ angle}$	2 °
Fault probability	10 ⁻³	I_{FA}	10 ⁻⁵

TABLE II
OBSERVATIONS FROM THE SIMULATION

	Landmark Density [m ⁻²]		
	3e-3	6e-3	9e-3
Avg. time window size (epochs)	5.7	3.3	3.0
Avg. # of landmarks in the window	21.9	26.3	34.3

of no detection becomes less sensitive to measurement faults. Thus, the difference between the worst-case fault and the actual (unknown) fault will have low impact on $P(HMI|H_i)$, and the bound will be tight.

VII. EMPIRICAL COMPARISON

This section quantifies localization safety for fixed lag smoothing using the two integrity monitoring algorithms. Simulation results compare the integrity risk bounds' tightness and computational efficiency, yielding guidelines for conditions in which each method is more suitable. Experiments demonstrate the performance of each method in real environments.

A. Simulation Results

The simulation consists of a constant-velocity bicycle model navigating predefined way-points in three environments of different landmark densities (see Fig. 1 and Table I). Relative measurements, steering angle and linear and angular velocities, are assumed to be fault free. Absolute measurements, range and bearing to mapped landmarks, can be faulted with a probability of 10⁻³. The time window is continuously resized to include the smallest number of landmark detections above 20 to maintain reasonable computational time [24]. For each landmark density, the two integrity monitoring methods were assessed using the same landmark locations (see Table II).

Fig. 2 shows the integrity risk bounds versus time for each landmark density. Both methods provide a guaranteed upper-bound on integrity risk, but the solution separation method

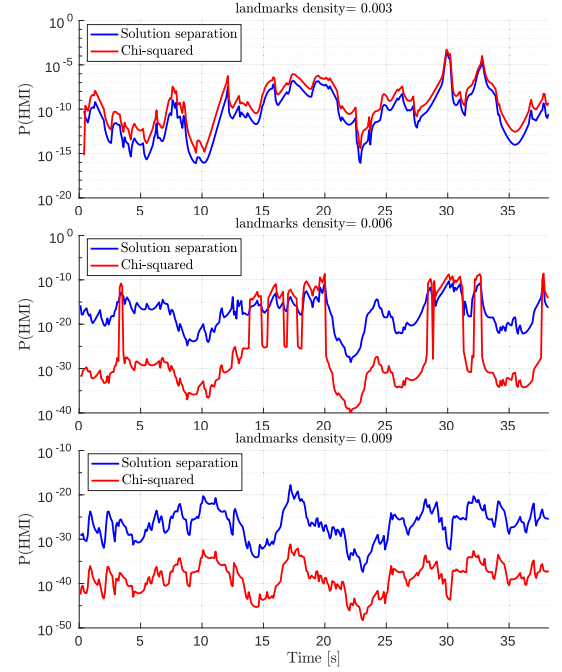


Fig. 2. Integrity risk bound for solution separation (blue) and chi-squared (red) for 0.003, 0.006, 0.009 landmark densities (top, middle, bottom).

provides a uniformly tighter bound than the chi-squared method when the number of measurements is small (see Fig. 2 top). As the number of measurements increases, the integrity risk bound becomes looser for solution separation and tighter for the chi-squared method (see Fig. 2, middle). Finally, when the number of measurements is relatively large, the chi-squared method provides a uniformly tighter integrity risk bound (see Fig. 2, bottom). This was expected, as illustrated in Section VI, because the most dominant (third) bound in solution separation becomes tighter as the number of measurements decreases, whereas the chi-squared method's only bound becomes tighter as the number of measurements increases.

To assess computational efficiency, the simulation was repeated multiple times for different landmark densities. Fig. 3 (top) shows that the average number of landmark detections in the time window increases as landmark density increases, whereas the middle shows that the average number of time epochs in the window decreases as landmark density increases to maintain the smallest number of landmark detections in the time window above 20. Fig. 3 (bottom) shows the scaled computational time versus landmark density in the map.

As outlined in Section VI, the chi-squared detector requires much less time than the solution separation detector for all landmark densities. The solution separation detector's computation time is nonlinear as a function of landmark density because when landmark density is very small, although the number of fault hypotheses (specified by the number of measurements) is low, the number of states to estimate per hypothesis (specified by the time window size) is very high, and so the elapsed time is high. However, as landmark density increases, the number of states to estimate per hypothesis reduces, but the number of fault hypotheses increases, and thus the elapsed time decreases and then, when the number of fault hypotheses dominates, increases.

Although the chi-squared detector does not require optimization and its computational time is proportional to the size of the

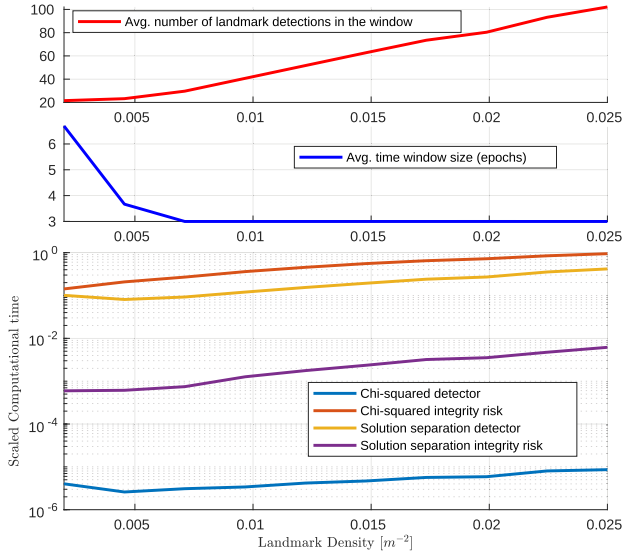


Fig. 3. Top—average number of landmark detections in the time window. Middle—average number of time epochs in the window. Bottom—scaled computational time (time per 5 s) for chi-squared and solution-separation, fault detector and integrity risk versus landmark densities in the map.

residual vector, $\mathbf{b}$, its computational time is also nonlinear as a function of landmark density. When landmark density is low, the number of epochs in the time window is high, and so the number of relative measurements is high. Thus, the total number of measurements, represented by the size of $\mathbf{b}$, is very high and so is the elapsed time. However, as landmark density increases, the number of relative measurements in the window reduces and the number of landmark detections increases. Thus, the elapsed time reduces and then, when the number of landmark detections dominates, increases. For both solution separation and chi-squared methods, the time needed to evaluate the integrity risk increases as the number of fault hypotheses increases, but the solution separation method needs much less time than the chi-squared method for all landmark densities and is more computationally efficient.

Integrity risk evaluation efficiency for both methods could be improved by around $1/n_H$ by using a separate processor thread for each conditional integrity risk, $P(H_{MI}|H_i)$, as they are independent. Multi-threading could also improve evaluation times for the solution separation fault detector. In addition, the computing time dedicated to evaluating the solution separation detector can be reduced by considering the sparse structure of the standardized observation matrix, $\mathbf{A}$, when solving the non-linear least squares problem [13]. Thus, the solution separation method has large potential for real-time applications. Accordingly, as a rule of thumb, solution separation should be used to quantify localization safety unless the number of measurements becomes relatively high. Ultimately, since the threshold for the number of measurements in the time window is based on sensor type, the quality of sensor measurements, and the quality of the feature extraction algorithm, the suitable method should be determined through some pre-analysis.

B. Experimental Results

The performance of the two methods was assessed using real-world data from a car traversing a university campus.

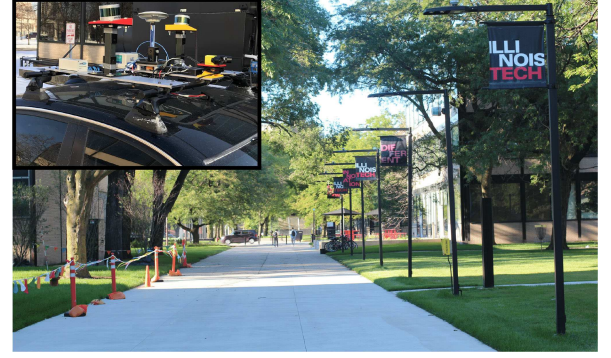


Fig. 4. Testing environment with test setup in the upper left. Both tree trunks and poles are used as landmarks for localization. The sensor suite consists of STIM-300 tactical-grade IMU, two Velodyne VLP-16 lidars, and Novatel SPAN-CPT DGPS attached to a roof-rack of a vehicle.

TABLE III
EXPERIMENT PARAMETERS

$\sigma_{DgpsPosXYZ}$	0.013, 0.006, 0.014 m	σ_{lidar}	0.2 m
$\sigma_{DgpsVelXYZ}$	0.042, 0.021, 0.044 m s ⁻¹	Alert limit	0.5 m
PSD _{AccelNoise}	0.002 m ² /s ⁵	I_{FA}	10 ⁻⁵
PSD _{GyroNoise}	3.05×10^{-6} rad ² /s ³	frequency _{Dgps}	1 Hz
PSD _{AccelBias}	2.40×10^{-7} m ² /s ⁶	frequency _{lidar}	10 Hz
PSD _{GyroBias}	2.12×10^{-12} rad ² /s ⁴	frequency _{IMU}	125 Hz

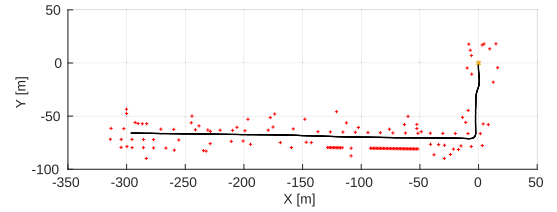


Fig. 5. Estimated vehicle trajectory. The vehicle starts from the yellow dot at (0, 0), and red-plus signs denote mapped landmarks (poles and tree trunks).

1) *Setup*: Fig. 4 shows the testing environment and sensor suite. Absolute measurements are range and bearing to light poles and tree trunks (landmarks) extracted from two synchronized Velodyne VLP-16 lidar point clouds and Differential GPS (DGPS) updates from a Novatel SPAN-CPT. A STIM-300 tactical grade Inertial Measurement Unit (IMU) provides relative measurements (see Table III).

The landmark map was obtained *a priori* using SLAM. The state vector has 15 states: six for pose, three for velocity, and six for IMU biases. Fig. 5 shows the estimated trajectory and landmark map. We assume only absolute measurements can be faulted (10^{-3} failure probability). The lateral-error is the state-of-interest with an alert limit of 0.5 m. Last, the time window is resized continuously so that it contains the minimum number of landmark detections or DGPS updates above 30.

2) *Results*: Fig. 6 shows the integrity risk bound. As before, solution separation is uniformly tighter at all times except between 5 and 25 s, when not enough landmarks are visible. This increases the time window, resulting in a large number of DGPS updates (see Fig. 7). Thus, the number of measurements in the time window increases, because there are six measurements per DGPS update, whereas each landmark detection includes only two measurements.

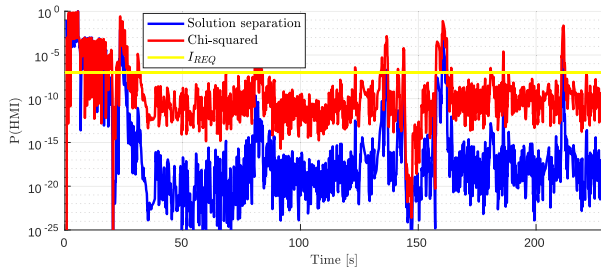


Fig. 6. Experimental integrity risk bound for solution separation (blue) and chi-squared (red) methods.

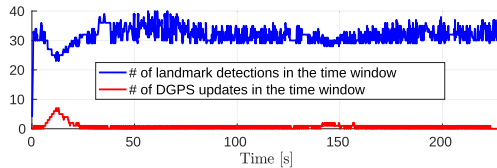


Fig. 7. The number of landmark detections and DGPS updates in the time window as a function of time.

The large variation in integrity risk is due to the extremely low nominal DGPS uncertainty with respect to landmark detections (see Table III). Unlike feature measurements, DGPS measurements are a direct observation of the robot state. Thus, if there are several DGPS measurements during the time window, the integrity risk decreases dramatically. Additionally, if the detected landmarks are spread apart enough with sufficient redundancy, adding or removing a single feature does not significantly affect integrity risk.

Availability, the percentage of time integrity risk is below a predefined requirement (chosen as $I_{REQ} = 10^{-7}$ based on [27]), for solution separation is 92% and 85% for the chi-squared method. This provides insight on how sensitive the tightness of the bound is on whether a robot can continue operation.

VIII. CONCLUSION

This letter presents a solution separation-based mobile robot localization safety method for fixed-lag smoothing estimators, and compares its performance with a chi-squared technique. The results show that solution separation is always more computationally efficient than the chi-squared method, but solution separation offers a tighter upper-bound than chi-squared only when small number of measurements are in the time-window, and vice-versa. In future work, we will derive integrity monitoring methods for the more challenging SLAM problem. Code to evaluate integrity can be found at <https://github.com/mspenko/RoboticsLab-Integrity-Evaluation>.

REFERENCES

- [1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Res. Part A: Policy Practice*, vol. 77, pp. 167–181, 2015.
- [2] ISO 26262-1 Road vehicles – Functional safety, International Standardization Organization Std.
- [3] ARP4754A Guidelines For Development Of Civil Aircraft and Sys., Aerospace Recommended Practice Std.
- [4] ISO/PAS 21448 Road vehicles – Safety of the intended functionality, International Standardization Organization Std.
- [5] N. A. Othman and H. Ahmad, "The analysis of covariance matrix for Kalman filter based slam with intermittent measurement," in *Proc. Int. Conf. Syst., Control, Informat.*, 2013, pp. 227–231.
- [6] R. Kelly and J. Davis, "Required navigation performance (RNP) for precision approach and landing with GNSS application," *Navigation*, vol. 41, no. 1, pp. 1–30, 1994.
- [7] M. Joerger *et al.*, "Landmark selection and unmapped obstacle detection in lidar-based navigation," in *Proc. 30th Int. Tech. Meeting Satell. Division Inst. Navigation (ION GNSS)*, 2017, pp. 1886–1903.
- [8] G. DuenasArana, M. Joerger, and M. Spenko, "Local nearest neighbor integrity risk evaluation for robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 2328–2333.
- [9] M. Joerger *et al.*, "A new approach to unwanted-object detection in GNSS/LiDAR-based navigation," *Sensors*, vol. 18, no. 8, 2018, Art. no. 2740.
- [10] C. Li and S. L. Waslander, "Visual measurement integrity monitoring for UAV localization," in *Proc. IEEE Int. Symp. Safety, Secur. Rescue Robot.*, Sep. 2019, pp. 22–29.
- [11] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3607–3613.
- [12] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.
- [13] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations Trends, Robot.*, vol. 6, pp. 1–139, 2017.
- [14] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 1879–1884.
- [15] R. Dubé *et al.*, "SegMap: Segment-based mapping and localization using data-driven descriptors," *Int. J. Robot. Res.*, vol. 39, pp. 339–355, 2019.
- [16] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 62–69.
- [17] V. Kadiramanathan *et al.*, "Particle filtering-based fault detection in nonlinear stochastic systems," *Int. J. Syst. Sci.*, vol. 33, pp. 259–265, 2002.
- [18] S. Hewitson and J. Wang, "Extended receiver autonomous integrity monitoring (e RAIM) for gnss/ins integration," *J. Surv. Eng.*, vol. 136, no. 1, pp. 13–22, 2010.
- [19] A. V. Kanhere and G. X. Gao, "Integrity for GPS/LiDAR fusion utilizing a RAIM framework," in *Proc. 31st Int. Tech. Meeting Satell. Division Inst. Navigation, ION GNSS*, 2018, pp. 3145–3155.
- [20] J. Blanch *et al.*, "Baseline advanced RAIM user algorithm and possible improvements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 713–732, Jan. 2015.
- [21] J. Blanch *et al.*, "Reducing computational load in solution separation for Kalman filters and an application to PPP integrity," in *Proc. Inst. Navigation Int. Tech. Meet.*, 2019.
- [22] B. W. Parkinson and P. Axelrad, "Autonomous GPS integrity monitoring using the pseudorange residual," *Navigation*, vol. 35, no. 2, pp. 255–274, 1988.
- [23] G. D. Arana, M. Joerger, and M. Spenko, "Efficient integrity monitoring for KF-based localization," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 6374–6380.
- [24] G. D. Arana, O. A. Hafez, M. Joerger, and M. Spenko, "Recursive integrity monitoring for mobile robot localization safety," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 305–311.
- [25] O. A. Hafez, G. D. Arana, and M. Spenko, "Integrity risk-based model predictive control for mobile robots," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 5793–5799.
- [26] G. D. Arana, O. A. Hafez, M. Joerger, and M. Spenko, "Integrity monitoring for kalman filter-based localization," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 6374–6380.
- [27] J. Blanch *et al.*, "Advanced RAIM user algorithm description: Integrity support message processing, fault detection, exclusion, and protection level calculation," in *Proc. 25th Int. Tech. Meeting Satell. Division Inst. Navigation (ION GNSS 2012)*, 2012, pp. 2828–2849.
- [28] M. Joerger *et al.*, "Solution separation versus residual-based RAIM," *Navigation: J. ION*, vol. 61, no. 4, pp. 273–291, 2014.