Landmark Augmentation for Mobile Robot Localization Safety

Yihe Chen, *Student Member, IEEE*, Osama Abdul Hafez, *Student Member, IEEE*, Boris Pervan, *Senior Member, IEEE*, and Matthew Spenko, *Senior Member, IEEE*

***Abstract*—As more robots, such as autonomous vehicles, are deployed in life-critical situations it is imperative to consider safety, and in particular, localization safety. While it would be ideal to guarantee localization safety without having to modify the environment, this is not always possible and one may have to add landmarks or active beacons for landmark-based localization. As such, this work introduces a method to identify the minimum places in an environment where landmarks can be added to ensure localization safety, as quantified using integrity risk, the probability of undetected sensor errors causing localization failure while accounting for measurement faults. The letter formulates the problem as a systematic minimization: given the robot's trajectory and the current landmark map, add the minimum number of new landmarks such that the integrity risk along the trajectory is below a given safety threshold. The letter proposes three algorithms: a naive approach, Integrity-based Landmark Generator (I-LaG), and Fast I-LaG. The computationally expensive naive algorithm serves as a reference to illustrate simple scenarios. I-LaG adds relatively fewer landmarks than the Fast I-Lag algorithm but is more computationally expensive. Simulation and experimental results validate the proposed algorithms.**

***Index Terms*—Autonomous vehicle navigation, automation technologies for smart cities, localization.**

I. Introduction

TO ENSURE widespread adoption of autonomous vehicles, safety must be thoroughly addressed beyond experimentation [1], which may require on the order of billions of test miles, [2]. Localization safety is one key component of overall vehicle safety [3]–[5] and can be measured as integrity risk, a measure of trust in a robot's sensors and localization algorithms. This metric has long been used in aviation [6]–[8] and has recently been extended to common mobile robot sensors, such as lidars [9]–[14]. The ideal would be to guarantee localization safety in any environment, but this is not always possible. Instead, it may be necessary to add landmarks when using landmark-based localization techniques. Rather than adding landmarks ad hoc, this paper introduces an automatic method to identify where to place the minimum number of landmarks, such that the desired minimum level of localization safety is guaranteed.

Manuscript received May 18, 2020; accepted September 28, 2020. Date of publication October 21, 2020; date of current version November 4, 2020. This letter was recommended for publication by Associate Editor Ashis Banerjee and Editor Dan Popa upon evaluation of the Reviewers' comments. This work was supported by NSF under Grant 1830642. *(Corresponding author: Yihe Chen.)*

The authors are with the Mechanical, Materials, and Aerospace Engineering Department, Illinois Tech, Chicago, IL 60616 USA (e-mail: ychen300@hawk.iit.edu; oabdulhafez@hawk.iit.edu; pervan@iit.edu; mspenko@iit.edu).

Digital Object Identifier 10.1109/LRA.2020.3032067

This is a new topic in robotics, and there is little prior related work. Most of the discussion on autonomous vehicle safety is on the vehicles themselves [15], [16]. One of the few papers to address landmark augmentation uses pseudolites to reinforce weak or non-existent Global Navigation Satellite System signals [17]. However, active pseudolites can be expensive. Instead, this paper focuses on landmarks that can be easily integrated into, and already readily populate, the urban landscape, such as trees, light-posts, and buildings.

Other work has proposed landmark selection algorithms for vision-based [18], [19], and lidar-based [20] navigation to improve localization accuracy and computational efficiency. However, such criteria do not address the case where available landmarks are insufficient for the desired localization safety.

This work formulates the landmark augmentation problem as a minimization of integrity risk, which is indirectly affected by the locations of new landmarks. We present three algorithms to solve the problem. The first is a naive approach that requires the fewest new landmarks; however, the algorithm is terribly computationally inefficient. The second, the Integrity-based Landmark Generator (I-LaG), may need more landmarks than the naive approach, but is more computationally efficient. The third, Fast I-LaG, may require the most new landmarks, but typically has the least computation time.

This work uses fixed-lag smoothing as a localizer [21], [22] and solution separation for integrity monitoring [12]. We assume that a precise landmark map exists, the robot's trajectory is known, and measurement errors are Gaussian with a non-zero mean (when faulted) and a known covariance.

The paper begins with a review of localization via fixed-lag smoothing that illustrates the quantification of localization safety using solution separation integrity monitoring method. Section III derives the integrity risk minimization problem and presents the three algorithms. Section IV shows simulation and experimental results. Section VI concludes the work and discusses future directions.

II. Background

This section provides the background to understand the integrity monitoring method used for the landmark augmentation algorithms described later. A description of fixed-lag smoothing localization is given, followed by the calculation of integrity risk using a solution separation fault detector.

A. Fixed-Lag Smoothing

This section begins with a description of fixed-lag smoothing's non-linear optimization problem. The resulting error in

mobile robot pose estimate is then expanded as a function of the errors in robot's sensor measurements.

Fixed-lag smoothing estimates the current pose, $\mathbf{x}_k \in \mathbb{R}^m$, by searching for the robot's states at each timestamp (epoch) within a preceding time window of size M that minimize the squared norm of the weighted measurement residual:

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \sum_{j=1}^{n} \|\mathbf{z}_j - \mathbf{h}_j(\mathbf{x})\|^2_{\mathbf{V}_j} \tag{1}$$

where $\mathbf{z}_j \in \mathbb{R}^{n_j}$ is the j^{th} measurement in the time window, $\mathbf{x} = [\mathbf{x}^{\mathbf{T}}_{k-M} \ \dots \ \mathbf{x}^T_{k-1} \ \mathbf{x}^T_k]^T$ is the robot's states within the time window, $\mathbf{h}_j(\cdot)$ is the j^{th} measurement observation function, and n is the total number of measurements within the window. Each measurement can be represented as:

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

where $\mathbf{v}_j \sim \mathbb{N}(\mathbf{0}, \mathbf{V}_j)$ is the Gaussian white noise of the j^{th} measurement with $\mathbf{V}_j$ as its covariance matrix, and $\mathbf{f}_j$ is the fault in the j^{th} measurement, such that $\mathbf{f}_j$ is a vector of zeros if the j^{th} measurement is non-faulted (see Section II-B in [12]). Measurement faults are rare, unknown deterministic errors that cannot be modeled using zero mean Gaussian white noise. In landmark-based navigation, $\mathbf{z}_j$ represents a feature's measurement extracted from a detected landmark and $\mathbf{h}_j(\cdot)$ relates robot states to a landmark's feature. Examples of a landmark's feature measurement fault include moving objects [20], [23] and data association errors [9], [24]. The non-linear optimization problem, in (1), can be expressed in batch form:

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \|\mathbf{z} - \mathbf{h}(\mathbf{x})\|^2_{\mathbf{V}} \tag{3}$$

where $\mathbf{z} \in \mathbb{R}^N$ is the measurement vector, $\mathbf{V} \in \mathbb{R}^{N \times N}$ is a block-diagonal matrix of the measurement noise covariance matrices, and $N = \sum_{j=1}^{n} n_j$ is the number of independent sensor measurements received within the time window. The optimization problem can now be solved by recursively linearizing the measurement function, $\mathbf{h}(\mathbf{x})$, e.g. using the Gauss-Newton algorithm. To define the pose estimate error, the measurement function, $\mathbf{h}(\mathbf{x})$, is linearized, after convergence, around the best estimate $\mathbf{x}^*$ (obtained in the optimization's last iteration):

$$\hat{\boldsymbol{\delta}} = \text{argmin}_{\boldsymbol{\delta}^*} \|\mathbf{z} - \mathbf{h}(\mathbf{x}^*) - \mathbf{H}\boldsymbol{\delta}^*\|^2_{\mathbf{V}} \tag{4}$$

where $\boldsymbol{\delta}^* = \mathbf{x} - \mathbf{x}^*$, and $\mathbf{H} = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}^*}$ is the Jacobian matrix of the measurement function. By defining $\mathbf{A} = \mathbf{V}^{-\frac{1}{2}}\mathbf{H}$ as the standardized measurement matrix and $\mathbf{b} = \mathbf{V}^{-\frac{1}{2}}(\mathbf{z} - \mathbf{h}(\mathbf{x}^*))$ as the standardized residual vector, the pose estimate error, in (4), can be rewritten in the general least squares form:

$$\hat{\boldsymbol{\delta}} = \text{argmin}_{\boldsymbol{\delta}^*} \|\mathbf{A}\boldsymbol{\delta}^* - \mathbf{b}\|^2 = \boldsymbol{\Lambda}^{-1}\mathbf{A}^T\mathbf{b} \tag{5}$$

where $\hat{\boldsymbol{\delta}} = \hat{\mathbf{x}} - \mathbf{x}$ is the robot's pose estimate error, $\boldsymbol{\Lambda} = \mathbf{A}^T\mathbf{A}$ is the information matrix of the pose estimate. By substituting (2) in (5), the error in the robot's pose estimate can be expressed as a function of measurement noises, $\mathbf{v} \in \mathbb{R}^N$, and faults, $\mathbf{f} \in \mathbb{R}^N$, in the preceding time window:

$$\hat{\boldsymbol{\delta}} = \boldsymbol{\Lambda}^{-1}\mathbf{A}^T\mathbf{V}^{-1/2}(\mathbf{v} + \mathbf{f}) \sim \mathbb{N}\left(\boldsymbol{\Lambda}^{-1}\mathbf{A}^T\mathbf{V}^{-1/2}\mathbf{f}, \boldsymbol{\Lambda}^{-1}\right) \tag{6}$$

Note, the mean of the estimate error is affected by the (unknown) measurement faults. Next, solution separation integrity monitoring for landmark-based localization will be presented.

B. Solution Separation Integrity Monitoring

This section reviews fixed-lag smoothing-based integrity monitoring using a solution separation fault detector. Integrity risk is the probability of Hazardous Misleading Information (HMI). HMI occurs when the estimate error in the state of interest (e.g. lateral error for autonomous vehicles) exceeds a predefined threshold or *alert limit*, and the fault detector, a statistical measure of measurement inconsistency, does not trigger an alarm [25]. HMI is given as:

$$HMI = \boldsymbol{\alpha}^T\hat{\boldsymbol{\delta}} > l \bigcap_{i=1}^{n_H} \Delta_i \leq T_{\Delta_i} \tag{7}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{(M+1)m}$ is the state-of-interest extraction vector; l is the alert limit; n_H is the number of fault hypotheses; and $\Delta_i, \forall i = 1, ..., n_H$ are a set of statistics that represent the solution separation fault detector such that it triggers an alarm when at least one of the statistics' magnitude exceeds its predefined threshold, T_{Δ_i}, defined as follows [12]:

$$\Delta_i = \boldsymbol{\alpha}^T(\hat{\mathbf{x}} - \hat{\mathbf{x}}_i), \quad T_{\Delta_i} = \Phi^{-1}\left[1 - \frac{I_{FA}}{2n_H}\right]\sqrt{\boldsymbol{\alpha}^T\boldsymbol{\Lambda}^{-1}_{\Delta_i}\boldsymbol{\alpha}} \tag{8}$$

where $\hat{\mathbf{x}}$ is the state estimated using all measurements within the time window; $\hat{\mathbf{x}}_i$ is the state estimated using only the non-faulted measurements as determined by the i^{th} hypothesis; $\Phi^{-1}[.]$ is the inverse Cumulative Distribution Function (CDF) for the standard normal random variable; I_{FA} is a predefined allocation that represents the desired upper-bound on the probability of false alarms; and $\boldsymbol{\Lambda}_{\Delta_i}$ is the information matrix for the i^{th} statistic of the solution separation fault detector, which can be expanded as:

$$\boldsymbol{\Lambda}^{-1}_{\Delta_i} = \boldsymbol{\Lambda}^{-1}_i - \boldsymbol{\Lambda}^{-1} \tag{9}$$

where $\boldsymbol{\Lambda}_i$ is the information matrix of the state estimate, $\hat{\mathbf{x}}_i$, obtained using only the fault-free measurements of the i^{th} hypothesis, and $\boldsymbol{\Lambda}$ is the information matrix of the state estimate, $\hat{\mathbf{x}}$, determined using all of the measurements in the time window (see Appendix B of [26] for proof).

Since the fault detector and the state-of-interest estimate error are both affected by measurement faults occurring within the time window, integrity risk, or the probability of HMI, $P(HMI)$, is quantified under a set of mutually exclusive, collectively exhaustive fault hypotheses, $H_i, \forall i \in \{0, ..., n_H\}$, that define the set of faulted and non-faulted measurements:

$$P(HMI) = \sum_{i=0}^{n_H} P(HMI|H_i)\,P(H_i) \tag{10}$$

where H_0 is the fault-free hypothesis, $P(HMI|H_i)$ is the conditional integrity risk for the i^{th} fault hypothesis, and $P(H_i)$ is the probability of the i^{th} hypothesis. Section III in [10] presents a method to evaluate $P(H_i)$ given the probability of failure for each measurement and Section V-A in [12] shows that the conditional integrity risk for the i^{th} hypothesis, $P(HMI|H_i)$, can be upper-bounded by:

$$P(HMI|H_i) \leq 2\Phi\left[\frac{T_{\Delta_i} - l}{\sqrt{\boldsymbol{\alpha}^T\boldsymbol{\Lambda}^{-1}_i\boldsymbol{\alpha}}}\right] \tag{11}$$

The next section will introduce algorithms that guarantee a minimum level of localization safety by modifying the landmark

map, such that the integrity risk, $P(HMI)$, always lies below a predefined safety requirement.

III. Landmark Augmentation Derivation and Implementation

This section introduces the proposed landmark augmentation algorithms that identify the locations of new landmarks, if needed, to satisfy the localization safety requirement.

A. Derivation

This subsection expresses the objective function, namely the integrity risk, as a function of new landmarks' locations. To do so, the impact of new landmarks on the conditional integrity risk defined in (11) will be isolated so that any minimization algorithm for the integrity risk in (10) does not need to re-address the impact of landmarks that already exist. This is done by augmenting the information matrices, the only factors that are impacted by landmarks, in (11). Substituting the i^{th} statistic's threshold T_{Δ_i} defined by (8) in (11) yields:

$$P\left(HMI|H_i\right) \leq 2\Phi\left[\frac{\Phi^{-1}\left[1-\frac{I_{FA}}{2n_H}\right]\sqrt{\boldsymbol{\alpha}^T\boldsymbol{\Lambda}_{\Delta_i}^{-1}\boldsymbol{\alpha}}-l}{\sqrt{\boldsymbol{\alpha}^T\boldsymbol{\Lambda}_i^{-1}\boldsymbol{\alpha}}}\right] \tag{12}$$

where the information matrices $\boldsymbol{\Lambda}_{\Delta_i}$ and $\boldsymbol{\Lambda}_i$, described in (9), will be expanded to illustrate the effect of the measurements coming from the new landmarks to be added in contrast with the existing measurements on the integrity risk bound.

$\boldsymbol{\Lambda}_i$ can be expanded into $\boldsymbol{\Lambda}_i = \mathbf{A}^T\mathbf{B}_i^T\mathbf{B}_i\mathbf{A}$ where $\mathbf{B}_i$ is the non-faulted measurement extraction matrix for the i^{th} hypothesis [12]. $\mathbf{B}_i\mathbf{A}$ can be expanded into $\mathbf{B}_i\mathbf{V}^{-\frac{1}{2}}\mathbf{H}$ such that $\boldsymbol{\Lambda}_i = \mathbf{H}^T\mathbf{V}^{-\frac{1}{2}}\mathbf{B}_i^T\mathbf{B}_i\mathbf{V}^{-\frac{1}{2}}\mathbf{H}$. The $n \times (M+1)m$ measurement matrix, $\mathbf{H}$, is a list of measurement models comprised of the prior measurements from the fixed-lag smoothing estimate at the previous epoch [12], state evolution measurements (e.g. odometery, IMU), and absolute measurements (e.g. feature measurements extracted from the old and new landmarks). Accordingly, the measurement matrix is augmented into $\mathbf{H} = [\mathbf{H_o^T}\ \mathbf{H}_n^T]^T$ where $\mathbf{H}_n$ is unknown since it is a function of the new landmarks to be added, and $\mathbf{H}_o$ is known since it is a function of the existing landmarks and the rest of existing measurements. Subsequently, the measurement covariance matrix, $\mathbf{V}$, can be augmented as follows:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V_o} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_n \end{bmatrix} \tag{13}$$

Therefore:

$$\boldsymbol{\Lambda}_i = \begin{bmatrix} \mathbf{H_o^T}\mathbf{V_o^{-\frac{1}{2}}} & \mathbf{H}_n^T\mathbf{V}_n^{-\frac{1}{2}} \end{bmatrix}\mathbf{B}_i^T\mathbf{B}_i\begin{bmatrix} \mathbf{H_o^T}\mathbf{V_o^{-\frac{1}{2}}} & \mathbf{H}_n^T\mathbf{V}_n^{-\frac{1}{2}} \end{bmatrix}^T \tag{14}$$

$\boldsymbol{\Lambda}_{\Delta_i}^{-1}$ was expanded in (9), and thus we only need to show how $\boldsymbol{\Lambda}$ is affected by the new landmarks. To that end, $\boldsymbol{\Lambda} = \mathbf{A}^T\mathbf{A} \Rightarrow \mathbf{H}^T\mathbf{V}^{-\frac{1}{2}}\mathbf{V}^{-\frac{1}{2}}\mathbf{H} = \mathbf{H}^T\mathbf{V}^{-1}\mathbf{H}$ such that:

$$\begin{aligned} \boldsymbol{\Lambda} &= \begin{bmatrix} \mathbf{H_o} \\ \mathbf{H}_n \end{bmatrix}^T \mathbf{V}^{-1} \begin{bmatrix} \mathbf{H_o} \\ \mathbf{H}_n \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{H_o} \\ \mathbf{H}_n \end{bmatrix}^T \begin{bmatrix} \mathbf{V_o^{-1}} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_n^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{H_o} \\ \mathbf{H}_n \end{bmatrix} \\ &= \mathbf{H}_o^T\mathbf{V}_o^{-1}\mathbf{H}_o + \mathbf{H}_n^T\mathbf{V}_n^{-1}\mathbf{H}_n \end{aligned} \tag{15}$$

Substituting (9), (14) and (15) in (12), and then the resulting $P(HMI|H_i)$ back in the integrity risk, defined in (10), yields (16), shown at the bottom of the page, where $\|\mathbf{Q}\|^{-2} = (\mathbf{Q}^T\mathbf{Q})^{-1}$ is the inverse of the squared norm. Equation (16) represents the objective function at a given epoch along the robot's trajectory as a function of new landmark locations, reflected by $\mathbf{H}_n$ and $\mathbf{V}_n$, that is to be minimized such that the desired level of localization safety is met at each epoch along the robot's trajectory. The next subsections will present the details of the proposed minimization problems.

B. A Naive Approach

The most straightforward solution to this problem would be to minimize the sum of the integrity risk along the entire robot trajectory over all potential landmark locations. This begins by minimizing the sum of the integrity risk with one additional new landmark. If the resulting $P(HMI)$ at each epoch along the trajectory lies under the predefined safety threshold, the process is complete. If not, the minimization repeats with two landmarks, and so on, until the integrity risk requirement is met.

This "naive" algorithm is given in Algorithm 1 where $minimize\_sum\_of\_integrity\_risk(lm\_number)$ is given as:

$$\begin{cases} \min_{LMs} \sum_{epoch=1}^{final\_epoch} P(HMI)_{epoch}, \text{where } |LMs| \\ \quad = lm\_number \\ \text{such that } P(HMI)_{epoch} \leq safety\_threshold, \ \forall epoch \\ \text{such that } LMs \in operation\_area \end{cases} \tag{17}$$

and $minimize\_integrity\_risk(lm\_number, epoch)$ is defined as:

$$\begin{cases} \min_{LMs} P(HMI)_{epoch}, \text{where } |LMs| = lm\_number \\ \text{such that } P(HMI)_{epoch} \leq safety\_threshold \\ \text{such that } LMs \in operation\_area \cap FoV_{epoch} \end{cases} \tag{18}$$

where LMs are the set of landmarks to be added with lm_number as their total number, FoV_{epoch} is the robot's field of view at a given epoch, and $operation\_area$ is the location where new landmarks can be placed. These minimization

$$P(HMI) \leq 2\sum_{i=0}^{n_H} P(H_i)\Phi\left[\frac{\Phi^{-1}\left[1-\frac{I_{FA}}{2n_H}\right]\sqrt{\boldsymbol{\alpha}^T\left(\left\|\mathbf{B}_i\begin{bmatrix}\mathbf{V_o^{-\frac{1}{2}}}\mathbf{H_o}\\ \mathbf{V}_n^{-\frac{1}{2}}\mathbf{H}_n\end{bmatrix}\right\|^{-2} - \left\|\begin{bmatrix}\mathbf{V_o^{-\frac{1}{2}}}\mathbf{H_o}\\ \mathbf{V}_n^{-\frac{1}{2}}\mathbf{H}_n\end{bmatrix}\right\|^{-2}\right)\boldsymbol{\alpha}} - l}{\sqrt{\boldsymbol{\alpha}^T\left\|\mathbf{B}_i\begin{bmatrix}\mathbf{V_o^{-\frac{1}{2}}}\mathbf{H_o}\\ \mathbf{V}_n^{-\frac{1}{2}}\mathbf{H}_n\end{bmatrix}\right\|^{-2}\boldsymbol{\alpha}}}\right] \tag{16}$$

Algorithm 1: Naive Approach.

Given: $original\_map$, $\mathbf{x}_{1:final\_epoch}$, $safety\_threshold$, n_{max}
$MAP \leftarrow original\_map$; $lm\_number \leftarrow 0$
while $lm\_number \leq n_{max}$ **do**
  for $epoch = 1 : final\_epoch$ **do**
    $integrity\_above\_threshold(epoch) \leftarrow$
     $eval\_integrity\_risk(epoch) -$
     $safety\_threshold > 0$
  end for
  if $\exists_{epoch} integrity\_above\_threshold(epoch)$ **then**
    $++ lm\_number$
    $[\sum_{epoch=1}^{final\_epoch} P(HMI)_{epoch}, LMs] \leftarrow$
     $minimize\_sum\_of\_integrity\_risk(lm\_number)$
  else
    $MAP = [MAP; LMs]$
    break
  end if
end while=0

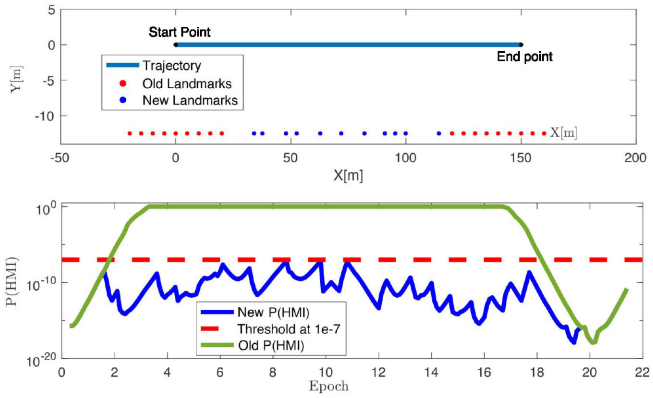

Fig. 1. Top—Simulation of a robot moving left to right in a straight line. Red dots indicate pre-existing landmarks. Blue dots indicate new landmark locations proposed by the naive algorithm. Bottom—The original $P(HMI)$ curve (green) and the $P(HMI)$ curve after adding the naive algorithm's landmarks (blue). The red dashed line indicates the safety threshold.

problems can be solved using any optimizer that can handle non-linear constraints—we use the interior point method.

This solution may add the fewest landmarks compared to the I-LaG and Fast I-LaG algorithms shown later, but it is computationally expensive because adding a new landmark requires the algorithm to evaluate integrity risk over the entire robot trajectory at each iteration of the algorithm.

Fig. 1 (top) shows a simple simulation that highlights this point. A robot traverses a straight 150 m path from left to right. Table I gives the simulation parameters. Landmarks are evenly distributed along the right side of the path at $y = -12.5$ m from $x = -25$ m to $x = 175$ m in 5 m increments. However, no landmarks exist between $x = 25$ m and $x = 125$ m.

Fig. 1 (bottom) shows the integrity risk in green, which starts small, increases in the middle where the landmarks are nonexistent, and then drops again at the end of the path when additional landmarks are identified by the robot. The naive algorithm places 11 new landmarks as shown in blue in Fig. 1 (top). The resulting

TABLE I
SIMULATION PARAMETERS

Velocity	$25\,\mathrm{km\,h^{-1}}$	$\sigma_{velocity}$	$1\,\mathrm{m\,s^{-1}}$
Time step	0.1 s	σ_{gyro}	$2^\circ\,\mathrm{s^{-1}}$
Sensor range	25 m	σ_{lidar}	0.2 m
Alert limit	0.5 m	$\sigma_{steering\ angle}$	2° +
Fault probability	10^{-3}	Width of LM belt	5 m
Total lane width	15 m	$safety\_threshold$	10^{-7}

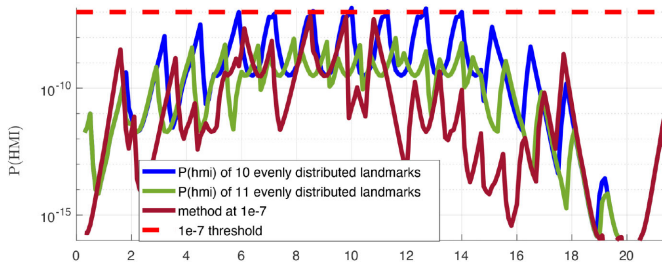

Fig. 2. $P(HMI)$ generated by the naive approach compared to the one given by evenly spaced landmarks on $y = -12.5$ m. The scarlet curve shows that the 11 landmarks placed by the naive approach guarantees that the $P(HMI)$ stays below the threshold as do the 11 evenly placed landmarks (green curve). However, the ten evenly placed landmarks yield a $P(HMI)$ curve (blue) that exceeds the safety threshold.

integrity risk at all epochs lies below the $1e^{-}7$ threshold as shown in Fig. 1 (bottom).

As a control, the integrity risk was calculated as if the gap in the landmarks between $x = 20$ m to $x = 120$ m was filled with evenly spaced landmarks. Fig. 2 shows that when placing ten landmarks, $P(HMI)$ exceeds the threshold, but 11 landmarks keep $P(HMI)$ below the threshold. This indicates that the naive approach generates reasonable landmark locations.

This solution, however, is extremely computationally expensive. This *highly* simplified and constrained scenario takes approximately 7 h to finish on a quad-core Intel Core i5 8259U microprocessor. The computation time would be considerably longer if the landmarks were not constrained to lie on a single y-value. Thus, a faster solution is needed.

C. I-LaG and Fast I-LaG

To address computation time, here we present the Integrity based Landmark Generator (I-LaG) algorithm and the even faster Fast I-LaG.

The I-LaG algorithm (see Algorithm 2) is given as follows:

1) Starting at the beginning of the trajectory, calculate $P(HMI)_{epoch}$ as the robot travels along the trajectory.
2) When $P(HMI)_{epoch}$ exceeds the safety threshold at a given epoch, minimize the integrity risk at only that epoch by adding one landmark to the environment.
3) If a location is found that reduces $P(HMI)_{epoch}$ to below the safety threshold, update the map and go to step 1.
4) If $P(HMI)_{epoch}$ is not reduced to below the given safety threshold, go to step 2, while increasing the number of new landmarks to be added in the minimization process.

When a landmark is added, the algorithm re-evaluates the integrity risk for the entire path. This is important because even though the newly added landmark is meant to reduce the integrity risk at a certain epoch, the landmark will likely be seen by the

Algorithm 2: I-LaG.

Given: $original\_map$, $\mathbf{x}_{1:final\_epoch}$, $safety\_threshold$, n_{max}
$MAP \leftarrow original\_map$; $epoch \leftarrow 1$
while $epoch \leq final\_epoch$
  $P(HMI)_{epoch} \leftarrow eval\_integrity\_risk(epoch)$
  if $P(HMI)_{epoch} > safety\_threshold$
    for $lm\_number = 1 : n_{max}$ **do**
      $[P(HMI)_{epoch}, LMs] \leftarrow$
       $minimize\_integrity\_risk(lm\_number,\ epoch)$
      if $P(HMI)_{epoch} \leq safety\_threshold$ **then**
        $MAP = [MAP; LMs]$
        $epoch \leftarrow 1$
        $break$
      end if
    end for
  else
    $++epoch$
  end if
end while=0

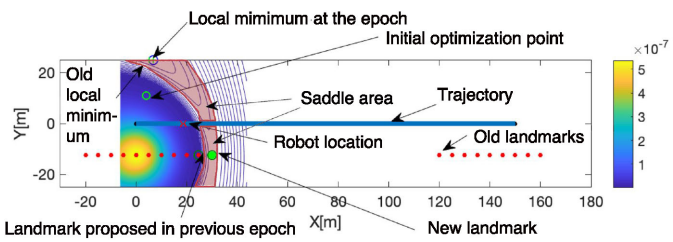

Fig. 3. I-LaG visualization. The simulation the same as in Fig. 1 (top). The contour represents the integrity risk if a new landmark is added at that location. The minimum of the contour (shaded red) is located at approximately (5,22), but since the simulation constrains the landmarks to lie on y =-12.5m, the new landmark is added on the green dot.

robot prior to that epoch, which will most likely reduce integrity risk further. In contrast, with Fast I-LaG (see Algorithm 3) the algorithm does not return to the first epoch when new landmarks are added, but instead proceeds.

Fig. 3 illustrates I-LaG algorithm. In this simplified scenario, the same as the one shown before, the robot has travelled 22m. At that point, it recognizes that its integrity risk has exceeded the threshold. The contour plot indicates the $P(HMI)$ corresponding to a landmark being placed at that location. The algorithm proposes a new landmark at the global minimum, approximately (4, 22.5). However, since the landmarks are constrained to lie on the y =-12.5m line, the algorithm chooses a new landmark location at the green dot.

If the $P(HMI)$ with this added landmark lies below the safety threshold, the algorithm continues; otherwise, the algorithm repeats the minimization with two additional landmarks, and so on, until the integrity risk is either below the safety threshold or a predefined maximum number of landmarks, n_{max}, has been attempted. The I-LaG algorithm takes approximately 3min to finish while the Fast I-LaG algorithm needs approximately 1.3min. A more detailed comparison between these two algorithms will be provided in the next section.

Algorithm 3: Fast I-LaG.

Given: $original\_map$, $\mathbf{x}_{1:final\_epoch}$, $safety\_threshold$, n_{max}
$MAP \leftarrow original\_map$; $epoch \leftarrow 1$
while $epoch \leq final\_epoch$ **do**
  $P(HMI)_{epoch} \leftarrow eval\_integrity\_risk(epoch)$
  if $P(HMI)_{epoch} > safety\_threshold$ **then**
    for $lm\_number = 1 : n_{max}$ **do**
      $[P(HMI)_{epoch}, LMs] \leftarrow$
       $minimize\_integrity\_risk(lm\_number,\ epoch)$
      if $P(HMI)_{epoch} \leq safety\_threshold$ **then**
        $MAP = [MAP; LMs]$
        break
      end if
    end for
  end if
  $++epoch$
end while=0

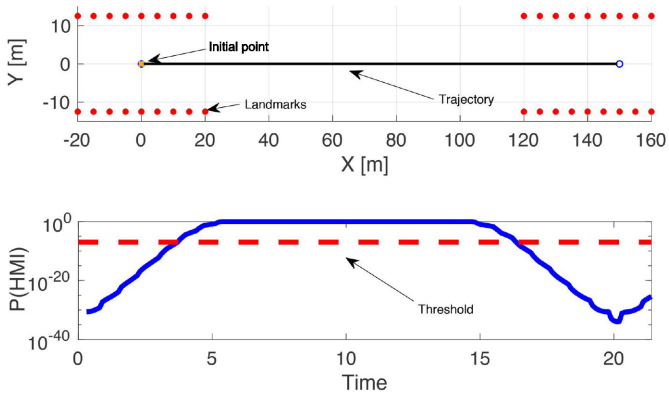

Fig. 4. The simulation environment (Top) and its $P(HMI)$ curve (Bottom).

IV. Simulation Results

This section shows I-LaG and Fast I-LaG simulation results for a mobile robot moving in an environment that mimics a roadway with landmarks on the left and right side of the street.

Fig. 4 (top) shows the simulation environment, and Fig. 4 (bottom) shows the resulting $P(HMI)$ curve, which exceeded the safety threshold. Table I gives the simulation parameters. In the simulation, a constant-velocity mobile robot travels in a straight line through a landmark-rich environment, traverses a stretch with no landmarks, and then returns to a landmark-rich environment. New landmarks are allowed to be placed in two 5m wide strips, representing the parkways on either side of a street, starting from x =20m to x =120m on the both sides of the 15m wide road. Absolute measurements, range and bearing to mapped landmarks, can be faulted with a probability of 10^{-3}. Relative measurements, steering angle and wheel velocity, are assumed to be fault-free [27].

Fig. 5 shows the resulting new landmark locations for each algorithm (I-LaG, bottom and Fast I-LaG, top). I-LaG produced ten new landmark locations; Fast I-LaG produced 12. The fact that Fast I-LaG proposed more landmarks is expected since the newly added landmarks, in the Fast I-LaG algorithm, are not being seen by any of the previous epochs when calculating the $P(HMI)$; thus, as shown in Fig. 6, the $P(HMI)$ value

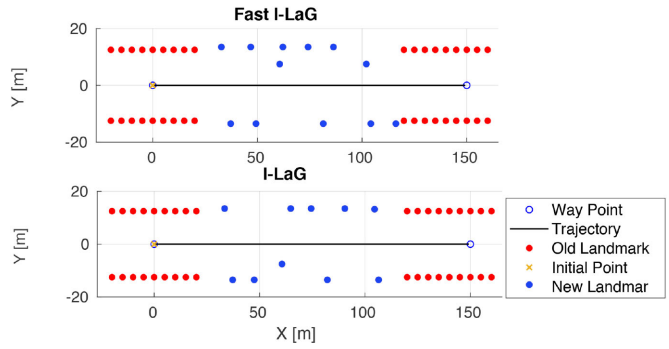

Fig. 5. Simulation results for the Fast I-LaG (12 additional landmarks) and I-Lag algorithms (ten additional landmarks).

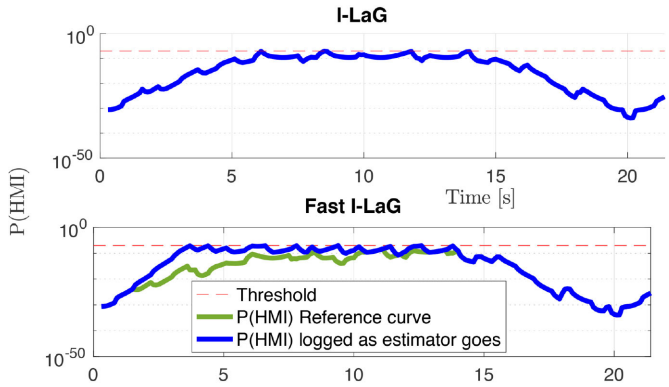

Fig. 6. A $P(HMI)$ comparison between I-LaG and Fast I-LaG. For I-LaG, the algorithm returns to the beginning of the trajectory after new landmarks are proposed. Newly added landmarks were not observed in past epochs when running Fast I-LaG algorithm; thus, the integrity risk curve generated by the Fast I-LaG algorithm is often higher than the actual integrity risk if evaluated from the start of the trajectory.

for the Fast I-LaG algorithm is higher than the actual value. Had integrity risk been calculated from the beginning of the trajectory, the integrity risk would be lower since additional landmarks would be sensed by the robot prior to the epoch in which they were added. Because Fast I-LaG does not need to start from the beginning of the trajectory each time a new landmark is added, it takes less time to finish.

For this simulation the I-LaG method completed in approximately 5.5min while Fast I-LaG stopped in 4.3min. For longer path segments, Fast I-LaG may consume significantly less time at the cost of more landmarks. While computation time may not be an issue in most cases, it could be helpful for some military or natural disaster scenarios.

V. Experimental Results

The experimental results highlight the I-Lag and Fast I-Lag algorithms for a car driving in a university campus.

A. Setup

Fig. 8 shows the sensor suite used for collecting experimental data. The sensor suite includes a Novatel SPAN-CPT GPS, a tactical grade STIM-300 IMU, and one Ouster OS-1 64-line lidar. The testing environment consisted of Illinois Tech's main campus in Chicago, IL USA (see Fig. 9). This allowed us to

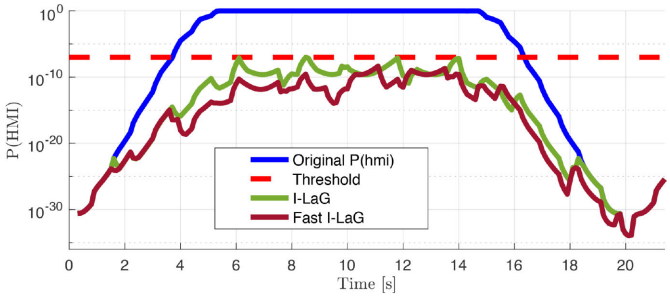

Fig. 7. The comparison between the $P(HMI)$ curves. The Fast I-LaG algorithm yields a lower integrity risk than the I-LaG algorithm due the the fact that the Fast I-Lag algorithm added more landmarks.

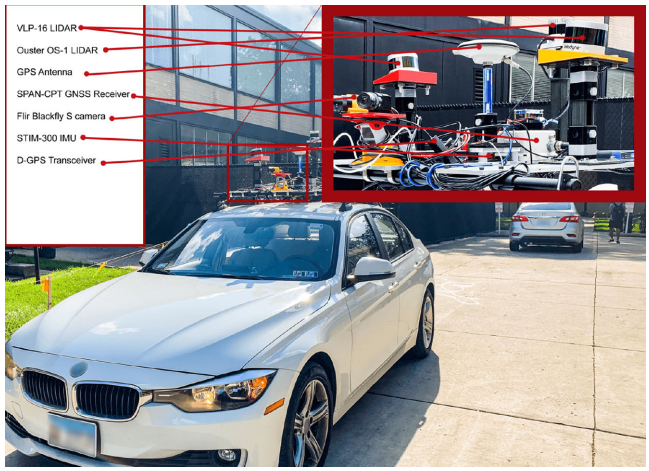

Fig. 8. The sensor suite consists of a STIM-300 tactical-grade IMU, two Velodyne VLP-16 lidars (not used in the experiments described in this paper), one ouster OS-1 64 beam lidar, and a Novatel SPAN-CPT DGPS.

Fig. 9. The experimental environment. No landmarks can be placed in the roadway, highlighted red. The green areas indicate locations where landmarks can be placed. Purple shows extracted features from lidar point cloud.

place new landmarks (concrete filler tubes) in the environment without disrupting the public streetscape.

Pole-like objects such as lamp posts, tree trunks, road signs, and parts of buildings are extracted from the lidar point cloud (see shaded purple in Fig. 9). The vehicle's trajectory is estimated using EKF-SLAM (see Fig. 10 and Table II).

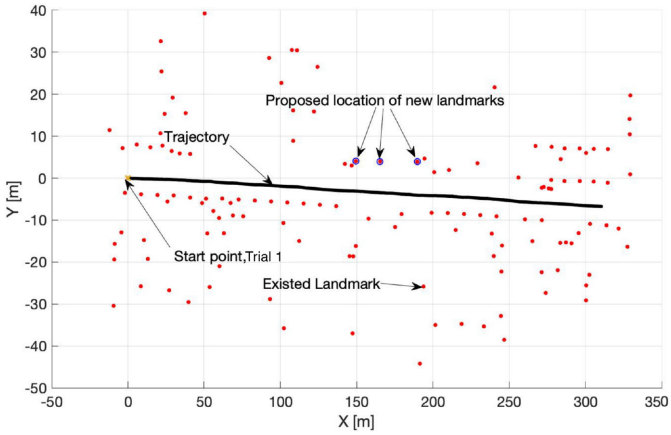

Fig. 10. This figure shows new poles that were placed in the actual scenario.

TABLE II
CONSTRAINT AND EXPERIMENTAL PARAMETERS

Lidar range	25 m	lane width	13 m
op. area width	7.5 m	supplemental area	$\emptyset$
σ_{lidar}	0.2 m	Alert limit	0.5 m
$\text{PSD}_{AccelNoise}$	$0.002\ \text{m}^2\ \text{s}^{-5}$	safety threshold	10^{-7}
$\text{PSD}_{GyroNoise}$	$3.05\ \mu\text{rad}^2\ \text{s}^{-3}$	frequency_{Dgps}	1 Hz
$\text{PSD}_{AccelBias}$	$0.24\ \mu\text{m}^2\ \text{s}^{-6}$	$\text{PSD}_{GyroBias}$	$2.12\ \text{prad}^2\ \text{s}^{-4}$
frequency_{IMU}	125 Hz		

Fig. 11. The vehicle path, existing landmarks, and proposed landmarks.

B. Experimental Procedure

Two trials were conducted between which the environment was modified to include new landmarks as suggested by the I-LaG algorithm. The experimental procedure is as follows:

1) Commence Trial 1, collect LIDAR and IMU data.
2) Run EKF-SLAM to estimate the vehicle's trajectory along with the landmark map from the data.
3) Evaluate the integrity risk along the vehicle's trajectory.
4) Run I-LaG to identify the locations of new landmarks.
5) Place the proposed new landmarks in the environment using a total station theodolite as shown in Fig. 11.
6) Begin Trial 2 and collect data while following as similar trajectory as possible to the Trial 1.
7) Run EKF-SLAM to estimate the vehicle's trajectory along with the landmark map of Trial 2, compare with the map gained in Trial 1, and verify that the newly added landmarks were placed in the locations proposed by the I-LaG algorithm, as shown in Fig. 14.
8) Evaluate the integrity risk of the second trial's trajectory, as shown in Fig. 15.

Regarding step 8, comparing the integrity risk curves for the two trials, note that there is some natural sensitivity of the integrity risk due to differences in the initial conditions (e.g. different GPS satellites are available at different testing times, slightly different starting locations), vehicle velocity, and the exact path taken. To highlight the differences in the integrity risk prior to presenting the experimental results, Fig. 12 shows three

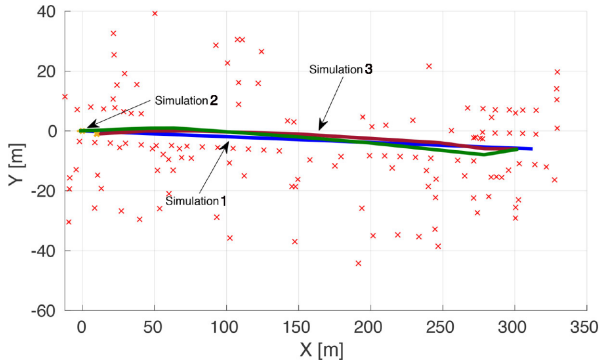

Fig. 12. Three simulated trajectories in a landmark map generated from experimental data.

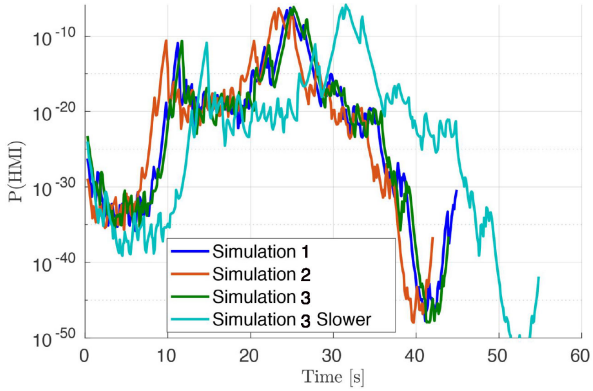

Fig. 13. Integrity risk for the trials described in Fig. 12. Note the differences in phase and magnitude.

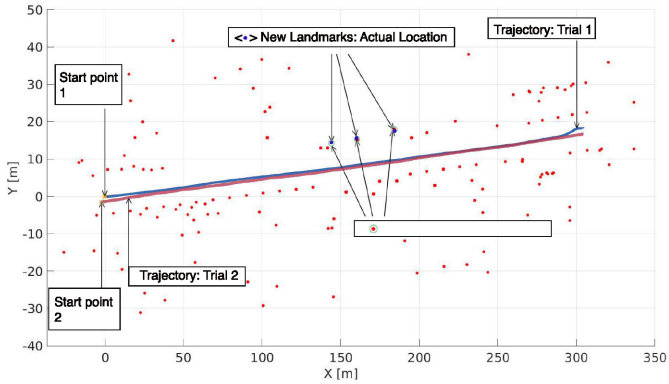

Fig. 14. The location of the new poles in the experiment.

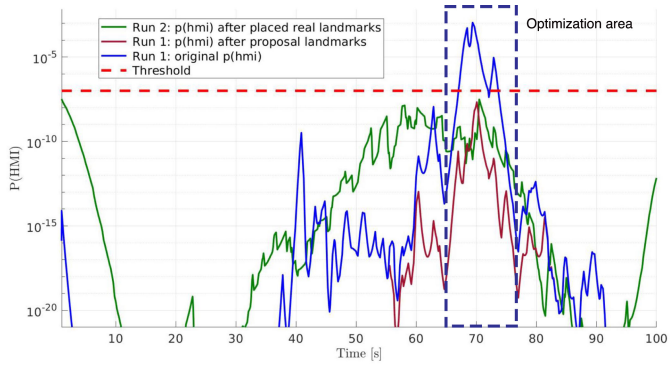

Fig. 15. The original integrity risk before running the I-LaG algorithm from trial 1 (blue), the integrity risk proposed by the algorithm for the trial 1 (brown), and the actual integrity risk after the landmarks were proposed. As shown in the "minimization area," the overall integrity risk is guaranteed to be less than the safety threshold.

simulated trajectories for a vehicle in a landmark map generated from Trial 1. In the first three runs, the robot travels along the paths represented in different colors at a constant 25km h^{-1}. A fourth trial also executed that follows the same trajectory as the third, but at 20km h^{-1}.

Fig. 13 shows the integrity risk curves for each simulation. The integrity risk is generally similar in form, but is shifted in phase due to the relative position change between the vehicle and the landmarks. There is also a difference in magnitude between the different speed trials. As such, if the robot were travelling at a variable speed, as in a real experiment, we would expect the $P(HMI)$ to be stretched and shrunk in time such that peaks and valleys in the integrity risk would occur at different epochs for each trial. Furthermore, these simulations do not account for differences in IMU data across each trial, which may create additional variation among trials.

C. Results

The trajectories before and after the I-LaG algorithm are shown in Fig. 14. The I-LaG algorithm proposed three landmarks (see red dots in Fig. 14). Fig. 15 compares the $P(HMI)$ curves where the $P(HMI)$ from Trial 1 exceeds the safety threshold. The post-I-LaG integrity risk differs from Trial 1, as expected, due to the aforementioned trajectory variations and initial conditions, but they follow similar trends. Most importantly, the new landmarks removed the integrity risk above the safety threshold.

VI. Conclusion

The simulation and experimental results show that I-LaG and Fast I-LaG can improve a robot's localization safety within a given environment by adding landmarks in key locations. The I-LaG algorithm proposes relatively fewer landmarks, but it requires more computation time. The additional landmarks in the Fast I-LaG algorithm may result in a lower integrity risk than what is given in I-LaG algorithm.

These algorithms may prove useful in situations where a robot must guarantee its localization safety, especially when evaluating stretches of roadway to identify where additional landmarks must be placed to ensure that autonomous vehicles are safe for their human passengers.

References

[1] T. Miller and D. McAslan, "With the launch of self-driving ride-share service 'waymo one,' whats next for cities?" 2019. [Online]. Available: https://www.greenbiz.com/article/launch-self-driving-ride-share-service-waymo-one-whats-next-cities, Accessed on: Dec. 25, 2019.
[2] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Res. Part A: Policy Pract.*, vol. 77, pp. 167–181, 2015.
[3] *ISO 26262-1 Road Vehicles – Functional Safety*, Std.
[4] *ARP4754A Guidelines For Development Of Civil Aircraft and Sys.*, Std.
[5] *ISO/PAS 21448 Road Vehicles – Safety of the Intended Functionality*, Std.
[6] Y. C. Lee, "Analysis of range and position comparison methods as a means to provide gps integrity in the user receiver," in *Proc. 42nd Annu. Meeting Inst. Navigation*, Jun. 1986, pp. 1–4.
[7] B. W. Parkinson and P. Axelrad, "Autonomous GPS integrity monitoring using the pseudorange residual," *Navigation*, vol. 35, no. 2, pp. 255–274, Summer 1988.
[8] G. X. Gao, M. Sgammini, M. Lu, and N. Kubo, "Protecting gnss receivers from jamming and interference," *Proc. IEEE*, vol. 104, no. 6, pp. 1327–1338, Jun. 2016.
[9] G. DuenasArana, M. Joerger, and M. Spenko, "Local nearest neighbor integrity risk evaluation for robot navigation," in *Proc. Int. Conf. Robot. Autom.*, 2018, pp. 2328–2333.
[10] 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.
[11] 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.
[12] O. A. Hafez G. D. Arana, M. Joerger, and M. Spenko, "Quantifying robot localization safety: A new integrity monitoring method for fixed-lag smoothing," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3182–3189, Apr. 2020.
[13] O. A. Hafez *et al.*, "On robot localization safety for fixed-lag smoothing: Quantifying the risk of misassociation," in *Proc. IEEE/ION Position, Location Navigation Sympo.*, vol. 5, no. 2, pp. 3182–3189, 2020.
[14] C. Li and S. L. Waslander, "Visual measurement integrity monitoring for uav localization," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot.*, Sep. 2019, pp. 22–29.
[15] T. Gu, J. Snider, J. M. Dolan, and J.-W. Lee, "Focused trajectory planning for autonomous on-road driving," in *Proc. IEEE Intell. Vehicles Sympo.*, 2013, pp. 547–552.
[16] D. Isele, "Interactive decision making for autonomous vehicles in dense traffic," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 3981–3986.
[17] W. R. R. Michalson and I. F. Progri, "Assessing the accuracy of underground positioning using pseudolites," in *Proc. 13th Int. Tech. Meeting Satel. Division Inst. Navigation*, 2000, Sep. 19–22.
[18] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, "Landmark selection for vision-based navigation," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 334–349, Apr. 2006.
[19] H. K. Mousavi and N. Motee, "Estimation with fast feature selection in robot visual navigation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3572–3579, Apr. 2020.
[20] M. Joerger, G. D. Arana, M. Spenko, and B. Pervan, "Landmark selection and unmapped obstacle detection in lidar-based navigation," in *Proc. 30th Int. Tech. Meeting Satell. Division Inst. Navigation*, 2017, pp. 1886–1903.
[21] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Found. Trends Robot.*, vol. 6, no. 1–2, pp. 1–139, 2017.
[22] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.
[23] 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, 2018, Art. no. 2740.
[24] G. D. Arana, O. A. Hafez, M. Joerger, and M. Spenko, "Integrity monitoring for kalman filter-based localization," *Int. J. Robot. Res.*, 2020.
[25] 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.
[26] M. Joerger Fang-Cheng Chan, and Boris Pervan, "Solution separation versus residual-based raim," *Navigation: J. Inst. Navigation*, vol. 61, no. 4, pp. 273–291, 2014.
[27] G. D. Arana, O. A. Hafez, M. Joerger, and M. Spenko, "Localization safety verification for autonomous robots," in *Proc. Int. Conf. Intell. Robots Syst.*, Apr. 2020.