Optimal Control for Kinematic Bicycle Model with Continuous-time Safety Guarantees: A Sequential Second-order Cone Programming Approach

Victor Freire and Xiangru Xu

***Abstract*—The optimal control problem for the kinematic bicycle model is considered where the trajectories are required to satisfy the safety constraints in the continuous-time sense. Based on the differential flatness property of the model, necessary and sufficient conditions in the flat space are provided to guarantee safety in the state space. The optimal control problem is relaxed into three second-order cone programs (SOCPs) solved sequentially, which find the safe path, the trajectory duration, and the speed profile, respectively. Solutions of the three SOCPs provide a sub-optimal but feasible trajectory in the original optimal control problem. Simulation examples and comparisons with state-of-the-art optimal control solvers are presented to demonstrate the effectiveness of the proposed approach.**

***Index Terms*—Optimization and Optimal Control, Motion and Path Planning, Constrained Motion Planning, Safety-Critical Control.**

I. Introduction

THE vehicle motion planning problem is well-studied. However, fast algorithms tend to lack formal safety guarantees while robust and safe planning methods are usually slow, making them unfit for real-time implementation. The search for fast and safe planning algorithms is key to achieving provably safe driving autonomy [1], [2].

The literature addressing the vehicle motion planning problem is vast. A common approach is the spatio-temporal division of the problem. On the one hand, most path-finding (spatial) algorithms parse the configuration space in search of minimum length or curvature sequences [3]–[5]. For example, in [4], the authors used RRT and B-splines to explore the space and generate kinodynamically feasible paths; however, their approach is computationally demanding because it needs to verify feasibility at each RRT sampling step. In [5], a multi-layer planning framework was proposed where sampling techniques modify a global path for obstacle avoidance; however, they rely on nonconvex minimization of the path's curvature to enforce kinodynamic feasibility. On the other hand, speed profile optimization (temporal) algorithms focus on minimum time and maximum rider comfort while navigating a given path [6]–[8]. For example, [6] showed that a minimum-time objective function can be reformulated in terms of the path parameter, and the problem was generalized for certain classes of systems in [7]. In [8], the authors used spatio-temporal separation to alternatively optimize the path and the speed profile; both problems are convex but the stopping criteria is ambiguous.

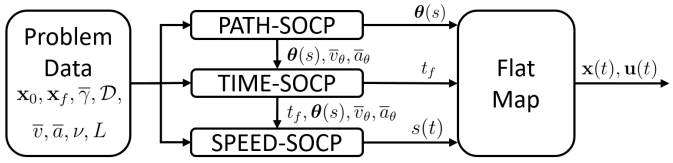

Fig. 1. Architecture of the sequential SOCP approach. (PATH-SOCP) finds a safe path $\boldsymbol{\theta}(s)$, (TIME-SOCP) finds an appropriate trajectory duration t_f, and (SPEED-SOCP) finds a safe speed profile $s(t)$. These quantities parameterize the flat outputs which can be converted into a state-space trajectory $\mathbf{x}(t)$ and $\mathbf{u}(t)$ by the differential flatness property of the kinematic bicycle model, where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ is a guaranteed feasible point of (OPT) by Theorem 1.

The kinematic bicycle model is widely used and captures the nonholonomic constraint present in vehicle dynamics. It has been used in optimal control problems for motion planning. For example, [9] studied the consistency of using the kinematic bicycle model for motion planning by comparing its results with a higher-fidelity model; [10] formulated an MPC problem based on the kinematic bicycle model, but it is nonconvex and provides no safety guarantees in the continuous-time sense; [11] demonstrated the high performance of stochastic MPC in a miniature racing environment, but their safety guarantees are given in the probabilistic sense.

In this work, we formulate an optimal control problem for vehicle motion planning using the kinematic bicycle model. The state and input safety constraints include maximum steering angle, position constraints, maximum velocity, and bounded acceleration. We use differential flatness to formulate conditions that guarantee continuous-time constraint satisfaction in the state space. We then use spatio-temporal separation and the convexity of B-splines to relax the original optimal control problem into three sequential SOCPs yielding a path, a trajectory duration, and a speed profile. We show that the SOCP solutions constitute a suboptimal but feasible solution to the original optimal control problem. The convex SOCPs are solvable by off-the-shelf solvers in real-time. The remainder of the paper is organized as follows: Sec. II describes the preliminaries and the considered problem; Sec. III provides necessary and sufficient conditions in the flat space that guar-

Manuscript received April 18, 2022; revised July 31, 2022; accepted August 23, 2022. (Corresponding author: Xiangru Xu.)

This letter was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments.

Victor Freire and Xiangru Xu are with the Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA (e-mail: freiremelgiz@wisc.edu; xiangru.xu@wisc.edu).

Digital Object Identifier (DOI): see top of this page.

antee safety in state space; Sec. IV presents convex relaxations and formulates the three SOCPs; Sec. V provides simulation examples and Sec. VI concludes the paper.

II. Preliminaries & Problem Statement

A. Kinematic Bicycle Model

The kinematic bicycle model is a commonly used, simple model which captures the nonholonomic constraint present in most wheeled vehicles, and it can be expressed as [12]:

$$\dot{\mathbf{x}}(t) = f\big(\mathbf{x}(t)\big) + g\big(\mathbf{x}(t)\big)\mathbf{u}(t), \tag{1}$$

where $f(\mathbf{x}) = (v\cos\psi, v\sin\psi, 0, 0)^T$, $g(\mathbf{x}) = (\mathbf{0}_{2\times 2}, I_2)^T$. State and input vectors are, respectively, $\mathbf{x} = (x, y, v, \psi)^T$ and $\mathbf{u} = (\dot{v}, \dot{\psi})^T$, where (x, y) is the position of the rear wheel, v is the speed and ψ is the heading angle (see Figure 2). The steering angle γ is also relevant and is given by $\gamma = \arctan(L\dot{\psi}/v)$, where $L > 0$ is the wheelbase length.

The kinematic bicycle model (1) is a special case of the classical *n-cart* system and is known to be *differentially flat* [13]. By choosing flat outputs as $\mathbf{y} = (x, y)^T$, the state and input can be expressed as functions of $\mathbf{y}$ and its derivatives:

$$\mathbf{x} = \Phi(\mathbf{y}, \dot{\mathbf{y}}), \quad \mathbf{u} = \Psi(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}), \tag{2}$$

where the flat maps Φ and Ψ are given as in [13]: $v = \|\dot{\mathbf{y}}\|_2$, $\psi = \arctan(\dot{y}/\dot{x})$, $\dot{v} = \dot{\mathbf{y}}^T\ddot{\mathbf{y}}/v$ and $\dot{\psi} = (\ddot{y}\dot{x} - \ddot{x}\dot{y})/v^2$. Generating a trajectory for differentially flat systems reduces to finding a sufficiently smooth flat output trajectory [14]. For system (1), $\mathbf{y}(t)$ needs to be at least twice-differentiable.

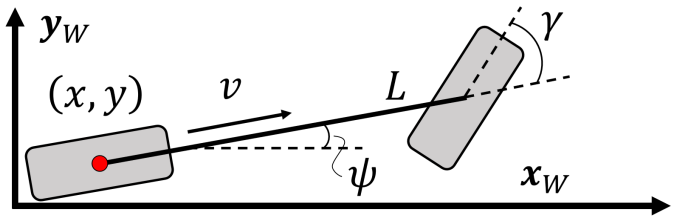

Fig. 2. Kinematic bicycle model and world (inertial) coordinate frame.

B. B-Spline Curves

B-splines are common in trajectory generation [4], [15], [16]. A d-th degree *B-spline basis* $\lambda_{i,d}(t)$ with $d \in \mathbb{Z}_{>0}$ is defined over a given *knot vector* $\boldsymbol{\tau} = (\tau_0, \ldots, \tau_\eta)^T$ satisfying $\tau_i \le \tau_{i+1}$ for $i = 0, \ldots, \eta - 1$ and is computed recursively by the Cox-de Boor recursion formula [17]. Additionally, we consider the *clamped, uniform B-spline basis*, which is defined over knot vectors satisfying:

$$\text{(clamped)} \quad \tau_0 = \ldots = \tau_d, \quad \tau_{\eta-d} = \ldots = \tau_\eta, \tag{3a}$$

$$\text{(uniform)} \quad \tau_{d+1} - \tau_d = \ldots = \tau_{N+1} - \tau_N, \tag{3b}$$

where $N = \eta - d - 1$. A d-th degree *B-spline curve* $\mathbf{s}(t)$ is a m-dimensional parametric curve built by linearly combining *control points* $\mathbf{p}_i \in \mathbb{R}^m (i = 0, \ldots, N)$ and B-spline bases of the same degree. Given that $\mathbf{s}(t) = P\boldsymbol{\Lambda}_d(t)$, we compute a B-spline curve's r-th order derivative with:

$$\frac{\mathrm{d}^r\mathbf{s}}{\mathrm{d}t^r}(t) = \sum_{i=0}^{N} \mathbf{p}_i \mathbf{b}_{r,i+1}^T \boldsymbol{\Lambda}_{d-r}(t) = PB_r\boldsymbol{\Lambda}_{d-r}(t), \tag{4}$$

where the control points are grouped into a matrix $P = (\mathbf{p}_0, \ldots, \mathbf{p}_N) \in \mathbb{R}^{m\times(N+1)}$, the basis functions are grouped into a vector $\boldsymbol{\Lambda}_{d-r}(t) = \big(\lambda_{0,d-r}(t), \ldots, \lambda_{N+r,d-r}(t)\big)^T \in \mathbb{R}^{N+r+1}$, and $\mathbf{b}_{r,j}^T$ is the j-th row of a time-invariant matrix $B_r \in \mathbb{R}^{(N+1)\times(N+r+1)}$ constructed as $B_r = M_{d,d-r}C_r$, where matrices $M_{d,d-r} \in \mathbb{R}^{(N+1)\times(N-r+1)}$ and $C_r \in \mathbb{R}^{(N-r+1)\times(N+r+1)}$ are defined in [16], [18].

Definition 1. [16] *The columns of $P^{(r)} \triangleq PB_r$ are called the r-th order virtual control points (VCPs) of $\mathbf{s}^{(r)}(t)$ and denoted as $\mathbf{p}_i^{(r)}$ where $i = 0, 1, \ldots, N + r$, i.e.,*

$$P^{(r)} = PB_r = \begin{bmatrix} \mathbf{p}_0^{(r)} & \ldots & \mathbf{p}_{N+r}^{(r)} \end{bmatrix}. \tag{5}$$

Note that we use parenthesized exponent $\mathbf{p}_j^{(r)}$ only to indicate the order of VCPs. B-splines have some nice properties such as continuity, convexity, and local support. The following result ensures continuous-time set inclusion by using such properties.

Proposition 1. [16] *Given a convex set $\mathcal{S}$ and the r-th derivative of a clamped B-spline curve $\mathbf{s}^{(r)}(t)$ defined as in* (4)*, if $\mathbf{p}_j^{(r)} \in \mathcal{S}$, with $j = r, \ldots, N$ holds, then $\mathbf{s}^{(r)}(t) \in \mathcal{S},\ t \in [\tau_0, \tau_\eta)$. Furthermore, if $\mathbf{p}_j^{(r)} \in S$, with $j = i - d + r, \ldots, i$ and $i \in \{d, \ldots, N\}$ holds, then $\mathbf{s}^{(r)}(t) \in \mathcal{S},\ t \in [\tau_i, \tau_{i+1})$.*

C. Second-order Cone Constraints

Second-order cone programs (SOCPs) are convex optimization problems of the following form [19]:

$$\text{min.} \quad \mathbf{f}^T\mathbf{x}, \quad \text{s. t.} \quad \|A_i\mathbf{x} + \mathbf{b}_i\|_2 \le \mathbf{c}_i^T\mathbf{x} + d_i, \quad i = 1, \ldots, m,$$

where $\mathbf{x} \in \mathbb{R}^n$ is the decision variable, $\mathbf{f} \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{(n_i-1)\times n}$, $\mathbf{b}_i \in \mathbb{R}^{n_i-1}$, $\mathbf{c}_i \in \mathbb{R}^n$ and $d_i \in \mathbb{R}$. The constraint above is the *second-order cone* (SOC) constraint. SOCPs are a generalization of more specialized types of convex optimization problems such as linear programs (LPs) and quadratic programs (QPs) [19]. SOCPs can be solved in polynomial time by interior-point methods, and specialized SOCP solvers also exist [20].

D. Problem Statement

We formulate the motion planning problem as the following constrained optimal control problem with variable horizon:

$$\min_{\mathbf{x}(\cdot),\mathbf{u}(\cdot),t_f} \quad \nu t_f + \int_0^{t_f} L(\mathbf{x}(t), \mathbf{u}(t))\,\mathrm{d}t \tag{OPT}$$

$$\text{s. t.} \quad \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) + g(\mathbf{x}(t))\mathbf{u}(t), \tag{6a}$$

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f, \tag{6b}$$

$$0 \le v(t) \le \overline{v}, \quad \forall t \in [0, t_f], \tag{6c}$$

$$|\dot{v}(t)| \le \overline{a}, \quad \forall t \in [0, t_f], \tag{6d}$$

$$|\gamma(t)| \le \overline{\gamma}, \quad \forall t \in [0, t_f], \tag{6e}$$

$$\mathbf{r}(t) \in \mathcal{D}, \quad \forall t \in [0, t_f], \tag{6f}$$

where f and g are defined in the kinematic bicycle model (1), $\mathbf{x}_0$ and $\mathbf{x}_f$ are given initial and final states, respectively, $\overline{v} > 0$ is the speed limit, $\mathcal{D} \subseteq \mathbb{R}^2$ is the obstacle-free region, $\mathbf{r} \triangleq (x, y)^T$ is the position vector, $\overline{a} > 0$ is the acceleration bound and $0 < \overline{\gamma} < \pi/2$ is the steering angle limit. The Lagrange cost functional $L : \mathbb{R}^4 \times \mathbb{R}^2 \to \mathbb{R}$ is chosen to promote smoothness properties for the trajectory, and the parameter $\nu > 0$ encodes the tradeoff between time-optimality and smoothness. The optimal control problem (OPT) is generally non-convex and computationally demanding to solve in real-time. Furthermore, constraints (6c)-(6f) are difficult to satisfy strictly in the continuous-time sense because optimal control solvers rely on discretization and these constraints are often enforced only at discrete time instances [21], [22].

In this paper, we solve (OPT) by proposing a sequential SOCP approach, which guarantees that the constraints (6c)-(6f) are rigorously satisfied in the continuous-time. While we won't compromise on safety (feasibility), we will trade optimality for computational efficiency. Our approach leverages the differential flatness property of the bicycle model and parameterizes flat outputs using a pair of convoluted B-spline curves. We consider a separation between space ($\mathbb{R}^2$) and time to first find a *path* with desirable properties; then, we use these properties to find a *speed profile* for navigating it. Convoluting the *path* with its *speed profile* results in the flat output trajectory required to recover the state-space trajectory.

III. Safety Constraints Satisfaction in Flat Space

In this section, we provide necessary and sufficient conditions on the flat output trajectory $\mathbf{y}(t)$ that can guarantee continuous-time safety in the state-space.

Consider a *path* $\boldsymbol{\theta}(s) \triangleq \big(x(s), y(s)\big)^T \in C^2 : [0,1] \to \mathbb{R}^2$ and a *speed profile* $s(t) \in C^2 : [0, t_f] \to [0, 1]$, where C^2 is the set of functions whose derivatives, up to 2nd order, exist and are continuous. The path and speed profile completely define the flat output and its derivatives [23]:

$$\mathbf{y}(t) = \boldsymbol{\theta}\big(s(t)\big), \tag{7a}$$
$$\dot{\mathbf{y}}(t) = \dot{s}(t)\boldsymbol{\theta}'\big(s(t)\big), \tag{7b}$$
$$\ddot{\mathbf{y}}(t) = \ddot{s}(t)\boldsymbol{\theta}'\big(s(t)\big) + \dot{s}^2(t)\boldsymbol{\theta}''\big(s(t)\big), \tag{7c}$$

where $\boldsymbol{\theta}'\big(s(t)\big)$ denotes differentiation of $\boldsymbol{\theta}$ with respect to s and taking values at $s(t)$, and similarly for $\boldsymbol{\theta}''\big(s(t)\big)$.

Remark 1. *The parameterization shown in* (7a) *provides a number of benefits. First, the map* (2) *has singularities if* $\|\dot{\mathbf{y}}(t)\|_2 = 0$ *for some* t *(i.e.,* $\dot{x}(t) = \dot{y}(t) = 0$*); however, the convoluted parameterization shown in* (7a) *allows one to avoid the singularity even in zero-speed situations [24]. Second, we can now consider the safety of the path* $\boldsymbol{\theta}$*, such as obstacle avoidance and steering angle constraints, independently of the speed profile* s *chosen later; that is, the parameterization makes spatial constraints independent from temporal constraints.*

In the following, we will overload the notation of the flat map (2) as $\Phi(t) \triangleq \Phi\big(\mathbf{y}(t), \dot{\mathbf{y}}(t)\big) = \mathbf{x}(t)$, $\Psi(t) \triangleq \Psi\big(\mathbf{y}(t), \dot{\mathbf{y}}(t), \ddot{\mathbf{y}}(t)\big) = \mathbf{u}(t)$, where $\mathbf{y}(t)$ and its derivatives are parameterized in (7a)-(7c).

A. Path Safety

Define the *steering angle safety set* and *drivable safety set*, respectively, as:

$$\mathcal{S}_\gamma \triangleq \{(\mathbf{x}, \mathbf{u}) \in \mathbb{R}^4 \times \mathbb{R}^2 : |\gamma| \le \overline{\gamma} < \pi/2\}, \tag{8}$$
$$\mathcal{S}_D \triangleq \{\mathbf{x} \in \mathbb{R}^4 : \mathbf{r} = (x, y)^T \in \mathcal{D}\}, \tag{9}$$

with γ the steering angle and $\mathcal{D} \subseteq \mathbb{R}^2$ the obstacle-free space.

Lemma 1. *The state-space trajectory* $\big(\Phi(t), \Psi(t)\big) \in \mathcal{S}_\gamma$ *for all* $t \in [0, t_f]$ *if and only if*

$$\|\boldsymbol{\theta}'(s) \times \boldsymbol{\theta}''(s)\|_2 / \|\boldsymbol{\theta}'(s)\|_2^3 \le \tan\overline{\gamma}/L, \quad \forall s \in [0,1]. \tag{10}$$

Proof. It can be shown from the expression of γ given in Section II-A and the considered parameterization (7a) that $\gamma(t) = \arctan\big(L(y''(s)x'(s) - x''(s)y'(s))/\|\boldsymbol{\theta}'(s)\|_2^3\big)$. Furthermore, over the range $\gamma \in (-\pi/2, \pi/2)$, $\tan|\gamma| = |\tan\gamma|$. Therefore, $\tan|\gamma|/L = \|\boldsymbol{\theta}'(s) \times \boldsymbol{\theta}''(s)\|_2/\|\boldsymbol{\theta}'(s)\|_2^3$. Observing that $\tan(|\gamma|) \le \tan(\overline{\gamma}), \gamma \in (-\pi/2, \pi/2) \iff |\gamma| \le \overline{\gamma} < \pi/2$, the conclusion follows immediately. □

Lemma 2. *The state-space trajectory* $\Phi(t) \in \mathcal{S}_D$ *for all* $t \in [0, t_f]$ *if and only if*

$$\boldsymbol{\theta}(s) \in \mathcal{D}, \quad \forall s \in [0,1]. \tag{11}$$

Proof. Recall that the speed profile $s \in C^2 : [0, t_f] \to [0, 1]$. Thus, the condition $\mathbf{y}(t) = \boldsymbol{\theta}(s(t)) \in \mathcal{D}$ must hold for all $t \in [0, t_f]$. The conclusion follows by the definition of $\mathcal{S}_D$. □

B. Speed Profile Safety

Define the *forward speed safety set* and *linear acceleration safety set*, respectively, as:

$$\mathcal{S}_v \triangleq \{\mathbf{x} \in \mathbb{R}^4 : 0 \le v \le \overline{v}\}, \tag{12}$$
$$\mathcal{S}_{\dot{v}} \triangleq \{\mathbf{u} \in \mathbb{R}^2 : |\dot{v}| \le \overline{a}\}, \tag{13}$$

with $\overline{v}$ and $\overline{a}$ the speed and acceleration bounds, respectively.

Lemma 3. *Let* $\boldsymbol{\theta}$ *be a path. The state-space trajectory* $\Phi(t) \in \mathcal{S}_v$ *for all* $t \in [0, t_f]$ *if and only if*

$$\dot{s}(t) \ge 0, \quad \dot{s}(t)\|\boldsymbol{\theta}(s(t))\|_2 \le \overline{v}, \quad \forall t \in [0, t_f]. \tag{14}$$

Proof. The conclusion follows from the flat map (2) describing the state v and the parameterization of $\dot{\mathbf{y}}(t)$ given in (7b). □

Lemma 4. *Let* $\boldsymbol{\theta}$ *be a path. The input trajectory* $\Psi(t) \in \mathcal{S}_{\dot{v}}$ *for all* $t \in [0, t_f]$ *if and only if*

$$|a_t(t) + a_n(t)| \le \overline{a}, \quad \forall t \in [0, t_f], \tag{15}$$

where $a_t \triangleq \ddot{s}\|\boldsymbol{\theta}'(s)\|_2$ *and* $a_n \triangleq \dot{s}^2\big(\boldsymbol{\theta}'(s) \cdot \boldsymbol{\theta}''(s)\big)/\|\boldsymbol{\theta}'(s)\|_2$.

Proof. Differentiating $v(t) = \dot{s}(t)\|\boldsymbol{\theta}'\big(s(t)\big)\|_2$ with respect to time we have $\dot{v}(t) = a_t(t) + a_n(t)$. The conclusion follows immediately by the definition of $\mathcal{S}_{\dot{v}}$. □

C. Flattened Optimal Control Problem

Consider now the following functional optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}(\cdot),s(\cdot),t_f} \quad & \nu t_f + \int_0^{t_f} L\big(\Phi(t),\Psi(t)\big)\,\mathrm{d}t, \qquad \text{(FLAT-OPT)} \\ \text{s. t.} \quad & \Phi(0)=\mathbf{x}_0, \quad \Phi(t_f)=\mathbf{x}_f, \\ & \text{(10), (11), (14) and (15) hold,} \end{aligned}$$

where $\boldsymbol{\theta} \in C^2 : [0,1] \to \mathbb{R}^2$ and $s \in C^2 : [0,t_f] \to [0,1]$.

Proposition 2. *If the duration t_f,* path $\boldsymbol{\theta}$ *and* speed profile s *is a solution of* (FLAT-OPT)*, then the corresponding trajectory given by $\mathbf{x}(t)=\Phi(t)$ and $\mathbf{u}(t)=\Psi(t)$ is a solution of* (OPT)*.*

Proof. The constraints on the initial and final states hold: $\Phi(0)=\mathbf{x}(0)=\mathbf{x}_0$ and $\Phi(t_f)=\mathbf{x}(t_f)=\mathbf{x}_f$. By the definition of each safety set $\mathcal{S}_\gamma, \mathcal{S}_D, \mathcal{S}_v$ and $\mathcal{S}_\dot{v}$ and by Lemmas 1, 2, 3 and 4, it follows that the state-space trajectory $\mathbf{x}(t)$ and $\mathbf{u}(t)$ satisfies all safety constraints in (OPT). The differential constraint in (OPT) is automatically satisfied by virtue of the differential flatness property [14] and C^2 smoothness of the *path* $\boldsymbol{\theta}$ and *speed profile* s. Finally, notice that the objective functionals are identical in both problems. □

Proposition 2 is a particular case of the observation in [25] that for differentially flat systems, optimal control problems can be cast as functional optimization problems without differential constraints by virtue of the flatness property. Intuitively, the state differential constraint is translated into a smoothness constraint in the flat output. However, (FLAT-OPT) is still intractable because the problem is nonconvex and we are minimizing over functions instead of vectors. In the next section, we will let the path $\boldsymbol{\theta}$ and the speed profile s be B-spline curves and optimize over their control points. We also use their convexity properties to reformulate the constraints into convex conditions with respect to their control points.

IV. Convexification of Problem (FLAT-OPT)

In this section, we describe a convexification approach for (FLAT-OPT) by splitting the problem into three sequential SOCP: the first SOCP finds a safe path $\boldsymbol{\theta}(s)$, the second SOCP finds a duration t_f for the trajectory, and the third SOCP computes a safe speed profile $s(t)$ (see Figure 1). The solution of these three convex programs together provides a feasible and possibly sub-optimal solution to (FLAT-OPT) with rigorous continuous-time constraint satisfaction guarantees.

Definition 2 (B-spline path)**.** *A* B-spline path *is a C^2, 2-dimensional, d_θ-degree B-spline curve defined as in* (4) *over a clamped, uniform knot vector $\boldsymbol{\zeta}$ segmenting the interval $[0,1]$ and control points $\boldsymbol{\Theta}_j \in \mathbb{R}^2,\ j=0,\dots,N_\theta$.*

Definition 3 (B-spline speed profile)**.** *A* B-spline speed profile *is a C^2, 1-dimensional, d_s-degree B-spline curve defined as in* (4) *over a clamped, uniform knot vector $\boldsymbol{\tau}$ segmenting the interval $[0,t_f]$ and control points $p_j \in \mathbb{R},\ j=0,\dots,N_s$.*

A. B-spline Path Optimization

The following propositions provide convex relaxations to the conditions of Lemma 1 and Lemma 2.

Proposition 3. *Let $\boldsymbol{\theta}(s)$ be a B-spline path. If there exist positive constant $\alpha > 0$, column unit vector $\hat{\mathbf{r}} \in \mathbb{R}^2$, and variables $\underline{v}_\theta, \overline{a}_\theta, \beta \in \mathbb{R}$ such that the B-spline path $\boldsymbol{\theta}(s)$ satisfies the following conditions:*

$$\hat{\mathbf{r}}^T \boldsymbol{\Theta}_i^{(1)} \geq \underline{v}_\theta, \quad j=1,\dots,N_\theta, \tag{16a}$$

$$\big\|\boldsymbol{\Theta}_i^{(2)}\big\|_2 \leq \overline{a}_\theta, \quad j=2,\dots,N_\theta, \tag{16b}$$

$$\Big\| \begin{bmatrix} 2\alpha & \frac{4\tan\overline{\gamma}}{L}\beta - 1 \end{bmatrix}^T \Big\|_2 \leq 4\beta\tan\overline{\gamma}/L + 1, \tag{16c}$$

$$\overline{a}_\theta \leq \alpha\underline{v}_\theta - \beta, \quad \beta, \underline{v}_\theta \geq 0, \tag{16d}$$

then the state-space trajectory $\big(\mathbf{x}(t),\mathbf{u}(t)\big) \in \mathcal{S}_\gamma,\ \forall t \in [0,t_f)$ where $\mathcal{S}_\gamma$ is the steering angle safety set *defined in* (8)*.*

Proof. By Proposition 1, conditions (16a)-(16b) imply that $\underline{v}_\theta \leq \|\boldsymbol{\theta}'(s)\|_2$ and $\|\boldsymbol{\theta}''(s)\|_2 \leq \overline{a}_\theta$ for all $s \in [0,1)$. Expand (16c) to observe that $0 \geq \alpha^2 - 4\beta\tan\overline{\gamma}/L\beta \triangleq \Delta_v$. Notice that Δ_v is the discriminant of the quadratic polynomial: $p(\underline{v}_\theta) \triangleq \underline{v}_\theta^2\tan\overline{\gamma}/L - \alpha\underline{v}_\theta + \beta$. It follows from $\Delta_v \leq 0$ that the roots of $p(\underline{v}_\theta)$ are either repeated and real, or complex conjugates. Therefore, the polynomial $p(\underline{v}_\theta)$ does not change sign. Since $p(0) \geq 0$ (because $\beta \geq 0$), we must have that $p(\underline{v}_\theta) \geq 0$. In particular, we can now observe that $\overline{a}_\theta \leq \alpha\underline{v}_\theta - \beta \leq \underline{v}_\theta^2\tan\overline{\gamma}/L \implies \|\boldsymbol{\theta}''(s)\|_2 \leq \|\boldsymbol{\theta}'(s)\|_2^2\tan\overline{\gamma}/L$ holds. Now multiply the implied inequality by $\|\boldsymbol{\theta}'(s)\|_2 \geq 0$ to establish $\|\boldsymbol{\theta}'(s)\|_2^3\tan\overline{\gamma}/L \geq \|\boldsymbol{\theta}''(s)\|_2\|\boldsymbol{\theta}'(s)\|_2 \geq \|\boldsymbol{\theta}'(s) \times \boldsymbol{\theta}''(s)\|_2$. The conclusion follows directly from Lemma 1. □

Proposition 4. *Let $\mathcal{D} \subseteq \mathbb{R}^2$ be a given SOC. If the control points $\boldsymbol{\Theta}_j$ of the B-spline path $\boldsymbol{\theta}(s)$ satisfy:*

$$\boldsymbol{\Theta}_j \in \mathcal{D}, \quad j=0,\dots,N_\theta, \tag{17}$$

then the state-space trajectory $\mathbf{x}(t) \in \mathcal{D}$ for all $t \in [0,t_f)$.

Proof. Condition (17) implies, by Proposition 1, that $\boldsymbol{\theta}(s) \in \mathcal{D}$ for all $s \in [0,1)$. The conclusion follows from Lemma 2. □

Remark 2. *While the obstacle-free space $\mathcal{D}$ is generally nonconvex, the convexity assumption is easily relaxed by considering the concept of "safe corridor" (union of convex sets) and enforcing the conditions of Proposition 4 segment-wise instead of globally. The reader is referred to our previous work [16] and to [26], [27] for more information.*

For fixed values of α and $\hat{\mathbf{r}}$, the conditions of Proposition 3 are convex and we can formulate the following SOCP to find a safe path $\boldsymbol{\theta}(s)$:

$$\begin{aligned} \min. \quad & \int_0^1 \|\boldsymbol{\theta}'''(s)\|_2^2\,\mathrm{d}s + \overline{v}_\theta - \underline{v}_\theta + \overline{a}_\theta \qquad \text{(PATH-SOCP)} \\ \text{s. t.} \quad & \boldsymbol{\Theta}_0 = \mathbf{r}_0, \quad \boldsymbol{\Theta}_{N_\theta} = \mathbf{r}_f, \\ & \boldsymbol{\Theta}_1^{(1)} = \overline{v}_\theta(\cos\psi_0, \sin\psi_0)^T, \\ & \boldsymbol{\Theta}_{N_\theta}^{(1)} = \overline{v}_\theta(\cos\psi_f, \sin\psi_f)^T, \\ & \big\|\boldsymbol{\Theta}_i^{(1)}\big\|_2 \leq \overline{v}_\theta, \quad j=1,\dots,N_\theta, \\ & \text{(16) and (17) hold,} \end{aligned}$$

with decision variables $\beta, \overline{v}_\theta, \underline{v}_\theta, \overline{a}_\theta$ and $\boldsymbol{\Theta}_0, \ldots, \boldsymbol{\Theta}_{N_\theta}$, where $\overline{\gamma}$ is the maximum steering angle, $\mathbf{r}_m \triangleq (x_m, y_m)^T,\ m \in \{0, f\}$ is the initial/final position vector and ψ_m is the initial/final heading angle. Note that the resulting B-spline path $\boldsymbol{\theta}$ has bounded derivatives given by $\overline{v}_\theta$ and $\overline{a}_\theta$ which will be used to guarantee the safety of speed profiles in Section IV-C.

Remark 3. *Proposition 3 contains two convex relaxations by lower-bounding a norm and a convex, quadratic function. Because of the relaxations, the size of the feasible region of* (PATH-SOCP) *depends on the choice of parameters α and $\hat{\mathbf{r}}$. We found the following heuristics worked well:*

$$\alpha = \frac{2\tan\overline{\gamma}}{L}\|\mathbf{r}_f - \mathbf{r}_0\|_2, \quad \hat{\mathbf{r}} = \frac{\mathbf{r}_f - \mathbf{r}_0}{\|\mathbf{r}_f - \mathbf{r}_0\|_2}.$$

B. Temporal Optimization

In this subsection, we find an appropriate trajectory duration t_f by considering a minimization of both t_f and the magnitude of the acceleration vector. Consider a B-spline path $\boldsymbol{\theta}(s)$ feasible in (PATH-SOCP), and the functions $b(s) \triangleq \dot{s}^2$ and $a(s) \triangleq \ddot{s}$, which must satisfy the differential condition $b'(s) = 2a(s)$. Following [6], we have that

$$t_f = \int_0^{t_f} 1\ \mathrm{d}t = \int_0^1 \frac{1}{\dot{s}}\ \mathrm{d}s = \int_0^1 b(s)^{-1/2}\ \mathrm{d}s. \tag{18}$$

Purely minimizing the trajectory duration given by t_f results in maximum-speed velocity profiles. We additionally minimize the acceleration to encourage trajectories with mild friction circle profiles [28] by choosing a Lagrange cost functional $L\big(\mathbf{x}(t), \mathbf{u}(t)\big) \triangleq \dot{v}^2(t) + v^2(t)\dot{\psi}^2(t) = \|\ddot{\mathbf{y}}(t)\|_2^2$. We can now write the objective functional entirely in terms of (and convex with respect to) the new functions $a(s)$ and $b(s)$ as follows:

$$J\big(a(s), b(s)\big) = \int_0^1 \frac{\nu}{\sqrt{b(s)}} + \|a(s)\boldsymbol{\theta}'(s) + b(s)\boldsymbol{\theta}''(s)\|_2^2\ \mathrm{d}s.$$

We follow a similar procedure as in [6] and consider $N_t + 1$ points partitioning the interval $[0, 1]$ into N_t uniform segments with width $\Delta s \triangleq 1/N_t$. The discretized Lagrange cost functional becomes $L(s_i, a_i, b_i) = \|a_i\boldsymbol{\theta}'(s_i) + b_i\boldsymbol{\theta}''(s_i)\|_2^2$, where a_i and b_i are the decision variables representing $a(s_i)$ and $b(s_i)$, respectively. Assuming that $a(s)$ is piecewise constant over each segment $[s_i, s_{i+1}), i = 0, \ldots, N_t - 1$ where $s_i \triangleq i\Delta s$, we can exactly evaluate the integral (18) to avoid the case when $\dot{s} = 0$ as shown in [6]. We formulate the following SOCP to obtain the trajectory duration t_f:

$$\begin{aligned}
\text{min.}\quad & \sum_{i=0}^{N_t-1} 2\nu\Delta s d_i + \sum_{i=0}^{N_t} L(s_i, a_i, b_i) && \text{(TIME-SOCP)}\\
\text{s. t.}\quad & \big\|(2c_i, b_i - 1)^T\big\|_2 \le b_i + 1, \quad i = 0, \ldots, N_t,\\
& \big\|(2, \overline{c}_i - d_i)^T\big\|_2 \le \overline{c}_i + d_i,\ i = 0, \ldots, N_t - 1,\\
& 2\Delta s a_i = b_i - b_{i-1}, \quad i = 1, \ldots, N_t,\\
& b_0\|\boldsymbol{\theta}'(0)\|_2^2 = v_0^2, \quad b_{N_t}\|\boldsymbol{\theta}'(1)\|_2^2 = v_f^2,\\
& b_i\|\boldsymbol{\theta}'(s_i)\|_2^2 \le \overline{v}^2, \quad i = 0, \ldots, N_t,\\
& \Big|a_i\|\boldsymbol{\theta}'(s_i)\|_2 + b_i f_i\Big| \le \overline{a}, \quad i = 0, \ldots, N_t,\\
& \overline{c}_i \triangleq c_{i+1} + c_i, \quad f_i \triangleq \big(\boldsymbol{\theta}'(s_i)\cdot\boldsymbol{\theta}''(s_i)\big)/\|\boldsymbol{\theta}(s_i)\|_2,
\end{aligned}$$

with decision variables $a_i, b_i, c_i, d_i,\ i = 0, \ldots, N_t$. The first two constraints are the SOCP embedding of (18) given in [6], the third constraint is from the differential constraint relating $a(s)$ and $b(s)$, the fourth sets initial and final speeds, the fifth ensures the speed bound is respected and the sixth ensures the acceleration bound is respected. From our assumption that the function $a(s)$ is constant over each $[s_i, s_{i+1})$ segment, we can recover the duration of each segment from the constant acceleration equation $\Delta t_i = (\sqrt{b_i} + \sqrt{b_{i-1}})/a_i,\ i = 1, \ldots, N_t$. The overall duration of the trajectory is then $t_f = \sum_{i=1}^{N_t}\Delta t_i$.

Remark 4. *The solution of* (TIME-SOCP) *provides a safe speed profile at discrete time instances. If continuous-time safety is not critical, it suffices to stop here and retrieve the discretized state-space solution. In addition, if the desired trajectory duration t_f is known, one can skip* (TIME-SOCP) *and proceed to the next SOCP after solving* (PATH-SOCP).

C. Speed Profile Optimization

Assume that t_f is given by the solution of (TIME-SOCP) or specified a priori. Let $\boldsymbol{\theta}(s)$ be a B-spline path feasible in (PATH-SOCP) and $s(t)$ be a B-spline speed profile. The following propositions provide convex relaxations to the conditions of Lemma 3 and Lemma 4.

Proposition 5. *If the condition*

$$0 \le \overline{v}_\theta p_j^{(1)} \le \overline{v}, \quad j = 1, \ldots, N_s, \tag{19}$$

holds, then the state-space trajectory $\mathbf{x}(t) \in \mathcal{S}_v,\ \forall t \in [0, t_f)$, where $\mathcal{S}_v$ is the forward speed safety set *defined in* (12).

Proof. By Proposition 1, (19) implies that $0 \le \overline{v}_\theta\dot{s}(t) \le \overline{v}$ for all $t \in [0, t_f)$. Because $\overline{v} \ge \overline{v}_\theta\dot{s}(t) \ge \dot{s}(t)\|\boldsymbol{\theta}\big(s(t)\big)\|_2 = v(t) \ge 0$, the conclusion follows. □

Proposition 6. *For any given nonnegative vectors $\overline{\boldsymbol{\kappa}} = (\overline{\kappa}_0, \ldots, \overline{\kappa}_{N_s-d_s})^T$ and $\overline{\boldsymbol{\epsilon}} = (\overline{\epsilon}_0, \ldots, \overline{\epsilon}_{N_s-d_s})^T$, if the following conditions*

$$0 \le p_j^{(1)} \le \overline{\kappa}_k, \quad j = k+1, \ldots, k+d, \tag{20a}$$

$$-\overline{\epsilon}_k \le p_j^{(2)} \le \overline{\epsilon}_k, \quad j = k+2, \ldots, k+d, \tag{20b}$$

$$\big\|A_{\dot{v}}(\overline{\kappa}_k, \overline{\epsilon}_k)^T + \mathbf{b}_{\dot{v}}\big\|_2 \le \begin{bmatrix}\overline{\kappa}_k & \overline{\epsilon}_k\end{bmatrix}\mathbf{c}_{\dot{v}} + d_{\dot{v}}, \tag{20c}$$

hold for all $k \in \{0, \ldots, N_s - d_s\}$, where $A_{\dot{v}} = \mathrm{diag}(\sqrt{2\overline{a}_\theta}, -\overline{v}_\theta/\sqrt{2})$, $\mathbf{b}_{\dot{v}} = (0, (\overline{a}-1)/\sqrt{2})^T$, $\mathbf{c}_{\dot{v}} = (0, -\overline{v}_\theta/\sqrt{2})^T$ and $d_{\dot{v}} = (\overline{a}+1)/\sqrt{2}$, then the trajectory $\mathbf{u}(t) \in \mathcal{S}_{\dot{v}},\ \forall t \in [0, t_f)$ where $\mathcal{S}_{\dot{v}}$ is the linear acceleration safety set *defined in* (13).

Proof. The first two conditions imply, by Proposition 1, that for any $k \in \{0, \ldots, N_s - d_s\}$, $0 \le \dot{s}(t) \le \overline{\kappa}_k$ and $|\ddot{s}(t)| \le \overline{\epsilon}_k$ hold $\forall t \in [\tau_{k+d_s}, \tau_{k+d_s+1})$. Expanding the last condition, we determine that for any $k \in \{0, \ldots, N_s - d_s\}$, $0 \le \overline{\kappa}_k^2\overline{a}_\theta + \overline{\epsilon}_k\overline{v}_\theta = |\overline{\kappa}_k^2\overline{a}_\theta| + |\overline{\epsilon}_k\overline{v}_\theta| \le \overline{a}$. Notice that for any $k \in \{0, \ldots, N_s - d_s\}$, the following conditions hold for all $t \in [\tau_{k+d_s}, \tau_{k+d_s+1})$: $|\overline{\epsilon}_k\overline{v}_\theta| \ge |\ddot{s}(t)|\|\boldsymbol{\theta}'(s(t))\|_2 = |a_t(t)|$, $|\overline{\kappa}_k^2\overline{a}_\theta| \ge \big|\dot{s}^2(t)\big(\boldsymbol{\theta}'(s(t))\cdot\boldsymbol{\theta}''(s(t))\big)/\|\boldsymbol{\theta}'(s(t))\|\big| = |a_n(t)|$, where a_t and a_n are defined in (15). Therefore, by the triangle inequality $\overline{a} \ge |a_t(t)| + |a_n(t)| \ge |a_t(t) + a_n(t)|$ holds for all $t \in [\tau_{k+d_s}, \tau_{k+d_s+1})$. Recalling that the knot vector $\boldsymbol{\tau}$ is clamped and uniform (3), segmenting the interval $[0, t_f]$, and

that the above inequality holds for all $k \in \{0, \dots, N_s - d_s\}$, we can establish that the inequality holds, in fact, for all $t \in [0, t_f)$. The conclusion now follows by Lemma 4. □

The vectors $\overline{\boldsymbol{\kappa}}, \overline{\boldsymbol{\epsilon}} \in \mathbb{R}^{N_s - d_s + 1}$ represent segment-wise bounds on the first and second order derivatives of $s(t)$, respectively. We will introduce them as optimization variables in the following SOCP to obtain the speed profile.

$$\begin{aligned} \underset{\overline{\boldsymbol{\kappa}}, \overline{\boldsymbol{\epsilon}}, p_0, \dots, p_{N_s}}{\text{min.}} \quad & \int_0^{t_f} \dddot{s}^2(t)\, \mathrm{d}t \qquad \text{(SPEED-SOCP)} \\ \text{s. t.} \quad & p_0 = 0, \quad p_{N_s} = 1, \\ & \overline{v}_\theta p_1^{(1)} = v_0, \quad \overline{v}_\theta p_{N_s}^{(1)} = v_f, \\ & \text{(19) and (20) hold,} \end{aligned}$$

where v_0 and v_f are given initial and final speeds, respectively.

Remark 5. *The spatio-temporal separation presents both a challenge and an opportunity: on the one hand, we lose optimality with respect to the original optimal control problem because the divided cost functions usually fail to represent the original one; on the other hand, we gain the flexibility to separately define desirable paths, trajectory durations and speed profiles. We found that the cost functions for the SOCPs yielded trajectories with small costs when evaluated in the original optimal control problem's cost function.*

D. Safety Analysis

The following theorem summarizes the main theoretical contributions of this paper.

Theorem 1. *Let $\boldsymbol{\theta}(s)$ and $s(t)$ be, respectively, a B-spline path feasible in* (PATH-SOCP) *and a B-spline speed profile feasible in* (SPEED-SOCP) *with duration t_f. The state-space trajectory of* (1) *obtained by passing the parameterized flat outputs* (7a) *through the flat map* (2) *satisfies the initial $\mathbf{x}(0) = \mathbf{x}_0$ and final $\mathbf{x}(t_f) = \mathbf{x}_f$ conditions as well as safety specifications:*

$$\big(\mathbf{x}(t), \mathbf{u}(t)\big) \in \mathcal{S}_\gamma, \quad \mathbf{x}(t) \in \mathcal{S}_\mathcal{D} \cap \mathcal{S}_v, \quad \mathbf{u}(t) \in \mathcal{S}_{\dot{v}}, \quad (21)$$

$\forall t \in [0, t_f)$. Thus, t_f, $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are feasible in (OPT).

Proof. The initial and final positions (x, y) are satisfied by $\boldsymbol{\theta}(0) = \mathbf{r}_0 = (x_0, y_0)^T$ and $\boldsymbol{\theta}(1) = \mathbf{r}_f = (x_f, y_f)^T$. Also, $\|\boldsymbol{\theta}(s)\|_2 = \overline{v}_\theta$ for $s \in \{0, 1\}$. Since $s(0) = 0$, $s(t_f) = 1$, $\overline{v}_\theta \dot{s}(0) = v_0$ and $\overline{v}_\theta \dot{s}(t_f) = v_f$, the flat map (2) implies that the obtained state trajectory $\mathbf{x}(t)$ satisfies the specified initial and final velocities v_0 and v_f. For ψ_0 and ψ_f, notice that $\boldsymbol{\theta}'(m) = \big(x'(m), y'(m)\big)^T = \overline{v}_\theta \big(\cos\psi_q, \sin\psi_q\big)^T$, with $(m, q) \in \{(0,0), (1,f)\}$. Since the heading $\psi(t)$ is given only by the path $\boldsymbol{\theta}(s)$, it follows that $\psi(p) = \arctan\big(\overline{v}_\theta \sin\psi_q / (\overline{v}_\theta \cos\psi_q)\big) = \psi_q$, $(p, q) \in \{(0,0), (1,f)\}$. Finally, (21) follows by Propositions 3, 4, 5 and 6. □

V. Simulation Examples

We evaluate the performance and efficiency of the proposed approach[1] by three simulation examples. In all examples, we use B-spline path $\boldsymbol{\theta}$ parameters $d_\theta = 4$ and $N_\theta = 20$, and B-spline speed profile s parameters $d_s = 4$ and $N_s = 20$. For (TIME-SOCP), we use $N_t = 40$. We solve the three SOCPs using YALMIP [29] with MOSEK solver [20].

[1]Source code: https://github.com/wisc-arclab/FlatVCP.

Example 1. *Consider* (OPT) *with the following problem data: initial state $\mathbf{x}_0 = \mathbf{0}_{4\times1}$, final state $\mathbf{x}_f = (100, 4, 0, 0)^T$, obstacle-free space $\mathcal{D} = \mathbb{R}^2$, maximum steering angle $\overline{\gamma} = 0.0044$ rad (0.25 degrees), maximum speed $\overline{v} = 4.2$ m/s, acceleration bound $\overline{a} = 0.6$ m/s^2, duration penalty factor $\nu = 1$ and wheelbase length $L = 2.601$ meters. Note that this problem requires rest-to-rest motion to be solved and, as described in previous sections, the flat map has singularities when the speed is zero. Nonetheless, our approach is able to handle this gracefully. We continue by solving the three proposed SOCP problems sequentially. We then pass the resulting path and speed profile through the flat map* (2) *to obtain corresponding state $\mathbf{x}(t)$ and input $\mathbf{u}(t)$ trajectories. The resulting state and input trajectories are shown in Figure 3 along with the steering angle $\gamma(t)$. The speed v the acceleration $\dot{v}$ and the steering angle γ respect their bounds for all time $t \in [0, t_f]$.*

Example 2. *We compare the performance and optimality of the proposed framework with two general optimal control solvers, ICLOCS2 [31] and OpenOCL [30], and a state-of-the-art algorithm for vehicle motion planning, Convex Elastic Smoothing (CES) [8]. The vehicle considered is a 2021 Bolt EV by Chevrolet with a wheelbase length of $L = 2.601$ meters. We assume a 2-lane, straight, road with a posted speed limit of 40 miles per hour and consider a left lane change. The initial and final states are specified as $\mathbf{x}_0 = (0, 0, 16, 0)^T$ and $\mathbf{x}_f = (75, 3.7, 17.5, 0)^T$, respectively. The other parameters, when applicable, are chosen as $\overline{\gamma} = 0.785$ (45 degrees), $\mathcal{D} = \mathbb{R}^2$, $\overline{v} = 19$ m/s, $\overline{a} = 2$ m/s^2 and $\nu = 1$. With these parameters, we solve the optimal control problem* (OPT) *with the proposed framework, ICLOCS2 with analytical derivatives provided, direct collocation transcription and 40 discretization samples, OpenOCL with fast, automatic differentiation using CasADi [32] and 40 discretization samples, and we use 40 trajectory samples and iterate 5 times over the CES algorithm alternating between elastic stretching and speed optimization. The resulting trajectories $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are shown in Figure 4. To compare the computational burden of each algorithm, we collect average solve times with each approach. The results are shown in Table I. In this example, the proposed approach achieved solve times nearly four times faster than the next leading method. In addition, the objective value of the proposed approach, while higher than some, was still comparable to that of the other solvers.*

Example 3. *We demonstrate the real-time capabilities of*

TABLE I
Efficiency and optimality of compared solvers.

	Avg solve time [ms]	**Objective value** [-]
Proposed method	28.8	6.8495
CES (5-iter) [8]	93.1	7.1334
OpenOCL [30]	94.3	6.5534
ICLOCS2 [31]	257	6.8134

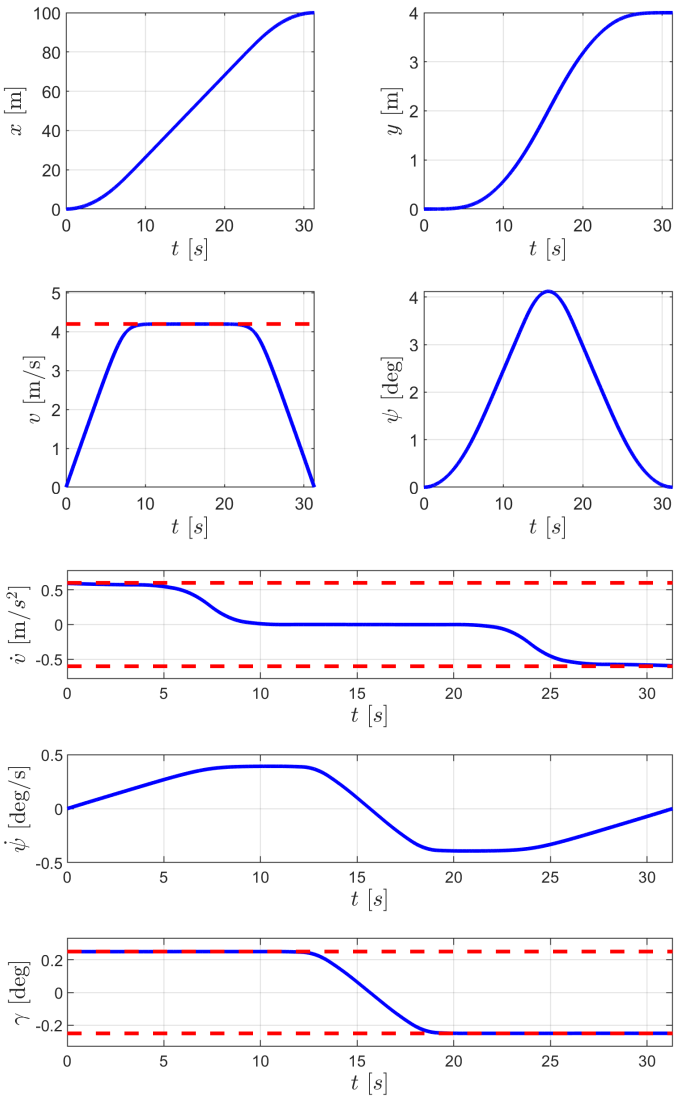

Fig. 3. The state (top) and input (bottom) trajectories of Example 1 show that the required safety constraints are all satisfied for all $t \in [0, t_f]$: the speed respects $v(t) \leq 4.2$ m/s, the acceleration respects $|\dot{v}(t)| \leq 0.6$ m/s^2 and the steering angle respects $|\gamma(t)| \leq 0.25$ deg.

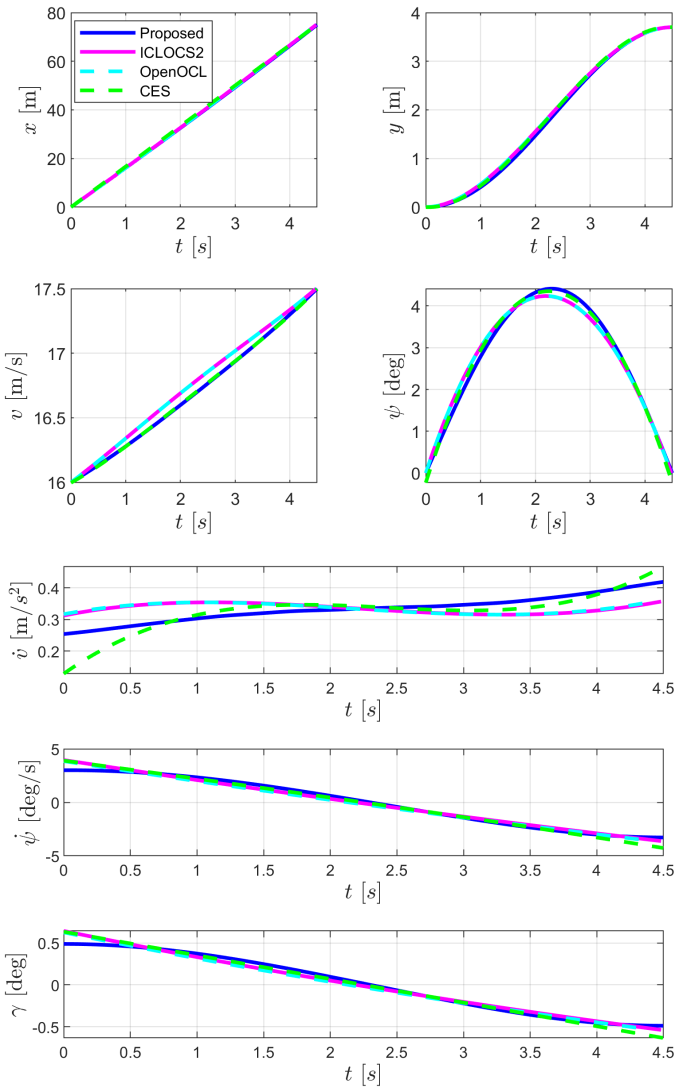

Fig. 4. Simulation results of Example 2 showing the state and input trajectories obtained with each solver.

the proposed approach in a simulated scenario located at Mcity's main roundabout. The vehicle begins from rest at the roundabout entrance and adjusts to existing traffic to take the roundabout's second exit. We enforce the constraint $v(t) \leq \overline{v} = 11.176$ m/s (25 mph). We also consider two actors driving around the roundabout with constant speed of 5 m/s. The adjustment to traffic is done by simple behavioral logic as follows: If the planned trajectory, with desired final position $\mathbf{r}_f$ located 15 meters ahead on the road, is obstacle-free, it is used; otherwise, if it collides with the vehicle in front, we adjust the endpoint of the trajectory to be 8 meters behind the leading vehicle (but still on the road's center line) and enforce a final speed $v_f = 2.5$ m/s. The proposed framework is executed at 25 Hz in the sense that the computation (including all three SOCPs and other algebraic operations) of each iteration can be done within 0.04 seconds. The scenario is rendered in MATLAB's 3D simulation environment powered by Unreal Engine [33]. We show snapshots of the trajectory at five different time steps in Figure 5. The figure also shows a bird's eye view of the scenario and the planned trajectory at the current time step (gray line).

VI. Conclusions

We presented a sequential SOCP approach to solve the optimal control problem with a kinematic bicycle model where solutions are guaranteed to satisfy the constraints in the continuous-time sense. We also compared the performance of the proposed method with state-of-the-art optimal control solvers/algorithms to demonstrate its efficiency in simulated scenarios. In future work, we will further explore the flexibility of the proposed approach such as achieving dynamic obstacle avoidance by only resolving the SPEED-SOCP with added constraints. We will also generalize our work to more realistic settings such as considering dynamic obstacle constraints and imperfect system models.

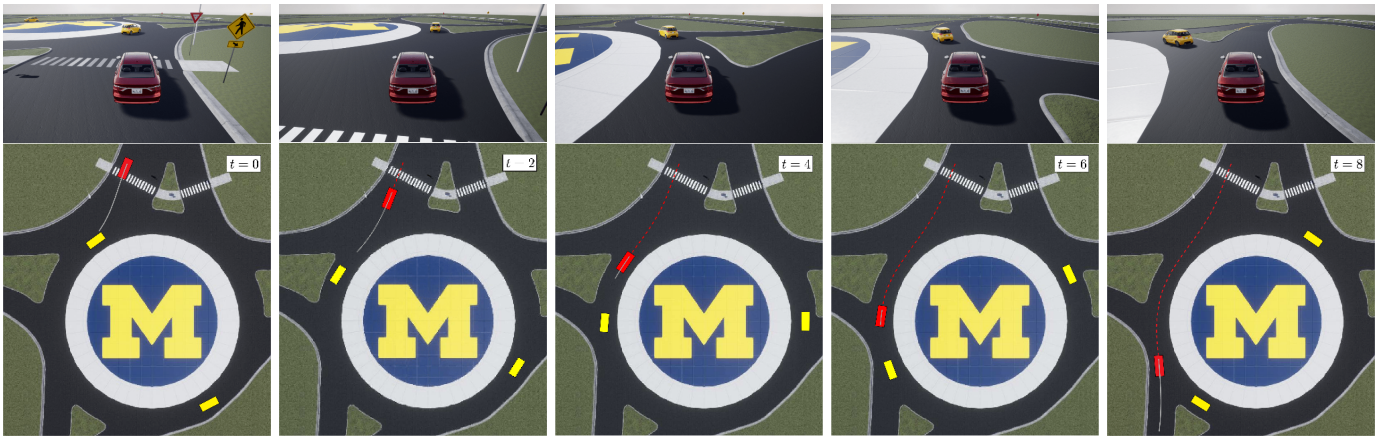

Fig. 5. Navigation using the proposed approach to solve (OPT) at 25 Hz in Example 3. The figure shows the scenario at times $t = 0, 2, 4, 6, 8$ seconds.

REFERENCES

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58 443–58 469, 2020.

[2] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1216–1229, 2018.

[3] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[4] M. Elbanhawi, M. Simic, and R. Jazar, "Randomized bidirectional B-spline parameterization motion planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 406–419, 2015.

[5] Y. Zhang, H. Chen, S. L. Waslander, J. Gong, G. Xiong, T. Yang, and K. Liu, "Hybrid trajectory planning for autonomous driving in highly constrained environments," *IEEE Access*, vol. 6, pp. 32 800–32 819, 2018.

[6] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[7] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, 2014.

[8] Z. Zhu, E. Schmerling, and M. Pavone, "A convex optimization approach to smooth trajectories for motion planning with car-like robots," in *IEEE Conference on Decision and Control*. IEEE, 2015, pp. 835–842.

[9] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "Guaranteeing consistency in a motion planning and control architecture using a kinematic bicycle model," in *IEEE American Control Conference*. IEEE, 2018, pp. 3981–3987.

[10] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *IEEE Intelligent Vehicles Symposium*. IEEE, 2015, pp. 1094–1099.

[11] A. Liniger, X. Zhang, P. Aeschbach, A. Georghiou, and J. Lygeros, "Racing miniature cars: Enhancing performance using stochastic MPC and disturbance feedback," in *IEEE American Control Conference*. IEEE, 2017, pp. 5642–5647.

[12] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *IEEE Intelligent Vehicles Symposium*. IEEE, 2017, pp. 812–818.

[13] P. Rouchon, M. Fliess, J. Lévine, and P. Martin, "Flatness and motion planning: The car with n trailers," in *European Control Conference*, 1993, pp. 1518–1522.

[14] M. J. Van Nieuwstadt and R. M. Murray, "Real-time trajectory generation for differentially flat systems," *International Journal of Robust and Nonlinear Control*, vol. 8, no. 11, pp. 995–1020, 1998.

[15] F. Stoican, I. Prodan, D. Popescu, and L. Ichim, "Constrained trajectory generation for UAV systems using a B-spline parametrization," in *Mediterranean Conference on Control and Automation*. IEEE, 2017, pp. 613–618.

[16] V. Freire and X. Xu, "Flatness-based quadcopter trajectory planning and tracking with continuous-time safety guarantees," *arXiv preprint arXiv:2111.00951*, 2021.

[17] C. De Boor and C. De Boor, *A Practical Guide to Splines*. Springer-Verlag: New York, 1978, vol. 27.

[18] F. Suryawan, "Constrained trajectory generation and fault tolerant control based on differential flatness and B-splines," Ph.D. dissertation, The University of Newcastle, 2012.

[19] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and Its Applications*, vol. 284, no. 1-3, pp. 193–228, 1998.

[20] MOSEK, *The MOSEK optimization toolbox for MATLAB manual. Version 9.3*, 2021.

[21] D. Dueri, Y. Mao, Z. Mian, J. Ding, and B. Açikmeşe, "Trajectory optimization with inter-sample obstacle avoidance via successive convexification," in *IEEE Conference on Decision and Control*. IEEE, 2017, pp. 1150–1156.

[22] A. Richards and O. Turnbull, "Inter-sample avoidance in trajectory optimizers using mixed-integer linear programming," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 4, pp. 521–526, 2015.

[23] D. P. Pedrosa, A. A. Medeiros, and P. J. Alsina, "Point-to-point paths generation for wheeled mobile robots," in *International Conference on Robotics and Automation*, vol. 3. IEEE, 2003, pp. 3752–3757.

[24] P. Martin, R. M. Murray, and P. Rouchon, "Flat systems, equivalence and trajectory generation," California Institute of Technology, Tech. Rep., 2003.

[25] I. M. Ross and F. Fahroo, "Pseudospectral methods for optimal motion planning of differentially flat systems," in *IEEE Conference on Decision and Control*, vol. 1. IEEE, 2002, pp. 1135–1140.

[26] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," in *IEEE International Conference on Robotics and Automation*. IEEE, 2018, pp. 344–351.

[27] W. Sun, G. Tang, and K. Hauser, "Fast UAV trajectory optimization using bilevel optimization with analytical gradients," in *IEEE American Control Conference*. IEEE, 2020, pp. 82–87.

[28] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[29] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *International Conference on Robotics and Automation*. IEEE, 2004, pp. 284–289.

[30] J. Koenemann, G. Licitra, M. Alp, and M. Diehl, "OpenOCL–open optimal control library," 2017.

[31] Y. Nie, O. Faqir, and E. C. Kerrigan, "ICLOCS2: Try this optimal control problem solver before you try the rest," in *UKACC 12th International Conference on Control*. IEEE, 2018, pp. 336–336.

[32] R. Verschueren, G. Frison, D. Kouzoupis, N. van Duijkeren, A. Zanelli, R. Quirynen, and M. Diehl, "Towards a modular software package for embedded optimization," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 374–380, 2018.

[33] MathWorks®, *Vehicle Dynamics Blockset™*. Release R2022a, 2022.