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 discuss an implementation of the RRT* optimal motion planning algorithm for the half-car dynamical model to enable autonomous high-speed driving. To develop fast solutions of the associated local steering problem, we observe that the motion of a special point (namely, 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, and to the computational efficiency afforded by these models. However, the resultant trajectories do not exploit the vehicle's maneuvering capabilities, and motion planning based on such vehicle models is unsuitable for enabling autonomous high-speed motion of car-like vehicles in complex and dynamic environments.

Motion planning for higher-dimensional, higher-fidelity vehicle dynamical models is, in general, difficult. Whereas computationally efficient motion planning in high-dimensional state spaces is made possible via 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 higher-fidelity vehicle dynamical models. Both of these approaches, however, rely on the availability of a *local steering* algorithm – a two-point boundary value (TPBV) problem solver – that computes a near-optimal control input to steer the vehicle between specified initial and final states, neither of which is necessarily an equilibrium state.

*Massachusetts Institute of Technology, Cambridge, MA.
**Aurora Flight Sciences Corp, Cambridge, MA.
†Open Source Robotics Foundation, Mountain View, CA.
††Georgia Institute of Technology, Atlanta, GA.

The main contributions of this paper are a fast local steering algorithm specific to the half-car dynamical model [6], and the implementation of the RRT* motion planning algorithm using this local steering algorithm. The half-car model of wheeled vehicles captures yaw dynamics and normal load transfer, and it includes acceleration constraints in the form of constraints on the ground-tire friction forces.

A. Motivation and Related Work

Whereas the solution of the local steering problem is difficult in general, in some specific cases it is possible to exploit the structure of the dynamical system to develop a fast solution algorithm suitable for real-time implementations. In particular, the property of *differential flatness* [7] 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).

In the context of motion planning, differential flatness of the vehicle dynamical model is particularly useful when the flat outputs are workspace[1] coordinates of a point associated with the vehicle. In such cases, the vehicle dynamical behavior may be inferred from workspace trajectories of this point. Conversely, the problem of motion planning subject to vehicle dynamical constraints may be transformed to a workspace trajectory generation problem, which is beneficial owing to the low dimensionality of the workspace and to the ease of collision checking with (workspace) obstacles.

For a half-car model of front-steered vehicles that incorporates wheel slip, special points associated with the model, called the *front and rear Huygens centers of oscillation* (CO) are associated with differential flatness. A crucial physical property of a CO is that its acceleration is independent of one of the lateral tire friction forces [8]. 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 normal load transfer [9].

More pertinent to 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 [10]. 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. The position coordinates of front and/or rear CO have been considered as reference inputs for trajectory tracking controllers in [11] (rear CO), in [12] (both front and rear CO with four-wheel steering), and

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

in [13] (front CO). The notion of "pseudo-flatness" is closely related to the notion of *near-identity diffeomorphisms* [14], wherein stabilization and tracking of non-holonomic systems are discussed. The crucial difference between the proposed work and [14] is that we address the more involved TPBV problem of optimal trajectory generation. In this context, numerical optimal control techniques have been applied in [15] for reproducing a 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 problem was solved numerically in a lower dimensional state space of the vehicle.

B. Contributions

The main contributions of this work are as follows. Firstly, we provide a fast 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*). The key idea that enables this fast local steering is its separation into a geometric path planning step and an optimal time parametrization step, while always maintaining guarantees of feasibility vis-a-vis the dynamical constraints and input constraints of the half-car model. This separation provides a significant advantage in the implementation of RRT*: rough collision checking can be performed after the geometric path planning step, and the computationally intensive optimal time-parametrization step can be skipped for cases where collisions are detected. The proposed local steering algorithm and the resulting RRT*-based motion planner constitute vast improvements in computation time over the implementation of [18], and enables the solutions of problems that were found to be impractically slow to solve using the approach of [18].

Secondly, we drop the simplifications to the half-car model adopted in [9], [10]: specifically, we allow *normal load transfer* between the front and rear tires – a phenomenon commonly utilized by rally racing drivers to control the yaw dynamics [15] – and we consider as inputs the longitudinal tire slips instead of the longitudinal tire forces. 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.

Thirdly, we map the constraints on tire friction forces to equivalent constraints on the flat output trajectory. This mapping is itself a novelty in the context of differential flatness-based trajectory generation and control algorithms for the half-car, because friction force constraints are ignored in similar earlier works [9], [10].

Finally, we study an implementation of RRT* with the proposed local steering algorithm, which is a fundamental way to develop an autonomous high-speed driving system that fully utilizes the vehicle's maneuvering capabilities.

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 discuss an efficient local steering algorithm for the half-car model. In Section IV, we provide simulation results of the said implementation of RRT*. Finally, we conclude the paper in Section V 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. [6], [9], [10], [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 of the velocity $\mathbf{v}$ of the CG. We denote by $\ell_{\rm f}$ and $\ell_{\rm r}$, respectively, the distances of the centers to 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 of the half-car model are:

$$\begin{aligned} m\dot{v}_x &= (F_{{\rm f}x}\cos\delta - F_{{\rm f}y}\sin\delta + F_{{\rm r}x}) + mv_y\dot{\psi}, \quad (1)\\ m\dot{v}_y &= (F_{{\rm f}x}\sin\delta + F_{{\rm f}y}\cos\delta + F_{{\rm r}y}) - mv_x\dot{\psi}, \quad (2)\\ I_z\ddot{\psi} &= \ell_{\rm f}\left(F_{{\rm f}x}\sin\delta + F_{{\rm f}y}\cos\delta\right) - \ell_{\rm r}F_{{\rm r}y}, \quad (3)\end{aligned}$$

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., $\xi = (p_{{\rm cg},x}, p_{{\rm cg},y}, \psi, v_x, v_y, \dot{\psi})$. The lateral slips $s_{{\rm f}y}$ and $s_{{\rm r}y}$ of the front and rear tires are:

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

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

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

where $F_{\alpha z}$ are the normal tire loads given by (cf. [15]):

$$\begin{aligned} F_{{\rm f}z} &= \frac{mg\left(\ell_{\rm r} - \mu_{{\rm r}x}h\right)}{\ell_{\rm f} + \ell_{\rm r} + h\left(\mu_{{\rm f}x}\cos\delta - \mu_{{\rm f}y}\sin\delta - \mu_{{\rm r}x}\right)}, \quad (6)\\ F_{{\rm r}z} &= mg - F_{{\rm f}z}, \quad (7)\end{aligned}$$

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

$$\begin{aligned} \mu_{\alpha\beta} &:= -(s_{\alpha\beta}/s_\alpha)\mu_\alpha \quad (8)\\ \text{with}\quad \mu_\alpha &= D_\alpha\sin\left(C_\alpha\tan^{-1}\left(B_\alpha s_\alpha\right)\right), \quad (9)\end{aligned}$$

for $\alpha \in \{{\rm f},{\rm r}\}$, $\beta \in \{x,y\}$, 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.

Following the work of Peters *et al* [10], we consider as a candidate flat output the position $\mathbf{p}_{\rm co}$ of the front CO [8], which is a point defined by

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

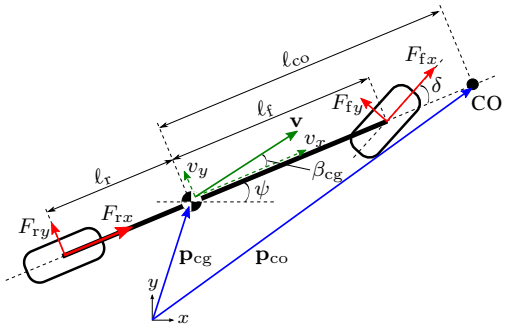


Fig. 1. The half-car dynamical model: position vectors are in blue, velocity vectors are in green, and forces are in red color.

where $\ell_{\rm co} := I_z/m\ell_{\rm r}$ and $R(\psi)$ is the rotation matrix. It may be shown (cf. [10]) that

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

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),

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

Firstly, observe that (12) is an under-determined system of equations, 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 $F_{\alpha\beta}$, $\alpha \in \{{\rm f},{\rm r}\}, \beta \in \{x, y\}$ depend on $s_{{\rm f}x}, s_{{\rm r}x}$ via (4)-(9).) However, we may treat one of the three inputs as an "exogenous" input, and subsequently determine the other two inputs.

Secondly, observe that the computation of ψ and $\dot{\psi}$ involves the solution of the ODE (3). Consequently, the mapping from $\mathbf{p}_{\rm co}$ and its derivatives to the vehicle control inputs is a system of coupled algebraic-differential equations, and $\mathbf{p}_{\rm co}$ may thus be considered a "pseudo-flat" output. 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.

Equations (5)–(8) may be manipulated to show that

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

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

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

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

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

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

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 $s_{{\rm f}x}$ of the front tire may be computed using (4).

Fact 1: There exists at least one real root to (16) whenever $u_{\rm t}$ and $u_{\rm n}$ satisfy (18).

Proof: Omitted for the sake of brevity. ■

Informally, Fact 1 states that whenever the commanded acceleration of the front CO satisfies the constraints (to be discussed in Section III) 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 $(s_{{\rm f}x}, s_{{\rm r}x}, \delta)$ 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. Local Steering for the Half-Car Model

Informally, the problem of optimal motion planning involves the determination of admissible control inputs for a nonlinear dynamical system such that (a) the state of the system is transferred from a pre-specified initial state ξ_0 to a pre-specified final state $\xi_{\rm f}$, (b) the resultant state trajectory does not intersect with a pre-specified subset of the state space, called the *obstacle space*, and (c) a pre-specified integral cost is minimized along the resultant state trajectory.

The RRT* algorithm solves the optimal motion planning problem (see [4] for details) by constructing a tree of state trajectories of the system. Each vertex of this tree is associated with a state of the nonlinear dynamical system, and each edge is associated with an admissible control input. Initially the aforesaid tree contains only one vertex associated with the initial state ξ_0. At each subsequent iteration, the RRT* algorithm samples a new state, extends the tree towards this state, and attempts to reassign the parent of each nearby vertex. The edge construction between vertices is achieved by a *local steering algorithm* (which we denote by STEER).

We note the following key issues [4]: (a) the number of times that RRT* invokes STEER is $O(n\log n)$; and (b) the state trajectory that corresponds to the control input found by STEER is subject to a further collision check to be included in the collision-free tree. It follows that a fast STEER is crucial to the speed of the overall motion planner;

furthermore, the computation time expended for instances of STEER that fail the subsequent collision check may be counted as "wasted" time, because these execution instances do not further advance the motion planner.

To design a fast STEER, 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 $\ddot{\mathbf{p}}_{\mathrm{co}} = \mathbf{u}$ that describes 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 curvature bounds on paths that the CO can feasibly traverse. Next, we approximate a time-optimal 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 using the computational procedure outlined in Section II.

As we will discuss in Section IV, the proposed approach for STEER is faster than a numerical optimal control-based approach. Furthermore, it also enables significant savings of the aforementioned "wasted" computation time by allowing collision checks to be performed after the (fast) geometric path planning step and before the (relatively slow) optimal time parametrization step.

A. Constraints on Pseudo-Flat Output Trajectories

The magnitude F_α, $\alpha \in \{\mathrm{f}, \mathrm{r}\}$, 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. 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 $\beta_{\mathrm{cg}} := \tan^{-1}(v_y/v_x)$, and the yaw rate $\dot{\psi}$), the maximum value μ_{f}^* of the front tire friction coefficient, and the rear tire longitudinal slip $s_{\mathrm{r}x}$, which was chosen in Section II as an "exogenous" input.

To this end, let $k_1 := -h/(\ell_{\mathrm{f}} + \ell_{\mathrm{r}})$, $k_2 := (\ell_{\mathrm{f}} + \ell_{\mathrm{r}})/\ell_{\mathrm{r}}$, and define the map $\sigma_3(\xi) := k_1(\ell_{\mathrm{co}}\dot{\psi}^2 - \ell_{\mathrm{r}}/h)$. Note that, by (13), $F_{\mathrm{f}z} = k_1 u_{\mathrm{t}} + \sigma_3(\xi)$. Following algebraic manipulations of (5), (8), and (12), we may then arrive at

$$\left(\frac{u_{\mathrm{t}} + \sigma_4(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}})}{\sigma_5(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}})}\right)^2 + \left(\frac{u_{\mathrm{n}}}{\sigma_6(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}})}\right)^2 = 1, \quad (17)$$

where σ_4, σ_5, and σ_6 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 $u_{\mathrm{t}} - u_{\mathrm{n}}$ plane. The constraints on the individual tire friction forces may now be mapped to the following elliptical constraint[2] on the acceleration of the CO:

$$\left(\frac{u_{\mathrm{t}} + \sigma_4(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}}^*)}{\sigma_5(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}}^*)}\right)^2 + \left(\frac{u_{\mathrm{n}}}{\sigma_6(\xi, s_{\mathrm{r}x}, \mu_{\mathrm{f}}^*)}\right)^2 \leqslant 1. \quad (18)$$

[2]It is straightforward to show that for any $\mu_{\mathrm{f}_1}, \mu_{\mathrm{f}_2}$ with $\mu_{\mathrm{f}_1} \leqslant \mu_{\mathrm{f}_2}$, the ellipse defined by (17) with $\mu_{\mathrm{f}} = \mu_{\mathrm{f}_1}$ is completely contained within the ellipse defined by (17) with $\mu_{\mathrm{f}} = \mu_{\mathrm{f}_2}$.

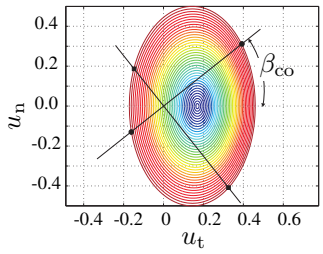


Fig. 2. Calculation of acceleration constraints on the CO from (18).

Let β_{co} denote the sideslip of the CO, i.e., the angle between $\dot{\mathbf{p}}_{\mathrm{co}}$ and the body x-axis. It can be shown that $\beta_{\mathrm{co}} = \tan^{-1}\left((v_y + \ell_{\mathrm{co}}\dot{\psi})/v_x\right)$. Then, the intersections of the line of inclination β_{co} passing through the origin of the $u_{\mathrm{t}} - u_{\mathrm{n}}$ plane with the boundary of the elliptical region described by (18) provide the upper and lower bounds on the pure tangential acceleration of the CO. Similarly, the intersections of the line of inclination $\beta_{\mathrm{co}} + \frac{\pi}{2}$ passing through the origin of the $u_{\mathrm{t}} - u_{\mathrm{n}}$ plane with the boundary of the elliptical region described by (18) provide bounds on the pure lateral acceleration of the CO (see Fig. 2).

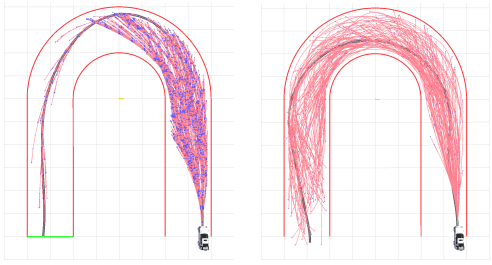
Geometric paths traversed by a particle with bounded lateral acceleration have bounded curvature. Specifically, the curvature bounds for left and right turns, respectively, are $v^2/u_\perp^{\max}$ and $v^2/|u_\perp^{\min}|$, assuming $u_\perp^{\max} > 0$ and $u_\perp^{\min} < 0$. The construction of the shortest curvature-bounded path with asymmetric (about the origin) bounds on the curvature is straightforward: it has been shown in [20] 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 subpaths, each of which is either a straight line segment or a circular arc of radius of equal to the minimum radius of turn.

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.

IV. SIMULATION RESULTS AND DISCUSSION

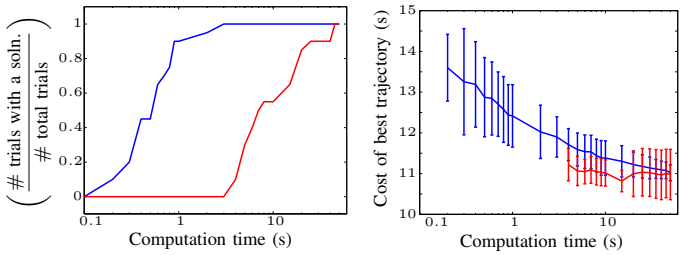
The proposed motion planner is fast, and its speed of execution makes it suitable for real-time implementations. To corroborate this claim, we present sample numerical simulation results for the proposed motion planner, including simulations with hard upper bounds on the execution time.

A preliminary implementation of RRT* for the half-car dynamical model was discussed in [18], where numerical methods were used in a reduced dimensional state space



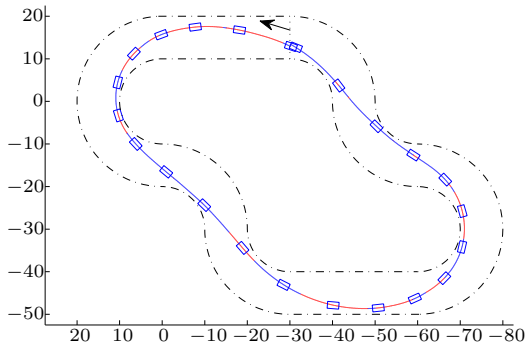
(a) From [18] (b) Proposed

Fig. 3. Trajectories obtained within a specified computation time when RRT* uses two different implementations of STEER for the half-car model.

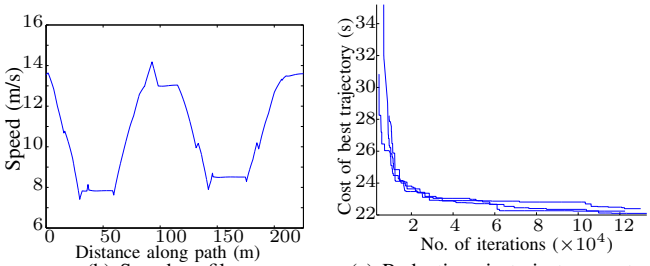


(a) Ratio of finding at least one solution over 20 trials within time. (b) Mean and standard deviation over 20 trials of the least known cost.

Fig. 4. Execution speed and resultant trajectory costs of the RRT* motion planner with the proposed STEER (blue) compared to that in [18] (red).



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



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

Fig. 5. Motion planning with the half-car model over a closed circuit.

to implement STEER. Fig. 3 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. As expected, the proposed implementation of RRT* achieved significantly better coverage than that discussed in [18].

The data in Fig. 4 was obtained over 20 trials for the problem of planning the 180-degree turn in Fig. 3. In particular, Fig. 4(a) shows that, within an execution time of 1s, 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. Fig. 4(b) shows the statistics for the costs of best trajectories achieved by the two implementations within specified execution times.

As discussed in Section II, we chose the rear tire longitudinal slip s_{rx} as an "exogenous" input, and we set it to a constant value $s_{rx,0}$. Consequently, the proposed implementation of RRT* converges asymptotically to an optimal control input within the class of admissible control inputs with $s_{rx} = s_{rx,0}$. In comparison, the implementation of RRT* in [18] converges asymptotically to a globally optimal control input (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 in [18]. For example, Fig. 5 illustrates the application of the proposed approach to motion planning on a closed circuit, similar to a race track. Fig. 5(a) illustrates the geometric path corresponding to a sample resultant trajectory, along with the vehicle's orientation (to indicate sideslip). Fig. 5(b) shows the speed profile over the sample resultant trajectory, and Fig. 5(c) shows the decreases in resultant trajectory cost with the progress of the algorithm, for three different trials.

As previously mentioned, the proposed STEER allows for a primary collision check (for the position of the front CO) to be performed immediately after the geometric path planning step. A secondary collision check (considering the finite size of the half-car and the heading angle) can be performed after the time-parametrization step. The computational advantage of this two-step collision check is that a large number potential collisions can be detected *before* the relatively slow time-parametrization computations. For the particular case of 180-degree turn illustrated in Fig. 3 over 20 trials, $98.76 \pm 0.05\%$ of the detected collisions were found immediately after the geometric path planning step. For the closed circuit case in Fig. 5, the ratio is higher as $99.02 \pm 0.08\%$ due to the challenging geometry of the obstacle-free space.

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

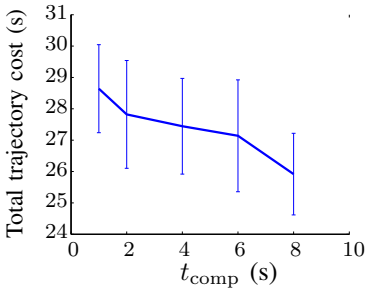


Fig. 6. Total trajectory costs using a receding-horizon approach to motion planning over the closed circuit shown in Fig. 5(a).

along the circuit. Next, the vehicle's motion is simulated for a pre-specified execution time and the process is repeated. Figure 6 shows the total trajectory cost obtained by the aforesaid receding-horizon planner, over a range of values of $t_{\rm comp}$. The total trajectory costs thus obtained are comparable to the trajectory costs shown in Fig. 5. A similar planner using the implementation of [18] was unable to find feasible trajectories for any of these values of $t_{\rm comp}$. The value $t_{\rm comp}$ implies $t_{\rm comp}$ seconds of computation on a Intel® Core™2 Extreme Q9300, 2.53GHz processor with 4GB RAM.

V. 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, the computation of a curvature-bounded geometric path, the imposition of a minimum-time speed profile, and the mapping of $\mathbf{u}$ to $(s_{{\rm f}x}, s_{{\rm r}x}, \delta)$—are all fast. Crucially, the proposed method for local steering, *by construction*, results in trajectories that are dynamically feasible and satisfy input constraints. Finally, the proposed approach enables motion planning in the space of the coordinates of $\mathbf{p}_{\rm co}$ and $\dot{\mathbf{p}}_{\rm co}$, instead of the full state space. Future work includes 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 and III-A are as follows:

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

$$\begin{aligned}
&\sigma_6(\xi, s_{{\rm r}x}, \mu_{\rm f}) :=\\
&\quad k_2\sqrt{\mu_{\rm f}^2\sigma_3^2 - (\ell_{\rm co}\dot{\psi}^2 + \mu_{{\rm r}x}(\sigma_3 - 1))^2 + (\sigma_4\sigma_7)^2},
\end{aligned}$$

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

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*, vol. 28, no. 2, pp. 379–395, 2012.

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

[7] 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.

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

[9] 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.

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

[11] 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, January 2006.

[12] 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.

[13] 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.

[14] R. Olfati-Saber, "Near identity diffeomorphisms and exponential ϵ-tracking and ϵ-stabilization of first order non-holonomic SE(2) vehicles," in *American Control Conference*, Anchorage, AK, USA, May 8 –10 2002, pp. 4690–4695.

[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 *IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, December 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.