UNIVERSITÀ
DEGLI STUDI
DI PADOVA

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

CORSO DI DOTTORATO IN INGEGNERIA INDUSTRIALE

CURRICULUM IN INGEGNERIA MECCANICA

Minimum-lap-time of race vehicles

Candidate:
Matteo VENERI

Supervisor:
Ch.mo Prof. Matteo MASSARO

Student ID:
1145132

Co-supervisor:
Ch.mo Prof. Vittore COSSALTER

Cohort:
XXXII

Coordinator:
Ch.mo Prof. Paolo COLOMBO

September, 30, 2019

Alla mia famiglia.

Abstract

This work deals with the numerical-optimisation methods employed for assessing the minimum-lap-time of race vehicles. A particular emphasis is given to the simulations based on the solution of an optimal-control problem.

First, a detailed review is devoted to the optimal-control theory, including direct and indirect methods, together with explicit and implicit formulations.

Three vehicle models are proposed for the implementation in steady-state and dynamic simulations. A steady-state double-track race-car model is presented first, implementing aerodynamic forces, Pacejka-based tyre model, fixed brake-ratio and steady-state lateral-load transfer. This model is then extended including the transient effects for the implementation in a dynamic minimum-time simulation. An essential but comprehensive steady-state motorcycle model is also presented, retaining the peculiar aspects of the motorcycle dynamics, such as the wheelie and stoppie conditions, while assuming the optimal braking-bias.

Then, an optimisation program is developed for computing the race car g-g envelope. A different approach is shown for the steady-state motorcycle model, which is employed for computing the g-g diagrams analytically.

Finally, the most widespread minimum-lap-time simulations are discussed and developed for the presented models, including a quasi-steady-state fixed-trajectory apex-finding approach and a quasi-steady-state fixed-trajectory optimal-control simulation. In addition, a novel approach that combines a steady-state vehicle model with the trajectory optimisation is presented and compared to the discussed simulations. This approach builds upon the formulation of an optimal-control problem and employs the g-g diagrams for limiting the vehicle performance. The results are employed for a sensitivity analysis, that underlines the effect of different parameters on the resulting race line. A dynamic race car model is also implemented in a free-trajectory optimal-control simulation. This program is employed for analysing the effect of different steering geometries on the vehicle performance in different test manoeuvres and during a track lap.

Contents

Chapter 1

Introduction

> On a given day, a given circumstance, you think you have a limit. And you then go for this limit and you touch this limit, and you think, 'Okay, this is the limit'. As soon as you touch this limit, something happens and you suddenly can go a little bit further. With your mind power, your determination, your instinct, and the experience as well, you can fly very high. Ayrton Senna, 1991.

In the motor-racing world, the research of the ultimate limit of the vehicle has always led the work of drivers (or riders) and engineers. When Ayrton Senna touches the walls of the tight roads of Monaco for squeezing every millisecond from his McLaren MP4/6, the limit hides a subjective meaning, a mixture of bravery, instinct and driving-skills that becomes tangible only when it is overtaken. Despite of that, an a-priori estimation of the limit is fundamental on the track, both for helping the driver to achieve the best performance and for obtaining the optimal strategy for the race. Moreover, the knowledge of the limit allows to optimise the performance of the vehicle also during the design stage. In this engineering framework, the limit assumes an objective meaning and its estimation is strongly connected to the concepts of simulation and optimisation. More specifically, simulating the vehicle at the limit leads to the optimisation of the action of a virtual driver, in order to minimise the time needed to complete a lap on a given track, i.e. solving the so-called minimum-lap-time problem. Typically, minimum-lap-time problems are faced employing offline simulations, which allow to solve the problem without real-time constraints, since both the vehicle and the driver are virtual and no cueing of a physical system is needed. For this reason, they can include complex vehicle models, capable of a realistic representation of the physical system, together with complex driver models that allow to optimise the control strategy of the vehicle. Instead, the online simulations, such as the driver-in-the-loop simulators, are performed in real-time, i.e. the model need be solved at the same time in which the real driver applies controls to the physical simulation system. This strategy limits the complexity of the models, since a reliable convergence time of the simulation need be maintained. In this case, the driver is a fundamental part of the simulation and a non-portable complex mechanical system is employed, in order to emulate the real vehicle.

There are a variety of minimum-lap-time problems: they may have a free or predetermined trajectory and may use dynamic or quasi-steady-state models. In the case of free-trajectory problems, the race line is determined by the optimisation solver, while for fixed-trajectory problems, the race line is determined from track data or from previous free-trajectory solutions. As a general rule, fixed-trajectory problems are faster to solve, since a lower number of variables is involved. The first attempts in computing the minimum-lap-time date back in the late 1950s, when basic simulations have been implemented to predict the performance of a Formula One vehicle employing heuristic methods [1]. In the late 1980s minimum-time simulations of sections of the Paul Ricard circuit were presented, employing quasi-steady-state model for computing the steady-state optimisation of the vehicle performance in each single point of the trajectory [2]. One of the first applications of optimal-control techniques for minimum-lap-time are presented in [3] (1996). A basic double-track model is employed for computing the maximum performance of a race car in simple manoeuvres. In [4] (1999) a simple motorcycle model is used together with an indirect optimal-control method for assessing the maximum performance and the optimal race line on the Mugello circuit. Nowadays, the most widespread combinations reported in the literature to solve minimum-lap-time problems are basically two: quasi-steady-state (QSS) models with predetermined trajectory and dynamic models with free-trajectory. The former approaches usually involve the computation of the well-known g-g map of the vehicle at different speeds (g-g-speed surface), together with the determination of the corner apexes on the trajectory (i.e. the points at which the vehicle achieves the maximum lateral acceleration). The latter approaches usually involve the solution of a nonlinear-optimal-control problem (OCP), which may be solved either with direct or indirect methods.

In this thesis, a detailed overview on the available minimum-lap-time methods is carried out, together with a comprehensive description of the optimisation methods and on the modelling approaches employed for assessing the minimum-lap-time. Different simulation programs are developed and compared, in order to stress the advantages and drawbacks in terms of computational effort, consistency with the real behaviour of the vehicle and complexity of implementation. In this framework, three main contributions of this work should be underlined. These studies mainly refer to the research work presented in [5] and [6]. The first contribution consists in a novel method for minimum-time simulations that combines the quasi-steady-state vehicle modelling with the trajectory optimisation, using optimal-control techniques. The method builds upon the g-g diagrams, which are commonly computed to analyse the vehicle performance. Numerical vehicle models of very high complexity can be employed, since all their features (e.g. tyre limits, power limits, aerodynamic drag and downforce, suspensions, etc.) are summarised by the related g-g diagrams, and do not affect the complexity of the OCP that needs to be solved. The method allows to employ even experimental g-g diagrams in place of numerical ones and is suitable for application to both cars and motorcycles. A second contribution can be found in the implementation of an essential motorcycle model in the quasi-steady-state simulations. The model retains some of the peculiar aspects of the motorcycle dynamics, such as the wheelie and

stoppie limits. The aerodynamic drag force is included, limiting the maximum acceleration achievable, while the optimal braking-ratio is employed for estimating the maximum deceleration performance. Due to its simplicity, the model requires a relatively small computational effort, providing a rapid tool for assessing a basic description of the vehicle performance. The third contribution aims to employ a dynamic simulation for assessing the performance of an FSAE race car and analyse the effect of steering geometry on the minimum-lap-time. This study is devoted to the implementation and the validation of a transient vehicle model for free-trajectory optimal-control simulations. The lap-time is computed in three manoeuvres (steady-turning, slalom and track lap), while the baseline FSAE steering system is compared against the Ackermann and parallel geometries.

The thesis is organised as follows. In Ch. 2 the optimal-control theory is presented, together with the optimisation strategies employed for solving the minimum-lap-time problems. Direct or indirect methods can be used for solving the optimal-control. In the direct approach the problem is discretised and traduced in a nonlinear-programming problem, which is solved iteratively. In the indirect approach, the problem is stated employing the first-order necessary conditions, which are then numerically solved. The solution of the set of equations obtained can be solved though explicit and implicit formulations. In the explicit case, the equations are solved though a time-marching technique, while in the implicit formulation, a large-scale system of algebraic equations is solved simultaneously for all the available mesh points. In Ch. 3 the modelling techniques employed for the minimum-lap-time computation are discussed. A steady-state race car and motorcycle model are presented first. These models are used for computing the g-g diagrams at different speeds, which are employed for implementing the vehicle model in the steady-state simulations. A race car dynamic model is also presented for the implementation in the dynamic-optimal-control simulations. Ch. 4 is devoted to the generation of the g-g diagrams. An optimisation-based program is developed for building the g-g in a finite number of points, at different speeds. Numerically computed car and motorcycle g-g envelopes are then obtained and the effect of different parameters is discussed. Ch. 5 represents the core chapter of the entire work, since the results obtained in the previous chapters are employed for building the minimum-lap-time problems, using different approaches. Three g-g based steady-state simulations are discussed in detail. A fixed-trajectory apex-finding program is presented first. In this method the speed is computed in the corner-apexes of the trajectory; then, the acceleration and braking speed profiles are calculated between two apexes using the g-g diagrams for limiting the vehicle performance. A fixed-trajectory optimal-control program is also presented. In this case, the g-g diagrams are employed in order to constrain the lateral and longitudinal accelerations of a simple vehicle model, allowed to move on a given trajectory. The previously-mentioned free-trajectory optimal-control program is then discussed and compared to the other simulations in terms of results and computational efficiency using both the race car and the race motorcycle models. In conclusion, a free-trajectory optimal-control method is employed for computing the minimum-lap-time of a dynamic vehicle model. In this case, the model is directly implemented in the optimal-control formulation, without involving the g-g

diagrams. As previously discussed, this method is applied to the analysis of the effect of steering geometry on a race car model in different manoeuvres.

Chapter 2

Optimal Control Problems

2.1 Historical notes

Optimal-control theory has been developed in the last six decades for many different purposes. A great progress on solving this kind of problems has arisen during the Cold War, for military purposes. In 1959, Lev Pontryagin, member of the URSS Academy of Sciences, developed a variational method for solving the optimal control problems: the famous *Pontryagin's Maximum Principle*, which poses the foundations for a family of optimal-control-solving strategies, known as indirect methods. In the same period, Richard E. Bellman, American researcher in the RAND Corporation, developed the theory of *Dynamic Programming* for achieving a numerical solution of the optimal-control problems. The *minimum-time-to-climb* is a famous problem formulated in these years: the target is to calculate the optimal flight trajectory, in order to achieve a given altitude in the minimum time [7]. Another example is related to the *minimum-intercepting-time* problem. In this case, the aim is to find the minimum-time trajectory of a missile for intercepting a target aircraft [8]. Further examples can be found among the space applications, such as the problem of minimum-fuel consumption in satellite orbit transfer, which is investigated in [9, 10]. Many other optimal-control applications span from the industrial-robot trajectory planning, to the parameter optimisation in chemical reactions and the optimisation of cancer treatments [11]. In the vehicle-engineering framework two main targets have been investigated during the years, employing optimal-control methods: optimisation of the driving efficiency of road vehicles and minimisation of the lap-time in motorsport applications. Minimising the fuel consumption has become a critical problem with the spreading of hybrid powertrains in road applications. The higher flexibility of these propulsion systems determines a notably higher complexity in selecting the best control strategies. Some efforts in facing these problems are discussed in [12], where a real-time control strategy is presented for a parallel-hybrid vehicle and compared to the global optimisation strategies. In [13] the Pontryagin's Maximum Principle is applied to a hybrid vehicle, together with proper assumptions for calculating an exact solution suitable for real-time applications. In [14] a direct-collocation approach is employed for the power-split optimisation of a hybrid vehicle. In motorsport applications, the optimal-control problems are widely

employed in a flavour of minimum-lap-time simulations, for evaluating and optimising the vehicle performance along a given track. In [3], the optimal control is found through gradient methods for a lane-change manoeuvre of a front-wheel drive and rear-wheel drive car. The vehicle is represented by a double-track model and the tyre-forces coupling is calculated through friction ellipses. In [4], a basic motorcycle model is applied to indirect-implicit optimal-control methods, in order to assess an index of manoeuvrability of the vehicle. A more complex car model, which includes of the effect of trim on aerodynamics, is employed in [15] to find the minimum-time of a Formula One car, using a direct-collocation method. The trim and aerodynamics variations are parametrised as a lookup table, considering their quasi-steady dependance on the tyre normal loads.

2.2 Fundamentals

In general, the dynamics of a controlled system can be written in terms of the state-space equations:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad \mathbf{x}(t_i) = \mathbf{x}_i, \quad t \in (0, t_f). \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ represents the states of the system, $\mathbf{u} \in \mathbb{R}^m$ represents the controls and $\mathbf{x}_i \in \mathbb{R}^n$ represents the state initial conditions. A *cost function* $\mathcal{J} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ can be defined as a function of the states $\mathbf{x}$ and the controls $\mathbf{u}$, and represents the target that the system has to achieve:

$$\mathcal{J}(\mathbf{x}(t), \mathbf{u}(t), t) = \boldsymbol{\phi}(\mathbf{x}(t_f), t_f) + \int_0^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t) dt. \tag{2.2}$$

The first term $\boldsymbol{\phi} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ of the cost function is called the *Mayer* part and determines the *terminal/final cost*, while the second term $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ represents the *Lagrange* term, or *running cost*; this formulation of the optimal-control problem is usually addressed as the *Bolza* problem.

The optimal-control problem (OCP) has the aim to find the control set $\mathbf{u}^*$ that minimises $\mathcal{J}$, subject to the dynamic constraints (2.1), in such a way that:

$$\mathcal{J}(\mathbf{x}^*(t), \mathbf{u}^*(t), t) \leq \mathcal{J}(\mathbf{x}(t), \mathbf{u}(t), t), \quad \forall \mathbf{u} \in \mathcal{U}, \tag{2.3}$$

where $\mathcal{U}$ represents the set of the admissible controls and $\mathbf{x}^*$ represents the state variables related to the optimal control $\mathbf{u}^*$.

The most widely used approach for solving the optimal-control problem of Eq. (2.3) relates to the *Pontryagin's Minimum Principle* (PMP), which will be discussed in Sec. 2.3. An alternative approach is based on the *Dynamic Programming* technique, which will not be included in the following discussion. The solution of an optimal-control problem is tightly related to the possibility of traducing the problem into a form suitable for the computation. In fact, many strategies have been developed for the numerical solution of the OCP during the last decades. In a simplified classification, an OCP can be solved adopting either explicit or implicit formulations, together with direct or indirect methods. The

explicit strategies (Sec. 2.4.1) allow to integrate the dynamics of the system starting from the previous-steps solutions, i.e. the solution at a given step is calculated from the solutions of the previous steps and this process is iterated until satisfying the boundary conditions. The implicit strategies (Sec. 2.4.2) allow to provide a solution of the problem without calculating the solution step-by-step, along the time domain: in this case, instead, the solution is calculated simultaneously in all the mesh points by solving a large-scale system of algebraic equations. Both explicit and implicit simulation strategies can be employed together with the direct and indirect approaches. In the direct methods (Sec. 2.4.3) the continuous optimal-control problem is first discretised into a *Nonlinear Programming Problem* (NLP) and then solved as an optimisation problem, in a finite-dimensional domain. The indirect methods (Sec. 2.4.4) are based on the research of a solution using the first-order necessary conditions discussed in Sec. 2.3. In this case the optimisation problem is stated first, through the first-order necessary conditions; then, the new set of differential equations is numerically solved. The main differences between direct and indirect approaches will be discussed in Sec. 2.4.5. Moreover, a widely diffused direct-implicit approach (GPOPS-II) will be presented in Sec. 2.5.

2.3 First-order necessary conditions

The optimal control problem (OCP) can be solved in a variational framework, considering the stationary points of the cost function. As reported in Sec. 2.3.3, this approach has many common features to the formulation of the variational-mechanics principles.

2.3.1 Unconstrained problem

The unconstrained representation of the OCP can be obtained introducing the dynamic constraints of the system (2.1) in the cost function (2.2), through the Lagrange multipliers vector $\boldsymbol{\lambda}(t)$, namely the *costate* vector of the OCP. The cost function of the unconstrained problem becomes[1]:

$$\begin{aligned}
\mathcal{J}_{\mathrm{a}}(\mathbf{x},\mathbf{u}) &= \boldsymbol{\phi}(\mathbf{x}(t_f)) + \int_0^{t_f} \left(\mathcal{L}(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^{\mathrm{T}}\left(\mathbf{f}(\mathbf{x},\mathbf{u}) - \dot{\mathbf{x}}\right)\right) dt \\
&= \boldsymbol{\phi}(\mathbf{x}(t_f)) + \int_0^{t_f} \left(\mathcal{L}(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^{\mathrm{T}}\,\mathbf{f}(\mathbf{x},\mathbf{u}) + \dot{\boldsymbol{\lambda}}^{T}\mathbf{x}\right) dt - \boldsymbol{\lambda}^{\mathrm{T}}\,\mathbf{x}\Big|_0^{t_f} \\
&= \boldsymbol{\phi}(\mathbf{x}(t_f)) - \boldsymbol{\lambda}^{\mathrm{T}}\,\mathbf{x}\Big|_0^{t_f} + \int_0^{t_f} \left(\mathcal{H} + \dot{\boldsymbol{\lambda}}^{T}\mathbf{x}\right) dt. \qquad (2.4)
\end{aligned}$$

where $\mathcal{H}$ is the *control Hamiltonian*, given by

$$\mathcal{H} = \mathcal{L}(\mathbf{x},\mathbf{u}) + \boldsymbol{\lambda}^{\mathrm{T}}\,\mathbf{f}(\mathbf{x},\mathbf{u}). \qquad (2.5)$$

[1] In the next sections the time dependency will be omitted.

A variation $\delta\mathbf{u}$ of the controls, produces a correspondent variation $\delta\mathbf{x}$ in the states and $\delta\,\mathcal{J}_\mathrm{a}$ in the cost function:

$$\delta\,\mathcal{J}_\mathrm{a} = \left[\left(\frac{\partial\boldsymbol{\phi}}{\partial\mathbf{x}} - \boldsymbol{\lambda}\right)^T \delta\mathbf{x}\right]_{t=t_f} + \int_0^{t_f}\left(\left(\frac{\partial\mathcal{H}}{\partial\mathbf{x}}\right)^T \delta\mathbf{x} + \left(\frac{\partial\mathcal{H}}{\partial\mathbf{u}}\right)^T \delta\mathbf{u} + \dot{\boldsymbol{\lambda}}^T\delta\mathbf{x}\right)dt. \tag{2.6}$$

The stationary points of the cost function can be found imposing the necessary condition $\delta\,\mathcal{J}_\mathrm{a}(\mathbf{x}^*,\mathbf{u}^*) = 0$. This condition can be satisfied by verifying the following set of equations:

$$\dot{\mathbf{x}}^*(t) = \mathbf{f}(\mathbf{x}^*(t),\mathbf{u}^*(t),t), \tag{2.7}$$

$$\dot{\boldsymbol{\lambda}}^* = -\left.\frac{\partial\mathcal{H}}{\partial\mathbf{x}}\right|_{\mathbf{x}^*,\boldsymbol{\lambda}^*,\mathbf{u}^*}, \tag{2.8}$$

$$0 = \left.\frac{\partial\mathcal{H}}{\partial\mathbf{u}}\right|_{\mathbf{x}^*,\boldsymbol{\lambda}^*,\mathbf{u}^*}, \tag{2.9}$$

together with the following boundary conditions (BCs):

$$\left.\frac{\partial\boldsymbol{\phi}}{\partial\mathbf{x}}\right|_{t_f} = \boldsymbol{\lambda}(t_f), \tag{2.10}$$

$$\mathbf{x}(t_i) = \mathbf{x}_i. \tag{2.11}$$

Equations (2.8) are known as *coequations*, equations (2.9) are known as *optimality equations* and equations (2.10) are the *transversality conditions*. To ensure that the stationary point is a minimum, a further condition have to be set:

$$\left.\frac{\partial^2\mathcal{H}}{\partial^2\mathbf{u}}\right|_{\mathbf{x}^*,\boldsymbol{\lambda}^*,\mathbf{u}^*} \geq 0. \tag{2.12}$$

Relations (2.7), (2.8), (2.9), (2.12) need be satisfied in order to solve the unconstrained optimal-control problem, with the boundary conditions defined in (2.10) and (2.11) for the final and initial point. For this reason this set of equations is known as two-points-boundary-value problem (TPBVP).

Equation (2.12) is not suitable for problems in which the controls are bounded in the admissible control set $\mathcal{U}$. In this case the condition assumes a more general form:

$$\mathcal{H}(\mathbf{x}^*(t),\mathbf{u}^*(t),\boldsymbol{\lambda}(t)^*,t) \leq \mathcal{H}(\mathbf{x}^*(t),\mathbf{u}(t),\boldsymbol{\lambda}(t)^*,t), \quad \forall\mathbf{u}\in\mathcal{U}, \tag{2.13}$$

which is equivalent to:

$$\mathbf{u}^*(t) = \underset{\mathbf{u}(t)\in\mathcal{U}}{\operatorname{argmin}}\,\mathcal{H}(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\lambda}(t),t). \tag{2.14}$$

It should be noticed that the minimum condition is translated from the minimisation of the cost function $\mathcal{J}(\mathbf{x},\mathbf{u})$ (see Eq. 2.3) to the minimisation of the control Hamiltonian $\mathcal{H}(\mathbf{x}^*,\mathbf{u})$, evaluated at the solution state (see Eq. 2.13). Equation (2.14) is the typical formulation of the *Pontryagin's Minimum Principle* (PMP).

Variable final condition

If the final time is not fixed, i.e. in the case of minimum-time applications, (2.6) can be updated, calculating the variation respect to time and introducing (2.8):

$$\begin{aligned}\delta \mathcal{J}_{\mathrm{a}} &= \left[\left(\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}} - \boldsymbol{\lambda}\right)^T \delta\mathbf{x} + \left(\mathcal{H} + \frac{\partial \boldsymbol{\phi}}{\partial t}\right)\delta t\right]_{t=t_f} + \\ &\int_0^{t_f} \left(\left(\frac{\partial \mathcal{H}}{\partial \mathbf{x}}\right)^T \delta\mathbf{x} + \left(\frac{\partial \mathcal{H}}{\partial \mathbf{u}}\right)^T \delta\mathbf{u} + \dot{\boldsymbol{\lambda}}^T \delta\mathbf{x}\right) dt. \end{aligned} \tag{2.15}$$

A new *transversality condition* arises from this formulation:

$$\mathcal{H}|_{t=t_f} = -\left.\frac{\partial \boldsymbol{\phi}}{\partial t}\right|_{t=t_f}. \tag{2.16}$$

2.3.2 Constrained problem

A more general statement of the optimal control problem, includes a cost term both on the initial and final states, together with path constraints (i.e. constraints that affect the system along the time domain) and boundary constraints (i.e. constraints related to the initial and final state). In this new general definition of the problem, the cost function and the constraints have the following form:

$$\mathcal{J}(\mathbf{x}(t), \mathbf{u}(t), t) = \boldsymbol{\phi}(\mathbf{x}(t_i), \mathbf{x}(t_f), t_i, t_f) + \int_{t_i}^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t) dt, \tag{2.17}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \tag{2.18}$$

$$\mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{0}, \tag{2.19}$$

$$\mathbf{b}(\mathbf{x}(t_i), \mathbf{x}(t_f), t_i, t_f) = \mathbf{0}. \tag{2.20}$$

Equation (2.18) represents the dynamic constraints, (2.19) represents the path constraints $\mathbf{c} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^p$, (2.20) represents the boundary constraints $\mathbf{b} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^q$. When introducing the constraints (2.18), (2.19) and (2.20), the cost function becomes:

$$\begin{aligned}\mathcal{J}_{\mathrm{a}}(\mathbf{x}, \mathbf{u}) &= \boldsymbol{\phi}(\mathbf{x}(t_f), \mathbf{x}(t_i)) - \boldsymbol{\nu}^{\mathrm{T}} \mathbf{b}(\mathbf{x}(t_f), \mathbf{x}(t_i)) + \\ &\int_{t_i}^{t_f} \left(\mathcal{L}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^{\mathrm{T}} (\mathbf{f}(\mathbf{x}, \mathbf{u}) - \dot{\mathbf{x}}) + \boldsymbol{\mu}^{\mathrm{T}} \mathbf{c}(\mathbf{x}, \mathbf{u})\right) dt, \end{aligned} \tag{2.21}$$

where $\boldsymbol{\nu}$ are the Lagrange multipliers associated to the boundary constraints $\mathbf{b}$ and $\boldsymbol{\mu}$ are the Lagrange multipliers associated to the path constraints $\mathbf{c}$. The new control Hamiltonian can be defined as:

$$\mathcal{H} = \mathcal{L}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{f}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\mu}^{\mathrm{T}} \mathbf{c}(\mathbf{x}, \mathbf{u}), \tag{2.22}$$

and the new cost-function variation becomes:

$$\begin{aligned}\delta\,\mathcal{J}_{\mathrm{a}} \;=\; & \left[\left(\frac{\partial\boldsymbol{\phi}}{\partial\mathbf{x}}-\boldsymbol{\lambda}-\left(\frac{\partial\mathbf{b}}{\partial\mathbf{x}}\right)^{T}\boldsymbol{\nu}\right)^{T}\delta\mathbf{x}+\left(\mathcal{H}+\frac{\partial\boldsymbol{\phi}}{\partial t}-\boldsymbol{\nu}^{\mathrm{T}}\,\frac{\partial\mathbf{b}}{\partial t}\right)\delta t\right]_{t=t_f} + \\ & \left[\left(\frac{\partial\boldsymbol{\phi}}{\partial\mathbf{x}}+\boldsymbol{\lambda}-\left(\frac{\partial\mathbf{b}}{\partial\mathbf{x}}\right)^{T}\boldsymbol{\nu}\right)^{T}\delta\mathbf{x}+\left(-\mathcal{H}+\frac{\partial\boldsymbol{\phi}}{\partial t}-\boldsymbol{\nu}^{\mathrm{T}}\,\frac{\partial\mathbf{b}}{\partial t}\right)\delta t\right]_{t=t_i} + \\ & \int_{t_i}^{t_f}\left(\left(\frac{\partial\mathcal{H}}{\partial\mathbf{x}}\right)^{T}\delta\mathbf{x}+\left(\frac{\partial\mathcal{H}}{\partial\mathbf{u}}\right)^{T}\delta\mathbf{u}+\dot{\boldsymbol{\lambda}}^{T}\delta\mathbf{x}\right)dt. \qquad (2.23)\end{aligned}$$

The fist-order necessary conditions become:

$$\dot{\boldsymbol{\lambda}} \;=\; -\frac{\partial\,\mathcal{H}}{\partial\mathbf{x}}, \qquad (2.24)$$

$$\mathbf{u} \;=\; \underset{\mathbf{u}\in\mathcal{U}}{\operatorname{argmin}}\,\mathcal{H}, \qquad (2.25)$$

while the transversality conditions are:

$$\boldsymbol{\lambda}(t_f)=\left(\frac{\partial\boldsymbol{\phi}}{\partial\mathbf{x}}-\left(\frac{\partial\mathbf{b}}{\partial\mathbf{x}}\right)^{T}\boldsymbol{\nu}\right)_{t=t_f},\quad \mathcal{H}|_{t=t_f}=\left(\frac{\partial\boldsymbol{\phi}}{\partial t}-\boldsymbol{\nu}^{\mathrm{T}}\,\frac{\partial\mathbf{b}}{\partial t}\right)_{t=t_f} \qquad (2.26)$$

$$\boldsymbol{\lambda}(t_i)=\left(-\frac{\partial\boldsymbol{\phi}}{\partial\mathbf{x}}+\left(\frac{\partial\mathbf{b}}{\partial\mathbf{x}}\right)^{T}\boldsymbol{\nu}\right)_{t=t_i},\quad \mathcal{H}|_{t=t_i}=\left(-\frac{\partial\boldsymbol{\phi}}{\partial t}+\boldsymbol{\nu}^{\mathrm{T}}\,\frac{\partial\mathbf{b}}{\partial t}\right)_{t=t_i} \qquad (2.27)$$

$$\begin{aligned}&\mu_j(t)=0, \quad \text{when} \quad c_j(\mathbf{x},\mathbf{u},t)<0, \quad j=1,...,p,\\ &\mu_j(t)<0, \quad \text{when} \quad c_j(\mathbf{x},\mathbf{u},t)=0, \quad j=1,...,p,\\ &\mu_j(t)>0, \quad \text{when} \quad c_j(\mathbf{x},\mathbf{u},t)>0, \quad j=1,...,p.\end{aligned} \qquad (2.28)$$

The left equations in (2.26) and (2.27) hold only if the final state $\mathbf{x}(t_f)$ is free (i.e. not fixed), while the right equations hold only if the final time t_f is free. Equations (2.28) are known as *complementary slackness conditions*, and have the aim to penalise the cost function if the path constraints given in (2.19) are not satisfied. This formulation of the problem is known as an *Hamiltonian-boundary-value-problem* (HBVP).

2.3.3 Remarks on the variational approach

A fascinating parallelism arises between the achievement of the PMP and the Hamiltonian mechanics. Hamilton's principle, or *principle of stationary action*, states that a mechanical system moves towards a configuration in which the so-called *action integral* is stationary for any possibile perturbation. The action integral is defined as:

$$\mathcal{A}=\int_{t_1}^{t_2}\mathcal{L}(\mathbf{q},\dot{\mathbf{q}},t)dt, \quad \mathcal{L}=T-V \qquad (2.29)$$

where $\mathbf{q}(t)$ represents the vector of generalised coordinates, T is the kinetic energy and V is the potential energy. Hamilton's principle requires that $\delta\mathcal{A} = 0$, that leads to the Euler-Lagrange equations:

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}}{\partial\dot{\mathbf{q}}}\right) - \frac{\partial\mathcal{L}}{\partial\mathbf{q}} = 0. \tag{2.30}$$

When introducing an holonomic constraint $\mathbf{b}(\mathbf{q}, t) = 0$, a new *augmented Lagrangian* can be defined by means of the Lagrange multiplier $\boldsymbol{\lambda}(t)$:

$$\hat{\mathcal{L}} = \mathcal{L} - \boldsymbol{\lambda}^T \mathbf{b}, \tag{2.31}$$

for which the same Euler-Lagrange equations (2.30) hold. In some sense, the Euler-Lagrange equations can be considered as a solution of an unconstrained OCP, in which the target is equal to the action integral. Therefore, in the optimal control framework, the action integral becomes the cost function, while the holonomic constraints become the dynamic equations that describe the control system. The Lagrangian approach make use of n generalised coordinated $\mathbf{q}$, their derivatives $\dot{\mathbf{q}}$ and the time t. In this framework n second-order differential equations need be solved in order to describe the dynamics of the system. In a Hamiltonian approach, the system can be described by n generalised coordinated $\mathbf{q}$, n generalised momenta $\mathbf{p}$ and the time t. This formulation leads to a $2n$ first-order set of differential equations (Hamilton's canonical equations). The conversion from Lagrange equations to Hamilton equations is carried out through the *Legendre transformation*. The generalised momenta are:

$$\mathbf{p} = \frac{\partial\mathcal{L}}{\partial\dot{\mathbf{q}}}, \tag{2.32}$$

while the Lagrangian $\mathcal{L}$ is replaced by the Hamiltonian:

$$\mathcal{H} = \mathbf{p}^T\dot{\mathbf{q}} - \mathcal{L}, \tag{2.33}$$

which is similar to the control Hamiltonian in 2.5. Posing $\delta\,\mathcal{H} = 0$ leads to the *canonical equations of Hamilton*:

$$\dot{\mathbf{q}} = \frac{\partial\mathcal{H}}{\partial\mathbf{p}}, \qquad \dot{\mathbf{p}} = \frac{\partial\mathcal{H}}{\partial\mathbf{q}}. \tag{2.34}$$

In this case, the generalised momenta $\mathbf{p}$ have the same role of the costates $\boldsymbol{\lambda}$ in 2.5. A deeper investigation of these topics is presented in [16].

2.4 Numerical solution of the OCP

The equations that give a representation of the minimum problem, have now to be solved using a numerical strategy, in order to find the optimal control. In this section the explicit and implicit formulations of the OCP will be discussed, together with the direct and indirect approach. Explicit and implicit strategies, differ from how the dynamics differential equations are integrated in each time step. Direct and indirect approaches are related to the moment in which the problem is discretised for a numerical solution. Both direct and indirect methods can implement an explicit or implicit formulation for the integration of the dynamics.

2.4.1 Explicit solution

In the case of an explicit integration method, the solution at the next step is obtained from the solution of the current and previous steps, with a time-marching technique. A typical explicit formulation allows to integrate the system dynamics, starting from a guess of the initial (unknown) condition and calculating the subsequent solutions of the differential equations (step-by-step) until the final time is reached. At the final step, the error between the solution and the final condition (known) is calculated. Then, the initial guess is adjusted and the evaluation of the solution at the final step is iterated, until the error on the final condition is lower than a given value. This formulation is known as *forward-shooting method*. In case the final state is unknown and the initial condition is known, a *backward-shooting method* can be used, integrating the differential equations from the final to the initial condition. In other words, with a *single-shooting method* the solution of differential equations is propagated from a known condition to an unknown state (or vice-versa).

The *multiple-shooting method* is an evolution of this concept, and allows to reduce the sensitivity of the single-shooting strategy while changing the initial condition. In the multiple shooting scenario, the time domain is divided into a set of smaller intervals. The single-shooting method is then applied to each interval, enforcing the continuity at each interval border. The new set of continuity conditions produce a higher dimension problem, although integrating on shorter intervals allows to overcome the high sensitivity related to the propagation of the adjusted initial condition to the solution at the final step (in case of forward integration).

2.4.2 Implicit solution

The implicit integration (or *collocation*) of the dynamics implies solving simultaneously, in each mesh point, a large-scale system of algebraic differential equations. The solution at each mesh interval $k = 1, ..., K$ is approximated as a linear combination of coefficients $\mathbf{a}$ and time-dependent basis functions $\boldsymbol{\psi}(t)$:

$$\mathbf{x}^{(k)}(t) \approx \mathbf{X}^{(k)}(t) = \sum_{j=1}^{J} \mathbf{a}_j^{(k)} \boldsymbol{\psi}_j^{(k)}(t). \tag{2.35}$$

Each mesh interval k is divided in I smaller integration intervals. The solution is calculated at each time-step $t_{ik} \in [t_{k-1}, t_k]$ for $i = 1, ..., I$:

$$\begin{aligned} \mathbf{X}(t_{ik}) &= \mathbf{X}(t_{k-1}) + \int_{t_{k-1}}^{t_{ik}} \mathbf{f}(\mathbf{X}(t), t) dt \\ &= \mathbf{X}(t_{k-1}) + \sum_{j=1}^{I} A_{ij}^{(k)} \, \mathbf{f}(\mathbf{X}_{jk}, t_{jk}), \end{aligned} \tag{2.36}$$

where $\mathbf{f}$ represents the equations to integrate (i.e. the dynamics equations), while $A^{(k)}$ is the integration matrix for interval k, associated to the rule of integration. The *defect conditions* (2.36) can be rewritten in the form $\mathbf{F}(\mathbf{X}(t), t) = 0$

by subtracting the right-hand-side of the equation to its left-hand-side. This set of equations, together with the continuity conditions between each mesh interval (similarly to the *multiple shooting*), can be solved for the coefficients $\mathbf{a}^{(k)}$, in each mesh interval, in order to calculate the solution $\mathbf{x}$. The implicit methods differ from the choice of basis functions $\boldsymbol{\psi}(t)$ (typically piecewise polynomials) and the integration scheme used (or quadrature method).

2.4.3 Direct methods

Direct methods are based on the transcription of the OCP into a nonlinear-programming-problem (NLP). The dynamics equations (2.1) and the cost function (2.2) are directly discretised, while the optimisation problem is solved in a finite-dimensional domain through a large-scale NLP solver (e.g. IPOPT, SNOPT).

Direct explicit solution

A *direct-shooting method* combines the simplicity of the explicit integration of the dynamics equations (2.18) to the direct transcription of the OCP into a discrete-time NLP. The controls of the system are approximated by piecewise polynomials: the set of parameters that describes these approximating functions need be determined by the optimiser. Then, an integration scheme is selected for solving the dynamics equations step-by-step, while a consistent quadrature formula is used for evaluating the cost function (2.17). The OCP is solved through the arisen NLP, which does not include the OCP Lagrange multipliers.

Direct implicit solution

In *direct-collocation methods*, both the states and the controls are approximated by sets of basis functions. A widely employed family of methods for direct collocation of OCPs is known as *orthogonal-collocation* or *pseudospectral* methods. In these methods the states are approximated by a linear combination of Lagrange polynomials (basis functions), while the collocation is based on Gauss quadrature, that guarantees exponential convergence. In this case, the accuracy of the approximation can be improved both adjusting the number and disposition of collocation points and adjusting the polynomial degree, introducing more flexibility than the classical fixed-order integration schemes (i.e. Runge-Kutta, Hermite-Simson, ...).

Considering an OCP defined by (2.18-2.20), the time domain $t \in [t_i, t_f]$ can be conveniently transformed in the new scaled time interval $\tau \in [-1, 1]$, through the affine transformation:

$$t = \frac{t_f - t_i}{2}\,\tau + \frac{t_f - t_i}{2}. \tag{2.37}$$

The interval considered is furthermore divided into K subintervals $[T_{k-1}, T_k]$, with $k = 1, \dots K$ and T_k are the *mesh points* in which the solution will be calculated.

In this discrete-time description of the OCP, the cost function becomes:

$$\mathcal{J} = \boldsymbol{\phi}(\mathbf{x}^{(1)}(-1), \mathbf{x}^{(K)}(+1), t_i, t_f) + \frac{t_f - t_i}{2} \sum_{k=1}^{K} \int_{T_{k-1}}^{T_k} \mathcal{L}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t(\tau, t_i, t_f)) d\tau, \tag{2.38}$$

while the dynamic constraints are:

$$\frac{d\mathbf{x}^{(k)}(\tau)}{d\tau} = \frac{t_f - t_i}{2} \mathbf{f}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t(\tau, t_i, t_f)), \tag{2.39}$$

with path constrints:

$$\mathbf{c}_{min} \leq \mathbf{c}(\mathbf{x}^{(k)}(\tau), \mathbf{u}^{(k)}(\tau), t(\tau, t_i, t_f)) \leq \mathbf{c}_{max}, \tag{2.40}$$

and boundary conditions:

$$\mathbf{b}(\mathbf{x}^{(1)}(-1), \mathbf{x}^{(K)}(+1), t_i, t_f) = \mathbf{0}. \tag{2.41}$$

Further boundary conditions $\mathbf{x}(T_k^-) = \mathbf{x}(T_k^+)$ need be included, in order to take into account for the continuity at each mesh point. In order to introduce the quadrature approximation, the dynamics equations (2.39) can be replaced by an equivalent integral form:

$$\mathbf{x}^{(k)}(\tau) = \mathbf{x}^{(k)}(T_{k-1}) + \frac{t_f - t_i}{2} \int_{T_{k-1}}^{\tau} \mathbf{f}(\mathbf{x}^{(k)}(\zeta), \mathbf{u}^{(k)}(\zeta), t(\zeta, t_i, t_f)) d\zeta. \tag{2.42}$$

In a similar way to Eq. (2.35) and (2.36), the states can be approximated employing a *Legendre-Gauss-Randau* (LGR) collocation scheme:

$$\mathbf{x}^{(k)} \approx \mathbf{X}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} \, l_j^{(k)}, \tag{2.43}$$

$$l_j^{(k)}(\tau) = \prod_{l=1}^{N_k+1} \frac{\tau - \tau_l^{(k)}}{\tau_j^{(k)} - \tau_l^{(k)}}, \quad l \neq j, \tag{2.44}$$

where $l_j^{(k)}(\tau)$, for $j = 1, \ldots, N_k+1$, represents a basis of *Lagrange polynomials*, and $\tau_1^{(k)}, \ldots, \tau_{N_k}^{(k)}$ are the *Legendre-Gauss-Randau* (LGR) collocation points. The collocation points, in each mesh interval k, are determined by the roots of a polynomial obtained by the sum of an (N_k)-degree and a (N_k-1)-degree *Legendre polynomial*. The peculiar formulation of the LGR polynomial generates an unevenly-spaced set of collocation points, which does not include the upper border of the mesh interval. Since the states and controls are evaluated (and approximated) in the collocation points, their location has an important role in the convergence of the OCP. Equation (2.42) can be further discretised in the interval $[T_{k-1}, T_k]$ by means of an LGR quadrature scheme:

$$\mathbf{X}_{i+1}^{(k)} = \mathbf{X}_1^{(k)} + \frac{t_f - t_i}{2} \sum_{j=1}^{N_k} A_{ij}^{(k)} \mathbf{f}(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}, t(\tau_j^{(k)}, t_i, t_f)), \quad i = 1, \ldots, N_k, \tag{2.45}$$

where $\mathbf{U}^{(k)}$ represents the approximation of the controls and $A_{ij}^{(k)}$ is an element of the *LGR integration matrix* in interval k. The NLP that arises from the LGR discretisation has the aim of minimising the approximated cost function

$$\mathcal{J} \approx \boldsymbol{\phi}(\mathbf{X}_1^{(1)}, \mathbf{X}_{N_k+1}^{(K)}, t_i, t_f) + \sum_{k=1}^{K} \sum_{j=1}^{N_k} \frac{t_f - t_i}{2} w_j^{(k)} \mathcal{L}(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}, t(\tau_j^{(k)}, t_i, t_f)), \quad (2.46)$$

where $w_j^{(k)}$ are the *LGR quadrature weights*. Moreover, in discrete-time, the NLP is subject to defect constraints

$$\mathbf{X}_{i+1}^{(k)} - \mathbf{X}_1^{(k)} - \frac{t_f - t_i}{2} \sum_{j=1}^{N_k} A_{ij}^{(k)} \mathbf{f}(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}, t(\tau_j^{(k)}, t_i, t_f)) = \mathbf{0}, \quad i = 1, \ldots, N_k, \quad (2.47)$$

path constraints

$$\mathbf{c}_{min} \leq \mathbf{c}(\mathbf{X}_i^{(k)}, \mathbf{U}_i^{(k)}, t(\tau_i^{(k)}, t_i, t_f)) \leq \mathbf{c}_{max}, \quad i = 1, \ldots, N_k, \quad (2.48)$$

and boundary conditions

$$\mathbf{b}(\mathbf{X}_1^{(1)}, \mathbf{X}_{N_k+1}^{(K)}, t_i, t_f) = \mathbf{0}. \quad (2.49)$$

The continuity conditions of the states at each mesh point are:

$$\mathbf{X}_{N_k+1}^{(k)} = \mathbf{X}_1^{(k+1)}, \quad k = 1, \ldots, K-1. \quad (2.50)$$

This condition can be deleted using the same variable for both $\mathbf{X}_{N_k+1}^{(k)}$ and $\mathbf{X}_1^{(k+1)}$. In summary, the NLP generated by the transcription of the OCP in a finite-dimensional domain (i.e. approximating the states, controls and cost function), aims to minimise (2.46), subject to the constraints (2.47 - 2.49). In this approach, both the states and controls are discretised, leading to a larger NLP than in the *explicit* solution (where only the controls are discretised).

The LGR collocation scheme used to explain the transcription steps is the method implemented in GPOPS-II. Though, different collocation strategies are available, i.e. Legendre-Gauss (LG) and Legendre-Gauss-Lobatto (LGL). These methods differ from the location of the collocation points, that is related to different combinations of the *Legendre polynomials*. In the next section further details on collocation schemes will be discussed.

NLP solution

Considering the minimisation of a function $\mathcal{J}(\mathbf{z})$, subject to equality constraints $\hat{\mathbf{b}}(\mathbf{z})$ and inequality constraints $\mathbf{c}(\mathbf{z})$, the NLP can be formulated as:

$$\min \mathcal{J}(\mathbf{z}) \quad \text{subject to} \quad \begin{cases} \hat{\mathbf{b}}(\mathbf{z}) = 0 \\ \mathbf{c}(\mathbf{z}) \leq \mathbf{0} \end{cases} \quad (2.51)$$

where $\mathbf{z} = (\mathbf{x}, \mathbf{u})$. The *Karush-Kuhn-Tucker* (KKT) conditions are fundamental to generalise the Lagrange multipliers method for a problem that includes inequality constraints. The Lagrangian can be formulated as:

$$L(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathcal{J}(\mathbf{z}) + \hat{\mathbf{b}}^T \boldsymbol{\lambda} + \mathbf{c}^T \boldsymbol{\mu}\,. \quad (2.52)$$

It should be noticed that this formulation is suitable also for the NLP that arises from the transcription of the OCP, considering that both (2.47) and (2.49) are included in the equality constraints $\hat{\mathbf{b}}$. The KKT necessary conditions that arise form $\nabla L(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = 0$ are:

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{z}} &= \frac{\partial \mathcal{J}}{\partial \mathbf{z}} + \left(\frac{\partial \hat{\mathbf{b}}}{\partial \mathbf{z}}\right)^T \boldsymbol{\lambda} + \left(\frac{\partial \tilde{\mathbf{c}}}{\partial \mathbf{z}}\right)^T \boldsymbol{\mu} = 0, && (2.53)\\
\frac{\partial L}{\partial \boldsymbol{\lambda}} &= \hat{\mathbf{b}}(\mathbf{z}) = 0, && (2.54)\\
\frac{\partial L}{\partial \boldsymbol{\mu}} &= \tilde{\mathbf{c}}(\mathbf{z}) = 0, \quad \boldsymbol{\mu} \geq 0, && (2.55)
\end{aligned}$$

where $\tilde{\mathbf{c}}(\mathbf{z})$ represents the *active inequality constraints*, i.e. the set of inequality constraints for which $\mathbf{c}(\mathbf{z}^*) = 0$. Condition (2.55) requires that, at the border of the *active* inequalities $\tilde{\mathbf{c}}(\mathbf{z})$, the gradient of the cost function and the gradient of the inequality constraints assume opposite directions, i.e. the frontier point $\mathbf{z}^*$ is a minimum. In fact, assuming $\hat{\mathbf{b}}(\mathbf{z}^*) = 0$, the Lagrangian gradient can be calculated as:

$$\begin{aligned}
0 &= \nabla L(\mathbf{z}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \nabla \mathcal{J}(\mathbf{z}^*) + \nabla \tilde{\mathbf{c}}^T(\mathbf{z}^*)\, \boldsymbol{\mu},\\
\mu_i &= -\frac{\nabla \mathcal{J}(\mathbf{z}^*)}{\nabla \tilde{c}_i(\mathbf{z}^*)}, \quad i = 1, \ldots, N_c, && (2.56)
\end{aligned}$$

where N_c is the number of active inequalities.

The solution of the NLP can be addressed with different strategies: the *penalty method*, the *augmented Lagrangian method*, the *sequential-quadratic-programming* and the *interior point* or *barrier method*. The *penalty* approach can be employed for NLPs with no inequality constraints. The equality constraints are multiplied by penalty terms in order to obtain an unconstrained problem. The new cost function becomes:

$$\mathcal{J}(\mathbf{z}) + \alpha \hat{\mathbf{b}}(\mathbf{z})^2, \tag{2.57}$$

where $\alpha > 0$ represents the penalty parameter. The penalty parameter is incremented until the constraints are satisfied. The *augmented Lagrangian methods* lead to a slightly different cost function:

$$\mathcal{J}(\mathbf{z}) - \boldsymbol{\lambda}^T \hat{\mathbf{b}}(\mathbf{z}) + \alpha \hat{\mathbf{b}}(\mathbf{z})^2, \tag{2.58}$$

where λ_i represents an estimate of the Lagrange multiplier, which need be updated at each iteration, together with α, in order to reach the solution.

In the *sequential-quadratic-programming* (SQP) the NLP is approximated to a linear-quadratic-programming (LQP) subproblem, at each iteration. The LQP solution defines the search direction $\mathbf{d}_k$ at each iteration k, i.e. the search direction that minimises the cost function calculated at iteration $k+1$ through a Taylor approximation:

$$\mathcal{J}(\mathbf{z}_{k+1}) \approx \mathcal{J}(\mathbf{z}_k) + \nabla \mathcal{J}(\mathbf{z}_k)^T \mathbf{d}_k + \frac{1}{\|\mathbf{d}_k\|} \mathbf{d}_k^T \nabla_{zz} L(\mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k) \mathbf{d}_k, \tag{2.59}$$

subject to:

$$\hat{\mathbf{b}}(\mathbf{z}_{k+1}) \approx \hat{\mathbf{b}}(\mathbf{z}_k) + \nabla\hat{\mathbf{b}}(\mathbf{z}_k)^T\mathbf{d}_k = 0, \tag{2.60}$$

$$\mathbf{c}(\mathbf{z}_{k+1}) \approx \mathbf{c}(\mathbf{z}_k) + \nabla\mathbf{c}(\mathbf{z}_k)^T\mathbf{d}_k \leq 0. \tag{2.61}$$

The last term of the cost function (2.59) relates to a common choice for improving the robustness of the algorithm, although it does not correspond to the proper Hessian of $\mathcal{J}$.

In the *interior-point methods* a barrier-penalty function is used to find a solution of the NLP. A new set of *slack variables* $\mathbf{s}$ is defined in order to reduce the original NLP into an equality-constrained NLP. The cost function becomes:

$$\mathcal{J}(\mathbf{z}) - \boldsymbol{\lambda}^T\hat{\mathbf{b}}(\mathbf{z}) - \boldsymbol{\mu}^T\left(\mathbf{c}(\mathbf{z}) - \mathbf{s}(\mathbf{z})\right) - \boldsymbol{\tau}\log\mathbf{s}(\mathbf{z}), \tag{2.62}$$

subject to:

$$\hat{\mathbf{b}}(\mathbf{z}) = 0, \tag{2.63}$$

$$\mathbf{c}(\mathbf{z}) - \mathbf{s} = 0, \tag{2.64}$$

$$\mathbf{s} \leq 0. \tag{2.65}$$

It should be noticed that the *slack variables* $\mathbf{s}$ are set equal to the inequality constraints $\mathbf{c}$, which are satisfied only if $\mathbf{s} \leq 0$. In order to find the solution of the NLP, the barrier parameter $\boldsymbol{\tau} > 0$ is reduced towards zero. The obtained solution is an approximation of the solution of the original NLP.

Mesh refinement

In the *direct-transcription methods* the solution of the NLP (namely the real optimisation step) is only one part of the algorithm. At this point the accuracy of the solution need be evaluated and compared to a desired value. If the accuracy is insufficient, a new refined mesh is generated and the new refined NLP is solved. This iteration is repeated until the desired accuracy is reached. The determination of the mesh points and collocation points is known as *mesh-refinement* algorithm. The h part of the mesh refinement method is related to the possibility of modifying the mesh-point number and location. The control of the number and location of the collocation points within a single mesh interval is related to the p part of the refinement strategy. A p collocation method employs a single mesh interval, while the subsequent mesh refinements are determined by varying the degree of the approximating polynomial. These methods give a good convergence when smooth solutions can be achieved. The hp collocation methods allow to vary both the number of mesh points and the degree of the approximating polynomials within each mesh interval, taking into account for the computational efficiency of the h methods, together with the advantages of using p methods where the solution is smooth.

Different collocation schemes can be employed for the direct transcription. For instance, a Legendre-Gauss (LG) scheme determines N collocation points within a single mesh interval by finding the roots of an N-degree Legendre polynomial

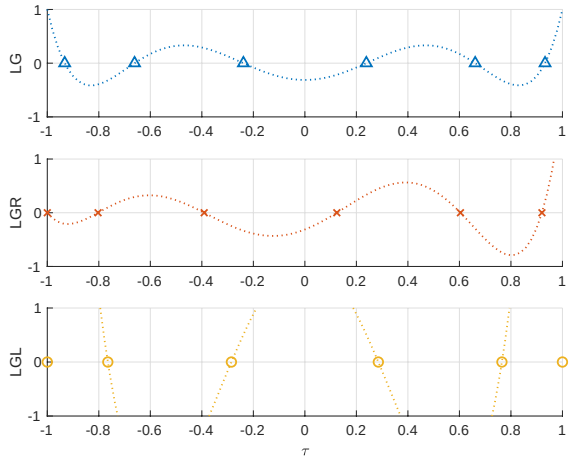


Figure 2.1: Comparison between the LG, LGR and LGR collocation schemes.

$P_N(\tau)$. In a Legendre-Gauss-Radau (LGR) scheme, the collocation points are determined by the N roots of $P_{N-1}(\tau)+P_N(\tau)$, while in a Legendre-Gauss-Lobatto (LGL) scheme, the N collocation points are evaluated from the $N-2$ roots of $\dot{P}_{N-1}(\tau)$, together with the two extreme points of the mesh interval. In Fig. 2.1 the LG, LGR and LGL collocation points are shown, together with the corresponding Legendre polynomials. Considering a normalised time interval $\tau \in (-1, 1)$, a set of approximating polynomials $l(\tau)$, and a set of quadrature weights w_i (dependent on the quadrature scheme selected), these three collocation schemes have the property that:

$$\int_{-1}^{+1} l(\tau)d\tau = \sum_{i=1}^{N} w_i l(\tau_i), \tag{2.66}$$

is an exact equation (i.e. not a quadrature approximation) for polynomial degree equal to $2N-1$ (LG), $2N-2$ (LGR) and $2N-3$ (LGL), i.e. the degree of the approximating polynomial determines the number of collocation points in a single mesh interval. The approximating polynomials can be for example the Lagrange polynomials (2.44), which are actually used in GPOPS-II. It should be underlined that the LGL scheme has proved to have minor convergence rate than LG and LGR, because it determines an oscillating costate dynamics, due to the intrinsical properties of the mathematical formulation (see [17]).

2.4.4 Indirect methods

In the indirect methods the optimisation problem is stated through the first-order necessary conditions, which come from (2.7) and (2.24), together with a relatively small optimisation subproblem (2.25). The arising set of differential equations is then numerically solved.

Indirect explicit solution

In an *indirect-shooting* approach, an initial guess on the unknown boundary condition is assumed first. Then, the differential equations (2.7) and (2.24) are integrated towards the known boundary with a time-marching criterium. The control law at each step is obtained from (2.25). At the last integration step, the final conditions are compared to those given in (2.10). The process is iterated adjusting the initial guess, until the error between the evaluated final condition and the actual terminal condition lays below a given threshold. Differently from the direct shooting method, in this case the OCP Lagrange multipliers are involved in the calculation.

Indirect implicit solution

In an *indirect-collocation* scenario, both the states and the Lagrange multipliers (costates) are approximated through piecewise polynomials, according to a certain integration scheme (Runge-Kutta, Hermite-Simpson, orthogonal collocation). The new set of discretised differential equations leads to a finite-dimensional algebraic equation system, that can be solved using an appropriate root-finding algorithm (Newton-Raphson, ...).

An example of indirect collocation is now briefly discussed, in order to underline the basic features of this class of methods. The equality-constrained problem defined by the dynamics equations (2.18) and the constraints (2.20) is transformed into an unconstrained problem through the formulation of the TPBVP, i.e. introducing the first-order necessary conditions (2.24)-(2.25). The inequality constraints (2.19) are then removed introducing a set of penalty functions $\mathbf{p}(\mathbf{x}, \mathbf{u})$. The Hamiltonian becomes:

$$\mathcal{H} = \mathcal{L}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^{\mathrm{T}}\, \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{p}^{T}\, \mathbf{c}(\mathbf{x}, \mathbf{u}), \tag{2.67}$$

where

$$p_i(\mathbf{x}, \mathbf{u}) = \begin{cases} \left(1 + \frac{c_i(\mathbf{x},\mathbf{u})}{h_i}\right)^{n_i} & \text{if} \quad t > -h_i, \\ 0 & \text{otherwise}, \end{cases} \tag{2.68}$$

represents an example of penalty function (with $i = 1, \ldots, N_c$), which increases if the i-th constraint is violated. The parameters n and h define the sharpness of the penalty function. It should be noticed that, with this penalty approach, the violation of the inequality constraints is permitted. Alternatively, a set of barrier functions can be used instead of the penalty functions: in this case no constraint violation is allowed. The arising algebraic-differential system of equations is then discretised into a finite-dimensional algebraic problem, through a proper integration scheme. To reduce the complexity of the problem, the controls are eliminated by the symbolical (when possible) or numerical solution $\mathbf{u}(\mathbf{x}, \boldsymbol{\lambda})$ of the optimality condition (2.25). It is noticed that the control set is explicitly determined only when the states $\mathbf{x}$ and costates $\boldsymbol{\lambda}$ are found. The use of penalty or barrier functions transforms the OCP in a smooth problem. The new set of discrete-algebraic equations $\mathbf{F}(\mathbf{z})$ can be handled by a proper numerical root-finding solver. For instance, a slightly modified Newton method can be implemented in order to improve the

convergence of the problem. The search direction $\mathbf{d}_j$ at iteration j is computed first by:

$$\mathbf{d}_j = -\left(\frac{\partial \mathbf{F}(\mathbf{z}_j)}{\partial \mathbf{z}_j}\right)^{-1} \mathbf{F}(\mathbf{z}_j), \tag{2.69}$$

while the unknown $\mathbf{z}$ can be computed by:

$$\mathbf{z}_{j+1} = \mathbf{z}_j - \alpha_j \mathbf{d}_j, \tag{2.70}$$

where the damping factor α_j is the first number in the sequence $\{1, q, q^2, q^3, \ldots, q_{max}\}$ with $q \in [0.5, 1)$ that satisfies the condition:

$$\left\| \frac{\partial \mathbf{F}(\mathbf{z}_j)}{\partial \mathbf{z}_j} \mathbf{F}(\mathbf{z}_{j+1}) \right\| \leq \kappa_j \|\mathbf{d}_j\|, \tag{2.71}$$

with $\kappa_j < 1$ chosen in the sequence $\{1 - q^i/2\}$. Convergence is reached when $\alpha_j = 1$ and $\|\mathbf{d}_j\|$ is lower than a predefined tolerance. The described procedure is implemented in the software PINS [18].

2.4.5 Remarks on direct and indirect methods

Two basic steps are necessary for the solution of the OCP: the calculation of the optimality conditions and the discretisation of the problem. The order of these steps represents the fundamental difference between direct and indirect methods. In the direct approach the OCP is discretised first, while the optimality conditions are stated in a finite-dimensional domain for solving the arisen discrete NLP. In the indirect approach, the first-order optimality conditions are achieved in a continuous-time domain, and are then discretised for obtaining a numerical (i.e. discrete) solution. Obtaining the first-order optimality conditions in a closed form, allows to achieve a good performance of the algorithm, as stated in [18]. Moreover, a symbolical or approximated formulation of the controls allows to employ the indirect approach in real-time applications, as discussed in [19]. On the other hand, this method results less intuitive, since the symbolic calculation of the PMP is not straightforward and the costates have to be expressed explicitly, as reported in [11]. Considering now the following unconstrained OCP:

$$\begin{aligned} &\mathcal{J} = \boldsymbol{\phi}(\mathbf{x}(T)), \\ &\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \\ &\mathbf{x}(t_0) = \mathbf{x}_0. \end{aligned} \tag{2.72}$$

In an indirect collocation, the PMP leads to:

$$\begin{aligned} \dot{\boldsymbol{\lambda}} &= -\frac{\partial \mathcal{H}}{\partial \mathbf{x}}, \\ 0 &= \frac{\partial \mathcal{H}}{\partial \mathbf{u}}, \\ \boldsymbol{\lambda}(T) &= \left.\frac{\partial \boldsymbol{\phi}}{\partial \mathbf{x}}\right|_{t=T}, \end{aligned} \tag{2.73}$$

with $\mathcal{H} = \boldsymbol{\lambda}^T \mathbf{f}$. In a direct collocation scenario, employing a single-state Euler quadrature method ($A_{ij}^{(k)} = h$ and $w_j^{(k)} = 1$):

$$\begin{aligned} \mathbf{X}_{k+1} &= \mathbf{X}_k + h\,\mathbf{f}(\mathbf{X}_k, \mathbf{U}_k), && (2.74) \\ \mathcal{J}_a &= \boldsymbol{\phi}(\mathbf{X}_k) + \sum_{k=1}^{K-1} \boldsymbol{\Lambda}_k^T (\mathbf{X}_{k+1} - \mathbf{X}_k - h\,\mathbf{f}(\mathbf{X}_k, \mathbf{U}_k)), && (2.75) \end{aligned}$$

while the KKTs become:

$$\begin{aligned} \frac{\partial \mathcal{J}_a}{\partial \mathbf{X}_k} &= -h\boldsymbol{\Lambda}_k \frac{\partial \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k)}{\partial \mathbf{X}_k} - \boldsymbol{\Lambda}_k + \boldsymbol{\Lambda}_{k-1} = 0, \\ \frac{\partial \mathcal{J}_a}{\partial \mathbf{U}_k} &= -h\boldsymbol{\Lambda}_k \frac{\partial \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k)}{\partial \mathbf{U}_k} = 0, \\ \frac{\partial \mathcal{J}_a}{\partial \mathbf{X}_K} &= \boldsymbol{\Lambda}_{K-1} + \frac{\partial \boldsymbol{\phi}(\mathbf{X}_K)}{\partial \mathbf{X}_K} = 0. \end{aligned} \qquad (2.76)$$

It should be noticed that, for $h \to 0$, the PMP equations 2.73 correspond to the KKT equations 2.76. Therefore, the costates in the direct approach are an approximation of the costates calculated in the indirect approach.

Some important insights on direct and indirect methods can be found in [20],[21] and [22].

2.5 GPOPS-II

GPOPS-II (*General Purpose Optimal Control Software*), represents a widespread implementation of the direct collocation methods, for the solution of optimal control problems; many details are discussed in [23]. The software takes advantage of a MATLAB implementation, that allows the user to build the problem by defining the fundamental parameters and functions that describe the system dynamics, together with the the set-up parameters of the OCP. The key components of the software relate to the transcription algorithm, based on the LGR orthogonal collocation, and the *hp*-adaptive-mesh-refinement method for the determination of the number of mesh points and the approximating polynomial degree. The optimisation step within a single mesh, is provided by a large-sparse NLP solver, such as IPOPT or SNOPT. First, the problem is discretised (i.e. transcribed) into an NLP through state and control approximation as in (2.43) and (2.44). An NLP is now generated, with the aim of minimising the cost function (2.46), subject to the constraints (2.47 - 2.49). The NLP is then presented to the optimiser, which calculates the solution of the OCP, at the current mesh iteration, through the KKT (2.53-2.55). After the optimisation step, the solution error is evaluated and compared to the user-defined error tolerance. If the error is higher than the accepted tolerance, a mesh refinement step is carried out. The new discretisation leads to a new NLP to be solved by the optimiser. These steps are repeated until the error tolerance on different meshes is satisfied.

In GPOPS-II both *hp* and *ph* adaptive-mesh-refinement schemes are implemented. The key difference between the *hp* and the *ph* methods lies in the manner

in which the decision is made to either increase the number of collocation points in a mesh interval or to refine the mesh. The *hp* strategy, described in [24], increases the degree of the approximating polynomial if the ratio of the maximum curvature over the mean curvature of the state in a particular mesh interval exceeds a user-specified threshold. In this case, the mesh spacing and the polynomial degree for the mesh refinement are determined simultaneously. The dynamics and the constraints are evaluated in the midpoints of the N_k collocation intervals, within a single mesh interval k. A midpoint is defined as:

$$\bar{\tau}_j^{(k)} = \frac{\tau_j^{(k)} + \tau_{j+1}^{(k)}}{2}, \quad j = 1, \ldots, N_k. \tag{2.77}$$

The j-th column of the *midpoint residual matrix* for the mesh interval k, can be defined as follows:

$$\mathbf{R}_j^{(k)} = \left| \sum_{l=1}^{N_k+1} D_{jl}^{(k)} \mathbf{X}_j^{(k)} - \frac{t_f - t_i}{2} \mathbf{f}(\mathbf{X}_j^{(k)}, \mathbf{U}_j^{(k)}, t(\tau_j^{(k)}, t_i, t_f)) \right| \tag{2.78}$$

where t_f and t_i represent the final and initial time at mesh interval k, $D_{jl}^{(k)}$ is an element of the LGR differentiation matrix, and $\mathbf{X}^{(k)}$ and $\mathbf{U}^{(k)}$ represent the approximation of the states and the controls respectively. The elements of vector $\mathbf{R}_j^{(k)}$ represent the residuals of the dynamics at the midpoints and provide a measure on the violation of the state dynamics. A vector $\mathbf{r}$ can be defined in such a way that each element contains the largest value of $\mathbf{R}_j^{(k)}$, for $j = 1, \ldots, N_k - 1$. A *scaled midpoint residual vector* can be defined as:

$$\boldsymbol{\beta} = \frac{\mathbf{r}}{\bar{r}}, \quad \bar{r} = \frac{\sum_{i=1}^{N_k-1} r_i}{N_k - 1}, \tag{2.79}$$

where r_i is an element of $\mathbf{r}$. The vector $\boldsymbol{\beta}$ is a scaled measure of how far the collocation condition is far from zero in a collocation midpoint, and gives a metric for the determination of the subsequent refinement strategy. If the user-defined error tolerance ε is lower than the maximum value of $\mathbf{r}$, a mesh-refinement step is needed for calculating the solution. If no elements of $\boldsymbol{\beta}$ are higher than a user-defined parameter ρ, the case of *uniform-type* errors occurs, and the number of collocation points is increased by a user-defined value. If some elements of $\boldsymbol{\beta}$ exceed the parameter ρ, the errors are *nonuniform-type* and new mesh points are added where the elements of the scaled midpoint residual vector $\beta_i > \rho$. In [25] a *ph* strategy is presented. The polynomial degree in a mesh interval is increased if the estimate of the required polynomial degree is lower than a user-defined limit. If the limit is exceeded, the mesh interval is divided into e defined number of mesh intervals, i.e. the number of mesh points is increased. In this case, the polynomial degree is incremented first (p refinement), while the mesh spacing (h refinement) is adjusted only if the polynomial approximation step fails (p-then-h). The error estimation is carried out by comparison of two different approximations of the states. The underlying idea is that, in a problem in which the state evolution is smooth, increasing the collocation points leads to a better approximation of the

solution, i.e. a solution that fits the state dynamics more accurately. For this reason, the estimate of the mesh-interval-error is calculated through the difference between the solution obtained at the current refinement step and the approximated solution obtained by increasing the LGR collocation points. Both the states and the controls are approximated through the Lagrange polynomials, at each mesh interval k (see Eq. 2.43). An improved state approximation is defined incrementing the polynomial order or, equivalently, the number of collocation points. The state at each new quadrature point j is:

$$\hat{\mathbf{X}}_j^{(k)} = \mathbf{X}_1^{(k)} + \frac{t_f - t_i}{2} \sum_{l=1}^{M_k} \hat{A}_{jl}^{(k)} \mathbf{f}(\mathbf{X}_l^{(k)}, \mathbf{U}_l^{(k)}, t(\tau_l^{(k)}, t_i, t_f)), \quad j = 2, \ldots, M_k + 1, \tag{2.80}$$

where $\mathbf{X}^{(k)}$ is the approximation of the states, $\mathbf{U}^{(k)}$ is the approximation of the controls, $M_k = N_k + 1$ is the new number of collocation points and $\hat{A}_{jl}^{(k)}$ is the LGR integration matrix corresponding to the points $(\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_{M_k}^{(k)})$. An *absolute* and *relative error* can be defined as:

$$E_j^{(k)} = \left| \hat{X}_j^{(k)} - X_j^{(k)} \right|, \tag{2.81}$$

$$e_j^{(k)} = \frac{E_j^{(k)}}{1 + \max \left| X_l^{(k)} \right|}, \quad l = 1, \ldots, M_k + 1, \tag{2.82}$$

where $X^{(k)}$ and $\hat{X}^{(k)}$ are the components of the approximated state vectors $\mathbf{X}^{(k)}$ and $\hat{\mathbf{X}}^{(k)}$. The *maximum relative error* is then:

$$e_{max}^{(k)} = \max \mathbf{e}^{(k)}, \quad \mathbf{e}^{(k)} = [e_1^{(k)}, \ldots, e_{N_x}^{(k)}], \tag{2.83}$$

where N_x is the number of states. In a global collocation scheme the error behaves like $O(N_k^{2.5-K_d})$, where N_k is the number of collocation points in the mesh interval k and K_d is the number of continuous derivatives in the solution ($K_d = N_k$ if the solution is smooth). Increasing the number of collocation points to $M_k = N_k + P_k$, the error decreases at least by the factor $N_k^{-P_k}$. If the error $e_{max}^{(k)}$ is higher than the desired error tolerance ε, a mesh refinement is needed in such a way that the error is reduced by a factor $\varepsilon / e_{max}^{(k)}$. In other words:

$$N_k^{-P_k} = \varepsilon / e_{max}^{(k)}, \tag{2.84}$$

then

$$P_k = \left\lceil \log_{N_k} \left(\frac{e_{max}^{(k)}}{\varepsilon} \right) \right\rceil. \tag{2.85}$$

In case $M_k > N_{max}$, i.e. the desired degree of approximating polynomials is higher than the user-defined maximum polynomial degree, a h-refinement is performed. The total number of collocation points in the new mesh intervals equals to the predicted polynomial degree M_k. Moreover, each new mesh interval contains the

user-defined minimum number of collocation points. Then, the number of newly-defined mesh intervals is:

$$B_k = \max\left(\left\lceil \frac{M_k}{N_{min}} \right\rceil, 2\right). \tag{2.86}$$

Another important aspect to underline is represented by the scaling of the NLP problem, that allows a faster and more robust convergence to the optimal solution: the software employs an automatic-scaling approach, that transforms a generic variable $x \in [a, b]$ to a new scaled variable $\tilde{x} \in [-1/2, 1/2]$ considering the following expressions:

$$\tilde{x} = v_x x + r_x, \tag{2.87}$$

$$v_x = \frac{1}{b-a}, \tag{2.88}$$

$$r_x = \frac{1}{2} - \frac{b}{b-a}. \tag{2.89}$$

2.5.1 IPOPT

IPOPT(*Interior Point Optimizer*) is a widespread NLP solver, based on an interior-point algorithm with a filter line-search method [26]. In the optimal control scenario, this algorithm has the task of solving the NLP generated by the OCP discretisation, at each mesh-refinement step computed by GPOPS-II. Considering the following NLP:

$$\min \mathcal{J}(\mathbf{z}) \quad \text{subject to} \quad \begin{cases} \mathbf{b}(\mathbf{z}) = 0 \\ \mathbf{c}(\mathbf{z}) \leq \mathbf{0} \end{cases}, \tag{2.90}$$

the interior-point method computes approximate solutions for a sequence of barrier problems defined as

$$\min \mathcal{J}(\mathbf{z}) - \tau \sum_{i=1}^{n} \ln(s_i) \quad \text{subject to} \quad \begin{cases} \mathbf{b}(\mathbf{z}) = 0 \\ \mathbf{c}(\mathbf{z}) = \mathbf{s} \\ \mathbf{s} \geq 0 \end{cases}, \tag{2.91}$$

for a decreasing sequence of *barrier parameters* τ converging to zero. The lower τ, the higher the sensitivity of the cost function to the effect of infeasibility of the inequality constraints. The Lagrangian becomes:

$$L(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{s}) = \mathcal{J}(\mathbf{z}) - \boldsymbol{\lambda}^T \mathbf{b}(\mathbf{z}) - \boldsymbol{\mu}^T \left(\mathbf{c}(\mathbf{z}) - \mathbf{s}\right) - \tau \sum_{i=1}^{n} \ln(s_i). \tag{2.92}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the Lagrangian multipliers for the equality and inequality

constraints respectively. The condition $\nabla L(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{s}) = 0$ leads to

$$\frac{\partial L}{\partial \mathbf{z}} = \frac{\partial \mathcal{J}}{\partial \mathbf{z}} + \left(\frac{\partial \mathbf{b}}{\partial \mathbf{z}}\right)^T \boldsymbol{\lambda} + \left(\frac{\partial \mathbf{c}}{\partial \mathbf{z}}\right)^T \boldsymbol{\mu} = 0, \tag{2.93}$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} = \mathbf{b}(\mathbf{z}) = 0, \tag{2.94}$$

$$\frac{\partial L}{\partial \boldsymbol{\mu}} = \mathbf{c}(\mathbf{z}) - \mathbf{s} = 0, \tag{2.95}$$

$$\frac{\partial L}{\partial \mathbf{s}} = \boldsymbol{\mu}\,\mathbf{s} - \tau = 0, \tag{2.96}$$

which represent the KKTs of the NLP for $\tau \to 0$. The method described implements an approximated solution of the barrier problem considering a fixed value of τ. The approximate solution of the next barrier problem is obtained from the previous one while decreasing the barrier parameter τ. The *optimality error* for the barrier problem can be defined as:

$$E_\tau(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{s}) \doteq \max\left\{ \frac{\|\nabla \mathcal{J}(\mathbf{z}) + \boldsymbol{\lambda}\,\nabla \mathbf{b}(\mathbf{z}) - \boldsymbol{\mu}\,\nabla \mathbf{c}(\mathbf{z})\|_\infty}{s_d}, \|\mathbf{b}(\mathbf{z})\|_\infty, \frac{\|\,\boldsymbol{\mu}\,\mathbf{s} - \tau\|_\infty}{s_c} \right\}, \tag{2.97}$$

where s_d and s_c are properly defined scaling parameters, that prevent the error to become very large when the multipliers diverge. Evaluating the error at $\tau = 0$ makes possible to obtain the *optimality error* for the original problem and definitely a terminal condition for the whole algorithm. In other terms, the algorithm terminates if the approximate solution $(\mathbf{z}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \mathbf{s}^*)$ satisfies

$$E_0(\mathbf{z}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*, \mathbf{s}^*) \leq \varepsilon_{tol}, \tag{2.98}$$

where $\varepsilon_{tol} > 0$ is a user-defined error tolerance. In interior-point problems, it is essential to implement a good update of the barrier parameter. At step $j+1$, the parameter is

$$\tau_{j+1} = \max\left\{ \frac{\varepsilon_{tol}}{10}, \min\left\{ \kappa_\tau \tau_j, \tau_j^{\theta_\tau} \right\} \right\}, \tag{2.99}$$

where $\kappa_\tau \in (0,1)$ and $\theta_\tau \in (1,2)$. With this update rule, the barrier parameter can be decreased at super-linear rate and an excessive reduction is prevented, avoiding numerical issues. The solution of the barrier problem can be found, for the k-th iteration, adopting a damped Newton's method. The search directions $(\mathbf{d}_k^z, \mathbf{d}_k^\lambda, \mathbf{d}_k^\mu, \mathbf{d}_k^s)$ are generated from a linearisation of the optimality conditions (2.93)-(2.96) at $(\mathbf{z}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k, \mathbf{s}_k)$:

$$\begin{bmatrix} \nabla^2_{xx} L & -\nabla_x \mathbf{b} & -\nabla_x \mathbf{c} & 0 \\ \nabla_x \mathbf{b} & 0 & 0 & 0 \\ \nabla_x \mathbf{c} & 0 & 0 & 1 \\ 0 & 0 & \mathbf{s}_k & \boldsymbol{\mu}_k \end{bmatrix} \begin{pmatrix} \mathbf{d}_k^z \\ \mathbf{d}_k^\lambda \\ \mathbf{d}_k^\mu \\ \mathbf{d}_k^s \end{pmatrix} = \begin{pmatrix} \nabla_x L \\ \mathbf{b}(\mathbf{z}_k) \\ \mathbf{s}_k - \mathbf{c}(\mathbf{z}_k) \\ \boldsymbol{\mu}_k\,\mathbf{s}_k - \tau \end{pmatrix}, \tag{2.100}$$

Considering now the step sizes (or damping factors) α_k, $\alpha_k^\mu \in (0,1]$, it is possible

to evaluate the following iterations as

$$\begin{aligned} \mathbf{z}_{k+1} &\doteq \mathbf{z}_k - \alpha_k \mathbf{d}_k^z, && (2.101)\\ \boldsymbol{\lambda}_{k+1} &\doteq \boldsymbol{\lambda}_k - \alpha_k \mathbf{d}_k^\lambda, && (2.102)\\ \boldsymbol{\mu}_{k+1} &\doteq \boldsymbol{\mu}_k - \alpha_k^\mu \mathbf{d}_k^\mu, && (2.103)\\ \mathbf{s}_{k+1} &\doteq \mathbf{s}_k - \alpha_k^\mu \mathbf{d}_k^s. && (2.104) \end{aligned}$$

The choice of step sizes is made in such a way that

$$\begin{aligned} \alpha_k^{max} &\doteq \max\left\{\alpha \in (0,1] : \mathbf{z}_k + \alpha \mathbf{d}_k^z \geq (1-\tilde{\tau}_j)\mathbf{z}_k\right\}, && (2.105)\\ \alpha_k^{\mu} &\doteq \max\left\{\alpha \in (0,1] : \boldsymbol{\mu}_k + \alpha \mathbf{d}_k^\mu \geq (1-\tilde{\tau}_j)\boldsymbol{\mu}_k\right\}, && (2.106) \end{aligned}$$

where

$$\tilde{\tau} = \max\left\{\tilde{\tau}_{min}, 1-\tau_j\right\}, \quad \tilde{\tau}_{min} \in (0,1). \tag{2.107}$$

The step size α_k is a value of the interval $(0, \alpha_k^{max}]$, determined by a *backtracking line-search* procedure that spans a decreasing sequence of trial steps sizes in the series

$$\alpha_{k,l} = 2^{-l}\alpha_k^{max}, \quad l = 0, 1, 2, \ldots, \tag{2.108}$$

while α_k^μ is the actual step size for updating $\boldsymbol{\mu}$.

In IPOPT, a *line-search filter method* is proposed instead of using *exact penalty merit functions* to enforce progress toward the solution. With the merit functions method, the trial points are accepted if they improve a combination of the objective function and the constraint violation. The typical formulation of a merit function is:

$$\boldsymbol{\Psi}_\beta = \mathcal{J}(\mathbf{z}) - \tau \sum_{i=1}^{n} \ln(s_i) + \frac{\beta}{2} \| \mathbf{b}(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{s}) \|^2 = \boldsymbol{\psi}(\mathbf{z}, \mathbf{s}) + \frac{\beta}{2} \boldsymbol{\theta}(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{s}), \tag{2.109}$$

where $\boldsymbol{\theta}(\mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{s})$ represents the infeasibility of the constraints, $\boldsymbol{\psi}(\mathbf{z}, \mathbf{s})$ is the interior-point cost function and β is a weighting parameter. At each Newton's iteration step, it exists a value of β such that the solution of (2.100) is a descent direction for the merit function (2.109). Therefore, a possible strategy is to choose *alpha* in the backtracking procedure (2.108) in order to ensure that the merit function is decreasing [27]. Filter methods, instead, allow to accept trial points if they improve either the objective function or the constraints violation. Considering a trial point $\mathbf{z}_k(\alpha_{k,l}) = \mathbf{z}_k + \alpha_{k,l}\mathbf{d}_k^z$ during the backtracking line-search procedure. This point is acceptable if either

$$\begin{aligned} &\boldsymbol{\theta}(\mathbf{z}_k(\alpha_{k,l})) \leq (1-\gamma_\theta)\,\boldsymbol{\theta}(\mathbf{z}_k),\\ \text{or} \quad &\boldsymbol{\psi}_j(\mathbf{z}_k(\alpha_{k,l})) \leq \boldsymbol{\psi}_j(\mathbf{z}_k) - \gamma_\psi\,\boldsymbol{\theta}(\mathbf{z}_k), \end{aligned} \tag{2.110}$$

are satisfied for fixed constants $\gamma_\theta, \gamma_\psi \in (0,1)$. If the constraints violation is lower than a given number θ_{min}, a switching condition is provided to ensure a sufficient progress in the barrier objective function. A filter $\mathcal{F}_k$ is defined for each step k, in order to restrict the possibile combinations $(\boldsymbol{\theta}, \boldsymbol{\psi}_j)$ and guarantee a successful

```
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  1.6109693e+01 1.12e+01 5.28e-01   0.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  1.8029749e+01 9.90e-01 6.62e+01   0.1 2.05e+00    -  2.14e-01 1.00e+00f  1
   2  1.8719906e+01 1.25e-02 9.04e+00  -2.2 5.94e-02   2.0 8.04e-01 1.00e+00h  1
```

Figure 2.2: IPOPT console output quantities.

trial point. A trial point is rejected if $(\boldsymbol{\theta}, \boldsymbol{\psi}_j) \in \mathcal{F}_k$. At the initial step, the filter is

$$\mathcal{F}_0 \doteq (\boldsymbol{\theta}, \boldsymbol{\psi}) \in \mathbb{R}^2 \,:\, \boldsymbol{\theta} \geq \theta^{max}, \tag{2.111}$$

where θ^{max} is the largest constraint violation for accepting a trial point. Then, the filter is updated:

$$\mathcal{F}_{k+1} \doteq \mathcal{F}_k \cup \Big\{ (\boldsymbol{\theta}, \boldsymbol{\psi}) \in \mathbb{R}^2 \,:\, \boldsymbol{\theta} \geq (1-\gamma_\theta)\, \boldsymbol{\theta}(\mathbf{z}_k),\, \boldsymbol{\psi} \geq \boldsymbol{\psi}_j(\mathbf{z}_k) - \gamma_\psi\, \boldsymbol{\theta}(\mathbf{z}_k) \Big\}. \tag{2.112}$$

This filters prevents the algorithm to return to the neighbourhood of $\mathbf{z}_k$. Finally, a minimum step size α_k^{min} can be defined for the cases in which the described procedure is not capable of finding a feasible trial step. If the backtracking line-search leads to a trial step size $\alpha_{k,l} \leq \alpha_k^{min}$, the algorithm reverts to a *feasibility restoration phase*. In this case, the algorithm tries to find a new iterate $\mathbf{z}_k$ which results acceptable for the filter $\mathcal{F}_k$ and satisfies (2.110). The restoration fails whether the program converges to a stationary point for the constraints violation, i.e. the problem seems locally infeasible and the constraints violation can not be reduced.

IPOPT provides a console tabular output in order to compute an evaluation for the most important parameters connected to the convergence of the problem; in figure (2.2) can be seen an example of the printed quantities. The meaning of the output parameters can be summarised as follows:

- *iter*: iteration count k;
- *objective*: unscaled objective value at current iteration;
- *inf_pr*: unscaled constraint violation at current iteration, or rather the infinity-norm (maximum) of the constraints;
- *inf_du*: scaled dual infeasibility at current point, or rather the infinity-norm of the internal dual infeasibility expressed in (2.93);
- *lg(mu)*: log_{10} of the barrier parameter τ;
- $\|d\|$: infinity-norm of the primal search direction $\mathbf{d}$;
- *lg(rg)*: depends on log_{10} of regularization of Hessian and Lagrangian;
- *alpha_du*: stepsize of dual variables α_k^z in (2.103);
- *alpha_pr*: stepsize of the primal variables α_k in (2.101);
- *ls*: number of backtracking line-search steps.

2.5.2 ADiGator

The evaluation of the objective function gradient, the constraint Jacobian and the Hessian of the Lagrangian is fundamental in order to obtain a solution for the NLP problem; the standard strategy adopted to evaluate these derivatives is the finite-differencing. Considering a generic function $f(x)$, its derivative can be approximated in an interval of length ε:

$$\left.\frac{df}{dx}\right|_x = \frac{f_{x+\varepsilon/2} - f_{x-\varepsilon/2}}{\varepsilon}. \tag{2.113}$$

The main advantage of this strategy consists in the fact that only evaluations of the function are required; nevertheless the finite-differencing strongly depends on the spacing between the chosen points: with a large spacing the approximation is coarse, while using small spacing may lead to an improper evaluation of the derivative.

An efficient and accurate method to compute the needed derivatives automatically is known as *automatic differentiation* (AD). Taking into consideration a differentiable computer program, it may be broken into a sequence of elementary operations, each one connected to a correspondent derivative rule; applying systematically the chain rule to each of the elementary operation rules, it is possible to achieve a derivative with accuracy coincident with machine precision. ADiGator is a MATLAB-based tool, that provides automatic differentiation. This package generates a MATLAB program for computing the possible non-zero derivatives of the original function program. In an optimal-control scenario, the tool can be easily implemented in GPOPS-II, leading to the following advantages:

1. no time penalties are added with the derivative evaluation because the derivative code is only evaluated by MATLAB library;

2. the tool can compute *vectorised* derivatives of the *vectorised* functions which result from the direct-collocation method provided by GPOPS-II, i.e. it improves the efficiency of the calculation of the OCP solution;

3. second-order derivatives can be provided simply applying recursively the same procedure.

ADiGator makes use of a source transformation via operator overloading in order to determine the derivatives of functions defined in MATLAB programs. In summary, the inputs of the software consist in a function (that has to be differentiated), the information on the sizes of inputs and the derivative information on the inputs; the operator is overloaded because different operators have different implementations depending on their arguments: the program in fact will be able to recall the functions using different input data-type available during the calculations. The algorithm first transforms the given MATLAB source-code into an intermediate code in which are included calls to *ADiGator* specific routines for transformation; this code is then evaluated multiple times using overloaded CADA objects, which include only information on the size of objects, symbolic identifiers and the possible nonzero derivative locations. The evaluations and the

information obtained are included in functions and derivative files. As result, the method provides a transformation of the original user-function into a MATLAB function that computes a numerical solution of the non-zero derivatives of the original function. All these topics are extensively discussed in [28, 29].

Chapter 3

Vehicle modelling

3.1 Introduction

The model development is a fundamental part of the minimum-time-simulation framework. The complexity of the model strongly affects the computational effort and, in some cases, even the possibility of achieving the convergence of the algorithm. In other words, a compromise between the accuracy of the results and the simulation time need be taken into account. Moreover, it is important to find an efficient mathematical representation of the vehicle, in which the constraints are smooth enough to ensure differentiability. Two different strategies will be discussed in this chapter: the steady-state vehicle models, in which only stationary effects are included, and the dynamic vehicle models, in which the transient effects are involved. Both these types of models have been applied to minimum-time problems and many examples are available in literature. In [30], a steady-state race car model is employed, including a friction-ellipse tyre model, together with a simple engine model, limited braking power, aerodynamic forces and the stiffness of the suspensions, the anti-roll-bars and the chassis. This formulation is employed in a quasi-steady-state fixed-trajectory simulation. In [31] and [32] a seven-degrees-of-freedom double-track model is used for assessing minimum-lap-time with quasi-steady-state fixed-trajectory simulations, while in [33] and [34] the same vehicle model is employed (neglecting the steady-state assumptions) for free-trajectory dynamic simulations using optimal-control methods. The chassis is represented by a rigid body with three degrees-of-freedom, while four degrees-of-freedom are related to the wheels spin. The suspensions are neglected, as well as the roll, pitch and heave motions. The Pacejka Magic Formula is used for the tyre model, and the force combination is obtained by means of weighting functions. The aerodynamic drag force is applied in the centre of gravity, while the downforce is applied considering the force distribution between the two axles. A parallel steering geometry is included. The powertrain model is based on experimental engine maps, while the gear ratio is computed from the vehicle speed. A speed-sensing differential is also considered for the rear-axle torque distribution. A similar model is employed in [35] for transient and steady-state simulations. In this case, the tyre model is obtained from a simplified Magic Formula, with a reduced number of coefficients. This model allows to compute analytically the force-peak

position for scaling the slip values. A more complex multibody car model is also implemented for simulating short manoeuvres, including a more accurate representation of the chassis for taking into account the roll, pitch and heave motions of the vehicle. The suspension model consists in four-bar-linkages for the front and rear axles, together with their springs and dampers. Also the unsprung masses and the tyre bodies are included. In [36] a steady-state double-track model is employed in a fixed-trajectory simulation. In this case, the vertical dynamics is modelled through a lookup-table, which computes the aerodynamic forces and the chassis motions from the tyre-normal loads. The TMeasy tyre model is included, together with a tyre-temperature model. A seven-degrees-of-freedom double-track car model is also discussed in [37] for the computation of g-g diagrams. The chassis can move in the road plane and the wheels spin is considered, while the pitch, roll and heave motions are neglected. A combined Magic Formula is employed for tyre modelling. The powertrain in taken into account through lookup tables, while a torque-sensing differential model is included. In [38] a double-track vehicle model is employed for free-trajectory dynamic simulations using an optimal-control approach. The model has seven degrees-of-freedom, which consist in the translations and yaw motion on the road plane plus the wheel spin. A speed-sensing differential model is also included, together with a constraint for avoiding wheel-locking during braking. The same dynamic model is used in [39] for computing the free-trajectory with an optimal-control approach, while including an energy recovery model for the power-unit. In the literature related to motorcycle simulations the modelling development is even more challenging due to the intrinsic instability of these vehicles. A milestone in the minimum-time approaches can be found in [4], where a simple dynamic model is implemented for computing a free-trajectory optimal-control simulation. The vehicle has four degrees-of-freedom, namely the longitudinal, lateral, yaw and roll motions. The gyroscopic effects are included, although both the suspensions and the rider motion are neglected. A linear tyre-model is employed together with the tyre relaxation-equation for including the transient behaviour. In [40] a comprehensive dynamic motorcycle model is implemented for the analysis of the effect of engine-spin direction. The model includes nine degrees-of-freedom, which consist in four chassis motions, steering angle, wheels spin and suspension travels. A linear tyre model is employed, together with friction ellipses for estimating the force coupling and relaxation equations for emulating the dynamics effects. The powertrain is taken into account for simulating the gear changes, while the engine is modelled though lookup tables. A very essential motorcycle model is presented in [41]. This model describes a steady-state single-track model, in which the roll motion is considered. The tyre performance is limited by constant friction ellipses, while the steering and the suspension travels are neglected. Optimal braking performance is also considered, together with roll-dependent wheelie and stoppie limits. This formulation allows to compute the g-g envelope of the vehicle analytically.

This chapter summarises the main modelling strategies implemented in the simulations described in Ch. 5. A double-track steady-state car model will be discussed in Sec. 3.2. This model is used for the g-g diagrams generation by employing nonlinear-optimisation techniques (see Ch. 4) and, consequently, for com-

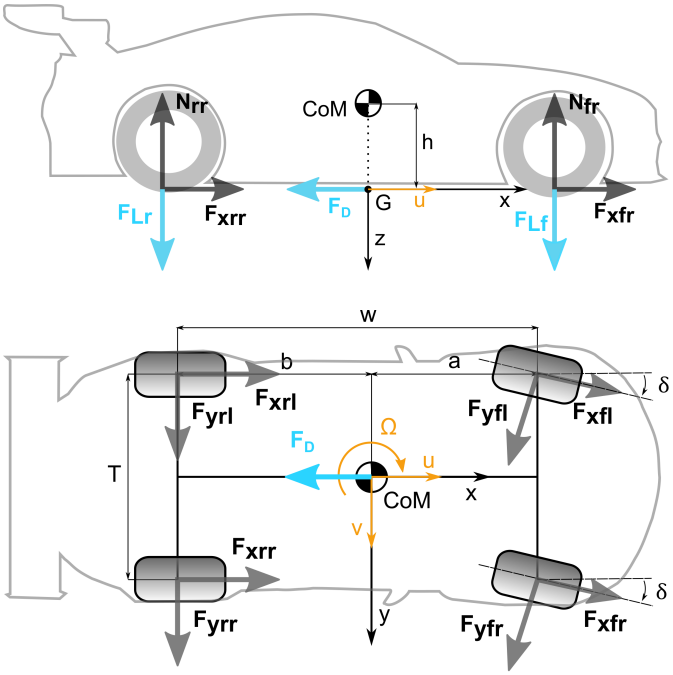


Figure 3.1: Race car model employed in quasi-steady-state simulations.

puting quasi-steady-state minimum-time simulations. In Sec. 3.3, the motorcycle model presented in [41] is modified to take into account for the drag force and the engine power limit. The model allows to compute the g-g-speed diagrams analytically (Ch. 4). The use of this simple model in minimum-time problems (Ch. 5) represents a novel approach to motorcycle simulations, since it retains the main features that characterise the dynamics of a motorcycle while requiring a low computational effort. Finally, in Sec. 3.4 the previously discussed steady-state car model is extended to a dynamic formulation for including the transient effects. This model is employed in a free-trajectory optimal-control simulation (see Ch. 5), which is used for assessing the effect of steering geometry on vehicle performance. For this reason, an in-depth discussion is carried out in this chapter, for listing and comparing the most widespread steering-geometry definitions.

3.2 Steady-state car model

A steady-state car model is fundamental for the computation of the g-g-V diagrams discussed in Ch. 4. In this case, a RWD double-track car model is employed for assessing the race car performance under steady-state conditions. The baseline parameters of the vehicle and their related meaning are shown in Tab. 3.1, together with the tyre coefficients used to compute the tyre forces. The wheel-steering angle δ is assumed small, the roll, pitch and bounce degrees of freedom are neglected, together with the suspension travels. The steady-state equations of the vehicle

are given by

$$
\begin{aligned}
ma_x &= (F_{xfl} + F_{xfr} + F_{xrl} + F_{xrr}) - (F_{yfl} + F_{yfr})\,\delta - F_D, && (3.1)\\
ma_y &= (F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr}) + (F_{xfl} + F_{xfr})\,\delta, && (3.2)\\
0 &= mg + F_{Lf} + F_{Lr} - N_{fl} - N_{fr} - N_{rl} - N_{rr}, && (3.3)\\
ma_y h &= \frac{T}{2}(N_{fl} - N_{fr} + N_{rl} - N_{rr}), && (3.4)\\
ma_x\, h &= aF_{Lf} - bF_{Lr} - a(N_{fl} + N_{fr}) + b(N_{rl} + N_{rr}), && (3.5)\\
0 &= \frac{T}{2}\,(F_{yfl} - F_{yfr})\,\delta - a\,(F_{xfl} + F_{xfr})\,\delta + \frac{T}{2}\,(-F_{xfl} + F_{xfr} - F_{xrl} + F_{xrr}) + \\
&\quad - a\,(F_{yfl} + F_{yfr}) + b\,(F_{yrl} + F_{yrr})\,, && (3.6)
\end{aligned}
$$

where the first three equations represent the force balance along the longitudinal, lateral and vertical direction respectively, while the latter three equations represent the moment balance around the roll, pitch and yaw axes through the projection of the centre of mass on the ground. In (3.1) and (3.6), the longitudinal acceleration is $a_x = \dot{u} - \Omega v$, where u and v are the longitudinal and lateral velocity of the vehicle respectively ($V = \sqrt{u^2 + v^2}$ is the total velocity), while Ω is the yaw rate. In (3.2) and (3.5), the lateral acceleration is $a_y = \Omega u$. It should be noticed that, due to the steady-state assumptions, the lateral acceleration $\dot{v}$ (computed in the reference frame fixed to the car) and the yaw acceleration $\dot{\Omega}$ are zero. The aerodynamics forces consist in the drag force F_D, the front F_{Lf} and rear F_{Lr} downforces. They are applied on the road plane (see Fig. 3.1), and are given by[1]

$$
F_D = \frac{1}{2}\,\rho_a\, C_D A\, u^2, \qquad F_{Lf} = \frac{1}{2}\,\rho_a\, C_{Lf} A\, u^2, \qquad F_{Lr} = \frac{1}{2}\,\rho_a\, C_{Lr} A\, u^2. \tag{3.7}
$$

The tyre longitudinal and lateral forces are given by F_{xij} and F_{yij} respectively, where $i = f, r$ (front, rear) and $j = l, r$ (left, right). The tyre forces are computed employing the Magic Formula with theoretical slips [43]:

$$
\begin{aligned}
F_x &= N\,\frac{\sigma_x}{\sigma}\,D_x \sin\{C_x\,\arctan\left[B_x\,\sigma - E_x\,(B_x\,\sigma - \arctan(B_x\,\sigma))\right]\}, && (3.8)\\
F_y &= N\,\frac{\sigma_y}{\sigma}\,D_y \sin\{C_y\,\arctan[B_y\,\sigma - E_y\,(B_y\,\sigma - \arctan(B_y\,\sigma))]\}, && (3.9)
\end{aligned}
$$

with

$$
\sigma_x = \frac{\kappa}{1+\kappa}, \quad \sigma_y = \frac{\tan(\lambda)}{1+\kappa}, \quad \sigma = \sqrt{\sigma_x^2 + \sigma_y^2}, \tag{3.10}
$$

where the theoretical slips σ_x, σ_y and σ are computed from the longitudinal and lateral (practical) slips κ and λ [43]. The lateral slips λ_{ij} of the current model are

$$
\begin{aligned}
\lambda_{fl} &= \delta - \frac{v + \Omega a}{u + \frac{T}{2}\Omega}, \qquad \lambda_{fr} = \delta - \frac{v + \Omega a}{u - \frac{T}{2}\Omega}, && (3.11)\\
\lambda_{rl} &= -\frac{v - \Omega b}{u + \frac{T}{2}\Omega}, \qquad \lambda_{rr} = -\frac{v - \Omega b}{u - \frac{T}{2}\Omega}. && (3.12)
\end{aligned}
$$

[1] Selecting to apply the aerodynamic forces on the road plane is consistent to the methods used for their measurement in the wind-tunnel using strain-gauge balances [42]. For this reason, the moment due to the actual drag-force (applied to the centre of pressure) is taken into account in the downforce coefficients.

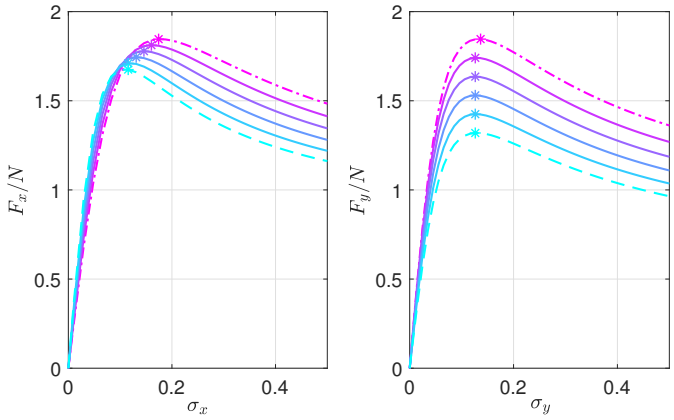


Figure 3.2: Longitudinal (left) and lateral (right) tyre forces for unit normal load as a function of the (theoretical) longitudinal σ_x and lateral σ_y slip respectively, for normal loads between 500 N (dash-dot-magenta) and 5500 N (dashed-cyan), with steps of 1000 N. Stars represent the curve peaks.

The longitudinal slip will be an input to the model. Finally, B, C, D and E are the Pacejka's coefficients, which are herein expressed in the simplified form introduced in [44]. The purpose is to avoid the full Magic Formula formulation, while retaining some of its key features, namely load dependent friction coefficients, load-dependent position of the peak of the force vs. slip curves, and load dependent cornering stiffness per unit load. The coefficients are

$$K_x = B_x\, C_x\, D_x = N\, pK_{x1} \exp(pK_{x3}\, df_z), \tag{3.13}$$
$$E_x = pE_{x1}, \tag{3.14}$$
$$D_x = (pD_{x1} + pD_{x2}\, df_z)\, \lambda_{\mu,x}, \tag{3.15}$$
$$C_x = pC_{x1}, \tag{3.16}$$
$$B_x = \frac{K_x}{C_x\, D_x\, N}, \tag{3.17}$$
$$K_y = N_0\, pK_{y1}\, \sin\left(2\, \arctan \frac{N}{pK_{y2}\, N_0}\right), \tag{3.18}$$
$$E_y = pE_{y1}, \tag{3.19}$$
$$D_y = (pD_{y1} + pD_{y2}\, df_z)\, \lambda_{\mu,y}, \tag{3.20}$$
$$C_y = pC_{y1}, \tag{3.21}$$
$$B_y = \frac{K_y}{C_y\, D_y\, N}, \tag{3.22}$$

where $df_z = \frac{N-N_0}{N_0}$ and N_0 is a reference load. The dataset of tyre parameters is given in Tab. 3.1, while the resulting tyre forces along the theoretical slips are shown in Fig. 3.2 at increasing normal loads N. The tyre normal forces are given by N_{ij}, where again $i = f, r$ and $j = l, r$ (see Fig. 3.1), and are computed from the system consisting of (3.3), (3.4) and (3.5), with the roll stiffness balance

$$ma_y \frac{h}{T}\xi = \frac{N_{fl} - N_{fr}}{2}, \tag{3.23}$$

where $\xi = \frac{K_{\phi f}}{K_{\phi f}+K_{\phi r}}$ is the roll stiffness ratio, $K_{\phi f}$ is the front-axle roll stiffness, and $K_{\phi r}$ is the rear-axle roll stiffness.

The total driving force F_x is split between the rear and front axle according to the distribution factor k_t, under the open-differential assumption:

$$F_{xfl} = \frac{(1-k_t)F_x}{2}, \qquad F_{xfr} = \frac{(1-k_t)F_x}{2}, \tag{3.24}$$

$$F_{xrl} = \frac{k_t F_x}{2}, \qquad F_{xrr} = \frac{k_t F_x}{2}. \tag{3.25}$$

During acceleration $k_t = 1$ for rear-wheel-drive (RWD) vehicles, $k_t = 0$ for front-wheel-drive (FWD) vehicles, and $0 < k_t < 1$ for all-wheel-drive (AWD) vehicles. During braking the distribution factor is given by

$$k_t = \frac{1}{1+\gamma}, \qquad \gamma = \frac{F_{xfl}+F_{xfr}}{F_{xrl}+F_{xrr}}, \tag{3.26}$$

where γ is the brake ratio, which is here defined as the ratio between the front and rear longitudinal tyre forces – the switch between the value of k_t in acceleration and the value of k_t in braking is implemented through an approximation of a regularised piecewise function, in order to avoid numerical issues.

Summarising, given the longitudinal velocity u, the longitudinal acceleration a_x, and the lateral acceleration a_y, the vehicle trim can be computed by solving the steady-state equations (3.1)-(3.6), the roll balance (3.23), the brake ratio (3.26), the open-differential conditions $F_{xfl} = F_{xfr}$ and $F_{xrl} = F_{xrr}$, for the four normal loads N_{ij}, the steering angle δ, the four tyre longitudinal slips κ_{ij}, and the lateral velocity v.

3.3 Steady-state motorcycle model

The g-g-V diagrams of the motorcycle are computed employing the essential model reported in [41], which is extended to include the drag force and the engine power limit. Both the suspensions and the steer angle are neglected. As in the car model, the yaw rate Ω is related to the lateral acceleration a_y and the velocity V by $\Omega = a_y/V$, under steady-state assumptions. In order to obtain a compact formulation, it is further assumed that the vehicle sideslip angle is small, that the motorcycle is symmetric with respect to the x-z plane (i.e. $I_{xy} = I_{yz} = 0$), that the cross moment of inertia I_{xz} is negligible, and that $I_y \approx I_z$. The benefit of such model is that a compact formulation is obtained, while retaining motorcycle peculiarities such as roll-dependent stoppie and wheelie limits.

The steady-state equations of motion are:

$$ma_x = F_{xr} + F_{xf} - F_D \tag{3.27}$$

$$ma_y = F_{yr} + F_{yf} \tag{3.28}$$

$$mg = N_r + N_f \tag{3.29}$$

$$ma_y h\cos\phi = mgh\sin\phi \tag{3.30}$$

$$ma_x h\cos\phi = bN_r - (w-b)N_f - F_D h_a\cos\phi \tag{3.31}$$

$$ma_x h\sin\phi = bF_{yr} - (w-b)F_{yf} - F_D h_a\sin\phi, \tag{3.32}$$

Table 3.1: Baseline car parameters and tyre coefficients.

Symbol	**Description**	**Value**
g	gravity	$9.81\,\mathrm{m/s^2}$
ρ_a	air density	$1.20\,\mathrm{kg/m^3}$
m	mass	$1300\,\mathrm{kg}$
h	height of centre of mass	$0.330\,\mathrm{m}$
w	wheelbase	$2.900\,\mathrm{m}$
b	longitudinal distance of the centre of mass	$1.535\,\mathrm{m}$
a	$w - b$	$1.365\,\mathrm{m}$
I_z	yaw inertia	$1700\,\mathrm{kgm^2}$
T	vehicle track	$2.016\,\mathrm{m}$
γ_0	brake ratio F_{xf}/F_{xr}	1.13
ξ_0	roll stiffness ratio	0.53
$C_D A$	drag area coefficient	$0.65\,\mathrm{m^2}$
$C_{Lf} A$	front lift area coefficient	$0.15\,\mathrm{m^2}$
$C_{Lr} A$	rear lift area coefficient	$0.35\,\mathrm{m^2}$
$P_{\max}$	maximum power	$415\,\mathrm{kW}$
$\delta_{\max}$	maximum wheel steer angle	$20\,\mathrm{deg}$
$pCx1$	longitudinal shape factor	1.6935
$pDx1$	max longitudinal friction coefficient	1.8757
$pDx2$	longitudinal friction load dependency factor	-0.127
$pEx1$	longitudinal curvature factor	0.07708
$pKx1$	max longitudinal stiffness coefficient	30.5
$pKx3$	max longitudinal stiffness coefficient	0.2766
λ_{μ_x}	longitudinal scaling factor	0.93
$pCy1$	lateral shape factor	1.733
$pDy1$	max lateral friction coefficient	1.8217
$pDy2$	lateral friction load dependency factor	-0.4388
$pEy1$	lateral curvature factor	0.29446
$pKy1$	max cornering stiffness coefficient	44.2
$pKy2$	max cornering stiffness coefficient	2.5977
λ_{μ_y}	lateral scaling factor	0.84
N_0	nominal load (where $df_z = 0$)	$3500\,\mathrm{N}$

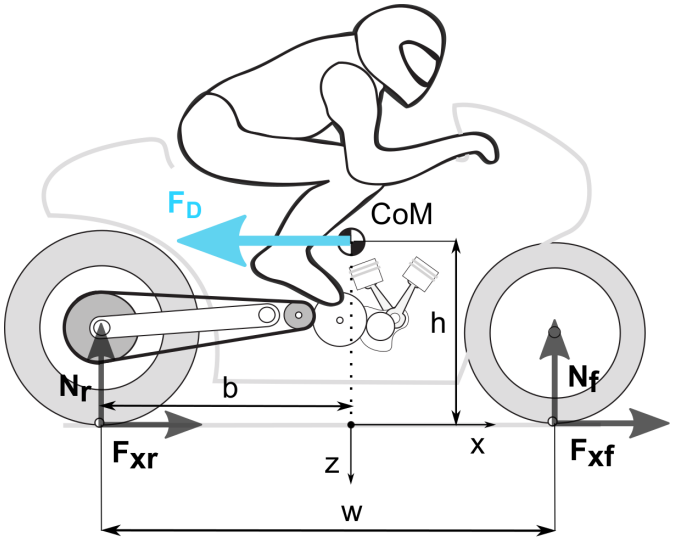


Figure 3.3: Motorcycle model.

where the first three equations represent the balance of forces along the longitudinal, lateral and vertical axes, while the latter three equations represent the moment balance around the roll, pitch and yaw axes. The meaning of the model parameters and values used in the simulations is reported in Tab. 3.2, while the forces and the axes convention are depicted in Fig. 3.3. The lateral forces are represented by F_{yr} and F_{yf} for the rear and front tyre respectively, while the roll angle ϕ is measured from the vertical plane x-z. During braking, the longitudinal forces F_{xr} and F_{xf} are related by

$$\gamma = \frac{F_{xf}}{F_{xr}}, \tag{3.33}$$

where the brake ratio γ (compare with (3.26) where γ was defined for cars) is zero during acceleration and is determined by the rider in braking. The seven steady-state equations (3.27)-(3.33) can be solved analytically for $\phi, N_r, N_f, F_{xr}, F_{xf}, F_{yr}, F_{yf}$. Such solution will be used to derive the g-g boundaries (Ch. 4).

The aerodynamic-drag force is applied on the centre of pressure, and is given by

$$F_D = \frac{1}{2}\rho_a C_D A V^2. \tag{3.34}$$

The coupling between longitudinal and lateral forces is included through friction ellipses, which (at the limit of friction) are given by

$$\left(\frac{F_{xr}/N_r}{\mu_x}\right)^2 + \left(\frac{F_{yr}/N_r}{\mu_y}\right)^2 = 1, \tag{3.35}$$

$$\left(\frac{F_{xf}/N_f}{\mu_x}\right)^2 + \left(\frac{F_{yf}/N_f}{\mu_y}\right)^2 = 1, \tag{3.36}$$

where μ_x and μ_y are the longitudinal and lateral friction coefficients respectively.

The maximum acceleration, i.e. the upper boundary of the g-g map, is obtained when introducing the analytical solution of the steady-state equations (3.27)-(3.33) into the rear friction ellipse (3.35), to give

$$\left(\frac{w(ma_x + F_D)\sqrt{a_y^2 + g^2}}{g((w-b)m\sqrt{a_y^2 + g^2} + ma_x h + F_D h_a)}\right)^2 \frac{1}{\mu_x^2} + \left(\frac{a_y}{g}\right)^2 \frac{1}{\mu_y^2} = 1, \tag{3.37}$$

which can be solved for the longitudinal acceleration a_x at different lateral accelerations a_y, using a standard root-finding solver.

Since the maximum engine power $P_{\max}$ is given, the longitudinal force balance (3.27) can be used to compute the related acceleration limit

$$a_x = \frac{P_{\max}}{mV} - \frac{F_D}{m}. \tag{3.38}$$

Differently from cars, the front tyre of a motorcycle can lift the road during acceleration (wheelie condition). The related acceleration limit is obtained from the condition $N_f = 0$, which can be solved analytically to give

$$a_x = \frac{b\sqrt{a_y^2 + g^2}}{h} - \frac{F_D h_a}{mh}. \tag{3.39}$$

In order to compute the minimum deceleration, it is assumed that the rider applies an optimal brake ratio: this condition occurs when the front and rear tyres are engaged equally, i.e. $F_{xr}/N_r = F_{xf}/N_f$. The condition (3.35)=(3.36) gives the following (longitudinal- and lateral-acceleration dependent) optimal brake ratio

$$\gamma_{opt} = \frac{mb\sqrt{a_y^2 + g^2} - ma_x h - F_D h_a}{m(w-b)\sqrt{a_y^2 + g^2} + ma_x h + F_D h_a}. \tag{3.40}$$

When introducing the analytical solution of the steady-state equations (3.27)-(3.33), together with (3.40) into either the front or rear friction ellipse (3.36), one obtains

$$-a_x = g\mu_x\sqrt{1 - \left(\frac{a_y}{g}\right)^2 \frac{1}{\mu_y^2}} + \frac{F_D}{m}. \tag{3.41}$$

Again, differently from cars, the rear tyre of a motorcycle can lift the road while braking (stoppie condition). The related deceleration limit is obtained from the condition $N_r = 0$, which can be solved analytically to give

$$-a_x = \frac{(w-b)\sqrt{a_y^2 + g^2}}{h} - \frac{F_D h_a}{mh}. \tag{3.42}$$

In sum, the g-g diagram is obtained by the envelope of (3.37), (3.38), (3.39), (3.41), and (3.42). The resulting g-g diagrams of the motorcycle (at different speeds), obtained with the dataset in Tab. 3.2, are discussed in Ch. 4.

Table 3.2: Motorcycle (plus rider) parameters.

Symbol	Description	Value
g	gravity	$9.81\,\mathrm{m/s^2}$
ρ_a	air density	$1.20\,\mathrm{kg/m^3}$
m	mass	$250\,\mathrm{kg}$
h	height of centre of mass	$0.69\,\mathrm{m}$
h_a	height of centre of pressure	$0.69\,\mathrm{m}$
w	wheelbase	$1.50\,\mathrm{m}$
b	longitudinal distance of the centre of mass	$0.73\,\mathrm{m}$
$C_D A$	drag area coefficient	$0.20\,\mathrm{m^2}$
$P_{\max}$	maximum power	$180\,\mathrm{kW}$
μ_x	longitudinal friction coefficient	1.2
μ_y	lateral friction coefficient	1.44

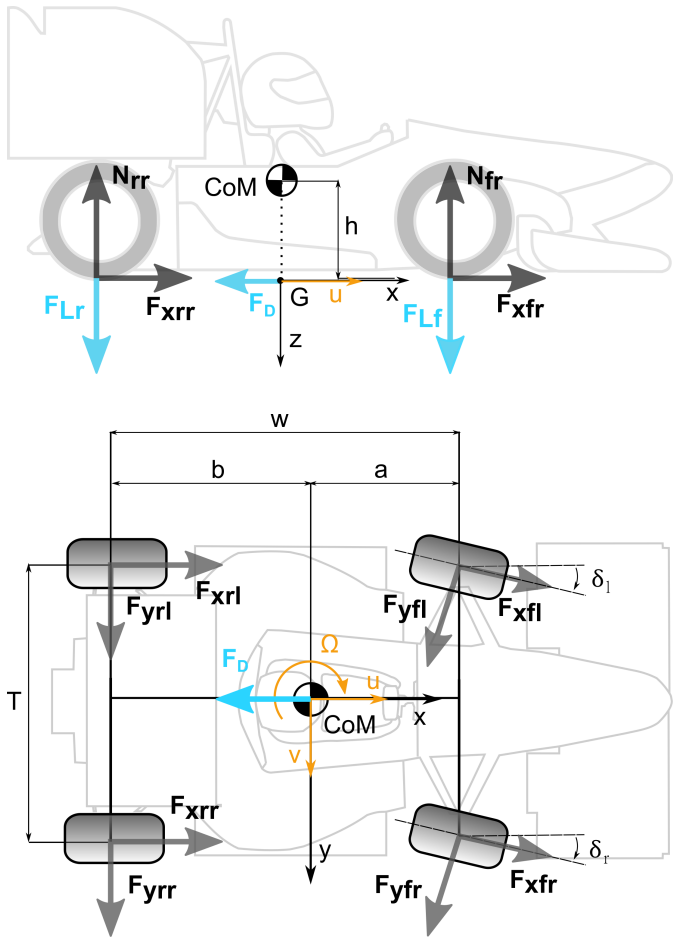


Figure 3.4: Dynamic car model, including different front steering angles.

3.4 Dynamic car model

A dynamics car model is derived from the steady-state RWD double-track model discussed in Sec. 3.2 for the implementation in the optimal-control dynamic simulation (see Ch. 5). In this case, no steady-state assumptions are included in order to take into account for the transient effects. As for the steady-state model, the roll, pitch and bounce degrees of freedom are neglected, together with the suspension travels. In this formulation different left and right steering angles can be employed at the front wheels: this feature is implemented for imposing a specific steering kinematics (see Ch. 5, in which the dynamic model performance is analysed for different steering systems). The main model quantities are depicted in Fig. 3.4. The model parameters are reported in Tab. 3.3 and describe an FSAE race car. The dynamic equations are:

$$
\begin{aligned}
ma_x &= F_{xfl}\cos\delta_{fl} + F_{xfr}\cos\delta_{fr} - F_{yfl}\sin\delta_{fl} - F_{yfr}\sin\delta_{fr} + \\
&\quad F_{xrl}\cos\delta_{rl} + F_{xrr}\cos\delta_{rr} - F_{yrl}\sin\delta_{rl} - F_{yrr}\sin\delta_{rr} - F_D, && (3.43)\\
ma_y &= F_{xfl}\sin\delta_{fl} + F_{xfr}\sin\delta_{fr} + F_{yfl}\cos\delta_{fl} + F_{yfr}\cos\delta_{fr} + \\
&\quad F_{xrl}\sin\delta_{rl} + F_{xrr}\sin\delta_{rr} + F_{yrl}\cos\delta_{rl} + F_{yrr}\cos\delta_{rr}, && (3.44)\\
0 &= mg + F_{Lf} + F_{Lr} - N_{fl} - N_{fr} - N_{rl} - N_{rr}, && (3.45)\\
ma_y h &= \frac{T}{2}(N_{fl} - N_{fr} + N_{rl} - N_{rr}), && (3.46)\\
ma_x h &= aF_{Lf} - bF_{Lr} - a(N_{fl} + N_{fr}) + b(N_{rl} + N_{rr}), && (3.47)\\
I_z\dot{\Omega} &= \frac{T}{2}\left(F_{xfl}\cos\delta_{fl} - F_{xfr}\cos\delta_{fr} - F_{yfl}\sin\delta_{fl} + F_{yfr}\sin\delta_{fr}\right) - \\
&\quad \frac{T}{2}\left(F_{xrl}\cos\delta_{rl} + F_{xrr}\cos\delta_{rr} - F_{yrl}\sin\delta_{rl} - F_{yrr}\sin\delta_{rr}\right) + \\
&\quad a\left(F_{xfl}\sin\delta_{fl} + F_{xfr}\sin\delta_{fr} + F_{yfl}\cos\delta_{fl} + F_{yfr}\cos\delta_{fr}\right) - \\
&\quad b\left(F_{xrl}\sin\delta_{rl} + F_{xrr}\sin\delta_{rr} + F_{yrl}\cos\delta_{rl} + F_{yrr}\cos\delta_{rr}\right), && (3.48)
\end{aligned}
$$

where the first three equations represent the force balance along the longitudinal, lateral and vertical direction respectively, while the latter three equations represent the moment balance around the roll, pitch and yaw axes through the projection of the centre of mass on the ground. In (3.43) and (3.47), the longitudinal acceleration is $a_x = \dot{u} - \Omega v$, where u and v are the longitudinal and lateral velocity of the vehicle respectively ($V = \sqrt{u^2 + v^2}$ is the total velocity), while Ω is the yaw rate. In (3.44) and (3.46), the lateral acceleration is $a_y = \dot{v} + \Omega u$. It should be noticed that, in this case, the steady-state conditions $\dot{v} = 0$ and $\dot{\Omega} = 0$ are neglected. The aerodynamic forces consist of the drag force F_D and the front and rear downforces, F_{Lf} and F_{Lr}. They are applied on the road plane (see Fig. 3.4), and are given by (3.7). The tyre longitudinal and lateral forces are given by F_{xij} and F_{yij} respectively, where $i = f, r$ (front, rear) and $j = l, r$ (left, right). The tyre forces are computed employing the Magic Formula with theoretical slips [43] and including the effect of camber on the lateral force:

$$
\begin{aligned}
F_x &= N\,\frac{\sigma_x}{\sigma}\,D_x \sin\{C_x\,\arctan\left[B_x\,\sigma - E_x\left(B_x\,\sigma - \arctan(B_x\,\sigma)\right)\right]\}, && (3.49)\\
F_y &= N\,\frac{\sigma_y}{\sigma}\,D_y \sin\{C_y\,\arctan[B_y\,\sigma - E_y\left(B_y\,\sigma - \arctan(B_y\,\sigma)\right)]\} + SV_y,
\end{aligned}
$$

with

$$\sigma_x = \frac{\kappa}{1+\kappa}, \quad \sigma_y = \frac{\tan(\lambda)}{1+\kappa}, \quad \sigma = \sqrt{\sigma_x^2 + \sigma_y^2}, \tag{3.50}$$

where the theoretical slips σ_x, σ_y and σ are computed from the longitudinal and lateral (practical) slips κ and λ:

$$\lambda_{fl} = \delta_{fl} - \frac{v + \Omega a}{u + \frac{T}{2}\Omega}, \qquad \lambda_{fr} = \delta_{fr} - \frac{v + \Omega a}{u - \frac{T}{2}\Omega}, \tag{3.51}$$

$$\lambda_{rl} = \delta_{rl} - \frac{v - \Omega b}{u + \frac{T}{2}\Omega}, \qquad \lambda_{rr} = \delta_{rr} - \frac{v - \Omega b}{u - \frac{T}{2}\Omega}. \tag{3.52}$$

The steering angles on each wheel δ_{ij}, where $i = f, r$ (front, rear) and $j = l, r$ (left, right), can be computed as

$$\delta_{fl} = \delta_l - \tau_f, \qquad \delta_{fr} = \delta_r + \tau_f, \tag{3.53}$$

$$\delta_{rl} = -\tau_r, \qquad \delta_{rr} = \tau_r, \tag{3.54}$$

where δ_l and δ_r are the left and right wheel-steering angles in a zero-toe configuration, while τ_f and τ_r are the toe angles at the front and rear tyres respectively (positive for toe-out). The longitudinal slip will be an input to the model. Similarly to the steady-state model (Sec. 3.2), the Pacejka's coefficients B, C, D and E are obtained using (3.14)-(3.22), while the camber-dependent term SV_y is given by

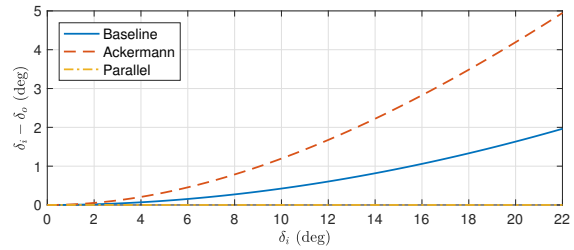
$$SV_y = N(pV_{y3} + pV_{y4}df_z)\phi\lambda_{\mu,y}, \tag{3.55}$$

where $df_z = (N - N_0)/N_0$, N_0 is a reference load and ϕ is the wheel-camber angle. A positive camber angle produces negative forces on the right tyres and positive forces on the left tyres, i.e. the top of the tyres is farther from the vehicle than the case of negative camber. The dataset of tyre parameters is given in Tab. 3.3. The tyre normal forces are given by N_{ij}, where again $i = f, r$ and $j = l, r$ (see Fig. 3.4), and are computed from the system consisting of (3.45), (3.46) and (3.47), with the roll stiffness balance obtained in (3.23). Similarly to the steady-state model, the total driving force F_x is split between the rear and front axle according to the distribution factor k_t, under the open-differential assumption, using the (3.24)-(3.25).

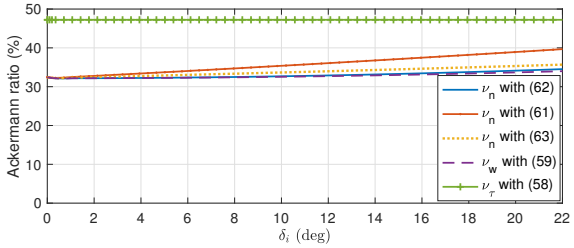
3.4.1 Steering geometry definitions

The possibility of choosing a specific steering kinematics when employing the discussed dynamic model, opens the problem of defining different steering geometries. In this section the main definitions available in literature are discussed and analysed, following the comparison work carried out in [6], where the Ackermann and parallel steering geometries are compared for an FSAE race vehicle.

The terms 'Ackermann steering' or 'Ackermann geometry' refer to the

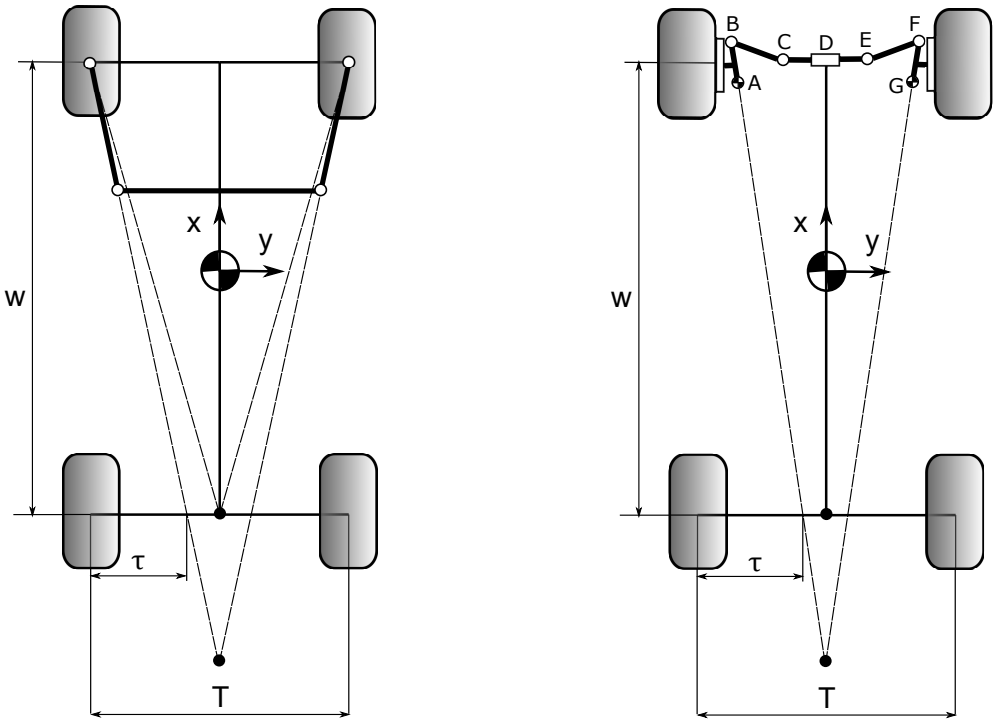


(a)



(b)

Figure 3.5: a) Comparison between the different steering kinematic characteristics selected. FSAE current configuration (solid), Ackermann (dashed) and parallel (dash-dot). b) Comparison between different Ackermann ratio definitions.



(a) (b)

Figure 3.6: a) Approximation of the Ackermann steering geometry using a four-bar linkages. b) Steering linkage of the FSAE vehicle under investigation.

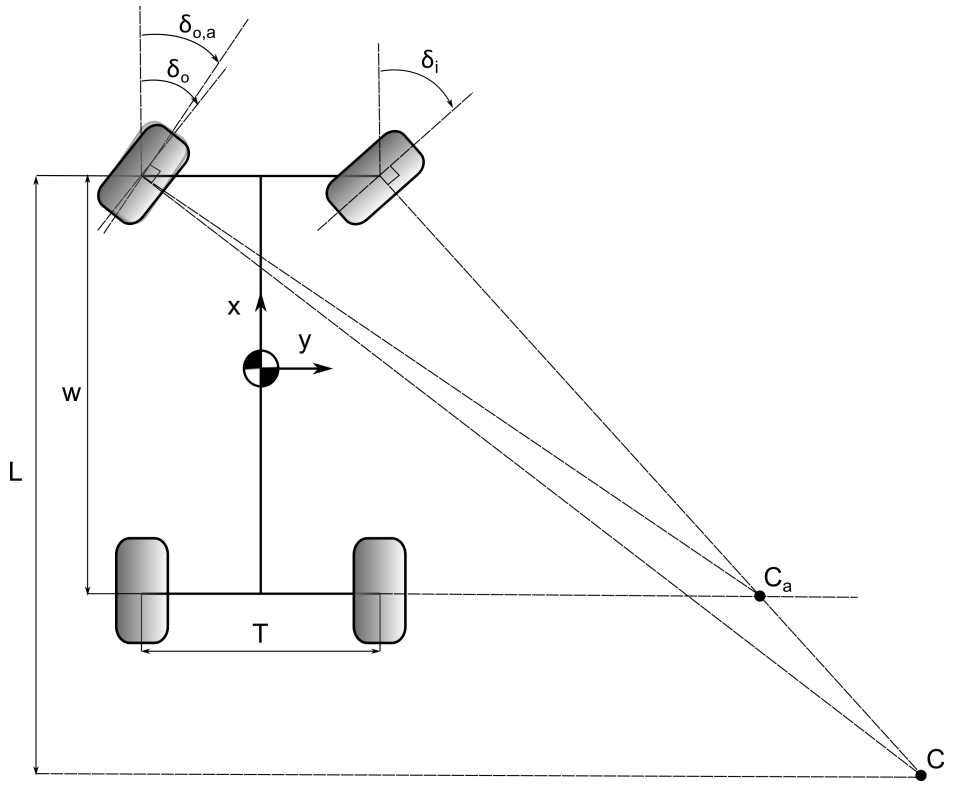


Figure 3.7: Actual steering vs. Ackermann steering.

> steering system kinematics where the wheel-spin axes of the steerable wheels intersect the centre of rotation, for non-zero steering-wheel angles at negligible lateral accelerations [...] A steering system with Ackermann geometry is said to be 100 % Ackermann, and one with equal steer angles (parallel steer) on the steerable axle is said to be 0 % Ackermann. [45]

A standard and well-known derivation give the following relationship between the inner δ_i and outer δ_o wheel angles in the case of Ackermann steering

$$\frac{1}{\tan \delta_o} - \frac{1}{\tan \delta_i} = \frac{T}{w}, \tag{3.56}$$

where δ_o is the steering angle of the outer wheel, δ_i is the steering angle of the inner wheel, T is the track width and w is the wheelbase.

For example, the steering configuration of the FSAE car model (see Tab. 3.3 for the main parameters) is compared with the Ackermann and the parallel steering geometries in Fig. 3.5a. The difference between the inner and outer wheel steering angles $\delta_i - \delta_o$ is shown at different internal wheel steering angles δ_i. When $\delta_i - \delta_o$ is negative (which never happens with the current setup), the geometry is said reverse-Ackermann [46, 47]. There are a number of different definitions in the literature when it comes to comparing the actual steering against the Ackermann steering. The most common definitions will now be concisely reviewed.

In 1878 C. Jeantaud proposed to approximate the Ackermann steering with a four-bar linkage mechanism that has two rockers pointing towards the middle of

the rear axle; see Fig 3.6a. Such layout could thus be considered 100 % Ackermann. If the two rockers point below the middle of the rear axle, the steering is less then 100 % Ackermann, parallel steering is obtained when the two rockers are parallel, while if the two rockers point above the middle of the rear axle the steering is more than 100 % Ackermann. Finally, if the intersection of the two rockers moves in front of the front axle, reverse-Ackermann steering is obtained, i.e. the steer of the inner wheel is smaller that the steer of the outer wheel. The distance between the connecting rod and the line between the steering pivots does not affect the steering geometry for small steer rotations, as long as the inclination of the two rockers remains constant. Indeed, basic kinematic considerations dictate that the position of the velocity centre of the connecting rod is at the intersection of the rocker axes. The Ackermann steering ratio is sometimes computed as

$$\nu_\tau = \frac{\tau}{T/2}, \tag{3.57}$$

where τ is the distance between the intersection of the rocker axis and the rear axle, while T is the track width, see Fig. 3.6a.

In the current industrial practice, the Jeantaud layout is modified by replacing the connecting rod with two tie rods (CB and EF) that connect the wheel hub to the steering rack (CDE); see Fig. 3.6b. As in the case of the four-bar linkage, when the two rods AB and FG point towards the middle of the rear axle the layout could be considered 100 % Ackermann. The conditions of different Ackermann ratios follow the considerations reported for the Jeantaud linkage.

A different method for defining the steering geometry relates to the position of the projection of the velocity centre of the front wheels on the vehicle wheelbase [48]. In this case the Ackermann ratio is given by

$$\nu_w = \frac{w}{L}, \tag{3.58}$$

where L is shown in Fig. 3.7. If the projection is on the rear axle, the steering system is 100 % Ackermann. If the projection is below the rear axle ($L > w$) the layout is less than 100 % Ackermann, if the projection lies at infinity ($L = \infty$) the parallel steering is obtained (0 % Ackermann).

A net-steer ratio definition is often employed. This is the ratio of the difference between the inner and outer wheel-steering angles to the difference between the inner and outer wheel-steering angles of the corresponding Ackermann geometry

$$\nu_n = \frac{\delta_i - \delta_o}{\delta_{i,a} - \delta_{o,a}}, \tag{3.59}$$

where δ_i and δ_o are the inner and outer wheel-steering angles of the current layout, while $\delta_{i,a}$ and $\delta_{o,a}$ are the inner and outer wheel-steering angles in the corresponding Ackermann geometry. There are two main widespread options to identify the Ackermann geometry associated to the current steering geometry. In the first option, the inner wheel-steering angle is kept fixed and equal to the inner-wheel steering angle of the current layout, i.e. $\delta_{i,a} = \delta_i$, while the outer wheel-steering

angle of the Ackermann geometry is readily computed from (3.56) as

$$\delta_{o,a} = \arctan\left(\frac{w\tan\delta_{i,a}}{w + T\tan\delta_{i,a}}\right). \tag{3.60}$$

In the second option, both the inner-wheel and outer-wheel steering angles are changed by the same quantity t in order to give the Ackermann steering geometry, i.e.

$$\delta_{i,a} = \delta_i + t, \qquad \delta_{o,a} = \delta_o - t. \tag{3.61}$$

In other words, in this case the Ackermann steering is obtained by applying a toe correction to the current steering layout. Such correction can be either positive (toe-out) or negative (toe-in) and is not constant. The latter option is employed e.g. in [49, 50].

Finally, a linearised version of the net-steer ratio (3.59) is sometimes used [48]. In this case the denominator of (3.59) is approximated with the following expression which is obtained from (3.56)

$$\delta_{i,a} - \delta_{o,a} \approx \delta_o\delta_i\frac{T}{w} \approx \delta_o^2\frac{T}{w}. \tag{3.62}$$

The different definitions are compared in Fig. 3.5b using the current FSAE steering layout. Application of (3.57) gives a (constant) Ackermann ratio of 47 % (green solid line with crosses). All the other definitions are almost coincident and equal to 32 % at very small wheel-steer angles. Definitions ν_w (3.58) and ν_n (3.59) with (3.61) remain very close in the whole steer range, raising from 32 % to 34 % (dashed violet and solid blue respectively). Definition ν_n (3.59) with (3.62) increases slightly more, and reaches a final value of 36 % (dotted yellow). Definition ν_n (3.59) with (3.60) is more sensitive to steer changes and raises to 40 % at the maximum steering considered (thick solid red). Finally, when applying a four-bar linkage with the two rockers pointing towards the middle of the rear axle, ratios in the range 24-33 % are obtained, depending on the definition selected.

Summarising, the net-steer definitions should be preferred over the practical ones, such as those related to the intersection of the rocker axes. The definition (3.59) with (3.61) is the one that is used in the steering characteristic analysis discussed in Ch. 5.

Table 3.3: Baseline car parameters and tyre coefficients.

Symbol	**Description**	**Value**
g	gravity	$9.81\,\mathrm{m/s^2}$
ρ_a	air density	$1.20\,\mathrm{kg/m^3}$
m	mass	$280\,\mathrm{kg}$
h	height of CoM	$0.315\,\mathrm{m}$
w	wheelbase	$1.535\,\mathrm{m}$
a	CoM from front axle	$0.767\,\mathrm{m}$
b	CoM from rear axle	$0.768\,\mathrm{m}$
ξ	roll stiffness ratio	0.489
I_z	yaw inertia	$109\,\mathrm{kgm^2}$
T_f	front track	$1.220\,\mathrm{m}$
T_r	rear track	$1.190\,\mathrm{m}$
T	$(T_f + T_r)/2$	$1.205\,\mathrm{m}$
$C_D A$	drag area coefficient	$1.38\,\mathrm{m^2}$
$C_{Lf} A$	front lift area coefficient	$0.89\,\mathrm{m^2}$
$C_{Lr} A$	rear lift area coefficient	$1.33\,\mathrm{m^2}$
$P_{\max}$	maximum power	$66.3\,\mathrm{kW}$
pC_{x1}	Longitudinal shape factor	2.31
pD_{x1}	Max longitudinal friction coefficient	-1.20
pD_{x2}	Max longitudinal friction coefficient	0.71
pE_{x1}	Longitudinal curvature factor	1.00
pK_{x1}	Max longitudinal stiffness coefficient	39.06
pK_{x2}	Max longitudinal stiffness coefficient	-0.32
pK_{x3}	Max longitudinal stiffness coefficient	-0.23
$\lambda_{\mu,x}$	Longitudinal friction scaling factor	1.00
pC_{y1}	Lateral shape factor	1.86
pD_{y1}	Max lateral friction coefficient	-2.48
pD_{y2}	Max lateral friction coefficient	0.06
pE_{y1}	Lateral curvature factor	0.93
pK_{y1}	Max cornering stiffness coefficient	53.91
pK_{y2}	Max cornering stiffness coefficient	2.57
pK_{y3}	Max cornering stiffness coefficient	3.95
pV_{y3}	Variation of shift with camber	-3.01
pV_{y4}	Variation of shift with camber and load	-1.51
$\lambda_{\mu,y}$	Lateral friction scaling factor	0.61
N_0	reference normal load (for $df = 0\,\mathrm{N}$)	$809\,\mathrm{N}$

Chapter 4

g-g diagrams

4.1 Introduction

The *g-g diagram* summarises the global vehicle performance, taking into account for the maximum attainable lateral and longitudinal acceleration of the vehicle. In other words, the g-g diagram is a map of the lateral vs. longitudinal acceleration that a given vehicle reaches when driving at the limit of its performance. This representation has an important role in the parameter optimisation or in the comparison of the performance achieved by different vehicles. This approach has been widely employed also in the simulation framework, in particular for minimum-lap-time problems. In fact, the synthesising power of the g-g diagrams allows to retain the complexity of the vehicle model out of the simulation. The computation of the maximum performance can be simply performed by evaluating the pre-calculated g-g at each simulation step, requiring a low computational effort for the minimum-time program, while guaranteeing a fast and reliable solution. Nevertheless, only a steady-state representation of the vehicle performance is allowed and no transient effects can be included in the pre-computed g-g diagrams. The steady-state effects involved in determining the g-g limits are [46]:

1. Power limit. The maximum engine power of the vehicle reduces the maximum longitudinal acceleration achievable since the traction force has to compensate for the drag force. The g-g upper limit is given by:

$$ma_x^{lim} = F_x - F_D = P_{max}/V - F_D, \tag{4.1}$$

   where a_x^{lim} is the longitudinal acceleration limit, F_x is the available traction force, F_D is the drag force and P_{max} is the maximum engine power.

2. Aerodynamics. The effect of the downforces is related to the increase of the normal loads at 'high' speeds. These forces allow higher lateral accelerations and may modify the brake balance and the stability balance. The drag force limits the available longitudinal acceleration (due to the power limit) and increases the maximum braking performance, since F_D and F_x have the same direction while braking.

3. Load transfer. The lateral and longitudinal load transfer affect the available slip of the vehicle and limits (or increases) the performance in each part of the g-g. This effect is optimised, for example, adjusting the suspension geometry and compliance.

4. Rollover. The lateral acceleration is reduced during cornering when the inner tyres lift from the ground.

5. Brake balance. For negative longitudinal accelerations, the brake balance affects the achievement of tyre limit on each axle.

Each of these effects are related to the achievement of the tyre-force limit. For this reason, in some sense, the g-g diagrams are a synthetic representation of the vehicle performance when one or more tyres reach their friction limit.

The generation of g-g diagrams is a consolidated topic and some examples are available in literature discussing their application on minimum-lap-time simulations. In [31] and [32] a seven degrees-of-freedom vehicle model is employed for computing the g-g-speed diagrams of a race car. The maximum speed achievable is computed first, while the maximum lateral and longitudinal performance of the car are obtained from optimisation methods for a finite set of speeds. The g-g profile is then computed by maximising the lateral acceleration of the vehicle at various increments of longitudinal acceleration. This process is repeated at different speeds. A similar procedure is presented in [37] by including a limited-slip-differential model. Differently, in [51] an experimental g-g diagram is used for constraining the vehicle performance in a fixed-trajectory simulation.

In this chapter, a method for the numerical computation of g-g-speed diagrams is discussed in detail. The method is mainly based on the approach presented in [31]-[32], although quite different strategies have been implemented. In particular, the presented approach make use of the polar-coordinate transformation in order to obtain a more homogeneous formulation of the optimisation problem. Moreover automatic differentiation (Ch. 2) is employed for providing the derivatives for the optimiser. In Sec. 4.2 the optimisation program used for generating the g-g diagrams is outlined. A deeper insight on the implementation strategies adopted for each part of the program is given in Sec. 4.3. In Sec. 4.4 the method is applied to a race car model and the effect of different parameters on the g-g is also discussed. Moreover, a set of race motorcycle g-g diagrams is obtained through analytical solution of an essential model (see Ch. 3) and compared to the race car results.

4.2 Basics

The numerical generation of the g-g diagrams at different speeds, is a rather consolidated practice in the vehicle simulation framework. Despite of that, the achievement of a robust and reliable solution of the steady-state optimisation problem is not trivial, due to the high number of suboptimal solutions of the steady-state vehicle model [52]. In general, a g-g diagram at a given speed V_i is obtained from the results of a series of dynamic equilibrium (i.e. steady-state) problems, that are solved through a constrained nonlinear-optimisation routine. In the following

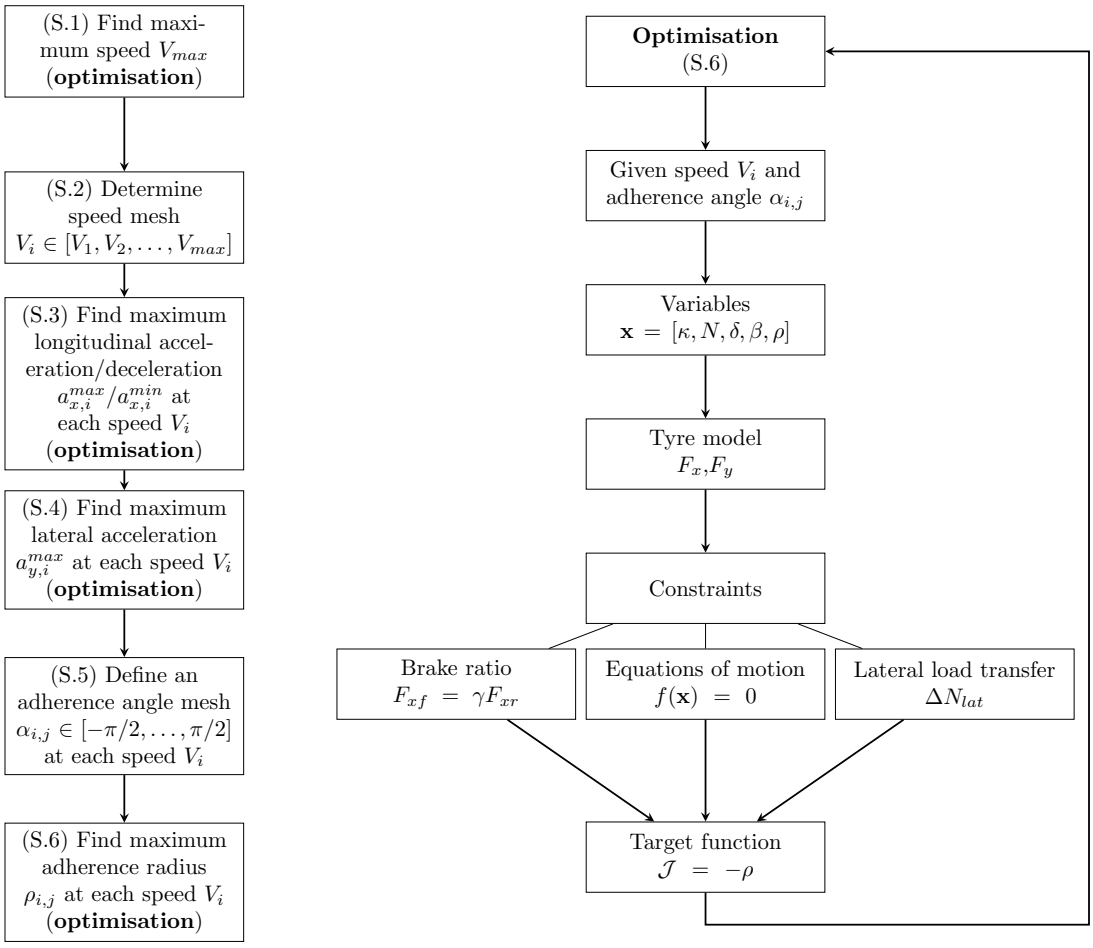


Figure 4.1: g-g calculation routine (left) and optimisation scheme for S.6 (right).

paragraphs, a method for the numerical calculation of the g-g diagrams will be presented, implementing the RWD vehicle steady-state model discussed in Ch. 3. The g-g generation routine (Fig. 4.1, left) consists of the following steps:

S.1 An optimisation program is employed for calculating the maximum speed V_{max} of the vehicle. In this case, only the variables involved in the straight constant-speed manoeuvre of the vehicle are considered. The variables of the problem are:

$$\begin{aligned} &\mathbf{x}_{i,j} = [\kappa_r, N_f, N_r]^T, \\ &\text{given} \quad \kappa_f = 0,\, \delta = 0,\, \beta = 0,\, \Omega = 0, \end{aligned} \tag{4.2}$$

where $\kappa_r = \kappa_{rl} = \kappa_{rr}$, $\kappa_f = \kappa_{fl} = \kappa_{fr}$, $N_r = N_{rl} = N_{rr}$ and $N_f = N_{fl} = N_{fr}$. The target is:

$$\mathcal{J} = -V. \tag{4.3}$$

The normal loads N and the longitudinal slips κ of left and right tyres of the same axle are considered equal, while the steering angle δ, the yaw rate Ω, the sideslip angle β and the front longitudinal slips κ_f are zero (straight traction manoeuvre).

S.2 A set of absolute vehicle speeds $V_i \in [V_1, V_2, \ldots, V_{max}]$ is defined. This set represents a discretisation mesh for the problem and the g-g diagrams will be calculated for each speed V_i.

S.3 The maximum longitudinal acceleration $a_{x,i}^{max}$ and the maximum longitudinal deceleration $a_{x,i}^{min}$ are calculated at speed V_i through an optimisation routine. Only the variables involved in the straight acceleration/braking manoeuvre are considered in the optimisation. This step is repeated for each element of the speed mesh. The variables of the problem are:

$$\begin{aligned} &\mathbf{x}_{i,j} = [\kappa_f, \kappa_r, N_f, N_r, a_x]^T, \\ &\text{given} \quad \delta = 0,\, \beta = 0,\, \Omega = 0,\, V_i, \end{aligned} \tag{4.4}$$

while the target is:

$$\mathcal{J} = -a_x \,(\text{acceleration}) \quad \text{or} \quad \mathcal{J} = a_x \,(\text{braking}). \tag{4.5}$$

In this case, the normal loads and the longitudinal slip of left and right tyres of the same axle are considered equal. The steering angle, the yaw rate and the sideslip angle are zero (straight traction/braking manoeuvre). In the braking manoeuvre a constraint is added for implementing a fixed brake-ratio.

S.4 The maximum lateral acceleration $a_{y,i}^{max}$ is calculated for longitudinal acceleration $a_x = 0$ and speed V_i, through an optimisation routine. The maximum lateral performance is considered symmetrical for left and right corners. Only the variables involved in the steady-turning accelerating manoeuvre are

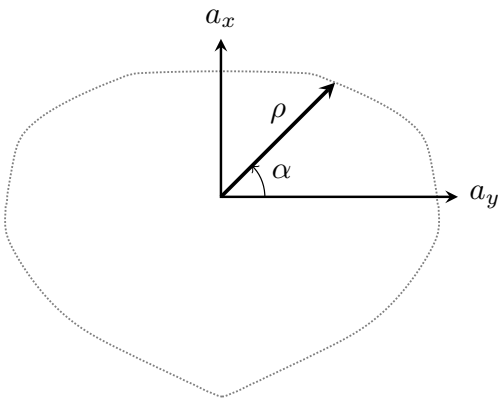


Figure 4.2: Polar coordinates.

considered in the optimisation. The condition $a_x = 0$ implies that the longitudinal tyre forces are balancing the drag force at the given speed. This step is repeated for each element of the speed mesh. The variables of the problem are:

$$\begin{aligned} &\mathbf{x}_{i,j} = [\kappa_{rl}, \kappa_{rr}, N_{fl}, N_{fr}, N_{rl}, N_{rr}, \Omega, \beta, \delta]^T , \\ &\text{given} \quad \kappa_{fl} = \kappa_{fr} = 0,\ a_x = 0,\ V_i, \end{aligned} \tag{4.6}$$

while the target is:

$$\mathcal{J} = -a_y. \tag{4.7}$$

This manoeuvre is simulated for $a_x = 0$ and traction conditions ($\kappa_{rl} > 0$, $\kappa_{rr} > 0$) are needed in order to balance the drag resistance.

S.5 A polar-coordinates reference-frame is considered in the (a_x, a_y) plane, with the origin in $(a_y = 0, a_x = 0)$. A point (ρ, α) of this frame is obtained from the polar transformation of the standard g-g diagram frame (a_y, a_x), using the following equalities:

$$\rho = \sqrt{a_x^2 + a_y^2}, \tag{4.8}$$

$$\alpha = \arctan\left(\frac{a_x}{a_y}\right), \tag{4.9}$$

where ρ is the adherence radius and α is the adherence angle (see Fig. 4.2). A mesh of adherence angles $\alpha_{i,j} \in [-\pi/2, \ldots, \pi/2]$ is now defined at each speed V_i. The g-g diagram will be calculated in each point of this adherence-angle mesh.

S.6 Given the speed V_i and the adherence angle $\alpha_{i,j}$, an optimisation routine is employed for calculating the maximum adherence radius $\rho_{i,j}$, at step (i, j). In this case, the full set of variables is included, in order to involve both the acceleration/deceleration equations and the lateral dynamics of the vehicle.

This step is repeated for each element of the adherence-angle mesh and for each element of the speed mesh. The variables of the problem are:

$$\begin{aligned} &\mathbf{x}_{i,j} = [\kappa_{fl}, \kappa_{fr}, \kappa_{rl}, \kappa_{rr}, N_{fl}, N_{fr}, N_{rl}, N_{rr}, \beta, \delta, \rho]^T, \\ &\text{given} \quad \alpha_{i,j},\, V_i, \end{aligned} \tag{4.10}$$

while the target is:

$$\mathcal{J} = -\rho. \tag{4.11}$$

It should be noticed that the steps S.3 and S.4 are not redundant. In fact the routine described in S.6 takes advantage of the knowledge of the previously-calculated extreme points ($a_{x,i}^{max}$, $a_{x,i}^{min}$ and $a_{y,i}^{max}$) in order to improve the generation of a guess for the optimisation (see Sec. 4.3). The optimisation routine for S.6 is described in Fig. 4.1, (right). The problem variables and the parameters V_i and $\alpha_{i,j}$ are introduced in a nonlinear tyre model for computing the tyre forces on each wheel. These information are used for calculating the nonlinear optimisation constraints, which consist in the steady-state equations of motion of the vehicle, the braking equations for introducing a fixed brake-ratio and the lateral load-transfer constraints. The aim of the program is to minimise the given target while satisfying the constraints. A similar procedure is also employed in the previous optimisation steps S.1, S.3 and S.4.

4.3 Implementation

In this section the g-g diagrams generation program for an RWD vehicle model (see Ch. 3) is analysed in more detail and the strategies adopted for enhancing the performance of the program are presented. The optimisation steps are solved with IPOPT, while ADiGator (see Ch. 2) is employed for the evaluation of derivatives.

4.3.1 Maximum speed

The maximum-speed subproblem is formulated as a constant speed simulation along a straight path. The longitudinal acceleration is zero ($a_x = 0$), while the lateral dynamics is neglected (steering angle $\delta = 0$, lateral speed $v = 0$ and acceleration $a_y = 0$). The optimisation variables are:

$$\mathbf{x} = [\kappa_r, N_f, N_r]^T, \tag{4.12}$$

where $\kappa_r = \kappa_{rl} = \kappa_{rr}$, $N_f = N_{fl} = N_{fr}$ and $N_r = N_{rl} = N_{rr}$ (no lateral load transfer). The optimisation (equality) constraints are the Newton-Euler equations of the vehicle in straight motion[1]:

$$0 = ma_x = F_{xrl} + F_{xrr} - F_D, \tag{4.13}$$

$$0 = ma_x\, h = aF_{Lf} - bF_{Lr} - a(N_{fl} + N_{fr}) + b(N_{rl} + N_{rr}), \tag{4.14}$$

$$0 = mg + F_{Lf} + F_{Lr} - N_{fl} - N_{fr} - N_{rl} - N_{rr}. \tag{4.15}$$

[1]The meaning of the symbols used is reported in Ch. 3.

Equation (4.13) is the force balance along the longitudinal axis, Eq. (4.14) is the momentum balance in the lateral direction and Eq. (4.15) is the force balance along the vertical axis. The longitudinal speed in the vehicle frame u can be explicitly shown in the Newton-Euler equations considering $a_x = \dot{u}$. An inequality constraint is considered for including the power limit of the engine.

$$(F_{xrl} + F_{xrr})u \leq P_{max}, \tag{4.16}$$

where P_{max} is the maximum power displaced by the engine. The tyre forces F_{xij} $(i = f, r,\ j = l, r)$ are obtained from the rear longitudinal slips κ_r and the normal loads N_{ij}, considering the simplified Pacejka formulas presented in Ch. 3. Since a traction force is needed for balancing the aerodynamic drag force, only the rear forces and longitudinal slips are considered (i.e. the braking condition is not taken into account and $\kappa_{fl} = \kappa_{fr} = 0$). The optimisation target is the longitudinal speed $u = V$ of the vehicle, i.e. $\mathcal{J} = -V$. Since the model is rather simple, the initial point at which the functions are evaluated (optimisation guess) can be estimated by simple steady-state considerations.

4.3.2 Maximum longitudinal performance

The maximum longitudinal performance is determined by solving two optimisation sub-problems: the research of the maximum longitudinal acceleration, and the research of the maximum braking performance of the vehicle. These points represent the upper and the lower limits of the g-g diagram, and depend on the manoeuvre speed (which is given for each sub-problem, see S.2). Both the maximum acceleration and the maximum deceleration of the vehicle are determined through a straight simulation at constant acceleration. The lateral dynamics is neglected (steering angle $\delta = 0$, lateral speed $v = 0$ and acceleration $a_y = 0$).

Maximum acceleration

In each optimisation, the longitudinal speed is given by $u = V_i$, where V_i is the speed value included in the mesh-speed vector (see S.2). Only traction condition is considered, i.e. $\kappa_{fl} = \kappa_{fr} = 0$. The optimisation variables are:

$$\mathbf{x} = [\kappa_r, N_f, N_r, a_x]^T, \tag{4.17}$$

where $\kappa_r = \kappa_{rl} = \kappa_{rr}$, $N_f = N_{fl} = N_{fr}$ and $N_r = N_{rl} = N_{rr}$ (no lateral load transfer). The equality constraints of the problem consist in the dynamic equations during straight acceleration

$$\begin{aligned} ma_x &= F_{xrl} + F_{xrr} - F_D, && (4.18)\\ ma_x h &= aF_{Lf} - bF_{Lr} - a(N_{fl} + N_{fr}) + b(N_{rl} + N_{rr}), && (4.19)\\ 0 &= mg + F_{Lf} + F_{Lr} - N_{fl} - N_{fr} - N_{rl} - N_{rr}. && (4.20) \end{aligned}$$

Equation (4.18) is the force balance along the longitudinal axis, Eq. (4.19) is the momentum balance in the lateral direction and Eq. (4.20) is the force balance along

the vertical axis. An inequality constraint is considered for including the power limit of the engine.

$$(F_{xrl} + F_{xrr})u \leq P_{max}, \tag{4.21}$$

where P_{max} is the maximum power displaced by the engine. Similarly to the maximum-speed problem, the tyre forces F_{xij} $(i = f, r,\ j = l, r)$ are obtained from the rear longitudinal slips κ_r and the normal loads N_{ij}, considering the simplified Pacejka formulas presented in Ch. 3. Since the vehicle is evaluated in a traction condition, only the rear forces and longitudinal slips are considered (i.e. the braking condition is not taken into account). The optimisation target is the longitudinal acceleration $a_x = \dot{u}$ of the vehicle, i.e. $\mathcal{J} = -a_x$. Since the model is rather simple, the initial point at which the functions are evaluated (optimisation guess) can be estimated by simple steady-state considerations.

Maximum deceleration

In each optimisation, the longitudinal speed is given by $u = V_i$, where V_i is a speed value included in the mesh-speed vector (see S.2). In this case, the braking condition is considered and the optimisation variables become:

$$\mathbf{x} = [\kappa_f, \kappa_r, N_f, N_r, a_x]^T, \tag{4.22}$$

where $\kappa_f = \kappa_{fl} = \kappa_{fr}$, $\kappa_r = \kappa_{rl} = \kappa_{rr}$, $N_f = N_{fl} = N_{fr}$ and $N_r = N_{rl} = N_{rr}$ (no lateral load transfer). The equality constraints of the problem consist in the dynamic equations during straight deceleration

$$\begin{aligned} ma_x &= F_{xfl} + F_{xfr} + F_{xrl} + F_{xrr} - F_D, && (4.23)\\ ma_x\, h &= aF_{Lf} - bF_{Lr} - a(N_{fl} + N_{fr}) + b(N_{rl} + N_{rr}), && (4.24)\\ 0 &= mg + F_{Lf} + F_{Lr} - N_{fl} - N_{fr} - N_{rl} - N_{rr}. && (4.25) \end{aligned}$$

Equation (4.23) is the force balance along the longitudinal axis, Eq. (4.24) is the momentum balance in the lateral direction and Eq. (4.25) is the force balance along the vertical axis. A further equality constraint is included for the fixed brake-ratio between front and rear axle:

$$\begin{aligned} \gamma &= \frac{F_{xfl} + F_{xfr}}{F_{xrl} + F_{xrr}}, && (4.26)\\ F_{xfl} + F_{xfr} &+ \gamma\,(F_{xrl} + F_{xrr}) = 0, && (4.27) \end{aligned}$$

where γ is the ratio between the front and rear longitudinal forces. This formulation of the constraint is valid only for braking (negative) forces; a slightly different formulation will be discussed in Sec. 4.3.4 for including both the traction and braking conditions. It is assumed that there is no limit on the maximum braking force, i.e. the brake system can always provide the necessary braking power. The longitudinal tyre forces F_{xij} $(i = f, r,\ j = l, r)$ are obtained from the front and rear longitudinal slips (κ_f and κ_r) and the normal loads N_{ij}, considering the simplified Pacejka formulas presented in Ch. 3. The vehicle is evaluated in a braking

condition and both the front and the rear tyre forces and longitudinal slips have to be taken into account. The optimisation target is the longitudinal acceleration $a_x = \dot{u}$ of the vehicle, i.e. $\mathcal{J} = a_x$. Since the model is rather simple, the initial point at which the functions are evaluated (optimisation guess) can be estimated by simple steady-state considerations.

4.3.3 Maximum lateral performance

Similarly to the maximum acceleration/deceleration optimisation, the longitudinal speed is given by $V = \sqrt{u^2 + v^2} = V_i$, where V_i is a speed value included in the mesh-speed vector (see S.2). The longitudinal speed of the vehicle is $u = V\cos\beta$ and the lateral speed of the vehicle is $v = V\sin\beta$. The maximum lateral performance is achieved by a steady-turning manoeuvre, considering $a_x = 0$. It is noticed that, in general, this g-g point does not represent the proper maximum lateral acceleration a_y attainable by the vehicle. In fact, the condition $a_x = 0$ is achieved when the aerodynamic drag force is balanced by the traction force of the vehicle. In other words, a positive traction force is needed for maintaining to zero the longitudinal acceleration and the lateral performance is reduced by the tyre force combination. The real maximum lateral performance is achieved for $a_x < 0$, at which the vehicle is decelerating because of the drag force. Despite of that, the g-g point at $a_x = 0$ has been chosen for the g-g generation in order to be consistent with the polar reference frame considered. The optimisation variables related to the steady-turning manoeuvre are:

$$\mathbf{x} = \left[\kappa_{rl}, \kappa_{rr}, N_{fl}, N_{fr}, N_{rl}, N_{rr}, \Omega, \beta, \delta\right]^T, \tag{4.28}$$

while the optimisation constraints consist in the steady-state equations of motion of the RWD vehicle in traction conditions:

$$\begin{aligned}
0 &= ma_x = (F_{xrl} + F_{xrr}) - (F_{yfl} + F_{yfr})\,\delta - F_D, && (4.29)\\
ma_y &= F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr}, && (4.30)\\
0 &= mg + F_{Lf} + F_{Lr} - N_{fl} - N_{fr} - N_{rl} - N_{rr}, && (4.31)\\
ma_y h &= \frac{T}{2}(N_{fl} - N_{fr} + N_{rl} - N_{rr}), && (4.32)\\
0 &= ma_x\, h = aF_{Lf} - bF_{Lr} - a(N_{fl} + N_{fr}) + b(N_{rl} + N_{rr}), && (4.33)\\
0 &= \frac{T}{2}\left(F_{yfl} - F_{yfr}\right)\delta + \frac{T}{2}\left(-F_{xrl} + F_{xrr}\right) - a\left(F_{yfl} + F_{yfr}\right) + \\
&\quad + b\left(F_{yrl} + F_{yrr}\right), && (4.34)
\end{aligned}$$

where the relation $a_y = \Omega u$ holds for the steady-turning manoeuvre. Equations (4.29), (4.30) and (4.31) represent the force balance along the x, y and z axes of the vehicle reference frame, while Eq. (4.32), (4.33) and (4.34) represent the moment balance with reference to the x, y and z axes. An equality constraint is included to account for the lateral load transfer

$$ma_y \frac{h}{T}\xi = \frac{N_{fl} - N_{fr}}{2}, \tag{4.35}$$

where ξ is the roll-stiffness ratio of the vehicle, i.e. the ratio between the front roll stiffness and the total (front plus rear) roll stiffness. The open-differential hypothesis produces the equality constraint

$$F_{xrl} = F_{xrr}, \tag{4.36}$$

while the lateral slip angles are limited through four inequality constraints:

$$\lambda_{min} \leq \lambda_{ij} \leq \lambda_{max}, \tag{4.37}$$

where $i = f, r$, $j = l, r$. The power limit is included as an inequality constraint

$$(F_{xrl} + F_{xrr})V \leq P_{max}, \tag{4.38}$$

where P_{max} is the maximum power displaced by the engine. The tyre forces F_{xij} and F_{yij} $(i = f, r,\ j = l, r)$ are obtained from the rear longitudinal slips κ_{ij}, the lateral slips λ_{ij} and the normal loads N_{ij}, considering the simplified Pacejka formulas presented in Ch. 3. Since the vehicle is evaluated in traction condition, only the rear forces and longitudinal slips are considered (i.e. the braking condition is not taken into account). The optimisation target is the lateral acceleration of the vehicle $a_y = \Omega u$, plus a penalty term for preventing the program to explore solution areas with 'high' slips

$$\mathcal{J} = -\Omega u + w_\beta \beta^2 + w_\lambda \sum_{i,j} \lambda_{ij}^2 + w_\kappa \sum_{i,j} \kappa_{ij}^2, \tag{4.39}$$

where w_β, w_λ, w_κ are the penalty weight related to the vehicle sideslip angle, the tyre sideslip-angles and the longitudinal slips respectively. The problem formulation is more complex than the case of pure longitudinal simulations. For this reason, the guess for each optimisation is fundamental for achieving a reliable solution of the problem and a more complex algorithm for the guess generation have to be implemented. Considering a generic step k related to the speed V_k. A preliminary guess is determined from the solution at step $k-1$ (if available), while simple steady-state considerations are employed for $k = 1$. This preliminary guess is used as starting point for the numerical solution of the system of equations formed by the equality constraints of the optimisation problem. A zero-finding algorithm is used for determining whether the preliminary guess satisfies the equality constraints. This previous-solution check is carried out for a mesh of lateral accelerations formed by the neighbourhood of the preliminary estimation of a_y. If the residuals of this numerical method are lower than a given tolerance (i.e. 10^{-8}), the new solution is employed as guess for the optimisation step k. Otherwise, a mesh of trial points is generated considering different combinations of steering angle δ, lateral acceleration a_y and longitudinal slip κ_{rj}. For each trial point $(\delta, a_y, \kappa)_k$, a guess for the other variables is deduced from steady-state considerations. This guess is employed as starting point for the zero-finding algorithm. The guess of the optimisation problem at step k is the solution of the zero-finding algorithm that both satisfies the constraints with residuals lower than a given tolerance and reduces the residuals to the minimum. If no solutions are suitable for this criterium, the optimisation guess is simply estimated from steady-state formulas.

4.3.4 Maximum adherence radius

Both the maximum longitudinal and lateral performance optimisations (Sec. 4.3.2-4.3.3) are employed in the guess estimation for the optimisation of the adherence radius ρ at different adherence angles α (see Fig. 4.2). Determining the extreme points of the g-g diagram is fundamental for employing reliable solutions as starting-points for the optimisation of the entire g-g, especially in the sections where the vehicle dynamics is more complex.

Guess generation

The maximum adherence radius problem is divided into three subproblems, in order to enhance the reliability of the solutions using a proper guess and a proper set of equations. For each subproblem the number of adherence angles α is selected (e.g. 30 points), in order to give a discrete representation of the g-g diagram. The subproblems are built through different definitions of the adherence-angle mesh, which spans different parts of the g-g diagram. The procedure for generating the adherence-angle mesh is outlined in S.5 and is now specified for each subproblem:

P.1 $\alpha \in [0, \ldots, -\pi/4]$. This part of the g-g is related to the deceleration of the vehicle while cornering. The g-g points are determined starting from $\alpha = 0$, $(a_x = 0)$ to $\alpha = -\pi/4$. The first point is known from the solution of S.4, and represents the guess for the first optimisation.

P.2 $\alpha \in [-\pi/2, \ldots, -\pi/4]$. In this part of the g-g, the vehicle performs hard braking manoeuvres. The g-g points are determined starting from $\alpha = -\pi/2$, $(a_y = 0)$ to $\alpha = -\pi/4$. The first point is known from the solution of S.3, and represents the guess for the first optimisation.

P.3 $\alpha \in [0, \ldots, \pi/2]$. In this part of the g-g, the vehicle accelerates. For this reason, the braking forces and constraints are neglected. The g-g is determined starting from $\alpha = 0$, $(a_x = 0)$ to $\alpha = \pi/2$, $(a_y = 0)$. Both the initial and the final points are obtained from solving S.4 and S.3 respectively. In this case the choice of the first point of the optimisation does not have much influence on the solution, since both the extreme points represent a good guess for the convergence of the first steps.

It should be noticed that only half of the g-g diagram is generated with this choice of adherence-angles mesh, since the vehicle model and the tyre forces are symmetrical (at least with the employed parameters). For this reason the adherence angle spans from $-\pi/2$ to $\pi/2$. Similarly to Sec. 4.3.3, the complexity of the problem imposes the implementation of a routine for the guess generation. For each g-g step (k, l), related to speed V_k and adherence angle α_{kl}, a preliminary guess is determined from the solution of step $(k, l-1)$ (if available). The preliminary guess related to the first optimisation point is determined considering the solution of S.4 at step k for P.1, the solution of S.3 at step k for P.2 and the solution of S.4 at step k for P.3. For P.1 and P.2, the preliminary guess is used as a starting-point for a zero-finding algorithm, used to determine whether

the equality constraints of the problem are satisfied at this starting-point. This previous-solution check algorithm is applied for a mesh of adherence radii, which consists of the neighbourhood of the preliminary estimation of ρ_{kl}. If the residuals of the solution are lower than a given tolerance (i.e. 10^{-8}), the new solution is employed as guess for the optimisation step (k, l). Otherwise, a mesh of trial points is generated considering different combinations of steering angle δ, adherence radius ρ and longitudinal slip κ_{ij}. For each trial point included in the mesh $(\delta, \rho, \kappa)_{kl}$, a guess for the other variables is deduced from steady-state considerations. This guess is employed as starting point for the zero-finding algorithm. The guess of the optimisation problem at step (k, l) is the solution of the zero-finding algorithm that both satisfies the constraints with residuals lower than a given tolerance and reduces the residuals to the minimum. For P.3, the guess of the optimisation step is determined by the preliminary guess, i.e. the solution at step $(k, l-1)$ or the solution of S.4 at speed k for step $(k, l=1)$.

Optimisation

The maximum adherence radius problem involves a combination of lateral and longitudinal performance. The longitudinal absolute speed $V = \sqrt{u^2 + v^2} = V_i$ is given for each optimisation step (i, j), where V_i is included in the speed-mesh vector obtained in S.2. Once the absolute speed is selected, the optimisation is performed on the set of adherence angles α_{ij} described in S.5. Therefore, also the adherence angle is given for each optimisation step (i, j).

For subproblems P.1 and P.2 both the acceleration and braking conditions need be considered, since for small longitudinal decelerations, the drag force have to be balanced by traction tyre forces. The optimisation variables are:

$$\mathbf{x}_{i,j} = \left[\kappa_{fl}, \kappa_{fr}, \kappa_{rl}, \kappa_{rr}, N_{fl}, N_{fr}, N_{rl}, N_{rr}, \beta, \delta, \rho\right]^T, \tag{4.40}$$

while the optimisation constraints involve the most general formulation of the Newton-Euler equations of the RWD vehicle,

$$\begin{aligned}
ma_x &= (F_{xfl} + F_{xfr} + F_{xrl} + F_{xrr}) - (F_{yfl} + F_{yfr})\,\delta - F_D, && (4.41)\\
ma_y &= (F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr}) + (F_{xfl} + F_{xfr})\,\delta, && (4.42)\\
0 &= mg + F_{Lf} + F_{Lr} - N_{fl} - N_{fr} - N_{rl} - N_{rr}, && (4.43)\\
ma_y h &= \frac{T}{2}(N_{fl} - N_{fr} + N_{rl} - N_{rr}), && (4.44)\\
ma_x\, h &= aF_{Lf} - bF_{Lr} - a(N_{fl} + N_{fr}) + b(N_{rl} + N_{rr}), && (4.45)\\
0 &= \frac{T}{2}\,(F_{yfl} - F_{yfr})\,\delta - a\,(F_{xfl} + F_{xfr})\,\delta + \frac{T}{2}\,(-F_{xfl} + F_{xfr} - F_{xrl} + F_{xrr}) + \\
&\quad - a\,(F_{yfl} + F_{yfr}) + b\,(F_{yrl} + F_{yrr})\,. && (4.46)
\end{aligned}$$

where the longitudinal and lateral accelerations are given by $a_x = \rho \sin\alpha$ and $a_y = \rho\cos\alpha$, and the yaw rate is $\Omega = a_y/u$ for the quasi-steady-state hypothesis. Equations (4.41), (4.42) and (4.43) represent the force balance along the x, y and z axes of the vehicle reference frame, while Eq. (4.44), (4.45) and (4.46) represent

the moment balance with reference to the x, y and z axes. An equality constraint is included to account for the lateral load-transfer

$$ma_y \frac{h}{T}\xi = \frac{N_{fl} - N_{fr}}{2}, \tag{4.47}$$

where ξ is the roll-stiffness ratio of the vehicle, i.e. the ratio between the front roll stiffness and the total (front plus rear) roll stiffness. The open-differential hypothesis produces the equality constraint

$$F_{xrl} = F_{xrr}, \tag{4.48}$$
$$F_{xfl} = F_{xfr}, \tag{4.49}$$

while the lateral slip angles are limited by four inequality constraints:

$$\lambda_{min} \leq \lambda_{ij} \leq \lambda_{max}, \tag{4.50}$$

where $i = f, r$, $j = l, r$. A further equality constraint is included for the fixed brake-ratio between front and rear axle:

$$\gamma = \frac{F_{xfl} + F_{xfr}}{F_{xrl} + F_{xrr}}, \tag{4.51}$$
$$F_{xfl} + F_{xfr} + \gamma[\min(F_{xrl} + F_{xrr}, 0)] = 0, \tag{4.52}$$

where γ is the ratio between the front and rear longitudinal forces. Eq. 4.52 activates the fixed brake-ratio constraint only if the sum of the rear tyre forces is negative and is obtained through regularised functions. The power limit is included as an inequality constraint

$$(F_{xrl} + F_{xrr})V \leq P_{max}, \tag{4.53}$$

where P_{max} is the maximum power displaced by the engine. It is assumed that there is no limit on the maximum braking force, i.e. the brake system can always provide the necessary braking power. The tyre forces F_{xij} and F_{yij} ($i = f, r$, $j = l, r$) are obtained from the longitudinal slips κ_{ij}, the lateral slips λ_{ij} and the normal loads N_{ij}, considering the simplified Pacejka formulas presented in Ch. 3. The optimisation target is the adherence radius ρ, plus a penalty term for preventing the program to explore solution areas with 'high' slip

$$\mathcal{J} = -\rho + w_\beta \beta^2 + w_\lambda \sum_{i,j} \lambda_{ij}^2 + w_\kappa \sum_{i,j} \kappa_{ij}^2, \tag{4.54}$$

where w_β, w_λ, w_κ are the penalty weight related to the vehicle sideslip angle, the tyre sideslip-angles and the longitudinal slips respectively.

Considering subproblem P.3, only acceleration condition during cornering need be evaluated. The front tyre forces and longitudinal slips are neglected and a reduced set of optimisation variables can be used:

$$\mathbf{x} = [\kappa_{rl}, \kappa_{rr}, N_{fl}, N_{fr}, N_{rl}, N_{rr}, \beta, \delta, \rho]^T, \tag{4.55}$$

while the optimisation constraints consist in the steady-state equations of motion of the RWD vehicle in traction condition:

$$\begin{aligned} ma_x &= (F_{xrl} + F_{xrr}) - (F_{yfl} + F_{yfr})\,\delta - F_D, && (4.56)\\ ma_y &= F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr}, && (4.57)\\ 0 &= mg + F_{Lf} + F_{Lr} - N_{fl} - N_{fr} - N_{rl} - N_{rr}, && (4.58)\\ ma_y h &= \frac{T}{2}(N_{fl} - N_{fr} + N_{rl} - N_{rr}), && (4.59)\\ a_x\, h &= aF_{Lf} - bF_{Lr} - a(N_{fl} + N_{fr}) + b(N_{rl} + N_{rr}), && (4.60)\\ 0 &= \frac{T}{2}\,(F_{yfl} - F_{yfr})\,\delta + \frac{T}{2}\,(-F_{xrl} + F_{xrr}) - a\,(F_{yfl} + F_{yfr}) + \\ &\quad + b\,(F_{yrl} + F_{yrr})\,, && (4.61) \end{aligned}$$

where the longitudinal and lateral accelerations are given by $a_x = \rho \sin\alpha$ and $a_y = \rho\cos\alpha$, and the yaw rate is $\Omega = a_y/u$ for the steady-state hypothesis. Equations (4.56), (4.57) and (4.58) represent the force balance along the x, y and z axes of the vehicle reference frame, while Eq. (4.59), (4.60) and (4.61) represent the moment balance with reference to the x, y and z axes. An equality constraint is included to account for the lateral load transfer

$$ma_y \frac{h}{T}\xi = \frac{N_{fl} - N_{fr}}{2}, \tag{4.62}$$

where ξ is the roll-stiffness ratio of the vehicle, i.e. the ratio between the front roll stiffness and the total (front plus rear) roll stiffness. The open-differential hypothesis produces the equality constraint

$$F_{xrl} = F_{xrr}, \tag{4.63}$$

while the lateral slip angles are limited by four inequality constraints:

$$\lambda_{min} \leq \lambda_{ij} \leq \lambda_{max}, \tag{4.64}$$

where $i = f, r$, $j = l, r$. The power limit is included as inequality constraint

$$(F_{xrl} + F_{xrr})V \leq P_{max}, \tag{4.65}$$

where P_{max} is the maximum power displaced by the engine. The tyre forces F_{xij} and F_{yij} ($i = f, r$, $j = l, r$) are obtained from the rear longitudinal slips κ_{ij}, the lateral slips λ_{ij} and the normal loads N_{ij}, considering the simplified Pacejka formulas presented in Ch. 3. The optimisation target is the adherence radius ρ, plus a penalty term for preventing the program to explore solution areas with 'high' slips

$$\mathcal{J} = -\rho + w_\beta \beta^2 + w_\lambda \sum_{i,j} \lambda_{ij}^2 + w_\kappa \sum_j \kappa_{rj}^2, \tag{4.66}$$

where w_β, w_λ, w_κ are the penalty-weights related to the vehicle sideslip angle, the tyre sideslip-angles and the rear longitudinal slips respectively.

4.3.5 Coding features

Although the numerical calculation of the g-g diagrams is a consolidated procedure, the solution of the g-g diagram optimisation is not in general an easy task, since many solutions exist for a steady-state vehicle model [52]. A first fundamental strategy that allows to improve the convergence to the desired solution is an effective determination of the problem guess. This topic is extensively discussed in Sec. 4.3.3 for S.4 and in Sec 4.3.4 for S.6. In this section, the expedients developed for facilitating the convergence of the optimisations are explained and the tools used for improving the performance of the routine are described.

Enhancing the problem statement

A first major improvement adopted for enforcing the convergence to 'low slip' solutions is the implementation of boundaries for the longitudinal and lateral slips. The maximum and minimum achievable slips are the values in which the lateral or the longitudinal force attain its maximum or minimum, i.e. the slip values correspondent to the peaks of the tyre forces. It is noticed that the peak value of the longitudinal (lateral) tyre force depends on both the longitudinal and lateral slips and on the tyre loads,

$$\kappa_{ij}^{\text{peak}} = \arg\max F_{xij}(\kappa_{ij}), \quad \text{given} \quad \lambda_{ij}, N_{ij}, \tag{4.67}$$

$$\lambda_{ij}^{\text{peak}} = \arg\max F_{yij}(\lambda_{ij}), \quad \text{given} \quad \kappa_{ij}, N_{ij}, \tag{4.68}$$

where $i = f, r$ and $j = l, r$. The values of $\kappa_{ij}^{\text{peak}}$ are obtained for a mesh of lateral slip angles λ_{ij} and normal loads N_{ij}, while the values of $\lambda_{ij}^{\text{peak}}$ are obtained for a mesh of longitudinal slips κ_{ij} and normal loads N_{ij}. The discrete representation obtained from the evaluation of the slip values correspondent to the maximum (minimum) tyre force is then interpolated by a two-dimensional spline surface, which can be evaluated by the optimisation program when computing the constraints. This strategy allows to avoid the solutions in which the tyre saturation is reached. For this reason, the number of solutions available for the optimiser is dramatically reduced and, consequently, the solutions achieved are similar to the ones of the neighbour g-g points and the guess generation from the previous solutions is more robust. Despite of that, there are some cases in which a higher performance can be obtained through solutions that allow the saturation of one or more tyres. These cases can not be properly handled by this formulation of the program.

A further program feature is related to the solution of subproblem P.3. Due to the vehicle limited power and to the drag resistance, the top of the g-g diagram is close to zero for all the positive longitudinal accelerations a_x. In these conditions, the adherence-angle mesh does not provide a good discretisation of the g-g. To avoid this problem, the polar reference frame (Fig. 4.2) is translated to the maximum deceleration point $(0, a_x^{min})$ and a new adherence-angle mesh is defined as $\alpha \in [\arctan(-a_x^{min}/a_y^{max}), \ldots, \pi/2]$. The resulting adherence radius is then transformed in order to return to the original polar-coordinate frame.

Algorithm 1: IPOPT interface for ADiGator

```
Data: guess x_0, options, parameters
Result: solution x
ADiGator setup;
if File-generation needed then
    ADiGator call → derivative functions;
    Jacobian pattern setup;
    Hessian pattern setup;
    Jacobian pattern save;
    Hessian pattern save;
else
    Jacobian pattern load;
    Hessian pattern load;
    Derivative-functions stored;
end
IPOPT call → x;
```

Figure 4.3: Pseudo-code for the IPOPT interface employed to allow an efficient usage of ADiGator.

Enhancing the program performance

Each optimisation point is obtained employing IPOPT and ADiGator (see Ch. 2) in order to improve the performance of the program in terms of time needed for solving the entire g-g. For this reason, a dedicated IPOPT interface is employed in order to communicate to a custom ADiGator wrapper, which allows to generate the derivatives only if the constraint equations are changed, i.e. there is no need to calculate the derivatives at each IPOPT call.

The custom wrapper for ADiGator adds some simple features to the original wrapper. It allows to save and store the sparsity information for the optimisation jacobian and hessian, in order to call them when the derivatives need not to be newly generated.

In the IPOPT interface two cases are distinguished. In the first case, ADiGator is called and the derivative files are generated. At this step the sparsity pattern of the jacobian and hessian are saved in separate files in order to be used later. This case is necessary when new derivative files need be generated (e.g. when a new problem or subproblem is solved for the first step). The second case is called when there is no need to generate new derivative files and the previously-generated files can be used. This condition occurs e.g. after the first optimisation point of the g-g, during each optimisation problem or subproblem. In order to correctly provide IPOPT with the previously-generated derivatives, the sparsity files saved after the ADiGator call (when the first case is called) are loaded, while the functions generated by ADiGator (when the first case is called) are stored in proper input variables for IPOPT. The pseudo-code for this IPOPT interface is reported in Fig. 4.3.

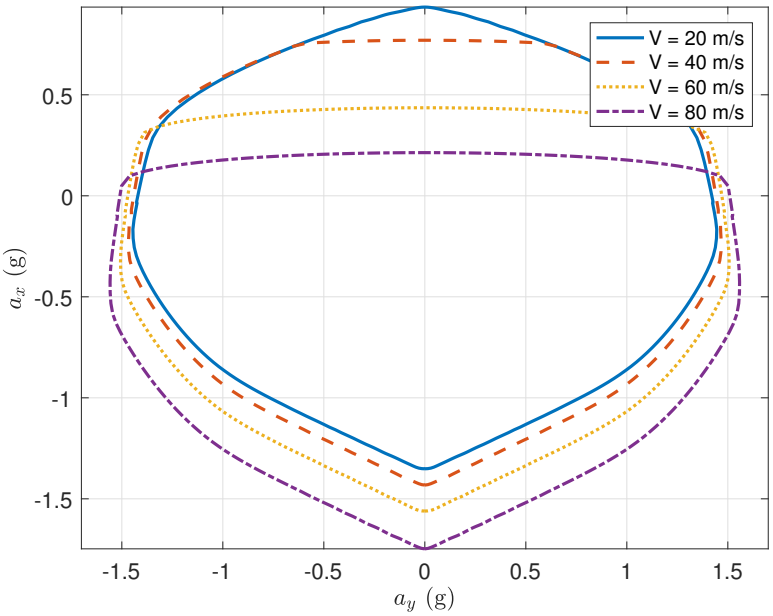


Figure 4.4: Baseline car g-g diagrams at different speeds.

4.4 Examples of application

Some examples of numerical g-g diagrams are now shown for the car model described in Sec. 4.3 and for a motorcycle model with analytical g-g diagrams (see Ch. 3). The g-g diagrams obtained with different parameters are compared and discussed in order to underline the effectiveness of this modelling strategy. The comparison between the car g-g and the motorbike g-g is also discussed, in order to highlight the main differences in modelling strategies and g-g generation.

4.4.1 Car model

A g-g diagram is shown in Fig. 4.4 using the baseline race car parameters (see Ch. 3). The boundaries of such diagram are found employing the optimisation procedures described in Sec. 4.3. At 'low' speeds, the g-g envelope is limited by the tyre-friction characteristics, both in acceleration and in braking. The boundary is reached when one or more tyres saturate. At 'high' speeds the effect of aerodynamics becomes dominant: the maximum acceleration is now limited by the power limit (the higher the speed, the lower the maximum acceleration available for traction), whereas in braking the drag force contributes to increasing the braking performance. The g-g diagrams shape changes significantly when the vehicle parameters are modified. Four parameters are taken into account for comparing the g-g diagrams at the same speed: the road adherence (modified through the tyre force scaling factors $\lambda_{\mu,x}$ and $\lambda_{\mu,y}$), the distance between centre of gravity of the vehicle and the rear axle b, the roll-stiffness ratio ξ and the front-to-total brake ratio β_b.

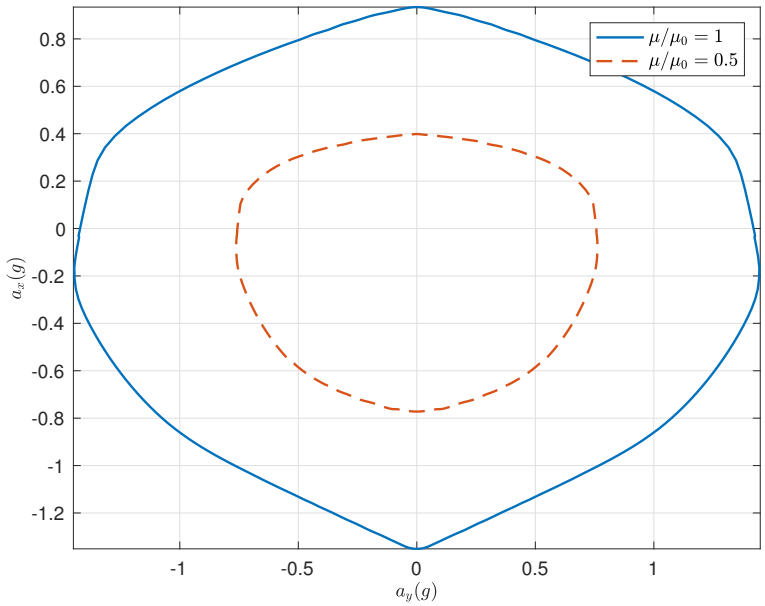


Figure 4.5: Car g-g diagrams at different adherence values μ ($V = 20\,\mathrm{m/s}$).

Road adherence

The g-g diagrams computed considering different values of road adherence are shown in Fig. 4.5. This result is obtained by reducing the $\lambda_{\mu,x}$ and $\lambda_{\mu,y}$ tyre-peak scaling factors by the same quantity, i.e. the condition of halved road adherence is obtained by halving both $\lambda_{\mu,x}$ and $\lambda_{\mu,y}$. The reduction of the adherence causes an obvious reduction of the maximum longitudinal and lateral acceleration achievable. Moreover, the combined lateral-longitudinal performance is reduced, and the shape of the g-g is changed.

Rear to CoG distance

The g-g diagrams computed considering different values of the distance between the rear axle and the centre of mass (while maintaining a constant wheelbase w) are shown in Fig. 4.6. This parameter mainly affects the longitudinal performance of the vehicle. Moving the centre of mass towards the rear of the vehicle (i.e. reducing b) increases the static load on the rear tyres and reduces the static load on the front tyres. As a consequence, higher traction forces can be produced in acceleration while during braking the front tyres are less affected from force saturation.

Roll stiffness ratio

The g-g diagrams computed considering different values of the roll-stiffness ratio, are shown in Fig. 4.7. This parameter mainly affects the lateral performance and the combined lateral-longitudinal manoeuvres. The maximum and minimum longitudinal accelerations are not influenced. A deeper investigation shows that

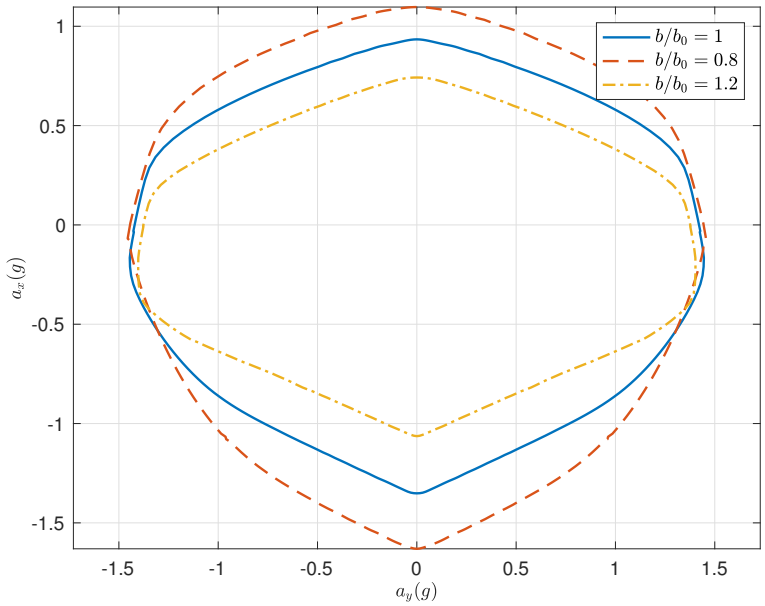


Figure 4.6: Car g-g diagrams at different b values ($V = 20\,\mathrm{m/s}$).

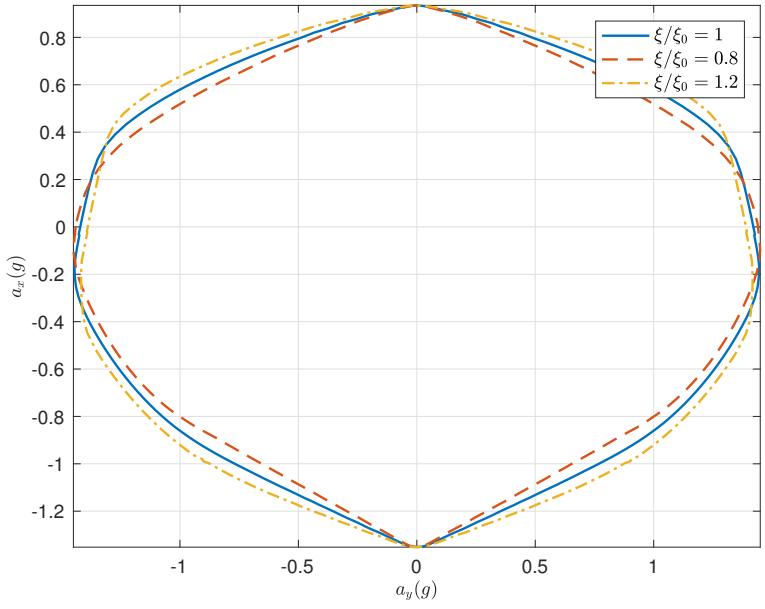


Figure 4.7: Car g-g diagrams at different roll stiffness ratio values ξ ($V = 20\,\mathrm{m/s}$).

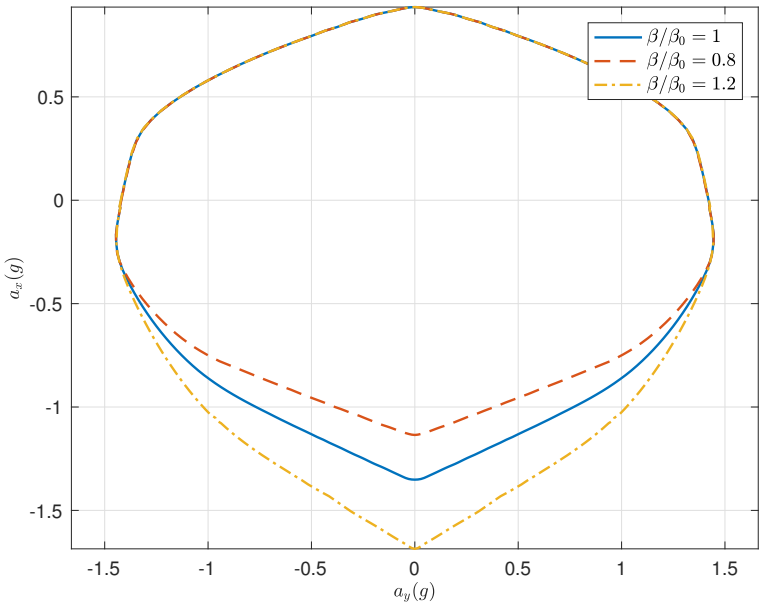


Figure 4.8: Car g-g diagrams at different bake ratio values β_b ($V = 20\,\mathrm{m/s}$).

increasing the roll-stiffness ratio is detrimental for the maximum achievable lateral acceleration, since the load transfer on the front axle is already large in the baseline configuration. However, while simultaneously braking (or accelerating) and cornering, increasing the roll stiffness ratio is beneficial, and higher deceleration (or acceleration) can be achieved. Indeed, in such condition the rear tyres saturate, and thus minimising the load transfer on the rear axle is beneficial for improving the performance, i.e. the g-g area. The optimal value of the roll-stiffness ratio is the best trade-off between these two effects.

Brake ratio

The g-g diagrams computed considering different values of front-to-total brake ratio, are shown in Fig. 4.8. This parameter is related to the front-to-rear brake ratio defined in Ch. 3 by $\gamma = \beta_b/(1-\beta_b)$. Increasing both β_b and γ means increasing the pressure delivered by the braking system to the front wheels, i.e. increasing the front braking power (or reducing the rear braking power). The braking performance increases when increasing β_b, until the saturation of front tyres is reached. Obviously, achieving the saturation is easier during combined lateral-longitudinal manoeuvres; for this reason the g-g assumes a sharper shape at high values of β_b, i.e. the braking performance is more affected by tyre-force combination.

4.4.2 Motorcycle model

The g-g diagram of the motorcycle, obtained with the essential model presented in Ch. 3, is shown in Fig. 4.9 at different speeds. The main differences with respect to the car g-g diagram (Fig. 4.4) are the 'dips' related to wheelie and stoppie con-

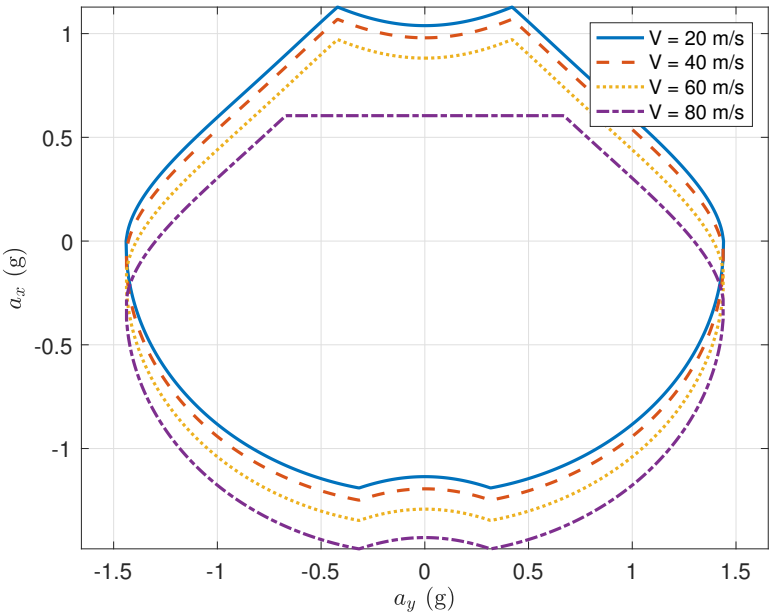


Figure 4.9: Baseline motorcycle g-g diagrams at different speeds.

ditions, which limit the motorcycle performance both in straight motion and with small lateral acceleration. Four parameters are taken into account for comparing the g-g diagrams at the same speed: the road adherence (modified through the tyre force scaling factors μ_x and μ_y), the distance between centre of gravity of the vehicle and the rear axle b, the height of the centre of mass h and the wheelbase w.

Road adherence

The g-g diagrams computed considering different values of road adherence, are shown in Fig. 4.10. This result is obtained by modifying the adherence parameters μ_x and μ_y that define the limits of the adherence ellipse used for this vehicle model (see Ch. 3). Halving the adherence reduces the longitudinal and lateral performance in any condition. Moreover, the effects of wheelie and stoppie are not visible since the adherence limit is achieved before lifting the front or rear tyre.

Rear to CoG distance

The g-g diagrams computed considering different values of the distance between the rear wheel and the centre of mass (without changing the wheelbase w), are shown in Fig. 4.11. As seen for the car, this parameter does not affect the pure lateral performance of the vehicle. Moving the centre of mass towards the rear (i.e. reducing b), reduces the effect of the stoppie during hard baking and increases the effect of the wheelie during acceleration, because the static load is augmented at the rear wheel.

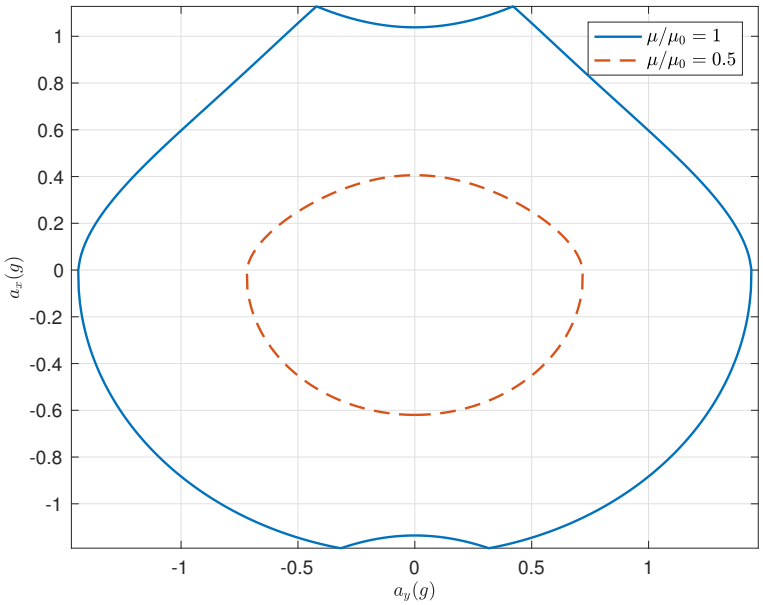


Figure 4.10: Motorcycle g-g diagrams at different adherence values μ ($V = 20\,\mathrm{m/s}$).

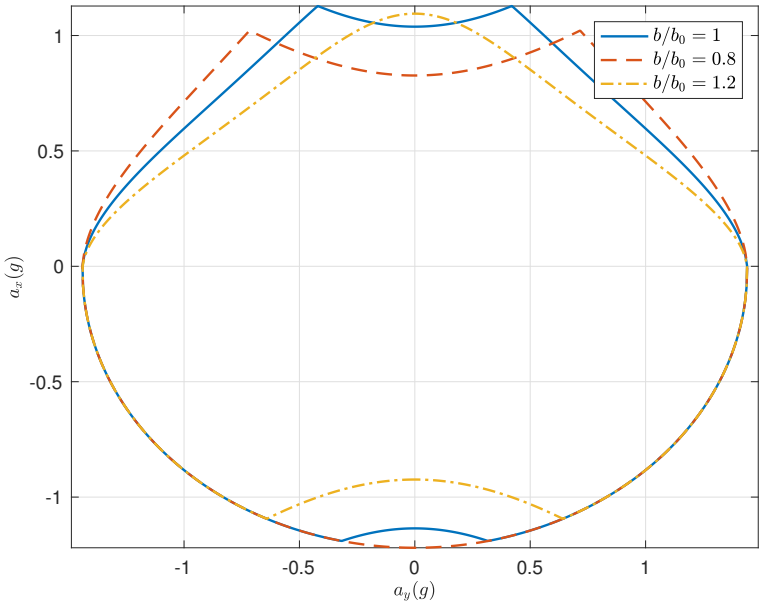


Figure 4.11: Motorcycle g-g diagrams at different b values ($V = 20\,\mathrm{m/s}$).

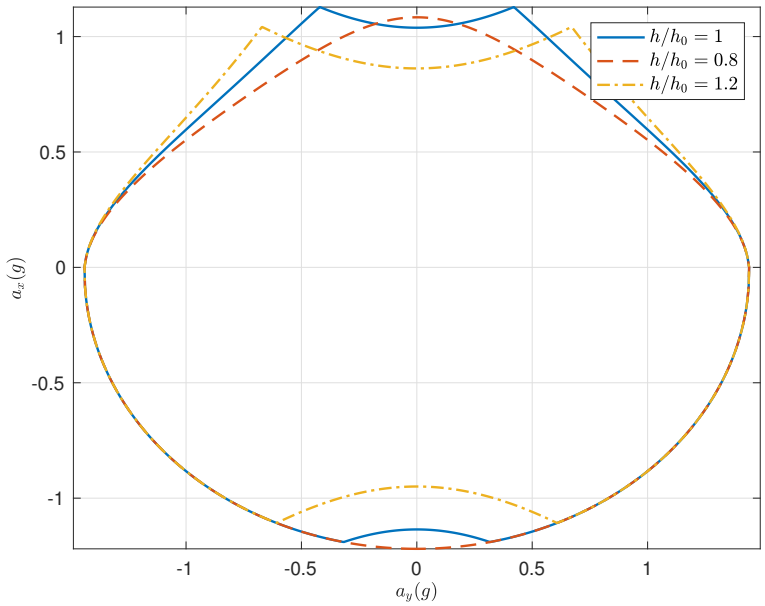


Figure 4.12: Motorcycle g-g diagrams at different h values ($V = 20\,\mathrm{m/s}$).

CoG height

The g-g diagrams computed considering different values of the height of the centre of mass, are shown in Fig. 4.12. Similarly to the position of the centre of mass, this parameter mainly affects the longitudinal performance of the vehicle. Higher values of h increase the longitudinal load transfer during acceleration and braking. For this reason both the wheelie and stoppie condition negatively affect the longitudinal performance while increasing h.

Wheelbase

The g-g diagrams computed considering different values of the vehicle wheelbase, are shown in Fig. 4.13. This parameter is changed while maintaining constant the b/w ratio, in order to highlight the only effects related to different values of w. As for the height of the centre of mass h, also w is related to the longitudinal load transfer of the vehicle, although the effects obtained while increasing this parameter are opposite to the ones obtained by increasing h. In fact, increasing the wheelbase reduces the longitudinal load transfer and, by consequence, decreases the effect of wheelie and stoppie to the longitudinal performance.

4.4.3 Comparison car and motorcycle

The presented car and motorcycle g-g diagrams, are obtained employing very different strategies. The car diagrams represent the solution of an optimisation problem for each g-g point, considering a simple but comprehensive vehicle model. This strategy has been extensively discussed in Sec. 4.3. Instead, the motorcycle diagrams are computed through a very simple model, that allows a fully-analytical

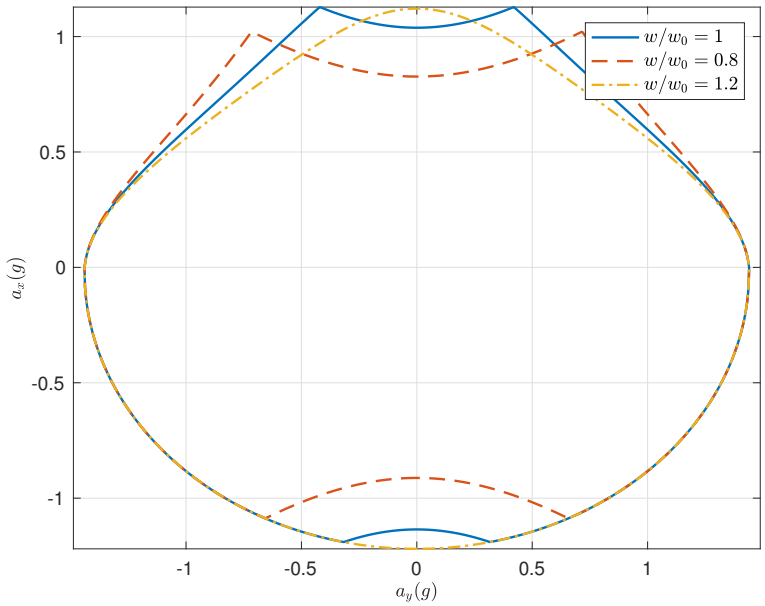
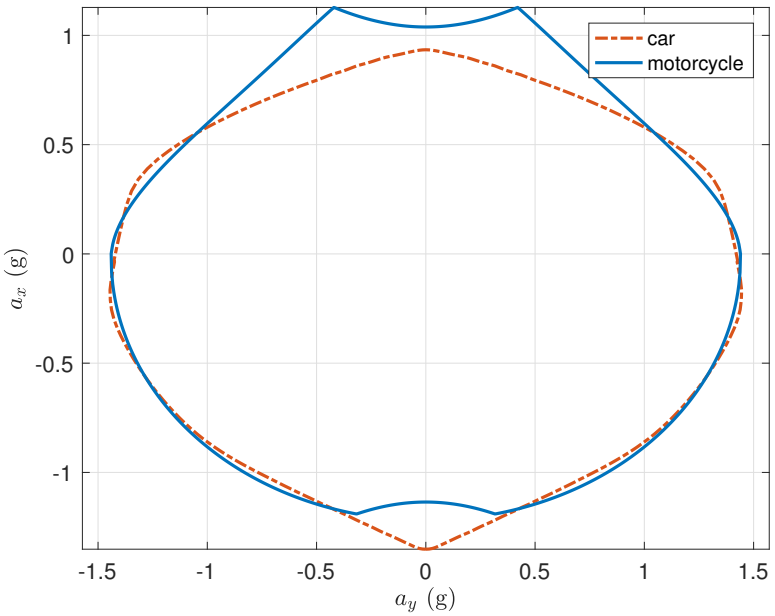
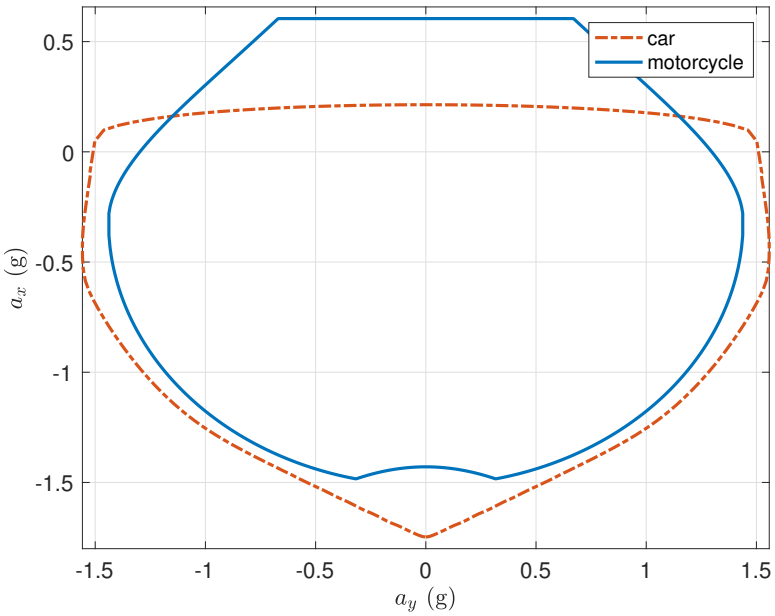


Figure 4.13: Motorcycle g-g diagrams at different wheelbase values w ($V = 20\,\mathrm{m/s}$).

description of the g-g (see Ch. 3). In order to compare these two modelling techniques, the friction coefficients of the motorcycle μ_x and μ_y have been chosen in order to have the maximum lateral acceleration and g-g diagram area similar to those of the race car at low speeds (indeed at high speeds the aerodynamics of the race car becomes a dominant factor); see Fig. 4.14 where the g-g diagrams are compared at 20 m/s and 80 m/s. The motorcycle g-g map includes the distinctive effects of wheelie and stoppie during acceleration and braking. This behaviour is typical for motorcycles and limits the maximum and minimum acceleration available (see Ch. 3). In spite of that, the motorcycle is capable of reaching higher accelerations, because of the larger height-to-wheelbase ratio (see Fig. 4.14a), lower drag area and higher power-to-mass ratio (see Fig. 4.14b). On the other hand, the downforces allow the car to potentially reach higher lateral accelerations (see Fig. 4.14b).



(a)



(b)

Figure 4.14: Car and motorcycle g-g diagrams at $V = 20\,\mathrm{m/s}$ (a) and $V = 80\,\mathrm{m/s}$ (b).

Chapter 5

Minimum-lap-time problems

5.1 Introduction

The use of minimum-lap-time simulations represents a widespread method for assessing and optimising the performance of a race vehicle in different conditions. This class of offline simulations permits to predict the lap-time achievable by a given vehicle, and to compute the quantities related to the model that describes the vehicle behaviour. These methods are employed both in a design stage and during the race weekends for the strategy analysis. The minimum-lap-time can be computed through either a quasi-steady-state approach, i.e. employing a steady-state model in each point of the simulation, or a dynamic approach, i.e. including in the model the transient behaviour of the vehicle; the trajectory travelled by the vehicle can be an input for the simulation (fixed-trajectory methods) or can be optimised while solving the problem (free-trajectory methods). The most common combinations reported in literature are basically two: quasi-steady-state models with fixed trajectory and dynamic models with free trajectory. The former approaches usually involve the computation of the well-known g-g map of the vehicle at different speed (g-g-speed surface), together with the determination of the corner apexes on the trajectory (i.e. the points at which the vehicle has maximum lateral acceleration). These steps usually identify an apex-finding method. The latter approaches usually involve the solution of a nonlinear-optimal-control problem, which may be solved either with direct or indirect methods (see Ch. 2). These two widespread minimum-time approaches have been combined in a third method, that employs a quasi-steady-state vehicle model together with a free-trajectory optimal-control simulation. This novel method will be discussed later in this chapter.

In the quasi-steady-state fixed-trajectory framework, many variants of the so-called apex-finding method have been employed for race car simulations. In [2] the minimum-lap-time of a quasi-steady-state vehicle model is computed for predetermined sections of the Paul Ricard circuit. A set of critical points is found for each corner of the track by simulating the model in constant-speed manoeuvres (e.g. steady-turning). The speed is increased until, at a certain point of the curve, a constraint is exceeded. At this point the maximum speed is achieved and the vehicle is assumed to switch between braking and acceleration. Between two critical points the speed profile is computed by merging the speed profiles

obtained in maximum accelerations and maximum braking simulations. In [30], a simple steady-state car model is employed for a quasi-steady-state fixed-trajectory simulation. The race line is obtained through optimisation of a set of meaningful points, in order to maximise the minimum cornering radius (i.e. maximising the steady-state cornering speed at the apex). These points are then interpolated through cubic splines. A common version of the apex-finding approach is then employed. In [31] and [32] the g-g-speed diagrams are produced for the implementation in an apex-finding method. In this case, the speed is computed for the corner apexes through the slip-limited g-g diagram. The speed profile is then computed between two apexes by merging the acceleration and braking speed profiles. An extension of the quasi-steady-state approach is presented in [36] for including the transient behaviour. The corner apexes are identified by calculating the maximum cornering speed, obtained on a given trajectory after maximising the sole lateral performance. The minima of this profile are the switching points between acceleration and braking phases. The acceleration and braking speed profiles are then calculated and merged by maximising the longitudinal performance between two switching points. This quasi-steady-state program is used iteratively in each segment for including the longitudinal, yaw and vertical dynamics of the vehicle, together with the transient tyre-temperature effects. Each dynamic state is computed from distinct space-dependent simulations, which employ specific dynamic models. The new values obtained for the dynamic-states represent the reference-state values in the following quasi-steady-state computation of the speed profile. This procedure is repeated until convergence.

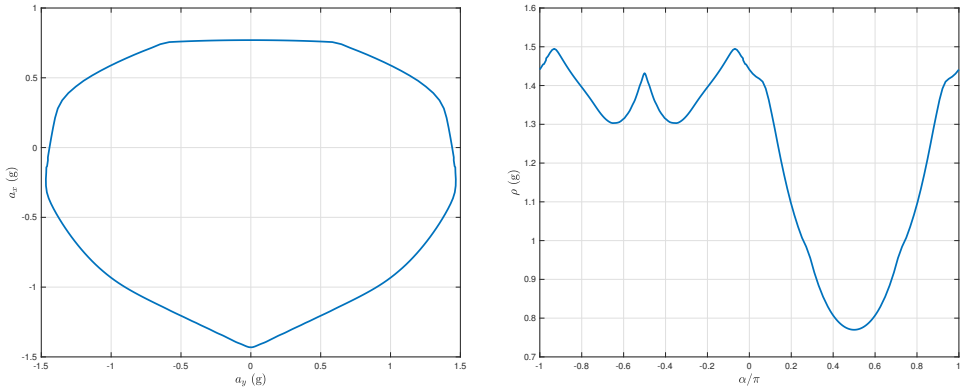
Many examples are available also for the implementation of dynamic models in optimal-control free-trajectory simulations. One of the earliest applications on race cars can be found in [3] for short manoeuvres with a single-track model. The optimal-control is solved through a direct approach based on a gradient method. In [4] a simple motorcycle model is used together with an indirect optimal-control method for assessing the maximum performance and the optimal race line on the Mugello circuit. A multiple-shooting algorithm is employed in [33] and [34] for computing the minimum-time optimal-control of a Formula One car model. The same vehicle model is used in [35], where the free-trajectory optimal-control is solved though sequential-quadratic-programming techniques. In [53], [54] and [44] an indirect collocation method is employed focusing on single cornering manoeuvres of race cars. The same method is used in [55] together with a quite comprehensive GP2 car model. A direct-collocation method is used in [38] and [39] for computing the free-trajectory optimal-control of a Formula One car. In [40] an indirect collocation method is used for assessing the performance and the optimal race line of a motorcycle model.

In this chapter some of the most widespread minimum-lap-time techniques are discussed in detail, with a special focus on their program implementations. In Sec. 5.2 the quasi-steady-state methods are presented. The fixed-trajectory apex-finding approach is discussed first. Then, a fixed-trajectory simulation is developed using optimal-control techniques. In this section, a novel approach for free-trajectory quasi-steady-state simulations employing optimal-control techniques is presented and compared with the other methods. This part of the chapter gives

an insight on the work published in [5] and represents one of the most important contributions of the entire thesis. The free-trajectory dynamic optimal-control approach is described in Sec. 5.3. An example of application is also carried out for the analysis of the effect of steering geometry on lap-time performance. This study is also published in [6].

5.2 Quasi-steady-state approach

The underlying concept of a quasi-steady-state (QSS) approach to a dynamic problem refers to the solution of a series of equilibrium (i.e. steady-state) manoeuvres of a given vehicle model. A set of mesh points need be defined on a track. In this way, the problem is discretised and the model equilibrium can be computed along a discrete path. The solution at each mesh point is affected by the previous-point solution, since a constant acceleration (or deceleration) profile is considered between two mesh points. It should be clear that, in this class of methods, the transient effects are neglected, since only steady-state solutions are considered. In this section, an overview of the different approaches available in the quasi-steady-state simulations framework is carried out, either considering a fixed (i.e. predetermined) or a free (i.e. non-predetermined) trajectory. The widespread fixed-trajectory apex-finding method is discussed first (Sec. 5.2.2). In this case, two strategies are available for the solution of the equilibrium at each mesh point. A first strategy relies on the generation of the g-g-speed diagrams of the vehicle (see Ch. 4). Each point of the g-g is obtained through the solution of a performance-optimisation problem, subject to the steady-state equations of the vehicle at a given speed. In this case, the optimisation is computed offline, and the previously-calculated g-g diagrams represent an input for the subsequent minimum-lap-time simulation. A second strategy consists in solving an optimisation problem at each mesh point of the minimum-lap-time simulation. The performance-optimisation problem is the same to the one defined for the g-g diagrams generation, although in this case its solution is computed online. These two strategies are equivalent if the underlying vehicle model is the same. Despite of that, in some cases the g-g-V diagrams need be computed considering a mesh of space-dependent parameters, which vary during a track lap. For instance, while using a three-dimensional road model, the g-g-V diagrams should be computed at different values of the road slopes and a three-dimensional description of the g-g diagrams is no more sufficient. This issue can be solved either employing an interpolation method in a the proper space-dimensionality (e.g. with a neural network approach) or solving the model steady-state equations considering the road characteristics at the current point of the track. In the following description, only the g-g-diagrams strategy is considered since, for the vehicle model employed (see Ch. 3), both the g-g-based and the point-by-point optimisation methods can be formulated employing the same steady-state equations. Moreover, the g-g-related method is adequate for the implementation in quasi-steady-state solutions obtained through an optimal-control strategy. Consequently, a fixed-trajectory optimal-control method is outlined in Sec. 5.2.3, while a novel free-trajectory optimal-control method is pre-



(a) (b)

Figure 5.1: Polar-coordinates transformation. Original g-g diagram (a) and correspondent version in polar coordinates (b). Note that the angle α is measured counterclockwise and the reference frame is centred in $(0, 0)$.

sented in Sec. 5.2.4. The different quasi-steady-state strategies are then compared (Sec. 5.2.5) and the sensitivity analysis on different parameters is assessed with reference to the lap time performance.

5.2.1 g-g surface interpolation

The g-g diagrams obtained in Ch.4 are consistent with the vehicle reference frame discussed in Ch.3. A change of coordinates need be introduced, in order to be consistent with the modelling choices implemented in the lap-time simulations. In the following sections, the acceleration of the vehicle are computed in the trajectory reference frame, in which the x-axis is tangent to the trajectory and the y-axis is normal to the trajectory. The frame transformation gives

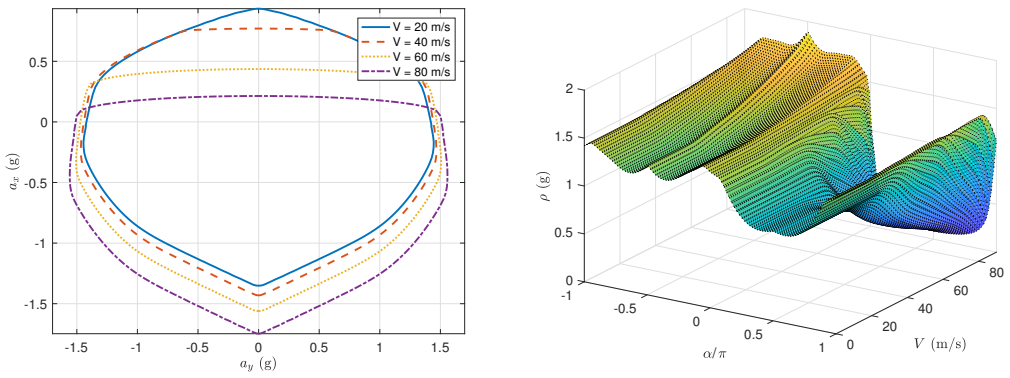
$$\begin{aligned} a_x &= \tilde{a}_x \cos\beta + \tilde{a}_y \sin\beta, \\ a_y &= \tilde{a}_x \sin\beta - \tilde{a}_y \cos\beta, \end{aligned} \tag{5.1}$$

where $\tilde{a}_x$ and $\tilde{a}_y$ are referred to the vehicle reference frame, while a_x and a_y refer to the trajectory reference frame, rotated by the vehicle-sideslip angle $\beta = \arctan(v/u)$. The g-g diagrams are then conveniently parametrised using polar coordinates (see Ch. 4):

$$\rho = \sqrt{\left(\frac{a_x}{g}\right)^2 + \left(\frac{a_y}{g}\right)^2}, \tag{5.2}$$

$$\alpha = \arctan\left(\frac{a_x}{a_y}\right). \tag{5.3}$$

where ρ is the adherence radius, and α is the related orientation. This will also make it easier to introduce the g-g constraint in the minimum-time-optimisation problem, which will be discussed in the next sections. The values of the adherence



(a) (b)

Figure 5.2: Spline interpolation. Original g-g diagrams at various speeds (a) and correspondent g-g-V surface (b).

radius at different speed are then interpolated by means of splines, to give the following surface

$$\rho = \rho(\alpha, V), \tag{5.4}$$

which is a function of the polar orientation α and the speed V. In Fig. 5.1 the original g-g diagram (a) is compared with the polar-coordinates representation (b) for a single speed. After producing the g-g diagrams at different speeds (Fig. 5.2a), the obtained g-g-V surface in polar coordinates is depicted in Fig. 5.2b.

5.2.2 Apex-finding method

The quasi-steady-state apex-finding approach represents a widely-employed method for solving minimum-lap-time problems. The simulation program allows to find the speed profile of a given vehicle along a given (i.e. fixed) trajectory. The simulation steps can be summarised as follows:

S.1 Identify the N_a curvature peaks (apexes) of the trajectory. It is assumed that in these points the lateral acceleration a_y is maximum, and the steady-state speed at the corner-apex is consequently

$$V_i = \sqrt{a_y^{max} \kappa_i}, \quad i = 1, \ldots, N_a, \tag{5.5}$$

where κ_i is the road curvature at apex i and a_y^{max} is the maximum lateral acceleration obtained from a 'low speed' g-g diagram. In other words, the speed at each corner apex is known before starting the program.

S.2 Corner exit. A mesh point j included in the interval defined by a pair of curvature peaks i and $i+1$ is now considered. The speed of the vehicle V_{j+1} at step $j+1$ can be computed by solving

$$a_{x,j} \approx \frac{\Delta V_j}{\Delta t_j} = \frac{V_{j+1} - V_j}{t_{j+1} - t_j} = a_{x,j}^{lim}\big|_{a_{y,j}^{lim}}, \tag{5.6}$$

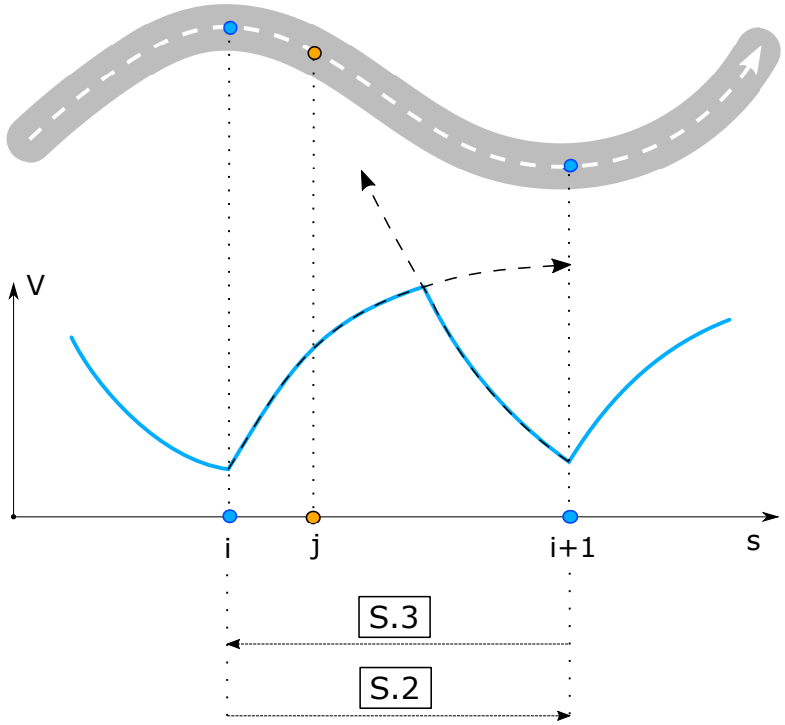


Figure 5.3: Apex-finding method.

where $a_{x,j}^{lim}$ is the maximum longitudinal acceleration, evaluated from the g-g-V diagrams at speed V_j and lateral acceleration $a_{y,j}^{lim} = V_j^2 \kappa_j$. The trajectory curvature κ_j at step j is given and the time-step Δt_j is calculated from the given trajectory-mesh-point distance $\Delta s_{T,j}$ as follows:

$$\Delta t_j = t_{j+1} - t_j = \frac{\Delta s_{T,j}}{\bar{V}} = 2\frac{s_{T,j+1} - s_{T,j}}{V_{j+1} + V_j}. \tag{5.7}$$

Eq. 5.6 is obtained considering that the vehicle employs the maximum acceleration achievable, while a constant acceleration profile is assumed in the mesh-point interval $(j, j+1)$. The overall speed profile during acceleration is obtained by repeating this step through a time-marching technique, i.e. the speed is computed step-by-step starting from the corner apex, in which the speed is known from (5.5).

S.3 Corner entrance. The same time-marching procedure described at step S.2 is employed for the deceleration manoeuvre. In this case a braking manoeuvre is assumed from apex i to the apex $i+1$, while in each mesh interval $(j, j+1)$ a constant deceleration profile is considered. The braking profile can be computed backwards from apex $i+1$ to apex i. In this case the speed V_{i+1} is known from Eq. 5.5, while the speed V_j is obtained from the solution of

$$a_{x,j+1} \approx \frac{\Delta V_j}{\Delta t_j} = \frac{V_j - V_{j+1}}{t_{j+1} - t_j} = a_{x,j+1}^{lim}\Big|_{a_{y,j+1}^{lim}}, \tag{5.8}$$

where $a_{x,j+1}^{lim}$ is the minimum longitudinal acceleration, computed from the

g-g-V diagrams at speed V_{j+1} and lateral acceleration $a_{y,j+1}^{lim} = V_{j+1}^2 \kappa_{j+1}$. The time-step Δt_j is obtained from (5.7).

S.4 Both a speed profile for acceleration and for deceleration are obtained between apex i and $i+1$. A *crossover point* is identified at the intersection of the two curves. The speed of the vehicle between two corners is then determined by joining the acceleration profile before the crossover point and the deceleration profile after the crossover point.

In summary, the corner apexes need be identified first: at each apex, where the trajectory curvature is maximum, the vehicle is assumed to have maximum lateral acceleration. Before the apex, the vehicle is braking along the g-g envelope, while after the apex the tyres are engaged with traction forces in order to keep the vehicle along the g-g envelope. The speed profiles between two apexes (acceleration out of turn i and deceleration into turn $i+1$) are then connected at their intersection point. Sometimes it is assumed that at each apex the longitudinal acceleration is zero, which is a slightly different condition (when compared with the maximum lateral acceleration condition): in this scenario, after the apex the vehicle is accelerating. It should be noticed that the apex-finding approach to minimum-time problems is based on the assumption that the corner apex (i.e. the point in which the curvature presents a maximum) is the point in which the vehicle instantly switches from braking to acceleration. This does not necessarily represent an accurate description of a real manoeuvre, in which different strategies might be preferred by the driver (e.g. acceleration/braking in straight and steady-state cornering).

Basic program

A basic version of the minimum-lap-time apex-finding program described in the previous section, is now presented for underlining the core aspects of this strategy. The vehicle model consists of a single-mass point and its g-g-V maps are described by the following analytical expression of the adherence limit at point $j \in (i, i+1)$

$$\left(\frac{1}{\mu_x}\frac{a_{x,j}}{g}\right)^2 + \left(\frac{1}{\mu_y}\frac{a_{y,j}}{g}\right)^2 = 1, \tag{5.9}$$

where μ_x and μ_y represent the maximum (normalised) longitudinal and lateral accelerations respectively and $a_{y,j} = V_j^2 \kappa_j$. In this case the g-g-V diagrams of the vehicle are described analytically by the adherence ellipse (5.9), which is not affected by the speed (i.e. it is constant). For the acceleration speed profile (see step S.2), the acceleration of the vehicle at point $j \in (i, i+1)$ is

$$a_{x,j} = \frac{V_{j+1}^2 - V_j^2}{2\,\Delta s_{T,j}}, \tag{5.10}$$

where $\Delta s_{T,j}$ is the mesh-interval of the curvilinear coordinate computed on the trajectory. Then, the speed V_{j+1} of the vehicle is computed from the speed V_j at

the previous step, by solving

$$\frac{V_{j+1}^2 - V_j^2}{2\,\Delta s_{T,j}} = \tilde{\mu}_x\, g \sqrt{1 - \left(\frac{a_{y,j}}{\mu_y\, g}\right)^2}, \tag{5.11}$$

which is obtained from a combination of (5.9) and (5.10). The maximum normalised acceleration in longitudinal direction $\tilde{\mu}_x$ is defined as

$$\tilde{\mu}_x = \min\left(\mu_x,\, \mu_{x,P}\right), \tag{5.12}$$

where μ_x refers to the longitudinal limit of the adherence ellipse (5.9), while $\mu_{x,P}$ represents the longitudinal acceleration limit due to the limited power displaced by the engine. Considering the longitudinal equilibrium of the single-mass point

$$ma_x = F_x - F_D, \tag{5.13}$$

the acceleration limit $\mu_{x,P}$ is computed as

$$\mu_{x,P} = \frac{P_{max}}{m\, g\, V_j} - \frac{F_D}{m\, g}, \tag{5.14}$$

where $P_{max} = F_x V_j$ is the maximum engine power and $F_D = 1/2\rho C_D A V_j^2$ is the aerodynamic drag-force. For the deceleration profile (see step S.3) the acceleration limit becomes

$$\tilde{\mu}_x = -\mu_x - \frac{F_D}{mg}, \tag{5.15}$$

and the speed at step V_j is computed backwards from the speed at step V_{j+1} by solving (5.11). The speed profiles obtained for the forward and backward solution are then joined at their intersection point (S.4) and the procedure is repeated for each apex pair.

Program based on g-g

The same program described in steps S.1 - S.4 is now employed together with the g-g-V surface, which summarises the quasi-steady-state performance of the vehicle. The generation of the g-g-V is reported in Ch. 4, while the interpolation method used for the implementation in the minimum-time problems is discussed in Sec. 5.2.1.

Similarly to the previous method, the speed profile (in acceleration and braking) is obtained by solving (5.6) for the speed V_j at point $j \in (i, i+1)$; nevertheless, in this case, the maximum longitudinal acceleration (or deceleration) is not constant, as occurs in (5.11), but is a function of the speed V_j and the maximum lateral acceleration $a_{y,j}^{lim}$. The g-g diagram at a given speed V_j is obtained by interpolating the surface spline described by (5.4). Then, a different formulation is introduced for positive and negative acceleration, at constant speed

$$\begin{aligned} a_x^+(\alpha^+) &= \rho(\alpha^+, V_j)\sin(\alpha^+), \qquad & a_y^+(\alpha^+) &= \rho(\alpha^+, V_j)\cos(\alpha^+), \\ a_x^-(\alpha^-) &= \rho(\alpha^-, V_j)\sin(\alpha^-), \qquad & a_y^-(\alpha^-) &= \rho(\alpha^-, V_j)\cos(\alpha^-), \end{aligned} \tag{5.16}$$

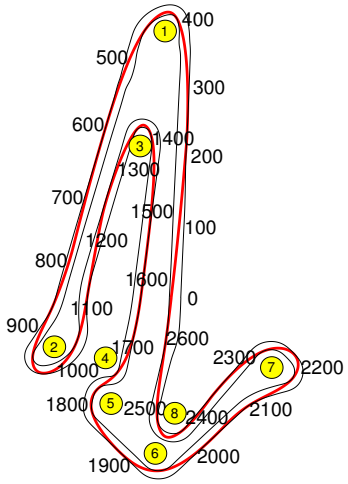
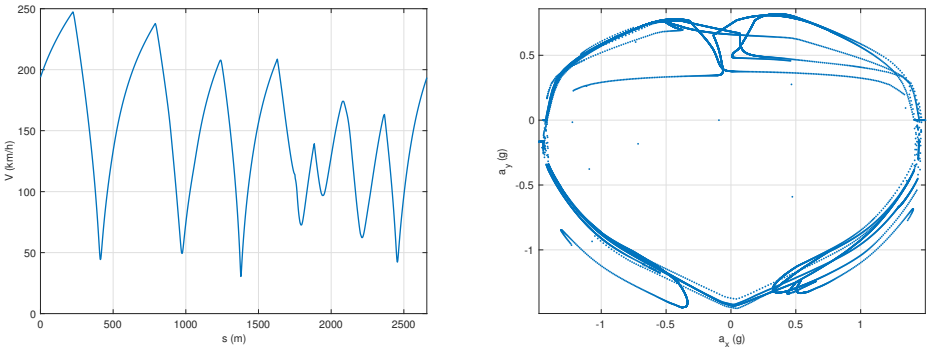


Figure 5.4: Trajectory of the baseline car on the Adria International Raceway, with corner numbers and distance (curvilinear abscissa s) from the start/finish line.

where $\alpha^+ \in (0, \pi)$ and $\alpha^- \in (-\pi, 0)$. The speed profile during acceleration (S.2) is now considered. The maximum lateral acceleration is given by $a_{y,j}^{lim} = V_j^2 \kappa$, while the correspondent adherence angle $\tilde{\alpha}$ is obtained by solving the equality $a_y^+(\alpha^+) = a_{y,j}^{lim}$, i.e. finding the angle $\tilde{\alpha}$ at which the equation is verified. Then, the maximum longitudinal acceleration $a_{x,j}^{lim}$ is computed from a_x^+ at the same adherence angle $\tilde{\alpha}$, i.e. $a_{x,j}^{lim} = a_x^+(\tilde{\alpha})$. For the braking profile (S.3) the procedure is similar. The adherence angle $\tilde{\alpha}$ at which the lateral acceleration assumes the value $a_{y,j}^{lim} = V_j^2 \kappa$, is obtained from equation $a_y^-(\alpha^-) = a_{y,j}^{lim}$. The maximum longitudinal deceleration is then $a_{x,j}^{lim} = a_x^-(\tilde{\alpha})$. These steps are repeated for updating $a_{x,j}^{lim}$ in (5.6) while solving for V_j. The speed profiles obtained for the forward and backward solution are then joined at their intersection point (S.4) and the procedure is repeated for each apex pair.

Car simulation

The g-g-based apex-finding method is employed for simulating the minimum-lap-time of the race car described in Ch. 3, on the Adria International Raceway. The g-g-V diagrams used for this simulation are obtained from a dual-track vehicle model, while the predetermined trajectory is computed offline and given as an input to the minimum-time program (see Fig. 5.4). The solution is achieved with a 0.1 m equally spaced mesh. The resulting speed profile and g-g map are reported in Fig. 5.5. The top speed is 245 km/h in the pit straight, while the minimum speed is 31 km/h at turn 3. The acceleration boundary is limited by the given g-g diagram, with a 1.45 g maximum lateral acceleration, a -1.45 g maximum braking deceleration and a 0.8 g maximum acceleration. The lap-time is 75.450 s.



(a) (b)

Figure 5.5: Apex-finding simulation results for the car model. Speed profile (a) and g-g diagram (b).

Motorcycle simulation

The g-g based apex-finding method is also employed for simulating the minimum-lap-time of the race motorcycle described in Ch. 3, on the Adria International Raceway. The g-g-V diagrams used for this simulation are obtained from a basic vehicle model, while the predetermined trajectory is computed offline and given as an input to the minimum-time program (see Fig. 5.6). The solution is achieved with a 0.1 m equally spaced mesh. The resulting speed profile and g-g map are reported in Fig. 5.7. The top speed is 281 km/h in the pit straight, while the minimum speed is 34 km/h at turn 3. The acceleration boundary is limited by the given g-g diagram, with a 1.44 g maximum lateral acceleration, a -1.29 g maximum braking deceleration and a 1.08 g maximum acceleration. The lap-time is 72.530 s.

5.2.3 Fixed-trajectory OCP

A minimum-lap-time problem with fixed trajectory can be also solved by means of an optimal-control problem. In this case a set of state and controls need be introduced to formulate the OCP (see Ch. 2). The main difference with respect to the apex-finding method (Sec. 5.2.2) consists in the fact that, in this case, the apexes are automatically identified by the optimiser and no assumption on their locations need be made.

Basic program

In a basic formulation of the OCP, the numerical model consists in a single-mass point that moves along the given trajectory by selecting the optimal throttle input. Its performance is limited by a constant adherence ellipse. The only state variable is the speed along the trajectory

$$\mathbf{x} = V, \tag{5.17}$$

while the only control variable is the longitudinal force employed by the mass-point, i.e. the force along the tangent to the trajectory that allows the vehicle to

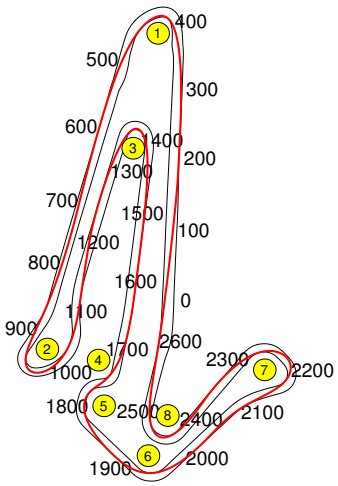
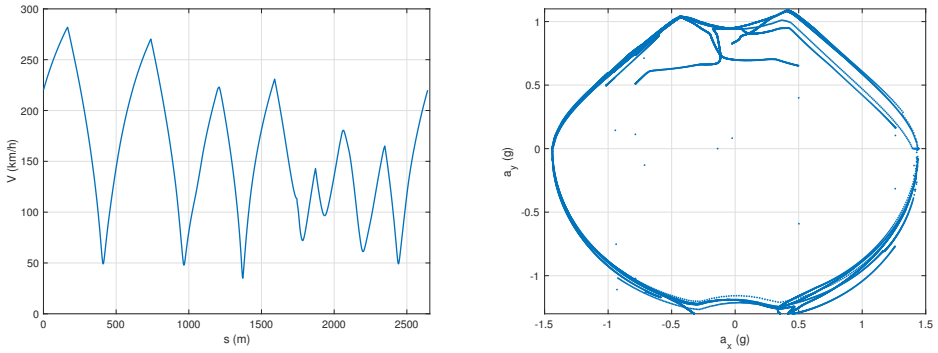


Figure 5.6: Trajectory of the baseline motorcycle on the Adria International Raceway, with corner numbers and distance (curvilinear abscissa s) from the stat/finish line.



(a) (b)

Figure 5.7: Apex-finding simulation results for the motorcycle model. Speed profile (a) and g-g diagram (b).

accelerate and brake

$$\mathbf{u} = F_x. \tag{5.18}$$

The related state equation is

$$\frac{dV}{ds_T} = \frac{dV/dt}{ds_T/dt} = \frac{\dot{V}}{V} = \frac{1}{m\,V}\,(F_x - F_D), \tag{5.19}$$

where s_T is the curvilinear coordinate along the trajectory and $F_D = 1/2\rho C_D A V^2$ is the drag force.

Two constraints are included: the first one takes account for the limited engine-power

$$F_x V \leq P_{max}, \tag{5.20}$$

while the second consist in the adherence limit

$$\left(\frac{1}{\mu_x}\frac{F_x}{mg}\right)^2 + \left(\frac{1}{\mu_y}\frac{V^2\kappa}{g}\right)^2 = 1, \tag{5.21}$$

where μ_x and μ_y represent the constant longitudinal and lateral adherence limits respectively, while κ represents the curvature of the given trajectory, which is employed for computing the steady-state lateral acceleration $a_y = V^2\kappa$. It is noticed that the adherence ellipse (5.21) represents a tyre-related boundary, which emulates the performance limits related to the maximum tyre-forces available.

The target function of the OCP is the total-manoeuvre time, which is computed from the vehicle speed

$$\mathcal{J} = \int (1 + \mathbf{p}(\mathbf{u}))dt = \int \frac{1}{\dot{s}_T}(1 + \mathbf{p}(\mathbf{u}))\,ds_T = \int \frac{1}{V}(1 + \mathbf{p}(\mathbf{u}))\,ds_T. \tag{5.22}$$

The penalty term $\mathbf{p}(\mathbf{u})$ is employed to penalise the controls

$$\mathbf{p}(\mathbf{u}) = wF_x^2, \tag{5.23}$$

where w is the penalty weight that have to be tuned for improving the convergence of the OCP ($\sim 10^{-4}$).

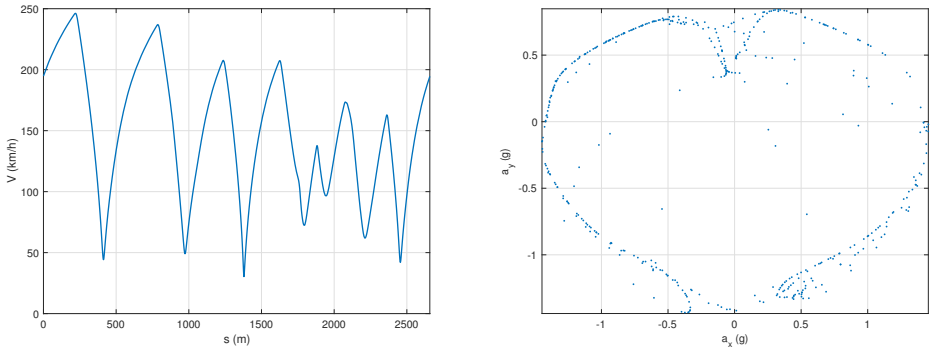
Program based on g-g

In case the performance is limited by the g-g-V surface (see Sec. 5.2.1), the OCP is obtained though a slightly different formulation. The numerical model still consists in a single-mass point that moves along the given trajectory by selecting the optimal throttle input. The state variable vector is

$$\mathbf{x} = \{V, a_x\}^T, \tag{5.24}$$

where V is the vehicle speed and $a_x = \dot{V}$ is the longitudinal acceleration directed along the trajectory. The only control variable is the time-derivative of a_x,

$$\mathbf{u} = \frac{da_x}{dt}. \tag{5.25}$$



(a) (b)

Figure 5.8: Fixed-trajectory optimal-control simulation results for the car model. Speed profile (a) and g-g diagram (b).

This choice allows to limit the time-derivative of a_x in order to avoid vibrations of the states and to improve the convergence of the simulation. The drawback of this strategy is that the number of states increases. The state equations are

$$\frac{dV}{ds_T} = \frac{1}{V} a_x, \tag{5.26}$$

$$\frac{da_x}{ds_T} = \frac{1}{V} \frac{da_x}{dt}. \tag{5.27}$$

Equations (5.19) and (5.26) are both expressions of the quasi-static equilibrium in the longitudinal direction, although in the latter, the acceleration a_x represents the overall longitudinal performance of the vehicle, including the effect of drag and limited power, as in the g-g diagrams definition. Instead, the force F_x in (5.19) is only comprehensive of the tyre-friction effects. One constraint is employed, for limiting the vehicle acceleration inside the given g-g envelope

$$\rho \leq \rho_{\max}, \tag{5.28}$$

where ρ is the adherence radius (5.2) and the lateral acceleration is $a_y = V^2\kappa$, where κ is the curvature of the given trajectory. The adherence radius limit $\rho_{\max}$ is computed from the g-g-V surface spline (5.4), after obtaining the current adherence angle α from (5.3).

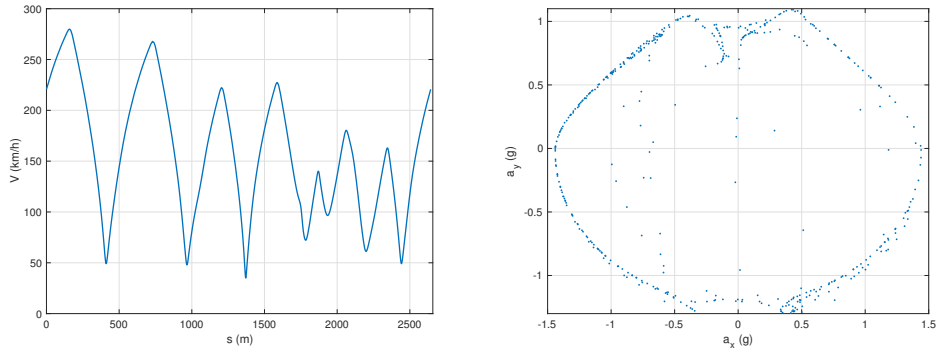
The OCP target is given by (5.22), while the penalty term is

$$\mathbf{p}(\mathbf{u}) = w \left(\frac{da_x}{dt} \right)^2, \tag{5.29}$$

where w is the penalty weight ($\sim 10^{-2}$).

Car simulation

The g-g-based fixed-trajectory-OCP method is employed for simulating the minimum-lap-time of the GT3 vehicle described in Ch. 3 on the Adria International Raceway. The g-g diagrams and the predetermined trajectory (see Fig. 5.4) are the



(a) (b)

Figure 5.9: Fixed-trajectory optimal-control simulation results for the motorcycle model. Speed profile (a) and g-g diagram (b).

same of the apex-finding method (Sec. 5.2.2). The OCP is solved through direct collocation by GPOPS-II, while IPOPT is used for solving each optimisation step employing the derivatives computed by ADiGator. An unevenly-spaced mesh in the range 25-100 m is provided at the first iteration. The resulting speed profile and g-g map are reported in Fig. 5.8. The solution of the OCP is achieved with an unevenly-spaced mesh, included in the range 0.1-10 m. Similarly to the apex-finding method, the top speed is 245 km/h in the pit straight, while the minimum speed is 31 km/h at turn 3. The acceleration boundary is limited by the given g-g diagram, with a 1.45 g maximum lateral acceleration, a -1.45 g maximum braking deceleration and a 0.8 g maximum acceleration. The lap-time is 75.456 s.

Motorcycle simulation

The g-g-based fixed-trajectory-OCP is also employed for simulating the minimum-lap-time of the race motorcycle described in Ch. 3, on the Adria International Raceway. The g-g diagrams and the predetermined trajectory (see Fig. 5.6) are the same of the apex-finding method (Sec. 5.2.2). Again, the OCP is solved through direct collocation by GPOPS-II, while IPOPT is used for solving each optimisation step employing the derivatives computed by ADiGator. An unevenly-spaced mesh in the range 25-100 m is provided at the first iteration. The solution of the OCP is achieved with an unevenly-spaced mesh (in the range 0.1-10 m). The resulting speed profile and g-g map are reported in Fig. 5.7. As for the apex-finding method, the top speed is 281 km/h in the pit straight, while the minimum speed is 34 km/h at turn 3. The acceleration boundary is limited by the given g-g diagram, with a 1.44 g maximum lateral acceleration, a -1.29 g maximum braking deceleration and a 1.08 g maximum acceleration. The lap-time is 72.543 s.

5.2.4 Free-trajectory OCP

A third approach can be employed by combining a free-trajectory minimum-time optimal-control problem together with the quasi-steady-state approach adopted

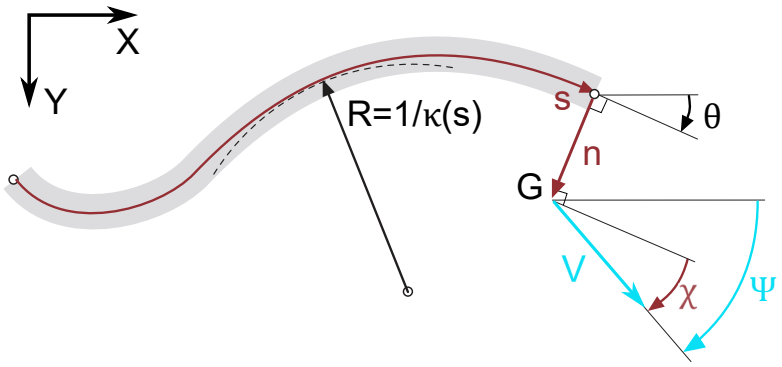


Figure 5.10: Curvilinear coordinates.

for the generation of the g-g diagrams. This novel approach is extensively discussed in [5].

Basic program

A basic version of the method is discussed first, employing a constant adherence ellipse constraint instead of the speed-dependent g-g diagrams. The model used in the OCP consists of a three-degrees-of-freedom model, which is constrained to move within the road borders, while satisfying the adherence limits.

The model state variables $\mathbf{x}$ are the absolute velocity V along the trajectory, the lateral position n with respect to the road centre line, and the relative orientation χ with respect to the tangent to the centre line (see Fig. 5.10):

$$\mathbf{x} = \{V,\, n,\, \chi\}^T. \tag{5.30}$$

The related state equations (in time domain) are

$$\dot{V} = \frac{1}{m}(F_x - F_D), \tag{5.31}$$

$$\dot{n} = \sin(\chi)\, V, \tag{5.32}$$

$$\dot{\chi} = \frac{d\chi}{dt}, \tag{5.33}$$

where F_x is the longitudinal traction force and F_D is the aerodynamic drag force. The control vector $\mathbf{u}$ of the OCP is

$$\mathbf{u} = \left\{F_x, \frac{d\chi}{dt}\right\}^T. \tag{5.34}$$

The model can be conveniently rewritten in the space domain

$$\mathbf{x}' = \frac{d\mathbf{x}}{ds} = \frac{\dot{\mathbf{x}}}{\dot{s}}, \tag{5.35}$$

using the curvilinear coordinate s along the centre line (see Fig. 5.10)

$$\dot{s} = \frac{\cos(\chi)}{1 - n\kappa}\, V, \tag{5.36}$$

where κ is the curvature of the road centre-line.

Moreover, the vehicle is constrained to move within the road borders

$$-r_{wl} \leq n \leq r_{wr}, \tag{5.37}$$

where r_{wl} and r_{wr} represent the left and right road limits, with a limited amount of power available

$$F_x V \leq P_{\max}, \tag{5.38}$$

where $P_{\max}$ is the maximum power produced by the engine. The adherence limit need also to be satisfied

$$\left(\frac{1}{\mu_x}\frac{F_x}{mg}\right)^2 + \left(\frac{1}{\mu_y}\frac{V\dot{\psi}}{g}\right)^2 = 1, \tag{5.39}$$

where the term $V\dot{\psi}$ represents the steady-state lateral acceleration and $\dot{\psi}$ is given by

$$\dot{\psi} = \dot{\chi} + \frac{\cos\chi}{1 - n\kappa} V\kappa. \tag{5.40}$$

The target $\mathcal{J}$ of the OCP is the manoeuvre time, which can be computed from the speed along the centre-line as follows

$$\mathcal{J} = \int (1 + \mathbf{p}(\mathbf{u}))dt = \int \frac{1}{\dot{s}}(1 + \mathbf{p}(\mathbf{u}))\, ds, \tag{5.41}$$

where

$$\mathbf{p}(\mathbf{u}) = w_{F_x} F_x^2 + w_\chi \left(\frac{d\chi}{dt}\right)^2, \tag{5.42}$$

where w_{F_x} ($\sim 10^{-4}$) and w_χ ($\sim 10^{-6}$) are the penalty weights for the controls. It is noticed that in this case the OCP integration domain is the curvilinear coordinate of the road centre-line s, while for the fixed trajectory methods (see Sec. 5.2.2-5.2.3) the curvilinear coordinate of the (given) trajectory s_T is selected as integration domain.

Program based on g-g

When including the g-g-V diagrams in the problem to summarise the vehicle performance, a slightly different formulation need be employed. In this case, the model used in the OCP consists of a three-degrees-of-freedom model, which is constrained to move within the road borders, while satisfying the limits of the g-g-speed surface. All the complexity of the vehicle, e.g. tyre nonlinearities, aerodynamics, etc., are included in the g-g map. Similarly to the basic description, the model state variables $\mathbf{x}$ are the absolute velocity V along the trajectory, the lateral position n with respect to the road centre-line, and the relative orientation χ with respect to the tangent to the centre-line (see Fig. 5.10):

$$\mathbf{x} = \{V,\, n,\, \chi\}^T. \tag{5.43}$$

The related state equations (in time domain) are

$$\dot{V} = a_x, \tag{5.44}$$
$$\dot{n} = \sin(\chi)\, V, \tag{5.45}$$
$$\dot{\chi} = a_y/V - \dot{\theta}, \tag{5.46}$$
$$\dot{\rho} = \frac{d\rho}{dt}, \tag{5.47}$$
$$\dot{\alpha} = \frac{d\alpha}{dt}, \tag{5.48}$$

where ρ is the adherence radius (5.2) and α is the adherence orientation (5.3), while θ is the orientation of the road centre-line with respect to the absolute reference frame. Consequently, the longitudinal acceleration is $a_x = \rho \sin\alpha$ and the lateral acceleration is $a_y = \rho\cos\alpha$. The first term on the right-hand side of (5.46) arises from the equation $a_y = \dot{\psi}V = \Omega V$, where $\dot{\psi} = \Omega$ is the yaw rate. The control vector $\mathbf{u}$ of the OCP consists of the time-derivatives of the adherence radius and adherence angle

$$\mathbf{u} = \left\{ \frac{d\rho}{dt}, \frac{d\alpha}{dt} \right\}^T. \tag{5.49}$$

It should be noticed that the equations (5.47) and (5.48) have been introduced in order to apply the control law to the state derivatives and, consequently, to obtain smoother acceleration profiles and improve the OCP convergence rate. The vehicle is constrained to move within the road borders

$$-r_{wl} \leq n \leq r_{wr}, \tag{5.50}$$

where r_{wl} and r_{wr} represent the left and right road limits, and to satisfy the g-g envelope

$$\rho \leq \rho_{\max}, \tag{5.51}$$

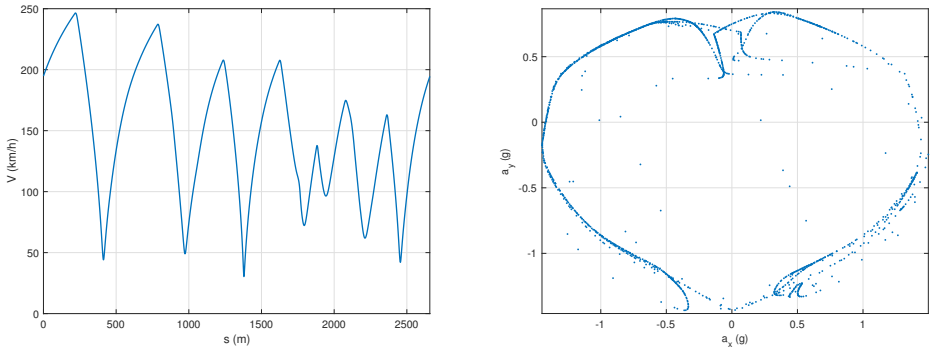
where $\rho_{\max} = \rho_{\max}(\alpha, V)$ is the maximum value of the adherence radius ρ (given by (5.2)) at the current orientation α (given by (5.3)) and speed V.

Also in this case, the problem is written in the centre-line curvilinear coordinate domain (see (5.35)-(5.36)) and the OCP target is the same as in (5.41). The penalty term is

$$\mathbf{p}(\mathbf{u}) = w_\rho \left(\frac{d\rho}{dt} \right)^2 + w_\alpha \left(\frac{d\alpha}{dt} \right)^2, \tag{5.52}$$

where w_ρ ($\sim 10^{-4}$) and w_α ($\sim 10^{-3}$) are the penalty weights related to the adherence radius and orientation.

The OCP can be solved numerically using direct or indirect methods [20]. It is noted that the resulting OCP is simple, although the underlying vehicle model can be complex.



(a) (b)

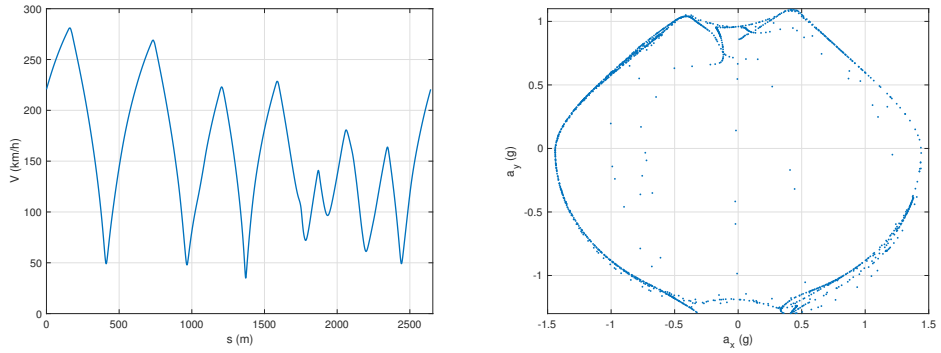
Figure 5.11: Free-trajectory optimal-control simulation results for the car model. Speed profile (a) and g-g diagram (b).

Car simulation

Similarly to the other quasi-steady-state methods, the g-g based free-trajectory-OCP approach is employed for simulating the minimum-lap-time of the GT3 vehicle described in Ch. 3 on the Adria International Raceway. The OCP is solved through direct collocation by GPOPS-II, while IPOPT is used for solving each optimisation step employing the derivatives computed by ADiGator. An unevenly-spaced mesh in the range 25-100 m is provided at the first iteration. In this case, the g-g-V diagrams are the same employed for the apex-finding and fixed-trajectory OCP, while the trajectory is a result of the simulation (see Fig. 5.4). It should be noticed that the resulting race-line is used an input for the fixed-trajectory simulations in Sec. 5.2.2 and 5.2.3. The resulting speed profile and g-g map are reported in Fig. 5.11. The solution of the OCP is achieved with an unevenly-spaced mesh, included in the range 0.1-10 m. Similarly to the fixed-trajectory methods, the top speed is 245 km/h in the pit straight, while the minimum speed is 31 km/h at turn 3. The acceleration boundary is limited by the given g-g diagram, with a 1.45 g maximum lateral acceleration, a -1.45 g maximum braking deceleration and a 0.8 g maximum acceleration. The lap-time is 75.451 s.

Motorcycle simulation

The g-g-based free-trajectory-OCP is also employed for simulating the minimum-lap-time of the race motorcycle described in Ch. 3, on the Adria International Raceway. As for the car, the g-g diagrams are the same employed for the fixed-trajectory methods. In this case, the trajectory is a result of the simulation (see Fig. 5.6) and it is used as an input for both the apex-finding and the fixed-trajectory OCP simulations. Again, the OCP is solved through direct collocation by GPOPS-II, while IPOPT is used for solving each optimisation step employing the derivatives computed by ADiGator. An unevenly-spaced mesh in the range 25-100 m is provided at the first iteration. The resulting speed profile and g-g map are reported in Fig. 5.7. The solution of the OCP is again achieved with



(a) (b)

Figure 5.12: Free-trajectory optimal-control simulation results for the motorcycle model. Speed profile (a) and g-g diagram (b).

an unevenly-spaced mesh (in the range 0.1-10 m). As for the other methods, the top speed is 281 km/h in the pit straight, while the minimum speed is 34 km/h at turn 3. The acceleration boundary is limited by the given g-g diagram, with a 1.44 g maximum lateral acceleration, a -1.29 g maximum braking deceleration and a 1.08 g maximum acceleration. The lap-time is 72.519 s.

5.2.5 Comparison

The presented quasi-steady-state methods achieve consistent results in terms of lap time, speed profiles and g-g envelopes. Nevertheless, the resulting computational effort is significantly different because of the different formulations employed. The computational effort is compared for the baseline car and motorcycle simulations through the inspection of the computation time needed and the problem size. The latter is measured through the number of equations that have to be solved and the number of related solver iterations. The fixed and free-trajectory-OCP methods are solved using GPOPS-II, while IPOPT is employed as optimisation solver together with ADiGator for computing the derivatives (see Ch. 2). The initial mesh in this case is unevenly-spaced in the range 25-100 m and an adaptive-mesh-refinement method is used for computing the solution at each mesh iteration, giving a resulting set of unevenly-spaced points. Differently, for the apex-finding method a 1 m equally-spaced mesh is employed. Since no optimisation is involved, the solution is computed in each point of the initial mesh and no refinement is needed. It should be noticed that the fixed-trajectory methods (i.e. apex-finding and fixed-trajectory OCP) require some pre-processing work in order to obtain a suitable race-line. Either an experimental race-line (obtained from telemetry) or a numerical race-line (e.g. computed using a free-trajectory optimisation) can be employed. In this case, the given trajectory is the one computed from the free-trajectory OCP approach (see Fig. 5.4 and 5.6).

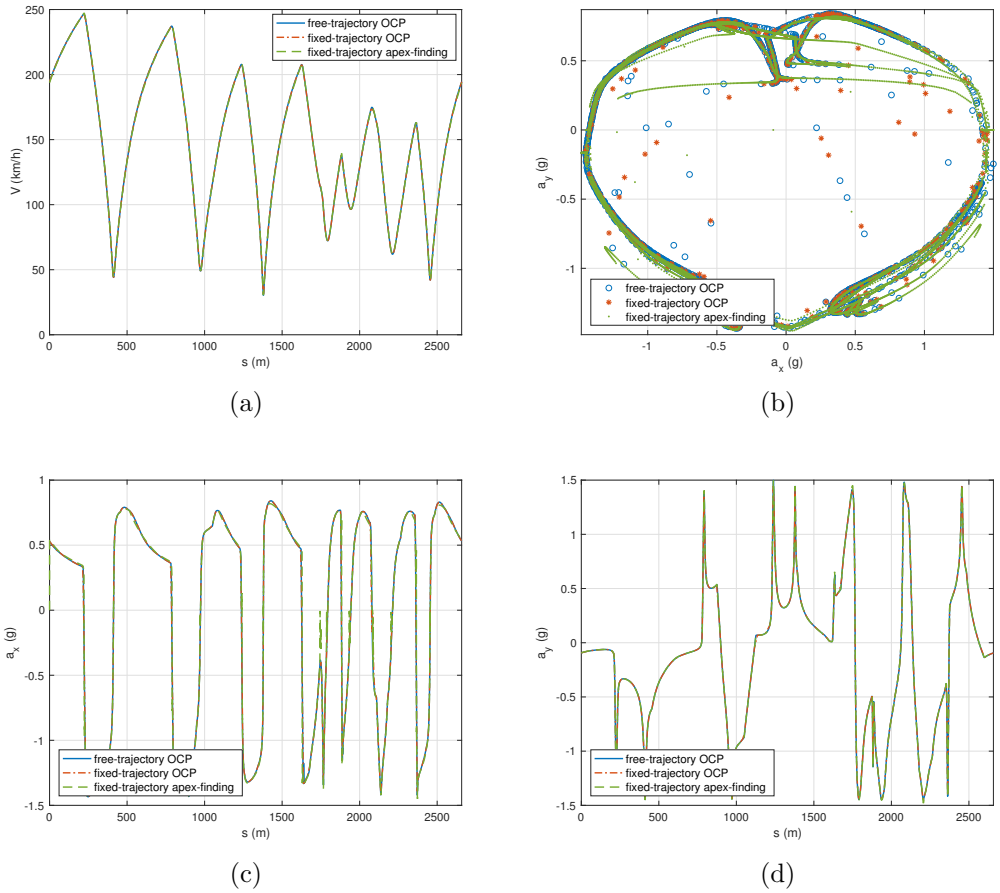


Figure 5.13: Race car: comparison between quasi-steady-state minimum-lap-time methods. a) Speed profile. b) g-g diagrams. c) Longitudinal acceleration. d) Lateral acceleration for free-trajectory OCP (solid), fixed-trajectory OCP (dash-dot) and fixed-trajectory apex-finding method (dashed).

Table 5.1: Car simulation. Computational effort overview.

	Apex-finding	**Fixed-OCP**	**Free-OCP**
Lap time [s]	75.450	75.456	75.451
Penalties [s]	-	0.011	0.002
CPU time [s]	249	99	345
Initial mesh-points	26596	41	41
Initial equations	26596	123	328
Final mesh-points	26596	446	1478
Final equations	26596	1338	11824
Equations solved	26596	6270	83952
Mesh iterations	0	7	10
Total solver iterations	102373	7558	2884

Car simulation

The speed-profile, g-g map and acceleration profiles obtained with the presented quasi-steady-state programs (based on the predetermined g-g-V surface) are reported in Fig. 5.13. The results are overlapped and the relative error is lower than 1%. Some points in the g-g diagrams of the OCP simulations (Fig. 5.13b) are not representative of the maximum performance, but lay inside the g-g envelope. In fact, the controls (i.e. the acceleration derivatives) are limited and penalised, giving a non-instantaneous dynamic (or quasi-static) behaviour.

A comparison of the lap-time and the problem size in the quasi-steady-state minimum-time simulation is shown in Tab. 5.1 for the race car model.

As expected, when solving the fixed-trajectory OCP, using the race line obtained with the free-trajectory OCP, the lap time is nearly identical (difference of 0.01%). When solving the lap-time problem with the fixed-trajectory apex-finding approach, using the race line obtained with the free-trajectory OCP, the lap time is again nearly identical (difference of 0.01%).

Overall, with the current implementation, the computation time of the free-trajectory OCP is about three times the computation time of the fixed-trajectory OCP, while the computation time of the fixed-trajectory apex-finding method is about one-and-a-half the computation time of the fixed-trajectory OCP.

Considering the problem size, the free-trajectory OCP needs to solve 328 equations (state equations plus constraints) with the initial mesh grid, which consists of 41 mesh points (in the range 25-100 m). In the final mesh grid, the number of mesh points is 1478 and the number of equations raises to 11824. The total number of equations solved after the 10 mesh-refinement steps is 83952, while the total number of iterations performed by IPOPT is 2884. The fixed-trajectory OCP needs to solve 123 equations with the initial mesh grid, which consists of 41 mesh points. In the final mesh grid the number of mesh points is 446 and the number of equations to solve is 1338. The total number of equations solved after the 7 mesh-refinement steps is 6270, while the total number of iterations performed by IPOPT is 7558. Finally, the apex-finding method has a constant number of 26596 mesh points (since there is no mesh refinement) and as many equations. The total number of iterations performed by its root-finding solver is 102373.

Motorcycle simulation

The speed-profile, g-g map and acceleration profiles obtained with the presented quasi-steady-state programs (based on the predetermined g-g-V surface) are reported in Fig. 5.13 for the race motorcycle model. The results are overlapped and the relative error is lower than 1%. As for the race car, some points in the g-g diagrams of the OCP simulations (Fig. 5.13b) are not representative of the maximum performance, but lay inside the g-g envelope. In fact, the controls (i.e. the acceleration derivatives) are limited and penalised, giving a non-instantaneous dynamic (or quasi-static) behaviour.

A comparison of the lap-time and the problem size in the quasi-steady-state minimum-time simulation is shown in Tab. 5.2 for the race motorcycle model.

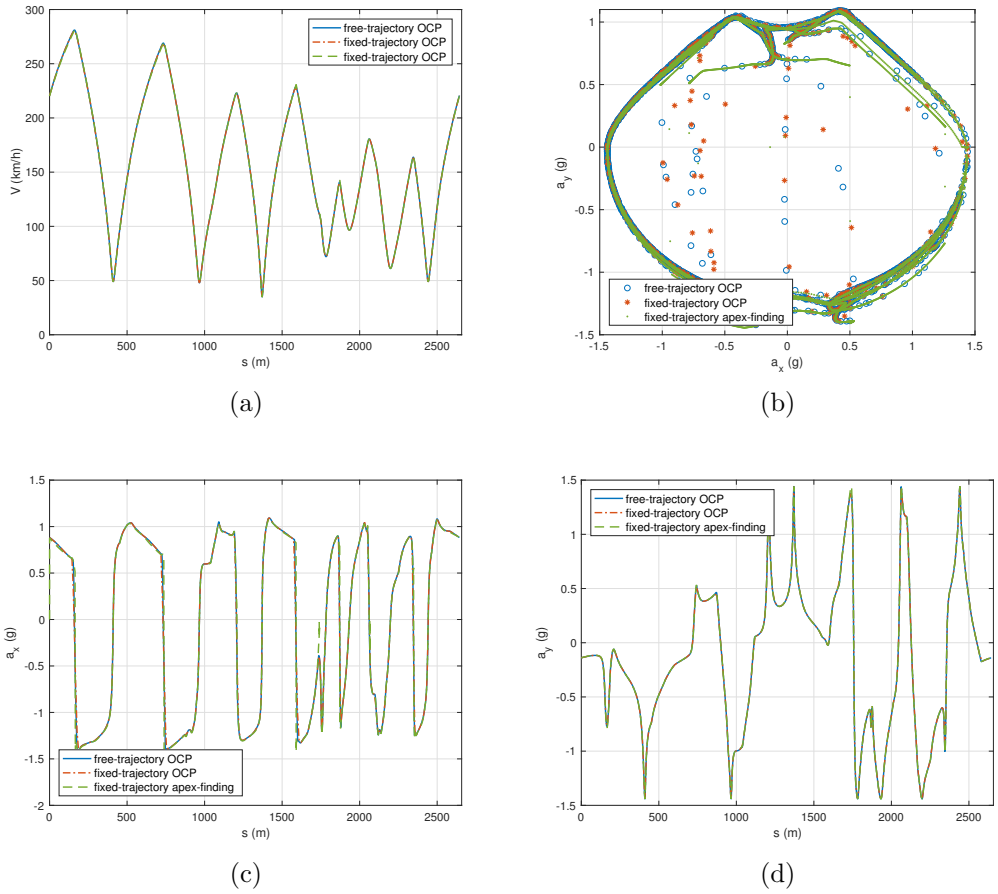


Figure 5.14: Race motorcycle: comparison between quasi-steady-state minimum-lap-time methods. a) Speed profile. b) g-g diagrams. c) Longitudinal acceleration. d) Lateral acceleration for free-trajectory OCP (solid), fixed-trajectory OCP (dash-dot) and fixed-trajectory apex-finding method (dashed).

Table 5.2: Motorcycle simulation. Computational effort overview.

	Apex-finding	**Fixed-OCP**	**Free-OCP**
Lap time [s]	72.530	72.543	72.519
Penalties [s]	-	0.025	0.012
CPU time [s]	296	84	75
Initial mesh-points	26596	41	41
Initial equations	26596	123	328
Final mesh-points	26596	446	1463
Final equations	26596	1386	11704
Equations solved	26596	7731	82696
Mesh iterations	0	8	10
Total solver iterations	100943	7434	512

Similarly to the race car scenario, even in the case of the motorcycle the lap time obtained with the fixed-trajectory methods (either OCP or apex-finding) almost coincides with the lap time of the free-trajectory OCP (difference around 0.02%). In fact, the predefined trajectory is the optimal race line obtained from the free-trajectory OCP.

With the current implementation, the computation time of the free-trajectory OCP is similar to the computation time of the fixed-trajectory OCP, while the computation time of the fixed-trajectory apex-finding method is about three-and-a-half the computation time of the fixed-trajectory OCP.

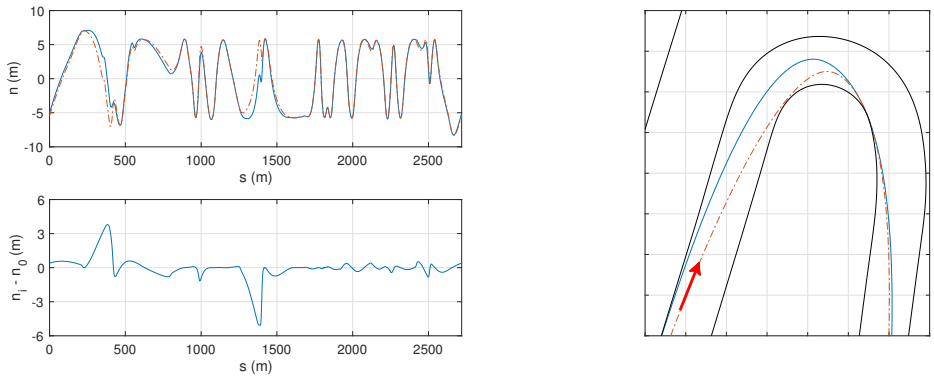
Considering the problem size, the free-trajectory OCP needs to solve 328 equations (state equations plus constraints) with the initial mesh grid, which consists of 41 mesh points – this is identical to the car problem. In the final mesh grid, the number of mesh points is 1463 and the number of equations raises to 11704. The total number of equations solved after the 10 mesh-refinement steps is 82696, while the total number of iterations performed by IPOPT is 512 – the number of equations is close to that of the car problem, the number of mesh steps is the same, while the number of iterations is smaller. The fixed-trajectory OCP needs to solve 123 equations with the initial mesh grid, which consists of 41 mesh points – again this is identical to the car problem. In the final mesh grid, the number of mesh points is 446 and the number of equations to solve is 1386. The total number of equations solved after the 8 mesh-refinement steps is 7731, while the total number of iterations performed by IPOPT is 7434 – the numbers are close to those of the car problem. Finally, the size of the apex-finding method is identical to the case of the race car, i.e. 26596 mesh points, and the number solver iterations is close (100943).

5.2.6 Sensitivity analysis

The main feature of the free-trajectory OCP method consists of the possibility of achieving the optimal race-line, using a quasi-steady-state vehicle model (i.e. a full dynamic model is not required). Trying to foresee the race-line that minimises the lap-time is not a trivial problem when the setup is changed, since the vehicle parameters may significantly affect the optimal trajectory. In the following sections, the effect of road friction, brake balance and roll-stiffness balance on the trajectory of the race car is presented. A sensitivity analysis on the lap-time and the race-line is carried out for each parameter. Moreover, a comparison of the quasi-steady-state minimum-time methods is included for the optimisation of the roll balance of the vehicle.

Effect of friction on the race-line

The coefficients $\lambda_{\mu,x}$ and $\lambda_{\mu,y}$ are halved, in order to simulate a low-friction condition, e.g. very wet vs. dry conditions. The obtained g-g-V diagrams are reported in Ch. 4, and are employed for limiting the performance of the OCP vehicle model (Sec. 5.2.4). In the baseline friction scenario, the lap time is 75.451 s, while in the low friction scenario the lap time increases to 102.727 s. The low-friction-OCP



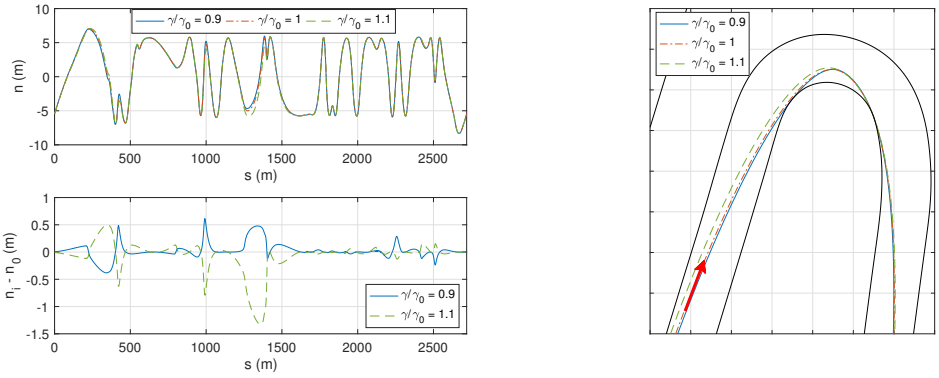
(a) (b)

Figure 5.15: a) Lateral position n at different road-tyre friction (top) and lateral position difference with respect to the baseline configuration (bottom). b) Magnified view of turn 3, with different race lines (right). Baseline (dot-dash), low friction (solid).

results are compared to the baseline simulation (i.e. baseline friction) in Fig. 5.15. The race line differs significantly in the two scenarios, in particular in the tight corners. When the vehicle enters turn 1 (300-500 m), the optimal lateral position n in the low-friction scenario is 4 m closer to the outer border with respect to the baseline friction scenario. The same behaviour is observed in turn 3 (1300-1500 m).

Effect of brake ratio on the race line

A small change in the brake ratio γ, defined in Ch. 3, can give a quite different behaviour of the vehicle during braking. Three γ configurations are chosen for the present investigation: the baseline configuration (A), a configuration with γ increased by 10% (B), and a configuration with γ reduced by 10% (C). The g-g-V maps obtained for each configuration are reported in Ch. 4, and are employed for limiting the performance of the OCP vehicle model (Sec. 5.2.4). The lateral position n of the three configurations is compared in Fig. 5.16. In general, the A and C configurations have an opposite behaviour during corners. The baseline brake ratio value γ_0 is reported in Ch. 3. The lap time is 75.451 s for vehicle A, 75.040 s for vehicle B and 75.844 s for vehicle C. At turn 1 (300-500 m), vehicle B approaches the corner closer to the outer border with respect to vehicle A, adopts a tighter trajectory during cornering, and moves towards the outer border again at the exit of the corner. The difference between the lateral position n of vehicle B and A, in this case, is 0.5 m. Vehicle C, instead, runs tighter to the inner border before the turn, goes larger during the turn, and exits the turn closer to the inner border. At turn 2 (900-1100 m) the behaviour is similar to turn 1. At turn 3 (1300-1500 m), vehicle B employs a wider race line both while entering and exiting the corner, while vehicle C employs the opposite strategy, with a trajectory closer to the inner side of the corner. Vehicle B has a maximum lateral position difference of 1.3 m, with respect to the baseline trajectory (A), while leaving the turn.



(a) (b)

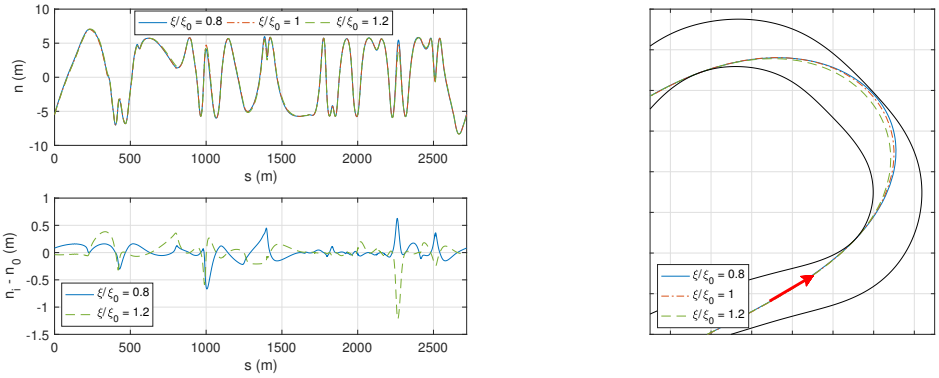
Figure 5.16: a) Lateral position of n at different brake ratios γ (top) and lateral position difference with respect to the baseline configuration (bottom). b) Magnified view of turn 3, with different race lines (right). Baseline (dot-dash), brake ratio -10% (solid), brake ratio +10% (dashed).

Effect of race-line on the roll stiffness optimum

The roll stiffness ratio ξ (see Ch. 3) affects the lateral load transfer on each axle, while the total lateral load transfer remains the same. The effect of such parameter on the race line is shown in Fig. 5.17 for three configurations: the baseline vehicle (A), the vehicle with ξ increased by 20% (B), and the vehicle with ξ reduced by 20% (C). The g-g-V diagrams obtained for each configuration are reported in Ch. 4, and are employed for limiting the performance of the OCP vehicle model (Sec. 5.2.4). The baseline roll-stiffness ratio ξ_0 is reported in Ch. 3. At turn 1 (300-500 m) and 2 (900-1100 m), vehicle B reaches the apex travelling on a race line closer to the inner kerb, and exits the corner with a larger trajectory with respect to vehicle A (baseline). At turn 3 (1300-1500 m), vehicle B drives closer to the outer border during the corner and then closer to the inner border in the exit phase. At turn 8 (2100-2300 m) the maximum difference of lateral position is observed (1.2 m): again vehicle B drives the corner closer to the inner kerb. In each case, vehicle C adopts an opposite strategy with respect to vehicle B.

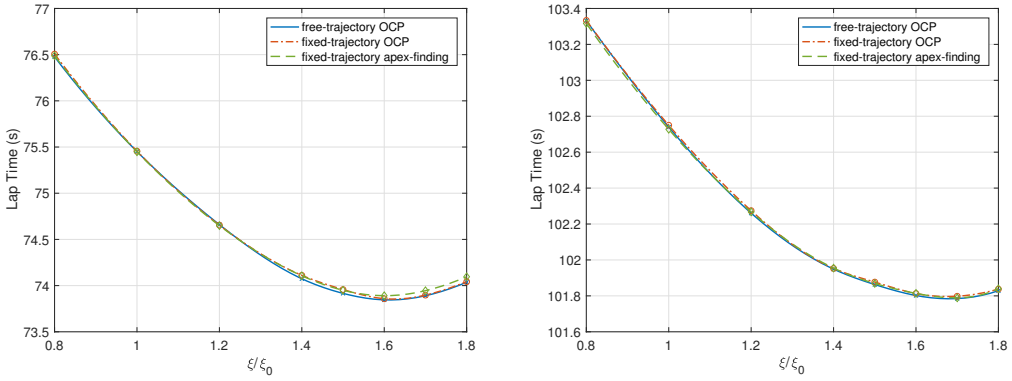
A parameter optimisation is run in order to find the roll ratio that gives the minimum-lap-time. Both the baseline adherence and the low adherence cases are considered. The optimisation is carried out by employing the free-trajectory OCP, the fixed-trajectory OCP, and the fixed-trajectory apex-finding method. The fixed-trajectory methods are run using the race line obtained with the corresponding free-trajectory OCP. The three methods give the same optimum, both in the baseline-friction ($\xi/\xi_0 = 1.6$) and low-friction ($\xi/\xi_0 = 1.7$) scenario (see Fig. 5.18). Indeed, as explained in Sec. 5.2.5, the three methods give roughly the same lap time, once the same race line is given.

A deeper investigation shows that increasing the roll-stiffness ratio is detrimental for the maximum achievable lateral acceleration, since the load transfer on the front axle is already large in the baseline configuration. However, while



(a) (b)

Figure 5.17: a) Lateral position n at different roll stiffness ratios ξ (top) and lateral position difference with respect to the baseline configuration (bottom). b) Magnified view of turn 8, with different race lines. Baseline (dot-dash), roll stiffness ratio -20% (solid), roll stiffness ratio +20% (dashed).



(a) (b)

Figure 5.18: Lap time versus the normalised roll-ratio parameter ξ/ξ_0 for baseline friction (a) and low friction (b), using the free-trajectory OCP method (solid), the fixed-trajectory OCP method (dash-dot), and the fixed-trajectory apex-finding method (dashed).

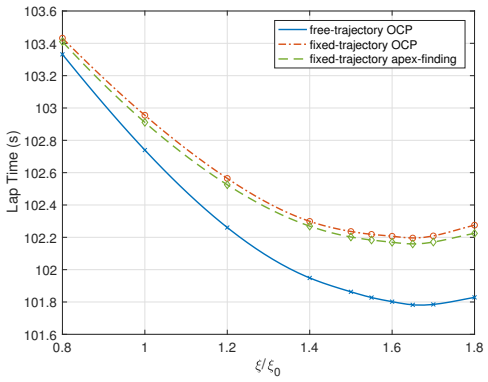


Figure 5.19: Lap time versus the normalised roll-ratio parameter ξ/ξ_0, using the low-friction free-trajectory OCP method (solid), the fixed-trajectory OCP method (dash-dot) computed on the baseline friction race line (with low-friction g-g), and the fixed-trajectory apex-finding method (dashed) computed on the baseline friction race line (with low-friction g-g).

simultaneously braking (or accelerating) and cornering, increasing the roll stiffness ratio is beneficial, and higher deceleration (or acceleration) can be achieved. Indeed, in such condition the rear tyres saturates, and thus minimising the load transfer on the rear axle is beneficial for improving the performance, i.e. the g-g area. The optimal value of the roll stiffness ratio is the best trade-off between these two effects (see Ch. 4).

In most of the practical cases, the race line is available for a certain configuration only. The fixed-trajectory methods allow us to investigate the effect of a given parameter, together with the g-g diagrams calculated with different values of such parameter, while still employing the same predetermined trajectory. As an example of application, the race line resulting from the baseline vehicle optimal lap (with the baseline friction) is selected. The fixed-trajectory OCP and the apex-finding method (Fig. 5.19) are employed on such trajectory, together with the g-g surfaces obtained for low friction conditions, at different values of the roll stiffness ratio ξ. The lap time is then compared against the lap time obtained with the free-trajectory OCP optimisation for low adherence conditions (Fig. 5.19). The lap times obtained with the fixed-trajectory methods are higher, since the race line used (which is optimal for baseline-friction condition) is not the optimal one for the g-g surfaces selected (which relate to the low-friction condition). However, when it comes to the parameter optimisation, the fixed-trajectory methods give the same optimal parameter value as the free-trajectory methods, although they are run on a suboptimal trajectory. This need not be always the case.

5.3 Dynamic approach

In the minimum-lap-time framework, the optimal-control methods are usually employed for computing the optimal trajectory of a transient vehicle model. In this case, the quasi-steady-state assumptions are discarded and the non-steady-state

effects need be taken into account by a more complex state-space representation. Differently to the presented free-trajectory quasi-steady-state OCP (Sec. 5.2.4), no preprocessing have to be dedicated for the computation of the g-g-V diagrams; in fact, the entire complexity of the vehicle model is included in the OCP formulation, giving a larger set of state variables and controls. Despite that, the same numerical strategies can be adopted for the solution of the OCP (see Ch. 2), i.e. both direct and indirect methods can be employed.

5.3.1 Free-trajectory OCP

This approach combines the advantage of a dynamic vehicle model together with the optimisation of the race-line. The state-space formulation includes the dynamic equations of the vehicle and its position on the road. As for the free-trajectory quasi-steady-state approach (Sec. 5.2.4) only the information on the centre-line and the road-borders are given as an input for the simulation.

In the following sections, the OCP state-equations will be briefly discussed for a race-car model, together with the controls and the constraints needed for building the simulation. Moreover, the simulation results will be reported using the same parameters of the quasi-steady-state analyses.

Optimal-control program

The same RWD double-track model used for computing the g-g-V diagrams (see Ch. 4) is employed for building the state-space equations of the OCP. In this case the steady-state assumptions are neglected, while the curvilinear-coordinates need to be defined considering the sideslip angle of the car (see Fig. 5.20). The dynamics equations of the vehicle are:

$$
\begin{aligned}
ma_x &= (F_{xrl} + F_{xrr}) - (F_{yfl} + F_{yfr})\,\delta - F_D, && (5.53)\\
ma_y &= F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr}, && (5.54)\\
0 &= mg + F_{Lf} + F_{Lr} - N_{fl} - N_{fr} - N_{rl} - N_{rr}, && (5.55)\\
ma_y h &= \frac{T}{2}(N_{fl} - N_{fr} + N_{rl} - N_{rr}), && (5.56)\\
a_x\, h &= aF_{Lf} - bF_{Lr} - a(N_{fl} + N_{fr}) + b(N_{rl} + N_{rr}), && (5.57)\\
I_z\dot{\Omega} &= \frac{T}{2}\,(F_{yfl} - F_{yfr})\,\delta + \frac{T}{2}\,(-F_{xrl} + F_{xrr}) - a\,(F_{yfl} + F_{yfr}) + \\
&\quad + b\,(F_{yrl} + F_{yrr})\,, && (5.58)
\end{aligned}
$$

where (5.53), (5.54) and (5.55) represent the force balance along the x, y and z axes of the vehicle reference frame, while Eq. (5.56), (5.57) and (5.58) represent the moment balance with reference to the x, y and z axes. Since no steady-state assumption is considered, the longitudinal and lateral accelerations are $a_x = \dot{u} - \Omega v$ and $a_y = \dot{v} + \Omega u$, where u and v are the longitudinal and lateral velocities of the centre of mass. The tyre longitudinal and lateral forces F_{xij} and F_{yij}, where $i = f, r$ (front, rear) and $j = l, r$ (left, right) are computed from the normal loads N_{ij} and the longitudinal and lateral slips κ_{ij} and λ_{ij} by employing the Pacejka

Magic Formula. In Ch. 3 the quantities and the parameters used in the model are described in more detail.

An equality constraint is included to account for the lateral load transfer

$$ma_y \frac{h}{T}\xi = \frac{N_{fl} - N_{fr}}{2}, \tag{5.59}$$

where ξ is the roll-stiffness ratio of the vehicle, i.e. the ratio between the front roll stiffness and the total (front plus rear) roll stiffness. The open-differential hypothesis produces the equality constraints

$$\begin{aligned} F_{xrl} &= F_{xrr}, \\ F_{xfl} &= F_{xfr}, \end{aligned} \tag{5.60}$$

while the lateral slip angles are limited by four inequality constraints:

$$\lambda_{min} \leq \lambda_{ij} \leq \lambda_{max}, \tag{5.61}$$

where $i = f, r$, $j = l, r$. The power limit is included as an inequality constraint

$$(F_{xrl} + F_{xrr} + F_{xfl} + F_{xfr})V \leq P_{max}, \tag{5.62}$$

where P_{max} is the maximum power displaced by the engine. It is assumed that there is no limit on the maximum braking force, i.e. the brake system can always provide the necessary braking power. A further equality constraint is included for the fixed brake-ratio between front and rear axle:

$$\gamma = \frac{F_{xfl} + F_{xfr}}{F_{xrl} + F_{xrr}}, \tag{5.63}$$

$$F_{xfl} + F_{xfr} + \gamma[\min(F_{xrl} + F_{xrr}, 0)] = 0, \tag{5.64}$$

where γ is the ratio between the front and rear longitudinal forces. Eq. 5.64 activates the fixed brake-ratio constraint only if the sum of the rear tyre forces is negative and is obtained through regularised functions.

The position of the vehicle on the road is described by the absolute motion of the road-centre-line frame, given by

$$\dot{s} = \frac{u\cos\chi - v\sin\chi}{1 - n\kappa}, \tag{5.65}$$

$$\dot{n} = u\sin\chi + v\cos\chi, \tag{5.66}$$

$$\dot{\chi} = \Omega - \kappa\frac{u\cos\chi - v\sin\chi}{1 - n\kappa}, \tag{5.67}$$

where s is the curvilinear coordinate of the road centre-line, n is the lateral position of the vehicle, κ is the road-centre-line curvature and χ is the angle between the tangent to the centre-line and the vehicle absolute speed (see Fig. 5.20). The vehicle is constrained to move within the road borders

$$-r_{wl} \leq n \leq r_{wr}, \tag{5.68}$$

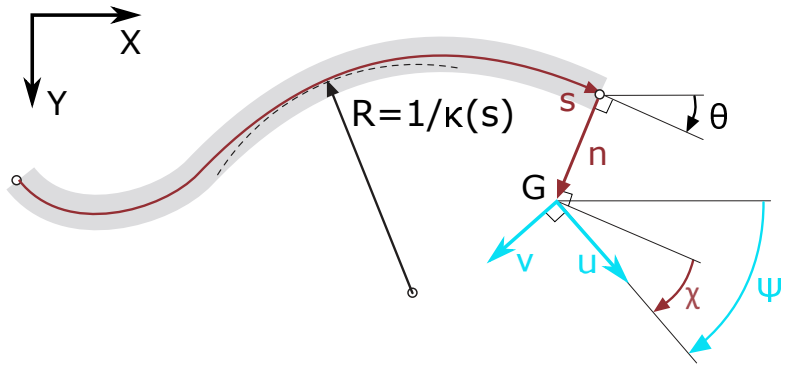


Figure 5.20: Curvilinear coordinates for the dynamic model.

where r_{wl} and r_{wr} represent the left and right road limits.

The state-space equations of the dynamic system are obtained from (5.53), (5.54), (5.58), (5.66), (5.67), while (5.55), (5.56) and (5.57) represent the problem constraints, together with (5.59)-(5.62) and (5.64). Consequently, the state and control vectors are:

$$\mathbf{x} = [u, v, \Omega, \delta, \kappa_{fl}, \kappa_{fr}, \kappa_{rl}, \kappa_{rr}, n, \chi]^T, \tag{5.69}$$

$$\mathbf{u} = [\dot{\delta}, N_{fl}, N_{fr}, N_{rl}, N_{rr}, \dot{\kappa}_{fl}, \dot{\kappa}_{fr}, \dot{\kappa}_{rl}, \dot{\kappa}_{rr}]^T, \tag{5.70}$$

where δ is the steering angle of the left wheel, N_{ij} and κ_{ij} are the loads and slips of the four tyres. Additional constraints are included to account for the limited steering angle and steering-angle rate

$$|\delta| \leq \delta_{\max}, \qquad |\dot{\delta}| \leq \dot{\delta}_{\max}, \tag{5.71}$$

and for the limited longitudinal slip rate

$$|\dot{\kappa}| \leq \dot{\kappa}_{\max}. \tag{5.72}$$

The model can be conveniently rewritten in the space domain

$$\mathbf{x}' = \frac{d\mathbf{x}}{ds} = \frac{\dot{\mathbf{x}}}{\dot{s}}, \tag{5.73}$$

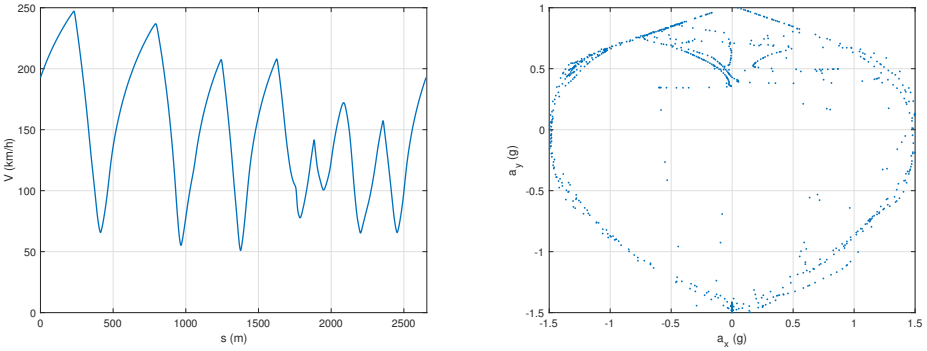
where $\dot{s}$ is computed from (5.65). The target $\mathcal{J}$ of the OCP is the manoeuvre time, which can be computed from the speed along the centre-line as follows

$$\mathcal{J} = \int (1 + \mathbf{p}(\mathbf{x}, \mathbf{u}))dt = \int \frac{1}{\dot{s}}(1 + \mathbf{p}(\mathbf{x}, \mathbf{u}))\, ds. \tag{5.74}$$

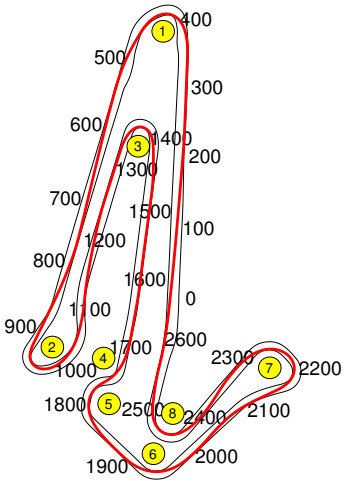
The penalty term $\mathbf{p}(\mathbf{x}, \mathbf{u})$ is given by

$$\mathbf{p}(\mathbf{x}, \mathbf{u}) = w_u \sum_{i=1}^{N_u} u_i^2 + w_\kappa \sum_{\substack{i=f,r \\ j=l,r}} \kappa_{ij}^2 + w_\lambda \sum_{\substack{i=f,r \\ j=l,r}} \lambda_{ij}^2, \tag{5.75}$$

where N_u is the number of controls, w_u ($\sim 10^{-2}$) is the penalty weight of the controls, w_κ ($\sim 10^{-4}$) is the penalty weight of the longitudinal slips and w_λ ($\sim 10^{-4}$) is the penalty weight of the sideslip angles.



(a) (b)



(c)

Figure 5.21: Free-trajectory optimal-control simulation results for the dynamic race-car model. Speed profile (a), g-g diagram (b) and optimal trajectory (c).

Simulation

The free-trajectory dynamic OCP is employed for simulating the minimum-lap-time of the GT3 vehicle (see Ch. 3) on the Adria International Raceway. GPOPS-II is used for the numerical solution of the OCP by direct collocation, together with the optimiser IPOPT and ADiGator for computing automatic derivatives. An unevenly-spaced mesh in the range 25-100 m is provided for the first iteration, while the solution is achieved e employing an unevenly-spaced mesh in the rage 0.1-10 m. The resulting speed profile, g-g map and optimal trajectory are shown in Fig. 5.21. It should be noticed that both the g-g diagram and the trajectory are obtained from the OCP solution and no assumptions on the maximum g-g performance are considered for constraining the OCP. In fact, the resulting acceleration boundary is generated considering the dynamic behaviour of the vehicle. The top speed is 247 km/h in the pit straight, while the minimum speed is 52 km/h at turn 3. The resulting acceleration boundary is limited by a 1.51 g maximum lateral acceleration, a -1.59 g maximum braking deceleration and a 1.01 g maximum acceleration. The lap-time is 73.854 s, with a 0.031 s penalty. The computation time is 850 s.

5.3.2 Example of application

The free-trajectory-dynamic-OCP approach is now applied to the analysis of the effect of the steering geometry on a Formula SAE race car. This section is mainly based on the results obtained in [6], with the aim of highlighting the peculiar aspects of this minimum-lap-time technique. The FSAE vehicle model is discussed first together with the OCP formulation used. Then, the model is validated against experimental data acquired during the Italian FSAE competition for three different test paths. Finally, three steering-geometry configurations are compared computing the minimum-lap-time on the testing paths and the Adria International Raceway.

Vehicle model and OCP

The vehicle model employed is based on the RWD double-track race car discussed in Sec. 5.3.1. In this case, different steering angles can be employed at the left and right front wheels, while their relative motion is constrained to follow a given steering-geometry rule. Moreover, the toe and camber angles of the front and rear wheels are considered, together with their effect on the tyre forces. More details on the modelling features and the model parameters are discussed in Ch. 3.

The OCP state and control vectors are:

$$\mathbf{x} = [u, v, \Omega, \delta_l, \delta_r, \kappa_{fl}, \kappa_{fr}, \kappa_{rl}, \kappa_{rr}, n, \chi]^T, \tag{5.76}$$

$$\mathbf{u} = [\dot{\delta}_l, \dot{\delta}_r, N_{fl}, N_{fr}, N_{rl}, N_{rr}, \dot{\kappa}_{fl}, \dot{\kappa}_{fr}, \dot{\kappa}_{rl}, \dot{\kappa}_{rr}]^T, \tag{5.77}$$

where δ_l is the steering angle of the left wheel, δ_r is the steering angle of the right wheel, N_{ij} and κ_{ij} are the loads and slips of the four tyres. An additional constraint is included to account for the fixed steering geometry

$$\delta_l = \delta_l(\delta_r), \tag{5.78}$$

Table 5.3: Wheel camber and toe for validation tests.

Symbol	Description	Value
Acceleration		
τ_f	toe angle at the front wheels (toe-in)	-0.5 deg
τ_r	toe angle at the rear wheels (toe-in)	-0.5 deg
ϕ_f	camber angle at the front wheels (camber-in)	-3.0 deg
ϕ_r	camber angle at the rear wheels (camber-out)	0.5 deg
Skidpad		
τ_f	toe angle at the front wheels (toe-out)	2.0 deg
τ_r	toe angle at the rear wheels (toe-in)	-0.5 deg
ϕ_f	camber angle at the front wheels (camber-in)	-3.0 deg
ϕ_r	camber angle at the rear wheels (camber-in)	-1.5 deg
Slalom		
τ_f	toe angle at the front wheels (toe-out)	2.8 deg
τ_r	toe angle at the rear wheels (toe-in)	-0.5 deg
ϕ_f	camber angle at the front wheels (camber-in)	-1.5 deg
ϕ_r	camber angle at the rear wheels	0.0 deg

i.e. the left steer angle is a function of the right steering angle, e.g. according to the Ackermann steering, the parallel steering, etc. The numerical solution of the OCP is obtained through a direct collocation approach, using GPOPS-II. ADiGator is used to speed-up the computation.

Validation

The numerical simulations have been compared against the experimental data in three manoeuvres: acceleration, skidpad and slalom. The first two tests have been performed within the 2018 Italian FSAE competition in Varano de' Melegari (PR), while the slalom manoeuvre is a benchmark for the autocross and endurance tests. In Tab. 5.3 the wheel camber and toe angles employed for each test are reported. The vehicle is provided with a Plex VMU-900 IMU for measuring chassis accelerations and angular velocities, two rotary potentiometers Avio Race Hall AR 006-10 for measuring the steering angle and the throttle pedal position, four linear potentiometers DIA 9,5-75 mounted on each spring-damper assembly for measuring the suspension travel and rate, and four wheel-speed sensors (Texsense M10). Since no GPS sensor is available, the use of a free-trajectory simulation is dramatically important.

The acceleration test is performed on a 75 m straight road, where the target is to reach the finish line in the minimum time. The simulation starts at a speed of 10 km/h, and reaches the speed of 100 km/h in 64 m. In the experimental data, the same speed is reached in 66 m (Fig. 5.22). The speed profiles are almost identical, although the gear change (see acceleration drops at 18 m and 37 m) is not included in the numerical model.

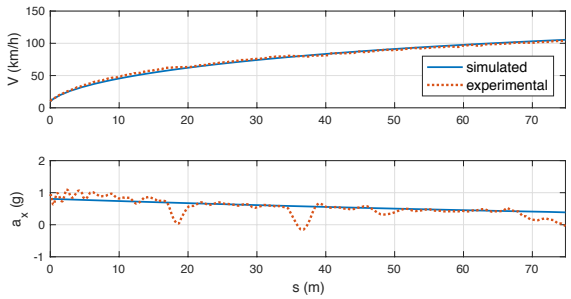


Figure 5.22: Acceleration: experimental and simulated speed (top) and longitudinal acceleration (bottom).

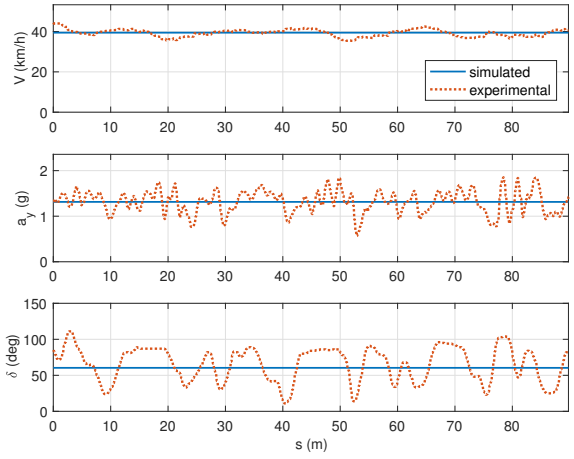
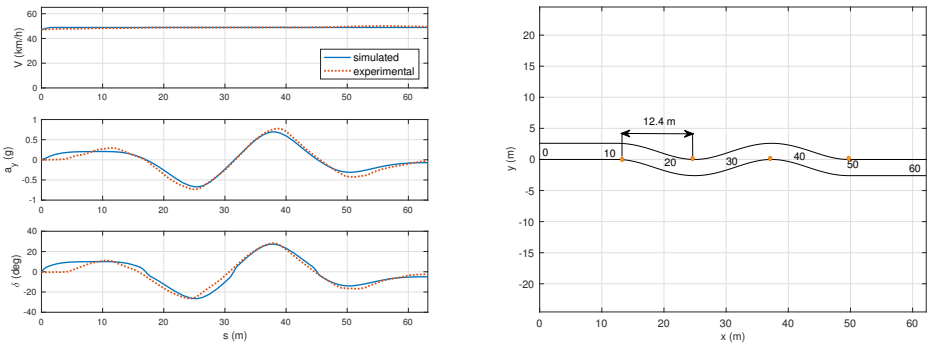


Figure 5.23: Skidpad: experimental and simulated speed (top), lateral acceleration (centre), steering wheel angle (bottom).



(a) (b)

Figure 5.24: Slalom. a) Experimental and simulated speed (top), lateral acceleration (centre), steering wheel angle (bottom). b) Slalom course.

The skidpad test consists in performing steady turning manoeuvres of given radius, at the maximum speed. The actual track geometry has an eight-shaped pattern, with an average curvature radius of 9.1 m and a width of 3.0 m. The driver has to perform two clockwise turns on the first circle of the eight and two counter-clockwise turns on the second circle. Since the vehicle numerical model is symmetric with respect to its vertical plane, only the clockwise turns can be considered – results on the counter-clockwise turns are clearly identical. In the OCP simulation the car enters the skidpad course and keeps turning for three turns. Steady-state conditions are reached after one turn. In steady state conditions the numerical driver has no oscillations on the steer and throttle inputs, differently from the real driver who is always adjusting the control inputs while trying to keep the car close to its physical limit. In the road tests the mean speed is 39 km/h, with a mean lateral acceleration of 1.34 g, a mean steering wheel angle of 62 deg (mean wheel angle of 10.7 deg); see Figure 5.23. In the OCP simulation the speed is limited to 39 km/h (same as the real road test) and the vehicle turns with a lateral acceleration of 1.31 g and a steering wheel angle of 60 deg (mean wheel angle of 10.3 deg). The skidpad test is then repeated at speeds between 10 and 15 km/h and the steering angle vs. lateral acceleration profile is obtained. The slope of this profile (i.e. the understeer gradient) for lateral accelerations between 0.1 and 0.2 g is 0.943 deg/g.

The slalom course consists of four cones, at a distance of 12.4 m: the target is turning around the cones in the minimum time. This path emulates sections of the autocross and endurance FSAE tracks, and represents an important benchmark for assessing the handling of the car. The experimental manoeuvre is obtained using the speed limiter at 49 km/h: the same speed is set as upper bound for the speed in the OCP simulation. In order to perform a slalom manoeuvre within the minimum time OCP framework, a S-shaped track is built, in which the corner apexes mimic the position of the cones, see Figure 5.24b. Since the OCP aims at moving the vehicle close to the apexes for the best performance, the width of the track has no effect on the results. A cosine function is employed to build the

S-shaped track. Initial and final straights of 12.5 m are included. In the measured data, the vehicle travels at 49 km/h, while the lateral acceleration is in the range ± 0.8 g and the steering wheel angle in the range ± 28 deg (± 4.6 deg at wheels) see Figure 5.24. The simulation results are consistent with the experimental data: the lateral acceleration is in the range ± 0.7 g, the steering wheel angle is in the range ± 28 deg (± 4.6 deg at wheels).

Effect of steering

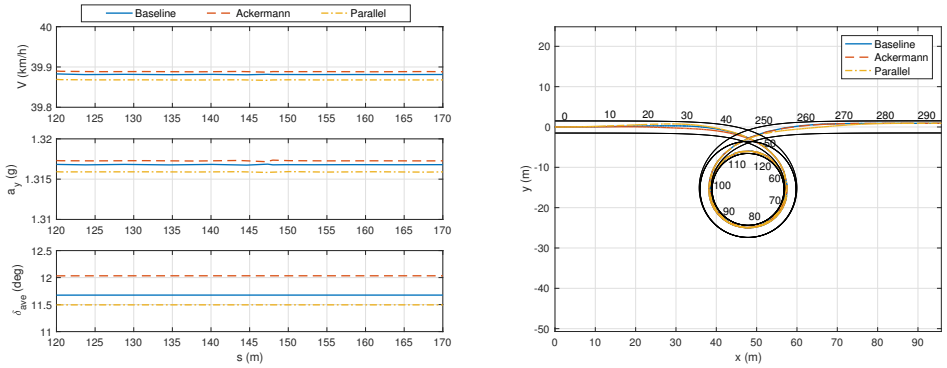
The understeer gradient is obtained from steady-state tests at different speeds. The OCP simulation is run with the same procedure described for the skidpad, during the validation analysis. The vehicle travels a constant-radius corner ($R = 50$ m), while the speed spans from 15 km/h to 100 km/h. The manoeuvre is consistent with the international standard [56].

The understeer gradient, computed as the slope of the steering angle vs. lateral acceleration between 0.1 and 0.2 g [57], is 0.021 deg/g for the baseline vehicle, 0.026 deg/g in case of Ackermann steering geometry and 0.019 deg/g for parallel steering.

The Ackermann steering vehicle is the most understeering, followed by the baseline and the parallel. Indeed, Ackermann steering is the least understeering configuration only at very low speeds, where the effects of tire slippage is negligible. As an example, when turning at a speed of 1 km/h with a 18 deg constant average steering-angle the Ackermann configuration performs the tightest circle, achieving the smallest turn radius (R = 4.50 m), while the parallel configuration reaches the worst turning performance (R = 4.75 m). The baseline geometry lays in between, travelling a larger circle than the Ackermann case and a smaller circle than the parallel case (R = 4.64 m). These results are obtained integrating the vehicle state-space equations, while selecting the speed and the steering angle inputs with a PI controller.

The understeer gradients may seem quite small. This is because toe and camber angles are neglected. In the case the toe angle is included, the values raise significantly. With the baseline vehicle and a 2 deg toe-out at the front, the understeer gradient raises to 2.761 deg/g, while with a 2 deg toe-in at the front, the understeer gradient raises to 0.458 deg/g. With 2 deg camber-in at the front wheels the understeer gradient reduces to -0.047 deg/g, while with 2 deg camber-out the understeer gradient (slightly) increases to 0.089 deg/g. These figures need be compared with 0.021 deg/g, which is computed in the case of zero toe and zero camber and shows that the effect of toe is the most significant on the steering characteristic. Finally, in the case toe and camber of the skidpad configuration are included, the understeer gradients are 1.144 deg/g for baseline, 1.157 deg/g for Ackermann and 1.138 deg/g for parallel; again, the values get larger but the ranking of configurations remains the same.

The skidpad manoeuvre is now examined to assess the maximum performance of the different steering configurations. The same course used in the validation analysis is considered. In Fig. 5.25a the speed, lateral acceleration and average steer angle at the wheels are shown during the steady-turning part of the skidpad



(a) (b)

Figure 5.25: Skidpad. a) Speed, lateral acceleration and average steering angle during steady-turning. b) Track and trajectory.

manoeuvre (for travelled distance from 120 m, i.e. after one turn of the circle, to 170 m, i.e. before exiting the circle; see Fig. 5.25b). The speed is 39.88 km/h for the baseline, 39.89 km/h for Ackermann (fastest) and 39.87 km/h for parallel (slowest). The corresponding lateral accelerations are 1.317 g, 1.317 g and 1.316 g respectively, while the average steering angles at wheels are 11.68 deg, 12.03 deg and 11.50 deg.

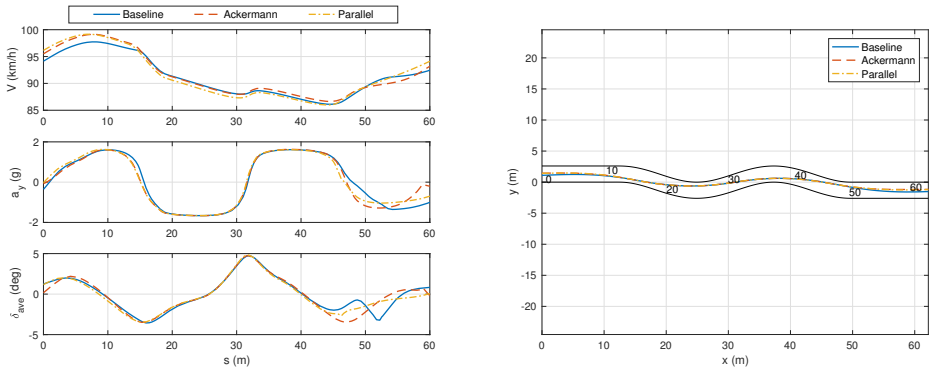
Overall, the effect of the steering geometry is very small on the maximum speed achievable (0.05 %). However, the related steering gradients ($R = 9.1$ m) are quite different: -0.117 deg/g for the baseline, -0.041 deg/g for Ackermann and -0.136 deg/g for parallel – all oversteering. It is worth noting that the understeer gradients are all positive (i.e. understeering behaviour) when computed on a turn with $R = 50$ m. However, the understeer ranking of configuration is the same: Ackermann, baseline and parallel.

In conclusion, a slightly larger steady-turning lateral acceleration (in the case of zero toe and zero camber) is achieved with the Ackermann configuration, which is also the least oversteering.

However, when toe and camber are included (as for validation), the speed raises to 44.36 km/h in the case of the baseline configuration, to 44.33 km/h in the case of Ackermann and to 44.40 km/h (fastest, by 0.16 %) in the case of parallel steering. The corresponding lateral acceleration are 1.662 g, 1.661 g and 1.667 g respectively, while the average wheel steering angles are 10.6 deg, 11.01 deg and 10.4 deg. Therefore the most performing configuration is the parallel one.

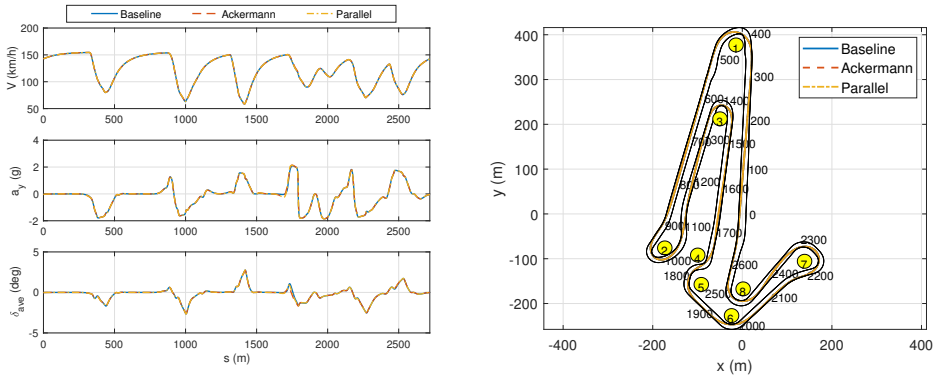
The slalom manoeuvre is performed on the same course used for the model validation, see Fig. 5.26b. The speed is constrained to assume the same values at the beginning and at the end of the slalom course (cyclic condition). No constraints are employed for limiting the speed to a constant value.

The maximum lateral acceleration is 1.658 g for the baseline, 1.663 g for Ackermann (highest) and 1.658 g for the parallel steering configuration. The maximum average wheel-steering-angle is 4.73 deg for baseline (lowest), 4.74 deg for Ackermann and 4.82 deg for parallel (highest), see Fig. 5.26a.



(a) (b)

Figure 5.26: Slalom. a) Speed, lateral acceleration and average steering angle. b) Vehicle trajectory.



(a) (b)

Figure 5.27: Racetrack. a) Speed, lateral acceleration and average steering angle. b) Racetrack and trajectory.

The manoeuvre time is 2.468 s for the baseline (slowest), 2.457 s for Ackermann (fastest) and 2.461 s for parallel. The maximum difference among the different configurations is 11 ms (0.4%). The Ackermann configuration is again the best, in the case of zero toe and zero camber.

When including toe and camber (as for validation) the manoeuvre times reduces to 2.130 s for the baseline, 2.130 s for Ackermann and 2.129 s (fastest, although almost identical to the previous configurations) for the parallel.

The different steering geometries are simulated also on a lap of the Adria International Raceway (Italy), characterised by a total length of 2702 m, and eight turns mainly paced at speeds below 80 km/h, see Fig. 5.27. The vehicle travels the track anticlockwise and reaches a maximum speed of ∼154 km/h at the pit-straight, while the minimum speed is ∼58 km/h at turn 3. The maximum lateral acceleration is ∼2.1 g at turn 4. More precisely, the maximum speed (pit-straight) is 154.55 km/h for baseline (fastest), 154.54 km/h for Ackermann, 154.51 km/h for parallel (slowest). The minimum speed (turn 3) is 58.69 km/h for baseline (fastest),

58.65 km/h for Ackermann, 58.62 km/h for parallel (slowest). The maximum lateral acceleration (turn 4) is 2.147 g for baseline (highest), 2.146 g for Ackermann, 2.124 g for parallel (lowest). The lap-time is 82.853 s for baseline (slowest), 82.827 s for Ackermann, 82.827 s for parallel. The maximum difference among the different configurations is 26 ms (0.03%). The mean difference of the lateral positions on the road (variable n in Fig. 5.20) of the Ackermann and parallel configurations with respect to the baseline are 0.040 m and 0.070 m respectively.

When the toe and camber are included (as for the slalom validation). the lap time reduces to 79.173 s (slowest) for the baseline, 78.844 s (fastest) for Ackermann and 78.891 s for parallel. The lap time difference in this case raises to 329 ms (0.4 %), which is much larger than the difference obtained in the case the sole steering configuration is changed. The mean difference of the lateral positions on the road (variable n in Fig. 5.20) of the Ackermann and parallel configurations with respect to the baseline configuration are around 0.5 m in both cases, again an order of magnitude larger than the differences observed when varying the steering configuration only.

Remarks

To conclude, the analysis showed that the effect of the steering configuration is significantly smaller than the effect of toe and camber. The toe is the most influential parameter when it comes to performance of the vehicle under investigation. Indeed the magnitude of the toe angle employed (around 2 deg) is comparable with the typical average wheel steering angle during a lap of the selected racetrack, more than one-half of the wheel steering angle employed during the slalom and about one-fourth of the steering angle employed during the skidpad.

Chapter 6

Conclusions

In this work the main minimum-lap-time strategies have been presented and reviewed, starting from the fundamental components needed for their implementation.

A detailed insight on the optimal-control problems has been carried out, since they represent a widespread and convenient approach for computing the minimum-lap-time. Both direct and indirect methods have been discussed, together with the explicit and implicit solutions. A direct implicit solver (GPOPS-II) has been chosen for developing most of the presented simulations, together with automatic differentiation (ADiGator) for evaluating derivatives.

Different modelling techniques have been discussed for cars and motorcycles. Steady-state models have been created for generating the g-g diagrams. In particular, both a race car and an essential race motorcycle model have been developed and implemented in the simulations. The car is represented by a double-track rear-wheel-drive model, that includes aerodynamic forces, Pacejka-based tyre model, fixed brake-ratio and the roll stiffness for computing the lateral-load-transfer. The presented motorcycle model retains the peculiar aspects of the motorcycle dynamics, such as the wheelie and stoppie conditions. The model includes the drag force and takes account for the limited amount of power available. Moreover, the motorcycle is assumed to perform the braking manoeuvres using an optimal braking-bias. The simplicity of this model permits to compute the g-g maps analytically, without employing optimisation methods. A dynamic race car model has also been discussed for its implementation in an optimal-control problem. The model is derived from the discussed steady-state double-track model, by including the transient effects.

Some effort has been dedicated to the g-g-speed envelopes generation, starting from steady-state vehicle models. An optimisation-based program has been developed for computing the g-g diagrams of a race car. A polar-coordinate parametrisation has been proposed, together with the automatic-differentiation, in order to speed-up the optimisation program. A set of g-g diagrams has been computed also for the motorcycle model; in this case the equations have been solved analytically, i.e. without relying on an optimisation procedure. A sensitivity analysis has also been carried out, for determining the effect of different parameters on the shape of the g-g diagrams, underlining their effectiveness in describing the steady-state

behaviour of the vehicle.

Starting from the theory and the results obtained in the previous part of the work, some of the most common minimum-time-simulation strategies have been discussed. Both quasi-steady-state (using steady-state vehicle models) and transient approaches have been employed, together with fixed (predetermined) or free (non-predetermined) trajectories.

In the quasi-steady-state framework, three methods have been presented: a fixed-trajectory apex-finding method, a fixed-trajectory optimal-control method and a free-trajectory optimal-control method. Each of these methods is based on the computation of the g-g diagrams, which are used to limit the performance of the vehicle. The quasi-steady-state apex-finding program is solved by computing step-by-step the vehicle speed during acceleration and deceleration phases between two corners. In summary, the corner apexes need be identified first: at each apex, where the trajectory curvature is maximum, the vehicle is assumed to have maximum lateral acceleration. Before the apex the vehicle is braking along the g-g envelope, while after the apex the tyres are engaged with traction forces in order to keep the vehicle along the g-g envelope. The speed profiles between two apexes are then connected at their intersection point. Performing the same procedure for all the pairs of apexes, permits to obtain the entire speed profile along the given trajectory and, consequently, the total lap-time. In the case of the optimal-control-based free-trajectory and fixed-trajectory simulations, the aim is to find the optimal driving inputs that minimise the total-manoeuvre time for completing a track lap. A simple vehicle model is used for the optimal-control problem formulation, since all the complexity is confined to the steady-state models employed for the generation of the g-g-speed envelopes. In the free-trajectory approach, the model has three degrees-of-freedom: the speed, the lateral position with respect to the road-centreline and the direction of the speed vector. The vehicle is constrained to move within the road borders, while satisfying the limits of the g-g-speed surface. This method represents a novel approach for calculating the minimum-lap-time, since it allows the computation of the optimal race line without involving a full dynamic model. In the fixed-trajectory approach, the number of degrees-of-freedom is reduced to one, since the lateral motion and the orientation of the velocity are constrained by the given race line, although the lateral and longitudinal performance is still limited by the g-g boundary. With the current implementation, the computation time of the free-trajectory approach lies between one and three times the computation time of the fixed-trajectory approaches. The advantage of optimising the trajectory has been underlined by discussing the effect of the tyre-road friction, brake ratio, and roll-stiffness ratio on the resulting race line. The effect of the fixed-trajectory assumption has been also investigated in the case of the optimisation of the roll-stiffness ratio. In particular, given a set of baseline parameters, the results of the free-trajectory optimisation have been compared to the results obtained employing the fixed-trajectory methods on a suboptimal race line; this has been repeated for different roll-stiffness ratios. In this case, the same optimal parameter has been obtained for both the free and fixed-trajectory simulations.

In addition, the presented dynamic car model has been implemented in a free-

trajectory optimal-control simulation. In this case, the entire vehicle model is included in the OCP formulation, without involving the computation of the g-g diagrams. This simulation has been employed for assessing the effect of the steering geometry on a race car. The model of a Formula SAE car has been built and validated against experimental data. The effect of the baseline FSAE geometry, the pure Ackermann geometry and the parallel steering geometry have been compared in skidpad, slalom and for a lap of the Adria International Raceway. It has been found that the effect of the Ackermann ratio on the maximum performance is small, at least when compared with the effect of toe, whose magnitude can be comparable with the wheel steering angle on a racetrack.

In conclusion, the work has addressed both the theoretical methods and the more practical applications of minimum-lap-time simulations. A significant effort has been spent in designing and implementing the presented simulation programs, together with suitable vehicle models. Also the g-g diagrams generation procedure has been developed from scratch, employing non-linear-programming and automatic differentiation for finding the optimal steady-state solution. The main contributions carried out in this work are basically three. First, a novel minimum-time method that combines steady-state models together with the race line optimisation has been presented and compared to other widespread simulation strategies. Second, an essential but comprehensive motorcycle model has been implemented for the minimum-time computation. Finally, a dynamic car model has been employed for clarifying the effect of the steering geometry on the performance of an FSAE car.

Bibliography

[1] H. Scherenberg, "Mercedes-benz racing design and cars experience," Tech. Rep. 1958-01-01, SAE Technical Paper, 1958.

[2] D. Metz and D. Williams, "Near time-optimal control of racing vehicles," *Automatica*, vol. 25, no. 6, pp. 841–857, 1989.

[3] J. Hendrikx, T. Meijlink, and R. Kriens, "Application of optimal control theory to inverse simulation of car handling," *Vehicle System Dynamics*, vol. 26, no. 6, pp. 449–461, 1996.

[4] V. Cossalter, M. Da Lio, R. Lot, and L. Fabbri, "A general method for the evaluation of vehicle manoeuvrability with special emphasis on motorcycles," *Vehicle System Dynamics*, vol. 31, no. 2, pp. 113–135, 1999.

[5] M. Veneri and M. Massaro, "A free-trajectory quasi-steady-state optimal-control method for minimum lap-time of race vehicles," *Vehicle System Dynamics*, 2019.

[6] M. Veneri and M. Massaro, "The effect of ackermann steering on the performance of race cars," *Vehicle System Dynamics*, 2019.

[7] A. E. Bryson Jr, M. N. Desai, and W. C. Hoffman, "Energy-state approximation in performance optimization of supersonicaircraft," *Journal of Aircraft*, vol. 6, no. 6, pp. 481–488, 1969.

[8] E. Davison and J. Seugnet, "A minimum time intercept problem," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 116–118, 1970.

[9] T. N. Edelbaum, "Optimum power-limited orbit transfer in strong gravity fields," *AIAA Journal*, vol. 3, no. 5, pp. 921–925, 1965.

[10] J. T. Betts, "Optimal low–thrust orbit transfers with eclipsing," *Optimal Control Applications and Methods*, vol. 36, no. 2, pp. 218–240, 2015.

[11] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*, vol. 19. Siam, 2010.

[12] A. Sciarretta, M. Back, and L. Guzzella, "Optimal control of parallel hybrid electric vehicles," *IEEE Transactions on control systems technology*, vol. 12, no. 3, pp. 352–363, 2004.

[13] N. Kim, S. Cha, and H. Peng, "Optimal control of hybrid electric vehicles based on pontryagin's minimum principle," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 5, pp. 1279–1287, 2011.

[14] L. V. Pérez and E. A. Pilotta, "Optimal power split in a hybrid electric vehicle using direct transcription of an optimal control problem," *Mathematics and Computers in Simulation*, vol. 79, no. 6, pp. 1959–1970, 2009.

[15] M. I. Masouleh and D. J. N. Limebeer, "Optimizing the aero-suspension interactions in a formula one car," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 912–927, 2016.

[16] C. Lanczos, *The variational principles of mechanics*. Courier Corporation, 2012.

[17] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.

[18] E. Bertolazzi, F. Biral, and M. Da Lio, "Symbolic-numeric efficient solution of optimal control problems for multibody systems," *Journal of computational and applied mathematics*, vol. 185, no. 2, pp. 404–421, 2006.

[19] E. Bertolazzi, F. Biral, and M. Da Lio, "Real-time motion planning for multibody systems," *Multibody System Dynamics*, vol. 17, no. 2-3, pp. 119–139, 2007.

[20] N. Dal Bianco, E. Bertolazzi, F. Biral, and M. Massaro, "Comparison of direct and indirect methods for minimum lap time optimal control problems," *Vehicle System Dynamics*, vol. 0, no. 0, pp. 1–32, 2018.

[21] D. J. N. Limebeer and M. Massaro, *Dynamics and optimal control of road vehicles*. Oxford University Press, 2018.

[22] N. Dal Bianco, *Optimal control for road vehicles: theory and applications*. PhD thesis, School of Industrial Engineering, University of Padova, 2017.

[23] M. A. Patterson and A. V. Rao, "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 (TOMS)*, vol. 41, no. 1, p. 1, 2014.

[24] C. L. Darby, W. W. Hager, and A. V. Rao, "An hp-adaptive pseudospectral method for solving optimal control problems," *Optimal Control Applications and Methods*, vol. 32, no. 4, pp. 476–502, 2011.

[25] M. A. Patterson, W. W. Hager, and A. V. Rao, "A ph mesh refinement method for optimal control," *Optimal Control Applications and Methods*, vol. 36, no. 4, pp. 398–421, 2015.

[26] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[27] H. Y. Benson, R. J. Vanderbei, and D. F. Shanno, "Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions," *Computational Optimization and Applications*, vol. 23, no. 2, pp. 257–272, 2002.

[28] M. J. Weinstein, M. A. Patterson, and A. V. Rao, "Utilizing the algorithmic differentiation package adigator for solving optimal control problems using direct collocation," in *AIAA Guidance, Navigation, and Control Conference*, p. 1085, 2015.

[29] M. J. Weinstein and A. V. Rao, "A source transformation via operator overloading method for the automatic differentiation of mathematical functions in matlab," *ACM Transactions on Mathematical Software (TOMS)*, vol. 42, no. 2, p. 11, 2016.

[30] M. Gadola, D. Vetturi, D. Cambiaghi, and L. Manzo, "A tool for lap time simulation," Tech. Rep. 1996-12-01, SAE Technical Paper, 1996.

[31] D. Brayshaw, *Use of numerical optimisation to determine on-limit handling behaviour of race cars.* PhD thesis, School of Engineering, Cranfield University, 2004.

[32] D. Brayshaw and M. Harrison, "A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 219, no. 6, pp. 725–739, 2005.

[33] D. Casanova, *On minimum time vehicle manoeuvring: The theoretical optimal lap.* PhD thesis, School of Engineering, Cranfield University, 2000.

[34] D. Casanova, R. S. Sharp, and P. Symonds, "Minimum time manoeuvring: The significance of yaw inertia," *Vehicle system dynamics*, vol. 34, no. 2, pp. 77–115, 2000.

[35] D. P. Kelly and R. S. Sharp, "Time-optimal control of the race car: influence of a thermodynamic tyre model," *Vehicle System Dynamics*, vol. 50, no. 4, pp. 641–662, 2012.

[36] T. Völkl, M. Muehlmeier, and H. Winner, "Extended steady state lap time simulation for analyzing transient vehicle behavior," *SAE International Journal of Passenger Cars-Mechanical Systems*, pp. 283–292, 2013.

[37] A. Tremlett, F. Assadian, D. Purdy, N. Vaughan, A. Moore, and M. Halley, "Quasi-steady-state linearisation of the racing vehicle acceleration envelope: a limited slip differential example," *Vehicle System Dynamics*, vol. 52, no. 11, pp. 1416–1442, 2014.

[38] G. Perantoni and D. J. N. Limebeer, "Optimal control for a formula one car with variable parameters," *Vehicle System Dynamics*, vol. 52, no. 5, pp. 653–678, 2014.

[39] D. J. N. Limebeer, G. Perantoni, and A. V. Rao, "Optimal control of formula one car energy recovery systems," *International Journal of Control*, vol. 87, no. 10, pp. 2065–2080, 2014.

[40] M. Massaro and E. Marconi, "The effect of engine spin direction on the dynamics of powered two wheelers," *Vehicle System Dynamics*, vol. 56, no. 4, pp. 604–620, 2018.

[41] F. Biral and R. Lot, "An interpretative model of g-g diagrams of racing motorcycle," in *Proceedings of the 3rd ICMEM International Conference on Mechanical Engineering and Mechanics*, 2009.

[42] J. Katz, *Race car aerodynamics*. Bentley Publishers, 2005.

[43] H. Pacejka and I. Besselink, *Tire and Vehicle Dynamics*. Elsevier Science, 2012.

[44] A. Tremlett, M. Massaro, D. Purdy, E. Velenis, F. Assadian, A. Moore, and M. Halley, "Optimal control of motorsport differentials," *Vehicle System Dynamics*, vol. 53, no. 12, pp. 1772–1794, 2015.

[45] BS-ISO-8855:2011, "Road vehicles–vehicle dynamics and road-holding ability–vocabulary," standard, International Organization for Standardization, 2011.

[46] W. F. Milliken, D. L. Milliken, *et al.*, *Race car vehicle dynamics*, vol. 400. Society of Automotive Engineers Warrendale, 1995.

[47] R. N. Jazar, *Vehicle dynamics: theory and application*. Springer, 2017.

[48] W. C. Mitchell, A. Staniforth, and I. Scott, "Analysis of ackermann steering geometry," *SAE Technical Paper*, 2006.

[49] M. Burgess, N. Fleming, M. Wootton, and S. Williams, "A tool for rapid vehicle suspension design," *SAE Technical Paper*, 2004.

[50] Lotus-Engineering, "Lotus SHARK manual," software manual, 2004.

[51] S. M. Savaresi, C. Spelta, D. Ciotti, M. Sofia, E. Rosignoli, and E. Bina, "Virtual selection of the optimal gear-set in a race car," *International Journal of Vehicle Systems Modelling and Testing*, vol. 3, no. 1-2, pp. 47–67, 2008.

[52] J. Edelmann and M. Plöchl, "Handling characteristics and stability of the steady-state powerslide motion of an automobile," *Regular and Chaotic Dynamics*, vol. 14, p. 682, Dec 2009.

[53] D. Tavernini, M. Massaro, E. Velenis, D. Katzourakis, and R. Lot, "Minimum time cornering: The effect of road surface and car transmission layout," *Vehicle System Dynamics*, vol. 51, no. 10, pp. 1533–1547, 2013.

[54] D. Tavernini, E. Velenis, R. Lot, and M. Massaro, "The optimality of the handbrake cornering technique," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 4, 2014.

[55] N. Dal Bianco, R. Lot, and M. Gadola, "Minimum time optimal control simulation of a GP2 race car," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 232, no. 9, pp. 1180–1195, 2017.

[56] BS-ISO-4138:2012, "Passenger cars —- steady-state circular driving behaviour – open-loop test methods," standard, International Organization for Standardization, 2012.

[57] D. F. Tandy, J. Colborn, J. C. Bae, C. Coleman, and R. Pascarella, "The true definition and measurement of oversteer and understeer," *SAE International Journal of Commercial Vehicles*, vol. 8, no. 2015-01-1592, pp. 160–181, 2015.