GENERAL INSTRUCTIONS

- **Your proof has been created based on Society and IEEE layout and editing requirements.**
- **Carefully check the page proofs (and coordinate with all authors). Please check author names and affiliations, funding, as well as the overall article for any errors prior to sending in your author proof corrections.**
- **Additional changes or updates WILL NOT be accepted after the article is published online/print in its final form.**
- **New source files WILL NOT be accepted at this time.**
- **Once the article is finalized, you will receive an invoice for any applicable charges from the Copyright Clearance Center (CCC). Please visit https://journals.ieeeauthorcenter.ieee.org/your-role-in-article-production/about-potential-article-processing-charges/ for a comprehensive list of article charges. If you have any questions regarding overlength page charges, need an invoice, or have any other billing questions, please contact mailto:apcinquiries@ieee.org.**

AQ:1 = IEEE style encourages the use of inclusive and people-first language. Please consider replacing the terms “unmanned vehicle,” “Unmanned Aerial Vehicles,” and “Unmanned” with one of the terms from the Inclusive Language Guide found in the IEEE Editorial Style Manual: https://docs.google.com/document/d/1OalFYqVfxKKMelF79U3B93SjmIEiTnJC12wPJrKeSDM/edit?tab=t.0#heading=h.ik3tdhg3xeq0

AQ:2 = Please confirm or add details for any funding or financial support for the research of this article.

AQ:3 = Please confirm whether the edits made in the current affiliation of all the authors are correct.

AQ:4 = Please provide the month for Refs. [3], [7], [13], [14], and [41].

AQ:5 = Please provide the organization location and report no. for Ref. [18].

AQ:6 = Please provide the page range for Ref. [25].

AQ:7 = Please provide the organization name and organization location for Ref. [37].

AQ:8 = Please provide the publisher name and publisher location for Ref. [40].

AQ:9 = Please confirm whether the edits made in the sentence “received the B.Eng. degree in computer...” are correct.

A Unified and Quality-Guaranteed Approach for Dubins Vehicle Path Planning With Obstacle Avoidance and Curvature Constraint

Xing Zhou, Lin Li, Xinglong Zhang, *Member, IEEE*, Hao Gao, Kangxing Yao, and Xin Xu, *Senior Member, IEEE*

***Abstract*—Robotic technologies and applications have recently witnessed remarkable advancements. A major challenge is the shortest-path planning problem of a curvature-bounded vehicle from the known starting configuration to visit a target point and finally return to the starting configuration in an obstacle environment. Spurred by this significant issue in robotic surveillance and patrolling applications, this paper proposed the Two-trip Obstacle-environment Relaxed Dubins Problem (TORDP). In TORDP, the vehicle's target-visiting heading is a critical variable. Analytical approaches have existed for simpler scenarios than TORDP. However, these approaches are unavailable when solving the complex TORDP simultaneously with bounded curvature, variable target heading and unified ability to tackle with- or without- obstacle cases. Hence, we develop the mixed-integer piecewise-linear program (MIPWLP) approach, making the otherwise intractable complex scenario unifiedly solved with guaranteed good quality. Extensive experiments demonstrate that the proposed approach demonstrates effective performance. Furthermore, the objective approximation error in some cases was analyzed to achieve a length near the optimal length within $h^2/(2\sqrt{2})$ tolerance where h is the approximation piece length. The proposed MIPWLP approach could also offer a generalizable optimization framework for broader robotic path-planning applications in constrained environments.**

***Index Terms*—Dubins vehicle, obstacle avoidance, mixed-integer piecewise-linear program, relaxed Dubins problem, intelligent system, path planning.**

I. Introduction

A. Background

SIGNIFICANT progress has recently been witnessed in unmanned vehicle path planning for various platforms such as Unmanned Aerial Vehicles (UAVs), cars, and Unmanned Surface Vehicles (USVs) [1], [2], [3]. These vehicles have been widely deployed in applications such as point-of-interest (POI) reconnaissance, surveillance, patrolling, search and rescue, and field exploration [4], [5]. However, shortest-path planning for these vehicles is often challenging because they are subject to physical constraints, specifically, a bound on the maximum turning curvature prohibiting the vehicle from turning sharply with a small radius [6]. These vehicles are categorized as Dubins vehicles.

Received 30 October 2023; revised 27 May 2024, 5 October 2024, and 7 April 2025; accepted 19 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62306325, Grant U24A20279, and Grant U21A20518; and in part by the Key Project of Hunan Provincial Department of Education under Grant 24A0671. The Associate Editor for this article was C. Hu. *(Corresponding author: Xinglong Zhang.)*

Xing Zhou, Xinglong Zhang, Hao Gao, Kangxing Yao, and Xin Xu are with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: zhangxinglong18@nudt.edu.cn).

Lin Li is with the School of Intelligent Manufacturing, Hunan First Normal University, Changsha 410205, China.

Digital Object Identifier 10.1109/TITS.2025.3606479

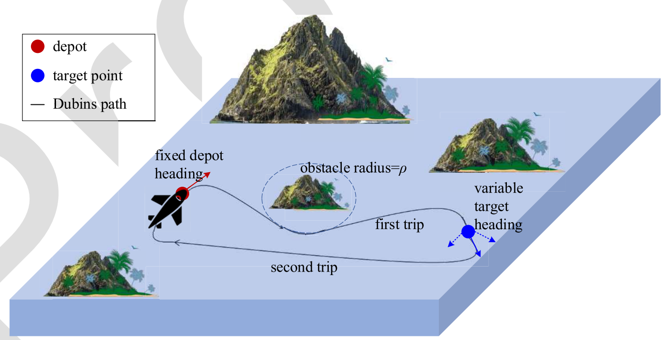


Fig. 1. Shortest-path planning of the curvature-bounded vehicle in environments with obstacles. Sometimes, the obstacles may not exist or exist but are ineffective, as in the second trip—the vehicle's path does not necessarily touch the obstacle.

This work deals with the vehicle's curvature-bounded shortest-path planning problems, illustrated in Fig. 1. In the examined scenario, the vehicle, e.g., a fixed-wing UAV, is taking off at the depot with a fixed heading angle (the runway angle). The vehicle is tasked to visit a point of interest within the work area **with any heading angle** and finally return back to the depot with the fixed heading angle, ensuring the entire surveillance two-trip path is the shortest. Application examples may be that a UAV daily frequently visits a coral reef target with known coordinates, optimizing the target-reaching angle to shorten the go-and-back two-trip for energy saving [6]. This scenario is called the Two-trip Obstacle-environment Relaxed Dubins Problem (TORDP) by us.

The main challenge lies in that the **curvature-bounded** vehicle's **visiting angle (heading)** at the target point is variable (i.e., free and should be optimized), and the environment may have an **obstacle** which represents an island or

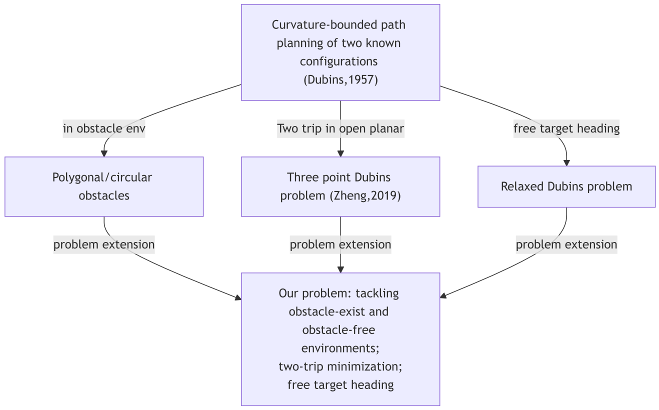


Fig. 2. Position and features of TORDP.

an adversarial area. Additionally, the obstacle is sometimes ineffective because it lies far from the shortest path or it does not exist. Hence, the proposed method requires a **unified tackling ability** for the effective or ineffective obstacle cases.

B. Related Work

TORDP is significant in various robotic applications and is complex, as shown in Fig. 2. Effective approaches exist for some simpler scenarios, e.g., considering only single-trip Dubins path [7], or fixed target heading [8], or obstacle-free scenarios [9]— some approaches are even analytical [10], [11], but we found that few of them have simultaneously considered the three features coming from robotic applications: (1) the entire two-trip Dubins path minimization, (2) the free target heading, and (3) the unification of obstacle and free-of-obstacle path. This missing is mainly because previous scenarios/applications did not encounter such a high complexity task; thus, their algorithm may not be capably applied to our scenario. The following is the detailed literature review.

In 1957, Dubins [10] conducted pioneering research on path planning under physical constraints [12]. He concluded that the shortest path between two positions with given heading angles, i.e., two configurations, must fall into one of the two path categories (each with no more than three path segments): CSC or subpaths, CCC or subpaths (where C represents minimum-radius circular-arc motion primitive, while S represents straight ahead motion primitive). Subsequently, researchers have followed the principles outlined by Dubins and Pontryagin [13], [14], [15] and have developed accelerated calculation methods for two known configurations [8], [16], [17]. When engaged with free target heading orientation angle (HOA) at the target point, it forms the Relaxed Dubins Problem (RDP) [18]. However, previous RDP solutions assumed obstacle-free environments and cannot be extended to the obstacle environment due to the difficulty of coupling the variable target heading and obstacles [9].

When obstacles exist in the work area, they impose a significant challenge in addressing the curvature-bounded vehicle [19], [20], [21]. Early studies focused on polygonal obstacles [22], [23], [24] for simple treatment of collision checking, without being fit for curvature-constrained complex settings. Subsequently, researchers focused on circular obstacles, and later approaches incorporated geometrical or analytical solutions for fixed target heading angles [25], [26]. Since these methods need the target heading to be geometrically fixed, they cannot handle free target angles in the two-trip minimization problem effectively because a periodic variable target angle brings structural difficulty to previous methods. Recent extensions have explored time-optimal lengthening and shortening in circular or other environments [27], [28]. However, these algorithms cannot unify obstacle-existing and obstacle-free conditions.

When engaged with two-trip length minimization, there were studies of the three-point Dubins path problem (3PDP) [29], [30], which involves three points, i.e., depot, middle, and target. In 3PDP, the aim is to decide the heading at the middle point to minimize the time from the starting to the middle to the end configurations. Although analytic solutions based on geometry have been proposed for the necessary conditions, current 3PDP solutions can hardly apply to open-plane scenarios. Moreover, scenarios with angle optimization in obstacle environments complicate the models [31], i.e., new discrete or binary variables emerging from angle optimizations complicate the feasible spaces so closed-form solutions hardly exist. Particularly, the periodic property of angles (or the congruent equation constraints) leads to integers in the feasible spaces, making derivative-based programming approaches incapable in this complicated feasible space. Recent research suggests solutions for visiting problems in an open environment in three-dimensional or similar settings [32], [33], [34], [35]. These schemes utilize differential geometry or handcrafted adaptive algorithms. Still, the heading orientation angle minimization is seldom considered.

C. Contribution

The limitations of previous research are that: previous work based on derivatives was hindered by the periodic property of HOA. Other approaches were hindered by constraints involving the trigonometric sine and cosine functions and by the lack of parameterized motion primitive derivations. Unlike existing methods, the proposed solution managed to break through. This work proposes a unified, quality-guaranteed mixed-integer piecewise-linear program (MIPWLP) for the TORDP, which exhibits high performance to problems that previous work cannot tackle featuring simultaneously free HOA, obstacle-existing, curvature-bounded constraints. This work's contributions can be summarized as follows:

1) **We propose a novel TORDP problem and establish it as a mathematical program.** The resulting program introduces integer variables to derive the three basic motion primitives with complex numbers and can unify scenarios with or without obstacles. It is probably the first program for free HOA, obstacle-existing, curvature-bounded, two-trip shortest-path planning.
2) **We develop mixed-integer and piecewise-linear strategies for the mathematical constraints involving congruent equations, trigonometric equations and direction consistencies.** MIPWLP is derivative-free,

TABLE I
THE NOTATIONS AND EXPLANATIONS

Symbol	Explanation
C/S/D	minimum-curvature circular/straight/obstacle-circumventing path segment
L/R	belonging to C, meaning left turn/right turn circular path segment
x_I, y_I, α_I	the known coordinates and heading at the depot
x_G, y_G, θ	the known coordinates and the variable heading at the target, with $\theta \in [0, 2\pi)$
(x, y, α)	a variable configuration, for parameterized move deduction use. $x, y \in R, \alpha \in [0, 2\pi)$
(x', y', β)	a variable configuration after a certain move from (x, y, α). $x', y' \in R, \beta \in [0, 2\pi)$
$\rho_{min}/\rho_1/\rho$	the normalized turn, the obstacle, and the obstacle-circumventing radii.
θ	heading $\in [0, 2\pi)$ at the target point. The coordinates of the target are also x', y'
ν	a move length or radian (the two will be the same because the turn radius is one)
f	f=+1 means vehicle's left turn; 0 means straight forward; -1 means right turn
$y = sin[x]$	y is constrained with a piecewise-linear approximation of the sine function
cross-product $xp = \vec{a} \times \vec{b}$	xp will > 0 if $\vec{b}$ is in the left-hand half-plane of $\vec{a}$; $= 0$ if co-linear; < 0 otherwise

making the program solvable by solvers. By approximating trigonometric functions with piecewise-linear functions and introducing integers for periodicity, we achieve an effective program. Moreover, several computational geometrical techniques, such as cross-product, dot-product, and directional consistencies, are employed to meet the constraints of the program.

3) **The approximation error (tolerance) between MIPWLP to the analytic optimal length in some cases is successfully analyzed**, guaranteeing a tolerance smaller than $h^2/(2\sqrt{2})$ concerning the piece length. Thus, MIPWLP is a quality-guaranteed (near-)shortest path planning approach.

4) **Our algorithm exhibits high performance across different cases, offers highly accurate solutions and unifiedly tackles cases with and without obstacle**. Notably, some cases were never solvable by any other existing approaches. It requires a nominal time cost in seconds. High-fidelity UAV simulation demonstrates the effectiveness of our method on path planning.

The remainder of this paper is organized as follows. Section II formally describes TORDP, the derivation of three motion primitives, and presents the preliminary piecewise-linear formulation and its mathematical equations. Section III introduces the proposed method, including the overall detailed approach. Section IV consolidates the experimental results and analyzes relatively simple cases. Section V discusses important considerations about our approach, and finally, Section VI concludes this work and proposes future research directions.

II. Problem Description and Preliminaries

This section first provides the basics for the Dubins vehicle and formulates the problem examined. Then, the parameterized motion primitives are deducted, followed by the basics of the piecewise-linear program. Table I lists and explains some notations that will be used throughout the paper.

A. Problem Description

The Dubins vehicle is defined as a vehicle that travels at a fixed and non-negative velocity with a curvature no larger than some specified value [5] and can usually be viewed as a mass point without sizes for path planning. The shortest path of the Dubins vehicle between two configurations is of type CCC or CSC (or a substring thereof), where $C \in \{\text{L, R}\}$ denotes left (L) or right (R) turns with the smallest radius permitted, i.e., with shortest paths involving six types: LRL, RLR, RSR, RSL, LSL, or LSR [8].

Without loss of generality, a Dubins vehicle can be normalized to travel at a constant speed of $\mu_s = 1$ m/s with a normalized minimum turning radius of $\rho_{min} = 1$ meter. This TORDP involves optimizing a length function: the path length from the known starting configuration $q_I(x_I, y_I, \alpha_I)$ where 'I' means 'initial', to reach the target point $q_G(x_G, y_G, \theta)$ where 'G' means 'goal', and then return to q_I, with only the target heading $0 \le \theta < 2\pi$ as the decision variable.

$$
\begin{aligned}
\min_{\theta}\ & Length = \int_0^{t_F} \sqrt{\dot{x}^2 + \dot{y}^2}\mathrm{d}t = \int_0^{t_m} 1\mathrm{d}t + \int_{t_m}^{t_F} 1\mathrm{d}t \\
\text{s.t.}\ & \dot{x} = \mu_s \cos\theta \\
& \dot{y} = \mu_s \sin\theta \\
& \dot{\theta} = u \cdot \frac{\mu_s}{\rho_{min}} = u \\
& Z(0) = q_I,\ Z(t_m) = q_G,\ Z(t_F) = q_I \\
& (Z(t, 0), Z(t, 1)) \in \mathcal{R}^2 \setminus \mathcal{C}_{obs},
\end{aligned}
\tag{1}
$$

where the control $u \in \{+1, 0, -1\}$ represents turning left, going straight, or turning right and u can be viewed as direction variables; t_m is the moment to reach q_G, and t_F is the time backing to the depot configuration; $Z(t)$ is the vehicle's state vector concerning time, $Z(t, 0)$ and $Z(t, 1)$ denote for the x, y coordinates of state $Z(t)$, and $\mathcal{C}_{obs}$ is the obstacle region.

B. Preliminaries

1) Proposed Derivation of Three Motion Primitives via Complex Numbers and Direction Decision-Variables: Our program involves vector rotations around a center with a certain radius. Therefore, it is fundamental to derive the parameterized rotation of the three motion primitives L, R, and S, illustrated in Fig. 3. The derivation is as follows: consider a vehicle at

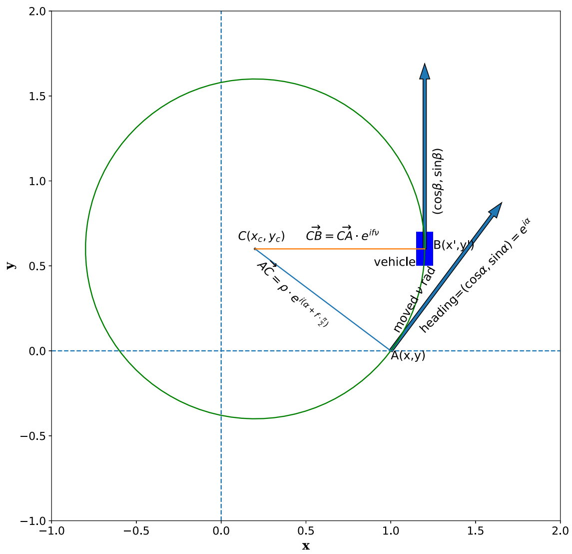


Fig. 3. A vehicle from the configuration $A(x, y, \alpha)$ to $B(x', y', \beta)$, via a parameterized rotating of ν radians with the radius ρ perpendicular to the velocity all the time, along the direction f (f=+1 left and -1 right). The coordinate system is the complex coordinate system.

any known former configuration $A(x, y, \alpha)$ with the heading angle α. It then moves a distance of ν radians on a circle with a radius of ρ. Note that $\rho \geq 1$ because the vehicle may merely conduct a normalized turn with $\rho_{min} = 0$ in an open environment, or may circumvent the obstacle with typical radius $\rho_1 \geq 1$ as assumed in [7]. The abstract of ρ contributes to unifying the obstacle-existing and obstacle-free conditions without explicitly distinguishing them individually. Let the configuration after the move be $B(x', y', \beta)$, where all three values are unknown. The derivation process utilizes complex numbers because of the convenience of representing rotation.

In Fig. 3, the current heading vector at point A is represented as a vector $(\cos\alpha, \sin\alpha)$ in Cartesian coordinates, or $e^{i\alpha}$ in complex number coordinates. Adding $\pi/2$ radians to the current heading yields a normal vector pointing from A toward the center of the left circle, while subtracting $\pi/2$ yields a normal vector pointing toward the right circle. These two normalized vectors are essential to derive the heading at point B. The normalized vector of $\overrightarrow{AC}$ using a complex number representation will be the previous complex number $e^{i\alpha}$ multiplied by (the exponential of $i \cdot$ *the rotated radian*), i.e., $e^{i\alpha} \cdot e^{i\pi/2} = e^{i(\alpha+\pi/2)}$.

Let an indicator integer variable $f = +1$ denote the vehicle turns left and $f = -1$ right. Based on f, one can derive the new configuration B after traveling a radian of ν from the current configuration A because (the complex number coordinates of B)= (coordinates of A) $+\overrightarrow{AC} + \overrightarrow{CB}$. So,

$$
\begin{aligned}
x' + iy' &= (x + iy) + \rho \cdot e^{i(\alpha + \frac{\pi}{2} f)} + \rho \left[-e^{i(\alpha + \frac{\pi}{2} f)} \right] \cdot e^{if\nu} \\
&= \left[x + \rho \cos\left(\alpha + \frac{\pi}{2} f\right) - \rho \cos\left(\alpha + \frac{\pi}{2} f + f\nu\right) \right] \\
&\quad + i \cdot \left[y + \rho \sin\left(\alpha + \frac{\pi}{2} f\right) - \rho \sin\left(\alpha + \frac{\pi}{2} f + f\nu\right) \right],
\end{aligned}
$$

where ρ is the circular radius, which can be 1 in Dubins path turns, or ≥ 1 in obstacle circumventing situations as defined before.

Locating the real and imaginary parts of the complex number $x' + iy'$ and considering the property of the odd multiple of $\pi/2$ provides:

$$
\begin{aligned}
x' &= \left[x + \rho \cos\left(\alpha + \frac{\pi}{2} f\right) - \rho \cos\left(\alpha + \frac{\pi}{2} f + f\nu\right) \right] \\
&= x - f\rho \sin(\alpha) + f\rho \sin(\alpha + f\nu) \\
y' &= \left[y + \rho \sin\left(\alpha + \frac{\pi}{2} f\right) - \rho \sin\left(\alpha + \frac{\pi}{2} f + f\nu\right) \right] \\
&= y + f\rho \cos(\alpha) - f\rho \cos(\alpha + f\nu).
\end{aligned}
$$

Note that, if the vehicle travels straight for a distance of ν meters, then $f = 0$. Incorporate it into the above equation as well, naturally:

$$
x' = x - f\rho \sin(\alpha) + f\rho \sin(\alpha + f\nu) + (1 - f^2)\nu \cos(\alpha), \quad (2)
$$

$$
y' = y + f\rho \cos(\alpha) - f\rho \cos(\alpha + f\nu) + (1 - f^2)\nu \sin(\alpha). \quad (3)
$$

It is evident that when turning left or right, $1 - f^2 = 0$. When $f = 0$, the x', y' expression is degraded to a straight movement of length ν.

Together with the heading at B,

$$
\beta = \alpha + f \cdot \nu. \quad (4)
$$

Hence, the left-turn/right-turn/straight motion primitives from (x, y, α) are derived using the equations Eq. (2)- (4). **These equations introduce as variables the directions, ν and ρ, which can all be optimized in our optimization program.**

Remark 1: Prior work [8] provided a normalized setting of a left/right/straight movement with ν radians/meters from an initial configuration (x, y, α) with radius 1, as follows:

$$
\begin{aligned}
L_\nu(x, y, \alpha) &= (x + \sin(\alpha + \nu) - \sin(\alpha), y - \cos(\alpha + \nu) \\
&\qquad + \cos(\alpha), \alpha + \nu) \\
R_\nu(x, y, \alpha) &= (x - \sin(\alpha - \nu) + \sin(\alpha), y + \cos(\alpha - \nu) \\
&\qquad - \cos(\alpha), \alpha - \nu) \\
S_\nu(x, y, \alpha) &= (x + \nu \cos(\alpha), y + \nu \sin(\alpha), \alpha).
\end{aligned}
$$

However, our proposed derivation generalized [8], as equations (2), (3), (4) are parameterized with turning radius ρ and direction f, unifying the obstacle-existing and obstacle-free conditions without explicitly distinguishing them individually.

2) Preliminaries of Mixed-Integer Piecewise-Linear Program: The developed MIPWLP should handle sin and cos functions as constraints, which require special treatment to make them tractable by solvers.

Fig. 4 explains the piecewise-linear approximation concept for the sin function within $[0, 2\pi]$. It reveals the linear approximation of $y = \sin(x)$ in an interval by multiple line segments (here, the segment number is chosen as 7 for an example value so as to enhance visualizing the approximation error). Suppose the objective is to minimize $(x - 4.76)^2$ where $x \in [0, 2\pi)$ and y, x have a sine relationship.

A PWL constraint specifies that a relationship $y = f(x)$ must hold between variables x and y, where f is a piecewise-linear function defined by breakpoints. PWL approximation surrogates curves using small line segments, with the tolerance error related to the length of these line segments. The $n-$breakpoint piecewise-linear relationship $y = PWL(x, ptx, pty)$ between

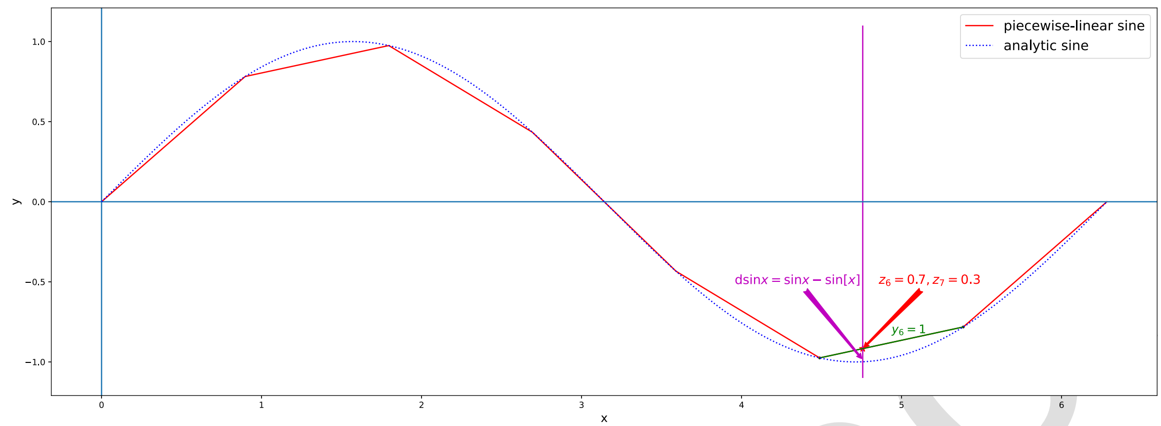


Fig. 4. Coarse illustration of piecewise-linear approximation of $\sin(x)$ on the interval $[0, 2\pi]$ with piece length $2\pi/7 \approx 0.9$ to enhance visualizing the error.

variables x and y, where ptx, pty are the breakpoints' x, y coordinates, can be constrained as [36]:

$$
\begin{aligned}
&z_1 \le y_1\\
&z_i \le y_{i-1} + y_i, \quad \forall i = 2, 3, \ldots, n-1\\
&z_n \le y_{n-1}\\
&y_1 + y_2 + \ldots + y_{n-1} = 1\\
&z_1 + z_2 + \ldots + z_n = 1\\
&x = z_1 \cdot ptx_1 + z_2 \cdot ptx_2 + \ldots + z_n \cdot ptx_n\\
&y = z_1 \cdot pty_1 + z_2 \cdot pty_2 + \ldots + z_n \cdot pty_n\\
&y_i \in \{0, 1\}, \quad \forall i = 1, 2, \ldots, n-1\\
&z_i \ge 0, \quad \forall i = 1, 2, \ldots, n,
\end{aligned}
$$

where y_i can only take binary values, and their sum equals one, i.e., only one segment y_i will be 1 (selected). Additionally, the non-negative variables z_i form a convex combination of the two endpoints of the selected segment—a point between the segment's two endpoints. The z_i, y_i relationships ensure that only two z_is are non-zero, and their sum equals 1. Solvers such as Gurobi [37] will automatically transform the $y = PWL(x, ptx, pty)$ constraints in the backend. These solvers tackle the mixed-integer piecewise-linear program via branch-and-bound or similar methods to find the best values for z_i and y_i, thus the best x and y, to optimize the objective. Our approach is not derivative-based. For our example, Gurobi uses a mixed-integer program that employs branch-and-bound techniques and finds the best (x, y), i.e., the red star in Fig. 4 at $x = 4.76$, $y = -0.92$, $y_6 = 1$ and $z_6 = 0.7$, $z_7 = 0.3$, while an analytic solution will be $x = 4.76, y = -0.99$.

An approximate error exists between the analytic sine and piecewise-linear sine $\mathrm{d}\sin x = \sin(x) - \sin[x]$, namely the line length between the red star and bowstring. Although the piece length is quite large (0.9 here) for illustration visualization, the approximate error is much smaller. Naturally, the approximation accuracy depends on the piece length and the breakpoints, i.e., the smaller the piece length, the smaller the error between the mathematical/analytical $\sin(x)$ and the PWL $\sin[x]$. The piece length should be selected considering the approximate accuracy and the computation time. An empirically typical value is 0.01.

III. The MIPWLP Approach

In this section, we first provide a detailed approach to the Single-trip Obstacle-environment Relaxed Dubins Problem (SORDP), which focuses on finding the shortest path from the starting configuration to the target configuration, with only the angle in the target point as a variable. In fact, there were no previous methods for SORDP other than ours. Since SORDP serves as the foundation for the Two-trip Obstacle-environment Relaxed Dubins Problem (TORDP), understanding SORDP's formulation is crucial. Once the principles and techniques for solving SORDP are established, we extend them to address TORDP, which involves a few additional constraints due to the need to return to the starting configuration, also considering the target heading as a variable. This presentation structure ensures clarity in our MIPWLP approach for TORDP and preserves the paper at a reasonable length.

A. SORDP: The Shortest Path From the Starting Configuration (x_I, y_I, α_I) to the Target Configuration (x_G, y_G, θ)

Fig. 5 illustrates an instance of SORDP. Typically, the vehicle's turning radius is normalized to one meter, and the distances between the depot, obstacle, and target are several kilometers. Under this mild condition, Yang [7], [38] proved via Pontryagin's minimum principle that the optimal path type is the 5-letter path type CSDSC, i.e., an optimal path will be first turning L or R using the minimum radius, going straight, circumventing the obstacle, going straight and finally turning L or R.

The essential variable in SORDP is the target heading θ, which should be optimized (see Alg. 1). Other variables such as t, p, q are secondary variables resulted from θ. The main idea for constraining SORDP is that the path CSDSC will sequentially pass through some critical points, such as tangent points and the target point, and constraints can be built on these critical points. After establishing the objectives and the constraints, MIPWLP solves SORDP through a comprehensive optimization framework.

Let $t, p, q, r, s \ge 0$ be the lengths of the five segments for a 5-letter Dubins path that act as optimizing variables related to the target heading variable θ. The SORDP model will be the

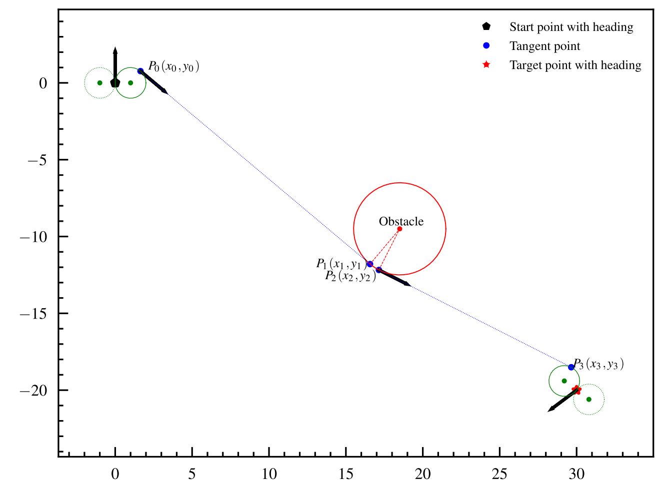


Fig. 5. SORDP's optimal path is CSDSC and the critical points of the 5-letter path, such as the depot, the tangent points and the target point, are shown. The angle at the target point is variable and should be optimized.

following equations with numbers in brackets. The objective of the SORDP is defined as follows:

$$\min_{t,p,q,r,s} \rho_{min} t + p + \rho q + r + \rho_{min} s, \tag{5}$$

where ρ_{min} represents the turning circle radius of the unmanned vehicle, normalized to 1, and ρ represents the vehicle's actual turning radius corresponding to the obstacle. Usually, $\rho \geq \rho_1$ and $\rho \geq \rho_{min}$ where ρ_1 is the radius of the obstacle. $\rho_1 \geq \rho_{min} = 1$, as assumed in [7].

The piecewise-linear constraints of sine and cosine functions are denoted as follows:

$$\sin[x] = \mathrm{PWL}(x, ptx, ptys)$$
$$\cos[x] = \mathrm{PWL}(x, ptx, ptyc),$$

where the PWL constraints are presented in Section II-B.

Note that $\sin[\cdot]$ ($\cos[\cdot]$) uses piecewise-linear approximation compared to the common brackets notation. The ptx represents the θ coordinate of the equally spaced interpolation points ($0 \leq ptx < 2\pi$), and the $ptys$ and $ptyc$ are the corresponding real values of $\sin(ptx)$ and $\cos(ptx)$. As mentioned before, the equal space of ptx should be selected considering the approximate accuracy and the computation time. Other radians outside the interval $0 \leq ptx < 2\pi$ can be transformed into this interval, as is well-known and will be seen later. Though MIPWLP only approximates the trigonometrical functions with linear pieces, as a consequence, the linear approximation will also impact the objective indirectly because there are equality relationships between the piecewise-linear variables in the constraints and normal variables (e.g., t, p, q) in the objective.

Assisted by the piecewise-linear constraints, the modulo operation, and computational geometry techniques, SORDP can be established as a tractable mixed-integer piecewise-linear program. The constraints, listed according to the chronological order of the vehicle's visit, are as follows:

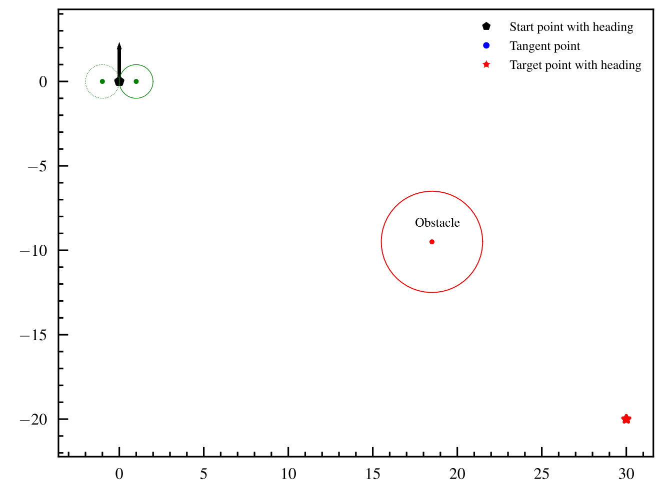
Algorithm 1 MIPWLP Algorithm for Problem SORDP

Input:
The starting configuration $q_I(x_I, y_I, \alpha_I) \in \mathbb{R}^2 \times \mathbb{S}$ and normalized turning radius ρ_{min}
end position $(x_G, y_G) \in \mathbb{R}^2$
obstacle position $(ob_x, ob_y) \in \mathbb{R}^2$ and obstacle radius $\rho_1 \geq \rho_{min}$
Decision variables (continuous) the heading $\theta \in [0, 2\pi)$ at the end position, the $t, p, q, r, s \in [0, +\infty)$ and the intermediate configurations.
Decision variables (discrete) $f_1, f_3, f_5 \in \{-1, 1\}$ for the first, third, and fifth segment direction (-1 for turning right)
Constants M, ϵ for big-M and small threshold
Output:
Optimal solution $\theta^*, t^*, p^*, q^*, r^*, s^*$ and optimal (single-trip) length
Step 1: Set the objective:
$\min_{t,p,q,r,s,\rho} \rho_{min} t + p + \rho q + r + \rho_{min} s$
Step 2: Model the constraints (modeled according to the chronological order of the vehicle's visit and the necessary property at each critical point and segment, for better logical understanding):
The first C segment constraints: From the starting configuration q_I via f_1 turning a t radian, the vehicle will arrive at intermediate configuration $P_0(x_0, y_0, \theta_C)$.
The first S segment constraints: From P_0 via going straight p length, with specific constraints in P_1, the vehicle will arrive at intermediate configuration $P_1(x_1, y_1, \theta_C)$.
The D segment constraints: From P_1 via f_3 turning a q radian, with specific constraints on f_3 and P_2, the vehicle will arrive at intermediate configuration $P_2(x_2, y_2, \theta_D)$.
The second S segment constraints: From P_2 via going straight r length, the vehicle will arrive at intermediate configuration $P_3(x_1, y_1, \theta_D)$, with specific constraints in P_3.
The second C segment constraints: From P_3 via f_5 turning a s radian, the vehicle will arrive at the **end configuration** (x_G, y_G, θ), with specific constraints on f_5 and s.
Step 3: Optimize the program: Solve the holistic mixed-integer program using an appropriate solver (e.g., Gurobi) and return the results constructed from the decision variables.

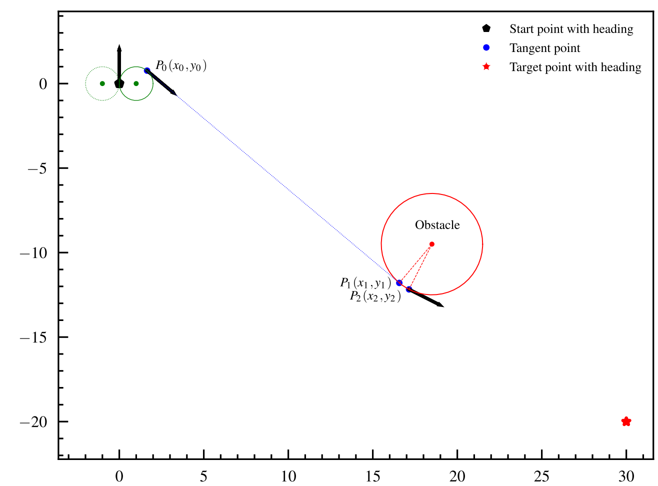
1) The First C Segment: Fig. 6(a)-(b) depict the initial setting of the scenario and the constraints/properties for the first C segment, the first S segment and the D segment. The intermediate variables $\theta_C, \theta_D \in [0, 2\pi)$ defined as the headings at points P_0, P_2 are presented in Fig. 5, respectively. Let the integer variables f_1, f_3, f_5 be the turning direction of the first, third, and fifth segments (the second and the fourth are straight already according to "CSDSC"). Thus, the following equation holds:

$$\theta_C = \alpha_I + f_1 \cdot t \pmod{2\pi}. \tag{6}$$

However, the mod operator cannot be handled directly to fit into solvers, causing periodicity and integers in the feasible



(a)



(b)

Fig. 6. The setting and the constraints for the first C, the first S and the D segments. The first C segment can turn left or right; the S segment will be tangent to both the turning circle and the obstacle-circumventing circle; the direction of D should be consistent with vector P_0P_1, clockwise or counterclockwise along the obstacle boundary.

space. Therefore, a proper transformation is required for the solvers. Hence, equation 6 is transformed into: $\theta_C = (\alpha_I + f_1 \cdot t) - \left\lfloor \frac{\alpha_I + f_1 \cdot t}{2\pi} \right\rfloor \cdot 2\pi$. By introducing an auxiliary integer variable $k = \left\lfloor \frac{\alpha_I + f_1 \cdot t}{2\pi} \right\rfloor$,

$$\frac{\alpha_I + f_1 \cdot t}{2\pi} - 1 < k \le \frac{\alpha_I + f_1 \cdot t}{2\pi}.$$

Hence, $\alpha_I + f_1 \cdot t - 2\pi < 2\pi k \le \alpha_I + f_1 \cdot t$ and $\theta_C = \alpha + f_1 \cdot t - 2\pi k$.

According to Fig. 6 and the complex number derivation presented in Section II, the following relational constraints hold for the tangent point P_0:

$$\begin{aligned} x_0 &= x_I + f_0 \cdot \rho_{min} \cdot \sin[\theta_C] - f_0 \cdot \rho_{min} \cdot \sin[\alpha] \\ y_0 &= y_I - f_0 \cdot \rho_{min} \cdot \cos[\theta_C] + f_0 \cdot \rho_{min} \cdot \cos[\alpha] \end{aligned} \tag{7}$$

Note that $[\cdot]$ denotes PWL constraints, which are essentially mixed-integer programs underneath, and solvers can use branch-and-bound techniques to solve them automatically.

2) The First S Segment: As shown in Fig. 6(b), a straight move from P_0 generates P_1, hence:

$$\begin{aligned} x_1 &= x_0 + p\cos[\theta_C] \\ y_1 &= y_0 + p\sin[\theta_C]. \end{aligned} \tag{8}$$

P_1 also has two properties: ① its distance ρ to the obstacle center is no less than the obstacle radius ρ_1. Therefore:

$$\begin{aligned} (b_x - x_1)^2 + \left(b_y - y_1\right)^2 &= \rho^2 \\ \rho &\ge \rho_1, \end{aligned} \tag{9}$$

② the shooting line from P_0 to P_1 is perpendicular to the line connecting the point P_1 and the obstacle center ob. Thus, their dot product is zero:

$$\cos[\theta_C] \cdot (ob_x - x_1) + \sin[\theta_C] \cdot (ob_y - y_1) = 0. \tag{10}$$

3) The D Segment: The directional tangent line P_0P_1 determines whether to follow the obstacle boundary clockwise or counterclockwise (see Fig. 6(b)), so the circumventing direction of the D segment is determined by the lateral position of the circle center with respect to the line P_0P_1, where the cross-product can be utilized to constrain the lateral position. Let xp_1 be the cross-product of P_0P_1 respect to vector $P_1(x_1, y_1) - (ob_x, ob_y)$:

$$xp_1 = \cos[\theta_C] \cdot (ob_y - y_1) - \sin[\theta_C] \cdot (ob_x - x_1), \tag{11}$$

where (ob_x, ob_y) is the center of the obstacle.

The sign of xp_1 must equal to the binary variable f_3, so it must hold:

$$f_3 = \mathrm{sgn}(xp_1). \tag{12}$$

To transform the non-trivial sign function $g = \mathrm{sgn}(\omega)$ to be tractable by solvers, one can use the big-M strategy:

$$\begin{aligned} &-M(1 - z_1) \le g + 1 \le M(1 - z_1), \ \omega \le -\epsilon + M(1 - z_1) \\ &-M(1 - z_2) \le g \le M(1 - z_2), \\ &-M(1 - z_2) - \epsilon \le \omega \le \epsilon + M(1 - z_2) \\ &-M(1 - z_3) \le g - 1 \le M(1 - z_3), \ \omega \ge \epsilon - M(1 - z_3) \\ &z_1 + z_2 + z_3 = 1 \end{aligned}$$

where z_i are auxiliary binary variables, and M is a sufficiently large (but as small as possible) constant. In the experiments, $M = 1000$. ϵ is a small equality constant set to 1e-5.

The constraint on θ_D is given by:

$$\theta_D = \theta_C + f_3 \cdot q \pmod{2\pi}, \tag{13}$$

which can be similarly transformed as the previous θ_C.

Circumventing (not necessarily touching) the obstacle boundary with q radian from P_1 (recall the complex number derivations), we reach P_2, and thus have:

$$\begin{aligned} x_2 &= x_1 + f_2 \cdot \rho \cdot \sin[\theta_D] - f_2 \cdot \rho \cdot \sin[\theta_C] \\ y_2 &= y_1 + f_2 \cdot \rho \cdot \cos[\theta_D] + f_2 \cdot \rho \cdot \cos[\theta_C]. \end{aligned} \tag{14}$$

4) The Second S Segment: The second S segment of the trip gives:

$$x_3 = x_2 + r \cdot \cos[\theta_D]$$
$$y_3 = y_2 + r \cdot \sin[\theta_D] \tag{15}$$

5) The Second C Segment: Let decision variable $finalLR = 1$ represent the left unit circle of the target configuration (x_G, y_G, θ) and $finalLR = -1$ represent the right unit circle. The circle center $(final_x, final_y)$ of this unit circle is parameterized as follows:

$$(final_x, final_y) = (x_G, y_G) + \rho_{min} \cdot finalLR \cdot (-\sin[\theta], \cos[\theta]) \tag{16}$$

Additional constraints mainly comprise clockwise or counterclockwise **direction consistencies**. Hence: (1) P_3 lies on the *finalLR* circle passed the final point ($\text{final}_x, \text{final}_y$) with a radius of $\rho_{min} = 1$. (2) The vector from P_2 to P_3 is perpendicular to the vector $P_3 P_f$. (3) The cross-product of $P_2 P_3$ to $P_3 P_f$ equals *FinalLR* and f_5. (4) The fifth constraint is that the heading value θ at the target point equals θ_D plus some circumventing radians. These consistencies can be translated into:

$$(x_3 - final_x)^2 + (y_3 - final_y)^2 = \rho_{min}^2$$
$$(x_3 - final_x) \cdot \cos[\theta_D] + (y_3 - final_y) \cdot \sin[\theta_D] = 0$$
$$xp_2 = \cos[\theta_D] \cdot (final_y - y_3) - \sin[\theta_D] \cdot (final_x - x_3)$$
$$f_5 = \text{sgn}(xp_2) = FinalLR$$
$$\theta = \theta_D + f_5 \cdot s (\text{mod } 2\pi). \tag{17}$$

B. TORDP: Shortest Path From (x_I, y_I, α_I) to (x_G, y_G, θ) to (x_I, y_I, α_I)

The equations (5)–(17) in the last subsection institute the objective and constraints for SORDP, which is from configuration (x_I, y_I, α_I) to configuration (x_G, y_G, θ). The two trips of the TORDP can be treated as the trip from the starting configuration (x_I, y_I, α_I) to the target configuration (x_G, y_G, θ) and the trip from the target configuration (x_G, y_G, θ) to the starting configuration (x, y, α), linked with the common θ variable at the target point.

Since the second trip of TORDP is planning a path from (x', y', θ) to (x, y, α) with CSDSC path type, we introduce an additional set of variables t_2, p_2, q_2, r_2, s_2 and extra constraints to the optimization objective. For TORDP, the objective and constraints of TORDP are:

$$\begin{aligned}
&\min \rho_{min} t + p + \rho q + r + \rho_{min} s \\
&\quad + rho_{min} t_2 + p_2 + \rho q_2 + r_2 + \rho_{min} s_2, \\
&s.\ t.\ \text{constraints (as Eqs. (6) – (16)) for the path from} \\
&\qquad \text{configuration } (x_I, y_I, \alpha_I) \text{ to configuration } (x_G, y_G, \theta); \\
&\qquad \text{constraints with } \mathrm{t_2, p_2, q_2, r_2, s_2} \text{ for the path from} \\
&\qquad \text{configuration } (x_G, y_G, \theta) \text{ to configuration } (x_I, y_I, \alpha_I),
\end{aligned} \tag{18}$$

where the decision variables are $t, p, q, r, s, t_2, p_2, q_2, r_2, s_2$, with t_2 the immediate path segment after visiting the target point. Similar to SORDP, all constraints of TORDP can be established by following a chronological order of CSDSC-CSDSC path type and constraining the critical points—the two CSDSC trips are linked with the common θ variable and the two trips are holistically solved by GUROBI. The additional constraints of TORDP than SORDP are the constraints of the path from $q_G(x_G, y_G, \theta)$ to $q_I(x_I, y_I, \alpha_I)$, which highly resembles the Eqs.(6)–(17) because of the one-to-one correspondence between $x_I \leftrightarrow x_G,\ y_I \leftrightarrow y_G,\ \alpha_I \leftrightarrow \theta$.

TABLE II

INFORMATION ON THE DIFFERENT COMBINATIONS AND THE FOUR EXPERIMENTAL SETS FROM EIGHT COMBINATIONS

Trip	Obstacle	Heading	Available methods
one	without	fixed	Classic Dubins methods
one	without	variable	Classic RDP methods
one	**with**	**fixed**	Tangent line method; ours
one	**with**	**variable**	No previous methods; ours
two	without	fixed	Classic Dubins methods
two	**without**	**variable**	Chen's algorithm; ours
two	with	fixed	Tangent line method
two	**with**	**variable**	A modified heuristic; ours

Remark 2: Our mixed-integer piecewise-linear programming (MIPL) approach differs fundamentally from traditional optimization techniques, such as the tangent-line method, Chen's method, and the angle-discretization heuristic: These traditional approaches rely on analytical formulations, which assume continuous and smooth solution spaces. While they work well for simpler Dubins path problems, they struggle to handle the complex constraints inherent in the TORDP with the three difficult features mentioned before.

IV. EXPERIMENTAL VALIDATIONS

The following experiments validate that MIPWLP is accurate, effective, and unifiedly capable. Some combinations have been solved well among the eight combinations of different trip types, obstacles, and target headings. Therefore, we conducted and compared four sets of numerical experiments to evaluate the high accuracy and unification performance of the proposed algorithm (presented in bold in Table II): (1) Single-trip experiments in obstacle-constrained environments with a given target heading. (2) Single-trip experiments in obstacle-constrained environments with a variable target heading. (3) Two-trip experiments in obstacle-free environments with a given target heading. (4) Two-trip experiments in obstacle-constrained environments with a variable target heading.

All test cases have a starting configuration of $q_I(0, 0, \pi/2)$, as in [7]. We normalized the x, y coordinates to have a unit of one. The approximation piece length is 0.01, the first breakpoint in *ptx* is 0.00, and the last is 6.29 (nearly 2π). The cutoff time is set to 1 hour. However, many cases are in the order of seconds, and only the fourth group needs one hour. Although our approach is a path planning algorithm for a vehicle with a bounded curvature, we validated the path planning and path following performance using the high-fidelity UAV toolbox from MATLAB. After experimenting with MIPWLP, the final part of this section analyzes the approximation errors,

TABLE III
INFORMATION AND COMPARISON RESULTS BETWEEN OUR ALGORITHM AND THE TANGENT LINE METHOD [7]

Information			Tangent line method [7]	Our algorithm		
Case	target point	obstacle radius	length	our length	absolute error	ρ
a	(18.5,-9.5)	3	38.233	38.232	0.001	3.000
b	(19.5,-8.5)	3	38.003	38.004	0.001	3.000
c	(20.5,-7.5)	3	37.966	37.967	0.001	3.956
d	(18.5,-9.5)	2	38.063	38.064	0.001	2.218
e	(18.5,-9.5)	3	38.233	38.232	0.001	3.000
f	(18.5,-9.5)	4	38.525	38.524	0.001	4.045

presenting quality-guaranteed proofs. The experimental setup involves an Intel Core i7 and a GPU using CUDA with 32 GB RAM. The operating system is Win10, and the solver is Gurobi 10.0 [37] with gurobipy interfaces.

A. Single-Trip Experiments in Obstacle Environments With Given Target Heading

Setup: The first experiment aims to reveal whether MIPWLP can obtain nearly optimal results compared to the analytical optimal results. In this scenario, the x, y coordinates and headings of the starting and target points are given, and the proposed algorithm is compared with the tangent line method [7]. The tangent line algorithm determines the positions of the tangent points and plans the collision-free path from the starting configuration to the ending configuration via the tangent points. The tangent line algorithm has proved to be optimal. Here, the target configuration adopts [7] and is fixed as $(30, -20, atan2(-4, -3))$.

This subsection conducted six experiments (among which the first and the fifth were identical— we added this duplicate to view the performance trends better). The obstacle center in the first three experiments is different (see Table III), while the radius of the obstacle in the last three experiments is different.

Results: Table III compares the results between the proposed algorithm and the tangent line method [7]. The comparison consists of different cases, i.e., the total length of the ground truth, our algorithm's length, the absolute error, and the circumventing radius that MIPWLP calculated. The shorter the total length, the better the algorithm's performance. **The results highlight that the total length of the proposed algorithm is nearly the same as the optimal path length [7], with absolute errors smaller than 0.001.**

Fig. 7 illustrates the impact of the obstacle centers and the obstacle radii. The total length decreases as the obstacle is farther away from the main path, and the total length increases as the obstacle radius increases in the second-row settings. The subfigure (c) reveals that our approach has a natural unification of the 5-letter case and the "CS0SC" case in a single trip, i.e., the length of q in the t, p, q, r, s vanishes to zero. In particular, in some cases where the starting and target points are far from obstacles, the length of some segments may be zero, so there will be one straight line rather than two passing near the obstacle.

Overall, in this experimental set, the proposed algorithm reaches the optimal, proving an effective and accurate solution to the path planning problem with given headings. It unifiedly found the optimal 5-letter path and the "CS0SC" path naturally.

B. Single-Trip Experiments in Obstacle Environments With Variable Target Heading

The first set of experiments verifies that the proposed algorithm provides the shortest path between the given start and target. This subsection verifies our algorithm's performance with a variable target heading in an obstacle environment.

Setup: The second set of experiments plans the shortest trip from a given starting configuration to the target point with a variable heading in an environment with obstacles. The depot's heading is given, but the target point heading is not fixed. Like Section IV-A, we conducted six experiments using the identical coordinates as in the experimental Set 1.

Results: We self-compare the results to those of Set 1 to preserve a reasonable article length. Table III highlights that the planned paths under variable target headings are always less than or equal to those with fixed headings since the unfixed target heading corresponds to a larger search space. For the "CS0SC" case, when the shortest path does not touch the obstacle, the length of q will be almost zero, and the circumventing radius will be larger than the obstacle radius, i.e., cases b and c in Table III. In addition, the planned paths are plotted in Fig. 8 to analyze and comprehend the results. The plotted paths infer that the algorithm chooses a proper θ that prevents the path from approaching the obstacle. Hence, the larger the obstacle radius, the longer the optimal path.

C. Two-Trip Experiments in Obstacle-Free Environments With Variable Target Headings

Setup: The experimental Set 3 verifies our algorithm's performance in calculating the two-trip Dubins path within obstacle-free environments while eliminating the obstacle constraints in MIPWLP. TORDP passes through three predefined points: the depot, the target point, and the depot (as the final destination). In [30], the authors have proved that the shortest three-point Dubins path in an obstacle-free environment is obtained by finding the **provable** optimal heading of the target point via a roots-finding algorithm. Therefore, we adopted their strategy using the SymPy Python package and compared it against the proposed algorithm in four cases. In all cases, the starting and ending points are the same (i.e., $(0, 0, \pi/2)$), while

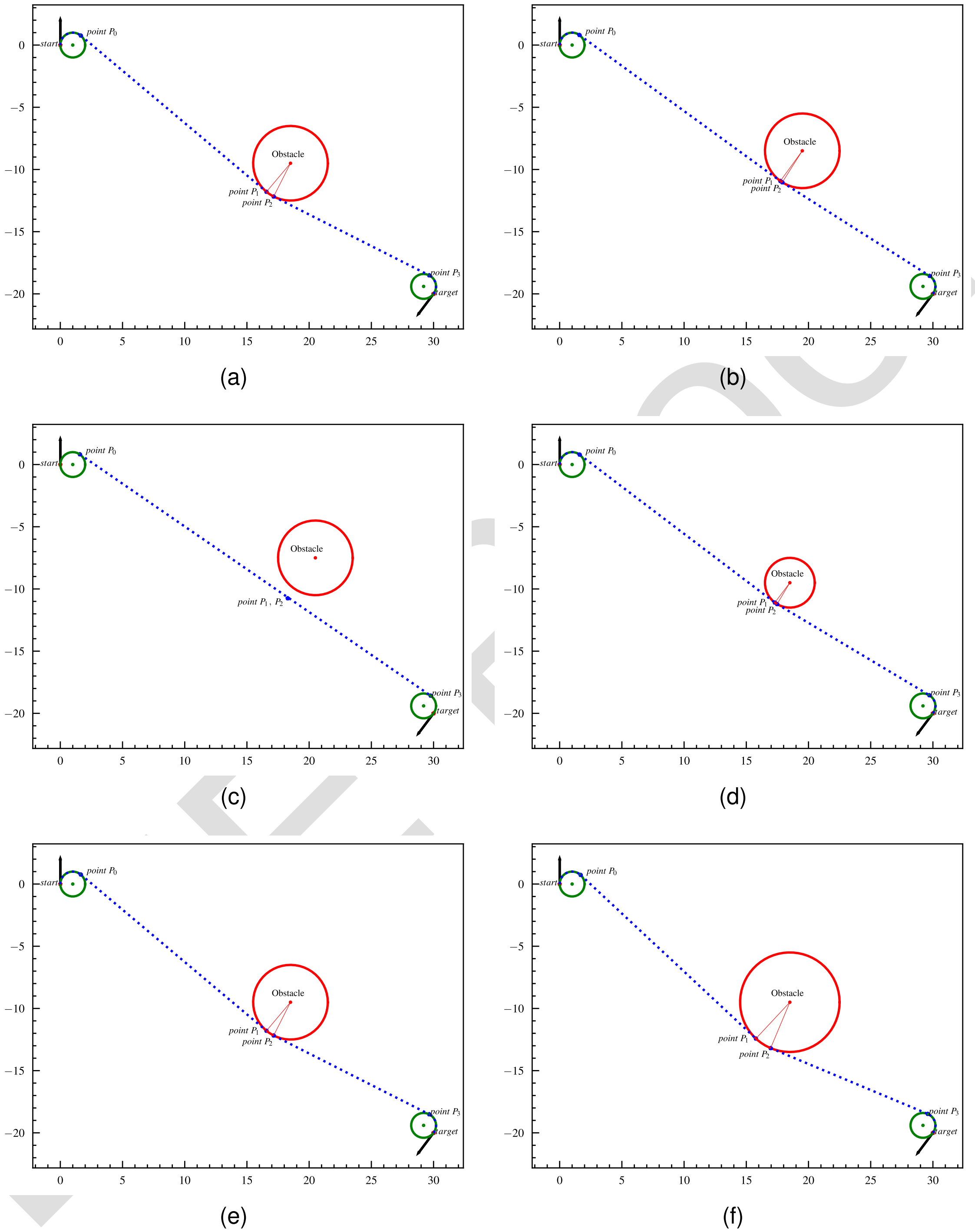


Fig. 7. The (near-)optimal solution of MIPWLP on the experimental set 1. It is a visual and detailed presentation of Table III.

TABLE IV

MIPWLP Results on the Experimental Set 2, Presenting Shorter Paths Than the Fixed Target Heading Cases. Cases b and c Have a "CS0SC" Optimal Path, Where q Is Almost Zero, and the Circumventing Radius Is Larger Than the Obstacle Radius

Case	t	p	q	r	s	optimal length (length in last table)	optimal target heading θ	ρ
a	2.271	19.506	0.155	15.131	0.147	**37.519** (38.232)	5.590	3.003
b	2.203	19.947	0.000	15.264	0.001	**37.417** (38.004)	5.650	3.078
c	2.203	20.164	0.000	15.050	0.000	**37.417** (37.967)	5.651	4.471
d	2.220	19.685	0.037	15.436	0.008	**37.423** (38.064)	5.680	2.000
e	2.271	19.506	0.155	15.131	0.147	**37.519** (38.232)	5.590	3.003
f	2.322	19.274	0.270	14.887	0.168	**37.731** (38.524)	5.970	4.000

the target points are $(30, -20), (30, 20), (-30, 20), (-30, -20)$, respectively.

Results: Fig. 9 and Table V qualitatively compare the proposed algorithm and [30]. The four cases are plotted within

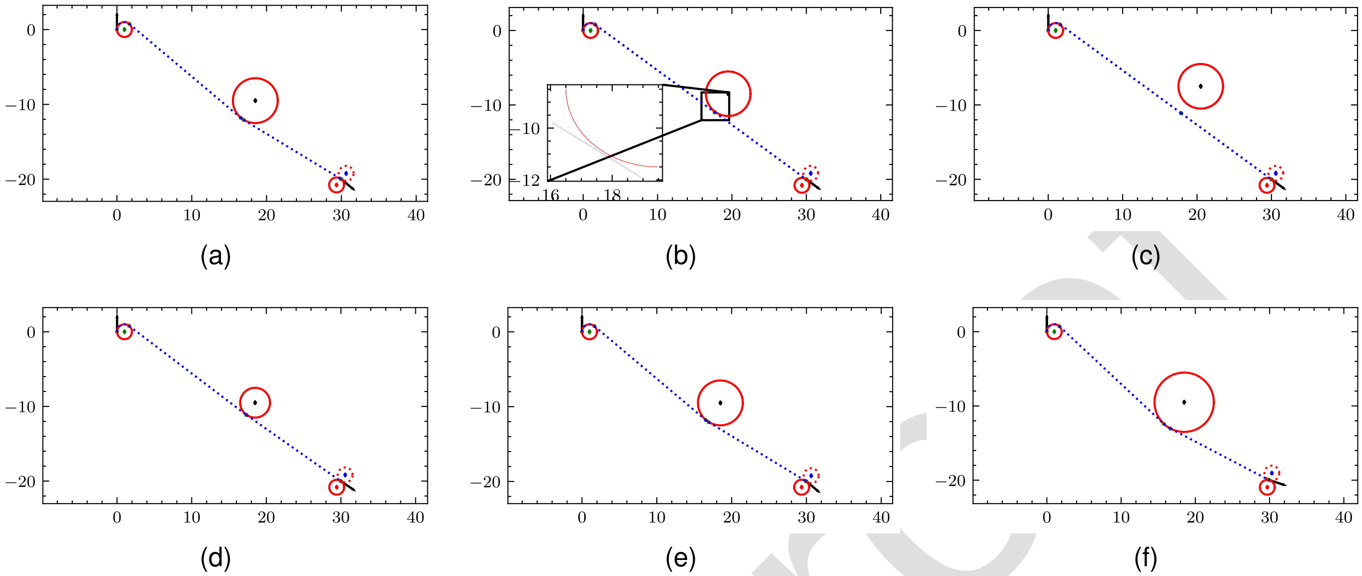


Fig. 8. Visual presentation of Table III, where the selected *finalLR* circles are depicted in solid line while the other circle is in dashed line. All solutions selected the right circle to pass the target point. (*b*) is zoomed in to view the details. The optimal heading at the target point is different from that of experimental Set 1.

TABLE V

COMPARISON OF OUR ALGORITHM AND CHEN'S ALGORITHM [30]. e_1 AND e_2 IS THE LENGTH AND θ ABSOLUTE ERROR, RESPECTIVELY

	Chen's algorithm						Our algorithm									
Case	t	p	q	q'	r	s	t	p	q	q'	r	s	length	e_1	e_2	T.(:s)
BR	2.170	34.230	1.570	1.570	34.230	0.970	2.175	34.228	1.572	1.570	34.228	0.967	**74.740**	0.001	0.003	**7.00**
TR	0.970	34.230	1.570	1.570	34.230	2.170	0.967	34.228	1.566	1.575	34.228	2.174	**74.740**	0.001	0.000	**2.98**
TL	0.970	34.230	1.570	1.570	34.230	2.170	0.968	34.230	1.550	1.591	34.228	2.174	**74.740**	0.001	0.011	**3.00**
BL	2.170	34.230	1.570	1.570	34.230	0.970	2.175	34.227	1.563	1.578	34.228	0.967	**74.739**	0.000	0.006	**3.00**

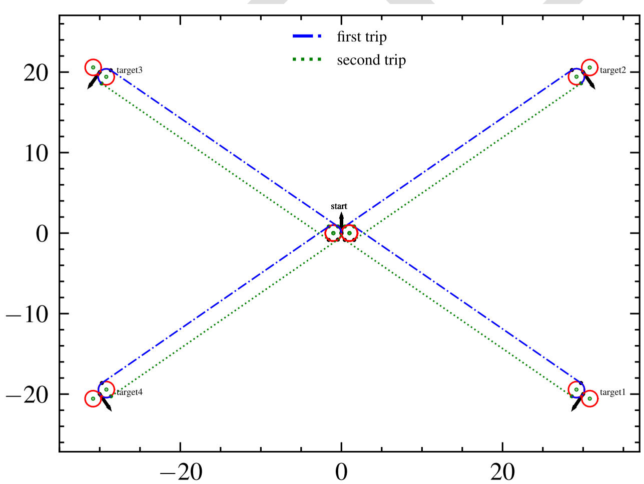


Fig. 9. Results of experimental Set 3 with four cases. The solutions are optimal and identical to Chen's ground-truth results [30]. The first and second trip in each TORDP is represented with '-.' and '.' line styles, respectively.

one figure to save article space. Since the physical environment is assumed to be open, MIPWLP has been modified accordingly to eliminate the obstacle constraints. Fig. 9 illustrates the four cases, where the circular segment just before the target point is denoted as q and just after as q'. The planned paths reveal that the bottom left (BL) and bottom right (BR) cases are symmetric and have obtained symmetric optimal results. The top left (TL) and top right (TR) cases are similar.

From the detailed data (Table V), it is evident that the proposed algorithm provides very close q and q', which aligns with the findings from the optimality-proved algorithm [30]. In addition, the proposed algorithm has **nearly the same optimal θ and optimal path length as Chen's algorithm** [30], which has been theoretically proved optimal. However, the computational burden of our MIPWLP is less than 7 seconds in each case, which is quite quick. This experimental set demonstrated our algorithm's accuracy and efficiency in non-obstacle scenarios.

D. Two-Trip Experiments in Obstacle Environments With Variable Target Headings

Setup: Unlike the experimental Set 3, Set 4 verifies the performance of the proposed algorithm in obstacle environments. It resembles Set 3, but obstacles are added with radius=3 centered at (18.5,-9.5), (18.5,9.5), (-18.5,9.5), (-18.5,9.5). This experimental set aims to reveal the performance of the developed scheme in this scenario and compare it with Set 3. Moreover, the results are compared with a prior heuristic that can be modified and applied to the free target heading optimization to prove the advantages of MIPWLP.

Results: Fig. 10 depicts the paths planned by our algorithm, where the first trip is in blue and the second trip is in dotted

TABLE VI
MIPWLP SOLUTIONS OF TORDPS FOR THE FOUR CASES

Case	t	p	q	r	s	ρ	t_2	p_2	q_2	r_2	s_2	ρ	length	θ
BR	2.270	19.510	0.221	14.043	1.605	3.000	1.645	14.393	0.034	19.761	0.950	3.454	74.961	4.200
TR	0.973	19.813	0.120	14.042	1.602	3.000	1.680	14.463	0.081	19.506	2.270	3.000	74.952	2.080
TL	0.972	19.814	0.135	14.009	1.798	3.020	1.483	14.454	0.098	19.502	2.271	3.020	75.006	0.880
BL	2.270	19.510	0.221	14.043	1.605	3.000	1.645	14.393	0.034	19.761	0.950	3.454	74.961	5.225

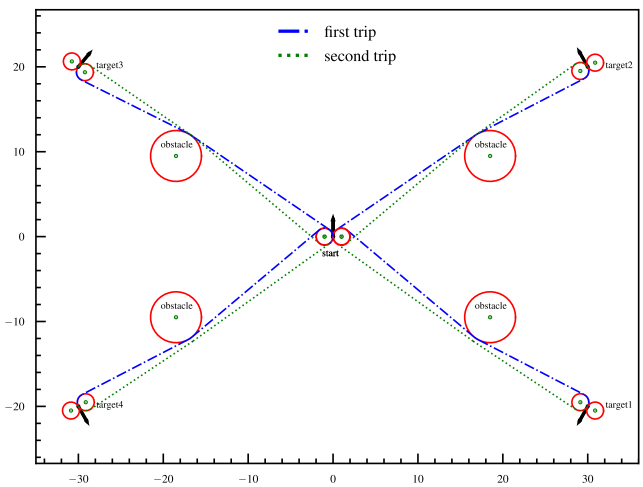


Fig. 10. Results of the experimental Set 4, with four cases, each with a target and obstacle. The symmetric cases have symmetrically identical solutions.

green. The two-trip length is smaller than twice the single-trip length. One may find the symmetries between the down-left and down-right cases, between the up-left and up-right cases—the cases will have the same ground-truth length, and MIPWLP indeed finds the same length and path.

Table VI presents the detailed length of each segment, where t_2 is the segment immediately after the target point and p_2 follows. The presence of obstacles increases the path length, leading to a slightly larger total length than the two-trip without obstacles— MIPWLP has provided quite good lengths.

Although no previous algorithms tackle TORDP, a heuristic (inspired by [39]) is used to validate the advancement of MIPWLP. The heuristic uniformly discretizes $[0, 2\pi)$ to $L(6 \leq L < 16)$ samplings and then transforms a TORDP to two one-trip fixed target heading problems. The two problems can be easily and optimally solved using [7]. We implemented the heuristic, calculated the shortest two-trip length S_L among the L samples, and recorded the corresponding sampled angle as A_L. The BR and TR cases in Table VI were experimented with this heuristic.

In each case, the S_L is slightly longer (0.5 meters) than MIPWLP's length, and the length difference decreases as the discretization level increases. Fig. 11 illustrates the angle difference between the heuristic and MIPWLP, revealing that the angle difference is between 5 and 20 degrees. This finding suggests that the optimized heading of the heuristic and the MIPWLP to visit the target differs greatly and that the difference has no obvious trend regarding the number of discretizations. This is reasonable

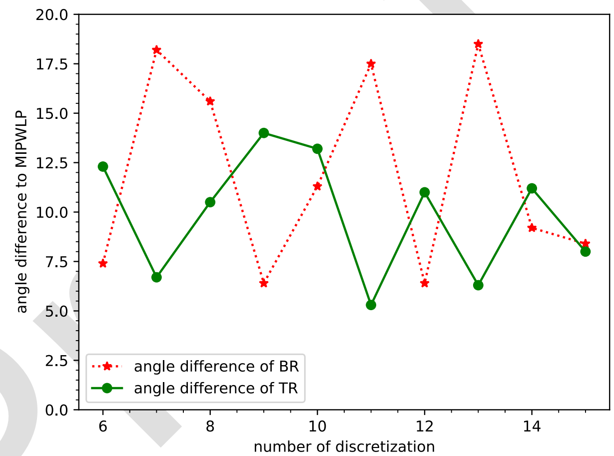


Fig. 11. The angle difference between the heuristic and MIPWLP, where the latter angles result in shorter path lengths.

because the performance of heuristics is hard to predict and guarantee.

E. Path Following Simulation Based on Uav Toolbox

This section validates the SORDP path-following performance of the paths planned by MIPWLP using a fixed-wing UAV model. The UAV Toolbox in MATLAB offers algorithms, simulation environments, and reference applications for designing, simulating, testing, and deploying unmanned aerial vehicle and drone applications. The UAV Guidance Model block includes fixed-wing UAV aerodynamics and an autopilot, approximating the kinematic behavior of a closed-loop system. The kinematics of a fixed-wing UAV at a constant height is that of a typical Dubins vehicle, offering a high-fidelity simulation. Since the UAV model is 3D, it is restricted to flying at a fixed height, turning it into a 2D Dubins vehicle to suit the MIPWLP algorithm. Samples with (x, y, θ) coordinates are generated along the MIPWLP planned (near-)shortest path and input to the UAV waypoint following modules for the path following.

The waypoint-follower configuration involves the Waypoint Follower block, which computes a desired heading for the UAV based on waypoints, current pose, and look-ahead distance inputs. The heading control block acts as a proportional controller to regulate the UAV's heading angle, with sliders available for tuning look-ahead distance and heading control values.

The simulated scenario represents a real scenario [7]: The initial and final configuration points are (0,1000) and (3000, -1000) in the meter unit. We scale the coordinates to 1/100 in the simulation and translate the initial configuration to (0,0), so the final configuration is (30,-20). The start heading is the

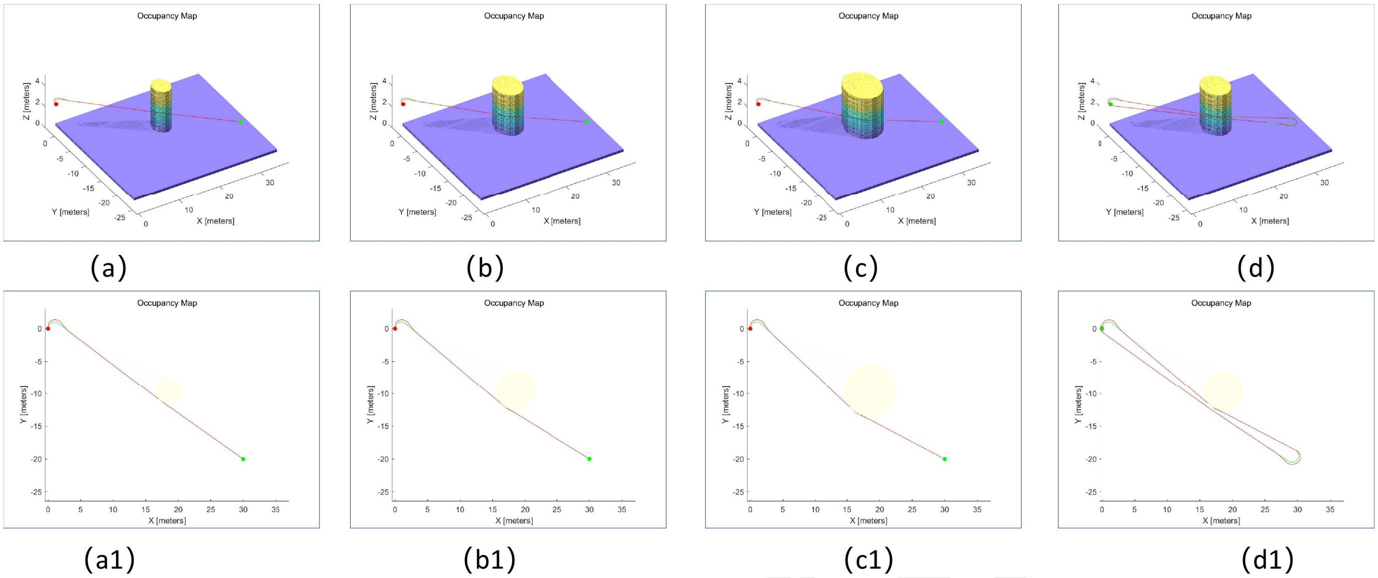


Fig. 12. Results of four simulation experiments. (a)-(d) are side views, and (a1)-(d1) are vertical views.

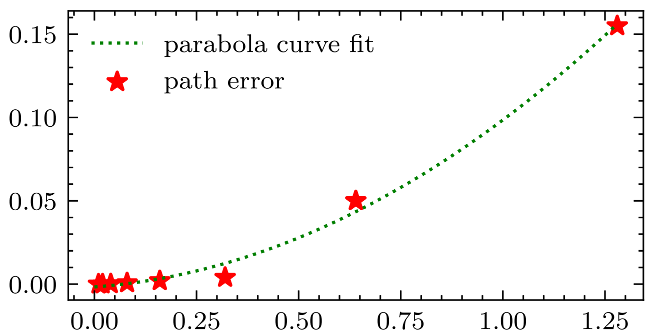


Fig. 13. Path length error of different piece lengths and a best-fit quadratic curve. The fit quadratic is about $0.07h^2$, showing better results than the worst-case quality guarantee analysis

TABLE VII

KEY PARAMETERS OF THE FIXED-WING UAVS

	Key Parameter	Value
1	MaxRollAngle	0.10168 rad
2	AirSpeed	1m/s
3	FlightPathAngleLimit	[-0.0001 0.0001] rad
4	g	9.8 N/kg

same as the real scenario. To adapt fixed-wing UAVs to 2D Dubins vehicles, we fix the UAV's height (100 meters in the real scenario) and limit the FlightPathAngle to a range close to zero. Detailed UAV parameters are outlined in Table VII to normalize the vehicle's turning radius (200 meters) as one meter and velocity as 1 unit (30 m/s in the real scenario). The obstacle in the real scenario has similarly been normalized to (18.5,-9.5) with a radius of three. The obstacle in a coral reef reconnaissance application can be islands or dangers and it is treated as a cylinder, as depicted in Fig. 12.

Four validation cases have been conducted in this subsection. Fig. 12(a)-(c) correspond to Fig. 8(d)-(f), while the fourth case corresponds to Fig. 10 BR. Fig. 8 and Fig. 10 depict the shortest single-trip and two-trip from the given starting configuration to the target point under a variable heading in an environment containing obstacles.

Fig. 12 presents the paths planned by MIPWLP (green paths) and the actual path followed by the UAV (red paths), highlighting that the two paths exhibit high coincidence. Indeed, there is only a slight deviation where the straight segment meets the arc due to the control change of the Dubins path at the junction. Future work could incorporate Clothoid curves to address curvature continuity and control smoothness at the junctions. These four experiments demonstrate that the fixed-wing UAV effectively follows the shortest path planned by MIPWLP.

F. Quality Guarantees

Next, the error (tolerance) between the optimal and MIPWLP solutions is theoretically analyzed in relatively simple cases. Given that comprehensively analyzing MIPWLP is quite complicated, thus it is left for future studies. Since the error estimation on the simple case can enhance our understanding of the proposed approach, this section demonstrates that the optimal length and the piecewise-linear approximated length have an absolute error less than $h^2/(2\sqrt{2})$, where h is the piece length. If $h = 0.01$, the length error is 1e-4. Suppose the optimal length is 74. Then, the calculated approximate length may be 74.0001.

We consider a path planning case from (x, y, α) to (x', y', β) under the path type *RS* without obstacles and suppose the setting is feasible for *RS*. The floating-point numerical issues are not considered, as we suppose the computers are precise enough. Regarding the analytical/mathematical computation, this optimization problem is:

$$\begin{aligned} &\min_{t,p} t + p \\ &s.t.\ x - \sin(\alpha - t) + \sin(\alpha) + p\cos(\alpha - t) = x' \\ &\quad y + \cos(\alpha - t) - \cos(\alpha) + p\sin(\alpha - t) = y' \\ &\quad \beta = \alpha - t (\bmod\ 2\pi). \end{aligned} \tag{19}$$

Notably, variables t and p can be analytically calculated because $t \in [0, 2\pi)$ can be determined first, and then the first two constraints have only one unknown p. Thus:

$$\begin{aligned} p\cos(\alpha - t) &= x' - x + \sin(\beta) - \sin(\alpha) \\ p\sin(\alpha - t) &= y' - y - \cos(\beta) + \cos(\alpha), \end{aligned} \tag{20}$$

By squaring and summing the two equations, we can solve p, which is a function of the known $\sin(\beta), \sin(\alpha), \cos(\beta), \cos(\alpha), x, y, x', y'$.

For the PWL computation: Let $\hat{t}, \hat{p}$ be the PWL solution. $\hat{t}$ is computed similarly to the analytic t because the mod operator can be transformed via mixed-integer constraints. The PWL solution $\hat{p}$ aims to find the solutions for these piecewise-linear constraints:

$$\begin{aligned} \hat{p}\cos[\alpha - t] &= x' - x + \sin[\beta] - \sin[\alpha] \\ \hat{p}\sin[\alpha - t] &= y' - y - \cos[\beta] + \cos[\alpha], \end{aligned} \tag{21}$$

where [.] denotes piecewise approximation. Therefore, $\hat{p} = f(\sin\alpha, \cos\alpha, \sin\beta, \cos\beta, x, y, x', y')$ where f is a known function with piecewise-linear variables.

To derive the absolute error $|p - \hat{p}|$ requires the following lemma. From the Lagrangian interpolation and tolerance theories [40], if the domain of the function is within $[a, b]$, and there exists an interpolation polynomial $L_n(x) \approx f(x)$ on $[a, b]$, then the difference $\mathrm{R}_n(x) = f(x) - L_n(x)$ is the interpolation remainder. In this paper, $n = 1$ in the Lagrangian interpolation polynomials L_n means linear interpolation.

Lemma 1: Let $f(x) \in C^{n+1}[a, b]$ (this denotes that $f(x)$ is continuously differentiable up to order $n+1$ on $[a, b]$), and let the interpolation points be $a \leqslant x_0 < x_1 < \cdots < x_n \leqslant b$. Then, for the interpolation polynomial $L_n(x)$, for any $x \in [x_i, x_{i+1}]$, we have

$$|R_n(x)| = |f(x) - L_n(x)| = \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_i)\left(x - x_{i+1}\right) \right|,$$

where $x_i < \xi < x_{i+1}$.

This lemma allows deriving the absolute error $|p - \hat{p}|$. Let the difference between the curvy and piecewise-linear sine be $\mathrm{d}\sin x = |sin(x) - sin[x]|$, as depicted in Fig. 4.

Then, according to Lemma 1,

$$\begin{aligned} \mathrm{dsin}x &\approx |\sin x - L_1(x)| \leqslant \frac{|f''(\xi_1)|}{2!} |(x - x_i)(x - x_{i+1})| \\ &\leqslant \frac{1}{2} \cdot \left|f''(\xi_1)\right| \frac{h^2}{4} = \frac{1}{2} \cdot \frac{h^2}{4}, \\ \mathrm{dcos}x &\approx |\cos x - p_1(x)| \leqslant \frac{1}{2} \left|g''(\xi_1)\right| \frac{h^2}{4} = \frac{1}{2} \cdot \frac{h^2}{4}, \end{aligned} \tag{22}$$

where h is the piece length, i.e., the x distance between two consecutive breakpoints. The second derivative functions of the sine and cosine functions are smaller than 1, and the rightmost $\leqslant$ applies the arithmetic-mean–geometric-mean inequality.

We denote the symbol S as the square of $\hat{p}$, $\Delta x = x - x'$, $\Delta y = y - y'$, and symbols $K_1 = \Delta x + \sin\beta - \sin\alpha$, $K_2 = \Delta y - \cos\beta + \cos\alpha$, then, according to Eq.(21):

$$\begin{aligned} S &= \hat{p}^2 = f^2(\sin\alpha, \cos\alpha, \sin\beta, \cos\beta, x, y, x', y') \\ &= (\Delta x + \sin\beta - \sin\alpha)^2 + (\Delta y - \cos\beta + \cos\alpha)^2. \end{aligned} \tag{23}$$

The tolerance of S is (similar as a difference of S):

$$\begin{aligned} \mathrm{d}S &= \frac{\partial S}{\partial \sin\alpha} \cdot \mathrm{d}\sin\alpha + \frac{\partial S}{\partial \cos\alpha}\mathrm{d}\cos\alpha + \frac{\partial S}{\partial \sin\beta}\mathrm{d}\sin\beta \\ &\quad + \frac{\partial S}{\partial \cos\beta}\mathrm{d}\cos\beta = -2(\Delta x + \sin\beta - \sin\alpha)\mathrm{d}\sin\alpha \\ &\quad + 2(\Delta y - \cos\beta + \cos\alpha)\mathrm{d}\cos\alpha \\ &\quad + 2(\Delta x + \sin\beta - \sin\alpha)\mathrm{d}\sin\beta \\ &\quad + (-2)(\Delta y - \cos\beta + \cos\alpha)\mathrm{d}\cos\beta \\ &= -2K_1\mathrm{d}\sin\alpha + 2K_2\mathrm{d}\cos\alpha + 2K_1\mathrm{d}\sin\beta + (-2)K_2\mathrm{d}\cos\beta \\ &\leqslant 2K_1\mathrm{d}\sin\alpha + 2K_2\mathrm{d}\cos\alpha + 2K_1\mathrm{d}\sin\beta + 2K_2\mathrm{d}\cos\beta \\ &\leqslant 2K_1 \cdot \frac{1}{2} \cdot \frac{h^2}{4} + 2K_2 \cdot \frac{1}{2} \cdot \frac{h^2}{4} + 2K_1 \cdot \frac{1}{2} \cdot \frac{h^2}{4} + 2K_2 \cdot \frac{1}{2} \cdot \frac{h^2}{4} \\ &= (K_1 + K_2) \cdot \frac{h^2}{2}, \end{aligned}$$

where the last $\leqslant$ applies Eq. 22.

From the relation $S = \hat{p}^2$, it induces $\mathrm{d}S = 2\hat{p}\mathrm{d}\hat{p}$, and thus the tolerance of p, i.e., $\mathrm{d}\hat{p}$, is expressed as:

$$\begin{aligned} \mathrm{d}\hat{p} &= \frac{\mathrm{d}S}{2\hat{p}} = \frac{\mathrm{d}S}{2\sqrt{(\Delta x + \sin\beta - \sin\alpha)^2 + (\Delta y - \cos\beta + \cos\alpha)^2}} \\ &= \frac{\mathrm{d}S}{2\sqrt{K_1^2 + K_2^2}} \\ &\leqslant \frac{(K_1 + K_2) \cdot \frac{h^2}{2}}{2\sqrt{(K_1^2 + K_2^2)}} \\ &\leqslant \frac{\sqrt{\left(\frac{K_1^2 + K_2^2}{2}\right)} \cdot \frac{h^2}{2}}{\sqrt{K_1^2 + K_2^2}} = \frac{h^2}{2\sqrt{2}}. \end{aligned}$$

The last $\leqslant$ applies the arithmetic-mean–quadratic-mean inequality.

This theoretical analysis demonstrates that the proposed algorithm achieves high precision with nominal error. The following empirical trials involve h taking piece lengths of 0.01, 0.02, 0.04,..., 0.64, and comparing the empirical errors to the theoretically analyzed errors. The start and target configurations are $(0, 0, \pi/2), (3, 1, 0)$. The absolute error of the length (the tolerance) is denoted as a star in the figure. The plotted length error demonstrates that the data fit a quadratic curve well, proving that the developed MIPWLP consistently provides quite accurate and guaranteed solutions, making it a reliable choice for the practical. Since setting piece length as 0.01 can generate paths with absolute error as small as 0.0001, and shorter segment lengths lead to excessive computational costs with diminishing returns, we recommend a piece length of 0.01 for path planning.

V. Discussions

The numerical experiments and analysis confirm the proposed algorithm's effectiveness, accuracy, and efficiency for curvature-constrained path planning with obstacles and free target heading. Deriving parameterized L/S/R primitives, handling obstacles, and the piecewise-linear strategy contribute to our method's exceptional performance, making it applicable and suitable for various surveillance and navigation applications. However, a few concerns should be further discussed.

1) **Robustness, generalizability, and potential applications**: The MIPWLP approach demonstrates robustness in handling complex path planning scenarios with curvature-bounded vehicles, particularly in obstacle-free and obstacle-existing environments. MIPWLP's general ability to manage both discrete and continuous variables allows it to approximate complex trigonometric and geometric constraints effectively. The approach can be extended to various applications [41], including UAV reconnaissance, autonomous ground vehicle navigation, and unmanned surface vehicle (USV) patrolling, where frequent mission planning with minimal energy consumption is critical. These characteristics make MIPWLP suitable for a wide range of path-planning tasks, particularly where curvature constraints, obstacle avoidance, and energy efficiency are essential. For real-world application deployment, several steps are required: First, the scenario should be modeled, and the vehicle's speed should be normalized to one. The MIPWLP computes the optimal heading after inputting the scenario parameters (as demonstrated in the pseudocode) into the MIPWLP solver. Subsequently, a PID or MPC controller can follow the path (as tested in our fixed-wing simulations). Given that the velocity is constant, the planned path is both the shortest and the quickest. Challenges may arise in the modeling and control, while the planning can be quite trustworthy.
2) This work has revealed intriguing findings in the path planning domain. Specifically, we have explored the challenges of finding the shortest path in environments characterized by circular obstacles. Previous work has proven that the decision version of the problem in polygonal obstacle environments with a curvature-bounded path length below a specific threshold belongs to the NP-complete class [24]. This work considers circular environments where a circle is considered a gigantically large number of polygons, and thus it is also NP-complete. The **NP-completeness** can also be viewed from the discrete variables: different breakpoints and selections can differ in the number of variables and constraints [42].
3) We employ mixed-integer programming (MIP) and approximation strategy to at most one obstacle in the two-trip. However, the path type and the target heading variables make the problem challenging. As to whether MIPWLP can effectively address scenarios with **multiple obstacles**, we believe it can, by constraining obstacles sequentially, identifying critical points, and leveraging properties like the perpendicular effect and the lateral cross-product. MIPWLP can also be promising to find the optimal path for **different vehicle types (e.g., Reeds-Shepp vehicle)** and to cooperate with RRT strategies [43].
4) While this paper represents a significant step forward, it is only a small step ahead, and many assumptions are **idealized**, e.g., the effect of wind current is not considered. The proposed approach is based on the distance between the circle centers larger than four, whose optimal path type is CSDSC. However, for dense or close circles, the optimal path type is yet an open question. MIPWLP has the potential to tackle these types (if determined) besides the "CSDSC" because it can introduce segment direction variables f_2 and f_4 to parameterize the path further.
5) Although the performance of our approach has been rigorously evaluated and compared against previous optimal algorithms in some scenarios, it has demonstrated close alignment with optimal values. We have also successfully applied our algorithm to scenarios that previous methods could not address. Combined with a comprehensive quality analysis of relatively simple cases, these findings instill trust in the algorithm's robust performance. We will build and publish a certain **benchmark** in the future.
6) The **error analysis** for a 5-letter two-trip is not yet available because there are several path segments, resulting in several stages, where the directions differ from each other, and each stage may be parameterized by f and t, p, q, r, s. Future work can draw inspiration from [44] and [45], and it may require error propagation tools to establish comprehensive quality guarantees in complex scenarios.

VI. Conclusion

This work presents a novel mathematical programming approach to solve the challenging problem of reconnaissance vehicles operating in obstacle environments with curvature constraints and free target heading. By formulating the reconnaissance scenario as a mathematical program, we effectively handle the challenging trigonometric constraints, periodic properties, and the direction consistencies (clockwise or counterclockwise) in this problem, which other approaches, such as derivative-based approaches, can not address. Our approach employs a piecewise-linear strategy to approximate these constraints, and we propose a unified derivation encompassing all possible primitive types and different numbers of segments to tackle this challenging problem.

The MIPWLP approach incorporates various techniques, resulting in a final mathematical formulation with theoretical properties and a bounded error. We extensively test our algorithm across four sets of scenarios and compare it with several algorithms in four sets of experiments, demonstrating its effectiveness (unification ability to tackle obstacle-free and obstacle-existing scenarios), computational efficiency (a few seconds), high accuracy and error bounds. Robustness, NP-completeness and a few topics about MIPWLP are also discussed thoroughly.

Looking ahead, we aim to apply this algorithm to other similar situations and explore online real-time or distributed implementations by drawing inspiration from convex optimization. Future work will enhance the algorithm's performance and analysis in real-world surveillance applications.

References

[1] Z. Liu, X. Zuo, M. Zhou, W. Guan, and Y. Al-Turki, "Electric vehicle routing problem with variable vehicle speed and soft time windows for perishable product delivery," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 6178–6190, Jun. 2023.

[2] T.-T. Nguyen, N.-H. Tran, T.-M.-D. Ho, and H. Nguyen, "Path planning for unmanned surface vehicle (USV) in obstacle-filled environments," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Oct. 2021, pp. 104–108.

[3] H. Chitsaz and S. M. LaValle, "Time-optimal paths for a Dubins airplane," in *Proc. 46th IEEE Conf. Decis. Control*, 2007, pp. 2379–2384.

[4] L. Li et al., "Collision-free coverage path planning for the variable-speed curvature-constrained robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 3600–3606.

[5] N. Karapetyan, J. Moulton, J. S. Lewis, A. Q. Li, J. M. O'Kane, and I. Rekleitis, "Multi-robot Dubins coverage with autonomous surface vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2373–2379.

[6] A. K. Rao, K. P. Singh, and T. Tripathy, "Curvature bounded trajectories of desired lengths for a Dubins vehicle," *Automatica*, vol. 167, Sep. 2024, Art. no. 111749.

[7] D. Yang, D. Li, and H. Sun, "2D Dubins path in environments with obstacle," *Math. Problems Eng.*, vol. 2013, pp. 1–6, 2013.

[8] A. M. Shkel and V. Lumelsky, "Classification of the Dubins set," *Robot. Auto. Syst.*, vol. 34, no. 4, pp. 179–202, Mar. 2001.

[9] Z. Chen and T. Shima, "Relaxed Dubins problems through three points," in *Proc. 27th Medit. Conf. Control Autom. (MED)*, Jul. 2019, pp. 501–506.

[10] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, Jul. 1957.

[11] B. Moon, S. Sachdev, J. Yuan, and S. Scherer, "Time-optimal path planning in a constant wind for uncrewed aerial vehicles using Dubins set classification," *IEEE Robot. Autom. Lett.*, vol. 9, no. 3, pp. 2176–2183, Mar. 2024.

[12] S. M. LaValle, *Planning Algorithms*. New York, NY, USA: Cambridge Univ. Press, 2006.

[13] J.-D. Boissonnat, A. Cerezo, and J. Leblond, "Shortest paths of bounded curvature in the plane," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1992, pp. 2315–2320.

[14] X.-N. Bui, J.-D. Boissonnat, P. Soueres, and J.-P. Laumond, "Shortest path synthesis for Dubins non-holonomic robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1994, pp. 2–7.

[15] L. Dai, "On the length of Dubins path with any initial and terminal configurations," *Pure Appl. Math. J.*, vol. 4, no. 6, pp. 248–254, 2015.

[16] K. Kučerová, P. Váň, and J. Faigl, "On finding time-efficient trajectories for fixed-wing aircraft using Dubins paths with multiple radii," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 829–831.

[17] D. De Palma and G. Parlangeli, "Shortest path type classification for real-time three-points Dubins problems," in *Proc. 30th Medit. Conf. Control Autom. (MED)*, Jun. 2022, pp. 520–525.

[18] X.-N. Bui and J. Boissonnat, "Accessibility region for a car that only moves forwards along optimal paths," INRIA, Tech. Rep., 1994.

[19] R. Chai, D. Liu, T. Liu, A. Tsourdos, Y. Xia, and S. Chai, "Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 3, pp. 1633–1647, Jul. 2023.

[20] J. Fu, G. Sun, J. Liu, W. Yao, and L. Wu, "On hierarchical multi-UAV Dubins traveling salesman problem paths in a complex obstacle environment," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 123–135, Jan. 2024.

[21] R. Chai, H. Niu, J. Carrasco, F. Arvin, H. Yin, and B. Lennox, "Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 4, pp. 5778–5792, Apr. 2024.

[22] M. Vendittelli, J.-P. Laumond, and C. Nissoux, "Obstacle distance for car-like robots," *IEEE Trans. Robot. Autom.*, vol. 15, no. 4, pp. 678–691, Apr. 1999.

[23] J. Backer and D. Kirkpatrick, "Finding curvature-constrained paths that avoid polygonal obstacles," in *Proc. 23rd Annu. Symp. Comput. Geometry*, 2007, pp. 66–73.

[24] P. K. Agarwal, T. Biedl, S. Lazard, S. Robbins, S. Suri, and S. Whitesides, "Curvature-constrained shortest paths in a convex polygon," *SIAM J. Comput.*, vol. 31, no. 6, pp. 1814–1851, Jan. 2002.

[25] J. Z. Ben-Asher, E. D. Rimon, M. Wetzler, and J. Diepolder, "Time optimal trajectories for a mobile robot with acceleration and speed limits in the presence of an obstacle," *J. Auto. Vehicles Syst.*, vol. 1, no. 1, Jan. 2021.

[26] M. Mosayebi and P. M. Kolahi, "Time optimal trajectory generation with obstacle avoidance by using optimal control theory for a wheeled mobile robot," *Gazi Univ. J. Sci.*, vol. 36, no. 1, pp. 430–439, Mar. 2023.

[27] B. Jha, Z. Chen, and T. Shima, "On shortest Dubins path via a circular boundary," *Automatica*, vol. 121, Nov. 2020, Art. no. 109192.

[28] W. Yao, N. Qi, C. Yue, and N. Wan, "Curvature-bounded lengthening and shortening for restricted vehicle path planning," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 15–28, Jan. 2020.

[29] A. Sadeghi and S. L. Smith, "On efficient computation of shortest Dubins paths through three consecutive points," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 6010–6015.

[30] Z. Chen and T. Shima, "Shortest Dubins paths through three points," *Automatica*, vol. 105, pp. 368–375, Jul. 2019.

[31] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. L. P. Chen, "Design and implementation of deep neural network-based control for automatic parking maneuver process," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1400–1413, Apr. 2022.

[32] J. Herynek, P. Vána, and J. Faigl, "Finding 3D Dubins paths with pitch angle constraint using non-linear optimization," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Aug. 2021, pp. 1–6.

[33] J. Müller, A. V. Ruiz, and I. Wieser, "Safe & sound: A robust collision avoidance layer for aerial robots based on acoustic sensors," in *Proc. IEEE/ION Position, Location Navigat. Symp.-PLANS*, May 2014, pp. 1197–1202.

[34] P. Pharpatara, B. Hérissé, and Y. Bestaoui, "3D-shortest paths for a hypersonic glider in a heterogeneous environment," *IFAC-PapersOnLine*, vol. 48, no. 9, pp. 186–191, 2015.

[35] T. Chen, Y. Cai, L. Chen, and X. Xu, "Trajectory and velocity planning method of emergency rescue vehicle based on segmented three-dimensional quartic Bezier curve," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3461–3475, Mar. 2023.

[36] M. Gerdts, S. Rogovs, and G. Valenti, "A piecewise linearization algorithm for solving MINLP in intersection management," in *Proc. 7th Int. Conf. Vehicle Technol. Intell. Transp. Syst.*, 2021, pp. 438–445.

[37] *Gurobi Optimizer Reference Manual*, 2014.

[38] G. Zhen-yu, Y. Dong-xiao, L. Jie, S. Ya-ling, and J. Zhi-qing, "Obstacle avoidance planning algorithm for uav based on Dubins path," *Trans. Beijing Inst. Technol.*, vol. 34, no. 6, pp. 570–575, 2014.

[39] I. Cohen, C. Epstein, P. Isaiah, S. Kuzi, and T. Shima, "Discretization-based and look-ahead algorithms for the Dubins traveling salesperson problem," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 383–390, Jan. 2017.

[40] M. Abramowitz, I. A. Stegun, and R. H. Romer, *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, 1988.

[41] R. Chai, Y. Guo, Z. Zuo, K. Chen, H. Shin, and A. Tsourdos, "Cooperative motion planning and control for aerial-ground autonomous systems: Methods and applications," *Prog. Aerosp. Sci.*, vol. 146, 2024, Art. no. 101005.

[42] S. Rebennack and V. Krasko, "Piecewise linear function fitting via mixed-integer linear programming," *INFORMS J. Comput.*, vol. 32, no. 2, pp. 507–530, Apr. 2020.

[43] A. Arab, K. Yu, J. Yu, and J. Yi, "Motion planning and control of autonomous aggressive vehicle maneuvers," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 2, pp. 1–13, Apr. 2024.

[44] R. Chai, A. Tsourdos, S. Chai, Y. Xia, A. Savvaris, and C. L. P. Chen, "Multiphase overtaking maneuver planning for autonomous ground vehicles via a desensitized trajectory optimization approach," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 74–87, Jan. 2023.

[45] R. Chai et al., "A two phases multiobjective trajectory optimization scheme for multi-UGVs in the sight of the first aid scenario," *IEEE Trans. Cybern.*, vol. 54, no. 9, pp. 5078–5091, Sep. 2024.

Xing Zhou received the B.Eng. degree in computer science from Hunan University, Changsha, China, in 2012, and the M.S. degree in computer science and the Ph.D. degree in software engineering from the College of Computer, National University of Defense Technology (NUDT), in 2014 and 2019, respectively. He is an Assistant Professor with NUDT. His research interests include robotics, optimization, and artificial intelligence.

Lin Li received the Ph.D. degree in computer science and technology from the National University of Defense Technology (NUDT), China. She currently works at the Intelligent Manufacturing College, Hunan First Normal University. Her current research interests include robot path planning, multi-robot systems, and swarm intelligence.

Xinglong Zhang (Member, IEEE) received the B.E. degree in mechanical engineering from Zhejiang University, Hangzhou, China, in 2011, and the Ph.D. degree in system and control from the Politecnico di Milano, Italy, in 2018. He is an Associate Professor at the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China. His research interests include Koopman operators, learning-based model predictive control, reinforcement learning and approximate dynamic programming, and their applications in automotive systems. He is on the Editorial Board of *Discover Vehicles* in Springer Nature and a Guest Editor of *Frontiers in Robotics and AI*.

Hao Gao is currently pursuing the Ph.D. degree in control science and engineering with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China. His research interests include cooperative multi-agent reinforcement learning, combinatorial optimization, and single-agent reinforcement learning.

Kangxing Yao received the bachelor's degree in aircraft design and engineering from Northwestern Polytechnical University, Xi'an, China, in 2021. He is currently pursuing the Ph.D. degree in control science and engineering with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha, China. His research interests include deep reinforcement learning and multi-agent systems.

Xin Xu (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the College of Mechatronics and Automation, National University of Defense Technology (NUDT), in 2002. He is currently a Professor with the College of Intelligence Science and Technology, NUDT. His current research interests include intelligent control, reinforcement learning, approximate dynamic programming, machine learning, robotics, and autonomous vehicles. He has served as an Associate Editor or a Guest Editor for *Information Sciences*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, *International Journal of Adaptive Control and Signal Processing*, and *Acta Automatica Sinica*.