Efficient Box Approximation for Data-Driven Probabilistic Geofencing

Pengcheng Wu[a], Jun Chen[b]

[a] *Department of Mechanical and Aerospace Engineering, University of California San Diego, La Jolla, CA 92093, US*
Department of Aerospace Engineering, San Diego State University, San Diego, CA 92182, US
E-mail: pwu2198@sdsu.edu, pcwupat@ucsd.edu

[b] *Department of Aerospace Engineering, San Diego State University, San Diego, CA, 92182, US*
E-mail: jun.chen@sdsu.edu

Advanced Air Mobility (AAM) using electrical vertical take-off and landing (eVTOL) aircraft is an emerging way of air transportation within metropolitan areas. A key challenge for the success of AAM is how to manage large-scale flight operations with safety guarantees in high-density, dynamic, and uncertain airspace environments in real time. To address these challenges, we introduce the concept of a data-driven probabilistic geofence, which can guarantee that the probability of potential conflicts between eVTOL aircraft is bounded under data-driven uncertainties. To evaluate the probabilistic geofences online, Kernel Density Estimation (KDE) based on Fast Fourier Transform (FFT) is customized to model data-driven uncertainties. Based on the FFT-KDE values from data-driven uncertainties, we introduce an optimization framework of Integer Linear Programming (ILP) to find a parallelogram box to approximate the data-driven probabilistic geofence. To overcome the computational burden of ILP, an efficient heuristic algorithm is further developed. Numerical results demonstrate the feasibility and efficiency of the proposed algorithms.

Keywords: Probabilistic Geofence; Box Approximation; Integer Linear Programming

1. Introduction

1.1. *Motivation and Related Work*

Due to the swift expansion of the unmanned aircraft system (UAS) industry, a dependable and effective system is required to manage the traffic of unmanned aerial vehicles (UAVs) and emerging new aircraft in the airspace.[1–4] The Advanced Air Mobility (AAM) system is designed for this mission, which includes Urban Air Mobility (UAM) and UAS traffic management (UTM). Around the world, various entities including industry, government, and academia have explored concepts in AAM. Airbus, Boeing, Bell, Joby, Archer, Lilium, and Aurora Flight Sciences are vying against each other to develop and test their recently designed electrical vertical take-off and landing (eVTOL) aircraft, as stated in.[5]

AAM, or UAM in specific, is expected to cater to a substantial portion of urban transportation demand by introducing numerous eVTOL aircraft in the limited urban airspace.[6,7] The environment in this airspace is highly uncertain due to the presence of tall buildings, resulting in inaccurate localization with disturbances such as strong Global Positioning System (GPS)[8] noise and high wind disturbance around these structures. Therefore, ensuring operation safety in real-time in high-density, dynamic, and uncertain airspace environments is a crucial obstacle that needs to be overcome for the success of UAM. Also, with the authorization of numerous eVTOL aircraft to enter the airspace, the airspace will become congested and surpass the capacity of the existing air traffic control system. To resolve this challenge, capable low-altitude traffic management is necessary to manage AAM. The existence of geofences is a key component of the traffic management system, which reserves airspace volumes as a feasible alternative to strictly route-based reservations.[9] The issue of an aircraft's location uncertainty or error is critical in geofencing to ensure safety.

Dynamic airspace geofencing algorithms are a recent addition to AAM and UTM. Geofencing designs have two equally crucial but different perspectives, namely local and global. The classical guidance, navigation, and control (GNC) approach is vehicle-centered and generates geofence layering solely for the individual eVTOL that has full knowledge of its control system. The aim of this perspective is to control the eVTOL and ensure that it does not breach the geofence boundaries (given expected trajectory tracing errors).[10] In this approach, each eVTOL monitors its real-time state vector concerning geofence boundaries to detect and react to possible breaches in the face of uncertainties due to sensor errors and wind disturbances.[11] Although vehicle-centered geofencing research is important, it does not account for the properties of the operating area airspace or the ground-based environment. Geofencing has also been studied from an airspace system perspective, where UTM manages geofences to organize the airspace structure and enhance Situational Awareness (SA). In this viewpoint, UTM will not model individual eVTOL capabilities and uncertainties in detail. Still, it can conservatively track eVTOL travel through an approved geofence to issue impending breach warnings to the eVTOL and actual boundary violations to all traffic. This paper introduces a concept called data-driven probabilistic geofences which can guarantee that the probability of potential conflicts is bounded under data-driven uncertainties. We further develop algorithms to efficiently evaluate the data-driven probabilistic geofences and to provide a parallelogram box to approximate the geofence in an online fashion.

Geofencing's design centers on determining a collision-free geofence-based flight trajectory. We evaluated numerous path planning algorithms in this regard. Ensuring collision-free flight trajectories is crucial for safe and reliable flying. However, in real-world scenarios, uncertainties can be present, which may make it difficult to guarantee feasibility using the current collision avoidance system. In such cases, identifying trajectories that ensure the probability of conflict occurrence is within a specified limit is a more practical approach.[12] Therefore, taking uncertainty into account in path planning is essential to ensure safety. Several methods have been proposed to address this issue. For example, probabilistic maps can be constructed using a likelihood function, and safe UAV paths can be generated by solving a probability minimization problem.[13] Although the sampling-based Monte Carlo method is accurate enough to estimate conflicts between aircraft stochastically, it is computationally expensive, and it takes a lot of time to compute the probability of conflict occurrence.[14,15] To balance planning conservatism and efficiency, stochastic constraints can be reformulated as tightened deterministic constraints through chance constraint formulation.[16,17] Chance-constrained programming models have been proposed to consider various uncertainties for conflict avoidance problems.[18,19] However, such formulations are computationally expensive and may not scale well as the dimensions of configuration spaces increase. Wu et al.[11] proposed a chance-constrained algorithm CCRRT that uses sampling-based methods to identify paths for linear systems subject to uncertainty. Sampling-based algorithms like CCRRT scale well because they perform trajectory-wise constraint checking.[20]

The previous studies have assumed that the uncertainty follows a Gaussian distribution, which may not align with practical scenarios. Some researchers have attempted to approximate non-Gaussian distributions with Gaussian distributions, even though this method may not be accurate and could compromise the safety constraints regarding the collision probability along the generated path. For instance, in,[21] researchers still use Gaussian distributions to approximate non-Gaussian ones. To address this issue, some studies have proposed non-Gaussian distribution-based methods.[22,23] Recent work by Han et al.[24] has shown that using Gaussian approximations can lead to suboptimal solutions and that considering non-Gaussian distributions is crucial for ensuring safety in collision avoidance problems.

This paper is dedicated to approximating probabilistic geofences using data obtained from experiments or simulations. Existing works, such as Devonport et al.[25] and Lew et al.,[26] employ different methods to estimate probabilistic geofences given bounded uncertainties. However, these methods may not be applicable for arbitrary probability distributions, including unbounded ones. To address this, the paper proposes a data-driven approach using Kernel Density Estimation (KDE) to approximate the unknown probability density function (PDF) of any arbitrary probability distribution. The level sets of the resulting PDF can be taken as probabilistic geofences. This method can be updated online through sensor measurements and can be accelerated using Fast Fourier Transform (FFT) to make it computationally efficient for real-time implementation.

The aim of this paper is to approximate probabilistic geofences using Kernel Density Estimation (KDE). However, there are some limitations to using the level sets of KDE as probabilistic geofences. Firstly, there is no closed-form expression for KDE and its level sets. Additionally, the shape of the level set is often irregular or non-convex. Therefore, this paper proposes a new approach to solving this problem. The authors provide a convex approximation of the probabilistic geofence by formulating an Integer Linear Programming (ILP) optimization problem using KDE values. A heuristic algorithm is implemented to solve the optimization problem efficiently. This approach is more accurate than using the convex hull or bounding box method, which can be overly conservative. The resulting convex parallelogram box is less conservative and more accurate, providing a larger feasible planning space. Additionally, the number of sides of the convex parallelogram box is constant, making it more practical for real-world applications.

1.2. *Contributions and Organization*

The major contributions of this paper are highlighted as follows:

1) To model data-driven distributions through online data samplings, FFT is encoded into traditional KDE to accelerate the computation greatly. Although most kernel estimators can use KDE to approximate the probability density function (PDF) of the data-driven distribution, the direct approach to evaluate KDE can be computationally expensive, especially when the number of data samplings becomes very large.[27] Instead, we present an alternative way to replace performing kernel evaluation on data points with that on grids. Further, FFT can be implemented to reduce computational complexity owing to the structure of discrete convolution of kernel evaluation. According to the results of the KDE evaluation, we develop an algorithm to evaluate the probabilistic geofence of data-driven uncertainties in real-time.

2) We build an optimization framework of ILP to seek a minimal parallelogram box approximation of the probabilistic geofence of data-driven uncertainties. This ILP formulation can help extend the traditional rectangle box to a general parallelogram box, which can reduce the conservatism of geofencing.

3) To solve the optimization framework online, we develop a heuristic algorithm that can find a near-optimal solution. This efficient algorithm can help translate the data-driven uncertainty into deterministic constraints for the path planning of eVTOL aircraft in AAM applications in an online fashion.

The content of the paper is organized as follows: in Section II, we will formally set up the problem and define the probabilistic geofences; in Section III, we will develop an online computational algorithm to capture the probabilistic geofences; in Section IV, we will build an optimization framework as an ILP problem to find a box approximation of the probabilistic geofence; In Section V, we develop a heuristic algorithm to solve for it; in Section VI, we conduct a simulation study to demonstrate the feasibility and efficiency of our proposed algorithms; last but not least, we will conclude this paper in Section VII.

2. Problem Formulation

For an eVTOL, its motion can be modeled by a general discrete-time dynamic system of the form

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

where state $\boldsymbol{x}_k \in \mathbb{R}^n$, control input $\boldsymbol{u}_k \in \mathcal{U}_k$, uncertain model parameter $\boldsymbol{\theta}_k \in \Theta$, uncertain disturbance $\boldsymbol{w}_k \in \mathbb{W}$, and initial state $\boldsymbol{x}_0 \in \mathcal{X}_0$. The sets $\mathcal{U}_k \subset \mathbb{R}^m, \Theta \subset \mathbb{R}^p, \mathbb{W} \subset \mathbb{R}^q$, and $\mathcal{X}_0 \subset \mathbb{R}^n$ are not limited to bounded sets but can be unbounded ones. Also, the dynamics $\boldsymbol{f}(\cdot)$ can be a general function that needn't be continuously differentiable. Given the uncertainties of $\boldsymbol{x}_k, \boldsymbol{\theta}_k, \boldsymbol{w}_k$, then $\boldsymbol{x}_{k+1}$ is a random vector which obeys an arbitrary unknown probability distribution.

To operate safely in an uncertain, dynamic environment, each eVTOL should try to avoid conflicts with other eVTOLs or environmental obstacles in the system. Hopefully, the probability of collision between aircraft and obstacles (or other aircraft) in the system at any time is less than a certain threshold, namely,

$$\Pr(\text{collision}) \leq 1 - \alpha \tag{2}$$

where α is a prescribed confidence level.

Therefore, we expect to find a bounded set such that the probability of the state $\boldsymbol{x}_k$ lying in the bounded set is greater than the confidence level. In this paper, the dynamic system Eq. (1) is assumed to be two-dimensional and the state $\boldsymbol{x}_k$ is the location of eVTOL at a particular time k. Note that this formulation can be easily extended to higher dimensions in future work. To this end, we can define a probabilistic geofence $\tilde{\mathcal{X}}_k$ as follows:

Definition 2.1 (Probabilistic Geofence). At time k, a bounded set is defined to be a probabilistic geofence $\tilde{\mathcal{X}}_k$ for the location state $\boldsymbol{x}_k$ of the dynamic system Eq. (1) at confidence level α if and only if

$$\Pr(\boldsymbol{x}_k \in \tilde{\mathcal{X}}_k) \geq \alpha \tag{3}$$

The goal of this paper is twofold:

1) We develop a data-driven approach to model the arbitrary unknown uncertainties and find a probabilistic geofence $\tilde{\mathcal{X}}_k$ for the state $\boldsymbol{x}_k$ of dynamic system Eq. (1);

2) Motivated by the requirement of realizing safety-critical real-time motion planning for uncertain systems, we aim to efficiently find an optimal parallelogram box to approximate the probabilistic geofence $\tilde{\mathcal{X}}_k$ without loss of accuracy.

3. Online Evaluation of Probabilistic Geofence

In this section, we are going to develop an online algorithm to capture the probabilistic geofence defined in the last section.

3.1. *Traditional Kernel Density Estimator*

A traditional method of KDE can be formulated by placing a kernel function $K(\cdot)$ on every data point x_i

$$\hat{f}(x) = \frac{1}{N_{ds}} \sum_{i=1}^{N_{ds}} K(x - x_i) \tag{4}$$

where N_{ds} represents the number of data samples.

Taking the bandwidth h of $\hat{f}(x)$ into consideration and assigning different weights w_i to different data points x_i, Eq. (4) can be rewritten as

$$\hat{f}(x) = \frac{1}{h} \sum_{i=1}^{N_{ds}} w_i K\left(\frac{x - x_i}{h}\right) \tag{5}$$

where $\sum_{i=1}^{N_{ds}} w_i = 1$.

The discretized form of traditional KDE can be obtained by evaluating the values of KDE over a mesh composed of N grid points $g_1, \ldots, g_N$ in each dimension

$$\hat{f}_j = \frac{1}{h} \sum_{i=1}^{N_{ds}} w_i K\left(\frac{g_j - x_i}{h}\right), \; j = 1, \ldots, N \tag{6}$$

An extension of Eq. (5) to d-dimensional scenarios is to write

$$\hat{f}(x) = h^{-1} \sum_{i=1}^{N_{ds}} w_i K\left(h^{-1}(x - x_i)\right) \tag{7}$$

where $\sum_{i=1}^{N_{ds}} w_i = 1$, and $K(\cdot)$ is a d-variate kernel function. Also, its discretized form can be derived as

$$\hat{f}_j = h^{-1} \sum_{i=1}^{N_{ds}} w_i K\left(h^{-1}(g_j - x_i)\right), \; j = 1, \ldots, N \tag{8}$$

For simplicity, Eq. (7) and Eq. (8) can be denoted as the following form

$$\hat{f}(x) = \sum_{i=1}^{N_{ds}} w_i K_h (x - x_i) \tag{9}$$

$$\hat{f}_j = \sum_{i=1}^{N_{ds}} w_i K_h (g_j - x_i), \; j = 1, \dots, N \tag{10}$$

3.2. *Online Identification of Probabilistic Geofence*

The computational intensity of traditional KDE is very high. Instead of direct evaluation, we can go through N_{ds} data points and assign weights to N equidistant grid points.[28] To speed up the evaluation of traditional KDE, the kernel function $K(\cdot)$ performed on N_{ds} data points x_i can be replaced by that on N grid points g_l.

For bivariate kernel estimators, as shown in Fig. 1, the mass associated with the data point X is distributed among each of the four surrounding grid points according to areas of the opposite sub-parallelograms induced by the position of the data point. By doing so, we can obtain the approximation of Eq. (8)

$$\hat{f}_j = \sum_{l=1}^{N} K_h (g_j - g_l) c_l, \; j = 1, \dots, N \tag{11}$$

where $c_l, l = 1, \dots, N$ are the grid weights assigned to every grid point g_l, which is determined by the number of data points x_i in the neighborhood of g_l. In this way, the number of kernel evaluations is only $O(N)$, which greatly saves running time, especially for large samples of data points.

Further, let $L = N - 1$ and then Eq. (11) can be reformulated as

$$\hat{f}_j = \sum_{l=-L}^{L} c_{j-l} k_l, \; j = 1, \dots, N \tag{12}$$

where

$$k_l = K_h (g_j - g_l) \tag{13}$$

Note that $c_l = 0, l \notin \{1, \dots, N\}$. By the symmetry of the kernel function $K_h(\cdot)$, it's only required to figure out k_l for $l = 0, 1, \dots, L$ where $L = N - 1$. Therefore, it is clear that no more than N kernel evaluations are required to obtain k_l. This is because there are only N distinct differences among different grid points. Indeed, Eq. (12) can be viewed as the discrete convolution of c_l and k_l. This means the approximation we use has a discrete convolution structure that can be computed quickly using FFT. Let C and K be the discrete Fourier transform of c_l and k_l respectively using FFT, and let F be the element-wise product of C and K. Then the values of KDE $\hat{f}_j$ can be extracted from the inverse FFT of F. By doing so, we can obtain KDE which approximates the PDF function of the data-driven distribution in real time.

There are two roles that grid points play in this process: the KDE function is evaluated on grid points; grid weights c_l are assigned to every grid point.

After obtaining KDE values, we can identify probabilistic geofence at any confidence level α for the data-driven distribution. Without violating the identified geofence, the probability of potential conflicts between eVTOL aircraft is within the bound $1 - \alpha$.

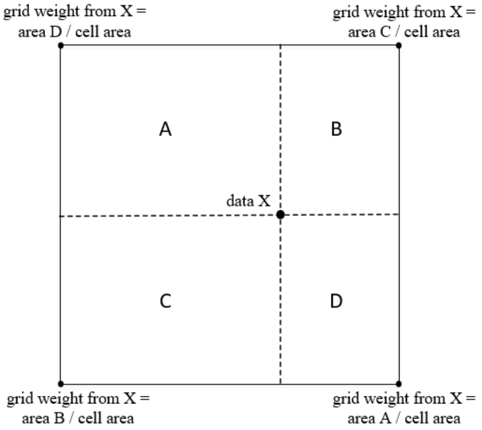

grid weight from X = area D / cell area
grid weight from X = area C / cell area
A
B
data X
C
D
grid weight from X = area B / cell area
grid weight from X = area A / cell area

Fig. 1: Assign weights to grid points[27]

Now we can present an algorithm to evaluate the probabilistic geofence of data-driven uncertainties online using FFT-based KDE in Algorithm 1. In the pseudo-code, N_{ds} is the number of sampled data points, N is the number of grid points in each dimension, α represents the confidence level of the data-driven distribution, and ϵ represents the error of the threshold. In Line 10, FFT is employed to speed up the computation of discrete convolution to obtain KDE values. In Line 40, the probabilistic geofence at the confidence level α of the data-driven distribution can be returned if successfully identified.

In summary, we develop an alternative way to replace performing kernel evaluation on data points with that on grids. Further, FFT can be implemented to reduce computational complexity owing to the structure of discrete convolution of kernel evaluation. Based on the results of the KDE evaluation, we propose an algorithm of online data-driven evaluation to capture probabilistic geofence. The identification of probabilistic geofence can help translate the data-driven uncertainty into deterministic constraints for the path planning of eVTOL aircraft in AAM applications.

Algorithm 1 Online Evaluation of Probabilistic Geofence

```
function GenDS($N_{ds}$)
    generate $N_{ds}$ data samplings $ds$
    return $ds$

function MeshGrid($ds$, $N$)
    get $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ from data samplings $ds$
    $x = \text{linspace}(x_{\min}, x_{\max}, N)$
    $y = \text{linspace}(y_{\min}, y_{\max}, N)$
    $g$ = Cartesian product of $x$ and $y$
    return $g$

function FFTKDE($ds$, $g$)
    obtain weights $c_l$ for all grids $g$ according to $ds$
    evaluate kernel functions $k_l$ on all grids $g$
    $C = \text{FFT}(c_l)$
    $K = \text{FFT}(k_l)$
    $F$ = element-wise product of $C$ and $K$
    $z^{kde} = \text{iFFT}(F)$
    return $g$, $z^{kde}$

function BisecSearch($z^{kde}$, $\alpha$, $\epsilon$)
    $low = \min(z^{kde})$
    $up = \max(z^{kde})$
    while $low < up$ do
        $mid = (low + up)/2$
        $z^{bin} = (z^{kde} \geq mid) \cdot 1$
        $z^{mix}$ = element-wise product of $z^{bin}$ and $z^{kde}$
        $pr = \text{sum}(z^{mix}) / \text{sum}(z^{kde})$
        if $\text{abs}(pr - \alpha) \leq \epsilon$ then
            $C^{kde} = mid$
            return $z^{bin}$, $C^{kde}$
        else if $pr < \alpha$ then
            $up = mid$
        else
            $low = mid$
    return $Failure$

function FindGeofence($N_{ds}$, $N$, $\alpha$, $\epsilon$)
    $ds$ = GenDS($N_{ds}$)
    $g$ = MeshGrid($ds$, $N$)
    $g$, $z^{kde}$ = FFTKDE($ds$, $g$)
    $z^{bin}$, $C^{kde}$ = BisecSearch($z^{kde}$, $\alpha$, $\epsilon$)
    return $g$, $z^{kde}$, $z^{bin}$, $C^{kde}$

FindGeofence($N_{ds}$, $N$, $\alpha$, $\epsilon$)
```

4. Formulation of ILP Optimization Framework

In this section, we are going to formulate an Integer Linear Programming (ILP) problem whose solution is a parallelogram box to approximate the probabilistic geofence of two-dimensional data-driven uncertainties given a confidence level α. The algorithms can be extended to any dimensional uncertainties.

Let (x_s, y_s) be the 2D coordinate of data sample s and N_{ds} is the number of data samples. Also, we denote (x_i, y_j), $\forall i, j \in \{1, \ldots, N\}$ as the 2D coordinates of the grid points established in Algorithm 1. The grid points can be viewed as a bijective map $g : (i, j) \to (x_i, y_j)$. Through the online evaluation of the geofence of a two-dimensional data-driven uncertainty using Algorithm 1, we can assign KDE values $z_{ij}^{kde} = \hat{f}(x_i, y_j)$ to $N \times N$ grid points $(i, j), i, j \in \{1, \ldots, N\}$.

Let (x_{p1}, y_{p1}) and (x_{p2}, y_{p2}) be the unit vector of two principal axes of the data samples, then the grid points can be established aligning with these two principal axes by performing a linear transformation, which is defined as follows.

The fixed point of the linear transformation is $\bar{\boldsymbol{x}} := (\bar{x}, \bar{y}) = \frac{1}{N_{ds}} \sum_{s=1}^{N_{ds}} (x_s, y_s)$. The transformation of grid points is then represented as $(x'_i, y'_j) = \boldsymbol{A}((x_i, y_j) - \bar{\boldsymbol{x}}) + \bar{\boldsymbol{x}}$, where (x'_i, y'_j) is the coordinate of a grid point after transformation, and $\boldsymbol{A}$ is determined by the principal axes as

$$\boldsymbol{A} = \begin{bmatrix} x_{p1} & x_{p2} \\ y_{p1} & y_{p2} \end{bmatrix} \tag{14}$$

The KDE values $\hat{f}(x'_i, y'_j)$ of every grid point (x'_i, y'_j) after transformation can be evaluated by

$$z_{ij}^{\prime kde} = \hat{f}(x'_i, y'_j) = \hat{f}(x_i, y_j) |\det(\boldsymbol{A})| \tag{15}$$

The grid points after transformation can also be viewed as a new bijective map $h : (i, j) \to (x'_i, y'_j)$, $\forall i, j \in \{1, \ldots, N\}$. The KDE values $z_{ij}^{\prime kde}$ after transformation can be normalized as w_{ij}

$$w_{ij} := \frac{z_{ij}^{\prime kde}}{\sum_{i=1}^{N} \sum_{j=1}^{N} z_{ij}^{\prime kde}} \quad i, j \in \{1, \ldots, N\} \tag{16}$$

to every grid point after transformation, such that

$$0 \leq w_{ij} \leq 1 \tag{17}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} = 1. \tag{18}$$

The ILP optimization framework will be established using these weights, and not with the original data samples. This makes it more tractable because the number of grid points is far less than the number of data samples, greatly reducing the number of decision variables and constraints.

The goal of the optimization problem is to find a minimum parallelogram box to approximate the geofence, by determining which weighted grid points should be covered by the box. To this end, we introduce a binary variable $z_{ij}, i, j \in \{1, \ldots, N\}$. If a grid point (i, j) is covered by the box, $z_{ij} = 1$; otherwise, $z_{ij} = 0$.

To get the box with minimal area, we can set the objective of the optimization as $\min \sum_{i}^{N} \sum_{j}^{N} z_{ij}$. The grid point which owns the greatest KDE value should be covered by the box, which means

$$z_{\arg\max_{i,j} \omega_{ij}} = 1. \tag{19}$$

To approximate the geofence given a confidence level α, we need to make sure the sum of the normalized KDE values of the grid points covered by the box is greater than the confidence level, in other words,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij} z_{ij} \geq \alpha \tag{20}$$

In addition, to find a parallelogram box for geofence approximation, we want to enforce that

$$(z_{ij} \wedge z_{i'j'}) \implies z_{uv} \tag{21}$$

where grid points $(i, j), (i', j') \in \{1, \ldots, N\} \times \{1, \ldots, N\}$ and $\min(i, i') \leq u \leq \max(i, i') \wedge \min(j, j') \leq v \leq \max(j, j')$, which means that if any two grid points (i, j) and (i', j') are covered by the box, then the grid point (u, v) within the parallelogram determined by (i, j) and (i', j') must also be covered.

Rewriting Eq. (21) in conjunctive normal form yields

$$\neg z_{ij} \vee \neg z_{i'j'} \vee z_{uv} \tag{22}$$

from which we obtain linear constraints

$$(1 - z_{ij}) + (1 - z_{i'j'}) + z_{uv} \geq 1 \tag{23}$$

or equivalently,

$$z_{ij} + z_{i'j'} - z_{uv} \leq 1 \tag{24}$$

for all (i, j), (i', j'), and (u, v) such that (u, v) is in the parallelogram determined by (i, j) and (i', j'). That is,

$$\min(i, i') \leq u \leq \max(i, i') \tag{25}$$
$$\min(j, j') \leq v \leq \max(j, j') \tag{26}$$

For brevity, we can denote index sets as follows:

$$\mathbb{N} = \{1, \ldots, N\} \tag{27}$$
$$\mathbb{I} = \{(i', j') : i < i' \vee (i = i' \wedge j < j')\} \tag{28}$$
$$\mathbb{J} = \{(u, v) : i \leq u \leq i', \min(j, j') \leq v \leq \max(j, j')\} \tag{29}$$

Above all, we can formally formulate the optimization framework as an ILP problem as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \sum_{j=1}^{N} z_{ij} \\ \text{s.t.} \quad & z_{ij} + z_{i'j'} - z_{uv} \leq 1 \quad \forall i, j \in \mathbb{N}, \forall (i', j') \in \mathbb{I}, \forall (u, v) \in \mathbb{J}; \\ & z_{\arg\max_{i,j} \omega_{ij}} = 1; \\ & \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} z_{ij} \geq \alpha; \\ & z_{ij} \in \{0, 1\} \quad \forall i, j \in \mathbb{N}. \end{aligned} \tag{30}$$

5. Solution Method

In this section, we develop a heuristic algorithm to efficiently solve the optimization problem of ILP formulated in the last section.

5.1. *ILP Optimal Algorithm*

Gurobi can be used to solve the formulated optimization framework of ILP in Eq. (30). As shown in Fig. 2, the Branch and Cut Algorithm built-in Gurobi explores the whole search space and given enough time, it will finally find a globally optimal solution. This paper applies the standard Branch and Cut Algorithm, so the discussion for the algorithm is omitted, please refer[29] for details.

Resorting to Branch and Cut Algorithm built-in Gurobi, we develop Algorithm 2 to solve the formulated optimization framework Eq. (30). Through implementing Algorithm 2, we can find an optimal binary solution of $z_{ij}, i, j \in \{1, \ldots, N\}$ to the ILP problem Eq. (30), which accounts for an optimal parallelogram box to approximate the probabilistic geofence computed by Algorithm 1.

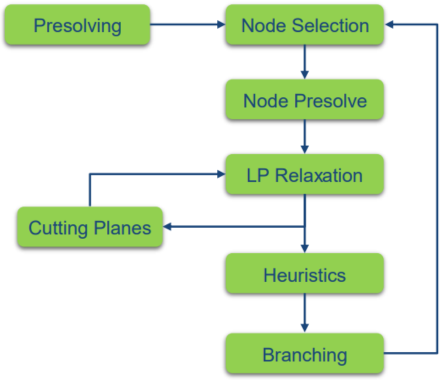


Fig. 2: Branch and Cut Algorithm built-in Gurobi

Algorithm 2 ILP Optimal Algorithm

```
function ILPSolver($\alpha$, $g$, $w$)
    $\bar{i}$, $\bar{j}$ = arg max($w$)
    determine $x_{\bar{i}}$, $y_{\bar{j}}$ according to $\bar{i}$, $\bar{j}$ and grid points $g$
    formulate Eq. (30) for grid points $g$ with weights $w$
    given confidence level $\alpha$
    implement Branch and Cut algorithm
    to solve Eq. (30)
    return $z_{ij}$

function FindBox($g$, $w$, $\alpha$)
    $z_{ij}$ = ILPSolver($\alpha$, $g$, $w$)
    return $z_{ij}$

FindBox($g$, $w$, $\alpha$)
```

Algorithm 3 ILP Heuristic Algorithm

```
function BoxApprox($r_{\min}, c_{\min}, r_{\max}, c_{\max}, \omega_{ij}$)
    $\omega_{ij}^{'} = \omega_{ij}$.copy()
    $z_{ij}[1:N][1:N] = 0$
    $r_{\min}, c_{\min} = \arg\max_{i,j} \omega_{ij}^{'}$
    $r_{\max} = r_{\min}$, $c_{\max} = c_{\min}$
    while $\sum_{i}^{N}\sum_{j}^{N} \omega_{ij} z_{ij} < \alpha$ do
        $i, j = \arg\max_{i,j} \omega_{ij}^{'}$
        $\omega_{ij}^{'}[i][j] = 0.0$
        $r_{\min} = \min(r_{\min}, i)$, $r_{\max} = \max(r_{\max}, i)$
        $c_{\min} = \min(c_{\min}, j)$, $c_{\max} = \max(c_{\max}, j)$
        $z_{ij}[r_{\min} : r_{\max}][c_{\min} : c_{\max}] = 1$
    return $z_{ij}$

BoxApprox($r_{\min}$,$c_{\min}$,$r_{\max}$,$c_{\max}$)
```

5.2. *ILP Heuristic Algorithm*

However, the formulated optimization framework Eq. (30) is an NP-hard problem, which makes the branch and cut algorithm computationally expensive. To address the issue, we propose a heuristic algorithm instead to solve the formulated optimization framework.

The main procedure of the proposed heuristic algorithm is summarized as follows: 1) First, the grid point (i, j) which has the greatest KDE value is selected to be covered by the box, namely, $z_{ij} = 1$; 2) Then, excluding the aforementioned grid point (i, j), the grid point (i', j') which owns the greatest KDE value in all the remaining grid points is selected to be covered, in other words, $z_{i'j'} = 1$; 3) Next, make all the grid points within the box determined by $(\min(i, i'), \min(j, j'))$ and $(\max(i, i'), \max(j, j'))$ be selected to be covered by the box; 4) Repeating the above steps while the sum of the normalized KDE values of the selected grid points not exceeding the confidence level α; 5) Last, if the sum exceeds the confidence level, the selected grid points form the parallelogram box that we want, which provides a near-optimal solution to the optimization framework Eq. (30).

The details of the proposed heuristic algorithm to solve the optimization framework Eq. (30) is given in Algorithm 3.

6. Main Results

In this section, comprehensive case studies are conducted to show the performance of four algorithms: ILP Optimal, ILP Heuristic, Gaussian Fit, and Bounding Box. They all provide 2D boxes to approximate the probabilistic geofence respectively. 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 64GB RAM.

The procedure of case studies is decomposed into two phases:

1) According to KDE values and the probabilistic geofence obtained from a collection of N_{ds} data samples through running Algorithm 1, we can formulate an ILP problem. Then, ILP Optimal and ILP Heuristics are implemented respectively to find box approximations of the probabilistic geofence. As a comparison, the Bounding Box algorithm and Gaussian Fit algorithm are also applied to approximate the probabilistic geofence;

2) For the boxes obtained, we can generate another new collection of $\gg N_{ds}$ data samples, and evaluate the ratio of the number of data samples inside the box to the total number of data samples generated. As the size of the collection increases, the ratio will converge to the true probability of the system state lying in the box.

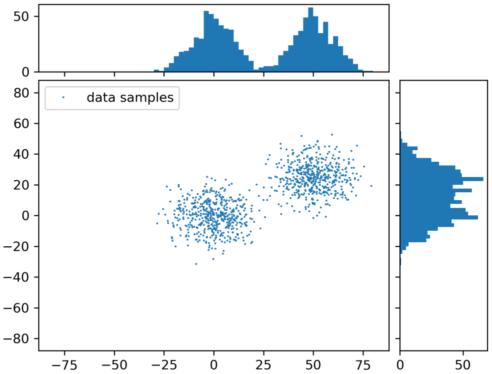


Fig. 3: Joint bimodal distribution generated by two marginal histograms

In this section, we display a scatter plot of the possible positions (data samples) $(x, y)^\top$ of a vehicle on the plane at a time point, which is generated by the measurement of marginal histograms for x and y respectively. Every iteration we measure the position of the vehicle once, and then different iterations yield different positions serving as different data samples. As shown in Fig. 3, the positions of the vehicle at that time point obey a bimodal distribution which is non-Gaussian.

The number of data samples is $N_{ds} = 1000$. The setting below is chosen as a baseline: the confidence level is $\alpha = 90\%$, and the number of grid points is $N^2 = 20^2$. In the following figures and tables, the baseline is marked by *. The time given in Table 1 and Table 2 is the computational time of probabilistic geofence estimation together with parallelogram box identification.

6.1. *Different Numbers of Grid Points*

In this part, we compare the performance of different algorithms given different numbers of grid points: $N \times N = 10^2$, 20^2, 25^2. The other parameters are the same as the baseline.

The results are given in Fig. 4 and Table 1. In Fig. 4, the yellow contour is the probabilistic geofence obtained by Algorithm 1; The yellow box that bounds the probabilistic geofence is the result of the Bounding Box algorithm; The blue box is the result of ILP Optimal, which is the optimal solution to the ILP problem Eq. (30) using Algorithm 2; The red box is the result of ILP Heuristic, which is a near-optimal solution to the ILP problem Eq. (30) using Algorithm 3; The green ellipse is the result of Gaussian Fit.

For ILP Optimal, as the number of grid points increases, the performance of ratio and area improves but computational time also significantly increases. This is because as the number of grid points increases, the number of decision variables and constraints increases, which contributes to the computational complexity. As shown in Table 1, the computational time of ILP Optimal increases from 1.04 s to 411 s. To achieve a balance, the number of grid points can be taken as 20^2. When the number of grid points is large enough, the gap between two boxes obtained by ILP Optimal and ILP Heuristic is small enough to be acceptable.

When the number of grid points is fixed to be 20^2, the area and ratio of ILP Optimal and ILP Heuristic are smaller than Gaussian Fit or Bounding Box, which suggests ILP Optimal and ILP Heuristic are less conservative and more accurate than Gaussian Fit or Bounding Box in the sense that they have smaller areas while ensuring small gaps between their ratios and the confidence level α. Further, for ILP Optimal and ILP Heuristic, their results of ratio and area are almost the same but the computational time of ILP Heuristic 0.02 s greatly outperforms ILP Optimal 103 s, demonstrating the efficiency of ILP Heuristic.

6.2. *Different Confidence Levels*

In this part, we compare the performance of implementing different algorithms at three different confidence levels: $\alpha = 90\%$, 95%, and 99%. The other parameters are the same as the baseline.

As illustrated in Fig. 5, compared with the probabilistic geofence based on irregular contour, every polygon obtained by ILP Optimal, ILP Heuristic, or Bounding Box respectively, is a conservative but convex approximation.

In Table 2, the area of the box obtained by ILP Heuristic is $4173.3\,\mathrm{m}^2$, which is less than $4399.9\,\mathrm{m}^2$ of Bounding Box or $5379.8\,\mathrm{m}^2$ of Gaussian Fit. This means the box obtained by ILP Optimal is far less conservative than Bounding Box or Gaussian Fit. Also, That is, ILP Heuristic is much more efficient. Meanwhile, the area of the box obtained by the ILP Heuristic is almost the same as ILP Optimal. However, the computational time 0.02 s of implementing the ILP Heuristic is much shorter than 103 s of ILP Optimal. Hence, ILP Heuristic runs much faster than ILP Optimal while ensuring accuracy.

6.3. *Near-optimality of ILP Heuristic Algorithm*

From Table 1, we have observed that ILP Heuristic greatly outperforms ILP Optimal in computational efficiency at the expense of sacrificing space efficiency. Naturally, we are concerned with the gap of space efficiency between ILP Optimal and ILP Heuristic.

The gap between the boxes of ILP Optimal and ILP Heuristic can be quantified using Jaccard distance. It compares members of two sets to see which members are shared and which are distinct. It's a measure of similarity between the two sets of data, with a range from 0 to 1. Formally, for any two sets A and B, Jaccard distance is defined as

$$d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}, \tag{31}$$

where $| \cdot |$ is the cardinality of a set. A smaller Jaccard distance means the two sets are more similar to each other.

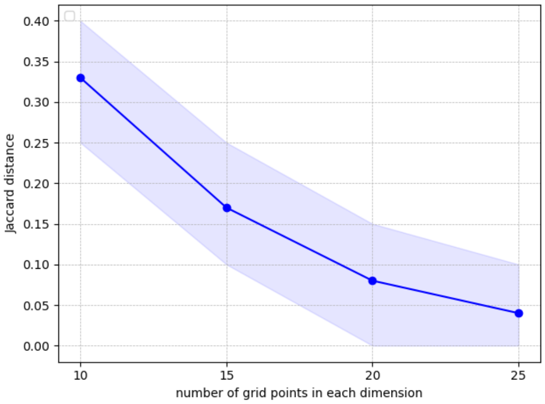
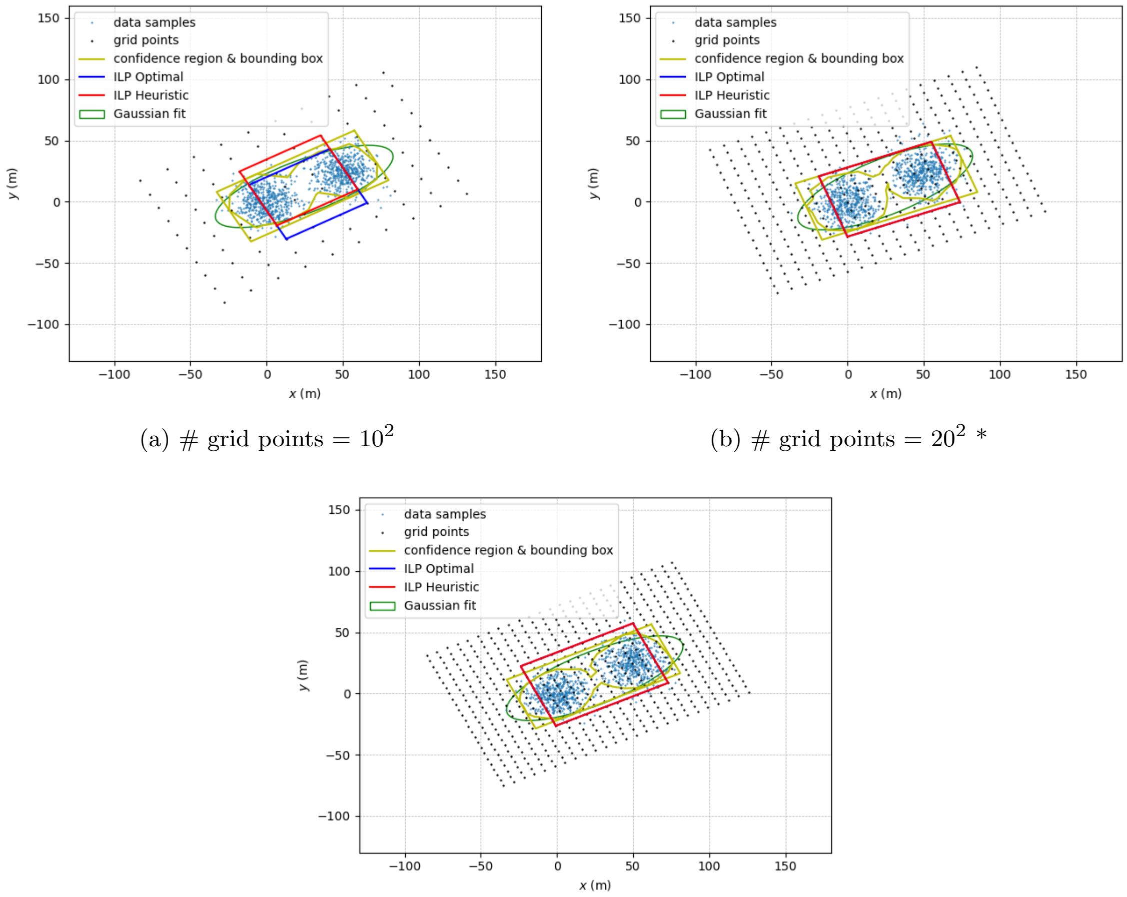


Fig. 6: Near-optimality analysis of ILP Heuristic algorithm

We analyze the Jaccard distance between the boxes obtained by ILP Heuristic and ILP Optimal with respect to the varying number of grid points. All the other parameters are the same as the baseline. Given a fixed number of grid points, the Jaccard distance may differ across different tests, because the randomness of data samples in every test leads to different boxes obtained by the algorithms.

Given a fixed number of grid points, we run ILP Optimal once to obtain an optimal box. Then we run ILP Heuristic ten times to obtain ten near-optimal boxes which



(a) # grid points = 10^2

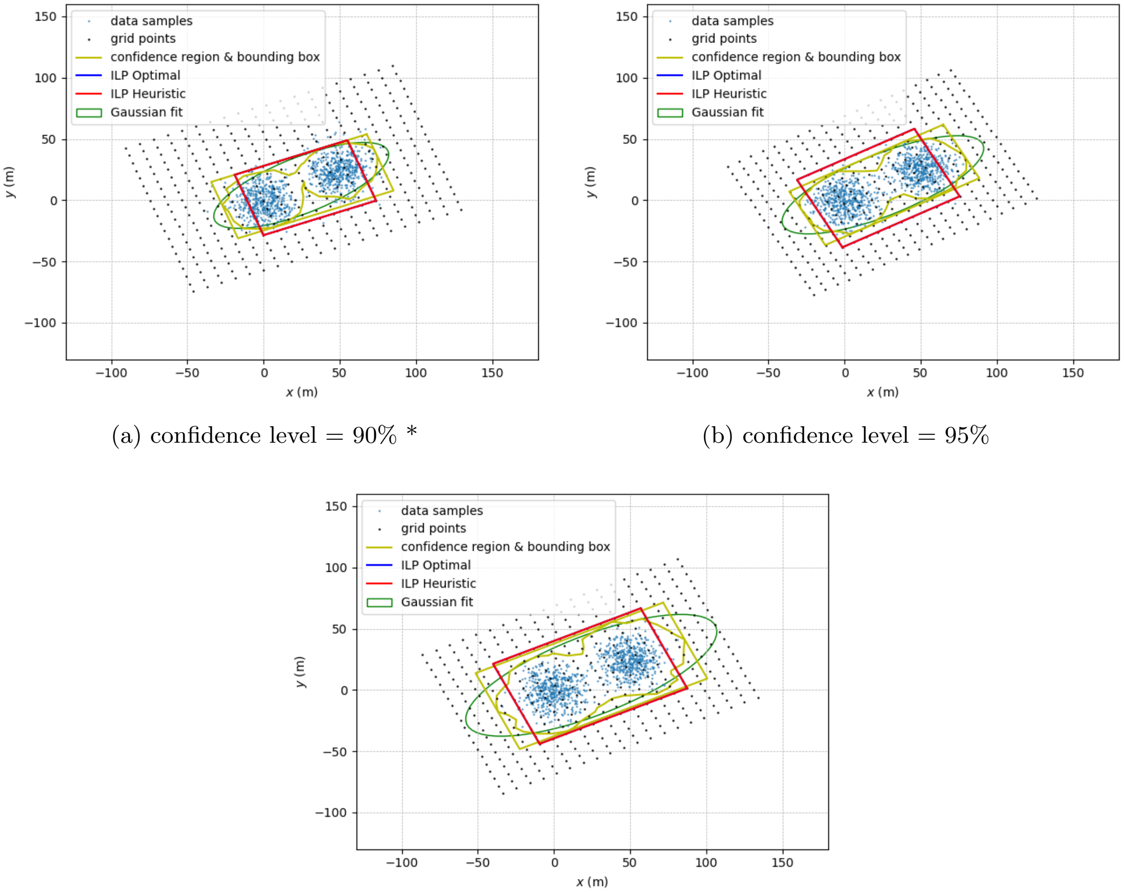
(b) # grid points = 20^2 *

(c) # grid points = 25^2

Fig. 4: Comparisons of different numbers of grid points

Table 1: Comparisons of different numbers of grid points

# grid points	Algorithm	Ratio	Area (m^2)	Time (s)
10^2	ILP Optimal	85.5%	3061.9	1.04
	ILP Heuristic	85.9%	3067.6	0.01
	Gaussian Fit	90.9%	4211.0	0.01
	Bounding Box	93.7%	4789.2	0.22
20^2 *	ILP Optimal	90.5%	4173.3	103
	ILP Heuristic	90.5%	4173.3	0.02
	Gaussian Fit	91.2%	4399.9	0.01
	Bounding Box	95.3%	5379.8	0.19
25^2	ILP Optimal	90.1%	4096.2	411
	ILP Heuristic	90.1%	4096.2	0.04
	Gaussian Fit	90.8%	4205.7	0.01
	Bounding Box	93.5%	4642.1	0.21



(a) confidence level = 90% *

(b) confidence level = 95%

(c) confidence level = 99%

Fig. 5: Comparisons of different confidence levels

Table 2: Comparisons of different confidence levels

Confidence level	**Algorithm**	**Ratio**	**Area** (m^2)	**Time** (s)
90% *	ILP Optimal	90.5%	4173.3	103
	ILP Heuristic	90.5%	4173.3	0.02
	Gaussian Fit	91.2%	4399.9	0.01
	Bounding Box	95.3%	5379.8	0.19
95%	ILP Optimal	95.4%	5485.0	91
	ILP Heuristic	95.4%	5485.0	0.04
	Gaussian Fit	95.7%	5597.8	0.01
	Bounding Box	96.1%	5717.9	0.22
99%	ILP Optimal	99.1%	7714.4	64
	ILP Heuristic	99.1%	7714.4	0.03
	Gaussian Fit	99.8%	8960.8	0.01
	Bounding Box	99.9%	9307.3	0.26

may differ every time. We evaluate the Jaccard distance between every near-optimal box and the optimal box and then obtain ten distances eventually. We repeat the operation above for 10^2, 15^2, 20^2, 25^2 grid points respectively.

As the number of grid points increases from 10^2 to 25^2, the average Jaccard distance decreases from 0.34 to 0.04. This means the box obtained by ILP Heuristic becomes more near-optimal with respect to the increasing number of grid points. As suggested in Table 1, as the number of grid points increases, the computational time increases from 0.01 s to 0.04 s. Therefore, the increase of grid points contributes to the burden of computational effort but reduces the difference between the boxes obtained by ILP Heuristic and ILP Optimal.

Accordingly, the computational efficiency of ILP Heuristic may come at the cost of loss of optimality. Therefore, to achieve a balance, we would like to select an appropriate parameter for the number of grids used by the ILP Heuristic. For example, it can be 20^2 which makes the box of ILP Heuristic pretty enough for its promising applications in real scenarios while enjoying accuracy, efficiency, and near-optimality.

7. Conclusions

How to ensure operation safety in high-density, dynamic, and uncertain airspace environments in real-time is a key challenge for the success of AAM. This paper introduces a concept called probabilistic geofences which can guarantee that the probability of potential conflicts is bounded in the face of data-driven uncertainties. To efficiently evaluate the probabilistic geofences, Kernel Density Estimation (KDE) combined with Fast Fourier Transform (FFT) is adopted to model data-driven uncertainties. After obtaining KDE values, we develop a heuristic algorithm based on the optimization framework of Integer Linear Programming (ILP) to find a parallelogram box to approximate the probabilistic geofence. Numerical simulation demonstrates that the heuristic algorithm can efficiently identify a box approximation of the probabilistic geofence with an acceptable gap between its near-optimal solution and the optimal solution found by Gurobi Optimizer.

The box approximation found in this paper is conservative in the sense that it must be a parallelogram rather than arbitrary convex polygons. In the future, we are going to develop an efficient optimization algorithm to provide a convex approximation for the probabilistic geofence given data-driven uncertainties, not limited to parallelogram boxes. Also, this work is limited to a 2D system. Next, we will extend our work to higher dimensions, and then apply it to the applications of safety-critical real-time motion planning for uncertain systems like robotics.

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.

The authors would like to thank Muhan Zhao from UC San Diego for the valuable comments and discussions.

References

[1] H. Liu, J. Chen, J. Feng and H. Zhao, An improved rrt* uav formation path planning algorithm based on goal bias and node rejection strategy, *Unmanned Systems* **11**(04) (2023) 317–326.

[2] S. Julius Fusic and R. Sitharthan, Improved rrt* algorithm-based path planning for unmanned aerial vehicle in a 3d metropolitan environment, *Unmanned Systems* (2023) 1–17.

[3] G. Skorobogatov, C. Barrado and E. Salamí, Multiple uav systems: A survey, *Unmanned Systems* **8**(02) (2020) 149–169.

[4] M. Radmanesh, M. Kumar, P. H. Guentert and M. Sarim, Overview of path-planning and obstacle avoidance algorithms for uavs: A comparative study, *Unmanned systems* **6**(02) (2018) 95–118.

[5] P. Wu, X. Yang, P. Wei and J. Chen, Safety assured online guidance with airborne separation for urban air mobility operations in uncertain environments, *IEEE Transactions on Intelligent Transportation Systems* (2022).

[6] X. Yang and P. Wei, Scalable multi-agent computational guidance with separation assurance for autonomous urban air mobility, *Journal of Guidance, Control, and Dynamics* **43**(8) (2020) 1473–1486.

[7] Z. Zhou, J. Chen and Y. Liu, Optimized landing of drones in the context of congested air traffic and limited vertiports, *IEEE Transactions on Intelligent Transportation Systems* **22**(9) (2021) 6007–6017.

[8] S. Kahne and I. Frolow, Air traffic management: Evolution with technology, *IEEE Control Systems* **16**(4) (1996) 12–21.

[9] M. Stevens and E. Atkins, Geofence definition and deconfliction for uas traffic management, *IEEE Transactions on Intelligent Transportation Systems* (2020).

[10] J. Kim and E. Atkins, Airspace geofencing and flight planning for low-altitude, urban, small unmanned aircraft systems, *Applied Sciences* **12**(2) (2022) p. 576.

[11] P. Wu, L. Li, J. Xie and J. Chen, Probabilistically guaranteed path planning for safe urban air mobility using chance constrained rrt*, *AIAA AVIATION 2020 FORUM*, (2020), p. 2914.

[12] L. Cheng, H. Wen and D. Jin, Uncertain parameters analysis of powered-descent guidance based on chebyshev interval method, *Acta Astronautica* **162** (2019) 581–588.

[13] J. P. Hespanha, H. H. Kizilocak and Y. S. Ateskan, Probabilistic map building for aircraft-tracking radars, *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, **6**, IEEE (2001), pp. 4381–4386.

[14] D. Althoff, D. Wollherr and M. Buss, Safety assessment of trajectories for navigation in uncertain and dynamic environments, *2011 IEEE International Conference on Robotics and Automation*, IEEE (2011), pp. 5407–5412.

[15] X. Yang, L. Deng, J. Liu, P. Wei and H. Li, Multi-agent autonomous operations in urban air mobility

with communication constraints, *AIAA Scitech 2020 Forum*, (2020), p. 1839.

[16] B. Du, J. Chen, D. Sun, S. G. Manyam and D. W. Casbeer, Uav trajectory planning with probabilistic geo-fence via iterative chance-constrained optimization, *IEEE Transactions on Intelligent Transportation Systems* (2021).

[17] P. Wu, H. Wen, T. Chen and D. Jin, Model predictive control of rigid spacecraft with two variable speed control moment gyroscopes, *Applied Mathematics and Mechanics* **38**(11) (2017) 1551–1564.

[18] L. Blackmore, M. Ono and B. C. Williams, Chance-constrained optimal path planning with obstacles, *IEEE Transactions on Robotics* **27**(6) (2011) 1080–1094.

[19] L. Blackmore, M. Ono, A. Bektassov and B. C. Williams, A probabilistic particle-control approximation of chance-constrained stochastic predictive control, *IEEE transactions on Robotics* **26**(3) (2010) 502–517.

[20] P. Lathrop, B. Boardman and S. Martínez, Distributionally safe path planning: Wasserstein safe rrt, *IEEE Robotics and Automation Letters* **7**(1) (2021) 430–437.

[21] S. Boone and J. McMahon, Spacecraft maneuver design with non-gaussian chance constraints using gaussian mixtures, *2022 AAS/AIAA Astrodynamics Specialist Conference*, (2022).

[22] W. Dai, B. Pang and K. H. Low, Conflict-free four-dimensional path planning for urban air mobility considering airspace occupancy, *Aerospace Science and Technology* (2021) p. 107154.

[23] P. Wu, J. Xie and J. Chen, Safe path planning for unmanned aerial vehicle under location uncertainty, *2020 IEEE 16th International Conference on Control & Automation (ICCA)*, IEEE (2020), pp. 342–347.

[24] W. Han, A. Jasour and B. Williams, Non-gaussian risk bounded trajectory optimization for stochastic nonlinear systems in uncertain environments, *2022 International Conference on Robotics and Automation (ICRA)*, IEEE (2022), pp. 11044–11050.

[25] A. Devonport, F. Yang, L. El Ghaoui and M. Arcak, Data-driven reachability analysis with christoffel functions, *2021 60th IEEE Conference on Decision and Control (CDC)*, IEEE (2021), pp. 5067–5072.

[26] T. Lew and M. Pavone, Sampling-based reachability analysis: A random set theory approach with adversarial sampling, *Conference on robot learning*, PMLR (2021), pp. 2055–2070.

[27] M. P. Wand and M. C. Jones, *Kernel smoothing* (CRC press, 1994).

[28] J. Fan and J. S. Marron, Fast implementations of nonparametric curve estimators, *Journal of computational and graphical statistics* **3**(1) (1994) 35–56.

[29] J. E. Mitchell, Branch-and-cut algorithms for combinatorial optimization problems, *Handbook of applied optimization* **1**(1) (2002) 65–77.