Optimal motion planning with the half-car dynamical model for autonomous high-speed driving

Optimal Motion Planning with the Half-Car Dynamical Model for Autonomous High-Speed Driving

Jeong hwan Jeon*, Raghvendra V. Cowlagi*†, Steven C. Peters**, Sertac Karaman*, Emilio Frazzoli*, Panagiotis Tsiotras††, and Karl Iagnemma**

***Abstract*— We investigate the application of the RRT* optimal motion planning algorithm to autonomous high-speed driving. Specifically, we discuss the implementation of RRT* for the half-car dynamical model. To enable fast solutions of the associated local steering problem, we observe that the motion of a special point (viz., the front center of oscillation) can be modeled as a double integrator augmented with fictitious inputs. We first map the constraints on tire friction forces to constraints on these augmented inputs, which provides instantaneous, state-dependent bounds on the curvature of geometric paths feasibly traversable by the front center of oscillation. Next, we map the vehicle's actual inputs to the augmented inputs. The local steering problem for the half-car dynamical model can then be transformed to a simpler steering problem for the front center of oscillation, which we solve efficiently by first constructing a curvature-bounded geometric path and then imposing a suitable speed profile on this geometric path. Finally, we demonstrate the efficacy of the proposed motion planner via numerical simulation results.**

I. Introduction

Motion planning for autonomous mobile vehicles [1] has traditionally focused on relatively simple unicycle-type kinematic models, owing both to the sufficiency of the resultant plans for low-speed vehicle motion in structured environments, and to the computational efficiency afforded by low-dimensional models, which enables real-time implementations. Typically, the resultant trajectories do not exploit advantageously the vehicle's maneuvering capabilities, and motion planning based on such simple vehicle models may thus be unsuitable for enabling autonomous high-speed driving in complex, dynamic, and unstructured environments.

However, motion planning for higher-dimensional, higher-fidelity vehicle dynamical models is, in general, difficult. Whereas computationally efficient motion planning in high-dimensional configuration- and state spaces, is enabled by randomized sampling-based algorithms [2], [3], these algorithms ignore the quality of the resultant motion, and often result in highly sub-optimal motion plans. Recent developments in optimal randomized sampling-based planning [4], and in deterministic approaches that include vehicle dynamical constraints [5] promise fast computation of near-optimal paths with high-fidelity vehicle dynamical models.

*Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139. Email: jhjeon, rcowlagi, sertac, frazzoli@mit.edu.

**Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139. Email: scpeters, kdi@mit.edu.

††School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332. Email: tsiotras@gatech.edu.

†Corresponding author.

Both of the aforementioned approaches, however, rely strongly on a *local steering* algorithm that computes an optimal (or near-optimal) control input to steer the vehicle from a specified initial state to a specified final state, neither of which is necessarily an equilibrium state. Whereas the solution of this problem is difficult for nonlinear dynamical systems in general, we may exploit, in some specific cases, the structure of the dynamical system to solve the local steering problem.

In particular, the property of *differential flatness* [6] may be advantageously used. The states and inputs of differentially flat dynamical systems can be fully recovered from the so-called *flat outputs* of the system (and their derivatives). Also, differentially flat systems can be linearized via feedback based on the flat outputs and their derivatives (called *endogenous* feedback [6]). In the context of motion planning for mobile robotic vehicles, differential flatness of the vehicle dynamical model is particularly useful when the flat outputs are workspace[1] position coordinates of some point (possibly fictitious) associated with the vehicle. In such cases, the vehicle dynamical behavior may be inferred from trajectories in the workspace. Conversely, the problem of motion planning subject to vehicle dynamical constraints may be transformed to a workspace trajectory generation problem, which is beneficial because of the low dimensionality of the workspace and because of the ease of collision checking with (workspace) obstacles.

In this paper, we consider the design of a computationally efficient technique for optimal, planar motion planning subject to the dynamical constraints imposed by the half-car model: in particular, we are interested in finding *time-optimal* trajectories for the vehicle. The half-car model [7] is a model of wheeled vehicles that captures yaw dynamics and includes acceleration constraints in the form of constraints on the ground-tire friction forces. We investigate the implementation of the RRT* algorithm [4] for this motion planning problem. This implementation entails a computationally efficient solution of the so-called *local steering problem*, which involves finding admissible control inputs that transfer the vehicle state between two pre-specified boundary conditions. To this end, we investigate the motion of a special point associated with the vehicle, that is called its *Huygens center of oscillation* (CO), and we transform the local steering problem for the vehicle to a problem of workspace trajectory generation for the CO.

[1]The *workspace* is the planar region in which the mobile vehicle operates.

The Huygens centers of oscillation at the front and rear of the vehicle are associated with differential flatness of models of front-steered vehicles that incorporate wheel slip[2]. A crucial physical property of a CO is that its acceleration is independent of one of the lateral tire friction forces. For example, the acceleration of the front CO is independent of the rear lateral tire force [9]. The coordinates in a body-fixed frame of the velocity of the rear CO have been identified as flat outputs of a half-car model that incorporates wheel slip but does not consider load transfer [10]. More pertinent to the context of motion planning, the coordinates in an inertial frame of the *position* of the front CO have been identified as "pseudo-flat" outputs for the half-car model [11]. The mapping from these outputs (and their derivatives) to the states and inputs of the vehicle involve not only algebraic equations, but also differential equations (of a significantly lower order than the vehicle's equations of motion). Furthermore, the position coordinates of front and/or rear CO have been considered as reference inputs for trajectory tracking controllers in [12] (rear CO), in [13] (both front and rear CO with four-wheel steering), and in [14] (front CO).

The problem of motion planning for the half-car model has not been well-addressed in the literature. Numerical optimal control techniques have been applied in [15] for reproducing autonomously a standard rally-driving aggressive maneuver (namely, the trail-braking maneuver), and in [16] for generating a minimum-time double lane-change maneuver. The generation of a minimum-time speed profile for a half-car traversing a given geometric path has been addressed in [17]. Preliminary results on the implementation of the RRT* algorithm for the half-car model have appeared in [18], where the local steering algorithm was implemented using (computationally intensive) numerical optimal control.

The main contributions of this work are as follows. Firstly, we drop the simplifications to the half-car model adopted in [10], [11]: specifically, we allow *normal load transfer* between the front and rear tires – a technique used commonly by rally racing drivers to control the yaw dynamics [15] – and we consider as inputs the longitudinal slips of the front and rear tires instead of the longitudinal tire forces (as considered in [10], [11]). Manipulating the longitudinal tire slips (with thrust/brakes) is more realistic than manipulating longitudinal forces because these forces depend on the *total* tire slips, not the longitudinal slips alone. Secondly, we map analytically the constraints on tire friction forces to equivalent constraints on the flat output trajectory: friction force constraints are ignored in similar earlier works [10], [11]. Thirdly, we provide a computationally efficient local steering algorithm for the half-car dynamical model, which may be applied independently in conjunction with motion planners different from that considered in this paper (RRT*). Finally, we study an implementation of RRT* with the preceding local steering algorithm, which is an important and fundamental component in the development of an autonomous high-speed driving system that utilizes the full envelope of the vehicle's maneuvering capabilities.

[2]The position of the rear wheel is a flat output of a front-steered half-car with no wheel slip [6], [8].

The rest of this paper is organized as follows. In Section II, we describe the half-car dynamics model and discuss its differential flatness properties. In Section III, we summarize the RRT* motion planner, and in Section IV, we discuss an efficient local steering algorithm for the half-car model that is invoked repeatedly by RRT*. In Section V, we provide simulation results of the said implementation of RRT*. Finally, we conclude the paper in Section VI with remarks about future extensions of this work.

II. The Half-Car Model

The half-car dynamical model is used in applications where the vehicle's position, heading, and sideslip are of primary interest (cf. [7], [10], [11], [15] and references therein). We consider a half-car model, as shown in Fig. 1, with mass m, and yaw moment of inertia I_z. We denote by $\mathbf{p}_{\rm cg}$ the position vector of the center of gravity (CG) with respect to a pre-specified inertial axis system; by ψ the heading of the vehicle, and by v_x and v_y the components in a body-fixed axis system (with positive x-axis along the vehicle's heading) of the velocity $\mathbf{v}$ of the CG. We denote by $\ell_{\rm f}$ and $\ell_{\rm r}$, respectively, the distances of the centers of the front and rear wheel from the center of gravity; by h the height of the CG; and by $F_{\alpha\beta}$, $\alpha \in \{{\rm f},{\rm r}\}$, $\beta \in \{x,y\}$, the components in axes attached to the tires (with the x-axis in the plane of the tire) of frictional forces of the front and rear tires. The equations of motion for the half-car model are:

$$m\dot{v}_x = (F_{{\rm f}x}\cos\delta - F_{{\rm f}y}\sin\delta + F_{{\rm r}x}) + mv_y\dot{\psi}, \quad (1)$$

$$m\dot{v}_y = (F_{{\rm f}x}\sin\delta + F_{{\rm f}y}\cos\delta + F_{{\rm r}y}) - mv_x\dot{\psi}, \quad (2)$$

$$I_z\ddot{\psi} = \ell_{\rm f}\left(F_{{\rm f}x}\sin\delta + F_{{\rm f}y}\cos\delta\right) - \ell_{\rm r}F_{{\rm r}y}, \quad (3)$$

where δ is the steering angle of the front wheel, which we consider a control input. In what follows, we denote by ξ the state of the vehicle, i.e., $\xi = (p_{{\rm cg},x}, p_{{\rm cg},y}, \psi, v_x, v_y, \dot{\psi})$. For simplicity, we do not consider suspension dynamics and wheel dynamics in this paper, and we assume that only the front wheel is steerable.

Next, the lateral slips $s_{\alpha y}$, $\alpha \in \{{\rm f},{\rm r}\}$, of the front and rear tires are given by

$$s_{{\rm f}y} = \frac{(v_y + \ell_{\rm f}\dot{\psi})\cos\delta - v_x\sin\delta}{v_x\cos\delta + (v_y + \ell_{\rm f}\dot{\psi})\sin\delta}, \quad s_{{\rm r}y} = \frac{v_y - \ell_{\rm r}\dot{\psi}}{v_x}, \quad (4)$$

We consider as control inputs the longitudinal slips $s_{\alpha x}$, $\alpha \in \{{\rm f},{\rm r}\}$, of the front and rear tires. The total tire slips s_α, $\alpha \in \{{\rm f},{\rm r}\}$, are given by $s_\alpha = \sqrt{s_{\alpha x}^2 + s_{\alpha y}^2}$. The tire friction forces $F_{\alpha\beta}$, $\alpha \in \{{\rm f},{\rm r}\}$, $\beta \in \{x,y\}$, may then be computed from the tire slips $s_{\alpha\beta}$ as follows:

$$F_{\alpha\beta} = \mu_{\alpha\beta}F_{\alpha z}, \qquad \alpha \in \{{\rm f},{\rm r}\},\ \beta \in \{x,y\}, \quad (5)$$

where $F_{\alpha z}$ are the normal loads on the front and rear tires, given by (cf. [15]):

$$F_{{\rm f}z} = \frac{mg\left(\ell_{\rm r} - \mu_{{\rm r}x}h\right)}{\ell_{\rm f} + \ell_{\rm r} + h\left(\mu_{{\rm f}x}\cos\delta - \mu_{{\rm f}y}\sin\delta - \mu_{{\rm r}x}\right)}, \quad (6)$$

$$F_{{\rm r}z} = mg - F_{{\rm f}z}, \quad (7)$$

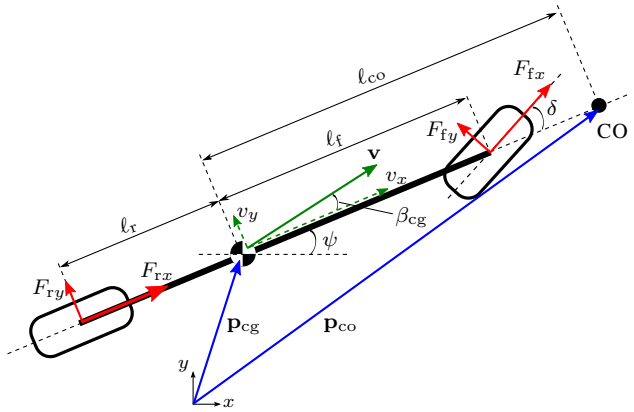


Fig. 1. The bicycle dynamical model: position vectors are in blue, velocity vectors are in green, and forces are in red color.

and $\mu_{\alpha\beta}$ are friction coefficients given by Pacejka's magic formula [19]:

$$\mu_{\alpha\beta} := -(s_{\alpha\beta}/s_\alpha)\mu_\alpha \quad (8)$$

$$\text{with} \quad \mu_\alpha = D_\alpha \sin\left(C_\alpha \tan^{-1}\left(B_\alpha s_\alpha\right)\right), \quad (9)$$

for $\alpha \in \{\mathrm{f},\mathrm{r}\}$, $\beta \in \{x, y\}$, where B_α, C_α, and D_α are constants. Note that (6)-(7) capture the load transfer effect, i.e., the normal tire loads depend upon the front and rear longitudinal tire slips, which relate to thrust/brake inputs.

A. Differential Flatness of the Half-Car Model

Following the work of Peters *et al* [11], we consider as a candidate flat output the position $\mathbf{p}_{\rm co}$ of the front CO, which is a point at a distance $\ell_{\rm co} := I_z/m\ell_{\rm r}$ from the CG along the vehicle's heading [9], i.e.,

$$\mathbf{p}_{\rm co} = \begin{bmatrix} p_{{\rm co},x} \\ p_{{\rm co},y} \end{bmatrix} := \mathbf{p}_{\rm cg} + R(\psi)\begin{bmatrix} \ell_{\rm co} \\ 0 \end{bmatrix}, \quad (10)$$

where $R(\psi)$ is the rotation matrix

$$R(\psi) := \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}.$$

It may be shown (cf. [11]) that

$$\ddot{\mathbf{p}}_{\rm co} = R(\psi)\begin{bmatrix} \dot{v}_x - v_y\dot{\psi} - \ell_{\rm co}\dot{\psi}^2 \\ \dot{v}_y + v_x\dot{\psi} + \ell_{\rm co}\ddot{\psi} \end{bmatrix}. \quad (11)$$

We designate as an augmented input $\mathbf{u} := \ddot{\mathbf{p}}_{\rm co}$ the inertial acceleration $\ddot{\mathbf{p}}_{\rm co}$ of the CO, and we denote by $(u_{\rm t}, u_{\rm n})$ the body-axis coordinates of the augmented input. To map the augmented input to the vehicle control inputs $(s_{{\rm f}x}, s_{{\rm r}x}, \delta)$, note that, by (1)-(3) and (11),

$$\begin{bmatrix} u_{\rm t} \\ u_{\rm n} \end{bmatrix} = \frac{1}{m}\begin{bmatrix} (F_{{\rm f}x}\cos\delta - F_{{\rm f}y}\sin\delta + F_{{\rm r}x}) - \ell_{\rm co}\dot{\psi}^2 \\ (F_{{\rm f}x}\sin\delta + F_{{\rm f}y}\cos\delta + F_{{\rm r}y}) + \ell_{\rm co}\ddot{\psi} \end{bmatrix}$$

$$= \frac{1}{m}\begin{bmatrix} (F_{{\rm f}x}\cos\delta - F_{{\rm f}y}\sin\delta + F_{{\rm r}x}) - m\ell_{\rm co}\dot{\psi}^2 \\ \left(\frac{\ell_{\rm f}+\ell_{\rm r}}{\ell_{\rm r}}\right)(F_{{\rm f}x}\sin\delta + F_{{\rm f}y}\cos\delta) \end{bmatrix}. \quad (12)$$

Firstly, observe that (12) is an under-determined system of equations (i.e., two equations in three unknowns), which implies that the three vehicle control inputs $(s_{{\rm f}x}, s_{{\rm r}x}, \delta)$ cannot be determined uniquely from the trajectory $t \mapsto \mathbf{p}_{\rm co}(t)$ of CO. (Note that the forces $F_{\alpha\beta}$, $\alpha \in \{\mathrm{f},\mathrm{r}\}, \beta \in \{x, y\}$ depend on $s_{{\rm f}x}, s_{{\rm r}x}$ via (4)-(9).) However, if one of the three inputs is treated as an "exogenous" input that may be manipulated independently of $\mathbf{p}_{\rm co}(t)$, then we may subsequently determine the other two inputs.

Secondly, observe that, whereas ψ and $\dot{\psi}$ appear in (12), the computation of ψ and $\dot{\psi}$ involves the solution of the second-order differential equation (3). Consequently, the mapping from the coordinates of $\mathbf{p}_{\rm co}$ and their derivatives to the control inputs of the vehicle is a system of coupled algebraic-differential equations. The position $\mathbf{p}_{\rm co}$ of the CO may thus be considered a "pseudo-flat" output.

Finally, observe that (12) is a complicated system of nonlinear equations. In what follows, we outline an analytic solution, if one exists, of (12), without recourse to numerical means of solution.

To this end, we first non-dimensionalize the physical quantities involved by dividing all lengths by $\ell_{\rm f}+\ell_{\rm r}$, all velocities by $\sqrt{g(\ell_{\rm f}+\ell_{\rm r})}$, all angular velocities by $\sqrt{g/(\ell_{\rm f}+\ell_{\rm r})}$, all accelerations by g, and all forces by mg. In a minor abuse of notation, all symbols in the remainder of the paper represent non-dimensionalized quantities. For analytical simplicity, we assume as an exogenous input the rear tire longitudinal slip $s_{{\rm r}x}$, which may be manipulated independently of $\mathbf{p}_{\rm co}(t)$. To compute the other two control inputs – the steering angle δ and the front tire longitudinal slip $s_{{\rm f}x}$ – after selecting $s_{{\rm r}x}$, we perform the following computations.

Let $(u_{\rm t}, u_{\rm n})$ denote the body-axis coordinates of the augmented input. Then, we may rewrite (12) as

$$u_{\rm t} = F_{{\rm f}x}\cos\delta - F_{{\rm f}y}\sin\delta + F_{{\rm r}x} - \ell_{\rm co}\dot{\psi}^2,$$
$$u_{\rm n} = (\ell_{\rm f}+\ell_{\rm r})\left(F_{{\rm f}x}\sin\delta + F_{{\rm f}y}\cos\delta\right)/\ell_{\rm r}.$$

Then, it follows after algebraic manipulations of (5)–(8) that

$$F_{{\rm f}z} = -h\left(u_{\rm t} - (\ell_{\rm r}/h) + \ell_{\rm co}\dot{\psi}^2\right)/(\ell_{\rm f}+\ell_{\rm r}), \quad (13)$$

and that

$$R(\delta)\begin{bmatrix} s_{{\rm f}x} \\ s_{{\rm f}y} \end{bmatrix}\frac{\mu_{\rm f}}{s_{\rm f}} = -\begin{bmatrix} \sigma_1(\xi, s_{{\rm r}x}) \\ \sigma_2(\xi) \end{bmatrix}, \quad (14)$$

where $(\xi, s_{{\rm r}x}) \mapsto \sigma_1(\xi, s_{{\rm r}x})$ and $\xi \mapsto \sigma_2(\xi)$ are maps whose detailed expressions are provided in the Appendix. We may compute the friction coefficient $\mu_{\rm f}$ of the front tire by (14) as

$$\mu_{\rm f} = \sqrt{\sigma_1^2(\xi, s_{{\rm r}x}) + \sigma_2^2(\xi)}, \quad (15)$$

and the total slip $s_{\rm f}$ of the front tire may then be computed immediately from (9). After algebraic manipulations of (14)–(15), we may arrive at the following equation in δ:

$$-v_x s_{\rm f}\sigma_2(\xi)\cos^2\delta + (v_y + \ell_{\rm f}\dot{\psi})s_{\rm f}\sigma_1(\xi, s_{{\rm r}x})\sin^2\delta$$
$$+ (v_x\sigma_1(\xi, s_{{\rm r}x}) - (v_y + \ell_{\rm f}\dot{\psi})\sigma_2(\xi))s_{\rm f}\sin\delta\cos\delta$$
$$- (v_y + \ell_{\rm f}\dot{\psi})\mu_{\rm f}\cos\delta + v_x\mu_{\rm f}\sin\delta = 0. \quad (16)$$

Following an appropriate transformation of variables, (16) may be transformed to a quartic polynomial equation, which

may be solved analytically for δ. Finally, the longitudinal slip $s_{\mathrm{f}x}$ of the front tire may be computed using (4).

Fact 1: There exists at least one real root to (16) whenever u_{t} and u_{n} satisfy (21).

Proof: See the Appendix. ∎

Informally, Fact 1 states that whenever the commanded acceleration of the front CO satisfies the constraints (to be discussed in Section IV) imposed by the ground-tire friction force characteristics, there exists at least one solution to the system of nonlinear equations in (12). We reiterate that all computations involved in computing the vehicle control inputs $(s_{\mathrm{f}x}, s_{rx}, \delta)$ from the acceleration $\mathbf{u}$ of the CO (i.e., the second derivatives of the flat output) are simple, and that none of these computations involve any computationally expensive numerical optimization or root-finding.

III. The RRT* Algorithm

In this section, we describe a general optimal motion planning problem, and we describe briefly the RRT* algorithm for the solution of this problem with probabilistic guarantees. The reader interested is referred to [4] for further details.

Let $X \subset \mathbb{R}^n$ and $U \subset \mathbb{R}^m$ be compact sets, and consider a nonlinear dynamical system described by

$$\dot{\xi}(t) = f(\xi(t), u(t)), \quad \xi(0) = \xi_0, \tag{17}$$

where $\xi(t) \in X$ and $u(t) \in U$ for all $t \geqslant 0$, $f : X \times U \to \mathbb{R}^n$, and $\xi_0 \in X$ is a pre-specified initial state. A trajectory $\xi : [0, T] \to X$, where $T \geqslant 0$, is said to be *dynamically feasible* if it satisfies (17) with a piecewise continuous function u that takes values in U. We denote by $\mathcal{X}$ the set of all dynamically feasible trajectories. Let $X_{\mathrm{obs}}, X_{\mathrm{goal}} \subset X$ be open sets, called the obstacle set and the goal set, respectively. Define the free space as $X_{\mathrm{free}} := \mathbb{R}^n \backslash X_{\mathrm{obs}}$. Finally, let $J : \mathcal{X} \to \mathbb{R}_+$ be a cost functional that associates each dynamically feasible trajectory with a non-negative real number.

The optimal motion planning problem is then defined as: Find a feasible trajectory $\xi^* \in \mathcal{X}$ such that (i) $\xi^*(T) \in X_{\mathrm{goal}}$, for some $T \geqslant 0$, i.e., the trajectory reaches the goal set; (ii) $\xi^*(t) \notin X_{\mathrm{obs}}$ for all $t \in [0, T]$, i.e., the trajectory avoids collisions with the obstacles; and (iii) $J(\xi^*) \leqslant J(\xi)$ for all $\xi \in \mathcal{X}$, i.e., the trajectory has minimum cost.

The RRT* algorithm, described in Algorithm 1, constructs a tree of dynamically feasible trajectories of the system. We denote by V and E, respectively, the set of vertices and edges of this tree, where each vertex in V is a state of the dynamical system, and associated with each edge in E is a dynamically feasible trajectory.

Algorithm 1: The RRT* Algorithm

```
1  V ← {z_init}; E ← ∅; i ← 0;
2  while i < N do
3      G ← (V, E);
4      z_rand ← Sample(i); i ← i + 1;
5      (V, E) ← Extend(G, z_rand);
```

Initially V contains only the initial state ξ_0 and E is empty (Line 1). At each subsequent iteration, the algorithm samples a new state (Line 4), and extends the tree towards this state (Line 5). The extension procedure of the RRT* algorithm (Algorithm 2), first extends the state associated with the nearest vertex in the tree towards the newly sampled state (Lines 2-4) by constructing a dynamically feasible trajectory between these two states. If this trajectory lies entirely in the obstacle-free space (Line 5), then the algorithm adds a new vertex to the tree associated with the newly sampled state (Line 6). A set of vertices near-by to this new vertex is computed in Line 8, and the vertex belonging to this set from which the new vertex is reached with minimal cost is assigned as the parent to the new vertex (Lines 9-14). The cost of the given trajectory from the root to a vertex z is denoted as $\mathtt{Cost}(z)$.

Algorithm 2: The Extend Procedure

```
1  V' ← V; E' ← E;
2  z_nearest ← Nearest(G, z);
3  (ξ_new, u_new, T_new) ← Steer(z_nearest, z);
4  z_new ← ξ_new(T_new);
5  if ObstacleFree(ξ_new) then
6      V' ← V' ∪ {z_new};
7      z_min ← z_nearest; J_min ← Cost(z_new);
8      Z_near ← NearVertices(G, z_new, |V|);
9      for all z_near ∈ Z_near do
10         (ξ_near, u_near, T_near) ← Steer(z_near, z_new);
11         if ξ_near(T_near) = z_new and
           ObstacleFree(ξ_near) and
           Cost(z_near) + J(ξ_near) < J_min then
12             J_min ← Cost(z_near) + J(ξ_near);
13             z_min ← z_near;
14     E' ← E' ∪ {(z_min, z_new)};
15     for all z_near ∈ Z_near \ {z_min} do
16         (ξ_near, u_near, T_near) ← Steer(z_new, z_near);
17         if ξ_near(T_near) = z_near and
           ObstacleFree(ξ_near) and
           Cost(z_new) + J(ξ_near) < Cost(z_near) then
18             z_parent ← Parent(z_near);
19             E' ← E' \ {(z_parent, z_near)};
20             E' ← E' ∪ {(z_new, z_near)};
21 return G' = (V', E')
```

Whenever a new vertex is successfully added to the tree, the algorithm attempts to "rewire" the near-by vertices as follows. If there exists a dynamically feasible collision-free trajectory that connects the new vertex z_{new} and a near-by vertex with a smaller accumulated cost, then the parent of this vertex is re-assigned to z_{new} (Lines 15-20).

The various primitive procedures that are invoked repeatedly by the RRT* algorithm are described as follows.

1) Sampling: The `Sample` procedure returns independent and identically distributed samples from the state space.

2) Near and Nearest Neighbors: Given a finite set $V \subset X$ and a state $z \in X$, the $\mathtt{Nearest}(z, V)$ procedure returns the element $z' \in V$ that is closest to z among all elements of V, i.e., $\mathtt{Nearest}(z, V) := \arg\min\{\|z' - z\| : z' \in V\}$, where $\|\cdot\|$ denotes the Euclidean norm.

The $\mathtt{Near}(z, V)$ procedure returns the set of all elements

of V that are close to z in the following sense:

$$\texttt{Near}(z, V) := \left\{ z' \in V : \|z' - z\| \leq \gamma \frac{\log(|V|)}{|V|} \right\},$$

where $|V|$ denotes the cardinality of V and γ is a constant.

3) Local steering: Given two states $z, z' \in X$, the $\texttt{Steer}(z, z')$ procedure returns a dynamically feasible trajectory $\xi \in \mathcal{X}$, a time of traversal $T \geqslant 0$, and a control input $u : [0, T] \to U$ associated with ξ such that $\xi(0) = z$ and $\xi(T) = z'$.

4) Collision checking: Given a trajectory $x : [0, T] \to X$, the $\texttt{CollsiionFree}(x)$ procedure returns true iff x avoids collision with obstacles, i.e., $x(t) \in X_{\text{free}}$ for all $t \in [0, T]$.

The local steering problem (to be solved by the `Steer` procedure) may be implemented using standard numerical optimal control techniques. Such an implementation of the `Steer` procedure for the half-car dynamical model described in Section II was discussed in [18]. This approach, however, was found to be computationally expensive, and the overall motion planner, whereas promising, was found to be impractically slow in its execution. To mitigate this drawback, we describe, in the next section, a computationally efficient solution of the local steering problem for the half-car dynamical model.

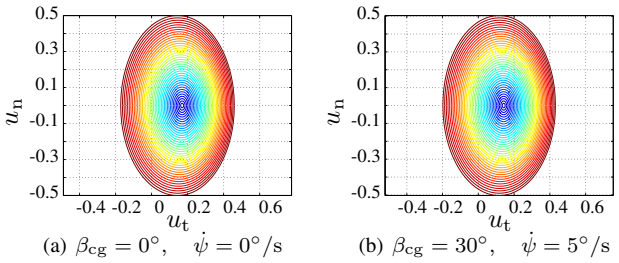
IV. Local Steering for the Half-Car Model

Our approach to the design of a fast local steering algorithm is summarized as follows. We leverage the "pseudo-flat" nature of the system to transform the steering problem for the half-car model to a steering problem for the simpler particle model $\ddot{\mathbf{p}}_{\text{co}} = \mathbf{u}$ that defines the motion of the CO. To this end, we map the constraints on the tire friction forces to equivalent constraints on the augmented input $\mathbf{u}$. These constraints on $\mathbf{u}$ impose bounds on the lateral acceleration of the CO, which in turn correspond to speed-dependent bounds on the curvatures of paths in the workspace that the CO can feasibly traverse. Next, we approximate a time-optimal local steering trajectory for the CO by first constructing a curvature-bounded geometric path and then by imposing a minimum-time speed profile on this path. Finally, we determine the acceleration $\mathbf{u}$ of the CO for tracking this trajectory, and we map $\mathbf{u}$ to the vehicle inputs $(s_{\text{f}x}, s_{\text{r}x}, \delta)$ using the computational procedure outlined in Section II-A.

A. Constraints on Pseudo-Flat Output Trajectories

The magnitude F_α, $\alpha \in \{\text{f}, \text{r}\}$, of the total friction force at each tire depends on the friction coefficient and the normal load on that tire: $F_\alpha = \mu_\alpha F_{\alpha z}$. It follows that $F_\alpha = \sqrt{F_{\alpha x}^2 + F_{\alpha y}^2} \leqslant \mu_\alpha^* F_{\alpha z}$, where μ_α^* is the maximum value of the tire friction coefficient (recall from Section II that μ_α depends on the total tire slip). The preceding circular constraint on the lateral and longitudinal forces at each tire, in turn, imposes constraints on the trajectory of the CO.

In what follows, we show that the acceleration $\mathbf{u}$ of the CO is constrained to lie within an ellipse, the dimensions of which depend on the vehicle state (in particular, the sideslip $\beta_{\text{cg}} := \tan^{-1}(v_y / v_x)$, and the yaw rate $\dot{\psi}$), the maximum value μ_{f}^* of the front tire friction coefficient, and the rear tire longitudinal slip $s_{\text{r}x}$, which was chosen in Section II-A as an exogenous input that may be chosen independent of the trajectory of the CO.



(a) $\beta_{\text{cg}} = 0°, \quad \dot{\psi} = 0°/\text{s}$ (b) $\beta_{\text{cg}} = 30°, \quad \dot{\psi} = 5°/\text{s}$

Fig. 2. Constraints on augmented inputs for different states, with the same rear longitudinal slip. Here $s_{\text{r}x} = -1$ for both cases. The vehicle speed is 10 m/s. The different ellipses correspond to different values of μ_{f} in (20), with increasing values of μ_{f} from blue to red.

To this end, define constants $k_1 := -h/(\ell_{\text{f}} + \ell_{\text{r}})$ and $k_2 := (\ell_{\text{f}} + \ell_{\text{r}})/\ell_{\text{r}}$, and define the map $\xi \mapsto \sigma_3(\xi) := k_1(\ell_{\text{co}}\dot{\psi}^2 - \ell_{\text{r}}/h)$. Note that, by (13),

$$F_{\text{f}z} = k_1 u_{\text{t}} + \sigma_3(\xi), \tag{18}$$

and that, by (12),

$$u_{\text{t}}^2 + \frac{u_{\text{n}}^2}{k_2^2} = (F_{\text{f}x}\cos\delta - F_{\text{f}y}\sin\delta + F_{\text{r}x} - \ell_{\text{co}}\dot{\psi}^2)^2 + (F_{\text{f}x}\sin\delta + F_{\text{f}y}\cos\delta)^2. \tag{19}$$

The R.H.S. of (19) involves the front and rear tire lateral and longitudinal forces, and we may use (5), (8), and (18) to express the R.H.S. of (19) in terms of ξ, $s_{\text{r}x}$, and μ_{f} to arrive at the following equation:

$$\left(\frac{u_{\text{t}} + \sigma_4(\xi, s_{\text{r}x}, \mu_{\text{f}})}{\sigma_5(\xi, s_{\text{r}x}, \mu_{\text{f}})}\right)^2 + \left(\frac{u_{\text{n}}}{\sigma_6(\xi, s_{\text{r}x}, \mu_{\text{f}})}\right)^2 = 1, \tag{20}$$

where $(\xi, s_{\text{r}x}, \mu_{\text{f}}) \mapsto \sigma_4(\xi, s_{\text{r}x}, \mu_{\text{f}})$, $\sigma_5(\xi, s_{\text{r}x}, \mu_{\text{f}})$, $\sigma_6(\xi, s_{\text{r}x}, \mu_{\text{f}})$ are maps whose detailed expressions are provided in the Appendix. Note that the values of these maps define the location of the center and the dimensions of an ellipse in the $u_{\text{t}} - u_{\text{n}}$ plane.

Figure 2 shows the ellipse defined by (20) for different sample values of the vehicle sideslip and yaw rate over a range of values of μ_{f}. It is straightforward to show that for any $\mu_{\text{f}_1}, \mu_{\text{f}_2}$ with $\mu_{\text{f}_1} \leqslant \mu_{\text{f}_2}$, the ellipse defined by (20) with $\mu_{\text{f}} = \mu_{\text{f}_1}$ is completely contained within the ellipse defined by (20) with $\mu_{\text{f}} = \mu_{\text{f}_2}$.

In light of the equation (20) and the preceding observation, the original circular constraints of the front and rear tire friction forces may be mapped to the following elliptical constraint on the acceleration $\mathbf{u}$ of the CO:

$$\left(\frac{u_{\text{t}} + \sigma_4(\xi, s_{\text{r}x}, \mu_{\text{f}}^*)}{\sigma_5(\xi, s_{\text{r}x}, \mu_{\text{f}}^*)}\right)^2 + \left(\frac{u_{\text{n}}}{\sigma_6(\xi, s_{\text{r}x}, \mu_{\text{f}}^*)}\right)^2 \leqslant 1. \tag{21}$$

Let β_{co} denote the sideslip of the CO, i.e., the angle between $\dot{\mathbf{p}}_{\text{co}}$ and the body x-axis. It can be shown that $\beta_{\text{co}} = \tan^{-1}\left((v_y + \ell_{\text{co}}\dot{\psi})/v_x\right)$. Then, the intersections of

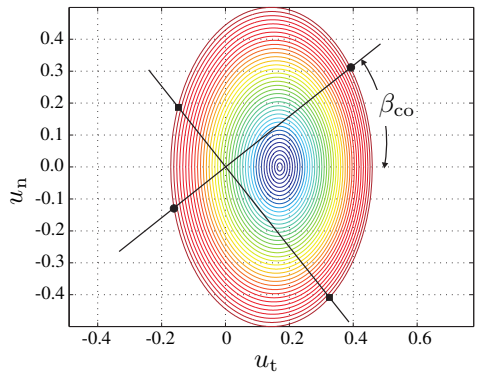


Fig. 3. Calculation of acceleration constraints on the CO from (21): The filled squares indicate bounds on pure lateral acceleration, and the filled circles indicate bounds on pure tangential acceleration.

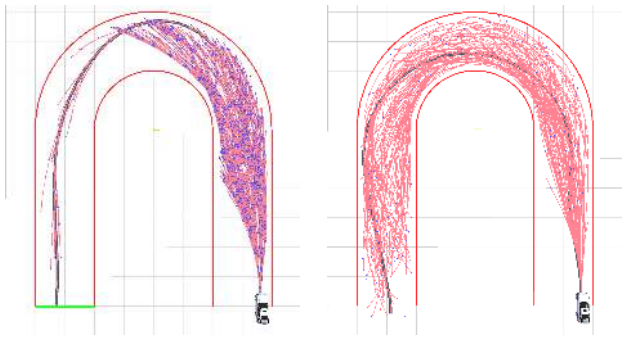
the line of inclination $\beta_{\rm co}$ passing through the origin of the $u_{\rm t} - u_{\rm n}$ plane with the boundary of the elliptical region described by (21) provide the upper and lower bounds on the pure tangential acceleration of the CO. Similarly, the intersections of the line of inclination $\beta_{\rm co}+\frac{\pi}{2}$ passing through the origin of the $u_{\rm t} - u_{\rm n}$ plane with the boundary of the elliptical region described by (21) provide bounds on the pure lateral acceleration of the CO (see Fig. 3).

Geometric paths traversed by a particle with bounded lateral acceleration have bounded curvature. Specifically, the curvature bounds for left and right turns, respectively, are $v^2/u_\perp^{\max}$ and $v^2/|u_\perp^{\min}|$, assuming[3] $u_\perp^{\max} > 0$ and $u_\perp^{\min} < 0$.

As previously mentioned, we approximate time-optimal trajectories for the CO by first constructing a shortest geometric path and then imposing on this path a minimum-time speed profile. The construction of the shortest curvature-bounded path between the initial and terminal states with asymmetric (about the origin) bounds on the curvature is straightforward: it has been shown in [20] show that the shortest path is contained within the Dubins family of paths. The Dubins family of paths consists of continuously differentiable paths that are obtained by concatenating at most three sub-paths, each of which is either a straight line segment or a circular arc of radius of equal to the minimum radius of turn (which is the reciprocal of the curvature bound).

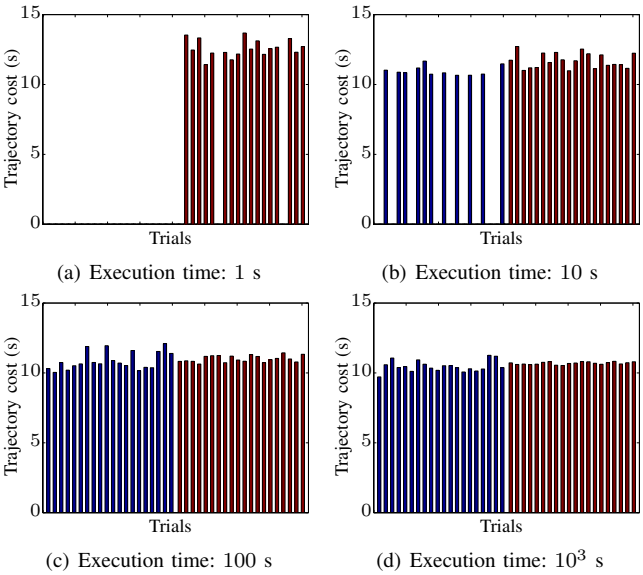
Similarly, the minimum-time speed profile on a prescribed curve for a particle with an elliptical acceleration constraint has been discussed in, for instance, [21]. Briefly, the approach in [21] involves determining switching points along the prescribed curve, such that, between two consecutive switches, the particle either travels with maximum possible tangential acceleration, or travels with maximum possible tangential deceleration, or travels at the critical speed. This critical speed is defined, pointwise along the prescribed curve, as the speed at which the centripetal acceleration required for the particle to change its direction of travel at the rate prescribed by the instantaneous curvature equals the maximum lateral acceleration of the particle. Here, the "maximum possible tangential acceleration/deceleration" refers to the tangential acceleration/deceleration that satisfies – together with the centripetal acceleration that corresponds to the change the particle's direction at the rate prescribed by the instantaneous curvature – the elliptical acceleration constraint as an equality.

[3]If both $u_\perp^{\max}$ and $u_\perp^{\min}$ are both positive (respectively, negative), then the particle can make only left (respectively, right) turns.



(a) From [18] (b) Proposed

Fig. 4. Comparison of coverages achieved within a specified computation time by different implementations of RRT* for the half-car model.



(a) Execution time: 1 s (b) Execution time: 10 s

(c) Execution time: 100 s (d) Execution time: 10^3 s

Fig. 5. Improvements in trajectory costs with increasing computation times, for the 180-degree turn illustrated in Fig. 4. The blue-colored bars indicate data for the motion planner in [18]; brown-colored bars indicate data for the proposed planner. Empty spaces indicate that no feasible trajectory was found within the stated execution time.

V. Simulation Results and Discussion

The proposed motion planner is fast, and its speed of execution makes it suitable for real-time implementations. To corroborate this claim, we present sample numerical simulation results for the proposed motion planner, including simulations with hard upper bounds on the execution time.

As previously mentioned, a preliminary implementation of RRT* for the half-car dynamical model was discussed in [18], where a numerical optimal control technique was used to implement the local steering algorithm. Figure 4 illustrates

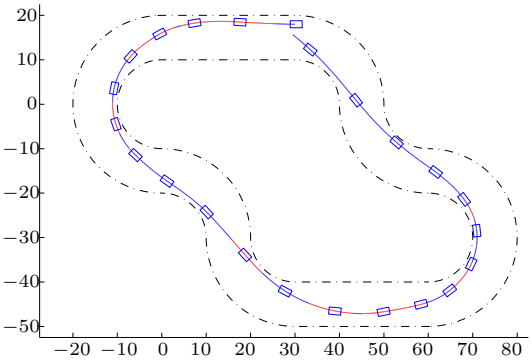
a sample simulation result comparing the coverage of the state space achieved by the RRT* algorithm implemented using the local steering method of [18] against that using the proposed method. Both of these algorithms were executed for a fixed period of time (20 s on a Intel® Core™2 Extreme Q9300, 2.53Ghz processor with 4GB RAM). As expected, the proposed implementation of RRT* achieved significantly better coverage than that discussed in [18].

Figure 5 shows comparisons of trajectory costs (i.e., the time of traversal) achieved by these two implementations within specified execution times. The data shown in Fig. 5 was obtained over 20 trials of each implementation for the problem of planning the 180-degree turn illustrated in Fig. 4. In particular, Fig. 5(a) shows that, within a specified execution time of 1 s on the aforesaid computer, no feasible trajectory was found in any of the trials of the implementation of [18], whereas feasible trajectories were found in all but two trials of the proposed implementation. In these simulations, the ratio of the average time required to find a first feasible solution with the implementation of [18] to that with the proposed implementation was 21.23. However, the ratio of the maximum time required to find a first feasible solution with the implementation of [18] to the minimum time required with the proposed implementation was 226.3.

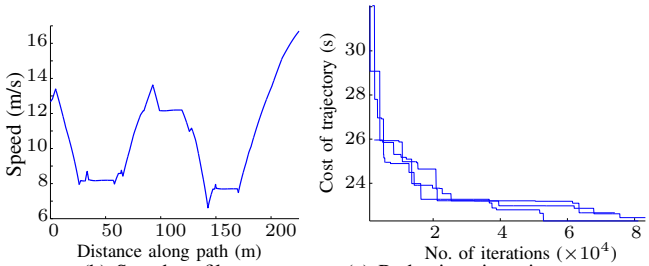
As discussed in Section II-A, the "pseudo-differentially-flat" feature of the half-car model requires one of the three vehicle inputs $(s_{\mathrm{f}x}, s_{\mathrm{r}x}, \delta)$ to be chosen independently of the flat output trajectory. For analytical simplicity, we chose the rear tire longitudinal slip $s_{\mathrm{r}x}$ as this independent input, and we set it to a constant value $s_{\mathrm{r}x,0}$. Consequently, the proposed implementation of RRT* converges asymptotically to an optimal control input within the class of admissible control inputs defined by $s_{\mathrm{r}x} = s_{\mathrm{r}x,0}$. On the other hand, the implementation of RRT* discussed in [18] converges asymptotically to a globally optimal control input, albeit at the cost of slower execution.

The speed of execution of the proposed motion planner enabled the solution of problems that were found to be impractically slow with the approach discussed in [18]. For example, Fig. 6 illustrates the application of the proposed approach to motion planning with the half-car model on a closed circuit, similar to a race track. Figure 6(a) illustrates the geometric path corresponding to a sample resultant trajectory, along with the vehicle's orientation (to indicate sideslip). Figure 6(b) shows the speed profile over the sample resultant trajectory, and Fig. 6(c) shows the decreases in resultant trajectory cost with the progress of the algorithm, for three different trials.

To further leverage the speed of execution of the proposed approach and to anticipate future real-time implementations with hard bounds on the execution time, we implemented the solution of the closed circuit motion planning problem using a receding-horizon approach. In this approach, the motion planner first computes, within a pre-specified computation time t_{comp}, a trajectory over a pre-specified horizon of length along the circuit. Next, the vehicle's motion is simulated for a pre-specified execution time and the process is repeated.



(a) Geometric path and vehicle orientation: the red-colored segments indicate braking; the blue-colored segments indicate acceleration.



(b) Speed profile (c) Reductions in trajectory cost

Fig. 6. Motion planning with the half-car model over a closed circuit.

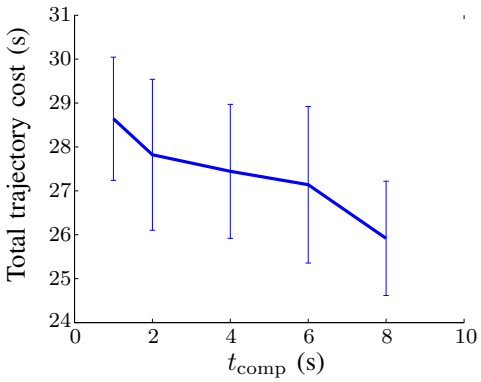


Fig. 7. Total trajectory costs using a receding-horizon approach to motion planning over the closed circuit shown in Fig. 6(a).

Figure 7 shows the total trajectory cost obtained by the aforesaid receding-horizon planner, over a range of values of t_{comp}. The total trajectory costs thus obtained are comparable to the trajectory costs shown in Fig. 6. A similar receding-horizon planner using the implementation of [18] was unable to find feasible trajectories for any of the values of t_{comp} shown in Fig. 7.

VI. Conclusions and Future Work

In this paper, we discussed a fast motion planner that incorporates the half-car dynamical model for wheeled vehicles. The proposed motion planner is based on the RRT* optimal motion planning algorithm, and the key to an efficient implementation of RRT* for the half-car model is a fast local

steering algorithm that we introduced here. The constituent algorithms involved in the proposed local steering method—namely, algorithms for the computation of a curvature-bounded geometric path, for the imposition of a minimum-time speed profile, and for mapping $\mathbf{u}$ to $(s_{\mathrm{f}x}, s_{\mathrm{r}x}, \delta)$—are all fast. Crucially, the proposed method for local steering, *by construction*, results in trajectories that are dynamically feasible. In particular, the mapping of the constraints on the tire friction forces to constraints on $\mathbf{u}$ saves significant computational effort in verifying *a posteriori* whether a candidate trajectory can be feasibly traversed by the vehicle. Finally, the proposed approach enables motion planning in the four-dimensional space of the coordinates of $\mathbf{p}_{\mathrm{co}}$ and $\dot{\mathbf{p}}_{\mathrm{co}}$, instead of the six-dimensional state space. Future work includes appropriate selection of the exogenous control input to better approximate the globally optimal control input.

Acknowledgements: This research was supported in part by ARO MURI award W911NF-11-1-0046.

APPENDIX

The expressions for the maps $\sigma_1, \ldots, \sigma_6$ in Sections II-A and IV-A are as follows:

$$\begin{aligned}
\sigma_1(\xi, s_{\mathrm{r}x}) &:= \frac{1}{h}\left(\frac{\ell_{\mathrm{r}} - h\mu_{\mathrm{r}x}}{F_{\mathrm{f}z}} - (\ell_{\mathrm{f}} + \ell_{\mathrm{r}})\right) + \mu_{\mathrm{r}x},\\
\sigma_2(\xi) &:= \ell_{\mathrm{r}} u_{\mathrm{n}}/(\ell_{\mathrm{f}} + \ell_{\mathrm{r}})/F_{\mathrm{f}z},\\
\sigma_3(\xi) &:= k_1(\ell_{\mathrm{co}}\dot{\psi}^2 - \ell_{\mathrm{r}}/h),\\
\sigma_4(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}}) &:=\\
&\Big((\mu_{\mathrm{r}x}k_1 + 1)(\ell_{\mathrm{co}}\dot{\psi}^2 + \mu_{\mathrm{r}x}(\sigma_3 - 1)) - \mu_{\mathrm{f}}^2 k_1 \sigma_3\Big)/\sigma_7^2,\\
\sigma_6(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}}) &:=\\
&k_2\sqrt{\mu_{\mathrm{f}}^2\sigma_3^2 - (\ell_{\mathrm{co}}\dot{\psi}^2 + \mu_{\mathrm{r}x}(\sigma_3 - 1))^2 + (\sigma_4\sigma_7)^2},\\
\sigma_5(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}}) &:= \sigma_6/(k_2\sigma_7),
\end{aligned}$$

where

$$(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}}) \mapsto \sigma_7(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}}) := \sqrt{(\mu_{\mathrm{r}x}k_1 + 1)^2 - \mu_{\mathrm{f}}^2 k_1^2}.$$

Proof: [of Fact 1] The algebraic manipulations involved in arriving at (16) from (14)-(15) include the equation

$$s_{\mathrm{f}x}^2 + s_{\mathrm{f}y}^2(\xi, \delta) = \frac{1}{B_{\mathrm{f}}^2}\tan^2\left(\frac{\sin^{-1}(\mu_{\mathrm{f}}/D_{\mathrm{f}})}{C_{\mathrm{f}}}\right),$$

and it follows that

$$(v_y + \ell_{\mathrm{f}}\dot{\psi} \mp \sigma_8 v_x)\cos\delta + (\mp\sigma_8(v_y + \ell_{\mathrm{f}}\dot{\psi}) - v_x)\sin\delta = 0, \tag{22}$$

where $\sigma_8 := \sqrt{\frac{1}{B_{\mathrm{f}}^2}\tan^2\left(\frac{\sin^{-1}(\mu_{\mathrm{f}}/D_{\mathrm{f}})}{C_{\mathrm{f}}}\right) - s_{\mathrm{f}x}^2}$. Equation (16) is obtained from (22) by eliminating $s_{\mathrm{f}x}$ from (22) using (14), and, consequently, the existence of real roots of (16) is equivalent to the existence of real roots of (22). It easy to show that, following an appropriate transformation of variables, that (22) is equivalent to a quadratic equation that has real roots whenever

$$(v_y + \ell_{\mathrm{f}}\dot{\psi} \mp \sigma_8 v_x)^2 + (\mp\sigma_8(v_y + \ell_{\mathrm{f}}\dot{\psi}) - v_x)^2 > 0,$$

which holds true whenever σ_8 is real, which in turn holds true whenever $|\mu_{\mathrm{f}}| \leqslant D_{\mathrm{f}} = \mu_{\mathrm{f}}^*$. By (15), and by the arguments leading to the constraint (21), it follows that $0 \leqslant \mu_{\mathrm{f}} \leqslant \mu_{\mathrm{f}}^*$ whenever u_{t} and u_{n} satisfy (21), and the result follows. ■

REFERENCES

[1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press, 2005.

[2] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[3] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[4] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, pp. 846–894, 2011.

[5] R. V. Cowlagi and P. Tsiotras, "Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles," *IEEE Transactions on Robotics*, 2011, to appear.

[6] M. Fliess, J. Levine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 71, no. 5, pp. 745–765, 1995.

[7] W. F. Milliken and D. L. Milliken, *Race Car Vehicle Dynamics*. Warrendale, PA, USA: SAE International, 1995.

[8] M. Schorn, U. Stahlin, A. Khanafer, and R. Isermann, "Nonlinear trajectory following control for automatic steering of a collision avoiding vehicle," in *Proceedings of the 2006 American Control Conference*, 2006.

[9] J. Ackermann, "Robust decoupling, ideal steering dynamics and yaw stabilization of 4WS cars," *Automatica*, vol. 30, no. 11, pp. 1761–1768, 1994.

[10] S. Fuchshumer, K. Schlacher, and T. Rittenschober, "Nonlinear vehicle dynamics and control – a flatness based approach," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, December 12–15 2005, pp. 6492–6497.

[11] S. C. Peters, "Optimal planning and control for hazard avoidance of front-steered ground vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.

[12] P. Setlur, J. Wagner, D. Dawson, and D. Braganza, "A trajectory tracking steer-by-wire control system for ground vehicles," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 76–85, 2006.

[13] T. Hiraoka, O. Nishihara, and H. Kumamoto, "Automatic path-tracking controller of a four-wheel steering vehicle," *Vehicle System Dynamics*, vol. 47, no. 10, pp. 1205–1227, 2009.

[14] K. Kritayakirana and J. C. Gerdes, "Using the center of percussion to generate feedforward steering for an autonomous race car," in *Proceedings of the 2011 IAVSD Symposium*. IAVSD, 2011.

[15] E. Velenis, P. Tsiotras, and J. Lu, "Optimality properties and driver input parametrization for trail-braking cornering," *European Journal of Control*, vol. 4, pp. 308–320, 2008.

[16] D. Casanova, R. S. Sharp, and P. Symonds, "Minimum time manoeuvering: The significance of yaw inertia," *Vehicle System Dynamics*, vol. 34, no. 2, pp. 77–115, 2000.

[17] E. Velenis and P. Tsiotras, "Optimal velocity profile generation for given acceleration limits: The half-car case," in *Proceedings of the IEEE International Symposium on Industrial Electronics ISIE 2005*, Dubrovnik, Croatia, June 20–23 2005, pp. 361–366.

[18] J. Jeon, S. Karaman, and E. Frazzoli, "Anytime computation of time-optimal off-road vehicle maneuvers using the RRT*," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, Dec. 12 – 15 2011.

[19] E. Bakker, L. Nyborg, and H. Pacejka, "Tyre modeling for use in vehicle dynamics studies," SAE Paper No. 870421, 1987.

[20] E. Bakolas and P. Tsiotras, "Optimal synthesis of the asymmetric sinistral/dextral Markov-Dubins problem," *Journal of Optimization Theory and Applications*, vol. 150, no. 2, 2011.

[21] E. Velenis and P. Tsiotras, "Minimum-time travel for a vehicle with acceleration limits: Theoretical analysis and receding horizon implementation," *Journal of Optimization Theory and Applications*, vol. 138, no. 2, pp. 275–296, 2008.