To appear in *Vehicle System Dynamics*
Vol. 00, No. 00, Month 20XX, 1–17

Three-dimensional fixed-trajectory approaches to the minimum-lap time of road vehicles

S. Lovato and M. Massaro*

Department of Industrial Engineering, University of Padova,
Via Venezia 1, 35131 Padova, Italy

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

Minimum-lap-time simulations with quasi-steady-state models and predefined trajectory have been in use for many years. However, most of the published works deal with two-dimensional roads and employ the so-called 'apex-finding' method. This paper focuses on the application to three-dimensional roads, both with an extension of the 'apex-finding' approach to three-dimensional scenarios and with an optimal control method. Both approaches are based on g–g–g diagrams, i.e. the three-dimensional extension of the well-known g–g maps. In addition, under the assumption that the predefined trajectory is determined from noisy data (e.g. logged from the real vehicle), the three-dimensional trajectory reconstruction problem is addressed, to find a smooth and drift-free racing line to be used in the minimum-time simulation — again an optimal control approach is employed for the trajectory reconstruction. Examples of application are given both for a car and a motorcycle.

Keywords: minimum-lap time; three-dimensional road; fixed trajectory; g–g–g map; cars; motorcycles; apex-finding.

1. Introduction

Minimum lap-time simulations (MLTS) of road vehicles have been in use for many years [1, 2]: they were historically formulated using quasi-steady-state (QSS) vehicle models on pre-defined two-dimensional trajectories [3–10], and evolved to employ transient vehicle models on three-dimensional roads [11–15], where the trajectory is a result of the optimization. Free-trajectory methods with QSS models have also been proposed both for two-dimensional [16] and three-dimensional roads [17, 18].

Although the approaches based on transient vehicle models with free trajectory have been shown capable of dealing with relatively sophisticated models and simulation scenarios including e.g. thermal tyre models [19, 20], variable tyre-road friction coefficients [21], aerodynamic maps dependent on vehicle trim [22], vehicles with electric motors and torque vectoring [23, 24], drifting and handbrake manoeuvring [25, 26], they nevertheless require computation times that are usually roughly one or two order of magnitude larger than those relate to QSS models on a fixed, i.e. pre-defined, trajectory. This is one of the reasons why the latter approach remains popular. Another reason is that QSS models are generally easier to implement.

The most common solution technique for MLTS with a QSS vehicle model on a fixed

*Corresponding author. Email: matteo.massaro@unipd.it trajectory is the so called ‘apex-finding’ method, which basically consists in i) finding the apexes of the trajectory (i.e. points where the curvature has peaks), ii) assuming that the lateral acceleration of the vehicle is maximum at such apexes, and iii) assuming braking before the apex (i.e. into the turn) and acceleration after the apex (i.e. out of the turn). The accelerations of the vehicle are constrained to remain on the boundaries of its g–g map, which is obtained from the solution of the QSS equations of motion and defines the performance envelope of the vehicle. Apparently, there is no documented literature for the application of the method on three-dimensional roads, i.e. with a three-dimensional racing-line trajectory.

The first contribution of this paper is to devise an ‘apex-finding’ method for the MLTS of road vehicles on three-dimensional roads with pre-defined racing line building upon g–g–g diagrams [17] (which are an extension of the classic g–g diagrams)—only two-dimensional apex-finding approaches are documented in the literature. The second contribution is the solution of the MLTS with an optimal control approach, again based on the g–g–g maps, which is alternative to the apex-finding approach. The third contribution is related to the trajectory reconstruction problem on three-dimensional roads (most methods discussed in the literature are two-dimensional). Indeed, the fixed-trajectory approaches need be fed with the racing-line trajectory. This can be obtained numerically, e.g. from the solution of a free-trajectory MLTS (any of those referenced above) or combining the minimum-distance and maximum-curvature trajectories [27–29] or experimentally, e.g. by direct measurement of the racing line on the track or by estimation from the sensors mounted on the vehicle [30]. A method based on optimal control is devised, which is similar to the one used to reconstruct the road.

An example of application is given both for a race car and a racing motorcycle, since the methods proposed are suitable for application on both vehicles. The Mugello circuit is considered, because its three-dimensional geometry is available from the literature [14, 15]. The g–g–g diagrams are generated using the semi-analytical models in [17], in order to make it simpler to reproduce the results presented. However, such maps can be derived from more complex models or even experimentally — all the complexity of the model is embedded in such diagrams and thus the methods proposed are insensitive to the model complexity.

The work is organised as follows. In Sec. 2 the basics of the three-dimensional road modelling and vehicle positioning using curvilinear coordinates are recalled, together with the derivation of the trajectory curvature (to be used in the apex-finding method). In Sec. 3 the racing-line reconstruction problem is formulated as an optimal control problem (OCP). In Sec. 4 the three-dimensional minimum-time problem is formulated, both using a three-dimensional apex-finding approach and an OCP approach. Finally, in Sec. 5 the examples of application are given, including both the racing line reconstruction from noisy data and the MLTS on such reconstructed trajectory with a race car and a race motorcycle.

2. Road modelling

Road modelling and vehicle positioning using curvilinear coordinates have been already discussed in the literature in a number of places, including [2, 11, 18, 31]. Therefore, only the fundamentals relationships are recalled below.

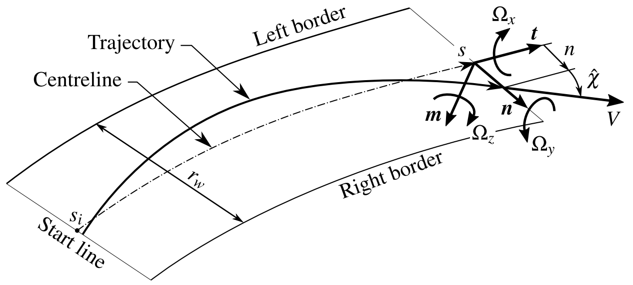

Figure 1.: Three-dimensional road modelling and vehicle positioning. The Darboux frame has basis vectors $\boldsymbol{t}, \boldsymbol{n}, \boldsymbol{m}$ and angular rate $\Omega_x, \Omega_y, \Omega_z$. The vehicle trajectory is described by the lateral position n from the road centreline and relative orientation $\hat{\chi}$ of the velocity V with respect to the tangent to the centreline.

2.1. *Road kinematics*

The road centreline is represented by a space curve, from which the road plane is created by adding width r_w. A moving trihedron, called Darboux frame, moves along the road centreline remaining tangent to the road surface; see Fig. 1. Its orthogonal basis $[\boldsymbol{t}, \boldsymbol{n}, \boldsymbol{m}]$ can be given by the attitude-pitch-roll convention as follows

$$
\begin{aligned}
R(\theta, \mu, \phi) &= R_z(\theta) R_y(\mu) R_x(\phi) \qquad (1)\\
&= \begin{bmatrix} \cos\theta\cos\mu & -\sin\theta\cos\phi + \cos\theta\sin\mu\sin\phi & \sin\theta\sin\phi + \cos\theta\sin\mu\cos\phi \\ \sin\theta\cos\mu & \cos\theta\cos\phi + \sin\theta\sin\mu\sin\phi & -\cos\theta\sin\phi + \sin\theta\sin\mu\cos\phi \\ -\sin\mu & \cos\mu\sin\phi & \cos\mu\cos\phi \end{bmatrix},
\end{aligned}
$$

where $R_z(\cdot), R_y(\cdot), R_x(\cdot)$ are the elementary matrix around the corresponding (subscripted) axis, while θ, μ, ϕ are the attitude, slope, and banking angles respectively. All variables are a function of the travelled distance along the road centreline, i.e. the curvilinear coordinate s.

The angles θ, μ, ϕ are obtained by integration of the relative torsion Ω_x, normal curvature Ω_y, and geodesic curvature Ω_z, which are the angular rates (rad/m) of the Darboux frame expressed in the Darboux frame itself. The resulting relationships are obtained from the well-known Frenet-Serret formula

$$
\begin{bmatrix} \phi' \\ \mu' \\ \theta' \end{bmatrix} = \frac{1}{\cos\mu} \begin{bmatrix} 1 & \sin\mu\sin\phi & \cos\phi\sin\mu \\ 0 & \cos\mu\cos\phi & -\cos\mu\sin\phi \\ 0 & \sin\phi & \cos\mu \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}, \qquad (2)
$$

where the prime means derivative with respect to s. In practice, the angular rates $\Omega_x, \Omega_y, \Omega_z$ are estimated from the measured coordinates of the road borders, see e.g. [2, 11, 18, 31]. The position $[x_c, y_c, z_c]^T$ of the origin of the Darboux frame, expressed in the absolute frame, is obtained by integration of the tangent vector $\boldsymbol{t}$, i.e. the first column of (1)

$$
\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \int \boldsymbol{t} ds = \int \begin{bmatrix} \cos\theta\cos\mu \\ \sin\theta\cos\mu \\ -\sin\mu \end{bmatrix} ds. \qquad (3)
$$

2.2. *Vehicle positioning*

Once the road has been defined, the vehicle trajectory is described in terms of the lateral distance n from the road centreline and relative yaw orientation $\hat{\chi}$ with respect to the tangent to the centreline. It is also worth noting that $\hat{\chi}$ refers to a frame aligned with the tangent to the racing line, and thus not necessarily aligned with the vehicle axes (unless the vehicle slip angle is zero). Indeed, when using QSS approaches it is convenient to use a trajectory-fixed frame which follows the racing line [16, 17] — in this framework, the hat is used to denote variables related to the trajectory-fixed frame.

The lateral position n and the relative orientation $\hat{\chi}$ can be related to the velocity V along the trajectory as follows

$$n' = \frac{1}{\dot{s}} V \sin \hat{\chi}, \tag{4}$$

where $\dot{s}$ is the speed along the road centreline

$$\dot{s} = \frac{V \cos \hat{\chi}}{1 - n\Omega_z}. \tag{5}$$

Equation (4) constraints n and $\hat{\chi}$, while the speed along the centreline $\dot{s}$ is the link between the space- and the time-domain formulations, since $dx/dt = \dot{s}dx/ds$. Indeed, it is common to solve minimum-time problems in the space domain instead of the time domain [1].

The angular velocities (rad/s) of the trajectory-fixed frame (expressed in the trajectory-fixed frame itself) are computed from the road curvatures (rad/m) as follows

$$\hat{\omega}_x = (\Omega_x \cos \hat{\chi} + \Omega_y \sin \hat{\chi})\, \dot{s}, \tag{6}$$

$$\hat{\omega}_y = (\Omega_y \cos \hat{\chi} - \Omega_x \sin \hat{\chi})\, \dot{s}, \tag{7}$$

$$\hat{\omega}_z = \left(\hat{\chi}' + \Omega_z\right) \dot{s}, \tag{8}$$

Equations (6) and (7) are necessary when solving the equations of motion, while (8) gives $\hat{\omega}_z$ as a function of $\hat{\chi}'$. The accelerations of the vehicle reference point, expressed in the trajectory-fixed frame, are

$$\hat{a}_x = \dot{V} + w\hat{\omega}_y, \tag{9}$$

$$\hat{a}_y = V\hat{\omega}_z - w\hat{\omega}_x, \tag{10}$$

$$\hat{a}_z = \dot{w} - V\hat{\omega}_y, \tag{11}$$

where w is the absolute vertical velocity of the trajectory-fixed frame (along the trajectory-fixed frame itself), given by

$$w = n\Omega_x \dot{s}. \tag{12}$$

Note that the terms $w\hat{\omega}_y$ and $w\hat{\omega}_x$ in (9) and (10) are negligible under the common assumption of small road curvatures Ω_x and Ω_y.

Finally, the (geodesic) curvature Γ of the racing line is obtained from the ratio of $\hat{a}_y$

to V^2

$$\Gamma(s) = (\chi' + \Omega_z)\frac{\cos\hat{\chi}}{1 - n\Omega_z} - n\Omega_x(\Omega_x\cos\hat{\chi} + \Omega_y\sin\hat{\chi})\left(\frac{\cos\hat{\chi}}{1 - n\Omega_z}\right)^2, \tag{13}$$

where $\hat{a}_y$ is given by (10) using (5), (6), (8) and (12). The second term in (13) is negligible if the road curvatures Ω_x and Ω_y are small.

It is worth noting that the MLTS on two-dimensional roads can be carried out based on the racing line only, i.e. without any knowledge of the centreline. However, in the three-dimensional scenario also the road geometry is necessary, because the slope and banking (as well as their rates) affect the solution of the MLTS. Such geometry is conveniently expressed as a function of the travelled distance s along the centreline.

3. Racing-line reconstruction

The aim is to identify the lateral position n and relative orientation $\hat{\chi}$ of the vehicle trajectory, i.e. the racing line, from the noisy data x, y of the vehicle position, e.g. logged from a GNSS system. It is assumed that the three-dimensional road model corresponding to the road where the noisy data x, y have been logged is available, in terms of its centreline and orientation of the road tangent plane. No additional signals, e.g. those related to an inertial measurement unit (IMU), are assumed available, for simplicity reasons.

The problem is formulated under OCP framework, in a fashion similar to that used for the identification of the road model parameters from the noisy coordinates of the road borders, which has been addressed in a number of places, including [2, 11, 18, 31]. This time the error is on the racing line, rather than on road borders. In addition, the noisy x and y coordinates are a projection of the three-dimensional trajectory on the ground plane. A number of modifications on the approach are thus necessary.

The objective is to minimize the reconstruction error, while at the same time obtaining a smooth reconstructed trajectory. The OCP takes the following form

$$\underset{\boldsymbol{u}}{\text{minimise}} \quad \mathcal{J} = \int_{s_i}^{s_f} \big(S(\boldsymbol{x}) + U(\boldsymbol{u})\big)\, ds, \tag{14}$$

$$\text{subject to} \quad \boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \tag{15}$$

where s is the elapsed distance along the road centreline and $s_f - s_i = L$ is the track length. The performance index $\mathcal{J}$ has two terms: $S(\boldsymbol{x})$ and $U(\boldsymbol{u})$. The first aims at minimizing the reconstruction error, while the second limits the optimal controls to enforce a smooth solution—their expressions are given below. The function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ accounts for the differential constraints between the road and the trajectory.

The state vector $\boldsymbol{x}$ includes the lateral position n with respect to the centreline, the relative orientation $\hat{\chi}$ with respect to the tangent to the road centreline together with its derivative $\hat{\chi}'$, and the travelled distance η_0 along the noisy data

$$\boldsymbol{x}(s) = \left[n, \hat{\chi}, \hat{\chi}', \eta_0\right]^T, \tag{16}$$

while the travelled distance along the projection of the reconstructed trajectory on the ground plane x-y is η; see Fig. 2, where both η and η_0 are shown. The state $\eta_0(s)$ is

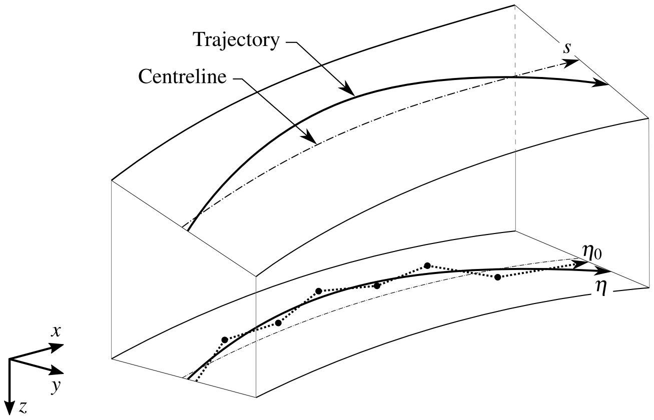

Figure 2.: Racing-line reconstruction with three-dimensional road (top) projected on the ground plane x-y (bottom). The reconstructed trajectory and the noisy data x, y are shown in solid line and dotted respectively. The travelled distance along the road centreline is s, the travelled distance along the projection of the reconstructed trajectory on the ground plane x-y is η, while the travelled distance along the noisy data is η_0.

necessary to evaluate the coordinates x and y of the noisy trajectory corresponding to a given position s along the centreline, and its final value $\eta_0(s_f)$ must be equal to the length η_{0f} of the noisy trajectory (obtained by numerical integration along the noisy data).

The OCP controls the rate of change (with respect to s) of the curvature $\hat{\omega}_z/\dot{s} = \hat{\chi}' + \Omega_z$ and rate of change of the noisy travelled distance η_0

$$\boldsymbol{u}(s) = [u_{\hat{\omega}z}, u_{\eta 0}]^T, \tag{17}$$

where

$$u_{\hat{\omega}z} = \frac{d}{ds}\left(\frac{\hat{\omega}_z}{\dot{s}}\right), \; u_{\eta 0} = \frac{\eta_0' - \eta'}{\eta'}. \tag{18}$$

The ‘stretching’ control $u_{\eta 0}$ allows to account for the difference between η_0 and η, in order to make it easier to satisfy the final value of η_0. This control becomes more important as the noise in the experimental data increases. It is chosen to control the derivative of $\hat{\omega}_z/\dot{s} = \hat{\chi}' + \Omega_z$ instead of $\hat{\chi}'$ because the geodesic curvature (13)—which needs be smooth for the successful computation of the MLTS—mainly depends on $\hat{\chi}' + \Omega_z$. Indeed, the speed profile in the MLTS is affected by the geodesic curvature of the trajectory, which needs be smooth in order to obtain a smooth speed profile.

The function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ in (15) is given by

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \left[\, (1 - n\Omega_z)\tan\hat{\chi}, \; \hat{\chi}', \; u_{\hat{\omega}z} - \Omega_z', \; \frac{1 - n\Omega_z}{\cos\hat{\chi}} c_\alpha (1 + u_{\eta 0}) \right]^T. \tag{19}$$

The first component is obtained from (4) with (5). The second and the third are the integrator chains for the relative orientation $\hat{\chi}$. The last component projects the rate of change of the travelled distance along the reconstructed trajectory on the ground plane.

Indeed, its three-dimensional components expressed in the ground frame are given by

$$R(\theta,\mu,\phi)R_z(\hat{\chi})\begin{bmatrix}V/\dot{s}\\0\\0\end{bmatrix}=\frac{1-n\Omega_z}{\cos\hat{\chi}}\begin{bmatrix}(\cos\mu\cos\hat{\chi}+\sin\mu\sin\phi\sin\hat{\chi})\cos\theta-\cos\phi\sin\hat{\chi}\sin\theta\\\cos\phi\sin\hat{\chi}\cos\theta+(\cos\mu\cos\hat{\chi}+\sin\mu\sin\phi\sin\hat{\chi})\sin\theta\\\cos\mu\sin\phi\sin\hat{\chi}-\sin\mu\cos\hat{\chi}\end{bmatrix}, \tag{20}$$

where $R(\theta,\mu,\phi)$ is from (1). The projection on the ground plane x-y is obtained from the norm of the first two components of (20)

$$\eta'=\frac{1-n\Omega_z}{\cos\hat{\chi}}c_\alpha, \tag{21}$$

where

$$c_\alpha=\sqrt{(\cos\mu\cos\hat{\chi}+\sin\mu\sin\phi\sin\hat{\chi})^2+(\cos\phi\sin\hat{\chi})^2}. \tag{22}$$

Note that $c_\alpha\approx 1$ under the often employed assumption of small slope μ and banking ϕ angles—however such assumption is not enforced here.

The state- and control-related terms in (14) are

$$U(\boldsymbol{u})=W_{\hat{\omega}z}u_{\hat{\omega}z}^2+W_{\eta 0}u_{\eta 0}^2, \tag{23}$$

$$S(\boldsymbol{x})=\left[x-x_0(\eta_0)\right]^2+\left[y-y_0(\eta_0)\right]^2, \tag{24}$$

with

$$x=x_c+(-\sin\theta\cos\phi+\cos\theta\sin\mu\sin\phi)n, \tag{25}$$

$$y=y_c+(\cos\theta\cos\phi+\sin\theta\sin\mu\sin\phi)n, \tag{26}$$

where $x_0(\eta_0), y_0(\eta_0)$ are interpolations of the measured coordinates of the vehicle position (expressed as a function of η_0), while $W_{\hat{\omega}z}, W_{\eta 0}$ are control weights. It is worth noting that the quantities $x_c, y_c, \theta, \mu, \phi, \Omega_z$ come from the road model — which is assumed available — as a function of s. Note also that it has been implicitly assumed, without loss of generality, that the weights related to x, y errors are unity.

Cyclic conditions are enforced in the case of closed trajectory

$$n(s_i)=n(s_f),\quad \hat{\chi}(s_i)=\hat{\chi}(s_f),\qquad \hat{\chi}'(s_i)=\hat{\chi}'(s_f), \tag{27}$$

while the final value of η_0 is enforced identical to that estimated from the noisy data

$$\eta_0(s_f)=\eta_{0f}. \tag{28}$$

4. Minimum-lap-time simulation

Two approaches for the solution of the MLTS on a three-dimensional track with given trajectory and QSS vehicle models are presented. The first is based on the application of the ‘apex-finding’ method to three-dimensional roads, while the second is based on the solution of an OCP. Both approaches build upon g–g–g diagrams, instead of the classic g–g.

4.1. *g-g-g diagrams*

In order to take into account the effects of road three-dimensionality, the classic g–g diagrams can be extended to g–g–g diagrams, to include the vertical acceleration in addition to the longitudinal and lateral accelerations. The approach introduced in [17] is employed. The fundamentals are recalled below.

Remaining within the g–g–g boundaries means satisfying the following condition

$$\tilde{\rho} \leq \tilde{\rho}_{\max}(\tilde{\alpha}, V, \tilde{g}), \tag{29}$$

with

$$\tilde{\rho} = \frac{1}{g}\sqrt{\tilde{a}_x^2 + \tilde{a}_y^2} \qquad \text{and} \qquad \tilde{\alpha} = \operatorname{arctan2}\left(\tilde{a}_x, \tilde{a}_y\right), \tag{30}$$

where $\tilde{\rho}$ is the engaged friction radius, $\tilde{\rho}_{\max}$ is the maximum friction radius (boundary of the g–g–g), $\tilde{\alpha}$ is the orientation of the engaged friction radius, V is the vehicle speed, g is the gravity acceleration, while $\tilde{a}_x$, $\tilde{a}_y$ and $\tilde{g}$ are the vehicle's apparent accelerations. These terms, in addition to the accelerations $\hat{a}_x$, $\hat{a}_y$, $\hat{a}_z$ in (9)–(11), also include the contributions of the gravitational effects related to the slope μ and banking ϕ of the road as well the the centrifugal effects related to the angular rates $\Omega_x, \Omega_y, \Omega_z$ of the road [17, 18]

$$\tilde{a}_x = \hat{a}_x + g\left(\sin\mu\cos\hat{\chi} - \cos\mu\sin\phi\sin\hat{\chi}\right), \tag{31}$$

$$\tilde{a}_y = \hat{a}_y - g\left(\cos\mu\sin\phi\cos\hat{\chi} + \sin\mu\sin\hat{\chi}\right), \tag{32}$$

$$\tilde{g} = -\hat{a}_z + g\cos\mu\cos\phi. \tag{33}$$

These effects make the vehicle 'heavier' ($\tilde{g}/g > 1$) when driven through a dip or on corners with banking, and 'light' ($\tilde{g}/g < 1$) when driven over the brow of a hill or on corners with adverse banking. Finally, the values of the maximum friction radius in different conditions can be interpolated, to give a 4-dimensional ($\mathbb{R}^3 \to \mathbb{R}$) continuously differentiable hyper-surface.

4.2. *Three-dimensional apex-finding*

The extension of the classic two-dimensional apex-finding method to three-dimensional scenarios in combination with the g–g–g diagrams is illustrated in this section.

The method begins with finding the apexes of corners on the given racing-line trajectory, where the vehicle is assumed to achieve maximum lateral acceleration. Before the apex, the vehicle is assumed to slow down, while after the apex the tyres are engaged with traction forces so as to remain along the g-g-g envelope; see Fig. 3. First, the positions s_k of the corner apexes are obtained by finding the peaks of the racing-line curvature in (13)

$$s_k = \underset{s_i \leq s \leq s_f}{\arg\max} \left|\Gamma(s)\right|. \tag{34}$$

Secondly, the maximum achievable lateral acceleration $\hat{a}_y$ of the vehicle on each apex is

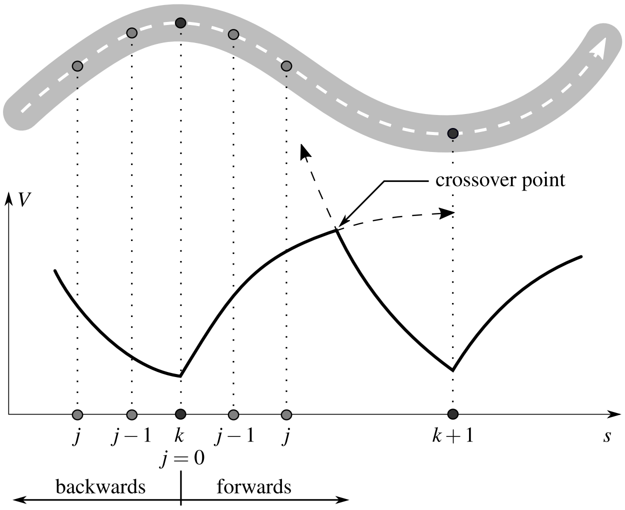

Figure 3.: Apex-finding method. The vehicle speeds up after the corner apex k and slows down before the corner apex $k+1$; the two speed profiles intersect at the crossover point.

obtained by solving the following non-linear programming (NLP) problem for V and $\hat{a}_x$

$$\underset{V,\,\hat{a}_x}{\text{maximise}} \quad |\hat{a}_y|\,, \tag{35}$$

$$\text{subject to} \quad \tilde{\rho}(\tilde{a}_x, \tilde{a}_y) = \tilde{\rho}_{\max}(\tilde{\alpha}, V, \tilde{g}). \tag{36}$$

In practice, the problem is to find the point of the g–g–g diagram that gives the maximum lateral acceleration, for the road geometry at the apex s_k. It is worth stressing that only in basic vehicle models the maximum $\hat{a}_y$ on flat roads is attained with $\hat{a}_x = 0$, while in general small and negative $\hat{a}_x$ are necessary to maximize $\hat{a}_y$ on flat roads, in order to have zero longitudinal force on the tyres (the vehicle slows down due to aerodynamic drag, rolling resistance, transmission losses, etc.). In three-dimensional roads even the basic vehicle models attain the maximum $\hat{a}_y$ for non-zero $\hat{a}_x$. This is the reason why also $\hat{a}_x$ is a decision variable of the NLP.

Once the speed of the k-th corner is known, a mesh between the previous and successive apexes is defined and the forwards and backwards speed profiles are computed by solving the following algebraic system for V_j and $\hat{a}_{x\,j}$ on each mesh point j sequentially

$$\tilde{\rho}(\tilde{a}_{x\,j}, \tilde{a}_{y\,j}) = \tilde{\rho}_{\max}(\tilde{\alpha}_j, V_j, \tilde{g}_j), \tag{37}$$

$$\frac{V_j - V_{j-1}}{s_j - s_{j-1}} = \frac{1}{2}\left[\frac{1}{\dot{s}_{j-1}}\left(\hat{a}_{x\,j-1} - w_{j-1}\hat{\omega}_{y\,j-1}\right) + \frac{1}{\dot{s}_j}\left(\hat{a}_{x\,j} - w_j\hat{\omega}_{y\,j}\right)\right], \tag{38}$$

where (37) constraints the vehicle to remain on the g-g-g envelope, while (38) stems from the integration with the trapezoidal rule of

$$V' = \frac{1}{\dot{s}}\left(\hat{a}_x - w\hat{\omega}_y\right), \tag{39}$$

which is obtained from (9) with $\dot{s}$ given by (5). The subscript j means evaluating the corresponding quantity at the mesh point j. All quantities at s_{j-1} are known from the previous mesh point, while for $j = 0$ (where $s = s_k$, see Fig. 3) they are given by the solution of the corresponding corner apex. The algebraic system in (37)-(38) remains the same for both the forwards and backwards speed profile: $s_j > s_k$ when exiting the apex (forward solution), while $s_j < s_k$ when braking into the apex (backward solution). In addition, for the forward solution a guess on the upper side of the g–g–g is provided, while a guess on the lower side is used for the backward solution. Between the corners k and $k+1$, the backwards and forwards integrations continue step by step until crossing each other at the crossover point; see Fig. 3. As a side remark, equation (38) can be analytically solved for V_j and introduced in (37) thus reducing the system to a single algebraic equation with unknown $\hat{a}_{x\,j}$.

Regardless the complexity of the model used to generate the g–g–g, the equations of the resulting apex-finding remain the same. The NLP in (35)–(36) is simple (only two decision variables and one equality constraint) and the algebraic system in (37)–(38) is a simple root-finding problem (one or two equations and as many variables, depending on the implementation). It goes without saying that a smooth racing line is fundamental to avoid unrealistic speed profiles in the MLTS.

4.3. *Three-dimensional fixed-trajectory OCP*

The MLTS on a three-dimensional road with fixed trajectory can also be solved as an OCP problem: the objective is to perform a lap along the given racing-line trajectory in the minimum time, while remaining within the acceleration envelope of the vehicle (g–g–g diagram).

The state of the OCP is the vehicle speed V, while the control manipulates the longitudinal acceleration $\hat{a}_x$. The differential constraint of the OCP is (39) with $\dot{s}$ given by (5). It is worth stressing that the lateral position n, relative orientation $\hat{\chi}$ and its derivative $\hat{\chi}'$ are given functions of s, since the vehicle trajectory is known (and fixed). The time derivatives of accelerations (i.e. the jerks) are usually controlled to obtain smooth solutions, as shown e.g. in [17]. However, in this work the longitudinal acceleration $\hat{a}_x$ has been preferred to improve the comparison with the apex-finding method. Indeed, apex-finding methods are characterized by speed profiles with cusps at the crossover points, which are difficult to obtain when using a jerk control, which tends to smooth the speed profiles around such points.

The cost function of the OCP is the manoeuvre time

$$\mathcal{J} = \int_{s_i}^{s_f} \frac{1}{\dot{s}} ds, \tag{40}$$

and cyclic conditions are enforced in the case of closed trajectory (e.g. in MLTS)

$$V(s_i) = V(s_f). \tag{41}$$

Finally, the vehicle is constrained to remain within the g–g–g envelope by enforcing (29).

Regardless the complexity of the model used to generate the g–g–g, the resulting OCP remains simple (one state and one control), which can be solved with one of the direct or indirect methods reported in the literature [1, 31]. As is in the apex-finding method, a smooth racing line is fundamental to avoid unrealistic speed profiles.

Table 1.: Weighting factors used in the racing-line reconstruction OCP.

Symbol	**Description**	**Value**
$u_{\eta 0,\max}$	maximum value of $u_{\eta 0}$	0.1
$W_{\hat{\omega}z}$	weighting on the control $u_{\hat{\omega}z}$	0.02
$W_{\eta 0}$	weighting on the control $u_{\eta 0}$	1

5. Examples of application

The examples are given for a car and a motorcycle, since the methods presented are suitable for application to both vehicles. The selected circuit is Mugello (Italy), because its geometry is available from the literature [14, 15]. The vehicle datasets and corresponding g–g–g maps are taken from [17], and resemble those of a Formula One car and a MotoGP bike. They are not reported here to avoid repetitions.

As a first step, the approach illustrated in Sec. 3 is employed to identify the racing line in terms of lateral position n and relative orientation $\hat{\chi}$ from a noisy set of x and y coordinates (racing-line reconstruction problem). As a second step, the three-dimensional apex-finding and OCP methods described in Sec. 4 are used to perform the MLTS on the reconstructed racing line. All the OCP in this work are solved using the direct transcription method in GPOPS-II [32], together with the NLP solver IPOPT [33].

5.1. *Racing-line reconstruction*

In order to test the proposed reconstruction approach, noisy datasets are artificially generated starting from the reference optimal racing line (x and y coordinates) computed numerically in [17]. These are re-sampled at 10 Hz, which is a typical sampling frequency of GNSS technologies. Then, noise is injected in the x and y coordinates of the racing line, so as to represent the typical disturbances. Three frequency distributions for the noise are considered to investigate the robustness of the reconstruction technique: white, i.e. Power Spectral Density (PSD) with slope 0 in a log-log diagram, flicker, i.e. PSD with slope -1, and 'random walk', i.e. PSD with slope -2. These are typical frequency distributions of coloured noise in GNSS technologies [34, 35]. The noise is generated from the PSD through inverse Fourier transform [36], using the PSD frequencies between 0.5 Hz and 5 Hz. The horizontal (radial) accuracy is assumed 2.5 m with confidence level of 95%, which is a typical positioning accuracy under good multipath conditions [37, 38]. This accuracy is obtained by injecting Gaussian noise with a standard deviation of $\sigma = 1.02\,\mathrm{m}$ into the x and y coordinates (according to the selected frequency distribution), to give a Rayleigh distribution with a scale parameter of σ, which gives an accuracy in the radial direction of $2.45 \times \sigma$ with confidence level of 95% [39].

The parameters used in the reconstruction are shown in Tab. 1. All variables in the OCP are scaled according to a length scale of 50 m. The solver starts with a coarse mesh consisting of 41 mesh points, which means 205 equations to solve. The solution takes about 12 s (on a Intel Core i7-7700HQ processor, running on Windows 10) and 5 mesh refinements with a mesh tolerance below 10^{-5}. The corresponding mesh consists of approximatively 4250 mesh points with mesh spacing between 0.17 m and 8.3 m, resulting in 21250 equations.

Fig. 4(a) shows the result of the reconstructed racing-line against a sample noisy (motorcycle) racing line, while Fig. 4(b) shows the reconstructed trajectory in terms of lateral position n, relative orientation $\hat{\chi}$ and local geodesic curvature Γ against the ref-

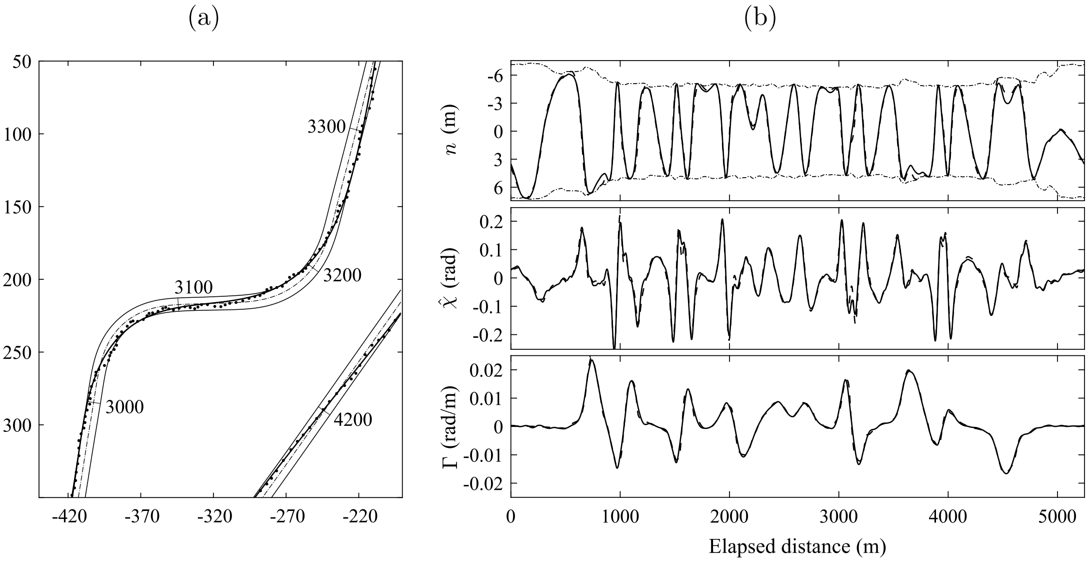

Figure 4.: Result of the racing-line reconstruction: (a) motorcycle noisy data (dots) and reconstructed racing-line (solid line) around Scarperia and Palagio (turns 10 and 11 of the Mugello circuit) with distances in meters; (b) lateral position n with road borders in dash-dot (top), relative orientation $\hat{\chi}$ (middle), and geodesic curvature Γ (bottom) of the reconstructed (solid) vs. original unperturbed racing line (dashed).

erence (unperturbed) racing line. Half-thousand noisy datasets are generated from the same reference racing line, and the reconstruction problem is solved as many times. The resulting RMS error between the reconstructed trajectory and the noisy data ranges between 1.25 m and 1.28 m (depending on the noise frequency distribution), while the RMS error between reconstructed and reference trajectory from 0.44 m to 0.53 m, with larger values related to the white noise. The travelled distance along the noisy data η_0 is typically 33–52 m larger than that along the reconstructed data η; see Fig. 2. When enforcing $u_{\eta 0} = 0$ in the OCP, the RMS errors with respect to the noisy data and reference trajectory increase to 3.04–3.34 m and 1.57–2.18 m respectively, which are at least 2 times larger than the above RMS errors. Therefore, the 'stretching' control $u_{\eta 0}$ helps significantly the reconstruction.

The comparison is repeated using the optimal trajectory of the car on the same circuit, again obtained from the free-trajectory optimization perturbed with noise. Again, half-thousand repetitions noisy datasets are generated from the reference racing line. The RMS error between the reconstructed trajectory and noisy data ranges between 1.26 m and 1.27 m, while the RMS error between reconstructed and original trajectory from 0.49 m to 0.54 m, again with larger values in the case of white noise. The travelled distance along the noisy data is typically 18–27 m larger than that along the reconstructed data. When enforcing $u_{\eta 0} = 0$, the RMS errors with respect to the noisy data and reference trajectory increase to 2.87–2.91 m and 0.99–1.21 m respectively. The results have the same order of magnitude in both the motorcycle and car scenarios. Therefore, the method proposed shows robust against the typical GNSS perturbations.

5.2. *Minimum lap time*

The three-dimensional apex-finding and OCP methods are now employed for the car and the motorcycle running on the corresponding trajectory obtained, starting from those in

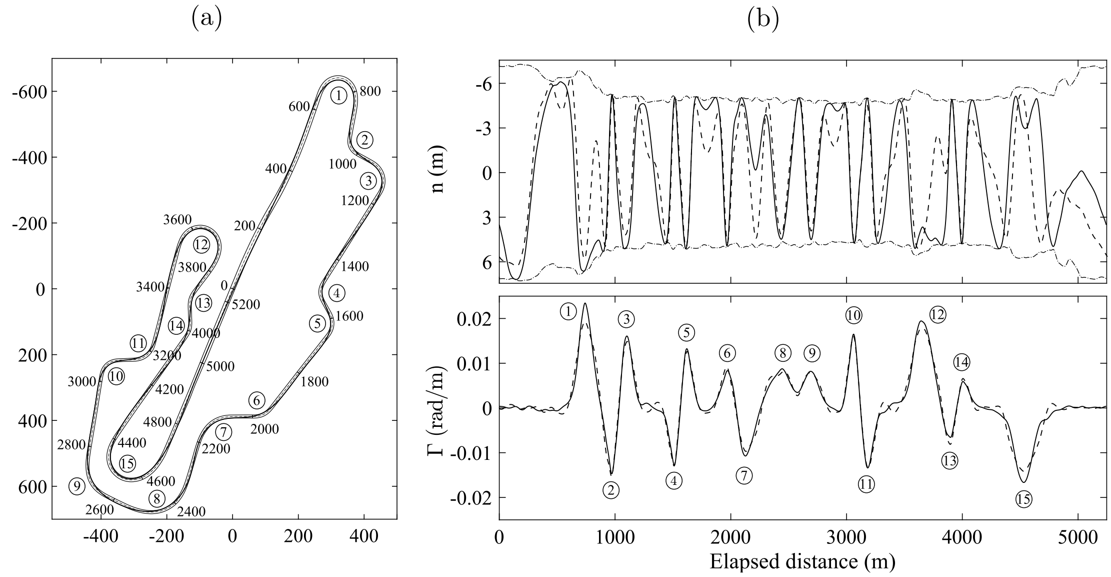

Figure 5.: (a) overview of the Mugello circuit with motorcycle's racing line; all distances are in meters. (b) lateral position n (top) and racing-line geodesic curvature Γ (bottom) of the motorcycle (solid) vs. car (dashed). The circles denote the numbering of the turns.

[17], through the reconstruction in the previous subsection; see Fig. 5. The results of the two methods are compared against each other and the effect of racetrack three-dimensionality is investigated.

In both cases, the g–g–g need be obtained. For the example presented the diagrams have been generated with speed V varying from 0 to 95 m/s by steps of 5 m/s, orientation $\tilde{\alpha}$ is varied with steps of 1 deg, and acceleration $\tilde{g}/g$ from 0.6 to 2 by steps of 0.1. For each vehicle, the g–g–g diagrams are combined to give the adherence radius hyper-surface $\tilde{\rho}_{\max}(\tilde{\alpha}, V, \tilde{g})$ and then interpolated. Finally, the fixed-trajectory OCP and the apex-finding method are run with the motorcycle and the car on the corresponding trajectory.

5.2.1. Apex-finding vs. OCP

When using the fixed-trajectory OCP, the lap times of motorcycle and car are 106.502 s and 79.637 s respectively, while they are 106.489 s and 79.637 s when employing the apex-finding method. In all lap-time simulations performed in this work the elapsed (running) time obtained with the apex-finding method, as compared with the corresponding solution found using the fixed-trajectory OCP, is below 0.02 s. Fig. 6 compares the speed profiles of the two method for the motorcycle and car. It is found that the speed profiles are in practice indistinguishable from each other, with maximum speed difference below 0.9 m/s.

All OCP have been started with a coarse mesh consisting of 41 mesh points; the tolerance for the mesh refinement in GPOPS-II is 10^{-5}. The solution is found after 8–10 mesh refinements and 25–45 s (on a Intel Core i7-7700HQ processor, running on Windows 10), depending on the vehicle. The corresponding mesh consists of approximatively 800 points, which means 1600 equations to solve. When using the apex-finding method on the mesh of the corresponding fixed-trajectory OCP, the problem consists of roughly 800 equations to solve, while the solution is found after 9–11 s, again depending on the vehicle. Therefore, the apex-finding method is 3–4 times faster than the fixed-trajectory OCP, in the current implementation.

As a final remark, 15 corner apexes are identified for the motorcycle in the apex-finding method. These actually correspond to the peaks of the racing-line curvature; see Fig. 5(b). On the contrary, the corner apexes reduces to 13 in the case of the car. Indeed, the car does not slow down in turns 13 and 14 at 3900–4000 m, even though the racing-line curvature has peaks there; see the two vertical dash-dot lines denoting turns 13 and 14 in Fig. 6(b). The speed profiles obtained from these corners when considered isolated are higher than the speed profile obtained while simulating the whole lap, see the two dotted lines in Fig. 6(b). In other words, the speed profile when braking with the car into turn 13 never intercepts the speed profile related to the acceleration out of turn 12. Similarly, the speed profile when accelerating out of turn 13 never finds the braking into turn 14. This highlights that the identification of apexes may require some manual intervention.

5.2.2. Effect on racetrack three-dimensionality

The effect of racetrack three-dimensionality is now investigated with both the fixed-trajectory OCP and apex-finding method. When running on the flat (two-dimensional) Mugello, the lap times of motorcycle and car increase by 4.167 s and 3.177 s with the fixed-trajectory OCP, while they increase by 4.188 s and 3.182 s when using the apex-finding method. Again, it is found that the speed profiles are in practice indistinguishable, with speed difference less than 0.5 m/s. The results found with the two approaches are both in line with those presented in [14, 15, 17], which employ full-dynamic models and free-trajectory QSS models.

When running on the two-dimensional road, small negative longitudinal acceleration at the corner apex is generally necessary to have zero traction force, so as to maximize the lateral acceleration. As a consequence, in this case the speed minima are always after the corresponding corner apex, see dashed-lines in Fig. 7. On the contrary, in the case of three-dimensional scenarios the longitudinal acceleration at the corner apex may be either negative (as in flat roads) or positive (e.g. when going downhill) to have zero traction force, and thus the speed minima may be either after or (slightly) before the corresponding corner apex. As opposite examples, compare the turn 9 with the motorcycle (solid line in Fig. 7(a)), where the speed minimum occurs 50 m after the corner apex, since the vehicle is going uphill (slope around +5 deg), and turn 6 with the car (solid line in Fig. 7(b)), where the speed minimum occurs about 1 m before the corner apex, since

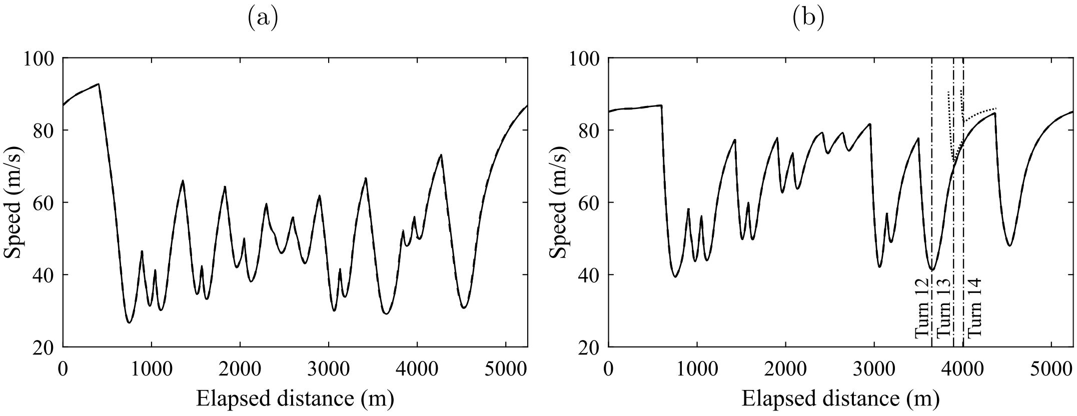

Figure 6.: Speed profiles on the Mugello circuit with the fixed-trajectory OCP (solid) and the apex-finding method (dashed): (a) motorcycle and (b) car. The speed profiles are in practice indistinguishable from each other.

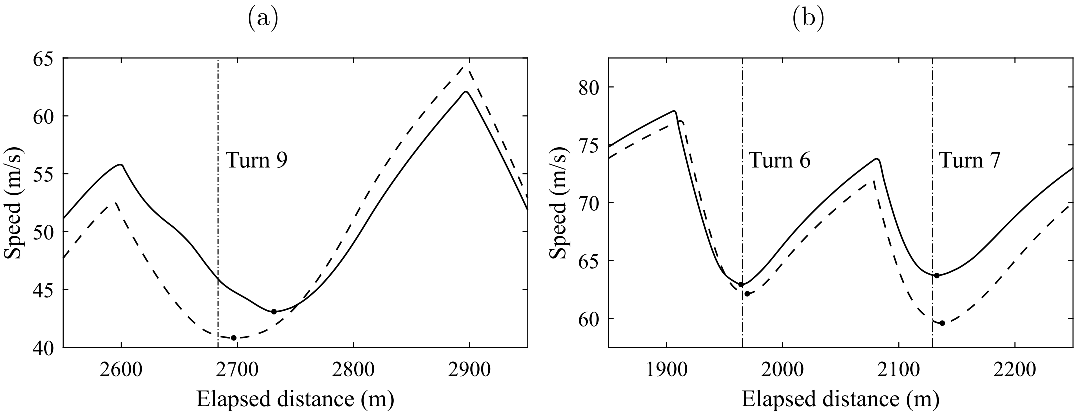

Figure 7.: Three- (solid) and two-dimensional (dashed) speed profiles on the Mugello circuit obtained with the apex-finding method: (a) motorcycle around turn 9 (Arrabbiata 2), and (b) car around turns 6-7 (Casanova and Savelli). The black dots are the speed minima, while the vertical dash-dot lines denote the locations of corner apexes, where Γ has peaks.

the vehicle is going downhill (slope around -4.5 deg).

5.2.3. Effect of the racing line

As a final investigation the MLTS of the motorcycle running on the car trajectory and the car running on the motorcycle trajectory are performed with the two methods, in order to highlight the effect of the racing line (an input to the methods presented in this work). With the fixed-trajectory OCP, the motorcycle and the car are 2.367 s and 0.580 s slower when switching the reference trajectories. Similarly, when using the apex-finding method the lap times increase by 2.370 s and 0.589 s respectively. The two methods give again similar results.

6. Conclusion

Two methods for the minimum lap time of road vehicles with quasi-steady-state models on a fixed three-dimensional racing line have been proposed, which extend the two-dimensional methods reported in the literature. The first method is an extension of the classic 'apex-finding' method to three-dimensional scenarios (i.e. including road elevation, slope and banking), while the second is based on the solution of a relatively simple optimal control problem (OCP), and is alternative to the apex-finding approach. Both approaches are based on the application of g–g–g diagrams, which are a generalisation of the classic g–g diagrams to three-dimensional scenarios. The racing line used in the minimum-lap-time problems is obtained from a noisy set of x and y coordinates (e.g. logged via GNSS or from trajectories drawn by hand). A solution to the trajectory reconstruction problems based on an optimal control approach has been proposed and tested with the typical noise of GNSS. The proposed method is able to account for the three-dimensionality of the racetrack—this is again an extension of the two-dimensional methods reported in the literature. Examples of application have been given for a race car and a race motorcycle on the Mugello circuit. The road three-dimensionality affects significantly the lap time (3-4 s in the examples provided), and the results reported are in line with those obtained in the literature with free-trajectory methods, both with quasi-steady and full-dynamic vehicle models. The computation time of the three-dimensional 'apex-finding' is 3-4 times lower than the fixed-trajectory OCP, in the implementation by the authors. Both methods require computation time that are roughly one order of magnitude lower than those corresponding to approaches involving quasi-steady-state models without pre-defined trajectory, and about two orders of magnitude lower than dynamic vehicle models with free trajectory.

References

[1] Massaro M, Limebeer DJN. Minimum-lap-time optimization and simulation. Vehicle System Dynamics. 2021;59(7):1069–1113.

[2] Limebeer DJN, Massaro M. Dynamics and optimal control of road vehicles. Oxford University Press; 2018.

[3] Scherenberg H. Mercedes-Benz Racing Design and Cars Experience. SAE Transactions. 1958; (580042):414–20.

[4] Metz D, Williams D. Near Time-Optimal Control of Racing Vehicles. Automatica. 1989;25(6):841–57.

[5] Milliken WF, Milliken DL. Race car vehicle dynamics. Warrendale, PA: SAE; 1995.

[6] Gadola M, Vetturi D, Cambiaghi D, Manzo L. A tool for lap time simulation. SAE Technical Papers. 1996;.

[7] Brayshaw D, Harrison M. A Quasi Steady State Approach to Race Car Lap Simulation in order to Understand the Effects of Racing Line and Centre of Gravity Location. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 2005;219(6):725–39.

[8] Brayshaw D, Harrison M. Use of Numerical Optimization to Determine the Effect of the Roll Stiffness Distribution on Race Car Performance. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 2005;219(10):1141?51.

[9] Hauser J, Saccon A. Motorcycle modeling for high-performance maneuvering. IEEE Control Systems Magazine. 2006;26(5):89–105.

[10] Lenzo B, Rossi V. A simple mono-dimensional approach for lap time optimisation. Applied Sciences. 2020;10(4).

[11] Perantoni G, Limebeer DJN. Optimal Control of a Formula One Car on a Three-Dimensional Track. Part 1: Track Modelling and Identification. ASME Journal of Dynamical Systems, Measurement, and Control. 2015;137(5).

[12] Limebeer DJN, Perantoni G. Optimal Control of a Formula One Car on a Three-Dimensional Track Part 2: Optimal Control. ASME Journal of Dynamical Systems, Measurement, and Control. 2015; 137(5).

[13] Lot R, Biral F. A curvilinear abscissa approach for the lap time optimization of racing vehicles. IFAC Proceedings Volumes. 2014;47(3):7559 – 7565; 19th IFAC World Congress.

[14] Marconi E, Massaro M. The effect of suspensions and racetrack three-dimensionality on the minimum lap time of motorcycles. In: Klomp M, Bruzelius F, Nielsen J, Hillemyr A, editors. Advances in Dynamics of Vehicles on Roads and Tracks. Cham: Springer International Publishing; 2020. p. 1368–1377.

[15] Leonelli L, Limebeer DJN. Optimal control of a road racing motorcycle on a three-dimensional closed track. Vehicle System Dynamics. 2020;58(8):1285–1309.

[16] Veneri M, Massaro M. A free-trajectory quasi-steady-state optimal-control method for minimum lap-time of race vehicles. Vehicle System Dynamics. 2020;58(6):933–954.

[17] Lovato S, Massaro M. A three-dimensional free-trajectory quasi-steady-state optimal-control method for minimum-lap-time of race vehicles. Vehicle System Dynamics. 2021;.

[18] Lovato S, Massaro M, Limebeer DJN. Curved-ribbon-based track modelling for minimum lap-time optimisation. Meccanica. 2021;56(8):2139–2152.

[19] Kelly DP, Sharp RS. Time-Optimal Control of the Race Car: Influence of a Thermodynamic Tyre Model. Vehicle System Dynamics. 2012;50(4):641–62.

[20] West WJ, Limebeer DJN. Optimal tyre management for a high-performance race car. Vehicle System Dynamics. 2020;0(0):1–19.

[21] Christ F, Wischnewski A, Heilmeier A, Lohmann B. Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients. Vehicle System Dynamics. 2021;4(59):588–612.

[22] Masouleh MI, Limebeer DJN. Optimizing the Aero–Suspension Interactions in a Formula One Car. IEEE Transactions on Control Systems Technology. 2016;24(3):912–27.

[23] Smith EN, Velenis E, Tavernini D, Cao D. Effect of handling characteristics on minimum time cornering with torque vectoring. Vehicle System Dynamics. 2018;56(2):221–248.

[24] Sedlacek T, Odenthal D, Wollherr D. Minimum-time optimal control for battery electric vehicles with four wheel-independent drives considering electrical overloading. Vehicle System Dynamics. 2020;0(0):1–25.

[25] Tavernini D, Massaro M, Velenis E, Katzourakis D, Lot R. Minimum Time Cornering: The Effect of Road Surface and Car Transmission Layout. Vehicle System Dynamics. 2013;51(10):1533–47.

[26] Tavernini D, Velenis E, Lot R, Massaro M. The Optimality of the Handbrake Cornering Technique. Journal of Dynamic Systems, Measurement, and Control. 2014;136(4).

[27] Braghin F, Cheli F, Melzi S, Sabbioni E. Race driver model. Computers and Structures. 2008; 86(13-14):1503–1516.

[28] Kapania N, Subosits J, Gerdes J. A sequential two-step algorithm for fast generation of vehicle racing trajectories. Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME. 2016;138(9).

[29] Heilmeier A, Wischnewski A, Hermansdorfer L, Betz J, Lienkamp M, Lohmann B. Minimum curvature trajectory planning and control for an autonomous race car. Vehicle System Dynamics. 2020; 58(10):1497–1527.

[30] Casanova D, Sharp RS, Symonds P. Technical note: Construction of race circuit geometry from on-car measurements. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 2001;215(9):1033–1042.

[31] Dal Bianco N, Biral F, Bertolazzi E, Massaro M. Comparison of Direct and Indirect Methods for Minimum Lap Time Optimal Control Problems. Vehicle System Dynamics. 2019;57(5):665–696.

[32] Patterson MA, Rao AV. GPOPS - II: A Matlab Software for Solving Multiple-Phase Optimal Control Problems Using *hp*–Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming. ACM Transactions on Mathematical Software. 2014;41(1).

[33] Wächter A, Biegler LT. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. Mathematical Programming. 2006;106:25–57.

[34] Kim DM, Suk J. Gps output signal processing considering both correlated/white measurement noise for optimal navigation filtering. International Journal of Aeronautical and Space Sciences. 2012;13(4):499–506.

[35] Ugazio S, Presti LL. Effects of colored noise in linear adaptive filters applied to gnss multipath detection. 2013 Conference on Design and Architectures for Signal and Image Processing. 2013; :126–133.

[36] Shinozuka M, Deodatis G. Simulation of Stochastic Processes by Spectral Representation. Applied Mechanics Reviews. 1991;44(4):191–204.

[37] Pesyna KM, Humphreys TE, Heath RW, Novlan TD, Zhang JC. Exploiting antenna motion for faster initialization of centimeter-accurate gnss positioning with low-cost antennas. IEEE Transactions on Aerospace and Electronic Systems. 2017;53(4):1597–1613.

[38] Specht M, Specht C, Dabrowski P, Czaplewski K, Smolarek L, Lewicka O. Road tests of the positioning accuracy of ins/gnss systems based on mems technology for navigating railway vehicles. Energies. 2020;13(17).

[39] Rayleigh distribution. John Wiley, Sons, Ltd; 2010. Chapter 39; p. 173–175.