IEEE Transactions on Aerospace and Electronic Systems, Volume 57, Issue 2, April 2021, pp. 1028-1045

DOI: 10.1109/TAES.2020.3037417

Fast Generation of Chance-Constrained Flight Trajectory for Unmanned Vehicles

Runqi Chai, *Member, IEEE,* Antonios Tsourdos, Al Savvaris, Shuo Wang, Yuanqing Xia, *Senior Member, IEEE,* and Senchun Chai, *Senior Member, IEEE,*

***Abstract*—In this work, a fast chance-constrained trajectory generation strategy incorporating convex optimization and convex approximation of chance constraints is designed so as to solve the unmanned vehicle path planning problem. A path-length-optimal unmanned vehicle trajectory optimization model is constructed with the consideration of the pitch angle constraint, the curvature radius constraint, the probabilistic control actuation constraint, and the probabilistic collision avoidance constraint. Subsequently, convexification technique is introduced to convert the nonlinear problem formulation into a convex form. To deal with the probabilistic constraints in the optimization model, convex approximation techniques are introduced such that the probabilistic constraints are replaced by deterministic ones, while simultaneously preserving the convexity of the optimization model. Numerical results, obtained from a number of case studies, validate the effectiveness and reliability of the proposed approach. A number of comparative studies were also performed. The results confirm that the proposed design is able to produce more optimal flight paths and achieve enhanced computational performance than other chance-constrained optimization approaches investigated in this paper.**

***Index Terms*—Chance-constrained, trajectory optimization, unmanned vehicle, convexification, convex optimization, convex approximation.**

I. Introduction

THE problem of designing flight trajectories for unmanned vehicles has been an active research area for the last two decades due to its increasing applications such as regional detection, rescue, and formation flying [1], [2]. An effective path generator can be refer to an autonomous approach that is able to produce a feasible flight trajectory connecting the vehicle initial position and the pre-specified target position. During the planning phase, a number of vehicle-related or environment-related requirements might also need to be taken into account. These requirements are usually mission-dependent and modeled into different types of constraints.

In early years, geometric path planners have been considered as the primary way to plan the movement of the unmanned aerial vehicle (UAV). In the literature, a large amount of geometric-based path planners have been reported for dealing with the unmanned vehicle path generation problem [3]–[5], among which the pioneering research investigation in [3] and [4] is particularly noteworthy. The authors in [3] proposed a 2-D dynamic trajectory generation approach to guide the unmanned vehicle flying across a set of pre-defined waypoints. A 3-D Dubins curve-based trajectory generation method was designed in [4], wherein both the pitch angle and curvature radius constraints were considered. However, it was shown that this algorithm can only be effective for large distance flying scenarios. In [5], the authors developed a Bezier curve-based path generation method with the consideration of obstacle avoidance. Although most geometric-based motion planners are able to achieve the feasibility of the path, they can hardly take the optimality of the path into account. Furthermore, the consideration of mission constraints is often problematic and the way of handling constraints might need to be re-designed for different problems.

In recent years, path planners based on optimization theory have attracted significant attention [6], [7]. The key idea of this type of planner is to formulate and solve a trajectory optimization model for the unmanned vehicle. Different types of mission requirements can then be modeled as constraints and entailed to the optimization model. Relative work on developing or applying optimization-based path generation techniques can be found in the open literature [8]–[10]. In [8], a genetic algorithm-based path generator was developed. This method was then applied to optimize the 3-D flight path for military unmanned vehicles with no-fly zone constraints. Besides, Kim and Lee developed a heuristic path generator in [9], wherein particle swarm optimization was applied to determine the optimal motion of the manipulator. The authors in [10] proposed a multi-layer optimization-based trajectory generator in order to calculate the optimal movement of an autonomous space vehicle. Moreover, a data-driven indirect algorithm, along with a sensitivity analysis tool, was designed by Tang and Hauser in [11] so as to quickly generate optimal control solutions for different benchmark problems. Although the results presented in the aforementioned papers confirmed the effectiveness of these optimization-based approaches, one critical problem of implementing these path planners is that their computational or convergence performance can hardly meet the real-time capacity. This issue becomes more challenging when the nonlinearity of the vehicle dynamics becomes high or complex constraints such as the path and control chance constraints are required to be satisfied.

In [12] and [13], the authors applied a dynamic Dubins-

R. Chai, A. Tsourdos and A. Savvaris are with the School of Aerospace, Transport and Manufacturing, Cranfield University, UK, e-mail: (r.chai@cranfield.ac.uk), (a.tsourdos@cranfield.ac.uk), and (a.savvaris@cranfield.ac.uk).

S. Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, e-mail: (shuo.wang@ia.ac.cn).

Y. Xia and S. Chai are with the school of Automation, Beijing Institute of Technology, Beijing, China, e-mail: (xia_yuanqing@bit.edu.cn), and (chaisc97@163.com).

Helix method to produce constrained trajectories for unmanned vehicles. In these two works, the mission scenario was extended from 2-D space to 3-D space. Also, the minimum curvature radius as well as pitch angle constraints were achieved in real time. However, there are three main drawbacks of applying this technique. The first one is that there are some points with discontinuous curvature in the trajectory presented in [12] and [13]. This is because the algorithm divides the entire flight into several segments and designs the path for each segment separately. The points with discontinuous curvature may cause relatively large position-error for the path tracker. Secondly, based on our experiments, the optimality of the flight path obtained via the previously-developed method is not as comparable as the optimization results for a variety of mission cases. Besides, this previously-designed motion planner does not have the capability in dealing with control chance constraints and probabilistic collision avoidance constraints.

It is important to note that usually in many real-world UAV flight circumstances, the constraint information might not be exactly known and could be perturbed by some uncertainties [14]. More precisely, uncharacterized temperature/weather influences, errors in localization and mapping could be some examples of uncertainty in the maximum attainable actuation level constraints as well as the obstacle avoidance constraints. Consequently, it is necessary to incorporate the chance constraints in the design of flight trajectories and new chance-constrained unmanned vehicle trajectory planning methods are highly demanded to serve as effective tools to explore the solutions.

In this work, we are interested in designing a trajectory planning algorithm that can: 1). produce the flight trajectory of a fixed-wing UAV with minimized path length; 2). simultaneously satisfy deterministic path constraints, probabilistic control and obstacle avoidance constraints; 3). achieve a real-time performance. To fulfill these goals, the proposed strategy incorporates convex optimization and convex approximation of chance constraints. Recently, there are many representative works that have been reported in the literature in terms of applying convex optimization methods for aerospace guidance and control problems [15]–[18]. For example, in [15] the authors developed a sequential convex programming method in order to convexify the nonlinear planetary-entry problem and optimize the flight trajectory. In their follow-up research, the developed convex trajectory optimization method was further enhanced and extended to other applications such as the low-thrust Earth-to-Mars orbital transfer problem [16], [17]. In addition, the authors considered an asteroid landing problem in [19], wherein a convex optimization method was applied to explore the minimum landing error and the time-optimal trajectories, respectively.

Apart from the aforementioned works, a large body of publications on UAV path planning problems using convex optimization can also be found in the literature [20], [21]. Specifically, a convex optimization-oriented path planning approach was proposed in [20], wherein flight experiments were executed on a quad-rotor to assess and verify the performance of the proposed method. In [21], a sequential convex optimization approach , modified by a line search update process, was established to address a UAV and unmanned ground vehicle (UGV) rendezvous problem. Based on the reported simulation results, enhanced convergence performance as well as real-time capability can be successfully achieved.

Nevertheless, the main motivation for the use of convex optimization is that if a problem can be formulated as a convex program, then it can be addressed in polynomial time [22]. However, existing designs cannot be directly applied to solve the problem considered in this study due to the existence of chance constraints. This is because the probabilistic functions are usually not deterministic. Note that in [23], the authors developed a chance-constrained optimization strategy and successfully applied it to address a similar chance-constrained UAV path planning problem. However, two problems still remain open. For instance, this method applied a nonconvex function to approximate the probabilistic constraints, thereby resulting in non-convexity in the transformed trajectory optimization model. As a result, it is likely to introduce a large amount of computational burdens during the optimization process. Moreover, in terms of the problem formulation, no uncertain obstacles were taken into account, thereby making the obtained results less attractive for practical applications. Hence, to address the remaining problems, we make an attempt to design a convex chance constraint approximation strategy in this work. This convex approximation strategy has the capability of transcribing the probabilistic control and obstacle avoidance constraints into a deterministic and convex version, which in turn preserves the convexity of the optimization model. Subsequently, this approximation method is embedded in the convexified path planning framework, thus making it a potentially suitable alternative for solving the chance-constrained UAV trajectory planning task.

In summary, the main contributions of this paper are threefold. Firstly, the chance-constrained UAV trajectory planning problem considered in [23] is further extended so as to take the uncertain effects caused by both the actuator and obstacles into account. Compared to the previous version, the extended problem formulation tends to be more representative and integrated. Another main contribution lies in the efforts on designing the convex approximation strategy such that the extended nonlinear UAV chance-constrained trajectory planning model can be reformulated into a deterministic and convex one. Thirdly, a number of comparative case studies are carried out in order to verify the effectiveness of the proposed fast chance-constrained trajectory generation method as well as its enhanced computational performance.

The structure of this paper is outlined as follows. Sec II illustrates the mathematical formulation of the considered trajectory planning problem. Sec III and Sec IV present, respectively, the convexified version of the unmanned vehicle trajectory optimization model as well as the convex chance constraint approximation method. Numerical results including the obtained optimal flight trajectories and a number of comparative case studies are demonstrated in Sec V. Finally, this paper is concluded in Sec VI.

II. Trajectory Planning Formulation

In this section, the unmanned vehicle trajectory planning model studied in this research is outlined. More precisely, Sec II.A presents the nonlinear system equations of the unmanned vehicle. Subsequently, in Sec II.B and Sec II.C, the geometric constraints imposed on the model, along with the relationship between geometric constraints and vehicle actual constraints, will be introduced. Sec II.D and Sec II.E illustrate, respectively, the control chance constraints and the probabilistic collision avoidance constraints. Based on the system model and mission constraints, a path length-optimal unmanned vehicle trajectory optimization formulation is established in Sec II.F.

A. Unmanned Vehicle System Equations

The equations of motion of a fixed-wing unmanned vehicle can be written as [4], [12]:

$$\begin{cases} \frac{dp_x}{ds} = \cos\gamma(s)\cos\phi(s) \\ \frac{dp_y}{ds} = \cos\gamma(s)\sin\phi(s) \\ \frac{dp_z}{ds} = \sin\gamma(s) \\ \frac{d\phi}{ds} = \mu_1 \\ \frac{d\gamma}{ds} = \mu_2 \end{cases} \tag{1}$$

where $p = (p_x, p_y, p_z) \in \mathbb{R}^3$ denotes the pose of the unmanned vehicle in the 3-D environment. s stands for the curvilinear abscissa along the path. In the last two subequations, ϕ and γ are, respectively, the heading and pitch angles. μ_1 and μ_2 represent the derivative value with regard to the heading and pitch angles, and they will play the role of control variables in the trajectory optimization model to be defined in Sec II.F.

Different from most previous studies [4], [5], [24], in this paper a variety of mission constraints are required to be taken into account during the flying. They are classified into geometric constraints, control chance constraints, and probabilistic collision avoidance constraints.

B. Geometric Constraints

One of the geometric constraints is the unmanned vehicle's minimum curvature radius. The aim for imposing a constraint on the curvature radius is to smoother the system state and control profiles. Besides, certain requirement should be given to the pitch angle for safety reasons in the 3-D space. This is achieved by restricting the pitch angle in a certain range. That is, the geometric constraints to be considered during the flight are:

i) The curvature radius $R(s)$ should satisfy $|R(s)| > R^{\min}$.
ii) The pitch angle should satisfy $\gamma^{\min} \leq \gamma \leq \gamma^{\max}$.

Note that in i), $R(s)$ is computed by:

$$R(s) = 1/\sqrt{\mu_1^2(s)\cos^2\gamma(s) + \mu_2^2(s)} \tag{2}$$

C. Relationship Between Geometric Constraints and Vehicle Actual Constraints

Different from the model described by Eq.(1), other forms of UAV dynamics may exist in the literature [7], [21]. It is worth noting that there exist certain links between the model given by (1) and those complicated nonlinear fixed-wing UAV dynamics widely applied in other works (e.g., [7] and [21]). This will become more apparent through the analysis of relationships between the geometric constraints defined in Sec II.B and the vehicle actual ones (e.g., the structural, propulsive, and envelope constraints). Specifically, to begin with, we recall the dynamics model used in [7] and [21]:

$$\begin{aligned} \dot{V} &= \frac{(T-D)}{m} - g\sin\gamma \\ \dot{\phi} &= \frac{L\sin\sigma}{mV\cos\gamma} \\ \dot{\gamma} &= \frac{L\cos\sigma - mg\cos\gamma}{mV} \end{aligned} \tag{3}$$

where $\dot{V}$, $\dot{\phi}$ and $\dot{\gamma}$ are the derivatives of vehicle velocity, heading angle and pitch angle with respect to time t. Other parameters such as T, m, g, and σ stand for the thrust, vehicle mass, gravity and roll angle, respectively. L and D are the aerodynamic lift and drag forces and their expressions are:

$$\begin{aligned} L &= \frac{\rho V^2 S C_L}{2} \\ D &= \frac{\rho V^2 S C_D}{2} \end{aligned} \tag{4}$$

in which ρ and S represent, respectively, the atmospheric density and the reference area of the vehicle. C_L is the lift coefficient, while the drag coefficient C_D can be modeled as a function of C_L.

Usually, to satisfy the vehicle propulsive and envelop limitations, constraints can be imposed on the variable σ and T. On the other hand, to satisfy other physical limitations such as the structural constraints, one may limit the value of the vertical load factor n_z, which can be written as:

$$n_z = \frac{L+T}{mg} \tag{5}$$

According to Eq.(4) and Eq.(5), two curvature radius components (e.g., the horizontal and vertical terms) can be defined in the form of:

$$\begin{aligned} R_h &= \frac{V}{\cos\gamma\dot{\phi}} = \frac{V^2}{n_z g\sin\sigma} \\ R_v &= \frac{V}{\dot{\gamma}} = \frac{V^2}{n_z g\cos\sigma - g\cos\gamma} \end{aligned} \tag{6}$$

From Eqs.(3)-(6), it is possible to find functional relationships for V, γ, R_h, and R_v such that:

$$\begin{aligned} V^2 &= F_V(n_z, T) \\ \gamma &= F_\gamma(n_z, T) \\ R_h &= F_{R_h}(n_z, T, \sigma) \\ R_v &= F_{R_v}(n_z, T, \sigma) \end{aligned} \tag{7}$$

Eq.(7) can be rewritten in a more compact mapping form:

$$\begin{pmatrix} V^2 \\ \gamma \\ R_h \\ R_v \end{pmatrix} = F(n_z, T, \sigma) \tag{8}$$

The above equation implies that we are able to map a domain regulated by the UAV physical variables (e.g., n_z, T, and σ) to another domain regulated by the geometric variables.

Remark 1. Note that if we compare the definition of $R(s)$ given by Eq.(2) with Eq.(6), it is easy to find the following relationship:

$$\begin{aligned} R(s) &= 1/\sqrt{\mu_1^2\cos^2\gamma+\mu_2^2} \\ &= 1/\sqrt{(\frac{\dot{\phi}}{V})^2\cos^2\gamma+(\frac{\dot{\gamma}}{V})^2} \\ &= 1/\sqrt{\frac{1}{R_h^2}+\frac{1}{R_v^2}} \end{aligned} \tag{9}$$

Here, $V=\frac{ds}{dt}$, $\mu_1=\frac{d\phi}{ds}$, and $\mu_2=\frac{d\gamma}{ds}$.

D. Control Chance Constraints

For the trajectory planning task considered in this work, the maximum and minimum attainable control actuation levels of the unmanned vehicle are not assumed to be deterministic. Alternatively, it is supposed that the control constraints will be affected by some uncertain parameters (e.g. ξ_{μ_1} and ξ_{μ_2}). That is,

$$\begin{cases} Pr\{|\mu_1+\xi_{\mu_1}|\leq\mu_1^{\max}\}\geq\epsilon_{\mu_1} & \text{(10a)} \\ Pr\{|\mu_2+\xi_{\mu_2}|\leq\mu_2^{\max}\}\geq\epsilon_{\mu_2} & \text{(10b)} \end{cases}$$

where ξ_{μ_1} and ξ_{μ_2} are two uncertain parameters with known probability density functions (PDFs). ϵ_{μ_1} and ϵ_{μ_2} are the acceptable probabilities of occurrence. $\mu_1^{\max}$ and $\mu_2^{\max}$ represent the maximum allowable values for μ_1 and μ_2, respectively.

E. Probabilistic Collision Avoidance Constraints

Collision avoidance path constraints are also imposed to the planning model. In this paper, the obstacles can be modeled as polygons or polyhedrons. Take the polyhedron as an example, a safe region can be defined as:

$$\Lambda=\{p\in\mathbb{R}^3:\bigwedge_{n=1}^{N_o}\bigvee_{m=1}^{M_j}a_{mn}^Tp+b_{mn}+\xi_{mn}>0\} \tag{11}$$

where $p=(p_x,p_y,p_z)$. N_o and M_j denote the number of obstacles and the number of faces of the jth obstacle, respectively. The characters $\bigwedge$ and $\bigvee$ represent the logical "and" and the logical "or" relations. $a_{mn}\in\mathbb{R}^3$ and $b_{mn}\in\mathbb{R}$, while $\xi_{ij}\in\mathbb{R}$ is the uncertain variable. The region occupied by the set Λ consists of areas outside all the obstacles existing in the environment. The pose of the unmanned vehicle must stay in the safe region so as to avoid any potential collisions. Due to the existence of uncertainty in the safe set Λ, we thereby model the collision avoidance constraint in a probabilistic inequality which is in the form of:

$$Pr\{p\in\Lambda\}\geq\epsilon_o \tag{12}$$

F. Objective and Optimization Model

The overall aim of the unmanned vehicle trajectory planning problem is to design a smooth flight path connecting the two poses (e.g., the initial pose $P_0(p_x,p_y,p_z)$ and the final pose $P_f(p_x,p_y,p_z)$) such that the flight path s of the vehicle can be optimized while simultaneously considering the geometric constraints as well as different chance constraints. Consequently, a path-length-optimal nonlinear chance-constrained unmanned vehicle trajectory optimization formulation can be established as follows:

$$\begin{cases} \underset{z(s)}{\text{minimize}} & J=\int_0^s ds \\ \text{s.t.} & \frac{dp_x}{ds}=\cos\gamma(s)\cos\phi(s) \\ & \frac{dp_y}{ds}=\cos\gamma(s)\sin\phi(s) \\ & \frac{dp_z}{ds}=\sin\gamma(s) \\ & \frac{d\phi}{ds}=\mu_1(s) \\ & \frac{d\gamma}{ds}=\mu_2(s) \\ & x(s_0)=x_0,\ x(s_f)=x_f \\ & |R(s)|>R^{\min} \\ & \gamma^{\min}\leq\gamma(s)\leq\gamma^{\max} \\ & |\mu_1|\leq\mu_1^{\max},|\mu_2|\leq\mu_2^{\max} \\ & Pr\{|\mu_1+\xi_{\mu_1}|\leq\mu_1^{\max}\}\geq\epsilon_{\mu_1} \\ & Pr\{|\mu_2+\xi_{\mu_2}|\leq\mu_2^{\max}\}\geq\epsilon_{\mu_2} \\ & Pr\{p\in\Lambda\}\geq\epsilon_o \end{cases} \tag{13}$$

In Eq.(13), $s\in(s_0,s_f)$ and $z(s)=(x(s),u(s))$, in which the system state variables are defined as $x(s)$=$[p_x(s),p_y(s),p_z(s),\phi(s),\gamma(s)]^T$. The boundary conditions are given by $x(s_0)=x_0$ and $x(s_f)=x_f$, respectively. Here, the boundary variables are denoted as x_0 and x_f. The control variable $u(s)$ consists of $\mu_1(s)$ and $\mu_2(s)$. That is, $u(s)=[\mu_1(s),\mu_2(s)]^T$.

III. Convex-Programming-based Trajectory Planning Approach

Currently, there are many numerical algorithms (e.g., the pseudospectral method [25], and the multiple shooting method [26]) that can be applied to address standard nonlinear optimal control problems. However, one critical issue of applying these techniques is that they tend to have low solution-finding efficiency due to the requirement of solving nonlinear programming problems directly. To effectively deal with this issue, the fast trajectory generation approach reported in this paper suggests to transcribe the original formulation into a convex program, thereby avoiding time-consuming nonlinear optimization process and improving the global convergence.

A. Convexification of System Equations and Constraints

To transform the optimization model, the first step is to linearize the system dynamics. Eq.(1) can be abbreviated as $\frac{dx}{ds}=f(x,u)$. Here, $f(x,u)$ has the following form:

$$f(x,u)=\begin{bmatrix}\cos\gamma(s)\cos\phi(s)\\ \cos\gamma(s)\sin\phi(s)\\ \sin\gamma(s)\\ \mu_1(s)\\ \mu_2(s)\end{bmatrix} \tag{14}$$

A linear unmanned vehicle system model is then obtained by differentiating $f(x,u)$ with respect to x and u, which can be written as:

$$\frac{dx}{ds}=A(x^r,u^r)x+B(x^r,u^r)u+c(x^r,u^r) \tag{15}$$

where (x^r,u^r) is the reference state and control pair and it satisfies $\frac{dx^r}{ds}=f(x^r,u^r)$. $c(x^r,u^r)$=$f(x^r,u^r)-A(x^r,u^r)x^r-B(x^r,u^r)u^r$.

In Eq.(15), $A(x^r, u^r)$ and $B(x^r, u^r)$ are, respectively, the partial derivative of $f(\cdot,\cdot)$ with respect to x and u at the operating point (x^r, u^r). That is,

$$\begin{aligned} A(x^r,u^r) &= \frac{\partial f(x,u)}{\partial x}\Big|_{\substack{x=x^r\\u=u^r}} \\ B(x^r,u^r) &= \frac{\partial f(x,u)}{\partial u}\Big|_{\substack{x=x^r\\u=u^r}} \end{aligned} \tag{16}$$

For the unmanned vehicle system given by Eq.(1), $A(x^r, u^r)$ and $B(x^r, u^r)$ can be written as:

$$\begin{aligned} &A(x^r,u^r) \\ &= \begin{bmatrix} 0 & 0 & 0 & \sin\phi(s)\cos\gamma(s) & -\cos\phi(s)\sin\gamma(s) \\ 0 & 0 & 0 & \cos\phi(s)\cos\gamma(s) & -\sin\phi(s)\sin\gamma(s) \\ 0 & 0 & 0 & 0 & \cos\gamma(s) \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{\substack{x=x^r\\u=u^r}} \end{aligned} \tag{17}$$

$$B(x^r,u^r) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}_{\substack{x=x^r\\u=u^r}} \tag{18}$$

Until now, in the original optimization model (13), the nonlinear dynamics have been transcribed into a linear form. The next step is to transcribe the nonconvex path constrain (e.g., the curvature radius constraint). The transformation method is similar to that of the dynamics. Specifically, $R(x, u)$ can be convexified as:

$$\begin{aligned} R(x,u) = R(x^r,u^r) &+ R_x(x^r,u^r)(x-x^r) \\ &+ R_u(x^r,u^r)(u-u^r) \end{aligned} \tag{19}$$

where R_x and R_u are partial derivatives of R with respect to x and u. This transformation is easy to implement and generally applicable to a variety of inequality constraints.

B. Convex Trajectory Optimization Model

To solve the problem, discretization techniques should be applied so as to convert the continuous-time problem into a static convex program. The discretization method used in this paper is the pseudospectral method (PS). Detailed formulations and convergence properties of the PS method can be refer to [25], [27] and are omitted here for space reasons.

It should be noted that the combination of pseudospectral method and convex optimization in solving aerospace optimal control problems can be found in a number of previously-published works such as [28] and [29]. Based on the reported results, it was verified that such a hybrid strategy has the capability of producing more accurate results in comparison to other standard transcriptions. Moreover, its computational time could be maintained in a reasonable level. Therefore, we apply this hybrid strategy in this paper to construct the convex trajectory optimization model for the considered problem.

By applying the PS method, the linearized dynamics can be further written as:

$$\begin{aligned} \sum_{i=0}^{N} D_{ik}x_i = \frac{s_f - s_0}{2}&(A(x_k^r,u_k^r)x_k \\ &+ B(x_k^r,u_k^r)u_k + c(x_k^r,u_k^r)) \end{aligned} \tag{20}$$

where s_0 and s_f are the initial and terminal s values. $k = 1, ..., N$, and D_{ik} is the differentiation matrix. With the convex transformation and the discretization of system dynamics and path constraints, a convexified unmanned vehicle trajectory planning model is established:

$$\left\{\begin{array}{ll} \underset{z_k}{\text{minimize}} & J = s_N \\ \text{s.t.} & \forall k \in \{1,...,N\} \\ & \sum_{i=0}^{N} D_{ik}x_i = \frac{s_f-s_0}{2}(A(x_k^r,u_k^r)x_k \\ & \qquad + B(x_k^r,u_k^r)u_k + c(x_k^r,u_k^r)) \\ & |R(x_k^r,u_k^r) + R_x(x_k^r,u_k^r)(x_k - x_k^r) \\ & \qquad + R_u(x_k^r,u_k^r)(u_k-u_k^r)| > R^{\min} \\ & x(s_0) = x_0,\ x(s_N) = x_f \\ & x^{\min} \le x_k \le x^{\max} \\ & |u_k| \le u^{\max} \\ & |x_k - x_k^r| \le \delta_x \\ & |u_k - u_k^r| \le \delta_u \end{array}\right. \tag{21}$$

Eq.(21) is a typical convex program, as the mission objective, equality and inequality constraints are all convex functions. In Eq.(21), $x(s_k)$ and $u(s_k)$ are abbreviated as x_k and u_k, respectively.

Linear approximation is used to convexify the nonlinear dynamics and constraints. However, it is well known that the linearization is only accurate at the vicinity of the reference trajectory. Therefore, in Eq.(21), the last two trust-region constraints are introduced (e.g., $|x_k - x_k^r| \le \delta_x$ and $|u_k - u_k^r| \le \delta_u$) to restrict the variation of the state and control profiles, thus enhancing the effectiveness of the linearization process. Note that based on our experiments, imposing these simple trust region constraints with fixed radius values (δ_x and δ_x) can be effective for addressing the considered problem. However, it is undeniable that trust region constraints with varying radius values can also be modeled and the optimization process is likely to benefit from this modification. For example, motivated by a related work [18], an alternative convex program can be constructed by introducing quadratic trust region constraints:

$$\left\{\begin{array}{ll} \underset{z_k^{'}}{\min} & J = s_N + w_x\|r_x\|_2 + w_u\|r_u\|_2 \\ \text{s.t.} & \forall k \in \{1,...,N\} \\ & \sum_{i=0}^{N} D_{ik}x_i = \frac{s_f-s_0}{2}(A(x_k^r,u_k^r)x_k \\ & \qquad + B(x_k^r,u_k^r)u_k + c(x_k^r,u_k^r)) \\ & |R(x_k^r,u_k^r) + R_x(x_k^r,u_k^r)(x_k - x_k^r) \\ & \qquad + R_u(x_k^r,u_k^r)(u_k-u_k^r)| > R^{\min} \\ & x(s_0) = x_0,\ x(s_N) = x_f \\ & x^{\min} \le x_k \le x^{\max} \\ & |u_k| \le u^{\max} \\ & [x_k - x_k^r]^T[x_k - x_k^r] \le r_{x_k} \\ & [u_k - u_k^r]^T[u_k - u_k^r] \le r_{u_k} \end{array}\right. \tag{22}$$

where w_x and w_u stand for positive weighting parameters. $r_x = [r_{x_1}, r_{x_2}, ...r_{x_N}]^T$ and $r_u = [r_{u_1}, r_{u_2}, ...r_{u_N}]^T$, respectively. $\|\cdot\|_2$ denotes the 2-norm. In this formulation, the optimization variable is redefined as $z_k^{'}$=$(x_k, u_k, r_{x_k}, r_{u_k})$. Besides, the trust region radius is adjusted during the iteration, thereby further enhancing the convergence and robustness of the optimization process.

The solution to the optimization problem (21) is usually obtained by iteratively addressing a sequence of convex optimization problems. Due to the implementation of linear approximations, one critical issue called "artificial infeasibility" identified in recent years may occur during the solution finding process [18], [21], [30]. Specifically, this problem will appear when a solution to the original nonconvex problem does not satisfy the linearized dynamics equations and path constraints in (21) and vice versa. Hence, certain treatments should be performed so as to reduce or remove these inconsistencies caused by the linearization errors. Otherwise, the optimization process may suffer from slow convergence or even fail to converge.

To overcome this issue, the works presented by Acikmese [30] and Wang [18], [21] are of particular importance. For instance, in [30], the authors eliminated the artificial infeasibility by adding a virtual control term in the linearized dynamics. Since no constraint was imposed on the virtual control variable, any feasible state values could be reached. Besides, in [18] the author suggested a line-search strategy in order to reduce the artificial infeasibility and enhance the convergence of the sequential optimization process. In their follow-up research [21], this line-search approach was successfully applied to address a UAV and a UGV rendezvous problem. It was verified that by applying the proposed line-search method, the error caused by linear approximation could be effectively reduced and the convergence of the successive optimization could be improved.

It should be noted that Eq.(21) and Eq.(22) are formulated without considering the chance constraints given by Eq.(10) and Eq.(12). In the next section, a convex chance constraint approximation method will be introduced and applied in order to formulate the fast chance-constrained trajectory planning model.

IV. Deterministic Chance-Constrained Trajectory Planning Formulation

In this section, we are aiming to find a proper chance constraint handling strategy in order to introduce the probabilistic control and collision avoidance constraints (10) to the convex formulation (21) or (22). The designed approach should be easy to implement and not damage the convexity of the optimization model. Actually, for the control chance constraints, one potential way is to simply apply the min-max strategy which is commonly-used in robust optimization tasks [31]. However, this strategy does not allow any constraint violations, thereby restricting the searching space of the optimization process and degrading the computational performance as well as the solution optimality. Alternatively, a convex chance constraint approximation strategy is suggested in this study, which will be detailed in the following subsection.

A. Convex Approximation of Control Chance Constraints

Let us consider the control chance constraints (10) in a general form $Pr\{g(u,\xi)>0\}\leq\epsilon$. If we treat the inequality $g(\cdot,\cdot)>0$ as an event, then the probability of occurrence can be calculated via:

$$Pr\{g(u,\xi)>0\}=E(H(g(u,\xi)))=\int_{\Omega}H(g(u,\xi))L(\xi)d\xi \tag{23}$$

where $E(\cdot)$ is the expectation function. $L(\xi)$ is the PDF of ξ. $H(\cdot)$ is the Heaviside function, which has the form of:

$$H(g(u,\xi))=\left\{\begin{array}{lll}1 & \text{if} & g(u,\xi)>0\\ 0 & \text{if} & g(u,\xi)\leq 0\end{array}\right. \tag{24}$$

The core idea of the convex chance constraint approximation strategy is to use a convex function $\Psi(g(u,\xi))$ to replace $H(g(u,\xi))$ in Eq.(23). The convex function suggested in this study has a simple form:

$$\Psi(g(u,\xi))=(g(u,\xi)+1)^{+} \tag{25}$$

where the superscript "+" stands for the max operation (e.g., $\max\{g(u,\xi)+1,0\}$). The next theorem indicates that by using Eq.(23), one can obtain an upper approximation of the original chance constraint.

Theorem 1. *Given that the convex approximation function in the form of Eq.(25), if an integral quantity can be defined in the form of*

$$V_c(u)=\int_{\Omega}\Psi(g(u,\xi))L(\xi)d\xi \tag{26}$$

then, V_c is an upper bound of the control chance constraint $Pr\{g(u,\xi)>0\}$.

Proof: According to the definition of $\Psi(g(u,\xi))$, it is obvious that $\Psi\geq 0$ for any $g(u,\xi)$ and $\Psi>1$ for $g(u,\xi)>0$. Hence, $\forall\xi\in\Omega$, we have

$$\begin{aligned}V_c(u)\geq&\int\limits_{\{\xi\in\Omega,g(u,\xi)>0\}}\Psi(g(u,\xi))L(\xi)d\xi\\ \geq&\int\limits_{\{\xi\in\Omega,g(u,\xi)>0\}}L(\xi)d\xi\\ =&Pr\{\xi\in\Omega,g(u,\xi)>0\}\end{aligned}$$

which completes the proof. ■

Theorem 1 can be used to create a relaxation of the control chance constraint. For example,

$$V_c(u)\leq\epsilon \tag{27}$$

If Eq.(27) is satisfied, the original chance constraint will be satisfied as well. To calculate the integral value in Eq.(26), the Markov chain Monte-Carlo (MCMC) sampling strategy is utilized as suggested in [23]. The motivation for the use of this method mainly relies on its simplicity in application. A set of random parameters are firstly generated $\{\xi^q\}_{q=1}^{N_\xi}$. Subsequently, $V_c(u)$ can be computed by:

$$V_c(u)\approx\frac{1}{N_\xi}\sum_{q=1}^{N_\xi}\Psi(g(u,\xi^q)) \tag{28}$$

As a result, the chance-constrained unmanned vehicle trajectory planning model is formulated as:

$$\begin{cases} \underset{z_k}{\text{minimize}} & J = s_N \\ \text{s.t.} & \forall k \in \{1, ..., N\} \\ & \sum_{i=0}^{N} D_{ik} x_i = \frac{s_f - s_0}{2}(A(x_k^r, u_k^r)x_k \\ & \qquad + B(x_k^r, u_k^r)u_k + c(x_k^r, u_k^r)) \\ & |R(x_k^r, u_k^r) + R_x(x_k^r, u_k^r)(x_k - x_k^r) \\ & \qquad + R_u(x_k^r, u_k^r)(u_k - u_k^r)| > R^{\min} \\ & \frac{1}{N_\xi}\sum_{q=1}^{N_\xi} \Psi(g(u_k, \xi^q)) \leq \epsilon \\ & x(s_0) = x_0, \; x(s_N) = x_f \\ & x^{\min} \leq x_k \leq x^{\max} \\ & |u_k| \leq u^{\max} \\ & |x_k - x_k^r| \leq \delta_x \\ & |u_k - u_k^r| \leq \delta_u \end{cases} \tag{29}$$

The next theorem indicates that Eq.(27) is a convex constraint and the constructed chance-constrained formulation (29) is a convex optimization problem.

Theorem 2. *The chance constraint approximation inequality (27) is a convex constraint and the chance-constrained optimization model given by Eq.(29) is a convex program.*

Proof: From the definition of the control constraints, it is clear that $g(u, \xi)$ is convex in u for any fixed $\xi \in \Omega$. In addition, function $\Psi(g(u, \xi))$ is convex and non-decreasing. Hence, $\Psi(g(u, \xi))$ is convex in u for $\forall \xi \in \Omega$.

As $L(\xi)$ is a probability measure on Ω and $L(\xi) \geq 0$, the integral $\int_\Omega \Psi(g(u, \xi))L(\xi)d\xi$ is also convex. The convexity of inequality (27), together with the convexity of the optimization model (21), confirms the convexity of the optimization model (29). This completes the proof. ■

B. Convex Approximation of Probabilistic Collision Avoidance Constraints

In order to solve the optimization model with the consideration of probabilistic obstacle avoidance constraints, we reformulate the inequality (12) into a tractable form. This is achieved by performing two steps. Firstly, we apply the big-M technique to transform the logic $\bigvee$ appeared in the safe region (11) to logic $\bigwedge$. Subsequently, Boole's inequality-based decomposition is used to split the joint constraint into single chance constraints, thus preserving the convexity of the optimization model and easying the solution-finding process.

Consider the disjunction shown in Eq.(11), its satisfaction is equivalent to

$$\begin{aligned} & \bigvee_{m=1}^{M_j} a_{mn}^T p + b_{mn} + \xi_{mn} > 0 \\ \Leftrightarrow & \bigwedge_{m=1}^{M_j} a_{mn}^T p + b_{mn} + \xi_{mn} + \mathbf{M} z_{mn} > 0 \\ & z_{mn} \in \{0, 1\}, \mathbf{M} > 0 \end{aligned} \tag{30}$$

where z_{mn} is a binary integer variable. There must exist a zero element among mth-indexed z, thereby resulting in an additional constraint $\sum_{m=1}^{M_j} z_{mn} < M_j$. $\mathbf{M}$ denotes a positive constant. Based on this big-M transformation, we can rewrite constraint (12) as:

$$Pr\{\bigwedge_{n=1}^{N_o}\bigwedge_{m=1}^{M_j} a_{mn}^T p + b_{mn} + \xi_{mn} + \mathbf{M} z_{mn} > 0\} \geq \epsilon_o \tag{31}$$

For a series of events $\{E_m\}$, Boole's inequality can be applied to split the joint in (31). Then a conservative yet easy form of the obstacle avoidance chance constraint can be obtained:

$$\begin{aligned} & \text{For} \quad \forall n \in \{1, ..., N_o\}, \forall m \in \{1, ..., M_j\} \\ & Pr\{a_{mn}^T p + b_{mn} + \xi_{mn} + \mathbf{M} z_{mn} > 0\} \geq \epsilon_{mn} \\ & \sum_{n=1}^{N_o}\sum_{m=1}^{M_j} \epsilon_{mn} \geq \epsilon_o \\ & z_{mn} \in \{0, 1\}, \mathbf{M} > 0 \end{aligned} \tag{32}$$

Until now, the original probabilistic collision avoidance constraint has been reformulated into single chance constraints. It is obvious that the convex transformation method developed in the previous subsection can be directly applied to approximate the probabilistic term in (32). Similar with (29), we are able to analogically construct the following mixed-integer convex program:

$$\begin{cases} \underset{z_k, z_{mn}}{\text{minimize}} & J = s_N \\ \text{s.t.} & \forall k \in \{1, ..., N\}, \forall n \in \{1, ..., N_o\}, \\ & \forall m \in \{1, ..., M_j\} \\ & \sum_{i=0}^{N} D_{ik} x_i = \frac{s_f - s_0}{2}(A(x_k^r, u_k^r)x_k \\ & \qquad + B(x_k^r, u_k^r)u_k + c(x_k^r, u_k^r)) \\ & |R(x_k^r, u_k^r) + R_x(x_k^r, u_k^r)(x_k - x_k^r) \\ & \qquad + R_u(x_k^r, u_k^r)(u_k - u_k^r)| > R^{\min} \\ & \frac{1}{N_\xi}\sum_{q=1}^{N_\xi} \Psi(g(u_k, \xi^q)) \leq \epsilon \\ & \frac{1}{N_\xi}\sum_{q=1}^{N_\xi} \Psi(a_{mn}^T p_k + b_{mn} + \xi_{mn}^q + \mathbf{M} z_{mn}) \leq \epsilon_{mn} \\ & \sum_{n=1}^{N_o}\sum_{m=1}^{M_j} \epsilon_{mn} \geq \epsilon_o \\ & z_{mn} \in \{0, 1\}, \mathbf{M} > 0 \\ & x(s_0) = x_0, \; x(s_N) = x_f \\ & x^{\min} \leq x_k \leq x^{\max} \\ & |u_k| \leq u^{\max} \\ & |x_k - x_k^r| \leq \delta_x \\ & |u_k - u_k^r| \leq \delta_u \end{cases} \tag{33}$$

where p_k represents the current 3-D position of the unmanned vehicle at node k. The convexity of problem (33) is a direct result from the definition of polyhedral obstacles, Boole's inequality-based decomposition, and Theorem 2. Different from the formulation given by Eq.(29), the branch-and-bound strategy can be applied to address this mixed-integer convex program [32].

Remark 2. Similar to that of Eq.(22), by introducing w_x and w_u (the weighting parameters), an alternative mixed-integer convex optimization formulation with varying trust-region radius r_x and r_u can be formulated as:

$$\begin{cases} \underset{z_k^{'}, z_{mn}}{\text{minimize}} & J = s_N + w_x\|r_x\|_2 + w_u\|r_u\|_2 \\ \text{s.t.} & \forall k \in \{1, ..., N\}, \forall n \in \{1, ..., N_o\}, \\ & \forall m \in \{1, ..., M_j\} \\ & \sum_{i=0}^{N} D_{ik}x_i = \frac{s_f - s_0}{2}(A(x_k^r, u_k^r)x_k \\ & \qquad + B(x_k^r, u_k^r)u_k + c(x_k^r, u_k^r)) \\ & |R(x_k^r, u_k^r) + R_x(x_k^r, u_k^r)(x_k - x_k^r) \\ & \qquad + R_u(x_k^r, u_k^r)(u_k - u_k^r)| > R^{\min} \\ & \frac{1}{N_\xi}\sum_{q=1}^{N_\xi}\Psi(g(u_k, \xi^q)) \leq \epsilon \\ & \frac{1}{N_\xi}\sum_{q=1}^{N_\xi}\Psi(a_{mn}^T p_k + b_{mn} + \xi_{mn}^q + \mathbf{M}z_{mn}) \leq \epsilon_{mn} \\ & \sum_{n=1}^{N_o}\sum_{m=1}^{M_j}\epsilon_{mn} \geq \epsilon_o \\ & z_{mn} \in \{0, 1\}, \mathbf{M} > 0 \\ & x(s_0) = x_0, \ x(s_N) = x_f \\ & x^{\min} \leq x_k \leq x^{\max} \\ & |u_k| \leq u^{\max} \\ & [x_k - x_k^r]^T[x_k - x_k^r] \leq r_{x_k} \\ & [u_k - u_k^r]^T[u_k - u_k^r] \leq r_{u_k} \end{cases} \tag{34}$$

Here, $z_k^{'}$=$(x_k, u_k, r_{x_k}, r_{u_k})$. In Eq.(34), varying-radius trust region constraints are considered. This can potentially enhance the convergence and robustness of the optimization process.

C. Overall Algorithm Framework

The overall algorithm framework is summarised in Algorithm 1.

Algorithm 1 Chance-constrained trajectory planning process

```
Input: Algorithm/mission-dependent parameters: $N$, $N_\xi$, $x^r$, $u^r$, $x_0$, $x_f$, and $R^{\min}$;
/*Main Iteration*/
Step 1: Perform the convexification process with respect to
        system dynamics and constraints;
Step 2: Generate the differential matrix $D_{ik}$ and discretize
        the convexified problem via PS;
Step 3: Generate the random parameter $\{\xi^q, \xi_{ij}^q\}_{q=1}^N$;
Step 4: Formulate the chance-constrained convex trajectory
        planning model via Eq.(33);
Step 5: Address the chance-constrained convex optimization
        model;
Step 6: Obtain the solution pair and update the
        reference pair $(x^{r+1}, u^{r+1})$;
Step 7: Check the stopping condition for convergence:
            $\max|x^{r+1} - x^r| \leq \epsilon_x$,
            $\max|u^{r+1} - u^r| \leq \epsilon_u$;
Step 8: If the stopping conditions can be satisfied, then
        output the solution. Otherwise, perform a line search
        process and go back to Step 1.
Output: The optimal trajectory pair $(x^*, u^*)$;
```

Note that in Algorithm 1, Step 6 requires the solution to the mixed-integer convex optimization model given by Eq.(33). Therefore, the sequential mixed-integer convex programming [32] is selected as the main optimization method to calculate the solution.

Motivated by related works [18], [21], a line search strategy can be established in Step 8 of Algorithm 1 to alleviate the artificial infeasibility issue for the considered problem. For the sake of completeness, this strategy is detailed in Algorithm 2.

Algorithm 2 A line-search process

```
Input: $z^r = (x^r, u^r)$, $z^{r+1}$ and line-search parameters: $\nu_1$, $\nu_2 \in (0, 0.5)$, and $0 < c_1 < c_2 < 1$;
/*Main line-search process*/
Step 1: Calculate $p^{r+1} = z^{r+1} - z^r$;
Step 2: Define the merit function $M$ in the form of:
    $M(z^{r+1}; \nu_1, \nu_2) = \nu_1\sum_{k=1}^{N}\|h_k(z^{r+1})\|_1 + \nu_2\sum_{k=1}^{N}\|g_k^+(z^{r+1})\|_1$
        where $g_k^+(z^{r+1}) = \max\{g_k(z^{r+1}), 0\}$;
Step 3: Calculate the directional derivative of $M$ via:
    $\Delta(M(z^{r+1}; \nu_1, \nu_2); p^{r+1}) =$
    $\lim_{\varepsilon \to 0}\left\{\frac{M(z^{r+1} + \varepsilon p^{r+1}; \nu_1, \nu_2) - M(z^{r+1}; \nu_1, \nu_2)}{\varepsilon}\right\}$
Step 4: Search $\alpha^{r+1}$ such that the following
        condition holds true:
    $M(z^{r+1}; \nu_1, \nu_2) + c_1\alpha^{r+1}\Delta(M(z^{r+1}; \nu_1, \nu_2); p^{r+1})$
      $\leq M(z^{r+1} + \alpha^{r+1}p^{r+1}; \nu_1, \nu_2)$
      $\leq M(z^{r+1}; \nu_1, \nu_2) + c_2\alpha^{r+1}\Delta(M(z^{r+1}; \nu_1, \nu_2); p^{r+1})$
Step 5: Execute $z^{r+1} = z^r + \alpha^{r+1}p^{r+1}$;
Output: The updated pair $(x^{r+1}, u^{r+1})$;
```

In Algorithm 2, h represents the equality constraints such as the linearization of the dynamics and g represents the inequality constraints such as the path constraints. Then, $h_k(z^{r+1})$ and $g_k^+(z^{r+1})$ measure the error due to the linearization process and the path constraint violation magnitude at node k, respectively. The value of $h_k(z^{r+1})$ and $g_k(z^{r+1})$ can be calculated via:

$$h_k(z^{r+1}) = \sum_{i=0}^{N} D_{ik}x_i^{r+1} - \frac{s_f - s_0}{2}(f(x_k^{r+1}, u_k^{r+1}) \tag{35}$$

$$g_k(z^{r+1}) = R^{\min} - R(x_k^{r+1}, u_k^{r+1}) \tag{36}$$

Based on Eq.(35) and Eq.(36), the total violation of the nonconvex constraints is penalized in the l_1 merit function $M(z^{r+1}; \nu_1, \nu_2)$ with the positive penalty factors denoted by ν_1 and ν_2.

Remark 3. Currently there are mainly two types of chance constraint approximation strategies: the convex approximations [33], [34] and the nonconvex approximations [23]. Most of the nonconvex methods aim to approximate the H function aggressively, thereby reducing the conservatism and improving the optimality of solutions. However, using this kind of technique might degrade the convexity of the original problem, thus resulting in more computational burdens. Due to this reason, in the present work, we have paid more attention to convex approximations. It was analyzed in [14] that compared with other convex approximation functions, the function Ψ tends to result in least conservative bounds for solving standard uncertain control problems. Hence, we apply this function to deal with the chance constraints existing in the trajectory planning task.

V. Numerical Results

A. Unmanned Vehicle Trajectory Generation

The proposed convex programming-based trajectory planning approach is firstly validated by performing a long distance case study that chance constraints (e.g., Eq.(10a) and Eq.(10b)) are not considered. The initial and terminal boundary settings for this particular case are assigned as $x_0 = [500m, 100m, 300m, 15°, 240°]$, and $x_f = [-100m, 400m, 0m, 15°, 45°]$, respectively. In terms of the geometric constraints, the numerical results were obtained under the condition of $R^{\min} = 40m$ and $\gamma \in [-15°, 20°]$. $\mu_1, \mu_2 \in [-1°/s, 1°/s]$.

In Fig. 1, the state and control trajectories of the unmanned system are displayed by performing the nonlinear pseudospectral algorithm (NPS) [27] and the convexification-based trajectory planning approach established in this paper. Note that to execute the NPS, the second generation general purpose optimal control software (GPOPS-II) is used [35]. On the other hand, for the convexified optimization model, a state-of-the-art solver developed in [36] using the primal-dual interior point algorithm with index of accuracy $\epsilon = 10^{-8}$ is applied to explore the optimal solution. The trust region constraints for x and u are specified as $\delta_x = [1000, 1000, 1000, \pi, 2\pi]^T$ and $\delta_u = [1, 1]^T$, respectively. Besides, the stop conditions are assigned as $\epsilon_x = [0.1, 0.1, 0.1, \frac{0.5\pi}{180}, \frac{1\pi}{180}]^T$ and $\epsilon_u = [0.01, 0.01]^T$, respectively. To start the optimization process for the sequential convex optimization process, an initial guess trajectory should be provided. For the considered mission case, linear interpolation between the state boundary conditions is performed and the resulting state sequence is applied as the initial state guess profile x^0. As for the initial control profile u^0, we simply choose it as zero. By assigning $x^r = x^0$ and $u^r = u^0$, the optimization process demonstrated in Algorithm 1 can be triggered.

Fig. 1(a) presents the projection of the unmanned vehicle trajectory on the y-z plane, whereas Fig. 1(b) shows the flight path on the 3-D plane. The heading angle profile, together with the pitch angle profile, is shown Fig. 1(c) and Fig. 1(d). Two control signals (e.g., μ_1 and μ_2) are presented in Fig. 1(e) and Fig. 1(f), respectively.

According to the obtained results, it can be seen that both the two approaches can produce flight path without violating constraints. Hence, their effectiveness can be verified to some extent. There are some differences in the state profiles produced by applying these two algorithms. These differences might be attributed to multiple potential reasons such as the linearization process with respect to the system dynamics and the path constraint. Furthermore, compared to the NPS

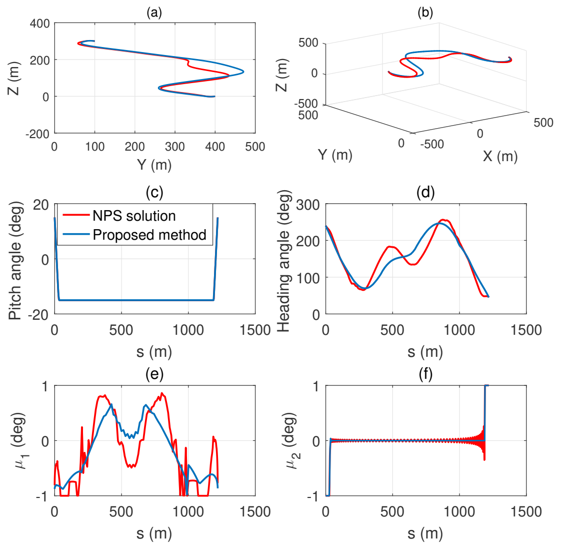

Fig. 1: Optimized trajectories for the unmanned vehicle

solutions, the state and control profiles computed using the proposed trajectory planning algorithm tend to be smoother. This is more apparent in the obtained control profiles (e.g., Fig. 1(e) and Fig. 1(f)) where high-frequency oscillations can be observed from the evolution trajectories planned via the NPS. By contrast, the control evolution profiles obtained via the proposed approach tend to be more stable and easier to achieve, thereby further enhancing its potential for real-world applications.

As for the optimality of the solution and the computational performance of the algorithm, detailed results are also collected. The optimal path length achieved via the two approaches are, respectively, $s_{NPS} = 1203.32m$ and $s_{prop} = 1205.33m$. However, the processing time for generating the solution of using the NPS is around $6.37s$, which is much higher than the time required by the proposed method (e.g., $0.77s$). Moreover, since the proposed algorithm is successfully converged, the global optimality of its solution can be guaranteed. While the solution obtained using the NPS can only be treated as a local optimal (or a near-optimal) solution.

B. Comparative Case Study: Without Chance Constrains

In this subsection, a number of comparative case studies without considering chance constrains were executed. The parameter specification for different test cases including the initial and final pose information of the unmanned vehicle is tabulated in Table I. Note that these test cases were designed in [12]. To make a fair comparison, we re-perform these test cases using the proposed algorithm. Also, it is worth mentioning that the selected test cases contain both short-range and long-range flying scenarios. Specifically, the first three test cases are likely to result in flight paths with relatively-small distances, whereas the last three test cases are likely to result in flight paths with relatively-large distances.

The performance of using the proposed approach to

TABLE I: Settings for different test cases

Case	Initial pose			Final pose		
	(p_x, p_y, p_z)	γ	ϕ	(p_x, p_y, p_z)	γ	ϕ
1	$(120, -30, 250)$	-10	100	$(220, 150, 100)$	-10	300
2	$(380, 230, 200)$	0	30	$(280, 150, 30)$	0	200
3	$(-80, 10, 250)$	0	20	$(50, 70, 0)$	0	240
4	$(400, -250, 600)$	0	350	$(600, -150, 300)$	0	150
5	$(-200, -200, 450)$	0	340	$(-300, -80, 100)$	0	100
6	$(-200, 200, 250)$	15	240	$(500, 800, 0)$	15	45

address different test case is compared against other well-developed techniques existing in the literature. For example, the NPS algorithm reported in [27], an in-flight waypoint-based algorithm investigated in [24], and the multiple shooting-based (MS) method developed in [26]. The detailed planning results including the lengths of trajectories s and execution times t_p for different flying scenarios are summarised in Table II.

From Table II, it can be observed that the optimal trajectory length values obtained via the proposed method and the NPS method are comparable. The proposed fast trajectory planning method can generally produce more optimal flight paths than the approach of Babaei et al. [24] and the MS-based method [26]. However, the execution time required by the strategy developed in this paper tends to be far less than its counterparts.

It should be noted that in [12] and [13], the authors proposed a geometric-based path generation approach to produce the flight path for the unmanned vehicle. It was shown that this method is able to produce feasible trajectory and the average execution time tends to be smaller than the one developed in this study. However, one critical problem of the previous design is that the produced trajectory might contain several transient disturbance points in the system state profiles. Moreover, based on our experiments, for some mission cases, the optimality of the flight path obtained via the previously developed method is not as comparable as the results obtained via the method developed in this paper.

C. Chance-Constrained Unmanned Vehicle Trajectory Generation

The impact of control chance constraints on the optimal unmanned vehicle flight trajectory is now analyzed. In Eq.(10), ξ_{μ_1} and ξ_{μ_2} are supposed to have an exponential distribution, and the PDF associated with them can be written as: $f(x;\lambda) = \lambda e^{-\lambda x}, x \geq 0$; $f(x;\lambda) = -\lambda e^{-\lambda x}, x < 0$ with the rate parameter $\lambda = 70$. Furthermore, the risk parameter is assigned as $1-\epsilon_{\mu_1} = 1-\epsilon_{\mu_2} = 0.05$ (5%).

The proposed fast trajectory generation approach is then applied to address the chance-constrained trajectory planning problem by incorporating the convex chance constraint approximation method stated in Sec IV. To transform the probabilistic constraints, a relatively large-sized sample (e.g. $N = 2 \times 10^5$) is selected. The effectiveness of this integrated framework is validated by performing a short distance mission case (e.g., case 1 in Table II). The optimized state and control profiles are portrayed in Fig. 2, where the blue line indicates the chance-constrained solution, while the red line indicates the solution without considering Eq.(10).

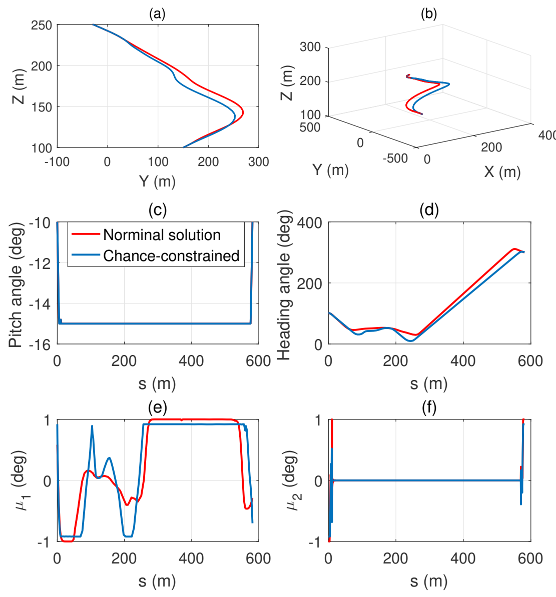

Fig. 2: Optimal Chance-constrained trajectories for the unmanned vehicle

From the results, it is calculated that the maximum violation rates for Eq.(10) are all smaller than 5%, which confirms the reliability of the convex chance constraint approximation technique. Besides, it can be seen from Fig. 2 that the consideration of control chance constraints results in a slight increase in terms of the path length. This is because based on the control profiles shown in Fig. 2(e) and Fig. 2(f), the control variables cannot reach their maximum or minimum allowable values at some time periods. Due to the lack of controllability, the unmanned vehicle might need to have a relatively longer flight in order to achieve the pre-specified final pose.

D. Comparative Case Studies: With Control Chance Constrains

Similar with the work presented in Sec V.B, in this subsection, a set of comparative case studies with the consideration of chance constraints were performed to assess the performance of the proposed chance-constrained smooth trajectory generation approach. The nominal trajectory design formulation (21) is incorporated with one nonconvex chance constraint approximation (NCCA) strategy and two additional

TABLE II: Comparative results for different approaches

Case	Proposed		NPS [27]		Babaei et al. [24]		MS-based [26]	
No.	s (m)	t_p (s)	s (m)	t_p (s)	s (m)	t_p (s)	s (m)	t_p (min)
1	580.75	0.38	582.73	5.43	750.82	6.29	583.48	1.11min
2	668.19	0.58	675.18	5.85	875.10	7.45	670.37	1.08min
3	977.18	0.42	978.52	5.64	1200.63	7.27	979.91	1.23min
4	1168.61	0.53	1168.61	5.72	1375.14	6.69	1169.74	1.00min
5	1360.80	0.61	1363.80	6.31	1667.44	6.83	1370.23	1.47min
6	1167.90	1.02	1169.95	6.07	1401.84	6.76	1168.83	2.05min

convex chance constraint approximation strategies. Specifically, the NCCA method developed in [23] (with index of optimization accuracy $\epsilon = 10^{-8}$), the exponential-function based method (also known as Bernstein method (BM)) suggested in [33], and a kinship function-based (KF) approach designed in [34]. The general idea of these three strategies is based on approximation of the probabilistic constraints. Different test cases listed in Table I were re-performed by applying the four chance-constrained trajectory planning schemes. The results regarding the path length and execution time are tabulated in Table III.

Firstly, a comparison is made between the method developed in this paper and the NCCA method proposed in [23]. According to the quantitative comparisons demonstrated in Table III, the proposed method is able to converge to a comparable objective with much faster convergence speed than the one developed in [23]. Actually, the method developed in [23] aims to approximate the H function aggressively, thus reducing the conservatism and improving the optimality of solutions. As a result, for some test cases (e.g., case 1, case 3 and case 4), the NCCA method can produce a slightly better solution than the proposed algorithm. However, using this approach will damage the convexity of the original problem, thus resulting in a large amount of computational burdens for the optimization process. A potential solution to this issue is to loosen the optimization tolerance value. This means the accuracy of the obtained solution might be decreased.

Next, the results obtained using different convex approximation-based methods are analyzed. According to Table III, compared to the BM and KF methods, the fast chance-constrained trajectory generation strategy suggested in this paper can produce a more optimal flight path in short time for different flight scenarios. This can partly reveal that the suggested convex chance constraint approximation method is less conservative than the one reported in [33] and [34], and it is more suitable to be applied for addressing the considered unmanned vehicle trajectory design task. Actually, if a convex approximation approach is too conservative, the resulting feasible set tends to be small. This indicates that the optimization process starts searching the solution in a highly limited space. As a result, the difficulty of finding the optimal solution will be increased, and the computational performance of the algorithm will be degraded significantly.

E. Comparative Case Studies: With Control and Obstacle Chance Constrains

In this subsection, we explore the impact of probabilistic collision avoidance constraints on the optimal unmanned vehicle flight trajectory. One short-range flight case and one long-range flight case designed in Table I (e.g., case 2 and case 4) were re-performed with the consideration of stochastic obstacles existing in the environment. Specifically, uncertain no-fly zone constraints are imposed on the x-y plane. That is, $p_y = a_{mn}p_x + b_{mn} + \xi_{mn}$. Detailed information regarding the obstacles is specified as follows. For flight case 2, two obstacles are considered:

$$O^1 \begin{cases} a_{11} = 0.5 & b_{11} = 140 \\ a_{21} = 0.05 & b_{21} = 330 \\ a_{31} = -4.7 & b_{31} = 1650 \\ a_{41} = -3.2 & b_{41} = 1433 \end{cases}$$

$$O^2 \begin{cases} a_{12} = 0.1 & b_{12} = 280 \\ a_{22} = 10 & b_{22} = -3620 \\ a_{32} = -0.3 & b_{32} = 505 \\ a_{42} = 10 & b_{42} = -4187 \end{cases}$$

For flight case 4, two obstacles are considered:

$$O^3 \begin{cases} a_{13} = -13 & b_{13} = 7590 \\ a_{23} = 1 & b_{23} = -1150 \\ a_{33} = -0.4 & b_{33} = -134 \end{cases}$$

$$O^4 \begin{cases} a_{14} = -1 & b_{14} = 620 \\ a_{24} = 1 & b_{24} = -970 \\ a_{34} = 1 & b_{34} = -1150 \\ a_{44} = -1 & b_{44} = 460 \end{cases}$$

The uncertain parameter ξ_{mn} is assumed to follow a Gaussian distribution of $\mathscr{N}(0, 1.5)$. The maximum allowable violation probability is set to 0.1.

Different from the strategy applied in previous subsections, the optimized solutions reported in Table II (e.g., the solutions obtained without considering the probabilistic control and obstacle avoidance constraints) are selected as the initial guess trajectories to trigger the iterative optimization process. Note that a detailed sensitivity study with respect to different initial guess generation methods will be provided in the next subsection. By addressing the formulation given by Eq.(33), the chance-constrained solutions are calculated.

Fig. 3 and Fig. 4 demonstrate the evolutions of the objective function value during the sequential optimization process for the two flight cases. In addition, Fig. 5 and Fig. 6 further illustrate the corresponding convergence histories with respect to the merit function value. It is worth noting that objective value can be used to reflect the optimality of the obtained solution, whereas merit function value measures the constraint violation. From Fig. 3 and Fig. 4, it is obvious that for the considered cases, the objective function

TABLE III: Chance-constrained results for different approaches

Case No.	Proposed		NCCA [23]		BM [33]		KF [34]	
	s (m)	t_p (s)	s (m)	t_p (s)	s (m)	t_p (s)	s (m)	t_p (s)
1	582.62	4.23	581.72	42.23	591.25	17.14	610.36	37.44
2	669.64	4.75	670.11	44.65	673.24	21.33	679.24	41.27
3	978.06	3.46	978.03	57.92	981.58	15.35	987.79	36.55
4	1169.83	2.62	1169.65	47.78	1172.11	12.27	1177.37	33.38
5	1362.75	4.13	1366.77	53.72	1368.85	22.34	1375.24	35.89
6	1174.74	6.15	1181.58	56.63	1202.29	31.61	1226.56	51.69

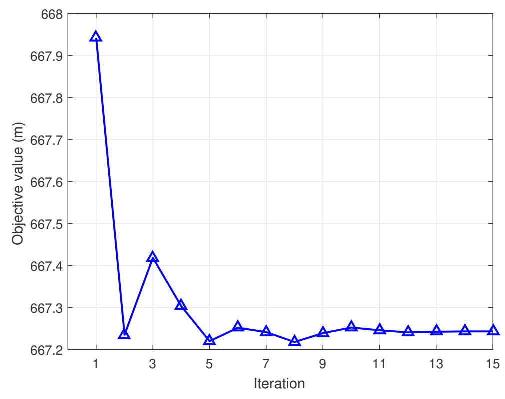

Fig. 3: Convergence history of the objective function: Case 2 with obstacles

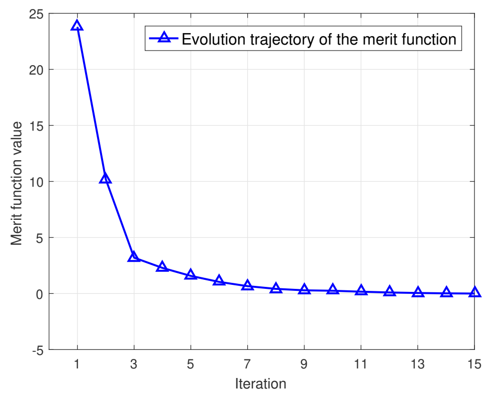

Fig. 5: Convergence history of the merit function: Case 2 with obstacles

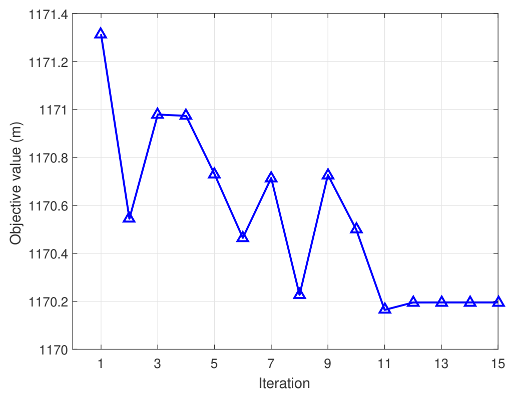

Fig. 4: Convergence history of the objective function: Case 4 with obstacles

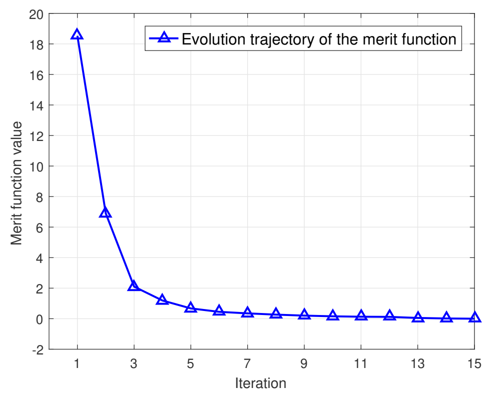

Fig. 6: Convergence history of the merit function: Case 4 with obstacles

eventually decreases to a converged value and remain stable until the specified tolerance level is reached (e.g., both of these two cases terminate after 15 iterations). Interestingly, by viewing the evolution histories of the objective value, it can be observed that the converged solutions do not have the minimum objective values. More precisely, the objective value is not monotonically decreasing and there exist higher/lower objective values of some intermediate iterations than that of the converged solution. Actually, by performing the line search process detailed in Algorithm 2, it is likely to obtain an updated solution pair which can result in a decrease in terms of the merit function value. This can be confirmed by the merit function trajectories presented in Fig. 5 and Fig. 6, where monotonically decreasing evolution histories are obtained for the two considered cases. However, there is no guarantee that the updated solution pair can result in a strict decrease or increase with respect to the objective value. For example, in some intermediate iterations, the solution optimality might be sacrificed so as to achieve a progress in terms of the merit function. This could be one potential reason for the oscillations existing in the evolution histories of the objective value.

Fig. 7 and Fig. 8 present the convergence history of the flight trajectory (projected on the x-y plane). To clearly present the convergence history, the initial trajectories are indicated by black dash lines, while the flight trajectories at different optimization iterations are indicated by solid lines and their colors are changed from dark blue to red. As can be seen from these two figures, the flight trajectories for these two cases tend

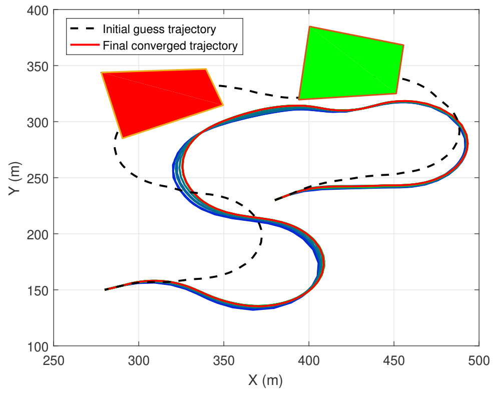

Fig. 7: Convergence history of the flight trajectory: Case 2 with obstacles

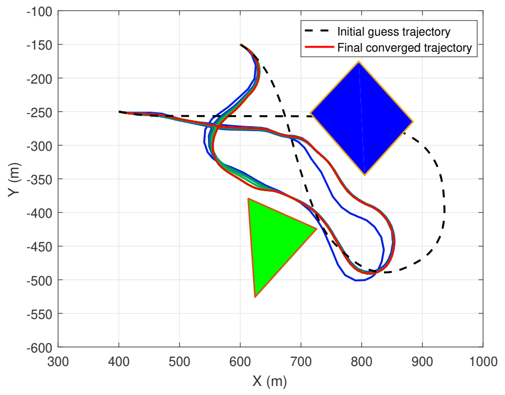

Fig. 8: Convergence history of the flight trajectory: Case 4 with obstacles

to become more aggressive as the iteration number increases. That is, there is a tendency for the vehicle to approach the uncertain obstacles so as to achieve a better objective value. Moreover, based on our observation, the trajectories become very close after 5 iterations for case 2 and after 11 iterations for case 4. This can also be reflected by viewing the objective function and merit function evolution profiles.

F. Sensitivity Analysis

In this subsection, a sensitivity study is firstly executed in order to analyze the impact of different initial guess generation methods on the convergence performance of the proposed approach. The methods selected for analysis are:

- Method A: The initial state trajectory x^0 is obtained by propagating the vehicle dynamics given by Eq.(1) from the initial boundary value x_0 via a specified initial control trajectory u^0. In the test, u^0 is simply assigned as zero.
- Method B: The initial state trajectory x^0 is obtained by performing the operation of linear interpolation between the state boundary conditions (e.g., x_0 and x_f), whereas the initial control trajectory u^0 is simply selected as zero.
- Method C: The solutions to the convex optimization problem without considering the probabilistic control and collision avoidance constraints are selected as the initial state/control guess trajectories.

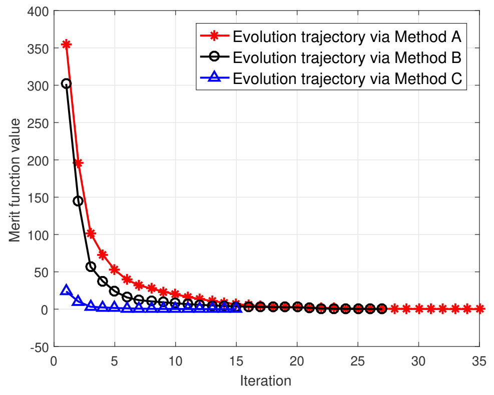

Fig. 9: Merit function histories for case 2 using different guess trajectories

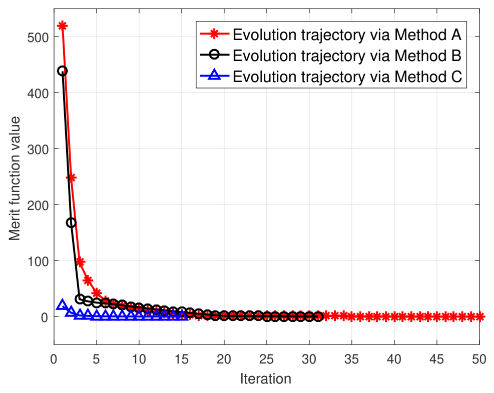

Fig. 10: Merit function histories for case 4 using different guess trajectories

Fig. 9 and Fig. 10 illustrate the resulting evolution trajectories of the merit function for mission case 2 and mission case 4, respectively. From the displayed results, it is obvious that for both mission cases, if the third initial guess generation method is applied, the sequential optimization process takes fewer iterations to converge. Actually, as shown in Fig. 10, the optimization process using the first initial guess generation method suffers from a convergence issue for mission case 4. That is, after reaching the maximum allowable iteration number (e.g., the iteration number reaches 50), the current solution still fails to satisfy the prescribed convergence condition. Based on these comparative results, we can conclude that the convergence performance of the proposed approach tends to be sensitive with respect to the initial guess trajectories. Moreover, for the optimization model given by Eq.(33), it is suggested to apply Method C to produce initial guess trajectories and start the sequential optimization process.

Another parameter which may have an impact on the optimized results is the maximum allowable violation probability ϵ_o for the probabilistic obstacle avoidance constraints. To further test the performance of the proposed convex probabilistic collision avoidance constraint approximation strategy, ϵ_o is assigned to different values. For example, ϵ_o is assigned to three levels (Level $1 = 0.1$, Level $2 = 0.05$, Level $3 = 0.01$) for testing. Fig. 11 illustrates the planned flight trajectory for mission case 2 in the x-y plane with different allowable constraint violation probabilities. Similarly, Fig. 12 portrays the simulated trajectory results for mission case 4.

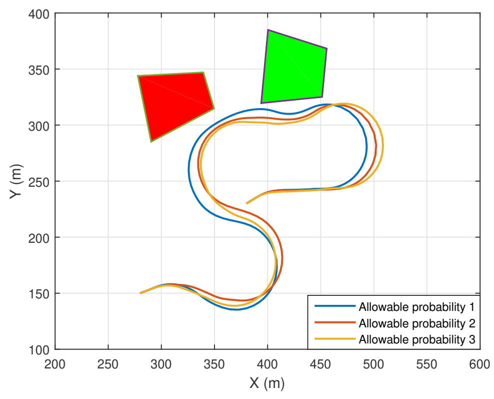

Fig. 11: Optimal Chance-constrained trajectories: Case 2 with obstacles

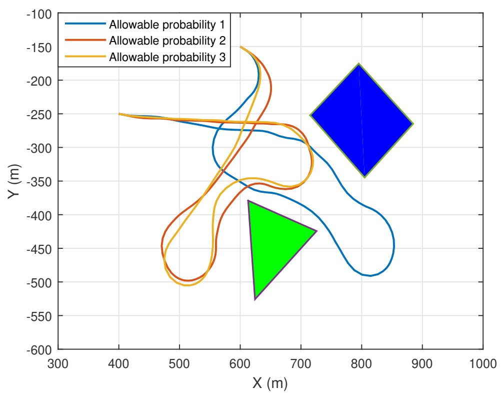

Fig. 12: Optimal Chance-constrained trajectories: Case 4 with obstacles

As can be observed from Fig. 11 and Fig. 12, the optimal chance-constrained flight trajectory tends to become more conservative as the maximum allowable constraint violation threshold decreases. This phenomenon becomes more obvious in the result of flight case 4, where the unmanned vehicle changes its maneuver from passing through the two obstacles to executing a much safer yet longer flight outside the obstacles. Detailed results such as the path length and execution time are tabulated in Table IV.

TABLE IV: Results with uncertain obstacles

Allowable probability	Case 2 results s (m)	Case 2 results t_p (s)	Case 4 results s (m)	Case 4 results t_p (s)
Level 1	674.23	5.81	1170.22	5.25
Level 2	675.35	6.98	1191.53	6.38
Level 3	675.82	8.33	1192.15	7.47

From Table IV, it is obvious that the execution time required by the strategy developed in this paper might experience a slight increase as the probabilistic collision avoidance constraint becomes tighter to satisfy. However, all the trials can still converge in few seconds. These results further confirm the effectiveness of using the proposed mixed-integer convex chance-constrained optimization model given by (33) to plan the flight trajectory of the unmanned vehicle with the consideration of probabilistic control and collision avoidance constraints.

VI. Conclusion

A fast chance-constrained trajectory planning algorithm incorporating convex optimization and convex approximation of probabilistic constraints is presented to solve the problem of unmanned vehicle path generation. One important feature of the proposed trajectory generation algorithm is that the constructed optimization model is a deterministic convex program even though probabilistic control and obstacle avoidance constraints are taken into account. By comparing against other trajectory generation strategies reported in the literature, the proposed convexification-based formulation has two main advantages. Firstly, the calculated system state and control profiles tend to be smooth. Another advantage is that it can significantly improve the computational performance while optimizing the flight path. These two advantages have been validated by a number of comparative case studies demonstrated in this paper. Hence, we believe the suggested approach and obtained results are of particular interest to the community that is involved within chance-constrained optimization applications and unmanned vehicle trajectory planning tasks.

References

[1] B. Zhang, L. Tang, J. DeCastro, M. J. Roemer, and K. Goebel, "A recursive receding horizon planning for unmanned vehicles," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 5, pp. 2912–2920, 2015.

[2] S. H. Ramesh and R. Padhi, "Three-dimensional nonlinear gravity assisted aiming point guidance," *Aerospace Science and Technology*, vol. 85, pp. 505–513, 2019.

[3] E. P. Anderson, R. W. Beard, and T. W. McLain, "Real-time dynamic trajectory smoothing for unmanned air vehicles," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 3, pp. 471–477, 2005.

[4] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, E. D. Lellis, and A. Pironti, "Path generation and tracking in 3-d for uavs," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 980–988, 2009.

[5] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 561–568, 2010.

[6] B. Alidaee, H. Wang, and F. Landram, "A note on integer programming formulations of the real-time optimal scheduling and flight path selection of uavs," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 839–843, 2009.

[7] A. Rucco, P. B. Sujit, A. P. Aguiar, J. B. d. Sousa, and F. L. Pereira, "Optimal rendezvous trajectory for unmanned aerial-ground vehicles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 2, pp. 834–847, 2018.

[8] V. Roberge, M. Tarbouchi, and G. Labonte, "Fast genetic algorithm path planner for fixed-wing military uav using gpu," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2105–2117, 2018.

[9] J. J. Kim and J. J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 620–631, 2015.

[10] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, “Trajectory optimization of space maneuver vehicle using a hybrid optimal control solver,” *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 467–480, 2019.

[11] G. Tang and K. Hauser, “A data-driven indirect method for nonlinear optimal control,” *Astrodynamics*, vol. 3, no. 4, pp. 345–359, 2019.

[12] Y. Wang, S. Wang, M. Tan, C. Zhou, and Q. Wei, “Real-time dynamic dubins-helix method for 3-d trajectory smoothing,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 730–736, 2015.

[13] Y. Wang, S. Wang, and M. Tan, “Path generation of autonomous approach to a moving ship for unmanned vehicles,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5619–5629, 2015.

[14] M. P. Vitus, Z. Zhou, and C. J. Tomlin, “Stochastic control with uncertain parameters via chance constrained control,” *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2892–2905, 2016.

[15] Z. Wang and M. J. Grant, “Constrained trajectory optimization for planetary entry via sequential convex programming,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 10, pp. 2603–2615, 2017.

[16] ——, “Optimization of minimum-time low-thrust transfers using convex programming,” *Journal of Spacecraft and Rockets*, vol. 55, no. 3, pp. 586–598, 2017.

[17] ——, “Minimum-fuel low-thrust transfers for spacecraft: A convex approach,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2274–2290, 2018.

[18] Z. Wang, “Optimal trajectories and normal load analysis of hypersonic glide vehicles via convex optimization,” *Aerospace Science and Technology*, vol. 87, pp. 357–368, 2019.

[19] H. Yang, X. Bai, and H. Baoyin, “Rapid generation of time-optimal trajectories for asteroid landing via convex optimization,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 3, pp. 628–641, 2017.

[20] M. Szmuk, C. A. Pascucci, and B. Acikmese, “Real-time quad-rotor path planning for mobile obstacle avoidance using convex optimization,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Conference Proceedings, pp. 1–9.

[21] Z. Wang and S. T. McDonald, “Convex relaxation for optimal rendezvous of unmanned aerial and ground vehicles,” *Aerospace Science and Technology*, vol. 99, p. 105756, 2020.

[22] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[23] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, Y. Xia, and S. Wang, “Solving trajectory optimization problems in the presence of probabilistic constraints,” *IEEE Transactions on Cybernetics*, pp. 1–14, 2019.

[24] A. R. Babaei and M. Mortazavi, “Three-dimensional curvature-constrained trajectory planning based on in-flight waypoints,” *Journal of Aircraft*, vol. 47, no. 4, pp. 1391–1398, 2010.

[25] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, “A unified framework for the numerical solution of optimal control problems using pseudospectral methods,” *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.

[26] S. Hota and D. Ghose, “Optimal trajectory planning for path convergence in three-dimensional space,” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 228, no. 5, pp. 766–780, 2014.

[27] T. Guo, J. Li, H. Baoyin, and F. Jiang, “Pseudospectral methods for trajectory optimization with interior point constraints: Verification and applications,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 2005–2017, 2013.

[28] M. Sagliano, “Pseudospectral convex optimization for powered descent and landing,” *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 2, pp. 320–334, 2017.

[29] ——, “Generalized hp pseudospectral-convex programming for powered descent and landing,” *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 7, pp. 1562–1570, 2019.

[30] Y. Mao, M. Szmuk, and B. Acikmese, “Successive convexification of non-convex optimal control problems and its convergence properties,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Conference Proceedings, pp. 3636–3641.

[31] P. Marantos, C. P. Bechlioulis, and K. J. Kyriakopoulos, “Robust trajectory tracking control for small-scale unmanned helicopters with model uncertainties,” *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 2010–2021, 2017.

[32] J. Kronqvist, D. E. Bernal, A. Lundell, and I. E. Grossmann, “A review and comparison of solvers for convex minlp,” *Optimization and Engineering*, vol. 20, no. 2, pp. 397–455, 2019.

[33] A. Nemirovski and A. Shapiro, “Convex approximations of chance constrained programs,” *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2006.

[34] C. Feng, F. Dabbene, and C. M. Lagoa, “A kinship function approach to robust and probabilistic optimization under polynomial uncertainty,” *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1509–1523, 2011.

[35] M. A. Patterson and A. V. Rao, “Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming,” *ACM Trans. Math. Softw.*, vol. 41, no. 1, pp. 1–37, 2014.

[36] A. Domahidi, E. Chu, and S. Boyd, “Ecos: An socp solver for embedded systems,” in *2013 European Control Conference (ECC)*, Conference Proceedings, pp. 3071–3076.

Runqi Chai (S’15-M’18) received the B.S. degree in information and computing science from the North China University of Technology, Beijing, China, in 2015 and the Ph.D. degree in Aerospace Engineering from Cranfield University, Cranfield, U.K, in August 2018. He is currently a research fellow at Cranfield University. His research interests include trajectory optimization, networked control systems, and multi-agent control systems.

Antonios Tsourdos obtained a MEng on Electronic, Control and Systems Engineering from the University of Sheffield, in 1995, an MSc on Systems Engineering from Cardiff University in 1996 and a PhD on Nonlinear Robust Autopilot Design and Analysis from Cranfield University in 1999. He joined the Cranfield University in 1999 as lecturer, appointed Head of the Centre of Autonomous and Cyber-Physical Systems in 2007 and Professor of Autonomous Systems and Control in 2009 and Director of Research - Aerospace, Transport and Manufacturing in 2015. He leads the research theme on autonomous systems within the School of Aerospace, Transport and Manufacturing at Cranfield University. He has diverse expertise in both unmanned and autonomous vehicles as well as networked systems.He conducts basic and applied research in the fields of guidance, control and navigation for single and multiple unmanned autonomous vehicles as well as research on cyber-physical systems.

Al Savvaris received the M.Eng. degree in aerospace systems engineering from the University of Hertfordshire, Hertfordshire, U.K., in 1998 and the Ph.D. degree in radiowave propagation and system design from the University of South Wales, Pontypridd, U.K., in 2004.

He is a Reader with the Centre for Cyber-Physical Systems, Cranfield University, Cranfield, U.K. He established the Autonomous Vehicle Dynamics and Control M.Sc. course and the COMAC training programme at Cranfield. His current research interests include systems integration, hybrid energy management, communication systems, embedded systems, guidance, and control. He is currently researching on Innovate U.K. Funded AirStart and USMOOTH Projects. In the past, he researched on the FLAVIIR and ASTRAEA UAS Projects, developing new technologies for unmanned systems, researching on hardware and system integration. He has published over 100 peer-reviewed journal and conference papers.

Shuo Wang received the B.E. degree in electrical engineering fromShenyang Architectural and Civil Engineering Institute, Shenyang, China, in 1995, the M.E. degree in industrial automation from Northeastern University, Shenyang, in 1998, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2001. He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include biomimetic robot, underwater robot, and multirobot systems.

Yuanqing Xia (M'15-SM'16) was born in Anhui Province, China, in 1971. He received the B.S. degree from the Department of Mathematics, Chuzhou University, Chuzhou, China, in 1991, the M.S. degree in fundamental mathematics from Anhui University, Wuhu, China, in 1998, and the Ph.D. degree in control theory and control engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001. His current research interests are in the fields of networked control systems, robust control and signal processing, active disturbance rejection control and flight control. He has published 8 monographs with Springer and Wiley, and more than 300 papers in journals. He has obtained Second Award of the Beijing Municipal Science and Technology (No. 1) in 2010, Second National Award for Science and Technology (No. 2) in 2011, and Second Natural Science Award of The Ministry of Education (No. 1) in 2012. He is a Deputy Editor of the Journal of the Beijing Institute of Technology, Associate Editor of Acta Automatica Sinica, Control Theory and Applications, the International Journal of Innovative Computing, Information and Control, and the International Journal of Automation and Computing.

Senchun Chai (M'19-SM'19) received the B.S and Master degree from Beijing Institute of Technology, Beijing, China from 1997 to 2004 and the Ph.D. degree in Networked Control System from University of South Wales, Pontypridd, U.K., in 2007.

He is currently a professor of School of Automation with Beijing Institute of Technology. He was a research fellow at Cranfield University, UK, from 2009 to 2010, and was a visiting scholar at University of Illinois at Urbana-Champaign Urbana, USA, from January 2010 to May 2010. He has published over 100 journal and conference papers. His current research interests focus on flight control system, networked control systems, embedded systems and multi-agent control systems.