



Comparison of direct and indirect methods for minimum lap time optimal control problems

Nicola Dal Bianco, Enrico Bertolazzi, Francesco Biral & Matteo Massaro

To cite this article: Nicola Dal Bianco, Enrico Bertolazzi, Francesco Biral & Matteo Massaro (2018): Comparison of direct and indirect methods for minimum lap time optimal control problems, *Vehicle System Dynamics*, DOI: [10.1080/00423114.2018.1480048](https://doi.org/10.1080/00423114.2018.1480048)

To link to this article: <https://doi.org/10.1080/00423114.2018.1480048>



Published online: 04 Jun 2018.



Submit your article to this journal



View related articles



View Crossmark data



Comparison of direct and indirect methods for minimum lap time optimal control problems

Nicola Dal Bianco  ^a, Enrico Bertolazzi  ^b, Francesco Biral  ^b and Matteo Massaro  ^a

^aDepartment of Industrial Engineering, University of Padua, Padova, Italy; ^bDepartment of Industrial Engineering, University of Trento, Trento, Italy

ABSTRACT

Minimum lap time simulations are especially important in the design, optimisation and setup of race vehicles. Such problems usually come in different flavours, e.g. quasi-steady state models vs. full dynamic models and pre-defined (fixed) trajectory problems vs. free trajectory problems. This work is focused on full dynamic models with free trajectory. Practical solution techniques include direct methods (i.e. solution of an nonlinear programming problem problem, widespread approach) and indirect method (i.e. based on Pontryagin's principle, less common, yet quite efficient in some cases). In this contribution the performance of the direct and indirect methods are compared in a number of vehicle-related problems.

ARTICLE HISTORY

Received 9 October 2017

Revised 23 April 2018

Accepted 11 May 2018

KEYWORDS

Optimal control; lap-time; direct methods; indirect methods; NLP; OCP; car simulation

1. Introduction

In the last decades the applications developed for minimum lap time problems have become a tool widely used to improve the performance of race vehicles. Minimum lap time problems not only are of great practical interest to help design, optimise and setup a vehicle for maximum performance but they also are a challenging theoretical and numerical problems. Early attempts to solve minimum lap time problems date back to the late 1950s [1]. Later in the 1980s authors of [2] simulated a section of the Formula One Circuit Paul Ricard in southern France using a quasi-steady state (QSS) optimisation routine to compute the optimal controls. Since then, many improvements have been introduced for solving minimum lap time problems with several authors proposing and developing different solving techniques and new theoretical and numerically efficient algorithms also supported by the increase of the cpu performance. Among all the methods that have been used until now, the ones that showed the best capabilities falls into four major categories: QSS, optimal control based, driver model based and evolutionary algorithm-based simulations.

In the QSS approach, the racing line is provided as input and it is divided into segments in which the vehicle is considered to be in stationary conditions except for few state variables such as the speed along the racing line. The curvature radius is then calculated for each segment and the vehicle is assumed to have zero longitudinal acceleration in those points where the curvature has a local maximum (at the corner apex); thus the maximum

vehicle speed can be calculated in correspondence of such points. Then, the vehicle speed is reconstructed backward and forward on the basis of the maximum g-g envelope. Examples of this method can be found in [3–8]. This method has shown both good robustness and fast computation times, together with the capability to use complex vehicle multibody models; however most of the transient behaviours is neglected (e.g. tyre loads dynamics, yaw dynamics and suspension damper effects) and the results obtained are less accurate when compared with other minimum time techniques. An extended iterative steady-state approach was presented [9] to include some of these effects, e.g. the suspension damping effects. Despite the additional benefits compared to standard QSS method a comparison with optimal transient solution is not given.

The second family of minimum lap time methods falls into the optimal control problem (OCP) theory, which are also called transient-optimal control to distinguish them from QSS simulations [5,10,11]. The general idea behind this approach is to translate the minimum lap time problem into an OCP, where the dynamic equations of motion become constraints of the optimisation process. The methods for solving OCPs can be mainly grouped into three main categories: dynamic programming (DP), indirect optimal control, direct optimal control. A comprehensive summary of optimal control approaches and numerical solution methods can be found in [12–14].

To the best of the authors' knowledge, despite the theoretical advantages of the DP, such as handling discrete/continuous variables and guarantee the global optimum, none minimum lap time application has been solved with this method in the literature. The main reason is that it suffers from the curse of dimensionality even with relatively small vehicle models [14,15]. To mitigate the limitations of DP the differential dynamic programming (DDP) was introduced that solves a sequence of quadratic subproblems obtained from the quadratic approximations of the objective function around a reference trajectory [16,17]. The DDP was further developed by many other authors to handle highly nonlinear dynamics subject to state and control constraints. The Hybrid DDP is currently the state of art of DDP algorithm that combines DDP with some well-proven nonlinear mathematical programming techniques and was successfully used to solve a large scale spacecraft trajectory optimisation problem [18,19]. Even if still in the early stage of development, it is a promising method thanks to its robustness and large convergence radius, yet it has not been applied to challenging minimum lap time problems.

Indirect methods for OCPs rely on the Pontryagin Maximum Principle [20], which gives the necessary conditions for optimality. Such conditions render themselves as a set of ordinary differential equations with both initial and final boundary conditions, i.e. a so called two-points boundary value problem (TPBVP), coupled with a minimisation problem to derive the optimal control law. Various numerical techniques can be used to solve such a problem. Early examples of application of indirect optimal control theory to minimum lap time simulations are reported in [21,22]. Since the late 90's various other works have used this approach both for motorbike lap time simulations [23,24] and car lap time simulations [25–29]. The TPBVP is solved both using single or multiple shooting (only controls are discretised) or with full collocation (i.e. states and controls are discretised).

Direct methods translate the OCP into a discrete constrained minimisation problem (i.e. direct transcription) also known as nonlinear programming problem (NLP). Similar to indirect methods, various numerical schemes can be used to discretise controls only (i.e. sequential discretisation or collocation) or controls and states (i.e. full discretisation

or collocation). Direct methods can also differ based on how the NLP is solved, e.g. using interior point algorithms, active set, sequential quadratic programming), etc. Among the sequential discretisation methods, direct multiple shooting method (also known as parallel shooting method) emerged as the most efficient because it is less affected by high sensitivities and naturally renders itself for parallelisation [30]. In [31] it was successfully applied to minimum time problem with gear choice using using MUSCOD-II software [32] and with a partial outer convexification to handle discrete variables.

Methods based on direct full collocation are largely diffused in the community because the resulting NLP can be easily solved thanks to the availability of IPOPT [33], a robust and efficient interior point algorithm. The approaches mainly differ from the discretisation method but IPOPT is the solver used by all of them, e.g. [34] used Lagrange Polynomials to discretise both control and states, [35] used a trapezoidal integration scheme to convert the OCP into an NLP problem by means of ICLOCS toolbox [36], various minimum lap time problems were formulated with a direct orthogonal collocation method based on Radau pseudo-spectral scheme by means of the software package GPOPS-II [37], which proved to be effective to illustrate the impact of optimal usage of energy recovery systems on fuel consumption saving [38] and the impact of aero-suspension interactions and adjustments on the lap-time performance of the car [39]. Direct methods have rarely been applied to motorbike models; to the authors' knowledge, the only relevant work is [40] where a full lap was simulated with a relatively complex motorbike model. However, he did not optimised at once the trajectory and the controls but a path following algorithm was used to make the motorbike follow a given trajectory.

Many authors proposed various alternative techniques in order to increase the robustness and to reduce the computational burden due to the size of the NLP resulting from long circuit/test course. One of these are the moving horizon techniques, which decompose the global OCP into a sequence of local OCPs over a finite horizon (i.e. preview length) that is moving forward in time [11,41–46] and satisfies appropriate continuity conditions. Optimisation of the preview length is necessary to guarantee for finding the global problem optimum. Additionally, this technique cannot be directly used when optimisation of global variables is required. Another proposed idea splits the problem into trajectory *planning* and *tracking*. Trajectory planning is usually solved as an optimisation problem on a simplified vehicle model and the tracking task is performed with a *driver model* (or controller) that guides the full vehicle along the pre-calculated trajectory. Examples are [47] that uses a direct collocation method for the trajectory planning; [48] that combines geometrical trajectory and speed profile optimisation with a simple driver model; [49] where the planning is based on a g-g diagram that reproduces the driver capabilities; [50] that relies on the indirect method for the trajectory planning; [51] where the tracking task is based on the decomposition of longitudinal and transverse dynamics; [52–55] that uses Model Predictive Control to implement the tracking task. However, all these approaches, despite their ability to be used in combination with complex vehicle models, should be considered as sub-optimal solutions since the driver model influences the lap time obtained.

Finally, evolutionary approaches, such as genetic algorithms, are the alternative numerical algorithms, for the solution of direct sequential optimal control, compared to derivative-based NLP solvers. A recent application to minimum lap time problem is reported in [56,57]. The procedure that leads to the lap time minimisation is the following:

for a given control history defined as piecewise linear, the equations of motion are integrated, resulting in a certain manoeuvring time; then the genetic algorithm optimises the control values at given number of knots in unknown time positions and the equations of motions are integrated with the new controls to evaluate the cost and the constraints. The genetic algorithm thus selects the control variations that brought to an improvement of the performance and the process is iterated again in a manner very similar to a single shooting methods. This approach demonstrated to be able to handle very complex vehicle models, however it also resulted to be significantly slow: the computation of a manoeuvre over two turns takes approximately one day to execute and the control law shape is derived from experimental data.

As final a remark of the literature review above, authors think it is worth it investigating benefits and limitations of optimal control methods for minimum lap time problems that exhibit the following characteristics:

- they possibility cope with non-trivial vehicle models, e.g. several dof ($> 5 - 7$) including highly nonlinear tyre model and aerodynamics interaction;
- they can provide simultaneous optimisation of the racing line and the controls;
- they have the ability to compute a simulation in a reasonable amount of time (e.g. less than few hours);
- they are sufficiently robust to variations on initial guess.

According to the literature review and to the authors knowledge and experience, three approaches emerged as the most effective to solve minimum lap time problems.

Following the historical development, the first method is an indirect approach implemented by the software PINS (formerly known as XOptima), which is described in [23]. It has been used since the late 1990s and the most representative minimum lap time results achieved with PINS are presented in [23,24,58] for motorbike applications, and in [25,26] for cars. The second method is a direct multiple shooting method implemented using the software MUSCOD-II and whose most successful result is in [31]. The last and most recent method is based on direct optimal control with full direct transcription via pseudospectral collocation and it is represented by the software GPOPS-II, [37]. It was released quite recently in the 2013 and its state of the art applications in this topic is described in [38,59].

As far as we are concerned, in the literature, there are no comparisons of the accuracy of the solution and numerical performance on minimum lap time problems among these methods. This gave the motivation for this work to compare the most promising optimal control approaches on the same OCPs, in order to highlight the similarities and the differences, and possibly understand which one is more indicated for a particular purpose. Unfortunately MUSCOD-II is not available and for this reason the work only focuses on the comparison of PINS and GPOPS-II software as representative of state of art methods of indirect and direct approach, respectively.

In Section 2, PINS and GPOPS-II and their approach are briefly presented, then in Section 3 they are tested against three vehicle OCPs. The comparison will focus on the solution accuracy, robustness with respect to guess perturbations, constraints enforcement and parameter sensitivity. Finally in Section 4, the results are summarised and the software performance discussed.

2. Minimum time manoeuvring OCP

In general, a minimum lap time problem can be formulated as a constrained optimisation process where the mathematical model of the vehicle is described by a set of differential equations $f(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}, \boldsymbol{\beta}, t) = \mathbf{0}$ where \mathbf{x} is the set of state variables, \mathbf{u} is the set of control variables, $\boldsymbol{\beta}$ the set of vehicle parameters, and t is the independent variable, e.g. time. A general OCP is stated as follows

$$\text{Minimise : } \Phi(\mathbf{x}(t_i), \mathbf{x}(t_f), \boldsymbol{\beta}) + \int_{t_i}^{t_f} \mathcal{L}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}, t) dt \quad (1a)$$

$$\text{subject to : } f(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{u}, \boldsymbol{\beta}, t) = \mathbf{0} \quad (1b)$$

$$\psi_{\min} \leq \psi(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}, t) \leq \psi_{\max} \quad (1c)$$

$$q_{\min} \leq \int_{t_i}^{t_f} Q(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}, t) \leq q_{\max} \quad (1d)$$

$$b_{\min} \leq b(\mathbf{x}(t_i), \mathbf{x}(t_f), \boldsymbol{\beta}) \leq b_{\max}, \quad (1e)$$

where (1e) are the initial and/or final boundary conditions, (1d) are integral constraints, (1c) is the set of algebraic inequalities or *path constraints* (e.g. track borders, engine power, tyre adherence, etc.) that may also include equality constraints when both bounds are coincident. Additionally, (1b) can be described by a system of ordinary differential equations or in a more general form by a system of differential algebraic constraints (DAEs) that is differential equations coupled with a set of algebraic equations that make some state variable dependent from the others. DAEs are quite common for multibody system such as complex vehicle models.

The OCP defined in (1) is quite general and may not fit the implementation of various software for solving OCPs. This is also the case for the software (PINS, GPOPS-II) that we have selected for comparison. In these cases some problem reformulations of the OCP problem may be required. Thus next Sections 2.1 and 2.2 briefly describe the two software with key features and limitations.

2.1. OCP: formulation and solution with PINS

PINS is a collection of libraries and programs mainly developed to symbolically formulate and numerically solve OCPs for nonlinear dynamical systems described by differential equations. PINS implements an indirect method with penalties and barriers to handle generic mixed state and control constraints. The optimal controls are explicitly derived using the Pontryagin Maximum Principle, which is done formally building a map that computes the controls as a function of state and co-state. The map is solved analytically whenever the problem makes it possible, otherwise a numerical procedure is used to obtain controls and their derivatives with respect to the states and co-states. The resulting TPBVP is approximated with finite difference (based on midpoint quadrature) to give a nonlinear algebraic system of equations, which is solved with a damped Newton Affine Algorithm specifically developed to exploit the Bordered Almost Block Diagonal (BABD)

structure of the jacobian [60]. PINS makes uses of the Maple[©] symbolic engine, via the XOPTIMA package, to symbolically formulate the OCP, automatically generate the equations of necessary conditions of optimality and the corresponding analytical jacobians, and finally translate into C++ code. The generated code can be, if necessary, further manipulated by the user and compiled and linked with libraries contained in MECHATRONIXTOOLKIT by the program pins, in order to produce a stand alone program or a callable library that can be used in custom code for different purposes (including real time applications). The MECHATRONIXTOOLKIT is a collection of C++ libraries of classes such as NonLinear system solvers, Boundary Value Problem Solvers, ODE-DAE solvers. The library is also complemented with utility classes such as vehicle components (e.g. tyre models, internal combustion engine models, etc), 2D–3D road models, splines and various interfaces to other languages such as Mruby and Lua and MATLAB[©]/Simulink.

The OCP formulation in PINS is:

$$\mathbf{A}(\mathbf{x}, \boldsymbol{\beta}, t) \dot{\mathbf{x}} = \tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}, t), \quad (2)$$

where \mathbf{A} is a square matrix that cannot depend on the controls. Multibody and vehicle dynamic equations can always be written in this form. In practice, the main difference between (1b) and (2) is that the latter assumes that the problem is linear in the velocities. This is an advantage compared to the majority of OCP solvers that require the differential equations to be in the explicit form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}, t)$ where inversion of matrix \mathbf{A} may be necessary as also found in almost all text books [61].

PINS does not directly handle integral constraints but they can be added to the problem by converting the integral constraints into differential constraints introducing additional states with proper boundary conditions.

PINS treats inequalities (1c) augmenting the target function (1a) with a weighted sum of penalty or barrier functions $\mathbf{P}(\psi(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}, t))$ for each inequality. The function \mathbf{P} is designed to be continuously differentiable of class C^3 in such a way that it evaluates nearly zero (i.e. ϵ) when ψ is at a distance h from bound. It then grows to 1 at the bound then pretty linearly to infinity out of the bound for the penalty and to infinite at the bound for barrier. Clearly barrier does not allow to break the bound. Parameters h and ϵ are used, respectively, to define when the penalty starts to increase the cost and how much.

The main limitation of PINS is the fact that cannot directly handle DAEs of index greater equal 1. For what concerns index-1 algebraic constrains only those linear in the algebraic variables can be used. For DAEs with index of higher order reduction techniques and penalisation terms in the cost function (1a) to avoid constraint drift are necessary [13].

Since PINS implements an indirect method [61] it has to solve the necessary conditions of optimality which are described by a pure TPBVP since penalty are also used to enforce constraints on controls. The TPBVP consists of a set of n_x differential equations for the states \mathbf{x} , n_x adjoint equations for the lagrange multipliers $\boldsymbol{\lambda}$, n_u algebraic equations for the optimal controls \mathbf{u} and $2n_x + n_b$ equations for the boundary conditions. The controls are formally solved either symbolically or numerically as a function of the states \mathbf{x} and Lagrange multipliers $\boldsymbol{\lambda}$. The TPBVP is discretised using a finite difference trapezoidal scheme and the resulting large set of algebraic equations is solved with a dumped Newton Affine scheme that exploits the Jacobian block diagonal structure [23,62].

Summarising, the key points of PINS approach for the solution of the indirect OCP are:

- dynamic equations (1b) can be given in implicit form, but it has to be linear in the state derivatives;
- inequality constraints (1c) are treated with penalty/barrier functions (constrained problem converted in equivalent unconstrained problem)
- only index-1 algebraic constraints linear in the algebraic variable can be handled directly
- the controls are formally solved analytically, if an explicit analytic solution is not available numerical methods are used
- states are discretised on the mesh points while controls are assumed constants on cells and thus the TP-BVP problem is solved as a large set of algebraic equations roughly of dimension $(2n_x + n_u)N + 2n_x + n_b$ where N is the number of mesh points.
- solution is obtained using custom nonlinear system numerical solver

Further details can be found in [23,62].

2.2. OCP: formulation and solution with GPOPS-II

GPOPS-II is a MATLAB[®] software intended to solve general OCPs for nonlinear dynamical systems described by differential-algebraic equations. GPOPS-II implements a direct full collocation approach by means of pseudo spectral method. The continuous-time OCP is approximated using a new class of a variable-order Legendre–Gauss–Radau quadrature orthogonal collocation polynomials resulting into a sparse NLP. This NLP is then solved using either the NLP solver IPOPT or the NLP solver SNOPT. A distinguishing feature is the adaptive mesh refinement method that determines the number of mesh intervals and the degree of the approximating polynomial within each mesh interval to achieve a specified accuracy. To achieve this GPOPS-II performs an a-posteriori solution error estimation and refine the mesh in those mesh segments where the error is higher than a certain threshold; this process is then repeated until all mesh interval satisfy the desired error threshold. Adopting the variable order for each mesh interval and the mesh refinement GPOPS-II is able to achieve high accuracy limiting the use of resources by putting refining only where it is necessary. By means of the free MATLAB[®] package ADIGATOR [63] symbolic gradients and jacobians could also be generated.

As for the general formulation of the OCP described by (1) GPOPS-II fits the definition with pretty much the same except it requires the dynamic equations (1b) to be in explicit form:

$$\dot{x} = \hat{f}(x, u, \beta, t). \quad (3)$$

In general, vehicle dynamic equations can often be reduced to the explicit form. However, it may not be possible when the model complexity increase, which is the case of advanced motorcycle models, e.g. [24,58,64]). For some problems the implicit form is more robust and the solver converge faster to solution [65]. Since GPOPS-II naturally treats algebraic equations of index 1 the implicit formulation can be also implemented at the cost of doubling the problem dimension adding n additional states y and n additional algebraic path



constraints as follows

$$\dot{x} = y, \quad (4)$$

$$\mathbf{0} = \tilde{f}(x, u, \beta, t) - A(x, \beta, t)y. \quad (5)$$

GPOPS-II approximate both states and controls using multiple-interval Legendre–Gauss–Radau quadrature orthogonal collocation method that transcribes the OCP into an NLP of dimension roughly of $n_x(N^{(k)} + 1) + n_uN^{(k)}$ for each $k = 1 \dots K$ intervals. The structure of the jacobians and hessian are sparse matrices that do not show any specific structure that can be exploited by the NLP solver. Additionally, since GPOPS-II uses IPOPT as NL solver internally the problem is augmented to handle the equality constraints via lagrange multipliers. This is transparent from the user perspective but has an effect on the computational performance point of view. Additionally, IPOPT, being an interior point method, uses barrier functions to treat inequality constraints [33]. However, it adopts several automatic strategies to adapts the barrier weights during algorithm convergence to satisfy the inequalities with the higher accuracy possible.

Summarising, the key points of GPOPS-II in the direct solution of the OCP are:

- dynamic equations (1b) must be in explicit form if one does not want to double the dynamic system dimension;
- algebraic constraints index-1 can be used
- automatic mesh refinement algorithm available
- automatic differentiation is available via Adigator to generate the necessary gradients, Hessian and Jacobians
- the solution is obtained using NLP solver IPOPT that implements a robust and quite fast interior point algorithm.

Further details on the software can be found in [37].

3. Test bench examples for vehicle OCPs

In this section the two software previously presented, PINS and GPOPS-II, are tested on three test bench vehicle OCPs. The first case of study deals with a simple motorcycle model; it has been chosen because the exact solution can be mathematically derived, thus it can be used for comparison with numerically calculated solutions. The second test benchmark consists in the reconstruction of a race circuit from experimental data; since this OCP does not include any path constraints, it leads to the same minimisation problem both through the indirect and direct approach. For this reason this problem has been chosen to test the robustness of PINS and GPOPS-II with respect to perturbations in the guess. Finally, the third and last test problem consists in a minimum lap time problem of a relatively simple, yet effective, car model; this represents a typical utilisation scenario for the studied software. The constraint enforcement, sensitivity to parameter variations and robustness to guess perturbations will be studied.

3.1. Basic two-wheeled minimum manoeuvre

The first OCP consists in the minimum manoeuvre time of a basic motorcycle model, that has to be moved from the upright configuration (zero roll angle) to a leaned configuration

(non-zero one) in the minimum time. The assumptions are that the tyres have zero slip, the suspensions are fixed and the speed is constant. Under these circumstances the model has one degree of freedom only, the roll angle, see Chapter 6 of [66].

The motorcycle model has two state variables, the roll angle ϕ and the roll rate $\dot{\phi}$, and one input, the steering angle δ . The related first-order differential equations are

$$\dot{\phi} = \dot{\phi}_{\text{dot}}, \quad I_{xx}\ddot{\phi}_{\text{dot}} = mh \left(g\phi - \frac{V^2}{L}\delta \right), \quad (6)$$

where m is the total (vehicle plus rider) mass, I_{xx} the roll moment of inertia about line joining the two tyre contact points, h the distance height of the centre of mass from ground, L is the wheelbase and g is the gravity. Equation (6) can be conveniently rewritten in order to highlight the independent model parameters:

$$\dot{\phi}_{\text{dot}} = \frac{mgh}{I_{xx}} \left(\phi - \frac{V^2}{gL}\delta \right) \equiv A(\phi - B\delta) \quad \text{where } A = \frac{mgh}{I_{xx}} \quad B = \frac{V^2}{gL}. \quad (7)$$

The minimum time to roll problem can be formulated as follows:

$$\begin{aligned} & \underset{t_f \geq t_0}{\text{minimise}} \quad t_f \\ & \text{subject to ODEs : } \dot{\phi} = \dot{\phi}_{\text{dot}}, \quad \dot{\phi}_{\text{dot}} = A(\phi - B\delta) \\ & \text{and constraints : } |\delta| \leq \delta^{\max}, \quad \phi(t_0) = 0, \quad \phi(t_f) = \phi_f, \\ & \quad \dot{\phi}_{\text{dot}}(t_0) = 0, \quad \dot{\phi}(t_f) = 0. \end{aligned} \quad (8)$$

The analytical solution is obtained using the PMP [66]:

$$t_f = \frac{1}{\sqrt{A}} (\ln(w) - \ln(1+f)), \quad \delta = \text{sign}(\phi_f) \delta^{\max} \times \begin{cases} +1 & 0 \leq t < t^*, \\ -1 & t^* \leq t \leq t_f, \end{cases} \quad (9)$$

where

$$f = -\frac{|\phi_f|}{B\delta^{\max}}, \quad w = 1 - f - \frac{f^2}{2} + \frac{\sqrt{f(f+4)(f^2-4)}}{2}, \quad t^* = \frac{\ln(1+w) - \ln(2)}{\ln(w) - \ln(1+f)}. \quad (10)$$

Finally, the solution exists only if $f > -1$ and w is real $w > -1$ which imply $f \in (-1, 0]$ and:

$$|\phi_f| < \delta^{\max} B. \quad (11)$$

It is worth noting that a positive steer angle δ corresponds to a positive roll angle ϕ in steady-state conditions, while the bike is performing a right or clockwise turn.

3.1.1. Solution analysis

The numerical solution obtained using the motorcycle dataset of Table 1 ($A = 42.92 s^{-1}$ and $B = 8.551$). The theoretical minimum time with the data used in this work is $t_f = 1.072369 \times 10^{-1} s$.

Table 1. Motorcycle problem dataset.

Variable	Value	Description
g	9.806[m/s ²]	Gravity
V	11[m/s]	Speed
m	273[kg]	Total mass
L	1.443[m]	Wheelbase
I_x	70.79[kg m ²]	Roll inertia moment
δ_{\max}	20°	Max steer angle
ϕ_f	20°	Final roll angle
t_f^g	1[s]	Guess on final time
$\phi_f t/t_f^g$		Guess on roll angle

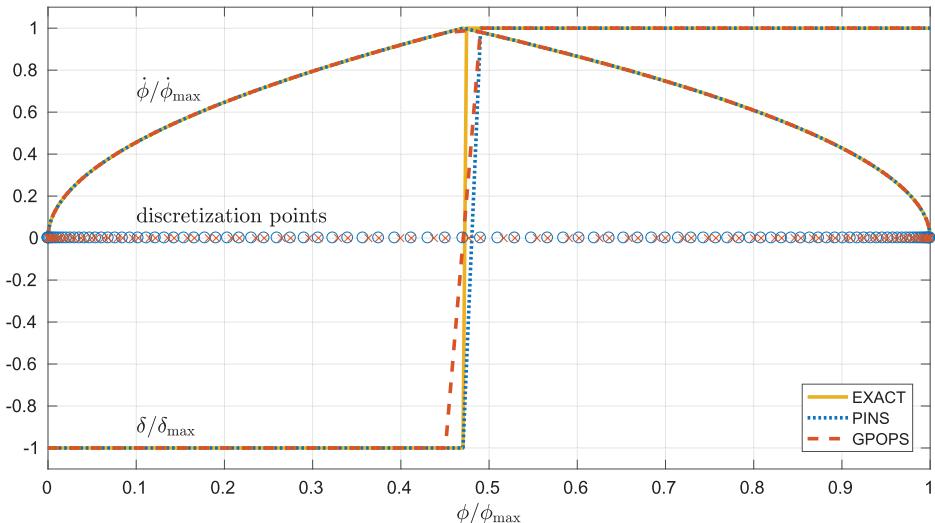


Figure 1. The control δ and roll rate $\dot{\phi}$ are shown as function of the roll angle ϕ . The solutions provided by PINS and GPOPS-II are shown together with the exact one. The circles represent PINS discretisation points, while crosses represent GPOPS-II points.

The solution given by the two solvers on a mesh with 100 discretisation points is shown in Figure 1. Settings used for GPOPS-II are reported in Table 2. It is noted that PINS uses an equally spaced grid while GPOPS-II puts the discretisation points at the Legendre–Gauss–Radau points [67], according to the LGR method therein implemented. Figure 1 highlights the bang-bang behaviour of the control δ , which switches from $-\delta_{\max}$ (steer opposite to the direction of turning) to $+\delta_{\max}$ (steer in the direction of turning) at a time equal to $t^* \approx 0.47t_f \approx 5 \times 10^{-2}$ s. Both solvers require one mesh interval to capture the complete change in the control; for this reason, GPOPS-II takes three discretisation points to capture the control switch¹, while PINS takes only two. The roll rate in Figure 1 shows an apex in correspondence of t^* , as consequence of the change in the control.

The accuracy of the target value t_f provided by the two solvers is shown in Figure 2 as a function of the number of discretisation points used. The error presents an wavy trend, which, at a first glance, may appear quite strange, as one would expect that the higher is the number of discretisation points the more accurate is the solution. Taking into consideration

Table 2. GPOPS-II settings.

Variable	Value
Minimum collocation points	2
Maximum collocation points	2
IPOPT tolerance	10^{-8}
Method	RPM-differentiation
Scaling	Automatic-bounds
Linear solver	ma57 [68]

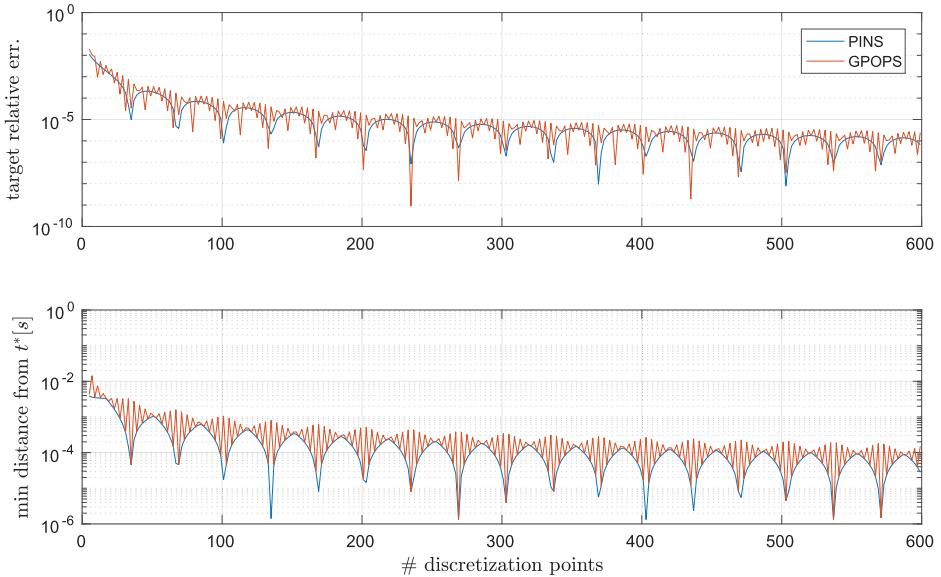


Figure 2. The relative error to the exact final time t_f (top plot) and the distance d^* of the time switching instant t^* from the nearest discretisation points (bottom plot) are shown as function of the number of discretisation points. The two plots suggest that the solution accuracy is dominated by the capability to capture the bang–bang trend in the control δ .

that the control δ has a bang-bang behaviour, the accuracy of the target value t_f strictly depends on the ability of the integration scheme to capture the discontinuity of the solution. In particular, the capability of the integration scheme to capture the control discontinuity depends on the distance d^* of the exact switching time t^* from the nearest mesh point: the higher d^* is, the less accurately the control discontinuity is captured. The distance d^* is shown as function of the number of discretisation points in the bottom plot of Figure 2. It is evident that the wavy trend of the target error is exactly the same of the distance d^* , thus the accuracy of the solution is dominated by the capability to capture the discontinuity of the δ control at t^* .

In order to better analyse the accuracy of the solution, the previous analysis has been re-performed using a large and fixed number of discretisation points near the switching time t^* . More precisely, 10^4 fixed discretisation points has been used in the interval $0.46t_f \leq t \leq 0.48t_f$, while only the number of discretisation points N in the rest of the domain has been changed. The results obtained are presented in Figure 3: GPOPS-II shows a steep

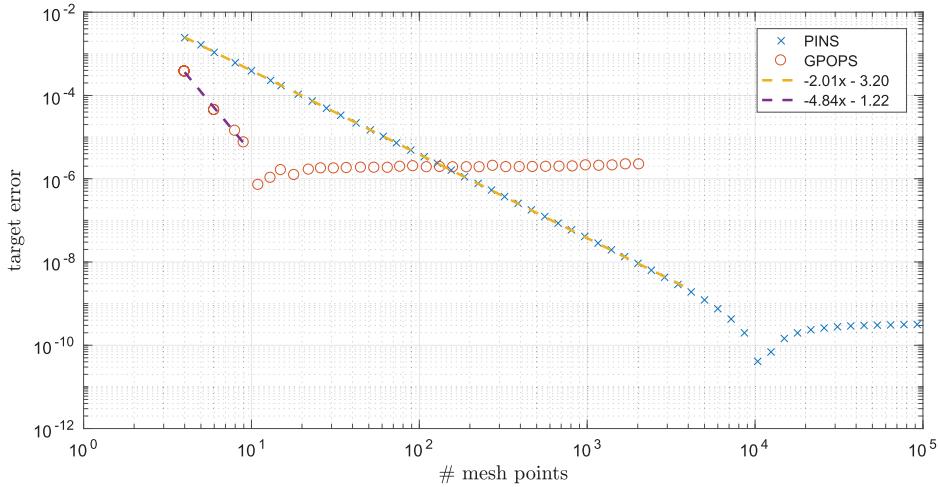


Figure 3. The relative error to the exact final time t_f is shown as function of the number of discretisation points used in the intervals $t \leq 0.46t_f$ and $t \geq 0.48t_f$. In the interval $0.46t_f \leq t \leq 0.48t_f$, 10^4 fixed discretisation points have been used. It can be noted that the straight lines fitting the error trends have a slope which is in agreement with the integration accuracy order of the two solvers.

decrease of the target error, and even with few points ≈ 10 it reaches a relative error of only 10^{-6} . The fitting highlights an accuracy, which goes as $\approx N^{-4.8}$. For a number of mesh points approximately greater than 10, the solution error does not decrease further and remains stuck at $\approx 10^{-6}$ even at $N = 10^3$. PINS differently presents a less rapid decrease of the solution error $\approx N^{-2}$, but increasing the number of mesh points it reaches a lower error; with $N \approx 10^4$ its error is $\approx 10^{-10}$. Numeric noise does not allow PINS to go below such error.

The authors increased the number of mesh point in the interval $0.46t_f \leq t \leq 0.48t_f$ (up to 10^5) and even tuned GPOPS-II parameters (e.g. using auto mesh refinement and/or lowering IPOPT tolerances, and/or imposing a mesh point at the switching point $t = t^*$) in order to achieve a better accuracy for the GPOPS-II solution. However, the best accuracy obtained was always not lower than $\approx 10^{-7}$. Moreover, when trying to increase the solution accuracy in GPOPS-II using a very fine mesh (either with fixed points or using the automatic mesh refinement), the solution so obtained suffered of control oscillations near the switching point $t = t^*$, as shown in Figure 4. It is opinion of the authors that if the controls could be analytically solved (as it is done in PINS) these oscillations may be reduced and the accuracy of the solution increased. However this cannot be done in GPOPS-II since the Lagrange multipliers are not available when the evaluation of the first-order equations is performed.

This case of study shows that GPOPS-II uses a better integration scheme that allows to achieve a good accuracy (relative to PINS) when the solution is smooth and few discretisation points are used. On the other side PINS demonstrates to be more accurate with a relative large number of discretisation points and to be able to get closer to the exact solution.

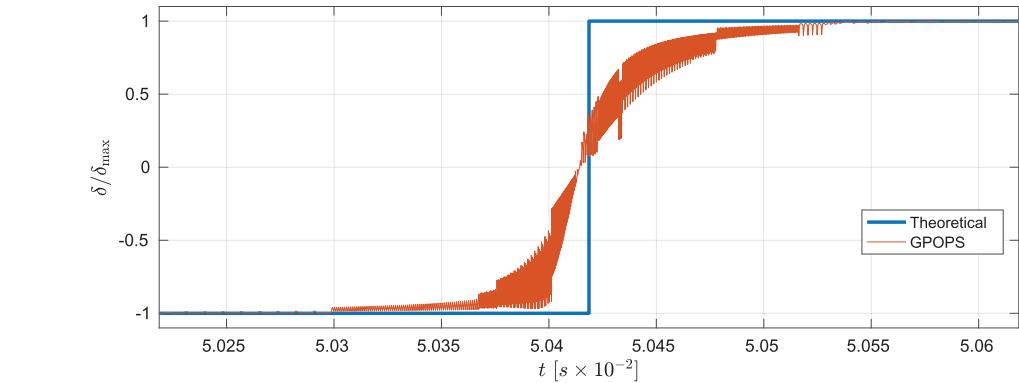


Figure 4. Detail of GPOPS-II solution control near the switching point t^* . The solution is calculated on a fine mesh (mesh tolerance error $\leq 10^{-11}$). The control is affected by noticeable oscillations that prevent the solution final time accuracy to be less than 10^{-7} .

3.2. Track reconstruction

The second OCP test consists in the reconstruction of a circuit from experimental data. In lap time simulations it is common to use a curvilinear abscissa approach [28,69,70] to track the vehicle position along the circuit. Thus the knowledge of the track geometry is fundamental for such purpose. The road geometry is given as the road orientation matrix \mathbf{R} (or, equivalently, the road curvatures) and the road width r_w as a function of the travelled space, i.e. the centre-line curvilinear abscissa s . The 3D coordinates of the track borders can be used to extract such informations.

Following the approach presented in [28], the road can be described by means of seven variables: the three coordinates of the road centre-line position $C(s) = [x(s), y(s), z(s)]^T$, together with the three angles determining the orientation matrix $\mathbf{R}(s) = \mathbf{R}(\theta(s), \sigma(s), \beta(s))$ and one additional variable for the road width $r_w(s)$. The orientation matrix \mathbf{R} is given by three successive rotations. Among the different alternatives, in this work the $z-y-x$ convention is employed, i.e. first a rotation of an angle θ about z axis, then of an angle σ about y axis, and then of an angle β about x axis:

$$\mathbf{R} = \mathbf{R}_z(\theta(s))\mathbf{R}_y(\sigma(s))\mathbf{R}_x(\beta(s)), \quad (12)$$

where θ is the track heading, σ is the road slope, β is the banking and $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ are three rotation matrices:

$$\begin{aligned} R_x(x) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(x) & -\sin(x) \\ 0 & \sin(x) & \cos(x) \end{bmatrix}, & R_y(x) &= \begin{bmatrix} \cos(x) & 0 & -\sin(x) \\ 0 & 1 & 0 \\ \sin(x) & 0 & \cos(x) \end{bmatrix} \\ R_z(x) &= \begin{bmatrix} \cos(x) & -\sin(x) & 0 \\ \sin(x) & \cos(x) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (13)$$

An effective approach to reconstruct the road geometry C, \mathbf{R}, r_w from a 3D map of the circuit is presented in [71] where the track reconstruction process is translated into an OCP, whose output are the functions $x(s), y(s), z(s), \theta(s), \sigma(s), \beta(s)$ and $r_w(s)$ that best matches

the known 3D map of the circuit. More precisely, the procedure can be summarised as follows:

- given the 3D map of the circuit, the coordinates of the road left ($x_{l0}(\zeta_0), y_{l0}(\zeta_0), z_{l0}(\zeta_0)$) and right borders ($x_{r0}(\zeta_0), y_{r0}(\zeta_0), z_{r0}(\zeta_0)$) are extracted as a function of the estimated curvilinear abscissa ζ_0 ;
- the calculated road angles θ, σ, β , the road width r_w and the curvilinear abscissa ζ are those such that the resulting road left (x_l, y_l, z_l) and right (x_r, y_r, z_r) borders minimise the squared error from the known road data ($x_{l0}, y_{l0}, z_{l0}, x_{r0}, y_{r0}, z_{r0}$).

This corresponds to an OCP, where the target to minimise is the error between the road borders and the reference borders obtained from the 3D map. The states of the OCP are the variables determining the road position, orientation and width, the controls are the rate of change of the road orientation and width:

$$\begin{aligned} & \underset{\boldsymbol{u}}{\text{minimise}} \quad \int_0^{\max(\zeta_0)} S(\boldsymbol{x}) + w_u U(\boldsymbol{u}) \, d\zeta_0 \\ & \text{subject to : } \frac{d\boldsymbol{x}}{d\zeta_0} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \end{aligned} \quad (14)$$

where

$$\begin{aligned} \boldsymbol{u}^T &= [u_\theta, u_\sigma, u_\beta, u_w, u_s] \\ \boldsymbol{x}^T &= [\theta, \sigma, \beta, \hat{\theta}, \hat{\sigma}, \hat{\beta}, r_w, x, y, z, \zeta] \\ \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) &= [\hat{\theta}, \hat{\sigma}, \hat{\beta}, u_\theta, u_\sigma, u_\beta, u_w, \cos \theta \cos \sigma (1 + u_s), \\ & \quad \sin \theta \cos \sigma (1 + u_s), -\sin \sigma (1 + u_s), 1 + u_s] \\ U(u_\theta, u_\sigma, u_\beta, u_w, u_s) &= \left(\frac{u_\theta}{\sigma_\theta} \right)^2 + \left(\frac{u_\sigma}{\sigma_\sigma} \right)^2 + \left(\frac{u_\beta}{\sigma_\beta} \right)^2 + \left(\frac{u_w}{\sigma_w} \right)^2 + \left(\frac{u_s}{\sigma_s} \right)^2 \\ S(\boldsymbol{x}) &= (x_r - x_{r0})^2 + (y_r - y_{r0})^2 + (z_r - z_{r0})^2 \\ & \quad + (x_l - x_{l0})^2 + (y_l - y_{l0})^2 + (z_l - z_{l0})^2 \\ x_r &= x + (r_w/2)(\cos \theta \sin \sigma \sin \beta - \sin \theta \cos \beta) \\ y_r &= y + (r_w/2)(\sin \theta \sin \sigma \sin \beta - \cos \theta \cos \beta) \\ z_r &= z + (r_w/2) \cos \sigma \sin \beta \\ x_l &= x - (r_w/2)(\cos \theta \sin \sigma \sin \beta - \sin \theta \cos \beta) \\ y_l &= y - (r_w/2)(\sin \theta \sin \sigma \sin \beta - \cos \theta \cos \beta) \\ z_l &= z - (r_w/2) \cos \sigma \sin \beta. \end{aligned}$$

This formulation allows to remove the noise affecting the input data (3D road map) by weighting the controls in the target function: the greater w_u is, the more the controls are penalised and the more the output data is filtered. That the optimal problem in Equation (12) can be easily modified to obtain a simpler problem to reconstruct 2D flat tracks: the

only modifications that are required are the removal of the variables related to the road elevation (σ , β , $\hat{\sigma}$ and z) and the removal of the related controls (u_σ , u_β).

The OCP (12) is in the form of a constrained least-square problem, where the constraints are originated by the first-order equations $dx/d\zeta_0 = f(x, u)$. No path (state) constraints are added to the OCP formulation (12), therefore the actual OCP solved by PINS and GPOPS-II is exactly the same (even if on slightly different meshes).

3.2.1. Solution analysis

PINS and GPOPS-II have been tested on the reconstruction problem of four different circuits, two three-dimensional and two two-dimensional: Adria (Italy, 2D), Montmelo (Spain, 2D), Imola (Italy, 3D) and Mugello (Italy, 3D). The numeric dataset used to feed equations (14) are reported in Table 3; GPOPS-II settings are the same of the previous example (see Table 2). A mesh grid size of 1 m has been used for all the four tracks.²

Figure 5 gives an overview of the reconstructed Mugello circuit, where the elevation variations along the track are noticeable. The detail of the RACC chicane of Montmelo circuit is shown in Figure 6. The reconstructed road borders (blue line for PINS, yellow for GPOPS-II) well matches the reference points (purple dots), moreover there is no noticeable difference in the solution provided by the two solvers. The resulting road curvature (bottom plot) highlights that the difference between the two solutions is approximately of the 1%.

The difference between the solutions obtained with the two solvers are relatively very small (up to $\approx 1\%$) and they are due to the different integration scheme adopted. Indeed, as previously said, in the track reconstruction problem there are no path constraints, which means that no penalty terms are used in the indirect approach OCP formulation. Therefore, the minimisation problem obtained through the indirect and direct approach is exactly

Table 3. Parameters used in the track reconstruction problem.

Variable	Value
w_u	1
σ_θ	7.0×10^{-3}
σ_σ	2.4×10^{-5}
σ_β	2.4×10^{-5}
σ_w	0.32
σ_s	0.20

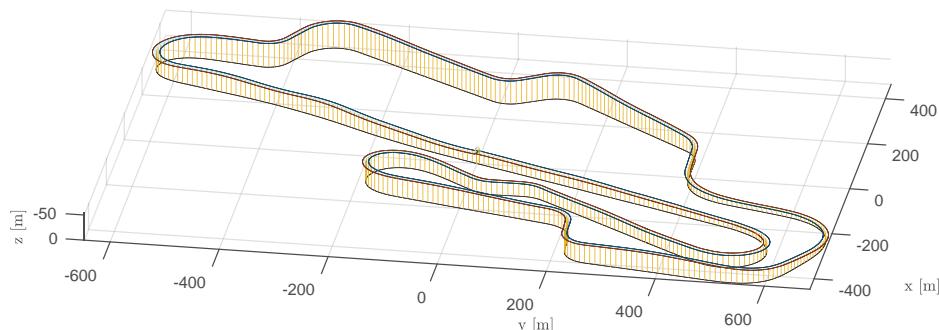


Figure 5. Overview of the 3D Mugello circuit.

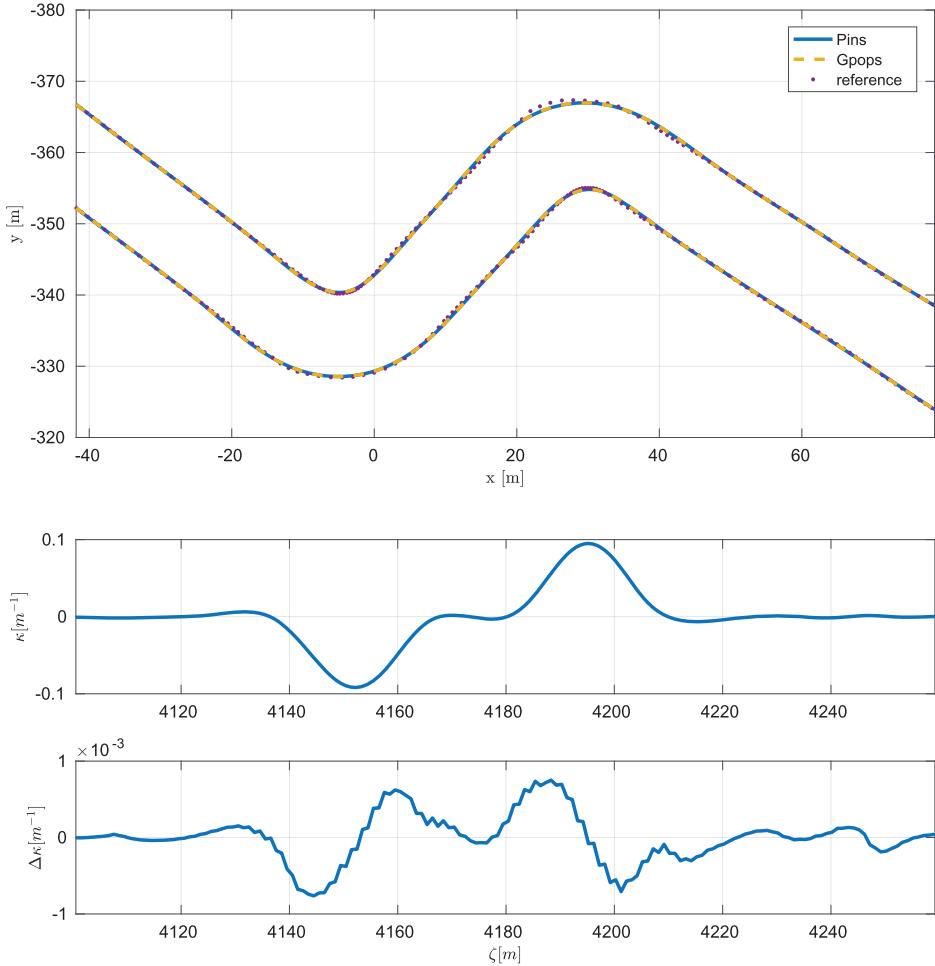


Figure 6. Detail of the RACC chicane of Montmelo circuit. In the top image, the road border reconstructed by PINS and GPOPS-II well matches the experimental data (purple dots) and no difference between the two solvers can be noticed by eye. The resulting road curvature (centre and bottom plots) shows that the difference between the two solutions is of approximately 1%.

the same. The detail of the difference between the solutions provided by the two solvers is reported in Table 4: for each circuit, the maximum error from the reference track borders and the root mean square of such error is calculated. Differences in the solutions are due to the different integration scheme.

Since the minimisation problem arising from the track reconstruction problem is exactly the same for PINS and GPOPS-II, it has been chosen as a test case for the robustness of the solver with respect to perturbation of the initial guess. The robustness has been tested through the following procedure:

- (1) A reference guess is first generated both for the states x^{ref} and controls u^{ref} . The reference guess is characterised by the following state initialisation: $x_r^{\text{ref}} = x_{r0}$, $y_r^{\text{ref}} = y_{r0}$,

Table 4. Summary of the maximum error from the reference track borders and the root mean square error for each circuit.

Circuit	PINS	GPOPS-II	PINS-GPOPS-II ratio
Maximum error			
Adria	2.02×10^0	2.01×10^0	1.00×10^0
Montmelo	1.50×10^0	1.54×10^0	9.69×10^{-1}
Mugello	9.01×10^{-1}	8.59×10^{-1}	1.05×10^0
Imola	8.99×10^{-1}	8.93×10^{-1}	1.01×10^0
Rms error			
Adria	4.01×10^{-3}	4.00×10^{-3}	1.00×10^0
Montmelo	2.51×10^{-3}	2.55×10^{-3}	9.87×10^{-1}
Mugello	1.79×10^{-3}	1.78×10^{-3}	1.00×10^0
Imola	1.93×10^{-3}	1.92×10^{-3}	1.00×10^0

Table 5. Variable relative noise amplitudes used to generate the noisy guesses for the track reconstruction problem.

Variable	x, y, z	r_w	θ, σ, β	$\hat{\theta}, \hat{\sigma}, \hat{\beta}$
Relative amplitude	5	1	$\pi/10$	0

Note: Units are those of SI.

$z_r^{\text{ref}} = z_{r0}$, $x_l^{\text{ref}} = x_{l0}$, $y_l^{\text{ref}} = y_{l0}$, $z_l^{\text{ref}} = z_{l0}$, $\theta^{\text{ref}} = \theta_0$ and $\kappa^{\text{ref}} = d\theta_0/d\xi_0$. All other variables and controls are set to zero (i.e. $\mathbf{u}^{\text{ref}} = 0$);

- (2) for each state variable x , a relative noise amplitude, name a_x , is chosen. The relative noise amplitudes used are reported in Table 5.
- (3) for each state variable x , its guess $x^{(g)}$ is given by the sum of the reference guess x^{ref} and a noisy term. The noisy term is the product of the relative noise amplitude a_x with a global noise variable ξ and a random variable r_x : $x^{(g)} = x^{\text{ref}} + a_x \xi r_x$. The global noise variable ξ is the same for all state variables, moreover the random variable r_x is uniformly distributed in the interval $[-1, 1]$ and sampled on each mesh point;
- (4) the global noise amplitude ξ is varied from 0 to 1.5 by step of 0.1, and 10 different guesses are generated for each value of ξ , for a total of 150 different guesses;
- (5) the track reconstruction problem is then solved both with PINS and GPOPS-II using the noisy guesses. A maximum limit of 500 iterations is used; if the solver does not manage to solve the problem within this iteration limit, it is considered to fail.

The above described procedure has been repeated for each circuit. The results obtained are reported in Figure 7, where the number of iterations required to solve the problem is shown as function of the noise global amplitude ξ . A number of iterations equal to the iteration limit (500) indicates the solver has not succeeded in finding the solution. The results highlight different trends for the 2D and 3D circuits. In 2D circuits (Adria and Montmelo) GPOPS-II is more robust and manages to solve the problem with all guesses, even at $\xi = 1.5$. A certain variance is observed in the number of iterations required to calculate the solutions for a given ξ , from approximately 20 iterations to 100. PINS fails to solve the problem for a noise amplitude greater than $\xi > 1$; however, for $\xi \leq 1$ it is able to calculate the solution in few iterations (≈ 6). The variance in the number of iterations for PINS is noticeably lower than that of GPOPS-II. In 3D circuits, PINS shows almost the same behaviour as with 2D circuits. GPOPS-II on the contrary presents a noticeably higher variability: while

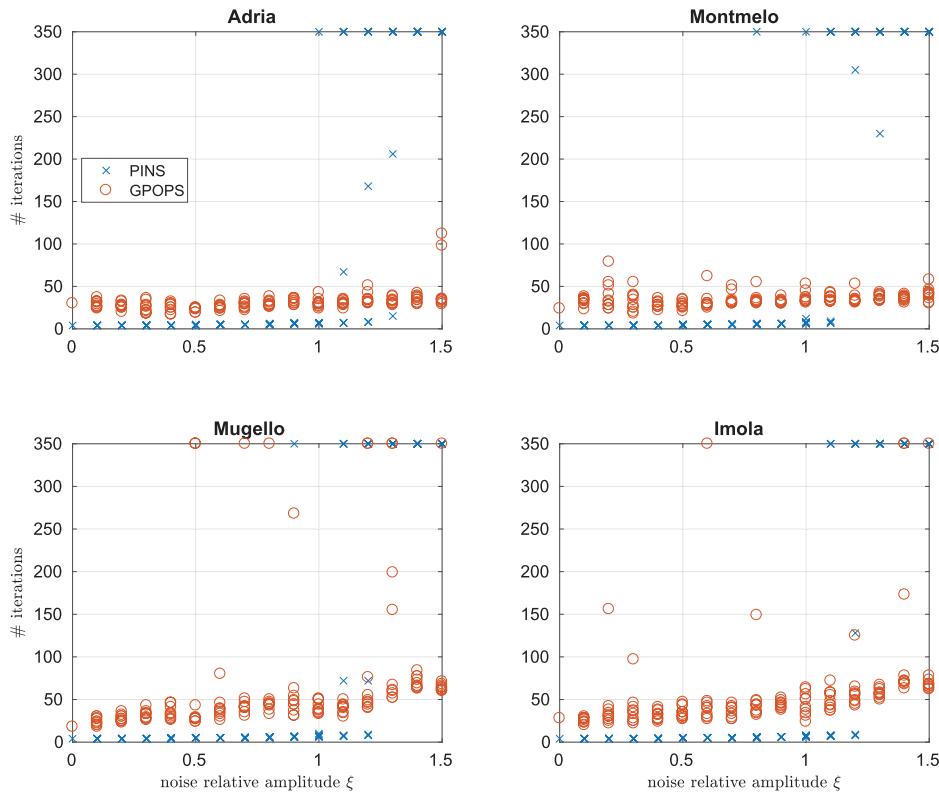


Figure 7. Robustness of the solvers with respect to noisy guess. The number of iterations required to compute the solution are plotted vs. the relative noise amplitude. An iteration limit of 500 iterations has been chosen for both solvers. Points located at an ordinate of 500 iterations refer to non-converged problems.

sometimes it still manages to solve the problem at $\xi = 1.5$, other times it fails even with a low relative noise ($\xi \approx 0.6$). On overall PINS demonstrates a more consistent behaviour, and GPOPS-II manages to handle guesses with an high noise but sometimes fails even with low-noise guesses. The greater robustness of GPOPS-II to less accurate guesses is probably due to the IPOPT initialisation procedure, and the heuristic therein used, to estimate the initial values for the lagrange multipliers. On the contrary the current release of PINS does not implement any of such a procedure and sets the lagrange initial values to zero. As a general result that authors have drawn there is that the lagrange multiplier initialisation is crucial for the robustness of both direct and indirect optimal control solvers.

3.3. Lap time simulation

The last OCP test that is analysed is the minimum lap time simulation of a racing GT car on the Adria International Raceway.

The simulation is performed with the well known 3-dof car model, which comprises QSS load transfers and load-dependent tyre adherence. Such car model is selected because it can be found on most classic vehicle dynamics books, e.g. [72–76]. Moreover similar

models have been extensively used for optimal control minimum lap time simulations, e.g. in [35,38,39,77] and with the addition of the wheel spin dynamics in [10,11,46,78]. The indirect and direct methods here presented have been used for minimum lap time simulations and validation also with significantly more complex car models [29,39]. However, the authors decided to keep the model relatively simple to let other researchers easily reproduce the reported results. The choice is in line with the main purpose of this work to compare the performance of the two different OCP methods, rather than computing accurate lap time simulations with the most complex car model available.

The three degrees of freedom of the car chassis are: the speed V , the sideslip angle λ and the yaw rate Ω . The car position along the circuit is tracked by means of three variables: the curvilinear abscissa s , the lateral displacement from the road centre line n and the heading angle α w.r.t. the road centre line. The first-order equations for the above mentioned six variables are:

$$\begin{aligned}\dot{s} &= \frac{V \cos(\alpha - \lambda)}{1 - n\kappa} \\ \dot{n} &= V \sin(\alpha - \lambda) \\ \dot{\alpha} &= \Omega - \frac{\kappa V \cos(\alpha - \lambda)}{1 - n\kappa}\end{aligned}\tag{15}$$

$$\begin{aligned}M\Omega V\lambda + M\dot{V} &= S_{rr} + S_{rl} + S_{fr} + S_{fl} - \delta(F_{fr} + F_{fl}) - D \\ M(\Omega V - \dot{V}\lambda - V\dot{\lambda}) &= \delta(S_{fr} + S_{fl}) + F_{rr} + F_{rl} + F_{fr} + F_{fl} \\ I_z\dot{\Omega} &= a(F_{fr} + F_{fl}) - b(F_{rr} + F_{rl}) + t_w(-S_{rr} + S_{rl} - S_{fr} + S_{fl}),\end{aligned}$$

where M is the car mass, S_{ij} is the tyre longitudinal force where $i=f,r$ indicates the front or rear tyre, and $j=r,l$ indicates the right or left side, F_{ij} is the tyre lateral force, δ is the steering angle, D is the drag force, I_z is the yaw inertia moment, a and b are, respectively, the distance of the front and rear axle from the centre of gravity and t_w is the car half width. In Equations (15) the simplifications $\cos\xi \approx 1$ and $\sin\xi \approx \xi$ have been adopted for the sideslip λ and steering δ angles.

The drag force D is proportional to the square of the speed:

$$D = \frac{1}{2}\rho C_a V^2,\tag{16}$$

where ρ is the air density, C_a is the drag coefficient.

Tyre lateral forces are computed from slip angles, while longitudinal forces are computed from a single control u_x , which is related to the normalised thrust:

$$\begin{aligned}S_{fl} = S_{fr} &= \frac{Mg}{2}f^-(u_x)\beta, \quad S_{rl} = S_{rr} = \frac{Mg}{2}(f^+(u_x) + f^-(u_x)(1 - \beta)), \\ F_{ij} &= N_{ij}K_\lambda\lambda_{ij},\end{aligned}\tag{17}$$

where f^+ and f^- return, respectively, the positive and negative part of the argument, β is the front braking bias, N_{ij} is the tyre load, λ_{ij} is the tyre sideslip angle and K_λ the tyre sideslip stiffness. Tyre saturation related to adherence limitations is accounted for using

a load dependent friction ellipse – see Section 3.3.1 – this makes the tyre model effectively nonlinear although approximate. The tyre sideslip angles are given by the following expression:

$$\begin{aligned}\lambda_{\text{rr}} &= \lambda + \frac{\Omega(b + \lambda t_w)}{V} & \lambda_{\text{fr}} &= \lambda + \delta - \frac{\Omega(a - \lambda t_w)}{V}, \\ \lambda_{\text{rl}} &= \lambda + \frac{\Omega(b - \lambda t_w)}{V} & \lambda_{\text{fl}} &= \lambda + \delta - \frac{\Omega(a + \lambda t_w)}{V}.\end{aligned}\quad (18)$$

The tyre loads N_{ij} depend on the delayed vehicle longitudinal a_x and lateral a_y accelerations:

$$\begin{aligned}N_{\text{rr}} &= \frac{Mg}{2} \frac{a}{a+b} + \frac{Mg}{4} \left(\frac{a_x h}{a+b} - a_y (1-\chi) \frac{h}{t_w} \right) \\ N_{\text{fr}} &= \frac{Mg}{2} \frac{b}{a+b} + \frac{Mg}{4} \left(-\frac{a_x h}{a+b} - a_y \chi \frac{h}{t_w} \right) \\ N_{\text{rl}} &= \frac{Mg}{2} \frac{a}{a+b} + \frac{Mg}{4} \left(\frac{a_x h}{a+b} + a_y (1-\chi) \frac{h}{t_w} \right) \\ N_{\text{fl}} &= \frac{Mg}{2} \frac{b}{a+b} + \frac{Mg}{4} \left(-\frac{a_x h}{a+b} + a_y \chi \frac{h}{t_w} \right),\end{aligned}\quad (19)$$

where χ is the roll stiffness. The longitudinal a_x and lateral a_y accelerations follow the actual vehicle accelerations with a low band pass filter of time constant τ_{a_x} and τ_{a_y} in order to simulate the suspension load transfer lag:

$$\tau_{a_x} \dot{a}_x + a_x = \dot{V} + \Omega V \lambda \quad \tau_{a_y} \dot{a}_y + a_y = \Omega V - (\dot{V} \lambda). \quad (20)$$

Summarising, the state space model comprises eight dof ($V, \lambda, \Omega, s, n, \alpha, a_x, a_y$), with the corresponding eight first-order equations (15), (20) and two controls (u_x, δ).

3.3.1. OCP constraints

The minimum lap time problem includes some constraints, which ensures that the power used is less than the a maximum threshold, that the car never exceeds the track boundaries, and that the tyre forces are less than the tyre maximum adherence. Such constraints can be expressed as follows:

$$\begin{aligned}c_p &= \frac{V(S_{\text{rr}} + S_{\text{rl}})}{P_{\max}} \leq 1 & c_n &= \frac{n}{n_{\max}} \leq 1 & c_n &= \frac{-n}{-n_{\max}} \leq 1 \\ c_{t_{ij}} &= \left(\frac{S_{ij}}{N_{ij} \mu_{ij}^x} \right)^2 + \left(\frac{F_{ij}}{N_{ij} \mu_{ij}^y} \right)^2 \leq 1 & i &= f, r & j &= r, l,\end{aligned}\quad (21)$$

where P_{\max} is the engine maximum power, n_{\max} is the maximum lateral displacement which is equal to half of the road width r_w minus half of the car width $n_{\max} = r_w/2 - t_w$. μ_{ij}^x and μ_{ij}^y are, respectively, the tyre longitudinal and lateral adherence, which depend on

Table 6. Car dataset used for the minimum lap time problem.

Variable	Value	Units	Description
M	1184	kg	Total mass
p	2.76	m	Wheelbase
b	1.404	m	Wheelbase to front axis distance
b	1.356	m	wheelbase to rear axis distance
t_w	0.807	m	Half track width
h	0.4	m	CoM height from ground
I_z	1775	kgm^2	Yaw inertia moment
β	0.62	–	Braking bias
χ	0.5	–	Roll stiffness
ρ	1.2	kg/m^3	Air density
CdA	0.88	m^2	Drag coefficient
μ_0^x	1.68	–	Tyre longitudinal adherence
μ_0^y	1.68	–	Tyre lateral adherence
K_μ	-0.5	–	Tyre adherence variation with load
K_λ	44	–	Tyre lateral stiffness
τ_{a_y}	0.2	s	Lateral load transfer time constant
τ_{a_x}	0.2	s	Longitudinal load transfer time constant
P_{\max}	215	kW	Maximum power

the tyre loads:

$$\mu_{ij}^x = \mu_0^x + K_\mu \frac{N_{ij}}{N_{0,ij}} \quad \mu_{ij}^y = \mu_0^y + K_\mu \frac{N_{ij}}{N_{0,ij}} \quad i = f, r \quad j = r, l, \quad (22)$$

where $N_{0,ij}$ is the tyre load in static conditions and K_μ is constant factor. The numerical data used to feed the car model is reported in Table 6.

3.3.2. Solution analysis

The above described car model has been used to simulate the lap time of a GT car on the Adria International Raceway. A mesh with 1 discretisation point per metre has been used.³ The lap time calculated by PINS and GPOPS-II are, respectively, 75.721 s and 75.429 s, with a relative difference of the approximately 0.3%.

The simulated speed profile is shown in Figure 8; the speed difference between the two solutions is always less than 3 km/h. In general, PINS simulated speed is higher than GPOPS-II one in the middle of the turns but it is lower in the straights and in the first part of the braking manoeuvres. The reason can be found in the use of the penalties, which in general do not allow to reach the control exact limits producing less braking/tractive forces. As a consequence the optimal solution found try to compensates maximising the speed in each corner looking for a slightly larger curvature radius.

The effect of the use of the penalties emerges when the optimal trajectories are compared (see Figure 9). The two solutions show a good agreement along all the circuit, except in the two straights track sections comprised between the red circles (the first) and between the green ones (the second). The difference in the trajectory is more evident in the car lateral displacement n from the road centre line, which is reported in the top plot of Figure 10. In the two mentioned track sections it is evident that PINS trajectory moves towards the road centre line, while GPOPS-II remains close to the track border in the two straights. In other words PINS tends to reduce the penalty (associated to the road border constraints,

see Equation (21)) by moving to the road centre line. It is opinion of the authors that this effect is probably evident in these two sections because there this manoeuvre affects only marginally the lap time.

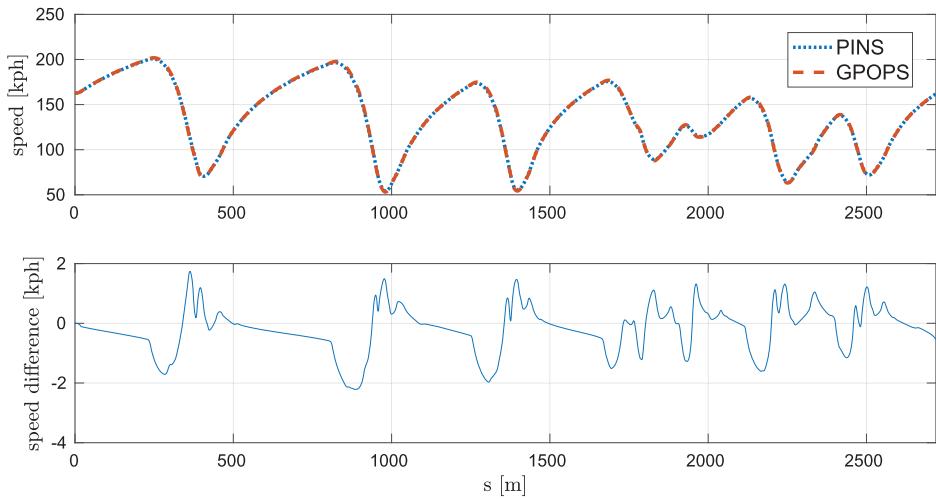


Figure 8. Simulated speed profile and speed difference (PINS minus GPOPS-II) along Adria circuit; the difference between the two solvers is always less than 3km/h. In general, PINS simulated speed is higher than GPOPS-II one in the middle of the turns but it is lower in the straights and in the first part of the braking manoeuvre.

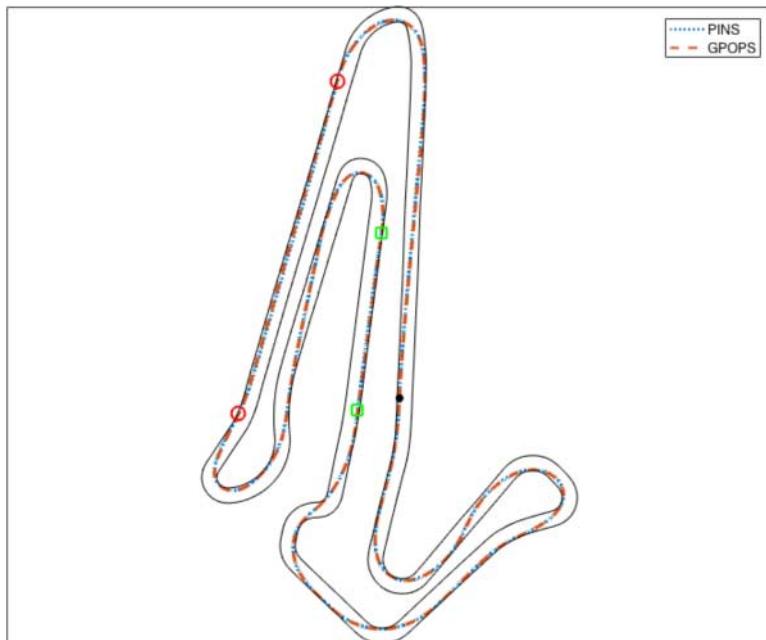


Figure 9. Simulated optimal trajectory along Adria circuit. The most significant trajectory difference is in the two intervals comprised between the circles and squares.

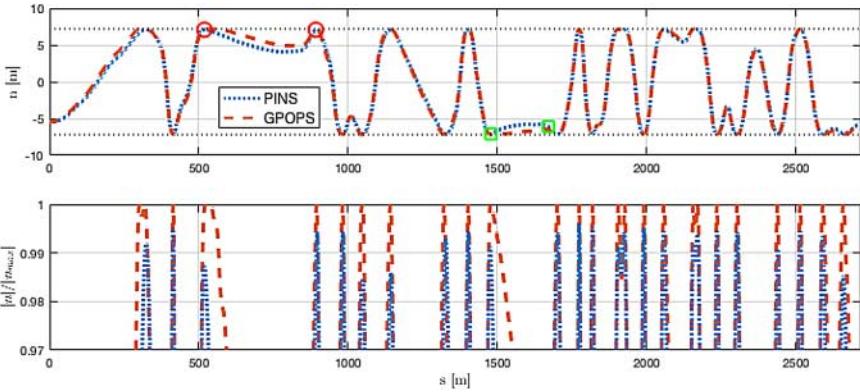


Figure 10. In the top figure, the lateral displacement n from the road centre line is shown. The trajectory differences in the two straight sections comprised between the circles and squares are noticeable. The bottom plot focuses on the road borders constraint enforcement $|c_n|$ (21) ($|c_n| = 1$ is the constraint limit). It can be noted that, while GPOPS-II solution arrives at a constraint value $|c_n| = 1$, PINS one does not go beyond $|c_n| = 0.995$.

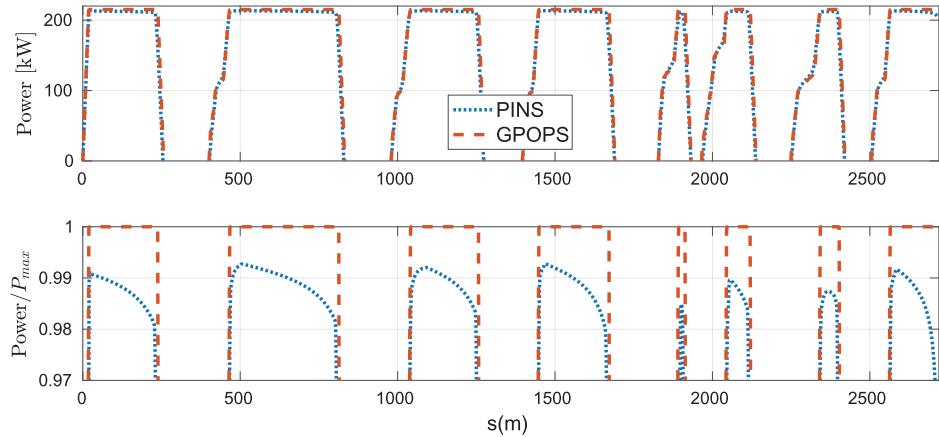


Figure 11. The power used $(S_{rr} + S_{rl})V$ is shown in the top plot; the two solutions present almost identical trends and both get close to the maximum power limit. The bottom plot focuses on the power-limit-constraint enforcement c_p (21) ($c_p = 1$ is the constraint limit). It is possible to notice that, while GPOPS-II arrives at $|c_p| = 1$, PINS arrives up to $c_p \approx 99\%$.

The bottom plot of Figure 10 shows the enforcement of the road borders constraint c_n (see Equation (21)). It is possible to notice that, while GPOPS-II solution almost touches the road borders ($|c_n| = 1$), PINS one does not go beyond $|c_n| = 0.995$. The same behaviour can be observed also in the other OCP constraints: the maximum power limit and the tyre adherence limit. The former constraint is shown in Figure 11; the bottom plot highlights that PINS solution uses up to $\approx 99\%$ of the maximum power, while GPOPS-II one uses up to 100% (at least to machine precision). Moreover, similar conclusions can be stated for the tyre engagement constraints c_{tij} (21), which are reported in Figure 12; in this case, PINS solution arrives up to $\approx 99.7\%$ of the maximum value.

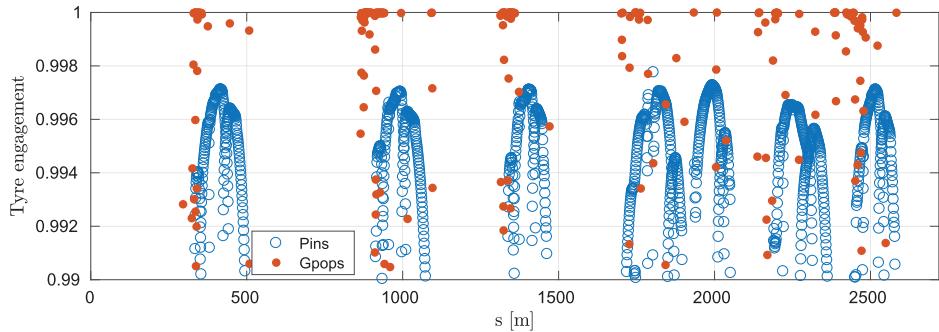


Figure 12. The tyre engagement c_{tj} (21) of the four wheels are shown as function of the curvilinear abscissa. Again, GPOPS-II solution touches the constraint limit, while PINS arrives up to the $\approx 99.7\%$ of the maximum value.

The fact that PINS solution does not reach the exact constraints bounds (track width, engine power, tyre adherence) is a consequence of the penalty approach used by indirect methods. However even IPOPT, that is based on an Interior Point algorithm, uses a similar penalty approach [33]. The main difference between PINS and IPOPT is that the latter implements an algorithm that automatically sharpens the penalties in order to better satisfy the constraints, while in the former the penalties can be fine-tuned only manually. Since in PINS this is achieved with a continuation procedure, it usually requires longer computational time and diminishes the robustness to convergence using a non-optimal tuning strategy. The values that have been used are, according to authors' opinion, the best compromise between constraint enforcement, robustness of the solver and low computational times.

At the beginning of this section it has been stated that PINS simulated lap time is approximately 0.3 s higher than GPOPS-II one, then the results showed that this difference can be attributed to the lower engine power and tyre engagement usage of PINS solution. While 0.3 s may seem a consistent discrepancy from an engineering point of view, it should be remembered that the absolute performance resulting from lap time simulations is not of primary importance because it depends on some parameters that are difficult to measure, first of all the friction between tyres and asphalt. Usually, the numeric dataset used to feed the mathematical model is tuned so as to make the simulated car performance match the telemetry data. Only after this calibration process is performed, the model is used for simulations and optimisation. Therefore, in lap time simulations the sensitivity to model parameter variations is much more important than the absolute lap time.

The sensitivity of the two solver has been compared on the optimisation of the braking bias β and roll stiffness χ . Simulations have been performed varying the braking bias and the roll stiffness, respectively, in the range $[0.55, 0.68]$ and $[0.63, 0.8]$; the lap times differences between PINS and GPOPS-II solutions are shown in Figure 13. One may note that, while the difference in the simulated lap time varies in the range $[0.29 \text{ s}, 0.32 \text{ s}]$, the location of the best lap time does not differ relevantly between the two solvers. In particular PINS minimum lap time is achieved with a braking bias of $\beta = 0.595$ and a roll stiffness of $\chi = 0.741$ (with an accuracy of 0.001 both for β and χ), while GPOPS-II minimum lap time is obtained for the same value of the rolling stiffness but for a braking

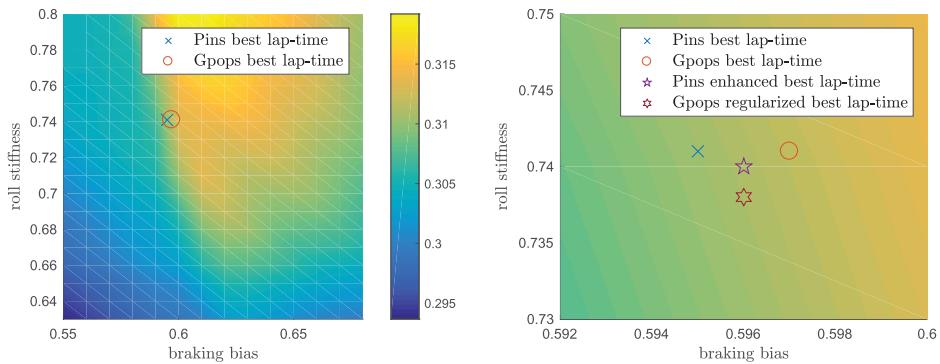


Figure 13. The difference of the simulated lap time (PINS minus GPOPS-II) is shown as function of the braking bias β (x-axis) and roll stiffness χ (y-axis). The location of the best lap time is also shown. The right plot focuses on a smaller range of the variables β and χ in order to better highlight the differences in the location of the best lap time. The best lap time for the PINS ‘enhanced’ and GPOPS-II ‘regularised’ models are also shown.

bias slightly higher, $\beta = 0.597$. This discrepancy in the location of optimum value of the parameters is mainly due to two different causes. The first is the different simulated performance: since PINS solution uses slightly less engine power and tyre adherence, this results in a different car dynamics and thus optimal braking bias and roll stiffness. The second cause is the control regularisation induced by the control penalties⁴ that makes the controls’ solution to be smoother, thus modifying again the simulated car dynamics. In order to verify that the different location of the best lap time is due to these two causes, the PINS model has been modified by increasing the engine maximum power and the tyre adherence and by augmenting the track width, so as to make the constraints actually reach their physical limits. In particular, the engine maximum power has been increased by a 1.015 factor, the tyre adherence by 1.004, and the track width by 1.01, in accordance with the distance from the constraints boundaries highlighted in Figures 10–12. This model, referred to as PINS “enhanced” model, shows a best lap time for $\beta = 0.596$ and $\chi = 0.74$, thus it is located closer to GPOPS-II minimum lap time but differences are still present. Finally GPOPS-II model has been regularised by adding a small regularisation term proportional (1×10^2) to the square of the controls into the Lagrange target; this model is referred to as GPOPS-II ‘regularised’. The regularisation is not exactly the same as that of PINS, yet tries to mimic it. The optimum parameters for the GPOPS-II regularised model are $\beta = 0.596$ and $\chi = 0.738$, as shown in the right plot in Figure 13. These results shows that the car design optimisations performed by PINS and GPOPS-II are in agreement if the same car performance is simulated and similar regularisation is adopted.

The robustness of the two methods with respect to perturbation of the initial guess has been studied with the same procedure described in Section 3.2.1. Differently from Section 3.2.1, the minimum lap time OCP includes constraints that are treated with different approaches by direct and indirect methods. The reference (i.e. without noise) guess is characterised by a non-zero guess speed $V_0 = 30$ kph, while all the other variables are left to zero. The noise relative amplitudes a_x for each state variable are reported in Table 7, and the noise global amplitude ξ spans from 0 to 1 by steps of 0.1. Moreover, the speed

Table 7. Variable relative amplitudes used to generate the noisy guesses for the minimum lap time problem.

Variable	n	α	V	λ	Ω	ax	ay	u_x	δ
Relative amplitude	7	0.6	60	0.05	1	10	15	1	0.15

Note: Units are those of SI.

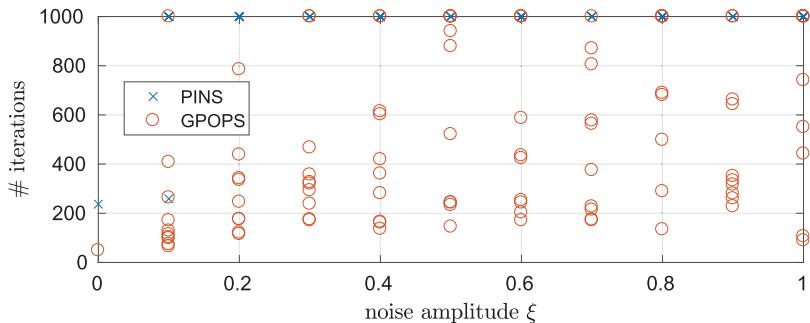


Figure 14. Robustness of the solvers with respect to noisy guess. The number of iterations required to compute the solution are plotted vs. the relative noise amplitude. An iteration limit of 10^3 iterations has been chosen for both solvers. Points located at an ordinate of 10^3 iterations refer to non-converged problems.

V is forced to be greater than 1 m/s: when the noised speed results to be less than such threshold, it is set to 1 m/s. This allows to avoid numerical singularities in the first-order equations, which can not be made explicit ODE in the curvilinear abscissa domain s when $V = 0$. The solver iterations limit is now increased to 10^3 since the problem is more difficult to solve.

The results obtained are summarised in Figure 14, and are significantly different from those of the reconstruction problem in Section 3.2.1. Even with a low noise amplitude $\xi = 0.1$ PINS fails to find the solution most of the times, and from $\xi = 0.3$ it is never successful. GPOPS-II, like in Section 3.2.1, shows a more variable behaviour and for all values of ξ it sometimes manages to find the solution, sometimes it fails. None of the two solvers appear to be robust with respect to the added noise. It is opinion of the authors that the high failure rate highlighted by PINS is consequence of the constraints violation in the provided guess. When starting from a noisy guess, PINS showed very high residual since the first iterations and then it straggle to find a feasible (i.e. with low residual) region. High values for the residual ($\approx 10^6$) are due to high penalty values (i.e. constraints violation), which make the problem highly ill conditioned and difficult to solve. Differently IPOPT, the NLP solver used for GPOPS-II, implements a feature that project the current state into the constraint-feasible region, thus even if the guess violates the constraints it automatically project it into a feasible region [33]. It is opinion of the authors that this is the main advantage that makes GPOPS-II more robust than PINS with respect to less accurate initial guesses. Finally, it should be noted that the use of more complex car models is likely to decrease the robustness of both solvers to initial guesses; the procedure here described could be likewise used to assess the resulting robustness.

4. Result discussion and note on performance

The results presented in the previous section, and in particular those of the basic motorcycle problem, confirmed that GPOPS-II implements a more accurate integration scheme (Gaussian in GPOPS-II, trapezoidal in PINS) that is able to provide a significantly higher accuracy than PINS using relatively few mesh points. Despite this, PINS is able to reach the absolute highest accuracy when using an higher number of mesh points.

Moreover GPOPS-II demonstrated to better handle perturbations (i.e. noise) in the guess, in particular when constraints are present in the OCP. Indeed, while PINS showed similar robustness results in the track reconstruction problem, it largely under performed in the minimum lap time one. It is opinion of the authors that this is mainly due to the robustness of IPOPT that implements a mechanism that project the variables into the feasible region, and adopts a reliable initialisation of the Lagrange multipliers. On the contrary, PINS uses a zero guess for the Lagrange multipliers, moreover, if the guess violates the constraints, it suffers of ill-conditioning due to the high penalty values (since constraints are not projected into a feasible region like in IPOPT).

In the minimum time problem, the constraint enforcement analysis showed that PINS solution does not reach the exact constraints boundaries (as GPOPS-II does). In particular, its solution did not make full usage of the track width, engine power and tyre adherence. This behaviour is a direct consequence of the lack of automatic tuning of the penalty weights and parameters during the convergence to the solution.

Finally, when tested on a typical utilisation scenario (i.e. optimisation of car design parameters) the two software provided similar outcomes. In particular, the location of the optimal braking bias and roll stiffness given by PINS and GPOPS-II were very close, with relative differences of the order of 10^{-3} . The small differences could be explained by the slightly different simulated car dynamics due to different constraint usage and regularisation.

In all previous test problems nothing has been stated about the solver performance related to the time required to compute the solution. A comparison of such performance for the two algorithm is not an easy task because the two solvers are developed with different programming languages. PINS is entirely written in C++ and, even if it uses a Ruby interpreter to setup the problem data, the solution algorithm and the problem function

Table 8. Solution computing times measured on a desktop computer equipped with an Intel Xeon E3-1270 v5 processor with 32 GB of ram, running on Ubuntu 16.04.

Problem	Number of mesh points	PINS time (s)	GPOPS-II time (s)	GPOPS-II to PINS time ratio
Motorcycle basic	100	0.2	0.9	4.5
Motorcycle basic	5000	7	2	0.3
Adria 2D	2719 (1 per metre)	0.1	9	90
Montmelo 2D	4650 (1 per metre)	0.1	11	110
Mugello 3D	5244 (1 per metre)	0.5	60	120
Imola 3D	4906 (1 per metre)	0.5	35	70
Car model	2719 (1 per metre)	6	50	8
Car model	54380 (20 per metre)	96	932	10
Car model	543800 (200 per metre)	1184	10,180	8.5

Notes: GPOPS-II has always been used with IPOPT as NLP solver and the ma57 linear solver[68]. PINS turns out to be significantly faster than GPOPS-II in most of the problems.

evaluation are computed by compiled code. GPOPS-II instead is developed as a MATLAB[®] library, but uses a compiled NLP solver (in this work a compiled version of IPOPT has been used). Thus, function evaluation is done at MATLAB[®] level (even if the user may compile the functions into mex files) while the NLP solution is calculated by compiled code. Thus a comparison of the solvers performance is not really meaningful to compare the efficiency of the algorithms, but it is certainly interesting from a practical (i.e. user) point of view. In the test problem analysed in this work, it has been noticed that PINS is generally significantly faster than GPOPS-II; the measured computing times are reported in Table 8.

5. Conclusions

In this work indirect and direct optimal control methods have been compared on three test problems related to vehicle OCPs. Numerical solvers PINS and GPOPS-II have been chosen, respectively, as representative for the indirect and direct approach since they are the software most used in literature for this purpose. The results showed that each software has advantages and disadvantages compared to the other. In particular, PINS excelled in computational time and absolute accuracy, while GPOPS-II resulted to be more robust and, when using coarse meshes, even more accurate. Moreover, from a user point of view, GPOPS-II may seem more user-friendly since it does not require any fine-tuning of the inequality constraints, which are automatically managed by the NLP solver. PINS instead requires the user to properly adjust the penalty parameters in order to achieve the best performance. While this is slightly time consuming, it offers a deeper control on the solution strategy and allows the user to choose whether to prefer fast solution computing or accurate constraint enforcement. However, from a general perspective, we can say that initialisation phase of the OCPs plays a major role in solver robustness. On overall results showed that indirect and direct methods on overall have a similar behaviour when dealing with minimum lap time problems; most of the observed differences between the two solvers can be explained with different numerical implementation features rather than with intrinsic differences of indirect and direct methods. Such implementation features include: integration scheme, multiplier initialisation, penalty tuning algorithm, projection of the state into feasible region.

Notes

1. Collocation points captures accurately smooth functions, but discontinuities in non-smooth solutions can be captured only at mesh interval boundaries.
2. It means that one mesh point per metre has been used in PINS, and one mesh interval every 2 m, with two collocation points per mesh interval have been used in GPOPS-II.
3. In GPOPS-II one mesh interval every 2 m and two collocation points per mesh interval have been used.
4. Control penalties are used in the indirect approach to keep the controls bounded.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Nicola Dal Bianco received the Master degree in Physics in 2013, and the PhD degree in Industrial Engineering in 2018, both from University of Padova. He is currently a research fellow, and his research fields include vehicle modelling and simulation, with particular interest in optimal control lap time simulations.

Enrico Bertolazzi received the Master degree in Mathematics from University of Trento, Trento in 1990, where he is currently Associate Professor. His research interests include numerical analysis and finite volumes for convective dominated and hypersonic flow, discrete maximum principle, constrained optimal control, and scientific programming.

Francesco Biral received the Master degree in Mechanical Engineering from the University of Padova in 1997 and the PhD in Mechanism and Machine Theory from University of Brescia in 2000. He is currently Associate Professor at the Department of Industrial Engineering at University of Trento. His research interests include symbolic and numerical multibody dynamics and optimisation, constrained optimal control, mainly in the field of vehicle dynamics with special focus on Intelligent Transportation Systems and minimum lap time problems.

Matteo Massaro is currently Associate Professor with the Department of Industrial Engineering of the University of Padova. He previously held positions as “Research Fellow” at Oxford University (UK, 2014–2015) and Padova University (2009–2013), where he graduated in 2005 (MSc) and was granted a PhD in 2009. His research themes include modelling, simulation and control of mechanical systems, with a focus on vehicle dynamics.

ORCID

Nicola Dal Bianco  <http://orcid.org/0000-0002-8892-638X>

Enrico Bertolazzi  <http://orcid.org/0000-0003-0487-5210>

Francesco Biral  <http://orcid.org/0000-0001-8098-7965>

Matteo Massaro  <http://orcid.org/0000-0001-6256-3384>

References

- [1] Scherenberg H. Mercedes-Benz racing design and cars experience; **1958**. (SAE Technical Paper; Report No.: 1958-01-01).
- [2] Metz D, Williams D. Near time-optimal control of racing vehicles. *Automatica*. **1989**;25(6): 841–857.
- [3] Gadola M, Vetturi D, Cambiaghi D, et al. A tool for lap time simulation; **1996**. (SAE Technical Paper; Report No.: 1996-12-01).
- [4] Siegler B, Deakin A, Crolla D. Lap time simulation: comparison of steady state, quasi-static and transient racing car cornering strategies. In: 2000 SAE Motorsports Engineering conference and Exposition of Automotive Engineers; SAE International Paper 2000-01-3563; **2000**. p. 9.
- [5] Brayshaw D. Use of numerical optimisation to determine on-limit handling behaviour of race cars [dissertation]. School of Engineering, Cranfield University; **2004**.
- [6] Brayshaw DL, Harrison MF. A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location. *Proc Inst Mech Eng D J Automob Eng*. **2005**;219(6):725–739.
- [7] Brayshaw DL, Harrison MF. Use of numerical optimization to determine the effect of the roll stiffness distribution on race car performance. *Proc Inst Mech Eng D J Automob Eng*. **2005**;219(10):1141–1151.
- [8] Savaresi SM, Spelta C, Ciotti D, et al. Virtual selection of the optimal gear-set in a race car. *Int J Veh Syst Model Test*. **2008**;3(1–2):47–67.

- [9] Völkl T, Muehlmeier M, Winner H. Extended steady state lap time simulation for analyzing transient vehicle behavior. *SAE International Journal of Passenger Cars-Mechanical Systems*; **2013**. p. 283–292.
- [10] Casanova D. On minimum time vehicle manoeuvring: the theoretical optimal lap [dissertation]. School of Engineering, Cranfield University; **2000**.
- [11] Kelly DP. Lap time simulation with transient vehicle and tyre dynamics [dissertation]. Cranfield University; **2008**.
- [12] Rao AV. A survey of numerical methods for optimal control. *Adv Astronaut Sci*. **2009**;135(1):497–528.
- [13] Gerdts M. A survey on optimal control problems with differential-algebraic equations. In: Ilchmann A, Reis T, editors. *Surveys in differential-algebraic equations II*. Cham: Springer International Publishing; **2015**. p. 103–161.
- [14] Biral F, Bertolazzi E, Bosetti P. Notes on numerical methods for solving optimal control problems. *IEEJ J Ind Appl*. **2016**;5(2):154–166.
- [15] Sorniotti A, Curto M. Racing simulation of a formula 1 vehicle with kinetic energy recovery system; **2008**. (SAE Technical Paper; 12. SAE International). p. 15.
- [16] Mayne D. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *Int J Control*. **1966**;3(1):85–95.
- [17] Gershwin SB, Jacobson DH. A discrete-time differential dynamic programming algorithm with application to optimal orbit transfer. *AIAA J*. **1970**;8(9):1616–1626. 2017/08/29.
- [18] Lantoine G, Russell RP. A hybrid differential dynamic programming algorithm for constrained optimal control problems. Part 2: application. *J Optim Theory Appl*. **2012 Aug**;154(2):418–442.
- [19] Lantoine G, Russell RP. A hybrid differential dynamic programming algorithm for constrained optimal control problems. Part 1: theory. *J Optim Theory Appl*. **2012**;154(2):382–417.
- [20] Pontryagin L, Boltyanskiy V, Gamkrelidze R, et al. *The mathematical theory of optimal processes*. New York: Interscience Publishers; **1962**.
- [21] Hendrikx JPM, Meijlink TJJ, Kriens RFC. Application of optimal control theory to inverse simulation of car handling. *Veh Syst Dyn*. **1996**;26(6):449–461.
- [22] Da Lio M. Analisi della manovrabilità dei veicoli un approccio basato sul controllo ottimo. ATA Associazione Tecnica Dell'Automobile. **1997**;50:35–42.
- [23] Bertolazzi E, Biral F, Da Lio M. Symbolic-numeric efficient solution of optimal control problems for multibody systems. *J Comput Appl Math*. **2006**;185(2):404–421.
- [24] Bobbo S, Cossalter V, Massaro M, et al. Application of the optimal maneuver method for enhancing racing motorcycle performance. *SAE Int J Passeng Cars Mech Syst*. **2009**;1(1):1311–1318.
- [25] Tavernini D, Massaro M, Velenis E, et al. Minimum time cornering: the effect of road surface and car transmission layout. *Veh Syst Dyn*. **2013**;51(10):1533–1547.
- [26] Tavernini D, Velenis E, Lot R, et al. The optimality of the handbrake cornering technique. *J Dyn Syst Meas Control*. **2014**;136(4):041019.
- [27] Biral F, Zendri F, Bertolazzi E, et al. A web based ‘virtual racing car championship’ to teach vehicle dynamics and multidisciplinary design. In: ASME, editor. *Proceedings of 2011 ASME International Mechanical Engineering Congress and Exposition*; 2011 Nov 11–17. Denver: ASME; **2011**. p. 391–401.
- [28] Lot R, Biral F. A curvilinear abscissa approach for the lap time optimization of racing vehicles. *IFAC Proc Vol*. **2014**;47(3):7559–7565. 19th IFAC World Congress.
- [29] Dal Bianco N, Lot R, Gadola M. Minimum time optimal control simulation of a GP2 race car. *Proc Inst Mech Eng D J Automob Eng*. **Forthcoming**. doi:[10.1177/0954407017728158](https://doi.org/10.1177/0954407017728158)
- [30] Casanova D, Sharp RS, Symonds P. Minimum time manoeuvring: the significance of yaw inertia. *Veh Syst Dyn*. **2000**;34(2):77–115.
- [31] Kirches C, Sager S, Bock HG, et al. Time-optimal control of automobile test drives with gear shifts. *Optim Control Appl Methods*. **2010**;31(2):137–153.
- [32] Diehl M, Leineweber DB, Schäfer A. *MUSCOD-II users’ manual*. Heidelberg: Universität Heidelberg; **2001**. (iwr-preprint 2001-25 ed.).

- [33] Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Program.* **2006**;106(1):25–57.
- [34] Berntorp K, Olofsson B, Lundahl K, et al. Models and methodology for optimal trajectory generation in safety-critical road-vehicle manoeuvres. *Veh Syst Dyn.* **2014**;52(10):1304–1332.
- [35] Perantoni G, Limebeer DJN. Optimal control for a formula one car with variable parameters. *Veh Syst Dyn.* **2014**;52(5):653–678.
- [36] Falugi P, Kerrigan E, van Wyk E. Imperial college london optimal control software user guide (ICLOCS). London: Imperial College London; **2010 May**.
- [37] 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 Trans Math Softw (TOMS).* **2014**;41(1):1.
- [38] Limebeer DJN, Perantoni G, Rao AV. Optimal control of formula one car energy recovery systems. *Int J Control.* **2014**;87(10):2065–2080.
- [39] Masouleh MI, Limebeer DJN. Optimizing the aero-suspension interactions in a formula one car. *IEEE Trans Control Syst Technol.* **2016 May**;24(3):912–927.
- [40] Sharp RS. A method for predicting minimum-time capability of a motorcycle on a racing circuit. *J Dyn Syst Meas Control.* **2014**;136(4):041007.
- [41] Gerdts M. A moving horizon technique for the simulation of automobile test-drives. *ZAMM – J Appl Math Mech.* **2003**;83(3):147–162.
- [42] Gerdts M, Karrenberg S, Müller-Beßler B, et al. Generating locally optimal trajectories for an automatically driven car. *Optim Eng.* **2008**;10(4):439–463.
- [43] Mavrouidakis B. About the simulations of formula 1 racing cars. Schriften aus dem Institut für Technische und Numerische Mechanik der Universität Stuttgart; Shaker; **2011**. German.
- [44] Kelly DP, Sharp RS. Time-optimal control of the race car: influence of a thermodynamic tyre model. *Veh Syst Dyn.* **2012**;50(4):641–662.
- [45] Timings JP, Cole DJ. Minimum maneuver time calculation using convex optimization. *J Dyn Syst Meas Control.* **2013 Mar**;135(3):031015-1–031015-9.
- [46] Timings J, Cole D. Robust lap-time simulation. *Proc Inst Mech Eng D J Automob Eng.* **2014**;228(10):1200–1216.
- [47] Butz T, Von Stryk O. Optimal control based modeling of vehicle driver properties. *SAE Technical Papers*; **2005**.
- [48] Braghin F, Cheli F, Melzi S, et al. Race driver model. *Comput Struct.* **2008 Jul**;86(13–14):1503–1516.
- [49] Fischer R, Butz T, Ehmann M, et al. A driver model for virtual control system development. *ATZautotechnology.* **2011**;11(6):22–25.
- [50] Lot R, Massaro M, Sartori R. Advanced motorcycle virtual rider. *Veh Syst Dyn.* **2008**;46(S1):215–224.
- [51] Saccon A, Hauser J, Beghi A. A virtual rider for motorcycles: maneuver regulation of a multi-body vehicle model. *IEEE Trans Control Syst Technol.* **2013**;21(2):332–346.
- [52] Cole DJ, Pick AJ, Odhams AMC. Predictive and linear quadratic methods for potential application to modelling driver steering control. *Veh Syst Dyn.* **2006**;44(3):259–284.
- [53] Keen SD, Cole DJ. Application of time-variant predictive control to modelling driver steering skill. *Veh Syst Dyn.* **2011**;49(4):527–559.
- [54] Massaro M. A nonlinear virtual rider for motorcycles. *Veh Syst Dyn.* **2011**;49(9):1477–1496.
- [55] Massaro M, Lot R. A virtual rider for two-wheeled vehicles. *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*; 2010 Dec 15–17; Atlanta, GA, USA; **2010**.
- [56] Maniowski M. Optimization of driver and chassis of fwd racing car for faster cornering. The dynamics of vehicles on roads and tracks: Proceedings of the 24th Symposium of the International Association for Vehicle System Dynamics (IAVSD 2015); 2015 Aug 17–21; Graz, Austria: CRC Press; **2016**. p. 77–86.
- [57] Maniowski M. Optimisation of driver actions in RWD race car including tyre thermodynamics. *Veh Syst Dyn.* **2016**;54(4):526–544.

- [58] Cossalter V, Lot R, Tavernini D. Optimization of the centre of mass position of a racing motorcycle in dry and wet track by means of the 'optimal maneuver method'. Proceedings of the 2013 IEEE International Conference on Mechatronics (ICM); 2013 Feb 27–Mar 1; Vicenza, Italy: IEEE; **2013**. p. 412–417.
- [59] Limebeer D. Optimising the aero-suspension interactions in a formula one car. *IEEE Trans Control Syst Technol.* **2014**;24(3):912–927.
- [60] Amadio P, Romanazzi G. Algorithm 859: BABDCR: a Fortran 90 package for the solution of bordered ABD linear systems. *ACM Trans Math Softw.* **2006 Dec**;32(4):597–608.
- [61] Bryson AE. Dynamic optimization. Vol. 1. Menlo Park (CA): Prentice Hall; **1999**.
- [62] Bertolazzi E, Biral F, Da Lio M. Real-time motion planning for multibody systems: real life application examples. *Multibody Syst Dyn.* **2007**;17(2–3):119–139.
- [63] Weinstein MJ, Rao AV. Algorithm 984: adigator, a toolbox for the algorithmic differentiation of mathematical functions in matlab using source transformation via operator overloading. *ACM Trans Math Softw.* **2017 Aug**;44(2):21:1–21:25.
- [64] Cossalter V, Peretto M, Bobbo S. Investigation of the influences of tyre–road friction and engine power on motorcycle racing performance by means of the optimal manoeuvre method. *Proc Inst Mech Eng D J Automob Eng.* **2010**;224(4):503–519.
- [65] De Groote F, Kinney AL, Rao AV, et al. Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem. *Ann Biomed Eng.* **2016**;44(10):2922–2936.
- [66] Tanelli M, Corno M, Saveresi S. Modelling, simulation and control of two-wheeled vehicles. Chichester: John Wiley & Sons; **2014**.
- [67] Abramowitz M, Stegun IA. Handbook of mathematical functions: with formulas, graphs, and mathematical tables. Vol. 55. Washington, DC: Courier Corporation; **1964**.
- [68] Duff IS. MA57 – a code for the solution of sparse symmetric definite and indefinite systems. *ACM Trans Math Softw.* **2004 Jun**;30(2):118–144.
- [69] Cossalter V, Da Lio M, Lot R, et al. A general method for the evaluation of vehicle manoeuvrability with special emphasis on motorcycles. *Veh Syst Dyn.* **1999**;31(2):113–135.
- [70] Limebeer DJ, Massaro M. Dynamics and optimal control of road vehicles. New York: Oxford University Press; **2018**.
- [71] Limebeer DJN, Perantoni G. Optimal control of a formula one car on a three-dimensional track – part 2: optimal control. *J Dyn Syst Meas Control.* **2015**;137(5):051019.
- [72] Gillespie TD. Fundamentals of vehicle dynamics. Warrendale, PA: SAE International; **1992**.
- [73] Pacejka H. Tire and vehicle dynamics. Oxford: Elsevier; **2005**.
- [74] Guiggiani M. The science of vehicle dynamics: handling, braking, and ride of road and race cars. Warrendale, PA: Springer Science & Business Media; **2014**.
- [75] Milliken WF, Milliken DL. Race car vehicle dynamics. Vol. 400. Warrendale: Society of Automotive Engineers; **1995**.
- [76] Jazar RN. Vehicle dynamics: theory and application. New York: Springer; **2017**.
- [77] Tremlett AJ, Limebeer DJN. Optimal tyre usage for a formula one car. *Veh Syst Dyn.* **2016**;54(10):1448–1473.
- [78] Tremlett AJ, Massaro M, Purdy DJ, et al. Optimal control of motorsport differentials. *Veh Syst Dyn.* **2015**;53(12):1772–1794.