Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Data-driven zonotopic approximation for n-dimensional probabilistic geofencing

Pengcheng Wu [a,b], Jun Chen [b,*]

[a] Department of Mechanical and Aerospace Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, CA, 92093, USA
[b] Department of Aerospace Engineering, San Diego State University, 5500 Campanile Drive, San Diego, CA, 92182, USA

ARTICLE INFO

Keywords:
Probabilistic geofence
Zonotopic approximation
Uncertain dynamic system
Kernel density estimator
Integer linear programming

ABSTRACT

Advanced air mobility is a promising way of metropolitan air transportation. One critical concern that arises is how to ensure operational safety in high-dense, dynamic, and uncertain airspace environments in real time. To address this challenge, we seek a probabilistic geofence that bounds system states with high confidence. To identify the n-dimensional probabilistic geofence for arbitrary unknown uncertainties not limited to Gaussian ones, we present an online algorithm based on a data-driven approach of kernel density estimator. Considering the irregular shape of the probabilistic geofence, we formulate an optimization framework of integer linear programming whose solution determines a zonotope which provides a convex approximation for the probabilistic geofence. Leveraging this formulation, a heuristic algorithm is developed to find its solution efficiently without losing notable accuracy. This heuristic algorithm is tested on case studies that demonstrate it enjoys efficiency, accuracy, near-optimality, and robustness simultaneously.

1. Introduction

1.1. Background and literature review

Given the rapid growth of the unmanned aircraft system (UAS), the management of the operation of unmanned aerial vehicles (UAVs) brings a demand for a reliable and effective air traffic control system. The advanced air mobility (AAM) system, including urban air mobility (UAM) and UAS traffic management (UTM), serves this purpose. Various entities around the world, including government, industry, and academia, are exploring the sector of AAM [1,2]. AAM is expected to serve the demand of urban air transportation by introducing multiple electric vertical take-off and landing (eVTOL) aircraft into the limited airspace environment [3,4].

Collision avoidance is a critical concern when it comes to the operations of the system [5]. However, there exist uncertainties like the disturbance of turbulence and GPS noise in the airspace environment [6,7]. Therefore, how to assure operational safety in the dynamic, uncertain, and high-dense airspace in real-time leads to a critical challenge for AAM. Also, the authorization of various eVTOL aircraft will be beyond the capacity of the system and congest the airspace. To address these issues, a low-altitude air traffic management system is introduced for AAM. Geofencing plays a pivotal role in the system, which reserves airspace volumes and bounds the possible locations of aircraft incorporating uncertainties to ensure operational safety [8]. The design of geofencing concentrates on identifying a collision-free trajectory based on geofences. If the identified trajectory does not violate the designed geofence, then collision avoidance can be guaranteed.

Researchers are faced with many challenges when it comes to geofencing:

(1) Unboundedness: When uncertainties are unbounded, it may be impossible to figure out a bounded geofence. However, most real-world applications need a bounded geofence due to physical limits [9];

(2) Unknown Distributions: Uncertainties are often assumed to obey Gaussian distributions in most literature. However, in practice, the uncertainties may be non-Gaussian [10];

(3) Convexity: The geofence computed by most algorithms is irregular or non-convex, which is hard to handle when it comes to the resolution of collision avoidance for the safe operation of [11];

(4) Efficiency: In order to realize the operation of AAM in real-time, the geofence must be computed efficiently. However, many algorithms are computationally expensive [12].

To address the challenges, in what follows, we will review some related literature, and then introduce the contributions of our paper. For unbounded uncertainties, it is often impossible to find a bounded geofence. Instead, in this paper we turn to find a probabilistic geofence that bounds the states of aircraft with high confidence, which is a more practical approach [9].

[*] Corresponding author.
E-mail addresses: pcwupat@ucsd.edu (P. Wu), jun.chen@sdsu.edu (J. Chen).

https://doi.org/10.1016/j.ress.2023.109923
Received 7 September 2023; Received in revised form 26 December 2023; Accepted 28 December 2023
Available online 3 January 2024

The previous studies usually make assumptions that the uncertainties obey Gaussian probability distributions, which may be inconsistent with real scenarios [13]. Some researchers fit the non-Gaussian uncertainties into Gaussian ones, which is inaccurate and may compromise the safety requirement [14,15]. Han et al. shows that fitting non-Gaussian uncertainties into Gaussian ones may find sub-optimal solutions and it is necessary to consider non-Gaussian uncertainties to ensure operational safety [10,16]. Data-driven approach is a popular way using data obtained from experiments for uncertainty quantification [17,18]. In this paper, we capture arbitrary unknown uncertainties, not limited to Gaussian ones, based on the data-driven approach of kernel density estimator (KDE), and take the level set of KDE as the probabilistic geofence of the system states. Also, the algorithm of fast Fourier transform (FFT) is applied to accelerate the evaluation of probabilistic geofence.

However, taking the level set of KDE as the probabilistic geofence brings limitations like the irregular shape of the probabilistic geofence obtained in this way. These limitations refrain from the probabilistic geofence from real applications. Instead, researchers turn to provide a convex approximation for the probabilistic geofence. The bounding box and convex hull are two existing methods that can provide convex bounds for the geofence [19]. Lew et al. employ a convex hull of samples to provide convex approximation [20]. Althoff et al. [21] figure out enclosing hulls as convex approximations. However, what they find are too conservative to be accurate approximations of the probabilistic geofence because they lack quantification. The number of facets (also called hyperfaces) of the convex hull is undetermined and a convex hull usually has too many facets leading to too many constraints, which is not convenient for the operations in practice. A bounding box can encompass data samples with a fixed number of facets but is more conservative [22]. Wu et al. [11] finds an efficient box approximation for data-driven probabilistic geofencing, but it does not apply to higher-dimensional space. In this work, using KDE results, an integer linear programming (ILP) problem is formulated, the solution to which determines a zonotope providing an accurate approximation for the probabilistic geofence in n-dimensional space. Then, a heuristic algorithm is developed to solve the formulated optimization problem efficiently.

1.2. Contributions and organization

This paper is dedicated to providing efficient and accurate convex approximations for the n-dimensional probabilistic geofence of uncertain dynamic systems. The major contributions of this work are summarized as:

(1) Based on KDE accelerated by FFT, an online algorithm is developed to effectively capture arbitrary unknown probability distributions and to efficiently evaluate the probabilistic geofence of the system states. This data-driven algorithm can learn from data online, and thus neither requires a priori information nor relies on strong assumptions.

(2) An optimization problem of ILP is formulated to provide a zonotopic approximation for n-dimensional probabilistic geofencing. The ILP problem is formulated not by using data samples directly but by grid points weighted by KDE results, which makes it tractable. As opposed to the existing approaches such as bounding box and Gaussian fit, the resulting zonotope is tighter and more accurate, and thus contributes to providing a larger feasible planning space for AAM.

(3) A heuristic algorithm is developed to efficiently solve the formulated ILP problem. Compared with the standard algorithm which finds a globally optimal solution to the optimization problem, this heuristic algorithm can efficiently find a near-optimal solution, without losing notable accuracy and robustness.

The rest of this paper is organized as follows. In Section 2, the problem with which we are concerned is formally stated. In Section 3, an algorithm is developed to capture the probabilistic geofence online. In Section 4, an ILP problem is formulated the solution to which determines a zonotope providing an accurate approximation for the probabilistic geofence in n-dimensional space. In Section 5, a heuristic algorithm is developed to solve the ILP problem formulated in the last section. In Section 6, case studies are conducted to demonstrate the feasibility and efficiency of the proposed heuristic algorithm. Last but not least, we conclude this paper and make suggestions for future research in Section 7.

Notations: $\mathbb{M} = \{1, \ldots, M\}$ indicates the index set of M data samples, $\mathbb{N} = \{1, \ldots, N\}$ indicates the index set of N grid points in each dimension, and $\mathbb{J} = \{(j_1, \ldots, j_n) : \min(i_1, i_1') \le j_1 \le \max(i_1, i_1'), \ldots, \min(i_n, i_n') \le j_n \le \max(i_n, i_n')\}$ indicates the index set of all grid points which lie between one grid point indexed by $(i_1, \ldots, i_n)$ and the other grid point indexed by $(i_1', \ldots, i_n')$.

2. Problem statement

In this section, we first investigate an uncertain discrete-time dynamic system and introduce the concept of probabilistic geofence of the system. We then present the goal of this paper.

Consider a discrete-time dynamic system of the form

$$\boldsymbol{x}_{t+1} = \boldsymbol{f}\left(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}_t, \boldsymbol{w}_t\right), \tag{1}$$

where $\boldsymbol{x}_t \in \mathbb{R}^n$ is the system state at time step t, $\boldsymbol{u}_t \in \mathbb{R}^m$ is the control input at time step t, the uncertain parameter $\boldsymbol{\theta}_t \in \mathbb{R}^p$ and the external disturbance $\boldsymbol{w}_t \in \mathbb{R}^q$ are two random vectors independent and identically distributed across time, whose probability distributions are unknown (and not necessarily Gaussian). The initial condition $\boldsymbol{x}_0$ may be known exactly or subject to an unknown probability distribution.

Collision avoidance is a critical concern when it comes to the operations of the system. If the uncertainties are unbounded, it is often impossible to guarantee that collision can be absolutely avoided. Instead, given a confidence level α, we hope to guarantee that the probability of collision is bounded by the threshold $(1-\alpha)$, i.e.,

$$\forall t, \Pr\Big(\bigvee_{i=1}^{N_o} \|\boldsymbol{x}_t - \boldsymbol{x}_{i_t}\| \le \delta\Big) \le (1-\alpha), \tag{2}$$

where $\bigvee$ represents the logical operator of "or", N_o is the number of obstacles, $\|\cdot\|$ is Euclidean norm, $\boldsymbol{x}_{i_t}$ is the state of the ith obstacle at time t, δ is the safe separation distance.

Eq. (2) is a chance constraint and we hope to convert it to a deterministic constraint equivalently. Since $\boldsymbol{x}_t$ and $\boldsymbol{x}_{i_t}$, $i \in \{1, \ldots, N_o\}$ are subject to unknown uncertainties which may be unbounded, we hope to find a bounded probabilistic geofence for each of them respectively. The notion of probabilistic geofence is formally defined as follows.

Definition 1 (*Probabilistic Geofence*)**.** A bounded set $\mathcal{X}_t \subseteq \mathbb{R}^n$ is a *probabilistic geofence* of the system state $\boldsymbol{x}_t \in \mathbb{R}^n$ at time t in Eq. (1) at confidence level α if and only if

$$\Pr(\boldsymbol{x}_t \in \mathcal{X}_t) \ge \alpha.$$

Through the introduction of probabilistic geofence, the probabilistic constraint Eq. (2) can be converted to the following deterministic constraint

$$\forall t, \mathcal{X}_t \cap \bigcup_{i=1}^{N_o} \mathcal{X}_{i_t} = \emptyset,$$

where $\bigcup$ represents the set operation of "union".

However, the shape of the probabilistic geofence $\mathcal{X}_t$ and $\mathcal{X}_{i_t}$, $i \in \{1, \ldots, N_o\}$ is usually nonconvex and irregular, which refrains it from real applications. In this paper, we hope to efficiently find a special type of bounded convex polytopes, termed *zonotopes*, to provide an accurate zonotopic approximation for the probabilistic geofence, where the precise definition of zonotopes will be introduced in Section 4.

Fig. 1 is used to help illustrate the notions like probabilistic geofence and its zonotopic approximation. In this figure, the black points

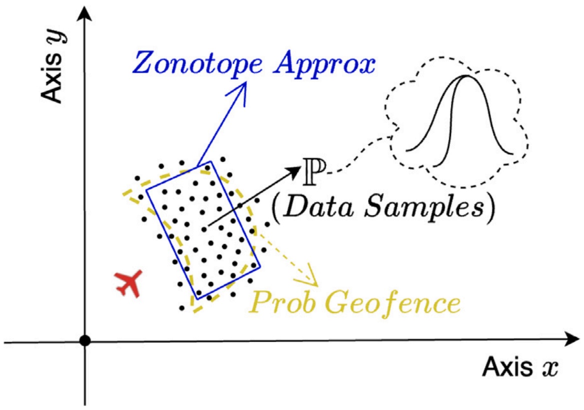


Fig. 1. Diagram of probabilistic geofence and its zonotopic approximation in 2D space.

indicate the data samples drawn from an unknown probability distribution $\mathbb{P}$, the yellow dashed curve indicates the probabilistic geofence, and the blue solid box indicates the zonotopic approximation of the probabilistic geofence.

Probabilistic geofencing plays a useful role in the safe operations of AAM in real-time. For the safe and reliable operations of AAM, in addition to the stationary obstacles like buildings, for a given vehicle, all the other vehicles relative to this vehicle can be viewed as moving obstacles. Thus, the probabilistic geofences can be used to bound the states of the obstacles with high confidence. If we can compute probabilistic geofences in real-time and enforce the planned trajectory does not violate these probabilistic geofences, then we can guarantee that the probability of collision is bounded by the threshold $(1-\alpha)$. In view of the irregular shape of the probabilistic geofences which is not convenient in practice, we hope to efficiently find zonotopes to provide polytopic approximations for the probabilistic geofences.

In summary, the goal of this paper is twofold:

(1) To efficiently and accurately capture an arbitrary unknown uncertainty based on a data-driven approach and get n-dimensional probabilistic geofences of the stochastic system state;

(2) Approximate the probabilistic geofence using a zonotope that satisfies: (i) **Convexity**: The zonotope is a bounded convex polytope, which benefits the convenience of AAM applications; (ii) **Efficiency**: The computation of finding the zonotope is tractable, which makes real-time motion planning of AAM possible; (iii) **Accuracy**: The probability of the system state $\boldsymbol{x}_t$ in the zonotope is close to the confidence level α, which brings safety guarantee for the operations of AAM; (iv) **Optimality**: The volume of the zonotope is as small as possible (not too conservative) while ensuring accuracy, which provides a larger feasible airspace for the safe operations of AAM.

We will come up with our research methodology from Section 3 through Section 5. A framework Fig. 2 is given to aid the readers' comprehension. The outline of our methodology is summarized as follows: (1) We obtain M data samples and establish N^n grid points encompassing these M data samples, where n is the number of dimension; (2) We assign a count to each grid point according to the distribution of M data samples and then we place a kernel function on each grid point and evaluate KDE values on the mesh N^n grid points such that each grid point is weighted by a KDE value; (3) We can identify the level set of KDE serving as the probabilistic geofence; (4) Using N^n grid points weighted by KDE values, we can formulate an ILP problem whose solution determines a zonotope which approximates the probabilistic geofence; (5) We develop an algorithm of Heuristic ILP to efficiently solve the formulated ILP problem.

3. Probabilistic geofence identification

In this section, we capture an arbitrary unknown n-variate probability distribution through FFT-based KDE and develop an algorithm to identify a probabilistic geofence of the system state online at confidence level α.

3.1. Gaussian fit

In previous research, an unknown n-variate probability distribution is often fitted to an n-variate Gaussian distribution whose probabilistic geofence is an n-dimensional ellipsoid.

Consider a collection of M data samples $\boldsymbol{x}_k = (x_{k1}, x_{k2}, \dots, x_{kn}) \in \mathbb{R}^n$, $k \in \mathbb{M}$. We can figure out its sample mean vector $\bar{\boldsymbol{X}}$ and sample covariance matrix $\boldsymbol{S}$ [23]. If the system state obeys a Gaussian distribution $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with the mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and the covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$, then $\bar{\boldsymbol{X}}$ and $\boldsymbol{S}$ of the data samples serve the estimation of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, respectively.

For the Gaussian distribution $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the quadratic form $(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \sim \chi_n^2$, where χ_n^2 is a chi-squared distribution with n degrees of freedom. Thus, it follows that

$$\Pr\left((\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \le F^{-1}(\alpha)\right) = \alpha,$$

where F^{-1} is the inverse cumulative density function (CDF) of χ_n^2.

The equation $(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = F^{-1}(\alpha)$ indicates an n-dimensional ellipsoid whose center is $\boldsymbol{\mu}$. The length of its ith principal axis is $2\sqrt{\frac{F^{-1}(\alpha)}{\lambda_i}}$ where λ_i is the ith eigenvalue of $\boldsymbol{\Sigma}^{-1}$, and the orientation of the ith principal axis aligns with an eigenvector which belongs to λ_i of $\boldsymbol{\Sigma}^{-1}$. This n-dimensional ellipsoid can serve as a probabilistic geofence of the system state at confidence level α.

3.2. FFT-based KDE

FFT-based KDE is a non-parametric way to capture unknown probability distributions, accelerated by fast Fourier transform (FFT). In this subsection, we introduce FFT-based KDE [24,25]. First, the concept of kernel density estimator is introduced below.

Definition 2 (*Kernel Density Estimator (KDE)*)**.** Let $\boldsymbol{x}_k \in \mathbb{R}^n, k \in \mathbb{M}$ be M data samples drawn from an n-variate probability distribution given by a PDF $f : \mathbb{R}^n \mapsto \mathbb{R}$. The *kernel density estimator* (KDE) $\hat{f} : \mathbb{R}^n \mapsto \mathbb{R}$ to approximate the PDF f is defined to be

$$\hat{f}(\boldsymbol{x}) = \frac{1}{M}\sum_{k=1}^{M} K(\boldsymbol{x}-\boldsymbol{x}_k),$$

where the function $K : \mathbb{R}^n \mapsto \mathbb{R}$ is the *kernel function* which is a symmetric n-variate density function.

The choice of the kernel function K is not critical to the accuracy of KDE [25]. Due to its convenient mathematical properties, a Gaussian kernel is often used, which is

$$K(\boldsymbol{x}) = (2\pi)^{-\frac{n}{2}}\det(\boldsymbol{H})^{-\frac{1}{2}}\exp(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{H}^{-1}\boldsymbol{x}),$$

where $\boldsymbol{H}$ is a symmetric and positive definite bandwidth matrix. The choice of bandwidth matrix is crucial to the accuracy of KDE approximating PDF [25]. In this paper, we choose the bandwidth matrix using Silverman's rule [24].

Consider a mesh of N^n *grid points* $\left\{(x_{1i_1}, \dots, x_{ni_n}) \in \mathbb{R}^n : i_1, \dots, i_n \in \mathbb{N}\right\}$ uniformly spaced on the space $\mathbb{R}^n$, the boundary of which encompasses all M data samples $\boldsymbol{x}_k \in \mathbb{R}^n$, $k \in \mathbb{M}$. The mesh of grid points can be viewed as a map $\boldsymbol{g} : \left\{(i_1, \dots, i_n) : i_1, \dots, i_n \in \mathbb{N}\right\} \mapsto \left\{(x_{1i_1}, \dots, x_{ni_n}) \in \mathbb{R}^n : i_1, \dots, i_n \in \mathbb{N}\right\}$ given by

$$(x_{1i_1}, \dots, x_{ni_n}) = \boldsymbol{g}(i_1, \dots, i_n),\ i_1, \dots, i_n \in \mathbb{N}, \tag{3}$$

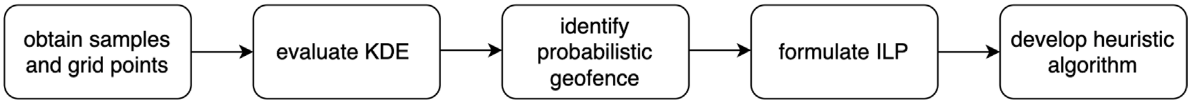

obtain samples and grid points
evaluate KDE
identify probabilistic geofence
formulate ILP
develop heuristic algorithm

Fig. 2. Framework of research methodology.

where $(i_1,\dots,i_n)$ is the index of a grid point whose coordinate is $(x_{1i_1},\dots,x_{ni_n})$. Throughout the rest of this paper, a coordinate $\boldsymbol{g}(i_1,\dots,i_n)$ is abbreviated as $\boldsymbol{g}_{i_1\cdots i_n}$.

The KDE $\hat{f}$ given in Definition 2 can be evaluated on the mesh of N^n grid points $\boldsymbol{g}$ in Eq. (3), which yields

$$\hat{f}_{i_1\cdots i_n}=\hat{f}(\boldsymbol{g}_{i_1\cdots i_n})=\frac{1}{M}\sum_{k=1}^{M}K(\boldsymbol{g}_{i_1\cdots i_n}-\boldsymbol{x}_k),$$
$$i_1,\dots,i_n\in\mathbb{N}.$$

One way to significantly accelerate the evaluation of KDE is *linear binning* [25]. Instead of placing the kernel functions K on M data samples, we can place them on N^n grid points $\boldsymbol{g}$ weighted by grid counts $c_{u_1\cdots u_n}$, $u_1,\dots,u_n\in\mathbb{N}$. A grid count $c_{u_1\cdots u_n}$ is a weight chosen to represent the amount of data samples near a grid point $\boldsymbol{g}_{u_1\cdots u_n}$. To obtain $c_{u_1\cdots u_n}$, we can go through every data sample and assign weights to its neighboring grid points.

Using the strategy of linear binning, the *binned* KDE $\tilde{f}:\mathbb{R}^n\mapsto\mathbb{R}$ to approximate the KDE $\hat{f}$ in Definition 2 is given by

$$\tilde{f}(\boldsymbol{x})=\frac{1}{M}\sum_{u_1,\dots,u_n=1}^{N}K(\boldsymbol{x}-\boldsymbol{g}_{u_1\cdots u_n})c_{u_1\cdots u_n},\tag{4}$$

where $c_{u_1\cdots u_n}$ is the grid count of the grid point $\boldsymbol{g}_{u_1\cdots u_n}$ indexed by $(u_1,\dots,u_n)$.

Evaluating the binned KDE $\tilde{f}$ in Eq. (4) on the mesh of N^n grid points $\boldsymbol{g}$ yields

$$\tilde{f}_{i_1\cdots i_n}=\tilde{f}(\boldsymbol{g}_{i_1\cdots i_n})=\frac{1}{M}\sum_{u_1,\dots,u_n=1}^{N}K(\boldsymbol{g}_{i_1\cdots i_n}-\boldsymbol{g}_{u_1\cdots u_n})c_{u_1\cdots u_n},$$
$$i_1,\dots,i_n\in\mathbb{N}.\tag{5}$$

Let $\tilde{\boldsymbol{f}}$ be the vector consisting of $\tilde{f}_{i_1\cdots i_n}$, $i_1,\dots,i_n\in\mathbb{N}$, $\boldsymbol{c}$ be the vector of $c_{u_1\cdots u_n}$, $u_1,\dots,u_n\in\mathbb{N}$, and $\boldsymbol{k}$ be the vector of $K(\boldsymbol{g}_{u_1\cdots u_n})$, $u_1,\dots,u_n\in\mathbb{N}$. Then, Eq. (5) indicates that $\tilde{\boldsymbol{f}}$ is the discrete convolution of $\boldsymbol{c}$ and $\boldsymbol{k}$. The discrete convolution can be computed by FFT, which significantly saves computational time. Let $\boldsymbol{C}$ and $\boldsymbol{K}$ be FFT of $\boldsymbol{c}$ and $\boldsymbol{k}$, and $\tilde{\boldsymbol{F}}$ be the element-wise product of $\boldsymbol{C}$ and $\boldsymbol{K}$. From the results of the inverse FFT of $\tilde{\boldsymbol{F}}$, we can acquire $\tilde{\boldsymbol{f}}$. Thus, using FFT, we can efficiently figure out the binned KDE $\tilde{f}_{i_1\cdots i_n}$, $i_1,\dots,i_n\in\mathbb{N}$ in Eq. (5) which can approximate the PDF f of an arbitrary unknown n-variate probability distribution [26,27].

We refer to the way of evaluating the binned KDE using FFT as FFT-based KDE. The computation of binned KDE given by Eq. (5) is equivalent to the computation of a discrete convolution, and FFT-based KDE and direct computation are just two different ways to compute the same discrete convolution. According to the convolution theorem, the results of two different ways are exactly the same. However, in practice, there may be a minor difference between the result of FFT-based KDE and binned KDE due to numerical errors, which has already been well discussed in [25,27].

3.3. Probabilistic geofence identification algorithm

We propose an efficient data-driven algorithm using FFT-based KDE to capture the probabilistic geofence at confidence level α of an unknown stationary uncertainty which obeys an n-variate probability distribution f whose data samples are gradually observed. See Algorithm 1.

In Algorithm 1, the symbol ε represents the tolerance of confidence error. From Lines 9 through 16, FFT is introduced to speed up computing the discrete convolution of $\boldsymbol{c}$ and $\boldsymbol{k}$ to obtain the binned KDE $\tilde{\boldsymbol{f}}$ evaluated on the mesh of N^n grid points $\boldsymbol{g}$ which aligns with the axes of the coordinate system. From Lines 17 through 32, a bisection search is implemented to find the critical binned KDE value C^{kde} at which the confidence level α is achieved. The level set of the KDE corresponding to C^{kde} is the probabilistic geofence that we aim to find. Note that the shape of a probabilistic geofence obtained in this way is often irregular, which precludes its use in subsequent applications requiring an optimal design. This will be solved in the next section.

As the number of the collected data samples increases, KDE captured from those data samples will be convergent to the true PDF of the unknown uncertainties [25]. In practice, the information about uncertainty is changing over time and the distribution captured by the algorithm is also updated with more data samples coming in. As the number of the collected data samples increases, KDE captured from those data samples will be convergent to the true PDF of the unknown uncertainties [25]. However, this poses a great challenge requiring that the proposed algorithm must run in real-time to perform re-computation every time newly observed data samples are added. Our proposed algorithm can handle this concern very well, which we will see in Section 6.5.

4. ILP formulation

In this section, we formulate an ILP optimization problem whose solution is a zonotope serving a convex approximation of the probabilistic geofence obtained in Algorithm 1.

4.1. Zonotope definition

A zonotope is a convex polytope that is the Minkowski sum of a finite set of vectors. The precise concept of zonotope is introduced as follows [21,28,29].

Definition 3 (*Zonotope*)**.** A *zonotope* Z generated by vectors $\bar{z},\boldsymbol{v}_1,\dots,\boldsymbol{v}_k\in\mathbb{R}^n$ is a set of points in n-dimensional space

$$Z=\left\{a_1\boldsymbol{v}_1+\cdots+a_k\boldsymbol{v}_k+\bar{z}\in\mathbb{R}^n:\forall i\in\{1,\dots,k\},\ a_i\in[0,1]\right\},\tag{6}$$

where $\bar{z}$ describes translation and $\boldsymbol{v}_1,\dots,\boldsymbol{v}_k$ are termed *generators*. Each a_i is a scalar between 0 and 1.

A zonotope is a bounded simply connected convex set. The use of zonotopes is motivated by the fact that they are closed under the operations of Minkowski addition and linear transformation [30,31].

Denoting $[\boldsymbol{v}_1,\dots,\boldsymbol{v}_k]$ by a matrix $\boldsymbol{G}$ and $(a_1,\dots,a_k)$ by a vector $\boldsymbol{a}$, then Eq. (6) can be rewritten in a compact form

$$Z=\left\{\boldsymbol{G}\boldsymbol{a}+\bar{z}\in\mathbb{R}^n:\forall i\in\{1,\dots,k\},\ \boldsymbol{a}[i]\in[0,1]\right\}.\tag{7}$$

In the special case where $k\geq n$ and $\operatorname{rank}(\boldsymbol{G})=n$, then the zonotope Z is termed a *full-dimensional zonotope*; In the special case where $k\leq n$, the zonotope Z is a (possibly degenerate) parallelotope. In this paper, we consider the zonotopes that satisfy $\operatorname{rank}(\boldsymbol{G})=n=k$.

Next, the following lemma shows that a zonotope is closed under the operation of an affine transformation.

Lemma 1. *Given a zonotope Z and an affine transformation $\boldsymbol{T}:\boldsymbol{x}\mapsto\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}$, $\boldsymbol{A}\in\mathbb{R}^{n\times n}$, then the affine transformation $\boldsymbol{T}(Z)$ of the zonotope Z is still a zonotope. Further, if for Z its $\operatorname{rank}(\boldsymbol{G})=n=k$ and for $\boldsymbol{T}$ its matrix $\boldsymbol{A}$ is invertible, then for $\boldsymbol{T}(Z)$ its $\operatorname{rank}(\boldsymbol{G}')=n=k$.*

Algorithm 1 Probabilistic Geofence Identification Algorithm

```
function SAMPLING($M$)
    Obtain $M$ samples $\boldsymbol{x}_k, k \in \mathbb{M}$
    return $\boldsymbol{x}_k$

function MESH($\boldsymbol{x}_k$, $N$)
    Get $x_{1\min}, x_{1\max}, \cdots, x_{n\min}, x_{n\max}$ from
    data samples $\boldsymbol{x}_k$
    $\boldsymbol{h}_1 = \text{linspace}(x_{1\min}, x_{1\max}, N), \cdots,$
    $\boldsymbol{h}_n = \text{linspace}(x_{n\min}, x_{n\max}, N)$
    $\boldsymbol{g}$ = Cartesian product of $\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n$
    return $\boldsymbol{g}$

function KDE($\boldsymbol{x}_k$, $\boldsymbol{g}$)
    Get grid counts $\boldsymbol{c}$ for grid points $\boldsymbol{g}$ according to $\boldsymbol{x}_k$
    Get kernel function values $\boldsymbol{k}$ evaluated on $\boldsymbol{g}$
    Get FFT result $\boldsymbol{C}$ of $\boldsymbol{c}$
    Get FFT result $\boldsymbol{K}$ of $\boldsymbol{k}$
    Get the element-wise product $\boldsymbol{F}$ of $\boldsymbol{C}$ and $\boldsymbol{K}$
    Get the inverse FFT result $\tilde{\boldsymbol{f}}$ of $\boldsymbol{F}$
    return $\tilde{\boldsymbol{f}}$

function BISEC($\tilde{\boldsymbol{f}}$, $\alpha$, $\varepsilon$)
    $l = \min(\tilde{\boldsymbol{f}})$
    $u = \max(\tilde{\boldsymbol{f}})$
    while $l < u$ do
        $m = 1/2(l + u)$
        $\boldsymbol{z}^{\text{bin}} = (\tilde{\boldsymbol{f}} \geq m) \cdot 1$
        $\boldsymbol{z}^{\text{mix}}$ = element-wise product of $\boldsymbol{z}^{\text{bin}}$ and $\tilde{\boldsymbol{f}}$
        $p = \sum \boldsymbol{z}^{\text{mix}} / \sum \tilde{\boldsymbol{f}}$
        if $|p - \alpha| \leq \varepsilon$ then
            $C^{\text{kde}} = m$
            return $\boldsymbol{z}^{\text{bin}}$, $C^{\text{kde}}$
        else if $p < \alpha$ then
            $u = m$
        else
            $l = m$
    return $Fail$

function PROBGEO($M$, $N$, $\alpha$, $\varepsilon$)
    $\boldsymbol{x}_k$ = SAMPLING($M$)
    $\boldsymbol{g}$ = MESH($\boldsymbol{x}_k$, $N$)
    $\tilde{\boldsymbol{f}}$ = KDE($\boldsymbol{x}_k$, $\boldsymbol{g}$)
    $\boldsymbol{z}^{\text{bin}}$, $C^{\text{kde}}$ = BISEC($\tilde{\boldsymbol{f}}$, $\alpha$, $\varepsilon$)
    return $\boldsymbol{g}$, $\tilde{\boldsymbol{f}}$, $\boldsymbol{z}^{\text{bin}}$, $C^{\text{kde}}$

PROBGEO($M$, $N$, $\alpha$, $\varepsilon$)
```

Proof. The affine transformation $\boldsymbol{T}(Z)$ of the zonotope Z is

$$\begin{aligned}\boldsymbol{T}(Z) &= \left\{ \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \in \mathbb{R}^n \,:\, \boldsymbol{x} \in Z \right\} \\ &= \left\{ \boldsymbol{G}'\boldsymbol{a} + \bar{z}' \in \mathbb{R}^n \,:\, \forall i \in \{1,\ldots,k\}\ \boldsymbol{a}[i] \in [0,1] \right\},\end{aligned} \tag{8}$$

where $\boldsymbol{G}' = \boldsymbol{A}\boldsymbol{G}$ and $\bar{z}' = \boldsymbol{A}\bar{z}+\boldsymbol{b}$. According to the definition of zonotope in Eq. (7), $\boldsymbol{T}(Z)$ is a zonotope. Further, if $\text{rank}(\boldsymbol{G}) = n = k$ and $\boldsymbol{A}$ is invertible which means $\text{rank}(\boldsymbol{A}) = n = k$, then $\text{rank}(\boldsymbol{G}') = \text{rank}(\boldsymbol{A}\boldsymbol{G}) = n = k$. Further, assume that $\text{rank}(\boldsymbol{G}) = n = k$ and $\text{rank}(\boldsymbol{A}) = n$ which means $\boldsymbol{A}$ is invertible. Since $n = k$, the matrix $\boldsymbol{G} \in \mathbb{R}^{n\times k}$ and $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ are square matrices. In addition, since $\text{rank}(\boldsymbol{G}) = n$ and $\text{rank}(\boldsymbol{A}) = n$, then $\boldsymbol{G}$ and $\boldsymbol{A}$ are full-rank square matrices. Since the product of two full-rank square matrices is still a full-rank square matrix, $\boldsymbol{G}' = \boldsymbol{A}\boldsymbol{G}$ is an $n \times n$ full-rank square matrix and therefore $\text{rank}(\boldsymbol{G}') = n = k$. ■

4.2. *ILP formulation of axis-aligned zonotope*

Consider M data samples $\boldsymbol{x}_k = (x_{k1}, \ldots, x_{kn}) \in \mathbb{R}^n$, $k \in \mathbb{M}$ of a random vector $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and a mesh of N^n grid points $\boldsymbol{g}$ aligning with the axes of the coordinate system. Evaluating Eq. (5) through the implementation of Algorithm 1, it follows that each grid point $\boldsymbol{g}_{i_1\cdots i_n}$, $i_1, \ldots, i_n \in \mathbb{N}$ has a binned KDE value $\tilde{f}_{i_1\cdots i_n}$, $i_1, \ldots, i_n \in \mathbb{N}$, respectively. We define a normalized weight matrix $\boldsymbol{w} = [w_{i_1\cdots i_n}]$ whose element is

$$w_{i_1\cdots i_n} = \frac{\tilde{f}_{i_1\cdots i_n}}{\sum_{i_1,\ldots,i_n=1}^{N} \tilde{f}_{i_1\cdots i_n}}, \quad i_1, \ldots, i_n \in \mathbb{N}, \tag{9}$$

which satisfies

$$0 \leq w_{i_1\cdots i_n} \leq 1,$$
$$\sum_{i_1,\ldots,i_n=1}^{N} w_{i_1\cdots i_n} = 1.$$

Instead of using M data samples $\boldsymbol{x}_k$ directly, we will formulate the ILP optimization problem using the grid points $\boldsymbol{g}$ with corresponding normalized weights $\boldsymbol{w}$ obtained in Eq. (9). This reduces the number of constraints and variables, and thus contributes to the feasibility and tractability of solving the ILP problem.

The ILP optimization problem is formulated to find an optimal zonotope, which serves as a convex approximation of the probabilistic geofence. The zonotope can be determined by the selection of grid points to be covered. To this end, we introduce decision variables $z_{i_1\cdots i_n} \in \{0,1\}$, $i_1, \ldots, i_n \in \mathbb{N}$. If a grid point $\boldsymbol{g}_{i_1\cdots i_n}$ is selected to be covered by the zonotope, then $z_{i_1\cdots i_n} = 1$; otherwise, $z_{i_1\cdots i_n} = 0$.

To get the smallest zonotope by area, the objective function of the ILP optimization problem can be formulated as $\min \sum_{i_1,\ldots,i_n=1}^{N} z_{i_1\cdots i_n}$.

The grid point whose weight is the greatest is selected to be covered by the zonotope, which means

$$z_{\bar{i}_1\cdots\bar{i}_n} = 1, \tag{10}$$

where $\bar{i}_1, \ldots, \bar{i}_n = \arg\max_{i_1,\ldots,i_n}(w_{i_1\cdots i_n})$.

In order to provide an approximation for the probabilistic geofence at confidence level α, we should enforce that the sum of the weights of the selected grid points exceeds α. That is,

$$\sum_{i_1,\ldots,i_n=1}^{N} w_{i_1\cdots i_n} z_{i_1\cdots i_n} \geq \alpha. \tag{11}$$

To make the approximation of the probabilistic geofence zonotopic, we can enforce that

$$(z_{i_1\cdots i_n} \wedge z_{i'_1\cdots i'_n}) \implies z_{j_1\cdots j_n}, \tag{12}$$

where $\min(i_1, i'_1) \leq j_1 \leq \max(i_1, i'_1), \ldots, \min(i_n, i'_n) \leq j_n \leq \max(i_n, i'_n)$. It implies that if any two grid points $\boldsymbol{g}_{i_1\cdots i_n}$ and $\boldsymbol{g}_{i'_1\cdots i'_n}$ are selected, then each grid point $\boldsymbol{g}_{j_1\cdots j_n}$ between those two grid points must also be selected.

Eq. (12) can be rewritten in a conjunctive normal form

$$\neg z_{i_1\cdots i_n} \vee \neg z_{i'_1\cdots i'_n} \vee z_{j_1\cdots j_n},$$

which can be converted to the linear constraints

$$(1 - z_{i_1\cdots i_n}) + (1 - z_{i'_1\cdots i'_n}) + z_{j_1\cdots j_n} \geq 1,$$

or equivalently,

$$z_{i_1\cdots i_n} + z_{i'_1\cdots i'_n} - z_{j_1\cdots j_n} \leq 1, \tag{13}$$

where $\min(i_1, i'_1) \leq j_1 \leq \max(i_1, i'_1), \ldots, \min(i_n, i'_n) \leq j_n \leq \max(i_n, i'_n)$.

Let $\mathbb{J}$ denote the set of $\left\{(j_1, \ldots, j_n) \,:\, \min(i_1, i'_1) \leq j_1 \leq \max(i_1, i'_1), \ldots, \min(i_n, i'_n) \leq j_n \leq \max(i_n, i'_n)\right\}$. Collecting the constraints Eqs. (10), (11), and (13), we can formally formulate the optimization framework as an ILP problem

$$\begin{aligned}\min\ & \sum_{i_1,\dots,i_n=1}^{N} z_{i_1\cdots i_n}\\ \text{s.t.}\ & z_{i_1\cdots i_n}+z_{i'_1\cdots i'_n}-z_{j_1\cdots j_n}\le 1 \quad i_1,i'_1,\dots,i_n,i'_n\in\mathbb{N},\\ & (j_1,\dots,j_n)\in\mathbb{J};\\ & z_{\bar{i}_1\cdots\bar{i}_n}=1;\\ & \sum_{i_1,\dots,i_n=1}^{N} w_{i_1\cdots i_n}z_{i_1\cdots i_n}\ge\alpha;\\ & z_{i_1\cdots i_n}\in\{0,1\}\quad i_1,\dots,i_n\in\mathbb{N}.\end{aligned}\tag{14}$$

Following the definition of zonotope in Eq. (6), we can find that the solution to the ILP optimization problem above determines a zonotope Z in n-dimensional space, which is given by

$$Z=\left\{a_1\boldsymbol{v}_1+\cdots+a_k\boldsymbol{v}_k+\bar{\boldsymbol{z}}\in\mathbb{R}^n : \forall i\in\{1,\dots,k\},\ a_i\in[0,1]\right\},\tag{15}$$

where

$$\begin{aligned}\bar{\boldsymbol{z}}&=\boldsymbol{g}_{i_{1\min}i_{2\min}\cdots i_{n\min}},\\ \boldsymbol{v}_1&=\boldsymbol{g}_{i_{1\max}i_{2\min}\cdots i_{n\min}}-\boldsymbol{g}_{i_{1\min}i_{2\min}\cdots i_{n\min}},\\ \boldsymbol{v}_2&=\boldsymbol{g}_{i_{1\min}i_{2\max}\cdots i_{n\min}}-\boldsymbol{g}_{i_{1\min}i_{2\min}\cdots i_{n\min}},\\ &\vdots\\ \boldsymbol{v}_n&=\boldsymbol{g}_{i_{1\min}i_{2\min}\cdots i_{n\max}}-\boldsymbol{g}_{i_{1\min}i_{2\min}\cdots i_{n\min}},\end{aligned}$$

and

$$\begin{aligned}i_{1\min}&=\min\big(\{i_1 : z_{i_1\cdots i_n}=1\}\big),\ i_{1\max}=\max(\{i_1 : z_{i_1\cdots i_n}=1\}),\\ i_{2\min}&=\min\big(\{i_2 : z_{i_1\cdots i_n}=1\}\big),\ i_{2\max}=\max(\{i_2 : z_{i_1\cdots i_n}=1\}),\\ &\vdots \qquad\qquad\qquad\qquad\qquad \vdots\\ i_{n\min}&=\min\big(\{i_n : z_{i_1\cdots i_n}=1\}\big),\ i_{n\max}=\max(\{i_n : z_{i_1\cdots i_n}=1\}).\end{aligned}$$

Also, this n-dimensional zonotope Z satisfies $\mathrm{rank}(\boldsymbol{G})=n=k$ where $\boldsymbol{G}$ is defined in Eq. (7).

The zonotope obtained in this way must align with the axes of the coordinate system. Indeed, the directions of the generators $\boldsymbol{v}_1,\dots,\boldsymbol{v}_k\in\mathbb{R}^n$ of a zonotope should be arbitrary. Thus, we hope that the orientation of the zonotope can be customized. This will be fully discussed in the next subsection.

4.3. ILP formulation of oriented zonotope

In this subsection, not limited to axis-aligned zonotopes, we find an oriented zonotope aligning with the principal axes of a collection of data samples, which provides a convex approximation for the probabilistic geofence.

Consider a random vector $\boldsymbol{x}=(x_1,\dots,x_n)\in\mathbb{R}^n$ obeying an unknown n-variate probability distribution with M data samples $\boldsymbol{x}_k=(x_{k1},x_{k2},\dots,x_{kn})\in\mathbb{R}^n$, $k\in\mathbb{M}$ drawn from the distribution. Let $\boldsymbol{e}_1,\dots,\boldsymbol{e}_n\in\mathbb{R}^n$ be n linear independent unit vectors which represent the orientation of the principal axes of those M data samples. Since $\boldsymbol{e}_1,\dots,\boldsymbol{e}_n\in\mathbb{R}^n$ are linearly independent vectors, we can define a new random vector $\boldsymbol{u}=(u_1,\dots,u_n)\in\mathbb{R}^n$ given by a bijective affine transformation of $\boldsymbol{x}=(x_1,\dots,x_n)$, which is

$$\boldsymbol{u}=\boldsymbol{T}(\boldsymbol{x})=\boldsymbol{A}(\boldsymbol{x}-\bar{\boldsymbol{x}})+\bar{\boldsymbol{x}},\ \boldsymbol{x}\in\mathbb{R}^n,\tag{16}$$

where $\boldsymbol{A}=[\boldsymbol{e}_1,\dots,\boldsymbol{e}_n]^{-1}$ is an invertible matrix. The point $\bar{\boldsymbol{x}}=(\bar{x}_1,\dots,\bar{x}_n)=\frac{1}{M}\sum_{k=1}^{M}\boldsymbol{x}_k=\frac{1}{M}\sum_{k=1}^{M}(x_{k1},\dots,x_{kn})$ is a fixed point such that $\boldsymbol{T}(\boldsymbol{e}_i)=(\delta_{i1},\dots,\delta_{in})$, $i\in\{1,\dots,n\}$ where δ_{ij} is Kronecker delta. The inverse transformation of $\boldsymbol{T}$ is therefore

$$\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{u})=\boldsymbol{A}^{-1}(\boldsymbol{u}-\bar{\boldsymbol{x}})+\bar{\boldsymbol{x}},\ \boldsymbol{u}\in\mathbb{R}^n,$$

where $\boldsymbol{A}^{-1}=[\boldsymbol{e}_1,\dots,\boldsymbol{e}_n]$ is the inverse matrix of $\boldsymbol{A}$.

Given the PDF of an original probability distribution, the PDF of another probability distribution which is the transformation of the original probability distribution can be obtained through the following Lemma 2 [32].

Lemma 2. *Let $\boldsymbol{x}=(x_1,\dots,x_n)\in\mathbb{R}^n$ be a random vector with a joint PDF $f_{\boldsymbol{x}}(x_1,\dots,x_n)$, and $\boldsymbol{u}=(u_1,\dots,u_n)\in\mathbb{R}^n$ be another random vector which is the transformation of $\boldsymbol{x}=(x_1,\dots,x_n)$, which is*

$$\boldsymbol{u}=\boldsymbol{T}(\boldsymbol{x}) : \begin{cases}u_1=g_1(x_1,\dots,x_n)\\ \quad\vdots\\ u_n=g_n(x_1,\dots,x_n).\end{cases}$$

If (1) the functions $g_1,\dots,g_n$ have continuous first partial derivatives; (2) The transformation $\boldsymbol{T}$ is bijective and thus has a unique inverse transformation

$$\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{u}) : \begin{cases}x_1=h_1(u_1,\dots,u_n)\\ \quad\vdots\\ x_n=h_n(u_1,\dots,u_n);\end{cases}$$

(3) The Jacobian determinant J associated with $\boldsymbol{T}^{-1}$ satisfies

$$\forall u_1,\dots,u_n,\ J(u_1,\dots,u_n)=\frac{\partial(x_1,\dots,x_n)}{\partial(u_1,\dots,u_n)}\neq 0,$$

then the joint PDF of $\boldsymbol{u}=(u_1,\dots,u_n)$ is

$$\begin{aligned}f_{\boldsymbol{u}}(u_1,\dots,u_n)=&f_{\boldsymbol{x}}\big(x_1(u_1,\dots,u_n),\dots,x_n(u_1,\dots,u_n)\big)\\ &|J(u_1,\dots,u_n)|,\end{aligned}$$

where $|J(u_1,\dots,u_n)|$ is the absolute value of $J(u_1,\dots,u_n)$.

According to Lemma 2, given the joint PDF $f_{\boldsymbol{x}}(\boldsymbol{x})$ of the random vector $\boldsymbol{x}=(x_1,\dots,x_n)\in\mathbb{R}^n$, then the joint PDF $f_{\boldsymbol{u}}(\boldsymbol{u})$ of the random vector $\boldsymbol{u}=(u_1,\dots,u_n)=\boldsymbol{T}(\boldsymbol{x})$, where $\boldsymbol{T}$ is a bijective affine transformation defined in Eq. (16), is given by

$$f_{\boldsymbol{u}}(\boldsymbol{u})=f_{\boldsymbol{x}}\big(\boldsymbol{T}^{-1}(\boldsymbol{u})\big)|\det(\boldsymbol{A}^{-1})|.\tag{17}$$

Given a mesh of N^n grid points $\boldsymbol{g}_{\boldsymbol{x}}$ aligning with the principal axes $\boldsymbol{e}_1,\dots,\boldsymbol{e}_n\in\mathbb{R}^n$ of M data samples $\boldsymbol{x}_k=(x_{k1},x_{k2},\dots,x_{kn})\in\mathbb{R}^n$, $k\in\mathbb{M}$, i.e.,

$$(x_{1i_1},\dots,x_{ni_n})=\boldsymbol{g}_{\boldsymbol{x}i_1\cdots i_n},\ i_1,\dots,i_n\in\mathbb{N},$$

then a new mesh of N^n grid points $\boldsymbol{g}_{\boldsymbol{u}}=\boldsymbol{T}\circ\boldsymbol{g}_{\boldsymbol{x}}$, aligning with the axes of the coordinate system, can be defined as the composition of $\boldsymbol{T}$ and $\boldsymbol{g}_{\boldsymbol{x}}$, i.e.,

$$\begin{aligned}&(u_{1i_1},\dots,u_{ni_n})=\boldsymbol{g}_{\boldsymbol{u}i_1\cdots i_n}=\boldsymbol{T}\big(\boldsymbol{g}_{\boldsymbol{x}i_1\cdots i_n}\big),\\ &i_1,\dots,i_n\in\mathbb{N},\end{aligned}$$

where $\boldsymbol{g}_{\boldsymbol{x}i_1\cdots i_n}=\boldsymbol{g}_{\boldsymbol{x}}(i_1,\dots,i_n)$ and $\boldsymbol{g}_{\boldsymbol{u}i_1\cdots i_n}=\boldsymbol{g}_{\boldsymbol{u}}(i_1,\dots,i_n)$.

According to Eq. (17), the connection between the PDF $f_{\boldsymbol{x}}(\boldsymbol{x})$ evaluated on the grid points $\boldsymbol{g}_{\boldsymbol{x}}$ and the PDF $f_{\boldsymbol{u}}(\boldsymbol{u})$ on the grid points $\boldsymbol{g}_{\boldsymbol{u}}$ is

$$f_{\boldsymbol{u}i_1\cdots i_n}=f_{\boldsymbol{x}i_1\cdots i_n}|\det(\boldsymbol{A}^{-1})|,\ i_1,\dots,i_n\in\mathbb{N},\tag{18}$$

where $f_{\boldsymbol{u}i_1\cdots i_n}=f_{\boldsymbol{u}}\big(\boldsymbol{g}_{\boldsymbol{u}}(i_1,\dots,i_n)\big)$ and $f_{\boldsymbol{x}i_1\cdots i_n}=f_{\boldsymbol{x}}\big(\boldsymbol{g}_{\boldsymbol{x}}(i_1,\dots,i_n)\big)$.

Given M original data samples $\boldsymbol{x}_k=(x_{k1},\dots,x_{kn})\in\mathbb{R}^n,k\in\mathbb{M}$ of a random vector $\boldsymbol{x}=(x_1,\dots,x_n)$ with N^n oriented grid points $\boldsymbol{g}_{\boldsymbol{x}}$, we can always transform it to the data samples of another random vector $\boldsymbol{u}=(x_1,\dots,x_n)=\boldsymbol{T}(\boldsymbol{x})$ where $\boldsymbol{T}$ is defined in Eq. (16) with N^n axis-aligned grid points $\boldsymbol{g}_{\boldsymbol{u}}=\boldsymbol{T}\circ\boldsymbol{g}_{\boldsymbol{x}}$.

Through the implementation of Algorithm 1, we can obtain the KDE values $\tilde{f}_{\boldsymbol{u}}$ evaluated on the axis-aligned grid points $\boldsymbol{g}_{\boldsymbol{u}}$, which estimates the PDF $f_{\boldsymbol{u}}$ for the random vector $\boldsymbol{u}$. Then according to Eq. (18), the KDE $\tilde{f}_{\boldsymbol{x}}$ of the random vector $\boldsymbol{x}=\boldsymbol{T}^{-1}(\boldsymbol{u})$ evaluated on the oriented grid points $\boldsymbol{g}_{\boldsymbol{x}}$, which estimates its PDF $f_{\boldsymbol{x}}$, is given by

$$\tilde{f}_{\boldsymbol{x}i_1\cdots i_n}=\tilde{f}_{\boldsymbol{u}i_1\cdots i_n}|\det(\boldsymbol{A}^{-1})|^{-1},\ i_1,\dots,i_n\in\mathbb{N},\tag{19}$$

where $\tilde{f}_{\boldsymbol{u}i_1\cdots i_n}=\tilde{f}_{\boldsymbol{u}}\big(\boldsymbol{g}_{\boldsymbol{u}}(i_1,\dots,i_n)\big)$ and $\tilde{f}_{\boldsymbol{x}i_1\cdots i_n}=\tilde{f}_{\boldsymbol{x}}\big(\boldsymbol{g}_{\boldsymbol{x}}(i_1,\dots,i_n)\big)$.

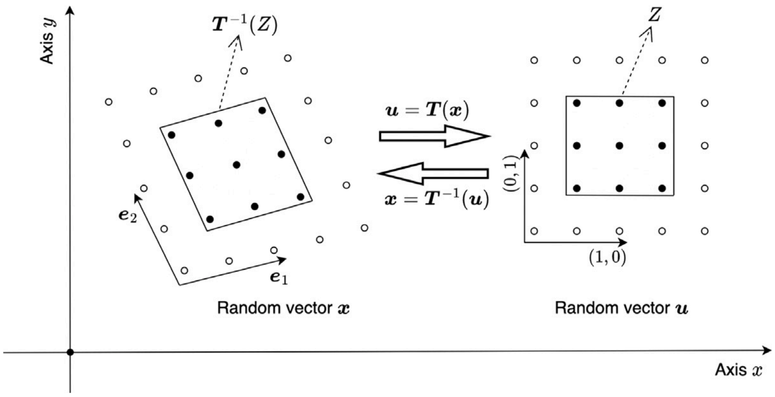


Fig. 3. Diagram of finding oriented zonotope in 2D space.

Given KDE values $\tilde{f}_{\boldsymbol{x}}$ amd $\tilde{f}_{\boldsymbol{u}}$, we can define a normalized weight matrix $\boldsymbol{w}_{\boldsymbol{x}} = [w_{\boldsymbol{x}i_1\cdots i_n}]$ for $\boldsymbol{x}$ whose element is

$$w_{\boldsymbol{x}i_1\cdots i_n} = \frac{\tilde{f}_{\boldsymbol{x}i_1\cdots i_n}}{\sum_{i_1,\ldots,i_n=1}^{N} \tilde{f}_{\boldsymbol{x}i_1\cdots i_n}}, \quad i_1,\ldots,i_n \in \mathbb{N},$$

and $\boldsymbol{w}_{\boldsymbol{u}} = [w_{\boldsymbol{u}i_1\cdots i_n}]$ for $\boldsymbol{u}$ whose element is

$$w_{\boldsymbol{u}i_1\cdots i_n} = \frac{\tilde{f}_{\boldsymbol{u}i_1\cdots i_n}}{\sum_{i_1,\ldots,i_n=1}^{N} \tilde{f}_{\boldsymbol{u}i_1\cdots i_n}}, \quad i_1,\ldots,i_n \in \mathbb{N}.$$

According to Eq. (19), it follows that $w_{\boldsymbol{u}i_1\cdots i_n} = w_{\boldsymbol{x}i_1\cdots i_n}$. Hence, for brevity, either $w_{\boldsymbol{u}i_1\cdots i_n}$ or $w_{\boldsymbol{x}i_1\cdots i_n}$ is referred to as $w_{i_1\cdots i_n}$ throughout the rest of this paper, i.e.,

$$w_{i_1\cdots i_n} = w_{\boldsymbol{u}i_1\cdots i_n} = w_{\boldsymbol{x}i_1\cdots i_n}, \tag{20}$$

and a normalized weight matrix for either $\boldsymbol{u}$ or $\boldsymbol{x}$ is defined as $\boldsymbol{w} = [w_{i_1\cdots i_n}]$.

As stated above, given M original data samples $\boldsymbol{x}_k = (x_{k1},\ldots,x_{kn}) \in \mathbb{R}^n, k \in \mathbb{M}$ of a random vector $\boldsymbol{x} = (x_1,\ldots,x_n)$ with N^n oriented grid points $\boldsymbol{g}_{\boldsymbol{x}}$, we can always transform it to the data samples of another random vector $\boldsymbol{u} = (x_1,\ldots,x_n) = \boldsymbol{T}(\boldsymbol{x})$ where $\boldsymbol{T}$ is defined in Eq. (16) with N^n axis-aligned grid points $\boldsymbol{g}_{\boldsymbol{u}} = \boldsymbol{T} \circ \boldsymbol{g}_{\boldsymbol{x}}$. For the random vector $\boldsymbol{u}$ with grid points $\boldsymbol{g}_{\boldsymbol{u}}$, according to Section 4.2, we can formulate an ILP optimization problem Eq. (14), whose solution determines an optimal axis-aligned zonotope Z in Eq. (15). For the random vector $\boldsymbol{x}$ with grid points $\boldsymbol{g}_{\boldsymbol{x}}$, we can also formally formulate another ILP problem which is formally identical to Eq. (14). Since Eq. (20) holds, the solution to the ILP problem of $\boldsymbol{x}$ is the same as that of $\boldsymbol{u}$. Hence, the solution to the ILP problem of $\boldsymbol{x}$ determines a set $\boldsymbol{T}^{-1}(Z)$. The diagram of finding $\boldsymbol{T}^{-1}(Z)$ is illustrated in the following Fig. 3.

According to Lemma 1, since Z is a zonotope that satisfies $\mathrm{rank}(\boldsymbol{G}) = n = k$, and $\boldsymbol{T}$ defined in Eq. (16) is invertible, then the set $\boldsymbol{T}^{-1}(Z)$ is also a zonotope that satisfies $\mathrm{rank}(\boldsymbol{G}') = n = k$ with $\boldsymbol{G}'$ defined in Eq. (8). Therefore, we find an optimal oriented zonotope $\boldsymbol{T}^{-1}(Z)$ which aligns with the principal axes of the original data samples $\boldsymbol{x}_k = (x_{k1},\ldots,x_{kn}) \in \mathbb{R}^n, k \in \mathbb{M}$.

5. Solution method

In this section, a heuristic algorithm is developed to efficiently solve the ILP optimization problem formulated in the last section.

5.1. Optimal ILP algorithm

The cutting planes and branching algorithm built in Gurobi can be utilized to solve an ILP optimization problem and find a globally optimal solution. Please refer to [33] for details of the algorithm. Based on cutting planes and branching algorithm built in Gurobi, Algorithm 2 is developed to solve the formulated ILP optimization framework Eq. (14). An optimal solution of $z_{i_1\cdots i_n} \in \{0,1\}, i_1,\ldots,i_n \in \mathbb{N}$ to the ILP problem Eq. (14) can be found by implementing Algorithm 2, which determines an optimal zonotope which approximates the probabilistic geofence identified in Algorithm 1.

Algorithm 2 Optimal ILP Algorithm

```
function Gurobi($\alpha$, $\boldsymbol{g}_{\boldsymbol{u}}$, $\boldsymbol{w}$)
    $\bar{i}_1, \cdots, \bar{i}_n = \arg\max(\boldsymbol{w})$
    Formulate Eq. (14) on grid points $\boldsymbol{g}_{\boldsymbol{u}}$ weighted by $\boldsymbol{w}$
    at confidence level $\alpha$
    Use Gurobi to solve Eq. (14) for $z_{i_1\cdots i_n}$
    return $z_{i_1\cdots i_n}$

function OptimalZonotope($\alpha$, $\boldsymbol{x}_k$)
    Get $n$ linear independent unit vectors $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n$
    representing the orientation of $M$ data samples $\boldsymbol{x}_k$
    Get $N^n$ grid points $\boldsymbol{g}_{\boldsymbol{x}}$ aligning with $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n$
    $\boldsymbol{A} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n]^{-1}$
    $\bar{\boldsymbol{x}} = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{x}_k$
    Get a bijective affine transformation
    $\boldsymbol{T}(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{x} - \bar{\boldsymbol{x}}) + \bar{\boldsymbol{x}}, \boldsymbol{x} \in \mathbb{R}^n$
    Get $N^n$ grid points $\boldsymbol{g}_{\boldsymbol{u}} = \boldsymbol{T} \circ \boldsymbol{g}_{\boldsymbol{x}}$ aligning with the
    axes of the coordinate system
    Evaluate KDE $\tilde{f}_{\boldsymbol{u}}$ on $\boldsymbol{g}_{\boldsymbol{u}}$ through implementing
    Algorithm 1 and get the normalized weight matrix $\boldsymbol{w}$
    $z_{i_1\cdots i_n}$ = Gurobi($\alpha$, $\boldsymbol{g}_{\boldsymbol{u}}$, $\boldsymbol{w}$)
    Determine the optimal axis-aligned zonotope $Z$
    by the solution $z_{i_1\cdots i_n}$
    Determine the optimal oriented zonotope $\boldsymbol{T}^{-1}(Z)$
    return $\boldsymbol{T}^{-1}(Z)$

OptimalZonotope($\alpha$, $\boldsymbol{x}_k$)
```

5.2. Heuristic ILP algorithm

Implementing the cutting planes and branching algorithm to solve the formulated ILP optimization framework Eq. (14) for an optimal solution is very computationally expensive. To overcome this issue, an alternative heuristic algorithm is developed to efficiently solve Eq. (14) for a near-optimal solution of $z_{i_1\cdots i_n} \in \{0,1\}, i_1,\ldots,i_n \in \mathbb{N}$ instead of an optimal solution, which determines a near-optimal zonotope to approximate the probabilistic geofence identified in Algorithm 1.

The procedure of the proposed heuristic algorithm is displayed in Algorithm 3. Its main idea is summarized as follows:

(1) The grid point $(i_1,\dots,i_n)$ whose weight is the greatest is selected to be covered by the near-optimal zonotope, namely, $z_{i_1\cdots i_n}=1$;

(2) The grid point $(i'_1,\dots,i'_n)$ whose weight is the second greatest is selected to be covered, in other words, $z_{i'_1\cdots i'_n}=1$. Also, all the grid points $(j_1,\dots,j_n)$ in $\left\{(j_1,\dots,j_n) : \min(i_1,i'_1)\le j_1\le\max(i_1,i'_1),\dots,\min(i_n,i'_n)\le j_n\le\max(i_n,i'_n)\right\}$ are selected;

(3) If the sum of weights of the selected grid points exceeds the confidence level α, jump to (5). Otherwise, the grid point $(i''_1,\dots,i''_n)$ whose weight is the third greatest and all the grid points in $\left\{(j_1,\dots,j_n) : \min(i_1,i'_1,i''_1)\le j_1\le\max(i_1,i'_1,i''_1),\dots,\min(i_n,i'_n,i''_n)\le j_n\le\max(i_n,i'_n,i''_n)\right\}$ are selected;

(4) Repeat Step (3) to select the grid point whose weight is the fourth greatest, fifth greatest $\cdots$, and all the grid points among the first four greatest grids, five greatest grids $\cdots$, until the sum of weights of the selected grid points exceeds the confidence level α;

(5) Then, the selection of grid points gives a near-optimal solution $z_{i_1\cdots i_n}, i_1,\dots,i_n\in\mathbb{N}$ to the ILP optimization problem Eq. (14), which determines a near-optimal zonotope that we want.

Algorithm 3 Heuristic ILP Algorithm

```
function ILPHeuristic($\alpha$, $\boldsymbol{g_u}$, $\boldsymbol{w}$)
    $\boldsymbol{w}'$= $\boldsymbol{w}$.copy()
    $z_{i_1\cdots i_n}[1:N]\cdots[1:N]=0$
    $i_{1\min},\cdots,i_{n\min}=\arg\max(\boldsymbol{w}')$
    $i_{1\max}=i_{1\min},\cdots,i_{n\max}=i_{n\min}$
    while $\sum_{i_1,\cdots,i_n=1}^{N} w_{i_1\cdots i_n}z_{i_1\cdots i_n}<\alpha$ do
        $i_1,\cdots,i_n=\arg\max(\boldsymbol{w}')$
        $\boldsymbol{w}'[i_1]\cdots[i_n]=0.0$
        $i_{1\min}=\min(i_{1\min},i_1)$, $i_{1\max}=\max(i_{1\max},i_1)$
        $\vdots$
        $i_{n\min}=\min(i_{n\min},i_n)$, $i_{n\max}=\max(i_{n\max},i_n)$
        $z_{i_1\cdots i_n}[i_{1\min}:i_{1\max}]\cdots[i_{n\min}:i_{n\max}]=1$
    return $z_{i_1\cdots i_n}$

function ApproxZonotope($\alpha$, $\boldsymbol{x}_k$)
    Get $n$ linear independent unit vectors $\boldsymbol{e}_1,\cdots,\boldsymbol{e}_n$
    representing the orientation of $M$ data samples $\boldsymbol{x}_k$
    Get $N^n$ grid points $\boldsymbol{g_x}$ aligning with $\boldsymbol{e}_1,\cdots,\boldsymbol{e}_n$
    $\boldsymbol{A}=[\boldsymbol{e}_1,\cdots,\boldsymbol{e}_n]^{-1}$
    $\bar{\boldsymbol{x}}=\frac{1}{M}\sum_{k=1}^{M}\boldsymbol{x}_k$
    Get a bijective affine transformation
    $\boldsymbol{T}(\boldsymbol{x})=\boldsymbol{A}(\boldsymbol{x}-\bar{\boldsymbol{x}})+\bar{\boldsymbol{x}}$, $\boldsymbol{x}\in\mathbb{R}^n$
    Get $N^n$ grid points $\boldsymbol{g_u}=\boldsymbol{T}\circ\boldsymbol{g_x}$ aligning with the
    axes of the coordinate system
    Evaluate KDE $\tilde{f}_{\boldsymbol{u}}$ on $\boldsymbol{g_u}$ through implementing
    Algorithm 1 and get the normalized weight matrix $\boldsymbol{w}$
    $z_{i_1\cdots i_n}$ = ILPHeuristic($\alpha$, $\boldsymbol{g_u}$, $\boldsymbol{w}$)
    Determine the near-optimal axis-aligned zonotope $Z$
    by the solution $z_{i_1\cdots i_n}$
    Determine the near-optimal oriented zonotope $\boldsymbol{T}^{-1}(Z)$
    return $\boldsymbol{T}^{-1}(Z)$

ApproxZonotope($\alpha$, $\boldsymbol{x}_k$)
```

6. Main results

In this section, we conduct comprehensive case studies to compare the performance of our proposed algorithm of Heuristic ILP with another algorithm of Optimal ILP, Gaussian fit, the oriented bounding box and the convex hull of probabilistic geofence [19]. All the algorithms provide different convex sets to approximate the probabilistic geofence. We also conduct robustness analysis which demonstrates the computational efficiency of the Heuristic ILP algorithm on finding a near-optimal solution while enjoying accuracy and robustness, in contrast to the Optimal ILP algorithm. The tests were implemented in Python 3.9 and on an Intel(R) Core(TM) i9-12900KF, 3187 Mhz, 16 Core(s), 24 Logical Processor(s) Desktop with 64 GB RAM.

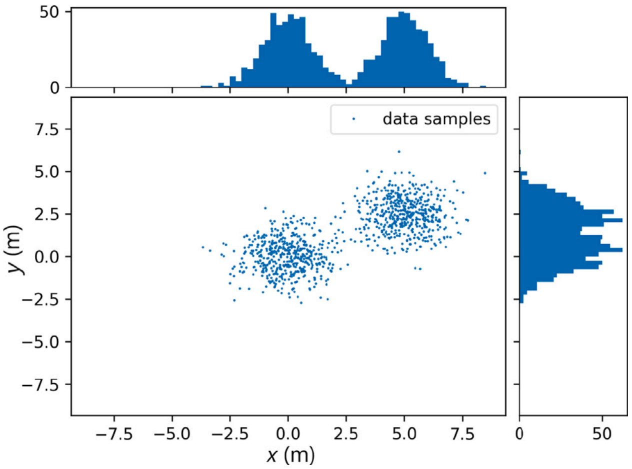


Fig. 4. Joint bimodal distribution generated by two marginal histograms.

There are two main stages in the case studies:

(1) Utilizing KDE values and the probabilistic geofence derived from a collection of M data samples using the methodology outlined in Algorithm 1, we construct an ILP optimization problem. Subsequently, we employ different algorithms to establish convex approximations of the probabilistic geofence;

(2) After obtaining these convex approximations, we generate a new collection of data samples whose size surpasses M significantly and assess the ratio of data samples within the convex approximation compared to the total number of generated data samples. As the number of data samples continues to increase, this ratio will tend towards the true probability of the system state which lies within the convex approximation.

6.1. Cases settings

6.1.1. Case I

In this case, we present a scatter plot depicting the possible instantaneous positions (data samples) (x,y) of a vehicle on the plane, which are generated by the marginal histograms of longitudinal position x and latitudinal position y respectively. As shown in Fig. 4, the instantaneous position of the vehicle obeys a non-Gaussian bimodal distribution.

6.1.2. Case II

In this case, we consider a free-flyer robot of six degrees of freedom under uncertainties in [34], whose system state is $\boldsymbol{x}=(\boldsymbol{p},\boldsymbol{v},\boldsymbol{q},\boldsymbol{\omega})\in\mathbb{R}^{13}$, control input is $\boldsymbol{u}=(\boldsymbol{F},\boldsymbol{M})\in\mathbb{R}^6$, and dynamics are $\dot{\boldsymbol{p}}=\boldsymbol{v}$, $m\dot{\boldsymbol{v}}=\boldsymbol{F}$, $\dot{\boldsymbol{q}}=\frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\boldsymbol{q}$, $\boldsymbol{J}\dot{\boldsymbol{\omega}}=\boldsymbol{M}-\boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{J}\boldsymbol{\omega}$ where $\boldsymbol{J}=\mathrm{diag}\left(\left[J_x,J_y,J_z\right]\right)$. We discretize the dynamics with $\Delta t=1\,\mathrm{s}$ such that $\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+\boldsymbol{f}_k(\boldsymbol{x}_k,\boldsymbol{u}_k,\boldsymbol{\theta}_k)\Delta t+\boldsymbol{w}_k$ where $\boldsymbol{w}_k\sim\mathcal{N}\left(\boldsymbol{0},\boldsymbol{\Sigma_w}\right)$ are i.i.d. disturbances. The mass and inertia are unknown with known bounds $m\in[7.1,7.3]$, $J_i\in[0.065,0.075]$, $\left|w_{ki}\right|\le 10^{-4}$ for $i=1,\dots,13$, and $\left|w_{ki}\right|\le 5\times10^{-4}$ for $i=4,5,6$. Our proposed method in the last section can find a 13-dimensional zonotopic approximation for the system state $\boldsymbol{x}\in\mathbb{R}^{13}$ of the robot. In the applications of AAM, we usually just need to consider 3D positions $\boldsymbol{p}\in\mathbb{R}^3$. Thus, we will find a 3D zonotopic approximation for the 3D projection $\boldsymbol{p}\in\mathbb{R}^3$ of the system state $\boldsymbol{x}\in\mathbb{R}^{13}$ at time step $k=20\,\mathrm{s}$.

In addition to the parameters in both cases, the bandwidth matrix $\boldsymbol{H}$ in Section 3.2 is $\left[\begin{smallmatrix}0.2 & 0\\ 0 & 0.2\end{smallmatrix}\right]$. The number of data samples is $M=500$ for

Case I and $M = 1000$ for Case II. In Case I, we establish a baseline with the following parameters: the confidence level $\alpha = 90\%$, the number of grid points $N^2 = 20^2$. In Case II, the baseline configuration includes the confidence level $\alpha = 90\%$ and the number of grid points $N^3 = 10^3$. In the subsequent figures and tables, the baseline is indicated using the symbol "*".

6.2. Different numbers of grid points

In this part, we compare the performance of different algorithms given different numbers N of grid points in each dimension. For Case I, $N^2 = 15^2$, 20^2, and 25^2. For Case II, $N^3 = 8^3$, 10^3, and 12^3. The other parameters are the same as the baseline.

For Case I, the impact of different numbers of grid points is shown in Figs. 5(a), 5(c), 5(e), and Table 1. The blue points indicate data samples. The black lattice indicates grid points. The probabilistic geofence in 2D space obtained by Algorithm 1 is represented by the yellow contour. The yellow box depicts the bounding box of this probabilistic geofence, while the green ellipse corresponds to the outcome of Gaussian fit. The blue (resp. red) box indicates the zonotope obtained by the algorithm of Optimal ILP (resp. Heuristic ILP), which is determined by the optimal (resp. near-optimal) solution to the ILP optimization problem Eq. (14) through the implementation of Algorithm 2 (resp. Algorithm 3). For Case II, the results are illustrated in Figs. 5(b), 5(d), 5(f), and Table 2. All the labels are the same as those in Case I except that the cyan box indicates the bounding box of the probabilistic geofence in 3D space obtained by Algorithm 1, and the green ellipsoid is obtained by Gaussian fit.

For Case I, the results obtained by different algorithms are different from each other. There can be a notable difference between the zonotope obtained by Optimal ILP and that obtained by Heuristic ILP, as shown in Fig. 5(a). This is because Heuristic ILP usually finds a near-optimal solution while Optimal ILP guarantees an optimal one. As shown in Table 1, compared with the ellipse of Gaussian fit and the bounding box, the area of the zonotope of Optimal ILP or Heuristic ILP is much smaller, which means the zonotopic approximation provided by Optimal ILP or Heuristic ILP for the probabilistic geofence is tighter. Also, the ratio of Optimal ILP or Heuristic ILP is much more accurate in the sense that the gap between the confidence level 90% and the ratio of Optimal ILP or Heuristic ILP is smaller. For example, when $N^2 = 20^2$, the ratio 90.8% of Optimal ILP or Heuristic ILP is more accurate than 97.8% of bounding box and 93.2% of Gaussian fit. This demonstrates the advantage of Optimal ILP or Heuristic ILP in terms of accuracy. Therefore, compared with the result of bounding box or Gaussian fit which can be overly conservative, the zonotopic approximation of probabilistic geofence is less conservative and more accurate. Thus, Optimal ILP or Heuristic ILP can provide a larger feasible airspace for AAM.

As indicated in Table 1, for Optimal ILP and Heuristic ILP in Case I, as the number of grid points rises from 15^2 to 20^2, the performance in terms of accuracy greatly improves while the area of zonotope increases. For example, the gap 0.8% between the ratio of Optimal ILP (resp. Heuristic ILP) and the confidence level is less than the gap 3.8% (resp. 1.7%) between the ratio of Optimal ILP (resp. Heuristic ILP) and the confidence level. At the same time, the area increases from $39.1\,\mathrm{m}^2$ (resp. $40.2\,\mathrm{m}^2$) of Optimal ILP (resp. Heuristic ILP) to $43.1\,\mathrm{m}^2$ of Optimal ILP (resp. Heuristic ILP). This makes sense because pursuing tightness cannot contradict the priority of ensuring accuracy. As the number of grid points rises from 20^2 to 25^2, the performance in terms of accuracy still improves while the area slightly increases. For example, for both algorithms, the gap between ratio and confidence level drops from 0.8% to 0.2% while the area slightly increases from $43.1\,\mathrm{m}^2$ to $43.6\,\mathrm{m}^2$. This suggests we can obtain more accurate results without increasing too much conservatism. However, for Optimal ILP, the increase in the number of decision variables and constraints arising from the increasing number of grid points leads to a significant increase in computational time, from 15 s to 376 s. Hence, optimality comes at the cost of computational efficiency. In contrast, the computational time of Heuristic ILP is very short, from 0.01 s to 0.03 s, while the zonotope obtained by Heuristic ILP is almost the same as Optimal ILP. These results suggest although losing optimality, Heuristic ILP guarantees near-optimality, accuracy, and efficiency.

For Case II, similar trends are followed, as shown in Figs. 5(b), 5(d), 5(f), and Table 2. Accordingly, Heuristic ILP outperforms the other three algorithms in the sense that it effectively trades off computational efficiency and accuracy.

Table 1
Case I comparisons of different numbers of grid points in each dimension N (Gap = Absolute Value of Ratio — Confidence Level).

Algorithm	Case I			
	# Grid points	Ratio/Gap	Area (m^2)	Time (s)
Optimal ILP	15^2	86.2%/3.8%	39.1	15
Heuristic ILP		88.3%/1.7%	40.2	0.01
Gaussian fit		93.5%/3.5%	43.6	0.01
Bounding box		98.0%/8.0%	54.9	0.02
Optimal ILP	20^2 *	90.8%/0.8%	43.1	93
Heuristic ILP		90.8%/0.8%	43.1	0.02
Gaussian fit		93.2%/3.2%	43.4	0.01
Bounding box		97.8%/7.8%	54.8	0.02
Optimal ILP	25^2	90.2%/0.2%	43.6	376
Heuristic ILP		90.2%/0.2%	43.6	0.03
Gaussian fit		92.8%/2.8%	44.5	0.01
Bounding box		97.0%/7.0%	53.9	0.03

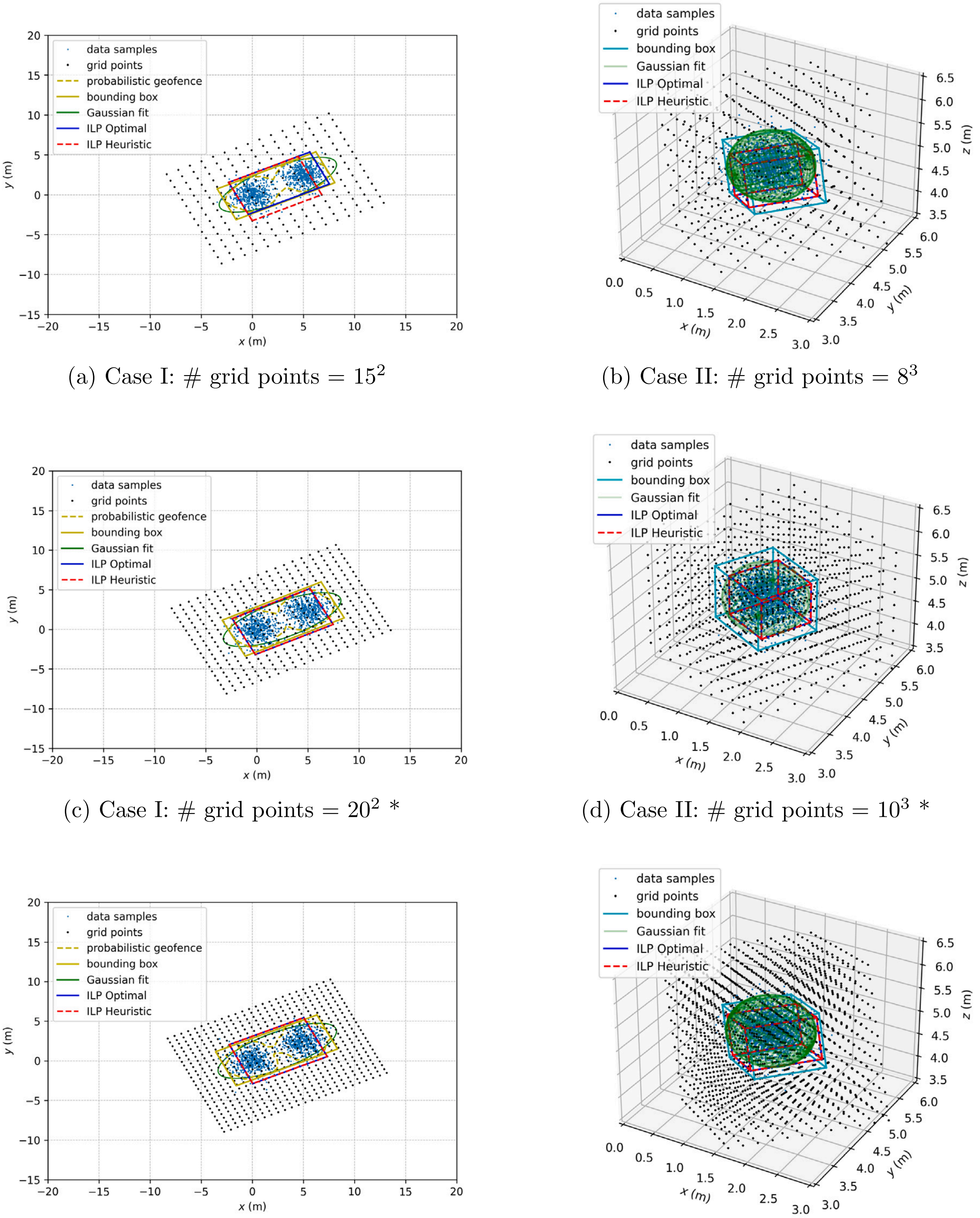
Table 2
Case II comparisons of different numbers of grid points in each dimension N (Gap = Absolute Value of Ratio — Confidence Level).

Algorithm	Case II			
	# Grid points	Ratio/Gap	Volume (m^3)	Time (s)
Optimal ILP	8^3	84.2%/4.2%	0.55	161
Heuristic ILP		84.2%/4.2%	0.55	0.01
Gaussian fit		93.9%/3.9%	0.99	0.02
Bounding box		98.1%/8.1%	1.31	0.01
Optimal ILP	10^3 *	91.1%/1.1%	0.69	916
Heuristic ILP		91.1%/1.1%	0.69	0.01
Gaussian fit		93.8%/3.8%	1.04	0.02
Bounding box		97.9%/7.9%	1.32	0.02
Optimal ILP	12^3	90.8%/0.8%	0.72	8376
Heuristic ILP		90.8%/0.8%	0.72	0.02
Gaussian fit		92.6%/2.6%	1.03	0.02
Bounding box		97.4%/7.4%	1.29	0.02

6.3. Different confidence levels

In this part, we compare the performance of different algorithms for both Case I and Case II with respect to three different confidence levels: $\alpha = 90\%$, 95%, and 99%. All the other parameters remain consistent with the baseline.

As shown in Fig. 6 and Tables 3, 4, for both Case I and Case II, when the confidence level is fixed, the zonotope of Optimal ILP or Heuristic ILP is tighter than the ellipse (or ellipsoid) of Gaussian fit and bounding box. Also, the ratio of Optimal ILP or Heuristic ILP is more accurate. For example, for Case II with respect to $\alpha = 90\%$, the volume $0.69\,\mathrm{m}^3$ of the zonotope obtained by Optimal ILP or Heuristic ILP is less than $1.04\,\mathrm{m}^3$ of the ellipsoid of Gaussian fit and $1.32\,\mathrm{m}^3$ of bounding box. Also, the gap between the confidence level 90% and the ratio 91.1% of Optimal ILP or Heuristic ILP is significantly smaller than 93.8% of Gaussian fit and 97.9% of Bounding box. Therefore, Optimal ILP or Heuristic ILP outperforms Gaussian fit and bounding box in terms of optimality and accuracy, and finds a zonotope that provides a convex approximation for the probabilistic geofence.



(a) Case I: # grid points = 15^2

(b) Case II: # grid points = 8^3

(c) Case I: # grid points = 20^2 *

(d) Case II: # grid points = 10^3 *

(e) Case I: # grid points = 25^2

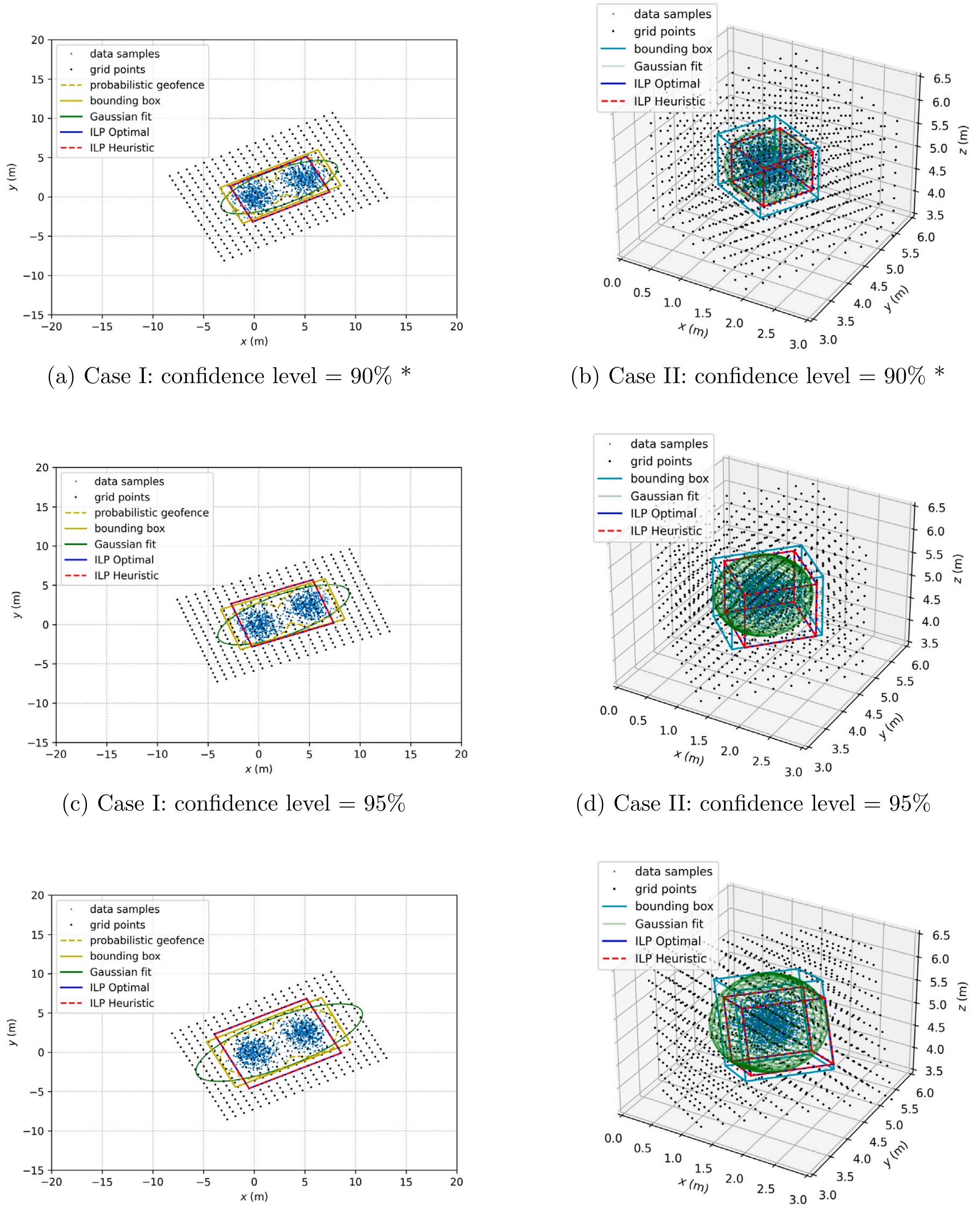
(f) Case II: # grid points = 12^3

Fig. 5. Comparisons of different numbers of grid points in each dimension N.

For Case I (resp. Case II), as the confidence level increases, the area (resp. volume) of the zonotope obtained by Optimal ILP or Heuristic ILP also increases, while the computational time slightly fluctuates. Given a common confidence level, the area (resp. volume) of the zonotope obtained by Heuristic ILP is close to that obtained by Optimal ILP. However, Heuristic ILP greatly outperforms Optimal ILP in terms of computational efficiency. For example, given a confidence level 90%, for Case II, the computational time 0.02 s of Heuristic ILP is greatly shorter than 916 s of Optimal ILP, while the volumes 0.69 m^3 of both zonotopes obtained by two different algorithms are the same. Therefore, Heuristic ILP enjoys computational efficiency without losing accuracy.

6.4. Zonotopic approximation versus convex hull

In addition to the algorithm of the bounding box, the algorithm of the convex hull is another existing method that can provide a convex bound for the probabilistic geofence. For Case I and II, we run



(a) Case I: confidence level = 90% *

(b) Case II: confidence level = 90% *

(c) Case I: confidence level = 95%

(d) Case II: confidence level = 95%

(e) Case I: confidence level = 99%

(f) Case II: confidence level = 99%

Fig. 6. Comparisons of different confidence levels α.

both the Optimal ILP and Heuristic ILP algorithms to obtain zonotopic approximations of the probabilistic geofence respectively, and compare them with the bounding box and convex hull. The parameters stay the same as the baseline.

The results of both Case I and II are shown in Figs. 7(a), 7(b) and Table 5. On the one hand, compared with the result of the bounding box, the result of the convex hull is tighter and more accurate. For example, for Case I shown in Table 5, the area of convex hull 43.2 m^2 is less conservative than bounding box 54.8 m^2, while the ratio of convex hull 91.0% is far more accurate than bounding box 97.8%. On the other hand, our proposed algorithm of Heuristic ILP beats the method of the convex hull in terms of convenience while still enjoying accuracy and computational efficiency. For instance, Case II, Table 5 shows that the zonotope obtained by Heuristic ILP is almost as accurate as the convex hull in the sense that the ratio/volume of Heuristic ILP 91.1%/0.69 m^3 is almost the same as that of convex hull 91.3%/0.71 m^3. In addition, Heuristic ILP 0.01 s runs faster than the method of convex hull 0.02 s, which demonstrates its computational efficiency against

Table 3
Case I comparisons of different confidence levels α (Gap = Absolute Value of Ratio — Confidence Level).

Algorithm	Case I			
	Confidence level	Ratio/Gap	Area (m^2)	Time (s)
Optimal ILP	90% *	90.8%/0.8%	43.1	93
Heuristic ILP		90.8%/0.8%	43.1	0.02
Gaussian fit		93.2%/3.2%	43.4	0.01
Bounding box		97.8%/7.8%	54.8	0.02
Optimal ILP	95%	95.4%/0.4%	53.4	77
Heuristic ILP		95.4%/0.4%	53.4	0.02
Gaussian fit		96.9%/1.9%	58.2	0.01
Bounding box		98.9%/3.9%	62.8	0.02
Optimal ILP	99%	99.4%/0.4%	78.4	66
Heuristic ILP		99.4%/0.4%	78.4	0.03
Gaussian fit		99.8%/0.8%	87.1	0.01
Bounding box		99.9%/0.9%	81.7	0.03

Table 4
Case II comparisons of different confidence levels α (Gap = Absolute Value of Ratio — Confidence Level).

Algorithm	Case II			
	Confidence level	Ratio/Gap	Volume (m^3)	Time (s)
Optimal ILP	90% *	91.1%/1.1%	0.69	916
Heuristic ILP		91.1%/1.1%	0.69	0.01
Gaussian fit		93.8%/3.8%	1.04	0.02
Bounding box		97.9%/7.9%	1.32	0.02
Optimal ILP	95%	95.8%/0.8%	0.97	861
Heuristic ILP		95.8%/0.8%	0.97	0.01
Gaussian fit		97.3%/2.3%	1.45	0.02
Bounding box		99.2%/4.2%	1.60	0.02
Optimal ILP	99%	99.3%/0.3%	1.59	780
Heuristic ILP		99.3%/0.3%	1.59	0.01
Gaussian fit		99.9%/0.9%	2.57	0.02
Bounding box		99.9%/0.9%	2.46	0.02

convex hull. However, as illustrated in Figs. 7(a) and 7(b), the number of facets (also called hyperfaces) of the convex hull is undetermined and a convex hull usually has too many facets. If we want to avoid collision in path planning, we need to formulate constraints that a point is outside the convex hull and every facet will lead to a constraint. Thus, too many facets can lead to too many constraints, which is not convenient for the operations in practice. In contrast, Heuristic ILP can overcome this limitation. In n-dimensional space, the number of facets of the zonotope obtained by Heuristic ILP is always $2n$. Therefore, our proposed Heuristic ILP algorithm performs as well as the method of convex hull in terms of accuracy and computational efficiency, and outperforms convex hull when it comes to the convenience of AAM operations.

6.5. Robustness of heuristic ILP algorithm

We have shown that compared with Optimal ILP, the proposed Heuristic ILP algorithm acquires computational efficiency at the cost of optimality when providing zonotopic approximation for the probabilistic geofence. However, there can be a difference between the two zonotopes obtained by these two algorithms. Also, given the same parameters, different data samples may lead to different zonotopes obtained by each algorithm. This raises the following concerns regarding the robustness of Heuristic ILP: (1) Space: we want to quantify the difference between the zonotope of Heuristic ILP and that of Optimal ILP every time, and evaluate whether the difference varies significantly at different times; (2) Time: we would also like to evaluate whether the computational time of Heuristic ILP differs a lot at different times under same condition. A major difference at different times means Heuristic ILP is not robust enough for applications in real scenarios.

Jaccard distance can be introduced to quantify the difference between the zonotope of Optimal ILP and that of Heuristic ILP. Formally speaking, given two sets S_1 and S_2, Jaccard distance between those two sets is defined to be [27]

$$d(S_1, S_2) = 1 - \frac{\mathrm{Card}(S_1 \cap S_2)}{\mathrm{Card}(S_1 \cup S_2)},$$

where the operation Card finds the cardinality of a set. A smaller Jaccard distance indicates a higher degree of similarity between two sets.

For both Case I and II, we analyze the Jaccard distance between the zonotopes obtained by Heuristic ILP and Optimal ILP with respect to the increasing number of grid points. All the other parameters are the same as the baseline. Given a fixed number of grid points, we generate different data samples twenty times in total and record the computational time of each algorithm and the Jaccard distance between the zonotopes obtained by both algorithms every time.

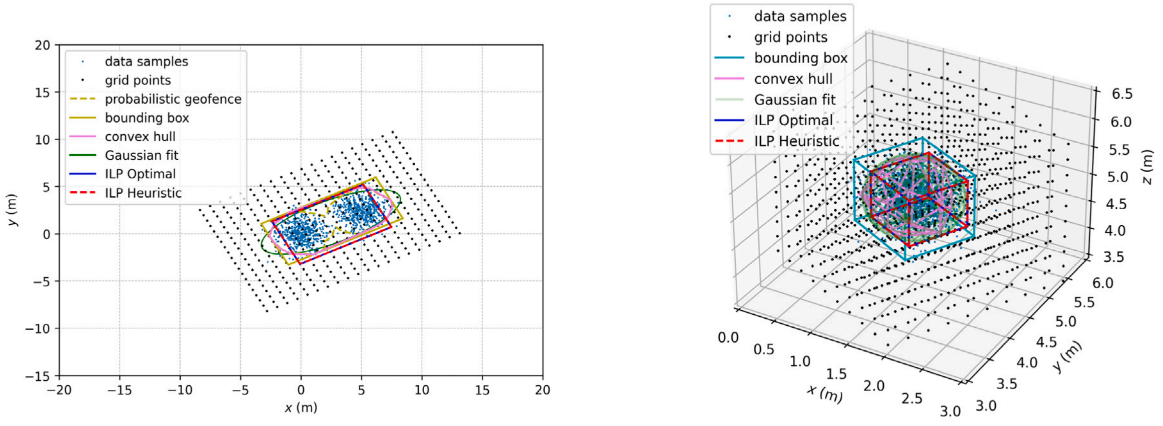
The results are illustrated in Fig. 8. On the one hand, for both cases, Figs. 8(b) and 8(d) show that when the number of grid points in each dimension is too small, different Jaccard distances fluctuate significantly around the average value, suggesting that Heuristic ILP has poor robustness in space under this condition. As N increases from 10 to 25 for Case I (resp. from 8 to 14 for Case II), the average Jaccard distance decreases from 0.15 to 0.06 for Case I (resp. from 0.17 to 0.07 for Case II) respectively. This implies that as the number of grid points increases, the zonotope acquired through Heuristic ILP approaches a more optimal solution. In addition, the spread of different Jaccard distances also decreases as N increases, which suggests that Heuristic ILP becomes more robust in space with respect to the increase of the number of grid points. On the other hand, Figs. 8(a) and 8(c) indicate that as N increases, the computational time of Heuristic ILP slowly increases from $0.01\,\mathrm{s}$ to $0.03\,\mathrm{s}$ for both cases. Additionally, as the number of grid points grows, the spread of computational time increases not significantly. Hence, Heuristic ILP enjoys computational efficiency and robustness in terms of time.

Therefore, the increase in the number of grid points slightly intensifies the computational effort while reducing the difference between the zonotopes obtained by Heuristic ILP and Optimal ILP. We would like to tune an appropriate parameter for the number of grid points, which makes Heuristic ILP robust enough in space and time. By doing so, Heuristic ILP enjoys near-optimality, accuracy, efficiency, and robustness simultaneously.

In practice, the information about uncertainty changes over time, and the distribution captured by the algorithm is also updated with more data samples coming in, which brings a great challenge requiring that the proposed algorithm must run in real-time to perform recomputation every time newly observed data samples are added. Our proposed method of Heuristic ILP, which consists of KDE computation and solving ILP problems for zonotopic approximation, can handle this concern very well. As indicated in Figs. 8(a) and 8(c), given a set of data samples, the computational time of Heuristic ILP is under $0.05\,\mathrm{s}$. Hence, the whole process of KDE computation together with zonotopic approximation can be updated in an online fashion. Every time the set of data samples is updated with new data samples coming in, we can rapidly recompute a new zonotopic approximation. This demonstrates the computational power of our proposed method and thus it can handle streaming data online, which is well-suited for the operations of AAM in real-time.

7. Conclusion

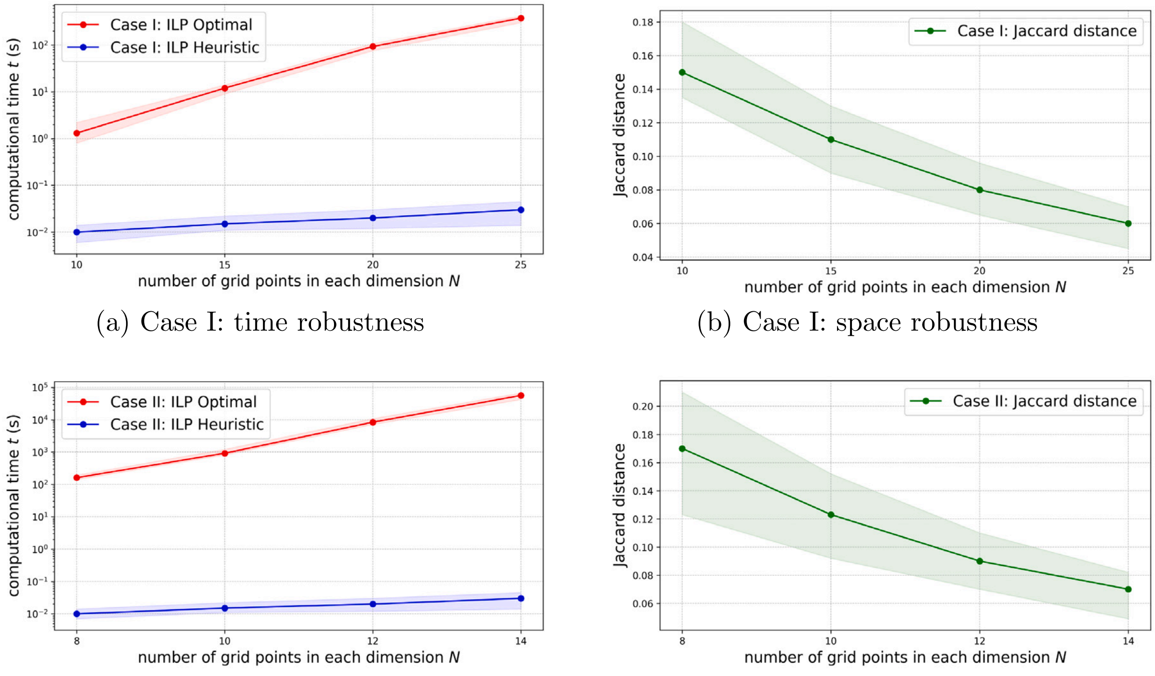
How to ensure safety in real-time in high-dense, dynamic, and uncertain airspace environments is a pivotal challenge for the successful operation of AAM. In this paper, we present an online algorithm to identify the probabilistic geofence of an arbitrary unknown probability distribution using the data-driven approach of FFT-based KDE. How-



(a) Case I: zonotope versus convex hull *

(b) Case II: zonotope versus convex hull *

Fig. 7. Zonotopic approximation versus convex hull of probabilistic geofence.



(a) Case I: time robustness

(b) Case I: space robustness

(c) Case II: time robustness

(d) Case II: space robustness

Fig. 8. Robustness analysis of Heuristic ILP algorithm.

Table 5
Zonotopic approximation versus convex hull of probabilistic geofence (Gap = Absolute Value of Ratio — Confidence Level).

Algorithm	Case I *			Case II *		
	Ratio/Gap	Area (m^2)	Time (s)	Ratio/Gap	Volume (m^3)	Time (s)
Optimal ILP	90.8%/0.8%	43.1	93	91.1%/1.1%	0.69	916
Heuristic ILP	90.8%/0.8%	43.1	0.02	91.1%/1.1%	0.69	0.01
Gaussian fit	93.2%/3.2%	43.4	0.01	93.8%/3.8%	1.04	0.02
Bounding box	97.8%/7.8%	54.8	0.02	97.9%/7.9%	1.32	0.02
Convex hull	91.0%/1.0%	43.2	0.02	91.3%/1.3%	0.71	0.02

ever, the irregularity or non-convexity of the shape of the probabilistic geofence refrains it from real applications. To address this issue, we formulate an ILP optimization problem whose solution determines an optimal zonotopic approximation of the probabilistic geofence. A computationally efficient algorithm of Heuristic ILP is then developed to solve the formulated ILP problem. Numerical studies are conducted to compare the performance of the Heuristic ILP algorithm with other algorithms ins providing convex approximations for the probabilistic geofence. Simulation results demonstrate that the proposed Heuristic ILP algorithm outperforms other algorithms in the sense that it enjoys the benefit of computational efficiency without losing accuracy. Also, the robustness analysis suggests that the Heuristic ILP algorithm can efficiently find a near-optimal solution while enjoying accuracy and robustness.

Future work will be dedicated to applying the method we proposed in this paper to the safe operations of AAM under uncertainty in real-time. Also, one significant limitation of our work is that under the environment where the obstacles are densely populated, the data samples of different obstacles may mess up, but our current method cannot tell the difference and thus is not able to find a zonotopic approximation of probabilistic geofencing for distinct obstacles respectively. This will be part of our future work.

CRediT authorship contribution statement

Pengcheng Wu: Writing – original draft, Methodology, Formal analysis. **Jun Chen:** Writing – review & editing, Project administration, Funding acquisition, Conceptualization,.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jun Chen reports financial support was provided by National Science Foundation.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Science Foundation under Grants CMMI-2138612. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not reflect the views of NSF.

References

[1] Wu P, Xie J, Liu Y, Chen J. Risk-bounded and fairness-aware path planning for urban air mobility operations under uncertainty. Aerosp Sci Technol 2022;127:107738.
[2] Pang B, Hu X, Dai W, Low KH. UAV path optimization with an integrated cost assessment model considering third-party risks in metropolitan environments. Reliab Eng Syst Saf 2022;222:108399.
[3] Yang X, Wei P. Scalable multi-agent computational guidance with separation assurance for autonomous urban air mobility. J Guid Control Dyn 2020;43(8):1473–86.
[4] Zhou Z, Chen J, Liu Y. Optimized landing of drones in the context of congested air traffic and limited vertiports. IEEE Trans Intell Transp Syst 2021;22(9):6007–17.
[5] Bensaci C, Zennir Y, Pomorski D, Innal F, Lundteigen MA. Collision hazard modeling and analysis in a multi-mobile robots system transportation task with STPA and SPN. Reliab Eng Syst Saf 2023;234:109138.
[6] Wu P, Li L, Xie J, Chen J. Probabilistically guaranteed path planning for safe urban air mobility using chance constrained RRT. In: AIAA aviation 2020 forum. 2020, p. 2914.
[7] Zou Y, Zhang H, Zhong G, Liu H, Feng D. Collision probability estimation for small unmanned aircraft systems. Reliab Eng Syst Saf 2021;213:107619.
[8] Stevens M, Atkins E. Geofence definition and deconfliction for UAS traffic management. IEEE Trans Intell Transp Syst 2020.
[9] Cheng L, Wen H, Jin D. Uncertain parameters analysis of powered-descent guidance based on Chebyshev interval method. Acta Astronaut 2019;162:581–8.
[10] Han W, Jasour A, Williams B. Non-Gaussian risk bounded trajectory optimization for stochastic nonlinear systems in uncertain environments. In: 2022 international conference on robotics and automation (ICRA). IEEE; 2022, p. 11044–50.
[11] Wu P, Chen J. Efficient box approximation for data-driven probabilistic geofencing. Unmanned Syst 2023.
[12] Wu P, Yang X, Wei P, Chen J. Safety assured online guidance with airborne separation for urban air mobility operations in uncertain environments. IEEE Trans Intell Transp Syst 2022.
[13] Luders B, Kothari M, How J. Chance constrained RRT for probabilistic robustness to environmental uncertainty. In: AIAA guidance, navigation, and control conference. 2010, p. 8160.
[14] Boone S, McMahon J. Spacecraft maneuver design with non-gaussian chance constraints using gaussian mixtures. In: 2022 AAS/AIAA astrodynamics specialist conference. 2022.
[15] Zhang R, Dai H. A non-Gaussian stochastic model from limited observations using polynomial chaos and fractional moments. Reliab Eng Syst Saf 2022;221:108323.
[16] Liu W, Shao Y, Li C, Li C, Jiang Z. Development of a non-Gaussian copula Bayesian network for safety assessment of metro tunnel maintenance. Reliab Eng Syst Saf 2023;109423.
[17] Subramanian A, Mahadevan S. Importance sampling for probabilistic prognosis of sector-wide flight separation safety. Reliab Eng Syst Saf 2022;222:108410.
[18] Lan H, Ma X, Qiao W, Deng W. Determining the critical risk factors for predicting the severity of ship collision accidents using a data-driven approach. Reliab Eng Syst Saf 2023;230:108934.
[19] Schneider P, Eberly DH. Geometric tools for computer graphics. Elsevier; 2002.
[20] Lew T, Pavone M. Sampling-based reachability analysis: A random set theory approach with adversarial sampling. In: Conference on robot learning. PMLR; 2021, p. 2055–70.
[21] Althoff M, Stursberg O, Buss M. Safety assessment for stochastic linear systems using enclosing hulls of probability density functions. In: 2009 European control conference (ECC). IEEE; 2009, p. 625–30.
[22] Ericson C. Real-time collision detection. Crc Press; 2004.
[23] Härdle WK, Simar L. Applied multivariate statistical analysis. Springer Nature; 2019.
[24] Silverman BW. Density estimation for statistics and data analysis. Routledge; 2018.
[25] Wand MP, Jones MC. Kernel smoothing. CRC Press; 1994.
[26] Wu P, Chen J. Online probabilistic collision detection for urban air mobility under data-driven uncertainty. In: AIAA scitech 2023 forum. 2023, p. 2539.
[27] Wu P, Chen J. Online evaluation for chance-constrained geofences under data-driven uncertainties. In: AIAA aviation 2022 forum. 2022, p. 3613.
[28] Alanwar A, Koch A, Allgöwer F, Johansson KH. Data-driven reachability analysis using matrix zonotopes. In: Learning for dynamics and control. PMLR; 2021, p. 163–75.
[29] Althoff D, Wollherr D, Buss M. Safety assessment of trajectories for navigation in uncertain and dynamic environments. In: 2011 IEEE international conference on robotics and automation. IEEE; 2011, p. 5407–12.
[30] Althoff M, Frehse G, Girard A. Set propagation techniques for reachability analysis. Annu Rev Control Robot Auton Syst 2021;4:369–95.
[31] Gaßmann V, Althoff M. Scalable zonotope-ellipsoid conversions using the Euclidean zonotope norm. In: 2020 American control conference (ACC). IEEE; 2020, p. 4715–21.
[32] Casella G, Berger RL. Statistical inference. Cengage Learning; 2021.
[33] Mitchell JE. Branch-and-cut algorithms for combinatorial optimization problems. In: Handbook of applied optimization, Vol. 1. 2002, p. 65–77.
[34] Lew T, Bonalli R, Pavone M. Chance-constrained sequential convex programming for robust trajectory optimization. In: 2020 European control conference (ECC). IEEE; 2020, p. 1871–8.