Embodied Footprints: A Safety-guaranteed Collision Avoidance Model for Numerical Optimization-based Trajectory Planning

Bai Li, Youmin Zhang, *Fellow*, *IEEE*, Tantan Zhang, Tankut Acarman, Yakun Ouyang,
Li Li, *Fellow*, *IEEE*, Hairong Dong, *Senior Member*, *IEEE*, and Dongpu Cao, *Senior Member*, *IEEE*

Abstract—**Numerical optimization-based methods are among the prevalent trajectory planners for autonomous driving. In a numerical optimization-based planner, the nominal continuous-time trajectory planning problem is discretized into a nonlinear program (NLP) problem with finite constraints imposed on finite collocation points. However, constraint violations between adjacent collocation points may still occur. This study proposes a safety-guaranteed collision-avoidance modeling method to eliminate the collision risks between adjacent collocation points in using numerical optimization-based trajectory planners. A new concept called embodied box is proposed, which is formed by enlarging the rectangular footprint of the ego vehicle. If one can ensure that the embodied boxes at finite collocation points are collide-free, then the ego vehicle's footprint is collide-free at any a moment between adjacent collocation points. We find that the geometric size of an embodied box is a simple function of vehicle velocity and curvature. The proposed theory lays a foundation for numerical optimization-based trajectory planners in autonomous driving.**

Index Terms—**Embodied footprint, optimal control, collision avoidance, trajectory planning, motion planning**

I. INTRODUCTION

TRAJECTORY planning, as an essential module in an autonomous vehicle, is responsible for finding a spatio-temporal curve that is kinematically feasible, free from collisions, comfortable for passengers, and efficient in travel time/energy [1,2]. The existing trajectory planners for an autonomous vehicle are classified as search-based, sampling-based, and numerical optimal control-based methods [3–6]. A

Bai Li, Tantan Zhang, and Yakun Ouyang are with the College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China (e-mails: libai@zju.edu.cn, zhangtantan@hnu.edu.cn, yakun@hnu.edu.cn).

Youmin Zhang is with the Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montreal, Canada (e-mail: ymzhang@encs.concordia.ca).

Tankut Acarman is with the Department of Computer Engineering, Galatasaray University, Istanbul, Turkey (e-mail: tacarman@gsu.edu. tr).

Li Li is with the Department of Automation, BNRist, Tsinghua University, Beijing 100084, China (e-mail: li-li@tsinghua.edu.cn).

Hairong Dong is with the School of Electronics and Information Engineering, Beihang University, Beijing 100191, China (e-mail: hrdong@ bjtu.edu.cn)

Dongpu Cao is with the School of Vehicle and Mobility, Tsinghua University, Beijing, China (e-mail: dongpu.cao@uwaterloo.ca).
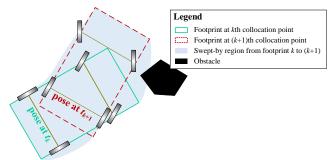
Fig. 1. Schematics on the violations of collision-avoidance constraints between adjacent collocation points. Two discretized footprints are collision-free but collisions occur between them.

search-/sampling-based planner coarsely finds a sub-optimal path/trajectory with global optimality while a numerical optimal control-based planner pursues local optimality. This paper is focused on numerical optimal control-based planning methods.

A numerical optimal control-based planner describes a nominal trajectory planning scheme as an optimal control problem (OCP). The OCP is discretized into a nonlinear program (NLP) problem before it is solved by a gradient-based NLP solver [7]. In this way, the original continuous-time OCP is converted into a problem with finite decision variables (denoted as collocation points) as well as finite constraints [8]. Unfortunately, the conversion is inequivalent because the time-continuous constraints in the original OCP are *only* imposed on those finite collocation points. This means that the constraint satisfaction between two adjacent collocation points is ignored. Setting the collocation points densely is a practical remedy for this ignorance, but 1) the constraint ignorance issue is still not thoroughly addressed, and 2) the formulated NLP is in an overly high dimension which requires excessive runtime. This constraint ignorance issue has been a common and silent drawback of numerical optimal control-based planners, especially when the constraints are highly nonlinear [9–11].

Collision-avoidance constraints are commonly known as the most complex constraint type in a trajectory planning problem [12]. As depicted in Fig. 1, ignorance of collision-avoidance constraints between adjacent collocation points easily makes a planned trajectory invalid [13]. This study aims to propose a constraint formulation approach that theoretically guarantees that a numerical optimal control-based trajectory planner is safe between adjacent collocation points.
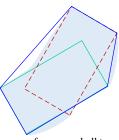
Fig. 2. Schematics on the usage of a convex hull to cover non-collocation point footprints. The edges of the convex hull are colored blue, which does not fully cover the swept-by region.
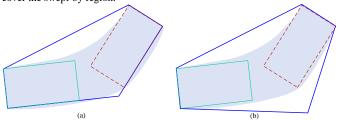

| (a) | (b) |

Fig. 3. Schematics on the usage of a polygonal region to cover the swept-by region between two adjacent vehicle footprints: (a) motion polygon proposed in [22]; (b) polygonal over-approximation method proposed in [23].
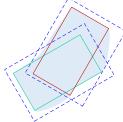

Fig. 4. Schematics on the usage of virtual protection frames to cover the swept-by region. Notably, the entire swept-by region is covered by two rectangles (i.e. the blue dashed boxes), which are derived by expanding the ego vehicle's footprints.

A. Related Works

This subsection reviews the previous studies that considered the violations of collision-avoidance constraints between adjacent collocation points/waypoints. For simplicity, violation of collision-avoidance constraints between adjacent collocation points is referred to as *non-collocation error* in the remainder of this paper.

Most of the previous works hold a silent assumption that the non-collocation errors do not occur if 1) the collocation points are *sufficiently* dense [14,15], 2) each polygonal obstacle is *sufficiently* dilated to form a large buffer [16,17], or 3) a cost function with *sufficiently* large penalty weights can repel the planned trajectory from being close to obstacles [18]. However, the absence of quantitative analysis about the *sufficiency* prevents such experiences from becoming unified knowledge.

A convex hull was naturally considered to cover the region swept by the vehicle footprint between two adjacent collocation points [19,20]. However, simply using a convex hull does not really guarantee to cover the whole swept-by region (Fig. 2). Schulman et al. [21] mentioned that the convex hull deserves to be expanded but they did not elaborate on how the expansion is implemented. Instead of imposing hard constraints, they suggested adding a repulsive penalty in the cost function as a practical remedy for the non-collocation error issue. However, collision risks would always exist unless hard constraints are

imposed, thus [21] is not a thorough solution. Scheuer and Fraichard [22] proposed a motion polygon to cover the swept-by region between adjacent collocation points, but part of the swept-by region is still out of the motion polygon (Fig. 3a). Besides that, their motion polygon cannot deal with the cases where adjacent poses are overlapped. Ghita and Kloetzer [23] proposed a polygonal over-approximation method, which is about building an expanded polygon with the intersection of two tangents to the swept-by region (Fig. 3b). However, this method is still too complex for a gradient-based NLP solver because the vertexes of the swept-by region are not easy to present. As seen from Fig. 3, the swept-by region is rather over-approximated in [22] or [23], which leads to overcautious trajectories or even solution failures (we prove this argument via simulations in Section IV). To summarize, modeling the swept-by region as a compact polygon is inapplicable.

In contrast to using compact polygons, Li et al. [24] deployed multiple polygons to cover the swept-by region. Concretely, equidistant waypoints between two adjacent collocation points are interpolated, and then a collision-avoidance constraint on each waypoint is imposed. However, this strategy is still not thorough because minor collision risks still exist. Zhang et al. [25] proposed a virtual protection frame method, which covers the swept-by region by expanded rectangles at the collocation points (Fig. 4). As opposed to the aforementioned compact polygon models, [25] guarantees that the OCP dimension does not increase after considering the non-collocation error issue. However, the size of each expanded rectangle is determined by trial and error, thus the OCP solution process has to be conducted repeatedly, which consumes much runtime. Similar ideas include adaptive mesh refinement by Yershov and Frazzoli [26], and moving finite element method by Chen et al. [27]. In [26], the trajectory resolution is improved by repeatedly adjusting the discretized mesh grids; [27] repeatedly moves the non-uniform collocation points before the non-collocation error is reduced to an acceptable level. However, either [26] or [27] still works in a trial-and-error mode, which hardly meets the real-time computation demand.

As a conclusion of the whole subsection, no previous study thoroughly addressed the non-collocation error issue with solution completeness and planning runtime both considered.

B. Motivations and Contributions

Reviewing the previous studies, we notice that 1) promoting the collocation point density would not thoroughly address the concerned issue, and 2) the compact polygon-based methods are overly complex. The virtual protection frame method is promising because it addresses the non-collocation error issue by operating *only* on collocation points. However, the virtual protection frame method is imperfect because it calls for an iterative planning process to determine the expansion buffers, which consumes much runtime. Motivated by the imperfect virtual protection frame method, this work expects to define the expansion buffers theoretically without a time-consuming iterative computation process.

The contribution of this work is the theoretical proposal of a collision-avoidance constraint modeling method for numerical optimal control-based planners that guarantees thorough safety between the adjacent collocation points. Concretely, the vehicle footprint at each collocation point is enlarged by buffers such

that the swept-by region between adjacent collocation points is fully covered; the buffer sizes are simple functions of the vehicle velocity and steering angle, which makes the proposed modeling method easy to implement in a numerical optimal control-based planner.

C. Organization

In the rest of this paper, Section II states the concerned problem in a generic trajectory planning background. Section III proposes a safe collision-avoidance constraint modeling method. Simulation and experimental results are provided and discussed in Section IV. Section V concludes the whole paper.

II. PROBLEM STATEMENT

This section states the safe collision-avoidance constraint formulation problem. As a preliminary step, we present a generic formulation of a trajectory planning task in the form of an OCP. Thereafter, the OCP is discretized into an NLP. The problem statement is made based on the NLP formulation.

A. Trajectory Planning Task Description

A trajectory planning task is formulated as the following OCP:

$$\underset{\mathbf{z}(t),\,\mathbf{u}(t),\,T}{\text{minimize }} J,$$

$$\text{s.t., } \dot{\mathbf{z}}(t) = f\big(\mathbf{z}(t),\,\mathbf{u}(t)\big),\ t \in [0,T];$$

$$\underline{\mathbf{z}} \leq \mathbf{z}(t) \leq \overline{\mathbf{z}},\ \underline{\mathbf{u}} \leq \mathbf{u}(t) \leq \overline{\mathbf{u}},\ t \in [0,T]; \quad (1)$$

$$\mathbf{z}(0) = \mathbf{z}_{\text{init}},\ \mathbf{u}(0) = \mathbf{u}_{\text{init}};$$

$$\mathbf{z}(T) = \mathbf{z}_{\text{end}},\ \mathbf{u}(T) = \mathbf{u}_{\text{end}};$$

$$fp(\mathbf{z}(t)) \subset \Upsilon_{\text{FREE}},\ t \in [0,T].$$

Herein, $\mathbf{z} \in \mathbb{R}^{n_z}$ represents the vehicle state profiles, $\mathbf{u} \in \mathbb{R}^{n_u}$ denotes the control profiles, and T represents the unfixed planning horizon length.

The kinematic constraints $\dot{\mathbf{z}}(t) = f\big(\mathbf{z}(t),\,\mathbf{u}(t)\big)$ are presented by the well-known single-track bicycle model [7]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x(t) \\ y(t) \\ v(t) \\ \phi(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} v(t) \cdot \cos\theta(t) \\ v(t) \cdot \sin\theta(t) \\ a(t) \\ \omega(t) \\ v(t) \cdot \tan\phi(t)/\mathrm{L_W} \end{bmatrix},\ t \in [0,T]. \quad (2)$$

Herein, (x, y) represents the coordinate value of the midpoint along the rear-wheel axle of the ego vehicle (Fig. 5), θ refers to the orientation angle, v is the vehicle velocity, ϕ represents the steering angle, a is the acceleration, ω denotes the angular velocity of the steering angle, and $\mathrm{L_W}$ is the wheelbase. The other geometric parameters marked in Fig. 5 include $\mathrm{L_F}$ (front hang length plus wheelbase), $\mathrm{L_R}$ (rear hang length), and $\mathrm{L_B}$ (width).

$[\underline{\mathbf{z}}, \overline{\mathbf{z}}]$ and $[\underline{\mathbf{u}}, \overline{\mathbf{u}}]$ denote the allowable intervals for $\mathbf{z}(t)$ and $\mathbf{u}(t)$, respectively, that is,

$$\begin{bmatrix} \mathrm{a_{min}} \\ 0 \\ -\Omega_{\text{max}} \\ -\Phi_{\text{max}} \end{bmatrix} \leq \begin{bmatrix} a(t) \\ v(t) \\ \omega(t) \\ \phi(t) \end{bmatrix} \leq \begin{bmatrix} \mathrm{a_{max}} \\ \mathrm{v_{max}} \\ \Omega_{\text{max}} \\ \Phi_{\text{max}} \end{bmatrix},\ t \in [0,T]. \quad (3)$$

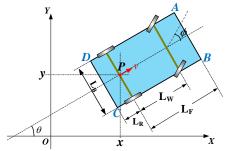Fig. 5. Schematics on vehicle kinematics and geometrics.

Here, $v(t) \geq 0$ indicates that backward maneuvers are ignored in this present work. $\mathrm{a_{min}}$, $\mathrm{a_{max}}$, $\mathrm{v_{max}}$, Ω_{max}, and Φ_{max} are boundary parameters.

$\mathbf{z}_{\text{init}}$ and $\mathbf{u}_{\text{init}}$ denote the initial values of $\mathbf{z}(t)$ and $\mathbf{u}(t)$. $\mathbf{z}_{\text{end}}$ and $\mathbf{u}_{\text{end}}$ denote their endpoint values, respectively.

$fp(\cdot): \mathbb{R}^{n_z} \to \mathbb{R}^2$ is a mapping from the vehicle's state profile $\mathbf{z}(t)$ to the footprint; thus, Υ_{FREE} denotes the free space in the 2D environment, $fp(\mathbf{z}(t)) \subset \Upsilon_{\text{FREE}}$ represents the collision-avoidance constraints. This work assumes that Υ_{FREE} is static, i.e., no moving obstacles are considered. For future usage, let us define the four vertexes of the ego vehicle's footprint as $A = (x_A, y_A)$, $B = (x_B, y_B)$, $C = (x_C, y_C)$, and $D = (x_D, y_D)$ (Fig. 5):

$$x_A(t) = x(t) + \mathrm{L_F} \cdot \cos\theta(t) - 0.5\mathrm{L_B} \cdot \sin\theta(t),$$

$$y_A(t) = y(t) + \mathrm{L_F} \cdot \sin\theta(t) + 0.5\mathrm{L_B} \cdot \cos\theta(t),$$

$$x_B(t) = x(t) + \mathrm{L_F} \cdot \cos\theta(t) + 0.5\mathrm{L_B} \cdot \sin\theta(t),$$

$$y_B(t) = y(t) + \mathrm{L_F} \cdot \sin\theta(t) - 0.5\mathrm{L_B} \cdot \cos\theta(t),$$

$$x_C(t) = x(t) - \mathrm{L_R} \cdot \cos\theta(t) + 0.5\mathrm{L_B} \cdot \sin\theta(t), \quad (4)$$

$$y_C(t) = y(t) - \mathrm{L_R} \cdot \sin\theta(t) - 0.5\mathrm{L_B} \cdot \cos\theta(t),$$

$$x_D(t) = x(t) - \mathrm{L_R} \cdot \cos\theta(t) - 0.5\mathrm{L_B} \cdot \sin\theta(t),$$

$$y_D(t) = y(t) - \mathrm{L_R} \cdot \sin\theta(t) + 0.5\mathrm{L_B} \cdot \cos\theta(t),\ t \in [0,T].$$

The concrete expression of $fp(\mathbf{z}(t)) \subset \Upsilon_{\text{FREE}}$ is not presented here because it is not the focus of this paper.

The cost function J is set to the completion time of the driving process, i.e., $J = T$. Other time-energy-optimal cost functions could also be used, but formulating a purely time-optimal OCP easily renders that the planned trajectories are close to the barriers of obstacles, which leverage the effect of our proposed method in the next section. That is why we set $J = T$.

B. NLP Formulation

The preceding subsection presents the trajectory planning scheme as an OCP, which is discretized into an NLP in this subsection. For simplicity, the OCP built in the preceding subsection is abstracted as

$$\underset{\mathbf{z}(t),\,\mathbf{u}(t),\,T}{\text{minimize }} J,$$

$$\text{s.t., } g\big(\mathbf{z}(t),\,\dot{\mathbf{z}}(t),\,\mathbf{u}(t)\big) \leq 0,\ t \in [0,T]. \quad (5)$$

Here, $g \leq 0$ represents all the constraints in inequality and equality forms.

Discretizing (5) into an NLP is about sampling finite moments along $t \in [0,T]$ such that each $\mathbf{z}(t)$ or $\mathbf{u}(t)$ could be represented by finite collocation points. In this way, the constraints $g \leq 0$ are imposed only on finite collocation points instead of on the entire time domain $[0,T]$. Suppose that $(\mathrm{N_{fe}} + 1)$ moments are sampled from 0 to T, which are collected in a

set $\{t_i | i = 0, ..., N_{fe}\}$ with
$$0 = t_0 < t_1 < t_2 < ... < t_{N_{fe}} = T. \quad (6)$$

Each $\mathbf{z}(t)$ is represented by ($N_{fe} + 1$) collocation points, which are denoted by $\{z_i | i = 0, ..., N_{fe}\}$. These collocation points are used to construct an infinite-dimensional variable $\mathbf{z}(t)$, which might be a piecewise constant, linear, or polynomial function. Similarly, $\mathbf{u}(t)$ is represented by $\{u_i | i = 0, ..., N_{fe}\}$. In this way, the original OCP is discretized into the following form:

$$\underset{z_i, u_i, T}{\text{minimize}} \ J(z_i, u_i, T),$$
$$\text{s.t., } g(z_i, u_i) \le 0, \ i = 0, ..., N_{fe}. \quad (7)$$

This work assumes that $\mathbf{z}(t)$ or $\mathbf{u}(t)$ is a *piecewise constant* function. Under this assumption, the discretization method is known as the explicit first-order Runge–Kutta method. Notably, the proposed method in Section III relies on this assumption.

C. Problem Statement

Nominally, one solves (7) via an NLP solver to derive a numerical solution to the original OCP (5). But the derived trajectory is not guaranteed to be safe because collision-avoidance constraints are only imposed on finite moments $\{t_i\}$. To address this issue thoroughly, we expect that each of the finite collision-avoidance constraints is "responsible". Specifically, the swept-by region from t_k to t_{k+1} is fully covered by an embodied box over the ego vehicle's footprint at t_k. As shown in Fig. 6, each embodied box is aligned with the vehicle footprint; the size of each embodied box is determined by four variables at t_k, which are referred to as $e_{\text{left}}(t_k)$, $e_{\text{right}}(t_k)$, $e_{\text{up}}(t_k)$, and $e_{\text{down}}(t_k)$. Obviously, e_{left}, e_{right}, e_{up}, and e_{down} are functions of $\phi(t_k)$ and $v(t_k)$. The core problem in this work is how to define the four variables such that the collision risks between adjacent collocation points are thoroughly eliminated.

III. EMBODIED BOX BASED COLLISION AVOIDANCE

A. Problem Normalization

As stated in Section II.C, the core problem is to define four variables $e_{\text{left}}(t_k)$, $e_{\text{right}}(t_k)$, $e_{\text{up}}(t_k)$, and $e_{\text{down}}(t_k)$ such that the swept-by region from t_k to t_{k+1} is fully covered by an embodied box at t_k. Without loss of generality, we rotate the vehicle in Fig. 6 to an axially aligned pose (see Fig. 7) for the convenience of analysis, i.e., $[x(t_k), y(t_k), \theta(t_k)] = [0, 0, \pi/2]$. Owing to this rotation, the four variables can be easily presented as

$$e_{\text{left}}(t_k) = -\frac{L_B}{2} - x_{\min},$$

$$e_{\text{right}}(t_k) = x_{\max} - \frac{L_B}{2},$$

$$e_{\text{up}}(t_k) = y_{\max} - L_F, \quad (8)$$

$$e_{\text{down}}(t_k) = -L_R - y_{\min},$$

where $x_{\min}$, $x_{\max}$, $y_{\min}$, and $y_{\max}$ respectively denote the lower/upper bounds of the swept-by region. The determination of $x_{\min}$, $x_{\max}$, $y_{\min}$, and $y_{\max}$ naturally determines $e_{\text{left}}(t_k)$, $e_{\text{right}}(t_k)$, $e_{\text{up}}(t_k)$, and $e_{\text{down}}(t_k)$.

Let us focus on the swept-by region from t_k to t_{k+1}. The piecewise constant assumption made in Section II.B yields that the steering angle variable $\phi(t)$ remains constant during $t \in [t_k, t_{k+1}]$, thus the path of the rear-axle midpoint located at $(x(t), y(t))$ is circular. Similarly, the velocity variable $v(t)$ is constant during $t \in [t_k, t_{k+1}]$. For simplicity, let us rewrite the
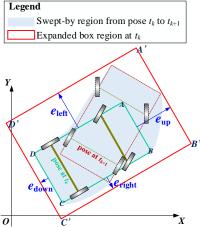

Fig. 6. Schematics on embodied box $A'B'C'D'$ that covers the swept-by region from t_k to t_{k+1}.
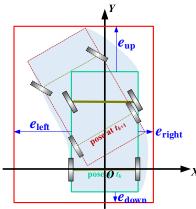

Fig. 7. A normalized angle of review of Fig. 6 to facilitate theoretical analysis.

constant steering angle as ϕ_k and the constant velocity as v_k. According to (2) and the Newton–Leibniz formula, the orientation angle $\theta(t)$ is determined as

$$\theta(t) = \theta(t_k) + \int_{\tau = t_k}^{t} \frac{v(t) \cdot \tan \phi(t)}{L_W} d\tau$$
$$= \frac{\pi}{2} + \frac{v_k \cdot \tan \phi_k}{L_W} \cdot (t - t_k), \ t \in [t_k, t_{k+1}]. \quad (9)$$

To further simplify the presentation, let us introduce a curvature variable $\kappa(t) \equiv \tan \phi(t) / L_W$ and set that $t_k = 0$ and $t_{k+1} = \Delta T$, then we have

$$\theta(t) = \frac{\pi}{2} + v_k \cdot \kappa_k \cdot t, \ t \in [0, \Delta T], \quad (10a)$$

where $\kappa_k = \tan \phi_k / L_W$. Similarly, $x(t)$ and $y(t)$ are defined as

$$x(t) = \frac{\cos(v_k \cdot \kappa_k \cdot t) - 1}{\kappa_k}, \ t \in [0, \Delta T], \quad (10b)$$

and

$$y(t) = \frac{\sin(v_k \cdot \kappa_k \cdot t)}{\kappa_k}, \ t \in [0, \Delta T]. \quad (10c)$$

Subscribing (10) into (4) yields the trajectories of the ego vehicle's four vertex points:

$$x_A(t) = \frac{\cos(v_k \cdot \kappa_k \cdot t) - 1}{\kappa_k} - L_F \cdot \sin(v_k \cdot \kappa_k \cdot t) - \frac{L_B}{2} \cdot \cos(v_k \cdot \kappa_k \cdot t),$$

$$y_A(t) = \frac{\sin(v_k \cdot \kappa_k \cdot t)}{\kappa_k} + L_F \cdot \cos(v_k \cdot \kappa_k \cdot t) - \frac{L_B}{2} \cdot \sin(v_k \cdot \kappa_k \cdot t),$$

$$x_B(t) = \frac{\cos(v_k \cdot \kappa_k \cdot t) - 1}{\kappa_k} - L_F \cdot \sin(v_k \cdot \kappa_k \cdot t) + \frac{L_B}{2} \cdot \cos(v_k \cdot \kappa_k \cdot t),$$

$$y_B(t) = \frac{\sin(v_k \cdot \kappa_k \cdot t)}{\kappa_k} + L_F \cdot \cos(v_k \cdot \kappa_k \cdot t) + \frac{L_B}{2} \cdot \sin(v_k \cdot \kappa_k \cdot t),$$

$$x_C(t) = \frac{\cos(v_k \cdot \kappa_k \cdot t) - 1}{\kappa_k} + L_R \cdot \sin(v_k \cdot \kappa_k \cdot t) + \frac{L_B}{2} \cdot \cos(v_k \cdot \kappa_k \cdot t),$$

$$y_C(t) = \frac{\sin(v_k \cdot \kappa_k \cdot t)}{\kappa_k} - L_R \cdot \cos(v_k \cdot \kappa_k \cdot t) + \frac{L_B}{2} \cdot \sin(v_k \cdot \kappa_k \cdot t),$$

$$x_D(t) = \frac{\cos(v_k \cdot \kappa_k \cdot t) - 1}{\kappa_k} + L_R \cdot \sin(v_k \cdot \kappa_k \cdot t) - \frac{L_B}{2} \cdot \cos(v_k \cdot \kappa_k \cdot t),$$

$$y_D(t) = \frac{\sin(v_k \cdot \kappa_k \cdot t)}{\kappa_k} - L_R \cdot \cos(v_k \cdot \kappa_k \cdot t) - \frac{L_B}{2} \cdot \sin(v_k \cdot \kappa_k \cdot t),$$

$$t \in [0, \Delta T]. \tag{11}$$

Connecting the four vertexes A, B, C, and D would yield a rectangular footprint, which moves throughout $t \in [0, \Delta T]$ to form a swept-by region. Obviously, boundaries of the swept-by region are caused by *vertexes* rather than *edges* of the footprint, thus

$$\begin{aligned}
x_{min} &= \min\left\{x_A(t), x_B(t), x_C(t), x_D(t), \ \forall t \in [0, \Delta T]\right\}, \\
x_{max} &= \max\left\{x_A(t), x_B(t), x_C(t), x_D(t), \ \forall t \in [0, \Delta T]\right\}, \\
y_{min} &= \min\left\{y_A(t), y_B(t), y_C(t), y_D(t), \ \forall t \in [0, \Delta T]\right\}, \\
y_{max} &= \max\left\{y_A(t), y_B(t), y_C(t), y_D(t), \ \forall t \in [0, \Delta T]\right\}.
\end{aligned} \tag{12}$$

The remainder of this whole section is focused on how to identify the four bounds x_{min}, x_{max}, y_{min}, and y_{max}.

B. Determination of y_{min}

This subsection identifies y_{min}. According to the definitions of $y_A(t)$, $y_B(t)$, $y_C(t)$, and $y_D(t)$ in (11), one can safely state that $y_A(t) \geq y_D(t)$ and $y_B(t) \geq y_C(t)$ for any t provided that

$$\left|v_k \cdot \kappa_k \cdot t\right| \leq \frac{\pi}{2}, \ \forall t \in [0, \Delta T]. \tag{13}$$

Given that (13) holds for any t, thus

$$\left|v_k \cdot \kappa_k \cdot \Delta T\right| \leq \frac{\pi}{2}. \tag{14}$$

Eq. (14) ensures that $y_A(t) \geq y_D(t)$ and $y_B(t) \geq y_C(t)$ for any t within $[0, \Delta T]$. Therefore, y_{min} would be the smaller one between $\min\left\{y_C(t)\right\}$ and $\min\left\{y_D(t)\right\}$ for $\forall t \in [0, \Delta T]$.

Recall that negative velocity is not considered in this present work as mentioned in (3), thus $v_k \geq 0$. The subsequent analyses are divided into two branches as per the sign of κ_k.

Condition 1: $\kappa_k \geq 0$

When κ_k is non-negative, one has $y_D(t) \leq y_C(t)$. Thus, y_{min} is the extremum of $y_D(t)$ on $t \in [0, \Delta T]$. Let us rewrite $y_D(t)$ in the following form:

$$\begin{aligned}
y_D(t) &= (\frac{1}{\kappa_k} - \frac{L_B}{2}) \cdot \sin(v_k \cdot \kappa_k \cdot t) - L_R \cdot \cos(v_k \cdot \kappa_k \cdot t) \\
&= M \cdot \sin(v_k \cdot \kappa_k \cdot t) - N \cdot \cos(v_k \cdot \kappa_k \cdot t) \\
&= \sqrt{M^2 + N^2} \cdot \sin(v_k \cdot \kappa_k \cdot t - \alpha),
\end{aligned} \tag{15}$$

where $M \equiv \frac{1}{\kappa_k} - \frac{L_B}{2}$, $N \equiv L_R$, and

$$\alpha = \arccos(\frac{M}{\sqrt{M^2 + N^2}}). \tag{16}$$

Herein, N is positive while the sign of M is pending. Let us discuss the sign of M.

If $M < 0$, then $M / \sqrt{M^2 + N^2}$ is negative, thus (16) yields that $\alpha \in [\pi/2, \pi]$. Let us analyze the monotonicity of $y_D(t)$ via its derivative $y_D'(t)$:

$$y_D'(t) = v_k \cdot \kappa_k \cdot \sqrt{M^2 + N^2} \cdot \cos(v_k \cdot \kappa_k \cdot t - \alpha).$$

Given that $v_k \cdot \kappa_k \cdot \sqrt{M^2 + N^2} > 0$, the sign of $y_D'(t)$ would be determined by $\cos(v_k \cdot \kappa_k \cdot t - \alpha)$. Obviously, $y_D'(0) \leq 0$, which means $y_D(t)$ has a descending trend at $t = 0^+$. We expect that the entire domain $[0, \Delta T]$ is a monotonic decreasing interval. If so, the minimal extremum value is $y_D(\Delta T)$. The reason why we expect $[0, \Delta T]$ to be monotonic is given as follows. If $[0, \Delta T]$ is not a monotonic decreasing interval, then the extremum is achieved at some $t^* \in [0, \Delta T]$ that satisfies

$$\sin(v_k \cdot \kappa_k \cdot t^* - \alpha) = -1,$$

which renders the following side effects: 1) the extremum is unrelated to v_k, thus making y_{min} overcautious; and 2) the extremum is a rather complex function of κ_k.

To make $[0, \Delta T]$ a monotonic decreasing interval, one needs to ensure that $\cos(v_k \cdot \kappa_k \cdot t - \alpha)$ remains negative throughout $[0, \Delta T]$, which yields $\cos(v_k \cdot \kappa_k \cdot \Delta T - \alpha) \leq 0$, i.e.,

$$v_k \cdot \kappa_k \cdot \Delta T - \alpha \leq -\frac{\pi}{2}. \tag{17}$$

This inequality is further described as

$$0 < v_k \cdot \kappa_k \cdot \Delta T \leq \arctan(-\frac{\frac{1}{\kappa_k} - \frac{L_B}{2}}{L_R}). \tag{18}$$

If (18) holds, then the extremum of $y_D(t)$ is

$$y_D(\Delta T) = (\frac{1}{\kappa_k} - \frac{L_B}{2}) \cdot \sin(v_k \cdot \kappa_k \cdot \Delta T) - L_R \cdot \cos(v_k \cdot \kappa_k \cdot \Delta T). \tag{19}$$

Nominally, one can state $y_{min} = y_D(\Delta T)$, but (19) is still not showing a simple enough relationship among y_{min}, v_k, and κ_k. Therefore, we simplify (19) further via inequality amplification skills in mathematics. According to the McLaughlin formula, one has

$$\sin(v_k \cdot \kappa_k \cdot \Delta T) \leq v_k \cdot \kappa_k \cdot \Delta T, \tag{20a}$$

and

$$\cos(v_k \cdot \kappa_k \cdot \Delta T) \leq 1. \tag{20b}$$

Then, we can provide a simplified lower bound for $y_D(\Delta T)$:

$$(\frac{1}{\kappa_k} - \frac{L_B}{2}) \cdot v_k \cdot \kappa_k \cdot \Delta T - L_R \leq y_D(\Delta T). \tag{21}$$

Accordingly, one has

$$y_{min} = \frac{L_B}{2} \cdot \kappa_k \cdot s_k - s_k - L_R, \tag{22}$$

wherein $s_k \equiv v_k \cdot \Delta T$ is deployed to simplify the presentation.

The aforementioned analysis is based on a prerequisite that $M < 0$. Symmetrically, if $M \geq 0$, then $\alpha \in [0, \pi/2]$ and $[0, \Delta T]$ is definitely a monotonic increasing interval. Therefore, the extremum value is $y_D(0)$, i.e., $y_{min} = -L_R$.

Condition 2: $\kappa_k < 0$

In dealing with $\kappa_k < 0$, we introduce a temporary variable $\kappa_k^* = -\kappa_k$. Since $\kappa_k^* > 0$, the analysis in **Condition 1** could be repeated. The concrete details are similar, thus we omit them.

Summarizing the analyses for $\kappa_k \geq 0$ and $\kappa_k < 0$ yields that

1° When $\dfrac{1}{|\kappa_k|} - \dfrac{L_B}{2} \leq 0$,

$$y_{\min} = \frac{L_B}{2} \cdot |\kappa_k| \cdot s_k - s_k - L_R, \qquad (23a)$$

which is associated with prerequisite

$$|\kappa_k| \cdot s_k \leq \arctan(-\frac{\dfrac{1}{|\kappa_k|} - \dfrac{L_B}{2}}{L_R}). \qquad (23b)$$

2° When $\dfrac{1}{|\kappa_k|} - \dfrac{L_B}{2} > 0$,

$$y_{\min} = -L_R, \qquad (23c)$$

which is associated with prerequisite

$$|\kappa_k| \cdot s_k \leq \frac{\pi}{2}. \qquad (23d)$$

Empirically, $2L_W > L_B \cdot \tan\phi_{\max}$ holds for a passenger vehicle, thus we have

$$\frac{1}{|\kappa_k|} - \frac{L_B}{2} = \left|\frac{L_W}{\tan\phi_k}\right| - \frac{L_B}{2} \geq \frac{L_W}{\tan\phi_{\max}} - \frac{L_B}{2} > 0. \qquad (24)$$

Therefore, the branch under $1/|\kappa_k| - L_B/2 \leq 0$ is discarded. The final conclusion of this subsection is that $y_{\min} = -L_R$ with $|\kappa_k| \cdot s_k \leq \pi/2$. We defaultly ignore the cases that involve $1/|\kappa_k| - L_B/2 \leq 0$ in the remainder of this section.

C. Determinations of y_{max}, x_{min}, and x_{max}

This subsection presents the definitions of $y_{\min}$, $x_{\min}$, and $x_{\max}$. The analyses are similar to those in Section III.B. The details are omitted.

$y_{\max}$ is written as

$$y_{\max} = s_k + \frac{L_B}{2} \cdot |\kappa_k| \cdot s_k + L_F, \qquad (25a)$$

together with the prerequisite

$$s_k \cdot |\kappa_k| \leq \arctan(\frac{\dfrac{1}{|\kappa_k|} + \dfrac{L_B}{2}}{L_F}). \qquad (25b)$$

$x_{\min}$ is written as

$$x_{\min} = -\frac{L_B}{2} - \max\left\{-L_R \cdot \kappa_k \cdot s_k, \ (L_F + \frac{s_k}{2}) \cdot \kappa_k \cdot s_k\right\}, \qquad (26a)$$

together with the prerequisite

$$s_k \cdot |\kappa_k| \leq \arctan(\frac{L_R}{\dfrac{1}{|\kappa_k|} + \dfrac{L_B}{2}}). \qquad (26b)$$

$x_{\max}$ is written as

$$x_{\max} = \frac{L_B}{2} + \max\left\{L_R \cdot \kappa_k \cdot s_k, -(L_F + \frac{s_k}{2}) \cdot \kappa_k \cdot s_k\right\}, \qquad (27a)$$

together with the prerequisite

$$s_k \cdot |\kappa_k| \leq \arctan(\frac{L_R}{\dfrac{1}{|\kappa_k|} + \dfrac{L_B}{2}}). \qquad (27b)$$

D. Definition of an Embodied box

Fig. 7 yields that

$$(-x_{\min}) - e_{\text{left}}(t_k) = \frac{L_B}{2},$$

$$x_{\max} - e_{\text{right}}(t_k) = \frac{L_B}{2}, \qquad (28)$$

$$(-y_{\min}) - e_{\text{down}}(t_k) = L_R,$$

$$y_{\max} - e_{\text{up}}(t_k) = L_F.$$

Eq.(28), Section III.B, and III.C render that

$$e_{\text{left}}(t_k) = \max\left\{-L_R \cdot \kappa_k \cdot s_k, \ (L_F + \frac{s_k}{2}) \cdot \kappa_k \cdot s_k\right\},$$

$$e_{\text{right}}(t_k) = \max\left\{L_R \cdot \kappa_k \cdot s_k, -(L_F + \frac{s_k}{2}) \cdot \kappa_k \cdot s_k\right\}, \qquad (29)$$

$$e_{\text{up}}(t_k) = s_k + \frac{L_B}{2} \cdot |\kappa_k| \cdot s_k,$$

$$e_{\text{down}}(t_k) = 0,$$

together with an intersection of prerequisites (23d), (25b), (26b), and (27b). Let us focus on the prerequisites (23d), (25b), (26b), and (27b). Eqs. (26b) and (27b) are identical, thus we only consider (23d), (25b), and (26b).

Eq. (25b) is rewritten as

$$\tan\left(s_k \cdot |\kappa_k|\right) \leq \frac{\dfrac{1}{|\kappa_k|} + \dfrac{L_B}{2}}{L_F}, \qquad (30a)$$

that is,

$$|\kappa_k| \cdot L_F \cdot \tan\left(s_k \cdot |\kappa_k|\right) \leq 1 + \frac{L_B}{2} \cdot |\kappa_k|. \qquad (30b)$$

Similarly, (26b) is rewritten as

$$\left(1 + \frac{L_B}{2} \cdot |\kappa_k|\right) \cdot \tan\left(s_k \cdot |\kappa_k|\right) \leq L_R \cdot |\kappa_k|. \qquad (31)$$

Based on the aforementioned analyses, the four vertexes of the ego vehicle at time instance t_k are expanded as the rectangle $A'B'C'D'$ shown in Fig. 6, where $e_{\text{left}}(t_k)$, $e_{\text{right}}(t_k)$, $e_{\text{up}}(t_k)$, and $e_{\text{down}}(t_k)$ are defined in (29). Simultaneously, prerequisites (23d), (30b), and (31) should hold. It deserves to emphasize again that the aforementioned conclusion is correct under a fundamental assumption that the kinematic parameters of the ego vehicle satisfy $2L_W > L_B \cdot \tan\phi_{\max}$.

Remark. According to (29), the embodied box becomes large when $|\kappa_k|$ and/or s_k are large, which is intuitively reasonable. Prerequisites (23d), (30b), and (31) are inherently setting an upper bound for $(t_{k+1} - t_k)$, i.e., the duration between adjacent collocation points.

E. Formulation of Safe Collision-avoidance Constraints

With the embodied box defined for each discretized moment during $[0, T]$, this subsection provides a safe formulation of the collision-avoidance constraints.

Let us define the four vertexes of the embodied box as $A' = (\widehat{x}_A, \widehat{y}_A)$, $B' = (\widehat{x}_B, \widehat{y}_B)$, $C' = (\widehat{x}_C, \widehat{y}_C)$, and $D' = (\widehat{x}_D, \widehat{y}_D)$:

$$\widehat{x}_A(t_k) = x(t_k) + \left(L_F + e_{\text{up}}(t_k)\right) \cdot \cos\theta(t_k) - \left(\frac{L_B}{2} + e_{\text{left}}(t_k)\right) \cdot \sin\theta(t_k),$$

$$\widehat{y}_A(t_k) = y(t_k) + \left(L_F + e_{\text{up}}(t_k)\right) \cdot \sin\theta(t_k) + \left(\frac{L_B}{2} + e_{\text{left}}(t_k)\right) \cdot \cos\theta(t_k),$$

$$\widehat{x}_B(t_k) = x(t_k) + \left(L_F + e_{\text{up}}(t_k)\right) \cdot \cos\theta(t_k) + \left(\frac{L_B}{2} + e_{\text{right}}(t_k)\right) \cdot \sin\theta(t_k),$$

$$\hat{y}_B(t_k) = y(t_k) + \left(L_F + e_{up}(t_k)\right) \cdot \sin\theta(t_k) - \left(\frac{L_B}{2} + e_{right}(t_k)\right) \cdot \cos\theta(t_k),$$

$$\hat{x}_C(t_k) = x(t_k) - \left(L_R + e_{down}(t_k)\right) \cdot \cos\theta(t_k) + \left(\frac{L_B}{2} + e_{right}(t_k)\right) \cdot \sin\theta(t_k),$$

$$\hat{y}_C(t_k) = y(t_k) - \left(L_R + e_{down}(t_k)\right) \cdot \sin\theta(t_k) - \left(\frac{L_B}{2} + e_{right}(t_k)\right) \cdot \cos\theta(t_k),$$

$$\hat{x}_D(t_k) = x(t_k) - \left(L_R + e_{down}(t_k)\right) \cdot \cos\theta(t_k) - \left(\frac{L_B}{2} + e_{left}(t_k)\right) \cdot \sin\theta(t_k),$$

$$\hat{y}_D(t_k) = y(t_k) - \left(L_R + e_{down}(t_k)\right) \cdot \sin\theta(t_k) + \left(\frac{L_B}{2} + e_{left}(t_k)\right) \cdot \cos\theta(t_k).$$
$$(32)$$

We require that the embodied box $A'B'C'D'$ does not overlap with obstacles at N_{fe} moments $\{t_k | k = 0,...,N_{fe}-1\}$. In this way, the ego vehicle's footprint is free from any collision risk at any moment throughout $[0,T]$.

F. Principle of Safety-guaranteed Trajectory Optimization

This subsection presents a standard principle to apply our collision-avoidance model in a numerical optimization-based trajectory planner. The basic steps are summarized into the following pseudo-codes.

Trajectory Optimization with Safe Collision-avoidance Constraints

---

Function $traj \leftarrow$ PlanTrajectory($config, map, obstacle$)

1. $task \leftarrow$ LoadTask($config, map, obstacle$);
2. $path_{coarse} \leftarrow$ PlanCoarsePath($task$);
3. $traj_{coarse} \leftarrow$ AttachVelocity($path_{coarse}, task$);
4. $ig \leftarrow$ FormInitialGuess($traj_{coarse}$);
5. $ocp \leftarrow$ FormOCP($task$);
6. $sol \leftarrow$ SolveOCP(ocp, ig);
7. $traj \leftarrow$ ConvertSolutionToTrajectory(sol);
8.   **return**

In the first step, the trajectory planning task should be loaded, which consists of the environment layout, obstacle distribution information, initial/goal pose of the ego vehicle, and related parameters.

In the second step, a sampling-based path planner is adopted to coarsely find a path.

In the third step, the coarse path is attached with a kinematically feasible and time-optimal velocity profile. This is achieved by solving a one-dimensional OCP via Pontryagin's Maximum Principle. In this way, a coarse trajectory can be obtained. As already mentioned in Section II, only static obstacles are considered, otherwise a yielding/overtaking decision should be made on each conflicting moving obstacle when planning the velocity. We do not consider moving obstacles because they are biased from the theme of this paper.

In the fourth step, an initial guess is formed to feed a numerical optimal control-based trajectory planner. Let us resample along the coarse trajectory with equidistant waypoints that are excessively dense. The values of state and control profiles, together with the time stamp, are recorded in each waypoint. The first waypoint wp_1 refers to the one at $t = 0$, i.e., $wp_1.t = 0$. Thereafter, one checks if the second waypoint wp_2 satisfies a relaxed version of the prerequisites (23d), (30b), and (31) with

$$\kappa_k = wp_1.\kappa,$$
$$s_k = wp_1.v \cdot \left(wp_2.t - wp_1.t\right).$$
$$(33)$$

Herein, the prerequisites (23d), (30b), and (31) are relaxed via a user-specified slack variable $\lambda \in (0,1)$:

$$|\kappa_k| \cdot s_k \le \lambda \cdot \frac{\pi}{2}, \quad (34a)$$

$$|\kappa_k| \cdot L_F \cdot \tan\left(s_k \cdot |\kappa_k|\right) \le \lambda \cdot \left(1 + \frac{L_B}{2} \cdot |\kappa_k|\right), \quad (34b)$$

$$\left(1 + \frac{L_B}{2} \cdot |\kappa_k|\right) \cdot \tan\left(s_k \cdot |\kappa_k|\right) \le \lambda \cdot \left(L_R \cdot |\kappa_k|\right). \quad (34c)$$

The usage of λ relaxes the right sides of inequalities (23d), (30b), and (31), thus making (34) a stricter version of the prerequisites (23d), (30b), and (31). The reason why λ is used would be explained later. Since the waypoints are densely resampled, wp_2 does not violate the prerequisites (34). Thereafter, one continues to check wp_3, wp_4, etc. until a violating waypoint wp_k is found, which means wp_{k-1} is the last valid one $(k > 2)$. Let us discard the intermediate waypoints between wp_1 and wp_{k-1}. After that, we treat wp_{k-1} like wp_1, and then repeat the aforementioned operations until the last resampled waypoint is reached. The waypoints that survive from the discard operation are no longer equidistant, but they do form a trajectory, which is our desired initial guess.

In the fifth step, an OCP is formulated in the form of (7) to present the concerned trajectory planning scheme. Particularly, the ego vehicle's contour is the rectangular $A'B'C'D'$ defined in (32) instead of $ABCD$ in (4). Prerequisites (23d), (30b), and (31) are also imposed. The OCP is discretized into an NLP consisting of $(N_{fe} + 1)$ collocation points $\{t_k | k = 0,...,N_{fe}\}$. The parameter N_{fe} is determined to match that in the initial guess. The collision avoidance-related constraints are only imposed on the first N_{fe} collocation points, i.e., $\{t_k | k = 0,...,N_{fe}-1\}$. It deserves to emphasize that the collocation points are not forced to be equidistant along $[0,T]$ in the formulated NLP. One only requires the collocation points to be ordered, that is,

$$t_0 = 0, \ t_{N_{fe}} = T, \quad (34a)$$

$$t_{k-1} \le t_k, \ k = 1,...,N_{fe}, \quad (34b)$$

Each t_k is a decision variable in the NLP, which brings more flexibility. Inherently, our proposed model is leveraging a relation among v, κ, and ΔT, the product of which should not be too large. For example, if v_k and $|\kappa_k|$ are large, then $\Delta T_k = t_{k+1} - t_k$ has to be small to avoid collision risks between adjacent collocation points. Setting the discretized moments $\{t_k\}$ as decision variables is a highlight of this study, which ensures the satisfaction of the prerequisites in a strict but flexible way.

In the sixth step, the formulated NLP is solved via an NLP solver, which is warm-started by the initial guess derived in step 4. The output of the NLP solution process is a series of waypoints, which represent an optimized trajectory.

Before the end of this section, the reason to introduce λ in (34) is presented. Recall that the collocation point number in the NLP is set to that in the initial guess, thus what criterion we use in step 4 has a direct influence on the value of N_{fe}. If one uses the nominal criterion (23d), (30b), and (31), then each $(t_{k+1} - t_k)$ is close to its maximum allowable value. This renders that the gross waypoint number of the initial guess is

nearly minimized. If one sets N_{fe} to such a low value, then the flexibility in the NLP would be insufficient. More specifically, every change in κ or v during the optimization process that differs from that in the initial guess may request a smaller bound on $(t_{k+1} - t_k)$, which calls for a gross collocation number larger than N_{fe}. Therefore, setting N_{fe} too small easily causes a loss of optimality and even causes an NLP solution failure. The introduction of $\lambda \in (0,1)$ in step 4 sets a harsher subset than the nominal criterion, thus rendering a larger waypoint number and bringing more flexibility to the NLP solution process.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

Simulations and field tests are conducted to validate the efficiency of the proposed modeling method.

A. Simulation Setup

Simulations were performed in MATLAB (R2019a) and executed on an i5-4460T CPU running at 1.90×2 GHz. IPOPT [29] is applied in the MATLAB+AMPL environment [30] as the NLP solver. The linear solver embedded in IPOPT is chosen as ma27 from the Harwell Subroutine Library (HSL) [31]. Hybrid A* search algorithm [4] is adopted as the sampling-based path planner in Section III.F. A triangle-area expression [7] is used to describe collision-avoidance constraints between two convex polygons. A 30m×30m workspace containing static polygonal obstacles is virtually set up. Parametric settings are listed in Table I. They validate that the basic requirement $2L_W > L_B \cdot \tan\phi_{max}$ is met, thus the application of our proposed modeling method is suitable.

TABLE I. PARAMETRIC SETTINGS FOR SIMULATIONS

Parameter	Description	Setting
L_F	Front hang length of the ego vehicle	0.96 m
L_W	Wheelbase of the ego vehicle	2.80 m
L_R	Rear hang length of the ego vehicle	0.929 m
L_B	Width of the ego vehicle	1.942 m
a_{min}, a_{max}	Lower and upper bounds of $a(t)$	-0.75 m/s², 0.75 m/s²
v_{max}	Upper bound of $v(t)$	5.0 m/s
Φ_{max}	Upper bound of $\lvert\phi(t)\rvert$	0.7 rad
Ω_{max}	Upper bound of $\lvert\omega(t)\rvert$	0.5 rad/s
λ	Slack parameter in (34)	0.75

B. Simulation Results and Discussions

The link www.bilibili.com/video/BV1Zt4y157FF presents typical simulation results, wherein the ego vehicle is bordered in green and the embodied box is marked in red. If the embodied box is free from collisions at finite collocation points, then the ego vehicle's footprint is safe at any of the infinite moments throughout $[0,T]$.

Specifically regarding Case 1 in the video, Fig. 8 marks the swept-by region of the ego vehicle footprint from 0 to T, which indicates the planned trajectory is safe with $N_{fe} = 90$. By contrast, if one uses the same setting of N_{fe} and sets the collocation points equidistant (this method is denoted as a naïve NLP method), then the resultant trajectory is not safe between adjacent collocation points, as depicted in Fig. 9. To investigate further, we test the naïve NLP method with N_{fe} incrementally added by 1 at each time. As it turns out, a thoroughly safe trajectory is not found until N_{fe} reaches 137. This comparison
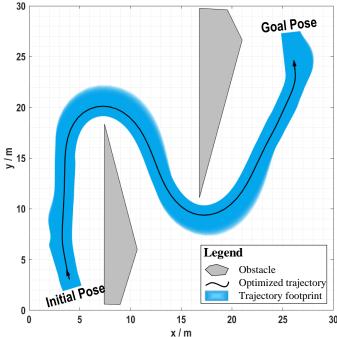

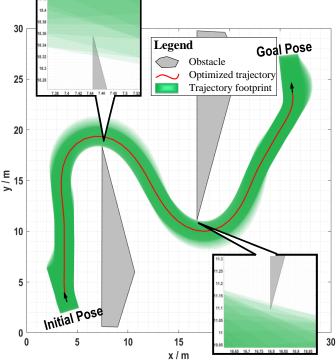Fig. 8. Swept-by region of the planned trajectory derived by the proposed modeling method ($N_{fe} = 90$).

Fig. 9. Swept-by region of the planned trajectory derived by the naïve NLP method ($N_{fe} = 90$).

shows that the proposed method guarantees trajectory safety with an obviously small number of collocation points. Another phenomenon we observe is that the optimized cost function value obtained by the naïve NLP method under $N_{fe} = 137$ is 16.8096 while that obtained by our proposed method is 17.1665, which means the proposed method suffers from a minor loss of optimality. This is intuitively understandable because the usage of an embodied box inevitably wastes some free space.

Let us further test the performance of the proposed modeling method under different settings of λ. Ten sets of initial/goal poses are randomly generated to form ten individual trajectory planning tasks in the same scenario as Fig. 8. Each task is handled by both our proposed method and the naïve method. In adopting the naïve NLP method, we gradually increase the parameter N_{fe} in a trial-and-error mode until trajectory safety is thoroughly achieved. Suppose that the proposed method needs $N_{proposed}$ collocation points to find a safe trajectory while the naïve method needs $N_{naïve}$. A *collocation number growth rate* index can be defined as ($N_{naïve}$ - $N_{proposed}$) / $N_{proposed}$. Fig. 10 presents the average collocation number growth rate under different settings of λ. The results show a reasonable trend and indicate that the proposed modeling method is robust under various settings of λ. Even when λ = 0.5, our proposed method is still slightly better than the naïve NLP method.

C. Field Test Setup, Results, and Discussions

Experiments were conducted on a small-sized autonomous vehicle platform in a 1.75m×1.20m indoor workspace (Fig. 11). Six infrared sensors were deployed for obstacle perception and ego vehicle localization. Concretely, reflective marking points were placed at the vertexes of each polygonal obstacle and the top of the ego vehicle. The infrared light sources would emit infrared beams, which are captured by the infrared sensors after being reflected by the marking points. Through this, the location of each marking point is localized [32]. The indoor infrared solution provided by NOKOV® is adopted in this work, and its infrared sensors are working with a frequency of 60 Hz. The perception and localization information was collected on a desktop PC, where a trajectory planning program was executed. The trajectory planning module ran once to generate an offline trajectory before the ego vehicle began to move. The planned trajectory, together with the localization information would be sent from the PC to the ego vehicle platform via ZigBee. Concerning the onboard tracking control, a PID controller was used for longitudinal tracking while a Pure Pursuit controller was used for lateral tracking. The frequency of either controller was set to 10 Hz. Parameters related to the vehicle geometrics and kinematics are listed in Table II, which satisfy $2L_W > L_B \cdot \tan\phi_{max}$, thus the proposed modeling method is suitable for the field tests.

Typical field test results are available in the video link provided in Section IV.B. Snapshots of the closed-loop tracking performance are shown in Fig. 12, which indicates that the planned open-loop trajectory is safe and easy to track (Fig. 13).

TABLE II. PARAMETRIC SETTINGS FOR FIELD TESTS

Parameter	Description	Setting
L_F	Front hang length of the ego vehicle	0.036 m
L_W	Wheelbase of the ego vehicle	0.143 m
L_R	Rear hang length of the ego vehicle	0.032 m
L_B	Width of the ego vehicle	0.191 m
a_{min}, a_{max}	Lower and upper bounds of $a(t)$	-0.02 m/s², 0.02 m/s²
v_{max}	Upper bound of $v(t)$	0.25 m/s
Φ_{max}	Upper bound of $|\phi(t)|$	0.38 rad
Ω_{max}	Upper bound of $|\omega(t)|$	0.10 rad/s

V. CONCLUSIONS

This paper has proposed a theoretical model for safe collision-avoidance constraints for numerical optimal control-based trajectory planners. The proposed modeling method is analyzed theoretically and validated via simulations/field tests.

The proposed method can be further extended to deal with precision-aware motion planning problems in other research areas than autonomous driving, provided that the kinematic principle is updated accordingly. The proposed modeling method may also work as a fast collision checker that only needs to examine a small number of waypoints along a to-be-checked path/trajectory.

Our future work is to further extend the current study so that reverse driving conditions can be supported, which are just symmetric to the analyses in this paper, though.
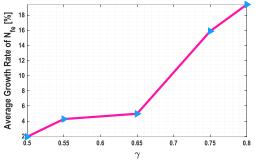

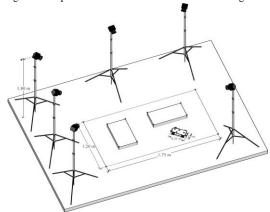Fig. 10. Comparative simulations under different setting of λ.

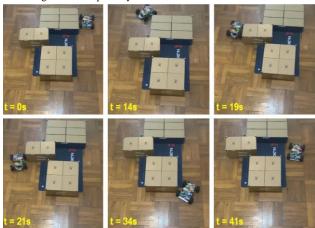Fig. 11. Workspace layout and indoor localization solution.

Fig. 12. Snapshots of closed-loop control performance when the ego vehicle is tracking a planned open-loop trajectory, which is derived by a trajectory planner with safe collision-avoidance constraints.
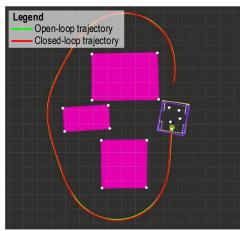
Fig. 13. Open-loop and closed-loop trajectories in the field test.

REFERENCES

[1] B. Li, Y. Ouyang, L. Li, and Y. Zhang, "Autonomous driving on curvy roads without reliance on Frenet frame: A Cartesian-based trajectory planning method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15729–15741, 2022.

[2] B. Li, Y. Ouyang, X. Li, et al., "Mixed-integer and conditional trajectory planning for an autonomous mining truck in loading/dumping scenarios: A global optimization approach," *IEEE Transactions on Intelligent Vehicles*, early access, Oct. 14, 2022, doi: 10.1109/TIV.2022.3214777.

[3] G. Feng, Y. Han, S. E. Li, S. Xu, and D. Dang, "Accurate pseudospectral optimization of nonlinear model predictive control for high-performance motion planning," *IEEE Transactions on Intelligent Vehicles*, early access, Feb. 23, 2022, doi: 10.1109/TIV.2022.3153633.

[4] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[5] C. Pek, and M. Althoff, "Fail-safe motion planning for online verification of autonomous vehicles using convex optimization," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 798–814, 2021.

[6] P. Scheffe, T. M. Henneken, M. Kloock, and B. Alrifaee, "Sequential convex programming methods for real-time optimal trajectory planning in autonomous vehicle racing," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 661–672, 2023.

[7] B. Li, and Z. Shao, "A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles," *Knowledge-Based Systems*, vol. 86, pp. 11–20, 2015.

[8] A. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, no. 1, pp. 497-528, 2009.

[9] S. Vasantharajan, and L. T. Biegler, "Simultaneous strategies for optimization of differential-algebraic systems with enforcement of error criteria," *Computers & Chemical Engineering*, vol. 14, no. 10, pp. 1083–1100, 1990.

[10] K. Kaczmarski, M. Mazzotti, G. Storti, and M. Morbidelli, "Modeling fixed-bed adsorption columns through orthogonal collocations on moving finite elements," *Computers & Chemical Engineering*, vol. 21, no. 6, pp. 641–660, 1997.

[11] B. Li, H. Zhang, W. Zheng, and L. Wang, "Spacecraft close-range trajectory planning via convex optimization and multi-resolution technique," *Acta Astronautica*, vol. 175, pp. 421–437, 2020.

[12] T. Schoels, P. Rutquist, L. Palmieri, et al., "CIAO*: MPC-based safe motion planning in predictable dynamic environments," *IFAC-Papers OnLine*, vol. 53, no. 2, pp. 6555–6562, 2020.

[13] M. Mukadam, J. Dong, X. Yan, et al., "Continuous-time Gaussian process motion planning via probabilistic inference," *The International Journal of Robotics Research*, vol. 37, no. 11, pp. 1319–1340, 2018.

[14] B. Li, and Z. Shao, "Simultaneous dynamic optimization: A trajectory planning method for nonholonomic car-like robots," *Advances in Engineering Software*, vol. 87, pp. 30–42, 2015.

[15] X. Zhou, Z. Wang, X. Wen, et al., "Decentralized spatial-temporal trajectory planning for multicopter swarms," arXiv preprint, 2021, https://arxiv.org/abs/2106.12481v2.

[16] B. Li, T. Acarman, Y. Zhang, et al., "Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11970–11981, 2022.

[17] Z. Zhu, E. Schmerling, and M. Pavone, "A convex optimization approach to smooth trajectories for motion planning with car-like robots," In *Proc. 2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 835–842, 2015.

[18] B. Li, Y. Zhang, T. Acarman, et al., "Trajectory planning for a tractor with multiple trailers in extremely narrow environments: A unified approach," In *Proc. 2019 International Conference on Robotics and Automation (ICRA)*, pp. 8557–8562, 2019.

[19] G. Van Den Bergen, Proximity queries and penetration depth computation on 3D game objects. In *Proc. Game Developers Conference*, vol. 170, pp. 1–17, 2001.

[20] C. Pek, V. Rusinov, S. Manzinger, et al., "CommonRoad drivability checker: simplifying the development and validation of motion planning algorithms," In *Proc. 2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1013–1020, 2020.

[21] J. Schulman, Y. Duan, J. Ho, et al., "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[22] A. Scheuer, and T. Fraichard, "Continuous-curvature path planning for car-like vehicles," In *Proc. 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS)*, pp. 997–1003, 1997.

[23] N. Ghita, and M. Kloetzer, "Trajectory planning for a car-like robot by environment abstraction," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 609–619, 2012.

[24] B. Li, and Z. Shao, "Precise trajectory optimization for articulated wheeled vehicles in cluttered environments," *Advances in Engineering Software*, vol. 92, pp. 40–47, 2016.

[25] Z. Zhang, S. Lu, L. Xie, et al., "A guaranteed collision-free trajectory planning method for autonomous parking," *IET Intelligent Transport Systems*, vol. 15, no. 2, pp. 331–343, 2021.

[26] D. S. Yershov, and E. Frazzoli, "Asymptotically optimal feedback planning using a numerical Hamilton-Jacobi-Bellman solver and an adaptive mesh refinement," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 565–584, 2016.

[27] W. Chen, Y. Ren, G. Zhang, and L. T. Biegler, "A simultaneous approach for singular optimal control based on partial moving grid," *AIChE Journal*, vol. 65, no. 6, pp. 1–10, 2019.

[28] B. Li, T. Acarman, Y. Zhang, et al., "Tractor-trailer vehicle trajectory planning in narrow environments with a progressively constrained optimal control approach," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 3, pp. 414–425, 2020.

[29] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[30] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. South San Francisco: The Scientific Press, 2003.

[31] HSL, "A collection of Fortran codes for large scale scientific computation," http://www.hsl.rl.ac.uk/.

[32] B. Li, et al., "Sharing traffic priorities via cyber physical social intelligence: A lane-free autonomous intersection management method in Metaverse," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, early access, Dec. 13, 2022, doi: 10.1109/TSMC.2022.3225250.

Bai Li (SM'13–M'19) received his B.S. degree in 2013 from Beihang University, China, and the Ph.D. degree from Zhejiang University, China. From Nov. 2016 to June 2017, he visited the Department of Civil and Environmental Engineering, University of Michigan (Ann Arbor), USA, as a joint training Ph.D. student. He is currently an associate professor in the College of Mechanical and Vehicle Engineering, Hunan University, China. Prior to joining Hunan University, he worked in JDX R&D Center of Automated Driving, JD Inc., China from 2018 to 2020 as an algorithm engineer. He has been the first author of nearly 70 journal/conference papers and 2 books in robotics. He was a recipient of the International Federation of Automatic Control (IFAC) 2014–2016 Best Journal Paper Prize. Since 2022, he serves as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT VEHICLES (TIV). He won the 2022 Best Associate Editor Award of TIV. His research interest is optimization-based trajectory planning for automated driving.

Youmin Zhang (M'99–SM'07–F'23) received the B.S., M.S., and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 1983, 1986, and 1995, respectively. He is currently a Professor with the Department of Mechanical, Industrial & Aerospace Engineering, as well as the Concordia Institute of Aerospace Design & Innovation, Concordia University, Montreal, Quebec, Canada. His main research interests include fault detection and diagnosis (FDD), fault-tolerant control (FTC), fault-tolerant cooperative control (FTCC) of single and multiple unmanned vehicles, smart grids, and applications of unmanned systems to forest fires, power lines, environment, natural resources and disasters monitoring, detection, and protection by combining with remote sensing techniques. He has authored 8 books, over 550 journal and conference papers, and book chapters. Dr. Zhang is a Fellow of CSME, a Senior Member of AIAA and IEEE, President of the International Society of Intelligent Unmanned Systems, and a member of the Technical Committee for several scientific societies. He has been an Editorial Board Member, Editor-in-Chief, Editor-at-Large, Editor, or Associate Editor of several international journals such as IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and Journal of Intelligent & Robotic Systems.

Tantan Zhang received the B.S. degree in 2012 from Hunan University, Changsha, China, the double M.S. degrees in 2015 from Politecnico di Torino, Turin, Italy, and Tongji University, Shanghai, China, and a Ph.D. degree in 2020 from Politecnico di Torino, Turin, Italy. He is currently an assistant professor in the College of Mechanical and Vehicle Engineering, Hunan University, China. His research interests include motion planning of automated vehicles.

Tankut Acarman (M'06) received a Ph.D. degree in electrical and computer engineering from Ohio State University, Columbus, OH, USA, in 2002. He is a Professor and the Head of the Department of Computer Engineering, Galatasaray University, Istanbul, Turkey. He is a co-author of the book entitled Autonomous Ground Vehicles. He serves as a Senior Editor for IEEE TRANSACTIONS ON INTELLIGENT VEHICLES. His research interests include aspects of intelligent vehicle technologies, driver assistance systems, and performance evaluation of inter-vehicle communication.

Yakun Ouyang (SM'20) received his B.S. degree in 2020 from the School of Information Engineering, Nanchang University, Nanchang, China. He is currently pursuing a Master's degree in the College of Mechanical and Vehicle Engineering, Hunan University, China. He was the first-prize recipient of the 2019 National University Students Intelligent Car Race. His research interests include decision making, trajectory planning, control, and software engineering of an autonomous vehicle system.

Li Li (F'17) is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China, working in the fields of artificial intelligence, complex systems, intelligent control and sensing, intelligent transportation systems, and intelligent vehicles. He has published over 100 SCI-indexed international journal articles and over 70 international conference papers as a first/corresponding author. He is a member of the Editorial Advisory Board for the Transportation Research Part C: Emerging Technologies, and a member of the Editorial Board for the Transport Reviews and ACTA Automatica. He serves as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and IEEE TRANSACTIONS ON INTELLIGENT VEHICLES.

Hairong Dong (M'12–SM'12) received the Ph.D. degree from Peking University in 2002. She was a Visiting Scholar with the University of Southampton in 2006 and The University of Hong Kong in 2008. She was also a Visiting Professor with the KTH Royal Institute of Technology in 2011. She is currently a Professor with the State Key Laboratory of Rail Traffic

Control and Safety, Beijing Jiaotong University, China. Her research interests include intelligent transportation systems, automatic train operation, intelligent dispatching, and complex network applications. She is a Fellow of the Chinese Automation Congress and the Co-Chair of the Technical Committee on Railroad Systems and Applications of the IEEE Intelligent Transportation Systems Society. She serves as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, IEEE INTELLIGENT TRANSPORTATION SYSTEMS MAGAZINE, and Journal of Intelligent and Robotic Systems.

Dongpu Cao (M'08) received the Ph.D. degree from Concordia University, Canada, in 2008. He is a Professor at Tsinghua University, China. His current research interests include driver cognition, automated driving and cognitive autonomous driving. He has contributed more than 200 papers and 3 books. He received the SAE Arch T. Colwell Merit Award in 2012, IEEE VTS 2020 Best Vehicular Electronics Paper Award and 6 Best Paper Awards from international conferences. Prof. Cao has served as Deputy Editor-in-Chief for IET Intelligent Transport Systems Journal, and an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and ASME Journal of Dynamic Systems, Measurement, and Control. Prof. Cao is an IEEE VTS Distinguished Lecturer.