Tight Collision Probability for UAV Motion Planning in Uncertain Environment

Tianyu Liu[1], Fu Zhang[1], Fei Gao[2], and Jia Pan[3]

Abstract— **Operating unmanned aerial vehicles (UAVs) in complex environments that feature dynamic obstacles and external disturbances poses significant challenges, primarily due to the inherent uncertainty in such scenarios. Additionally, inaccurate robot localization and modeling errors further exacerbate these challenges. Recent research on UAV motion planning in static environments has been unable to cope with the rapidly changing surroundings, resulting in trajectories that may not be feasible. Moreover, previous approaches that have addressed dynamic obstacles or external disturbances in isolation are insufficient to handle the complexities of such environments. This paper proposes a reliable motion planning framework for UAVs, integrating various uncertainties into a *chance constraint* that characterizes the uncertainty in a probabilistic manner. The chance constraint provides a probabilistic safety certificate by calculating the collision probability between the robot's Gaussian-distributed forward reachable set and states of obstacles. To reduce the conservatism of the planned trajectory, we propose a tight upper bound of the collision probability and evaluate it both exactly and approximately. The approximated solution is used to generate motion primitives as a reference trajectory, while the exact solution is leveraged to iteratively optimize the trajectory for better results. Our method is thoroughly tested in simulation and real-world experiments, verifying its reliability and effectiveness in uncertain environments.**

I. INTRODUCTION

Nowadays, UAVs are increasingly deployed in various scenarios and are facing more complicated operating environments. They may fly under unknown air turbulence in the wild or work in close physical proximity to walking humans in the logistics factory. These uncertain environments, along with robot intrinsic uncertainty such as modeling errors and inaccurate localization, makes safe motion planning a challenging and computationally expensive problem [1], [2]. Online planners designed for deterministic environments are inadequate to handle uncertainty, relying solely on replanning mechanisms or feedback controllers. Thus, it is essential for motion planners to reason about uncertainty online and react intelligently with safety guarantees.

The majority of prior research on planning under uncertainty can be divided into *chance constrained approach*, and *set bounded approach*. The former leverages a probabilistic representation of uncertainty and guarantees the probability of failure is below a specified threshold [1]. This approach has been adopted to handle uncertainty arising from dynamic obstacles [3]–[5] and system modeling errors [6].

1 Department of Mechanical Engineering, The University of Hong Kong.
2 Institute of Cyber-Systems and Control, Zhejiang University.
3 Department of Computer Science, The University of Hong Kong.
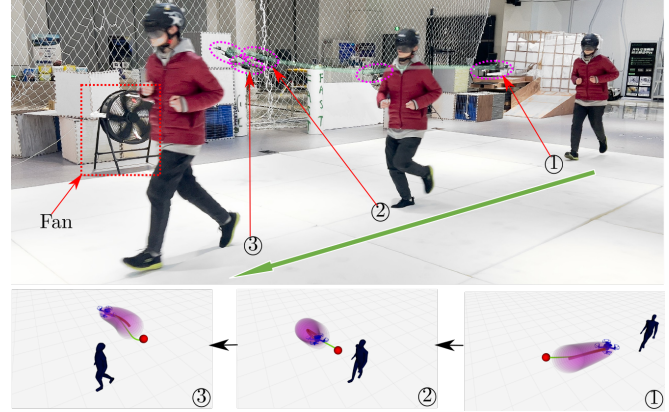Corresponding author: Tianyu Liu, `tianyu@connect.hku.hk`.

Fig. 1. The UAV aims to hover at a fixed point in a windy environment while actively avoiding a moving human. **Top**: snapshots of real world experiment. **Bottom**: data visualization in Rviz. **Red ball**: goal point. **Green line**: reference trajectory. **Brown line**: optimized trajectory. **Purple ellipsoid**: FRS.

The latter method typically ensures safety certification for robots by achieving robust planning [7] while assuming a disturbance bound, particularly for wind [2], [7], [8]. In this paper, we present a reliable motion planning framework for UAVs that combines both approaches to address the uncertainties associated with dynamic obstacles and wind disturbances. Specifically, we propose an integrated chance constraint that restricts the collision probability between the robot's Gaussian-distributed forward reachable set (FRS) and the predicted states of obstacles. However, no closed-form solution exists for estimating this probability, and previous methods have either evaluated loose upper bounds or approximations with significant errors. With various kinds of uncertainty considered, these collision estimation methods will result in overly conservative and inefficient trajectories. To this end, we propose a tight probability upper bound and give its exact and approximated solutions. The approximated solution is computed from numerical integral using Gauss-Hermit quadrature. It is served for collision chance checking of motion primitives, which compose the reference trajectory for optimization. The exact solution is derived from the series expansions of a quadratic form in random Gaussian variables. It is utilized to iteratively optimize the trajectory for better results. As shown in Figure 1, the real-world experiment demonstrated the reliability and effectiveness of our planning framework.

To conclude, the contributions of this paper are summarized as follows:

- A tight collision probability upper bound with its exact

and approximated solutions.

- A reliable motion planning framework for UAVs in highly uncertain environments with dynamic obstacles and external disturbances.
- Extensive Validation of the proposed method in simulation and real-world experiments.

II. LITERATURE REVIEW

A. Collision Probability Estimation

Collision probability estimation is essential for motion planning in probabilistic contexts. Monte Carlo (MC) provides a powerful tool for this purpose, but it requires a large number of simulation rollouts. The simulation was accelerated in [9] by approximating the double summation from the MC integration with a single summation. Assuming that obstacles are punctual objects, a representative density point on the distribution can be selected and multiplied by the volume of the robot. Park et al. [10] took the maximum density value at the robot surface and provided an upper bound, while Du et al. [11] took the density at the center. Despite their efficiency, these methods usually yield large errors. Some other methods marginalized the probability over the collision region of the configuration space. Hardy et al. [12], and Zhu et al. [3] both obtained upper bounds with collision regions as a rectangular bounding box and half hyperspace respectively. Dai et al. [6] proposed to approximate the probability via quadrature rules under the assumption that each distribution dimension is independent. Antony et al. [13] claimed to find the exact solution for ellipsoidal-shaped robots and obstacles, but there seems to be an oversight of taking random variables as constant in their theory derivation[1].

B. Planning Under Uncertainty

Planing under uncertainty can mainly be divided into *chance constrained approach* [1], [3], [5], [6] and *set bounded approach* [2], [7], [8], [14], [15]. Considering the obstacles as convex polyhedrons, Blackmore et al. [1] approximated the chance-constrained optimization problem as a disjunctive convex programming. Zhu et al. [3] proposed transforming the chance constraint to a deterministic one, typically for ellipsoidal obstacles. Benefiting from the two strategies, a hybrid solution is proposed in [5] to trade off performance and efficiency. These methods are mainly designed to handle uncertain dynamic obstacles. While in [6], the robot motion and state uncertainty is focused, and a chance-constrained motion planning system is proposed for high-dimensional robots. *Set bounded approach* reasons about the influence under bounded disturbances (e.g. air turbulence) and usually provides a safety guarantee. Anirudha et al. [2] computed funnel library offline via sum-of-squares programming and sequentially composed them

[1]See our comments https://github.com/Acmece/IROS2023_Com for more details

online for robust trajectories. The approach has been extended to multirotor systems, where the rotor drag is explicitly estimated rather than treated as an unknown disturbance [15]. Using Hamilton-Jacobi reachability, Seo et al. [8] conservatively approximate the FRS with ellipsoidal parameterization, which can be computed analytically online and was further integrated with external force estimator for the quadrotor in [14]. Manchester et al. [7] computed the ellipsoid bounds around nominal trajectory locally in the closed form under the LQR feedback controller and designed a robustness cost for trajectory optimization. Apart from these two main categories, the chance constraint can be converted to a set-bounded one by enlarging the robot with 3-σ uncertainty ellipsoids [10], [16]. Wang et al. [17] conducted dynamic avoidance in font-end path searching and then refine the trajectory by gradient descent. In [18], the safety margin library is computed via MC simulations offline and adaptively adjusted on-the-fly for disturbances.

III. MOTION PLANNING FRAMEWORK

Considering a robot moving in an uncertainty environment with dynamic obstacles, external disturbances as well as static structures, we aim to find efficient trajectories with probabilistic safety certificates. Given the models of robot and M dynamic obstacles, we discretize the planned trajectory $(\mathbf{x} \in \mathbb{R}^{n_x}, \mathbf{u} \in \mathbb{R}^{n_u})$ into N knot points in time. The motion planning problem is then formulated by the following nonlinear programming:

$$\min_{\mathbf{u}_0,\ldots,\mathbf{u}_{N-1}} J\left(\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}, \mathbf{x}_0, \ldots, \mathbf{x}_N\right), \tag{1}$$

$$\text{s.t. } \mathbf{x}_{t+1} = f\left(\mathbf{x}_t, \mathbf{u}_t\right) + \mathbf{w}_t, \tag{1a}$$

$$\mathbf{y}_{t+1}^i = g^i\left(\mathbf{y}_t^i\right) + \mathbf{v}_t^i, \tag{1b}$$

$$\mathcal{E}_t = h(\mathbf{x}_t, \mathbf{u}_t, t), \tag{1c}$$

$$\mathbf{w}_t \sim \mathcal{N}\left(\mathbf{0}, W_t\right), \quad \mathbf{v}_t^i \sim \mathcal{N}\left(\mathbf{0}, V_t^i\right), \tag{1d}$$

$$\mathbf{x}_0 \sim \mathcal{N}\left(\hat{\mathbf{x}}_0, \Sigma_{\mathbf{x},0}\right), \quad \mathbf{y}_0 \sim \mathcal{N}\left(\hat{\mathbf{y}}_0^i, \Sigma_{\mathbf{y},0}\right),$$

$$\mathbf{x}_{t+1} \in \mathcal{F}_t, \ \mathbf{u}_t \in \mathcal{U} \tag{1e}$$

$$\mathcal{P}\left(\mathcal{G}_t(\mathcal{E}_t, \mathbf{y}_t^1, \cdots \mathbf{y}_t^M) < \mathbf{0}\right) < \mathbf{\Delta}_t, \tag{1f}$$

$$\forall t = 1, 2, \ldots, N, \ \forall i = 1, 2, \ldots, M,$$

where f is the nonlinear dynamic system model of robot, g^i is the prediction model of the i-th dynamic obstacles, $\mathcal{E}_t$ is the propagated FRS computed under bounded disturbance, $\mathcal{F}_t$ is the collision free regions with regards to static structures, and $\mathcal{U}$ is the feasible control set. The function $\mathcal{G}_t(\cdot)$ encode the region that robot is in collision with dynamic obstacles. $\mathcal{P}$ indicates the probability of an event. Δ_t is the prescribed collision probability for step t. This work utilizes the Gaussian distribution and Forward Reachable Set (FRS) to address the uncertainties arising from dynamic obstacles and external disturbance, respectively. By applying the chance constraint (1f), we ensure that the collision probability between dynamic obstacles and every state in FRS is below a user-specified threshold.

In this work, we make two key assumptions. Firstly, the noise of robot and obstacle models are all in the represen-

tation of Gaussian distribution. This is valid because of the central limit theorem as the robot system evolves over time. While the Gaussian distribution of obstacles' state encodes the belief of the prediction model. Secondly, we assume that shapes of robots and dynamic obstacles are all in ellipsoids. Actually, for an arbitrary geometric shape, we can always find its minimum-volume enclosing ellipsoids, the *Löwner-John ellipsoids*, by convex optimization [19].

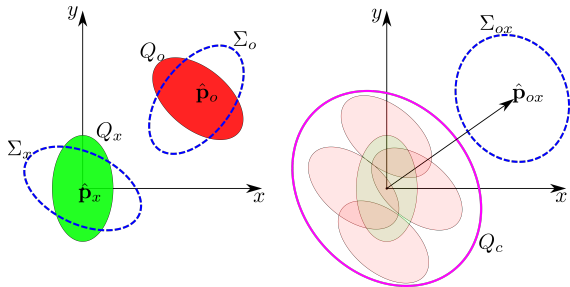## IV. TIGHT COLLISION PROBABILITY

Fig. 2. Illustration of the collision region upper bound Q_c (pink ellipse) in 2D. **Red ellipse**: obstacle shape. **Green ellipse**: robot shape. **Blue ellipse**: Gaussian distribution.

Suppose we have an ellipsoidal-shaped robot and obstacle with shape matrix of Q_x, $Q_o \in \mathbb{S}_+^3$, respectively. The collision region in configuration space is thus the Minkowski sum of these two ellipsoids. As shown in Figure 2, we approximate the region with an optimal outer ellipsoid which is denoted by:

$$Q_c = Q_x \boxplus Q_o = (1 + \alpha)Q_x + (1 + \frac{1}{\alpha})Q_o, \quad (2)$$

where $\alpha = \mathrm{Tr}(Q_o)/\mathrm{Tr}(Q_x)$. Suppose the positions and uncertainty covariances of the robot and the obstacle are $\mathbf{p}_x \sim \mathcal{N}(\hat{\mathbf{p}}_x, \Sigma_x)$, $\mathbf{p}_o \sim \mathcal{N}(\hat{\mathbf{p}}_o, \Sigma_o)$, with $\mathbf{p}_x, \mathbf{p}_o \in \mathbb{R}^3$. We can calculate the upper bound of the collision probability as:

$$\mathcal{P}_c = \int_{\|\mathbf{p}_o - \mathbf{p}_x\|_{Q_c^{-1}} < 1} (\mathbf{p}_o - \mathbf{p}_x)d(\mathbf{p}_o - \mathbf{p}_x) > \mathcal{P}, \quad (3)$$

where $\mathbf{p}_o - \mathbf{p}_x = \mathbf{p}_{ox}$ is the relative position between robot and obstacle, and $\| \mathbf{p}_{ox} \|_{Q_c^{-1}} = \mathbf{p}_{ox}^T Q_c^{-1} \mathbf{p}_{ox} < 1$ is the outer ellipsoid Q_c. The probability (3) is the integral of the relative position distribution $\mathbf{p}_{ox} \sim \mathcal{N}(\hat{\mathbf{p}}_o - \hat{\mathbf{p}}_x, \Sigma_o + \Sigma_x)$ over the ellipsoid Q_c.

In the following subsections, we present the approaches to evaluate $\mathcal{P}_c$ and how to achieve the chance constraint (1f) in the form of

$$\mathcal{P}_c\Big(\mathcal{G}_t\big(\mathcal{E}_t, y_t^1, \cdots y_t^M\big) < 0\Big) < \Delta_t. \quad (4)$$

A. Collision Probability Estimation

In the view of the random variable $\mathbf{p}_{ox}$, the probability (3) can be rewritten as

$$\mathcal{P}_c = \mathcal{P}(\mathbf{p}_{ox}^T Q_c^{-1} \mathbf{p}_{ox} < 1) = \mathcal{P}(v < 1) = F_v(1), \quad (5)$$

where $v = \mathbf{p}_{ox} Q_c^{-1} \mathbf{p}_{ox}^T$ is the quadratic form in random variable $\mathbf{p}_{ox}$ and F_v is the cumulative distribution function

(CDF) of v. According to [20], F_v has an exact expression in the form of series expansions, as stated by the following theorem.

Theorem 1: *Suppose the quadratic form $v = \mathbf{p}^T A\mathbf{p}$ with $A = A^T > 0$, $\mathbf{p} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma), \Sigma > 0$. With the affine transformation $\mathbf{z} = \Sigma^{-\frac{1}{2}} \mathbf{p} - \Sigma^{-\frac{1}{2}} \boldsymbol{\mu}$ and the diagonalization $P^T \Sigma^{\frac{1}{2}} A \Sigma^{\frac{1}{2}} P = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, the quadratic form v can be expressed as:*

$$v = (\mathbf{w} + \mathbf{b})^T \mathrm{diag}(\lambda_1, \ldots, \lambda_n)(\mathbf{w} + \mathbf{b}),$$

where P is an orthogonal matrix ($PP^T = I$), $\mathbf{w} = P^T \mathbf{z} = (w_1, \ldots, w_n)^T$ and $\mathbf{b} = P^T \Sigma^{-\frac{1}{2}} \boldsymbol{\mu} = (b_1, \ldots, b_n)^T$. In this representation, the CDF of v is

$$F_v(q) = P(v \le q) = \sum_{k=0}^{\infty} (-1)^k c_k \frac{q^{\frac{n}{2}+k}}{\Gamma\left(\frac{n}{2} + k + 1\right)},$$

where Γ denotes the gamma function and

$$c_0 = \exp\left(-\frac{1}{2}\sum_{i=1}^n b_i^2\right) \prod_{i=1}^n (2\lambda_i)^{-\frac{1}{2}},$$

$$c_k = \frac{1}{k}\sum_{i=0}^{k-1} d_{k-i}c_i,$$

$$d_k = \frac{1}{2}\sum_{i=1}^n \left(1 - kb_i^2\right)(2\lambda_i)^{-k}.$$

In our case, the $n = 3$ and $q = 1$. The convergent proof of this infinite series as well as the derivation of the truncation error upper bound, is presented in [21], [22].

Additionally, the probability $\mathcal{P}_c$ can be computed numerically via Gauss-Hermit quadrature. First of all, we apply the affine coordinate transform $\mathbf{r} = R^T \mathbf{p}_{ox}$, where R is an orthogonal matrix that diagonalizes Σ_{ox}, i.e. $R^T \Sigma_{ox} R = \Lambda$. Thus, each dimension $\mathbf{r}^i$ of $\mathbf{r}$ is independent from each other. Under the new coordinates, $\mathbf{p}_{ox}$ and Q_c become $\mathbf{r}$ and Q_r:

$$\begin{aligned} \mathbf{r} &\sim \mathcal{N}(\hat{\mathbf{r}}, \Sigma_r) = \mathcal{N}(R^T \hat{\mathbf{p}}_{ox}, \Lambda), \\ Q_r &= (RQ_c^{-1}R^T)^{-1}. \end{aligned} \quad (6)$$

Then the probability $\mathcal{P}_c$ is rewritten as

$$\mathcal{P}_c = \int_{\mathbb{R}^3} I(\mathbf{r})p(\mathbf{r})d\mathbf{r}, \quad (7)$$

where $I(\cdot)$ is a function to indicate whether $\mathbf{r}$ is in the collision region:

$$I(\mathbf{r}) = \begin{cases} 1, & \mathbf{r}^T Q_r^{-1} \mathbf{r} < 1, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Since all the dimensions of $\mathbf{r}$ are independent, we have :

$$\begin{aligned} p(\mathbf{r}) &= p(r^1)p(r^2)p(r^3), \\ p(r^i) &= \frac{1}{\sigma_i \sqrt{2\pi}} \exp(-\frac{(r^i - \mu_i)^2}{2\sigma_i^2}). \end{aligned} \quad (9)$$

Applying (9) to (7) yields:

$$\mathcal{P}_c = \int_{-\infty}^{\infty} p(r^1) \int_{\mathbb{R}^2} p(\mathbf{r}^{2:3})I(\mathbf{r})d\mathbf{r}^{2:3}dr^1. \quad (10)$$

Let $g(r^1) = \int_{\mathbb{R}^2} p(\mathbf{r}^{2:3}) I(\mathbf{r}) d\mathbf{r}^{2:3}$, then:

$$
\begin{aligned}
\mathcal{P}_c &= \int_{-\infty}^{\infty} p(r^1) g(r^1) dr^1 \\
&= \int_{-\infty}^{\infty} \frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left(-\frac{(r^1 - \mu_1)^2}{2\sigma_1^2}\right) g(r^1) dr^1.
\end{aligned}
\tag{11}
$$

Gauss-Hermite quadrature approximates the value of integrals by calculating the weighted sum of the integrand function at a finite number of reference points, i.e.

$$
\int_{-\infty}^{\infty} e^{-z^2} h(z) dz \approx \sum_{j=1}^{n} w_j h(z_j),
\tag{12}
$$

where n is the number of sampled points, z_j are the roots of the Hermite polynomial $H_n(z)$ and the associated weights w_j are given by:

$$
w_j = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 [H_{n-1}(z_j)]^2}.
\tag{13}
$$

To match (11) to (12), we take the following variable change:

$$
z_1 = \frac{r^1 - \mu_1}{\sqrt{2}\sigma_1} \Leftrightarrow r^1 = \sqrt{2}\sigma_1 z_1 + \mu_1,
\tag{14}
$$

such that the Equation (11) becomes

$$
\mathcal{P}_c = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} e^{-(z_1)^2} g(\sqrt{2}\sigma_1 z_1 + \mu_1) dz_1.
$$

Hence, the value of $\mathcal{P}_c$ can then be approximated through Gauss-Hermite quadrature rule:

$$
\mathcal{P}_c \approx \frac{1}{\sqrt{\pi}} \sum_{j=1}^{n_1} w_{1,j} g\left(\sqrt{2}\sigma_1 z_{1,j} + \mu_1\right).
\tag{15}
$$

By iteratively applying this procedure from r_1^t to r_3^t, we can obtain a numerical integral of $\mathcal{P}_c$:

$$
\begin{aligned}
\mathcal{P}_c \approx \pi^{-\frac{3}{2}} \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \sum_{j_3=1}^{n_3} &\left(\prod_{i=1}^{d} w_{i,j_i}\right) I([\sqrt{2}\sigma_1 z_{1,j_1} \\
&+ \mu_1, \sqrt{2}\sigma_2 z_{2,j_2} + \mu_2, \sqrt{2}\sigma_3 z_{3,j_3} + \mu_3]^T).
\end{aligned}
\tag{16}
$$
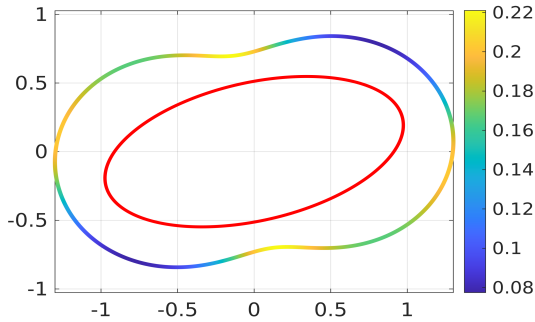
B. Iterative Trajectory Optimization

Fig. 3. 0.3-level set for the linearized collision probability in 2D. The red ellipse indicates the collision region, and the color on level set encodes the ground truth probability value.

The linearized collision probability $\mathcal{P}_l$ proposed in [3] has the ability to convert the chance constraint $\mathcal{P}_l < \Delta$

Algorithm 1: Iterative Trajectory Optimization

Input: Total allowed risk Δ, the optimized steps n
Result: the optimized trajectory Γ

1 **Function IterTrajOpt():**
2      $\Delta_l \leftarrow 0, \ \delta_l \leftarrow 0$
3      $\Delta_h \leftarrow n, \ \delta_h \leftarrow 0$
4      $\Gamma_0 \leftarrow$ trajectory optimization without chance constraint
5      $\delta_0 \leftarrow$ get trajectory risk $n\mathcal{P}_c$ of Γ_0
6      $\delta_h \leftarrow \delta_0$
7   **if** $\delta_0 \leq \Delta$ **then return** Γ_0
8
9      $k \leftarrow 0$
10   **while** $|\delta_k - \Delta| > precision$ **do**
11          $k \leftarrow k + 1$
12          $\Delta_k \leftarrow \Delta_l + (\Delta_h - \Delta_l)/(\delta_h - \delta_l)(\delta - \delta_l)$
13          $\Gamma_k \leftarrow$ trajectory optimization with $\mathcal{P}_l < \Delta_k/n$
14          $\delta_k \leftarrow$ get trajectory risk $n\mathcal{P}_c$ of Γ_k
15   **if** $\delta_k < \Delta$ **then** $\Delta_l = \Delta_t, \ \delta_l = \delta_k$
16   **else** $\Delta_h = \Delta_t, \ \delta_h = \delta_k$
17   **return** Γ_k

---

into a deterministic one, thus enabling efficient trajectory optimization. We summarize the method in Conclusion 1.

Conclusion 1: *Let the collision region in configuration space be the ellipsoid $\mathbf{p}^T Q^{-1} \mathbf{p} < 1$ with $\mathbf{p} \in \mathbb{R}^3, Q \in \mathbb{S}_+^3$. Denote the robot-obstacle relative position distribution as $\mathbf{p} \sim \mathcal{N}(\hat{\mathbf{p}}, \Sigma)$. The linearized collision probability $\mathcal{P}_l$ is given by*

$$
\mathcal{P} < \mathcal{P}_l = \frac{1}{2} + \frac{1}{2} \operatorname{erf}\left(\frac{1 - \mathbf{a}^T Q^{-\frac{1}{2}} \hat{\mathbf{p}}}{\sqrt{2\mathbf{a}^T Q^{-\frac{1}{2}} \Sigma Q^{-\frac{1}{2}} \mathbf{a}}}\right),
$$

where $\mathbf{a} = Q^{-\frac{1}{2}} \hat{\mathbf{p}}/\|Q^{-\frac{1}{2}} \hat{\mathbf{p}}\|$. The chance constraint $\mathcal{P}_l < \Delta$ can then be transformed to the a deterministic one:

$$
\mathbf{a}^T Q^{-\frac{1}{2}} \hat{\mathbf{p}} - 1 > \operatorname{erf}^{-1}(1 - 2\Delta)\sqrt{2\mathbf{a}^T Q^{-\frac{1}{2}} \Sigma Q^{-\frac{1}{2}} \mathbf{a}}.
$$

where $\operatorname{erf}(\cdot)$ is the error function: $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$.

However, as illustrated in Figure 3, this method overestimates the probability and gives rise to a loose chance constraint. And our exact solution of $\mathcal{P}_c$ equals the ground truth value, while the true estimation of $\mathcal{P}_l$ ranges from 0.0774 to 0.2211 for the 0.3-level set. By combining with the linearized collision probability, we propose an iterative trajectory optimization method to achieve the tight constraint (4).

In particular, given a trajectory $\Gamma = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, we take uniform risk allocation $\Delta_t = \Delta/n$ for every step. The collision probability for each step is taken as the maximum one among all obstacles. As mentioned in [23], summing the collision probabilities at discrete time steps will double count the probability. Thus the risk here (i.e. the sum of all probabilities in different steps) is not necessarily to be lower than one. Notice that trajectory optimization under linearized probability $\mathcal{P}_l$ tightens the original problem, while trajectory optimization without chance constraint is a relaxation. Hence our goal is to achieve the tight chance constraint (4) between them. As shown in Algorithm 1, trajectory optimization with $\mathcal{P}_l < \Delta/n$ is solved iteratively. After each solving step, the exact solution of $\mathcal{P}_c$ is employed to evaluate the risk of

optimized trajectory and provide a guess of the next suitable value of $\mathcal{P}_l$. Observing that $\mathcal{P}_l$ has positive correlation with tight probability $\mathcal{P}_c$ in general, we guess the next value of $\mathcal{P}_l$ via proportional interpolation. It has been verified in our experiments that this heuristic can help us quickly reach the tight chance constraint. To reduce the computation time, the previously obtained solution can serve as an initial guess for the next iteration of the problem.
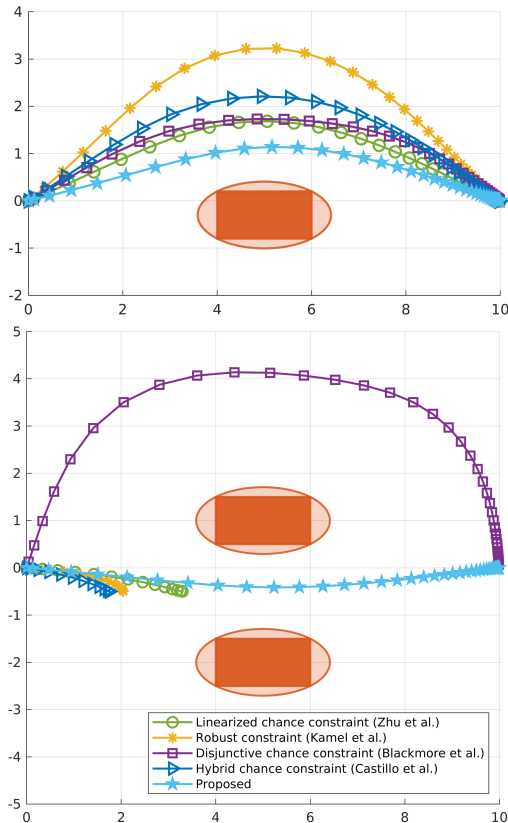
C. Benchmark Comparisons

Fig. 4. Benchmark comparison of our tight chance constraint against the linearized chance constraint [3], robust constraint [16], disjunctive chance constraint [1] and hybrid chance constraint [5].

We statistically compare our collision probability estimation methods with others[2] in a setting where the mean position of the first ellipsoid is at the origin, the mean position of the second ellipsoid is uniformly generated within a $4 \times 4 \times 4$ cube centered at the origin, and the covariance matrices are randomly generated with $\sigma^2 \in [0.01, 2]$ for each independent dimension. The semi-axis length of the two ellipsoids ranges in $[0.2, 2]$; the first ellipsoid is axis aligned and the orientation of the second one is randomly generated. We evaluate and benchmark our methods in this setting through ten thousand cases and the resulting performance is presented in Table I, showing that our methods achieve the best result in the limited time. Although the fast Monte Carlo method has the most accurate result, it requires a

TABLE I

BENCHMARK STATISTICS ON COLLISION PROBABILITY ESTIMATION

Methods	Collision Probability Error	Computation Time (s)
Fast Monte Carlo [9]	-0.0020 $\pm$ 0.0024	4.3425 $\pm$ 0.9781
Max Point Approximation [24]	0.5483 $\pm$ 0.3957	0.0883 $\pm$ 0.0361
Center Point Approximation [11]	0.3390 $\pm$ 0.2390	0.0004 $\pm$ 0.0002
Linearized Probability [3]	0.4314 $\pm$ 0.1630	0.0016 $\pm$ 0.0008
Rectangular Bounding Box [12]	0.3224 $\pm$ 0.1547	0.0035 $\pm$ 0.0011
Sphere Approximation [21]	0.4928 $\pm$ 0.1885	0.0066 $\pm$ 0.0019
Markov's Inequality [13]	0.5479 $\pm$0.7966	0.0013 $\pm$ 0.0003
Ours-Approximate ($n = 10$)	0.1799 $\pm$ 0.1925	0.0018 $\pm$ 0.0010
Ours-Approximate ($n = 200$)	0.1372 $\pm$ 0.1432	0.0201 $\pm$ 0.0065
Ours - Exact	0.1257 $\pm$0.1364	0.0063 $\pm$ 0.0032

TABLE II

BENCHMARK STATISTICS ON ONE HORIZON PLANNING

	Scenes	**Proposed**	**Zhu**	**Kamel**	**Blackmore**	**Castillo**
	Obj	**1.00**	1.03	1.22	1.05	1.09
I	Time	1.00	0.83	0.52	119.26	**0.46**
	Risk	**0.4001**	0.0530	0	0.0314	0.0037
	Obj	**1.00**	2.03	2.41	1.69	2.65
II	Time	1.00	0.31	0.27	6246.49	**0.22**
	Risk	**0.0107**	0.0038	0	0	0

significant amount of time for sampling. On the other hand, our approximation method with a low number of quadrature points ($n = 10$) can provide a good result in a short amount of time, but further efforts are required to enhance its accuracy to match that of the exact method. Thus, the method is adopted to check motion primitives to provide a reference trajectory for optimization.

The iterative trajectory optimization algorithm is tested in two one-horizon planning scenes. A robot of single mass point starts from the origin and is expected to reach the goal at $[10, 0]$. All the obstacles are within a bounding box of size $[1, 0.5]$. In the first scene, there is only one obstacle with mean position $p = [5, -0.3]$ and covariance $\Sigma = \text{diag}(0.5, 0.3)$. The second scene has two obstacles and the mean positions and covariances are $p_1 = [5, -2]$, $p_2 = [5, 1]$, $\Sigma_1 = \text{diag}(0.4, 0.07)$, $\Sigma_2 = \text{diag}(0.4, 0.07)$, respectively. We set a planning horizon of 8 seconds with $N = 40$ steps. The prescribed collision risk is 0.4 for scene I and 0.01 for scene II. As shown in Figure 4 and Table II, our iterative trajectory optimization method achieves the least conservative results compared to all the other methods in scene I. In scene II, only our method can go through the narrow gap and all the other methods fall into the local minima except [1]. Although the method [1] can find the global optimal solution, but its use of the linearized collision probability leads to an overly conservative solution. All these methods are implemented in Matlab using CasADi framework [25]. Our method iterates three times for scene I while spending a comparable time to [3]. In scene II, our method iterates two times and the methods [1], [5], [16] early exit because of reaching local minima.

V. UAV Motion Planning

In this section, we implement the motion motion planning framework (1) in the Model Predictive Control (MPC) fashion for a quadrotor and conduct experiments in both simulation and real world.

A. Robot model

We consider the dynamics model with accurate control of Euler angles, which assumes that its rate can accurately track the desired command. The system state is $\mathbf{x} = [\mathbf{p}, \mathbf{v}, \phi, \theta, \psi]^T \in \mathcal{X} \subset \mathbb{R}^{n_x}$, where $\mathbf{p} = [p_x, p_y, p_z]^T$, $\mathbf{v} = [v_x, v_y, v_z]^T \in \mathbb{R}^3$ denotes the position and velocity of the quadrotor. $\phi, \theta, \psi \in \mathbb{R}$ are the roll, pitch and yaw angles. The control input is $\mathbf{u} = \left[\dot{\phi}_c, \dot{\theta}_c, \dot{\psi}_c, f_c\right]^T \in \mathcal{U} \subset \mathbb{R}^{n_u}$ in which $\dot{\phi}_c, \dot{\theta}_c, \dot{\psi}_c \in \mathbb{R}$ are the command rates of the Euler angles, $f_c \in \mathbb{R}$ is the total thrust command of the quadrotor in the body frame. We model the quadrotor dynamics as follows:

$$\dot{\mathbf{p}} = \mathbf{v}$$
$$\dot{\phi} = \dot{\phi}_c, \dot{\theta} = \dot{\theta}_c,$$
$$\dot{\mathbf{v}} = \frac{1}{m}\left(R\begin{bmatrix}0\\0\\f_c\end{bmatrix}\right) - \begin{bmatrix}0\\0\\g\end{bmatrix}, \quad (17)$$

where $R \in \mathbb{R}^{3\times 3}$ is the rotation matrix parameterized by the Euler angles, $g \in \mathbb{R}$ is the magnitude of gravitational acceleration, and $m \in \mathbb{R}$ is the mass of the quadrotor. The discrete dynamics in (1a) is obtained through 4-th order Runge-Kutta integration of (17).

B. Dynamic Obstacle Model

The constant velocity model is used for dynamic obstacle prediction because it outperforms most other models in many aspects [26]. Thus, defining the obstacle state as $\mathbf{y}_t^i = \left[\mathbf{q}_t^i, \mathbf{v}_t^i, \phi_t^i, \theta_t^i, \psi_t^i, \dot{\phi}_t^i, \dot{\theta}_t^i, \dot{\psi}_t^i\right]$, the constant velocity model is given by:

$$\dot{\mathbf{q}}^i = R\left(\phi^i, \theta^i, \psi^i\right)\mathbf{v}^i,$$
$$\dot{\mathbf{v}}^i = \mathbf{0}, \ddot{\phi}^i = \ddot{\theta}^i = \ddot{\psi}^i = 0, \quad (18)$$

where $\mathbf{q}^i$ is the position, ϕ^i, θ^i, ψ^i are the roll pitch and yaw angles respectively, and $\mathbf{v}^i$ is the velocity in the body frame. The discrete dynamics $g^i(\mathbf{y}_t^i)$ in (1b) is obtained by Euler integration of (18).

C. Uncertainty Propagation

To evaluate the chance constraint (1f), the uncertainty covariances of the states in (1a), (1b) and the FRS in (1c) need to be propagated over time. For the uncertainty of the robot and obstacle states, since in our case the planning horizon is short, we propagate the covariances using an EKF-like update to achieve real time performance:

$$\Sigma_{x,t+1} = F_{x,t}\Sigma_{x,t}(F_{x,t})^T + W_t,$$
$$\Sigma_{y,t+1}^i = F_{y,t}^i\Sigma_{y,t}^i(F_{y,t}^i)^T + V_t, \quad (19)$$

where $F_{x,t} = \left.\frac{\partial f}{\partial \mathbf{x}_t}\right|_{\hat{\mathbf{x}}_t, \mathbf{u}_t}$ and $F_{y,t}^i = \left.\frac{\partial g}{\partial \mathbf{y}_t^i}\right|_{\hat{\mathbf{y}}_t^i}$. For the uncertainty results from external disturbances, we use [8] to calculate the FRS in a representation of ellipsoid Q_d.

D. Cost Function

We define the cost function as:

$$J = \left\|\mathbf{p}_N - \mathbf{p}_N^{ref}\right\|_{l_p^N} + \sum_{t=1}^{N-1}\left\|\mathbf{p}_t - \mathbf{p}_t^{ref}\right\|_{l_p}$$
$$+ \sum_{t=1}^{N}\|\mathbf{u}_{t-1}\|_{l_u} + \sum_{t=1}^{N-1} + \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_{l_{\Delta u}} \quad (20)$$

where l_p^N, l_p, l_u, $l_{\Delta u}$ are norm-related weighting matrixes, and $\mathbf{p}^{ref}$ is the position obtained from the reference trajectory. The reference trajectory is generated using a kinodynamic hybrid-state A* algorithm proposed in [27], which samples in the control space to generate motion primitives. We assess the risk of these primitives using the approximation method proposed in Section IV-A. The cost function (20) is designed to penalize the trajectory for tracking performance, energy efficiency, and smoothness.

E. Collision Free Constraint

For the static structure-related collision-free constraint in (1e), we construct the flight corridor, a list of connected convex polyhedrons, using the method presented in [28]. The flight corridor indicates the collision free region and we always constrain the FRS-augmented robot shape $Q_a = Q_x \boxplus Q_d$ within it to ensure safety under bounded disturbance. Specifically, we denote Q_a at time t as $Q_{a,t}$ and then assign a polyhedron $\{A\mathbf{p} \leq \mathbf{b}\}$ to constrain it, with $A \in \mathbb{R}^{n\times 3}, \mathbf{b} \in \mathbb{R}^n$. Then, the point on the ellipsoid $Q_{a,t}$, with the minimum signed distance to the i-th plane of the polyhedron, is

$$\mathbf{z}_t^i = \mathbf{p}_t + \frac{Q_{a,t}A_i}{\left\|Q_{a,t}^{1/2}A_i\right\|}. \quad (21)$$

We achieve the collision-free constraint in (1e) by enforcing all such points $\mathbf{z}_t^i$ to locate within their corresponding polyhedrons:

$$A\mathbf{z}_t^i \leq \mathbf{b} \Leftrightarrow A\mathbf{p}_t + \left\|Q_{a,t}^{1/2}A\right\| < \mathbf{b}. \quad (22)$$

F. Chance Constraint

The chance constraint, as defined in equation (1f), is designed to maintain the probability of collision below a user-specified threshold over a forthcoming time horizon. We use our proposed method in Section IV-B to iteratively reach the tight chance constraint (4). Since the robot shape is augmented with the FRS Q_d, the collision region now becomes $Q_c = Q_x \boxplus Q_o \boxplus Q_d$. For real-time planning, we limit the iterative optimization time to a prescribed value. Additionally, we set $\Delta_t = \Delta$ in Algorithm 1 and solve it in the first iteration, so that the worst case result is that we reach the optimized trajectory with linearized collision probability.

G. Simulation Experiment

The experiments are conducted in two scenarios: crowd navigation (CN) and factory patrolling (FP), as shown in Figure 6. In the first scenario, the robot navigates in a $20 \times 10\,m$ environment populated with 20 humans. In

Scenarios		Succ R (%)	Mis T (s)	Risk	Comp T (ms)
	Ours	**96**	**22.5**	**0.18**	52.7
CN	[30]	90	26.2	0.016	43.4
	[17]	78	27.1	3.5	**8.3**
	Ours	**99**	15.7	**0.15**	45.5
FP	[30]	92	23.8	0.004	38.6
	[17]	85	**13.9**	2.6	**9.1**

Fig. 5. The breakdown of the computation time for the proposed motion planning framework.

the second scenario, the robot travels in a $20 \times 12\,m$ environment with factory-like static structure and 12 randomly walking humans. All humans are driven by the social force model[3] with randomly generated goals in the space and an update rate of $50\,Hz$. The mission of robot is to move back and forth between the two corners on the diagonal. The noise on the position of robot and humans is set to $\Sigma_p = \mathrm{diag}(0.05, 0.05, 0.05)\,m^2$ and $\Sigma_q = \mathrm{diag}(0.05, 0.05, 0)\,m^2$, respectively. The velocity uncertainty is considered as $\mathrm{diag}(0.03, 0.03, 0.03)\,m^2/s^2$ and $\Sigma_p = \mathrm{diag}(0.03, 0.03, 0)\,m^2/s^2$. The shapes of robot and human are modeled as ellipsoids with semi-axis $[0.22, 0.22, 0.1]\,m$ and $[0.3, 0.3, 0.875]\,m$, respectively. For our planning framework, we select a planning horizon of $1\,s$ with 20 steps and the obstacles within a 5-meter radius of the robot are taken into consideration. The prescribed trajectory risk is 0.2 with uniform allocation strategy $\Delta_t = 0.2/20 = 0.01$. The bounded disturbance on the external force is $0.3\,mg$. The limit of the trajectory optimization time is $45\,ms$ and the maximum speed of the robot and humans is $2\,m/s$. We use the code generation engine FORCES Pro [29] to generate our optimization solver and implement our motion planning framework in C++ 11. The simulations are run on a laptop with Intel i7-9750H CPU.

We compared our work with [30] and [17], which can handle the static environment with dynamic obstacles. Each method runs for two hours in the two scenarios and the average performance on success rate, mission completion time, trajectory risk and computation time is compared. The planning is regarded as failure whenever the robot collides with moving humans. The trajectory and computation time is only calculated when there exist humans in the planning horizon. From the Table III , we can conclude that the proposed planning framework is more reliable and less conservative. Compared to [17], our method has a longer computation time but still meets the real-time planning requirement.

H. Real-world Experiment

As illustrated in Figure 1, the robot is faced with an unintentionally aggressive human walking in its vicinity, along with the wind disturbance from a nearby fan. The robot is ordered to hover in a fixed position and respond appropriately when the human is at risk of colliding with it. To track the movements of both the robot and the human,

---
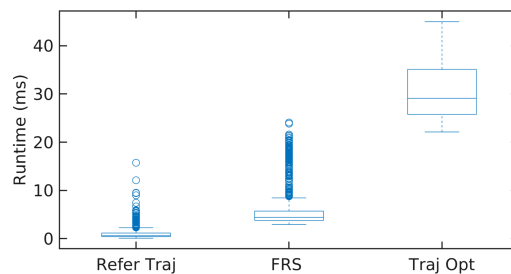
[3]https://github.com/srl-freiburg/pedsim_ros

a motion capture system is employed to capture their poses, which are then processed by the EKF algorithm with the models (17) and (18). The human's position and velocity uncertainty are set to $\Sigma_q = \mathrm{diag}(0.10, 0.10, 0), m^2$ and $\Sigma_q = \mathrm{diag}(0.10, 0.10, 0), m^2/s^2$, respectively. The allocated collision probability for each step is set to the value of 0.03. Furthermore, to account for the impact of wind, an external force of $1.5\,mg$ is set in the planning parameters. All the other parameters are the same as those described in Section V-G. The MPC is running on the onboard computer Orin NX, and the breakdown of the computation time is displayed in Figure 5. As shown in the accompanying video, our planner can reliably operate in highly uncertain environments with real-time performance.

VI. CONCLUSION

This paper proposed a reliable motion planning framework for UAVs operating in highly uncertain environments with dynamic obstacles and external disturbances. To enhance the efficiency of the planned trajectory, we proposed a tight upper bound for collision probability and evaluated it using both exact and approximate methods. Future research directions include integrating our visibility planning algorithm [31] to reduce localization uncertainty and developing a robust planning framework that can effectively handle the challenges of complex environments.

REFERENCES

[1] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011. I, I, II-B, 4, IV-C

[2] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017. I, I, II-B

[3] 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. I, II-A, II-B, IV-B, 4, I, IV-C

[4] J. Lin, H. Zhu, and J. Alonso-Mora, "Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2682–2688. I

[5] M. Castillo-Lopez, P. Ludivig, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos, "A real-time approach for chance-constrained motion planning with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3620–3625, 2020. I, II-B, 4, 2, IV-C

[6] S. Dai, S. Schaffert, A. Jasour, A. Hofmann, and B. Williams, "Chance constrained motion planning for high-dimensional robots," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8805–8811. I, II-A, II-B
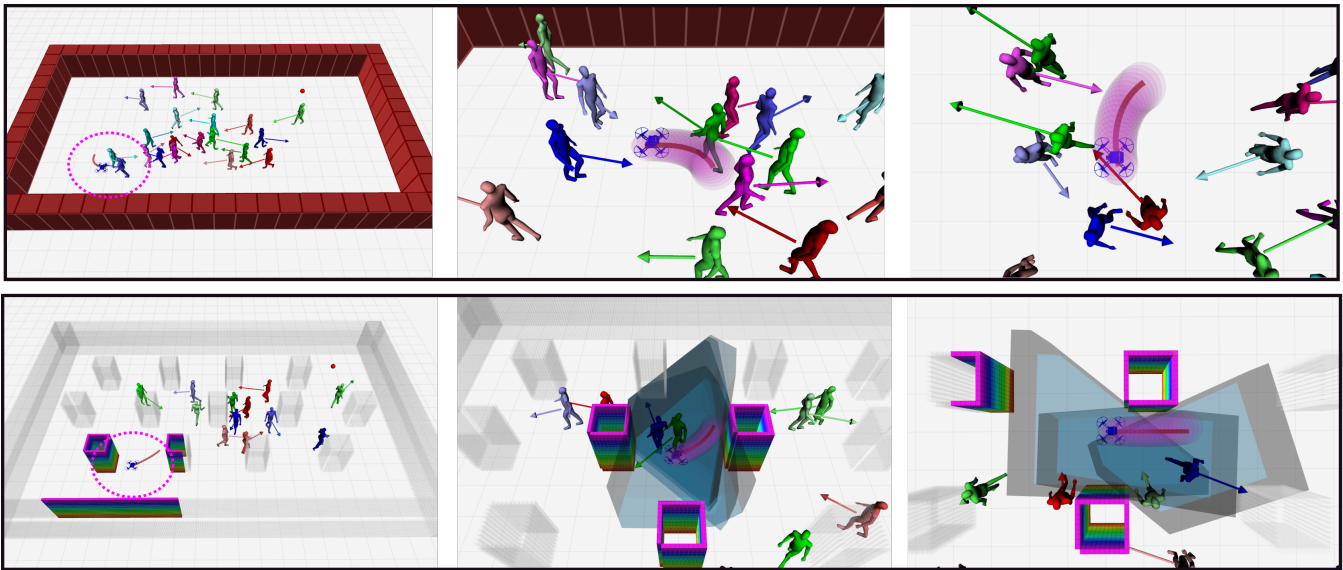
Fig. 6. Snapshots of our planning framework in simulation. **Top**: scenario I. **Bottom**: scenario II. **Arrow vector**: velocity of the human. **Colored map**: known static structure. **Translucent map**: unknown static structure. **Blue polyhedron**: flight corridor. Other labels are the same as Figure 1

[7] Z. Manchester and S. Kuindersma, "Dirtrel: Robust trajectory optimization with ellipsoidal disturbances and lqr feedback." in *Robotics: Science and Systems*. Cambridge, MA, USA, 2017. I, II-B

[8] H. Seo, D. Lee, C. Y. Son, C. J. Tomlin, and H. J. Kim, "Robust trajectory planning for a multirotor against disturbance based on hamilton-jacobi reachability analysis," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3150–3157. I, II-B, V-C

[9] A. Lambert, D. Gruyer, and G. Saint Pierre, "A fast monte carlo algorithm for collision probability estimation," in *2008 10th International Conference on Control, Automation, Robotics and Vision*. IEEE, 2008, pp. 406–411. II-A, I

[10] C. Park, J. Pan, and D. Manocha, "Itomp: Incremental trajectory optimization for real-time replanning in dynamic environments," in *Twenty-Second International Conference on Automated Planning and Scheduling*, 2012. II-A, II-B

[11] N. E. Du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 809–815, 2011. II-A, I

[12] J. Hardy and M. Campbell, "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 913–929, 2013. II-A, I

[13] A. Thomas, F. Mastrogiovanni, and M. Baglietto, "Exact and bounded collision probability for motion planning under gaussian uncertainty," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 167–174, 2021. II-A, I

[14] Y. Wu, Z. Ding, C. Xu, and F. Gao, "External forces resilient safe motion planning for quadrotor," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8506–8513, 2021. II-B

[15] S. Kim, D. Falanga, and D. Scaramuzza, "Computing the forward reachable set for a multirotor under first-order aerodynamic effects," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2934–2941, 2018. II-B

[16] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 236–243. II-B, 4, IV-C

[17] Y. Wang, J. Ji, Q. Wang, C. Xu, and F. Gao, "Autonomous flights in dynamic environments with onboard vision," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1966–1973. II-B, III, V-G

[18] C. Ho, J. Patrikar, R. Bonatti, and S. Scherer, "Adaptive safety margin estimation for safe real-time replanning under time-varying disturbance," *arXiv preprint arXiv:2110.03119*, 2021. II-B

[19] E. Rimon and S. P. Boyd, "Obstacle collision detection using best ellipsoid fit," *Journal of Intelligent and Robotic Systems*, vol. 18, pp. 105–126, 1997. III

[20] A. M. Mathai and S. B. Provost, *Quadratic Forms in Random Variables: Theory and Applications*. Quadratic forms in random variables :. IV-A

[21] A. Thomas, F. Mastrogiovanni, and M. Baglietto, "An integrated localization, motion planning and obstacle avoidance algorithm in belief space," *Intelligent Service Robotics*, vol. 14, pp. 235–250, 2021. IV-A, I

[22] S. Kotz, N. L. Johnson, and D. Boyd, "Series representations of distributions of quadratic forms in normal variables. i. central case," *The Annals of Mathematical Statistics*, vol. 38, no. 3, pp. 823–837, 1967. IV-A

[23] K. M. Frey, T. J. Steiner, and J. P. How, "Collision probabilities for continuous-time systems without sampling [with appendices]," *arXiv preprint arXiv:2006.01109*, 2020. 17

[24] C. Park, J. S. Park, and D. Manocha, "Fast and bounded probabilistic collision detection for high-dof trajectory planning in dynamic environments," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 980–991, 2018. I

[25] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018. IV-C

[26] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020. V-B

[27] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019. V-D

[28] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, 2017. V-E

[29] A. Zanelli, A. Domahidi, J. Jerez, and M. Morari, "Forces nlp: an efficient implementation of interior-point... methods for multistage nonlinear nonconvex programs," *International Journal of Control*, pp. 1–17, 2017. V-G

[30] Z. Xu, D. Deng, Y. Dong, and K. Shimada, "Dpmpc-planner: A real-time uav trajectory planning framework for complex static environments with dynamic obstacles," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 250–256. III, V-G

[31] T. Liu, Q. Wang, X. Zhong, Z. Wang, C. Xu, F. Zhang, and F. Gao, "Star-convex constrained optimization for visibility planning with application to aerial inspection," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7861–7867. VI