To appear in *Vehicle System Dynamics*
Vol. 00, No. 00, Month 20XX, 1–41

Minimum-lap-time optimization and simulation

M. Massaro[a*] and D.J.N. Limebeer[b]

[a]*Department of Industrial Engineering, University of Padova,*
Via Venezia 1, 35131 Padova, Italy.
[b]*Department of Electrical and Electronic Engineering Science,*
University of Johannesburg, Johannesburg, South Africa.

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

The paper begins with a survey of advances in state-of-the-art minimum-time simulation for road vehicles. The techniques covered include both quasi-steady-state and transient vehicle models, which are combined with trajectories that are either pre-assigned, or free to be optimized. The fundamentals of nonlinear optimal control are summarized. These fundamentals are the basis of most of the vehicular optimal control methodologies and solution procedures reported in the literature. The key features of three-dimensional road modelling, vehicle positioning, and vehicle modelling are also summarized with a focus on recent developments. Both cars and motorcycles are considered.

Keywords: lap-time simulation; optimization; road modelling; quasi-steady-state; optimal control; nonlinear programming; g-g map; free-trajectory; Formula One; MotoGP; NASCAR.

1. Introduction

A classic metric used to assess the performance of road vehicles is the time taken to travel a prescribed distance. The quest to improve vehicular performance has promoted the development of numerical methods that estimate these times. The aim of this paper is to provide an overview of the predominant approaches reported in the literature, and to discuss some recent advances.

Minimum-time simulation is a classical optimal control problem (OCP)[1]. It is thus no surprise that OCP methods are an important component of minimum-manoeuvre-time and minimum-lap-time simulations (MLTS) [1]. In the generic OCP framework the driver is the 'optimal controller' who takes charge of the application of drive power, braking and steering. Subject to the physical limits associated with the road boundaries and the vehicle's performance, the control algorithm, 'the optimal driver', seeks to minimise some performance criterion such as the lap time. In the most general formulation, the control objective is 'drive the vehicle so as to complete a prescribed number of laps in minimum time'. In order to make the solution to these problems practically feasible using the computational facilities currently available, a number of simplifications have been proposed. These simplifications might include such things as prescribing a priori

*Corresponding author. Email: matteo.massaro@unipd.it

[1] In order to avoid awkward usage, such as OCPs, we will always use acronyms in their singular form.

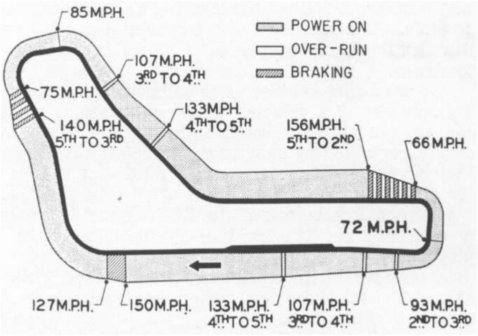


Figure 1. Driving the Monza circuit in Italy, from [2].

the vehicle's trajectory, using quasi-steady-state (QSS) models (as opposed to dynamic models), and using simplified tyre and aerodynamics models.

In order to limit the scope of this review we will not cover techniques that are based predominantly on feedback control. These methods use a feedback loop to minimise an error relative to a prescribed optimal trajectory (in this context 'trajectory' refers to the target controls and the consequent state response). We will also limit our discussion of 'prescribed-trajectory' approaches, as these are now well understood.

We begin in Section 2 with an historical overview of the main developments in MLTS with summary assessments of the (published) state-of-the-art. After briefly reviewing the fundamentals of the OCP in Section 3, the most popular solution techniques are discussed in Section 4. We will also look at some of the practicalities such as scaling, the computation of gradients, and problem regularization. Section 5 is devoted to three-dimensional road modelling. The identification of the road-related parameters—the road reconstruction problem—is covered in Section 6. Vehicle positioning kinematics is covered in Section 7. Section 8 is devoted to vehicle models. In Section 9 we summarize the formulation of MLTS problems, with an example application for both a car and a motorcycle given in Section 10.

2. Historical overview

2.1. *The early days*

Early attempts to numerically predict the lap-time performance of road vehicles date back to the late 1950s. The complete driving profiles of Formula One cars were routinely computed before races, including the speed on different sections of the track, braking distances and gear-shifting points. An example application of this type is discussed for the Monza circuit (Italy) in [2]; see Figure 1.

In 1971 a computer analysis of the Watkins Glen GP circuit (USA) was carried out with the aim of computing the speed and acceleration profiles of racing cars [3]. The vehicle was treated as a point mass, with prescribed acceleration limits, driven along an empirically determined 'racing groove'. This is an example of a QSS method featuring the early use of 'g-g' diagrams, which began to appear in published forms in the 1970s [4, 5]. These diagrams show the accelerations achievable in a horizontal plane (the 'flat road' assumption), which is a useful way of characterizing the vehicle's performance envelope. The limit accelerations, especially in the longitudinal direction, are functions of speed. For that reason, speed-dependent families of g-g diagrams are of obvious utility—these assemblies of g-g diagrams are called 'g-g-v' diagram; see for example Figure 2. The

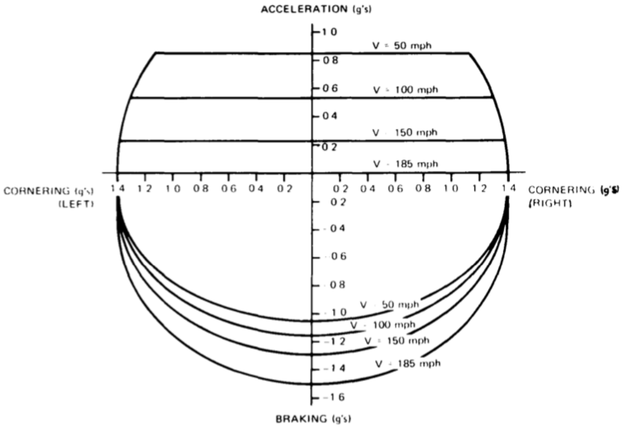


Figure 2. The g-g-v diagram of a racing car, from [5].

lateral accelerations are less sensitive to speed, unless there are significant aerodynamic downforces, which may be the case.

A MLTS can be computed using g-g diagrams once the trajectory is given (either the racing line, or a sequence of straights and constant-radius turns). The method begins from the corner apices (or 'critical points', see Figure 3), where the vehicle is assumed to attain the maximum lateral acceleration. The corresponding speed, sometimes called 'the critical speed', is easily obtained since the curvature at the apex is also assumed known. Before the apex, the vehicle is braking along the g-g envelope (along the predefined trajectory), while after the apex it is accelerating (again along the predefined trajectory). The speed profiles between two apexes (the acceleration out of turn k and the deceleration into turn $k+1$) are then connected at their intersection point.

The minimum transit time for a predefined trajectory is considered in [6] using a dynamic planar car model that includes longitudinal, lateral and yaw degrees-of-freedom (DOF). The calculation procedure begins by dividing a track section into a series curves. After identifying the critical points, and the corresponding critical speeds, the equations are integrated assuming maximum acceleration after the critical point and maximum deceleration before the critical points to find the velocity profile. A section of the Paul Ricard circuit (in France) is simulated using a dataset resembling that of a Formula One car. The results, namely the g-g diagrams, are compared with the performance measured for a Formula One car driven by three racing drivers (Andretti, Peterson and Stewart).

MLTS that are based on predefined trajectories and QSS models are considered in [7]. While MLTS based on g-g diagrams is now standard, a new departure is the use of a predefined trajectory obtained by an optimization procedure. Minimizing the distance travelled through the corner, and maximising the cornering speed, both appear to be sensible objectives. To see how these objectives are related, we observe that by balancing lateral forces (neglecting aerodynamic forces)

$$m\frac{v_{\max}^2}{\rho} = \mu m g \qquad \Rightarrow \qquad v_{\max} = \sqrt{\mu \rho g}, \tag{1}$$

where $v_{\max}$ is the vehicle's maximum cornering speed, ρ is the trajectory radius of curvature, and μ is the tyre-road coefficient of friction. This relationship shows that minimising the distance travelled, and maximising the cornering speed are conflicting objectives—the shortest path leads to a low radius of curvature trajectory, while a high radius of curva-

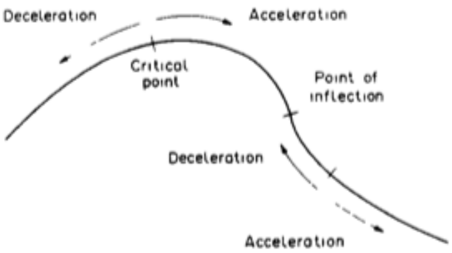


Figure 3. Procedure for QSS optimization, from [6].

ture is required for higher speeds. The method proposed in [7] assumes that the optimal trajectory through each corner equates to maximizing the minimum cornering radius. The trajectory through each turn is parametrized in terms of four points: the turn-in point (assumed to be on the outer periphery of the track), the brake release point, the throttle re-application point (mid corner and on the inside track boundary), and the corner exit (on the outer periphery of the track). Once these points have been selected, they are interpolated by a cubic spline. Simulations are shown for a front-wheel-drive D2 Super Touring saloon car on the Hockenheimring (Germany).

An alternative, but related approach to computing the optimal trajectory is discussed in [8]. This paper considers a constrained optimisation problem that takes into account both the geometry of the circuit and the dynamics of the vehicle. The important point is that the optimal trajectory through the corner is strongly influenced by the dynamics of the vehicle, and is thus a combination of the shortest path and the minimum curvature trajectory. In [9] the minimum curvature path is generated by solving a quadratic programming (QP) problem. The key contribution of this paper is the extension of the QP formulation used for determining accurate curvature approximations and improved lap-time estimates. Improvements in relation to the results in [8] are also presented.

An early example of optimising a vehicle's performance by solving a (nonlinear) OCP is presented in [10]. The idea is to minimize the manoeuvre time (cost function), while satisfying the dynamic equations (constraints) of the vehicle, as well as other constraints such as the road boundaries. Strictly speaking, [10] does not consider a pure minimum-time problem, as the cost function also contains a 'deviation from a reference path' term. A number of approximation are introduced which makes finding a solution less burdensome; only lane-change manoeuvres are considered. This work also introduces the elapsed distance (rather than elapsed time) as the independent variable—this is a feature of almost all subsequent MLTS work. This change of variable transforms the free-final-time OCP into a fixed-terminal-distance OCP (with the problem state dimension reduced by one).

2.2. *The 1990s*

Some early examples of MLTS solved as minimum-time OCP appeared in the 1990s. A hairpin curve at the Fuji Speedway (Japan) is considered in [11]; the cost function is the manoeuvre time. Unlike earlier studies, no pre-assigned reference trajectory is provided—the trajectory is determined by the optimizer. A single-track three DOF car model is employed (the well-known 'bicycle model'); the longitudinal and lateral tyre forces are constrained by friction circles. The lateral forces are modelled as linearly-saturating: they grow linearly with slip (at fixed cornering stiffness) until saturation is reached. The side-forces remain constant thereafter; load transfer effects are not con-

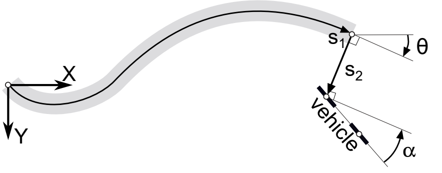


Figure 4. Vehicle positioning through curvilinear coordinates, adapted from [16].

sidered. Both front-wheel-drive (FWD) and rear-wheel-drive (RWD) configurations are considered, together with front- and rear-wheel steering variants. The solution of the OCP is obtained using a sequential conjugate-gradient-restoration algorithm [12, 13].

A more general vehicle model is employed in [14], which is again represented by a three DOF planar model, but this time with four-wheels (two tracks) and steady-state (longitudinal and lateral) load transfer. Both drag and downforces are considered. The inputs are the steering rate and the total drive/braking longitudinal force. The drive force is split equally between the two propulsive wheels (*i.e.* an open differential), while braking influences are split between the front and rear axles according to a prescribed brake bias, and applied symmetrically. The tyre lateral forces depend on the side slip and normal load through a Magic Formula. The coupling between longitudinal and lateral forces recognises a friction ellipse. The maximum engine torque vs. engine rpm curve is converted into a relationship between the tyre longitudinal force and the vehicle speed. It is thus assumed that the driver engages the gear ratio that provides the maximum torque at the current speed—this is an approach that is employed in almost all of the subsequent work on MLTS. Gear shifts are assumed instantaneous. The cost function of the OCP comprises the manoeuvre time and a barrier term relating to the mass centre of the car violating the road boundaries. The driver's physical limitations are accounted for by limiting the maximum steering rate. The OCP is solved by an indirect gradient projection method [15]. Only lane-change manoeuvres, using both a RWD Formula One car, and FWD passenger car, are considered.

MLTS for motorcycles are considered for the first time in [16]. The vehicle model includes a roll freedom, in addition to longitudinal, lateral and yaw DOF typical of planar car models. Additional states include the front tyre vertical load, the rear tyre vertical load, and the rear tyre lateral force. There are three control inputs: the lateral force on the front tyre, the longitudinal tyre forces and the brake bias that divides the longitudinal force between the front and rear tyres under braking. The physical controls are integrals of the OCP inputs, which are constrained in order to avoid abrupt bang-bang physical inputs. The equations of motion are supplied as a set of implicit differential equations (IDE), which makes the application 'nonstandard', since most OCP solvers only deal with ordinary differential equations (ODE). Referring to Figure 4, curvilinear coordinates are employed to track the vehicle's position using the travelled distance s_1 along the centreline, the lateral position s_2 of the vehicle with respect to the centreline, and the relative orientation α of the vehicle with respect to the tangent to the centreline. In order to avoid a free-end-time formulation, the distance travelled is maximized in a prescribed time. The cost function also includes penalty-function terms that account for tyre-force saturation, the rate of change of the vehicle's physical inputs, and violation of the road boundaries. A section of the Mugello circuit (Italy) is simulated with a dataset representative of a racing motorcycle. The OCP is solved using the indirect solver SOLVDE [17].

2.3. *The 2000s*

In the early part of the new millennium a number of research outputs appeared that relate to the MLTS optimal control of Formula One cars. In [18] a seven DOF planar car model is employed, which includes the standard longitudinal, lateral and yaw motions of the car's chassis, and the spin motion of the four wheels. The tyre forces depends on slips and loads through Magic Formulas. The equations are written in the space domain. The cost of the OCP is the manoeuvre time. The track is described in terms of the curvature of the centreline and is divided into segments. The continuity of the state trajectories across the k to $k+1$ segment boundary being enforced by a constraint of the form $x_{init}(k+1) - x_{final}(k) = 0$. In this study, there are typically 3-5 control segments for each track partition. The optimal trajectory for each road section can be solved separately once the initial conditions and the controls have been selected by the optimizer. This process is repeated to convergence as is standard in multiple-shooting techniques. The sequential-quadratic-programming (SQP) algorithm implemented in the Matlab$^{\text{TM}}$ Optimisation Toolbox was used. The track segments have lengths of between 8 m (tight corners) and 120 m (straights), and 20–30 m in medium and fast bends, while the control segments have lengths of between 4 m (tight corners) and 20 m (straights). This methodology is used to study the effect of the yaw inertia on a 300 m double lane-change manoeuvre. The effect of mass on a full lap of the Montmeló (Spain) and Suzuka (Japan) circuits are reported in [19]—this time the SQP solver is SNOPT [20]; the results trend with the (then-current) experimental value (0.036-0.037 s/kg/lap). The effect of the longitudinal position of the centre of mass is analysed in [21], again for both the Montmeló (Spain) and Suzuka (Japan) circuits. In [22] automatic differentiation (AD) is used to replace finite differences, and a ten-fold decrease in computation time is demonstrated. In this study AD is employed to evaluate the first-order derivatives of the objective and constraint functions with respect to the state and control variables.

Advances in the MLTS of motorcycles are reported in [23, 24], where the motorcycle model is extended to include the steer angle (in addition to the longitudinal, lateral, yaw and roll DOF described in [16]). The rider is again assumed rigidly attached to the chassis, *i.e.* he/she only affects the vehicle's inertia. Relaxation equations are included for both the front and rear tyre forces, while tyre saturation is again modelled using friction circles. The motorcycle inputs are the front and rear wheel longitudinal forces and the steering torque (between the handlebar and the chassis). The OCP inputs are the rate of change of the motorcycle's physical inputs. The independent variable is the centreline distance travelled, while the cost is the minimum-manoeuvre time. The resulting OCP is solved as a two-points boundary-value problem (TPBVP). A symbolic procedure is used to find the adjoint equations, which are then discretized by a finite-difference scheme. The forward and backward equations constitute a large set of algebraic equations, which are solved with a modified version of the affine-invariant Newton scheme. Constraints such as the road width are included in the cost function using barrier functions [25]. A racing motorcycle is simulated on a lap of the Adria International Raceway (Italy).

In [26] the motorcycle model is extended to include a realistic engine torque map using a technique similar that described in [14]—the rider only 'sees' the engine torque envelope at the rear wheel. A full lap on the Adria circuit, and a two-dimensional representation of the Mugello (Italy) circuit, are considered with a view to finding optimal gearbox ratios for a 1000cc SuperBike class motorcycle. The problem is again solved with the indirect method described in [24].

A motorcycle model that includes the motion of the rider is discussed in [27]. Referring to Figure 5, the rider's upper body has fore-and-aft rotation γ, and lateral lean (η)

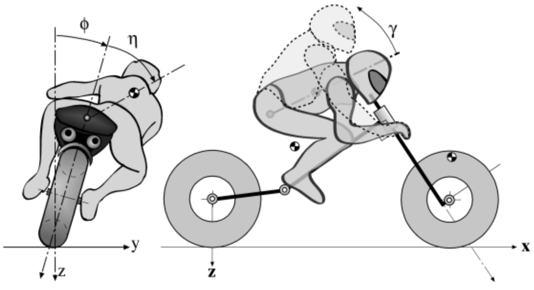


Figure 5. Rider model with lateral lean and fore-and-aft DOF, from [27].

freedoms. The for-aft rotation takes place around a lateral axis through the rider's seat. The lean roll freedom occurs around a longitudinal axis close to the ground. In combination, these freedoms are used to replicate the rider's motion. The aerodynamic drag coefficient is a function of the rider's forward lean; lower drag being associated with a prone 'tucked-in' position. The model inputs are torques that determine the lean motions of the rider, the steering torque, and the front- and rear-wheel longitudinal tyre forces. The OCP controls are filtered with second-order low-pass filters prior to injection into the vehicle model in order to recognise the limited bandwidth of human riders; this is an alternative to saturating the control rates. The minimum-time manoeuvres studied are for a straight section and a U-turn. The solution of the OCP is obtained with the indirect method described in [24]. It is demonstrated that the rider's air drag variation due to fore-aft leaning dominates the effect of the changes in the centre of mass position. The lateral lean motion has a much greater influence on the vehicle's dynamics. For example, lateral leaning reduces the need for 'counter-steering' on the entry to corners, and reduces the roll angle in the turn (as compared to the rigidly attached rider).

At this point the MLTS reported in the literature, for both cars and motorcycles, use 'flat' two-dimensional tracks, with the racing line assumed free to be determined as part of the optimization process.

An early attempt to compare the results of the newly developed dynamic methods, without predefined trajectory, and the QSS methods with predefined racing lines is reported in [28] for an open-wheel race car on a full lap. The vehicle model used is the seven DOF car described in [18, 19, 21]; the reference dynamic MLTS is the Montmeló OCP simulation reported therein. In order to obtain the corresponding QSS simulations, the g-g diagrams of the same car model are computed at different speeds, and then used for the MLTS with a second-order polynomial fit to the racing line obtained from the OCP solution. Good agreement is demonstrated between the longitudinal and lateral accelerations profiles. However, there appears to be a delay in some corner entry and exit points, which is attributed to the different braking and acceleration crossover points—the points where the accelerating and braking speeds between two apexes reach the same value. The difference in the reported lap time is 2.3 s. Variations in the position of the centre of mass of the car produces similar results with both methods. When it comes to the computation time, the difference is two order of magnitude.

2.4. *The 2010s*

An alternative approach to the MLTS problem for race cars is reported in [29]. In this study the optimiser was only given limited access to the vehicle model—the states which are common to all representative vehicle models. These include the absolute position coordinates of the car's mass centre, the longitudinal and lateral velocities of the car, the vehicles yaw rate and lateral acceleration, the longitudinal slip ratio at each wheel, as well as the tyre loads. The idea is that the method can potentially be applied to models of any complexity. In this work the manoeuvre under study is defined in terms of suitably spaced waylines, each being perpendicular to the track centreline, and with a length representing the width of the track. At each wayline, longitudinal and lateral controls are found that satisfy the track-boundary constraints. Other constraints are introduced that limit the search space, these include the slip ratio, the understeer (angle), the vehicle's slip, and tyre load maxima. An interior-point feasible sequential quadratic programming code [30], is used with forward-finite-difference gradient calculations. A receding horizon of approximately 250 m is used. An example application is given for a Formula One car on Jerez circuit (Spain).

In 2011 a survey of automotive optimal control was published [31], with the third section devoted to minimum-time manoeuvring. Some of the papers discussed above are included, while others are not.

Another example of the application of direct multiple shooting methods to MLTS is reported in [32]. A single-track car model with nonlinear tyres representative of a Porsche 911 Club Sport stock car is employed. The MLPS was conducted on the Formula One racing circuit Hockenheimring (Germany), with a lap time of 112.7s obtained using 324 shooting intervals. The works builds upon prior studies [33–36], where vehicle equations, datasets, and solution approach details are provided. The position of the vehicle is given in terms of the car's progress on, and lateral deviation from, the centreline. The independent variable is the travelled distance. A notable feature of the approach is that the gear ratios are treated as 'discrete', which leads to a mixed-integer OCP. The discrete controls are treated alongside the standard continuous controls (steering rate, braking force and throttle position). Mixed-integer programming problems are difficult to solve, but can be avoided using relaxation 'tricks'. The resulting large-scale nonlinear programming problem (NLP) is solved using the SQP solver within MUSCOD [37].

An alternative to QSS approaches and nonlinear OCP is presented in [38–40]. The central idea is to linearise the vehicle model at each time step, and then solve the resulting convex QP optimization problem. This is in contrast to the prior use of nonlinear models in combination with a nonlinear programme solver. The problem is formulated within a model predictive control (MPC) framework, in which the minimum time problem is formulated as one of maximizing the distance travelled along the track centreline. The track and vehicle trajectories are linearised by considering small displacements from a fixed reference. The method was developed initially for predicting optimal trajectories at constant speed. It was then extended to solve the minimum-lap-time problem, and to investigate 'robustness' to disturbances and driver errors. The vehicle model considered is a single-track car model extended to include the spin of the front and rear wheels *i.e.* five DOF. The tyre response is modelled with a Magic Formula. The model inputs are the steering angle and the total wheel torque. A second-order low-pass filter (cut-off frequency 3 Hz and damping ratio 0.7) is used to represent the neuromuscular system dynamics of the driver. The results are compared with those from standard nonlinear optimizations on a 90^o bend—good agreement was found. The method is then demonstrated on a short racing circuit, with road roughness included as a disturbance. A linear quadratic regulator (LQR) was used to keep the vehicle close to the nominal vehicle path and speed profile in the presence of disturbances.

Another example of an MPC-based approach to MLTS is reported in [41]. A cascade optimization structure is employed to overcome possible shortcomings associated with a finite preview (receding) horizon. The objective of the low-level MPC problems is to find control inputs that achieve a blend of minimising the manoeuvre time and maximising the exit velocity in each preview horizon. These low-level problems are solved using GPOPS-II [42]. The outer optimization problem selects the weights (manoeuvre time to exit speed) to be used on each (local) MPC horizon, in order to minimize the lap time. This global problem is solved using a genetic algorithm on supercomputing clusters. A Formula One car model is considered on (two-dimensional) Hockenheim and Nürburgring circuit models. The previews lengths for the local MPC problems are in the range 200–250 m. The elapsed distance is the independent variable, and curvilinear coordinates are used for the vehicle positioning.

The combination of a QSS paradigm and a transient vehicle model is described in [43], in which the trajectory is pre-defined. During the calculation of the optimal MLTS strategy the transient states are treated as distance-dependent parameters. The QSS outcome is used post-facto to calculate a revised dynamic response, which induces, in turn, a revised QSS solution. This iterative process is repeated until the dynamic states have settled. The technique accommodates dynamics effects such as the tyres' damping and temperature—a grip scaling factor is used that depends on the tyre surface temperature, and a cornering stiffness factor that depends on the tyre core temperature. To be clear, the method does not use pre-computed g-g diagrams, instead, the states of the QSS model are computed step-by-step when solving an NLP problem. The given application considers a 2009 Le Mans prototype on the Circuit de la Sarthe. The computing time is approximately twelve minutes on a 2.5 GHz Intel core 2 workstation.

The lap time of a race series-hybrid electric vehicle (S-HEV) is considered in [44]. The powertrain considered contains an internal combustion engine (ICE), a motor-generator system and a battery. The ICE is connected mechanically to the permanent magnet synchronous (PMS) motor generator, which converts the engine's output mechanical power into an alternating current (AC) supply. This is then rectified and connected to a direct current (DC) link. The battery acts as an energy store, or a source of additional power when required. Under driving conditions power is taken from the DC link and supplied to the PMS motor-generator that is mechanically coupled to the road wheels. Under braking the power flow is reversed, and the vehicle's kinetic energy is converted into electrical power, which is used to recharge the battery. One additional state, the battery's state of charge, augments a standard three DOF single-track vehicle model with saturated tyres. A sport car is simulated on the Mugello circuit using an indirect solver [24]. It was found that the battery is used primarily to boost the initial phases of acceleration—it is then recharged under braking. Since the optimal control problem being considered is of minimum-time type, the power flow to and from the battery shows bang-bang type characteristics [1]. Under some circumstances the ICE is used for battery recharging purposes.

Minimum-lap-time optimal control combined with the optimization of vehicle setup parameters is considered in [45]. The vehicle is represented as a standard double-track configuration with longitudinal, lateral and yaw freedoms. The model assumes instantaneous longitudinal and lateral load transfers, combined-slip nonlinear tyres, an adaptive braking ratio (*i.e.* the brake bias is not fixed), a limited-slip (speed sensing) differential, and simple aerodynamic force representations. The control inputs are the rate of change of steering angle, the rates of change of the longitudinal tyre slips, and the normal loads on the four tyres. A direct collocation method is employed to transcribe the nonlinear OCP as a NLP. The transcription is carried out using ICLOCS [46] with a trapezoidal integration scheme, while the NLP solver is IPOPT [47]. The derivatives required include first- and second-order partial derivatives of the cost function, the path constraints, the system dynamics, and the boundary constraints, all with respect to the decision variables including the inputs, the states and the parameters. Analytical derivatives were

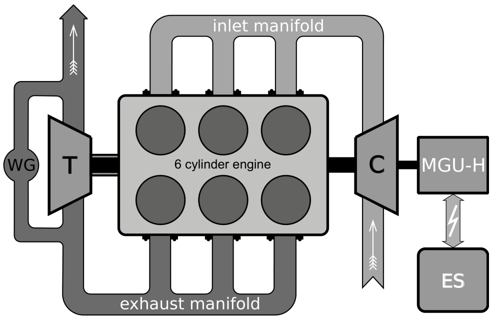


Figure 6. Thermal energy recovery system in [51]; the MGU-K is not shown.

provided. The solution is shown to converge on the Barcelona-Catalunya circuit with mesh size >0.5 m; the computing time is of the order one hour, and approximately 15 min with a 2 m mesh. The problem of finding the centreline that best fits experimental GPS data is solved separately as a another OCP, where the cost function is the error between the reconstructed centreline and measured GPS survey data. The problem input is the rate-of-change of track curvature (as a function of elapsed distance).

Three-dimensional road models are introduced in the context of minimum-lap-time OCP for cars in [48, 49] and in [50]. In the former study a double-track car model, with quasi-static load transfer, is employed with the OCP solved for Barcelona-Catalunya circuit. A direct transcription process is used to transform the OCP into a NLP using GPOPS-II [42] in combination with the solver IPOPT [47]. In the latter study a single-track car model, with a delayed load transfer, is solved on the Mugello circuit—the OCP is solved using an indirect approach [24]. The study in [50] assumes that the road inclination and camber angles are 'small'. The three-dimensionality of the circuit introduces a number of heretofore unfamiliar dynamical phenomena. These include inclination-dependent top speeds, camber-angle-dependent lateral accelerations in corners, and normal tyre loads that responds to elevation rates-of-change (suspension squatting in 'dips' and suspension load reductions on the brows of hills). In three-dimensional road studies the centreline is modelled using three curvatures (the geodesic and normal curvatures, and the relative torsion), which are continuous functions of the travelled distance—in 'flat road' studies the track is characterized by the geodesic curvature alone. For computational convenience, the road curvatures are parametrized in terms of Euler angles in the attitude-slope-banking convention. The orientation and lateral position of the vehicle are computed relative to the centreline and its tangent; this is a routine extension to the two-dimensional case. The problem of finding curvatures that best fit experimental survey data, related to the left and right boundaries of the track, is solved as another OCP [48]. The cost function is the error between the reconstructed and measured boundaries. The control inputs are the rates of change of the track-related Euler angles, which are elapsed-distance dependent.

Energy recovery systems that include both conventional kinetic energy recovery system (KERS) and exhaust gas heat scavenging were introduced into Formula One in 2014. In combination, these are usually referred to as energy recovery systems (ERS). A schematic of the thermal part of the ERS system is shown in Figure 6. The main power source is a conventional 1.6 l turbo-boosted ICE. Inlet pressure boosting is achieved using a single-stage compressor that is linked to a motor-generator unit heat (MGU-H) and an exhaust turbine—the connection shaft is not the ICE crank shaft. When the waste gate is open, the compressor is powered by the MGU-H operating as a motor that is supplied by the energy store (ES). This is an energy inefficient way of operating the power train and is only used in qualifying—opening the waste gate reduces the exhaust manifold pressure thereby increasing the ICE's output power. When the waste gate is closed, the single-stage exhaust turbine drives both the compressor and the MGU-H, which operates as a generator—this is the normal mode of operation under racing conditions. The power train includes a second motor-generator unit that is not shown in the figure. This is the motor-generator unit kinetic (MGU-K), which converts the vehicle's kinetic energy into stored electrical energy under braking, or as a power-boosting motor on corner exit, or when overtaking. The optimal usage of these ERS is considered in [51], where the 2014 FIA technical regulations [52] are fully explained. A number of constraints on the allowable energy flows and the per-lap stored/expended energy need be enforced. The vehicle model includes four additional states that monitor energy- and power-flow-related constraints. These include total fuel usage that is limited to 100 kg per race—the regulations also impose a fuel mass-flow-rate constraint of 100 kg/h. The stored energy is limited to 4 MJ, the energy flow from the ES to the MGU-K is limited to 4 MJ per lap, while the energy flow from the MGU-K to the ES is limited to 2 MJ per lap. If the car is to be competitive, this asymmetry mandates the use of thermal energy recovery, which was a relatively new technology at the time. The power flow to/from the MGU-K is limited to ±120 kW, while the power to/from the MGU-H is unrestricted. In the OCP setup one must include power balance constraints on the ICE-MGU-K and compressor-turbine-MUG-H shafts. A Formula One car is simulated on the Barcelona-Catalunya circuit (this time in its 'flat road' form). It is shown that the optimal operation of the ERS is very different during racing and qualifying; in qualifying the profligate short-term use of fuel and stored electrical energy is permitted by the technical regulations.

In most forms of motor racing optimising the vehicle's aerodynamic performance is of paramount importance. With this motivation in mind, the aero-suspension interaction of an open-wheeled race car was considered in [53]. In this study the drag and lift coefficients were functions of the car's roll angle, steer angle, slip angle and the front- and rear-axle ride heights; the roll angle and ride heights depended, in turn, on the suspension travel which was computed with a QSS meta-model. Since Formula One cars have 'hard' suspensions, the suspension travel is limited (of the order of the tyre squash) and 'almost' linear. At each integration time step the tyre loads are computed and fed into the suspension meta-model in order to find the vehicle trim. These trim conditions were then used to update the aerodynamic coefficients. For the avoidance of confusion, the suspension meta-model does not comprise the suspension kinematic equations. In the case of the cross-coupled multi-link suspension used in this study, there are 26 equations associated with each axle. The orientations and locations of the wheel carriers are described by 12 equations, there are another 12 equations associated with the forces on the suspension arms, and 2 equations that determine rocker angles and cross-coupled heave spring displacements. These equations are a function of the force and moment on the tyre contact patch. Each axle roll angle is computed as the wheel height difference divided by the axle track, while the axle ride heights are computed as the average wheel height. In the OCP solver a constraint is included that forces the front- and rear-axle roll angle to be equal; this replaces the conventional rigid-chassis roll balance equation. In the example reported, the suspension travel was found to be a function of the normal load only (the other force components and moments had a negligible effect). It is shown that parameter optimization problems involving the car's suspension and aerodynamics are readily soluble. The results demonstrate that the track-specific optimization of these systems can substantially reduce the lap time. The solution technique used is described in [49].

The effects of aerodynamic maps on the MLTS is also considered in [54]. In this case

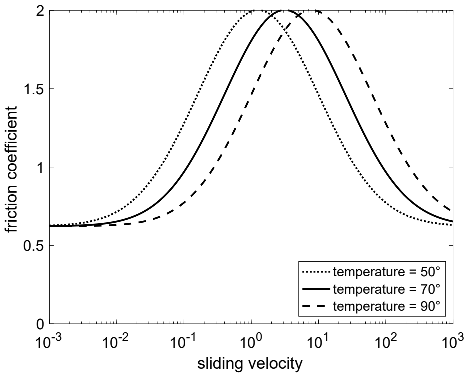


Figure 7. Friction coefficient with the reference dataset in [61].

the drag and downforce coefficients depend on the front- and rear-ride heights through a third order polynomial. The classical QSS assumption on load transfers is lifted. The car model has 14 DOF, which include suspension travel and the wheel spin, together with the six DOF of the chassis. A 2012 GP2 car is simulated on the Barcelona circuit with the indirect collocation method described in [24], with a focus on suspension dynamics.

The MLTS for a go-kart is discussed in [55]. A seven DOF model is employed, with six DOF associated with the motion of the chassis, and one with the spin velocity of the rear axle. The front axle spin dynamics are neglected, because the go-kart considered only has braking and propulsion influences on the rear axle. A race go-kart is simulated and compared with experiments conducted on the 'Pista Azzurra' near Jesolo (Italy). After some parameter tuning the simulated and experimental lap times were roughly equal at 53.6 s, with the simulation time approximately 230 s on a laptop equipped with an Intel Core-i7 640M processor. The OCP is solved with the indirect method described in [24]. The lack of a differential meant that high lateral load transfers were required to minimize the lap times. This produced some inside-wheel 'lifting' under turning in order to maintain peak loading on the outside wheels. Limit-handling is also considered in [56, 57], where it is shown that in some scenarios 'drifting' and 'handbrake turning' techniques arise as solutions to the minimum-time OCP. A long-distance track is considered in [58], where the 60 km Snaefell Mountain Course is simulated for the design of an electric motorcycle for the annual Isle of Man TT Zero Challenge. The motorcycle is represented as a mono-wheel with the track elevation considered (without cambering). The electric power train consists of a lithium battery pack, a brushed DC drive motor, and a motor controller. Ackermann effects on MLTS are studied in [59]. State-of-the-art engine models are employed to investigate the minimization of emissions production in [60].

A simple thermodynamic tyre model is introduced in the MLTS in [61], where the solution method remains the direct approach described in [29]. The tyre is divided conceptually into carcass and tread: the carcass is considered rigid with the tyre's compliance coming from the elastic bristles of the tread. The contact patch is divided into sliding and non-sliding regions as taught in the standard brush model [62]. The patch is assumed rectangular, with a fixed width, and with the length determined by the inflation pressure. The normal load distribution is assumed symmetric and approximately parabolic. The friction coefficient is assumed to depend on the sliding velocity and tread temperature in the manner of the well-known 'time-temperature' equivalence [63]. The master curve of the tread compound is assumed to be Gaussian as shown in Figure 7. The sliding velocities at which maximum friction is attained increases with the temperature. The road-tyre friction coefficient is assumed to reduce linearly with the contact pressure. A

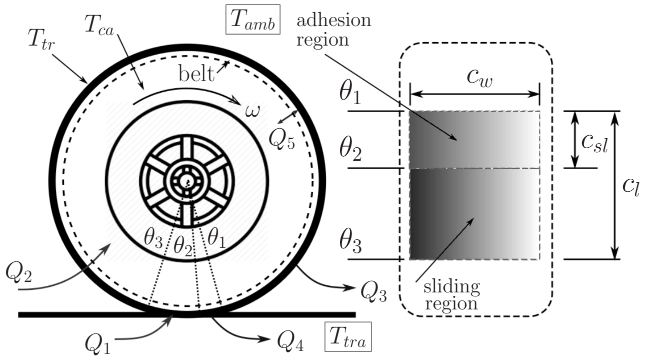


Figure 8. Tyre temperatures and heat flows [64].

lumped-parameter thermodynamic model is considered, with three states representing the temperature of the tread, the gas and the bulk material. Track temperature and ambient air temperature are assumed constant, with the heat flows governed by Newton's law of cooling. The model assumes convective heat flows between the tread and the surrounding air; the heat transfer coefficient increases linearly with the vehicle's speed. Conductive heat transfers occur between the tread and the road in the adhesion regions (the front part of the contact patch). Radiation effects are neglected. The heat that warms the tyres comes from carcass deflection (whose heating power is assumed to be a linear function of the tyre forces), and from the tread sliding on the road (the heating power is a function of the sliding friction coefficient, the normal load, and the sliding velocity). A diagram representative of this configuration is shown in Figure 8, where Q_1 is the heat flow generated by tyre slip, Q_2 is generated by flexing of the tyres' structure and carcass, Q_3 is heat lost by convective cooling, Q_4 represents heat lost by conductive cooling in the adhesion region of the contact patch, and Q_5 represents the heat transfer between the tread and the carcass; the heat flows from the carcass to the gas is neglected. The Silverstone national circuit (UK) is used in the simulation work. It is shown that the carcass and inflation gas temperatures vary slowly, while the tread temperature variations are larger and faster. In short manoeuvres the carcass and gas temperatures are essentially constant.

The purely thermodynamic tyre work considered in [61] is upgraded to include wear in [65] and [64]. Following the results in [61], the gas and tyre carcass temperatures are neglected in [65], while an estimate of tyre wear is introduced, using an additional state. The sensitivity of tyre friction coefficient to tyre temperature variations is based on experiment. The length of the sliding region is modelled using an empirical formula—30% of the contact patch is assumed to be 'sliding' at 0 deg of slip angle, which increases to 80% when the slip angle increases to 8 deg; linear interpolation on the slip angle is employed, with the influence of longitudinal slip neglected. The tyre wear rate is computed using a combination of tyre frictional power, 'graining' and 'blistering'. Graining is used to describe particles of rubber breaking away from the tread surface as a consequence overworking, while the tyre is below its optimal operating temperature range. Blistering is used to describe accelerated wear resulting from localised hot spots. The results suggest that earlier, smoother braking into corners, and reduced throttle usage in high-speed corners optimises tyre wear. A trade-off between lap times and tyre wear is demonstrated using the Barcelona circuit and the direct method described in [49]. This study also addresses the possibility of tailoring the cars differential setup to minimise tyre wear. In particular, the value of opening the differential on corner entry and then locking it on corner exit is illustrated. This strategy maximises traction and longitudinal acceleration—similar results are reported in [66].

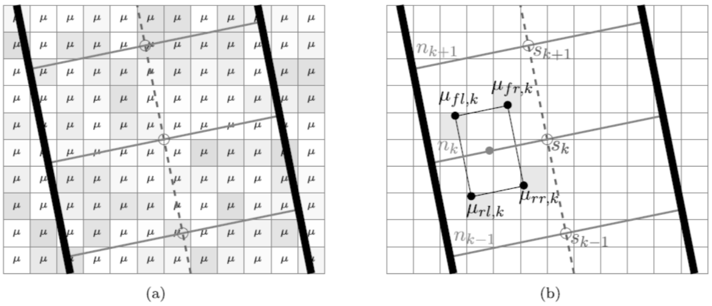
These tyre management studies are extended in [64]; see Figure 8. The complexity of the tyre model is in-between that used in [61] and that employed in [65]. In this new study the thermodynamic states are the tyre tread and the tyres bulk material temperatures; in particular, the gas temperature state used in [61] was removed. This is because the dominant heat flows are from the carcass to the tread, and from the tread to the road—smaller heat flows are associated with the tread to ambient, and the carcass to ambient, with the smallest (and negligible) heat flow from the carcass to the gas. On a racing lap the tread will typically undergo rapid temperature variations. On the other hand the carcass temperature reacts more slowly making it more difficult to manage—this is particularly true in cases where carcass-temperature reductions are required. In order to recognise the long-run impact of tyre degradation, the grip is modelled as a function of both tyre wear [67] and the tread temperature [68]. Simulations on the Circuit de Catalunya show that with new tyres the first lap tends to be slow reflecting the combined influences of a low operating temperature and tyres that have not yet been 'rubbed in'. The tyres then operate optimally for a number of laps with the lap times gradually increasing due to wear. After that tyre wear begins to affect the performance more noticeably, and the lap times begin to degrade. At some point the drop-off in performance increases to the extent that the car is no longer competitive, and a tyre change is required. The numerics associated with these thermal- and wear-related phenomena are clearly car, track and tyre dependent.

The MLTS of electric vehicles (EV) equipped with separate motors on each wheel (wheel-independent motors (WIM)) is considered in [69]. A standard two-track planar vehicle model is employed. The solution is obtained by transcribing the OCP into a NLP. A forward-Newton discretisation is employed, and the resulting NLP is solved with IPOPT [47]. Simulations on a hairpin suggest that during combined braking and steering, the optimal torque distribution generates an 'understeering yaw-moment', while an 'oversteering yaw-moment' is produced during combined acceleration and steering. Larger torques are applied to the wheels with larger normal loads. The results of the OCP are then used to design a causal torque allocation strategy, which assumes knowledge of the drivers inputs, the vehicle speed and accelerations, and produces torque references for the four WIM.

A similar study is presented in [70], where the effect of altering the passive handling on the minimum time performance of the EV with 4 WIM is studied. The solution is obtained using GPOPS-II [42], with the derivatives computed algebraically using the automatic differentiation tool ADIGATOR [71], while IPOPT [47] is the NLP solver. It is found that the manoeuvre time for the reference-following torque-vectoring (TV) vehicle is largely insensitive to target understeer gradients.

Recent work reported in [72] extends the investigation of MLTS to EV, using a standard planar nonlinear two-track car model and a simple Thevenin battery model. The Nuerburgring Grand Prix course is the demonstration track. The transcription of the continuous OCP is obtained using Hermite-Simpson collocation within the JuMP framework [73, 74], and the programming language Julia [74]. The derivative information is computed via automatic differentiation using [75]. The NLP is IPOPT [47]. The same solution technique is employed in [76] to study the effect of active rear-axle steering, and longitudinal torque allocation. The illustration track is again the Nuerburgring. It is found that the torque allocation potential is greater than that possible with rear-axle steering.

A road with variable (location-dependent) road-tyre friction coefficient is introduced



(a)

(b)

Figure 9. Variable friction road map [77]. (a) Location-dependent friction map, and (b) Determining the friction coefficients for each wheel.

into MLTS in [77]. The track-friction map is given in terms of the absolute coordinates (x and y), and then converted into curvilinear coordinates (the travelled distance s and the lateral position n with respect to the centreline); see Figure 9. A linear regression with Gaussian basis functions is employed to limit the tyre-road friction; both single-track and double-track car models are considered. The problem is solved with IPOPT [47] after transcribing the OCP as a NLP through orthogonal Legendre-Gauss collocation. Algorithmic differentiation is employed using CasADi [78]. A Roborace car is simulated on the Berlin Formula E racetrack, with changes in the lap time and racing line, due to the introduction of the variable-friction map demonstrated. The computation times are approximately 100 s for the single-track model and 120 s for the double-track model on a standard laptop computer, with a mesh sampling interval of 3 m. Variations in the tyre-road friction of ±0.1 show variations in the optimal trajectory up to 2.9 m.

The MLTS of motorcycles on three-dimensional tracks is considered in [79] and [80]. The former includes suspensions and a linearly-saturated tyre model (combined with an indirect solution method [24]), while the latter employs a slightly embellished variant of the classical 'Whipple model' [1] using Magic Formula based tyres (combined with a direct solution method [42]). In [80] measured data was obtained from an instrumented motorcycle ridden by a professional rider. Both studies simulate a racing motorcycle on the three-dimensional Mugello circuit, with similar results and trends found. The comparison between the measured and simulated data highlights the possibility of predicting the directional behaviour of the vehicle, in a minimum-time manoeuvring framework. These results also provide novel insight into the riding style of top-level racing riders on high-power machines. It was found that the most important influences on the lap time were: (i) the centrifugal forces induced by the road pitch rate, which provide additional down forces that can broaden the vehicles performance envelope, and (ii) the road banking that makes it possible to support higher cornering speeds without exceeding the tyre limits.

A QSS method without a pre-defined trajectory is introduced in [81]—recall that QSS simulations are usually solved using a pre-defined trajectory that is determined by other means. The method builds on established g-g diagram type ideas, with an OCP solver used to compute the optimal trajectory and the related lap-time. The overarching idea is to encapsulate the vehicle model complexities in g-g diagrams that 'encode' the performance envelope of the vehicle; these diagrams can be determined either numerically or experimentally. The related OCP is simple, and forces the vehicle represented by the g-g diagrams to follow the track in the minimum time. Comparison between the standard MLTS with a predefined trajectory are discussed. The optimal controls are the longitu-

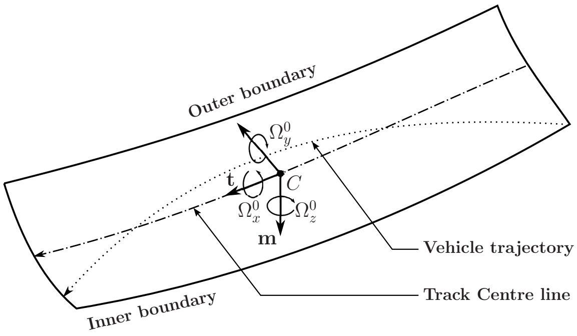


Figure 10. Curved ribbon model in [83]

dinal and lateral accelerations, while the states are the speed, the lateral position and relative orientation with respect to the centreline. The method is applied to both cars and motorcycles. The OCP is solved using the direct method described in [42].

The QSS method, without a pre-defined trajectory, is extended to deal with three-dimensional track in [82]. The g-g maps are replaced by g-g-g diagrams; vertical acceleration is added to the longitudinal and lateral accelerations. The method is again demonstrated for both a MotoGP motorcycle and a Formula One car on the Mugello and Barcelona-Catalunya tracks. The optimal controls are the rates-of-change of the accelerations, while the states remain those of the two-dimensional model, *i.e.* speed, lateral position and orientation with respect to the centreline on the road tangent plane. The road is defined using three curvatures as summarized in [1]. Semi-analytical motorcycle and car models are used to compute representative g-g-g diagrams. It is found that the computation time for the free-trajectory QSS approach is approximately 10 times faster than the free-trajectory approach using a dynamic model, and about 10 times slower than the pre-defined trajectory QSS approach—there are thus two orders of magnitude difference between the typical solution times obtained with free-trajectory full-dynamic models, and fixed-trajectory QSS simulations.

Most recently, an extended three-dimensional road model is described in [83]. The idea is to lift the assumption that transverse sections of the road are 'straight', *i.e.* that the road banking does not change when moving laterally across the track at fixed elapsed distance. This variable transverse camber feature is particularly important on some NASCAR ovals; at least one such track is characterized by camber angle variations of up to 10 deg. NASCAR tracks are also characterized by large banking angles (of up 30 deg), and the oft-employed 'small camber angle' assumption is not justifiable. In this work the classical ribbon model with flat lateral strips is replaced by a curved ribbon as in Figure 10. The geodesic and normal curvatures, and the relative torsion, are still used to describe the road centreline, where the moving Darboux frame is located. In contrast to prior work, the road elevation in the transverse direction is modelled using a multivariate polynomial fitted to experimental data. The road reconstruction problem is again cast as an OCP, where the cost is the error between the reconstructed road profile and a 3D mesh of experimental data. The performance of a NASCAR car is simulated on the Darlington Raceway (USA), showing that lateral shifts in the optimal trajectory of several meters, as compared to a flat ribbon, are required in some corners. These lateral position shifts exploit the increased camber angles at the outside periphery of the track. The MLTS is obtained using the method in [82]; the road model developed is equally suitable for application with dynamic models.

We mention in conclusion that two popular direct and indirect solution methods for vehicular OCP are compared and contrasted in [84](PINS [24, 85] and GPOPS-II [42]).

Three scenarios relevant to vehicle applications are considered. A minimum-time manoeuvre with a basic two-wheeled model, the road reconstruction problem, and a MLTS for a GT car on the Adria International Raceway. The results showed that each approach has advantages and disadvantages—PINS excelled in terms of computation time and absolute accuracy, while GPOPS-II was more robust and showed high accuracy on initially coarse meshes. Results showed that direct and indirect methods have a similar overall performance when dealing with MLTS problems. Most of the observed differences between the two solvers can be explained in terms of the numerical implementations rather than intrinsic differences. These variations in implementation include: the integration scheme used, mesh refinement schemes, multiplier initialisations, penalty tuning algorithms, derivative computations, and the state projection used in feasible regions.

2.5. *Summary remarks*

The above survey indicates that there are three main approaches to the MLTS for road vehicles: QSS simulations with a predefined trajectory, transient simulations without a predefined trajectory, and QSS simulations without a predefined trajectory. The first is the oldest and simplest method, usually solved using the so-called ‘apex-finding’ technique, or as an OCP; most implementations are built around g-g diagrams. The second is the most general method, and began to gain traction in the 2000s. This approach employs dynamic vehicle models, and track descriptions of varying complexity. The OCP is then solved by transcription and a NLP solver, or with an indirect method that is based on the solution to a TPBVP. In indirect methods the constraint equations are discretised along with appropriate adjoint equations. The third method is the most recent, and seeks to exploit some features of the other two techniques. This approach generally results in simpler and faster-to-solve OCP.

Several common features characterizes most of the work reported in the literature. These include: (i) describing equations are written in the spatial domain (rather than the time domain), (ii) some form of magic-formula-type tyre descriptions are used, (iii) curvilinear track and vehicle kinematics descriptions are used, (iv) the OCP inputs comprise rates-of-change of the physical model input (or filtered versions thereof), (v) typically collocation methods are employed to solve the OCP, *i.e.* both the states and controls are discretized on a mesh, with the resulting equations solved simultaneously.

Both cars and motorcycles have been studied in detail. In the case of cars, both single- and double-track descriptions have been employed. Quasi-steady-state load transfer descriptions form an integral part of the vehicles mathematical description. Most detailed 3D track descriptions are based on a ribbon model, with recent work extending this idea to so called ‘curved ribbons’. Many problem-specific embellishments have been included in the basic vehicular optimal control framework. These include: (i) vehicle-trim-dependent aerodynamic models, (ii) thermodynamic tyre models, including wear models, and multi-lap simulations, (iii) variable tyre-road friction descriptions, (iv) complex hybrid power trains that include KERS systems, thermal energy recovery systems, and pure-electric drivetrains, (v) torque vectoring and rear-axle steering, (vi) emissions and detailed ICE models. It is reasonable to expect that this list will continue to lengthen.

3. Fundamentals of OCP

Most of the MLTS referred to above rely on the numerical solution of a nonlinear OCP. The purpose of this section is to review some of the relevant fundamentals. The material discussed here is mainly from [1], with additional background available in a number of classic books on optimal control including [15, 86–89].

The dynamical system, which in the case of MLTS, is the vehicle and its related subsystems, is usually described by a set of ordinary differential equations (ODE)

$$\boldsymbol{x}'(s) = \boldsymbol{f}(\boldsymbol{x}(s), \boldsymbol{u}(s), s) \qquad s_i \leq s \leq s_f. \tag{2}$$

The prime denotes derivatives with respect to the (independent) elapsed-distance variable s; s_i and s_f are the initial and final values of s respectively. In most MLTS, this distance is measured along the centreline of the track. The state vector is $\boldsymbol{x}(s) \in \mathbb{R}^n$ and $\boldsymbol{u}(s) \in \mathbb{R}^m$ are the controls; $\mathbb{R}^k$ denotes the space of k-dimensional vectors of real variables.

In the so-called Bolza form, the cost function, or performance index (PI) is

$$J(\boldsymbol{x}(s), \boldsymbol{u}(s), s) = \mathcal{M}(\boldsymbol{x}(s_i), s_i, \boldsymbol{x}(s_f), s_f) + \int_{s_i}^{s_f} \mathcal{L}(\boldsymbol{x}(s), \boldsymbol{u}(s), s) ds. \tag{3}$$

In the case of a MLTS the prototypical PI is

$$J = \int_{s_i}^{s_f} \frac{1}{\dot{s}} ds = T, \tag{4}$$

where $\dot{s} = ds/dt$ is the velocity of the vehicle projected on to the track centreline, and T the manoeuvre execution time. The travelled distance associated with an arbitrary manoeuvre (such at executing a U-turn) is $s_f - s_i$. In the case of closed-circuit racing

$$J = \oint \frac{1}{\dot{s}} ds, \tag{5}$$

is used. The path integral is evaluated along the centre-line (or spine) of the track, with the path length the lap distance.

The objective of the OCP is to minimize J, while simultaneously satisfying the dynamic equations (2), and the path constraints

$$\boldsymbol{c}(\boldsymbol{x}(s), \boldsymbol{u}(s), s) \leq \boldsymbol{0}. \tag{6}$$

These constraints may, for example, include ‘stay on the track’ conditions. The boundary constraints are given by

$$\boldsymbol{b}(\boldsymbol{x}(s_i), s_i, \boldsymbol{x}(s_f), s_f) = \boldsymbol{0}. \tag{7}$$

In the case of closed-circuit racing, these boundary constraints might be used to enforce continuity condition across the start-finish line. It is intuitively clear, for example, that the tyre carcass temperatures must be the same an instant before, and an instant after the start-finish line—these are continuity, or ‘no jumps allowed’ constraints.

In the case that the solution is tackled using an indirect approach, the *Control Hamiltonian*

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{u}, s) = \mathcal{L}(\boldsymbol{x}(s), \boldsymbol{u}(s), s) + \boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}(s), \boldsymbol{u}(s), s) + \boldsymbol{\mu}^T \boldsymbol{c}(\boldsymbol{x}(s), \boldsymbol{u}(s), s) \tag{8}$$

is of importance; $\boldsymbol{\lambda}$ is the co-state, while $\boldsymbol{\mu}$ are the Lagrange multipliers associated with the inequality constraints.

The optimal control problem gives rise to the following first-order necessary conditions:

$$\boldsymbol{x}' = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}, \tag{9}$$

$$\boldsymbol{\lambda}' = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}, \tag{10}$$

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\mathcal{U}} \mathcal{H}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \boldsymbol{u}, s), \tag{11}$$

where $\mathcal{U}$ is the set of admissible controls. The Lagrange multipliers associated with the inequality constraints must satisfy the *complementary slackness* conditions

$$\begin{aligned} \mu_j(s) = 0, \text{ when } \quad & c_j(\boldsymbol{x}, \boldsymbol{u}, s) < 0, \quad j = 1, \dots, p \\ \mu_j(s) < 0, \text{ when } \quad & c_j(\boldsymbol{x}, \boldsymbol{u}, s) = 0, \quad j = 1, \dots, p \\ \mu_j(s) > 0, \text{ when } \quad & c_j(\boldsymbol{x}, \boldsymbol{u}, s) > 0, \quad j = 1, \dots, p. \end{aligned} \tag{12}$$

The (mixed) boundary conditions are given by

$$\begin{aligned} &\text{either} \quad \boldsymbol{x}(s_i) \quad \text{fixed,} \quad \text{or} \quad \boldsymbol{\lambda}(s_i) = -\frac{\partial \mathcal{M}}{\partial \boldsymbol{x}}(s_i) + \left(\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{x}}(s_i)\right)^{\mathsf{T}} \boldsymbol{\nu} \\ &\text{either} \quad \boldsymbol{x}(s_f) \quad \text{fixed,} \quad \text{or} \quad \boldsymbol{\lambda}(s_f) = +\frac{\partial \mathcal{M}}{\partial \boldsymbol{x}}(s_f) - \left(\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{x}}(s_f)\right)^{\mathsf{T}} \boldsymbol{\nu}, \end{aligned} \tag{13}$$

where $\boldsymbol{\nu}$ is the lagrange multiplier associated to the boundary constraints. This means that (9), (10) and (13) constitute a TPBVP. One typically has initial conditions on $\boldsymbol{x}$, and terminal conditions on $\boldsymbol{\lambda}$. In combination, equations (9) and (10) form a Hamiltonian system, in which (9) are the state equations and (10) the co-state equations. Condition (11) is the *Pontryagin Minimum Principle* (PMP).

If (8) is linear in the controls, and one collects these control-related terms in a function $\boldsymbol{\Phi}(s)$, called a *switching function*[2], the PMP demands that

$$\boldsymbol{u}^*(s) = \begin{cases} \boldsymbol{u}_{\max} & \text{if } \Phi(s) < 0 \\ \text{unknown} & \text{if } \Phi(s) = 0 \\ \boldsymbol{u}_{\min} & \text{if } \Phi(s) > 0 \end{cases} \tag{14}$$

where $\boldsymbol{u}_{\max}$ and $\boldsymbol{u}_{\min}$ are, respectively, upper and lower bounds on the admissible controls $\boldsymbol{u}(s)$. If the condition $\boldsymbol{\Phi}(s) = 0$ does not occur over any finite interval, the controls are of *bang-bang* type—controls of this type are endemic in MLTS problems. In the case that $\boldsymbol{\Phi}(s) = 0$ over a finite interval $\Delta s > 0$, the PMP says nothing about the optimal controls on that interval. These problems are called *singular.* Consider a minimum-time problem for a long straight section of road with a powerful car and a finite speed constraint (limit). It is clear that the optimal strategy would be to apply maximum power until the car reaches the speed limit $v = v_{max}$. After that, one would simply apply enough power to overcome such things as drag, thereby maintaining $v = v_{\max}$ until the end of the road

[2] The switching function must be evaluated on an optimal trajectory.

section. In this case the engine power satisfies $u_{\min} < u^* < u_{\max}$, and is consequently a *bang-singular* problem—it is 'bang' while $u^* = u_{\max}$, and singular thereafter. If the initial speed is $v_{\max}$, the entire arc is singular. Finally, if one applies a terminal speed constraint such as $v_f < v_{\max}$, the problem becomes *bang-singular-bang.* The second 'bang' arc would be $u^* = u_{\min}$, which is used to brake the car as quickly as possible so as to meet the terminal constraint $v(s_f) = v_f$. While important, these problems often present numerical difficulties.

The most common solution methods for indirect formulations are discussed in Section 4, and involve discretizing equations (9) and (10) with (11) on a finite mesh along the centre line of the track. This discretization process reduces the infinite-dimensional TPBVP into one of finite dimension.

In the case that the OCP is tackled using a direct approach, the control Hamiltonian (8), the co-state equations (10), and the PMP equations (11) need not be derived. Instead, the OCP in (2)-(7) can be solved by transcription using a NLP solver. That said, when searching for the minimum, most optimization routines use internally necessary conditions called the Karush-Kuhn-Tucker (KKT) conditions [90–93]. In the limit $\Delta s_i \to 0$, the KKT equations become the first-order necessary conditions associated with indirect methods [1, 88].

4. Solution methods

Due to their inherent complexity, most practical optimal control problems must be tackled numerically; this implies immediately that some form of discretization will be required. Techniques that make explicit use of first-order necessary conditions such as those described in Section 3, are called *indirect methods.* Other methods, such as *dynamic programming* [15], or the transcription of the continuous OCP into a NLP, will be referred to as *direct methods* [88].

Once the solution methodology has been chosen, the problem is solved using either *time marching* or *collocation-based* simulation. In time marching, the dynamics are integrated using an algorithm in which the solution at a given time step is obtained from the solution of the dynamics at one or more previous time steps. In collocation-based methods the solution at every steps is obtained simultaneously by solving a set of algebraic equations—in this framework any notion of time marching, is suppressed.

4.1. *Indirect time marching*

An 'obvious' solution methodology to the TPBVP given by (6)–(13) is some form of trial-and-error iteration combined with a time-stepping numerical integration algorithm. In *indirect shooting*, the known initial conditions are used in combination with estimates of those currently missing. Equations (9)-(10) are then solved from s_i to s_f using time marching, with the optimal control given by (11). One then compares the resulting terminal conditions with those already provided in (13). Any mismatch is then used to revise the unknown initial conditions with the process repeated to convergence. This process is reminiscent of how students converge on the solutions to their homework exercises, by invoking multiple update visits to their tutors. The same approach can be applied using the the known final conditions in combination with estimates of those currently missing.

While this shooting methodology might be intuitively obvious, it has a significant shortcoming—the TPBVP associated with Hamiltonian systems have inherently unstable dynamics. This solution procedure will be problematic when the time interval of interest is long compared with the growth rates of the Hamiltonian system dynamics in a neighbourhood of the optimal solution.

In order to overcome this difficulty the *multiple-shooting* method was developed. In this case the interval $[s_i, s_f]$ is divided into K subintervals $\mathcal{S}_k = [s_{k-1}, s_k]$, where $s_0 = s_i$ and $s_K = s_f$ with $\bigcup_{k=1}^{K} \mathcal{S}_k = [s_i, s_f]$. The shooting method is then applied over each subinterval $[s_{k-1}, s_k]$. In order to enforce continuity, the following conditions

$$\boldsymbol{x}(s_k^-) - \boldsymbol{x}(s_k^+) = \boldsymbol{0}, \quad k = 1, \ldots, K-1, \tag{15}$$

must be recognised in combination with (13). Despite being of higher dimension, the multiple-shooting method represents an improvement, because the shorter integration intervals reduce the sensitivity to errors in the unknown terminal conditions.

4.2. *Direct time marching*

Another basic time-marching method is *direct shooting.* In this case the control is parametrized using

$$\boldsymbol{u}(s) = \sum_{j=1}^{J} \boldsymbol{a}_j \psi_j(s), \tag{16}$$

where $\psi_j(s)$ are pre-assigned basis functions and $\boldsymbol{a}_j$ are parameters to be determined by the optimization. The system dynamics can be evaluated by integrating the differential equations (2) over the interval of interest $[s_i, s_f]$; this is an initial value problem (IVP). The cost function (3) is also determined using a numerical approximation that is consistent with the numerical integrator used to solve the differential equations. The following standard NLP arises from the approximation of the optimal control problem:

$$\min \ F(\boldsymbol{z}) \text{ subject to } \begin{cases} \boldsymbol{z}_{\min} \leq \boldsymbol{z} \leq \boldsymbol{z}_{\max} \\ \boldsymbol{g}(\boldsymbol{z}) \leq \boldsymbol{0}, \end{cases} \tag{17}$$

which is typically ‘large’, containing the decision variables $\boldsymbol{z}$ and constraints $\boldsymbol{g}$—in this particular case $\boldsymbol{z}$ consists of the $\boldsymbol{a}_j$ coefficients in (16).

In a manner similar to that used for indirect methods, *direct multiple-shooting* may be employed, with the interval $[s_i, s_f]$ again divided into K subintervals $\mathcal{S}_k$. In this way the problem is decomposed into K-IVPs with

$$\boldsymbol{u}^{(k)}(s) = \sum_{j=1}^{J} \boldsymbol{a}_j^{(k)} \psi_j^{(k)}(s). \tag{18}$$

The values of the states $\boldsymbol{x}(s_{k-1})$ at the beginning of each subinterval and the unknown coefficients $\boldsymbol{a}_j^{(k)}$ in the control parameterization (18) are the unknowns in the NLP. To enforce continuity (15) are again employed at the subinterval interfaces. The key feature of these problems, which makes then tractable, is the fact that the NLP is *sparse.* Well known solvers such as SNOPT [20], IPOPT [47] and KNITRO [94] may be used for solving such large sparse NLPs.

4.3. *Collocation methods*

Collocation methods produce (smooth) approximations to the solutions $\boldsymbol{x}(s)$ in (2) in the form of polynomial functions. These polynomials satisfy the given differential equation on a finite set of points in the integration interval, and coincides exactly with the initial condition $\boldsymbol{x}(s_{k-1})$ in each integration sub-interval. Suppose the solution interval $[s_i, s_f]$ is divided into K mesh intervals $\mathcal{S}_k$, as in the case of multiple shooting. In each k–interval $\mathcal{S}_k$, the solution of the dynamic equations (2) is approximated in terms of a set of basis functions $\psi_j^{(k)}(s)$

$$\boldsymbol{x}^{(k)}(s) \approx \boldsymbol{X}^{(k)}(s) = \sum_{j=0}^{J} \boldsymbol{c}_j^{(k)} \psi_j^{(k)}(s). \tag{19}$$

These basis functions are usually in the form $\psi_j^{(k)}(s) = (s - s_{k-1})^j$, so that the approximation $\boldsymbol{X}^{(k)}(s)$ is a J^{th}-order polynomial, whose coefficients $\boldsymbol{c}_j^{(k)}$ are defined in order to match the initial condition at the beginning of the interval, and the dynamic equations (2) at the *collocation points* $s_j = s_{k-1} \leq s_1, \cdots s_J \leq s_k$ within $\mathcal{S}_k$—at these points $\boldsymbol{X}'(s_j) = \boldsymbol{f}\left(\boldsymbol{x}(s_j), \boldsymbol{u}(s_j), s_j\right)$. If there are only two collocation points at s_{k-1} and s_k, $\boldsymbol{X}^{(k)}(s)$ is quadratic, and the well-known trapezoidal integration rule results—the default used is ICLOCS [46]. In the case that there is only one collocation point in the middle of the interval, the midpoint integration rule is obtained. This is the rule employed in PINS [24] to discretize the TPBVP to give a large set of nonlinear algebraic equations (typical size for MLTS $> 100,000$). If three collocation points are used, at the beginning, middle and end of the interval, the Simpsons rule results and $\boldsymbol{X}^{(k)}(s)$ is cubic; this method is used e.g. in SOCS [88].

Assuming basis functions of order J in (19), this parameterization is used to integrate the equations from s_{k-1} to 'stage times' $s_{ik} \in \mathcal{S}_k$, where $s_{k-1} \leq s_{1k} < s_{2k} < \cdots < s_{(J-1),k} < s_{Jk} \leq s_k$ for all $i \in [1, \ldots, J]$ using

$$0 = \boldsymbol{X}_{ik} - \boldsymbol{X}_{k-1} - \sum_{j=1}^{J} A_{ij}^{(k)} \boldsymbol{f}(\boldsymbol{X}_{jk}, \boldsymbol{U}_{jk}, s_{jk}), \tag{20}$$

where $\boldsymbol{X}_{jk}$ and $\boldsymbol{U}_{jk}$ are the values of the function approximations $\boldsymbol{X}(s)$ and $\boldsymbol{U}(s)$ at the stage points s_{jk}; both the states and controls are discretized. The matrices $A_{ij}^{(k)}$ are *integration matrices* associated with the particular integration (quadrature) rule being employed. These defect conditions are satisfied at all the collocation points within each mesh interval. The subinterval boundary conditions (15) must also be satisfied. Given the controls at each collocation point, the goal is to solve for the coefficients $\boldsymbol{c}_j^{(k)}$ in each mesh interval. As opposed to time-marching methods, all of the defect equations are solved *simultaneously* as a large-scale algebraic system.

Of particular importances are methods that use collocation points that are the roots of *orthogonal* polynomials, which are called *pseudo-spectral* methods. Of relevance here are the *Legendre-Gauss* (LG), the *Legendre-Gauss-Labbato* (LGL) and *Legendre-Gauss-Radua* (LGR) schemes that are all slightly different. In the case of LG schemes, the

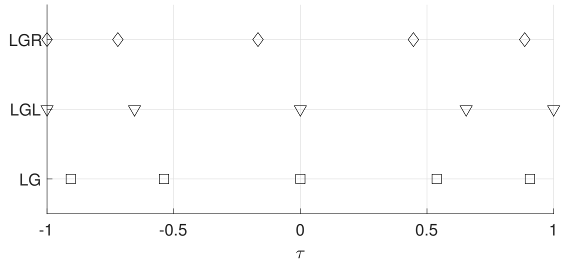


Figure 11. LG, LGL and LGR collocation points in the of $J = 5$.

collocation points are the J roots of the *Legendre polynomials*, which are given by

$$P_J(\tau) = \frac{1}{2^J J!}\frac{\partial^J}{\partial \tau^J}(\tau^2 - 1)^J, \tag{21}$$

with the basis functions given by the *Lagrange polynomials*

$$L_j(\tau) = \sum_{\substack{1 \le m \le J \\ m \ne j}} \left(\frac{\tau - \tau_m}{\tau_j - \tau_m}\right). \tag{22}$$

There are three important points to note here: (i) the Legendre polynomials have simple distinct roots that lie in the interval $-1 < \tau < 1$, (ii) $L_j(\tau_j) = 1$ and $L_j(\tau_i) = 0$ for $i \ne j$ (the isolation property), and (iii) the root locations of the Legendre polynomials mandate the normalization of the independent variable so that

$$\tau = \frac{s_{k-1} + s_k - 2s}{s_{k-1} - s_k}. \tag{23}$$

It is clear that $\tau = 1$ when $s = s_k$, and $\tau = -1$ when $s = s_{k-1}$. The 'isolation' property of the Lagrange polynomials means that

$$\boldsymbol{x}(\tau) \approx \boldsymbol{X}(\tau) = \sum_{j=1}^{J} \boldsymbol{X}(\tau_j) L_j(\tau). \tag{24}$$

One of the advantages of LG collocation, is that operating with J points is comparable in terms of accuracy, to operating with $2J$ equidistant points [95]. Comparison with (19) reveals that the $\boldsymbol{c}_j$'s are approximations of the states $\boldsymbol{X}$ at the collocation points, while the basis functions ψ are replaced by the Lagrange polynomial $L_j(\tau)$.

The closely related LGR and LGL methods require modifications that we will now describe. In the case of the LGL method, we use collocation points that are the roots of $P'_{J-1}(\tau)$ (rather than $P_J(\tau)$) plus the terminal points $\tau = -1$ and $\tau = 1$; this is the primary method used in OTIS [96]. In the case of LGR schemes the collocation points are the roots of $(P_{J-1}(\tau) + P_J(\tau))/(\tau + 1)$, which include the initial point of the interval [97]; this is the primary method used in GPOPS [42, 98] and is also available in ICLOCS2 [99]. Both LGR and LGL are used in DYMOS [100], which is within OpenMDAO [101]. A further variation is the use of Hermite polynomials in place of Lagrange polynomials. Figure 11 shows the collocation points for the three methods in the case of $J = 5$. It can also be observed that the midpoint rule is the most basic LG scheme, while the trapezoidal and Simpson are basic LGL schemes [88]. The LGR method has also been shown to be a good estimator for the co-state of the OCP from the Lagrange multipliers of the KKT conditions of the related NLP problem [102, 103]. This suggests that solving the OCP with a direct method, *i.e.* without using the co-state equations, will give good estimates of the multipliers related to the co-state equations. Regardless of the specific choice of the collocation points, the primary solver for the resulting NLP problem is either SNOPT [20] or IPOPT [47]. These methods converge spectrally (*i.e.* at an exponential rate) as a function of the number of collocation points (for smooth problems), hence the name *pseudo-spectral* [104].

4.4. *Mesh refinement*

A key aspect of efficiently generating accurate solutions to OCP using collocation methods is the placement of both the mesh segment boundaries, and the collocation points. The algorithmic determination of the mesh boundaries and the related collocation points is called *mesh refinement.* In a traditional fixed-order collocation method the degree p of the approximating polynomial in each interval is fixed, and the mesh refinement algorithm determines the location of the mesh boundaries—these are called h-type methods. In contrast, p-type algorithms use fixed mesh in combination with variable-order approximating polynomials. When both h and p are varied, the algorithm is of h-p type; the h and p variants can be considered special cases of h-p algorithms.

Both pure h and pure p method have their limitations. Achieving high accuracy may require an extremely fine mesh (in the case of an h method), or it may require the use of an unreasonably high-order degree polynomial in the case of a p method. A specific limitation of p methods is their ability to deal with discontinuities that occur routinely in minimum-time OCP—increasing the order of approximating polynomials for discontinuous functions is of limited utility. It is better to use an h algorithm to place discontinuities on segment boundaries. The use of fixed low-order polynomials (h methods) for smooth functions with poor polynomial representations also has obvious drawbacks. In a general MLTS setting h-p orthogonal collocation mesh-refinement methods are thus well motivated. Algorithms based on h-p mesh refinement are described in a number of papers including [105–107]. As a general rule, if the error across a particular segment is large and uniformly distributed, the polynomial order should be increased. If a significant error occurs at an isolated point within a segment, the segment should be subdivided.

4.5. *Scaling*

Scaling usually has a significant influence on the performance of optimization algorithms. Since convergence tolerances, and other criteria, are necessarily based on notions of ‘small’ and ‘large’ quantities, problems with unbalanced scaling may cause difficulty. One notion of scaling is to transform the variables from their original representation, which may reflect the physical nature of the problem, to dimensionless quantities that have desirable properties in terms of the convergence of optimization algorithms. In the case of vehicular problems, it is common to make the length of the car or motorcycle the fundamental unit of length—after scaling, the vehicle has a dimensionless length of one. In the same way, the fundamental unit of mass is the mass of the vehicle—after scaling, the vehicle has a dimensionless mass of one. In order to scale time, we give the vehicle unit weight, and so a time scale of $\sqrt{g/l_0}$ is used in which l_0 is the vehicle’s length and g is the acceleration due to gravity ($\sqrt{l_0/g}$ is thus the fundamental unit of time). Once normalizing scale factors for length, mass, and time have been defined, all other scale factors can be defined in terms of combinations of these quantities. During racing a vehicle's speed might be 90 m/s; after scaling this becomes 15.58 in dimensionless units if $l_0 = 3.4$ m. During firm acceleration and fast cornering, the car's longitudinal and lateral tyre forces might reach 8,000 N, which becomes 1.236 in dimensionless units if the vehicle's mass is 660 kg; the force scale factor is derived from the mass, length, and time scale factors. This scaling scheme forces the nonlinear solver's decision variables into a 'more spherical' space as opposed to an elongated hyperellipsoid one.

4.6. *Gradients*

NLP solvers usually require derivatives of the cost function and constraints with respect to both the state and control variables. For this reason non-smooth problem features have to be approximated by differentiable functions in a way that does not change significantly the problem's solution. Functions such as $\min(x, 0)$ and $\max(x, 0)$ have undefined derivatives at $x = 0$, and can be conveniently approximated using

$$\max(x,0) \approx \frac{x + \sqrt{x^2+\epsilon}}{2} \quad \text{and} \quad \min(x,0) \approx -\frac{-x + \sqrt{x^2+\epsilon}}{2}, \tag{25}$$

in which ϵ is a 'small' constant. Since

$$\frac{\partial}{\partial x}\left(\frac{\pm x + \sqrt{x^2+\epsilon}}{2}\right) = \frac{1}{2}\left(\frac{x}{\sqrt{x^2+\epsilon}} \pm 1\right), \tag{26}$$

one sees that the derivative approximations are now well defined at $x = 0$. One may also make use of $\max(a, b) = a + \max(b - a, 0)$. In the case of $|x|$, one might use

$$|x| \approx \sqrt{x^2+\epsilon}.$$

In this case

$$\frac{\partial}{\partial x}\sqrt{x^2+\epsilon} = \left(\frac{x}{\sqrt{x^2+\epsilon}}\right),$$

with the derivative approximation at $x = 0$ again well defined. These approximations become more accurate as the value of ϵ is reduced, with values in the range $10^{-5} \leq \epsilon \leq 10^{-2}$ typical for MLTS.

When it comes to computing gradients practically, there are at least four possible approaches. The first and most obvious is to compute derivatives analytically, either by hand, or by using a computer algebra package such as Maple$^{\text{TM}}$ or Mathematica$^{\text{TM}}$. While the precision of this approach is appealing, and generally results in faster rates of convergence, it is sometimes not practical to compute derivatives this way.

Another approach is to compute derivatives numerically using finite (forward, backward or central) differencing. In the case of forward differencing one uses

$$\frac{df}{dx} \approx \frac{f(x+\Delta) - f(x)}{\Delta}$$

in which Δ is the step length. The difficulty with this approach is that Δ must be small enough to provide a good approximation to the derivative, but not so small that round-off errors occur when calculating the difference $f(x+\Delta) - f(x)$—this is the well-known *step-size dilemma.*

Another approach is *complex-step differentiation*, which is both accurate and efficient [108–110]. In the case of higher-order derivatives a more general version of the complex-step derivative approximation, called the multi-complex-step derivative approximation, can be used as a way of computing Hessians without truncation errors [111].

Another powerful approach is algorithmic or *automatic differentiation* (AD), which computes function derivatives to the precision that one would achieve using analytic or symbolic differentiation [71, 78, 112, 113]. While any of these approaches can be used, symbolic and automatic differentiations are most common in vehicular minimum-time problems.

Finally, relaxation 'tricks' can be used to avoid discontinuous Mixed-Integer Programming problems, such as gearbox modelling. In this case one might approximate the gearbox with a continuously-variable transmission. Suppose there are n gear ratios of $\kappa_1, \cdots \kappa_n$, with corresponding gear-change speeds v_i. One can compute the engaged gear ratio κ using

$$\kappa = \kappa_1 + \sum_{i=1}^{n} \left(\frac{\kappa_{i+1} - \kappa_i}{2} \right) (1 + \sin(\tan^{-1}(k(v - v_i)))), \tag{27}$$

where k is a 'switching' constant that is selected to provide robust numerical performance. If the vehicle speed $v \leq v_i$, gear κ_i is engaged.

4.7. *Regularization*

When the controls enter the system dynamics and performance index linearly, the possibility of bang–bang controls and/or singular arcs exists; see Section 3. The majority of general-purpose optimal control software codes have difficulties with problems containing singular arcs. In order to avoid oscillatory solutions and non-convergent behaviours with these problems, it is often useful to introduce into the performance index small terms that are quadratic (*i.e* not linear) in the controls. These terms ensure, at least in theory, that the Pontryagin minimum principle can be used to find the optimal controls (for the perturbed problem). If these terms are 'small' relative to the main cost, they do not change the problem in any significant way. The regularized problem has the following cost

$$\hat{J} = J + \int_{s_i}^{s_f} \sum_{i=1}^{m} \epsilon_i u_i^2 ds, \tag{28}$$

in which the ϵ_i's are small constants with the u_i's the controls.

5. Road modelling

As was mentioned in Section 2, the majority of the published work on MLTS is based on two-dimensional road models with uniform friction characteristics. More advanced road models are three-dimensional and include variable camber and elevation—variable

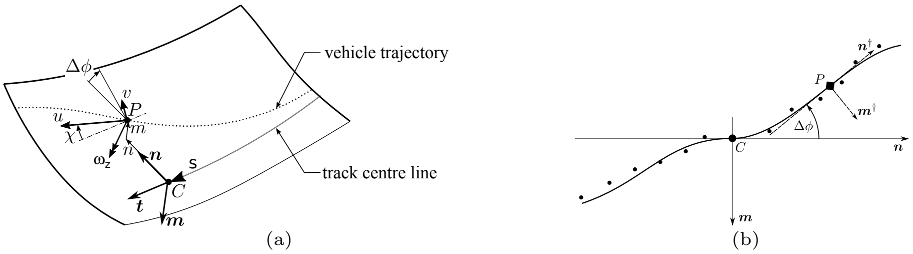


Figure 12. Three-dimensional road model, from [83].

tyre-road friction studies are unusual, but have nonetheless been reported in publications such as [77]. Curvilinear coordinate based road models are ubiquitous and will now be summarized in the three-dimensional case.

In accordance with classical Frenet-Serret theory, the track centre line is a space curve that can be described by two curvature functions, namely the curvature κ and the torsion τ. Both variables are a function of the elapsed distance s. When the centre line is ‘broadened out’ into a ribbon-based track description (think of a strip of tagliatelle bent to represent the road), the road’s twist rate ν' is required to represent the track’s camber [1]. A moving trihedron, called the *Darboux frame*, moves with the vehicle along the road’s centreline with its x-y plane tangent to the road surface. An orthogonal basis for the Darboux frame is $[\boldsymbol{t}, \boldsymbol{n}, \boldsymbol{m}]$, which can be described using the conventional attitude-pitch-roll convention as follows

$$R = R_z(\theta)R_y(\mu)R_x(\phi) = \begin{bmatrix} c_\theta c_\mu & c_\theta s_\mu s_\phi - s_\theta c_\phi & c_\theta s_\mu c_\phi + s_\theta s_\phi \\ s_\theta c_\mu & s_\theta s_\mu s_\phi + c_\theta c_\phi & s_\theta s_\mu c_\phi - c_\theta s_\phi \\ -s_\mu & c_\mu s_\phi & c_\mu c_\phi \end{bmatrix}, \tag{29}$$

where $R_z(\theta), R_y(\mu)$ and $R_x(\phi)$ are respectively elementary rotation matrices around the z, y and x axes. The angles θ, μ and ϕ represent the attitude, slope and camber angles of the road; ‘s’ and ‘c’ refer to the sine and cosine of the corresponding (subscripted) Euler angles. The moving Darboux frame can be seen in Figure 12 (a) moving along the road centre line. The Euler angles, θ, μ and ϕ are obtained by integrating the curvature variables that are referred to as the *relative torsion* Ω_x^0, the *normal curvature* Ω_y^0, and the *geodesic curvature* Ω_z^0 [1, 48]. The angular spatial rate of the Darboux frame is the vector $\boldsymbol{\Omega}^0 = [\Omega_x^0, \Omega_y^0, \Omega_z^0]^T$. The Euler angles and the angular rate are related by

$$\begin{bmatrix} \Omega_x^0 \\ \Omega_y^0 \\ \Omega_z^0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s_\mu \\ 0 & c_\phi & c_\mu s_\phi \\ 0 & -s_\phi & c_\mu c_\phi \end{bmatrix} \begin{bmatrix} \phi' \\ \mu' \\ \theta' \end{bmatrix}, \tag{30}$$

where $\boldsymbol{\Omega}^0$ is expressed in the Darboux frame—a skew-symmetric representation of the angular rate (expressed in rad/m) is given by $R^T R'$. The location of the origin of the Darboux frame is given by

$$\begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} = \int \boldsymbol{t} ds = \int \begin{bmatrix} c_\phi c_\mu \\ s_\phi c_\mu \\ -s_\mu \end{bmatrix} ds, \tag{31}$$

where $\boldsymbol{t}$ (the first columns of (29)) is the tangent to the road centreline. This road model is laterally flat, *i.e.* the road camber does not vary laterally.

In order to accommodate lateral camber variations, which are required for some NASCAR oval models for example, it is assumed that the real road surface can float above, or submerge below the x-y plane of the Darboux frame [83]. To make this extension the road surface is described by

$$\mathcal{S} = \left\{ \boldsymbol{x}_P(s,n) = \boldsymbol{x}_C(s) + \boldsymbol{n}(s)n(s) + \boldsymbol{m}(s)m(s,n) \in \mathbb{R}^3 \right\}, \tag{32}$$

where $s \in [s_i, s_f]$ and $n(s) \in \frac{1}{2}[-r_w(s), r_w(s)]$; the $\boldsymbol{m}(s)m(s,n)$ term introduces lateral camber variations. The origin of a vehicle-fixed axis system is given in terms of a vector $\boldsymbol{x}_P$—this point is assumed to lie on the road surface and is commonly the projection of the vehicle's centre of mass for cars, and the rear ground-contact contact point for bikes; see Figure 12 (a). The vehicle's trajectory is again assumed to start at s_i and terminate at s_f; it is also assumed that n, and the road width r_w, are functions of s, with the position-dependent elevation $m(n,s)$ a function of both n and s. The legacy three-dimensional road model corresponds to the case $m(s,n) = 0$. The vectors $\boldsymbol{n}$ and $\boldsymbol{m}$ are represented by the second and third columns of (29).

If the surface $\mathcal{S}$ is to remain tangent to the road at positions away from the centre line, the camber angle must be adjusted to $\phi(s) + \Delta\phi(s,n)$, where[3]

$$\Delta\phi(s,n) = \operatorname{atan}(\partial_n m) \approx \partial_n m \tag{33}$$

for small changes in road elevation (in Darboux coordinates). These camber variations are illustrated in Figure 12 (b), where the dots represent the survey data and the solid line an approximating function $m(s,n)$. The impact that this camber change will make on the track curvature estimate at $\boldsymbol{x}_P$ can be calculated from (30), by replacing ϕ with $\phi + \Delta\phi$, to obtain

$$\begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \approx \begin{bmatrix} \Omega_x^0 \\ \Omega_y^0 \\ \Omega_z^0 \end{bmatrix} + \begin{bmatrix} \Delta\phi' \\ \Delta\phi\Omega_z^0 \\ -\Delta\phi\Omega_y^0 \end{bmatrix}, \tag{34}$$

where for small angles

$$\Delta\phi' \approx n'\partial_{nn}m + \partial_{sn}m. \tag{35}$$

The curve $m(s,n)$ can be any (sufficiently continuous) curve; in [83] an algebraic function is assumed.

In sum, the three-dimensional road model is described by the three road curvatures related to the centreline $(\Omega_x^0(s), \Omega_y^0(s), \Omega_z^0(s))$, plus the coefficients of the algebraic function representing the transverse undulations $m(s,n)$. The Euler angles associated with the Darboux frame, and the vehicle-fixed frame with origin $\boldsymbol{x}_P$, can be found by integrating the curvatures $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}^0$ respectively [83].

6. Road reconstruction

Once the structural properties of the road model have been defined (Section 5), the associated parameters have to be identified from measured survey data—this is the *road*

[3] We use the notation $\partial_n m$ to denote $\frac{\partial m}{\partial n}$, and $\partial_{sn} m$ to denote $\frac{\partial^2 m}{\partial s \partial n}$ and so on.

(model) reconstruction problem. The direct derivation of curvature information from survey data (GPS, LiDar, etc.) has a number of difficulties, including drift, measurement noise distortion, and the failure of the ends of closed-circuit models to meet. A number of different approaches are reported in [9, 114, 115], and the references therein, with techniques based on the solution of secondary OCP effective and simple to implement.

The OCP aims to minimize the error between the reconstructed road model and the experimental survey data—importantly, road-closure constraints are used to prevent the beginning and end of closed-circuit models from 'missing each other'. The optimal controls are derivatives of the Euler angles described in (29), as well as the derivatives of the road width for variable-width tracks.

In 'flat ribbon' modelling one usually requires survey data relating to the track boundaries, which might be noisy. The optimal control framework described in Section 3 is utilized. The road model state vector comprises three position variables for the track centre line, integrator chains for each of the Euler angles, and a road-width variable

$$\boldsymbol{x} = [x_C, y_C, z_C, \phi, \phi', \mu, \mu', \theta, \theta', r_w]^T. \tag{36}$$

The controls are the rates of change of the three Euler angles in (29) and the road width

$$\boldsymbol{u} = [u_\phi, u_\mu, u_\theta, u_{rw}]^T, \tag{37}$$

which are manipulated so that the trajectory and orientation of the Darboux frame circumnavigates the race track while minimising an estimation error given by (3) with

$$J = \int_{s_i}^{s_e} \left(\mathcal{R}(\boldsymbol{u}) + \mathcal{Q}(\boldsymbol{x})\right) ds. \tag{38}$$

The various terms therein are given by

$$\mathcal{R}(\boldsymbol{u}) = r_\phi u_\phi^2 + r_\mu u_\mu^2 + r_\theta u_\theta^2 + r_{rw} u_{rw}^2, \tag{39}$$

$$\begin{aligned}\mathcal{Q}(\boldsymbol{x}) = {} & q_x(x_l - x_{l0})^2 + q_y(y_l - y_{l0})^2 + q_z(z_l - z_{l0})^2 + \\ & q_x(x_r - x_{r0})^2 + q_y(y_r - y_{r0})^2 + q_z(z_r - z_{r0})^2.\end{aligned} \tag{40}$$

The track boundaries are described by

$$x_{l,r} = x_C \mp (-s_\theta c_\phi + c_\theta s_\mu s_\phi)\frac{r_{rw}}{2}, \quad y_{l,r} = y_C \mp (c_\theta c_\phi + s_\theta s_\mu s_\phi)\frac{r_{rw}}{2}, \quad z_{l,r} = z_C \mp c_\mu s_\phi \frac{r_{rw}}{2}.$$

The control weights are r_ϕ, r_μ, r_θ, r_{rw}, while q_x, q_y, q_z are weights on the reconstruction errors in (40). The survey data for the right and left boundaries are given by, respectively, x_{r0}, y_{r0}, z_{r0} and x_{l0}, y_{l0}, z_{l0}. The weight selection is used to emphasise, or de-emphasise, the relative importance of the tracking error, the rejection of measurement noise, and the smoothness of the curvature estimates. Excessively 'busy' curvature estimates are likely to cause convergence difficulties in subsequent MLTS calculations.

The estimation dynamic system (2) is described by

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = [c_\theta c_\mu,\ s_\theta c_\mu,\ -s_\mu,\ \phi',\ u_\phi,\ \mu',\ u_\mu,\ \theta',\ u_\theta,\ u_{rw}]^T. \tag{41}$$

In the case of closed race tracks, cyclic boundary conditions must be enforced, *i.e.* $\boldsymbol{x}(s_i) = \boldsymbol{x}(s_f)$ for all the state variable except the yaw angle θ, which must satisfy $\theta(s_i) - \theta(s_f) =$

$\pm 2\pi$—the sign is determined by the running direction. In combination, the cyclicity conditions ensure that the start and end of the track marry up at the start-finish line.

A number of minor variants have been reported in the literature, e.g. the third derivatives of the Euler angles are controlled instead of the second derivatives (in this case three more states must be added to (36)). This variation is required, for example, when the spacial rates of change of the track curvature are required. One might also include a term in (40) that weights the error between the reconstructed and measured centreline. In the more recent formulations, including the curved ribbon [83], for example, the first- or second-order spatial derivatives of the $m(s, n)$ polynomial coefficients are added to the controls (36). While these details are beyond the scope of this survey, the cost function (40) may also require error terms relating to the fitting errors associated with a track-surface mesh; see Figure 12 (b). Examples of application to several tracks have been reported in the literature, these include Barcelona-Catalunya [45, 48], Spa-Francorchamps [1], Adria and Imola [84], Mugello [79, 80] and the Darlington Raceway [83]. In the last example, the NASCAR track is reconstructed using a curved ribbon model, which shows variations in the road camber in the lateral direction of up to 10 deg. This is combined with large absolute camber angles of up to 30 deg.

7. Vehicle positioning

Once a road model has been constructed, one must consider the kinematics of vehicle positioning on the road surface. Referring again to Figure 12 (a), it is clear that the vehicle's position is determined by the distance travelled s, the lateral and vertical displacements n and m respectively, and the yaw of the vehicle relative to the centreline tangent χ. These variables specify the position of the vehicle with respect to the Darboux frame, whose location and orientation are known once s is known; see Sections 5 and 6.

In the case of a flat-ribbon road model these quantities are determined by

$$\dot{s} = \frac{uc_\chi - vs_\chi}{1 - n\Omega_z^0} \qquad \dot{n} = u\sin\chi + v\cos\chi \qquad \dot{\chi} = \omega_z - \Omega_z^0\dot{s} \tag{42}$$

which are derived in several places including [1]. The longitudinal and lateral components of the vehicle's velocity are u and v, respectively, or more precisely, the velocity of the projection of the vehicle's reference point on to the road plane. The angular velocity (rad/s) of the vehicle is ω_z, while Ω_z^0 is the geodesic curvature (rad/m) of the Darboux frame.

When the road prototype is extended to include lateral curvature, these equations include additional terms that relate to road height variations $m(s, n)$ relative to the Darboux frame [83]:

$$\dot{s} = \frac{uc_\chi - vs_\chi}{1 - n\Omega_z^0 + m\Omega_y^0} \qquad \dot{n} = u\sin\chi + v\cos\chi + (m - n\partial_n m)\,\Omega_x^0\dot{s}, \qquad \dot{\chi} = \omega_z - \Omega_z\dot{s}. \tag{43}$$

The ribbon's lateral curvature gives rise to $m(s, n)$-related terms in (43), while the Ω_z^0 term becomes Ω_z, which is the geodesic curvature (rad/m) of the frame tangent to the road plane at the vehicle's reference point (not on the centreline). The relationship between Ω_z^0 and Ω_z is given by (34). The position and orientation of the vehicle are obtained integrating (43), which are appended to the vehicle's equations of motion.

8. Vehicle models

A variety of models have been proposed to describe the dynamics of cars and motorcycles in MLTS, see Section 2. The aim of this section is to highlight the essential features of these models in order to incorporate them in OCP.

Car models (both single- and twin-track variants) include the following states

$$\boldsymbol{x} = [u, v, \omega_z]^T, \tag{44}$$

which represent, respectively, the longitudinal and lateral velocity of a vehicle-fixed reference point, with the vehicle's yaw velocity given by ω_z. These quantities are usually expressed in a vehicle-fixed reference frame. The control inputs include (but are not limited to) the steering angle δ, and the propulsive/braking torque T_w

$$\boldsymbol{u} = [\delta, T_w]^T. \tag{45}$$

The torque T_w is split between the front and rear axles in accordance with the vehicle's drivetrain layout, which might be rear-wheel drive (RWD), front-wheel-drive (FWD), or all-wheel-drive (AWD). The distribution of the braking torque depends on a front-to-rear wheel braking ratio. In 'simple' models one will typically assume: (i) that the drive torque is split symmetrically between the wheels on each axle (*i.e.* an open differential), (ii) symmetric braking torques, (iii) no states associated with wheel spin—tyre forces are applied to ground-contact points that are assumed 'fixed' in the car's body-fixed reference frame, and the propulsive/braking torques being related by the wheel radius (spin inertia effects are neglected), (iv) identical left- and right-wheel steering angles (*i.e.* Ackermann and toe effects neglected), and (v) 'instantaneous' longitudinal and lateral load transfers (suspension and inertial effects are neglected).

Simple motorcycle models include typically the following states

$$\boldsymbol{x} = [u, v, \omega_z, \varphi, \dot{\varphi}, \delta, \dot{\delta}]^T. \tag{46}$$

The first three states are defined as for cars (44), while φ and $\dot{\varphi}$ are required to represent the motorcycle's roll angle and roll rate. Unlike car models, the steering angle δ and steering rate $\dot{\delta}$ are introduced into the state rather than the input vector. This change is required in order to preserve the basic dynamical properties of any two-wheeled vehicle (particularly 'weave', 'castoring' and 'capsize'); see the classic Whipple model in [1]. The bicycle/motorcycle inputs comprise steering and propulsive/braking torques respectively

$$\boldsymbol{u} = [T_h, T_w]^T. \tag{47}$$

The drive torque acts on the rear wheel and reacts on the main frame, while the steer torque T_h acts on the steered assembly and reacts on the main frame. Unlike cars, motorcycles are only conditionally stable—it is self-evident that at very low speed a two-wheeled machine will 'fall over'. Issues relating to 'wobble', 'weave', 'capsize' and conditional stability are discussed in detail in [1]; we refer the interested reader to that source and the many references therein. One often finds the term 'bicycle model' used to refer to single-track car models, but this term is misleading, because the absence of a roll freedom means that car-related bicycle models have nothing to do with bicycles.

Tyres are an important component of any vehicle model. Be it a car or motorcycle tyre, the forces and moments generated on the contact patch are nonlinear functions of the tyre's side-slip angle, slip ratio, camber and normal load—other variables may enter the tyre description in complex tyre models. Another important issue relates to cross-coupling between the longitudinal and lateral forces through such things as weighting functions, friction ellipses, and so on. No matter what model is used, the tyre forces and torque must remain within the limits dictated by the available friction. In most MLTS models the tyre response is instantaneous, especially in the case of cars, while for motorcycles the tyre response lag may be represented by relaxation equations that involve a single state. A number of tyre model variations have been reported in the literature, but they all share these common features. An in-depth discussion of tyre modelling is reported in the classic [62]; an extended discussion is also available in [1].

The equations of motion are usually derived in the time domain, either by pencil-and-paper, or with the aid of a multibody modelling tool, see for example [116–125]. As explained in Section 3 the OCP is usually formulated in the spatial domain, which is achieved by the change of variable $dt = ds/\dot{s}$. The vehicle can thus be described by

$$\boldsymbol{x}' = \boldsymbol{f}(\boldsymbol{x}(s), \boldsymbol{u}(s), s), \tag{48}$$

in which the prime denotes a derivative with respect to s.

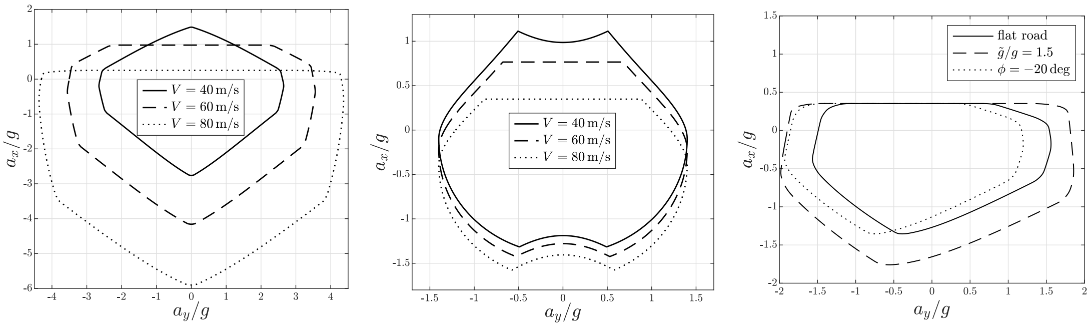
In some applications QSS models are used instead of their dynamic counterparts—dynamic models with QSS components, or approximations, are also widely employed. These include such things as vehicle models with instantaneous load transfers, non-dynamic suspension models, tyre models with an instantaneous response and so on. In these cases the differential equations describing the dynamic system in (2) are replaced by algebraic equations for the corresponding steady-state system

$$\boldsymbol{0} = \boldsymbol{f}(\boldsymbol{x}_{QSS}, \boldsymbol{u}_{QSS}). \tag{49}$$

These equations can be solved for the equilibrium states $\boldsymbol{x}_{QSS}$ and controls $\boldsymbol{u}_{QSS}$ once the vehicle's motion has been determined in terms of speed and lateral acceleration for example. In the case of a basic car model with three states (44), and two controls (45), one could pre-assign the (constant) longitudinal velocity u_{QSS} and the yaw rate $\omega_{z,QSS}$. The resulting algebraic equations (49) can then be solved for the lateral velocity v_{QSS}, steer angle δ_{QSS} and wheel torque $T_{w,QSS}$. The same general approach can be used for motorcycle models.

In a more general setting, longitudinal acceleration (at constant speed) can be emulated by including apparent inertial forces. In this case one obtains a quasi-steady-state solution rather than a steady-state solution—regardless of SS or QSS conditions, the vehicles state is determined by a set of (usually nonlinear) algebraic equations. The performance envelope of the vehicle is determined by QSS calculations that can be presented graphically as g-g diagrams, which are plots of the vehicles normalized longitudinal acceleration limit (at a particular speed) versus the normalized lateral acceleration at the same speed. In practice, (49) are solved iteratively until a physical limit of the vehicle is reached—limits include the maximum engine power, a tyres' friction limit, the rollover limit of a car, or the 'wheelie' (front tyre lift) or 'stoppie' (rear tyre lift) limit of a motorcycle. Figure 13 shows the performance envelope of three vehicles, obtained using the semi-analytical vehicle models given in [82].

Figure 13 (a) shows the g-g diagram of a prototypical Formula One car at different speeds. At low speeds (solid line) the acceleration limits are dictated by tyre friction. As the speed increases (dashed and dotted lines), the maximum acceleration is limited by the engine power (the top horizontal lines in the g-g diagram), while the maximum



(a) (b) (c)

Figure 13. (a) g-g diagrams for a formula car, (b) a racing motorcycle, both at different speeds. (c) is for a NASCAR at 70 m/s at different camber angles and apparent gravity.

available deceleration increases with speed due to increased aerodynamic drag and down force. The g-g diagram also 'widen out' at higher speeds, because increased downforce facilitate increases in the maximum available lateral acceleration.

Figure 13 (b) shows the g-g diagram of a prototypical MotoGP bike at different speeds. At low speeds (solid line), the acceleration and deceleration limits have a quite different shape as compared with cars. At lows speeds the maximum acceleration is limited by the onset of a 'wheelie', while the deceleration limit is determined by the onset of a 'stoppie'—these limits appear as concave arcs on top and bottom of the g-g diagram. The lateral accelerations are still limited by the tyre friction. As the speed increases, the maximum acceleration is limited by engine power, while more braking capacity is available due to aerodynamic drag. The lateral acceleration limits are largely independent of speed, because there are no significant speed-dependent aerodynamic effects at play in this example.

Figure 13 (c) show the g-g diagram of a Nascar car. Most NASCAR ovals comprise only left turns and so the tyres, suspension, and aerodynamics are set up to favour left-handed cornering. This asymmetry is evident at all speeds. The dashed line represents the car's g-g characteristics on a road with a camber of 20 deg (in the direction of turning), which represents another reason for the asymmetry, and increases the maximum lateral acceleration capacity for left-handed cornering. The dot-dash line represents the acceleration envelope when the apparent gravity (due to centrifugal acceleration and slope variations) is increased by 50 %—as one would expect the whole g-g map enlarges.

This last example suggests conceptually that the standard g-g-V diagram should be extended to a more general g-g-g-V diagram, which is indeed one of the approaches to deal with effects related to road three-dimensionality in MLTS with QSS models.

9. The MLTS problem

In this section we summarizes the steps needed to solve the MLTS problem.

The first step in the solution of a MLTS is to survey the track and construct a track model using the processes outlined in Sections 5 and 6. At the end of this step one will have available the track curvature and track-width parameters, which will be required in the construction of inequality constraints that confine the car to the road.

Supposing one wants to build a dynamic simulation without a pre-defined race line, the next step is to construct a dynamic vehicle model (2). This model provides additional constraints that describe how the system states and controls are related to each other. There are two elements to this: (i) the vehicle's equations of motion (48), and (ii) the kinematic differential equations related to the vehicle positioning (43). In combination, these equations will allow the optimal control solution algorithm to drive the car around the track and seeks out an optimal control strategy. The propulsive and steering controls will produce lateral, longitudinal and yaw velocity profiles consistent with the vehicle's equations of motion. These profiles, using (42), will 'drag' the Darboux frame along the track centre line. In this way the vehicle drives along the road plane of a moving coordinate system which pitches, rolls and turns with the track (as described by the track model).

The OCP controls are usually the rates of change of the physical controls required by the vehicle model. For example, the steering input might be $u_\delta = \dot{\delta}$, or a steering torque version of it (in the case of motorcycles). The introduction of these additional integrators in the vehicle model serve two main purposes, namely (i) it tends to improve the numerical behaviour of the MLTS, and (ii) it allows one to introduce rate constraints on the controls. The input vectors for simple car and motorcycle models are

$$\boldsymbol{u} = [u_\delta, u_{T_w}]^T \qquad \text{and} \qquad \boldsymbol{u} = [u_{T_h}, u_{T_w}]^T, \tag{50}$$

respectively. The state vectors are typically

$$\boldsymbol{x} = [n, \chi, \delta, T_w, u, v, \omega_z]^T \qquad \text{and} \quad \boldsymbol{x} = [n, \chi, T_h, T_w, u, v, \omega_z, \varphi, \varphi', \delta, \delta']^T \tag{51}$$

for cars and motorcycle respectively. The first two states are governed by the second and third equations in (43). The third and fourth states come from integrating the inputs as described above. The remaining equations, *i.e.* $f_5 \cdots f_7$ for cars and $f_5 \cdots f_{11}$ for bikes, come from the equations of motion used to represent the vehicle's dynamics.

For MLTS the cost function (3) is the minimum lap time, which is (4), or (5), plus addition regularization terms should they be required; see Section 4.7. If regularization terms are introduced, the 'true' lap time can be computed separately introducing an auxiliary 'time-measuring' state.

The controls should be limited to reflect the neuro-muscular capabilities of a human driver/rider. These limits may include, for example, the steering torque (say 100-150 Nm) and the rate of change of steering angle. One may also want to feed the controls through low-pass filters with a bandwidth of few Hertz. There will also be limits on the steering travel, or suspension travel, in more advanced models. The driving, braking and lateral capabilities of the tyre should take care of themselves if the tyre model is appropriately parametrised. The engine power should be enforced using constraints such as

$$F_x u \leq P_{\max} \qquad \text{or} \qquad T_w \omega_r \leq P_{\max}, \tag{52}$$

where $P_{\max}$ is the mechanical engine power (at the road in the former expression and at wheels in the latter), F_x is the longitudinal drive force, u is the vehicle's speed, ω_r is the angular velocity of the drive wheels. Note that $F_x u \leq T_w \omega_r$, since the former expression does not recognise power losses due to longitudinal tyre slip and spin inertial effects.

In order to keep the vehicle on the track, one would include the constraint

$$|n| \leq \frac{r_w}{2} - \frac{t_w}{2}, \tag{53}$$

where r_w is the road width, and t_w is the vehicle's track width. If one wants to take account of vehicle yawing, or differing front and rear track widths, four more complicated constraints are required—one for each wheel ground-contact point.

Particularly in the case of motorcycles, one may include 'no wheel lift-off' constraints. One can get wheel lift-off in cars, for example, if one 'over cooks' a corner entry, but far more likely with powerful motorcycles are 'wheelies' and 'stoppies', both of which could result in numerical problems during the solution of MLTS.

Implicit in free-trajectory MLTS, is the idea that the optimal control algorithm will automatically 'seek out' the optimal racing line—in this problem configuration one seeks to minimise the lap time without violating any vehicle or track-related constraints. In simpler problems, the racing line may be pre-assigned as another constraint set.

One may choose to carry out a MLTS using a QSS model instead of a dynamic model (or partially dynamic model). In this case g-g diagrams are often used to impose limits on the vehicle's performance. When the trajectory is predefined—the *fixed-trajectory* problem—the problem becomes one of computing the speed profile that minimises the lap time while remaining within the limits imposed by the g-g diagrams. This approach has the advantage that, in principle, the g-g diagram could be measured and no vehicle modelling exercise is required. In these simulations the classical apex-finding method is usually employed (although an OCP approach is also possible), which begins with the identification of the apexes in the pre-defined trajectory. At each apex a minimum speed and maximum lateral acceleration is assumed. Braking comes before the apex, and acceleration after the bend; see discussions in Section 2.

The third possibility is to use a QSS vehicle representation combined with a free trajectory that will be determined by the associated OCP solution. In this case the MLTS is again solved as an optimal control problem using the cost functions (4), or (5). The optimal controls are the longitudinal and lateral accelerations

$$\boldsymbol{u} = [a_x, a_y]^T, \tag{54}$$

with the equations of motion replaced by acceleration constraints, which can be enforced using a polar representation of the g-g diagrams. That is

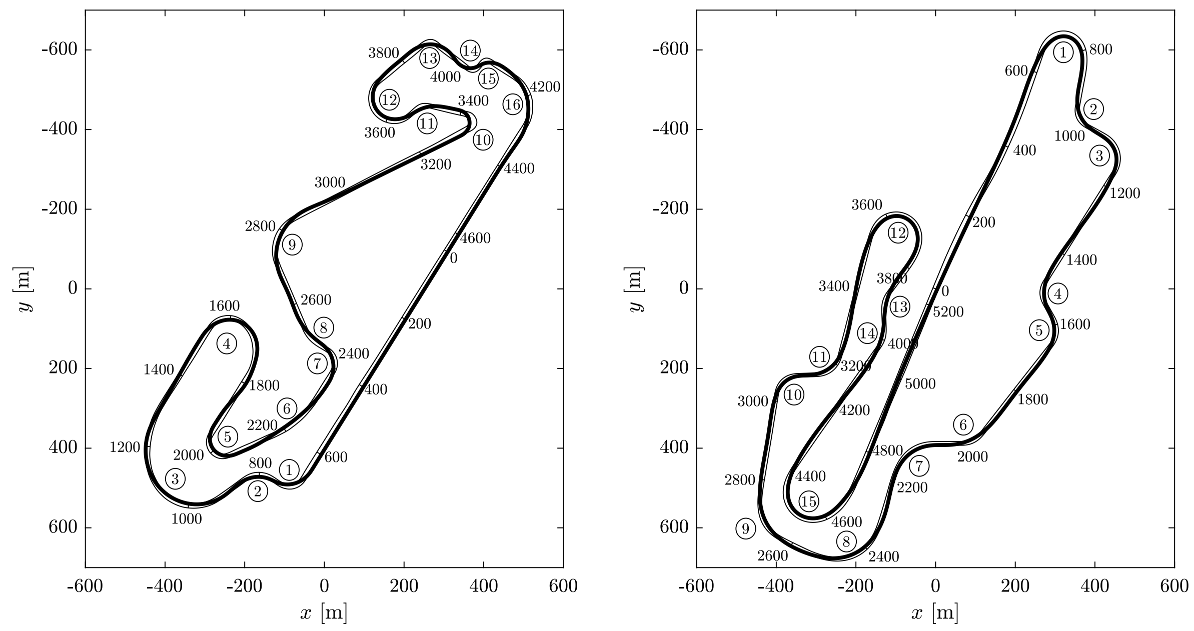
$$\tilde{\rho} \leq \rho_{\max}(\tilde{\alpha}, V, \tilde{g}), \tag{55}$$

where

$$\tilde{\rho} = \frac{1}{g}\sqrt{\tilde{a}_x^2 + \tilde{a}_y^2} \qquad \text{and} \qquad \tilde{\alpha} = \arctan 2(\tilde{a}_x, \tilde{a}_y), \tag{56}$$

while $\tilde{a}_x, \tilde{a}_y, \tilde{g}$ are the apparent longitudinal, lateral and vertical accelerations on the vehicle and $\rho_{\max}$ is the boundary of the g-g diagram. The dynamic equations of the OCP reduces to the second and third equations in (43), together with $\dot{V} = a_x$, with all the other vehicle-model-related complexities encoded in the g-g-V or g-g-g-V diagrams.

We conclude with some clarifying comments: (i) elevation changes and banking will reflect on the vehicle's performance through the $\tilde{g}$ parameter; (ii) in more complex situations the influence of track camber must be recognised that may change laterally [83]; (iii) considerations such as fuel consumption constraints will require additional dynamic equations; (iv) to obtain smoother solutions, the derivatives of the accelerations (*i.e.* jerks) may be used as control inputs—in this case the input vector becomes $\boldsymbol{u} = [j_x, j_y]^T$ with the additional equations $\dot{a}_x = j_x$ and $\dot{a}_y = j_y$ required; (v) fixed-trajectory method is easily recovered by making the track width arbitrarily small ($|n| \leq \epsilon$ with ϵ small)—in this case the track becomes the racing line.



(a)

(b)

Figure 14. (a) Car on Barcelona-Catalunya and (b) motorcycle on Mugello, from [82].

10. Examples of application

MLTS on three-dimensional track models of the Barcelona-Catalunya and Mugello circuits are shown in Figure 14. The (a) part of the figure was obtained using g-g diagrams of a car (Figure 13 (a)) on the Barcelona track, while the (b) part is for a motorcycle (Figure 13 (b)) on Mugello. The free-trajectory QSS methodology discussed in Section 9 was employed. The results from these calculations include the optimal trajectories, and the associated speed profiles. All the other vehicle states can be found by back calculation; each simulation point correspond to a point of the corresponding g-g diagram. The MLTS was solved using a direct approach, with the original continuous-time problem discretised and solved as an NLP using IPOPT [47]. The total number of equations being solved simultaneously was approximatively 12,000, with mesh points spaced between 0.2 m (on tight corners) and 25 m (on straights) along the centreline. The controls were longitudinal and lateral 'jerks', *i.e* the time rates of change of acceleration. Three-dimensional effects result in a change of approximatively 1 s in the lap time for the car on the Barcelona circuit (the flat circuit being faster), and a change of approximatively 4 s for the motorcycle on the Mugello circuit (the flat circuit being slower in this case). We refer the reader to [82] for further details.

11. Conclusion

An historical overview of advances in the state-of-the-art of minimum-lap-time simulations of road vehicles is provided, with a commentary on the most widely used modelling and solution strategies included. Most approaches use either quasi-steady-state vehicle models with pre-assigned trajectories, or dynamic vehicle models with free trajectories found by optimization. A combination of quasi-steady-state models, with free trajectories, has been proposed and researched recently. Most of the studies reported in the literature use two-dimensional road models. More recently, three-dimensional road models based on geometric ribbons have been developed—these models are laterally flat. In another recent devolvement, road models with a laterally varying camber have been proposed. The use of the ‘elapsed distance’ as the independent variable is now ubiquitous, as are curvilinear coordinates for track and vehicle-positioning descriptions. The optimal control problem resulting from the minimum-lap-time problem is usually solved with a direct approach based on nonlinear programming. That said, several studies using indirect methods (based on first-order necessary conditions) have also been reported. The most commonly employed solution techniques are based on collocation (both direct and indirect), with a number of studies employing multiple-shooting also reported.

References

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

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

[3] Roland RD, Thelin CF. Computer simulation of Watkins Glen grand prix circuit performance. Cornell Aeronautical Laboratory; 1971. Report No.: ZL-5002-K-1.

[4] Roy S, Alianello DA. A driver characterizing function-the g-g diagram. Cornell Aeronautical Laboratory; 1970. Report No.: VJ-2882-K.

[5] Rice RS. Measuring car-driver interaction with the g-g diagram. SAE. 1973;(730018).

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

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

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

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

[10] Hatwal H, Mikulcik EC. Some inverse solutions to an automobile path-tracking problem with input control of steering and brakes. Vehicle System Dynamics. 1986;15(2):61–71.

[11] Fujioka T, Kimura T. Numerical simulation of minimum-time cornering behavior. JSAE review. 1992;13(1):44–51.

[12] Wu A, Miele A. Sequential conjugate gradient-restoration algorithm for optimal control problems with non-differential constraints and general boundary conditions, part i. Optimal Control Applications and Methods. 1980;1(1):69–88.

[13] Wu A, Miele A. Sequential conjugate gradient-restoration algorithm for optimal control problems with non-differential constraints and general boundary conditions, part ii. Optimal Control Applications and Methods. 1980;2(1):119–130.

[14] Hendrikx JPM, Meijlink TJJ, Kriens RFC. Application of Optimal Control Theory to Inverse Simulation of Car Handling. Vehicle System Dynamics. 1996;26(6):449–61.

[15] Kirk DE. Optimal control theory: An introduction. Englewood Cliffs, NJ: Prentice-Hall; 1970.

[16] Cossalter V, Da Lio M, Lot R, Fabbri L. A General Method for the Evaluation of Vehicle Manoeuvrability with Special Emphasis on Motorcycles. Vehicle System Dynamics. 1999;31(2):113–35.

[17] Press WH, Teukolsky SA, Vetterling WT, Flannery BP. Numerical recipes: The art of scientific computing. 3rd ed. Cambridge University Press; 2007.

[18] Casanova D, Sharp RS, Symonds P. Minimum Time Manoeuvring: The Significance of Yaw Inertia. Vehicle System Dynamics. 2000;34(2):77–115.

[19] Casanova D, Sharp RS, Symonds P. Sensitivity to mass variations of the fastest possible lap of a formula one car. Vehicle system dynamics. 2001;35(2):119–134.

[20] Gill PE, W Murray W, Saunders MA. SNOPT: An SQP Algorith for Large-Scale Constrained Optimization. SIAM Review. 2005;47(1):99–131.

[21] Casanova D, Sharp RS, Symonds P. On the optimisation of the longitudinal location of the mass centre of a formula one car for two circuits. In: Proceedings of AVEC; Vol. 2; 2002. p. 6–12.

[22] Christianson B, Casanova D, Sharp RS, Symonds P. Application of automatic differentiation to race car performance optimisation. Automatic Differentiation of Algorithms. 2002;.

[23] Bertolazzi E, Biral F, Da Lio M. Symbolic-numeric indirect method for solving optimal control problems for large multibody systems. Multibody Syst Dyn. 2005;(13):233–252.

[24] Bertolazzi E, Biral FB, Da Lio M. Symbolic-Numeric Efficient Solution of Optimal Control Problems for Multibody Systems. Journal of computational and applied mathematics. 2006;185(2):404–21.

[25] Bertolazzi E, Biral F, Da Lio M. Real-Time Motion Planning for Multibody Systems. Multibody System Dynamics. 2007;17:119–39.

[26] Bobbo S, Cossalter V, Massaro M, Peretto M. Application of the Optimal Maneuver Method for Enhancing Racing Motorcycle Performance. SAE International Journal of Passenger Cars—Mechanical Systems. 2009;1(1):1311–18.

[27] Bertolazzi E, Biral F, Da Lio M, Cossalter V. The influence of rider's upper body motions on motorcycle minimum time maneuvering. In: Proceedings of the ECCOMAS Thematic Conference, Multibody Dynamics; Milan, Italy; 2007.

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

[29] Kelly DP, Sharp RS. Time-optimal control of the race car: a numerical method to emulate the ideal driver. Vehicle System Dynamics. 2010;48(12):1461–1474.

[30] Zhou J, Tits A, Lawrence C. Users guide for FSQP version 3.0: A fortran code for solving nonlinear (minimax) optimization problems, generating iterates satisfying all inequality and linear constraints. Institute for Systems Research, University of Maryland, College Park, MD.; 1997. Report No.: TR 92-107.

[31] Sharp RS, Peng H. Vehicle dynamics applications of optimal control theory. Vehicle System Dynamics. 2011;49(7):1073–1111.

[32] Kehrle F, Frasch JV, Kirches C, Sager S. Optimal control of formula 1 race cars in a vdrift based virtual environment. IFAC Proceedings Volumes. 2011;44(1):11907 – 11912; 18th IFAC World Congress.

[33] Bock H, Plitt K. A multiple shooting algorithm for direct solution of optimal control problems. IFAC Proceedings Volumes. 1984;17(2):1603 – 1608; 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984.

[34] Leineweber DB, Schfer A, Bock HG, Schlder JP. An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization (part i and part ii). Computers & Chemical Engineering. 2003;27(2):157 – 174.

[35] Gerdts M. Solving mixed-integer optimal control problems by branch&bound: a case study from automobile test-driving with gear shift. Optimal Control Applications and Methods. 2005;26(1):1–18.

[36] Sager S, Bock HG, Reinelt G. Direct methods with maximal lower bound for mixed-integer optimal control problems. Mathematical Programming. 2009;118(1):109–149.

[37] Hoffmann C, Kirches C, Potschka A, Sager S, Wirsching L, Diehl M, Leineweber DB, Sch afer AAS. Muscod-ii users manual. University of Heidelberg. 2011;.

[38] Timings JP, Cole DJ. Vehicle trajectory linearisation to enable efficient optimisation of the constant speed racing line. Vehicle System Dynamics. 2012;50(6):883–901.

[39] Timings J, Cole DJ. Minimum Maneuver Time Calculation Using Convex Optimization. Journal of Dynamic Systems, Measurement, and Control. 2013;135(3).

[40] Timings J, Cole DJ. Robust Lap-Time Simulation. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 2014;228(10):1200–16.

[41] Anderson J, Ayalew B. Modelling minimum-time manoeuvering with global optimisation of local receding horizon control. Vehicle System Dynamics. 2018;56(10):1508–1531.

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

[43] 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;6(1).

[44] Lot R, Evangelou SA. Lap Time Optimization of a Sports Series Hybrid Electric Vehicle. In: Proceedings of the World Congress on Engineering; Vol. III. London; 2013.

[45] Perantoni G, Limebeer DJN. Optimal Control of a Formula One Car with Variable Parameters. Vehicle Systems Dynamics. 2014;52(5):653–78.

[46] Van Wyk EJ, Falugi P, Kerrigan EC. Imperial college london optimal control software user guide (iclocs). 2010; Available from: http://www.ee.ic.ac.uk/ICLOCS.

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

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

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

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

[51] Limebeer DJN, Perantoni G, Rao AV. Optimal Control of Formula One Car Energy Recovery Systems. International Journal of Control. 2014;87(10):2065–80.

[52] Fédération Internationale de l'Automobile (FIA). 2014 Formula One Technical Regulations; 2014. Report no.:.

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

[54] Dal Bianco N, Lot R, Gadola M. Minimum time optimal control simulation of a gp2 race car. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 2018;232(9):1180–1195.

[55] Lot R, Dal Bianco N. Lap time optimisation of a racing go-kart. Vehicle System Dynamics. 2016; 54(2):210–230.

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

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

[58] Dal Bianco N, Lot R, Matthys K. Lap time simulation and design optimisation of a brushed dc electric motorcycle for the isle of man tt zero challenge. Vehicle System Dynamics. 2018;56(1):27–54.

[59] Veneri M, Massaro M. The effect of ackermann steering on the performance of race cars. Vehicle System Dynamics. 2020;0(0):1–21.

[60] Pease G, Limebeer DJN, Fussey P. Fuel consumption minimization, with emissions constraints, for diesel powered cars. IEEE Transactions on Control System Technology. 2020;28(4):1243–1257.

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

[62] Pacejka HB. Tire and vehicle dynamics. 3rd ed. Oxford: Butterworth Heinemann; 2012.

[63] Williams M, Landel R, Ferry J. The Temperature Dependence of Relaxation Mechanisms in Amorphous Polymers and Other Glass-Forming Liquids. Journal of the Americal Chemical Society. 1955;77(14):3701–7.

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

[65] Tremlett AJ, Limebeer DJN. Optimal Tyre Usage for a Formula One Car. Vehicle System Dynamics. 2016;54(10):1448–73.

[66] Tremlett A, Massaro M, Purdy D, Velenis E, Assadian F, Moore A, Halley M. Optimal Control of Motorsport Differentials. Vehicle System Dynamics. 2015;53(12):1772–94.

[67] Farroni F, Sakhnevych A, Timpone F. Physical modelling of tire wear for the analysis of the influence of thermal and frictional effects on vehicle performance. Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications. 2017;231(1-2):151–161.

[68] Capra D, Farroni F, Sakhnevych A, Salvato G, Sorrentino A, Timpone F. On the implementation of an innovative temperature-sensitive version of pacejkas mf in vehicle dynamics simulations. Lecture Notes in Mechanical Engineering. 2019;:1084–1092.

[69] De Castro R, Tanelli M, Arajo RE, Savaresi SM. Minimum-time manoeuvring in electric vehicles with four wheel-individual-motors. Vehicle System Dynamics. 2014;52(6):824–846.

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

[71] 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 Transactions on Mathematical Software. 2017;44(2).

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

[73] Dunning I, Huchette J, Lubin M. Jump: A modeling language for mathematical optimization. SIAM Review. 2017;59(2):295–320.

[74] Bezanson J, Edelman A, Karpinski S, Shah V. Julia: A fresh approach to numerical computing. SIAM Review. 2017;59(1):65–98.

[75] Revels J, Lubin M, Papamarkou T. Forward-mode automatic differentiation in julia. ArXiv. 2016;abs/1607.07892.

[76] Sedlacek T, Odenthal D, Wollherr D. Minimum-time optimal control for vehicles with active rear-axle steering, transfer case and variable parameters. Vehicle System Dynamics. 2020;0(0):1–29.

[77] Christ F, Wischnewski A, Heilmeier A, Lohmann B. Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients. Vehicle System Dynamics. 2019;0(0):1–25.

[78] Andersson JAE, Gillis J, Horn G, Rawlings JB, Diehl M. CasADi: a software framework for nonlinear optimization and optimal control. Math Prog Comp. 2019;:1–36.

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

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

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

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

[83] Lovato S, Massaro M, Limebeer DJN. Curved-ribbon-based track modelling for minimum lap-time optimisation. Meccanica. 2021;Under Review.

[84] Dal Bianco N, Biral F, Bertolazzi E, Massaro M. Comparison of Direct and Indirect Methods for Minimum Lap Time Optimal Control Problems. Vehicle System Dynamics. 2018;.

[85] Biral F, Bertolazzi E, Bosetti P. Notes on Numerical Methods for Solving Optimal Control Problems. IEEJ Journal of Industry Applications. 2016;5(2):154–66.

[86] Bryson AE, Ho YC. Applied optimal control: Optimization, estimation, and control. New York: Hemisphere Publishing; 1975.

[87] Bryson AE. Dynamic optimization. Addison-Wesley; 1999.

[88] Betts JT. Practical Methods for Optimal Control and Estimation using Nonlinear Programming. 2nd ed. Philadelphia, PA: SIAM; 2001.

[89] Athans MA, Falb PL. Optimal control: An introduction to the theory and its applications. Mineola, NY: Dover; 2006.

[90] Karush W. Minima of Functions of Several Variables with Inequalities as Side Constraints [master's thesis]. Dept. of Mathematics, Univ. of Chicago; 1939.

[91] Kuhn HW, Tucker AW. Nonlinear Programming. In: Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press; 1951.

[92] Kjeldsen TH. A Contextualized Historical Analysis of the Kuhn–Tucker Theorem in Nonlinear Programming: The Impact of World War II. Historia Mathematica. 2000;27:331–61.

[93] Garg D, Patterson MA, Hager WW, Rao AV, Benson DA, Huntington GT. A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods. Automatica. 2010;46(11):1843–51.

[94] Byrd RH, Nocedal J, Waltz RA. Large-scale nonlinear optimization. Boston, MA: Springer; 2006. Chapter Knitro: An Integrated Package for Nonlinear Optimization; p. 35–59.

[95] Lanczos C. Applied analysis. Prentice-Hall; 1956.

[96] Paris S, Riehl J, Sjauw W. Enhanced procedures for direct trajectory optimization using nonlinear programming and implicit integration. Vol. 2; 2006. p. 852–870.

[97] Abramowitz M, Stegun I. Handbook of mathematical functions with formulas, graphs, and mathematical tables. Mineola, NY: Dover; 1965.

[98] Agamawi YM, Rao AV. Cgpops: A c++ software for solving multiple-phase optimal control problems using adaptive gaussian quadrature collocation and sparse nonlinear programming. ACM

Trans Math Softw. 2020 Jul;46(3); Available from: https://doi.org/10.1145/3390463.

[99] Imperial College London. iclocs2 user guide. 2018.

[100] Falck R, Gray J. Optimal control within the context of multidisciplinary design, analysis, and optimization. AIAA Scitech 2019 Forum. 2019;.

[101] Gray J, Hwang J, Martins J, Moore K, Naylor B. Openmdao: an open-source framework for multidisciplinary design, analysis, and optimization. Structural and Multidisciplinary Optimization. 2019;59(4):1075–1104.

[102] Garg D, Patterson MA, Darby CL, Francolin C, Huntington GT, Hager WW, Rao AV. Direct Trajectory Optimization and Costate Estimation of Finite-Horizon and Infinite-Horizon Optimal Control Problems via a Radau Pseudospectral Method. Computational Optimization and Applications. 2011;49(2):335–58.

[103] Francolin CC, Hager WW, Rao AV. Costate Approximation in Optimal Control Using Integral Gaussian Quadrature Collocation Methods. Optimal Control Applications and Methods. 2014; 36:38197.

[104] Rao AV. A Survey of Numerical Methods for Optimal Control. Advances in the Astronautical Sciences. 2009;135(1):497–528.

[105] Patterson MA, Hager WW, Rao AV. A *ph* Mesh Refinement Method for Optimal Control. Optimal Control Applications and Methods. 2014;36:398–421.

[106] Liu F, Hager WW, Rao AV. Adaptive mesh refinement method for optimal control using nonsmoothness detection and mesh size reduction. J Frankl Inst. 2015;352:4081–4106.

[107] Liu F, Hager WW, Rao AV. Adaptive mesh refinement method for optimal control using decay rates of legendre polynomial coefficients. IEEE Transactions on Control Systems Technology. 2018; 26(4):1475–1483.

[108] Squire W, Trapp G. Using Complex Variables to Estimate Derivatives of Real Functions. SIAM Review. 1998;40(1):110–2.

[109] Martins JRRA, Sturdza P, Alonso JJ. The Connection Between the Complex-Step Derivative Approximation and Algorithmic Differentiation. AIAA paper. 2001;(0921).

[110] Martins JRRA, Sturdza P, Alonso JJ. The Complex-Step Derivative Approximation. ACM Transactions on Mathematical Software. 2003;29(3):24562.

[111] Lantoine G, Russell RP, Dargent T. Using Multicomplex Variables for Automatic Computation of High-Order Derivatives. ACM Transactions on Mathematical Software. 2012;38(3).

[112] Griewank A, Walther A. Evaluating derivatives: Principles and techniques of algorithmic differentiation. Philadelphia, PA: SIAM; 2008.

[113] Houska B, Ferreau H, Diehl M. ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization. Optimal Control Applications and Methods. 2011;32(3):298–312.

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

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

[116] Rosenthal D. Triangulization of equations of motion for robotic systems. J Guid Control Dyn. 1988;11(3).

[117] Schiehlen, editor. Multibody systems handbook. Springer; 1990.

[118] Sayers M. Symbolic computer language for multibody systems. J Guid Control Dyn. 1991; 14(6):11531163.

[119] Sayers M. Autosim. Vehicle System Dynamics. 1993;22:53–56.

[120] Shi P, McPhee J. Symbolic programming of a graph-theoretic approach to flexible multibody dynamics. Mech Struct Mach. 2002;30(1).

[121] Samin J, Fisette P. Symbolic modeling of multibody systems. Springer; 2003.

[122] Tiller M. Introduction to physical modeling with modelica. Springer; 2001.

[123] Docquier N, Poncelet A, Fisette P. Robotran: A powerful symbolic gnerator of multibody models. Mechanical Sciences. 2013;4(1):199–219.

[124] Kurz T, Eberhard P, Henninger C, Schiehlen W. From neweul to neweul-m2: Symbolical equations of motion for multibody system analysis and synthesis. Multibody System Dynamics. 2010; 24(1):25–41.

[125] Lot R, Massaro M. A Symbolic Approach to the Multibody Modeling of Road Vehicles. International Journal of Applied Mechanics. 2017;9(5).