

An Efficient Minimum-Time Trajectory Generation Strategy for Two-Track Car Vehicles

Alessandro Rucco, Giuseppe Notarstefano, and John Hauser

Abstract—In this paper, we propose a novel approach to compute minimum-time trajectories for a two-track car model, including tires and (quasi-static) longitudinal and lateral load transfer. Given the car model and a planar track, including lane boundaries, our goal is to find a trajectory of the car minimizing the traveling time subject to steering and tire limits. Moreover, we enforce normal force constraints to avoid wheel liftoff. Based on a projection operator nonlinear optimal control technique, we propose a minimum-time trajectory generation strategy to compute the fastest car trajectory. Numerical computations are presented on two testing scenarios, a 90° turn and a real testing track. The computations allow us to both demonstrate the efficiency and accuracy of the proposed approach and highlight important features of the minimum-time trajectories. Finally, we integrate our strategy into a commercial vehicle dynamics software, thus computing minimum-time trajectories for a complex multibody vehicle model. The matching between the predicted trajectory and the one of the commercial toolbox further highlights the effectiveness of the proposed methodology.

Index Terms—Minimum-time, nonlinear optimal control, race car, trajectory optimization, two-track, vehicle dynamics.

I. INTRODUCTION

THE main goal of car racing teams is to build the fastest and best performing cars possible. Driver and engineers work together to find the best setup, race line, and velocity profile, minimizing the time on a given track. The basic idea is that if we want to perform a real testing, then we need a driver, hopefully with great skills, and a real car, and need to make him drive it on a real track, get some data from telemetry, and then analyze these data to fine-tune the setup and so on. In this paper, we carry out this idea in a similar way by developing a virtual prototyping tool based on nonlinear optimal control techniques, so as to compute the fastest trajectory on a given track.

Many computational strategies have been introduced in the literature to solve the minimum-time problem, see [2]

Manuscript received April 7, 2014; revised September 2, 2014; accepted November 14, 2014. Date of publication February 24, 2015; date of current version June 12, 2015. Manuscript received in final form November 26, 2014. Recommended by Associate Editor S. Di Cairano.

A. Rucco and G. Notarstefano are with the Department of Engineering, Università del Salento, Lecce 73100, Italy (e-mail: alessandro.rucco@unisalento.it; giuseppe.notarstefano@unisalento.it).

J. Hauser is with the Department of Electrical, Computer, and Energy Engineering, University of Colorado, Boulder, CO 80309 USA (e-mail: john.hauser@colorado.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCST.2014.2377777

for a survey. Using a common classification, we can identify two main classes of strategies: 1) *quasi-steady-state* methods and 2) *transient optimal* methods. The key idea of quasi-steady-state methods is to approximate the vehicle trajectory as a concatenation of equilibrium conditions [3], [4]. The track is split into a sequence of segments of suitable length. For each segment it is assumed that the vehicle is in stationary conditions, i.e., the vehicle goes along the segment with a constant speed and lateral acceleration. The lateral tire force needed to ensure the maximum lateral acceleration in the segment is computed. Then, using a combined tire model, the remaining tire forces are sought to compute the longitudinal acceleration. Even if this method provides a sufficiently realistic estimate of minimum-time trajectories, it has two main limitations: 1) it computes the maximum speed profile for a given racing line and 2) important information on transient dynamics is lost, especially when the car is at the limit of its capabilities. Conversely, transient optimal methods, based on optimal control techniques, generally aim at solving or approximating the minimum-time problem by maximizing or minimizing a suitable objective function subject to the dynamics. Different objective functions are discussed in the literature: maximize the distance traveled along a race track in a given fixed time [5], [6], maximize the vehicle speed at the final point of the maneuver [7], and minimize the time [8], [9]. Different optimization methods are used to solve the minimum-time optimal control problem. Based on Bellman's Principle of Optimality, in [10], a semianalytical method is proposed to generate minimum-time velocity profiles for a point-mass model. An extension of that methodology is provided in [11] for a half-car model. In [12] and [13], the gear shift is considered and the resulting mixed-integer control problem is addressed using a direct multiple shooting method. Hendrikx *et al.* [8], Casanova *et al.* [9], and Tavernini *et al.* [14] (see also [15]) propose the use of the distance travelled along the road center-line as an independent variable. The authors solve the problem by applying a gradient descent method, a parallel shooting method, and a symbolic-numerical (indirect) method, respectively. Moreover, the sensitivity of the minimum-time with respect to the yaw inertia for a two-track model is considered in [9] and the effect of road surface and transmission layout of a single-track model is investigated in [14]. A comparison between pure steady-state, quasi-steady-state, and transient optimal methods is discussed in [16].

The contributions of this paper are as follows. First, we develop an optimal control-based strategy, called minimum-time trajectory generator (MTGO), to efficiently solve the optimal control problem and compute realistic minimum-time trajectories. Based on the idea developed in [9], we set up an equivalent constrained optimal control problem in terms of longitudinal and transverse coordinates. The main advantages with respect to the original problem are: the horizon is fixed and the time-varying state constraints are converted into space-varying state constraints. The proposed strategy combines a barrier function relaxation, introduced in [17], to handle the point-wise constraints with a continuation update rule. The main idea is the following. We start with a feasible car trajectory, i.e., a trajectory satisfying the state (lane) and input (steering and tire) constraints. Then, we repeatedly apply the projection operator Newton method [18] to solve the relaxed optimal control problem. At each iteration the relaxed problem is modified by updating the barrier functional parameters (increasing the barrier climb) thus pushing the optimal trajectory toward the constraint boundaries. The interesting consequence of applying the proposed strategy is that the convergence to the optimal trajectory happens in an interior point fashion. This resembles, in our opinion, the learning process of a real driver in pushing the car toward its limit capabilities.

We highlight that our strategy and the ones used in [9] and [14] share the same optimal control problem setup, yet they use significantly different approaches. In [9], a shooting approach is used. The minimum-time trajectory is computed at a number of fixed positions along the independent variable, while the intermediate values are estimated by means of interpolation techniques (enforcing the continuity of the state trajectories). In [14], the minimum-time trajectory is computed using a symbolic-numerical approach. First, the constrained problem is transformed into an unconstrained problem using a symbolic manipulation software. Then, a finite difference discretization leads to a (large) finite-dimensional algebraic problem that is solved using a numerical scheme. In contrast to the previous approaches, the strategy proposed in this paper exploits a state feedback approach of the optimal control solver ensuring the computation of feasible (intermediate) car trajectories. The combination with a continuation method allows us to approximate the minimum-time trajectory in an efficient way. We refer the reader to [19] for a discussion of the aforementioned optimization techniques (including both direct and indirect methods). Notice that a detailed comparison of the proposed approach with respect to previous works goes beyond the scope of this paper, while we are interested to show the effectiveness of the proposed strategy in terms of both accuracy of the solution and computational effort. In this view, numerical computations are reported suggesting that the proposed strategy is less demanding than the ones based on shooting methods from a computational point of view, while exhibiting a comparable accuracy of the solution. Moreover, by considering a suitable constraint on the normal forces, we are able to predict the wheel liftoff condition (i.e., a wheel is losing contact with the ground). This condition has been studied in [20] for rollover threat

assessment, yet, to the best of our knowledge, it has never been discussed in the literature for computing minimum-time trajectories.

Second, through numerical computations, we highlight and discuss a set of interesting features of minimum-time trajectories. First, a common virtual testing maneuver is used as a test case, that is a 90° maneuver. This computation allows us to show the main strength of the proposed strategy. We are able to capture important features of the minimum-time trajectory, e.g., apex points and rally-like maneuvers. In the second test, we compute the minimum-time trajectory for a long race circuit (real testing track). Modifying the center of mass position, we show that the normal force constraints can be active thus affecting the minimum-time trajectory. We want to stress that in computing the minimum-time trajectory, we do not need to split the track into a sequence of short sections. Finally, we provide an application scenario, in which we validate the proposed strategy in a commercial vehicle dynamics software for motor-sport dynamic analysis.

The rest of this paper is organized as follows. In Section II, we briefly introduce the car model developed in previous works. In Section III, we formulate the minimum-time problem (we define cost function, constraints, and boundary conditions) and its equivalent formulation using the transverse dynamics. In addition, we describe the MTGO strategy. Finally, in Section IV, we provide numerical computations. Vehicle behavior, convergence, and computational time are discussed.

II. DYNAMICS OF A TWO-TRACK CAR MODEL

In this section, we briefly introduce a simplified two-track car model: a *reduced-order model* for a two-track car vehicle called RigidCar that considers tire models, and a quasi-static model for longitudinal and lateral load transfer. The RigidCar model is based on the following main assumption: we neglect the suspension dynamics. It is worth noting that, despite this assumption, the model captures an important feature for studying vehicle handling, i.e., the longitudinal and lateral load transfer.

The vehicle dynamics is based on a simplified car model developed in [21]. The model is modified to include front wheel steer rate, traction, and brake force distribution (these would yield more realistic results from a physical point of view). The basic idea of this model is to introduce suitable constraints into the equations of motion which allow computation of the normal forces by means of the reaction forces. In view of its redundant degree of constraints (i.e., the vehicle interacts with the ground at four contact points, yet only three constraints are needed to constrain the vehicle's motion in a plane) the reaction forces are not univocally determined and the vehicle is called a *hyper-static* (or *redundant*) structure. Using the principle of least work, we are able to get an additional equation (called compatibility equation), thus resolving the indeterminateness. A planar view of the rigid car model is shown in Fig. 1. In Table I, we provide a list of the car symbols used in this paper. A rigorous definition of these symbols will

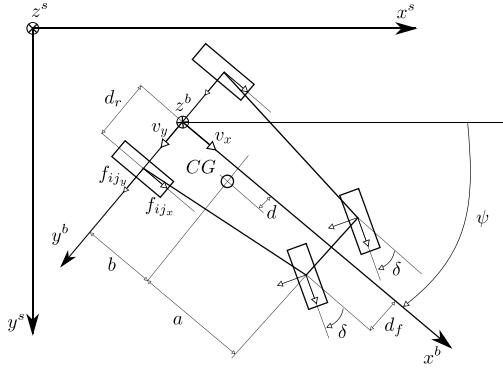


Fig. 1. RigidCar model. The figure shows the quantities used to describe the model.

TABLE I
NOMENCLATURE

a	distance between front axle and CG	m
b	distance between rear axle and CG	m
ℓ	wheelbase, $\ell = (a + b)$	m
d_r	distance between rear contact points and x^b	m
d_f	distance between front contact points and x^b	m
d	distance between CG and x^b	m
m	vehicle's mass	kg
h	height of the CG from the ground	m
I_{zz}, I_{xz}	inertia's terms (wrt z -axis and $x-z$ plane)	kg m^2
μ_{ijx}, μ_{ijy}	longitudinal, lateral force coefficients	-
β_{ij}	sideslip angle	rad
g	gravity acceleration	m/s^2
f_x, f_y, f_z	longitudinal, lateral, normal, tire forces	N
x, y	longitudinal, lateral position midpoint rear axle	m
ψ	car yaw angle	rad
v_x, v_y	longitudinal, lateral velocity midpoint rear axle	m/s
r	car yaw rate	rad/s
δ	front wheel steer angle	rad
u_δ	front wheel steer rate	rad/s

be given in the following model development. The car model is a single planar rigid body with 6 DOF, the longitudinal, lateral, and vertical displacements and the yaw, roll, and pitch angles. It is constrained to move in a plane interacting with the road at four body-fixed contact points according to a suitable tire model. The center of mass and the front and the rear wheels of the vehicle are located, respectively, at distances d , d_f , and d_r from the longitudinal axis. We denote by a and b the distances of the center of mass from the front and the rear contact points, so that $\ell = a + b$ is the wheelbase. The origin of the body frame of the car is located at the midpoint of the rear (contact points) axle with $x^b-y^b-z^b$ axes oriented in a forward-right-down fashion, in accordance with the SAE J670e standard. The orientation R of the (unconstrained) rigid car model is parameterized using roll-pitch-yaw parametrization $R = R(\psi, \theta, \phi) = R_z(\psi)R_y(\theta)R_x(\phi)$, where ϕ , θ , and ψ are, respectively, the roll, pitch, and yaw angles.

Each contact-point/road-plane interaction is modeled using a linear tire model. Before, we clarify our notation. We associate two indices, $i \in \{f, r\}$ (front/rear) and $j \in \{r, l\}$ (right/left), to each tire quantity when we want to provide

an expression holding for a generic tire. Thus, for example, we denote the generic normal tire force f_{ijz} , meaning that we are referring to f_{frz} (f_{flz}) for the front-right (left) tires and f_{rrz} (f_{rlz}) for the rear-right (left) ones. The linear tire model relies on the following assumptions: 1) the longitudinal force is directly controlled; 2) the relationship between the lateral force f_{ijy} and the sideslip β_{ij} is linear (see [22] for experimental data supporting this assumption); and 3) the longitudinal and lateral forces, f_{ijx} and f_{ijy} , are decoupled. Notice that in Section III, we add a suitable constraint such that the longitudinal and lateral forces become in fact coupled, thus capturing the main features of more realistic tire models.

Our model utilizes the following, somewhat reasonable, assumption.

Assumption 2.1: The forces tangential to the road plane, f_{ijx} and f_{ijy} , depend linearly on the normal forces f_{ijz} , that is

$$\begin{aligned} f_{ijx} &= -f_{ijz}\mu_{ijx} \\ f_{ijy} &= -f_{ijz}\mu_{ijy}(\beta_{ij}) \end{aligned}$$

where μ_{ijx} and μ_{ijy} are the longitudinal and lateral force coefficients. \square

The front forces expressed in the body frame, f_{fjx}^b and f_{fjy}^b , are obtained by rotating the forces in the tire frame according to the steer angle δ , so that, e.g., $f_{fjx}^b = f_{fjx}\cos\delta - f_{fjy}\sin\delta$. Substituting the above expressions for f_{fjx} and f_{fjy} , we get

$$\begin{aligned} f_{fjx}^b &= -f_{fjz}(\mu_{fjx}\cos\delta - \mu_{fjy}(\beta_{fj})\sin\delta) \\ &:= -f_{fjz}\tilde{\mu}_{fjx}(\mu_{fjx}, \beta_{fj}, \delta). \end{aligned}$$

In the sequel, abusing notation, we will suppress the tilde and use μ_{fjx} to denote $\tilde{\mu}_{fjx}$.

We assume to control the front-wheel steer rate, u_δ , and the traction and brake coefficient forces, u_t and u_b . The traction and braking coefficient forces are distributed between front and rear wheels as follows:

$$\begin{aligned} \mu_{fjx} &= K_t u_t - K_b u_b \\ \mu_{rjx} &= (1 - K_t)u_t - (1 - K_b)u_b \end{aligned}$$

where K_t and K_b are the traction and brake ratios, respectively.

We start from the (unconstrained) 6 DOF rigid body model. To do that, we derive the dynamics of the system with respect to the following generalized coordinates:

$$q = [x, y, \psi, z, \varphi, \theta]^T = [q_r, q_c]^T.$$

The coordinates $q_r = [x, y, \psi]^T$ are the unconstrained coordinates of the reduced-order car model, while $q_c = [z, \varphi, \theta]^T$ are the constrained ones.

Then, we constrain the four contact points to the road plane to compute the normal tire forces as reaction forces. However, due to the hyper-static nature of the contact point constraints, the expression of the reaction forces cannot be univocally obtained (i.e., the system is underdetermined). To overcome this issue, we apply the principle of least work to recover the reaction forces in a hyper-static structure. In particular, we assume that the contact between the four contact points and the ground happens according to a strain

model and assume that the car structure is rigid. Then, we compute the total strain energy of the car due to the reaction forces at the four contact points, and minimize this energy. The resulting equation, known as compatibility equation, enables the explicit derivation of four reaction forces due to the static load distribution, and the longitudinal and lateral load transfer. Please see [21] for a detailed derivation of the compatibility equation and the reaction forces computation. Using the Euler–Lagrange formulation, we have a dynamic model explicitly depending on the unconstrained coordinates q_r and an explicit expression for the normal forces that can be used to predict the longitudinal and lateral load transfer. Notice that in Section III, the explicit formulation of the normal forces is exploited to avoid the wheel liftoff condition. Moreover, such formulation allows us to ensure the validity of the RigidCar model (the ground is a unilateral constraint and, therefore, cannot generate a positive reaction force).

Since the dynamics does not depend on the positions x and y , and the orientation ψ , we can work directly with the longitudinal velocity v_x , the lateral velocity v_y , and the yaw rate r . Expressing the dynamics in the body frame, the equations of motion are given by

$$\begin{aligned}\dot{q}_r &= f_{q_r}(q_r, q_v) \\ \dot{q}_v &= f_{q_v}(q_v, u)\end{aligned}\quad (2)$$

where $q_v = [v_x, v_y, r, \delta]^T$ and $u = [u_\delta, u_t, u_b]^T$. In this way, we can decouple the dynamics of the vehicle, $\dot{q}_v = f_{q_v}(q_v, u)$, from its kinematics, $\dot{q}_r = f_{q_r}(q_r, q_v)$. The dynamics of the vehicle is described in (1), as shown at the bottom of this page, and the kinematics is given by

$$\begin{aligned}\dot{x} &= v_x \cos \psi - v_y \sin \psi \\ \dot{y} &= v_x \sin \psi + v_y \cos \psi.\end{aligned}\quad (3)$$

For the sake of space, in (1) we omit the dependence of μ_{ijx} and μ_{ijy} from β_{ij} , and δ we define $\mu_{fr_{xy}} = \ell \mu_{fr_y} - d_f \mu_{fr_x}$, $\mu_{fl_{xy}} = \ell \mu_{fl_y} + d_f \mu_{fl_x}$, $\tilde{I}_{zz} = I_{zz} + m(b^2 + d^2)$, and $\tilde{I}_{xz} = I_{xz} + mbh$.

III. NONLINEAR OPTIMAL CONTROL BASED MINIMUM-TIME TRAJECTORY COMPUTATION

In this section, we address the minimum-time problem and describe the optimal control-based strategies used to compute the minimum-time trajectory of the car vehicle.

A. Minimum-Time Problem and Equivalent Formulation

To formulate the minimum-time problem, we need to define the cost function to be optimized and specify the state/input constraints and boundary conditions. In this paper, we consider the problem of finding a trajectory of the RigidCar system that minimizes the time T to complete a given track with fixed initial point, track boundary constraints, input control, and normal force constraints

$$\begin{aligned}&\min_{q_r(\cdot), q_v(\cdot), u(\cdot)} \int_0^T 1 \, d\tau \\ &\text{s.t. dynamics constraints} \\ &\dot{q}_r = f_{q_r}(q_r, q_v) \quad q_r(0) = q_{r0} \\ &\dot{q}_v = f_{q_v}(q_v, u) \quad q_v(0) = q_{v0} \\ &\text{point-wise state/input constraints} \\ &\text{pwc}_w(x(t), y(t), t, w_{\text{track}}) \leq 0 \quad (\text{track boundary}) \\ &\text{pwc}_\delta(\delta(t), \delta_{\max}) \leq 0 \quad (\text{steer angle}) \\ &\text{pwc}_{u_\delta}(u_\delta(t), u_{\delta\max}) \leq 0 \quad (\text{steer rate}) \\ &\text{pwc}_t(q_v(t), d_{X\max}, d_{Y\max}) \leq 0 \quad (\text{tire force}) \\ &f_{ij_z}(t) \geq 0 \quad (\text{normal force}) \\ &u_t(t) \geq 0 \quad (\text{traction coefficient force}) \\ &u_b(t) \geq 0 \quad (\text{brake coefficient force})\end{aligned}\quad (4)$$

where pwc_w , pwc_δ , pwc_{u_δ} , and pwc_t represent, respectively, the track boundary, steer angle, steer rate, and tire point-wise constraints. Specifically, the distance between the midpoint of the rear (contact points) axle and the center of the race track is limited by the track width w_{track} . We choose to constrain only the midpoint of the rear axle to the race track for simplicity. The front wheel steer angle, δ , and rate, $\dot{\delta}$, are bounded in module by δ_{\max} and $u_{\delta\max}$. To consider the limited grip of the tires, we constrain the longitudinal and lateral forces into an ellipse (friction ellipse), with maximum pure longitudinal (lateral) force defined by $d_{X\max}$ ($d_{Y\max}$). To avoid the liftoff condition, the normal forces, f_{ij_z} , are imposed to be nonnegative ($f_{ij_z} \geq 0$). The traction and brake force coefficients, u_t and u_b , are constrained to be nonnegative as well.

To formulate an equivalent version of the minimum-time problem, we define a new set of coordinates and split the dynamics into a longitudinal one and a transverse one. We define the new set of coordinates by replacing the

$$\begin{bmatrix} m & 0 & -md & 0 & \mu_{fr_x} & \mu_{fl_x} & \mu_{rr_x} & \mu_{rl_x} \\ 0 & m & mb & 0 & \mu_{fr_y} & \mu_{fl_y} & \mu_{rr_y} & \mu_{rl_y} \\ -md & mb & \tilde{I}_{zz} & 0 & \mu_{fr_{xy}} & \mu_{fl_{xy}} & -d_r \mu_{rr_x} & d_r \mu_{rl_x} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 0 & mh & \tilde{I}_{xz} & 0 & -d_f & d_f & -d_r & d_r \\ -mh & 0 & mhd & 0 & \ell & \ell & 0 & 0 \\ 0 & 0 & 0 & 0 & -d_r & d_r & d_f & -d_f \end{bmatrix} \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{r} \\ \dot{\delta} \\ f_{fr_z} \\ f_{fl_z} \\ f_{rr_z} \\ f_{rl_z} \end{bmatrix} + \begin{bmatrix} -mr(v_y + br) \\ mr(v_x - dr) \\ mr(dv_y + bv_x) \\ 0 \\ 0 \\ mhr(v_x - dr) \\ mhv_y r + \tilde{I}_{xz} r^2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -mg \\ -mgd \\ mgb \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ u_\delta \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (1)$$

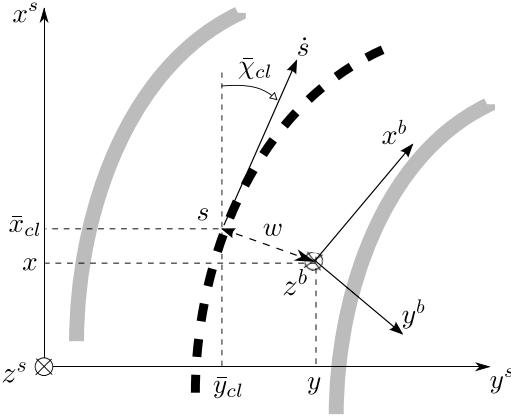


Fig. 2. Local coordinates around the path. The bold dashed line indicates the center line.

position coordinates with: a longitudinal coordinate s , representing the position along the center line of the track, and a lateral displacement w representing the displacement transverse to the central line. In our context, a natural choice of the new parametrization is the arc-length of the curve

$$s(t) = \int_0^t \left(\sqrt{\dot{x}_{cl}^2(\tau) + \dot{y}_{cl}^2(\tau)} \right) d\tau$$

where $x_{cl}(\cdot)$ and $y_{cl}(\cdot)$ are the longitudinal and lateral coordinates of the center line, respectively. If the velocity, $\dot{s}(t)$, is bounded away from zero, the mapping $t \mapsto s(t)$ is strictly increasing, so that we can compute its inverse. This allows us to define the state $q_r(t)$, $q_v(t)$, and input $u(t)$ trajectories as a function of the arc-length instead of time, i.e., $\bar{q}_r(s)$, $\bar{q}_v(s)$, and $\bar{u}(s)$, respectively. From now on, the bar symbol indicates that a quantity is expressed as a function of the longitudinal parameter s rather than the time t .

Given the arc-length parametrization of the track center line, $s \mapsto (\bar{x}_{cl}(s), \bar{y}_{cl}(s))$, we choose a set of local coordinates in a tube around it. The coordinates of the midpoint of the rear (contact points) axle (x, y) can be defined with respect to the arc-length of the point at minimum distance from the track center line, see Fig. 2, that is

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \bar{x}_{cl}(s) \\ \bar{y}_{cl}(s) \end{bmatrix} + \begin{bmatrix} -w \sin \bar{\chi}_{cl}(s) \\ w \cos \bar{\chi}_{cl}(s) \end{bmatrix}. \quad (5)$$

From now on, we use a prime symbol to denote the first derivative of a variable with respect to the arc-length s . For example, $\bar{x}'(s) = d\bar{x}(s)/ds$, while $\dot{x}(t) = dx(t)/dt$.

Remark 3.1: The transverse displacement w is orthogonal to the unit tangent vector $(\bar{x}'(s), \bar{y}'(s)) = (\cos \bar{\chi}_{cl}(s), \sin \bar{\chi}_{cl}(s))$. The inverse of the map $(s, w) \mapsto (x, y)$ is locally well defined around points (s, w) satisfying $w\bar{\sigma}_{cl}(s) < 1$, where $\bar{\sigma}_{cl} = \bar{\chi}'_{cl}$ is the curvature of the map at $(\bar{x}(s), \bar{y}(s))$. \square

By differentiating (5) with respect to time t , we get the expression of \dot{s} and \dot{w} , which are

$$\begin{aligned} \dot{s} &= \frac{v_x \cos(\psi - \bar{\chi}_{cl}(s)) - v_y \sin(\psi - \bar{\chi}_{cl}(s))}{1 - \bar{\sigma}_{cl}(s)w} \\ \dot{w} &= v_x \sin(\psi - \bar{\chi}_{cl}(s)) + v_y \cos(\psi - \bar{\chi}_{cl}(s)). \end{aligned} \quad (6)$$

Defining the local heading angle as $\mu = \psi - \bar{\chi}_{cl}(s)$, the nonlinear system (2) can be written with respect to the new set of coordinates $(s(\cdot), w(\cdot), \mu(\cdot), q_v(\cdot), u(\cdot))$

$$\begin{aligned} \dot{s} &= \frac{v_x \cos \mu - v_y \sin \mu}{1 - \bar{\sigma}_{cl}(s)w} \\ \dot{w} &= v_x \sin \mu + v_y \cos \mu \\ \dot{\mu} &= \dot{\psi} - \bar{\sigma}_{cl}(s) \frac{v_x \cos \mu - v_y \sin \mu}{1 - \bar{\sigma}_{cl}(s)w} \\ \dot{q}_v &= f_{q_v}(q_v, u). \end{aligned} \quad (7)$$

The nonlinear differential equation (7) is the longitudinal and transverse dynamics form of the nonlinear system (2).

The constraints introduced in the minimum-time problem (4) can be rewritten in the new set of coordinates. The constraint on the track boundary assumes a very simple form. Indeed, it is given by $|\bar{w}(s)| \leq w_{\text{track}}$. To have a smooth function defining the constraint, we rewrite it in the equivalent form

$$\left(\frac{\bar{w}(s)}{w_{\text{track}}} \right)^2 - 1 \leq 0 \quad \forall s \in [0, S]$$

where S is the total length of the track.

Remark 3.2: This constraint ensures also that the change of coordinates is well defined in a tubular neighborhood of the center line $(\bar{x}_{cl}(\cdot), \bar{y}_{cl}(\cdot))$. The condition $\bar{w}(s)\bar{\sigma}_{cl}(s) < 1$ in Remark 3.1 is true for all $s \in [0, S]$, provided that w_{track} is appropriately chosen. \square

In the same way, we define the constraint on the front wheel steer angle and rate as

$$\begin{aligned} \left(\frac{\bar{\delta}(s)}{\delta_{\max}} \right)^2 - 1 &\leq 0 \quad \forall s \in [0, S] \\ \left(\frac{\bar{u}_{\delta}(s)}{u_{\delta\max}} \right)^2 - 1 &\leq 0 \quad \forall s \in [0, S]. \end{aligned}$$

To prevent the tires from operating in a high-saturation region, we consider the linear tire model and we constrain the total force acting on each wheel limited within the ellipse of maximum tire force, that is

$$\left(\frac{\bar{\mu}_{ij_x}(s)}{d_{X\max}} \right)^2 + \left(\frac{\bar{\mu}_{ij_y}(s)}{d_{Y\max}} \right)^2 - 1 \leq 0 \quad \forall s \in [0, S].$$

Remark 3.3: The combination of the linear tire model, introduced in the previous section, with the ellipse constraint gives a nonlinear tire model approximating the commonly used Pacejka model [23]. \square

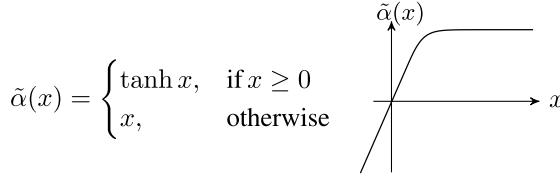
Remark 3.4: It is worth noting that the longitudinal and lateral forces are coupled when the tire forces are saturated. In reality, a coupling exists also when the forces do not saturate. However, when computing minimum-time trajectories, the tire forces work on the ellipse constraints for almost all the time [3], [7]. For this reason, we use a linear tire model and impose the combined ellipse constraint, so that the longitudinal and lateral forces are coupled when the tire forces are saturated. \square

We constrain the normal forces on each wheel limited within a maximum and a minimum normal force, that is

$$\left(\frac{2\bar{f}_{ij_z}(s) - (f_{Z\max} + f_{Z\min})}{f_{Z\max} - f_{Z\min}} \right)^2 - 1 \leq 0 \quad \forall s \in [0, S]$$

where $0 \leq f_{Z\min} < f_{Z\max} \leq mg$.

Finally, to consider positive values of traction and brake coefficient forces, u_t and u_b , we use the following \mathcal{C}^2 -smooth function:



and we consider

$$\begin{aligned} \tilde{\alpha}(\bar{u}_t(s)) &\geq 0 \\ \tilde{\alpha}(\bar{u}_b(s)) &\geq 0 \quad \forall s \in [0, S]. \end{aligned}$$

Now, we address the minimum-time problem with respect to the new parametrization. We eliminate the time t and write the dynamics as a differential equation where the longitudinal coordinate s becomes the independent variable. The transverse coordinates, the generalized velocity coordinates, and the input control at location s are $\bar{w}(s)$, $\bar{\mu}(s)$, $\bar{q}_v(s)$, and $\bar{u}(s)$, and the dynamics (7) becomes

$$\begin{aligned} \bar{w}' &= (\bar{v}_x \sin \bar{\mu} + \bar{v}_y \cos \bar{\mu}) \frac{1 - \bar{\sigma}_{cl}(s)\bar{w}}{\bar{v}_x \cos \bar{\mu} - \bar{v}_y \sin \bar{\mu}} \\ \bar{\mu}' &= \bar{r} \frac{1 - \bar{\sigma}_{cl}(s)\bar{w}}{\bar{v}_x \cos \bar{\mu} - \bar{v}_y \sin \bar{\mu}} - \bar{\sigma}_{cl}(s) \\ \bar{v}'_x &= f_{q_v}^1(\bar{q}_v, \bar{u}) \frac{1 - \bar{\sigma}_{cl}(s)\bar{w}}{\bar{v}_x \cos \bar{\mu} - \bar{v}_y \sin \bar{\mu}} \\ \bar{v}'_y &= f_{q_v}^2(\bar{q}_v, \bar{u}) \frac{1 - \bar{\sigma}_{cl}(s)\bar{w}}{\bar{v}_x \cos \bar{\mu} - \bar{v}_y \sin \bar{\mu}} \\ \bar{r}' &= f_{q_v}^3(\bar{q}_v, \bar{u}) \frac{1 - \bar{\sigma}_{cl}(s)\bar{w}}{\bar{v}_x \cos \bar{\mu} - \bar{v}_y \sin \bar{\mu}} \\ \bar{\delta}' &= f_{q_v}^4(\bar{q}_v, \bar{u}) \frac{1 - \bar{\sigma}_{cl}(s)\bar{w}}{\bar{v}_x \cos \bar{\mu} - \bar{v}_y \sin \bar{\mu}} \end{aligned} \quad (8)$$

where $f_{q_v}^1(\cdot)$, $f_{q_v}^2(\cdot)$, $f_{q_v}^3(\cdot)$, and $f_{q_v}^4(\cdot)$ are the components of the function $f_{q_v}(\cdot)$. In compact form

$$\bar{q}'_s(s) = \bar{f}(\bar{q}_s(s), \bar{u}(s), \bar{\sigma}_{cl}(s)) \quad (9)$$

where $\bar{q}_s(\cdot) = [\bar{w}(\cdot), \bar{\mu}(\cdot), \bar{v}_x(\cdot), \bar{v}_y(\cdot), \bar{r}(\cdot), \bar{\delta}(\cdot)]^T$.

An equivalent formulation of the minimum-time problem, expressed in the new set of coordinates and in the new

variable s , is given by

$$\begin{aligned} &\min_{\bar{q}_s(\cdot), \bar{u}(\cdot)} \int_0^S \frac{1 - \bar{\sigma}_{cl}(s)\bar{w}(s)}{\bar{v}_x(s) \cos \bar{\mu}(s) - \bar{v}_y(s) \sin \bar{\mu}(s)} ds \\ &\text{s.t. } \bar{q}'_s = \bar{f}(\bar{q}_s(s), \bar{u}(s), \bar{\sigma}_{cl}(s)), \quad \bar{q}_s(0) = q_{s0} \\ &\quad \left(\frac{\bar{w}(s)}{w_{\text{track}}} \right)^2 - 1 \geq 0 \\ &\quad \left(\frac{\bar{\delta}(s)}{\delta_{\max}} \right)^2 - 1 \geq 0 \\ &\quad \left(\frac{\bar{u}_\delta(s)}{u_{\delta\max}} \right)^2 - 1 \geq 0 \\ &\quad \left(\frac{\bar{\mu}_{ij_x}(s)}{d_{X\max}} \right)^2 + \left(\frac{\bar{\mu}_{ij_y}(s)}{d_{Y\max}} \right)^2 - 1 \geq 0 \\ &\quad \left(\frac{2\bar{f}_{ij_z}(s) - (f_{Z\max} + f_{Z\min})}{f_{Z\max} - f_{Z\min}} \right)^2 - 1 \geq 0 \\ &\quad \tilde{\alpha}(\bar{u}_t) \geq 0 \\ &\quad \tilde{\alpha}(\bar{u}_b) \geq 0. \end{aligned} \quad (10)$$

Problems (4) and (10) are equivalent in the sense that trajectories that solve problem (4) can be mapped to trajectories that solve (10) and vice versa. The last form has several benefits. The main one is that the problem is a fixed-horizon free-endpoint problem.

B. Minimum-Time Trajectory Generation Strategy

Now, we describe the optimal control-based strategy used to compute the minimum-time trajectory. We address the constrained optimal control problem by computing solutions of a suitable unconstrained optimal control problem.

We are interested in developing unconstrained optimal control strategies that can be used to find approximate solutions to the constrained optimal control problem (10). To facilitate the local exploration of trajectories of the nonlinear system, we use the projection operator-based Newton method for trajectory optimization (PRONTO) (Appendix B). The key point is the design of a projection operator \mathcal{P} which maps the curve $\xi_c = (\bar{q}_c(\cdot), \bar{u}_c(\cdot))$ to the trajectory $\xi = (\bar{q}_s(\cdot), \bar{u}(\cdot))$. Specifically, the space-varying trajectory-tracking control law

$$\begin{aligned} \bar{q}'_s(s) &= \bar{f}(\bar{q}_s(s), \bar{u}(s), \bar{\sigma}_{cl}(s)), \quad \bar{q}_s(0) = q_{s0} \\ \bar{u}(s) &= \bar{u}_c(s) + K(s)(\bar{q}_c(s) - \bar{q}_s(s)) \end{aligned} \quad (11)$$

defines the projection operator $\mathcal{P} : \xi \mapsto \eta$. The definition of \mathcal{P} involves the solution to a suitable linear quadratic optimal control problem providing the projection operator with a stability-like property. It is worth noting that we are dealing with finite-time trajectories. The feedback system (11) is guaranteed to exponentially stabilize the trajectory as the time horizon goes to infinity [18], [24]. This also guarantees a numerical stability in the optimization procedure avoiding common numerical instabilities that happen in single-shooting methods.

Following [17], we use a barrier function relaxation to consider the constraints. Formally, let $c_j(\bar{q}_s(s), \bar{u}(s))$, $j \in \{1, \dots, 7\}$ denote the constraints in problem (10).

For a given state-input curve $\xi = (\bar{q}_s(\cdot), \bar{u}(\cdot))$, a barrier functional can be defined as

$$b_\nu(\xi) = \int_0^S \Sigma_j \beta_\nu(-c_j(\bar{q}_s(\cdot), \bar{u}(\cdot))) ds$$

where

$$\beta_\nu(x) = \begin{cases} -\log x, & x > \nu \\ -\log \nu + \frac{1}{2} \left[\left(\frac{x-2\nu}{\nu} \right)^2 - 1 \right], & x \leq \nu. \end{cases}$$

Let us denote the cost functional of problem (10) as

$$h_0(\xi; \bar{\sigma}_{cl}) = \int_0^S \frac{1 - \bar{\sigma}_{cl}(s)\bar{w}}{\bar{v}_x \cos \bar{\mu} - \bar{v}_y \sin \bar{\mu}} ds - \log \bar{v}_x(S) \quad (12)$$

where the terminal cost $-\log \bar{v}_x(S)$ is added to have the final velocity as large as possible. Using the barrier functional defined above, the relaxed version of problem (10) is given by

$$\begin{aligned} \min_{\xi} h_0(\xi; \bar{\sigma}_{cl}) + \epsilon b_\nu(\xi) \\ \text{s.t. } \bar{q}'_s(s) = \bar{f}(\bar{q}_s(s), \bar{u}(s), \bar{\sigma}_{cl}(s)), \quad \bar{q}_s(0) = q_{s0}. \end{aligned} \quad (13)$$

The strategy is to start with a reasonably large ϵ and ν . Then, for the current ϵ and ν , the relaxed problem (13) is solved using the PRONTO method starting from the current trajectory. The projection operator Newton method, being a descent method, guarantees the convergence to a local minimum of the optimal control problem relaxation (13). A naive application of the method and, even more, a naive choice of the initial trajectory, ξ_0 , may let the algorithm converge to a (local minimum) trajectory that is too far from the minimum-time trajectory. To deal with this issue, we develop an exploration strategy based on a continuation idea to iteratively apply the projection operator Newton method.

First, we define the center line of the track. The center line is described by the following parameters (related to each other): the longitudinal and lateral coordinates with respect to a fixed spatial frame, $\bar{x}_{cl}(\cdot)$ and $\bar{y}_{cl}(\cdot)$, the radius of curvature, $\bar{\sigma}_{cl}(\cdot)$, and the tangent angle, $\bar{\chi}_{cl}(\cdot)$. Then, we have to design the initial trajectory to initialize the projection operator Newton method. The initial trajectory is based on the center line of the track. In particular, given the curvature of the center line, $\bar{\sigma}_{cl}(\cdot)$, we choose a suitable constant velocity profile, v_0 , and, for $s \in [0, S]$, we define the following (state input) curve $\xi_0 = [0, 0, v_0, 0, v_0 \bar{\sigma}_{cl}(\cdot), 0, 0, 0, 0]^T$. This curve is not a trajectory of the RigidCar system [i.e., it does not satisfy the dynamics (9)]. Using the trajectory-tracking projection operator (11), the curve is projected onto the set of system trajectories to get a suitable initial trajectory, i.e., $\xi_0^1 = \mathcal{P}(\xi_0)$.

Now, with the center line track and the initial trajectory in hand, we repeatedly perform the following steps. We compute the optimal time trajectory using the projection operator Newton method, $\xi_{opt}^1 = \text{PRONTO}(\xi_0^1)$. Then, at the i th step, we update the initial trajectory, ξ_0^i , with the previous optimal trajectory, ξ_{opt}^{i-1} , and the constraints parameters, ϵ and ν , to solve problem (13) with an updated barrier functional. Algorithm 1 gives a pseudo code description of the exploration strategy. An illustration of the approach is shown in Fig. 3.

Algorithm 1 Min-Time Trajectory Generator, MTGO

Given: center-line track $(\bar{x}_{cl}(\cdot), \bar{y}_{cl}(\cdot), \bar{\chi}_{cl}(\cdot), \bar{\sigma}_{cl}(\cdot))$; set a “suitable” initial velocity and compute $\xi_0(\bar{\sigma}_{cl}(\cdot))$; compute transverse dynamics $q'_s = f(\bar{q}_s, \bar{u}, \bar{\sigma}_{cl})$; generate an initial “low-speed/centerline” trajectory ξ_0^1 ; set $\epsilon = 1, \nu = 1$; **for** $i = 1, 2, \dots$ **do**
 set the functional

$$\tilde{h}(\xi; \bar{\sigma}_{cl}) = h_0(\xi; \bar{\sigma}_{cl}) + \epsilon b_\nu(\xi)$$

 compute: $\xi_{opt}^i = \text{PRONTO}(\tilde{h}(\xi; \bar{\sigma}_{cl}), \xi_0^i)$;
 update: ϵ, ν , and set $\xi_0^{i+1} = \xi_{opt}^i$;
end for
Output: $\xi_{opt} = \xi_{opt}^{end}$.

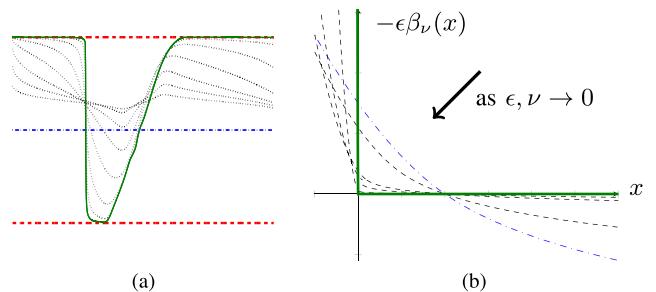


Fig. 3. (a) Sequence of input trajectory. (b) Barrier function relaxation. Minimum-time trajectory generation approach: start with a feasible, low speed, trajectory [see the dashed-dotted blue line in (a)]; compute the new intermediate optimal trajectory by solving the relaxed optimal control problem (13) [see the light dotted black lines in (a)]; increase the barrier climb [see the corresponding (finite dimensions) approximate barrier function in (b)]; and use the continuation approach, thus pushing the intermediate trajectory closer and closer to the boundary constraints [see the dashed red line in (a)]. The solid green line is the input trajectory obtained as candidate minimum-time trajectory.

IV. NUMERICAL ANALYSIS OF MINIMUM-TIME TRAJECTORIES

In this section, we provide numerical computations showing the effectiveness of the proposed strategy in terms of both accuracy of the solution and computational effort. To evaluate the performance of the algorithm, we use the distance from the apex and kissing points as indexes of accuracy of the solution. The apex is referred to the point where the vehicle is closest to the lane boundaries (i.e., $w = w_{\text{track}}$) on the inside of the turn (see the apex point for a 90° maneuver in Fig. 4 highlighted with 2). When the vehicle is traveling before or after a given turn and $w = w_{\text{track}}$, then we refer to a kissing point (highlighted with 1 and 3 in Fig. 4). To gain initial insights into the minimum-time trajectory, we start with a relatively simple benchmark scenario: the 90° maneuver. Then, as an application on a more realistic scenario, the minimum-time trajectory for a real track is computed. Finally, we validate the strategy by comparing the computed minimum-time trajectories with the ones of a complex vehicle model showing how, we are able to capture important behaviors of complex car models.

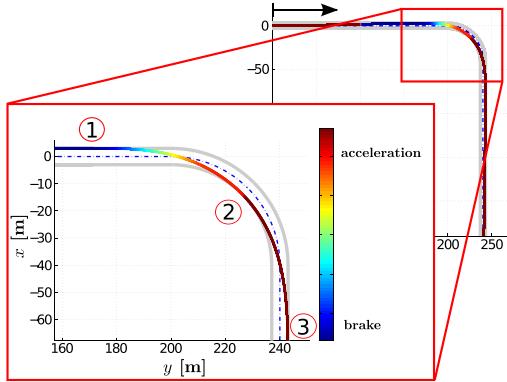


Fig. 4. Minimum-time trajectory: 90° maneuver. The dashed-dotted and solid lines represent the center line of the track and the RigidCar minimum time race line.

A. Minimum-Time Trajectory for a 90° Maneuver

The maneuver consists of a 90° right turn with a radius of 40m and straight sections of 200m before and after the turn (Fig. 4). The road width is taken to be 6m. The curvature $\bar{\sigma}_{cl}(\cdot)$ is approximated analytically by a suitable combination of two hyperbolic tangent functions. The longitudinal and lateral coordinates, $(\bar{x}_{cl}(\cdot), \bar{y}_{cl}(\cdot))$, and the tangent angle, $\bar{\chi}_{cl}(\cdot)$ are given by integration of

$$\begin{aligned}\bar{x}'_{cl} &= \cos \bar{\chi}_{cl} \\ \bar{y}'_{cl} &= \sin \bar{\chi}_{cl} \\ \bar{\chi}'_{cl} &= \bar{\sigma}_{cl}\end{aligned}$$

The minimum-time (optimal) trajectory for a sports car with a rear-wheel drive is shown in Figs. 4 and 5. The car parameters are given in Appendix A.

We stress that the initial trajectory is feasible. By setting as initial longitudinal velocity $v_{x0} = 5$ m/s, the constraints are not active, see dash-dotted blue line in Fig. 5(d)–(f). In particular, the initial race line is almost on the center line of the track. Thus, the maximum value of the lateral displacement is about -0.6 m [Fig. 5(d)]. Note that only the steer wheel angle profile is found to be close to the saturation bound (the maximum value is 3.8°). Moreover, we observe that the normal forces are almost constant [Fig. 5(c)] (the maximum deviation is about 120 N).

The MTGO algorithm starts with the constraint parameters $\epsilon = 1$ and $\nu = 1$. The (intermediate) optimal trajectory is achieved in six iterations and the algorithm shows a quadratic convergence rate [we recall that the PRONTO has a structure of a standard Newton method (Appendix B)]. Next, the constraint parameters are decreased. We have found $\epsilon \leftarrow \epsilon/6$ and $\nu \leftarrow \nu/6$ to work well. We want to stress that the algorithm works in an interior point fashion. Thus the intermediate optimal trajectories, see the light dotted lines in Fig. 5, are strict feasible as ϵ and ν are reduced.

It is interesting to see the effect of the apex point on the velocity profile. The velocity decreases and increases again, with a higher and higher slope as ϵ and ν decrease, in a symmetric way with respect to the apex point [Fig. 5(a) and

(b)]. The constraints start to be active at the fifth iteration. The car approaches the apex point at $s = 233$ m, see Figs. 4 (highlighted with 2) and 5(d), and the lateral tire forces are almost saturated, see Fig. 5(f). Then, the trajectory is pushed closer and closer to the boundary constraints. At the eighth iteration the barrier parameters are about 10^{-6} . The car approaches the kissing points at $s = 168$ m and $s = 303$ m and the tire forces work on the ellipse constraints for almost all the track, see Figs. 4 (highlighted with 1 and 3) and 5(d) and (f). Note that only the steer angle and normal force constraints are not active, see Fig. 5(c) and (e).

In Table II, we provide the traveling time, the indexes of accuracy, and the associated CPU time (on a commercial notebook with a 2 GHz Intel core i7 and with 8 GB of RAM). The minimum traveling time is 13.10 s. Note that the traveling time is monotonically decreasing as desired. The apex and kissing point distances show a high accuracy of the computed minimum-time trajectory. The overall computational time is under 8 min. The number of optimization variables is approximately 41 000 (note that the track length is 462.8 m and the space discretization is 0.1 m).

Confirming our intuition, the minimum-time trajectory has the following features. First, the car accelerates in the first straight portion of the track. Then, it moves toward the outside edge of the corner (first kissing point), brakes, and turns into the corner through the apex point. Finally, the car starts the exit from the corner, approaches to the second kissing point, and accelerates with maximum traction force. It is worth noting that, as it appears in some maneuvers performed by race drivers, the vehicle does not remain on the track boundary. Due to the presence of the lateral dynamics, a sharp variation of the steer angle would be needed to remain on the boundary, thus decreasing the maximum longitudinal force and, therefore, the traveling time.

We also emphasize the significant and rapid load transfer [Fig. 5(c)]. In static conditions (i.e., longitudinal and lateral accelerations are zero) 21% of the total weight is on the front wheels and 29% is on the rear wheels (remember that $d = 0$ m). In the acceleration phase, we note that 12% of the weight is on the front wheels and 38% is on the rear wheels. Before the right turn, due to the brake action, the weight is shifted toward the front axle: the 33% (17%) of the weight is on the front (rear) wheels. Then, the vehicle turns into the corner. The front and rear right wheels become unloaded: the lateral acceleration is pushing the weight on the left side of the vehicle and only 2% (7%) of the weight is on the front (rear) right wheel. This analysis confirms the importance of considering the longitudinal and lateral load transfer to compute minimum-time trajectories.

Now, let us take a look at the steer angle profile at $s \in [130, 180]$ m, see Fig. 5(e). The steer angle has a small swing with respect to its mean value right before the turn. This behavior is highlighted by the lateral velocity profile [Fig. 5(b)] and it starts to appear when the tire forces approach the saturation bound. Such a feature is related to the steer-braking phenomenon. A detailed explanation of the steer-braking phenomenon goes beyond the scope of this paper, and we refer the reader to [25] for a detailed explanation of this phenomenon.

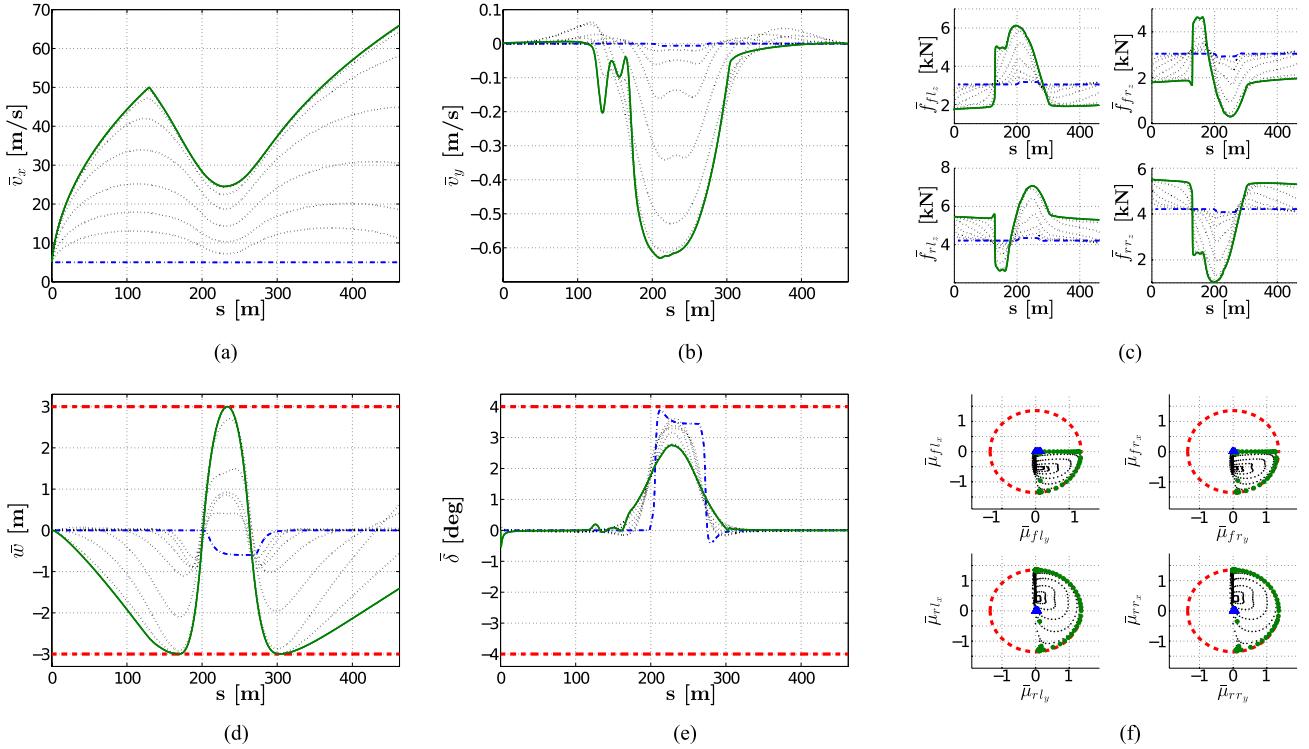


Fig. 5. Minimum-time trajectory for a 90° maneuver. The initial (dashed-dotted line), the intermediate (light dotted line), and the minimum-time (solid line) trajectories are shown. Constraints are in dashed line. (a) Longitudinal velocity \bar{v}_x . (b) Lateral velocity \bar{v}_y . (c) Normal forces \bar{f}_{ij_z} . (d) Lateral displacement \bar{w} . (e) Steer angle $\bar{\delta}$. (f) Ellipse constraints $\bar{\mu}_{ij_x}$ versus $\bar{\mu}_{ij_y}$.

TABLE II

90° MANEUVER: TRAVELING TIME, INDEXES OF ACCURACY, AND COMPUTATIONAL TIME

Values for ϵ and ν	1	1.6e-01	2.7e-02	4.6e-03	7.7e-04	1.2e-04	2e-05	3e-06
traveling time [s]	41.95	29.59	21.01	16.15	13.94	13.27	13.13	13.10
apex point distance [m]	2.58	2.14	2.07	1.50	0.30	0.01	9e-05	1e-05
kissing point distance [m]	2.32	2.07	2.12	1.90	0.90	0.08	4e-03	2e-05
computational time [s]	40	25	25	46	74	52	129	96

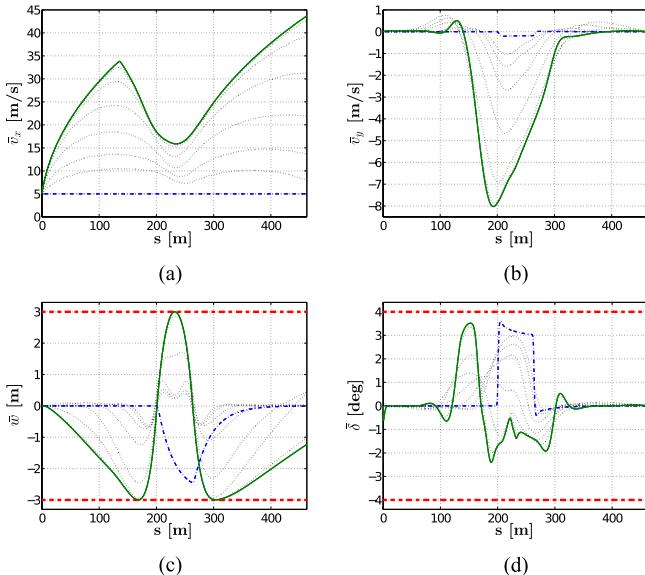


Fig. 6. Minimum-time trajectory for a 90° maneuver: the pendulum turn. (a) Longitudinal velocity \bar{v}_x . (b) Lateral velocity \bar{v}_y . (c) Lateral displacement \bar{w} . (d) Steer angle $\bar{\delta}$.

Another interesting phenomenon is the one in which the road-tire characteristics are the ones of rally cars. As discussed in [14], [15], and [26], rally drivers use high sideslip (drifting)

maneuvers. To investigate such a feature, we compute the minimum-time trajectory by considering a different road-tire setup of the sports car. In particular, we consider a tire with an off-road characteristic curve [Fig. 11] in Appendix A. We want to stress that the proposed strategy does not require any *ad hoc* setup: we compute the minimum-time trajectory without changing the problem setup, strategy, and initial trajectory, but (once again) only the road-tire parameter.

In Fig. 6, the minimum-time trajectory is shown. First, similar to the previous computation, the optimal race line profile approaches the apex point at $s = 233$ m and the kissing points at $s = 168$ m and $s = 303$ m [Fig. 6(c)]. Due to the low tire friction, the longitudinal velocity profile has a lower slope, maximum and minimum values, see Fig. 6(a). Next, looking at the lateral velocity profile, see Fig. 6(b), we see that it denotes a drifting trajectory. The maximum value is about -8 m/s (i.e., equivalent to $\beta = \arctan v_y/v_x = -26^\circ$). Fig. 6(d) shows an interesting feature of the proposed strategy. The sequence of optimal steer angle profiles shows that the expert driver starts with the usual driving technique (light dotted lines): the driver moves the vehicle on the left of the track, then increases the steer angle to steer the vehicle up to the apex point, and finally starts to accelerate with the maximum traction force. Thanks

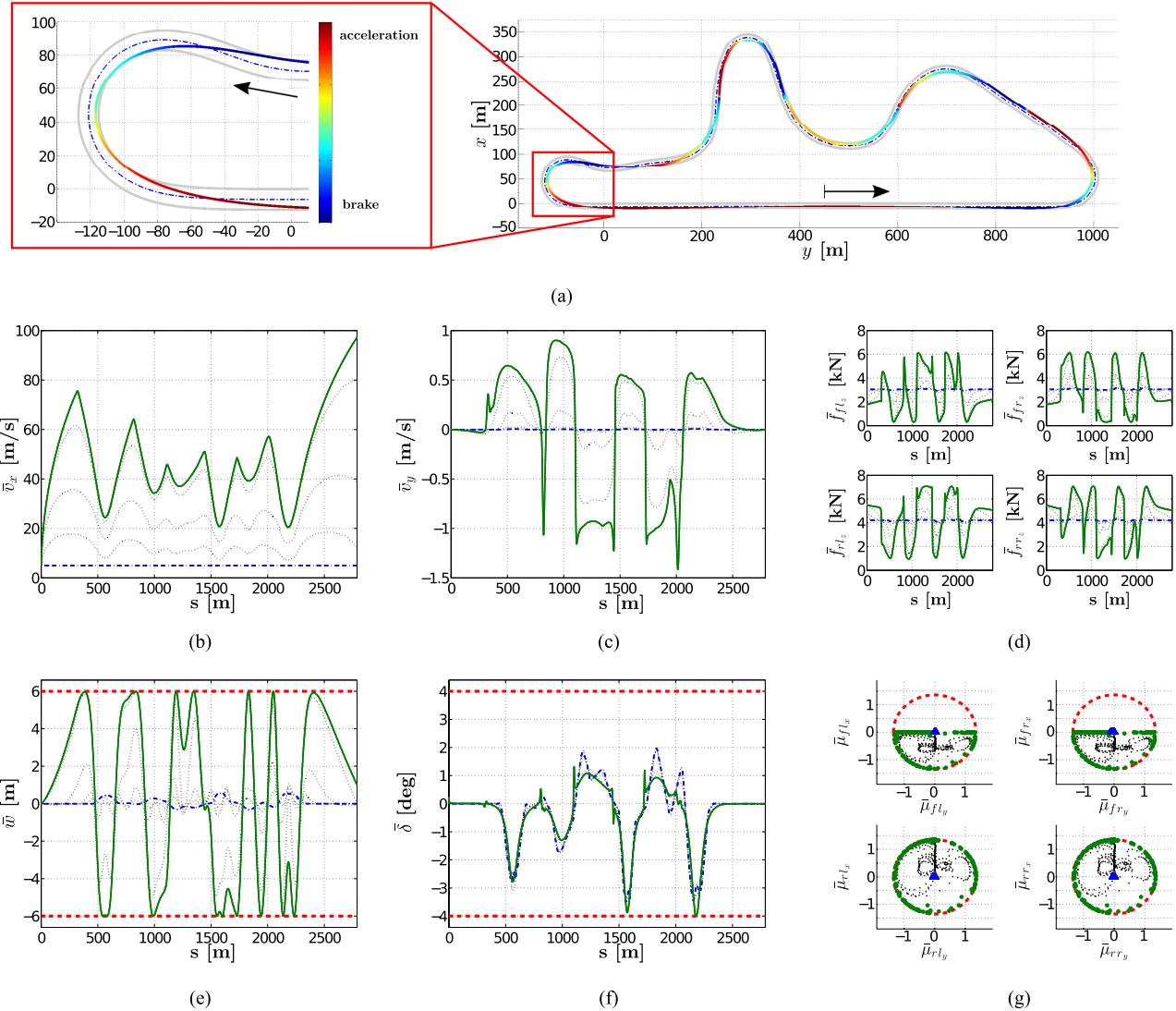


Fig. 7. Minimum-time trajectory for a real testing track. The initial (dashed-dot line), the intermediate (light dotted line), and the minimum-time (solid-line) trajectories are shown. Constraints are in dashed line. (a) Path (x, y). (b) Longitudinal velocity \bar{v}_x . (c) Lateral velocity \bar{v}_y . (d) Normal forces \bar{f}_{ijz} . (e) Lateral displacement \bar{w} . (f) Steer angle $\bar{\delta}$. (g) Ellipse constraints $\bar{\mu}_{ijx}$ versus $\bar{\mu}_{ijy}$.

TABLE III
REAL TESTING TRACK: TRAVELING TIME, INDEXES OF ACCURACY, AND COMPUTATIONAL TIME

Values for ϵ and ν	1	1.6e-01	2.7e-02	4.6e-03	7.7e-04	1.2e-04	2e-05	3e-06
traveling time [s]	230.56	162.07	113.98	84.66	70.22	65.49	64.55	64.14
apex point distance [m]	5.13	4.8	3.74	1.98	0.38	0.01	4e-04	2e-05
kissing point distance [m]	25.58	5.26	4.39	2.35	0.62	0.04	1e-03	1e-04
computational time [s]	206	128	127	234	288	251	434	341

to the learning process (continuation approach of the strategy), the expert driver discovers a new driving technique: pendulum turn. First, it steers the vehicle inward the turn and then, in the mid turn, uses the counter-steering. This computation confirms that the drifting maneuver is a (candidate) minimum-time trajectory for rally cars.

B. Minimum-Time Trajectory for a Real Testing Track

As an application on a more realistic scenario, the minimum-time trajectory for a real testing track is

computed.¹ The testing track was also used in [27] for dynamics exploration. The track is a planar circuit and consists of straights, turns, a chicane, and a hairpin turn. The track length is $S = 2789.2\text{m}$ and its width is 12m. In Fig. 7, the computed minimum-time trajectory is shown. For clarity of presentation, only a few intermediate optimal trajectories are shown. In Table III, we provide the traveling time, the indexes of accuracy, and the associated CPU time. The

¹See <http://www.nardotechnicalcenter.com/> for details on the track.

overall computational time is under 35 min with 251 000 optimization variables (the space discretization of the track is 0.1 m).

Once again, by setting as initial longitudinal velocity $v_{x0} = 5$ m/s, the initial trajectory is feasible, see the dash-dotted blue lines in Fig. 7(e)–(g). We highlight the last (aggressive hairpin) turn of the track, see the zoomed-in view of Fig. 7(a). When entering the turn, the vehicle moves to the right of the track to reduce the path curvature. Then, to generate the required lateral forces in the turn the tires have a high lateral velocity. Finally, the car starts to exit the turn with the maximum traction force. Now, consider the lateral displacement, see Fig. 7(e). An interesting feature appears: the vehicle hits two apex points. First, the vehicle turns into the (first) apex point at $s = 2145$ m. Then, it drifts toward the middle of the road: it reaches the maximum distance from the track boundary, the minimum longitudinal velocity value, and the maximum steer angle value [Fig. 7(b), (e), and (f)], respectively. Then, the vehicle hits the (second) apex point, $s = 2333$ m, and starts the exit from the turn: it approaches the kissing point and accelerates with the maximum traction force (as performed by race drivers). Notice that this behavior is different from the one obtained when trying to travel the turn at a high constant velocity, as observed in [27]. There the optimal trajectory has a single apex point.

Next, looking at the global picture of the minimum-time trajectory, we note that the longitudinal velocity has (local) *minimum* values at the same points of the track as the (local) *maximum* absolute values of the steer angle. Moreover, we note the relationship between the (longitudinal and lateral) load transfer and the (longitudinal and lateral) velocity. When the vehicle enters left turns, the lateral velocity is positive and the load shifts to the (front and rear) right wheels. We observe a space-delay of the maximum peak between the front and rear normal forces. This is due to the longitudinal load transfer induced by the strong braking force. A similar behavior is observed for the right turns. Once again, we highlight: 1) the high accuracy in computing the apex and kissing points (Table III); 2) the rapid load transfer [Fig. 7(d)]; and 3) the tire forces working on the ellipse boundary [Fig. 7(g)].

Finally, we show how the (longitudinal and lateral) load transfer affects the minimum-time trajectory. We compute the minimum-time trajectory for the rigid car model without load transfer, called two-track model, and we compare it with respect to the RigidCar minimum-time trajectory for $d = 0.2$ m and $d = -0.2$ m. Notice that such a comparison allows us to strengthen the importance of the load transfer effect in the optimal solution.

In Fig. 8, the minimum-time trajectories are shown. This computation highlights four important features. First, the driver (i.e., the optimization routines) exploits the load transfer to increase the maximum acceleration and braking capabilities. Even if the velocity profile may seem similar at a first look, they present a significant difference in the braking points and in the braking and acceleration rates [Fig. 8(a)]. The different rate of acceleration can be explained in the following way. When the rear-wheel drive transmission vehicle accelerates,

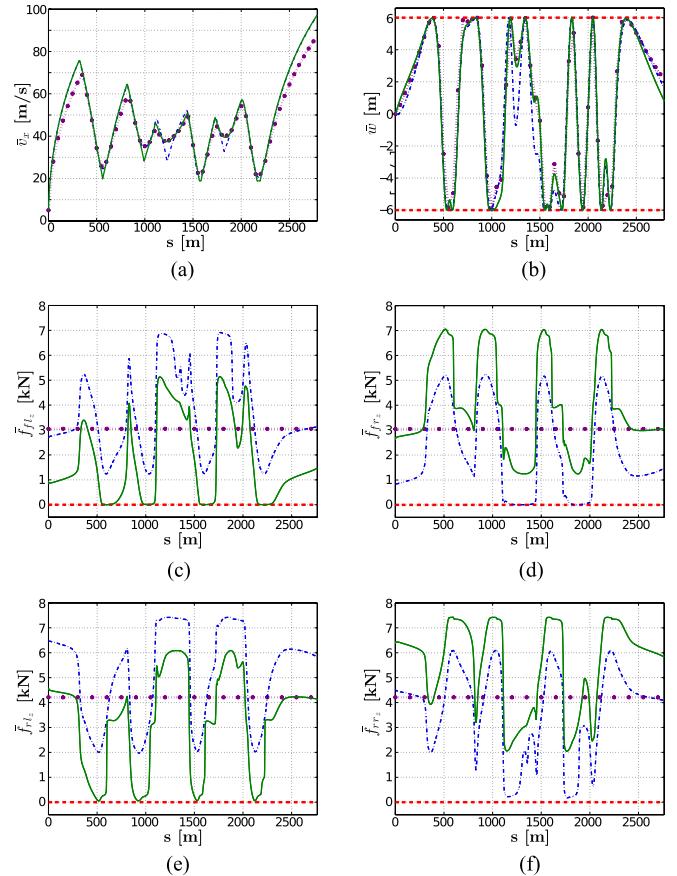


Fig. 8. Minimum-time trajectory for a real testing track. Comparison among the two-track car model, no load transfer, (dotted magenta line), RigidCar with $d = -0.2$ m (dashed-dotted blue line), and RigidCar with $d = 0.2$ m (solid green line). (a) Longitudinal velocity \bar{v}_{x_1} . (b) Lateral displacement \bar{w} . (c) Normal forces \bar{f}_{flz} . (d) Normal forces \bar{f}_{frz} . (e) Normal forces \bar{f}_{rlz} . (f) Normal forces \bar{f}_{rrz} .

the rear tire forces saturate. However, due to the effect of the longitudinal load transfer, the weight of the RigidCar shifts to the rear axle increasing the longitudinal traction force. Similarly, the hard braking increases the front normal load, thus increasing the longitudinal brake force. Clearly, this phenomenon (i.e., the longitudinal load transfer) is not captured at all by the two-track car model without load transfer. Notice that a comparison between the minimum-time trajectory of the single track rigid model with longitudinal load transfer and the bicycle model is provided in [1], which highlights the importance of the longitudinal load transfer effect. Second, the explicit formulation of the normal forces allows us to predict an important feature: the liftoff condition. Since for $d = 0.2$ m the (static) weight is shifted on the right side of the vehicle, when the vehicle enters the left turns, the normal force constraint of the front left wheel becomes suddenly active [solid green line in Fig. 8(c)]. The weight is shifted onto the rear right wheel causing the front left wheel almost to lift its weight from the ground. This behavior is reversed for $d = -0.2$ m, see dash-dotted blue line in Fig. 8(d). To regain the load on the wheel (or, in other words, to avoid rollover condition), the driver has to decrease the velocity [Fig. 8(a)]. Clearly, the behavior described above does not appear in the two-track model without load transfer, as well as in

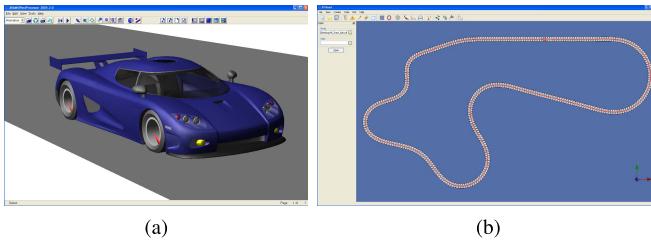


Fig. 9. VI-grade's simulation environment. (a) VI-CarRealTime model. (b) Virtual testing track (*VI_track_flat*).

the single-track model used in [1]. Third, RigidCar includes some parameters not present in the single-track model (e.g., the parameter d) affecting the minimum-time trajectory. It is interesting to note in Fig. 8(a) (see dashed-dotted blue and solid green lines) the different shape of the optimal velocity profile due to the different value of d . Finally, we note that the RigidCar minimum-time race line is affected by the two configurations: in Fig. 8(b), the lateral displacement is different, especially in the middle section of the track. This latter feature highlights the importance of computing the optimal race line for effectively solving the minimum-time problem (it is an important limitation of the quasi-steady-state methods).

C. Strategy Validation in a Virtual Prototyping Toolbox

We validate the MTGO strategy by integrating it into a commercial virtual prototyping toolbox. We show how the MTGO strategy predicts with high-fidelity minimum-time trajectories (including velocities, accelerations, and longitudinal, lateral, and normal tire forces) of a complex multibody vehicle model. We use VI-CarRealTime developed by VI-Grade GmbH, a vehicle dynamics software tool for motor-sport dynamic analysis, [Fig. 9(a)]. The VI-CarRealTime model (fully compatible with Adams/Car) is based on a fairly complex vehicle model which includes all the actual car subsystems, e.g., steering, front and rear suspensions, chassis, front and rear tires, powertrain, and brake system. The model includes 14 DOF (the vehicle chassis has 6 DOF, while the wheel parts have 2 DOF each), suspension and steering system properties are described by lookup tables, brakes, and powertrain subsystems are described using differential and algebraic equations, tire-ground interaction is modeled using a Pacejka magic formula tire model.

We compute a so-called flying lap trajectory, i.e., a minimum *lap-time* trajectory in which the (free) initial condition has to be equal to the (free) final condition, for the RigidCar model on the virtual testing circuit shown in Fig. 9(b), called *VI_track_flat*. The track length is $S = 1723$ m and its width is 10 m. The car parameters are based on those given in the VI-grade's simulation environment. The computed minimum-time path and velocity profiles are set as reference signal to the VI-CarRealTime driver. The VI-CarRealTime vehicle is driven by means of such a closed loop controller on the basis of the reference signal computed according to the MTGO strategy.

In Fig. 10, we show a comparison between the minimum-time trajectory predicted by the MTGO strategy on the RigidCar model and the one obtained when the VI-CarRealTime model is driven using that reference signal.

The predicted minimum traveling time is 53.15 s and the one obtained by driving the VI-CarRealTime model is 53.21 s. We observe a good trajectory matching. Except for transients due to the suspension and gear shifting, e.g., the velocity profile during upshift phase at about $s = 1130$ m in Fig. 10(b) and (d), the RigidCar model predicts position, velocity, and the lateral acceleration of the VI-CarRealTime model [Fig. 10(a)–(c)]. Moreover, we highlight that the RigidCar model predicts the longitudinal, lateral, and normal tire forces of the VI-CarRealTime model [Fig. 10(d)–(f)]. These results confirm that the MTGO strategy (combined with the RigidCar model) allows one to compute meaningful reference signals which can be used to drive a complex vehicle model, such as the VI-CarRealTime model, at the maximum of its capabilities.

Remark 4.1: In Fig. 10, some differences between the RigidCar and the VI-CarRealTime trajectories can be observed at about $s = 700$ m and $s = 980$ m. In these segments of the circuit, the driver (i.e., the closed-loop controller of VI-CarRealTime) seems to have some problem in driving the vehicle along the predicted path at the predicted velocity profile, so that the vehicle slightly loses traction (it drifts and seems not to use its full potential). This is confirmed by the lateral displacement and lateral acceleration profiles in Fig. 10(a) and (c). We believe that by designing a high-performance driver, based on the full state-input minimum-time trajectory (predicted by the MTGO strategy), the complex vehicle model could be driven in a more effective way. \square

D. Discussion

The results show that we are able to capture interesting behaviors of minimum-time trajectories with a low computational effort. We want to highlight some meaningful differences with respect to the existing literature.

First, it is interesting to note that a pendulum turn maneuver appears as the result not only of a velocity maximization, as shown in [7], but also of a time minimization as we show in our computations. Although the two cost functions are strictly related, it is interesting that this maneuver appears in these two different scenarios.

Second, in [14], the pendulum turn trajectory of a single-track car model is shown for a 180° turn by considering an extreme off-road setup. It seems that the car does not approach the kissing points, thus showing that the approach used in computing the optimal trajectory is missing some important features of the minimum-time trajectory. However, this behavior could be due to the different structure of the track (in [14] a straight section of 30 m is considered) and further investigations would be needed.

Finally, in [9], the minimum-time trajectory for a double lane change track is computed. The authors stated that the (average) computational time was 2 h. The track length was 300 m, with a space discretization of 12 m, and a two-track car model with 7 DOF was considered. Clearly, part of the high computational effort was mainly due to the limited performance of the processor used (this paper was published

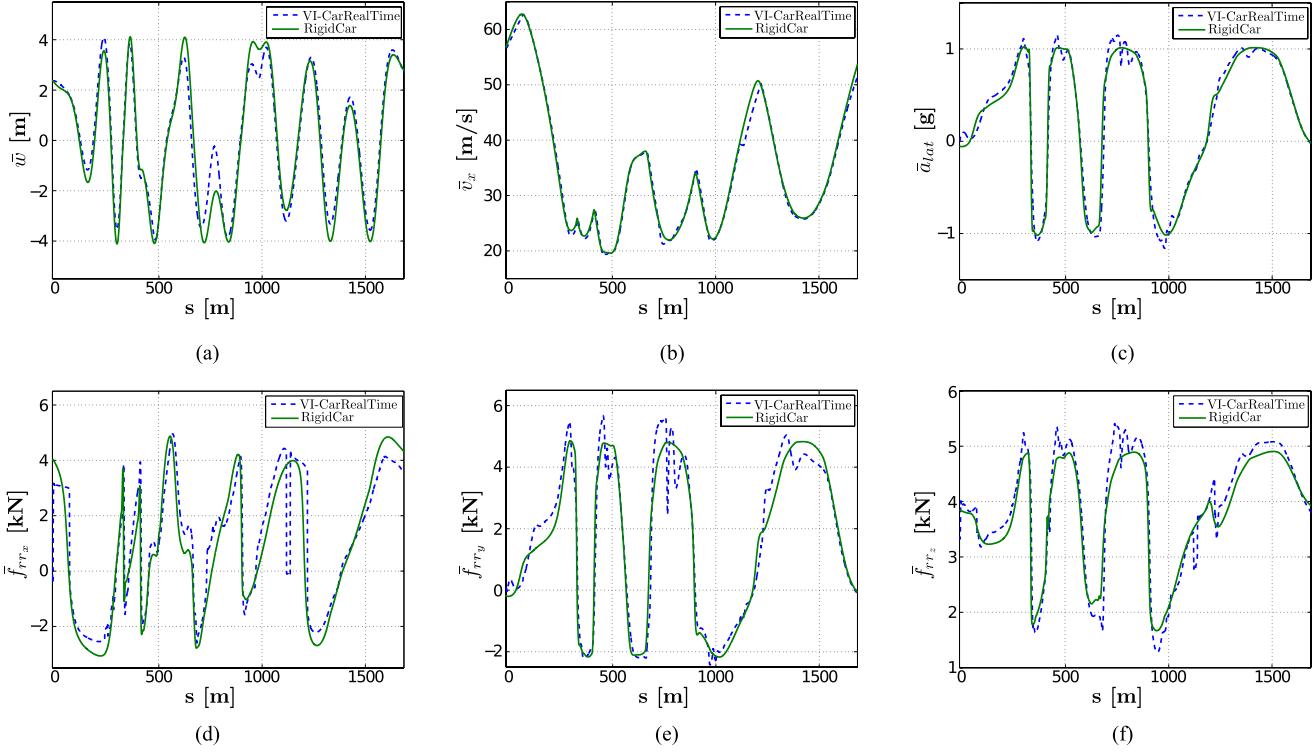


Fig. 10. Virtual testing circuit. Comparison between the minimum-time trajectory generated by MTGO (solid green line) and the trajectory obtained by using VI-CarRealTime model (dashed blue line). The path, longitudinal velocity, lateral acceleration, longitudinal, lateral, and normal tire forces profiles are shown. Due to space limitation, only the rear right tire forces are shown. (a) Lateral displacement \bar{w} . (b) Longitudinal velocity \bar{v}_x . (c) Lateral acceleration \bar{a}_{lat} . (d) Longitudinal tire forces \bar{f}_{rrx} . (e) Lateral tire forces \bar{f}_{rry} . (f) Normal tire forces \bar{f}_{rrz} .

10 years ago). For the numerical computations shown in this paper, we consider a 90° maneuver of 462 m and a real testing track of 2789 m. For both the paths the space discretization of 0.1 m is used. The overall computational time for the 90° scenario is under 8 min (Table II) and for the real testing track scenario is under 35 min (Table III).

To conclude, from these comparisons, our strategy provides a meaningful improvement to compute minimum-time trajectories of two-track car models.

V. CONCLUSION

In this paper, we addressed the minimum-time problem for a two-track rigid car that considers tire models and load transfer. We provided an optimization strategy to estimate the minimum-time trajectory on a given track. This strategy is based on an equivalent formulation of the optimal control problem and combines optimal control techniques with a continuation method to find a (candidate) minimum-time trajectory. The strategy allows us to capture important dynamic aspects of the vehicle with a low computational effort. Thus, the performance of the MTGO strategy appears to be far promising than those of shooting algorithms. This fact was proved on a simple test case, 90° maneuver, and on a real testing track. Moreover, the minimum-time path and velocity profiles (computed for a standard virtual testing circuit) were used as reference trajectories for a high-fidelity vehicle simulator. The combination of the MTGO strategy with the RigidCar model has been found to perform well in the presence of unmodeled dynamics, such as that due to the suspension system and gear shifting.

APPENDIX A CAR MODEL PARAMETERS

The car parameters are the ones given in [23]. Large rear-wheel drive sports car

$$\begin{aligned} m &= 1480[\text{kg}], \quad a = 1.421[\text{m}], \quad b = 1.029[\text{m}], \quad h = 0.42[\text{m}] \\ d_f &= 0.751[\text{m}], \quad d_r = 0.789[\text{m}], \quad d = 0.0[\text{m}] \\ K_t &= 0, \quad K_b = 0.5 \\ I_b &= \begin{bmatrix} 590 & 0 & -50 \\ 0 & 1730 & 0 \\ -50 & 0 & 1950 \end{bmatrix}. \end{aligned}$$

We set the following, somewhat reasonable, maximum front wheel steer angle and steer rate:

$$\delta_{\max} = 4[\text{deg}], \quad u_{\delta\max} = 20[\text{deg/s}]$$

and the following minimum and maximum normal tire force:

$$f_Z_{\min} = 0[\text{N}], \quad f_Z_{\max} = mg[\text{N}].$$

The cornering stiffness and the peak friction of the tire are

$$\begin{aligned} C_f &= 62[\text{rad}^{-1}], \quad C_r = 52[\text{rad}^{-1}] \\ d_{X\max} &= 1.355, \quad d_{Y\max} = 1.355. \end{aligned}$$

For the off-road scenario, they are

$$\begin{aligned} C_f &= 4[\text{rad}^{-1}], \quad C_r = 3[\text{rad}^{-1}] \\ d_{X\max} &= 0.677, \quad d_{Y\max} = 0.677. \end{aligned}$$

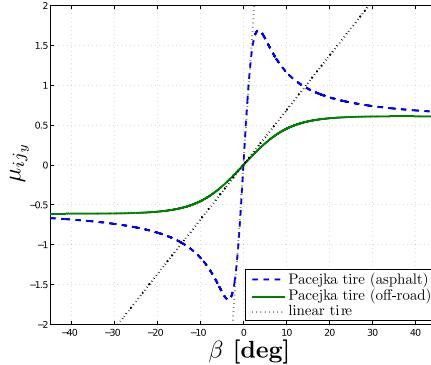


Fig. 11. Lateral force coefficient is plotted as a function of lateral slips. Dashed line: linear approximation.

The qualitative trend of the lateral force coefficients, as function of lateral slips, is reported in Fig. 11.

APPENDIX B PROJECTION OPERATOR-BASED NEWTON METHOD

The PRONTO [18] allows us to address the optimal control problem (13). The problem is equivalent to

$$\min_{\xi \in \mathcal{T}} h_0(\xi; \bar{\sigma}_{cl}) + \epsilon b_v(\xi) \quad (14)$$

where \mathcal{T} denotes the manifold of bounded trajectories $(\tilde{q}_s(\cdot), \tilde{u}(\cdot))$ on $[0, S]$. Using the projection operator \mathcal{P} , (11), to locally parametrize the trajectory manifold, the *constrained* optimization problem (14) is converted into the following *unconstrained* optimization problem:

$$\min_{\xi} h_0(\mathcal{P}(\xi); \bar{\sigma}_{cl}) + \epsilon b_v(\mathcal{P}(\xi)). \quad (15)$$

The constrained and unconstrained optimization problems are equivalent in the sense that if a trajectory ξ^+ is a (unconstrained) local minimum of (15), then $\xi^* = \mathcal{P}(\xi^+)$ is a (constrained) local minimum of (14), see [18]. Keeping in mind the usual Newton (or quasi-Newton) method in finite dimensions, a descent method for trajectory optimization is used for the minimization of the cost functional

$$\tilde{h}(\mathcal{P}(\xi); \bar{\sigma}_{cl}) = h_0(\mathcal{P}(\xi); \bar{\sigma}_{cl}) + \epsilon b_v(\mathcal{P}(\xi)).$$

Minimization of the cost functional \tilde{h} is accomplished by iterating over the Algorithm 2, where ξ_i indicates the current trajectory, ξ_0 an initial trajectory, and $D\tilde{h}(\mathcal{P}(\xi_i); \bar{\sigma}_{cl}) \cdot \zeta$ and $D^2\tilde{h}(\mathcal{P}(\xi_i); \bar{\sigma}_{cl}) \cdot (\zeta, \zeta)$ are, respectively, the first and second Fréchet derivatives of the functional $\tilde{h}(\mathcal{P}(\xi); \bar{\sigma}_{cl})$ at ξ_i in the direction ζ . The search direction ζ_i is obtained by solving a linear quadratic optimal control problem, where the functional to be minimized, $D\tilde{h}(\mathcal{P}(\xi_i); \bar{\sigma}_{cl}) \cdot \zeta + 1/2D^2\tilde{h}(\mathcal{P}(\xi_i); \bar{\sigma}_{cl}) \cdot (\zeta, \zeta)$, is the quadratic model functional given by the first two terms of the Taylor expansion of the functional $\tilde{h}(\mathcal{P}(\xi_i + \zeta))$ with respect to ζ . A new trajectory ξ_{i+1} is computed by combining the search direction ζ_i and the step size γ_i (obtained through the standard backtracking line search).

It is worth noting that, due to the projection operator, at each iteration of the PRONTO algorithm a system trajectory (i.e., a curve satisfying the dynamics) is available.

Algorithm 2 Projection Operator Newton Method, PRONTO

Given: initial trajectory $\xi_0 \in \mathcal{T}$

for $i = 0, 1, 2 \dots$ **do**

design K defining \mathcal{P} about ξ_i

search for descent direction

$$\zeta_i = \arg \min_{\zeta \in T_{\xi_i} \mathcal{T}} D\tilde{h}(\mathcal{P}(\xi_i); \bar{\sigma}_{cl}) \cdot \zeta + \frac{1}{2} D^2\tilde{h}(\mathcal{P}(\xi_i); \bar{\sigma}_{cl}) \cdot (\zeta, \zeta)$$

step size $\gamma_i = \arg \min_{\gamma \in (0, 1]} \tilde{h}(\mathcal{P}(\xi_i + \gamma \zeta_i); \bar{\sigma}_{cl})$;
project $\xi_{i+1} = \mathcal{P}(\xi_i + \gamma_i \zeta_i)$.

end for

REFERENCES

- [1] A. Rucco, G. Notarstefano, and J. Hauser, "Computing minimum lap-time trajectories for a single-track car with load transfer," in *Proc. IEEE Conf. Decision Control*, Maui, HI, USA, Dec. 2012, pp. 6321–6326.
- [2] R. S. Sharp and H. Peng, "Vehicle dynamics applications of optimal control theory," *Veh. Syst. Dyn.*, vol. 49, no. 7, pp. 1073–1111, Jun. 2011.
- [3] W. F. Milliken and D. L. Milliken, *Race Car Vehicle Dynamics*. Warrendale, PA, USA: SAE, 1995.
- [4] D. L. Brayshaw and M. F. Harrison, "A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location," *J. Automobile Eng.*, vol. 219, no. 6, pp. 725–739, Jun. 2005.
- [5] D. P. Kelly, "Lap time simulation with transient vehicle and tyre dynamics," Ph.D. dissertation, School Eng., Automotive Studies Group, Cranfield Univ., Cranfield, U.K., 2008.
- [6] J. P. Timings and D. J. Cole, "Minimum maneuver time calculation using convex optimization," *J. Dyn. Syst., Meas., Control*, vol. 135, no. 3, p. 031015, Mar. 2013.
- [7] E. Velenis and P. Tsotras, "Minimum time vs maximum exit velocity path optimization during cornering," in *Proc. IEEE Int. Symp. Ind. Electron.*, Dubrovnik, Croatia, Jun. 2005, pp. 355–360.
- [8] J. P. M. Hendrikx, T. J. J. Meijlink, and R. F. C. Kriens, "Application of optimal control theory to inverse simulation of car handling," *Veh. Syst. Dyn.*, vol. 26, no. 6, pp. 449–461, 1996.
- [9] D. Casanova, R. S. Sharp, and P. Symonds, "Minimum time manoeuvring: The significance of yaw inertia," *Veh. Syst. Dyn.*, vol. 34, no. 2, pp. 77–115, 2000.
- [10] E. Velenis and P. Tsotras, "Optimal velocity profile generation for given acceleration limits: Theoretical analysis," in *Proc. Amer. Control Conf.*, Portland, OR, USA, Jun. 2005, pp. 1478–1483.
- [11] E. Velenis and P. Tsotras, "Optimal velocity profile generation for given acceleration limits: the half-car model case," in *Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, Dubrovnik, Croatia, Jun. 2005, pp. 361–366.
- [12] C. Kirches, S. Sager, H. G. Bock, and J. P. Schlöder, "Time-optimal control of automobile test drives with gear shifts," *Optim. Control Appl. Methods*, vol. 31, no. 2, pp. 137–153, Mar./Apr. 2010.
- [13] S. Sager, C. Kirches, and H. G. Bock, "Fast solution of periodic optimal control problems in automobile test-driving with gear shifts," in *Proc. IEEE Conf. Decision Control*, Dec. 2008, pp. 1563–1568.
- [14] D. Tavernini, M. Massaro, E. Velenis, D. I. Katzourakis, and R. Lot, "Minimum time cornering: The effect of road surface and car transmission layout," *Veh. Syst. Dyn.*, vol. 51, no. 10, pp. 1533–1547, 2013.
- [15] D. Tavernini, E. Velenis, R. Lot, and M. Massaro, "On the optimality of handbrake cornering," in *Proc. IEEE 52nd Conf. Decision Control*, Dec. 2013, pp. 2233–2238.
- [16] B. Siegler, A. Deakin, and D. Crolla, "Lap time simulation: Comparison of steady state, quasi-static and transient racing car cornering strategies," SAE Technical Paper, 2000, pp. 247–254.
- [17] J. Hauser and A. Saccon, "A barrier function method for the optimization of trajectory functionals with constraints," in *Proc. IEEE Conf. Decision Control*, Dec. 2006, pp. 864–869.
- [18] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," in *Proc. IFAC World Congr.*, Barcelona, Spain, 2002, p. 310.

- [19] J. M. Maciejowski, *Predictive Control: With Constraints*. Upper Saddle River, NJ, USA: Pearson Education, 2002.
- [20] M. J. Spenko, "Hazard avoidance for high-speed rough-terrain unmanned ground vehicles," Ph.D. dissertation, Dept. Mech. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 2005.
- [21] A. Rucco, G. Notarstefano, and J. Hauser, "Development and numerical validation of a reduced-order two-track car model," *Eur. J. Control*, vol. 20, no. 4, pp. 163–171, Jul. 2014.
- [22] S. Di Cairano, H. E. Tseng, D. Bernardini, and A. Bemporad, "Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 4, pp. 1236–1248, Jul. 2013.
- [23] G. Genta, *Motor Vehicle Dynamics: Modeling and Simulation*. Singapore: World Scientific, 2006.
- [24] A. Jadbabaie, J. Yu, and J. Hauser, "Unconstrained receding-horizon control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 46, no. 5, pp. 776–783, May 2001.
- [25] A. Rucco, J. Hauser, and G. Notarstefano, "Non-existence of minimizing trajectories for steer-braking systems," in *Proc. IFAC Symp. Nonlinear Control Syst.*, Toulouse, France, Sep. 2013, pp. 205–210.
- [26] E. Velenis, P. Tsiotras, and J. Lu, "Modeling aggressive maneuvers on loose surfaces: The cases of trail-braking and pendulum-turn," in *Proc. Eur. Control Conf.*, Kos, Greece, Jul. 2007, pp. 1233–1240.
- [27] A. Rucco, G. Notarstefano, and J. Hauser, "Optimal control based dynamics exploration of a rigid car with longitudinal load transfer," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1070–1077, Aug. 2013.



Alessandro Rucco received the master's degree in automation engineering and the Ph.D. degree in information engineering from the Università del Salento, Lecce, Italy, in 2009 and 2014, respectively.

He was a Visiting Student with Supélec, University of Paris-Sud, Gif-sur-Yvette, France, in 2010, and the University of Colorado Boulder, Boulder, CO, USA, in 2012. He was a Team Leader of the VI-RTUS Team that won the International Student Competition Virtual Formula in 2012. From 2013 to 2014, he was a Research Fellow with the Department of Innovation Engineering, Università del Salento. He is currently a Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, Porto, Portugal, where he is involved in motion planning and guidance of autonomous unmanned aerial vehicles. His current research interests include nonlinear optimal control, modeling, trajectory optimization, and high-performance maneuvering for car and aerial vehicles.



Giuseppe Notarstefano received the Laurea degree (*summa cum laude*) in electronics engineering from the Università di Pisa, Pisa, Italy, in 2003, and the Ph.D. degree in automation and operation research from the Università di Padova, Padua, Italy, in 2007.

He was a Visiting Scholar with the University of Stuttgart, Stuttgart, Germany, in 2009 and 2013, the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2005 and 2009, and the University of Colorado Boulder, Boulder, CO, USA, in 2006. He has been an Assistant Professor (Ricercatore) with the Università del Salento, Lecce, Italy, since 2007. He coordinated the VI-RTUS Team that won the International Student Competition Virtual Formula in 2012. His current research interests include distributed optimization, motion coordination in multi-agent networks, applied nonlinear optimal control, and modeling, trajectory optimization, and aggressive maneuvering of aerial and car vehicles.

Dr. Notarstefano serves as an Associate Editor on the Conference Editorial Board of the IEEE Control Systems Society, the European Control Conference, the IFAC World Congress, and the IEEE Multi-Conference on Systems and Control.



John Hauser received the B.S. degree from the United States Air Force Academy, Colorado Springs, CO, USA, in 1980, and the M.S. and Ph.D. degrees from the University of California at Berkeley, Berkeley, CA, USA, in 1986 and 1989, respectively.

He was the Fred O'Green Assistant Professor with the University of Southern California, Los Angeles, CA, USA, from 1989 to 1992. Since 1992, he has been with the Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, Boulder, CO, USA. He has held visiting positions with the Instituto Superior Técnico, Lisbon, Portugal, the University of Stuttgart, Stuttgart, Germany, the University of Padova, Padua, Italy, the California Institute of Technology, Pasadena, CA, USA, the Lund Institute of Technology, Lund, Sweden, and École Supérieure d'Électricité, Paris, France. His recent work has focused on the development of optimization and optimal control tools and techniques for trajectory exploration with an eye toward characterizing the trajectory space of highly maneuverable nonlinear systems. His current research interests include nonlinear dynamics and control, optimization and optimal control, aggressive maneuvering for high-performance motorcycles and aircraft, and dynamic visualization.

Prof. Hauser was a recipient of the Presidential Young Investigator Award from the National Science Foundation in 1991.