Chinese Journal of Aeronautics, 2013,26(3): 646–660

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

Adaptive path planning for unmanned aerial vehicles based on bi-level programming and variable planning time interval

Liu Wei [a,b,c,*], **Zheng Zheng** [a,b], **Cai Kaiyuan** [a]

[a] *School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China*
[b] *Science and Technology on Aircraft Control Laboratory, Beijing 100191, China*
[c] *High-Tech Institute of Xi'an, 710025, China*

Received 6 June 2012; revised 3 September 2012; accepted 3 December 2012
Available online 16 May 2013

KEYWORDS

Adaptive;
Bi-level programming;
Motion planning;
Unmanned aerial vehicles;
Variable time interval

Abstract This paper presents an adaptive path planner for unmanned aerial vehicles (UAVs) to adapt a real-time path search procedure to variations and fluctuations of UAVs' relevant performances, with respect to sensory capability, maneuverability, and flight velocity limit. On the basis of a novel adaptability-involved problem statement, bi-level programming (BLP) and variable planning step techniques are introduced to model the necessary path planning components and then an adaptive path planner is developed for the purpose of adaptation and optimization. Additionally, both probabilistic-risk-based obstacle avoidance and performance limits are described as path search constraints to guarantee path safety and navigability. A discrete-search-based path planning solution, embedded with four optimization strategies, is especially designed for the planner to efficiently generate optimal flight paths in complex operational spaces, within which different surface-to-air missiles (SAMs) are deployed. Simulation results in challenging and stochastic scenarios firstly demonstrate the effectiveness and efficiency of the proposed planner, and then verify its great adaptability and relative stability when planning optimal paths for a UAV with changing or fluctuating performances.

1. Introduction

As a crucial subject of unmanned aircraft systems (UAS), path planning for heterogeneous unmanned aerial vehicles (UAVs) has attracted substantial attention.[1,2] Path planning methods for UAVs are to generate safe reference trajectories that navigate a UAV from its present location to a desired target in a hostile environment.[3] Motivated by the advent of new UAVs which encompass a broad range of mission capabilities, such as reconnaissance, strike, and signal collection,[2] a satisfactory path planning method should be practicable and tailored to various UAVs when executed in environments with different obstacle distributions. Therefore, a challenging idea for path planning is how to adapt a method to not only various operational environments, but also UAVs that have different performances with regard to kinematic properties, maneuverability, and sensory capability. This study is driven by the above idea and focuses on adaptive path planning for UAVs with variant performances.

* Corresponding author at: School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China. Tel.: +86 10 82338767.
E-mail addresses: everwl@gmail.com (W. Liu), zhengz@buaa.edu.cn (Z. Zheng), kycai@buaa.edu.cn (K. Y. Cai).

Peer review under responsibility of Editorial Committee of CJA.

1000-9361
http://dx.doi.org/10.1016/j.cja.2013.04.041

1.1. Prior work

Pervious research on path planning has well addressed the method's adaptability to diverse environments distributed with boundary obstacles (e.g., mountains and buildings) or radiation threats (e.g., radars and missiles).[3,4] For completely known environments, methods are put forward to directly find a globally optimal path, such as the Voronoi diagram methods,[5,6] a directed graph based method,[7] a probabilistic approach,[8] and silhouettes.[9] With the growing complexity of flight tasks and operational spaces, threats are increasingly hard to be fully characterized at the start of a mission. Many methods are further developed to generate flight paths in real time, as investigated below.

Rapidly-exploring random trees (RRTs) based algorithms[4,10,11] are to bias the exploration toward undetected space by randomly sampling points, in which the differential constraints (arising from non-holonomy and dynamics) are considered. Mixed integer linear programming (MILP) with alternative receding horizon control (RHC) provides a powerful optimization planning framework for hybrid dynamic models.[12,13] Both probabilistic roadmaps (PRMs)[14] and potential field approaches[15] using generalized sigmoid functions are feasible with very modest computation. Besides, bouncing algorithms[16] are superefficient in path generation with limited information obtaining and unrestricted maneuver, in addition to the behavior coordination and virtual (BCV) goal based algorithm[17] and the feedback based compositional rule of inference (FBCRI) algorithm[18] in our previous work.

Besides the aforementioned methods that have been successfully applied in the presence of multifarious obstacles, other methods embedded with adaptation strategies have dealt with the adaptability to UAVs' different abilities, but only specific subsets of the abilities are considered. To adapt to performance degradation caused by ice accumulation, an evolutionary computation method is presented to flexibly plan paths for UAVs.[19] A simple on-line adaptive path planning algorithm,[20] combined with a nonlinear lateral guidance control law, is designed to reconfigure the current flight path in the event of aircraft performance reduction. Except for the maneuver adaptation in the above methods, another adaptive path planning algorithm for vertical take-off and landing (VTOL) UAVs pays attention to the sensor module adaptation.[21] In a novel fuzzy virtual force (FVF) method,[22] planning parameters are adaptively set by using fuzzy logic reasoning theories and the Bayesian belief network. Moreover, the adaptive immigrant scheme genetic algorithm[23] and the max–min adaptive ant colony optimization (ACO) approach[24] can adaptively improve planning effect and solving efficiency.

1.2. Problem analysis and motivation

Most existing methods are capable of adapting to variant environments, and several of them are flexible to the change of a UAV's single performance. Nevertheless, they neither explicitly exhibit nor completely take into account the adaptability to multiple crucial properties. In fact, better adaptability would conduce to more extensive applications and smaller adjustments when certain of the UAV's operational performances fluctuate. For example, an autonomous rotary-wing aircraft is able to stop and make quick turns on a spot. On the contrary, an autonomous fixed-wing aircraft has to maintain the minimal flight velocity and cannot turn at a large angle instantaneously. If a flight path obtained from a general planning method demands many agile or abrupt maneuvers, it would be hard for the fixed-wing aircraft to fly along, or even completely impossible to track. This may also occur to the same type of UAVs with different maneuverabilities and onboard detection sensors. Consequently, it is insufficient in practice for a planning method to only aim at an invariable UAV model of steady maneuver and exploration abilities.

The fixed planning step, which is commonly adopted to equidistantly determine new waypoints,[3,10,12] may restrict the method's applicability with regard to path safety and solving efficiency. When a UAV is close enough to obstacles at a high speed, a generated path may fail to steer clear of the threat region because of the UAV's inability to decelerate in time. A similar accident is also likely to happen if the UAV's turning ability is strictly limited. Besides, in real-time planning an improper fixed step may result in frequent but unnecessary replanning implementation, which could increase the computational burden and reduce the generated path length and smoothness.

Rather than a fixed step, variable planning step is introduced to build paths or new reference waypoints only when necessary.[25] Typically, an anytime algorithm (AA) is embedded in several existing path planning approaches, such as RRTs,[26] PRMs,[27] and particle swarm optimization (PSO),[28] to gradually improve the path quality as the execution time increases. It can react to current environmental changes and output paths at any time, which brings variable planning intervals and adaptation to the surrounding obstacles and airborne equipment. Likewise, an improved RRT method[4] also adopts the randomly variable planning step to increase path search efficiency.

Accordingly, we extend the considerations of diverse environments and a UAV's single ability change to an integrated demand on multiple performance changes or fluctuations for better adaptability. Meanwhile, to overcome the obstacles of the fixed step mode, a new idea of variable planning step is adopted to adaptively update flight paths. These two novel ideas are our improvements and advantages over the existing path planning formulations in a sense that they deal with the adaptability problem under uncertain conditions.

This study focuses on the development of an adaptive path planner that is able to quickly search optimal or near-optimal flight paths for general UAVs with different performances, including sensory capability, maneuverability, and flight velocity limit. With the introduction of the variable planning time interval as a decision variable, a discrete solving algorithm based on bi-level programming (BLP) is presented as an alternative to build a real-time adaptive path planner. The interactive leader's and follower's objectives, defined as the angle of deviation from a target and the distance to destination respectively, are both minimized in response to the convergence guarantee and the optimization requirement. Through the trade-off between the current flight state and the performance fluctuations in the currently known threat environment, the planning interval (flight time between any two waypoints) is independently determined to adaptively make decisions about new

control input, reference flight state (waypoint and flight direction), and trajectory segment.

2. Adaptive path planning problem

2.1. Problem statement

This section develops the adaptive path planning problem statement for UAVs, which is formulated as a discrete-time system and encompasses the adaptability property in the planning process. The subsequent analyses and model are one of our technical contributions in this study.

Let $\boldsymbol{s}_k \in \boldsymbol{S}$ be the UAV's flight state at time t_k, where $\boldsymbol{S} \subseteq \mathbf{R}^{n_s}$ is the state space with dimension n_s. The UAV's control input at t_k is indicated by $\boldsymbol{u}_k$ which lies in the action space $\boldsymbol{U}$ with dimension n_u, $\boldsymbol{u}_k \in \boldsymbol{U} \subseteq \mathbf{R}^{n_u}$. Moreover, the flight state indicates the UAV's situation and can be formulated as a tuple:

$$\boldsymbol{s}_k = \langle \boldsymbol{w}_k, \theta_k, v_k \rangle \tag{1}$$

where $\boldsymbol{w}_k$ denotes the UAV's position at t_k in the operational space $\boldsymbol{G} \subseteq \mathbf{R}^x$. θ_k is the UAV's heading angle at t_k and is defined by the smallest absolute value rotating from north (positive y-axis) to the UAV's flight direction. It is stipulated that θ_k is positive by clockwise rotation, i.e., $\theta_k \in (-\pi, \pi]$. v_k is the UAV's flight velocity at t_k.

The adaptability is defined to adapt the planner to the variation of a UAV's performances with respect to sensory capability, turning ability, and maximal effective velocity. Let $\boldsymbol{c}_k \in \boldsymbol{C} \subseteq \mathbf{R}^{n_c}$ and $\boldsymbol{\delta}_k \in \boldsymbol{C} \subseteq \mathbf{R}^{n_c}$ indicate the capability input and capability fluctuation of above three aspects at t_k, respectively, where $\boldsymbol{c}_k$ is determined by a UAV's design performances, and $\boldsymbol{\delta}_k$ is affected by flight tasks and surrounding environments. In particular, both $\boldsymbol{c}_k$ and $\boldsymbol{\delta}_k$ are assumed to be mutually independent from current flight state and previous capability inputs. This paper takes the following two conditions into consideration for $\boldsymbol{c}_k$ and $\boldsymbol{\delta}_k$ while calculating flight paths in real time:

(1) Performances are invariable in one flight task (one planning process) but distinct in different flight tasks.
(2) Performances fluctuate in a single flight task.

Consequently, in specified flight tasks, the adaptive real-time path planning for a given UAV model can be described by the following state transition equation group of a discrete-time system:

$$\begin{cases} \boldsymbol{s}_{k+1} = f(\boldsymbol{s}_k, \boldsymbol{u}_k, \boldsymbol{c}_k, \boldsymbol{\delta}_k) \\ \boldsymbol{s}_0 = \boldsymbol{s}_{\text{start}} \\ \boldsymbol{s}_N = \boldsymbol{s}_{\text{target}} \end{cases} \tag{2}$$

where N is the waypoint number, and $\boldsymbol{s}_{\text{start}} = \langle \boldsymbol{w}_{\text{s}}, \theta_{\text{s}}, v_{\text{s}} \rangle$ and $\boldsymbol{s}_{\text{target}} = \langle \boldsymbol{w}_{\text{t}}, \theta_{\text{t}}, v_{\text{t}} \rangle$ indicate the initial and target flight states, respectively. f: $\boldsymbol{S} \times \boldsymbol{U} \times \boldsymbol{C}^2 \to \boldsymbol{S}$ is the state transition function. Under condition (1), the planner to be proposed generates paths when $\boldsymbol{c}_k$ changes in different tasks and keeps invariant in each task, and $\boldsymbol{\delta}_k = \boldsymbol{0}$. Under condition (2), the planner generates paths when $\boldsymbol{\delta}_k \not\equiv \boldsymbol{0}$ ($\boldsymbol{\delta}_k$ is assumed to follow a random distribution) and $\forall k = 0, 1, \ldots, N-1$, $\boldsymbol{c}_k = \boldsymbol{c}_{k+1}$.

Besides, the adaptive real-time path planning should guarantee the flight path safety and meet the path optimization demand. The constraints of obstacle avoidance and a UAV's performance must be modeled to search paths in safe operational spaces with available control inputs. To evaluate the flight paths and optimize the length, computational cost, and other crucial aspects, a BLP-based adaptive path planning idea is analyzed in the next section.

2.2. Basic path planning idea using BLP

The original formulation for BLP was initially mentioned by Bracken and McGill in 1973.[29] Candler and Norton formally put forward its definition in their research report in 1977,[30] who considered a multi-level formulation in the context of agricultural economics. From then on, hundreds of papers have been devoted to this topic.[31–35] Up to now, the BLP theory has been widely used in multiple fields, like military, economics, and operation research.[36–38]

In this paper, we introduce BLP into the UAV's path planning application for an adaptive path planner through construction of a leader–follower decision and optimization model. The adaptability description and the BLP-based planner are the key novelty and another contribution of the study.

The adaptive path generation idea using BLP is directly and closely related to planning objectives, flight decisions regarding flight state and control input (decision variables), and path search constraints. Objectives at two levels are designed to adapt to the flight path improvement by exhibiting a hierarchical characteristic and a principal–subordinate relationship, which is also beneficial to the priority distinction between the two levels' functions. The leader's objective intends to enhance the convergence to the target and obtain a holistic feasible trajectory. It has a higher priority than the follower that optimizes the path in local and conversely influences the leader through the combined decision variables. In addition, the flight time decision in the follower realizes a variable step that can enhance the adaptability. Different search constraints can be easily formulized in the BLP model and guarantee the feasibility of the generated flight paths. Detailed construction of the three components is exhibited as follows:

(1) Path planning objectives. They are defined to guarantee both the planner's feasibility and path optimization. With the current state $\boldsymbol{s}_k$, a new path segment is uniquely determined by f and the control input decision. We evaluate a new decision against the following two criteria: (i) whether the flight direction at $\boldsymbol{w}_{k+1}$ exactly points to $\boldsymbol{w}_{\text{t}}$; and (ii) whether the distance between $\boldsymbol{w}_{k+1}$ and $\boldsymbol{w}_{\text{t}}$ is the shortest. According to the two criteria, the objectives of the leader and follower levels can be constructed as follows (see Fig. 1):
(a) Leader's objective: minimize the **ANGLE** between the new flight direction θ_{k+1} and the line connecting the new waypoint w_{k+1} with the target $\boldsymbol{w}_{\text{t}}$. It is denoted by η_{k+1} and is defined positive if the rotation from the line to the flight direction is clockwise.
(b) Follower's objective: minimize the **DISTANCE** between the new waypoint $\boldsymbol{w}_{k+1}$ and the target $\boldsymbol{w}_{\text{t}}$. The **DISTANCE** is denoted by Dis_{k+1}.
(2) Decision variables. In order to achieve the two objectives, control input is calculated to determine a new satisfactory flight state, and then a path segment to

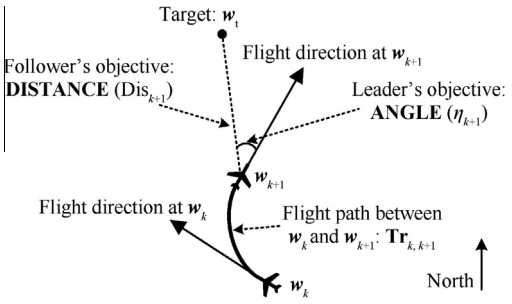


Fig. 1 Flight state transition based on the leader's and follower's objectives.

the new state. In this paper we consider the operational space is two-dimensional so as to simplify the problem. Next, we further define that $\boldsymbol{u_k}$ is composed of yaw rate and flight time: $\boldsymbol{u_k} = [\omega_k \ t_k]$, so $\boldsymbol{s}_{k+1}$ can be quickly generated. In fact, the operational space $\boldsymbol{G}$ can easily be extended to three-dimension if the UAV's rolling and pitching angles are introduced. In this case, ω and t are the decision variables of the BLP-based planner. Against the influences of $\boldsymbol{c_k}$ and $\boldsymbol{\delta_k}$ on a new flight state decision, t_k results in variable planning time intervals that take into account the UAV's performance uncertainty and provide a flexible way for the decisions of ω_k and t_k in each step.

(3) Path search constraints. A feasible flight path must be identified in the safe area of $\boldsymbol{G}$. Let Tr denote the flight path from $\boldsymbol{w}_s$ to $\boldsymbol{w}_t$ and $\mathbf{Tr}_{k,k+1}$ the path segment from $\boldsymbol{w_k}$ to $\boldsymbol{w}_{k+1}$, and then $\mathbf{Tr} = \bigcup_{k=0,1,\ldots,N-1} \mathbf{Tr}_{k,k+1}$. Let $\boldsymbol{X}_{\text{free}} \subseteq \boldsymbol{G}$ be the obstacle-free space, so the path search constraint about flight path safety can be described as: $\mathbf{Tr} \subseteq \boldsymbol{X}_{\text{free}}$. Besides, a new reference waypoint should be determined under the UAV's maneuverability and sensor limits, so that the planner can achieve good performances on path tracking and smoothing. The formulations of $\boldsymbol{X}_{\text{free}}(t_k)$ ($\boldsymbol{X}_{\text{free}}$ at $\boldsymbol{w}_k$), $\mathbf{Tr}$, and the UAV's related performances are provided in Section 3. All above conditions, denoted by Res(ω, t), constitute together the path search constraint on decision variables in the BLP-based path planner.

To generate the control input $\boldsymbol{u}_k$ and the new flight state $\boldsymbol{s}_{k+1}$, η_{k+1} is firstly calculated to determine a feasible decision space for ω. Then Dis_{k+1} is calculated with the given decision space and constraints, as well as the changes and fluctuations of $\boldsymbol{c}_k$ and $\boldsymbol{\delta}_k$. In the end, the leader level measures all the possible decisions and provides an optimal or near-optimal solution to guide the UAV toward the target point. An adaptive decision is achieved in a sense that, by using BLP, the changes of sensor and UAV's performances are integrated in the two levels' objectives and constraints. A satisfactory solution, therefore, can always be obtained in a cooperative or competitive manner.

Accordingly, the BLP-based adaptive path planning can be formally described as:

$$\begin{aligned} &\min_{\omega} |\eta_{k+1}| \\ &\quad \min_{t} |\text{Dis}_{k+1}| \\ &\text{subject to } \text{Res}(\omega_k, t_k) \end{aligned} \tag{3}$$

where both the yaw rate $\omega_k \in \mathbf{R}$ and the flight time $t_k \in \mathbf{R}^+$ are subject to Res(ω, t) at $\boldsymbol{w}_k$, i.e., Res(ω_k, t_k), which provides necessary restrictions for path decisions of the two levels. The above description is used and thoroughly implemented in the subsequent section for adaptive path planner design.

3. BLP-based adaptive real-time path planner

3.1. Design of planner components

For the convenience of the adaptive path planner construction, relevant component definitions and assumptions including obstacles avoidance, sensory capability, maneuverability, flight velocity and time, are given in the first instance. They are also the important parts of the adaptability requirement and lay the foundation for Res(ω, t).

3.1.1. Obstacle description

Opposition obstacles such as radars or surface-to-air missiles (SAMs) in $\boldsymbol{G}$ may identify or shoot down a UAV when it enters the range of radiation or attack. Therefore, the obstacles must be modeled to construct the obstacle avoidance constraints. The threat degree to a UAV depends on the positions and strengths of the obstacles. We assume that the threat sources are SAMs whose resultant risk distribution is deterministically represented.

A deterministic risk model is considered based on probabilistic risk that has been adopted in many literatures and shown great applicability in path planning applications.[3,5,16–18,25] It must be pointed out that the path planner to be proposed is not restricted to the deterministic risk models. At the obstacle description stage, our planner can be easily employed to accommodate various stochastic events and polygonal types of risks in different environments.

Probabilistic risk is defined to quantificationally measure the threat degree of exposure to all the obstacles at position $\boldsymbol{w}$ ($\boldsymbol{w} \in \boldsymbol{G}$), and is denoted as $P(\boldsymbol{w})$. Suppose M is the number of SAM units in $\boldsymbol{G}$ with different hitting ranges (denoted as $R_{\{s,m,l\}}$), and $\boldsymbol{O}$ is the obstacle set. $\forall \boldsymbol{O}_i \in \boldsymbol{O}$ ($i = 1, 2, \ldots, M$), and its position is $\boldsymbol{L}_i = [x \ \ y]^{\mathrm{T}} \in \boldsymbol{G}$. Then, when a UAV reaches $\boldsymbol{w}$, $P(\boldsymbol{w})$ can be calculated by the following hit probability formula[5]:

$$P(\boldsymbol{w}) = 1 - \prod_{i=1}^{M} P_i(\boldsymbol{w}) \tag{4}$$

where $P_i(\boldsymbol{w})$ is the probabilistic risk of exposure to $\boldsymbol{O}_i$ at $\boldsymbol{w}$, and is described in Appendix A.

In the real-time path planning process, the number of detected obstacles at $\boldsymbol{w}$ does not exceed M because of the UAV's limited sensory ability. Let $\boldsymbol{O}(\boldsymbol{w})$ be the detected obstacle set and $M(\boldsymbol{w})$ the detected obstacle number, the UAV has to evaluate its real-time probabilistic risk at $\boldsymbol{w}$ (denoted as $P^{\mathrm{R}}(\boldsymbol{w})$) only according to $\boldsymbol{O}(\boldsymbol{w})$. In this case, $P^{\mathrm{R}}(\boldsymbol{w}) \leqslant P(\boldsymbol{w})$. $P^{\mathrm{R}}(\boldsymbol{w})$ can be calculated by:

$$P^{\mathrm{R}}(\boldsymbol{w}) = 1 - \prod_{i=1}^{M(w)} P_i(\boldsymbol{w}) \tag{5}$$

In order to guarantee the UAV's flight safety, the probabilistic risk threshold (denoted as ρ) is defined to indicate the minimal real-time probabilistic risk that the UAV can steer clear of the

hitting ranges of the detected obstacles. When the UAV reaches $\boldsymbol{w}_k$, the obstacle-free space is:

$$\boldsymbol{X}_{\text{free}}(t_k) = \{\boldsymbol{w} | P^{\text{R}}(\boldsymbol{w}) < \rho\} \tag{6}$$

The safety condition in a real-time planning mode can be described as:

$$\forall \boldsymbol{w}_k, k \in [0, N]: \quad \boldsymbol{w}_k \in \boldsymbol{X}_{\text{free}}(t_k) \tag{7}$$

and

$$\forall \mathbf{Tr}_{k,k+1}, k \in [0, N-1]: \quad \mathbf{Tr}_{k,k+1} \subseteq \boldsymbol{X}_{\text{free}}(t_k) \tag{8}$$

3.1.2. UAV's performances

Three related UAV's performances are systematically described to prepare for the path planner's hard constraint components in the path search process as follows.

3.1.2.1. Sensory capability. A UAV must detect the surrounding environment in real time during the flight toward a specified target. The UAV's sensory capability for obstacle detection directly affects the path planning result, and the success probability of the flight mission as well.

Consequently, the UAV's sensory capability should be incorporated in the path planner with initially unknown obstacles. We describe it by *detection range*, without regard to the dynamic variation of the airborne radar cross-section caused by the UAV's changing altitude angle. The detection range (denoted by $\boldsymbol{D}_{\boldsymbol{w}}$) indicates a constant and finite detectable area where the UAV's current position $\boldsymbol{w}$ is the center:

$$\boldsymbol{D}_{\boldsymbol{w}} = \{\boldsymbol{w}' | \|\boldsymbol{w}' - \boldsymbol{w}\| \leqslant R_{\text{d}}, \boldsymbol{w}' \in \boldsymbol{G}\} \tag{9}$$

where R_{d} ($R_{\text{d}} > 0$) is the circle radius (i.e., sensory radius).

R_{d} may change if either the environmental impact or the diverse detection equipment is taken into consideration. It is assumed that R_{d} keeps constant in one flight mission and may alter in another. As shown in Fig. 2, it is also assumed that R_{d} is so large that the UAV can detect the obstacles before flying into their hitting ranges, i.e., $\forall\ \boldsymbol{w} \in \boldsymbol{G}$ and $P^{\text{R}}(\boldsymbol{w}) < \rho$, and R_{d} satisfies:

$$R_{\text{d}} > \left|\left|\boldsymbol{w} - \boldsymbol{L}_i\right|\right| \quad (i = 1, 2, \ldots, M) \tag{10}$$

The above assumptions are necessary and reasonable. As shown in Fig. 2(a), the UAV enters the dangerous region (gray filled region) due to the obstacle's great strength and the UAV's limited detection range (dotted circles). Conversely, the dangerous region can be successfully avoided, as shown in Fig. 2(b).

3.1.2.2. Turning ability. A generated feasible path must be trackable for a UAV within its maneuverability limit. Let $\omega_{\max}$ ($\omega_{\max} > 0$) be the maximal change value of the heading angle per unit time, and then the yaw rate ω satisfies:

$$\omega \in [-\omega_{\max}, \omega_{\max}] \tag{11}$$

It is assumed that $\omega > 0$ if the heading angle changes clockwise, and $\omega = 0$ if it keeps unchanged. The turning ability should be incorporated in the path planner considering the path trackability. As shown in Fig. 3(a), three different flight paths starting from the UAV's current position $\boldsymbol{w}_0$ are generated and indicated by $\mathbf{Tr}_{0,\,i}$(i = 1, 2, 3). If the limit on ω is ignored, it is reasonable that $\mathbf{Tr}_{0,\,2}$ or $\mathbf{Tr}_{0,\,3}$ would be chosen as a reference trajectory, given the shortest distance to the target point or the proper flight direction at the point. However, the UAV with limited maneuverability would fail to proceed in practice, as shown in Fig. 3(b). The feasible maneuver range (gray filled region, denoted as $\boldsymbol{A}_k$) restricts the decision space, so that only $\mathbf{Tr}_{0,\,1}$ could be chosen.

Furthermore, the variation of $\omega_{\max}$ in diverse flights should also be handled in the path planner to strengthen its adaptability to maneuverability changes. For example, with possible weather deterioration in a long-distance flight mission, the turning ability may degrade due to unanticipated wind vectors and ice accumulation on pitot tubes or aerofoil.[2,19,41,42] The requirement of setting up sensitive loads or security-critical equipment would demand smooth trajectories all along.[1,25] As a result, the hard constraints in regard to turning ability and new reference waypoint decision-making can be described as follows:

$$\boldsymbol{w}_{k+1} \in \boldsymbol{A}(\boldsymbol{w}_k, \omega_{\max}) \tag{12}$$

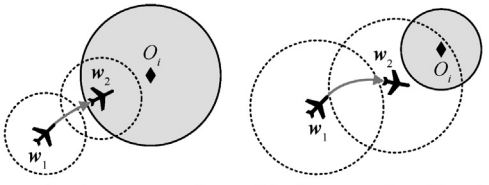
where $\boldsymbol{A}(\boldsymbol{w}_k, \omega_{\max})$ (abbreviated as $\boldsymbol{A}_k$, see Fig. 3(b)) indicates the UAV's feasible maneuver range at $\boldsymbol{w}_k$. With Eq. (11), the turning ability constraint can be further denoted as:

$$\boldsymbol{w}_{k+1} \in \boldsymbol{A}_k = \left\{ \boldsymbol{w} \middle| \boldsymbol{w} = \boldsymbol{w}_k + \left[\frac{2v_k}{|\omega|} \sin\left(\frac{|\omega t_k|}{2}\right) \sin\left(\theta_k + \frac{|\omega t_k|}{2}\right), \times \frac{2v_k}{|\omega|} \sin\left(\frac{|\omega t_k|}{2}\right) \cos\left(\theta_k + \frac{|\omega t_k|}{2}\right) \right]^{\text{T}} \right\} \tag{13}$$

3.1.2.3. Flight velocity and flight time. Let $v_{\min}$ and $v_{\max}$ denote a UAV's minimal and maximal flight velocities in the path planning process, then:

$$\forall v_k, k \in [0, N]: \quad v_{\min} \leqslant v_k \leqslant v_{\max} \tag{14}$$

Because in the real-time process the UAV (at $\boldsymbol{w}_k$) is unable to identify obstacles outside the detection range, the waypoint $\boldsymbol{w}_{k+1}$ to be determined should stay within $\boldsymbol{D}_{\boldsymbol{w}}$ to ensure the safety of the planned path. According to Eq. (9), it is known that the distance between $\boldsymbol{w}_k$ and $\boldsymbol{w}_{k+1}$ is shorter than R_{d}. Let t be the flight time between $\boldsymbol{w}_k$ and $\boldsymbol{w}_{k+1}$, and then t is



(a) Process of traversing dangerous region (b) Process of avoiding dangerous region

Fig. 2 Distinct processes of detection and flight based on different sensory capabilities and hitting ranges.

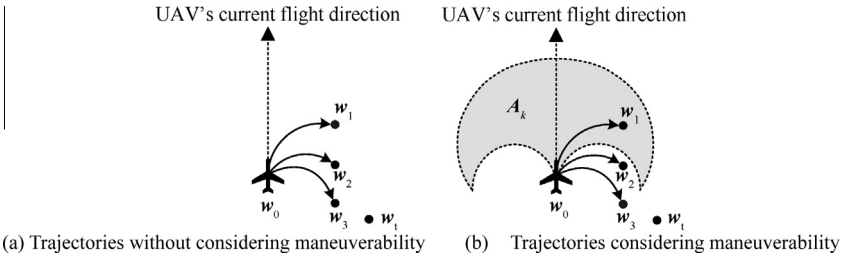


Fig. 3 Different trajectories in different flight direction changing modes.

a decision variable for path optimization in the planner, which satisfies:

$$t \in [0, R_d/v] \tag{15}$$

where $v \in [v_{min}, v_{max}]$, and v tends to take the maximal value in the flight process.

Although it is feasible for the planner to limit the flight velocity below v_{max}, the planner tends to make the UAV remain at the maximal velocity and subject to Eq. (13), so that it can reach the target as soon as possible. On the basis of a given velocity, the changed value of the heading angle after t should be smaller than 2π to prevent the UAV from flying back to the current waypoint. Then, the flight time from $\boldsymbol{w}_k$ to $\boldsymbol{w}_{k+1}$ must satisfy the following constraint:

$$\omega t \in (-2\pi, 2\pi) \tag{16}$$

Noting that if $\omega_{max} R_d/v < 2\pi$ (caused by the relatively low sensory and turning abilities), ωt inevitably satisfies the constraint in Eq. (16), so the actual value range of ωt is:

$$\omega t \in [-\omega_{max} R_d/v, \omega_{max} R_d/v] \tag{17}$$

3.2. Path planner structure

According to the design components based on obstacle avoidance and relevant performance constraints, a BLP-based adaptive and real-time path planner is constructed, as illustrated in Fig. 4.

Both the UAV's start and target states are imported at the start of the flight mission. The BLP-based path search is the key module of the planner that makes real-time decisions for it. The operational space, including the UAV model, obstacles, and its detection module, is integrated into the planner to form a closed-loop runtime environment.

In the BLP-based real-time path search module, objectives are used to calculate the new flight state s_{k+1} and path segment $\mathbf{Tr}_{k,k+1}$ subject to the constraints of $\mathrm{Res}(\omega_k, t_k)$ at t_k and capability fluctuation $\boldsymbol{\delta}_k$. The objective function definitions and operational principle have been demonstrated in Section 2. $\mathrm{Res}(\omega_k, t_k)$ is composed of four types of restrictions, with respect to obstacle avoidance, sensory capability, maneuverability, flight velocity and time, which have also been given in Section 3.1. $\boldsymbol{u}_k$ is the decision result in the kth planning step and further determines how the UAV flies toward the target from the current flight state.

In particular, $\boldsymbol{c}_k$ is embedded in $\mathrm{Res}(\omega_k, t_k)$, where it is invariable in the planning process for one flight mission. $\boldsymbol{\delta}_k$ is provided as a flight time function, and the uncertainty caused by it is addressed by the two levels' repetitive decision scheme. When making a new decision, the leader level optimizes the global convergence of the generated path, whereas the follower level enhances the local trend of approaching the target. The tradeoff decision between the two levels eliminates the uncertainty in a sense that the influence of $\boldsymbol{\delta}_k$ on one state is dispersed into the whole planning procedure. Besides, variable flight time (planning time interval) between two reference waypoints is conducive to the performance changes or

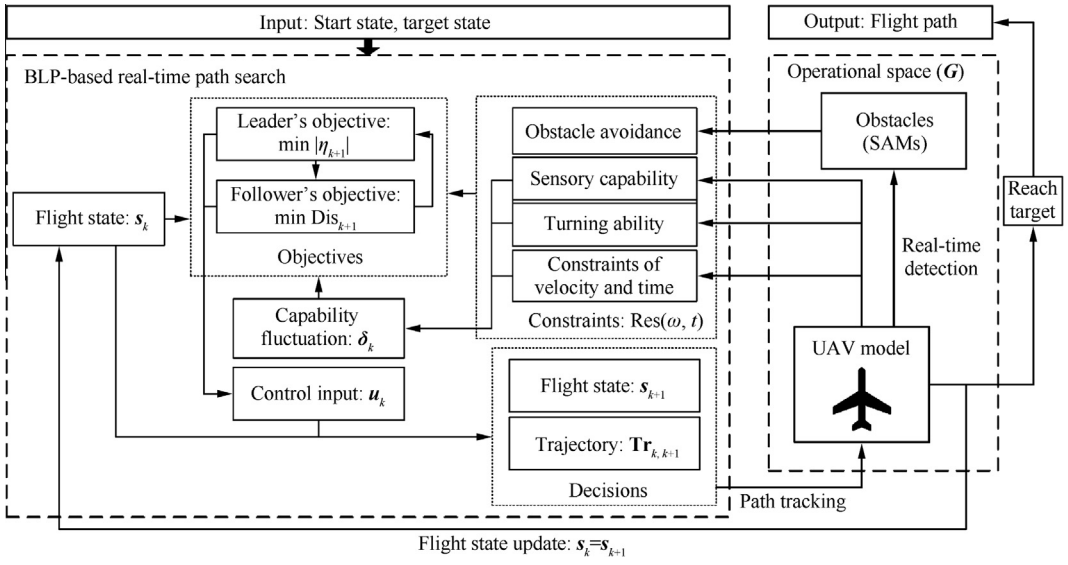


Fig. 4 BLP-based adaptive real-time path planner.

fluctuations, thanks to its adaptive value-taking as the follower's decision item.

For each decision and segmented trajectory, the planner estimates whether the new waypoint is close enough to the target point. If the UAV has not arrived at the target, a new iteration would be required; otherwise, the planner accomplishes its mission. A complete flight path is composed of necessary reference waypoints and flight directions at variable time intervals.

4. Path search solution

This section presents a highly effective discrete trajectory search algorithm for the aforementioned BLP-based path planner, in which several supplementary strategies are further developed and embedded to reduce the flight path decision space and then alleviate the computational burden.

4.1. Discrete search algorithm

The path planning problem has been analyzed and considered as PSPACE in the presence of obstacles,[9] where the computation of an exact solution is shown to be NP-hard.[39] Additionally, the BLP problems are intrinsically hard to solve, even the "simplest instance" of a linear BLP problem is strongly NP-hard.[40] In this case, most existing solution algorithms are not applicable to the proposed planner. However, those observational characteristics of the path planning process can provide valuable references for a novel solution algorithm of the planner.

On one hand, the calculation at the planner's each step is based on both the state transition equation of a discrete-time system and the two levels' objectives and constraints. We can further discretize the decision variables in their feasible sets (flight state and control input sets), so that the nonlinear continuous decision problem can be transformed into a discrete linear BLP-based decision problem.[33] Although the later leads to near-optimal solutions, it is relatively more tractable and insensitive to conversion error and discrete precision. On the other hand, the follower's decision-making can be appropriately simplified to reduce the inducible region. The goal of the planner is to timely provide a satisfactory decision favorable to both global and local path improvement. The reduced inducible region is still effective on the path optimization in global, because a temporary local decision of the follower level exerts little influence on the whole flight path with many flight states and control signals. Besides the above major factors, the implicit constraints on decision variables, derived from the indirect relationship between the UAV's diverse performances, are also worthy of full consideration in the search procedure. They induce further strategies to be presented in Section 4.2.

As a result, a near-optimal discrete trajectory search algorithm for the planner, based on BLP optimal solution definition[36,37] and flight process particularity, is summarized in Fig. 5:

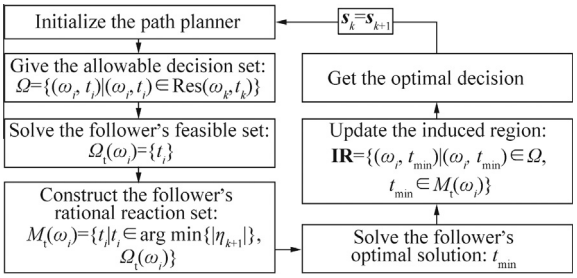


Fig. 5 BLP solution procedure for trajectory search.

Step 1 Initialize the path planner: input the UAV's start and target flight states into the planner. Initialize $k = 0$ and update $\boldsymbol{X}_{\mathrm{free}}(t_k)$ for the first time. Define **IR** as the inducible region of BLP and initialize it to an empty set. Based on $\mathrm{Res}(\omega, t)$, discretize the leader's and follower's decision variables as strictly monotonic increasing nonempty bounded sequences (ω_i) and (t_j), respectively.

Step 2 Give the problem's allowable decision set: $\forall\omega_i$, t_j, and $(\omega_i, t_j) \in \mathrm{Res}(\omega_k, t_k)$, give the allowable decision set Ω which includes all the possible solutions in the problem's decision space.

Step 3 Solve the follower's feasible set: For each element in (ω_i), ascendingly search all the elements in (t_j) and determine the follower's feasible set $\Omega(\omega_i)$ for each ω_i.

Step 4 Construct the follower's rational reaction set: For each element in (ω_i), the follower's rational reaction set $R_k(\omega_i)$ comprises the follower's feasible decisions that also belong to arg min $|\eta_{k+1}|$. The decision (ω_i, t_j) is rendered rational if ω_i meets $\mathrm{Res}(\omega_k, t_k)$ and $t_j \in R_k(\omega_i)$.

Step 5 Solve the follower's optimal solution: Based on $R_k(\omega_i)$ and the follower's decision objective, the follower's optimal solution is denoted by $t_{\min} = \min\{t_j\colon t_j \in R_k(\omega_i), (\omega_i, t_j) \in \Omega\}$.

Step 6 Update the inducible region: With $t_{\min}$, ascendingly search (ω_i) again. For each element in (ω_i), calculate the leader's objective function and determine whether $(\omega_i, t_{\min})$ satisfies $\mathrm{Res}(\omega_k, t_k)$. If $(\omega_i, t_{\min}) \in \mathrm{Res}(\omega_k, t_k)$, add it to **IR** and keep on searching.

Step 7 Get the optimal decision: ascendingly search **IR** to get the solution (control input) of the kth path search step to minimize $|\eta_{k+1}|$. Then, construct a new flight state $\boldsymbol{s}_{k+1}$ and flight path $\mathbf{Tr}_{k,\,k+1}$. Update the threat environment $\boldsymbol{X}_{\mathrm{free}}(t_k)$ and $k = k + 1$.

4.2. Optimization strategies

In the BLP-based path search process, traversals of various discrete sets are frequently performed. To reduce the search space in discrete sets and speed up the solution, the following optimization strategies are embedded.

(1) Division and accelerating strategy. When discretizing the leader's decision variable in Step 1, the discrete sequences (ω_i) can be divided into two subsequences: define $\Xi_1 = [-\omega_{\max}, 0]$, $\Xi_2 = [0, \omega_{\max}]$, and then constraints in Eqs. (7) and (12) can be denoted by $\omega \in \Xi_1 \cup \Xi_2$. Discretize Ξ_1 and Ξ_2 as two nonempty finite

subsequences of $(\omega_i)_1$ (which is strictly monotone increasing) and $(\omega_i)_2$ (which is strictly monotone decreasing), respectively. Based on the above division, the traversal of (ω_i) in Steps 3 and 4 can be divided into two parts: the traversal of $(\omega_i)_1$ indicating a counterclockwise search process for a yaw rate within $\boldsymbol{A_k}$ from $\omega = 0$; and the traversal of $(\omega_i)_2$ indicating a clockwise search process for a yaw rate within $\boldsymbol{A_k}$ from $\omega = 0$. Once a satisfactory decision of the leader level is obtained in any traversal, the other will be immediately abandoned.

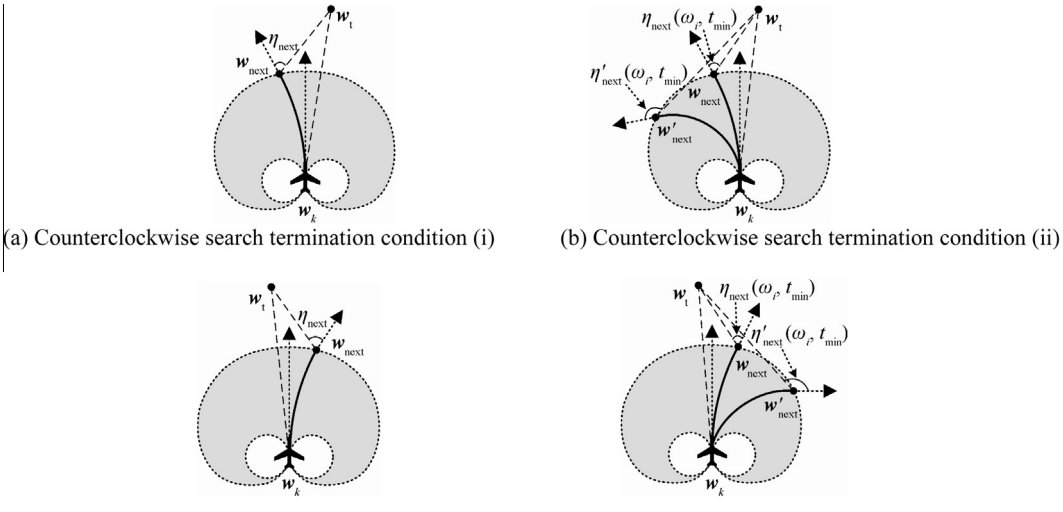
(2) Dynamic termination strategy. The traversal of (ω_i) can also be divided and performed like (1) in order to efficiently construct the inducible region in Step 6. The search in $(\omega_i)_1$ will be immediately terminated on condition that: (i) if $\eta_k \leqslant 0$ and $\eta_{\text{next}} \leqslant 0$ (as illustrated in Fig. 6(a)), where η_{next} is the **ANGLE** between the target and any point w_{next} within $\boldsymbol{A}_k$; or (ii) if $-\pi < \eta_k < \pi$ and the length of **IR** is not zero, the search will be terminated once the value of $|\eta_{\text{next}}|$ increases, i.e., $|\eta'_{\text{next}}(\omega_i, t_{\min})| > |\eta_{\text{next}}(\omega_i, t_{\min})$, as illustrated in Fig. 6(b). Similarly, the search in $(\omega_i)_2$ will be terminated immediately on condition that: (i) if $\eta_k \geqslant 0$ and $\eta_{\text{next}} \geqslant 0$, as illustrated in Fig. 6(c); or (ii) if $-\pi < \eta_k < \pi$ and the length of **IR** is not zero, the search will be terminated once the value of $|\eta_{\text{next}}|$ (where $t = t_{\min}$) increases, as seen in Fig. 6(d).

(3) Interruption strategy. In Step 6, whether the elements in $(\omega_i)_1$ and $(\omega_i)_2$ satisfy both $\boldsymbol{A}_k$ and obstacle avoidance constraints should be determined by a new traversal of (t_i). According to the variation law of η_{k+1} with t, the search process in (t_i) can be interrupted under some special conditions of counterclockwise and clockwise search state changes. Let 0, $-\pi$, and π denote $\eta_{k+1} = 0$, $\eta_{k+1} = -\pi$, and $\eta_{k+1} = \pi$, respectively. Let $-$ and $+$ denote the η_{k+1} value ranges of $-\pi < \eta_{k+1} < 0$ and $0 < \eta_{k+1} < \pi$, respectively. We then give the potential interruption conditions:

(a) In a counterclockwise search, for each element in $(\omega_i)_1$, the changing trend of η_{k+1} can be classified into four types with a strictly monotone increase of t: (i) if $-\pi < \eta_k < 0$, the change of η_{k+1} is: $- \to -\pi \to +$; (ii) if $\eta_k = 0$, the change is: $0 \to - \to -\pi \to +$; (iii) if $0 < \eta_k < \pi$, the change is: $+ \to 0 \to - \to -\pi \to +$; (iv) if $\eta_k = \pi$, the change is: $-\pi \to - \to 0$. Under the circumstances of (i), (ii), and (iii), the search should be interrupted if $\eta_{k+1} = -\pi$, because $|\eta_{k+1}|$ takes the maximum value in its feasible range; When (iv) takes place, the search should be interrupted if $\eta_{k+1} = 0$, because $|\eta_{k+1}|$ takes the minimum value in its feasible range.

(b) Likewise, in a clockwise search, for each element in $(\omega_i)_2$, the changing trend of η_{k+1} can also be classified into four types with a strictly monotone increase of t: (i) if $0 < \eta_k < \pi$, the change of η_{k+1} is: $+ \to \pi \to -$; (ii) if $\eta_k = 0$, the change is: $0 \to + \to \pi \to -$; (iii) if $-\pi < \eta_k < 0$, the change is: $- \to 0 \to + \to \pi \to -$; (iv) if $\eta_k = -\pi$, the change is: $\pi \to + \to 0$. Under the circumstances of (i), (ii), and (iii), the search should be interrupted if $\eta_{k+1} = \pi$, because $|\eta_{k+1}|$ takes the maximum value in its feasible range; When (iv) takes place, the search should be interrupted if $\eta_{k+1} = 0$, because $|\eta_{k+1}|$ drops to the minimum from the maximum value in its feasible range.

(c) Safety improvement strategy. It is worth noting that the continuous variable discretization precision influences path safety. Low precision may result in a problem that the probability risk is smaller than ρ at a certain discrete point, but larger than ρ at those sampling points, on the line connecting two discrete points. That is to say, there is no guarantee that the continuous path does not intersect with the obstacles. Therefore, we present a reduction coefficient (denoted as u,



(a) Counterclockwise search termination condition (i)

(b) Counterclockwise search termination condition (ii)

(c) Clockwise search termination condition (i)

(d) Clockwise search termination condition (ii)

Fig. 6 Termination conditions.

$0 < u \leqslant 1$) of the probabilistic risk threshold to address this problem. In the solving process, ρ will change into $u\rho$ to improve the success probability of obstacle avoidance.

5. Simulation results

In order to validate the feasibility and adaptability of the BLP-based real-time path planner for UAVs, numerical simulations are carried out in a great number of random scenarios and several representative scenarios used in many motion planning literatures.[3,16–18] Comparisons are also drawn between our planner and other classical methods[10,14,16,18] to highlight the advantages of ours in optimization and adaptability. All simulations are performed on a Microsoft visual studio platform using a 3.4 GHz CPU.

5.1. Feasibility validation

Two representative scenarios including many different SAMs[16–18] (denoted as S_A and S_B) are adopted to evaluate the effectiveness and efficiency of the proposed planner. The values of start position, target position, initial flight velocity, and initial heading angle are taken as $\boldsymbol{w}_s = [20\ 20]$ km, $\boldsymbol{w}_t = [180\ 180]$ km, $v_s = 50$ m/s, and $\theta_s = \pi/4$ (pointing to the target), respectively. Other parameters are given in Table 1.

The generated flight paths are shown in Fig. 7. We can see that the proposed planner generates convergent and feasible flight paths without many unnecessary twists and turns. In S_A and S_B, the path lengths are 252.5 km and 321.5 km, and the peak risks of exposure to all obstacles (max $P(\boldsymbol{w})$) are 0.0948 and 0.0919, respectively, which means the UAV is able to be kept safe in the whole flight processes. Besides, the planner calculates control inputs and flight states only 17 and 25 times in the very large operational spaces of S_A and S_B. Even in a highly complex scenario[16] shown in Fig. 8, the planner can escape from a *double bug trap* and generate an optimal flight path.

Table 1 Parameters of the proposed path planner.

Parameter	ω_{max}(rad/s)	R_d(km)	ρ	u	v_{min}(km/s)	v_{max}(km/s)
Value	$\pi/60$	40	0.1	0.9	0.01	0.05

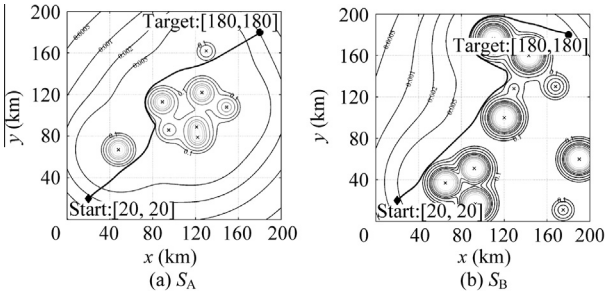


Fig. 7 Generated flight paths in representative scenarios.

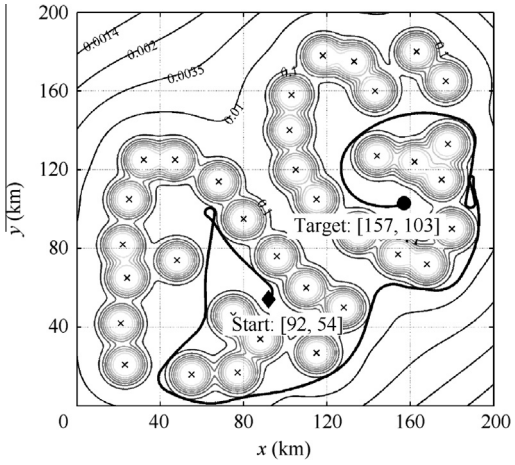


Fig. 8 Generated flight path in a highly complex scenario including a double bug trap.

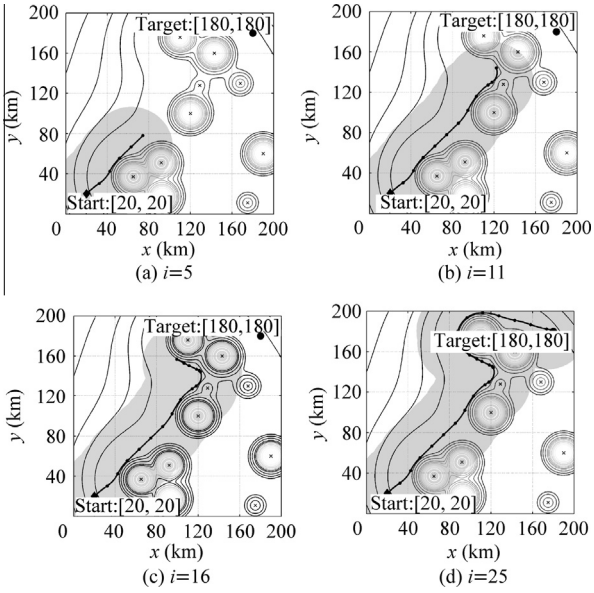


Fig. 9 Real-time path planning process in S_B.

The planning process in S_B with two big traps is shown in Fig. 9. In Fig. 9(a), the UAV fails to detect the distant obstacles in the fifth step, so the planned path is directed toward the target without regard to unknown obstacles. In Fig. 9(b), more obstacles are detected and the planned path keeps away from a trap. After that, all the obstacles in the surrounding area are detected, as shown in Fig. 9(c) and (d), and the subsequent path steers clear of obstacles all along. The adaptive decision-making frequency guarantees both the safety and optimization of the path.

To further test our planner's adaptability to diverse complex threat environments, 100 scenarios are randomly generated. Each scenario incorporates eight SAMs deployed within a very large operational space $\boldsymbol{G} = [0, 200]$ km $\times$ [0, 200] km. The center positions and hitting ranges of the SAMs are subjected to a uniform distribution. For the sake of a consistent evaluation standard, the corresponding parameters take

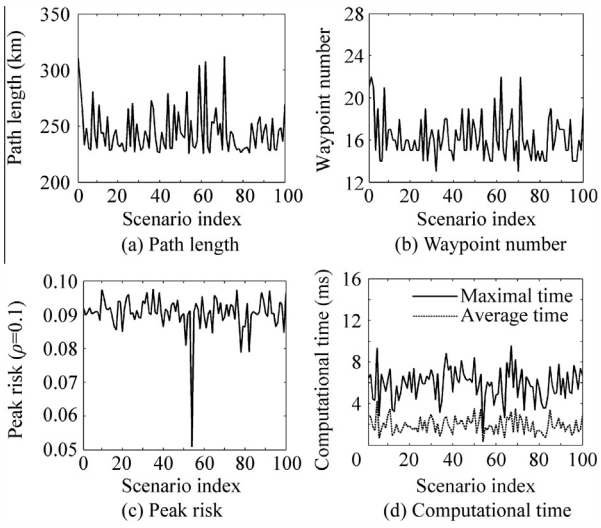


Fig. 10 Path planning results in 100 stochastic scenarios.

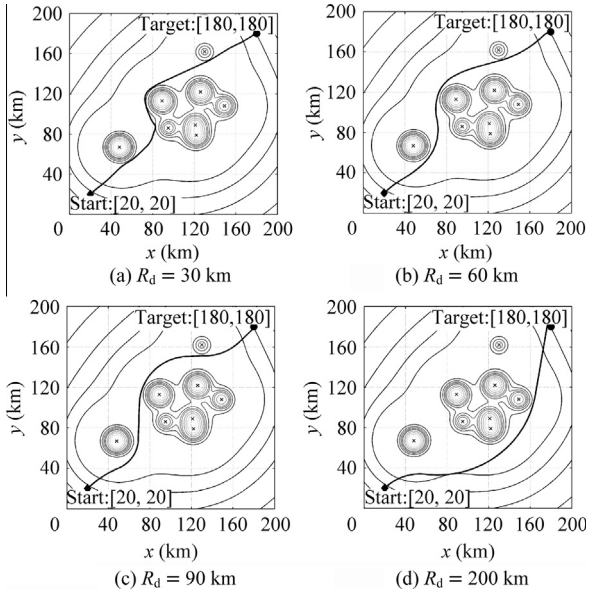


Fig. 11 Generated paths with different sensory capabilities.

the same values as in the aforementioned simulations in Section 5.1.

The statistical results in regard to path length, peak risk, total number of reference waypoints, and computational time of making a decision (calculate a new control input and flight state), are shown in Fig. 10. In all scenarios, our planner can provide convergent and safe paths with short flight distances and a small quantity of reference waypoints (the average length and waypoint number are 244.9 km and 16, respectively). Considering the variable flight time between any two waypoints (belong to [13, 320] s), the maximal and average computational time to make a decision are only 9.59 ms and 4.25 ms among scenarios, which completely meet the real-time planning requirement (the ranges of the maximal and average computational time are [1.14, 9.59] ms and [0.34, 4.25] ms, respectively).

5.2. Adaptability to performance variations

The challenging scenario S_A, with a large dangerous region and two independent obstacles between the start position and the target, is suitable to test the adaptability when the UAV's related performance constraints change or fluctuate.

5.2.1. Different sensory capabilities

With the UAV's increasing sensory capability, the obtained information in a certain location also increases. In particular, if $\boldsymbol{G} \subseteq \boldsymbol{D}_w$, the real-time path planning problem will be translated into a static one due to the completely known threat environments. When R_d changes from 30 km to 200 km, the generated flight paths are illustrated in Fig. 11. All paths are safe and smooth whether R_d is strictly limited or not (when R_d = 200 km, $\boldsymbol{G} \subseteq \boldsymbol{D}_w$).

Quantitative results are given in Table 2. The total number of reference waypoints decreases significantly because a larger R_d leads to fewer planning executions. Similarly, with more obtained information, the path safety is obviously improved through more rational obstacle avoidance and a little longer flight distance. Besides, the relatively stable and short computational time to make a decision decreases the overall computational burden.

5.2.2. Different turning abilities

When ω_{max} changes from $\pi/30$ to $\pi/1080$ rad/s, the feasible maneuver range $\boldsymbol{A}_k$ and the corresponding planning results are illustrated in Figs. 12 and 13, respectively. The gray filled regions indicate $\boldsymbol{A}_k$ with representative values when the UAV is at $\boldsymbol{w}_k$. Despite the sharp distinction, the impact of different turning abilities on the generated flight paths is insignificant.

Table 2 Planning results with different values of R_d.

R_d (km)	Path length (km)	Peak risk	Waypoint number	Maximal time (ms)	Average time (ms)
30	251.5	0.0989	23	7.07	2.27
40	252.5	0.0948	17	7.11	2.59
50	247.8	0.0877	13	7.04	2.73
60	248.3	0.0847	12	8.29	4.23
70	252.5	0.0817	11	5.82	3.87
80	255.4	0.0839	9	6.02	3.79
90	252.4	0.0816	9	7.12	4.49
100	262.1	0.0736	8	6.51	3.92
200	263.7	0.0734	5	4.19	3.30

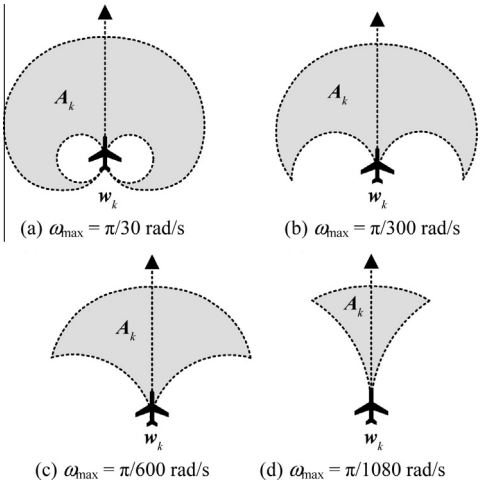


Fig. 12 Feasible maneuver ranges with different turning abilities (gray filled region).

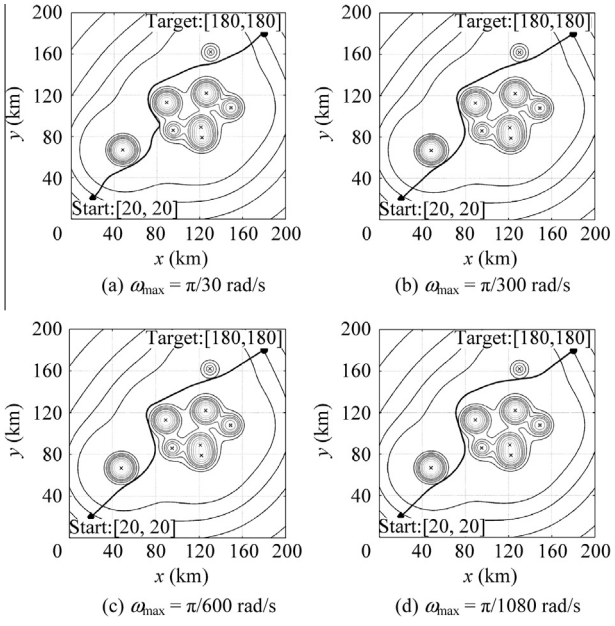


Fig. 13 Generated paths with different turning capabilities.

Quantitative results are given in Table 3. All the evaluation items remain stable as the turning ability decreases to a great degree. Their standard deviations are 12.1 km, 0.0031, 10.8 ms (maximal time), and 3.41 ms (average time), respectively, which means that the planner can adapt all the real-time decisions to the UAV's maneuverability variations and timely provide optimal flight paths without remarkable differences.

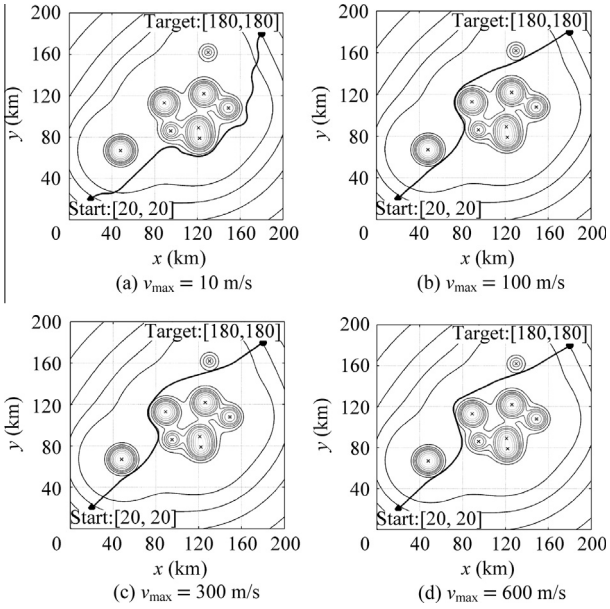


Fig. 14 Generated paths with different velocity constraints.

5.2.3. Different velocity constraints

When v_{max} changes from 10 m/s to 600 m/s, the generated flight paths are illustrated in Fig. 14. If the velocity is limited within a small range ($v = 10$ m/s), the UAV is more agile with the same maneuverability and sensory capability, so that the decision space is larger and the planned path may include many minor zigzags, as shown in Fig. 14(a). Except for that, the planned paths are all smooth, as shown in Fig. 14(b)–(d).

Quantitative results are given in Table 4. When $v_{max} < 50$ m/s, the flight distances are a little longer, but the paths are safer due to a larger decision space of the planner. The planner adapts the flight velocity to the velocity constraints and timely plans flight paths with stable quality and average computational time.

5.2.4. Performance fluctuation

Considering the influence of the random disturbances on R_d, ω_{max}, and v_{max} at the kth stage of a planning process, $\boldsymbol{\delta}_k$ is superimposed on the corresponding properties to test the planner's adaptability. For each component of $\boldsymbol{\delta}_k$, the path planning is independently carried out 100 times. Typical flight

Table 3 Planning results with different values of ω_{max}.

ω_{max} (rad/s)	Path length (km)	Peak risk	Waypoint number	Maximal time (ms)	Average time (ms)
$\pi/30$	254.1	0.0964	17	3.15	1.21
$\pi/60$	252.5	0.0948	17	7.11	2.59
$\pi/120$	250.6	0.0925	18	11.48	3.94
$\pi/240$	251.5	0.0934	17	9.41	4.16
$\pi/480$	254.1	0.0933	16	13.29	3.91
$\pi/720$	258.4	0.0938	16	10.95	4.10
$\pi/1000$	254.0	0.0937	16	16.85	5.35
$\pi/1080$	264.7	0.0938	16	9.32	4.71

Table 4 Planning results with different values of v_{max}.

v_{max} (m/s)	Path length (km)	Peak risk	Waypoint number	Maximal time (ms)	Average time (ms)
10	267.9	0.0817	19	2.45	1.26
20	255.3	0.0898	17	4.04	1.59
50	252.5	0.0948	17	7.11	2.59
100	250.6	0.0925	18	10.85	3.67
200	251.0	0.0933	16	7.67	3.49
340	253.9	0.0943	15	10.93	2.77
400	254.5	0.0941	15	11.50	2.88
500	251.4	0.0941	16	7.60	2.41
600	252.6	0.0936	16	8.27	1.74

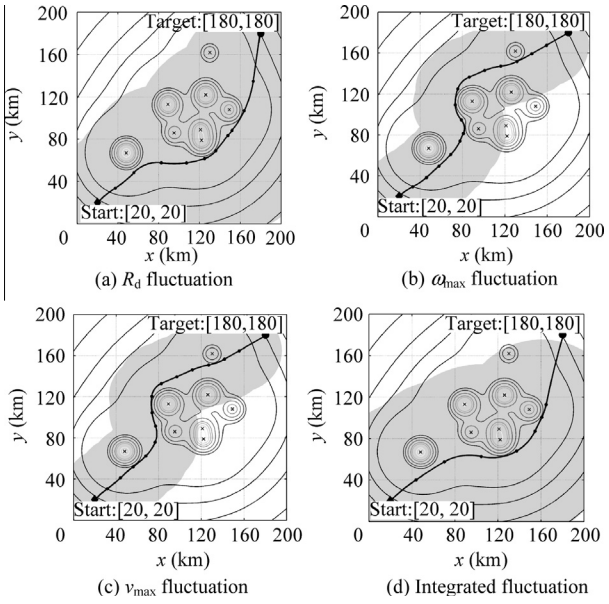


Fig. 15 Generated paths with performances fluctuation.

paths with the introduction of $\boldsymbol{\delta}_k$ are exhibited in Fig. 15(a)–(c), where the performance fluctuation ranges of R_d, ω_{max}, and v_{max} are [40,100] km, [π/1200, π/30] rad/s and [50, 100] m/s, respectively. The path synthetically affected by all the three fluctuations is exhibited in Fig. 15(d), where the grayed regions denote the real-time detected areas. We can see that satisfactory paths are all successfully generated, whether the three violent disturbances impact the planner independently or jointly.

Furthermore, the means and standard deviations of the 100 results, with regard to path length, peak risk, total number of reference waypoints, and computational time to make a decision, are shown in Table 5. Based on the small values of the standard deviations, we know that the planner can be adaptively adjusted to the disturbances and fluctuations that exist in R_d, ω_{max}, and v_{max}, and then provide optimal flight paths with a stable operational performance.

5.3. Comparison analysis

We compare the simulation results obtained from our planner with those from two classical methods of a RRT-based algorithm[4,10] and a Dijkstra-based PRM algorithm,[14] and two novel methods of a bouncing based A_{2d} algorithm[16] and a (compositional rule of inference (CRI)) based algorithm.[18] Quantificational results of the five key measurements in typical scenarios are listed in Table 6, and the generated paths are shown in Fig. 7 and Fig. 16. All the parameters take the same values as in the simulations in Section 5.1, and R_d = 40 km. In particular, considering the stochastic ideas of RRT and PRM, their results are the average values of 1000 implements.

It is manifest that the path obtained from our planner is shorter and smoother than those from others, and our planner generates only 25 and 17 references waypoints, making the planning results more applicable for realistic flight tasks. Despite the computational time is longer than RRT, A_{2d}, and CRI (whereas much shorter than PRM), the BLP can still timely provide decisions whenever needed. Moreover, the peak risks of the paths from CRI exceed ρ a little due to ignorance of the hard constraint of safety, whereas our planner can fully describe it through the BLP constraints and provide a lower threat degree (0.0919 and 0.0948) in the whole flight. Although the path quality obtained from PRM is superior in length (S_A) and peak risk (S_B), the enormously long computational time (caused by its two-phase planning idea) makes the method practically infeasible in real-time missions.

Additional comparisons are drawn in the representative scenario S_A to verify better adaptability of our planner than the other four methods, as shown in Fig. 17. If some of the UAV's performances change to their critical values, such as R_d < 40 km, ω_{max} < π/300 rad/s, and v > 150 m/s, the BLP-based planner is always capable of generating feasible or

Table 5 Planning results with $\boldsymbol{\delta}_k$.

Fluctuation item	Path length (km)	Peak risk	Waypoint number	Maximal time (ms)	Average time (ms)
R_d fluctuation	253.5/4.1	0.0842/0.0069	12/1	8.53/1.30	4.11/0.62
ω_{max} fluctuation	252.7/3.80	0.0927/0.0007	17/1	8.44/1.99	2.75/0.41
v_{max} fluctuation	249.9/13.2	0.0928/0.0029	17/1	7.55/0.66	2.94/0.25
Integrated fluctuation	256.8/14.9	0.0875/0.0018	11/1	11.99/3.15	5.10/0.86

Note: −/− corresponds to means/standard deviations.

Table 6 Planning results from five different methods.

Scenario	Method	Path length (km)	Peak risk	Waypoint number	Maximal time (ms)	Total time (ms)
S_A	BLP	321.5	0.0919	25	11.55	133.54
	RRT	540.8	0.0992	135	9.80	264.99
	PRM	265.5	0.0926	36	575.44	10004.40
	A_{2d}	2312.1	0.0964	1974	3.31	102.46
	CRI	328.0	0.1016	329	0.27	22.96
S_B	BLP	252.5	0.0948	17	14.58	85.42
	RRT	355.9	0.0989	89	14.34	220.67
	PRM	302.7	0.0631	40	76.73	1634.67
	A_{2d}	258.3	0.0984	249	1.08	39.46
	CRI	257.0	0.0982	258	0.26	21.64

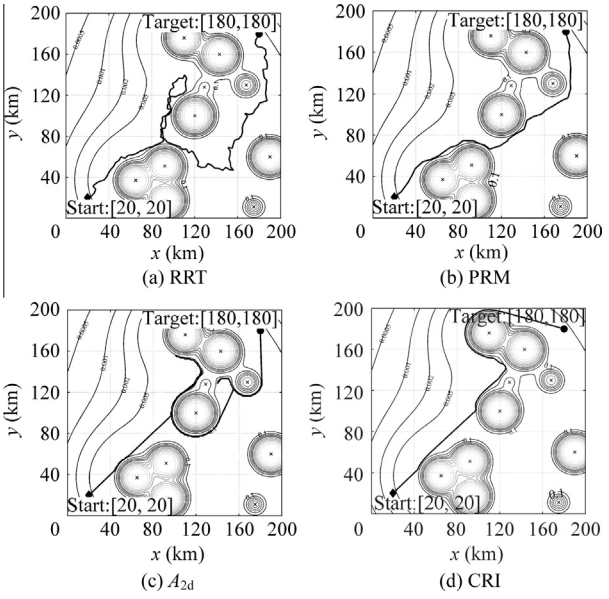


Fig. 16 Generated paths from four methods in S_B.

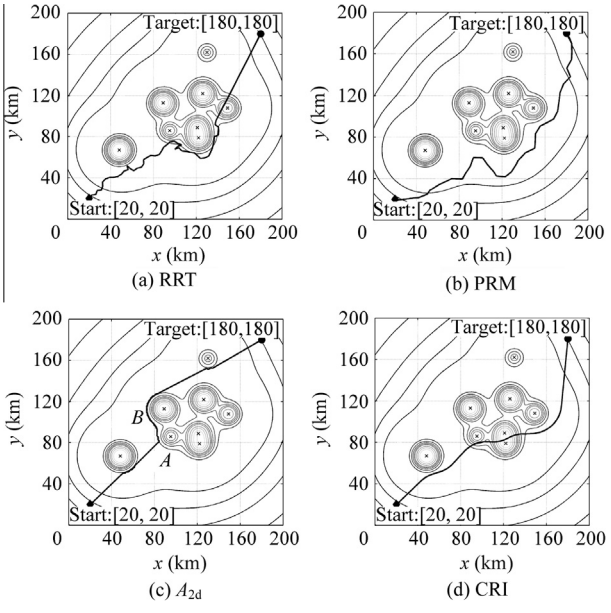


Fig. 17 Generated paths from four different methods.

near-optimal paths, as shown in Fig. 11 and Figs. 13–15. However, the unpractical paths from RRT and CRI traverse dangerous regions because of the UAV's high speed and poor detection, which means failure to arrive at the target, as shown in Fig. 17(a) and (d). The A_{2d} algorithm has great effects with limited sensors, but under the integrated preference constraints of R_d, ω_{max}, and v, it can only generate a path containing a segment of dangerous trajectory $\mathbf{Tr}_{A,B}$ (the peak risk is 0.1079), as shown in Fig. 17(c). When the value of R_d is especially small (R_d = 20 km/s), PRM has to frequently re-plan new paths (six times and 5.90 s in Fig. 17(b)) to avoid newly detected obstacles, which drastically increases the time burden and further limit the method's real-time implementation when the flight velocity v is very high (v > 300 m/s). In conclusion, the BLP-based planner shows better adaptability in most evaluations of the real-time path planning when the performances change or fluctuate.

Except the path optimization and adaptability advantages of the planner, there are two deficiencies in real-time applications. On one hand, the computational complexity is higher than those based on sampling or bouncing algorithms, so the computational time to get a reference waypoint and control input is relatively long. Even so, the extremely low ratios of computational time to planning intervals (available flight time between two waypoints) can guarantee real-time flights in practice. On the other hand, the convergence or completion of the BLP model is hard to be theoretically guaranteed due to the intrinsically NP-hard property and the possible sensory capability degradation. One feasible improvement is introducing the technique of model predictive control (MPC)[12] into the BLP-based path search process. As a result, it is able to predict future behaviors of the UAV and help to generate a holistic trajectory to the goal in each planning step.

6. Conclusions and future work

This paper puts forward an initial thought in regard to a BLP-based planner for UAVs' adaptive real-time path planning. We anticipate in-depth research on it in the future. The main contributions of the study are listed below:

(1) Adaptability description of path planning. Considering the changes or fluctuations of a UAV's performances in different flight missions or at different stages of a mission, the adaptability plays a crucial part in real-time path planning. We introduce its description in the proposed planner to meet more planning requirements in practice.

(2) BLP-based real-time path planner. A novel path planning idea based on BLP is adopted in the adaptive planner to generate reference waypoints and control inputs at variable planning time intervals. It can address the performance variations and adaptively plan smooth flight paths only when necessary.
(3) Discrete flight path search algorithm for the BLP-based planner. The characteristics of BLP and path planning procedure are both considered in the planner's implementation process, so as to construct a feasible and efficient discrete search algorithm for the adaptive planner.

Some directions are possible for further works. An immediate work will be dedicated to the motion planning problem in three-dimensional operational space by introducing MPC and new variables, such as flight height, pitch angle, and rolling angle. Another valuable research is to address more uncertainties caused by airborne equipment or external disturbances, and adaptively plan optimal flight paths in dynamic and uncertain environments (DUEs).

Acknowledgements

This study was supported by the National Natural Science Foundation of China (No. 60904066). Many thanks also go to the anonymous reviewers for their critical and constructive reviews of the manuscript, as well as Ms. Xin Liu for her language polishing efforts.

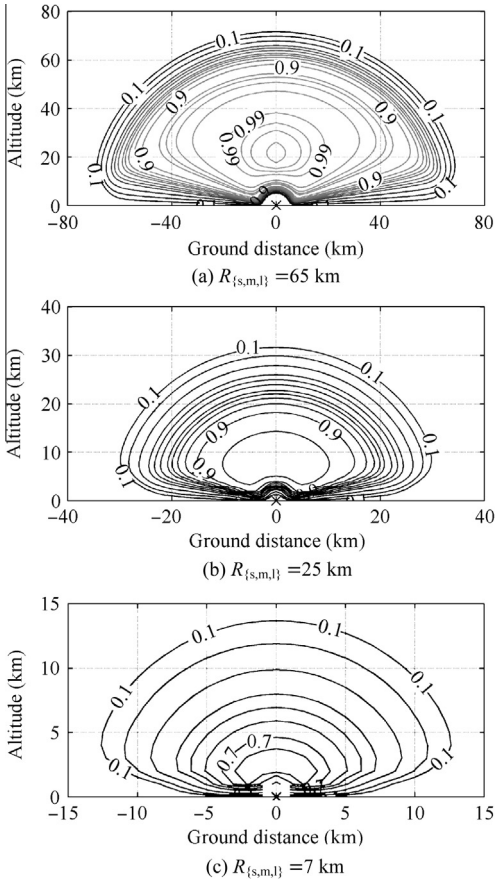


Fig. A1 Probability risk (hit probability) distributions.

Appendix A. Obstacle model

The probability risk based obstacle model is defined to describe a SAM threat distribution (hit probability distribution).[5] The risk of exposure to the ith SAM at $\boldsymbol{w}$ is calculated based on the following formula:

$$P_i(\boldsymbol{w}) = [1 - \mathrm{Step}(d_i, R_{\{s,m,l\}}, k_1)] \cdot \mathrm{Step}(d_i, 0.1 \cdot R_{\{s,m,l\}}, k_2) \cdot \mathrm{Step}\left(\arcsin\frac{h}{d_i}, \gamma, k_3\right) \quad \text{(A1)}$$

where the function $\mathrm{Step}(a, a_0, k)$ is defined by:

$$\mathrm{Step}(a, a_0, k) = \frac{1}{2}\left(1 + \frac{a - a_0}{\sqrt{k^2 + (a - a_0)^2}}\right) \quad \text{(A2)}$$

and, i is the SAM index, $d_i = \|\boldsymbol{w} - \boldsymbol{L}_i\|$ the ground distance between $\boldsymbol{w}$ and the ith SAM, k_i (i = 1,2,3) the softness parameters of the function $\mathrm{Step}(a, a_0, k)$, h the flight altitude of the UAV, γ the lowest coverage angle of the airborne radar (assumed to be 0.17 rad), and $R_{\{s,m,l\}}$ the hitting range. The probability risk distribution of a single SAM taking the $R_{\{s,m,l\}}$ values of *short* (7 km), *medium* (25 km), and *long* (65 km) are illustrated in Fig. A1.

References

1. United States Air Force. Unmanned aircraft systems flight plan 2009–2047. Unmanned Systems Special Report. Washington D.C.: Headquarters, United States Air Force; 2009 May.
2. Weatherington D. Unmanned aircraft systems roadmap 2005–2030. Final report. Report No.: 05-S-1620. Washington D.C.: Office of the Secretary of Defense; 2005 Aug.
3. Gu DW, Postlethwaite I, Kim Y. A comprehensive study on flight path selection algorithms. In: *The IEE seminar on target tracking: algorithms and applications*, Birmingham, UK. Piscataway: Institution of Electrical Engineers; 2006. p. 77–90.
4. LaValle SM. *Planning algorithms*. Cambridge, UK: Cambridge University Press; 2006.
5. Gu DW, Kamal W, Postlethwaite I. A UAV waypoint generator 2002. Report No.: AIAA-2004-6227.
6. Bortoff SA. Path planning for UAVs. In: *Proceedings of the American control conference*, 2000 Jun 28–30, Chicago, IL, USA. Piscataway: IEEE; 2000. p. 364–8.
7. Tarjan RE. A unified approach to path problems. *J Assoc Comput Mach* 1981;**28**(3):577–93.
8. Dogan A. Probabilistic approach in path planning for UAVs. In: *Proceedings of IEEE international symposium on intelligent control*, 2003 Oct 5–8, Houston, TX, USA. Piscataway: IEEE; 2003. p. 608–13.
9. Canny JF. *The complexity of robot motion planning*. Cambridge: MIT Press; 1988.
10. LaValle SM, Kuffner Jr JJ. Randomized kinodynamic planning. *Int J Rob Res* 2001;**20**(5):378–400.
11. Karaman S, Walter MR, Perez A, Frazzoli E, Teller S. Anytime motion planning using the RRT*. In: *Proceedings of IEEE International conference on robotics and automation*, 2011 May 9–13, Shanghai, China. Piscataway: IEEE; 2011. p. 1478–83.
12. Schouwenaars T. Safe trajectory planning of autonomous vehicles dissertation. Boston: Massachusetts Institute of Technology; 2006.
13. Alidaee B, Wang HB, Landram F. A note on integer programming formulations of the real-time optimal scheduling and flight path selection of UAVs. *IEEE Trans Control Syst Technol* 2009;**17**(4):839–43.

14. Kavraki LE, Švestka P, Latombe JCC, Overmars MH. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans Rob Autom* 1996;**12**(4):566–80.
15. Ren J, McIsaac KA, Patel RV, Peters TM. A potential field model using generalized sigmoid functions. *IEEE Trans Syst Man Cybern Part B Cybern* 2007;**37**(2):477–84.
16. Kim Y, Gu DW, Postlethwaite I. Real-time path planning with limited information for autonomous unmanned air vehicles. *Automatica* 2008;**44**(3):696–712.
17. Wu SJ, Zheng Z, Cai KY. Real-time path planning for unmanned aerial vehicles using behavior coordination and virtual goal. *Control Theory Appl* 2011;**28**(1):131–6 [Chinese].
18. Zheng Z, Wu SJ, Liu W, Cai KY. A feedback based CRI approach to fuzzy reasoning. *Appl Soft Comput* 2011;**11**(1):1241–55.
19. Rubio JC, Vagners J, Rysdyk R. Adaptive path planning for autonomous UAV oceanic search missions; 2004. Report No.: AIAA-2004-6228.
20. Ducard G, Kulling KC, Geering HP. A simple and adaptive online path planning system for a UAV. In: *2007 Mediterranean conference on control and automation*, 2007 Jul 27–29, Athens, Greece. Piscataway: IEEE; 2007. p. 1–6.
21. Meister O, Frietsch N, Ascher C, Trommer GF. Adaptive path planning for VTOL-UAVs. *IEEE Aerosp Electron Syst Mag* 2009;**24**(7):36–41.
22. Dong ZN, Zhang RL, Chen ZJ, Zhou R. Study on UAV path planning approach based on fuzzy virtual force. *Chin J Aeronaut* 2010;**23**(3):341–50.
23. Fu XW, Gao XG. Genetic algorithm with adaptive immigrants for dynamic flight path planning. In: *IEEE International conference on intelligent computing and intelligent systems*, 2010 Oct 29–31, Xi'an, China. Piscataway: IEEE; 2010. p. 630–4.
24. Duan HB, Zhang XY, Wu J, Ma GJ. Max-min adaptive ant colony optimization approach to multi-UAVs coordinated trajectory replanning in dynamic and uncertain environments. *J Bionic Eng* 2009;**6**(2):161–73.
25. Liu W, Zheng Z, Zhao LM, Cai KY. Smooth path planning in online mode for unmanned air vehicles. In: *Proceedings of IEEE conference on industrial electronics and applications*, 2011 Jun 21–23, Beijing, China. Piscataway: IEEE; 2011. p. 2599–604.
26. Ferguson D, Stentz A. Anytime RRTs. In: *Proceedings of IEEE/RSJ international conference on intelligent robots and systems*, 2006 Oct 9–15, Beijing, China. Piscataway: IEEE; 2006. p. 5369–75.
27. van den Berg J, Ferguson D, Kuffner JJ. Anytime path planning and re-planning in dynamic environments. In: *Proceedings of IEEE international conference on robotics and automation*, 2006 May 15–19, Orlando, FL, USA. Piscataway: IEEE; 2006. p. 2366–71.
28. Sujit B, Beard R. Multiple UAV path planning using anytime algorithms. In: *Proceedings of American control conference*, 2009 Jun 10–12, Hyatt Regency Riverfront, St. Louis, MO, USA. Piscataway: IEEE; 2009. p. 2978–83.
29. Bracken J, McGill JT. Mathematical programs with optimization problems in the constraints. *Oper Res* 1973;**21**(1):37–44.
30. Candler W, Norton R. Multilevel programming. Report No.: 20. Washington D.C.: World Bank Development Research Center; 1977.
31. Colson B, Marcotte P, Savard G. Bilevel programming: a survey. 4OR. 2005; 3(2): 87–107.
32. Zheng Z, Lu J, Zhang GQ, He Q. Rule sets based bilevel decision model and algorithm. *Expert Syst Appl* 2009;**36**(1):18–26.
33. Vicente L, Savard G, Judice J. Discrete linear bilevel programming problem. *J Optim Theory Appl* 1996;**89**(3):597–614.
34. Lu J, Shi CG, Zhang GQ. On bilevel multi-follower decision making: general framework and solutions. *Inf Sci* 2006;**11**(3), 1607-7.
35. Gao Y, Zhang GQ, Ma J, Lu J. A λ-cut and goal-programming-based algorithm for fuzzy-linear multiple-objective bilevel optimization. *IEEE Trans Fuzzy Syst* 2010;**18**(1):1–13.
36. Vicenet LN, Calamai PH. Bilevel and multilevel programming: a bibliography review. *J Global Optim* 1994;**5**(3):291–306.
37. Bard JF. *Practical bilevel optimization: algorithms and applications*. Boston: Kluwer Academic Publishers; 1998.
38. Dempe S. *Foundations of bilevel programming (non-convex optimization and its applications)*. Boston: Kluwer Academic Publishers; 2002.
39. Canny J, Reif J, Donald B, Xavier P. On the complexity of kinodynamic planning. In: *Proceedings of IEEE 29th annual symposium on foundations of computer science*, 1988 Oct 24–26, White Plains, NY, USA. Piscataway: IEEE; 1988. p. 306–16.
40. Hansen P, Jaumard B, Savard G. New branch- and bound rules for linear bilevel programming. *SIAM J Sci Stat Comput* 1992;**13**(5):1194–217.
41. McGee TG, Spry S, Hedrick JK. Optimal path planning in a constant wind with a bounded turning rate. 2005. Report No.: AIAA-2005-6186.
42. Andrew CV. Path planning and control of unmanned aerial vehicles in the presence of wind dissertation. Berkeley: University of California; 2003.

Liu Wei received his B.S. and M.S. degrees in system engineering from National University of Defense Technology in 2002 and 2004, respectively, and then became an assistant professor in the School of Science at the High-Tech Institute of Xi'an, Xi'an, China. He is now working towards his Ph.D. degree in guidance, navigation, and control at Beihang University. His current research interests are motion planning, flight task decision and optimization.

Zheng Zheng is an associate professor and M.S. advisor in the School of Automation Science and Electrical Engineering at Beihang University, Beijing, China. He received his Ph.D. degree from the Institute of Computing Technology at Chinese Academy of Sciences in 2006. His current research interests are flight task planning and software fault localization.

Cai Kaiyuan is a full professor and Ph.D. advisor in the School of Automation Science and Electrical Engineering at Beihang University, Beijing, China, where he received his Ph.D. degree in 1991. He is also a Cheung Kong Scholar (Chair Professor), jointly appointed by the Ministry of Education of China and the Li Ka Shing Foundation of Hong Kong in 1999. His area of research includes software reliability and testing, autonomous flight control, and software cybernetics.