Online Probabilistic Collision Detection for Urban Air Mobility under Data-Driven Uncertainty

Pengcheng Wu*
University of California San Diego, La Jolla, CA, 92093
San Diego State University, San Diego, CA, 92182

Jun Chen†
San Diego State University, San Diego, CA, 92182

Urban air mobility (UAM) using unmanned aerial vehicles (UAV) is an emerging way of air transportation within metropolitan areas. For the sake of the successful operations of UAM in dynamic and uncertain airspace environments, it is important to provide safe path planning for UAVs. To achieve the path planning with safety assurance, the first step is to detect collisions. Due to uncertainty, especially data-driven uncertainty, it's impossible to decide deterministically whether a collision occurs between a pair of UAVs. Instead, we are going to evaluate the probability of collision online in this paper for any general data-driven distribution. A sampling method based on kernel density estimator (KDE) is introduced to approximate the data-driven distribution of the uncertainty, and then the probability of collision can be converted to the Riemann sum of KDE values over the domain of the combined safety range. Comprehensive numerical simulations demonstrate the feasibility and efficiency of the online evaluation of probabilistic collision for UAM using the proposed algorithm of collision detection.

I. Introduction

The applications of urban air mobility (UAM) are increasingly drawing great attention from many institutes such as NASA, the Federal Aviation Administration (FAA), and airlines [1], [2]. The vision of UAM is to use revolutionary unmanned aerial vehicles (UAV) to provide efficient and on-demand air transportation service between places previously underserved by the current aviation market [3]. To well serve a significant proportion of urban transportation demand, UAM will introduce a large number of UAVs in the limited urban airspace [4], [5], where the environment is also highly uncertain due to inaccurate localization with disturbances from high buildings, such as strong Global Positioning System (GPS) [6] noise and high wind disturbance around those buildings [7]. Therefore, a key challenge for the success of UAM is how to ensure operation safety in high-density, dynamic and uncertain airspace environments in real time. Collision detection is the first step to do path planning for UAVs with safety assurance.

The most important part of probabilistic collision detection is how to fast estimate collision probability based on position uncertainty. For conventional piloted aircraft, Gaussian distribution was often used for modeling trajectory prediction error [8], [9], an thus some fast algorithms for collision or conflict probability estimation could be derived based on the properties of Gaussian distribution. For the two-dimensional situation, Paielli and Erzberger [8] first combined predicted position errors of an aircraft pair into a single relative position error that was still a Gaussian distribution and then converted this bivariate normal distribution into a standard normal distribution based on linear transformation. Zou et al. [10], [11] extended the above method to work for different shapes of safety ranges. However, all the aforementioned works adopt the assumption that the uncertainty obeys Gaussian distribution, which may not be consistent with the engineering practice [12], [13].

To efficiently evaluate the probability of collision in an online fashion, a data-driven method is developed for any general empirical distribution. Kernel Density Estimation (KDE) is a classical way to handle data-driven distributions, we can directly perform evaluations to figure out KDE values based on all samples [14]. However, a significant issue that arises in the practice of traditional KDE is its computational intensity, the direct KDE approach runs slowly because it needs to have a kernel on every data point [15]. Alternatively, we will present a way to speed up the computation of KDE at the cost of replacing kernel estimators by their approximations. The essential idea behind the operation is to

*Joint PhD Student, Department of Aerospace Engineering, Student Member AIAA, pwu@sdsu.edu,pcwupat@ucsd.edu
†Assistant Professor, Department of Aerospace Engineering, Member AIAA, jun.chen@sdsu.edu

perform kernel estimators on the grid points instead of the sampled data points. Further, the algorithm of FFT can be integrated to speed up the computation of discrete convolution to evaluate KDE values. Once we obtain kernel density estimator (KDE) to approximate the probabilistic distribution of the uncertainty, then the probability of collision can be converted to the Riemann sum of KDE values over the domain of the combined safety range.

In this paper, we are going to develop an online algorithm for probabilistic collision detection to provide support for the safe operation of UAVs. The rest of the paper is organized as follows. In Section II, the non-Gaussian distribution is captured from real applications and several fundamental concepts are formally defined. The integration to evaluate collision probability is also formulated. In Section III, the fast algorithm of collision detection based on KDE is developed. In Section IV, numerical study is conducted to demonstrate the feasibility and efficiency of the proposed algorithm. In Section V, some conclusions and summaries are drawn.

II. Problem Formulation

For a UAV in UAM, its location $\boldsymbol{x}_t$ at a particular time t may be stochastic, due to uncertainty arising from the inaccurate sensor of self-position or environmental disturbance like wind. Assume that all the UAVs fly at the same altitude. The predicted position of a UAV can be described by

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + (v\cos\theta, v\sin\theta)\Delta t \tag{1}$$

where $\boldsymbol{x}_t$ is the current position, v is the magnitude of the velocity, θ is the heading angle of the velocity, and Δt is the time step.

We use the sampling method to do reachability analysis for UAVs [16]–[18]. Assuming that v and θ both obey Gaussian distribution, then $\boldsymbol{x}_{t+1}$ doesn't obey Gaussian distribution in most cases. Instead, we can do samplings to obtain a data-driven distribution of $\boldsymbol{x}_{t+1}$. In this way, it describes the predicted position of UAV in the form of probabilistic distribution and then the collision probability of any pair of UAVs can be predicted for collision detection. When the collision probability is greater than a certain threshold , a collision can be considered to occur. Collision detection based on this can be called probabilistic collision detection under data-driven uncertainty.

A safety range is a minimum envelope that covers a UAV [19]. For simplicity, in this paper, the boundary of the safety range is assumed to be circular and its radius is constant throughout all the time steps.

Definition 1 **(Collision Occurrence)** A collision occurs when the safety ranges of both UAVs overlap

$$\|\boldsymbol{x}_{jt} - \boldsymbol{x}_{it}\| < r_j + r_i \tag{2}$$

where $\boldsymbol{x}_{jt}$ and $\boldsymbol{x}_{it}$ are the positions of UAV j and UAV i at time step t respectively. r_j and r_i are the safety ranges for both UAVs.

We have learned that $\boldsymbol{x}_{jt}$ and $\boldsymbol{x}_{it}$ are both random vectors obeying data-driven distributions. Thus, it's impossible to decide deterministically whether a collision occurs between a pair of UAVs. Instead, we are going to consider the probability of collision.

Definition 2 **(Probabilistic Collision Detection)** A collision is considered to be detected when the probability of collision occurrence is greater than a certain threshold

$$\Pr(\text{collision}) > \alpha \tag{3}$$

where α is the threshold associated with the risk level of the data-driven distribution.

The occurrence of collision can be represented by an indicator function

$$I_D(\Delta\boldsymbol{r}) = \begin{cases} 1 & \Delta\boldsymbol{r} \in D \\ 0 & \Delta\boldsymbol{r} \notin D \end{cases} \tag{4}$$

where 1 indicates that a collision happens and 0 indicates that there is no collision. D is the combined safety range and Δr is the relative position between a pair of UAV i and UAV j, i.e., $\Delta\boldsymbol{r} = \|\boldsymbol{x}_{jt} - \boldsymbol{x}_{it}\|$. t is omitted for simplicity,

The relative position $\Delta\boldsymbol{r}$ should be a random vector. Hence the probability of collision can be evaluated by

$$Pr(\Delta\boldsymbol{r} \in D) = \iiint_{\Delta\boldsymbol{r}\in D} f(\Delta\boldsymbol{r})d\Delta\boldsymbol{r} \tag{5}$$

where $f(\Delta \boldsymbol{r})$ is the PDF of the random vector $\Delta \boldsymbol{r}$. Since the position $\boldsymbol{x}_t$ of a UAV obeys a data-driven distribution as discussed above, we can obtain that the relative position $\Delta \boldsymbol{r}$ obeys another data-driven distribution.

Therefore, the evaluation of collision probability for UAVs is converted to the integral of the PDF of a data-driven distribution. The integration region is the combined safety range D of a UAV pair, and the integrand is the PDF of a data-driven distribution. Note that although we assume that the safety range of a UAV is circular in this paper, this way of evaluating collision probability applies to any shapes of safety ranges and therefore the combined safety range D.

III. Fast Algorithm of Collision Detection

According to the problem formulation above, the key point of collision detection for UAVs is to evaluate the integral in eq. (5). However, there is no analytical solution for the integral because the CDF of a data-driven distribution cannot be expressed in closed-form only by elementary functions [20]. Therefore, we are going to develop algorithms based on kernel density estimator to approximate the PDF of the data-driven distribution and therefore estimate the collision probability for UAVs.

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

$$\hat{f}(x) = \frac{1}{N} \sum_{i=1}^{N} K(x - x_i) \tag{6}$$

where N represents the number of data points.

Taking the bandwidth h of $\hat{f}(x)$ into consideration and assigning different weights w_i to different data points x_i, eq. (6) can be rewritten as

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

where $\sum\limits_{i=1}^{N} w_i = 1$.

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

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

An extension of eq. (7) to d-dimensional scenarios is to write

$$\hat{f}(x) = h^{-1} \sum_{i=1}^{N} w_i K\left(h^{-1}(x - x_i)\right) \tag{9}$$

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

$$\hat{f}_j = h^{-1} \sum_{i=1}^{N} w_i K\left(h^{-1}(g_j - x_i)\right),\ j = 1, \ldots, M \tag{10}$$

For simplicity, eq. (9) and eq. (10) can be denoted as the following form

$$\hat{f}(x) = \sum_{i=1}^{N} w_i K_h\left(x - x_i\right), \quad \hat{f}_j = \sum_{i=1}^{N} w_i K_h\left(g_j - x_i\right),\ j = 1, \ldots, M \tag{11}$$

The computational intensity of traditional KDE is very high. Instead of direct evaluation, we can go through N data points and assign weights to M equidistant grid points [15]. To speed up the evaluation of traditional KDE, the kernel function $K(\cdot)$ performed on N data points x_i can be replaced by that on M grid points g_l.

The idea is to go through every data point and then assign weights to its neighbour grid points. We need to obtain the grid weights c_l. The weights are determined by the proportion of the volume of a hyper-cube that is enclosed by the data point. For bi-variate kernel estimators, 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-rectangles induced by the position of the data point. By doing so, we can obtain the approximation of eq. (10)

$$\hat{f}_j = \sum_{l=1}^{M} K_h\left(g_j - g_l\right) c_l, \ j = 1, \ldots, M \tag{12}$$

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

Further, let $L = M - 1$ and then eq. (12) can be reformulated as

$$\hat{f}_j = \sum_{l=-L}^{L} c_{j-l} k_l, \ j = 1, \ldots, M \tag{13}$$

where

$$k_l = K_h\left(g_j - g_l\right) \tag{14}$$

Note that $c_l = 0$ for l not in the index set $\{1, \ldots, M\}$. By the symmetry of the kernel function $K_h(\cdot)$, it's only required to figure out k_l for $l = 0, 1, \ldots, L$ where $L = M - 1$. Therefore, it is clear that no more than M kernel evaluations are required to obtain k_l. This is because there are only M distinct differences among different grid points. Indeed, eq. (13) can be viewed as the discrete convolution of c_l and k_l. This means the approximation we use has a discrete convolution structure which 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 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.

Once we obtain KDE, we can use KDE to approximate the probabilistic distribution of the uncertainty in eq. (5). And then the integral of PDF can be converted to the Riemann sum of KDE values over the cells covered by the combined safety range D.

The details of the proposed algorithm are stated in **Algorithm 1**. In the pseudo codes, N is the number of sampled data points. M is the number of grid points in each dimension. α represents the risk level of the data-driven distribution. In Line 9 and 10, FFT is employed to speed up the computation of discrete convolution to obtain KDE values. In Line 14 and 15, the probability of collision is converted to the Riemann sum of KDE values over the domain of the combined safety range in eq. (5). The collision is detected if the probability of collision is greater than α; otherwise not.

In summary, we present an alternative way to replace performing kernel evaluation on data points by 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 KDE evaluation, we propose a fast algorithm of online data-driven evaluation to find probability collision for UAVs.

Algorithm 1 Fast Algorithm of Collision Detection

```
function DataGenerate(N)
    generate N data points randomly
    return data

function KernelDensityEstimator(data, M, α)
    mesh with M grid points in each dimension
    obtain grid weights c_l according to data
    evaluate kernel functions k_l on grids
    C = FFT(c_l); K = FFT(k_l)
    kde = iFFT(C * K)
    return kde

function CollisionDetection(kde, α)
    f(Δr) = kde
    discretize eq. (5) and evaluate collision probability
    if  collision probability > α then
        return collision detected
    else
        return collision not detected
```

IV. Numerical Study

A. Test Settings

In this section, we are going to evaluate the collision probability given a data-driven distribution using the fast algorithm of collision detection proposed in last section. The tests were implemented in Python 3.8 and on an Intel(R) Core(TM) i5-8400T 1.70GHz PC with 8GB RAM.

B. KDE Approximation

In this part, tests are conducted at risk level $\alpha = 20\%$. The number of data points is $N = 10^4$. The number of grid points is $M = 128 * 128$.

The confidence domain of the given data-driven distribution approximated by KDE at risk level $\alpha = 20\%$ is displayed in fig. 1.

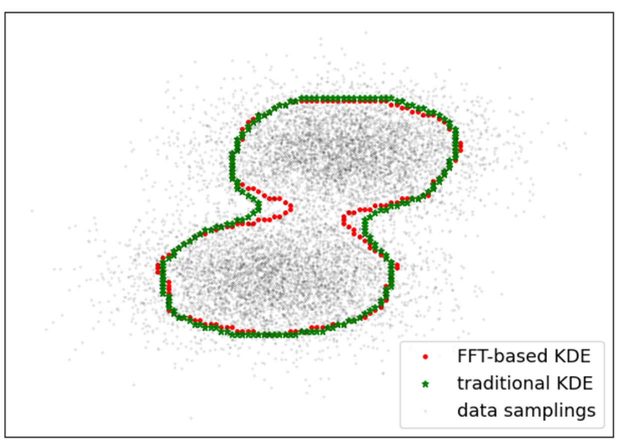

Fig. 1 Risk level = 20%

C. Collision Probability Evaluation

. . . . . .

V. 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 UAM. For uncertain UAM systems, the first step to do path planning for UAVs with safety assurance is to evaluate the collision probability between any pair of UAVs. Due to uncertainty, it's impossible to decide deterministically whether a collision occurs between a pair of UAVs. Instead, we are going to evaluate the probability of collision online. A sampling method based on kernel density estimator (KDE) is introduced to approximate the probabilistic distribution of the uncertainty, and then the probability of collision is converted to the Riemann sum of KDE values over the domain of the combined safety range. Comprehensive numerical simulations demonstrate the feasibility and efficiency of the online evaluation of probabilistic collision for UAM using the proposed method.

Acknowledgments

This work was supported by the National Science Foundation under Grants CMMI-2138612.

References

[1] L. Gipson, *Nasa embraces urban air mobility, calls for market study*, `https://www.nasa.gov/aero/nasa-embraces-urban-air-mobility`, Accessed: 2020-02-15, 2017.

[2] J. Holden and N. Goel, "Fast-forwarding to a future of on-demand urban air transportation," *San Francisco, CA*, 2016.

[3] J. Chen, l. Chen, and D. Sun, "Air traffic flow management under uncertainty using chance-constrained optimization," *Transportation Research Part B: Methodological*, vol. 102, pp. 124–141, 2017.

[4] X. Yang and P. Wei, "Scalable multi-agent computational guidance with separation assurance for autonomous urban air mobility," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 8, pp. 1473–1486, 2020.

[5] 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*, vol. 22, no. 9, pp. 6007–6017, 2020.

[6] S. Kahne and I. Frolow, "Air traffic management: Evolution with technology," *IEEE Control Systems*, vol. 16, no. 4, pp. 12–21, 1996.

[7] P. Wu, L. Li, J. Xie, and J. Chen, "Probabilistically guaranteed path planning for safe urban air mobility using chance constrained rrt," in *AIAA Aviation 2020 Forum*, 2020, p. 2914.

[8] R. A. Paielli and H. Erzberger, "Conflict probability estimation for free flight," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 3, pp. 588–596, 1997.

[9] ——, "Conflict probability estimation generalized to non-level flight," *Air Traffic Control Quarterly*, vol. 7, no. 3, pp. 195–222, 1999.

[10] Y. Zou, H. Zhang, D. Feng, H. Liu, and G. Zhong, "Fast collision detection for small unmanned aircraft systems in urban airspace," *IEEE Access*, vol. 9, pp. 16 630–16 641, 2021.

[11] Y. Zou, H. Zhang, G. Zhong, H. Liu, and D. Feng, "Collision probability estimation for small unmanned aircraft systems," *Reliability Engineering & System Safety*, vol. 213, p. 107 619, 2021.

[12] 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*, p. 107 154, 2021.

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

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

[15] J. Fan and J. S. Marron, "Fast implementations of nonparametric curve estimators," *Journal of computational and graphical statistics*, vol. 3, no. 1, pp. 35–56, 1994.

[16] V. Lefkopoulos and M. Kamgarpour, "Using uncertainty data in chance-constrained trajectory planning," in *2019 18th European Control Conference (ECC)*, IEEE, 2019, pp. 2264–2269.

[17] T. Lew, L. Janson, R. Bonalli, and M. Pavone, “A simple and efficient sampling-based algorithm for general reachability analysis,” in *Learning for Dynamics and Control Conference*, PMLR, 2022, pp. 1086–1099.

[18] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, “Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns,” *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[19] 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.

[20] H. Zhu and J. Alonso-Mora, “Chance-constrained collision avoidance for mavs in dynamic environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.