2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Localization Safety Validation for Autonomous Robots

Guillermo Duenas Arana[1], Osama Abdul Hafez[1], Mathieu Joerger[2], and Matthew Spenko[1]

Abstract— **This paper presents a method to validate localization safety for a preplanned trajectory in a given environment. Localization safety is defined as integrity risk and quantified as the probability of an undetected localization failure. Integrity risk differs from previously used metrics in robotics in that it accounts for unmodeled faults and evaluates safety under the worst possible combination of faults. The methodology can be applied prior to mission execution and thus can be employed to evaluate the safety of potential trajectories. The work has been formulated for localization via smoothing, which differs from previously reported integrity monitoring methods that rely on Kalman filtering. Simulation and experimental results are analyzed to show that localization safety is effectively quantified.**

I. Introduction

In recent years, advances in autonomous navigation technologies have been developed at an accelerated pace, to the point where the first self-driving taxi was launched in Arizona in December 2018 [1]. These fast-paced developments in critical applications such as autonomous vehicles (AV) have brought the attention of regulators that are trying to create a regulatory framework to enable further AV testing in public roads [2], [3]. However, no clear and definite safety standard exists for AVs at the moment. Most efforts focus on adapting current safety standards such as ISO26260 and ARP4754, where the AV system is divided into interdependent subsystems that are separately certified in order to prove an overall level of safety [4], [5]. This paper deals with the safety of one of those subsystems, the localization module.

Localization is paramount in autonomous navigation since failures may have catastrophic consequences, e.g. a lateral localization error on the order of decimeters could mislead the controls subsystem into driving into an adjacent lane. Most robotics publications evaluate localization performance as a measure of the estimate variance, which is understood as not being sufficient when unmodeled faults occur. Faults are rarely occurring events not modeled by the usual Gaussian noise assumption. Examples include incorrect associations, misleading measurements due to moving objects, and previously unmapped static objects mistaken by parts of the map.

This paper utilizes a more suitable safety metric: the localization integrity risk. Integrity is a quantifiable performance metric used to set certifiable requirements on individual system components to achieve and prove a level of safety for the overall system [6]. More precisely, localization integrity risk is the probability that a robot's pose estimation lies outside pre-defined acceptable limits and no alarm is triggered. For decades, integrity risk has been the primary safety metric on open-sky GNSS-based aviation applications, and recent work has aimed at bringing similar techniques to terrestrial robots [7], [8], [9]. In this work, we propose a methodology that, given a map and system specifications, validates localization safety for a predefined trajectory prior to execution.

A. Related Work

Numerous publications have focused on improving localization and mapping performance over the last decade [10], [11], [12]. However, relatively few deal with localization safety. Of those that do, most focus on the experimental evaluation of different fault detector mechanisms [13], [14]. There is also some other work in robust filtering techniques, such as the H_∞ filter [15], which reduces the maximum localization error at the expense of nominal performance, and work in reducing the risk of collisions [16].

Integrity monitoring differs from other approaches in that it upper bounds the risk of undetected failures while using an optimal filter. [17] presented the first integrity monitoring methodology for GNSS-based applications that uses chi-squared tests for fault detection; since then, integrity monitoring has experienced great improvements in performance and efficiency [18], [19].

Recent work by the authors has focused on transitioning integrity monitoring methods from open-sky applications to the more challenging case of terrestrial robots, where GNSS must be combined with other sensors to provide the necessary sub-meter accuracy. [20] presents a sequential method to monitor integrity in Kalman Filter (KF)-based localization, but it is not suitable for AV applications where landmarks come in and out of sight. An efficient method to monitor integrity using a KF without the previous limitations is presented in [21], and a more complex KF-based approach that uses a preceding time window for integrity monitoring in [22]. Additionally, the risk of faults in the data association process is upper bounded in [8], the benefits of incorporating an IMU for data associations is analyzed in [23], and the use of an integrity risk metric in a model predictive control framework to generate safe trajectory in [24].

This work differs from previous integrity monitoring methods in two ways. First, the approach is intended for localization via smoothing, which offers better accuracy than filtering and, thanks to relatively recent methods that exploit the sparseness of the problem such as [10], [11], can be solved very efficiently. This differs from previous work by the authors that focus on Kalman Filter-based localization.

*This work was supported by NSF Grant #1637899.

[1]GD. Arana, OA. Hafez and M. Spenko are with the Mechanical, Materials and Aerospace Department, Illinois Institute of Technology, Chicago, IL, USA gdueasar@hawk.iit.edu

[2]M. Joerger is with the Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA, USA

Second, the method can be applied off-line, i.e. before the mission is carried out, which may be helpful to quantify the localization safety of a given trajectory.

B. Overview

The remainder of the paper begins with a brief review of fixed-lag smoothing for localization within a map of landmarks in Section II. Unlike most robotics publications, we explicitly indicate that measurements may be affected by faults and present the relation between those and the estimate error. Similarly, the residuals' norm is employed as the fault detector, which is also expressed as a function of the measurements' faults.

Section III presents the main steps for offline integrity monitoring. Integrity risk is upper bounded for a predefined trajectory estimated from the predictive mission model. Estimate error and detector distributions are determined for the hypothesized fault modes, which indicate which measurements are faulted. Integrity risk is computed under the worst-case fault assumption, the fault that maximizes integrity risk, corresponding to each fault mode.

The methodology is implemented in Section IV. Simulations show that the methodology computes a conservative upper bound on the actual integrity risk and experimental results show its applicability to real-world situations. Finally, Section V presents conclusions and future work.

II. Fixed-Lag Smoothing

This section presents the basic elements of fixed-lag smoothing localization. First, we state the general optimization problem, which is recursively solved to obtain the pose estimate. Then, different possible measurement models are unified into a generalized model that allows us to leverage prior work in integrity monitoring. Last, we define the estimate error and fault detector, exposing their connection with the faults.

A. Problem Statement

Fixed-lag smoothing estimates $M+1$ states by minimizing the weighted squared norm of the measurements residuals:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i}^{n_z} \|\mathbf{z}_i - \mathbf{h}_i(\mathbf{x})\|^2_{\mathbf{V}_i} \tag{1}$$

where $\|\mathbf{a}\|^2_{\mathbf{A}} = \mathbf{a}^T\mathbf{A}^{-1}\mathbf{a}$ and the robot states are stacked in the state vector, $\mathbf{x} = \begin{bmatrix}\mathbf{x}_{k\text{-}M}^T & \dots & \mathbf{x}_{k\text{-}1}^T & \mathbf{x}_k^T\end{bmatrix}^T$, such that $\mathbf{x}_k$ is state at the current time. Each of the n_z measurements during the time window, $\mathbf{z}_i$, is a vector of dimension n_i that can be expressed as nonlinear function of the states corrupted by noise and possibly a fault:

$$\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}) + \mathbf{v}_i + \mathbf{f}_i \quad \text{where} \quad \mathbf{v}_i \sim \mathbb{N}(\mathbf{0}, \mathbf{V}_i) \tag{2}$$

is white Gaussian noise with covariance $\mathbf{V}_i$ and $\mathbf{h}_i(\cdot)$ is a known measurement model function. In addition to the usual Gaussian noise, measurements can be corrupted by rarely occurring faults not modeled by the Gaussian assumption. The possibility of faults is represented by the fault vector $\mathbf{f}_i$, which is only nonzero for the faulted measurements.

Next, different types of measurements are rewritten to fit the general measurement model in (2).

B. Measurements Models

Measurements are usually categorized as either absolute or relative. Absolute measurements provide an update on the robot pose with respect to an external framework, while relative measurements estimate the motion between two consecutive poses. Both of these measurements are expressed to fit the format specified in (2), as follows.

1) *Absolute measurements*, such as GNSS or landmark detections within a map usually follow (2). For example, in landmark-based navigation, $\mathbf{h}_i(\cdot)$ relates the position of landmarks with robot states, $\mathbf{z}_i$ contains the detected landmark measurements, and $\mathbf{f}_i$ is nonzero whenever a landmark is incorrectly detected or associated. Examples of landmark measurement faults include moving landmarks and data association faults. A special case of absolute measurement is the *prior measurement*, $\mathbf{x}^*_{k\text{-}M}$, which contains the state estimate at the last epoch in the time window:

$$\underbrace{\mathbf{x}^*_{k\text{-}M}}_{\mathbf{z}_i} = \underbrace{\mathbf{x}_{k\text{-}M}}_{\mathbf{h}_i(\mathbf{x})} + \underbrace{\boldsymbol{\delta}_{k\text{-}M}}_{\mathbf{v}_i} + \underbrace{\mathbf{f}_{k\text{-}M}}_{\mathbf{f}_i} \quad \text{where} \quad \boldsymbol{\delta}_{k\text{-}M} \sim \mathbb{N}\left(\mathbf{0}, \boldsymbol{\Lambda}^{-1}_{k\text{-}M}\right) \tag{3}$$

and $\boldsymbol{\Lambda}_{k\text{-}M}$ is the information matrix corresponding to the prior state. Note that some localization methods do not make use of the prior as its impact may be small when using lengthy time windows. Nevertheless, this measurement will be included in the remainder of the paper for completeness.

2) *Relative measurements*, $\mathbf{u}$, such as the ones provided by wheel encoders or IMUs are naturally expressed as:

$$\mathbf{x}_{k+1} = \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k + \mathbf{f}_u \quad \text{where} \quad \mathbf{w}_k \sim \mathbb{N}(\mathbf{0}, \mathbf{W}_k) \tag{4}$$

and $\mathbf{g}(\cdot,\cdot)$ is a known function. Reorganizing (4), we obtain a format equivalent to (2) in which the measurement is always the null vector:

$$\underbrace{\mathbf{0}}_{\mathbf{z}_i} = \underbrace{\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_{k+1}}_{\mathbf{h}_i(\mathbf{x})} + \underbrace{\mathbf{w}_k}_{\mathbf{v}_i} + \underbrace{\mathbf{f}_u}_{\mathbf{f}_i} \tag{5}$$

An alternative model can be obtained considering $\mathbf{z}_i = \mathbf{u}$, but that usually results in a more complex $\mathbf{h}_i(\mathbf{x})$.

Finally, stacking all measurements together in vector $\mathbf{z}$ of dimension $n = \sum_{i=1}^{n_z} n_i$, we obtain the batch measurement model for the time window as:

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{v} + \mathbf{f} \quad \text{where} \quad \mathbf{V} \sim \mathbb{N}(\mathbf{0}, \mathbf{V}) \tag{6}$$

Note that the elements of $\mathbf{f}$ are only nonzero when the corresponding measurements are faulted, i.e. when $\mathbf{f}_i \neq \mathbf{0}$.

C. Estimate

The optimization problem in (1) can be rewritten using the batch notation in (6) as:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{z} - \mathbf{h}(\mathbf{x})\|^2_{\mathbf{V}} \tag{7}$$

Pre-multiplying by $\mathbf{V}^{-1/2}$ to whiten the measurements and linearizing the measurement function around the best estimate, $\mathbf{x}^*$, yields:

$$\boldsymbol{\Delta}^* = \underset{\boldsymbol{\Delta}}{\operatorname{argmin}} \|\mathbf{A}\boldsymbol{\Delta} - \mathbf{b}\|^2 \tag{8}$$

where the new Jacobian and residual are respectively:

$$\mathbf{A} = \mathbf{V}^{-1/2} \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}^*} \quad \text{and} \quad \mathbf{b} = \mathbf{V}^{-1/2}(\mathbf{z} - \mathbf{h}(\mathbf{x}^*)) \tag{9}$$

Many publications, such as [10] and [11], explore efficient methods to solve (7). These solvers take advantage of $\mathbf{A}$'s sparse structure to efficiently solve a factorized version of the normal equations obtained from (8). Since this methodology is applied off-line, we are not concerned with the specific method used to solve the normal equations; it suffices to know that the path deviation is:

$$\boldsymbol{\Delta}^* = \boldsymbol{\Lambda}^{-1}\mathbf{A}^T\mathbf{b} \tag{10}$$

where $\boldsymbol{\Delta}^* = \mathbf{x} - \mathbf{x}^*$ and $\boldsymbol{\Lambda} = \mathbf{A}^T\mathbf{A}$ is the information matrix and the state estimate is updated as $\mathbf{x}^* := \mathbf{x}^* + \boldsymbol{\Delta}^*$. Once convergence is reached, the estimate error is defined as the difference between the true and estimated poses as:

$$\boldsymbol{\varepsilon} = \mathbf{x}^* - \mathbf{x} = \boldsymbol{\Lambda}^{-1}\mathbf{A}^T\mathbf{V}^{-1/2}(\mathbf{v} + \mathbf{f}) \tag{11}$$

Note that $\boldsymbol{\varepsilon}$ is a function of the measurement error, which includes both Gaussian noise and faults.

D. Fault Detector

As a sanity check, a chi-squared test on the norm of the residuals is commonly used, such that an alarm is triggered when the residuals' norm—which is the fault detector—surpasses a threshold defined to limit the occurrences of false alarms, i.e. when $q > T$. The fault detector is defined as:

$$q = \|\mathbf{b}\|^2 = \|\mathbf{v} + \mathbf{f}\|^2_{\mathbf{V}} \tag{12}$$

which is again a function of Gaussian noise and faults.

This section has presented the basic elements of fix-lag smoothing applied to localization applications. The next section defines Hazardous Misleading Information and presents a method to monitor integrity.

III. Integrity Monitoring

This work quantifies localization safety for a given trajectory prior to execution, which requires a predictive model that includes the environment map, waypoints defining the robot's trajectory, and the system model, which can be based on previously known sensor models and manufacturer specifications. Then, we compute an upper bound on the localization integrity risk, which is defined as the probability of Hazardous Misleading Information (HMI).

HMI occurs when the error in a state—or linear combination of states—of interest (e.g. lateral error in AV applications) is greater than a pre-defined alert limit ($\varepsilon > l$), and at the same time, the fault detector does not trigger and alarm ($q < T$). This is the especially dangerous situation in which the robot suffers a localization failure, but does not realize

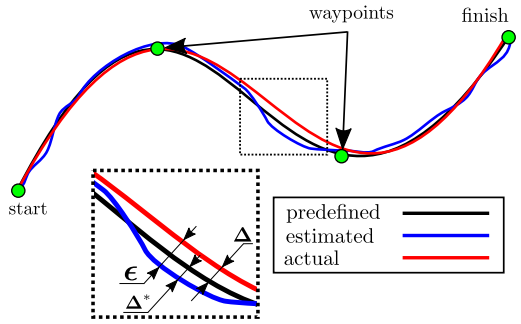

Fig. 1. Difference between the predefined, estimated, and actual trajectories. The error between the estimated and actual trajectories is $\boldsymbol{\varepsilon}$, which is the difference between the computed, $\boldsymbol{\Delta}^*$, and the actual, $\boldsymbol{\Delta}$, deviations from the predefined trajectory.

it and therefore does not initiate some emergency action. The probability of HMI, or integrity risk, is calculated under different fault modes, i.e. under the assumption that different measurements during the time window are faulted. Then, given a set of n_h fault modes, the integrity risk is:

$$P\big(\underbrace{\varepsilon > l \cap q < T}_{HMI}\big) = \sum_{h=1}^{n_h} P(HMI \mid h)P(h) \tag{13}$$

where the error in the state of interest, $\varepsilon = \mathbf{t}^T\boldsymbol{\varepsilon}$, is extracted from the full estimate error using vector $\mathbf{t}$, l is a predefined alert limit, and h indexes a fault mode, which may contain multiple faulted measurements. Determination of a mutually exclusive, jointly exhaustive set of fault modes and their corresponding probabilities, $P(h)$, is included in [18]. Direct computation of (13) is unfeasible. Fortunately, we can leverage classical GNSS-based integrity monitoring methods to compute an upper bound on the HMI probability and thus, guarantee localization safety.

Integrity monitoring is carried out in a three steps process that involves determining a predefined trajectory, computing the estimate error and fault detector distribution parameters, and calculating the HMI probability under the worst possible combination of faults.

A. Predefined Trajectory

To establish a plausible trajectory through the waypoints, the state evolution model in (4) is coupled with a control algorithm to generate a set of poses. Fig. 1 shows example predefined, estimated, and actual (unknown) trajectories. Since a better reference trajectory prior to the mission does not exist, we assume that deviations are calculated from the predefined trajectory. Thus, the results will only be valid if the predefined and actual trajectories are close with respect to system nonlinearities; otherwise, linearization errors might invalidate the results.

The next section determines the statistical distributions of the estimate error and fault detector for each state in the predefined trajectory.

B. Estimate Error & Fault Detector Distributions

The statistical distribution of ε and q are necessary to upper bound (13). [25] proved that the random parts of the

estimate error and fault detector for least squares estimators are independent. Thus, the summation terms in (13) become:

$$P(HMI \mid h) = P(\varepsilon > l \mid h) P(q < T \mid h) \tag{14}$$

where the estimate error and the fault detector are normally and chi-squared distributed, respectively, as [25]:

$$\begin{aligned} &\varepsilon \sim \mathbb{N}\left(\mathbf{\Lambda}^{-1}\mathbf{A}^T\breve{\mathbf{f}},\ \mathbf{\Lambda}^{-1}\right) \\ &q \sim \chi^2_{n-m,\lambda} \quad \text{where} \quad \lambda = \breve{\mathbf{f}}^T\left(\mathbf{I} - \mathbf{A}\mathbf{\Lambda}^{-1}\mathbf{A}^T\right)\breve{\mathbf{f}} \end{aligned} \tag{15}$$

Here, $\breve{\mathbf{f}} = \mathbf{V}^{-1/2}\mathbf{f}$ is the whitening fault and $\chi^2_{n-m,\lambda}$ denotes a non-central chi-squared distribution with $n-m$ degrees of freedom and non-centrality parameter λ, where m is the dimension of the state vector. The detection threshold is usually set to limit the frequency of false alarms, which occur when the the detector triggers the alarm under nominal (i.e. fault-free) operation. Thus, limiting the probability of false alarms to I_{FA}, the threshold is set to: $T = X^{-2}_{n-m}[1 - I_{FA}]$.

The only unknown in (15) is the fault vector that originates an estimate bias and the non-centrality of the detector. The next section uses (15) to evaluate (14) under the worst possible scenario.

C. Probability of HMI under Worst-case Fault

Faults are rarely occurring events that are not captured by the Gaussian noise assumption. Moreover, these are low frequency incidents that do not follow any clear pattern, which makes them difficult to model statistically. Here, faults are modeled as unknown deterministic quantities; therefore, integrity monitoring can be seen as an optimization problem of finding the worst-case fault that maximizes the HMI probability under each fault mode.

Integrity risk is evaluated under the worst-case fault for each fault mode, h, where each fault mode hypothesizes a different set of faulted measurements. Previous work derives methods to efficiently find such worst-case fault and determine the integrity risk [25]. First, the worst-case fault direction is analytically calculated as:

$$\breve{\mathbf{f}}_h^{direction} = \mathbf{E}_h^T\left[\mathbf{E}_h\left(\mathbf{I} - \mathbf{A}\ \mathbf{\Lambda}^{-1}\mathbf{A}^T\right)\mathbf{E}_h^T\right]^{-1}\mathbf{E}_h\mathbf{A}\ \mathbf{\Lambda}^{-1}\mathbf{t} \tag{16}$$

where $\mathbf{E}_h$ is composed of ones and zeros such that only the faulted components of $\mathbf{f}$—or equivalently $\breve{\mathbf{f}}$—are extracted. Then, using the fault direction in (16) to calculate the statistical distributions in (15), the integrity risk for each fault mode, $P(HMI \mid h)$, is obtained by maximizing (14) over the fault magnitude. Finally, every fault mode is weighted by its probability of occurrence and added in (13) to compute an upper bound on the integrity risk.

This section presented a methodology to quantify localization integrity for a given mission. In order to validate localization safety, the computed integrity risk is compared against an integrity requirement, such that only those areas with high enough integrity are validated.

TABLE I
SIMULATION PARAMETERS

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

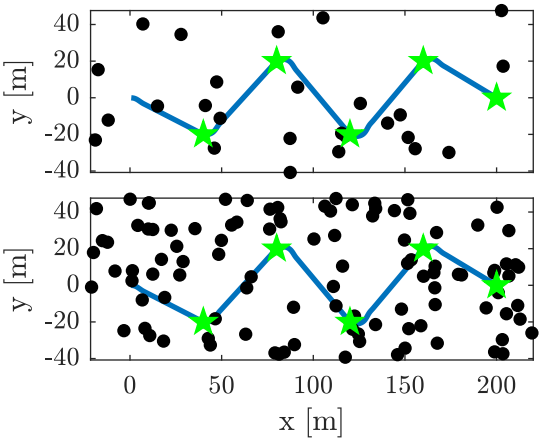

Fig. 2. Predefined trajectory and two randomly generated maps for two different landmark densities (0.001 landmarks/m^2 above and 0.005 landmarks/m^2 below). Note that the predefined trajectory remains the same.

IV. RESULTS

This section implements the proposed method to monitor integrity and validate localization safety. Simulations show that the integrity risk properly upper bounds the probability of HMI, and experimental results demonstrate the applicability of this method to a real system.

A. Simulation Results

The simulation depicts a robot traversing fixed waypoints in a 2D plane. Fig. 2 shows the predefined trajectory as a result of applying a simple steering angle controller to a constant-velocity bicycle model. The predefined trajectory, along with the specifications in Table I, remains constant for all simulations. However, the number and location of landmarks are modified such that ten landmark maps are randomly generated for each density ranging from 0.001 to 0.005 landmarks per square meter, exemplified in Fig. 2.

The relative measurements are the robot's linear and angular velocities as well as its steering angle. Absolute measurements are provided by range and bearing sensors. Relative measurements are assumed fault free, while absolute measurements have a fault probability of 10^{-3}. To maintain reasonable computational requirements, the time window length is continuously resized to preserve at least ten landmark detections. Extra epochs are removed from the time window. This technique maintains a more stable integrity risk than a fixed time window in number of epochs.

The results of the trajectory validation method for a lateral error of 0.5 m are given in Table II as localization availability, the percentage of the trajectory where the computed integrity risk is lower than the integrity requirement, I_{REQ}. Availability

TABLE II
AVAILABILITY

	Landmark Density ρ [m^{-2}]				
map #	0.001	0.002	0.003	0.004	0.005
1	32	78	91	100	97
2	63	83	97	100	97
3	59	77	98	90	92
4	53	73	97	98	96
5	68	89	97	99	100
6	30	82	93	97	99
7	80	92	92	90	100
8	55	89	88	97	100
9	58	78	96	99	95
10	49	84	95	100	100
average	55	83	94	97	98

Availability is the percentage of time $P(HMI)$ is lower than the integrity requirement, I_{REQ}, for different landmark density maps. Ten maps are randomly generated for each density.

increases as landmark density increases, sometimes reaching complete availability for maps with higher densities. In those 100% availability cases, localization safety is verified (for an integrity requirement of I_{REQ}), which means that a robot following such trajectory will only encounter HMI situations with a probability lower than I_{REQ}.

To validate the method, the actual mission is simulated 30 times for each generated map. In these simulations, all measurements are corrupted by Gaussian noise; however, only absolute measurements are corrupted by faults, which are randomly generated from a uniform distribution. Note that integrity risk is not monitored during these runs because only the fault detector and detector threshold are computed during the actual mission. Results show that HMI only occurs for landmark densities lower than 0.002 in regions where localization safety could not be guaranteed by the offline integrity monitoring method.

As an example, Fig. 3(top) shows the integrity risk for the predefined trajectory, while the middle and lower figures show the lateral error and detector for one of the missions shown in Fig. 2(top). The integrity risk is calculated only once, prior to the mission, and it applies to all subsequent mission executions as long as the mission model remains the same. However, Fig. 3(middle and bottom) are calculated for each mission execution. In this case, there are two HMI events shown in yellow bands—recall that HMI occurs when the lateral error surpasses the predefined alert limit and the fault detector stays below its threshold—that occur when the localization safety could not be guaranteed by the proposed method, i.e. when $P(HMI) > I_{REQ}$. Specifically, these HMI events occur when the computed upper bound on the integrity risk reaches one, which means that no safety bound can be guaranteed at those times.

Note that there are four measurement faults during this mission execution. The first three, occurring between 19 and 25 s, correspond to areas where safety is guaranteed and are detected as $q > T$ at those times. The forth fault occurs at time 30 s when the HMI probability is 2×10^{-4}. At this time, there is not enough measurement redundancy and the

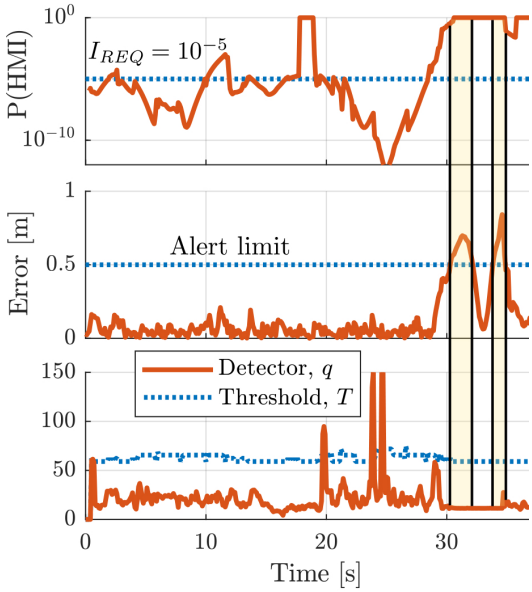

Fig. 3. Upper figure: upper bound on the integrity risk computed before the mission execution; the mission is depicted on the upper part of Fig. 2. Middle figure: predefined alert limit and absolute value of the lateral error for the example mission execution. Lower figure: fault detector and detector threshold for the example mission execution. Note that the detector is dimensionless.

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

fault is not detected; this fault is not damaging enough to immediately generate a localization failure, but it results in an estimate bias that, due to the lack of subsequent fault-free measurements, is propagated until HMI occurs.

In summary, simulation results demonstrate that integrity risk can reliably validate a mission's localization safety. The next section presents experimental results of applying this same methodology to a real system.

B. Experimental Results

In this section, we apply the integrity risk prediction methodology to validate safety using real-world data from an automobile on a college campus. The environment and the vehicle's sensors are shown in Fig. 4. Relative measurements are provided by a STIM-300 tactical grade Inertial Measurement Unit (IMU) at 125Hz. Light poles and tree trunk landmarks are extracted from two synchronized Velodyne VLP-16 lidar point clouds at 10Hz. Range and bearing

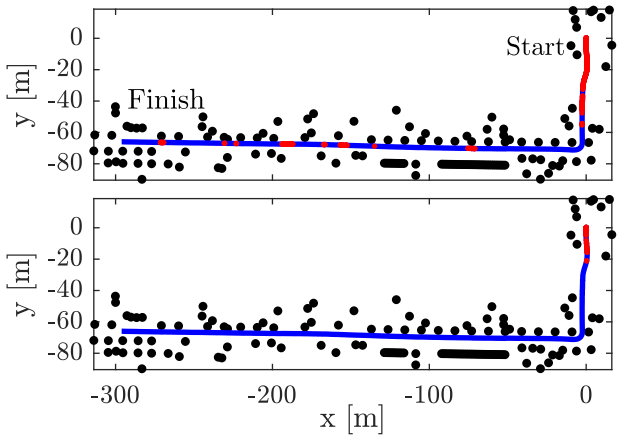

Fig. 5. Predefined trajectory and landmark map for the experiment. In red, the regions where localization safety cannot be guaranteed when defining an alert limit of 0.5m (above) and 1m (below).

estimates, together with differential GPS updates at 1Hz from a Novatal SPAN/CPT, compromise absolute measurements.

Fig. 5 shows the landmark map and estimated trajectory obtained via SLAM. In this experiment, the state vector contains 15 states: six for the 3D pose (position and orientation), three for the linear velocity, and six for the IMU biases (accelerometers and gyros). For integrity monitoring, we assume that only absolute measurements can be faulted and that both GPS and LiDAR measurements have a failure probability of 10^{-3}. The monitored state of interest is the lateral error—which is critical in AV applications—and the alert limit is set to 0.5 m. For better performance, the length of the time window is set at each epoch to maintain at least thirty absolute measurements.

The availability is measured for $I_{REQ} = 10^{-7}$, the same used by the FAA to prevent *extremely remote hazardous events*. Fig. 5 shows the trajectory regions where a lateral error of 0.5 m (top) and 1 m (bottom) cannot be guaranteed. Except for the problematic region around (-40,0), the trajectory is validated for a 1 m alert limit, which yields an overall availability of 98%. This contrasts with the 84% availability corresponding to the 0.5 m alert limit where multiple regions along the trajectory fail to present enough redundancy to guarantee safety against possible faults. The low integrity regions around (-40,0) is due to the lack of detectable landmarks, which makes localization rely heavily on GPS. In the absence of other redundant absolute measurements, a GPS fault can result in large errors, and thus, not even the 1 m alert limit can be guaranteed.

V. Conclusions and Future Work

This paper presents the first method to evaluate integrity risk localization safety for a pre-defined trajectory prior to executing the trajectory. The method also is formulated for localization via smoothing, which offers advantages over previous Kalman filter-based techniques.

The method has two main drawbacks: 1) the predefined trajectory must be close to the actual trajectory to avoid linearization errors, and 2) misdetection of landmarks due to occlusions during the mission might result in a worse than predicted localization performance and thus, the integrity bound might be invalidated. The former has not presented any complications in the experimental case and can be addressed by further validating similar trajectories to the original one. The latter can be the subject of future work where the misdetection probability is accounted for in the integrity risk calculation.

References

[1] M. Laris, "Waymo launches nation's first commercial self-driving taxi service in arizona," *The Washington Post*, December 2018.

[2] "Automated driving systems: A vision for safety 2.0," National Highway Traffic Safety Administration (NHTSA), Tech. Rep., Sept. 2017.

[3] "Preparing for the future of transportation: Automated vehicle 3.0," U.S. Department of Transportation, Tech. Rep., 2018.

[4] *ISO 26262:2011(en) Road vehicles — Functional safety*, International Standardization Organization Std.

[5] *ARP4754A Guidelines For Development Of Civil Aircraft and Systems*, Aerospace Recommended Practice Std.

[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, M. Jamoom, M. Spenko, and B. Pervan, "Integrity of laser-based feature extraction and data association," in *IEEE/ION PLANS*, April 2016, pp. 557–571.

[8] G. D. Arana, M. Joerger, and M. Spenko, "Local nearest neighbor integrity risk evaluation for robot navigation," in *ICRA*. IEEE, 2018, pp. 2328–2333.

[9] M. Joerger, G. D. Arana, M. Spenko, and B. Pervan, "A new approach to unwanted-object detection in gnss/lidar-based navigation," *Sensors*, vol. 18, no. 8, p. 2740, 2018.

[10] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, pp. 1–139, 2017.

[11] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *ICRA*. IEEE, 2011, pp. 3607–3613.

[12] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[13] S. Hewitson and J. Wang, "Extended receiver autonomous integrity monitoring (e raim) for gnss/ins integration," *Journal of Surveying Engineering*, vol. 136, no. 1, pp. 13–22, 2010.

[14] V. Kadirkamanathan, P. Li, M. H. Jaward, and S. G. Fabri, "Particle filtering-based fault detection in non-linear stochastic systems," *International Journal of Systems Science*, vol. 33, pp. 259–265, 2002.

[15] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[16] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, 2005.

[17] B. W. Parkinson and P. Axelrad, "Autonomous gps integrity monitoring using the pseudorange residual," *Navigation*, vol. 35, 1988.

[18] J. Blanch *et al.*, "Baseline advanced raim user algorithm and possible improvements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 713–732, January 2015.

[19] "Working group c - araim technical subgroup," EU-U.S. Cooperation on Satellite Navigation, Tech. Rep., 2016.

[20] M. Joerger *et al.*, "Landmark selection and unmapped obstacle detection in lidar-based navigation," in *ION GNSS+*, 2017.

[21] G. D. Arana, M. Joerger, and M. Spenko, "Efficient integrity monitoring for kf-based localization," *ICRA*, 2019.

[22] G. D. Arana, O. A. Hafez, M. Joerger, and M. Spenko, "Recursive integrity monitoring for mobile robot localization safety," *ICRA*, 2019.

[23] A. Hassani *et al.*, "Lidar data association risk reduction, using tight integration with ins," *ION GNSS+*, September 2018.

[24] O. A. Hafez, G. D. Arana, and M. Spenko, "Integrity risk-based model predictive control for mobile robots," *ICRA*, 2019.

[25] M. Joerger, F.-C. Chan, and B. Pervan, "Solution separation versus residual-based raim," *Navigation: Journal of The Institute of Navigation*, vol. 61, no. 4, pp. 273–291, 2014.