Optimal Control and Stability of Four-Wheeled Vehicles

Mehdi Imani Masouleh
New College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2017

To my mother, Afsaneh

Acknowledgements

First and foremost I would like to thank my supervisor David Limebeer for his support and invaluable insight. David has not only been a beacon of guidance in my academic life but also in my personal life. I will always see him as a great mentor and will continue to seek his advice on important decisions in future.

I would like to extend my deepest gratitude to Chris Laws and Enrico Sampo for sharing their in-depth knowledge on vehicle modelling and also providing the data required for some of the studies in this thesis. Without their support this work would not have been possible.

My time at Oxford would not have been the same without my friends at the office. It was a great pleasure to have the company of Reza, Chen, Ross, Andreas, Ming, Gareth, Dhruva and Manuel. The hours we spent discussing every issue from the future of control engineering to various social issues have added greatly to my intellectual rigour. I will very much miss the dinners, the socials and, of course, our AoE adventures.

Oxford Judo has played an important role in my life since the beginning of my DPhil. I am deeply indebted to my coach Chris Doherty for teaching me everything I know and helping me grow both as a judoka and as a man. I am grateful to all my friends at Oxford Judo especially Tom, Ralph and Emma for being a great source of positive energy. In my college, I had a pleasure of meeting many wonderful people. Amongst them, I am especially thankful to Ben Vaidya for being such a great friend and also for reading this thesis.

Finally, I would like to thank my family for their unconditional support and love. My mother, Afsaneh, and father, Reza, have been relentless in providing the fertile ground for my academic growth. My siblings, Maryam, Sara and Mobin have always been a huge source of encouragement and motivation. Thank you all for being there.

Abstract

Two vehicular optimal control problems are visited. The first relates to the minimum lap time problem, which is of interest in racing and the second the minimum fuel problem, which is of great importance in commercial road vehicles. Historically, minimum lap time problems were considered impractical due to their slow solution times compared with the quasi-steady static (QSS) simulations. However, with increasing computational power and advancement of numerical algorithms, such problems have become an invaluable tool for the racing teams. To keep the solution times reasonable, much attention still has to be paid to the problem formulation. The suspension of a Formula One car is modelled using classical mechanics and a meta-model is proposed to enable its incorporation in the optimal control problem. The interactions between the aerodynamics and the suspension are thereby studied and various related parameters are optimised.

Aerodynamics plays a crucial role in the performance of Formula One cars. The influence of a locally applied perturbation to the aerodynamic balance is investigated to determine if a compromise made in design can actually lead to lap time improvements. Various issues related to minimum lap time calculations are then discussed.

With the danger of climate change and the pressing need to reduce emissions, improvements in fuel consumption are presently needed more than ever. A methodology is developed for fuel performance optimisation of a hybrid vehicle equipped with an undersized engine, battery and a flywheel. Rather than using the widely used driving cycles, a three-dimensional

route is chosen and the optimal driving and power management strategy is found with respect to a time of arrival constraint. The benefits of a multi-storage configuration are thereby demonstrated.

Finally, the nonlinear stability of a vehicle model described by rational vector fields is investigated using region of attraction (RoA) analysis. With the aid of sum-of-squares programming techniques, Lyapunov functions are found whose level sets act as an under-approximation to the RoA. The influence of different vehicle parameters and driving conditions on the RoA is studied.

Contents

List of Figures

Notation

$\mathbb{R}^n$ denotes the n-dimensional Euclidean space, $\mathbb{R}^{n\times m}$ is the set of $n\times m$ real matrices. Lower capital letters denote scalars e.g. x, unless defined otherwise in the text. Bold lower capital letters denote column vectors e.g. $\boldsymbol{x} = [x_1, ..., x_n]^T$. The i^{th} element of a vector is identified by the unbolded vector symbol with subscript i, e.g. x_i is the i^{th} element of $\boldsymbol{x}$. A bold capital letter denotes a matrix, e.g. $\boldsymbol{X}$. Partial differential operation on vector function $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), ..., f_m(\boldsymbol{x})]^T$ results in a $m \times n$ matrix.

$$\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \cdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

The gradient of a scalar function $f(\boldsymbol{x})$ with respect to a vector $\boldsymbol{x}$ is denoted by $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ and is a $n \times 1$ column vector. The Hessian of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ is denoted by $\nabla^2_{\boldsymbol{xx}} f(\boldsymbol{x})$ and is a $n \times n$ square matrix. The subscript is often omitted if $f(.)$ is only a function of a single vector.

Physical units are given using the SI convention in plain fonts. Calligraphic characters are used for sets, e.g. $\mathcal{D}$. Unit vectors are denoted by a hat, e.g. $\hat{\boldsymbol{x}}$. Euclidean norm on the vector $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\|_2$.

Chapter 1

Introduction

Ever since the industrial revolution, engineers have constantly attempted to find ways to control physical systems with minimal human interaction. Classical control theory, based on analytical methods such as Laplace transforms and the Routh criterion, provided a framework to enable stabilisation of unstable systems. Graphical methods such as root locus and Nyquist diagrams were then used to improve certain transient behaviours.

Optimal control theory was introduced to not only obtain a controller that is merely acceptable, but to find one which performs best with respect to some desired performance measure. In space applications, it is of paramount benefit to find the trajectory that places the spacecraft in the desired orbit with minimal fuel expenditure. In the Goddard rocket problem, given a limited amount of fuel, we seek to find a thrust profile that results in the rocket achieving its maximum height. For an industrial robot, we might want to know how the actuators of the manipulator arm should be controlled such that it is taken from its current configuration to another required configuration in the least possible time. Problems of this kind can be tackled elegantly in an optimal control framework.

With the ever increasing power of digital computing, solution of many such difficult problems has now become achievable. Often, by posing the problem differently or merely viewing it from a different perspective, we can obtain solutions that are otherwise perceived impractical. For example, in proving the stability of a non-linear system, rather than attempting to solve the differential equations analytically, we can try to find a Lyapunov function whose properties guarantee the system's stability. An optimal control problem can be transformed into a finite-dimensional problem using gradient approximation, which can then be seen as a root-finding problem. Each view angle has certain mathematical properties, some of which are better suited to digital computers. Each transformation can open gates to allow the employment of powerful numerical routines developed for otherwise seemingly unrelated applications.

The purpose of this thesis is to take advantage of such advancements to analyse ground vehicles. Since the invention of cars, a continuous effort has been put to improve their performance, efficiency and safety. In this thesis, these criteria are studied separately in different chapters and will be discussed in detail.

Vehicle performance is of special concern in racing. The history of auto-racing is as old as the history of cars themselves and has been a positive force for constant innovation in the auto industry. Today, Formula One represents the pinnacle of auto-racing. It is a place where constructors pursue maximum performance by exploiting every millisecond out of the car to make theirs faster than that of their competitors. The teams have to react quickly to redesign the car in response to changing regulations. At every race weekend, the car has to be configured for a different track, whilst considering several factors, ranging from the vehicle upgrades added by engineers to changing weather conditions and information about other teams' cars and strategies. With the limited time allowed for on-track testing and expensive development costs, simulations become an indispensable tool. Realistic models need to be used to predict the car's behaviour with respect to parameter changes to shorten development efforts and significantly cut expenditures.

Minimum lap time problems entail calculating controls along with the state trajectories that result in the vehicle traversing the circuit in the least time. Optimal control is a natural framework to study this problem and allows simultaneous optimisation of vehicle parameters. The improvements in processing power of computers and the advancement of numerical methods have now made such calculations of practical benefit to racing teams.

These simulations, which are also known as dynamic lap time simulations in the racing community, replace the driver with an ideal controller. The results from these calculations represent the fastest time possible with respect to the considered physical model and not one that can necessarily be achieved by an actual driver. The main goal of these optimal control calculations is to find relative improvements rather than estimating the minimum time as accurately as possible. Finding an accurate minimum time would require a significantly complex model, which would make the optimal control calculations impractical. Furthermore, in real racing, many external factors including the presence of other vehicles, driver mistakes and weather changes would influence the lap time. In addition, a very accurate estimate of the fastest lap is not of significant benefit to the race engineers as the main concern is to find ways to add performance to the vehicle relative to its current set-up. Another use of such calculations is to inform drivers on possible strategies that they may be unaware of, thereby assisting them in improving their performance.

It is crucial to use vehicle models that are suited to the chosen numerical routine so that practical solution times remain feasible, whilst maintaining a model that reasonably matches the actual behaviour. For example, including components that involve fast dynamics such as the stiff springs of the suspension systems can significantly inhibit solution times. Although it can be tempting for a vehicle dynamicist to include a complex vehicle model, an insight into the underlying optimisation algorithms is essential for an appropriate formulation that results in successfully solving the optimal control problem. One aim of this thesis is, therefore, to model Formula 1 car suspensions and propose surrogate meta-models for them such that they can be employed in dynamic lap time simulations.

Today, there are around 1 billion road vehicles across the globe and this number is expected to double in the next 20 years [8]. With the ever increasing consumption of fossil fuels and the associated issues with climate change, air pollution and supply pressures, improvements in fuel efficiency are urgently needed. In recent years, hybrid vehicles have proven to be a viable option for significant improvements in fuel economy.

In this thesis, we will propose a hybrid electric vehicle that incorporates a flywheel as an extra source of energy storage. It will be shown that such devices have the potential to result in significant amounts of fuel saving. Flywheels become particularly valuable in aggressive driving during which abundant braking regions manifest themselves as sources of energy recovery. Furthermore, we propose an optimal control formulation that simultaneously finds the best driving and power management strategies. As opposed to using a driving cycle, our approach allows a deeper insight into the possible fuel saving strategies that can be obtained by having control over the vehicle speed. A three-dimensional path description is used to allow timely trade-offs between the kinetic and potential energies. It will be shown that with careful formulation, optimal control calculations for a somewhat complex vehicle model can remain tractable.

Minimum lap time calculations attempt to operate the racing car such that it remains close to its limit as much as possible. The dynamical models in these formulations do not take into account the effects of unpredictable external factors and driver imperfections. The racing vehicle executing these optimal strategies can thus lose yaw stability, which can result in the car spinning out of the track. More importantly, in the case of road vehicles, loss of stability can have serious safety implications. It is therefore important to study the vehicle stability for safe and reliable manoeuvre completion. Merely determining whether the car, in a certain manoeuvre, is stable does not offer a complete picture of how far the system is from instability. By finding the region of attraction around the equilibrium point of the nonlinear system, we can determine the direction and size of the external disturbance that will cause the vehicle to enter the unstable region.

The basin of attraction of an equilibrium is defined by the set of initial conditions for which the system trajectories converge to that equilibrium. An analytical solution for the ordinary differential equation (ODE) system can answer this question but such solutions often do not exist for most ODEs. Simulations can be used to gain insight into the system behaviour but they do not offer any mathematical guarantees, regardless of how finely the states are sampled. Lyapunov devised an ingenious alternative method in 1892 that can establish whether an equilibrium is stable, without the need to solve the differential equations [9]. He proposed that the equilibrium stability can be proved by finding an energy-like positive definite function, referred to as a Lyapunov function, whose time derivative is negative along the system trajectories. A level set of such a Lyapunov function behaves like a trapping region, where an incoming trajectory would remain inside it without being able to escape. Therefore, the region of attraction can be estimated by finding the largest level set for which the Lyapunov function satisfies the conditions of asymptotic stability.

Until recently, there was no algorithmic method for searching for a Lyapunov function. However, by relaxing the search to positive polynomials that admit a sum-of-squares (SOS) decomposition, this problem reduces to a convex semi-definite program [10]. In a final study, we will employ SOS programming techniques to estimate the region of attraction for the nonlinear lateral equations of motion of a vehicle.

1.1 Thesis Outline and Contributions

A summary of each chapter with an emphasis on contribution is given here. At the beginning of each chapter, a more detailed introduction will be provided with a review of relevant existing literature.

Since the majority of this thesis is based on solving vehicular optimal control problems, in Chapter 2, an overview of optimal control theory is provided to equip the unfamiliar reader with the necessary background. Constrained and unconstrained optimisation are discussed, followed by a description of existing numerical optimal control methods. It is shown how calculus of variations can be used to derive the necessary optimality conditions. The indirect methods, which aim to solve the boundary value problem resulting from these necessary conditions, are discussed. An overview of the direct methods, which do not require the derivation of optimality conditions, is subsequently given. The choice of the numerical method employed in this thesis is justified and the components of the algorithm are discussed in detail. This knowledge will be invaluable for the proper problem formulation that enables successful solution of the optimal control problem.

Chapter 3 focuses on suspension modelling of Formula One cars. The analysis is based on considering the kinematics of the connecting rods using classical mechanics. Currently, no such analysis dedicated to a Formula One car suspension has been published in the literature (at least to the author's knowledge). The analysis provided here can thus be used by other researchers, who are seeking such models or want to obtain a deeper understanding of suspensions systems of F1 cars. Following its development, we propose a suitable meta-model that acts as a surrogate for the full non-linear model to reduce the computational burden so that it can be employed in the optimal control calculations.

In Chapter 4, we employ the suspension meta-model developed in the previous chapter to quantify its influence on lap times. The vehicle model in [73] is upgraded with an aerodynamics map that is sensitive to the chassis configuration, which is determined by the suspension meta-model. The interactions between the aerodynamics and the suspension are thereby studied and an extensive set of related parameters are optimised. The proposed methodology can be used both in car set-up before the racing weekend and in the longer term to guide the vehicle design. It will be shown that the solution times remain reasonably low, making them invaluable to performance engineers. The results from this chapter and the previous are largely published in [11] and [12].

In Chapter 5, we study how a local perturbation applied to the aerodynamic balance influences the lap time along the manoeuvre. This can help engineers to see if any compromise made in the aerodynamics at the design stage would actually translate to lap time improvements. Using data from a real F1 driving simulator, the minimum lap time solutions were validated and the vehicle was calibrated for use in the study. After visiting the assumptions behind the popular quasi-steady-state (QSS) methods, we discuss how some of these assumptions can be incorporated into dynamic lap time simulations. The influence of aero-balance perturbations on lap time is studied with these assumptions in place. The influence of including some driver limits on the simulations is also investigated.

In Chapter 6, we solve a minimum fuel problem for a hybrid vehicle with a battery and flywheel as auxiliary storage systems. The problem will be solved over a three-dimensional track to take road gradient information into account. By constraining the vehicle to complete the journey within a given arrival time, we offer a new methodology to standardise the fuel performance evaluation of road vehicles. The advantages of the auxiliary storage systems for different levels of driving aggressiveness are studied. The results from this chapter are summarised in [13].

In Chapter 7, nonlinear stability of vehicular lateral dynamics is studied. Using sum-of-squares programming techniques, the region of attraction (RoA) of a vehicle under constant speed cornering and straight line forward motion is found. The tyre forces are approximated using rational functions, which capture the nonlinear behaviour of the tyre much better than the polynomial approximations that are often used in the literature for RoA estimation. It is shown that with the proposed method, a better RoA estimate can be obtained. The influence of vehicle parameters and driving conditions on the stability region is then investigated. The results from this chapter can be found in [14] and [15].

Finally, we conclude this thesis in Chapter 8 and give directions for future research.

Chapter 2

Optimal Control

CONTROL theory deals with systems whose behaviour can be influenced by external inputs. Optimal control is a field in control theory with a distinguished history and is concerned with choosing the inputs of a system such that it is driven to a desired state whilst minimising some criteria. Its early development dates back to the brachistochrone problem [16] posed by Johann Bernoulli in 1696, which asks the following question: What shape should a curve take, such that a frictionless bead starting at rest at some point can travel to another point in the least time? This problem motivated the start of a new branch of mathematics known as *Calculus of Variations*. What distinguishes this field from the ordinary calculus, is that rather than studying how a change in a quantity can affect the value of a function, we are concerned with how function changes influence the value of a functional[1]. Calculus of variations is the cornerstone of modern optimal control and has had wide-reaching implications in many other branches of modern science and engineering.

The exposition given here aims to provide the necessary background for the understanding of numerical optimal control algorithms used for solving the practical problems presented in this thesis. Therefore, the treatment focuses on highlighting the main results and ideas behind the relevant methods rather than rigour or detail. More importantly, we show how a change in viewpoint in interpretation of theoretical results can enable the development of numerical methods that are better suited to

[1]A functional is a mapping from a set of functions to the real numbers.

certain classes of problems. Nonetheless, necessary references are provided, which the interested reader can refer to for more details.

Many formulations of numerical optimal control rely on methods developed in numerical optimisation. Hence, a brief overview of unconstrained and constrained optimisation methods are provided in Sections 2.1 and 2.2. Necessary conditions of optimality for optimal control problems are then derived and presented in Section 2.4 using the tools arising from calculus of variations described in Section 2.3. The existing numerical methods for solving optimal control problems, which can be divided into two general categories, namely, direct and indirect, are explained in Sections 2.5 and 2.6. The popular methods in each category are then discussed and their relative strengths and weaknesses are highlighted. The pseudospectral method employed for tackling the optimal control problems presented in the rest of the thesis is described in Section 2.7. Finally, some issues related to numerical optimal control methods are discussed in Section 2.8 and the summary is given in Section 2.9.

2.1 Unconstrained Optimisation

The problem of finding the largest or smallest value of a quantity is referred to as an *extremum problem* in mathematics. Typical examples of such problems, to name a few, could be finding the highest peak of a hill, maximising profit of a company or finding the fastest way to travel between two points. It is not difficult to see why studying such problems are of the utmost importance in many branches of engineering and science. To begin our formal exposition, let us consider a simple example. Suppose we need to find the lowest point of a valley represented by $y = f(x)$. To be at the bottom of the valley, we have to find the point that lies below all the other points. If we restrict our attention to only the region nearby our current position, the lowest trough is referred to as the *local minimum*. In contrast, finding the lowest point across the entire valley is the problem of finding the *global minimum*. This concept is illustrated in Figure 2.1.

Definition 2.1 *[17] Consider a function $f : \mathcal{D} \rightarrow \mathbb{R}$ where $\mathcal{D} \subseteq \mathbb{R}^n$ and let the standard Euclidean norm on $\mathbb{R}^n$ to be denoted by $\|.\|$. A point $\boldsymbol{x}^*$ is said to be a 'local minimum' of a function $f(x)$, if there is an $\epsilon > 0$ such that for all the points in $\|\boldsymbol{x}^* - \boldsymbol{x}\| < \epsilon$, the following holds*

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}). \tag{2.1}$$

Conversely, for x^* to be a 'local maximum', we require that

$$f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}). \tag{2.2}$$

Furthermore, if the inequalities (2.1) *or* (2.2) *hold for all $x \in \mathcal{D}$, then the extremum is 'global' over $\mathcal{D}$.*

To establish whether the current position is a local minimum, by definition, the rate of change of valley height must be zero in whichever direction we proceed. If this was not the case and the rate of change was negative in a certain direction, we could move in that direction to reach a lower point, as demonstrated in Figure 2.1. Furthermore, if there is a direction for which the rate of change of height is positive, we could proceed in the opposite direction to arrive at a lower height. The same logic would apply if we were to find the maximum height of a hill, since this can be viewed as minimisation of the negative height. Therefore, we can see that for a point to be an extremum, we require that the rate of change of $f(x)$ has to be zero at that point and this is known as the *first order necessary condition.* The second order conditions can then assist us to determine whether this extremum is a minimum, maximum or a saddle point.

Definition 2.2 ***The first order necessary condition for a function*** $f(\boldsymbol{x})$ *to have $\boldsymbol{x}^*$ as an extremum requires that the differential of $f(\boldsymbol{x})$ to vanish in all directions at $\boldsymbol{x}^*$.*

$$\nabla f(\boldsymbol{x}^*) = \mathbf{0} \tag{2.3}$$

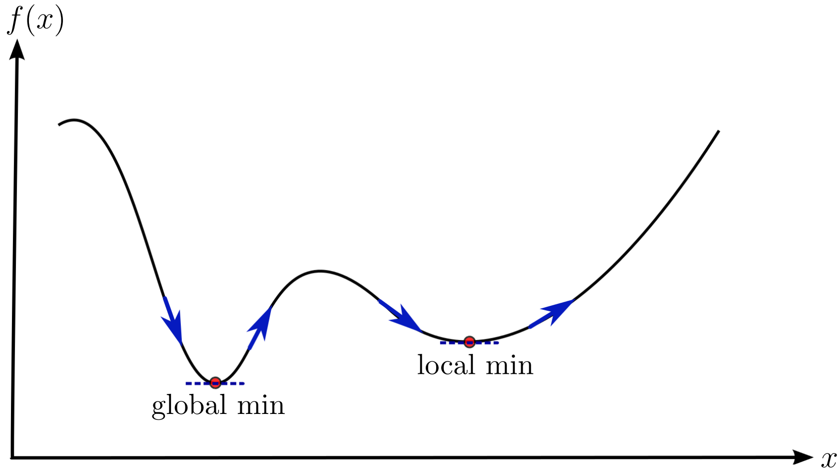

Figure 2.1: Illustration of global and local minima.

where

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)^T. \tag{2.4}$$

A point $\boldsymbol{x}^*$ *satisfying* (2.3) *is called a 'stationary point' of* $f(\boldsymbol{x})$.

Definition 2.3 ***The second order necessary condition for a function*** $f(\boldsymbol{x})$ *to have* $\boldsymbol{x}^*$ *as a local minimum requires that the gradient of* $f(\boldsymbol{x})$ *to vanish in all directions at* $\boldsymbol{x}^*$

$$\nabla f(\boldsymbol{x}^*) = \mathbf{0} \tag{2.5}$$

and the second derivative matrix to satisfy

$$\nabla^2 f(\boldsymbol{x}^*) \geq \mathbf{0}. \tag{2.6}$$

Furthermore, if condition (2.6) *is strengthened so that* $\nabla^2 f(\boldsymbol{x}^*)$ *is positive definite rather than positive semi-definite, then the following condition, along with* (2.5), *are sufficient for the point* $\boldsymbol{x}^*$ *to be a local minimum of* $f(\boldsymbol{x}^*)$.

$$\nabla^2 f(\boldsymbol{x}^*) > \mathbf{0} \tag{2.7}$$

2.1.0.1 Newton's Iterative Method

Suppose we need to find the roots of the nonlinear function $f(x) = 0$. By writing the first order Taylor expansion of $f(x)$, the function value at another point $\tilde{x}$ near x can be approximated by

$$f(\tilde{x}) \approx f(x) + f'(x)(\tilde{x} - x). \tag{2.8}$$

If we then set $f(\tilde{x}) = 0$, we can obtain an expression in terms of the new point $\tilde{x}$, which takes us closer to a root.

$$\tilde{x} = x - f'(x)^{-1} f(x) \tag{2.9}$$

This process can be repeated to improve the root estimate and thus generates the following iterative rule

$$x_{k+1} = x_k - f'(x_k)^{-1} f(x_k). \tag{2.10}$$

Suppose now that rather than finding a root of a nonlinear function, we need to find a local **minimum** of a multi-variable scalar valued function $f(\boldsymbol{x})$. Since, the first order necessary condition requires that $\nabla f(\boldsymbol{x}) = 0$, the iterative method (2.10) can be extended to obtain

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1} \nabla f(\boldsymbol{x}_k), \tag{2.11}$$

where $\nabla f(\boldsymbol{x}_k)$ and $\nabla^2 f(\boldsymbol{x}_k)$ can be interpreted as the Jacobian $\boldsymbol{G}_k$ and Hessian $\boldsymbol{H}_k$ matrices evaluated at $\boldsymbol{x_k}$ to give

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \boldsymbol{H}_k^{-1} \boldsymbol{G}_k. \tag{2.12}$$

2.1.0.2 Quasi-Newton Methods

Consider again the problem of minimising $f(\boldsymbol{x})$. Our aim is to find an iterative rule of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_K \boldsymbol{D}_k \boldsymbol{d}_k, \tag{2.13}$$

where $\alpha_k > 0$ is the step length, $\boldsymbol{D}_k$ is a positive definite matrix and $\boldsymbol{d}_k$ is the search direction, which takes us recursively closer to the minimum from some initial guess $\boldsymbol{x}_0$.

If we start by considering the first order expansion of $f(\boldsymbol{x}_{k+1})$

$$f(\boldsymbol{x}_{k+1}) \approx f(\boldsymbol{x}_k) + \boldsymbol{G}^T(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k), \tag{2.14}$$

we can then substitute $\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = \alpha_K \boldsymbol{D}_k \boldsymbol{d}_k$ into (2.14) to obtain

$$f(\boldsymbol{x}_{k+1}) \approx f(\boldsymbol{x}_k) + \boldsymbol{G}_k^T(\alpha_k \boldsymbol{D}_k \boldsymbol{d}_k). \tag{2.15}$$

Since $\alpha_k > 0$ and we require that the function's value decreases at every step, i.e. such that $f(\boldsymbol{x}_{k+1}) < f(\boldsymbol{x}_k)$, we must have

$$\boldsymbol{G}_k^T \boldsymbol{D}_k \boldsymbol{d}_k < 0. \tag{2.16}$$

One example of such an iterative rule is the *Steepest Descent* method where $\boldsymbol{D}_k = \boldsymbol{I}$ and the search direction is in the opposite direction to the function gradient and thus given by $\boldsymbol{d}_k = -\boldsymbol{G}_k$. The *Steepest Descent* iteration can therefore be expressed as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{G}_k \tag{2.17}$$

As can be seen in (2.17), *Steepest Descent* only uses gradient information to calculate the next step and hence is a first order method. In comparison, Newton's iteration (2.12) utilises both the Jacobian and Hessian information, making it a second order method. This enables Newton's method to have quadratic convergence, which is much faster than the linear convergence of *Steepest Descent.* However, constructing the Hessian information is a costly operation, which should preferably be avoided. *Quasi-Newton* methods approximate the Hessian information recursively and combine the robustness of Steepest Descent with the fast convergence of Newton's iteration.

The idea is to replace $\boldsymbol{D}_k$ in (2.13) by an approximation of $\boldsymbol{H}_k^{-1}$ such that

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_K \boldsymbol{B}_k^{-1} \boldsymbol{d}_k \tag{2.18}$$

where $\boldsymbol{B}_k$ is the Hessian approximation that is recursively updated based on the function and gradient information. There are many algorithms for this update, for example DFP, BFGS and SR1 to name a few. These algorithms are described in detail in [18] and hence we only outline the widely used BFGS algorithm. This update is given by

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k - \frac{\boldsymbol{B}_k \boldsymbol{p}_k \boldsymbol{p}_k^T \boldsymbol{B}_k^T}{\boldsymbol{p}_k^T \boldsymbol{B}_k \boldsymbol{p}_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{p}_k^T \boldsymbol{y}_k} \tag{2.19}$$

where $\boldsymbol{p}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ is the function difference at step k and $\boldsymbol{y}_k = \boldsymbol{G}_{k+1} - \boldsymbol{G}_k$ is the gradient difference. These algorithms are then combined with a *Line Search* method, which finds an appropriate value of step length α_k.

2.2 Constrained Optimisation

Many optimisation problems are subject to equality and inequality constraints. A good treatment of constrained optimisation can be found in [18–20]. The first order necessary conditions for these problems are provided by the *Karush-Kuhn-Tucker* (*KKT*) conditions given in the following theorem.

Theorem 2.1 *[19] Consider the following optimisation problem*

$$\begin{aligned} &\min_{\boldsymbol{x}} && f(\boldsymbol{x}) && (2.20a) \\ &\textit{subject to} && \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0} && (2.20b) \\ & && \boldsymbol{h}(\boldsymbol{x}) \leq \boldsymbol{0} && (2.20c) \end{aligned}$$

with $f : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^{n_g}$ *and* $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^{n_h}$. *Define the Lagrangian function to be*

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^T \boldsymbol{h}(\boldsymbol{x}) \tag{2.21}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{n_g}$ *and* $\boldsymbol{\mu} \in \mathbb{R}^{n_h}$ *are referred to as multipliers. The optimisation problem* (2.20) *is called a Nonlinear Program (NLP). Suppose* $\boldsymbol{x}^*$ *is a local solution to the* *described NLP, and* $\boldsymbol{x}^*$ *is a regular point*[2] *for the constraints. There then exists multipliers* $\boldsymbol{\lambda}^*$ *and* $\boldsymbol{\mu}^*$ *such that*

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \mathbf{0} \tag{2.22a}$$

$$\boldsymbol{g}(\boldsymbol{x}^*) = \mathbf{0} \tag{2.22b}$$

$$\boldsymbol{h}(\boldsymbol{x}^*) \leq \mathbf{0} \tag{2.22c}$$

$$\boldsymbol{\mu}^* \geq \mathbf{0} \tag{2.22d}$$

$$(\boldsymbol{\mu}^*)^T \boldsymbol{h}(\boldsymbol{x}^*) = \mathbf{0}. \tag{2.22e}$$

The conditions (2.22) *provide the first order necessary conditions of local optimality for the NLP described in* (2.20) *and are referred to as the KKT conditions. The condition* (2.22e) *is known as the complementarity condition and implies that if* $h_i(\boldsymbol{x}^*)$ *is inactive then we must have* $\mu_i^* = 0$.

2.2.1 Sequential Quadratic Programming

Consider the following equality constrained optimisation problem

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) \tag{2.23a}$$

$$\text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}. \tag{2.23b}$$

The KKT necessary conditions for this problem are

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0} \tag{2.24a}$$

$$\boldsymbol{g}(\boldsymbol{x}^*) = \mathbf{0} \tag{2.24b}$$

where L is the Lagrangian function given by $L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x})$. This set of nonlinear equations arising from the KKT conditions can be solved using Newton's iterative method (2.11). The iteration can be written as [18]

$$\begin{bmatrix} \nabla^2_{\boldsymbol{xx}} L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k) & \nabla \boldsymbol{g}(\boldsymbol{x}_k) \\ \nabla \boldsymbol{g}(\boldsymbol{x}_k)^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \\ \boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}_k \end{bmatrix} = - \begin{bmatrix} \nabla f(\boldsymbol{x}_k) + \nabla \boldsymbol{g}(\boldsymbol{x}_k) \boldsymbol{\lambda}_k \\ \boldsymbol{g}(\boldsymbol{x}_k) \end{bmatrix} \tag{2.25}$$

[2]A point $\boldsymbol{x}$ is said to be regular if at $\boldsymbol{x}$ the gradients of active constraints remain linearly independent.

Substituting $\boldsymbol{p}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ and noting that $\boldsymbol{\lambda}_k$ will be eliminated after simplification, we can write

$$\begin{bmatrix} \nabla^2_{\boldsymbol{xx}} L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k) & \nabla \boldsymbol{g}(\boldsymbol{x}_k) \\ \nabla \boldsymbol{g}(\boldsymbol{x}_k)^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_k \\ \boldsymbol{\lambda}_{k+1} \end{bmatrix} = - \begin{bmatrix} \nabla f(\boldsymbol{x}_k) \\ \boldsymbol{g}(\boldsymbol{x}_k) \end{bmatrix}. \tag{2.26}$$

Solving (2.26) iteratively would then takes us to the solution of the equality constrained optimisation problem (2.23). We can therefore see that the minimum can be found in a similar manner to the unconstrained case.

Now if we consider the following quadratic program (QP)

$$\min_{\tilde{\boldsymbol{p}}} \quad \nabla f(\boldsymbol{x}_k)^T \tilde{\boldsymbol{p}} + \frac{1}{2} \tilde{\boldsymbol{p}}^T \nabla^2_{\boldsymbol{xx}} L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k) \tilde{\boldsymbol{p}} \tag{2.27a}$$

$$\text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}_k) + \nabla \boldsymbol{g}(\boldsymbol{x}_k)^T \tilde{\boldsymbol{p}} = \boldsymbol{0} \tag{2.27b}$$

The KKT conditions for (2.27) imply that

$$\nabla_{\tilde{\boldsymbol{p}}} \tilde{L}(\tilde{\boldsymbol{p}}^*, \tilde{\boldsymbol{\lambda}}^*) = \nabla f(\boldsymbol{x}_k) + \nabla^2_{\boldsymbol{xx}} L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k) \tilde{\boldsymbol{p}} + \nabla \boldsymbol{g}(\boldsymbol{x}_k) \tilde{\boldsymbol{\lambda}} = \boldsymbol{0} \tag{2.28a}$$

$$\boldsymbol{g}(\boldsymbol{x}_k) + \nabla \boldsymbol{g}(\boldsymbol{x}_k)^T \tilde{\boldsymbol{p}} = \boldsymbol{0}. \tag{2.28b}$$

By setting $\tilde{\boldsymbol{p}} = \boldsymbol{p}_k$ and $\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}_{k+1}$ we can see that (2.26) and (2.28) become identical and hence the solution of QP (2.27) and the original problem (2.23) will be the same [18, 20]. Newton's method for solving the equality constrained problem can thus be interpreted as a *Sequential Quadratic Program* (SQP), where in each iteration $\boldsymbol{p}_k$ and $\boldsymbol{\lambda}_{k+1}$ are found by solving a QP. This viewpoint is extremely useful in the treatment of inequality constraints.

The NLP problem with equality and inequality constraints (2.20) can thus be solved iteratively by finding the solution of the following QP at each step

$$\min_{\tilde{\boldsymbol{p}}} \quad \nabla f(\boldsymbol{x}_k)^T \tilde{\boldsymbol{p}} + \frac{1}{2} \tilde{\boldsymbol{p}}^T \nabla^2_{\boldsymbol{xx}} L(\boldsymbol{x}_k, \boldsymbol{\lambda}_k) \tilde{\boldsymbol{p}} \tag{2.29a}$$

$$\text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}_k) + \nabla \boldsymbol{g}(\boldsymbol{x}_k)^T \tilde{\boldsymbol{p}} = \boldsymbol{0} \tag{2.29b}$$

$$\boldsymbol{h}(\boldsymbol{x}_k) + \nabla \boldsymbol{h}(\boldsymbol{x}_k)^T \tilde{\boldsymbol{p}} \leq \boldsymbol{0} \tag{2.29c}$$

The inequality constrained QP sub-problem can be solved by using any well-known QP implementation [18]. One approach is the *Active Set Method* where the active inequality constraints are identified and treated as equality constraints. The inactive inequality constraints, on the other hand, are ignored. The main challenge is thus finding the correct set of active constraints at each iteration, which can be done by solving an auxiliary sub-problem or using some rules based on the estimates of the Lagrange multipliers. The resulting equality constrained QP is particularly easy to solve, as the inequality constraints appear linearly. A leading Active Set SQP software code is the SNOPT [21] which has found special appeal in solving large scale NLPs.

2.2.2 Interior Point Methods

Consider again the NLP problem with equality and inequality constraint as given in (2.20). Rather than trying to find the active constraints and applying the full KKT conditions (2.22) to find the optimal solution, we can solve a perturbed set of KKT conditions using a homotopy method. To this end, we replace the non-smooth L-shaped set resulting from (2.22c), (2.22d) and the complementarity condition (2.22e) with a smooth approximation. For instance, we can use the hyperbolic approximation $\mu_i = -\frac{\gamma}{h_i(\boldsymbol{x})}$ where $\gamma > 0$ is a smoothing parameter such that as it tends to zero, makes the approximation approach the original L-shaped set. The modified KKT conditions in this case are given by [18]

$$\nabla f(\boldsymbol{x}^*) + \nabla \boldsymbol{g}(\boldsymbol{x})\boldsymbol{\lambda}^* + \nabla \boldsymbol{h}(\boldsymbol{x}^*)\boldsymbol{\mu}^* = \mathbf{0} \tag{2.30a}$$

$$g(\boldsymbol{x}^*) = \mathbf{0} \tag{2.30b}$$

$$\mu_i^* h_i(\boldsymbol{x}^*) + \gamma = \mathbf{0}, \quad i = 1, \ldots, n_h \tag{2.30c}$$

This set of equations is then solved iteratively with decreasing values of γ, such that at the final iteration, the local solution coincides with that obtained from the original KKT conditions.

Another interpretation of this approach is by noticing that the perturbed KKT conditions (2.30) are indeed the necessary conditions for the following barrier problem

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) - \gamma \sum_{i=1}^{n_h} \ln(-h_i(\boldsymbol{x})) \tag{2.31a}$$

$$\text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}. \tag{2.31b}$$

Noting that $\nabla \ln(-h_i(\boldsymbol{x})) = \nabla h_i(\boldsymbol{x}) h_i(\boldsymbol{x})^{-1}$, the KKT conditions for this problem can be written as [19]

$$\nabla f(\boldsymbol{x}^*) - \gamma \sum_{i=1}^{n_h} \nabla h_i(\boldsymbol{x}^*) h_i(\boldsymbol{x}^*)^{-1} + \nabla \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{\lambda}^* = \mathbf{0} \tag{2.32a}$$

$$\boldsymbol{g}(\boldsymbol{x}^*) = \mathbf{0}, \tag{2.32b}$$

which after comparing with (2.30) imply that $\mu_i^* = \gamma h_i(\boldsymbol{x})^{-1}, i = 1, \ldots, n_h$ as given in (2.30c).

We can observe that as $h_i(\boldsymbol{x}) \to 0$, the objective barrier function (2.31a) approaches infinity, even for a very small value of γ. This forces the inequality constraint to remain interior to the feasible region. Solving the barrier problem KKT conditions (2.32) using a Newton-type method results in strong ill-conditioning of the matrices for small values of γ. Therefore, it is more common in practice to solve the full set of perturbed KKT conditions (2.30), which also explicitly solve for the dual variables $\boldsymbol{\mu}$.

The interior point methods have become a strong competitor to SQP methods. One example of such a software is IPOPT [22], which is particularly efficient for systems with a large number of decision variables. A Newton-based method has to be incorporated with appropriate globalisation strategies to ensure that iterations converge to a local minimum from arbitrary initial guesses [20]. IPOPT uses a filter method where the step is only accepted if it improves either the objective function or the constraint violation. It then employs a line search method to choose an appropriate step size when a step is rejected.

2.2.3 Gradient Approximation

In a Newton-type optimisation it is necessary to construct gradient information for the Jacobian and sometimes the Hessian. For some problems, this can be done using analytical derivatives, but this has to be supplied by the user and is often cumbersome. Therefore, in most instances these derivatives have to be approximated.

2.2.3.1 Finite Differencing

The simplest way to approximate the gradient is the finite difference method given by

$$\frac{df}{dx} \approx \frac{f(x+\epsilon) - f(x)}{\epsilon} \tag{2.33}$$

where ϵ is a perturbation value that should be small enough such that a good approximation is obtained but not so small that the error due to subtractive cancellation becomes dominant. The value ϵ should thus be chosen such that the sum of errors due to Taylor series truncation and computer finite precision are minimised.

2.2.3.2 Complex Differentiation

One approach that avoids the issues due to subtractive cancellation on a finite precision machine is the use of complex differentiation. Consider a function $f = u + iv$ of the complex variable $z = x + iy$ that is differentiable in the complex plane. For such a function, the *Cauchy-Riemann* equations state that

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \tag{2.34a}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}. \tag{2.34b}$$

Using (2.34a), we can then write

$$\frac{\partial u}{\partial x} = \lim_{\epsilon \to 0} \frac{v(x + i(y+\epsilon)) - v(x+iy)}{\epsilon}. \tag{2.35}$$

Since we are only interested in real-valued functions of real variables, we can set $y = 0$, $v(x) = 0$ and $u = f(x)$, which allows us to write (2.35) as

$$\frac{\partial f}{\partial x} = \lim_{\epsilon \to 0} \frac{v(x+i\epsilon)}{\epsilon} = \lim_{\epsilon \to 0} \frac{\mathrm{Im}(f(x+i\epsilon))}{\epsilon} \tag{2.36}$$

Therefore, for a small ϵ we have the following *complex-step derivative approximation*

$$\frac{\partial f}{\partial x} \approx \frac{\mathrm{Im}(f(x+i\epsilon))}{\epsilon}. \tag{2.37}$$

This approximation can be shown to be $\mathcal{O}(\epsilon^2)$ accurate, in contrast to the finite difference method which is only $\mathcal{O}(\epsilon)$. Furthermore, since no subtraction occurs in the formula, it does not suffer from subtractive cancellation and hence extremely small values of ϵ can be used to obtain accurate derivatives up to machine precision. However, since it relies on complex arithmetic, it requires twice the memory when compared to the finite difference method.

2.2.3.3 Automatic Differentiation

Automatic differentiation [23] (AD) exploits the fact that any differentiable function is composed of elementary operations and functions such as addition, subtraction, multiplication, sin, cos, log, etc. AD can then use the chain rule of calculus to differentiate the elementary operations such that derivatives of arbitrary order are obtained accurate up to machine precision. The chain rule, however, is applied to the numerical values rather than symbolic expressions. Therefore, whilst providing the same precision as the analytical derivatives, it avoids the symbolic derivation of mathematical expressions, which become recursively more complicated as we proceed through the chain rule. This is especially advantageous for derivative evaluation of large matrices where symbolic manipulation can be slow. There are two main approaches to AD, namely *source transformation* and *operator overloading*. In the former, the user source code is taken as input and associated derivative producing files are generated accordingly. In the context of the numerical optimal control calculations in this thesis, we make use of the *source transformation* AD tool ADiGator [24] based in MATLAB. *Operator overloading* on the other hand, relies on the capabilities of object oriented programming languages to overload operators such that they record operands and their results on a 'tape'. This tape is then interpreted to provide the required derivatives. As such, it negates the need to generate separate files for the derivative information.

2.3 Calculus of Variations

A functional $J : \Omega \rightarrow \mathbb{R}$ is a correspondence rule that maps a function of one or several variables in a certain class Ω to a real number in $\mathbb{R}$. In an manner analogous to the ordinary calculus case, to find a stationary value of the functional J, we require a perturbation on the function $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ to check whether this results in a zero rate of change of the functional. To this end, we introduce a perturbed function $\bar{\boldsymbol{f}}(\boldsymbol{x})$

$$\bar{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) + \epsilon\boldsymbol{\phi}(\boldsymbol{x}), \tag{2.38}$$

where $\boldsymbol{\phi}(\boldsymbol{x})$ is an arbitrary function and ϵ is a small variable parameter tending towards zero. The difference between the original function $\boldsymbol{f}(\boldsymbol{x})$ and perturbed function $\bar{\boldsymbol{f}}(\boldsymbol{x})$ evaluated at $\boldsymbol{x}$ is referred to as the *variation* of the function $\boldsymbol{f}(\boldsymbol{x})$ and is given by

$$\delta\boldsymbol{y} = \bar{\boldsymbol{f}}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}) = \epsilon\boldsymbol{\phi}(\boldsymbol{x}). \tag{2.39}$$

Definition 2.4 *Let $\|.\|$ denote the function norm. A functional $J : \Omega \rightarrow \mathbb{R}$ is said to have a 'local minimum' at $\boldsymbol{y}^*$ if there is an $\epsilon > 0$ such that for all functions $\boldsymbol{y} \in \Omega$ that satisfy $\|\boldsymbol{y}^* - \boldsymbol{y}\| < \epsilon$, the following holds*

$$\Delta J = J(\boldsymbol{y}) - J(\boldsymbol{y}^*) \geq 0 \tag{2.40}$$

Conversely, $\boldsymbol{y}^*$ is a 'local maximum' if

$$\Delta J = J(\boldsymbol{y}) - J(\boldsymbol{y}^*) \leq 0. \tag{2.41}$$

Furthermore, if the inequalities (2.40) *and* (2.41) *hold for all $\boldsymbol{y} \in \Omega$, the extremum is said to be 'global' over Ω [25, 26].*

Theorem 2.2 *[25]* ***The first order necessary condition for a functional*** *$J(\boldsymbol{y})$ to have $\boldsymbol{y}^*$ as an extremum requires that the variation of the functional has to vanish at $\boldsymbol{y}^*$ for all $\delta\boldsymbol{y} \in \Omega$, i.e.*

$$\delta J(\boldsymbol{y}^*, \delta\boldsymbol{y}) = 0 \tag{2.42}$$

Proof. *The increment of functional J at $\boldsymbol{y}^*$ is defined as*

$$\Delta J(\boldsymbol{y}^*, \delta\boldsymbol{y}) = J(\boldsymbol{y}^* + \delta\boldsymbol{y}) - \Delta J(\boldsymbol{y}^*) = \delta J(\boldsymbol{y}^*, \delta\boldsymbol{y}) + g(\boldsymbol{y}^*, \delta\boldsymbol{y}) \cdot \|\delta\boldsymbol{y}\|. \tag{2.43}$$

The $\delta J(\boldsymbol{y}^*, \delta\boldsymbol{y})$ is the linear part of the increment and is referred to as the variation of $J(\boldsymbol{y}^*, \delta\boldsymbol{y})$. The higher order terms are represented by $g(\boldsymbol{y}^*, \delta\boldsymbol{y})$, which goes to zero faster than $\delta\boldsymbol{y}$ does as $\|\delta\boldsymbol{y}\|$ approaches zero

$$\lim_{\|\delta\boldsymbol{y}\|\to 0} g(\boldsymbol{y}^*, \delta\boldsymbol{y}) = 0. \tag{2.44}$$

The sign of ΔJ therefore depends on the sign of δJ for sufficiently small $\|\delta\boldsymbol{y}\| < \epsilon$. Now, we assume that for some variation $\delta\boldsymbol{y}_0 \in \Omega$, the variation of the functional is not zero,

$$\delta J(\boldsymbol{y}^*, \delta\boldsymbol{y}_0) \neq 0. \tag{2.45}$$

For any $\alpha > 0$ such that $\|\alpha\delta\boldsymbol{y}_0\| < \epsilon$, the principle of linearity implies that

$$\delta J(\boldsymbol{y}^*, \alpha\delta\boldsymbol{y}_0) = \alpha\delta J(\boldsymbol{y}^*, \delta\boldsymbol{y}_0) \tag{2.46}$$

$$\delta J(\boldsymbol{y}^*, -\alpha\delta\boldsymbol{y}_0) = -\alpha\delta J(\boldsymbol{y}^*, \delta\boldsymbol{y}_0). \tag{2.47}$$

As mentioned earlier, for $\|\delta\boldsymbol{y}\| < \epsilon$, the signs of the increment and variation are the same. Therefore, $\Delta J(\boldsymbol{y}^*, \alpha\delta\boldsymbol{y}_0)$ can be made to have either sign. However, by Definition 2.4, for $\boldsymbol{y}^*$ to be an extremal, ΔJ has to have the same sign in the neighbourhood of $\boldsymbol{y}^*$. This is a contradiction and proves the theorem. ■

Lemma 2.1 *[27] δ-process has commutative properties. More specifically, the 'derivative of the variation' is equal to the 'variation of the derivative':*

$$\frac{d}{dx}\delta y = \delta\,\frac{d}{dx}y \tag{2.48}$$

and the 'variation of a definite integral' is equal to the 'definite integral of the variation':

$$\delta\int_a^b f(x)dx = \int_a^b \delta f(x)dx \tag{2.49}$$

Proof. *The left hand side of the property* (2.48) *can be written as*

$$\frac{d}{dx}\delta y = \frac{d}{dx}(\bar{f}(x) - f(x)) = \epsilon\phi'(x) \tag{2.50}$$

and the right hand side

$$\delta\frac{d}{dx}y = \bar{f}'(x) - f'(x) = (y' + \epsilon\phi') - y' = \epsilon\phi'(x) \tag{2.51}$$

and hence we arrive at (2.48).

For the definite integral case,

$$\begin{aligned}\delta\int_a^b f(x)dx &= \int_a^b \bar{f}(x)dx - \int_a^b f(x)dx = \int_a^b (\bar{f}(x) - f(x))dx \\ &= \int_a^b \delta f(x)dx.\end{aligned} \tag{2.52}$$

■

2.3.1 The Euler-Lagrange Equation

One of the earliest examples of a dynamics problem solved using variational techniques is the brachistochrone problem formulated by Johann Bernoulli [17]. In problems of this kind, a function $y(x)$ needs to be found such that the definite integral J

$$J(y) = \int_a^b L(y(x), y'(x), x)dx \tag{2.53}$$

shall be stationary subject to the boundary conditions given by

$$y(a) = y_a, \quad y(b) = y_b \tag{2.54}$$

A candidate function $y(x)$ that results in the stationary value of $J(y)$ should, by Theorem 2.2, result in the vanishing of the first variation of J denoted by δJ.

$$\delta J = \delta\int_a^b L(y(x), y'(x), x)dx = 0. \tag{2.55}$$

Since only variations on the function y are meaningful as x is only a ‘dummy’ variable, we consider the variation of x to be zero.

$$\delta x = 0. \tag{2.56}$$

Furthermore, the variations at the boundary conditions (2.54) are zero as the function y is fixed at these points

$$\delta y|_{x=a} = 0 \tag{2.57}$$

$$\delta y|_{x=b} = 0 \tag{2.58}$$

Using Lemma 2.1, we can write (2.55) as

$$\delta J = \int_a^b \delta L(y(x), y'(x), x)dx \tag{2.59}$$

and using the chain rule

$$\delta J = \int_a^b \left(\frac{\partial L}{\partial y}\delta y + \frac{\partial L}{\partial y'}\delta y' \right) dx. \tag{2.60}$$

Integration by parts can be applied to the term involving $\delta y'$ to obtain

$$\int_a^b \frac{\partial L}{\partial y'}\delta y' dx = \frac{\partial L}{\partial y}\delta y\Big|_a^b - \int_a^b \frac{d}{dx}\frac{\partial L}{\partial y'}\delta y dx \tag{2.61}$$

and using (2.57) and (2.58), the first term in (2.61) vanishes, and (2.60) can be written as

$$\delta J = \int_a^b \left(\frac{\partial L}{\partial y} - \frac{d}{dx}\frac{\partial L}{\partial y'} \right) \delta y dx = 0 \tag{2.62}$$

Lemma 2.2 *[28] If function $h(x)$ is continuous in $[a, b]$, and if*

$$\int_a^b h(x)\delta y(x)dx = 0 \tag{2.63}$$

for every function δy that is continuous in $[a, b]$ such that $\delta y(a) = 0$ and $\delta y(b) = 0$, then h must be zero everywhere in $[a, b]$.

Proof. *Suppose $h(x)$ is positive for some $x \in [a, b]$. Since $h(x)$ is continuous in $[a, b]$, there is some subinterval $[x_1, x_2]$ contained in $[a, b]$, where $h(x)$ is also positive. If we choose $\delta y(x)$, which is arbitrary, to be*

$$\delta y(x) = \begin{cases} (x - x_1)(x_2 - x) & x \in [x_1, x_2] \\ 0 & otherwise \end{cases} \tag{2.64}$$

then $\delta y(x)$ is continuous on $[a, b]$ and clearly positive on $[x_1, x_2]$. This implies that

$$\int_a^b h(x)\delta y(x)dx = \int_{x_1}^{x_2} h(x)(x - x_1)(x_2 - x)dx > 0, \tag{2.65}$$

which is a contradiction and proves the lemma. ■

Invoking Lemma 2.2 on (2.62) we arrive at the celebrated *Euler-Lagrange* differential equation

$$\boxed{\frac{\partial L}{\partial y} - \frac{d}{dx}\frac{\partial L}{\partial y'} = 0,} \tag{2.66}$$

which provides the necessary and sufficient condition for the definite integral (2.53) to take a stationary value.

Example 2.1 *Find the path $y(x)$ that connects the two points (x_1, y_1) and (x_2, y_2) with the shortest length.*

Using Pythagoras' theorem, $ds^2 = dx^2 + dy^2$, the path length can be defined by:

$$S(f) = \int_{x_1}^{x_2} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx = \int_{x_1}^{x_2} \sqrt{1 + y'^2} dx \tag{2.67}$$

Setting the Lagrangian to be $L = \sqrt{1 + y'^2}$ the Euler-Lagrange equation gives

$$\frac{\partial L}{\partial y} - \frac{d}{dx}\frac{\partial L}{\partial y'} = 0 - \frac{d}{dx} y'(1 + y'^2)^{-1/2} = y''(1 + y'^2)^{-3/2} \tag{2.68}$$

Since, the term $(1+y'^2)^{-3/2}$ is always positive, setting the right-hand side of (2.68) *to zero results in*

$$\frac{d^2 y^*(x)}{dx^2} = 0, \tag{2.69}$$

which implies $y^*(x)$ must be a constant gradient line with equation $y^* = mx + c$. The constants m and c can be determined using the boundary conditions to give,

$$y^*(x) = \frac{y_2 - y_1}{x_2 - x_1} x + \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1}. \tag{2.70}$$

Thus proving that the shortest distance between two points has to be a straight line.

The derivation provided here is for a single degree of freedom, however, it can be easily modified to account for n degrees of freedom. If the definite integral is a function of several functions it can then be written as

$$J(\boldsymbol{y}) = \int_a^b L(\boldsymbol{y}(x), \boldsymbol{y}'(x), x) dx \tag{2.71}$$

with boundary conditions:

$$\boldsymbol{y}(a) = \left[y_{1a}, y_{2a}, \ldots, y_{na}\right]^T, \quad \boldsymbol{y}(b) = [y_{1b}, y_{2b}, \ldots, y_{nb}]^T. \tag{2.72}$$

In the same manner as the single function case, we can proceed to take the variation of J to obtain

$$\delta J = \int_a^b \left(\frac{\partial L}{\partial \boldsymbol{y}}\delta \boldsymbol{y} + \frac{\partial L}{\partial \boldsymbol{y}'}\delta \boldsymbol{y}'\right) dx = \int_a^b \left(\sum_{i=1}^{n} \frac{\partial L}{\partial y_i}\delta y_i + \sum_{i=1}^{n} \frac{\partial L}{\partial y_i'}\delta y_i'\right) dx. \tag{2.73}$$

Again using integration by parts on the second term and noting that $\delta \boldsymbol{y}(a) = 0$ and $\delta \boldsymbol{y}(b) = 0$, we arrive at

$$\delta J = \int_a^b \sum_{i=1}^{n} \left(\frac{\partial L}{\partial y_i} - \frac{d}{dx}\frac{\partial L}{\partial y_i'}\right) \delta y_i dx = 0. \tag{2.74}$$

Since δy_i's are independent and arbitrary, we proceed by assuming that all but the first one is zero across x. We can then apply Lemma 2.2 to obtain $\frac{\partial L}{\partial y_1} - \frac{d}{dx}\frac{\partial L}{\partial y_1'} = 0$. The same logic can be applied for other δy_i's to arrive at the necessary condition for the multiple degrees of freedom case.

$$\frac{\partial L}{\partial y_i} - \frac{d}{dx}\frac{\partial L}{\partial y_i'} = 0, \quad \text{for } i = 1, \ldots, n \tag{2.75}$$

or more compactly in vector form,

$$\boxed{\frac{\partial L}{\partial \boldsymbol{y}} - \frac{d}{dx}\frac{\partial L}{\partial \boldsymbol{y}'} = \boldsymbol{0}} \tag{2.76}$$

2.4 Variational Optimal Control

In a general optimal control problem we would like to find the control decision vector $\boldsymbol{u}(t)$ that results in the system with state dynamics [17, 25, 26]

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \tag{2.77}$$

taking the minimum value of the cost functional J given by

$$J(\boldsymbol{u}) = \Phi(\boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} L\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) dt. \tag{2.78}$$

Here, t_0 and t_f are the initial final times, L is the *Lagrange* cost-to-go function and Φ is the terminal cost. General optimal control problems are typically subject to path constraints, limits on the values of the control and also boundary conditions. For now, we omit these complications for the sake of clarity and consider a system with fixed initial states $\boldsymbol{x}(t_0) = \boldsymbol{x}_{t_0}$ and initial time t_0. However, we let the final time t_f and terminal state $\boldsymbol{x}(t_f)$ be free. Systems with a more complicated set of constraints will be dealt with later.

The cost functional J (2.78) can be augmented with the constraint (2.77) using the Lagrangian method to make the problem unconstrained. To this end, we introduce the augmented cost functional J_a

$$J_a(\boldsymbol{u}) = \Phi(x(t_f), t_f) + \int_{t_0}^{t_f} \left[L\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) + \boldsymbol{\lambda}^T(t) \left(\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) - \dot{\boldsymbol{x}}(t) \right) \right] dt, \tag{2.79}$$

where $\boldsymbol{\lambda}$ is the adjoint associated with the state dynamics, and we define the *Hamiltonian* $\mathcal{H}$ to be

$$\mathcal{H}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\lambda}(t), t) = L\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) + \boldsymbol{\lambda}^T(t) \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t). \tag{2.80}$$

Since the first order condition for optimality requires that the variation of the functional has to be zero on an extremal, we start by expressing the variations of each term.

$$\delta J_a(\boldsymbol{u}) = \delta\Phi + \delta \int_{t_0}^{t_f} \left(\mathcal{H} - \boldsymbol{\lambda}^T \dot{\boldsymbol{x}} \right) dt. \tag{2.81}$$

The variation on the terminal cost term can be expressed as

$$\delta\Phi = \frac{\partial\Phi}{\partial \boldsymbol{x}(t_f)} \delta\boldsymbol{x}(t_f) + \frac{d\Phi}{dt_f} \delta t_f, \tag{2.82}$$

since only $\boldsymbol{x}(t_f)$ and t_f are free and $\boldsymbol{x}(t_0)$ and t_0 are assumed as fixed in this case. Using the chain rule, the coefficient of δt_f in (2.82) can be written as

$$\frac{d\Phi}{dt_f} = \frac{\partial\Phi}{\partial \boldsymbol{x}(t_f)} \frac{d\boldsymbol{x}(t)}{dt}\bigg|_{t_f} + \frac{\partial\Phi}{\partial t_f} = \frac{\partial\Phi}{\partial \boldsymbol{x}(t_f)} \dot{\boldsymbol{x}}(t_f) + \frac{\partial\Phi}{\partial t_f}. \tag{2.83}$$

The variation in the final value of $\boldsymbol{x}$ comes from a combination of variations caused by t_f and $\boldsymbol{x}(t_f)$ changes, hence

$$\delta \boldsymbol{x}_f = \delta \boldsymbol{x}(t_f) + \dot{\boldsymbol{x}}(t_f)\delta t_f. \tag{2.84}$$

Substituting (2.84) into (2.82) and (2.83) gives

$$\delta \Phi = \frac{\partial \Phi}{\partial \boldsymbol{x}(t_f)}\delta \boldsymbol{x}_f + \frac{\partial \Phi}{\partial t_f}\delta t_f. \tag{2.85}$$

Applying Lemma 2.1 to the second term in (2.81) and noting that t_f is not fixed,

$$\delta \int_{t_0}^{t_f} \left(\mathcal{H} - \boldsymbol{\lambda}^T \dot{\boldsymbol{x}}\right) dt = \int_{t_0}^{t_f} \left(\delta \mathcal{H} - \delta \boldsymbol{\lambda}^T \dot{\boldsymbol{x}} - \boldsymbol{\lambda}^T \delta \dot{\boldsymbol{x}}\right) dt + \left(\mathcal{H} - \boldsymbol{\lambda}^T \dot{\boldsymbol{x}}\right)\Big|_{t_f} \delta t_f. \tag{2.86}$$

Variation in *Hamiltonian* $\mathcal{H}$ is

$$\delta \mathcal{H} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}\delta \boldsymbol{x} + \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}\delta \boldsymbol{\lambda} + \frac{\partial \mathcal{H}}{\partial \boldsymbol{u}}\delta \boldsymbol{u}. \tag{2.87}$$

In a similar fashion to (2.61) in the derivation of the *Euler-Lagrange* equation, the term $-\boldsymbol{\lambda}^T \delta \dot{\boldsymbol{x}}$ in (2.86) can be integrated by parts to obtain

$$\int_{t_0}^{t_f} -\boldsymbol{\lambda}^T \delta \dot{\boldsymbol{x}} dt = -\boldsymbol{\lambda}^T(t_f)\delta \boldsymbol{x}(t_f) + \int_{t_0}^{t_f} \dot{\boldsymbol{\lambda}}^T \delta \boldsymbol{x} dt. \tag{2.88}$$

Noticing that $\delta \boldsymbol{\lambda}^T \dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}^T \delta \boldsymbol{\lambda}$ and substituting (2.87) and (2.88) into (2.86) we obtain

$$\begin{aligned} \delta \int_{t_0}^{t_f} \left(\mathcal{H} - \boldsymbol{\lambda}^T \dot{\boldsymbol{x}}\right) dt = & \int_{t_0}^{t_f} \left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}\delta \boldsymbol{x} + \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}\delta \boldsymbol{\lambda} + \frac{\partial \mathcal{H}}{\partial \boldsymbol{u}}\delta \boldsymbol{u} - \dot{\boldsymbol{x}}^T \delta \boldsymbol{\lambda} + \dot{\boldsymbol{\lambda}}^T \delta \boldsymbol{x}\right) dt \\ & + \left[\mathcal{H}(t_f) - \boldsymbol{\lambda}^T(t_f)\dot{\boldsymbol{x}}(t_f)\right] \delta t_f - \boldsymbol{\lambda}^T(t_f)\delta \boldsymbol{x}(t_f). \end{aligned} \tag{2.89}$$

Using (2.84) in (2.89) and finally regrouping all the derived terms from (2.85) and (2.89) in (2.81), we arrive at

$$\begin{aligned} \delta J_a(\boldsymbol{u}) = & \int_{t_0}^{t_f} \left[\left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}} + \dot{\boldsymbol{\lambda}}^T\right) \delta \boldsymbol{x} + \left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}} - \dot{\boldsymbol{x}}^T\right) \delta \boldsymbol{\lambda} + \frac{\partial \mathcal{H}}{\partial \boldsymbol{u}}\delta \boldsymbol{u}\right] dt \\ & + \left(\mathcal{H}(t_f) + \frac{\partial \Phi}{\partial t_f}\right) \delta t_f + \left(\frac{\partial \Phi}{\partial \boldsymbol{x}(t_f)} - \boldsymbol{\lambda}^T(t_f)\right) \delta \boldsymbol{x}_f. \end{aligned} \tag{2.90}$$

Setting $\delta J_a = 0$ and invoking Lemma 2.2, we can obtain the well-known optimality necessary conditions

$$\dot{\boldsymbol{x}}^{*T}(t) = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{\lambda}^*(t), t) \tag{2.91}$$

$$\dot{\boldsymbol{\lambda}}^{*T}(t) = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{\lambda}^*(t), t) \tag{2.92}$$

$$\boldsymbol{0} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{u}}(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{\lambda}^*(t), t) \tag{2.93}$$

for $t_0 \leq t \leq t_f$ with transversality conditions

$$\mathcal{H}(\boldsymbol{x}^*(t_f), \boldsymbol{u}^*(t_f), \boldsymbol{\lambda}^*(t_f), t_f) = -\frac{\partial \Phi}{\partial t_f}(\boldsymbol{x}^*(t_f), t_f) \tag{2.94}$$

$$\boldsymbol{\lambda}^{*T}(t_f) = \frac{\partial \Phi}{\partial \boldsymbol{x}(t_f)}(\boldsymbol{x}^*(t_f), t_f) \tag{2.95}$$

If we denote the Hamiltonian on the optimal trajectory to be $\mathcal{H}^*$, then its time derivative is given by

$$\begin{aligned}\frac{d\mathcal{H}^*}{dt} = \frac{\partial H^*}{\partial \boldsymbol{x}}\dot{\boldsymbol{x}}^* + \frac{\partial H^*}{\partial \boldsymbol{\lambda}}\dot{\boldsymbol{\lambda}}^* + \frac{\partial \mathcal{H}^*}{\partial \boldsymbol{u}}\dot{\boldsymbol{u}}^* + \frac{\partial \mathcal{H}^*}{\partial t} &= -\dot{\boldsymbol{\lambda}}^{*T}\dot{\boldsymbol{x}}^* + \dot{\boldsymbol{x}}^{*T}\dot{\boldsymbol{\lambda}}^* + \frac{\partial \mathcal{H}^*}{\partial t} \\ &= \frac{\partial \mathcal{H}^*}{\partial t}\end{aligned} \tag{2.96}$$

where $\mathcal{H}^* = \mathcal{H}(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{\lambda}^*(t), t)$. Furthermore, if the Hamiltonian is not explicitly a function of time t, then it is invariant and constant across the optimal trajectory,

$$\mathcal{H}(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{\lambda}^*(t)) = \text{constant}, \tag{2.97}$$

and its value can be determined by (2.94).

The problem considered here assumes that the initial time and states are given. However, extending the derivation to account for free t_0 and $\boldsymbol{x}(t_0)$ is straightforward. This can be done by considering the variations that are caused by the variations of t_0 and $\boldsymbol{x}(t_0)$ in the Lagrange and terminals costs. Furthermore, the boundary conditions constraint can be dealt with by augmenting the cost function and taking the variation of the augmented cost in a similar manner to the above treatment. The necessary conditions for the more complicated case will be stated in the next section.

2.4.1 Pontryagin's Minimum Principle

Optimal control problems in practice typically have bounds on the controls. The total torque provided to the wheels of a vehicle is limited by the engine power, a space rocket can only have a limited thrust and actuators of an industrial robot are subject to saturation. When the control is subject to a bound, the control variation $\delta \boldsymbol{u}$ might enter an inadmissible region and therefore equation (2.93), obtained by requiring $\delta J(\boldsymbol{u}, \delta \boldsymbol{u})$ to vanish, can no longer provide the necessary condition for optimality of the control. Pontryagin and his students in 1956 formulated the necessary conditions for the optimal control in the presence of control bounds [29]. If the set of all admissible controls is denoted by $\mathcal{U}$, this principle states that, the optimal control $\boldsymbol{u}^*$ that results in the minimum value of cost functional J must satisfy

$$\mathcal{H}(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{\lambda}^*(t), t) \leq \mathcal{H}(\boldsymbol{x}^*(t), \boldsymbol{u}(t), \boldsymbol{\lambda}^*(t), t), \quad \forall \boldsymbol{u} \in \mathcal{U} \tag{2.98}$$

for all for $t_0 \leq t \leq t_f$.

Therefore, the optimal control $\boldsymbol{u}^*$ is the one that causes the Hamiltonian to take its global minimum. The principle is also often presented as

$$\boldsymbol{u}^*(t) = \arg\min_{\boldsymbol{u} \in \mathcal{U}} \mathcal{H}(\boldsymbol{x}^*(t), \boldsymbol{u}, \boldsymbol{\lambda}^*(t), t) \tag{2.99}$$

In summary, if the $\boldsymbol{u}^*$ lies on the boundary of $\mathcal{U}$, then (2.98) or (2.99), which are known as the weak form of the Minimum Principle, can be readily employed to find the optimal control. In contrast, if $\boldsymbol{u}^*$ lies inside of $\mathcal{U}$, then the strong form of Minimum Principle $\nabla_{\boldsymbol{u}} \mathcal{H} = \mathbf{0}$, also given in (2.93) can be applied.

Equations (2.91), (2.92), (2.94), (2.95) and (2.98) or (2.98) offer the necessary but not the sufficient conditions for optimality. In much of classical literature and Pontryagin's original work, the Hamiltonian is defined with an opposite sign and thus this result is also referred to as the Maximum Principle. A rigorous proof of the Minimum Principle can be found in [17, 29, 30].

We will now consider the more general optimal control problem with boundary conditions, free initial and final times, with bounds on control, and path constraints.

The first order necessary condition for the following optimal control problem: Minimise the cost functional

$$J(\boldsymbol{u}) = \Phi(x(t_0), t_0, x(t_f), t_f) + \int_{t_0}^{t_f} L\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) dt, \tag{2.100a}$$

subject to the state dynamics

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \tag{2.100b}$$

the boundary conditions

$$\phi(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) = \mathbf{0}, \tag{2.100c}$$

and the path constraints

$$\boldsymbol{c}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \leq \mathbf{0}. \tag{2.100d}$$

are given by

$$\left\{\begin{array}{ll}
\dot{\boldsymbol{x}}^* = \nabla_{\boldsymbol{\lambda}} \mathcal{H}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*) & (2.101a)\\
\dot{\boldsymbol{\lambda}}^* = -\nabla_{\boldsymbol{x}} \mathcal{H}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*) & (2.101b)\\
\boldsymbol{u}^* = \arg\min\limits_{\boldsymbol{u} \in \mathcal{U}} \mathcal{H}(\boldsymbol{x}^*, \boldsymbol{u}, \boldsymbol{\lambda}^*) & (2.101c)\\
\mathbf{0} = \phi(\boldsymbol{x}(t_0), t_0, \boldsymbol{x}(t_f), t_f) & (2.101d)\\
\boldsymbol{\lambda}^T(t_0) = -\dfrac{\partial \Phi}{\partial \boldsymbol{x}(t_0)} + \boldsymbol{\nu}^T \dfrac{\partial \phi}{\partial \boldsymbol{x}(t_0)}, \quad \boldsymbol{\lambda}^T(t_f) = \dfrac{\partial \Phi}{\partial \boldsymbol{x}(t_f)} - \boldsymbol{\nu}^T \dfrac{\partial \phi}{\partial \boldsymbol{x}(t_f)} & (2.101e)\\
\mathcal{H}(t_0) = \dfrac{\partial \Phi}{\partial t_0} - \boldsymbol{\nu}^T \dfrac{\partial \phi}{\partial t_0}, \quad \mathcal{H}(t_f) = -\dfrac{\partial \Phi}{\partial t_f} + \boldsymbol{\nu}^T \dfrac{\partial \phi}{\partial t_f} & (2.101f)\\
\mu_j(t) = 0, \text{ when } \quad c_j(\boldsymbol{x}, \boldsymbol{u}, t) < 0, \quad j = 1, \ldots, n_c & (2.101g)\\
\mu_j(t) \leq 0, \text{ when } \quad c_j(\boldsymbol{x}, \boldsymbol{u}, t) = 0, \quad j = 1, \ldots, n_c & (2.101h)
\end{array}\right.$$

where $\boldsymbol{\mu}(t)$ is the path constraint multiplier, $\mathcal{H}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{u}, t) = L + \boldsymbol{\lambda}^T \boldsymbol{f} - \boldsymbol{\mu}^T \boldsymbol{c}$ is the control Hamiltonian and $\boldsymbol{\nu}$ is the Lagrange multiplier associated with the boundary conditions. Note that we have omitted the optimality superscript * on (2.101d)-(2.101h) for the sake of clarity.

2.5 Indirect Optimal Control

The first order necessary conditions given by (2.101) result in a two-point boundary value problem (TPBVP) as there are boundary conditions that need to be satisfied at both ends. In an indirect optimal control method, the TPBVP is solved numerically to find the optimal control and system trajectory. The simplest indirect method is perhaps the single shooting [31].

2.5.1 Indirect Shooting

Indirect shooting involves making a guess of the boundary values and then numerically integrating the differential equations from one end to the other. The error between the propagated dynamics and the final boundary conditions are calculated and the initial guess is adjusted until this error becomes smaller than a desired threshold. To illustrate this, we consider the optimal control problem in Section 2.4 where the problem has fixed initial conditions but free final time and states. The indirect single shooting for this problem can be summarised as:

1. Define the guess vector to contain the unknown variables
   $\boldsymbol{g} := (\tilde{\lambda}(t_0)_1, \ldots, \tilde{\lambda}(t_0)_n, \tilde{t_f})$
2. Integrate the dynamics vector $\dot{\boldsymbol{p}} = (\dot{\boldsymbol{x}}, \dot{\boldsymbol{\lambda}}) = \left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}^*), -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}^*\right)$ to find $\tilde{\boldsymbol{p}}(t_f)$.
3. Define the error vector to be $\boldsymbol{E}(\boldsymbol{g}) := \begin{bmatrix} \tilde{\boldsymbol{\lambda}}(t_f) - \left.\frac{\partial \Phi}{\partial \boldsymbol{x}(t_f)}\right|_{\boldsymbol{x}(t_f)=\tilde{\boldsymbol{x}}(t_f), t_f=\tilde{t_f}} \\ \mathcal{H}\left(\tilde{\boldsymbol{x}}(t_f), \tilde{\boldsymbol{\lambda}}(t_f), \boldsymbol{u}^*\right) + \left.\frac{\partial \Phi}{\partial t_f}\right|_{\boldsymbol{x}(t_f)=\tilde{\boldsymbol{x}}(t_f), t_f=\tilde{t_f}} \end{bmatrix}$
4. Use some nonlinear solver to find the guess vector $\boldsymbol{g}$ which satisfies the boundary conditions by having $\boldsymbol{E}(\boldsymbol{g}) = 0$. For Newton's iterative method:
   $\left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{g}}(\boldsymbol{g^i})\right)(\boldsymbol{g}^{i+1} - \boldsymbol{g}^i) = -\boldsymbol{E}(\boldsymbol{g^i})$.

The BVP dynamics $\dot{\boldsymbol{p}} = (\dot{\boldsymbol{x}}, \dot{\boldsymbol{\lambda}})$ constitute a Hamiltonian system and by Liouville's theorem [32], its volume in the phase space must be preserved. This means that if the dynamics of $\boldsymbol{x}$ are very stable and contracting fast, then the dynamics of $\boldsymbol{\lambda}$

are expanding and very unstable. Hence, as the dynamics are integrated, any error made in the initial guess will grow, which can result in wild trajectories in the state space. This ill-conditioning of the dynamics can in practice present many numerical difficulties.

Indirect multiple shooting [33] can be used to alleviate the sensitivity issues associated with the single shooting. In this method, the optimisation horizon is broken into multiple intervals and the simple shooting is applied at each subinterval $t \in [t_i, t_{i+1}]$. To ensure continuity of the solution, the end values of states and costates from the integration are constrained to be equal to those at the start of the next interval.

$$\boldsymbol{p}(t_i^-) = \boldsymbol{p}(t_i^+) \tag{2.102}$$

The logic is that by breaking the problem into smaller subintervals, the integration is carried out over a shorter time, thereby reducing the sensitivity, which reduces the chances of divergence problems. The disadvantage of multiple shooting compared to simple shooting is the increased size of the problem as continuity constraints are introduced into the problem. Initial guesses for state and costate equations are also needed for every subinterval introduced, hence the dimension of the root finding problem is significantly increased. Nevertheless, even at the cost of higher number of decision variables, multiple shooting typically performs better than direct shooting due to the reduced sensitivity to initial guesses. Another added benefit of the multiple shooting method is that subinterval simulations can be solved for in parallel to reduce the computation time.

2.5.2 Indirect Collocation

An alternative indirect method which can significantly help with overcoming the ill-conditioning problems of the Hamiltonian dynamics is the simultaneous collocation. In this method, the algebraic-differential system from the necessary conditions are discretised over a defined gird and turned into a finite-dimensional algebraic problem. For the purpose of demonstration, suppose initial and final times are given and define

$$\boldsymbol{y}(\boldsymbol{p}, \boldsymbol{u}) = \dot{\boldsymbol{p}}, \tag{2.103}$$

where $\boldsymbol{p} = (\boldsymbol{x}, \boldsymbol{\lambda})$, to represent the dynamics. We can write the optimal control rule as

$$\boldsymbol{g}(\boldsymbol{p}, \boldsymbol{u}) = \nabla_{\boldsymbol{u}} \mathcal{H} = \boldsymbol{0}, \tag{2.104}$$

and we can combine the boundary conditions into a lumped vector function $\boldsymbol{b}$ such that

$$\boldsymbol{b}\left(\boldsymbol{p}(t_0), \boldsymbol{p}(t_f)\right) = \boldsymbol{0}. \tag{2.105}$$

If we now employ the midpoint rule, and split the time interval $[t_0, t_f]$ into N subintervals defined by t_i for $i = 0, \ldots, N$, we can approximate (2.103) by

$$\boldsymbol{y}\left(\frac{\boldsymbol{p}(t_i) + \boldsymbol{p}(t_{i+1})}{2}, \boldsymbol{u}\left(\frac{t_i + t_{i+1}}{2}\right)\right) - \frac{\boldsymbol{p}(t_{i+1}) - \boldsymbol{p}(t_i)}{t_{i+1} - t_i} = \boldsymbol{0}, \tag{2.106}$$

and (2.104) by

$$\boldsymbol{g}\left(\frac{\boldsymbol{p}(t_i) + \boldsymbol{p}(t_{i+1})}{2}, \boldsymbol{u}\left(\frac{t_i + t_{i+1}}{2}\right)\right) = \boldsymbol{0}. \tag{2.107}$$

If we then define $\boldsymbol{p}_i = \boldsymbol{p}(t_i)$ and $\boldsymbol{u}_i = \left(\frac{t_i + t_{i+1}}{2}\right)$, we can combine (2.105), (2.106) and (2.107) into the following finite-dimensional problem.

$$\boldsymbol{\zeta}(\boldsymbol{V}) = \boldsymbol{0}, \quad \boldsymbol{V} = \left[\boldsymbol{p}_0, \ldots, \boldsymbol{p}_N, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\right]^T. \tag{2.108}$$

Equation (2.108) constitutes a root finding problem and again can be solved using a Newton-type method. This approach was taken in [34] to solve the minimum lap time problem for a motorcycle on the Adria circuit. The path constraints were handled by augmenting the cost functional with penalty functions. This allows the strong form of the Minimum Principle (2.93) to be used and simplifies the first order necessary conditions. The penalty function used to replace a constraint inequality of type $s \geq 0$ was

$$p_b(s) = \begin{cases} (1 + s/h)^n & s > -h \\ 0 & \text{otherwise} \end{cases} \tag{2.109}$$

with the h and n being the tuning parameters used to control the sharpness of the p_b. This penalty function is depicted in Figure 2.2 for $n = 10$ and different values of h.

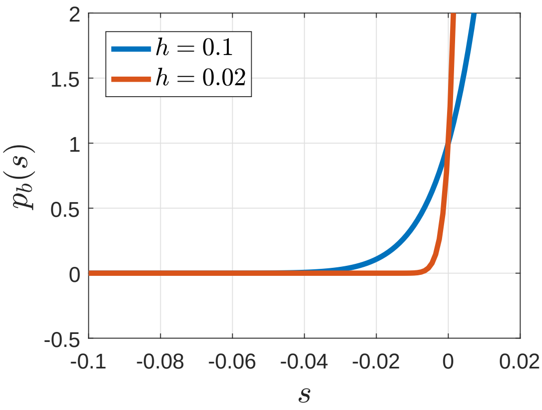

Figure 2.2: Penalty function p_b for $h = 0.1$ and $h = 0.2$ when $n = 10$.

As the constraint is violated, this value of the cost function increases rapidly, which forces the variables to move towards the interior of the admissible region.

The first order necessary conditions were derived using symbolic manipulation software. By focusing on multi-body systems with holonomic constraints, the structure of the conditions was exploited to improve the formulation of the boundary value problem. For the particular studied example, there were 19 states and 3 controls. Discretising the BVP on a grid with 3000 subintervals resulted in over 120000 equations, which were then solved using a modified version of the affine invariant Newton scheme [35].

2.6 Direct Optimal Control

Direct methods aim to first discretise the optimal control problem and then optimise [20]. In indirect methods, the slope of the augmented objective function is found and the roots of the necessary conditions are then located. In the direct approach, we try to find the minimum of the objective by first approximating the infinite-dimensional problem into a finite-dimensional nonlinear program through parametrising the decision variables. Nonlinear programming (NLP) techniques are then applied to solve the optimal control problem. This has the added benefit that problem constraints can be seamlessly integrated into the nonlinear program. Furthermore, sparse solvers now exist that can exploit the structure of the large discretised system to obtain the required solutions fast and robustly. Direct methods are typically divided into two general categories, shooting and collocation. We will describe both the methods and discuss the associated advantages.

2.6.1 Direct Shooting

In a direct shooting method, the controls are parametrised using a specified function approximation such as

$$\boldsymbol{u}_p(t, \boldsymbol{q}) = \sum_{k=1}^{M} q_k \psi_k(t) \tag{2.110}$$

where $\psi_k(t)$'s are the predetermined basis functions and q_k's are the function coefficients, which comprise the problem decision variables. Alternatively, instead of using a global function approximation, the control time horizon can be segmented into N intervals and the control can be represented by piecewise polynomials [36].

$$\boldsymbol{u}_p(t, \boldsymbol{q}) = \sum_{k=1}^{M} q_{ik} p_k(t_i - t), \quad \text{for} \quad t \in [t_i, t_{i+1}]. \tag{2.111}$$

In this case the most common parametrisation is actually the piecewise constant control, where $\boldsymbol{u}_p(t) = q_i$ for $t \in [t_i, t_{i+1}]$. For an OCP with fixed initial conditions and no path constraints, the resulting NLP is

$$\min_{\boldsymbol{q}} J(\boldsymbol{u}_p(t, \boldsymbol{q})) = \Phi(\boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} L\left(\boldsymbol{x}(t), \boldsymbol{u}_p(t, \boldsymbol{q}), t\right) dt. \tag{2.112}$$

subject to

$$\boldsymbol{x}(t_f) - \int_{t_0}^{t_f} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_p(t, \boldsymbol{q})) = \mathbf{0} \tag{2.113}$$

Evaluation of the integral terms naturally requires numerical integration and should be treated as an initial value problem. At every iteration, the gradient information has to be determined by considering the change in the cost function with respect to the perturbation in every single parameter. Hence, each gradient component evaluation requires the integration of the dynamics and the Lagrange cost.

The parametrisation of type (2.110) has shown its effectiveness in launch vehicle and orbit transfer applications [37]. In such applications, the control profile can be approximated accurately with few parameters at each phase. This results in the NLP having a small number of variables. The second type of parametrisation (2.111) can solve a wider class of optimal control problems. However, by having a larger set of parameters, each NLP iteration becomes much more expensive.

2.6.2 Direct Multiple Shooting

In a similar manner to the indirect multiple shooting, to deal with error propagation problems with the single shooting, the trajectory horizon can be split into several segments. The dynamics are then integrated over each interval $[t_i, t_{i+1}]$, which need to be joined together using continuity conditions at the boundaries. The control parametrisation would then take the form of (2.111). However, this time the dynamics $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}_p)$ are integrated from t_i with artificial values $\boldsymbol{s}_i = \boldsymbol{x}(t_i)$ to t_{i+1} over each segment.

$$\boldsymbol{s}_{i+1} = \int_{t_i}^{t_{i+1}} \boldsymbol{f}(\boldsymbol{s}_i, \boldsymbol{q_i}) dt \tag{2.114}$$

The total Lagrange cost is then the sum of

$$l_i(\boldsymbol{s_i}, \boldsymbol{q}_i) = \int_{t_i}^{t_{i+1}} L\left(\boldsymbol{x}(t), \boldsymbol{u}_p(t, \boldsymbol{q}_i), t\right) dt. \tag{2.115}$$

The optimal control problem can then be transcribed into the following NLP.

$$\min_{\boldsymbol{q}} J(\boldsymbol{u}_p(t, \boldsymbol{q})) = \Phi(\boldsymbol{s}_N, t_f) + \sum_{i=0}^{N-1} l_i(\boldsymbol{s_i}, \boldsymbol{q}_i). \tag{2.116}$$

subject to

$$\boldsymbol{x}(t_0) - \boldsymbol{s}_0 = \mathbf{0} \tag{2.117}$$

$$\boldsymbol{x}(t_{i+1}) - \boldsymbol{s}_{i+1} = \mathbf{0}, \quad i = 0, \dots, N-1 \tag{2.118}$$

where (2.117) ensures that the initial conditions are met and (2.118) enforces continuity between each segment. The direct multiple shooting has a significantly larger number of decision variables and constraints compared to the single shooting method. However, both the Jacobian and Hessian matrices of the resulting NLP in this method are sparse and can be solved efficiently using appropriate solvers. Nevertheless, even with the added robustness due to the segmentation, the limitations of shooting still apply.

2.6.3 Direct Collocation

In a direct collocation method both the control and the states are parametrised with appropriate basis functions. Unlike shooting, rather than integrating the dynamics over the specified time segments, the differential equations are discretised and turned into algebraic defect constraints. This would mean that the mesh intervals are typically much smaller in collocation compared to multiple shooting. The optimal control problem is then transformed into an NLP, where the aim is to minimise the objective function with respect to the dynamics defect constraints. This has the added benefit that the path constraint can be naturally dealt with as they can be enforced by the NLP at each discretisation point. Furthermore, the direct collocation method still has the benefit of other direct methods where there is no need to find first order conditions that involve handling the transversality, Pontryagin's minimum principle and adjoint equations.

To illustrate this concept, we start by considering the simple Euler Forward collocation where the state is approximated by

$$\boldsymbol{x}_{k+1} \approx \boldsymbol{x}_k + h_k \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k), \qquad \text{for} \quad k = 0, \ldots, N-1 \tag{2.119}$$

with h_k being the step length at the discretisation point t_k. These equations can be written as defect constraints that need to be enforced at the collocation points $t_0 < t_1, \ldots, < t_{N-1}$.

$$\boldsymbol{\zeta}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k - h_k \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k) = \mathbf{0}, \quad \text{for} \quad k = 0, \ldots, N-1 \tag{2.120}$$

Note that collocation points are the points at which the gradient of the differential equation is approximated. Therefore all collocation points are discretisation points but the converse may not be necessarily true. For example, we might have discretisation points at the boundaries but not treat them as collocation points.

The cost functional can be approximated using the same integration scheme to obtain

$$J_d = \phi(\boldsymbol{x}_N, t_N) + \sum_{k=0}^{N-1} h_k L(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k) \tag{2.121}$$

If we combine the boundary conditions and the defect constraints into a single equality constraint vector $\boldsymbol{g}(\boldsymbol{z})$, where $\boldsymbol{z} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N, \boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1})$ are the problem's decision variables, we arrive at the following NLP.

$$\begin{aligned} &\min_{\boldsymbol{z}} J_d(\boldsymbol{z}) \\ &\text{subject to} \\ &\boldsymbol{g}(\boldsymbol{z}) = \boldsymbol{0} \end{aligned} \tag{2.122}$$

For a typical optimal control problem with 8 states and 2 controls, if 1000 collocation points are employed, we would have more than 10000 decision variables in the NLP. However, most of the elements of the NLP Jacobian are comprised of the defect constraint gradients. A perturbation on a decision variable would only affect the nearby constraints since by design each constraint is only a function of a few decision variables. For example, using the Euler Forward method (2.120), only the adjacent constraints can be influenced by $\boldsymbol{x}_k$. It is this decoupling that differentiates direct collocation from the shooting methods and helps to overcome the sensitivity issues. Furthermore, this decoupling implies that most of the other elements of the Jacobian are zero, which makes the matrix structure sparse. Efficient algorithms can then be used to evaluate the Jacobian and the Hessian matrices of the NLP [20], which makes the direct optimisation feasible for practical applications.

2.6.3.1 NLP Lagrangian Versus the Optimality Adjoint

Suppose the cost function of the optimal control problem is given in the Mayer form

$$J = \Phi(\boldsymbol{x}(t_f), t_f) \tag{2.123}$$

If the cost functional is of the Bolza form (2.78), it is straightforward to convert it into the Mayer form by introducing an extra state where

$$x_{n+1}(t_f) = \int_{t_0}^{t_f} L\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) dt. \tag{2.124}$$

and $\dot{x}_{n+1} = L\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right)$. The cost functional can then be written as

$$J = \tilde{\Phi}(\tilde{\boldsymbol{x}}(t_f), t_f) = \Phi(\boldsymbol{x}(t_f), t_f) + x_{n+1}(t_f) \tag{2.125}$$

The Hamiltonian for the Mayer form is given by $\mathcal{H} = \boldsymbol{\lambda}^T \boldsymbol{f}$, where $\boldsymbol{f}$ should be treated as the dynamics vector augmented with the added variable. In this case, the NLP objective function can be written as

$$F(\boldsymbol{z}) = \Phi(\boldsymbol{x}_N). \tag{2.126}$$

The Lagrangian of this NLP can be written as

$$L(\boldsymbol{z}, \boldsymbol{\lambda}) = F(\boldsymbol{z}) - \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{z}) = \Phi(\boldsymbol{x}_N) - \sum_{k=0}^{N-1} \boldsymbol{\lambda}_k^T \left(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k - h_k \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \right) \tag{2.127}$$

The necessary conditions for this NLP are

$$\nabla_{\boldsymbol{\lambda}_k} L = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k - h_k \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{0} \tag{2.128}$$

$$\nabla_{\boldsymbol{x}_k} L = \boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{k-1} + h_k \boldsymbol{\lambda}_k^T \nabla_{\boldsymbol{x}_k} \boldsymbol{f} = \boldsymbol{0} \tag{2.129}$$

$$\nabla_{\boldsymbol{u}_k} L = h_k \boldsymbol{\lambda}_k^T \nabla_{\boldsymbol{u}_k} \boldsymbol{f} = \boldsymbol{0} \tag{2.130}$$

$$\nabla_{\boldsymbol{x}_N} L = -\boldsymbol{\lambda}_{N-1} + \nabla_{\boldsymbol{x}_N} \Phi = \boldsymbol{0} \tag{2.131}$$

It is not difficult to see that in the limiting case where $h_k \to 0$, (2.128) becomes the state dynamics $\dot{\boldsymbol{x}} = \boldsymbol{f}$, (2.129) approaches $\dot{\boldsymbol{\lambda}} = -\nabla_{\boldsymbol{x}} \mathcal{H}$, which is the adjoint equation, (2.130) becomes the optimal control $\nabla_{\boldsymbol{u}} \mathcal{H} = \boldsymbol{0}$ and (2.131) becomes the transversality condition $\boldsymbol{\lambda}(t_f) = \nabla_{\boldsymbol{x}(t_f)} \Phi$. This demonstrates that the necessary conditions that arise from solving the NLP of the discretised problem approach that of the optimal control. This concept becomes crucial in recovering the adjoint $\boldsymbol{\lambda}$ when a direct collocation method is used. By having the adjoint vector across the time horizon, we will be able to check that the optimal control necessary conditions are met and we will have all the information that an indirect method would provide as part of the solution.

2.6.4 Orthogonal Collocation

In the standard direct collocation methods such as Euler and Runge-Kutta transcription, the order of collocation polynomial is constant and convergence is achieved by reducing the step size h. In an orthogonal collocation or pseudospectral method, the order of the interpolating polynomial is increased to reduce error. Such methods have exponential convergence as a function of the number of collocation points for smooth and well-behaved problems as opposed to Runge-Kutta methods, which only have polynomial convergence. The collocation points are non-uniform and are obtained from the roots of orthogonal polynomials such as Chebyshev and Legendre polynomials. The basis functions for state approximation can be Chebyshev or Lagrange polynomials. However, the majority of the pseudospectral methods employ Lagrange polynomials because they satisfy the isolation property, which results in simpler collocation conditions.

2.6.4.1 Gaussian Quadrature

The quadrature rules based on equidistant points such as the Newton-Cotes formulae can exhibit oscillatory behaviour at the edges of the interval when a high degree polynomial is used. This problem, which can even occur in the quadrature of well-behaved integrands, is known as the Runge's phenomenon, where increasing the polynomial order does not result in an error reduction [38]. Gaussian quadrature rules can overcome this issue and are highly accurate even at relatively low polynomial degrees.

Suppose we use a quadrature rule to approximate the following integral

$$\int_{-1}^{+1} f(\tau)dt \approx \sum_{i=1}^{N} w_i f(\tau_i) \tag{2.132}$$

where w_i are the quadrature weights and τ_i are the associated evaluation nodes. In Gaussian quadrature, these weights and nodes are chosen such that the quadrature error

$$E(f) = \int_{-1}^{+1} f(\tau)dt - \sum_{i=1}^{N} w_i f(\tau_i) \tag{2.133}$$

remains zero for as a high degree of f as possible [39]. It turns out that if the nodes are chosen to be the zeros of the N^{th} degree Legendre polynomial $P_N(\tau)$, with

$$P_n(\tau) = \frac{1}{2^n n!}\frac{d^n}{d\tau^n}[(\tau^2 - 1)^n], \tag{2.134}$$

then the quadrature weights are given by [40]

$$w_i^{LG} = \frac{2}{(1-\tau_i)^2[P_N'(\tau_i)]^2} \tag{2.135}$$

where $P'_N(\tau)$ denotes the derivative of $P_N(\tau)$ with respect to τ. In this case, the quadrature is known as the Legendre-Gauss (LG) quadrature and the nodes are referred to as the LG points. LG quadrature is exact for a polynomial integrand f of degree up to $2N-1$. All LG points lie in the interior of the interval. There are two other related quadrature schemes, widely known as Legendre-Gauss-Radau (LGR) and Legendre-Gauss-Lobbato (LGL). In LGR, one of the interval boundaries is also chosen to be a quadrature point, making it accurate up to degree $2N-2$. The LGR nodes are given by the zeros of the sum of degree N and $N-1$ Legendre polynomials, $P_N(\tau)+P_{N-1}(\tau)$. The weights are given by

$$w_1^{LGR} = \frac{2}{N^2}, \quad w_i^{LGR} = \frac{1}{(1-\tau_i)[P'_{N-1}(\tau_i)]^2} \quad \text{for } i = 2,\ldots,N \tag{2.136}$$

LGL has both boundary points $\tau=-1$ and $\tau=+1$ as quadrature points. Since two degrees of freedom are lost due to the boundary fixed points, LGL is only accurate up to polynomial order $2N-3$. The interior LGL nodes are the zeros of $P'_{N-1}(\tau)$, which is the derivative of the degree $N-1$ Legendre polynomial. The LGL quadrature weights are given by [40]

$$w_{1,N}^{LGL} = \frac{2}{N(N-1)}, \quad w_i^{LGL} = \frac{2}{N(N-1)[P_{N-1}(\tau_i)]^2} \quad \text{for } i = 2,\ldots,N-1 \tag{2.137}$$

The quadrature points for $N=5$ are shown in Figure 2.3. In practice, the quadrature nodes and weights are typically included in computer codes in tabular format or are calculated using dedicated fast algorithms.

Since LGL points have collocation points at both sides of the interval, it makes them the most natural choice for optimal control problems as they include the boundary conditions at each end. However, the LGL scheme leads to a defect in optimality conditions at the boundary points, which makes the costate estimation potentially non-convergent [41]. LG and LGR points do not suffer from this deficiency. The LGR scheme has the initial boundary collocated, which means that the control at the initial point is provided by the NLP solution. This is especially useful when a multiple segment strategy is employed, where using a LGR points would lead to an efficient formulation as the end boundary coincides with the initial collocation point of the next mesh [41].

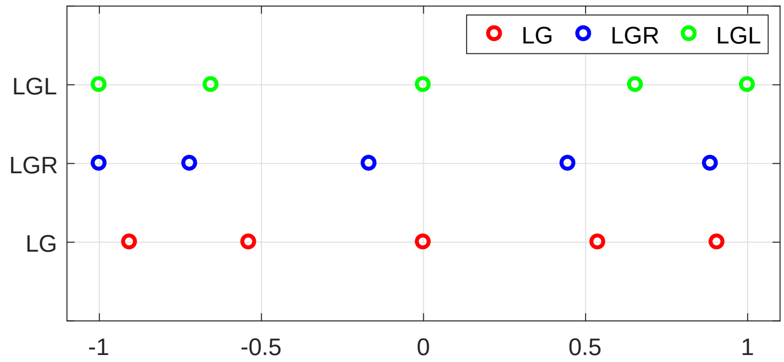

Figure 2.3: $N = 5$ Quadrature points for Legendre-Gauss, Legendre-Gauss-Radau and Legendre-Gauss-Lobato.

2.7 LGR Pseudospectral Method

Throughout this thesis, a LGR pseudospectral method will be used to solve difficult optimal control problems. These problems have the general form of (2.100). In this section we will explain how the infinite-dimensional optimal control problem can be turned into a finite-dimensional problem by using LGR points as collocation points and LGR quadrature for cost functional approximation. The exposition provided here is largely based on [42, 43] and references therein, which entail the operation of the software code GPOPS-II. Since pseudospectral quadrature methods are typically defined on the interval $\tau \in [-1, +1]$, it is natural to introduce the following time transformation

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}, \tag{2.138}$$

which maps the time $t \in [t_0, t_f]$ into the fixed interval $\tau \in [-1, +1]$. This fixed interval is then split into K subintervals with each segment being represented by $\mathcal{S}_k = [T_{k-1}, T_k]$, $k = 1, \ldots, K$, where $-1 = T_0 < T_1 < \ldots < T_K = +1$ are the mesh points. If we define $\boldsymbol{x}^{(k)}(\tau)$ and $\boldsymbol{u}^{(k)}(\tau)$ to be the state and the control in the subinterval $\mathcal{S}_k$, the optimal control problem (2.100) can then be written as follows.

Minimize the cost functional

$$J = \Phi(\boldsymbol{x}^{(1)}(-1), t_0, \boldsymbol{x}^{(K)}(+1), t_f) + \frac{t_f - t_0}{2}\sum_{k=1}^{K}\int_{T_{k-1}}^{T_k} \mathcal{L}(\boldsymbol{x}^{(k)}(\tau), \boldsymbol{u}^{(k)}(\tau), \tau; t_0, t_f)\, d\tau, \tag{2.139}$$

subject to the dynamic constraints

$$\frac{d\boldsymbol{x}^{(k)}(\tau)}{d\tau} = \frac{t_f - t_0}{2}\boldsymbol{f}(\boldsymbol{x}^{(k)}(\tau), \boldsymbol{u}^{(k)}(\tau), \tau; t_0, t_f), \quad k = 1, \ldots, K, \tag{2.140}$$

the path constraints

$$\boldsymbol{c}_{\min} \leq \boldsymbol{c}(\boldsymbol{x}^{(k)}(\tau), \boldsymbol{u}^{(k)}(\tau), \tau; t_0, t_f) \leq \boldsymbol{c}_{\max}, \quad k = 1, \ldots, K, \tag{2.141}$$

and the boundary conditions

$$\boldsymbol{b}(\boldsymbol{x}^{(1)}(-1), t_0, \boldsymbol{x}^{(K)}(+1), t_f) = \boldsymbol{0}. \tag{2.142}$$

Similar to any other multi-interval method, since the state must remain continuous between subintervals, we require that the continuity conditions $\boldsymbol{x}(T_k^-) = \boldsymbol{x}(T_k^+)$ are satisfied at $T_1, \ldots, T_{K-1}$.

The multiple-interval form of the LGR collocation method [41] is then stated as follows. First, the state of the continuous-time optimal control problem is approximated using Lagrange polynomials in $\mathcal{S}_k,\ k \in [1, \ldots, K]$ by

$$\boldsymbol{x}^{(k)}(\tau) \approx \boldsymbol{X}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} \boldsymbol{X}_j^{(k)} \ell_j^{(k)}(\tau), \tag{2.143}$$

where $\ell_j^{(k)}(\tau)$ is the Lagrange polynomials basis given by

$$\ell_j^{(k)}(\tau) = \prod_{\substack{l=1 \\ l \neq j}}^{N_k+1} \frac{\tau - \tau_l^{(k)}}{\tau_j^{(k)} - \tau_l^{(k)}}, \quad j = 1, \ldots, N_{k+1}. \tag{2.144}$$

Using a Lagrange polynomial basis is especially useful as it satisfies the isolation property

$$\ell_j^{(k)}(\tau_l) = \begin{cases} 0 & , \quad j \neq l \\ 1 & , \quad j = l \end{cases} \tag{2.145}$$

which results in the interpolation being exact at the collocation point, i.e. $\boldsymbol{x}^{(k)}(\tau_j) = \boldsymbol{X}^{(k)}(\tau_j)$. The support points for the Lagrange polynomials are the LGR collocation points in $\mathcal{S}_k = [T_{k-1}, T_k)$, with $\tau_{N_k+1}^{(k)} = T_k$ being a noncollocated point. Differentiating $\boldsymbol{X}^{(k)}(\tau)$ in (2.143) with respect to τ gives

$$\frac{d\boldsymbol{X}^{(k)}(\tau)}{d\tau} = \sum_{j=1}^{N_k+1} \boldsymbol{X}_j^{(k)} \frac{d\ell_j^{(k)}(\tau)}{d\tau}. \tag{2.146}$$

Then collocating the dynamics (2.140) at the LGR points, we obtain

$$\sum_{j=1}^{N_k+1} D_{ij}^{(k)} \boldsymbol{X}_j^{(k)} - \frac{t_f - t_0}{2} \boldsymbol{f}(\boldsymbol{X}_i^{(k)}, \boldsymbol{U}_i^{(k)}, \tau_i^{(k)}; t_0, t_f) = \boldsymbol{0}, \quad i = 1, \ldots, N_k \tag{2.147}$$

where

$$D_{ij}^{(k)} = \frac{d\ell_j^{(k)}(\tau)}{d\tau}\bigg|_{\tau=\tau_i^{(k)}}, \quad i = 1, \ldots, N_k, \quad j = 1, \ldots, N_k + 1 \tag{2.148}$$

are the elements of the $N_k \times (N_k + 1)$ LGR differentiation matrix $\boldsymbol{D}^{(k)}$ associated with the mesh interval $\mathcal{S}_k, k \in [1, \ldots, K]$. Alternatively, the dynamics can be collocated using an equivalent *implicit integral form* [41] to obtain the following integral collocation condition

$$\boldsymbol{X}_{i+1}^{(k)} - \boldsymbol{X}_1^{(k)} - \frac{t_f - t_0}{2} \sum_{j=1}^{N_k} I_{ij}^{(k)} \boldsymbol{f}(\boldsymbol{X}_i^{(k)}, \boldsymbol{U}_i^{(k)}, \tau_i^{(k)}; t_0, t_f) = \boldsymbol{0} \quad i = 1, \ldots, N_k. \tag{2.149}$$

where $I_{ij}^{(k)}$ are the elements of the $N_k \times N_k$ LGR integration matrix associated with the mesh interval $\mathcal{S}_k, k \in [1, \ldots, K]$, and is obtained from the inversion of a submatrix of the LGR differentiation matrix as given below.

$$\boldsymbol{I}^{(k)} = \left[\boldsymbol{D}_2^{(k)}, \ldots, \boldsymbol{D}_{N_k+1}^{(k)}\right]^{-1}. \tag{2.150}$$

Furthermore, it was shown in [41] that $\boldsymbol{I}^{(k)} \boldsymbol{D}_1^{(k)} = \boldsymbol{-1}$, where $\boldsymbol{-1}$ denotes a column vector of all ones with length N_k.

The cost functional (2.139) can be approximated with an LGR quadrature

$$J \approx \Phi(\boldsymbol{X}_1^{(1)}, t_0, \boldsymbol{X}_{N_K+1}^{(K)}, t_f) + \sum_{k=1}^{K} \sum_{j=1}^{N_k} \frac{t_f - t_0}{2} w_j^{(k)} \mathcal{L}(\boldsymbol{X}_j^{(k)}, \boldsymbol{U}_j^{(k)}, \tau_j^{(k)}; t_0, t_f), \tag{2.151}$$

where $w_j^{(k)}, j = 1, \ldots, N_k$ are the LGR quadrature weights [40] in the mesh interval $\mathcal{S}_k$. Finally, the optimal control problem can be transcribed into a finite-dimensional nonlinear programming problem (NLP) where the approximated cost functional (2.151) is minimised subject to the defect constraints (2.149), the discretised path constraints

$$\boldsymbol{c}_{\min} \leq \boldsymbol{c}(\boldsymbol{X}_i^{(k)}, \boldsymbol{U}_i^{(k)}, \tau_i^{(k)}, t_0, t_f) \leq \boldsymbol{c}_{\max}, \quad i = 1, \ldots, N_k, \tag{2.152}$$

and the discretised boundary conditions

$$\boldsymbol{b}(\boldsymbol{X}_1^{(1)}, t_0, \boldsymbol{X}_{N_K+1}^{(K)}, t_f) = \boldsymbol{0}. \tag{2.153}$$

The continuity condition in the state at the interior mesh points can be automatically satisfied by using the same variable for $\boldsymbol{X}_{N_k+1}^{(k)} = \boldsymbol{X}_1^{(k+1)}, k = 1, \ldots, K-1$.

It is worth mentioning that some optimal control problems are also subject to integral equality and inequality of type

$$\boldsymbol{q}_{min} \leq \int_{t_0}^{t_f} \boldsymbol{q}(\boldsymbol{x}^{(k)}(\tau), \boldsymbol{u}^{(k)}(\tau), \tau; t_0, t_f) d\tau \leq \boldsymbol{q}_{max}. \tag{2.154}$$

Inclusion of such an integral constraint is straightforward in this framework. This can be done by applying LGR quadrature to (2.154) so that it is transformed into the following algebraic inequality

$$\boldsymbol{q}_{min} \leq \sum_{k=1}^{K} \sum_{j=1}^{N_k} \frac{t_f - t_0}{2} w_j^{(k)} \boldsymbol{q}(\boldsymbol{X}_j^{(k)}, \boldsymbol{U}_j^{(k)}, \tau_j^{(k)}; t_0, t_f) \leq \boldsymbol{q}_{max}, \tag{2.155}$$

which can be directly included in the NLP.

It is also important to mention that inclusion of static parameters is also very convenient as they can simply be added as extra decision variables in the NLP.

2.7.1 Mesh Refinement

Traditional fixed order methods such as Euler, Trapezoidal and Runge-Kutta rely on mesh width reduction for error convergence and hence are known as local collocation or h-methods. In global collocation or pseudospectral methods, one approximating polynomial is used across the entire horizon. The order of this polynomial is then increased to reduce the error, hence giving the name p-methods. For problems, which are infinitely smooth and well-behaved, p-methods benefit from exponential convergence but in the presence of any non-smoothness this behaviour is lost. In this situation, error convergence becomes very slow and an excessively large order polynomial is needed to achieve an acceptable error threshold.

Many optimal control problems have piecewise continuous solutions with discontinuous first order derivatives. In these problems, a p-method would be inefficient and will exhibit a poor error behaviour. One reason for this inefficiency is that a large polynomial order approximation would result in dense Jacobian and Hessian matrices of the defect constraints. The main success of collocation methods arises from the fact they result in sparse NLPs upon transcription. This loss of sparsity at high polynomial orders significantly inhibits the solution speed of the NLP. In contrast to p-methods, an adaptive h-method can use a denser grid at positions where non-smoothness occurs to obtain a relatively good estimate. However, having a fixed order approximation prevents such methods benefiting from the exponential convergence in places where the solution is smooth. This means that they might require an unreasonably large number of mesh segments to achieve the desired error threshold, which will in turn increase the size of the NLP.

In order to combine the benefits of both methods, an h- and p-adaptive pseudospectral method can be implemented where the mesh position and width, and the polynomial order are simultaneously determined based on the error behaviour [44]. If the error across a particular segment has a uniform behaviour, the number of collocation points can be increased. Conversely, if the error at an isolated point within a segment is significantly larger than the errors at other points within the segment, the segment can be subdivided and an appropriate order polynomial can be used in each subinterval. As such, the exponential convergence in the smooth sections is maintained, whilst ensuring that NLP sparsity is not significantly diminished in the presence of non-smoothness.

2.7.1.1 Error Estimation

The error between the calculated solution and the actual solution is the main indicator of where a finer mesh or higher degree polynomial is required. Since the exact solution is not known, this error has to be estimated using some method. For example, in a h-method, a finer mesh can be used and the difference between the solution from the original mesh and the finer mesh can act as an error estimate. In an LGR method where Lagrange polynomials are used with LGR points as support nodes, one strategy would be to use interpolation between the collocation points to find a higher order estimate of dynamics by using the state derivative approximation and comparing them against the values that are produced by the state dynamics equation. More precisely, suppose in the mesh $\mathcal{S}_k = [T_{k-1}, T_k]$, we have N_k collocation points at $\tau_j^{(k)}, j = 1, \ldots, N_k$ with $\tau_{N_k+1}^{(k)} = T_k$ being the non-collocated point at the end of the interval. The state approximation denoted by $\boldsymbol{X}^{(k)}(\tau)$ in this case is as was given in (2.143). Likewise, the control approximation can be written as

$$\boldsymbol{u}^{(k)}(\tau) \approx \boldsymbol{U}^{(k)}(\tau) = \sum_{j=1}^{N_k} \boldsymbol{U}_j^{(k)} \hat{\ell}_j^{(k)}(\tau), \quad \hat{\ell}_j^{(k)}(\tau) = \prod_{\substack{l=1 \\ l \neq j}}^{N_k} \frac{\tau - \tau_l^{(k)}}{\tau_j^{(k)} - \tau_l^{(k)}} \tag{2.156}$$

with $\boldsymbol{U}_j^{(k)}, j = 1, \ldots, N_k$ giving the control values at the collocation points.

Now we will introduce a finer estimate with M_k LGR points such that $M_k > N_k$. We will denote the new set of support points by $\hat{\tau}_j^{(k)}, j = 1, \ldots, M_k + 1$. The right-hand side of the dynamics, i.e. the function $\boldsymbol{f}$ can then assist us to obtain an improved approximation of the state derivative. To this end, if we use the *differential form* of LGR collocation, we obtain

$$\hat{\dot{\boldsymbol{X}}}(\hat{\tau}_j{}^{(k)}) = \sum_{l=1}^{M_k+1} \hat{D}_{jl}^{(k)} \hat{\boldsymbol{X}}(\hat{\tau}_j{}^{(k)}), \quad j = 1, \ldots, M_k, \tag{2.157}$$

where $\hat{D}_{jl}^{(k)}$ is the $M_k \times M_k + 1$ LGR derivative matrix corresponding to the finer grid with collocation points at $\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_{M_k}^{(k)}$. On the other hand,

$$\dot{\boldsymbol{X}}(\hat{\tau}_j{}^{(k)}) = \frac{t_f - t_0}{2} \boldsymbol{f}(\boldsymbol{X}^{(k)}(\hat{\tau}_l{}^{(k)}), \boldsymbol{U}^{(k)}(\hat{\tau}_l{}^{(k)}), \hat{\tau}_l{}^{(k)}), \quad j = 1, \ldots, M_k, \tag{2.158}$$

gives the state derivatives from interpolation of the original state derivatives. Clearly, the state derivative absolute error estimate can then be written as [44]

$$\boldsymbol{E}_{\dot{\boldsymbol{X}}}^{(k)}(\hat{\tau}_j{}^{(k)}) = |\hat{\dot{\boldsymbol{X}}}(\hat{\tau}_j{}^{(k)}) - \dot{\boldsymbol{X}}(\hat{\tau}_j{}^{(k)})|, \quad j = 1, \ldots, M_k. \tag{2.159}$$

Alternatively, if we seek a state error estimate, we can define

$$\hat{\boldsymbol{X}}(\hat{\tau}_j{}^{(k)}) = \boldsymbol{X}^{(k)}(\tau_{k-1}) + \frac{t_f - t_0}{2} \sum_{l=1}^{M_k} \hat{I}_{jl}^{(k)} \boldsymbol{f}(\boldsymbol{X}^{(k)}(\hat{\tau}_l{}^{(k)}), \boldsymbol{U}^{(k)}(\hat{\tau}_l{}^{(k)}), \hat{\tau}_l{}^{(k)}), \tag{2.160}$$

for $j = 2, \ldots, M_k + 1$ with $\hat{I}_{jl}^{(k)}$ being the $M_k \times M_k$ LGR integration matrix corresponding to LGR points at $\hat{\tau}_1^{(k)}, \ldots, \hat{\tau}_{M_k}^{(k)}$. Then, by interpolating the original state variables, we can determine $\boldsymbol{X}(\hat{\tau}_j{}^{(k)})$ to arrive at the following state absolute error estimate [43].

$$\boldsymbol{E}_{\boldsymbol{X}}^{(k)}(\hat{\tau}_j{}^{(k)}) = |\hat{\boldsymbol{X}}(\hat{\tau}_j{}^{(k)}) - \boldsymbol{X}(\hat{\tau}_j{}^{(k)})|, \quad j = 1, \dots, M_k + 1. \tag{2.161}$$

The relative state error vector can then be defined by

$$\boldsymbol{e}_{\boldsymbol{X}}^{(k)}(\hat{\tau}_j{}^{(k)}) = \frac{\boldsymbol{E}_{\boldsymbol{X}}^{(k)}(\hat{\tau}_j{}^{(k)})}{1 + \max\limits_{l \in [1, \dots M_k+1]} |\boldsymbol{X}(\hat{\tau}_l{}^{(k)})|}. \tag{2.162}$$

The maximum relative state error, $e_{\max}^{(k)}$, is then given by the largest element of $\boldsymbol{e}_{\boldsymbol{X}}^{(k)}$ across all the new LGR points in mesh $\mathcal{S}_k$. By considering an example that had an analytical solution, it was shown in [43] that the state error estimates from this method match the actual errors very closely.

2.7.1.2 Mesh Adaptation

Many mesh adaptation strategies have already been developed for pseudo-spectral methods with some examples including [43–45]. The goal of these algorithms is to decide whether a segment should be divided into more intervals or the order of approximating polynomial should be increased when the desired error threshold is not met across that segment. One example of such an algorithm is the ph-adaptive method described in [43]. In this scheme, the order of polynomial is first increased to check whether the required error threshold can be achieved. The size of this order increase is based on the following rule

$$P_k = \left\lceil \log\left(\frac{e_{\max}^{(k)}}{\epsilon}\right) \right\rceil, \tag{2.163}$$

where $e_{\max}^{(k)}$ is the maximum relative error estimate as explained earlier, ϵ is the desired relative error threshold and $\lceil . \rceil$ denotes the ceiling operation. This rule was obtained by considering the error behaviour of pseudospectral methods. The user defined parameters $N_{\min}$ and $N_{\max}$ set the minimum and maximum number of collocation points in each mesh. If N_k is the number of collocation points in mesh $\mathcal{S}_k$ at the current iteration, then the polynomial order is increased by P_k, as long as $N_k + P_k$ is less than $N_{\max}$. In contrast, if $N_k + P_k > N_{\max}$, then the mesh has to be subdivided. Each new mesh is then started with $N_{\min}$ collocation points with the number of new subintervals in S_k being given by

$$B_k = \max\left(\left\lceil \frac{N_k + P_k}{N_{\min}} \right\rceil, 2\right). \tag{2.164}$$

This process is then repeated until the maximum relative error across all meshes is reduced under the desired level or the maximum number of iterations is reached. It is worth mentioning that if the user defines $N_{\min} = N_{\max}$, the described scheme becomes a standard fixed order method with adaptive step-size.

2.8 Related Issues

In this section, we discuss two of the limitations of the direct optimal control methods and suggest remedies to overcome these issues. One is the presence of discontinuities in the problem formulation and the other the presence of singular arcs along the solution.

2.8.1 Non-smooth Features

Newton-type algorithms typically require first-order derivative information and sometimes also second-order information for functions defining the cost and constraints. For this reason any discontinuity or non-smooth problem features have to be approximated in a way that does not significantly change the problem's solution. Functions such as $\min(x, 0)$ and $\max(x, 0)$ have undefined derivatives at $x = 0$, and are therefore approximated using [4],

$$\max(x, 0) \approx \frac{x + \sqrt{x^2 + \epsilon}}{2} \quad \text{and} \quad \min(x, 0) \approx -\frac{-x + \sqrt{x^2 + \epsilon}}{2}, \tag{2.165}$$

in which ϵ is a 'small' constant. Since

$$\frac{\partial}{\partial x}\left(\frac{1}{2}\left(x + \sqrt{x^2 + \epsilon}\right)\right) = \frac{1}{2}\left(\frac{x}{\sqrt{x^2 + \epsilon}} + 1\right), \tag{2.166}$$

one sees that the derivative approximation for $\max(x, 0)$ at $x = 0$ is now well-defined. One may also make use of $\max(a, b) = a + \max(0, b - a)$.

In the case of $|x|$, one can use

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

in this case

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

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

2.8.2 Singular Arcs

In many optimal control problems, the Hamiltonian is linear in control and can be written as

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = \mathcal{H}^0(\boldsymbol{x}, \boldsymbol{\lambda}) + \boldsymbol{u}\mathcal{H}^1(\boldsymbol{x}, \boldsymbol{\lambda}). \tag{2.167}$$

where $\mathcal{H}^1$ is referred to as the switching function,

In such situations, if the control is bounded, according to PMP (2.99) the optimal control would be switching between the boundaries and the profile would typically be bang-bang so long as the switching function remains non-zero. However, when $\mathcal{H}^1 = 0$, PMP provides no information as to how the optimal control should be chosen. This situation manifests itself as a singularity in the transcribed NLP where the optimisation routine becomes unable to choose the correct control. Along the singular arc the control then becomes oscillatory, which inhibits the convergence of the solution. However, this behaviour is caused by the nature of numerical algorithms and it is not in fact the optimal solution.

Application of (2.99) in this case results in

$$\nabla_{\boldsymbol{u}}\mathcal{H}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*) = \mathcal{H}^1(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}, \tag{2.168}$$

which also implies $\nabla^2_{\boldsymbol{uu}}H = \mathbf{0}$ as $\mathcal{H}^1$ does not involve $\boldsymbol{u}$. In this case (2.168) can be differentiated with respect to time repeatedly until an expression explicit in $\boldsymbol{u}$

appears. The additional necessary condition of optimality for the singular arcs is given by the generalised Legendre-Clebsch condition [46]

$$(-1)^k \frac{\partial}{\partial u}\left(\frac{d^{2k}\nabla_{\boldsymbol{u}}\mathcal{H}}{dt^{2k}}\right) \geq 0, \qquad k = 1, 2, \ldots \tag{2.169}$$

In context of a direct optimal control method, the additional necessary condition can then be used as a feedback law to remove the singularity and enable the optimiser to choose the optimal control. This can be done by having a multi-phase problem description, where the derived feedback law is inserted in the phase where the singularity occurs.

Nevertheless, one of the main motivations of using a direct method is avoiding the cumbersome procedure of deriving the optimality conditions. This means that it is highly desirable to avoid deriving the higher order necessary conditions along singular sub-arcs. Some remedial solutions have been suggested in the literature to alleviate this issue. In [47] it was proposed that a perturbation term of form $\epsilon/2\boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u}$ be added to the Lagrange cost, with $\boldsymbol{R}$ being a positive definite weight matrix. This would make $\boldsymbol{H}_{uu}$ non-singular along the singular sub-arcs for some $\epsilon > 0$.

$$\widetilde{\boldsymbol{H}}_{uu}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) = \boldsymbol{H}_{uu}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}) + \epsilon\boldsymbol{R} > 0. \tag{2.170}$$

The problem can then solved successively for deceasing values of ϵ. This makes the solution approach that of the original problem in a similar way to a homotopy method. Results from each step are then used as the starting guess for the next iteration. When ϵ reaches below a pre-defined threshold, the result is then chosen as the optimal solution. However, it was found that as ϵ approaches zero, the problem would become singular again, at which point the algorithm should be stopped [47].

In [48] a similar idea was proposed which did not require ϵ to go to zero. A *piece-wise derivative variation* perturbation term was introduced in the cost objective and it was shown that the solution of transcribed problem would approach that of the original as the number of mesh points increased whilst subduing the erratic oscillations. A similar but simpler to implement approach is suggested here, in which a regularisation term is added for the control, such that the modified Lagrange cost becomes

$$\tilde{J} = \int_{t_0}^{t_f} \big(L(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \epsilon/2\dot{\boldsymbol{u}}^T \boldsymbol{R}\dot{\boldsymbol{u}}\big)\, dt. \tag{2.171}$$

Intuitively, since the cost functional is augmented with a term that penalises fast changes in control, the optimisers choose a control that is more smooth, which helps the problem converge faster.

2.9 Summary

In this chapter, we provided the necessary background for understanding numerical optimal control algorithms. We started with an explanation of optimisation methods and presented the main general techniques of solving nonlinear programs (NLP). It was shown how the main results from calculus of variations can be used to derive the necessary optimality conditions for general optimal control problems.

Indirect optimal control methods aim to solve the boundary value problem (BVP) arising from the necessary conditions. Various methods of solving the BVP were then discussed. In this thesis, a direct LGR pseudospectral method is used. We showed how LGR collocation can be used to transcribe an optimal control problem into a finite-dimensional problem, which can then be solved using a suitable NLP solver. An insight into the practical implementation of an LGR pseudospectral method was then provided to illustrate the inner workings of the algorithm used for solving the optimal control problems presented in subsequent chapters.

Chapter 3

Suspension Modelling of a Formula-1 Car

3.1 Introduction

Suspension systems are sprung mechanisms that connect the chassis of a car to the wheel carriers. This flexible connection uses a parallel spring-damper combination to support the weight of the vehicle, improve road holding and isolate the driver from road disturbances. An introductory discussion of vehicular suspension systems can be found in [49]. A number of independent suspensions are surveyed in [50], where the synthesis of suspension linkages is studied, including their kinematic structure, their dimensions, and their compliance properties. The well-known MacPherson strut suspension is modelled in [51] as a three-dimensional kinematic mechanism; both linear and nonlinear analyses are provided. A MacPherson suspension mechanism model is also developed [52], where a parameter estimation technique based on the singular value decomposition is studied.

Multiple link suspension systems, of the type used in open-wheeled race cars, have been analysed in both industrial and academic papers. The design of a race car suspension is considered in [53]. This paper outlines a procedure for determining optimal suspension parameters, and contains an example that includes a heave spring, anti-roll bar, torsion bar and rocker assembly. The kinematic and dynamic behaviours of a five-link suspension unit are studied in [54], where the movement of the wheel carrier is constrained by five rods terminated at each end with a spherical joint. Rigid-body analyses of five-rod suspensions are extended to mechanisms with flexible joints in [55]. Flexible joints with experimentally determined linearized compliance matrices are considered, leading to an algorithm for the elasto-kinematic analysis of the mechanism. The optimal synthesis of a five-link independent suspension system is considered in [56]. The synthesis objective is the minimisation of variations in the wheel track, and the toe and camber angles during the jounce and rebound of the wheel. The kinematic design of double-wishbone suspension systems is studied in [57], where multi-objective design optimisation is employed. The design parameters include the lengths of the links and the position of the ground joints. An in-plane 2-DOF kineto-dynamic quarter-car model for a double wishbone suspension was developed in [58] to obtain equivalent stiffness and damping rates for response analysis and suspension synthesis. The kinematics of a MacPherson strut mechanism and a five rod Mercedes 190 series rear wheel suspension are studied in [59] using symbolic computation and interval analysis.

Aerodynamics influences play a major role in the performance of Formula One cars. It is important for high-performance car designers to minimise drag, while simultaneously seeking to maximise downforce effects in order to improve road holding and tyre performance. Although Formula One regulations limit the use of ground effect aerodynamics, they are still an effective means of creating downforce with only a small drag penalty. Since aerodynamic effects are strongly influenced by the orientation and proximity of the car to the road, the vehicle's suspension system plays an important part in determining the aerodynamic performance of the car.

A typical Formula One car deploys a double wishbone suspension system with a push rod, or pull rod for force transmission, and a rack-and-pinion steering arrangement. In this chapter, we aim to model a general purpose double wishbone suspension system so that its effect on aerodynamic performance can be studied and optimised in Chapter 4. The kinematics of the suspensions are derived by studying its geometrical properties. Using force analysis, the normal wheel loads are then related to the suspension configuration. In order to keep the computational burden to an acceptable

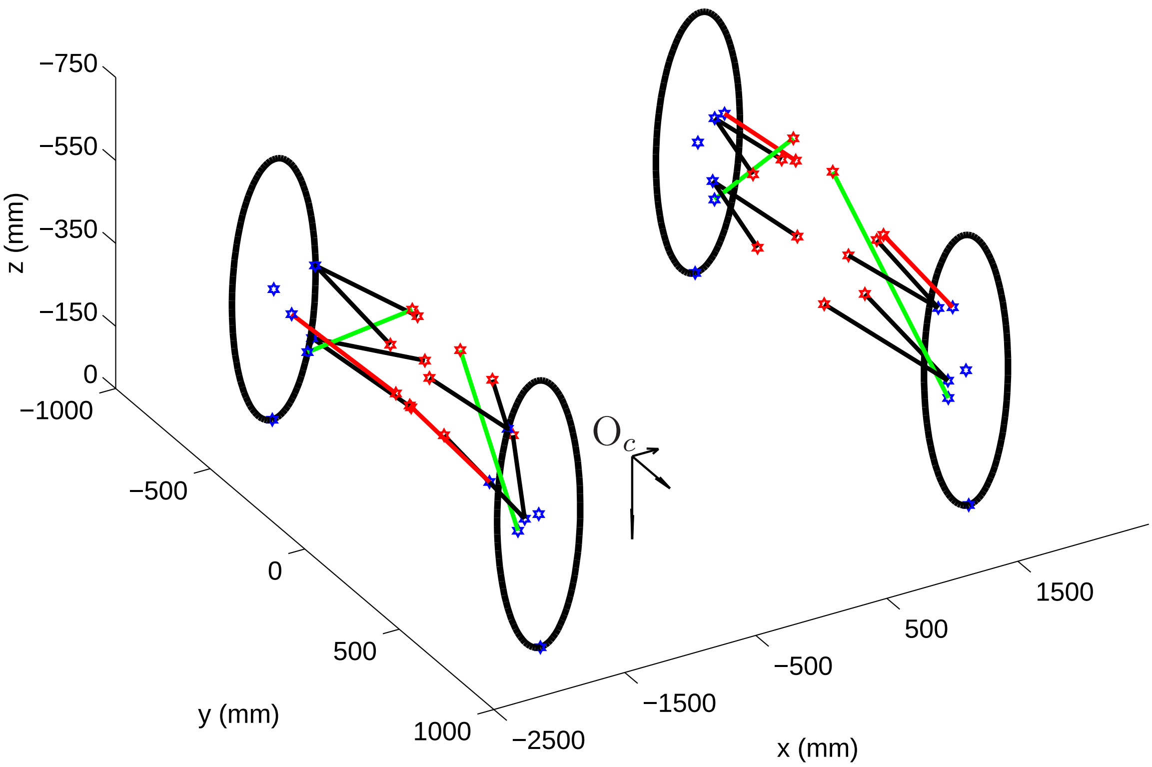

Figure 3.1: The suspension system of a typical Formula One car. The wishbones are shown in black, the suspension push rods are shown in green and the track rods are shown in red. The origin of the SAE coordinate system is O_c.

level for use in optimal calculations, an appropriate meta-model, which captures the suspension behaviour accurately is presented.

In Figure 3.1 the 24 elements of a Formula One car suspension are shown; the wishbones are coloured black, the push rods are coloured green and the track rods are coloured red. The wheel carrier fixed points are shown as blue hexagrams. The car-body fixed points, the chassis ends of the push rods and the chassis ends of the track rods are marked with red hexagrams.

3.2 Suspension Analysis

The kinematics of a suspension system are determined by its motion constraints. To fix ideas, we will focus on Figure 3.2 that shows the front-left suspension of a Formula One car. Rods one to four make up the double-wishbone suspension elements. Rod five is the steering track rod, which connects the wheel carrier to the steering

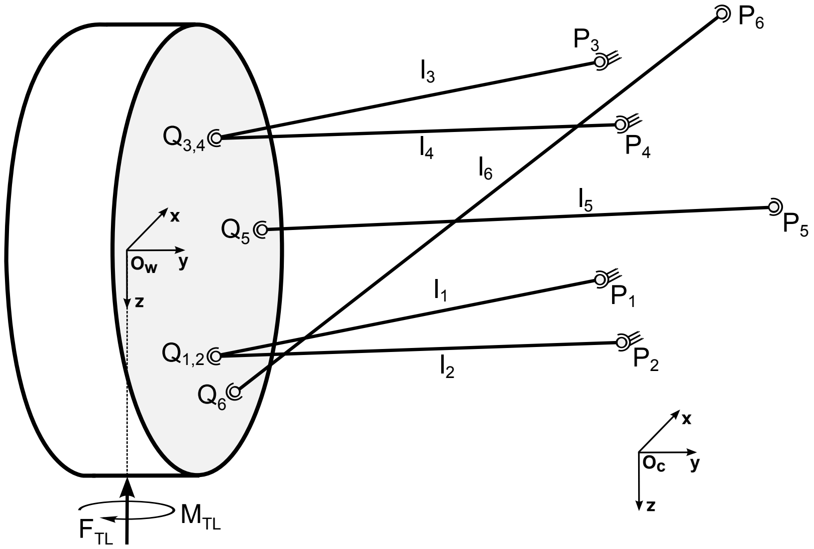

Figure 3.2: Front-left suspension configuration of a typical Formula One car with a push rod suspension system. The axis system O_Cxyz is fixed in the car body, while O_Wxyz is fixed in the wheel carrier with its origin at the wheel's geometric centre. The points P_i are in the car chassis (with points $P_1 \cdots, P_4$ fixed in the car chassis)and the points Q_i are fixed in the wheel carrier. The lengths of the double wishbones are $l_1 \cdots, l_4$, the push rod l_5 and the track rod l_6.

rack. Rod six is the suspension push bar, which connects the wheel carrier to a suspension strut rocker mechanism mounted within the car's body; see Figure 3.3. In the following analysis all the joints in Figure 3.2 are treated as spherical.

In order to describe the suspension motion constraints in detail, we introduce car-fixed and wheel-carrier-fixed reference frames O_Cxyz and O_Wxyz respectively. As shown in Figure 3.2, points P_1 to P_4 are fixed in car chassis and are connected to Q_1 to Q_4 that are fixed in the wheel carrier using fixed-length wishbones of length l_i $i = 1, \cdots, 4$. Points P_5 and Q_5 are at opposite ends of the steering track rod with Q_5 fixed in the wheel carrier. Point P_5 is connected to the steering rack and moves in the car body as the steering wheel position is varied. Points P_6 and Q_6 are at opposite ends of the suspension push rod with Q_6 fixed in the wheel carrier. Point P_6 is connected to a rocker mechanism and moves in the car body in sympathy with movements of the push rod; see Figure 3.3. The steering track rod and suspension push rod have lengths l_5 and l_6 respectively. This system has two degrees of freedom

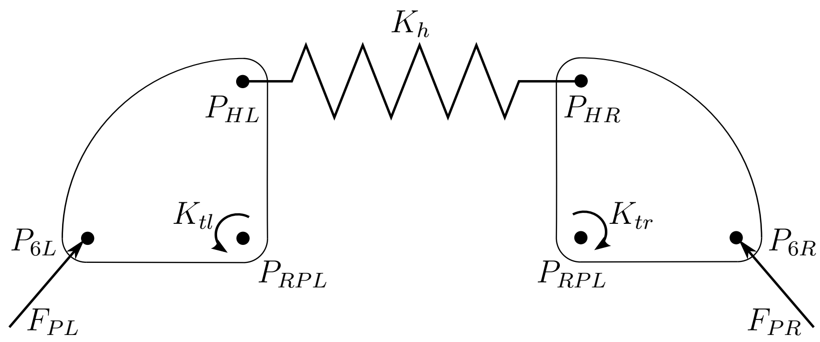

Figure 3.3: Push-rod and rocker set-up. The heave spring has stiffness K_h. Right- and left-hand quantities are labelled with subscripts R and L respectively.

corresponding to changes in the steering angle and to changes in the angular position of the suspension rocker mechanism. The suspension rockers are also connected to a parallel heave spring and damper combination, a torsion bar and an anti-roll bar; the damper is not considered in the quasi-static analysis presented here. A tyre moment M_{TL} is applied to the wheel carrier body, with a tyre force F_{TL} applied at the wheel's ground contact point.

Six motion constraints derive from the fixed lengths of the six suspension elements. If we suppose that P_i and Q_i are vector representations of the corresponding points in Figure 3.2 (expressed in a common reference frame), then the magnitudes of each of the six differences $P_i - Q_i$ (lengths) must be fixed too. That is

$$\|P_i - Q_i\|_2^2 = l_i^2 \qquad i = 1, \cdots, 6 \tag{3.1}$$

in which the lengths l_i are constant. A steering angle δ results in a displacement of the steering rod mount point in the y-axis direction, which is proportional to the radius r of the steering pinion

$$P_5 = \begin{bmatrix} 0 & -r\delta & 0 \end{bmatrix}^T + P_5^0; \tag{3.2}$$

P_5^0 is the straight-running position of P_5.

These equations determine a hypersurface over which the twelve points in Figure 3.2 must move. In order to describe points that are fixed in the wheel carrier, in the car's reference frame, we need to recognise the relative motion of the two. There are six degrees of relative motion movement that are constrained by (3.1).

The translational freedoms of the wheel carrier (relative to the car) are

$$D = \left[\begin{array}{ccc} \Delta x & \Delta y & \Delta z \end{array} \right]^T. \tag{3.3}$$

The rotational freedoms of the wheel carrier relative to the vehicle's chassis are described in terms of the elementary rotations $R_x(\phi)$, $R_y(\theta)$ and $R_z(\psi)$, which are described by

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}_\phi & -\mathrm{s}_\phi \\ 0 & \mathrm{s}_\phi & \mathrm{c}_\phi \end{bmatrix} \; R_y(\theta) = \begin{bmatrix} \mathrm{c}_\theta & 0 & \mathrm{s}_\theta \\ 0 & 1 & 0 \\ -\mathrm{s}_\theta & 0 & \mathrm{c}_\theta \end{bmatrix} \; R_z(\psi) = \begin{bmatrix} \mathrm{c}_\psi & -\mathrm{s}_\psi & 0 \\ \mathrm{s}_\psi & \mathrm{c}_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

in which s_ϕ and c_ϕ are the sine and cosine, respectively, of ϕ. The same notation is used for θ and ψ. As a result

$$x_{ci} = R_z(\psi)R_x(\phi)R_y(\theta)x_{wi} + D \tag{3.5}$$

describes a point x_{wi} that is fixed in the wheel carrier in the chassis frame. When working with (3.1) all points are referred to the chassis frame. Equations (3.1) and (3.5) provide six motion constraints that determine the kinematic behaviour of the suspension system.

As shown in Figure 3.3, the left- and right-hand sides of the suspension system are coupled through a heave spring (and a parallel damper which is not shown) and a rocker assembly. The right- and left-hand rockers are mounted on torsion bars with stiffnesses K_{tr} and K_{tl} respectively. The two rockers are also coupled by an anti-roll bar that resists chassis roll. This is represented by a moment $(\alpha_R+\alpha_L)K_{arb}$ applied to each rocker, in which α_R and α_L are the right- and left-hand rocker angles. The anti-roll bar stiffness is denoted by K_{arb}. In Figure 3.3, F_{PL} and F_{PR} are the magnitudes of the right- and left-hand push rod forces, P_{6L} and P_{6R} are the push rod attachment points, P_{HL} and P_{HR} are the end points of the heave spring, R_{RPL} and R_{RPR} are the rocker pivot points and K_h is the heave spring stiffness.

In order to compute a moment balance for the rockers, we introduce the rotation matrix $R = R(\hat{\boldsymbol{n}}, \alpha)$ that describes a rotation through α around the unit vector $\hat{\boldsymbol{n}}$:

$$R(\hat{\boldsymbol{n}}, \alpha) = cos(\alpha)I + (1 - cos(\alpha))\hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^T + sin(\alpha)S(\hat{\boldsymbol{n}}) \tag{3.6}$$

in which the skew-symmetric matrix $S(\hat{\boldsymbol{n}})$ is defined by

$$S(\hat{\boldsymbol{n}}) = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix}. \tag{3.7}$$

Taking moments around the pin joint P_{RPL} gives

$$\begin{aligned} 0 \;=\; & (F_{PL}(R(\hat{\boldsymbol{n}}_{RL}, \alpha_L)(P_{6L} - P_{RPL})) \times \hat{\boldsymbol{d}}_6).\hat{\boldsymbol{n}}_{RL} \\ & + (F_{heave}(R(\hat{\boldsymbol{n}}_{RL}, \alpha_L)(P_{HL} - P_{RPL})) \times \hat{\boldsymbol{d}}_{hs}).\hat{\boldsymbol{n}}_{RL} \\ & - K_{tl}\alpha_L - K_{arb}(\alpha_L + \alpha_R) \end{aligned} \tag{3.8}$$

in which $\hat{\boldsymbol{n}}_{RL}$ is the left rocker pivot rotation axis, $\hat{\boldsymbol{d}}_6$ is a unit vector in the direction of l_6, $\hat{\boldsymbol{d}}_{hs}$ is a unit vector in the heave spring direction pointing from P_{HL} to P_{HR}, F_{heave} is the heave spring force magnitude, '.' represents the dot product and $\times$ represents the cross product.

In the same way one can take moments around P_{RPR} to obtain

$$\begin{aligned} 0 \;=\; & (F_{PR}(R(\hat{\boldsymbol{n}}_{RR}, \alpha_R)(P_{6R} - P_{RPR})) \times \hat{\boldsymbol{d}}_6).\hat{\boldsymbol{n}}_{RR} \\ & - (F_{heave}(R(\hat{\boldsymbol{n}}_{RR}, \alpha_R)(P_{HR} - P_{RPR})) \times \hat{\boldsymbol{d}}_{hs}).\hat{\boldsymbol{n}}_{RR} \\ & - K_{tr}\alpha_R - K_{arb}(\alpha_L + \alpha_R) \end{aligned} \tag{3.9}$$

in which the various term in (3.9) must be interpreted in the context of the right-hand suspension assembly. The heave spring force F_{heave} is a function of the change in its length, and is given by

$$F_{heave} = K_h(\Delta L)\Delta L \tag{3.10}$$

in which $\Delta L = |P_{HR} - P_{HL}| - L_0$ with L_0 the unloaded length. In general, the heave spring stiffness is non-linear and tends to 'harden' as it is compressed.

The equations of motion describing the suspension system will now be determined using force and moment balances with the aid of the free body diagram shown in Figure 3.4 below. We will suppose that the reaction force acting at point Q_i has magnitude R_{Li} and acts in direction $\hat{\boldsymbol{d}}_i$. It should be noted that although the reaction forces are vector-valued quantities, it is only the magnitudes of these vectors that are

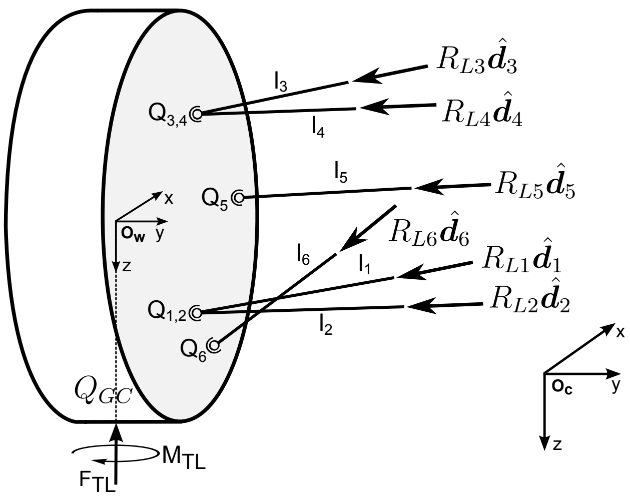

Figure 3.4: Free body diagram of the front-left wheel carrier. The reaction force magnitudes are given by R_{Li} with their directions given by $\hat{\boldsymbol{d}}_i$ for $i = 1, \cdots, 6$. The external torque acting on the wheel career is M_{TL} and F_{TL} is the external tyre force applied at the ground contact point Q_{GC}.

unknown, since their directions are determined by the kinematic constraints. The external forces acting on the wheel carrier are the tyre force F_{TL}, and the gravitational force due to the wheel carrier's mass m_L, which is given by

$$F_{Lg} = \begin{bmatrix} 0 & 0 & m_L g \end{bmatrix}^T. \tag{3.11}$$

The tyre force F_{TL} and the gravitational force F_{Lg} are expressed in the chassis reference frame, which is assumed for present purposes to be aligned with the inertial frame. Summing forces on the left-hand wheel carrier gives

$$\begin{aligned} 0 &= F_{TL} + F_{Lg} + R_{L1}\hat{\boldsymbol{d}}_1 + R_{L2}\hat{\boldsymbol{d}}_2 + R_{L3}\hat{\boldsymbol{d}}_3 + \\ &\quad R_{L4}\hat{\boldsymbol{d}}_4 + R_{L5}\hat{\boldsymbol{d}}_5 + F_{PL}\hat{\boldsymbol{d}}_6 \end{aligned} \tag{3.12}$$

in which $R_{L6} = F_{PL}$ is the force acting on the left-hand rocker. This equation provides three scalar equations in the six left-hand unknown reaction force magnitudes. A similar force balance on the right-hand front wheel carrier gives

$$\begin{aligned} 0 &= F_{TR} + F_{Rg} + R_{R1}\hat{\boldsymbol{d}}_1 + R_{R2}\hat{\boldsymbol{d}}_2 + R_{R3}\hat{\boldsymbol{d}}_3 + \\ &\quad R_{R4}\hat{\boldsymbol{d}}_4 + R_{R5}\hat{\boldsymbol{d}}_5 + F_{PR}\hat{\boldsymbol{d}}_6, \end{aligned} \tag{3.13}$$

where $R_{R6} = F_{PR}$ is the force acting on the right-hand suspension rocker. In this case

$$F_{Rg} = \begin{bmatrix} 0 & 0 & m_R g \end{bmatrix}^T \tag{3.14}$$

with the $\hat{\boldsymbol{d}}_i$'s interpreted in a like manner in the context of the right-hand front wheel. This equation provides another three equations in the six right-hand reaction force magnitudes.

As we will now show the remaining equations come from moment balance calculations. To begin, we define a set of vectors that point from point Q_1, which is fixed in the wheel carrier, to points $Q_i \quad i = 2, \cdots, 6$, O_W and the tyre ground contact point Q_{GC}. These vectors are given by $\boldsymbol{d}_{12}$, $\boldsymbol{d}_{13}$, $\boldsymbol{d}_{14}$, $\boldsymbol{d}_{15}$, $\boldsymbol{d}_{16}$, $\boldsymbol{d}_{1W}$ and $\boldsymbol{d}_{1GC}$ respectively. With the exception of $\boldsymbol{d}_{1GC}$, these vectors are known and fixed in the wheel carrier.

In order to find the vector $\boldsymbol{d}_{1GC}$, which represents a moving point in the wheel frame, we need to study the geometry of the ground contact point. To that end we introduce the vector $\hat{\boldsymbol{e}}_w = [0\ 1\ 0]^T$, which is fixed in the wheel carrier, and points in the direction of the wheel spindle. The wheel spindle unit vector, expressed in the car frame, is thus given by

$$\hat{\boldsymbol{e}}_{cw} = R_z(\psi) R_x(\phi) R_y(\theta) \hat{\boldsymbol{e}}_w. \tag{3.15}$$

If the pitch and roll of the car relative to the road is 'small', the unit road-normal expressed in the chassis frame can be approximated by $\hat{\boldsymbol{e}}_{rn} = [0\ 0\ 1]^T$. This means that the vector pointing from the wheel centre to the ground contact point is given by

$$\boldsymbol{r}_w = R_{fw}(F_{TL}) \hat{\boldsymbol{r}}_w \tag{3.16}$$

where $\hat{\boldsymbol{r}}_w$ is the wheel's unit road normal vector expressed in car frame defined by

$$\hat{\boldsymbol{r}}_w = -\frac{\hat{\boldsymbol{e}}_{cw} \times (\hat{\boldsymbol{e}}_{cw} \times \hat{\boldsymbol{e}}_{rn})}{\|\hat{\boldsymbol{e}}_{cw} \times (\hat{\boldsymbol{e}}_{cw} \times \hat{\boldsymbol{e}}_{rn})\|_2} \tag{3.17}$$

and hence $\boldsymbol{d}_{1GC} = \boldsymbol{r}_w + \boldsymbol{d}_{1W}$. Figure 3.5 illustrates the loaded tyre radius characteristic used here, which is given by

$$R_{fw} = 326.0 + 4.00 \times 10^{-3} F_{TL} + 4.075 \times 10^{-8} F_{TL}^2, \tag{3.18}$$

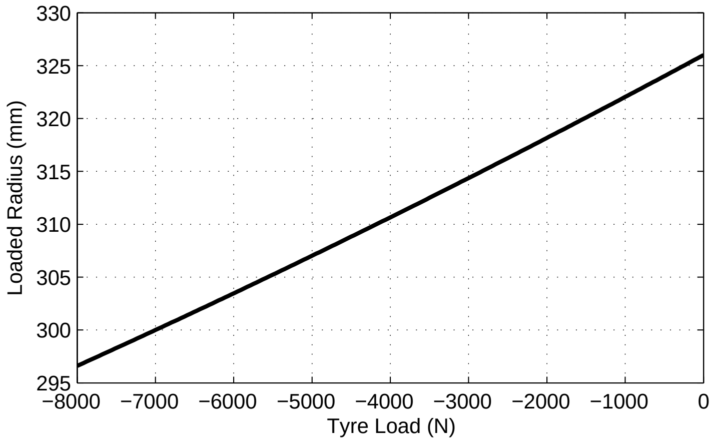

Figure 3.5: Tyre loaded radius as a function of the normal tyre load.

in which F_{TL} is the normal tyre load and is given in Newtons.

Taking moments around Q_1 in Figure 3.4 gives

$$\begin{aligned} 0 &= M_{TL} + \boldsymbol{d}_{1GC} \times F_{TL} + \boldsymbol{d}_{1W} \times F_{Lg} + R_{L2}(\hat{\boldsymbol{d}}_2 \times \boldsymbol{d}_{12}) \\ &\quad + R_{L3}(\hat{\boldsymbol{d}}_3 \times \boldsymbol{d}_{13}) + R_{L4}(\hat{\boldsymbol{d}}_4 \times \boldsymbol{d}_{14}) + R_{L5}(\hat{\boldsymbol{d}}_5 \times \boldsymbol{d}_{15}) \\ &\quad + R_{L6}(\hat{\boldsymbol{d}}_6 \times \boldsymbol{d}_{16}). \end{aligned} \tag{3.19}$$

A similar calculation for the front right wheel gives

$$\begin{aligned} 0 &= M_{TR} + \boldsymbol{d}_{1GC} \times F_{TR} + \boldsymbol{d}_{1W} \times F_{Rg} + R_{R2}(\hat{\boldsymbol{d}}_2 \times \boldsymbol{d}_{12}) \\ &\quad + R_{R3}(\hat{\boldsymbol{d}}_3 \times \boldsymbol{d}_{13}) + R_{R4}(\hat{\boldsymbol{d}}_4 \times \boldsymbol{d}_{14}) + R_{R5}(\hat{\boldsymbol{d}}_5 \times \boldsymbol{d}_{15}) \\ &\quad + R_{R6}(\hat{\boldsymbol{d}}_6 \times \boldsymbol{d}_{16}) \end{aligned} \tag{3.20}$$

in which the vectors $\hat{\boldsymbol{d}}_2, \cdots, \hat{\boldsymbol{d}}_6$, $\boldsymbol{d}_{1GC}$ and $\boldsymbol{d}_{12}, \cdots, \boldsymbol{d}_{16}$ must be interpreted in an analogous manner in the context of the right-hand suspension.

In sum, equations (3.1) are used to determine the right- and left-hand Euler angles (3.4) and the right- and left-hand wheel carrier displacements in (3.3); twelve variables in total. Equations (3.8) and (3.9) determine the two rocker angles. Equations (3.12), (3.13), (3.19) and (3.20) determine the twelve right- and left-hand push rod force magnitudes. Equations (3.1), (3.8), (3.9), (3.12), (3.13), (3.19) and (3.20) can be solved for the twenty six unknowns associated with each axle using the nonlinear equation solver *fsolve* in MATLAB$^{\text{TM}}$. Figures 3.6 and 3.7 illustrate the force-displacement behaviour of the front-left suspension system as a function of the normal tyre load; the front-right hand tyre is assumed un-loaded. To focus the attention on the suspension behaviour, the tyre is assumed to be rigid and tyre squash is ignored. It is evident from Figure 3.6 that the wheel's camber and toe angles vary very little as a result of normal load variations. These two quantities are defined by

$$\delta_{camber} = \arcsin(\hat{\boldsymbol{e}}_{cw} \cdot \hat{\boldsymbol{e}}_{rn}) \tag{3.21}$$

$$\gamma_{toe} = \arctan\left(\frac{e_{cpy}}{e_{cpx}}\right) \tag{3.22}$$

where e_{cpx} and e_{cpy} are the x- and y-axis components of $\hat{\boldsymbol{e}}_{cp}$, which is the unit vector perpendicular to the wheel spindle axis and the unit road normal vector $\hat{\boldsymbol{e}}_{rn}$, given by

$$\hat{\boldsymbol{e}}_{cp} = \frac{\hat{\boldsymbol{e}}_{cw} \times \hat{\boldsymbol{e}}_{rn}}{\|\hat{\boldsymbol{e}}_{cw} \times \hat{\boldsymbol{e}}_{rn}\|_2}. \tag{3.23}$$

Figure 3.7 shows that the suspension movement resulting from normal load variations is almost entirely in the vertical direction. As the load increases on the left tyre, the ground contact point moves upwards and slightly outwards as the wheel cambers. The conclusions relating to Figures 3.6 and 3.7 are not necessarily general ones, and relate only to the particular race car suspension geometry being studied here. The description of the nonlinear equations provided here could be alternatively derived using a multi-body modelling software.

3.2.1 Suspension Metamodels

Given its complexity, the suspension model described by Equations (3.1), (3.8), (3.9), (3.12), (3.13), (3.19) and (3.20), and the tyre squash, are replaced by a metamodel, which acts as a "surrogate" for these equations. The intention is to preserve the accuracy of the physical kinematic model, while seeking simultaneously to significantly increase its solution speed in the context of optimal control calculations. The

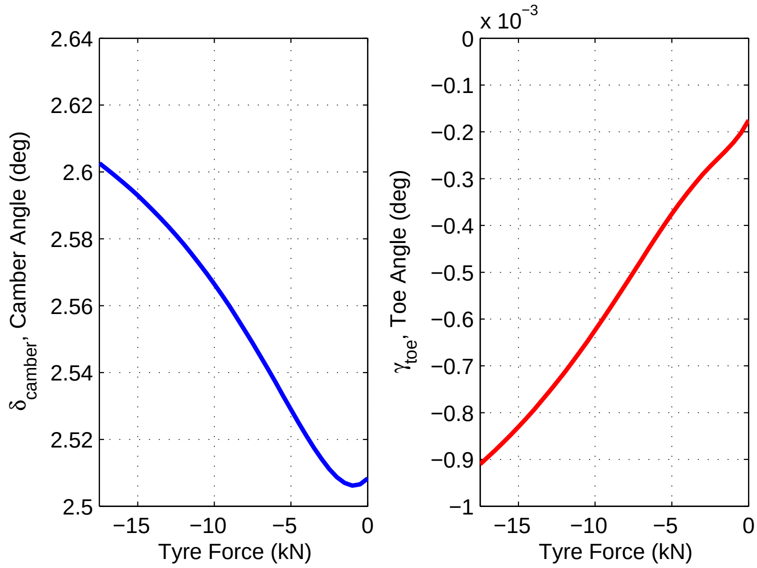

Figure 3.6: Variations in the front-left wheel carrier toe and camber angles as a function of tyre vertical load. These results are expressed in an SAE coordinate system fixed in the car body - a right-hand rule is used for angles.

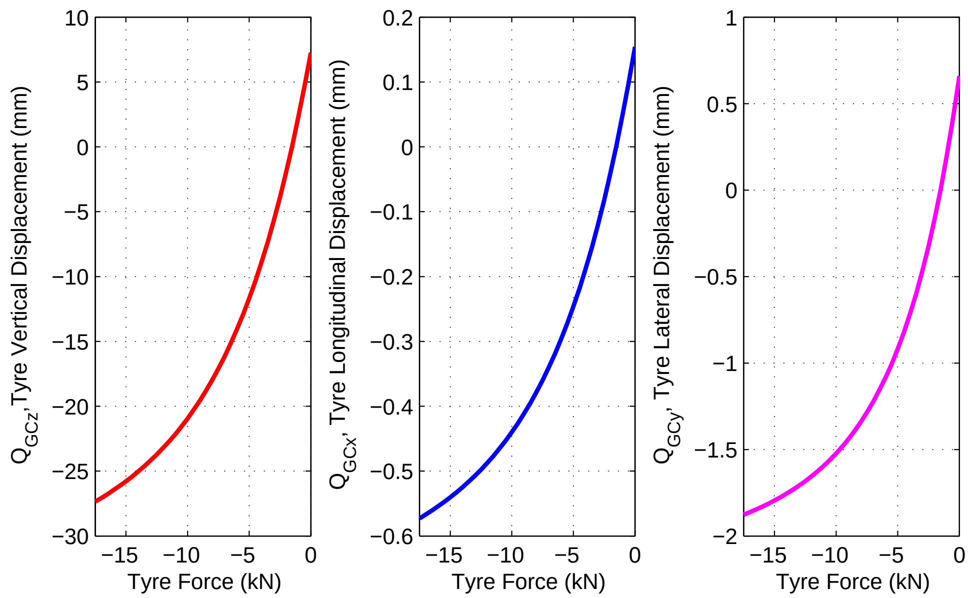

Figure 3.7: Ground contact point displacements as a function of tyre vertical load.

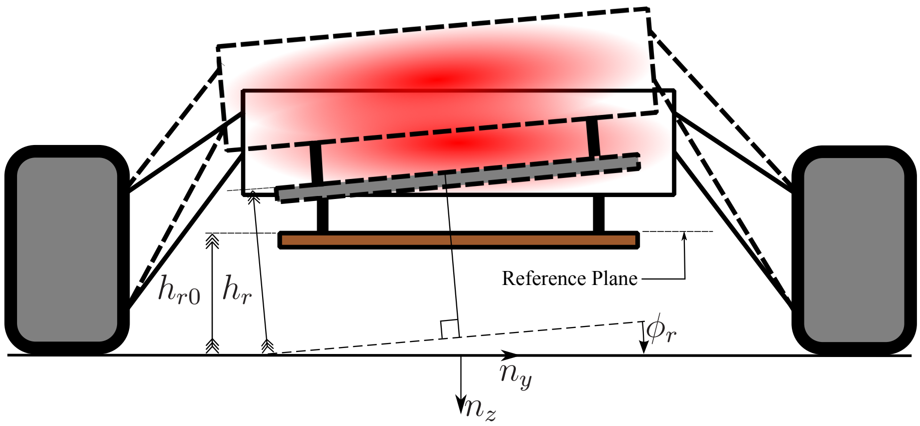

Figure 3.8: Suspension and tyre squash induced roll angle ϕ_r and ride height h_r at the rear axle; the car is driving into the page and into a right-hand bend. The rear-axle ride height in the nominal configuration is h_{r0} and is measured relative to the reference plane (fixed in the car); see Drawing 7 in [1]. The front axle is treated in the same way with ϕ_r, h_r and h_{r0} replaced with ϕ_f, h_f and h_{f0}. The ride height is expressed in the inertial coordinate system given by n_x and n_y.

interested reader is referred to [60] for a review of *metamodelling* in the context of general engineering design optimisation. Models based on approximating polynomials, splines, radial basis functions, neural networks and Kriging are all discussed.

Since the suspension deflection is dominated by the normal tyre load, the tyre moments M_{TL} and M_{TR}, the x- and y-axis components of F_{TL} and F_{TR}, and the steering angle δ can all set be to zero for the purpose the aerodynamic force studies carried out later. Although these forces and moments can become sizeable in certain manoeuvres, their influence on the vertical displacement of the wheel remains small compared to that of the normal wheel load. This is especially true in the F1 car considered, where a large part of the chassis vertical movements is due to tyre squash, which is largely dictated by the normal tyre load. Nevertheless, the influence of other components can be taken into account by augmenting the metamodel with sweeping values of the required components. In these studies we make use of a SAE coordinate system that is fixed in the road.

Figure 3.8 shows the rear axle of the car with its associated roll angle and ride height. In its nominal configuration[1] the front ride height is $h_{f0} = -14\,\text{mm}$, while the

[1]The car is stationary on the garage floor.

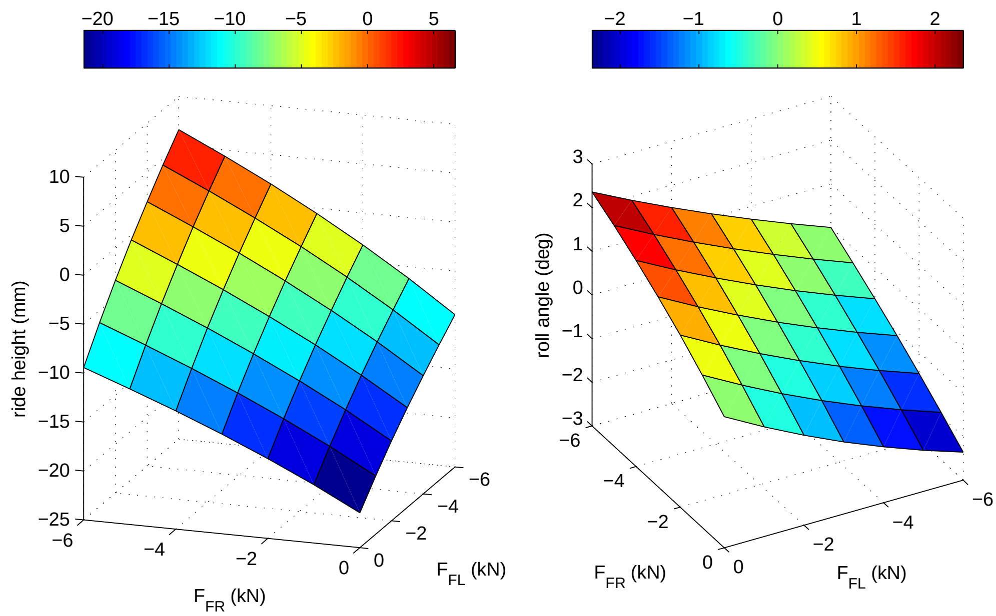

Figure 3.9: Front axle ride height and roll angle variations as a function of right- and left-hand normal tyre load changes. The left-hand plot shows the front axle ride height, while the right-hand plot shows the front axle roll angle.

rear ride height is $h_{r0} = -74.1$ mm. The nominal values of the heave spring stiffness are chosen to be 1000 kN/m and 650 kN/m for the front and rear axles respectively. In the case of the front axle, the normal tyre loads are swept from zero to -6 kN, with the resulting ride height and roll angle variations shown in Figure 3.9. If the tyre normal loads are swept from zero to -8 kN at the rear axle, the resulting ride height and roll angle variations are shown in Figure 3.10.

Figure 3.10 shows that the rear ride height varies between -88 mm and -27 mm, with the rear axle roll angle ranging over $\pm 5^o$. The front ride height varies between -24 mm and 7 mm, while the front axle roll angle varies over the $\pm 2.4^o$ range as shown in Figure 3.9. Under cornering conditions the car rolls out of the bend and so right-hand corners result in negative roll angles; this is the situation illustrated in Figure 3.8. As the tyre loads increase, the car moves towards the road (in the positive n_z-axis direction; see Figure 3.8).[2]

[2]The fact that the reference frame can pitch relative to the road means that ride heights can take on positive and negative values within the suspension's operating range.

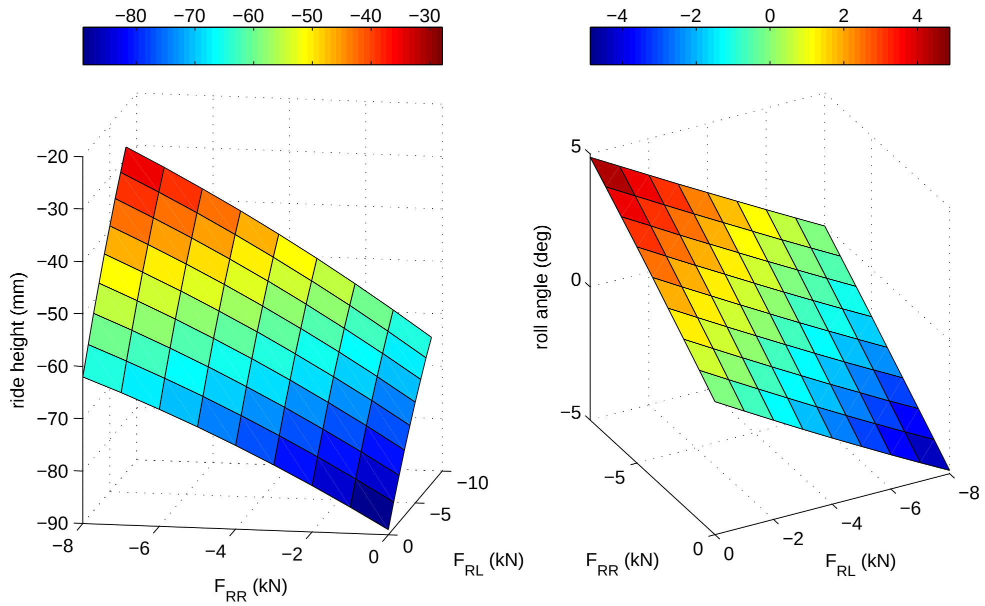

Figure 3.10: Rear axle ride height and roll angle variations as a function of right- and left-hand normal tyre load changes. The left-hand plot shows the front axle ride height, while the right-hand plot shows the front axle roll angle.

In optimal lap time simulations it will be necessary to perform these calculations many times over and so it is necessary to perform these calculations fast. To that end, we will fit multivariate polynomials to pre-computed data of the type shown in Figures 3.9 and 3.10 using the least squares algorithm described in Appendix A. The smooth nature of the ride height and roll angle data shown in these figures suggests that this data can be accurately represented by relatively low-order multivariate polynomials. Table 3.1 gives the coefficients associated with the multivariate polynomial descriptions of the rear- and front-axle ride height and roll angles.

-	1	F_l	F_r	F_l^2	$F_l F_r$	F_r^2
h_f	-21.52	-2.513	-2.513	-0.08324	0.1126	-0.08324
ϕ_f	0	0.00851	-0.00851	0.000256	0	-0.0002562
h_r	-89.04	-4.1305	-4.1305	-0.09450	0.1119	-0.09450
ϕ_r	0	0.01118	-0.01118	6.134×10^{-05}	0	-6.134×10^{-05}

Table 3.1: Multivariate polynomial coefficients of the front- and rear-axle ride height (in mm) and roll angle (in rad) in *Graded Reverse Lexicographic Order*. The left- and right-wheel normal loads F_l and F_r, respectively, are given in (kN).

-	1	F_l	F_r	K_h	F_l^2
h_f	-2.53e1	-3.70e0	-3.70e0	-1.19e-3	-1.07e-1
ϕ_f	6.56e-16	8.57e-3	-8.57e-3	-6.2e-19	2.88e-4
h_r	-9.07e1	-6.90e0	-6.90e0	-1.83e-2	-9.71e-2
ϕ_r	-3.95e-16	1.11e-2	-1.11e-2	7.2e-19	6.88e-5
-	F_lF_r	F_lK_h	F_r^2	F_rK_h	K_h^2
h_f	1.22e-1	9.77e-4	-1.07e-1	9.77e-4	3.13e-6
ϕ_f	6.49e-18	2.24e-7	-2.88e-4	-2.24e-7	1.8e-22
h_r	9.43e-2	4.04e-3	-9.71e-2	4.04e-3	3.01e-5
ϕ_r	1.91e-18	2.28e-7	-6.88e-5	-2.28e-7	-3.5e-22

Table 3.2: Multivariate polynomial coefficients of the front- and rear-axle ride height (in mm) and roll angle (in rad) in *Graded Reverse Lexicographic Order* with heave spring stiffness included as an input. The left- and right-wheel normal loads F_l and F_r, respectively, are given in (kN) and heave spring stiffness K_h in (kN/m).

Despite the relatively low order of these polynomial descriptions, the largest rear axle ride height error across all mesh points is only 0.03 mm with a corresponding maximum roll angle error of 0.027^o. The largest front axle ride height error across all mesh points is 0.147 mm with a corresponding maximum roll angle error of 0.038^o.

In the context of racing lap time optimisation, it is often desirable to optimise the suspension parameters. One important example of such a parameter is the heave spring stiffness K_h value. Suspension optimisation can be achieved by including the required parameter in the meta-model. To this end, the ride height and roll angle of each axle can be solved for a range of varying spring stiffnesses in addition to the normal tyre forces. The solution dataset can again be fitted to a multivariate polynomial using the linear least squares algorithm in Appendix A to obtain the desired meta-model. Table 3.2 gives the coefficients of the fitted multivariate polynomials with the front and rear heave spring stiffness values included as input variables. Even though the polynomials are only of quadratic order, the maximum error across all mesh points for the front axle ride height was 0.44 mm and 0.047^o for the roll angle, with similar results obtained for the rear.

Figure 3.11 shows the front ride heights when equal normal tyre loads are applied on each wheel for a range of heave spring stiffness values. As one can expect, with

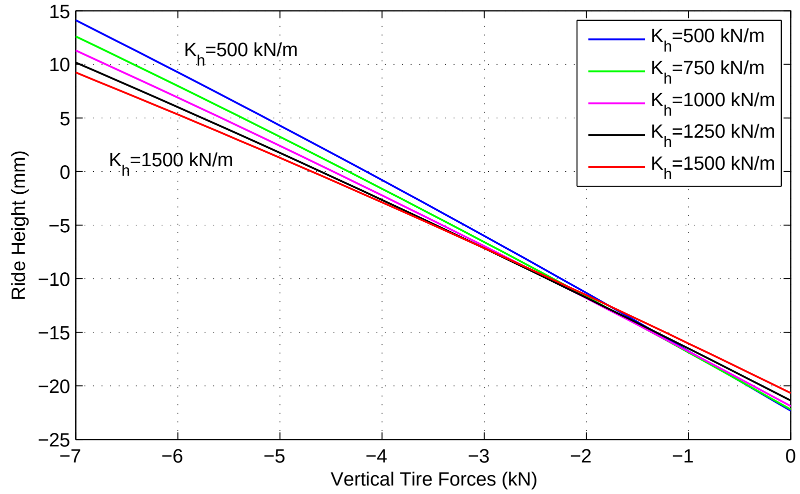

Figure 3.11: Front axle height ride for different values of heave spring stiffness as the tyre loads are increased equally.

increasing tyre load the ride height increases (in SAE coordinates) thus bringing the car closer to the road. A lower heave spring stiffness results in a higher variation in the ride height.

Figure 3.12 shows the front axle roll angle for the case where the vertical force on the left tyre is increased whilst the right tyre is kept unloaded. During a right-hand turn, the load on the left tyre is higher than the right and this causes the car to have a negative roll angle. In a high roll angle configuration, the bulk of the external forces is resisted by the anti-roll and torsion bars. The value of the heave spring has a relatively small influence on the axle roll angle since it essentially acts as a force transmitter between the two rockers. Nevertheless, a more stiff heave spring results in a larger roll angle variation as its length reduces less under a high normal wheel load, which in turn causes the opposite rocker to rotate more.

3.3 Summary

In this chapter, the suspension of a typical Formula One car in push rod configuration was modelled. This was done by considering the kinematics of the connecting rods and deriving the equations which relate the external forces acting on the tyre to the movements of the ground contact points. The developed model has a fairly high

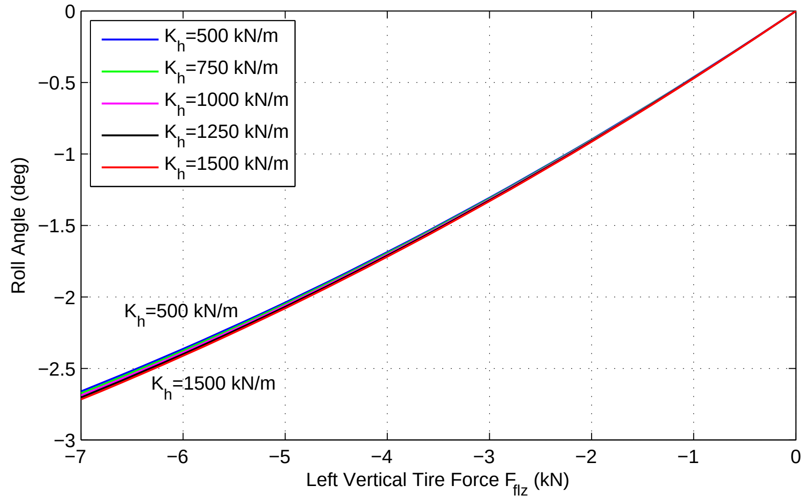

Figure 3.12: Front axle roll angle for different values of heave spring stiffness. The left tyre load is varied while the right wheel load is kept at zero.

fidelity and takes into account the suspension geometry by considering the suspension pick-up points and the length of suspension linkage bars. It also models the effects of heave springs, anti-roll and torsion bars, tyre squash and movements of tyre ground contact point as the configuration of the wheel carrier changes. This results in a relatively large set of nonlinear equations that are suited to off-line computations. In the following chapter, we aim to study the interactions of the vehicle's aerodynamics with the suspension and optimise various related quantities. Therefore, we proposed to use multi-variate polynomials to act as a surrogate for the full suspension model such that it can be employed in the dynamic lap time optimisation routines. It was shown that low order multi-variate polynomials can approximate the behaviour of stiff Formula One cars accurately. Various suspension parameters can be included in the meta-model for purpose of optimisation as required.

Chapter 4

Aero-Suspension Analysis and Optimisation

4.1 Introduction

Minimum lap time problems are of great importance to Formula One racing teams. Such problems involve calculating the trajectory along with controls which allow the car to traverse the circuit in minimum time. These problems are typically solved by replacing the driver with a perfect controller that drives the car at its absolute limits around the circuit. Using this approach, the performance of the car against parameter changes can also be studied and optimised. In some races, even a millisecond a lap can mean a different finishing grid position and hence the racing teams try to optimise every aspect of the car to suit the circuit as best as possible. The first attempt to tackle problems of this type appears to be due to Mercedes-Benz race analysis department in 1950s [61]. The track was dissected into straights and constant radius corners and optimum speeds together with required gears were calculated using steady-state methods to determine the optimal lap times. The results from these simulations were then discussed with the racing drivers to provide them with guidance on possible ways to improve their manoeuvre times.

Quasi-steady-state (QSS) methods [62] have been widely used in racing teams for many years for track-side optimisation, where fast run-times are required. To evaluate the vehicle's performance, QSS methods rely on using the so-called g-g diagrams, which represent the boundaries of maximum lateral and longitudinal accelerations at each specific speed. The positive longitudinal acceleration limit is largely dictated by the engine power, while the braking and cornering accelerations are limited by the available tyre grip. Although QSS methods have proven to be a very useful tool in lap time analysis, they all have an inherent limitation; they assume the lap can be broken into a series of segments where the longitudinal and lateral accelerations and the yaw rate are assumed constant. This makes them unsuitable for studying vehicle transient behaviour. Another pitfall of such methods is that the racing line is typically required a priori. In order to alleviate these issues, an optimal control formulation can be used.

As mentioned in Chapter 2, there are two general methods of solving optimal control problems numerically, direct and indirect methods. An indirect collocation approach was taken in [63], where manoeuvrability of motorcycles was studied. The resulting two point boundary value problem (TPBVP) from the first-order necessary conditions was solved using a relaxation method, SOLVDE, from [64]. In this method, the TPBVP is discretised and turned into a set of finite-difference equations, which can then solved using a root finding algorithm. In this paper, vehicle position and orientation were described using curvilinear coordinates. This has the advantage of formulating the track boundary violation constraint in a more convenient manner as well as reducing the size of problem variables by one by using a distance dependent description.

In [34] symbolic manipulations tools were used to obtain the optimality conditions for large multi-body systems, whilst exploiting the structure of differential algebraic equations. A modified Newton Affine Invariant scheme [35] was then used to solve the discretised TPBVP. The minimum lap time problem for a motorcycle on Adria raceway was solved using this method. Minimum time cornering manoeuvres of cars were studied using this methodology for different road surfaces and transmission layouts in [65].

One of the downsides of the indirect methods is the need to derive the necessary conditions for optimality. Even for simple problems, this requires a user with at least some knowledge of optimal control theory. Symbolic manipulation software can be employed to lift much of this burden, but for larger problems with multiple phases even this can become error prone and tedious. The path constraints are usually needed to be handled using penalty functions, or more complication will be added to the necessary conditions. Furthermore, in process of deriving the optimality conditions, the number of dynamic equations is doubled due to the addition of Lagrange multipliers associated with the state dynamics vector. Finally, for certain problems such as ones with look-up tables, these methods are not readily available [20]. To avoid these problems direct methods can be applied.

Direct methods aim to discretise the problem and then optimise [20] as opposed to indirect methods where optimality conditions are first derived and then solved by discretisation. Direct methods involve approximating the infinite-dimensional problem with a finite-dimensional problem and then applying nonlinear programming (NLP) techniques to solve the optimal control problem. Such methods are typically divided into two general categories, shooting and collocation.

In [66], a two-dimensional, three degrees of freedom car model with nonlinear tyres was used to study time-optimal lane change manoeuvres. Front wheel steering rate and longitudinal force were the controls of the problem. The car was constrained to stay within the road boundary by adding a penalty to the cost functional. A Gradient method [67] was then used to solve the optimal control problem for a front- and a rear- wheeled drive car. The optimal strategy was shown to be significantly different for each vehicle configuration.

A direct multiple shooting method [68] was used to solve the minimum lap time problem for a race car in [69]. In such methods, the time interval is broken into subintervals and then the direct shooting is applied for each segment. To ensure that the segments join at the boundaries, continuity constraints are enforced. The vehicle model in [69] was represented as a single point mass with seven degrees of freedom. The model assumed a rigid chassis with yaw, lateral and longitudinal freedoms. The wheels, rigidly attached to the body, were allowed to spin around their own axis relative to the chassis frame thus adding four more degrees of freedom to the system. Constant drag and lift coefficients were used to calculate the aerodynamic forces. Tyre forces were calculated using the Magic Formula tyre model from [70]. The vertical load acting on each wheel was evaluated by summing the contributions from the static weight distribution, downforce, and longitudinal and lateral load transfers.

The model used steering angle for lateral control and throttle aperture/braking for the longitudinal control. The torque available to each wheel was calculated by considering gear ratios and using an engine map, which took throttle and engine rotational velocity as inputs. The vehicle model also included a speed sensitive limited-slip differential. Under braking, the total negative torque applied was obtained by the product of the longitudinal control with the maximum braking torque available. This total torque was then distributed between front and rear axles according to a constant ratio. The front wheels were set to share the torque equally, whereas the rear wheels also took into account the effect of the limited-slip differential.

The track position of the car was described using curvilinear coordinates, as proposed in [63]. To obtain an initial guess for the racing line, the map of the circuit was loaded from a pre-stored data file that contained the track information such as the distances from the start/finish line, curvatures and centre line tangent angles. Initially, the racing line was reconstructed from on-board telemetry data using two different algorithms. However, it was found that this racing line did not usually match the loaded map. The initial guess for the racing line was therefore obtained by drawing it over the map via a graphical user interface (GUI) and then fitting a spline across the selected points. The velocities were initialised from a complete lap of measured data on a real racing car.

The resulting NLP problem from direct multiple shooting was solved using SNOPT [21]. Automatic Differentiation (AD) [23] was used to provide SNOPT with accurate first-order derivatives at low computational costs. The control discretisation grid points were chosen to be closer together when the rate of change of controls was large, for example during braking or cornering, to improve solution accuracy. Conversely, lower density grids were used when the rate of change of controls was small to decrease the problem size. Finally, double lane change manoeuvres were set up to study the influence of yaw inertia on vehicle performance. Minimum lap times were calculated for Barcelona and Suzuka circuits and their sensitivity to vehicle mass and weight distribution was evaluated. The run-time for a full lap of Barcelona circuit on a Sun Spark station with six CPUs running at 336 MHz was around 28 hours for the case that converged and 60 hours for cases that did not satisfy the tolerance within the maximum major iteration limit of 200.

Building on this work, [71] used a more elaborate car model to study a set of short manoeuvres and full laps. The feasible sequential quadratic programming (FSQP) method, which is based on SQP, was used to solve the NLP. FSQP generates iterates that remain feasible with respect to the inequality constraints, thus making it similar to interior point methods in that regard. FSQP was found to solve the problem more robustly when stability constraints were added to help the optimiser avoid objective function evaluations inside the highly non-linear regions. A sprung vehicle model was also considered. The suspension consisted of four pairs of rigid links, which connected the wheel carrier to the chassis. Each link pair had one rotation degree of freedom, hence giving the suspension four degrees of freedom in total. Each axle was coupled with linear heave and roll springs and dampers. The effects of the suspension stiffness and damping rates on the vehicle stability and performance over a short cornering manoeuvre with and without kerbs were studied. It was stated that it is difficult to obtain reliable results for kerbs above a certain height. The influence of suspension movements on aerodynamics was ignored. A thermodynamic tyre model was also developed and its effects on lap times were investigated. A full lap time of the Jerez circuit for the basic vehicle model with 10 meters control spacing on a 2.4 GHz single core processor took approximately 8 hours.

A direct collocation method based on the trapezoidal rule was used to calculate minimum lap times for the Barcelona circuit in [2]. The track was modelled using curvilinear coordinates. Track curvatures were reconstructed from global positioning system (GPS) data by solving an auxiliary optimal control problem to avoid issues associated with integration drift error and ensure track closure. The optimal control problem was transcribed into an NLP using MATLAB based ICLOCS [72] toolbox and then solved using the NLP solver IPOPT, which is based on the interior point method.

Analytical derivatives were supplied to IPOPT using symbolic manipulation software. A set of parameter optimisation problems including mass centre position, location of aerodynamic centre of pressure, roll balance ratio and differential viscosity constant was also considered. A full lap of Barcelona circuit was calculated in 15 minutes on a quad-core 3.4 GHz desktop station for states and controls sampling distance of 2 meters [73].

In a later paper [7], the minimum lap time problem for cars with energy recovery systems (ERS) was studied. GPOPS-II [42] software, which implements a variable-order adaptive pseudospectral method with LGR collocation points [44] was used to solve the optimal control problem. To obtain the derivatives required by the NLP solver, complex-step differentiation was used to avoid subtractive cancellation errors associated with finite-differencing and increase approximation accuracy. It was shown that the less powerful turbo-engined car with ERS could achieve comparable lap times with only a two-third of the fuel required by the 2013 and earlier Formula One cars.

Three-dimensional track modelling and identification of roads was considered in [74]. The track was modelled as a ribbon using an augmented version of Frenet-Serret equations to represent track widths. A twist angle was then introduced to allow representation of the ribbon camber vector. As with the 2D case, to avoid issues with noise and drift errors, the problem was formulated and solved as an auxiliary optimal control. In a 3D formulation, the car has a gravitational vector that varies in the body frame and centripetal forces can augment or undermine the car's downforce when travelling through dips and peaks. Road camber affects the side forces, which are particularly important around the corners, and gyroscopic moments can impact the load transfers. A minimum lap time calculation was carried out for the Barcelona track and it was demonstrated that using a 3D track can lead to different optimal racing lines and vehicle maximum speeds. The 3D model was shown to result in a lap time that is 0.67 s slower.

Suspensions allow the car to heave, pitch and roll relative to the road. They play a crucial role in the performance of Formula One cars, where aerodynamic characteristics are sensitive to attitude and height of the chassis. Including a complete suspension model in the optimal control problem will result in impractical solution times, as fine meshes will be required to capture the fast dynamics of the stiff suspension. Furthermore, this approach will increase the number of states, which in turn will considerably inflate the number of decision variables in the NLP problem. Therefore, the suspension meta-model that was described in Chapter 3 will be employed here to study the interactions between the suspension and the aerodynamics. The meta-model is used to map the normal tyre loads to suspension configuration, which is then used to determine the aerodynamic forces acting on the vehicle. A full lap of Barcelona is simulated using a pseudospectral collocation method and a set of parameter optimisation studies are carried out to find the suspension set-ups and aerodynamics characteristics that result in optimal lap times.

The rest of this chapter is organised a follows. The car and track models appear in Sections 4.2. Vehicle-related modelling features that have been used before are dealt with briefly with appropriate references provided and the new model upgrades are described. The optimal control problem is summarised in Section 4.3. The results appear in Section 4.4 with the conclusions given in Section 4.5.

4.2 Car and Track Model

The car model used in this chapter is an extension of one that has been used elsewhere [2, 4, 7], and so will be described only briefly. Relative to prior art models, the key upgrades relate to the introduction of a suspension system, and the introduction of an aerodynamic model that is responsive to the car's disposition relative to the road and its direction of travel. The three degree of freedom description, tyre and wheel torque distribution models remain the same. The load transfer equations reflect the inclusion of a suspension model. The positivity of vertical tyre forces are ensured more elegantly by having hard bounds on them in the optimal control calculations rather than introducing extra constraints.

In order to study suspension-related influences, one might consider augmenting existing vehicle models with dynamic suspension models, and main-body heave, roll

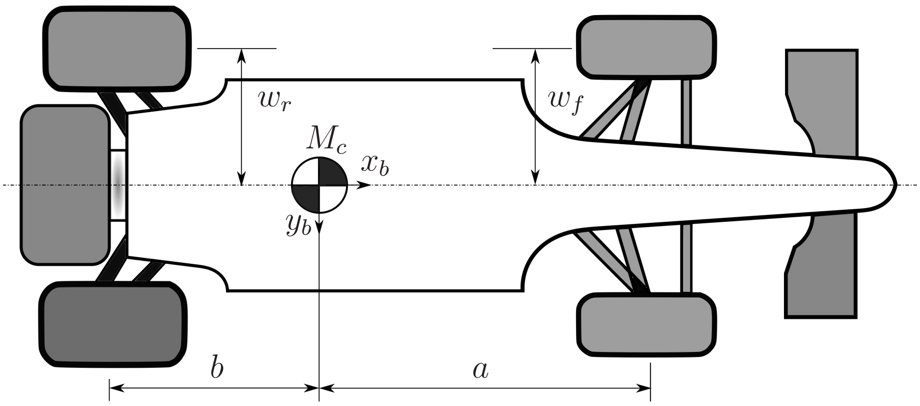

Figure 4.1: Plan view of a Formula One car with its basic geometric parameters. The car's mass centre is at M_c, with the body-fixed axes x_b and y_b in the ground plane beneath M_c.

and pitch freedoms, which would inform the aerodynamic maps given in Table 4.1 with the appropriate ride height and body-angle information. A model of this type would be complex, since it would have to include three new main-body freedoms and the multi-link suspension system described in Section 3.2. Another complication, from an optimal control perspective, related to the fact that the suspension dynamics would contain 'fast' dynamics requiring a fine integration mesh. Rather than pursuing this line of investigation, we will use the quasi-static meta model given in Table 3.1 to compute the front- and rear-axle ride heights, and the front and rear vehicular roll angles. These suspension-related variables are then fed directly into the aerodynamic map (see Figure 4.4). If one wishes to model the car body as 'rigid', the front- and rear-axle roll angles can be constrained in the optimal control problem to be equal. The remaining modelling elements are now described briefly, with further details widely available in the literature.

As has been explained elsewhere [2, 7, 74], the track and car kinematics are modelled using ideas from classical differential geometry. The car model is standard and is based on a rigid-body representation of a chassis with longitudinal, lateral and yaw freedoms. We will use the tyre description given in [71] in combination with the upgraded aerodynamic model given in Table 4.1. The important geometric modelling quantities are shown in plan view in Figure 4.1.

4.2.1 Track Model

We will model the track using a curvilinear coordinate system that follows the vehicle using the track *spine* (centre line) position as the curvilinear abscissa [63]. Referring to Figure 4.2, we describe the location of the mass centre of the vehicle in terms of the curvilinear abscissa $s(t)$ and the vector $\boldsymbol{n}(s(t))$. The former quantity defines the distance travelled along the track centre line, while the latter gives the position of the vehicle's mass centre in a direction perpendicular to the track spine tangent vector $\boldsymbol{t}(s(t))$. It is assumed that the travelled distance $s(t)$ is an increasing function of time, and that 'time' and 'distance' are alternate independent variables. The standard dot notation will be used to signify derivatives with respect to time. At any point s, the track's radius of curvature is given by $\mathcal{R}$. The track spine tangent vector $\boldsymbol{t}$ will be described in terms of the track orientation angle θ, with the track's (possibly unequal) half-widths given by $\mathcal{N}_r$ and $\mathcal{N}_l$. The yaw angle of the vehicle is given by ψ and the angle between the vehicle and the track by ξ; $\psi = \theta + \xi$. In this coordinate system constraints on the track width are easily expressed in terms of constraints on the magnitude of $\boldsymbol{n}$. The car and track kinematics have been extended to three dimensions in [4] and [74].

It follows by routine calculation [2] that

$$\dot{s} = \frac{u\cos\xi - v\sin\xi}{1 - n\mathcal{C}}, \tag{4.1}$$

in which u and v are the longitudinal and lateral components of the car's velocity and $\mathcal{C}$ is the track curvature. This equation shows how the car 'drags' s along the track's spine. The rate of change of n is given by

$$\dot{n} = u\sin\xi + v\cos\xi. \tag{4.2}$$

Differentiating $\psi = \xi + \theta$ with respect to time results in

$$\dot{\xi} = \dot{\psi} - \mathcal{C}\dot{s}. \tag{4.3}$$

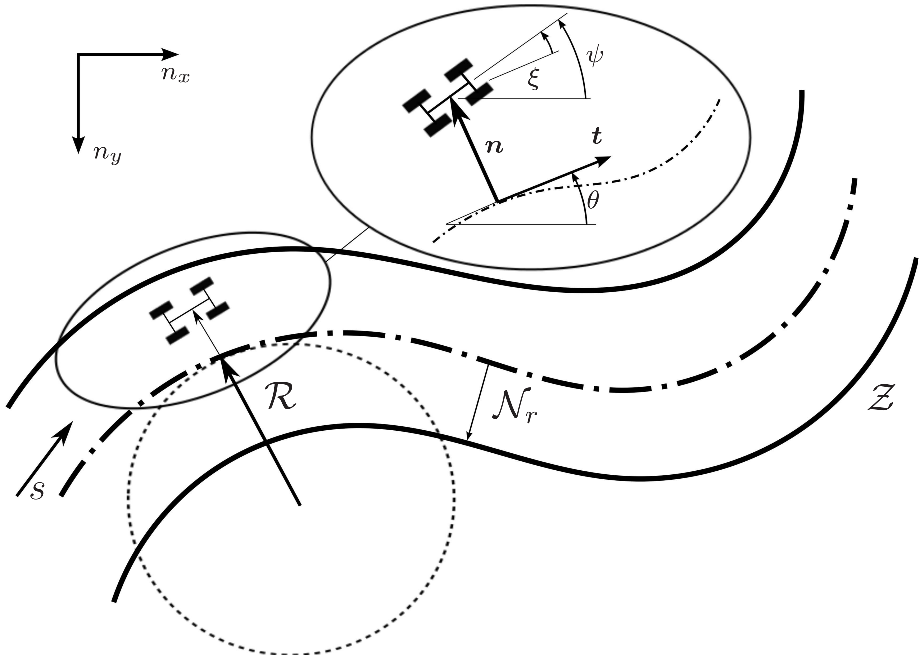

Figure 4.2: Curvilinear-coordinate-based description of a track segment $\mathcal{Z}$. The independent variable s represents the elapsed centre-line distance travelled, with $\mathcal{R}(s)$ the radius of curvature and $\mathcal{N}_r(s)$ is the the track's right-hand half-width; n_x and n_y represent an inertial reference frame [2].

4.2.1.1 Change of Independent Variable

The 'distance travelled', rather than time, will be used as the independent variable. This has the advantage of maintaining an explicit connection with the track position, as well as reducing (by one) the requisite number of state variables. Suppose

$$dt = \frac{dt}{ds}ds = S_f(s)ds,$$

where S_f comes from (4.1) as follows

$$S_f = \left(\frac{ds}{dt}\right)^{-1} = \frac{1 - n\mathcal{C}}{u\cos\xi - v\sin\xi}. \tag{4.4}$$

The quantity S_f is the reciprocal of the component of the vehicle velocity in the track-tangent direction (on the centre line at s). There follows

$$\frac{dn}{ds} = S_f\left(u\sin\xi + v\cos\xi\right) \tag{4.5}$$

from (4.2), and

$$\frac{d\xi}{ds} = S_f\omega - \mathcal{C} \tag{4.6}$$

from (4.3); $\omega = \dot{\psi}$ is the vehicle yaw rate.

4.2.2 Car Model

Each tyre produces longitudinal and lateral forces that are responsive to the tyres' slip [75]. These forces together with the steer and yaw angle definitions are illustrated in Figure 4.3.

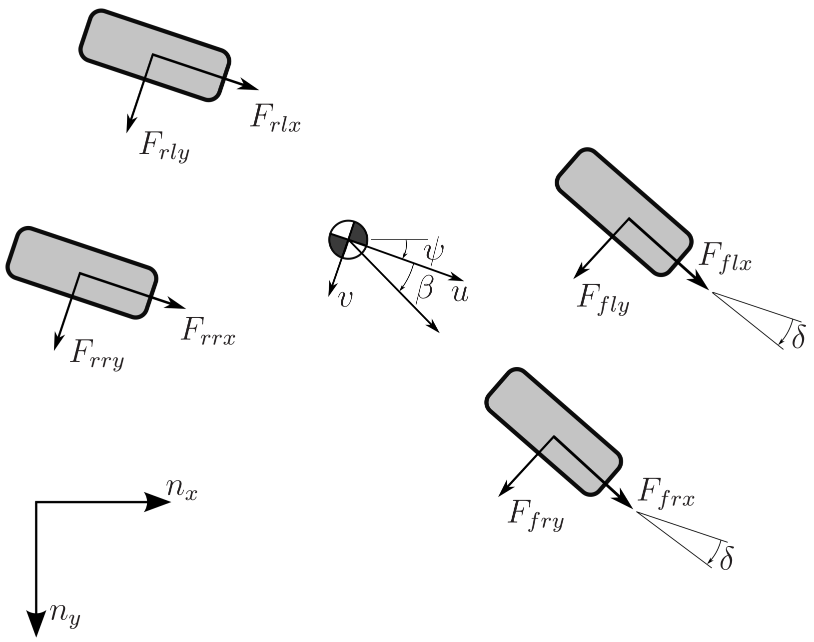

Figure 4.3: Tyre force system. The car's yaw angle is given by ψ relative to the inertial reference frame n_x and n_y. The body side-slip angle is given by $\beta = \tan^{-1}\left(\frac{v}{u}\right)$, where u and v are the longitudinal and lateral components, respectively, of the car's mass centre velocity.

$$
\begin{aligned}
M\frac{d}{dt}u(t) &= M\omega v + F_x \\
M\frac{d}{dt}v(t) &= -M\omega u + F_y \\
I_z\frac{d}{dt}\omega(t) &= a\left(\cos\delta(F_{fr\,y} + F_{fl\,y}) + \sin\delta(F_{fr\,x} + F_{fl\,x})\right) \\
&\quad +w_f\left(\sin\delta F_{fr\,y} - \cos\delta F_{fr\,x}\right) - w_r F_{rr\,x} \\
&\quad +w_f\left(\cos\delta F_{fl\,x} - \sin\delta F_{fl\,y}\right) + w_r F_{rl\,x} - b\left(F_{rr\,y} + F_{rl\,y}\right), \qquad (4.7)
\end{aligned}
$$

in which F_x and F_y are the longitudinal and lateral forces, respectively, acting on the car. These forces are given by

$$
\begin{aligned}
F_x &= \cos\delta(F_{frx}+F_{flx}) - \sin\delta(F_{fry}+F_{fly}) + (F_{rrx}+F_{rlx}) + F_{ax} && (4.8)\\
F_y &= \cos\delta(F_{fry}+F_{fly}) + \sin\delta(F_{frx}+F_{flx}) + (F_{rry}+F_{rly}) && (4.9)
\end{aligned}
$$

in which F_{ax} is the aerodynamic drag force. These equations can be expressed in terms of the independent variable s as follows:

$$
\begin{aligned}
\frac{du}{ds} &= S_f(s)\dot{u} && (4.10)\\
\frac{dv}{ds} &= S_f(s)\dot{v} && (4.11)\\
\frac{d\omega}{ds} &= S_f(s)\dot{\omega}. && (4.12)
\end{aligned}
$$

4.2.3 Tyre Forces

The tyre forces have normal, longitudinal and lateral components that act on the vehicle's chassis at the tyre ground contact points and react on the inertial frame. The rear-wheel tyre forces are expressed in the vehicle's body-fixed reference frame, while the front tyre forces are expressed in a steered reference frame; refer again to Figure 4.3. In each case these forces are a function of the normal load and the tyre's longitudinal slip coefficient κ and a slip angle α [75]. Following standard conventions we use

$$
\begin{aligned}
\kappa &= -\left(1+\frac{R\omega_w}{u_w}\right), && (4.13)\\
\tan\alpha &= -\frac{v_w}{u_w}, && (4.14)
\end{aligned}
$$

where R is the wheel radius and ω_w the wheel's spin velocity. The quantities u_w and v_w are the absolute speed components of the wheel centre in a wheel-fixed coordinate system. The four tyre slip angles are given by

$$
\begin{aligned}
\alpha_{rr} &= \arctan\left(\frac{v-\dot{\psi}b}{u-\dot{\psi}w_r}\right),\\
\alpha_{rl} &= \arctan\left(\frac{v-\dot{\psi}b}{u+\dot{\psi}w_r}\right),\\
\alpha_{fr} &= \arctan\left(\frac{\sin\delta(\dot{\psi}w_f-u)+\cos\delta(\dot{\psi}a+v)}{\cos\delta(u-\dot{\psi}w_f)+\sin\delta(\dot{\psi}a+v)}\right),\\
\alpha_{fl} &= \arctan\left(\frac{\cos\delta(\dot{\psi}a+v)-\sin\delta(\dot{\psi}w_f+u)}{\cos\delta(\dot{\psi}w_f+u)+\sin\delta(\dot{\psi}a+v)}\right).
\end{aligned}
\qquad (4.15)
$$

The complete tyre model with parameters is provided in Appendix B.

4.2.4 Load Transfer

In our optimal control codes the tyre normal forces are treated as time-varying non-positive inputs normal to the ground plane that must be constrained by the laws of mechanics. To this end we balance the forces acting on the car in the n_z direction and balance moments around the body-fixed x_b- and y_b-axes; see Figure 4.1. Balancing the vertical forces gives

$$0 = F_{rrz} + F_{rlz} + F_{frz} + F_{flz} + Mg + F_{az}^{f} + F_{az}^{r}, \tag{4.16}$$

in which the $F_{\cdot\cdot z}$'s are the vertical tyre forces for each of its four wheels, g is the acceleration due to gravity, and F_{az}^{f} and F_{az}^{r} are the front and rear, respectively, aerodynamic downforces acting on the car. Balancing moments around the car's body-fixed x_b-axis gives

$$0 = w_r(F_{rlz} - F_{rrz}) + w_f(F_{flz} - F_{frz}) + hF_y, \tag{4.17}$$

in which F_y is the lateral inertial force acting on the car's mass centre; see (4.9). Balancing moments around the car's body-fixed y_b-axis gives

$$0 = b(F_{rrz} + F_{rlz} + F_{az}^{r}) - a(F_{frz} + F_{flz} + F_{az}^{f}) + hF_x, \tag{4.18}$$

where F_x is the longitudinal inertial force acting on the car's mass centre (see (4.8)).

Equations (4.16), (4.17) and (4.18) are a set of linear equations in the four unknown normal wheel loads. A unique solution for the tyre loads is obtained by including the suspension-related roll constraint

$$\phi_f = \phi_r, \tag{4.19}$$

which ensures that the front- and rear-axle roll angles are the same; see Figure 3.8. Equation (4.19) enforces a rigid vehicular main body, which is not the same as using a suspension roll stiffness parameter [2, 4, 7].

4.2.5 Wheel Torque Distribution

In order to optimise the vehicle's performance, one needs to control the torques applied to the wheels. The braking system applies equal pressure to the brake callipers on each axle, with the braking pressures between the front and rear axles satisfying a design ratio, which is stipulated by the Formula One technical regulations [1]. The drive torques applied to the rear wheels are controlled by a differential mechanism.

4.2.5.1 Brakes

We approximate equal brake calliper pressures with equal braking torques when neither wheel on a particular axle is locked. If a wheel 'locks up', the braking torque applied to the locked wheel may be lower than that applied to the rolling wheel. For the front wheels this constraint is modelled as follows

$$0 = \max(\omega_{fr}, 0) \max(\omega_{fl}, 0)(F_{frx} - F_{flx}), \tag{4.20}$$

in which ω_{fr} and ω_{fl} are the angular velocities of the front right and front left wheel, respectively. If either road wheel 'locks up', the corresponding angular velocity will be non-positive and the braking torque constraint (4.20) becomes inactive.

4.2.5.2 Differential

The drive torque is delivered to the rear wheels through a limited-slip differential, which is modelled by

$$R(F_{lrx} - F_{rrx}) = -k_d(\omega_{lr} - \omega_{rr}), \tag{4.21}$$

in which ω_{lr} and ω_{rr} are the rear-wheel angular velocities, R is the wheel radius and k_d is a torsional damping coefficient. The special cases of an open- and a locked-differential correspond to $k_d = 0$ and k_d arbitrarily large respectively. Limited slipping occurs between these extremes.

4.2.6 Aerodynamics

The external forces acting on the car come from the tyres and from aerodynamic influences. The dynamic car model used in the optimal control calculations compute the normal tyre loads $F_{rrz} \cdots F_{flz}$, and the steering and side slip angles δ and β, respectively, for each track location on an optimal lap. As shown in Figure 4.4 these data are supplied to the suspension and aero models to produce the aerodynamic coefficients, which are then updated continuously. The front and rear axle downforce coefficients are given by C_L^f and C_L^r respectively, while the drag coefficient is C_D. The

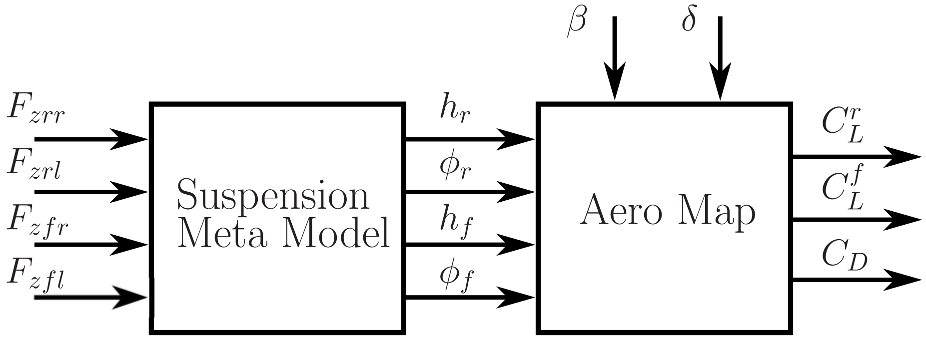

Figure 4.4: Quasi-static aero-suspension model. The dynamic model generates the normal tyre loads, and the steering and side-slip angles, while the quasi-static meta model generated the aero dynamic drag and downforce coefficients.

resulting aerodynamic downforces are computed using

$$F_{az}^f = \frac{1}{2} C_L^f \rho A u^2, \tag{4.22}$$

$$F_{az}^r = \frac{1}{2} C_L^r \rho A u^2, \tag{4.23}$$

which are applied at the centres of the front and rear axles, while the drag is given by

$$F_{ax} = -\frac{1}{2} C_D \rho A u^2, \tag{4.24}$$

which is applied at the car's mass centre. The drag has a negative sign, since it acts in the negative x-axis direction. The drag and down-force coefficients used in this study derive from track and wind tunnel measurements and are given in Table 4.1. These coefficients are responsive to the front and rear ride heights, the side-slip angle (β); see Figure 4.3, and the steer (δ) and roll (ϕ) angles. Figure 4.5 depicts the influence of front and rear ride heights on the aerodynamics downforce coefficients when β, δ and ϕ are assumed to be at a zero constant. The front downforce coefficient increases as the front axle is brought closer to the floor, whilst at a given front ride height it decreases as the rear axle sinks towards the ground. The rear downforce coefficient variation is dominated by the rear ride height with the maximum achieved around $h_r = -50\,\text{mm}$.

C_L^f	C_L^r	C_D	Input
1.2000	0.9494	0.9455	Nominal Value
10.0000	1.0000	0.1000	h_f (m)
-5.0000	-22.2222	-0.1000	h_r (m)
0	-222.2222	0	h_r^2 (m^2)
-5.1294	-10.2588	0	β^2 (rad^2)
-0.1641	-0.1641	0	δ^2 (rad^2)
-8.2070	-32.8281	0	ϕ^2 (rad^2)

Table 4.1: Downforce and drag coefficients and sensitivities.

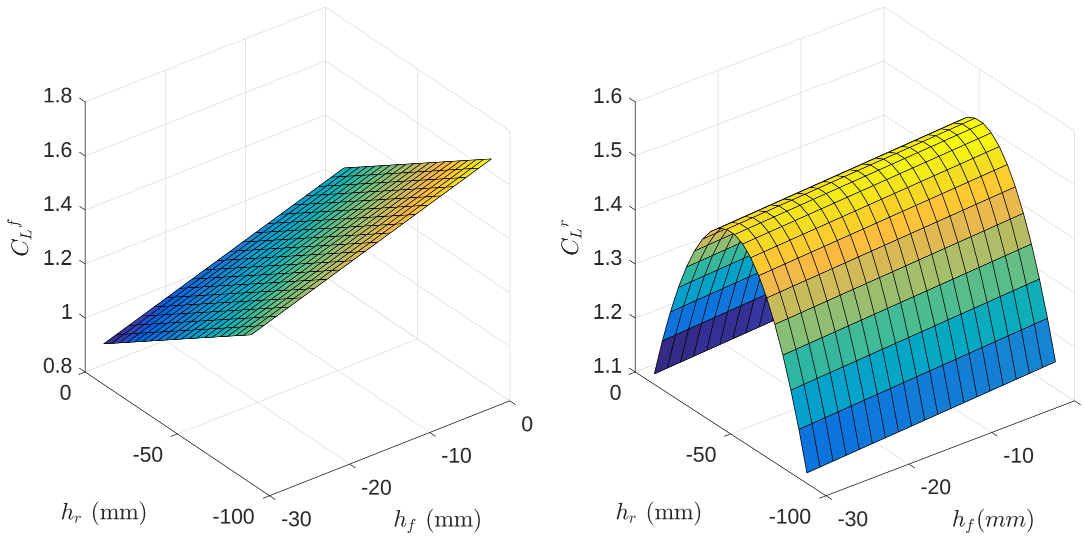

Figure 4.5: The front and rear downforce coefficient variations with respect to the front and rear ride heights in absence of side-slip, steering and roll.

4.3 Optimal Control

In this section, we will describe the optimal control problem used to calculate the minimum lap time. The states of this problem for the basic case are the vehicle perpendicular distance to the centre-line, the orientation angle with respect to the track, the longitudinal and lateral velocities, and the yaw rate.

$$\boldsymbol{x} = \begin{bmatrix} n & \xi & v & u & \omega \end{bmatrix} \tag{4.25}$$

The controls are the vehicle steering angle, the longitudinal slip and the normal load of the four wheels.

$$\boldsymbol{u} = \begin{bmatrix} \delta & \kappa_{fl} & \kappa_{fr} & \kappa_{rl} & \kappa_{rr} & F_{flz} & F_{frz} & F_{rlz} & F_{rrz} \end{bmatrix} \tag{4.26}$$

The aim is to find the controls and values of the static parameters that minimise the lap time given by

$$J = \int_{s_0}^{s_f} S_f(s)\, ds, \tag{4.27}$$

where S_f is the reciprocal of the vehicle speed given in (4.4). The s_0 and s_f are the start and end track centre-line distances. The problem is subject to state dynamics of the form

$$\frac{d\boldsymbol{x}(s)}{ds} = \boldsymbol{f}(\boldsymbol{x}(s), \boldsymbol{u}(s), \boldsymbol{p}) \tag{4.28}$$

given in equations (4.5), (4.6), (4.10), (4.11) and (4.12). The path constraints are given by equations (4.16)-(4.21). The vehicle perpendicular distance to the centre-line, n, is also bounded to not exceed the track width. The static parameters, $\boldsymbol{p}$, will be included in the problem for optimisation studies carried out later and will be explained in detail in Section 4.4. In racing, the car completes many laps of the circuit before the race is finished. Therefore, it is more useful to study the periodic lap time, where the vehicles' states are considered continuous immediately before and after the start/finish line. To that end, the following boundary condition is imposed

$$\boldsymbol{x}(s_0) - \boldsymbol{x}(s_f) = \boldsymbol{0}. \tag{4.29}$$

General purpose numerical optimal control problem solvers are now widely available. When solving new problems for the first time, it is prudent to check the solutions from multiple initial conditions, and if possible against measured data. One should also verify that parameter changes produce the trends expected, and if possible, consistency against alternative optimisation algorithms should be checked. The pseudospectral numerical optimal control solver GPOPS-II [76] based on the Legendre-Gauss-Radau (LGR) collocation scheme was used here to solve the minimum lap time problem. In this scheme the states are approximated using Lagrange polynomials as basis functions. The state dynamics are then transcribed into algebraic constraints using the derivative of the approximating polynomial. The optimal control problem is then represented by an NLP problem whose cost function is obtained by approximating the cost functional using LGR quadrature. This NLP problem is subject to the transcribed algebraic constraints and path constraints of the original problem imposed at the collocation points.

GPOPS-II uses a segment and polynomial degree adaptation scheme [44] so that error reduction can still be achieved in the presence of non-smooth problem features. The transcribed NLP problem is typically very large but sparse. The IPOPT [77] software library (based on interior point methods) was used here to solve the NLP problem. Automatic Differentiation was used to provide IPOPT with fast and accurate first and second order derivatives.

4.3.1 Scaling

Scaling can have a significant influence on the performance of optimisation algorithms. One scaling methodology is to transform variables from their original physical quantities into dimensionless variables that have desirable optimisation properties. In our work we take the length of the car to be the fundamental unit of length — after scaling the car has a dimensionless length of one. In the same way the fundamental unit of mass is the mass of the car. To scale time we choose a non-dimensional gravitational constant of unity and so a time scale of $\sqrt{g/l_0}$ is used in which l_0 is the car's length. This scaling of the basic length, mass and time variables induces a scaling on all the other physical quantities involving acceleration, velocity, force, torque, power and so on. This scaling scheme forces the NLP's decision variables into a 'more spherical' space than that associated with the original physical units.

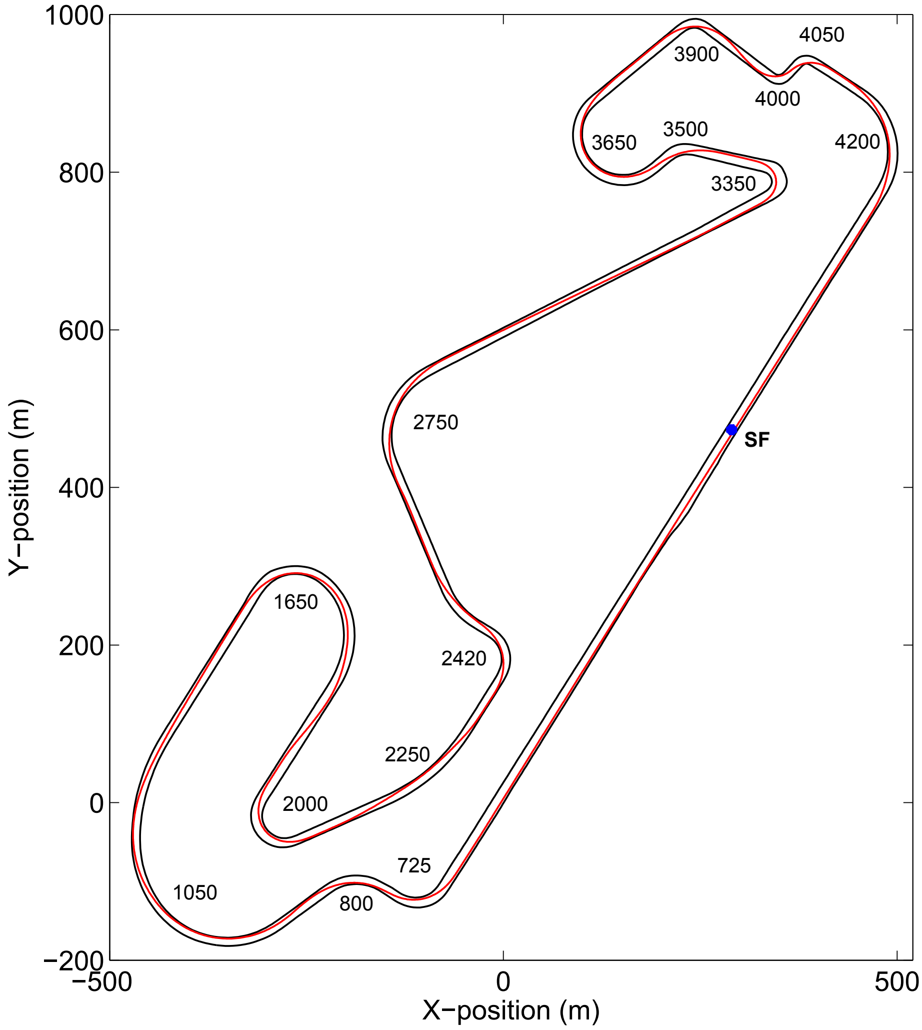

Figure 4.6: Racing line with distances from Start/Finish (SF) line in (m) for Circuit de Catalunya-Barcelona. The suspension and aerodynamic parameters are set to their base-case values.

4.4 Results

The study presented here is based on minimum-time optimal laps of a 2D model of the Circuit de Barcelona-Catalunya, which is shown in Figure 4.6. This particular track has sixteen corners with their approximate distances from the start-finish given on the same figure. The track's corner distances are useful features when seeking to understand the detailed behaviour of the car.

We will begin our study by considering the case when the car's suspension is 'frozen' in its nominal configuration ($h_f =$ -14 mm and $h_r =$ -74.1 mm), and with an aero map that has no angular sensitivities; in this case the last three rows of Table 4.1 are set to zero. The optimal lap time is computed as 83.22 s as shown in case (A) in Table 4.2. Introducing the steer angle sensitivity (row 6 in Table 4.1) causes a marginal increase in the lap time that is too small to compute reliably. As shown in row (B), when the steer and sideslip sensitivities are recognised, the lap time increases to 83.67 s.

The baseline car, corresponding to row (C) Table 4.2, includes the aerodynamic coefficients given in Table 4.1 and the suspension system described by Table 3.1. Nominal values for the car and tyre parameters can be found in Appendix C.1 and Appendix B respectively. The introduction of the suspension system means that the car has roll and pitch freedoms and all the rows in Table 4.1 come into effect. As compared with rows (A) and (B) in Table 4.2, the car will now benefit from increased downforce due to reduced ride heights at high speed, but it will also suffer the detrimental effects of roll motions on the downforce coefficients.

Figure 4.7 shows the whole-track speed profile of the baseline car, with the optimal racing line for the baseline car shown on Figure 4.6; this figure shows the trajectory of the car's mass centre. In the optimal control problem formulation used here, the car's mass centre is constrained to remain within ± 4 m of the track centre. The track width is approximately 12 m on almost all parts of the track. This constraint ensures that the car's front wheels remains on the track whatever the vehicle's orientation (relative to the track). This explains why the racing line on the turn around 3350 m appears slightly conservative, where the track widens to approximately 20 m. A constraint set that allows for a variable width track, and which ensures that each of the four wheels remain on the track, is easy to implement, but takes longer to process. The differences between the solutions with these alternative problem formulations are not visible on plotted graphs and so the more complex constraint formulation was not used.

The mesh refinement and collocation adaptation scheme is demonstrated in Figure 4.8, which focuses on the upper-right section of the track. It is evident that the optimal control solver chooses smaller segments on corner entries and small radius

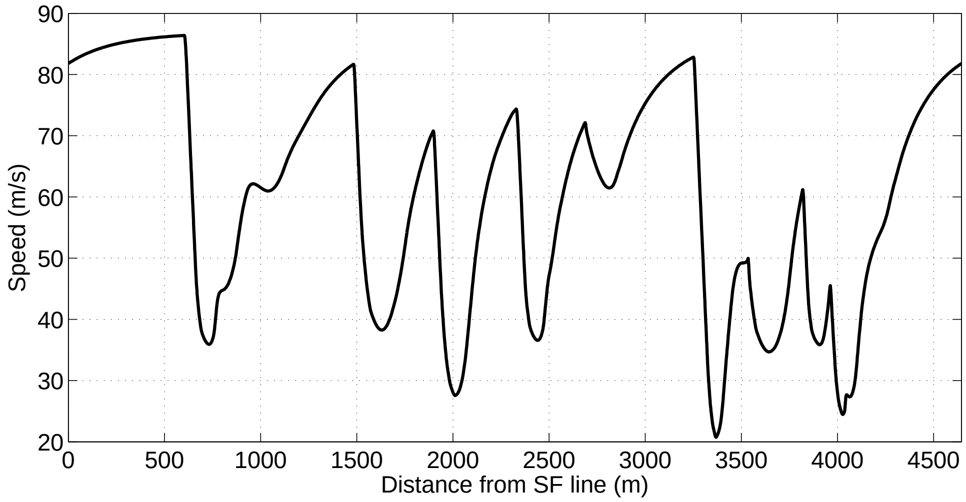

Figure 4.7: Speed for one lap of the Circuit de Barcelona. The suspension and aerodynamic parameters are set to their baseline values; see row (C) in Table 4.2.

turns, where the rate of change of controls are high. Conversely, on the straights, larger meshes and a smaller number of collocation points are required to achieve the desired error threshold.

The suspension ride heights for the baseline car are shown in Figure 4.9. One can see that as the car accelerates on the straight crossing the SF line towards the first corner (track positions 4400 m to 600 m), the air flow over the front wing causes the front of the car to move closer to the road under the influence of increased downforce. The car's increasing speed, and the increased airflow under the vehicle, also causes the car's rear axle to move closer to the road.

The high-speed straight running car operates with a very small roll angle on this section of the track as shown in Figure 4.10. As the car's speed drops into the right-hand corner, the car's suspension is unloaded and car moves away from the track surface, with the vehicle simultaneously rolling out of the corner; see the roll angle decrease at 725 m. Following on immediately after the turn, is the left-hand at 800 m, which causes the roll angle to switch from approximately -1.8^o to 1.8^o. As the car accelerates towards the third turn, the vehicle moves closer to the track with the roll angle remaining negative in the long, fast right-hand corner. The roll angle and ride heights are derived from the normal tyre loads and the suspension meta-model

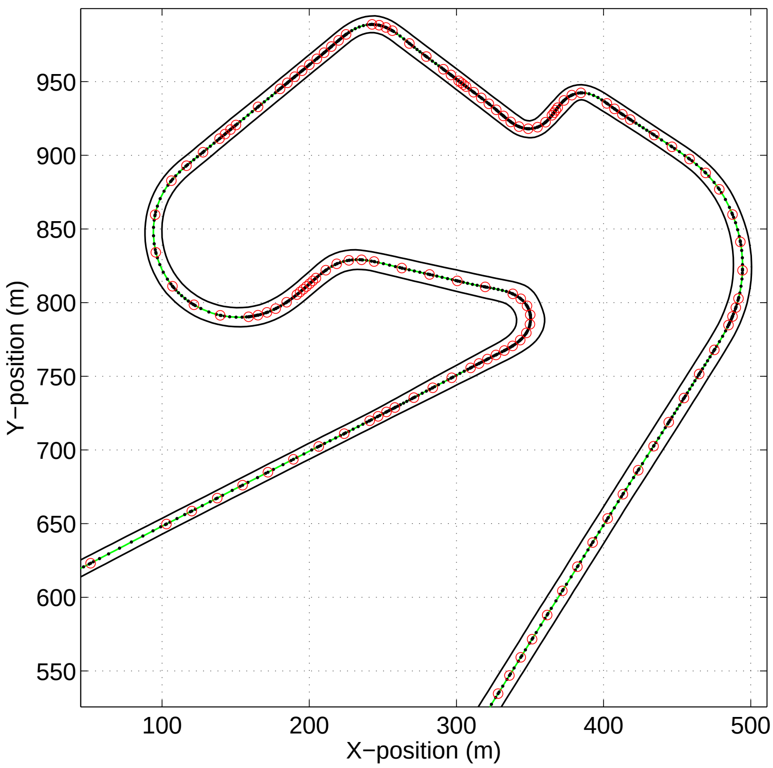

Figure 4.8: Mesh and polynomial degree adaptation for the upper-right section of the track (approximately distance 3000 m-4500 m) for the baseline case. Collocation points are shown in black dots and the mesh points in red circles. The solid green line is the centre-line.

given in Table 3.1. The variations in the suspension posture produce changes in the aerodynamic drag and downforce coefficients in accordance with Table 4.1. These changes, as well as the accompanying changes in the drag and downforces, are shown in Figure 4.11 and Figure 4.12 respectively.

It is evident that the drag Coefficient C_D is essentially constant (at approximately 0.95). To explain this, we observe that the aerodynamic model in Table 4.1 shows that the drag is linearly dependent on the difference between the front and rear ride heights; $C_D = 0.9455 + 0.1(h_f - h_r)$. Further, a pitch angle of 1^o, which is large for a F1 race car, would produce a ride height difference of about 59 mm. This difference represents a change of 0.006 to the drag coefficient. For that reason the drag coefficient is not significantly affected by suspension pitch movements, or differential

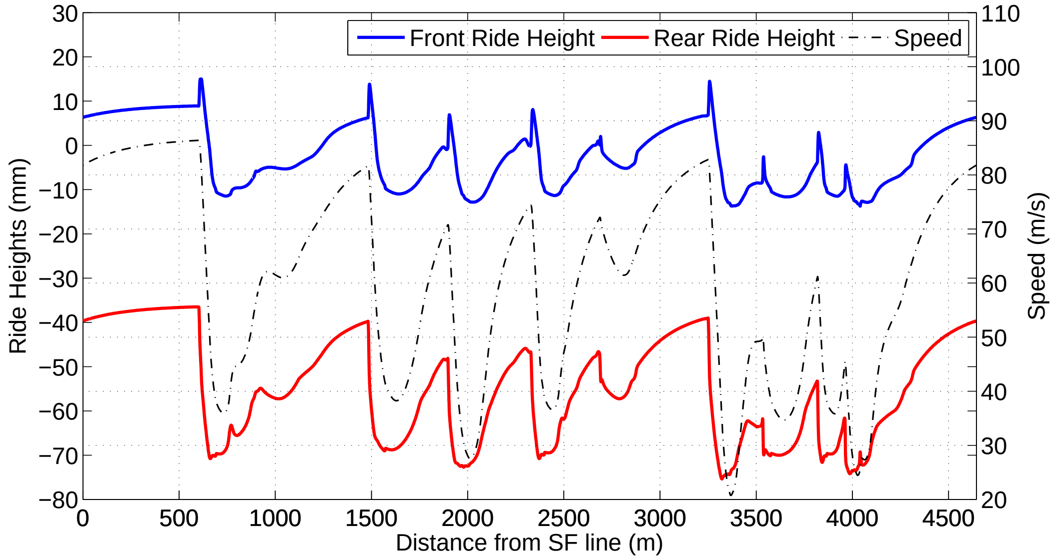

Figure 4.9: Suspension ride heights with the suspension and aerodynamic parameters set to their base-case values; see row (C) in Table 4.2. The figure shows the vehicle speed (black dot-dash), the front ride height (blue) and the rear ride height (red).

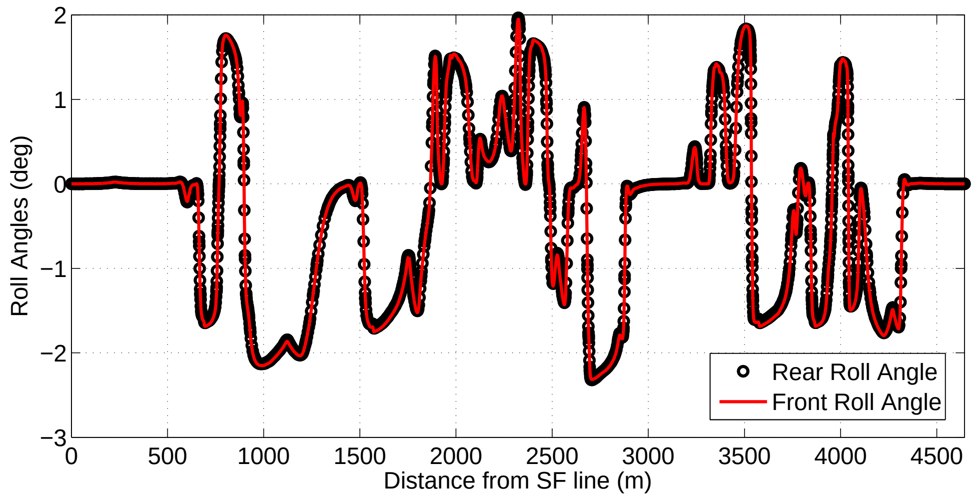

Figure 4.10: The front and rear axle roll angles for the baseline car, which are constrained to be equal in the optimisation process.

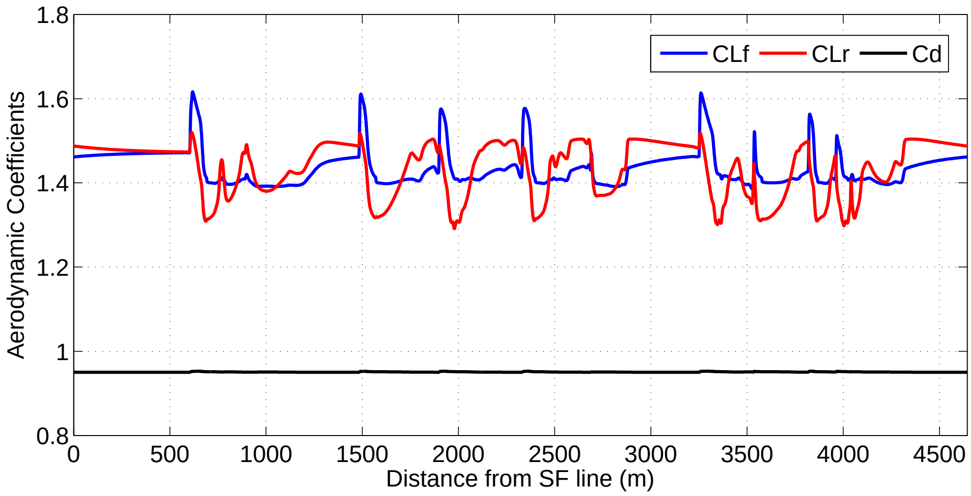

Figure 4.11: Aerodynamic coefficients with the suspension and aerodynamic parameters set to their base values; see row (C) in Table 4.2. The drag coefficient is shown in black, the front downforce coefficient in blue, and the rear downforce coefficient in red.

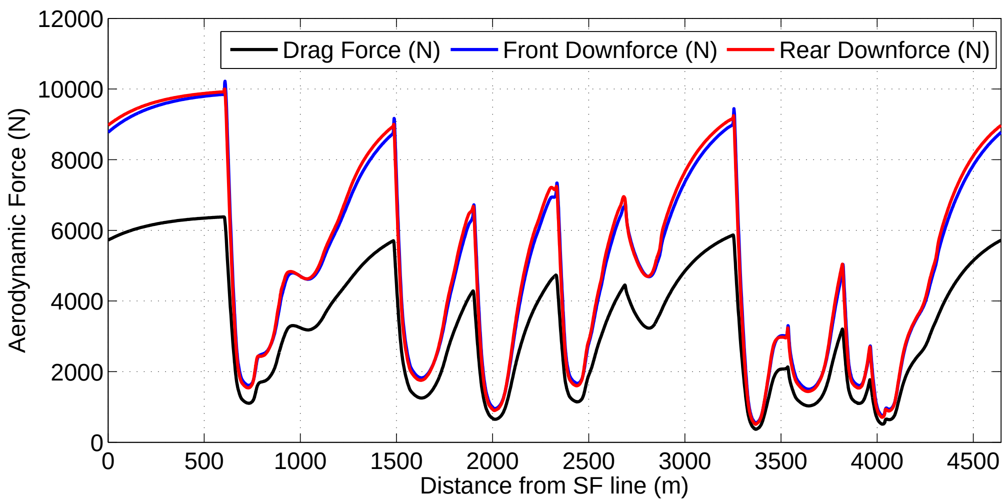

Figure 4.12: Aerodynamic forces with the suspension and aerodynamic parameters set to their base values; see row (C) in Table 4.2. The drag force is shown in black, the front downforce in blue, and the rear downforce in red.

ride height variations. In contrast the downforce coefficients C_L^f and C_L^r vary significantly between approximately 1.3 and 1.6. Figure 4.12 shows that the total downforce reaches approximately 20 kN, which is roughly 3.5 times the weight of the car.

The remainder of Table 4.2 shows the effect on the optimal lap time of a variety of suspension and aerodynamic set-up changes, some of which act in combination.

In row (D) the front and rear ride heights can be adjusted by $\pm$ 10mm and $\pm$ 30mm respectively. In order to minimise the lap time, the optimiser finds that the front ride height should be brought closer to the road by 6.0 mm, while the rear ride height should be adjusted by 12.7 mm. The front lift coefficient increases linearly with ride height (see Table 4.1) and hence more downforce can be generated by operating the front of the car closer to the road. In all the cases investigated here, the front ride height of the car was constrained to not exceed 15 mm. This prevents the car from sinking too close to the road surface, which can potentially cause excessive wear on the under-body plank of the car. This explains why the optimiser does not increase the front ride height by the maximum allowed. The rear downforce coefficient is a quadratic function of the rear ride height as shown in Figure 4.5 and by increasing h_r, it is kept closer to its optimum operating point. These ride height adjustments reduce the predicted lap time by 0.46 s.

In case (E), the front and rear heave springs were allowed to vary between $\pm$500 kN/m and $\pm$300 kN/m of their nominal values respectively. The optimiser softened both springs by the maximum allowed – apparently to drop the car closer to the road thereby gaining an aerodynamic advantage.

In case (F), both the heave spring stiffnesses and ride heights for the front and rear axles were allowed to be optimised by the same amount as in case (D) and (E). The results show that ride heights are increased by 7.6 mm and 13.8 mm at the front and rear respectively and both axles are stiffened to their maximum value. The ride heights for case (E) and (F) are shown in Figure 4.13, whilst the difference between the front downforce coefficients is highlighted in Figure 4.14. One can see that the ride height and heave spring optimised car benefits from higher front downforce coefficients for the majority of the lap. It is evident that at high speeds when downforce

Rear Axle \ Front Axle	Δh_f [a]	Δh_r	$\frac{\partial C_L}{\partial \beta^2}$	$\frac{\partial C_L}{\partial \delta^2}$	$\frac{\partial C_L}{\partial \phi^2}$	Aero Balance [b]	ΔK_{hf}	ΔK_{hr}	Lap Time
(A) Rigid suspension (no aero sensitivities) [c]	—	—	—	—	—	—	—	—	83.22
(B) Rigid suspension (with aero sensitivities)	—	—	—	—	—	—	—	—	83.67
(C) Baseline with suspension	—	—	—	—	—	—	—	—	84.50
(D) Ride height (RH) offset	6.0	12.4	—	—	—	—	—	—	84.04
(E) Heave spring stiffness (K_h)	—	—	—	—	—	—	-500	-300	84.30
(F) Heave spring and RH	7.6	13.8	—	—	—	—	500	300	84.01
(G) Aero side-slip sensitivity	—	—	-5 [d] / -10	—	—	—	—	—	84.25
(H) Aero steer sensitivity	—	—	—	-0.15 / -0.15	—	—	—	—	84.46
(I) Aero roll sensitivity	—	—	—	—	5 / 20	—	—	—	84.42
(J) All-angle sensitivity	—	—	-5 / -10	-0.15 / -0.15	5 / 20	—	—	—	84.19
(K) All-angle and RH	5.6	11.8	-5 / -10	-0.15 / -0.15	5 / 20	—	—	—	83.71
(L) All-angle, RH and K_h	7.1	13.4	-5 / -10	-0.15 / -0.15	5 / 20	—	500	300	83.66
(M) Fixed aero-balance	—	—	—	—	—	0.1926	—	—	83.68
(N) Fixed aero-balance and RH offset	9.6	1.6	—	—	—	0.2554	—	—	83.18
(O) All-angle and aero-balance	—	—	-5 / -10	-0.15 / -0.15	5 / 20	0.1874	—	—	83.33
(P) All-angle, RH, K_h and aero-balance	10.0	3.0	-5 / -10	-0.15 / -0.15	5 / 20	0.2427	500	300	82.78
(Q) Point-wise Aero Balance	—	—	—	—	—	Continuous	—	—	82.49

Table 4.2: Lap times for various optimisation cases. Ride height offsets are in (mm), heave spring stiffness offsets are in (kN/m) and lap times are in (s).

[a] Positive values bring the car closer to the ground since SAE convention is used.

[b] Positive values move the aero-dynamic centre of pressure rearwards.

[c] The influence of side-slip β and steer δ on aerodynamics (Table. 4.1) are ignored.

[d] Front axle in the North-East corner.

levels are significant, the front ride height increases, thereby supporting further gains in C_l^f. Furthermore, with the increased heave spring stiffnesses, the ride height offsets can be increased by more than the amount achieved by the ride height only optimised car (case E). As shown in Figure 4.13, the ride height variations are smaller in case (F) compared with case (E). Stiffer suspensions, in general, can lead to faster manoeuvre times in F1 cars as this allows them to be kept closer to their optimum aerodynamic operating configuration. However, in practice, there is a trade-off between the performance and ride comfort and care has to be taken to not arbitrarily increase the suspension stiffness. The total reduction in lap time for case (F), when compared with case (E), was 0.29 s.

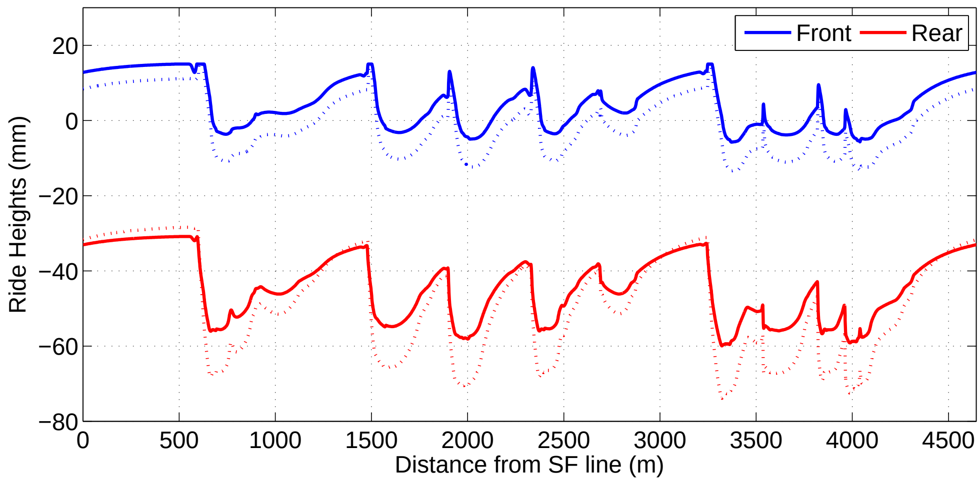

Figure 4.13: Ride height comparisons between the car with ride height and heave spring offsets optimised (see row (F) in Table 4.2) against the car with only heave springs optimised (see row (E) in Table 4.2). The solid curves correspond to row (F) results and the dashed curves to row (E). Front ride heights are shown in blue and the rear ride heights in red.

In the next three tests, rows (G-I) in Table 4.2, the effect of changing the aerodynamic sensitivities to side slip, steer and roll are investigated. If the changes in the side slip, steer and roll sensitivities are ΔC_L^β, ΔC_L^δ and ΔC_L^ϕ, then changes in the nominal downforce coefficients ΔC_L is constrained by

$$\Delta C_L = -\left(w_1 \Delta C_L^\beta + w_2 \Delta C_L^\delta + w_3 \Delta C_L^\phi\right) \qquad w_i \geq 0,$$

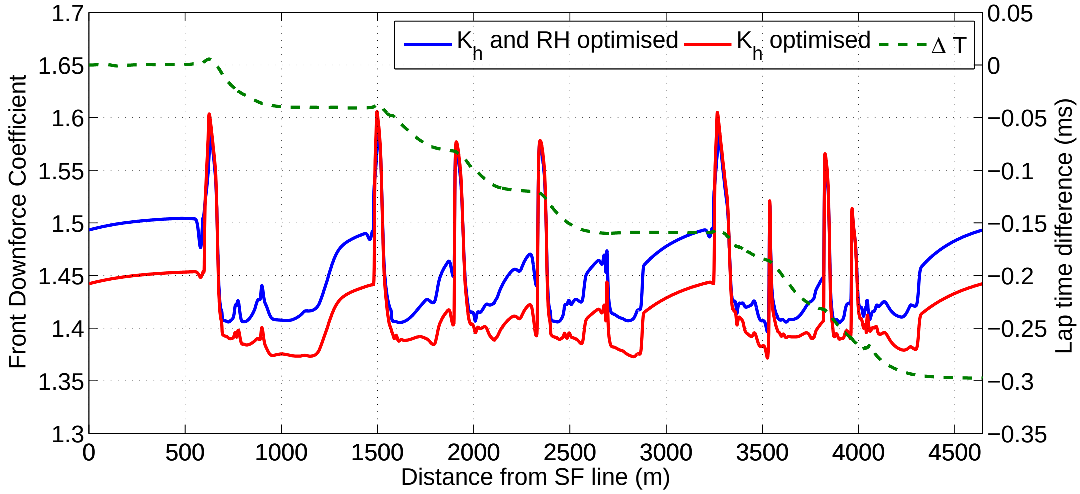

Figure 4.14: Front downforce coefficient comparison between the car with ride height and heave spring offsets optimised (shown in blue and corresponds to row (F) in Table 4.2) against the car with only heave springs optimised (shown in red and corresponds to row (E) in Table 4.2). The lap times difference between the two cases is shown in dashed green.

for weights w_i. The purpose of this constraint is to ensure that the optimisation process is not offered a 'free lunch', whereby the benefits of reducing the aerodynamic sensitivities can be introduced without reducing the nominal downforce coefficient in compensation. One example of a constraint of this form is a water-bed constraint that preserves the "area under the aerodynamic downforce curves". In this case the change in the nominal downforce coefficients are given by

$$\begin{aligned} 0 \;=\; & \int_{-\beta_{max}}^{\beta_{max}} \int_{-\delta_{max}}^{\delta_{max}} \int_{-\phi_{max}}^{\phi_{max}} \Big(\Delta C_L + \Delta C_L^{\beta} \beta^2 \\ & + \Delta C_L^{\delta} \delta^2 + \Delta C_L^{\phi} \phi^2 \Big) \, d\beta d\delta d\phi, \end{aligned}$$

which means that

$$\Delta C_L = - \left(\frac{\beta_{max}^2}{3} \Delta C_L^{\beta} + \frac{\delta_{max}^2}{3} \Delta C_L^{\delta} + \frac{\phi_{max}^2}{3} \Delta C_L^{\phi} \right) \tag{4.30}$$

in which $\beta_{max} = \frac{4\pi}{45}$ rad, $\delta_{max} = \frac{\pi}{12}$ rad and $\phi_{max} = \frac{\pi}{90}$ rad are the ranges of validity of the aerodynamic maps.

In row (G) we allow the front and rear side slip aero-dynamic sensitivities to be changed by ± 5 and ± 10 respectively, with the downforce coefficients constrained by

(4.30). The nominal values for these sensitivities are -5.1294 and -10.2588 respectively; see the fifth row in Table 4.1. In this case, the optimal strategy is to maximise the front and rear side slip sensitivities (more negative values) in order to increase the nominal front and rear downforce coefficients. The associated lap time reduction is 0.25 s. In row (H) the front and rear steer aerodynamic sensitivities are allowed to change by ± 0.15. The nominal values for these sensitivities are -0.1641; see the sixth row in Table 4.1. Again, the water-bed constraint allows a higher nominal downforce coefficient to be 'bought' with increased sensitivity. In this case, there is a small gain in the predicted lap time of 0.04 s.

In row (I), the front and rear roll aerodynamic sensitivities are allowed to change by ± 5 and ± 20 respectively. The nominal values for these sensitivities are -8.2070 and -32.8281 respectively; see the seventh row in Table 4.1. In this case, the sensitivities are minimised thereby decreasing the nominal front and rear downforce coefficients. The rear downforce coefficients for the car with roll sensitivity optimised is compared against the baseline car in Figure 4.15. It is evident that on the straights, where the roll angle of the car is negligible, the baseline vehicle benefits from a higher rear downforce coefficient. Conversely, during high speed cornering, where roll angles are large, the roll optimised car has the higher coefficient. The overall lap time reduction for this case compared with the baseline was 0.08 s.

The three angular sensitivities are optimised simultaneously in row (J), again with constraint (4.30) in place. The strategy turns out to the same as those used when these parameters were optimised separately. The predicted lap time reduction is 0.31 s. The row (K) considers the optimisation of the aerodynamic angular sensitivities along with the ride heights, which makes the car faster than case (J) by 0.51 s. If the heave springs offsets are also added to the optimisation, a further time reduction of 0.05 s is achieved. The associated optimal parameter values are shown in row (L).

In row (M) the aerodynamic balance of the car is adjusted by facilitating a relocation of the aerodynamic centre of pressure. The optimal control calculation is allowed to add a fixed aero offset $-1 \leq a_{off} \leq 1$ to the nominal rear downforce coefficient of

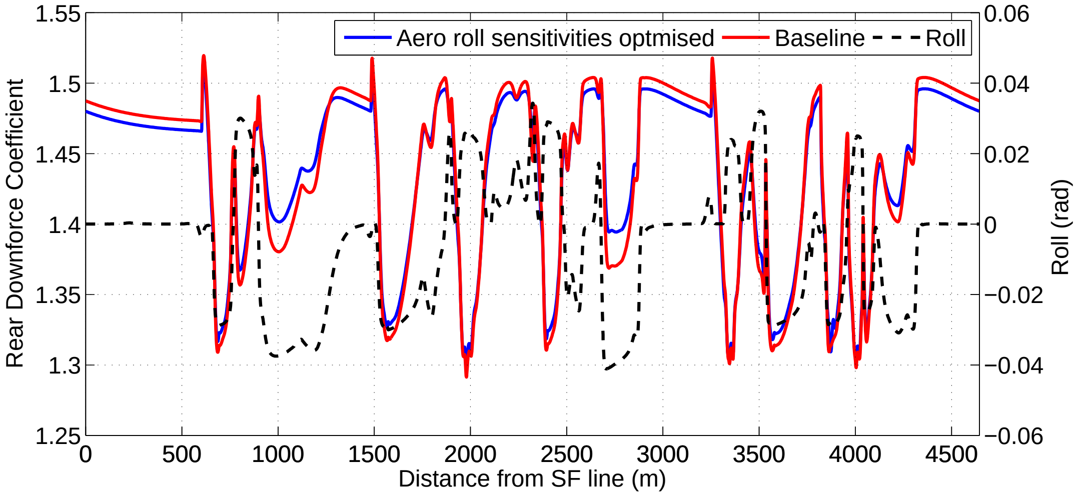

Figure 4.15: Rear downforce comparison between the vehicle with aerodynamic roll sensitivity optimised (see row (I) in Table 4.1) and the baseline car (see row (C) in Table 4.1). The roll angle for the optimised car is shown in dashed black.

0.9494; see Table 4.1. If a_{off} is added to the rear, then it is subtracted in compensation from the front nominal downforce coefficient of 1.20 in order to keep the total downforce constant. In effect, positive values move the centre of pressure (CoP) rearwards, while negative values move the CoP forwards. In the case of the vehicle, tyres and aero map used here, the optimal aero balance adjustment is to move the CoP rearwards by a_{off}= 0.1926, with a resulting lap time reduction of 0.82 s compared to the baseline. Row (N) considers the aero balance optimisation along with the ride heights. By moving the centre of aerodynamics pressure rearwards, the front ride height can now be increased to near its maximum value. This drops the lap time by a further 0.50 s when compared with the case (M). Row (O) considers the aero-balance optimisation in combination with the aerodynamic angular sensitivities with the results being self-explanatory. Row (P) shows the effect of adjusting the ride heights and heave springs in combination with the three aerodynamic sensitivities and the aero balance parameter a_{off} described above. The optimal aero balance coefficient is 0.2427, with the front ride height dropped by the maximum 10 mm, while the rear ride height is dropped by only 3 mm. It appears that adjusting the aero balance coefficient renders rear ride height adjustments virtually unnecessary, because these

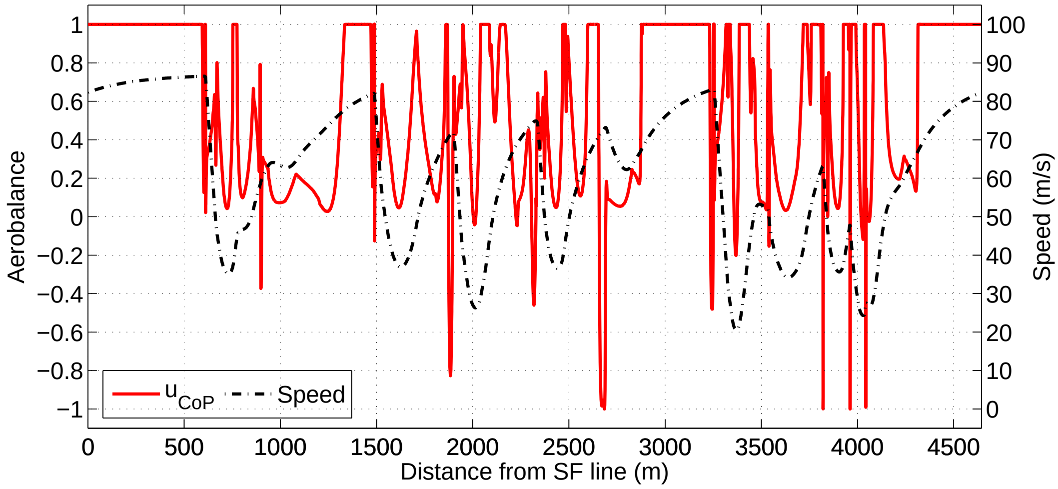

Figure 4.16: Continuously adjustable aero-balance u_{CoP} (red) and vehicle speed (dashed black); see row (P) in Table 4.2. The red curve shows a continuously variable aero-balance control; positive values increase the rear downforce coefficient.

changes produce the same effect. These changes, in combination, reduce the predicted lap time of the baseline car by a significant 1.72 s.

The study given in row (Q) treats the aero balance coefficient as an idealised and continuously variable control input $-1 \leq u_{CoP}(t) \leq 1$. The front downforce coefficient is given by $C_L^f - u_{CoP}(t)$, while the rear downforce coefficient is given by $C_L^r + u_{CoP}(t)$ thereby keeping $C_L = C_L^r + C_L^f$ constant. Positive values of $u_{CoP}(t)$ move the CoP rearwards, while negative values move the CoP forwards. Figure 4.16 shows the aero balance control u_{CoP} (in red) with the vehicle speed (dashed black). The optimal strategy is to move the CoP rearwards under firm acceleration when high normal loads on the rear tyres are required. The CoP is then moved forwards under braking and prior to turning into corners when high front tyre side forces are needed. The influence of this hypothetical control is substantial and reduces the lap time by over 2 s, thereby emphasising the importance of good aerodynamic balance.

By way of comparison, Figure 4.17 shows the changes in the aerodynamic forces produced by optimised suspension and aerodynamic parameters (Row P) against the baseline case. It is evident that the rear downforce is increased significantly all around the track by moving the centre of pressure rearwards. This results in a lower front downforce along the straights. However, similar front downforces are achieved in cornering, where they are needed the most, due to the added benefits from optimisation of the other suspension and aerodynamic parameters. There is a sizeable increase in the total downforce and a slight reduction in the drag force as a result of the optimisation. In the optimised case, the front downforce coefficient was observed to be rarely less than 1.5, while this minimum was approximately 1.4 in the base case. A similar improvement was witnessed in the rear coefficients.

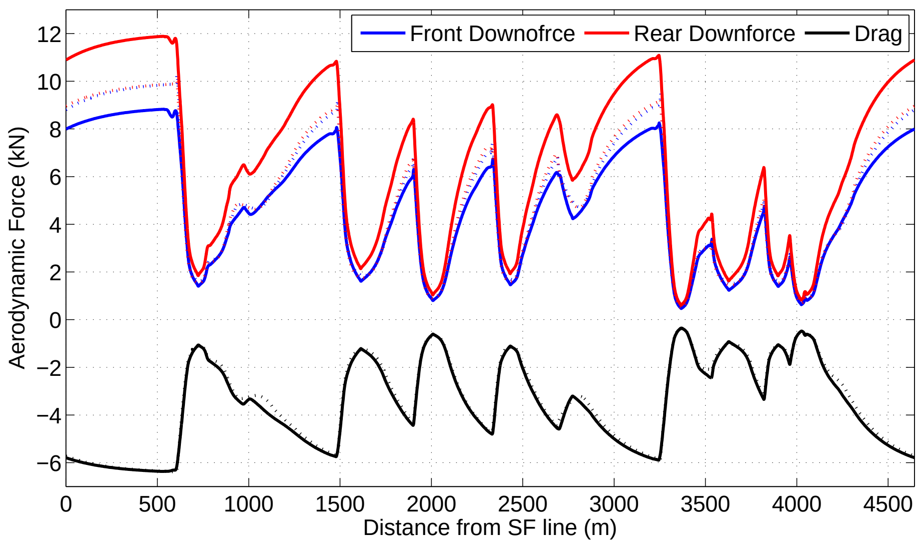

Figure 4.17: Aerodynamic force comparisons between the baseline case (dash lines) and the study with optimised ride heights, heave spring, aero-balance and aero-sensitivities (solid lines); see rows (C) and (P) respectively in Table 4.2. The front downforce is shown in blue, the rear in red and the drag force is shown in black.

Figure 4.18 highlights the improvements in optimal lap time and speed due to the suspension and aerodynamic parameter optimisation in row (P). The main optimisation gains appear to be on corner exits, with some small benefit achieved almost everywhere else. On average, the simulation times were less than 10 minutes on an 8 core 3.4 GHz machine.

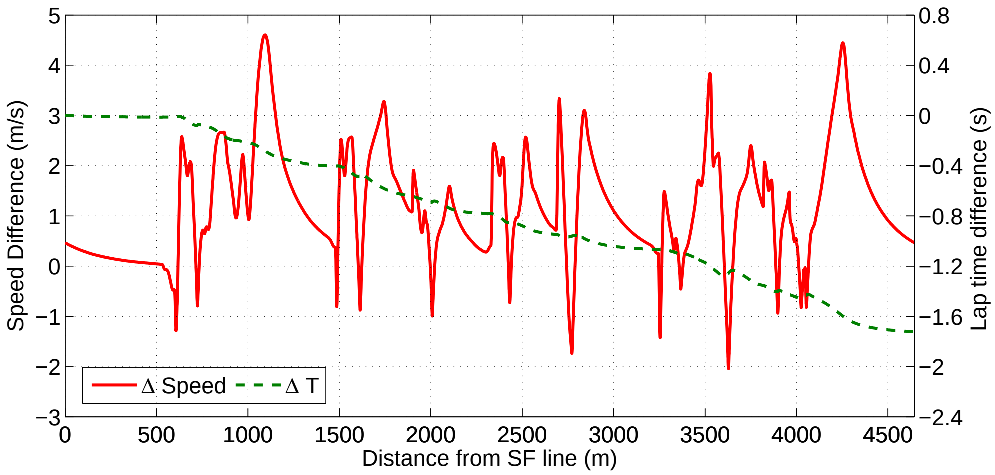

Figure 4.18: Speed difference between the study with optimised ride heights, aero-balance and aero-sensitivities and the baseline case; see rows (P) and (C) respectively in Table 4.2. Positive values indicate that the optimised car is faster. The resulting lap time improvement along the circuit is shown in dashed green.

4.5 Conclusion

Pseudospectral methods for solving numerical optimal control problems are a useful tool for the aerodynamic performance optimisation of high-performance race cars. The results presented in this chapter focus on the interactions between the vehicle's aerodynamic performance and its suspension system. The kinematics of multi-link suspension systems that are typical of open-wheeled race cars involve the solution of a complex set of nonlinear algebraic equations. One source of complication are the torsion bars and heave spring rocker assemblies that produce coupling influences between the road wheels on each axle. It is not practical, or necessary, to include these equations in the vehicle dynamics model used for optimal control calculations. Instead, the solution to these equations can be accurately approximated by a multivariate polynomial, or 'meta-model', which is fitted off-line to the solution of the describing kinematic equations. In this way, the interactions between the car's aerodynamics and its pull-rod suspension system can be accurately represented by smooth (almost linear) polynomials that can be quickly evaluated as part of an optimal control calculation. It is shown that parameter optimisation problems, involving the car's suspension and aerodynamics, are readily soluble. The results demonstrate that the track-specific optimisation of these systems can substantially reduce the lap time. The specific figures will depend on the track, the tyre and vehicle properties, and the closeness of the base vehicle to its optimal set-up. A good mesh-refinement strategy along with the use of Automatic Differentiation for supply of fast and accurate derivatives to the optimisation routine can significantly help improve solution times. This work suggests that ever more complex vehicular optimisation problems may now become tractable.

Chapter 5

Aero-balance Perturbation

5.1 Introduction

The aim of this chapter is to analyse how and to what extent the lap time changes in response to a local perturbation applied to the aerodynamic balance. The aerodynamic balance (aero-balance) is defined by the ratio of the front downforce coefficient to the total of front and rear coefficients and has a significant influence on vehicle performance. This study involves perturbing the aero-balance over a short section of the track and then moving position of this perturbation across the entire manoeuvre. This will enable us to understand how the lap time changes as a function of the application point of perturbation, thereby providing an insight into the aerodynamic operation of the vehicle, which can then be used to guide the car design.

Problems of this type are relatively straightforward to formulate and solve in an optimal control framework, as opposed to the quasi-steady state (QSS) methods that constitute the predominant track-side analysis tool used by F1 teams. Therefore an aim of this study is to investigate how the assumptions made in the QSS simulations would affect the solutions synthesised from an optimal control framework. For instance, QSS methods typically presume that the racing line is given and remains fixed along the manoeuvre. Another assumption is that the car is temporarily in total yaw equilibrium. We will investigate both of these assumptions in more detail subsequently, and will discuss how they can be incorporated into the optimal control problem. As such, one outcome of this chapter is a methodology allowing for reconciliation of the dynamic lap time simulations with the QSS methods.

We commence the chapter with a brief description of QSS methods. We will then calibrate the vehicle against simulator data to both validate the optimal control solutions and obtain a baseline for comparison purposes. The influence of aero-balance perturbation location is then studied on the manoeuvre time. The effect of fixing the racing line and neglecting the yaw dynamics are then highlighted. Finally, the effect of including driver performance limits on the lap time sensitivity to aero-balance perturbations is studied.

5.1.1 QSS simulations

Quasi-steady state simulations take advantage of the so called g-g diagrams to calculate the minimum-time manoeuvre of vehicles. The underlying principle is that a minimum-time manoeuvre must involve the vehicle being at its limits at all times. The g-g diagram is a way of representing these limits. It is a contour plot of the vehicle's maximum lateral and longitudinal accelerations. F1 cars of today have strongly speed dependent characteristics as they make use of aerodynamic downforce to increase grip and traction. This means that the g-g boundary of the vehicle has to be determined at different speeds to obtain the complete picture. A typical g-g diagram is illustrated in Figure 5.1. The solid contour shows the boundary for the car at one particular speed, while the dashed contour demonstrates how this boundary will change at a higher speed. One can see that the higher speed vehicle has a lower longitudinal acceleration limit, but enjoys a higher lateral acceleration limit. This is due to the quadratic dependency of drag and downforces on the speed.

The g-g boundary for a vehicle can be established by setting up a nonlinear program (NLP) for various operating boundary points at each given speed [78]. QSS methods assume that $\dot{u} = constant$, $\dot{v} = 0$ and $\dot{\omega} = 0$, for short time periods (see Section 4.2.2 for a vehicle model). In a straight line, the vehicle has zero yaw rate and side slip velocity, i.e. $\omega = 0$ and $v = 0$. The maximum longitudinal acceleration and deceleration, shown as point A and B in Figure 5.1, can be determined by solving an

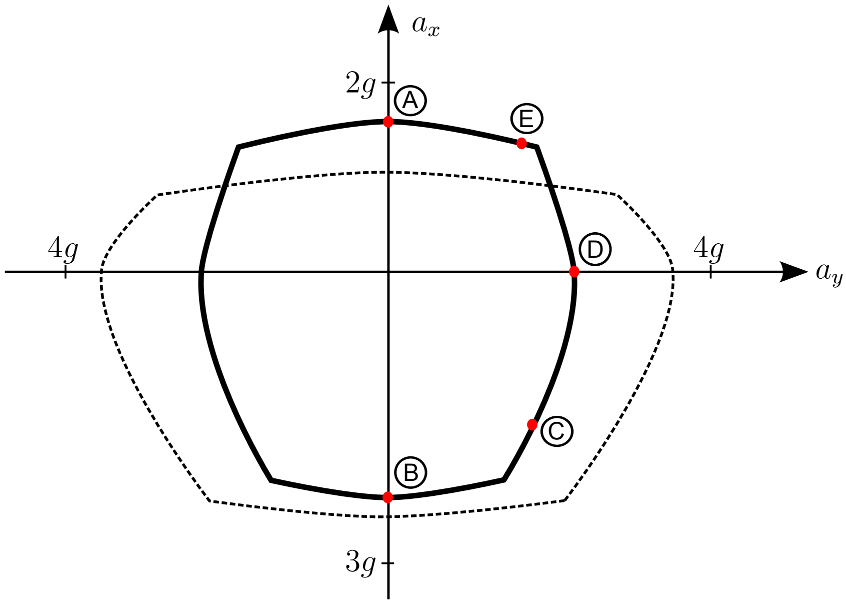

Figure 5.1: A typical GG-digram. The horizontal axis is the lateral acceleration and the vertical axis is the longitudinal acceleration, which are denoted by a_y and a_x respectively. The dash-lined contour corresponds to the vehicle a higher speed.

NLP whose objective is to maximise $\pm\dot{u}$, subject to the QSS conditions mentioned and the vehicle model equations. The decision variables of this NLP are the state and control vectors of vehicle. The other boundary points, such as C, D and E, which have a combination of longitudinal and lateral accelerations, can be found by setting the objective to maximise the steady-state lateral acceleration given by $u\omega$ for different values of longitudinal acceleration $a_x = a_x^{ref}$ between points A and B.

After the g-g diagram boundaries have been found at different speeds, the track is then broken into multiple segments. At each segment, the vehicle accelerations are chosen to lie on the boundary at all points, thus keeping the vehicle at its absolute limit. To illustrate this, let us consider the hairpin manoeuvre shown in Figure 5.2. The vehicle starts at rest and aims to complete the manoeuvre in the least time possible. To achieve this, it must start by accelerating at its maximum longitudinal velocity on the straight. This corresponds to the point A on the g-g diagram shown in Figure 5.1. The vehicle then starts braking at some point B, to prepare for cornering. In cornering, the switch from braking to acceleration, assumed to occur at the apex (point D), implies zero longitudinal acceleration. Moreover, the speed at point D has to satisfy $u^2(s) = a_y(s)/\tilde{\mathcal{C}}(s)$ where $\tilde{\mathcal{C}}$ is the racing line curvature. The apex is the point with the highest curvature across the corner, which has to be identified from the path data. By working backwards from the apex in the braking region and forwards in the acceleration region until the two meet, we can then determine the speed profile from point A to D [3]. This concept is demonstrated in Figure 5.3.

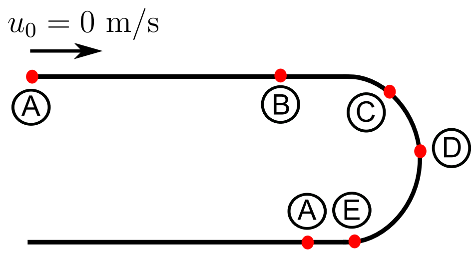

Figure 5.2: The hairpin manoeuvre with the different operating points marked by capital letters.

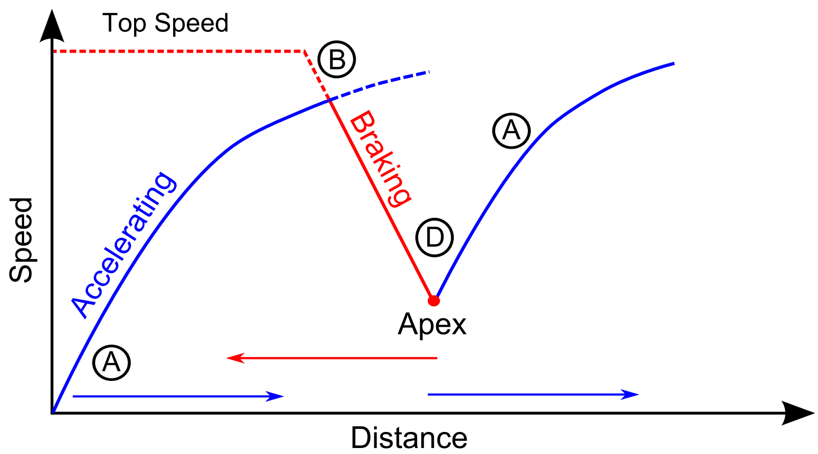

Figure 5.3: Speed profile reconstruction using the approach proposed in [3].

Across each segment the longitudinal acceleration is constant, therefore the following relations can be used to move forward or backward in each segment [3].

$$u(s_{k+1})^2 = u(s_k)^2 + 2a_x(s_k)\Delta s_k \tag{5.1}$$

$$a_y(s_{k+1}) = u(s_{k+1})^2\tilde{\mathcal{C}}(s_{k+1}) \tag{5.2}$$

where $\Delta s_k = s_{k+1} - s_k$. The curvature before and after the apex is smaller than the maximum and hence the lateral acceleration is also smaller. To keep the car at its limit, we therefore have to choose a longitudinal acceleration that puts the vehicle on the boundary of the g-g diagram, corresponding to points C and E. It is also important to note that in this scheme, immediately before and after the apex, the longitudinal velocity remains constant as $a_x = 0$. The segment widths adjacent to the apex therefore need to be made small so that a_x remains zero for as small a distance as possible.

Once the speed profile has been determined for the entire path, the minimum manoeuvre time is simply given by [3]

$$T_{min} = \sum_{k=0}^{M-1} \frac{\Delta s_k}{\left[u(s_{k+1}) + u(s_k)\right]/2}. \tag{5.3}$$

5.2 Vehicle Calibration

In order to generate a baseline for the analysis in this chapter, the vehicle model had to first be calibrated to the data obtained from an F1 driving simulator. This was done to bring the model into line with the actual behaviour of the car on the track and more importantly, to validate the vehicle model and the optimal control calculations. Overall, very good agreement was found between the simulator data and the optimal control results.

Four laps of representative simulator data for the Barcelona circuit were considered for the purpose of model validation. This included the time elapsed, distance travelled, race-line coordinates, yaw velocity and acceleration, longitudinal and lateral velocities and accelerations of the vehicle, wheel slips and tyre forces. Since the optimal control problem formulation in this thesis is based on curve-linear coordinates, it could not readily take the race-line given in Cartesian coordinates as an input. Furthermore, the optimal control problem is parameterised in terms of the distance travelled on the centre-line rather than the actual distance travelled by the vehicle. Therefore, the simulator data provided and optimal control problem results had to be made consistent first. To this end, the curvature and arc length of the race-line given in Cartesian coordinates were extracted and then used as the centre-line in the optimal control problem. The perpendicular distance from the centre-line (state n) was then set to zero, forcing the vehicle to conform to the intended race-line. The tuning of parameters was performed by reverse calculations using the equations of motion and system constraints. To make this calibration easier, speed dependent aerodynamic coefficients were employed, and the suspension was removed. This stands in contrast to the aerodynamic map and suspension meta-model used in the previous chapter. This is the model utilised in the subsequent studies of the chapter.

A low-pass filter was used to remove the high frequency oscillations present in the simulator data. Some vehicle parameters such as mass centre height or suspension roll stiffness could be calculated using simple algebraic manipulation. More complicated parameters such as those involved in the tyre model were calculated using heuristics that compared the outputs produced by related model inputs. After estimation of all parameters, the model was updated and the optimal control problem was resolved. The calculated states and controls were in good agreement with the simulator data but are not published here due to freedom of information constraints associated with competitive F1 racing.

5.3 Baseline Generation

Once the vehicle was calibrated, a track section approximately connecting the midway of the third and fourth corners with the midway of the fourth and fifth corners of the Barcelona track was chosen for the studies of the chapter. This corresponds to centre-line distances of $\sim$1370 m to $\sim$1860 m on the original track data. The optimal control problem solver, GPOPS-II, was set up to divide the track into 1000 meshes, which resulted in a mesh size of $\sim$0.5 m. In order to avoid inconsistencies that could arise from different mesh strategies, the maximum number of mesh-refinement iterations was set to one. The total number of collocation points across each mesh segment was fixed at four by setting the minimum and maximum number of collocations to four. This essentially forces the solver to have a fixed order and mesh size and disables any mesh refinement. The same fine mesh grid is used across all subsequent studies. Finally, tolerance of IPOPT was lowered to 10^{-8} from the default value of 10^{-7} to get more accurate and reliable results. In Figure 5.4, we can see the race-line from the full lap simulation of calibrated vehicle in red. For the shorter manoeuvre, the vehicle starts with fixed initial conditions obtained from the full lap simulation results at the starting point of the shorter manoeuvre. The race-line for this manoeuvre is shown on the same plot in blue (referred to as the '4th Corner Baseline'). The speed profile for both the full lap and the shorter manoeuvre are shown in Figure 5.5.

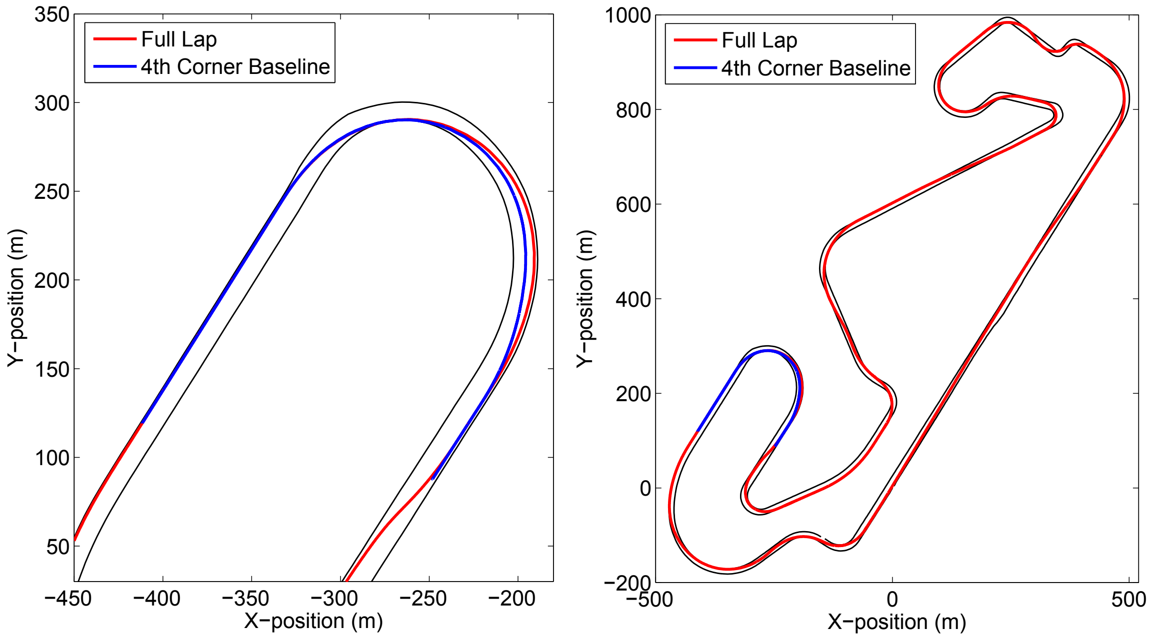

Figure 5.4: Racing line of the full lap and fourth corner for the calibrated vehicle.

It is interesting to note that the speed profile of shorter manoeuvre does not match that of the full lap as only the initial conditions are taken from the full lap simulation. The end boundary conditions are free and the optimiser chooses to take a shorter path at the expense of sacrificing speed, in order to minimise time. This reduced the manoeuvre time by around 20 ms. This also highlights the importance of preview information. In the *infinite horizon* case, the vehicle has to prepare itself for the next corner but this is not the case for the shorter manoeuvre. It can therefore take advantage of the extra freedom to lower the manoeuvre time by taking the path that has a higher curvature so that it reduces the total distance it has to travel. At the same time, it is evident that the initial part of the manoeuvre closely matches

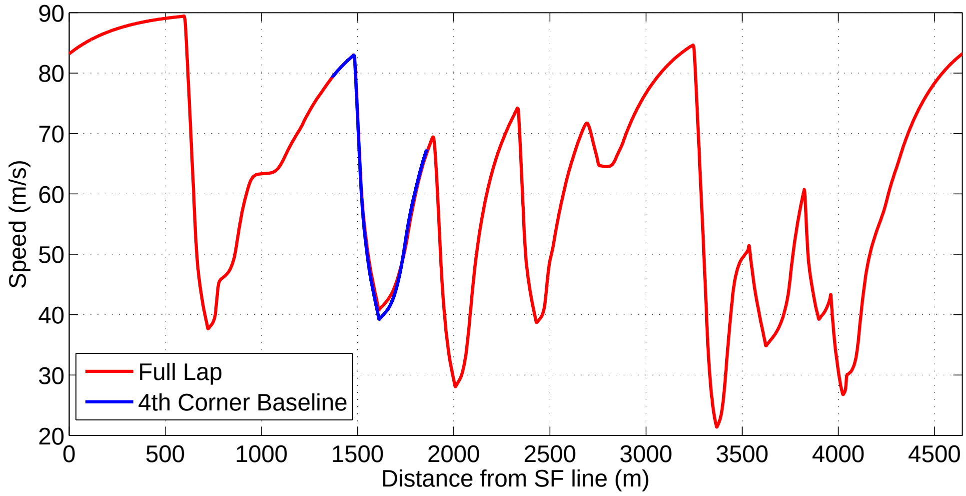

Figure 5.5: Speed profile for the fourth corner manoeuvre along with the full lap. The initial conditions of the manoeuvre are taken from the full lap.

that of the full lap. This implies that it is potentially feasible to solve for the entire lap by solving an overlapping set of *finite horizon* problems, so long as a large enough preview distance is chosen. This problem, however, is beyond the scope of this thesis.

5.4 Aero-Balance Perturbation Analysis

The corner simulation was repeated but this time a localised change of 1% was applied in the aerodynamic balance over a short track section of ∼10 m. The aerodynamic balance is defined as

$$\alpha_{bal} = \frac{C_{lf}}{C_{lf} + C_{lr}}. \tag{5.4}$$

The local perturbation was applied so as to increase C_{lf} by 1% and decrease C_{lr} by the same amount, thus keeping the sum of the downforce coefficients constant. The location of this perturbation was then shifted along the track as shown in Figure 5.6 and the manoeuvre time was recorded. In this way, we obtained a picture of how the manoeuvre time varies as the perturbation is moved through the corner.

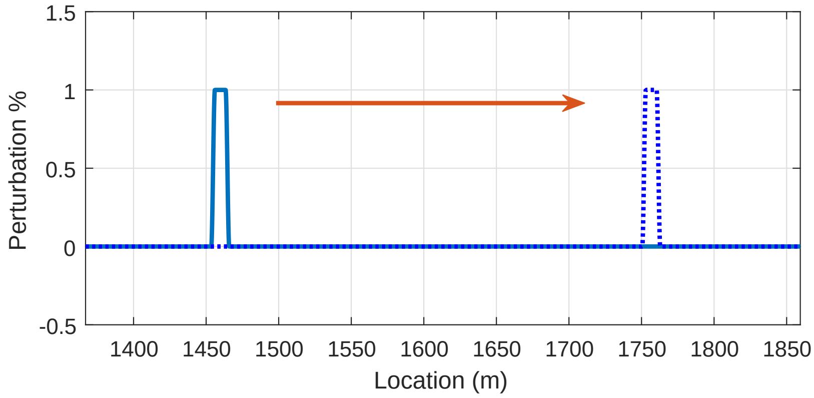

Figure 5.6: Aerodynamics balance perturbation profile. The perturbation is applied for a fixed distance and then shifted along the path in the next simulation.

5.4.1 Fixed Race-line

The aero-balance simulation was first carried out by fixing the race-line to be the same as the baseline manoeuvre. This can be done by introducing an extra path constraint which makes the perpendicular distance to the centreline, state n, to be the same as that recorded from the baseline manoeuvre. The speed profile for the case where the race-line from baseline was used is shown in Figure 5.7. The 1% perturbation on aerodynamic balance seems to not make a major difference in speed profile. The largest time difference occurs when the perturbation is applied from 1603 m to 1613 m. This lap time difference is in the order of milliseconds so the difference is not visible on the speed plot. Figure 5.8 shows the speed profile for different application points of perturbation. The plots are magnified so as to highlight the speed differences at various locations along the corner.

5.4.2 Race-line Optimisation

To quantify the effect of aero-balance perturbations on the race-line, the simulation was repeated but this time additionally optimising the race-line. The results for this case are shown in Figure 5.9 and Figure 5.10. We can see that the 1% perturbation has a barely observable effect on the race-line and speed. The difference in the racing line was on the order of a few centimetres and the speed profiles were quite similar to those of the fixed racing line case.

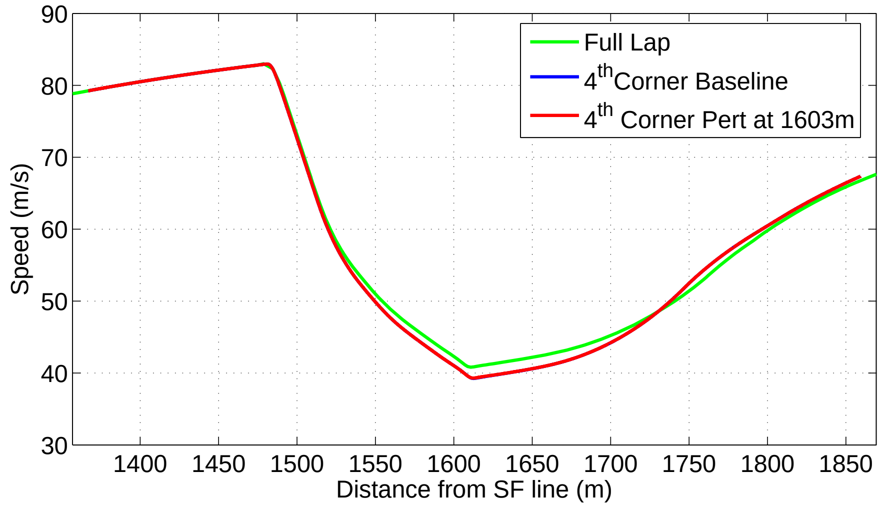

Figure 5.7: Speed profile with the racing line fixed.

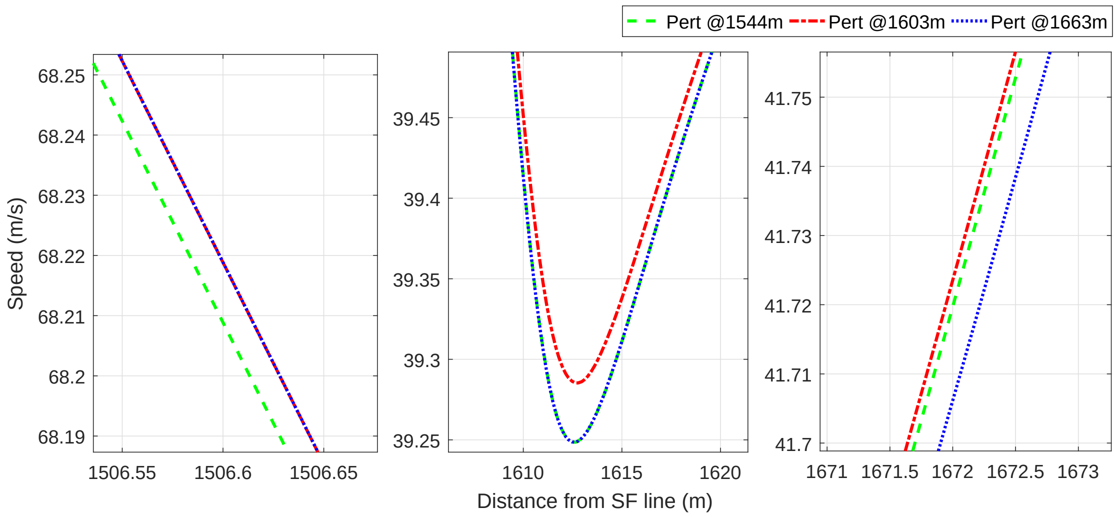

Figure 5.8: Speed profile zoomed in at the corner entry, mid-corner and corner exit. The racing line was fixed. Green corresponds to the perturbation being applied at 1544 m, red at 1603 m, and blue at 1663 m.

5.4.3 Aero-Balance Perturbation Results

If the simulation is repeated as the location of perturbation is shifted along the track, a picture of how the lap time changes with the perturbation location emerges. This result is depicted in Figure 5.11 for both the case where a fixed race-line was used and the case where the race-line was allowed to be optimised. The first point on the plot corresponds to the case where no perturbation was applied, i.e. the baseline

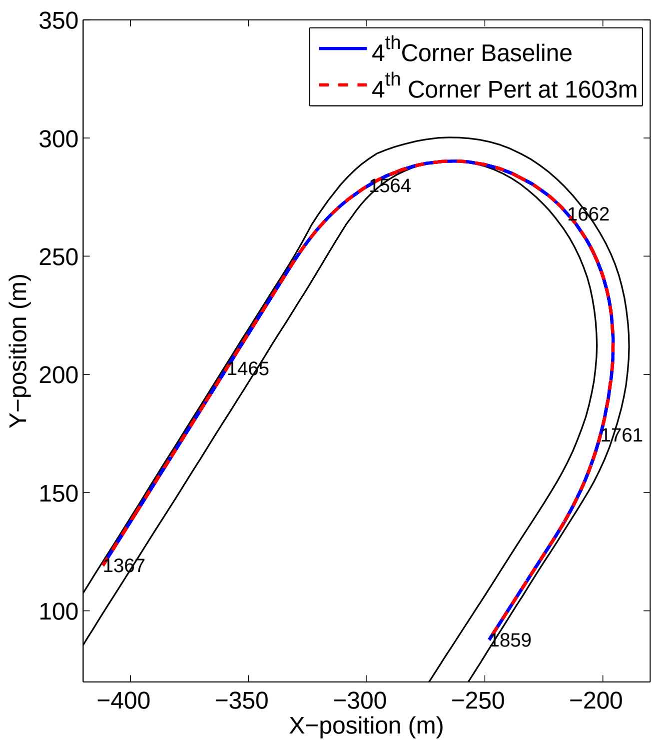

Figure 5.9: Racing line for the aero-balance perturbation at 1603 m when the racing line was allowed to be optimised.

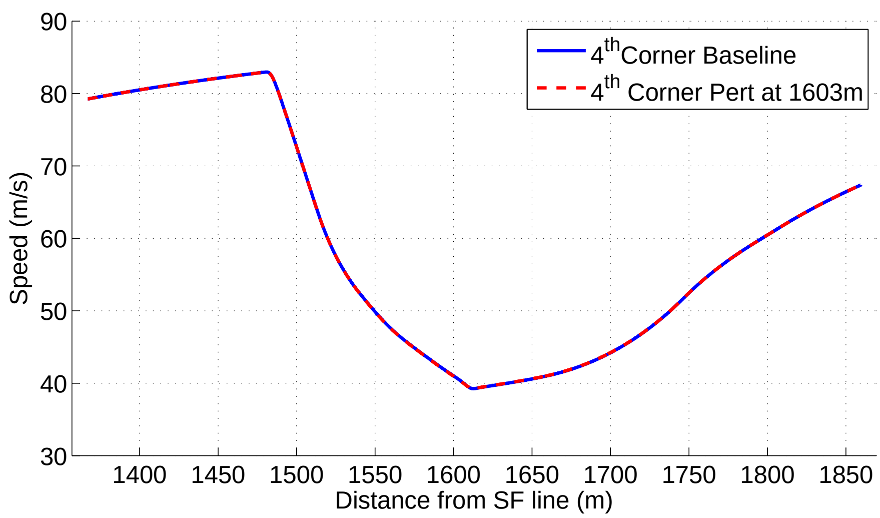

Figure 5.10: Speed profile for the aero-balance perturbation at 1603 m with race-line optimisation.

case. We can see that on the straight, the perturbation essentially makes no difference to the lap time. On the corner entry, a higher value of C_{lr} is beneficial to the lap time as the car is limited by the rear tyre grip. A decrease in rear tyre downforce causes larger lateral slips on the rear tyres, which in turn leads to an oversteering behaviour. The vehicle, in this case, travels at a lower speed through the corner entry, thus increasing manoeuvre time. However, at mid corner a higher front downforce coefficient is advantageous as the front tyres, which provide most of the turning force, can generate more lateral force due to the increased load. Finally, at corner exit, the sudden transfer of rear lift to front causes the rear tyres to take larger lateral slips. This results in the vehicle needing a steering correction to maintain its path. Furthermore, since the vehicle is under traction, the lower rear downforce results in a lower longitudinal acceleration. For these reasons, the perturbation at corner exit results in a higher lap time. Figure 5.8 highlights the speed differences caused by these behaviours.

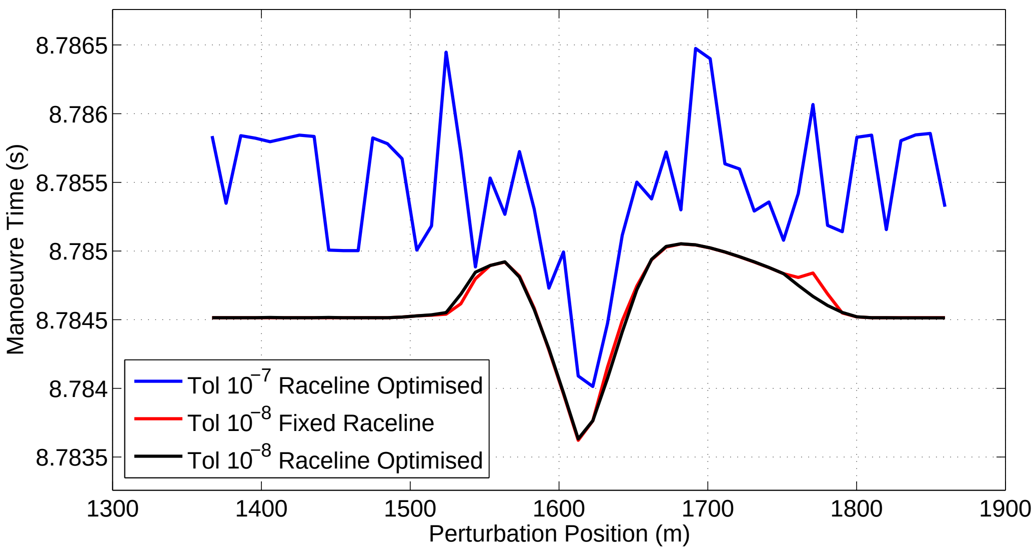

Figure 5.11: Manoeuvre times against the location of application of aero-balance perturbation.

We can see that race-line optimisation has a negligible effect on the aero-balance perturbation results. The simulation was originally carried out with the IPOPT tolerance set to its default value of 10^{-7}. The numerical error, in this case, was found to be too large for a clear picture of the lap time evolution to appear (see the blue trace on Figure 5.11). The NLP solver tolerance was therefore decreased to 10^{-8} to obtain more accurate results.

5.5 Effects of Yaw Dynamics

As a large proportion of the vehicle simulations carried out within F1 are based on quasi-steady state (QSS) methods, it is important to quantify the effect of yaw dynamics on the aero-balance perturbation simulation. To ignore yaw dynamics, the sum of the yaw moments around the car needs to be constrained to zero and the yaw rate must be set to the product of speed and curvature, with the car slip angle free to vary. This was done by introducing two additional control inputs that provided direct control over yaw rate and lateral velocity (denoted by $\tilde{\omega}$ and $\tilde{v}$ respectively). More precisely, we introduce the following two constraints

$$\sum M_z = 0 \tag{5.5a}$$

$$\tilde{\omega} - \frac{F_y \cos\beta - F_x \sin\beta}{M\sqrt{u^2 + \tilde{v}^2}} = 0, \tag{5.5b}$$

where

$$\beta = \arctan\frac{\tilde{v}}{u}, \tag{5.6}$$

is the side slip angle. The first constraint corresponds to setting the yaw acceleration to zero, where M_z denotes the moment around z-axis given by the right-hand side of (4.7). The second constraint sets the yaw rate to be the product of speed and path curvature, using the fact that

$$\tilde{\omega} = \dot{s} \times \text{Path Curvature} = \frac{\text{Centripetal Acceleration}}{\dot{s}}. \tag{5.7}$$

The centripetal acceleration can be obtained by projecting the longitudinal and lateral forces F_x and F_y, as given in (4.8) and (4.9), to be perpendicular to the path. Note the distinction between the path curvature, and the centre-line curvature denoted by $\mathcal{C}$.

In terms of implementation, rather than converting the states ω and v to inputs, one could set the ω and v derivatives to be equal to the new controls. This would mean that $\tilde{\omega}$ and $\tilde{v}$ are able to remain as states, and $\dot{\tilde{\omega}}$ and $\dot{\tilde{v}}$ instead act as the inputs, which replace the original dynamics such that they can be arbitrarily controlled whilst satisfying the constraints given in (5.5).

Having the state v as an input allows the optimiser to arbitrarily set the side-slip angle to any value and thus exploit the available grip. The gross effect of neglecting yaw dynamics can be highlighted by comparing the solutions of the original optimal control problem against that of the modified problem. Figure 5.12 shows the racing line for the case with yaw dynamics included and neglected. On the same plot, the racing line for the model with neglected yaw dynamics and with the aero-balance perturbation applied at 1613 m, which corresponds to the largest lap time difference, is shown in green. We can see that ignoring yaw dynamics negligibly affects the race-line. The perturbation at 1613 m has an even smaller effect on the race-line. Indeed, as with the case including yaw dynamics discussed in the previous section, the difference is on the order of a few centimetres.

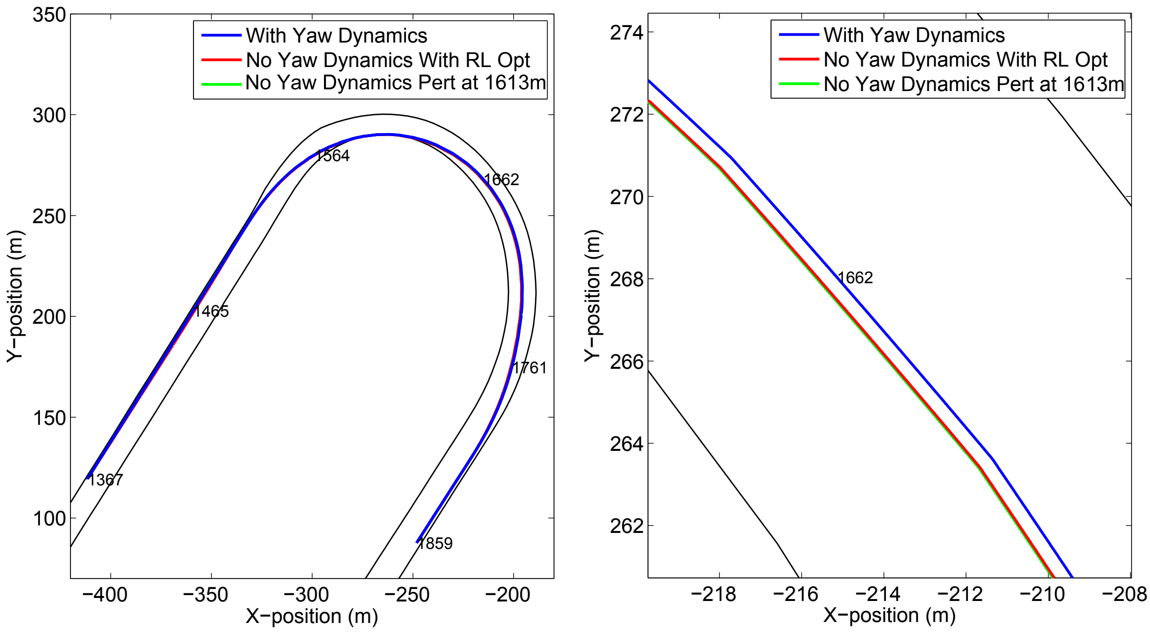

Figure 5.12: Race-line for the case with yaw dynamics included and yaw dynamics ignored.

The vehicle speed in both the presence and absence of yaw dynamics is shown on Figure 5.13. Ignoring yaw dynamics appears to cause the vehicle to brake slightly earlier and to have a higher cornering speed. One can see that influence of race-line optimisation on speed is small but visible. In Figure 5.14 the lateral velocity, which is controlled as an input in the case of dropped yaw dynamics (with and without race-line optimisation), is shown along with the baseline case. Having the lateral velocity as an input could potentially inject artificial external energy into the system. However, Figure 5.14 shows that the magnitude of the difference in lateral velocity between the two cases remains relatively small. The yaw rates are shown in Figure 5.15.

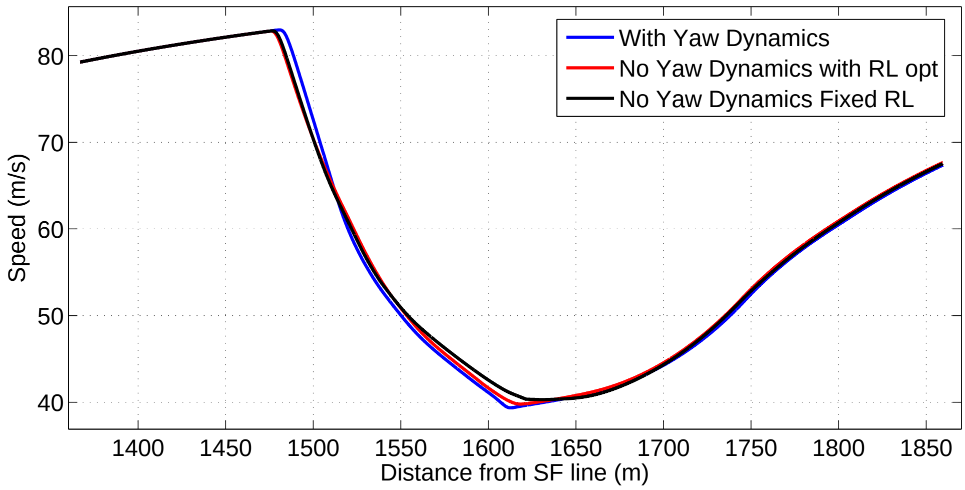

Figure 5.13: Speed profile comparison between the baseline and dropped yaw dynamics case with and without race-line optimisation.

Manoeuvre times for the aero-balance perturbation simulation are shown in Figure 5.16 for the baseline case, and the dropped yaw dynamics case with and without race-line optimisation. When a fixed racing line from the baseline case was utilised, ignoring yaw dynamics resulted in a manoeuvre time which was 62 ms faster. In the model with yaw dynamics neglected, race-line optimisation reduced the manoeuvre time by 12 ms. In order to compare the shape of the aero-perturbation lap time sensitivity curve for these three cases, the normalised lap times were calculated. These results are shown in Figure 5.17. We can observe the same pattern as when the yaw dynamics are included (see Figure 5.11). In the latter case, a higher C_{lf} is advantageous in mid-corner but disadvantageous in corner entry and exit. We can therefore

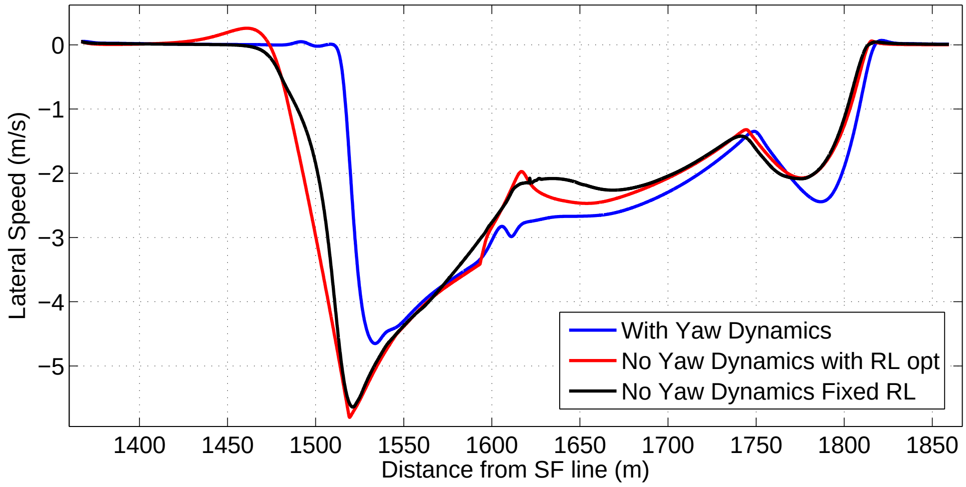

Figure 5.14: Lateral velocity comparison between the baseline and dropped yaw dynamics case.

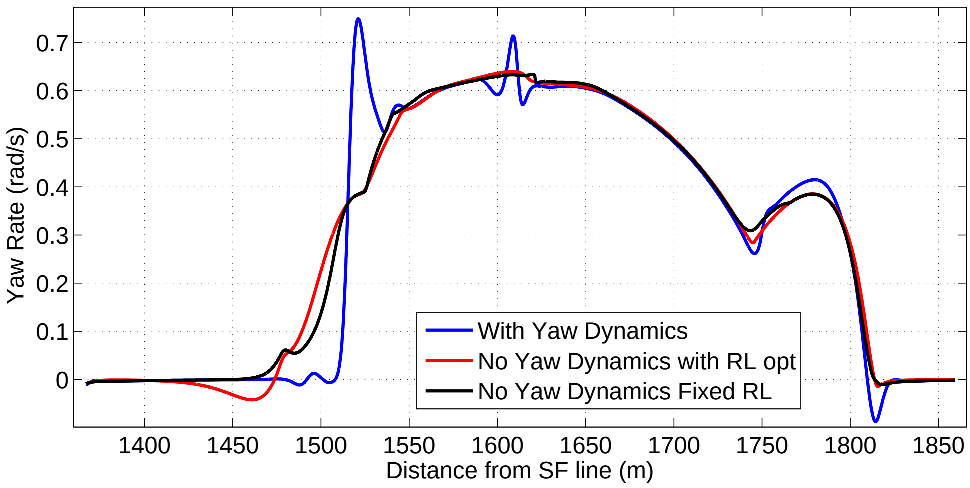

Figure 5.15: Yaw rate comparison between the baseline and dropped yaw dynamics case.

conclude that the inclusion of yaw dynamics negligibly affects the aero-balance perturbation simulation.

5.6 Driver Limits

In order to determine the extent to which driver characteristic limits can influence the aero-perturbation simulation, slew rate limits are imposed on the steering angle and longitudinal slips. By comparing the optimal control results against the simu-

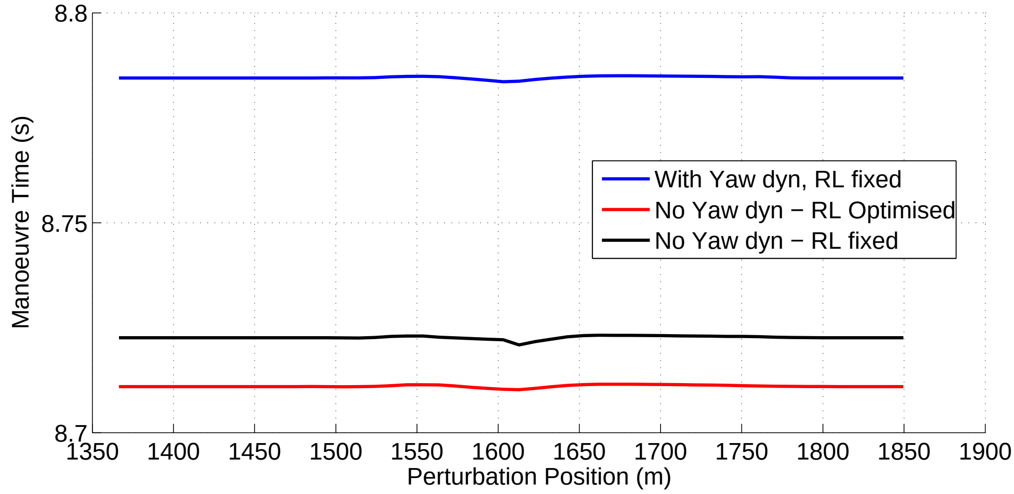

Figure 5.16: Manoeuvre time vs. location of aero-balance perturbation for the case where yaw dynamics are ignored (blue is with race-line optimisation and red without) and the case where yaw dynamics are included (green).

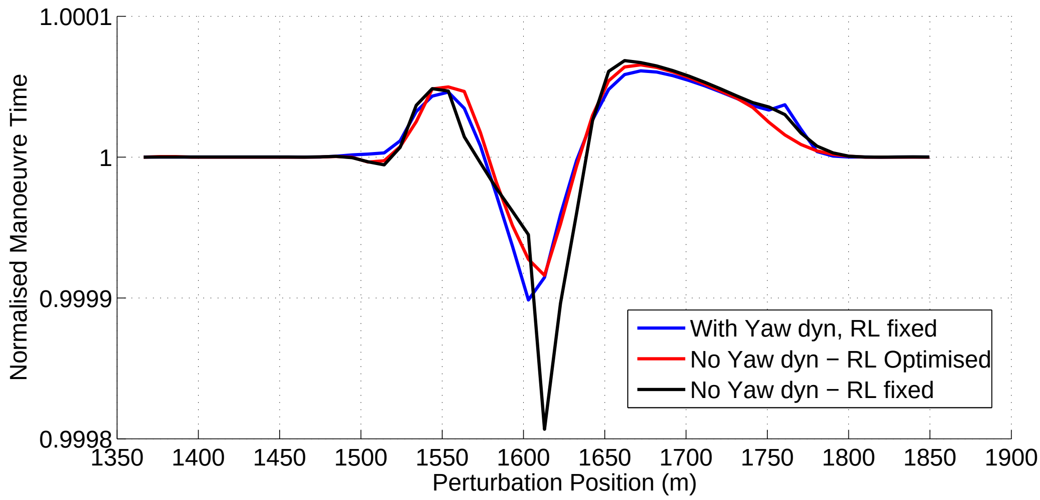

Figure 5.17: Normalised lap times vs. the location of aero-balance perturbation.

lator data, it was determined that slew rates of 20 deg/s on steering and 0.17/s on longitudinal slips would be appropriate. The influence of including these limits on the steering and longitudinal slips are demonstrated on Figure 5.18 and Figure 5.19. Slew rate limits were imposed by bounding the rate of change of steering and longitudinal slips such that $|\dot{u}_{control}|$ was less than the chosen slew rate limit.

The aero-balance perturbation simulation was repeated with the mentioned slew rate limits in place. The results are summarised in Figure 5.20 and Figure 5.21. The

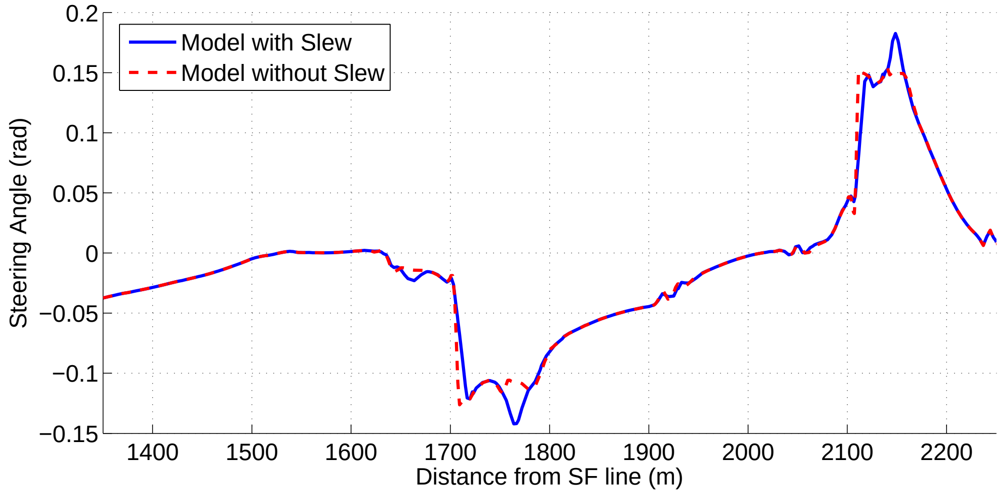

Figure 5.18: Comparison of rear right tyre longitudinal slip, κ_{rr}, between the vehicle model with and without slew rates imposed

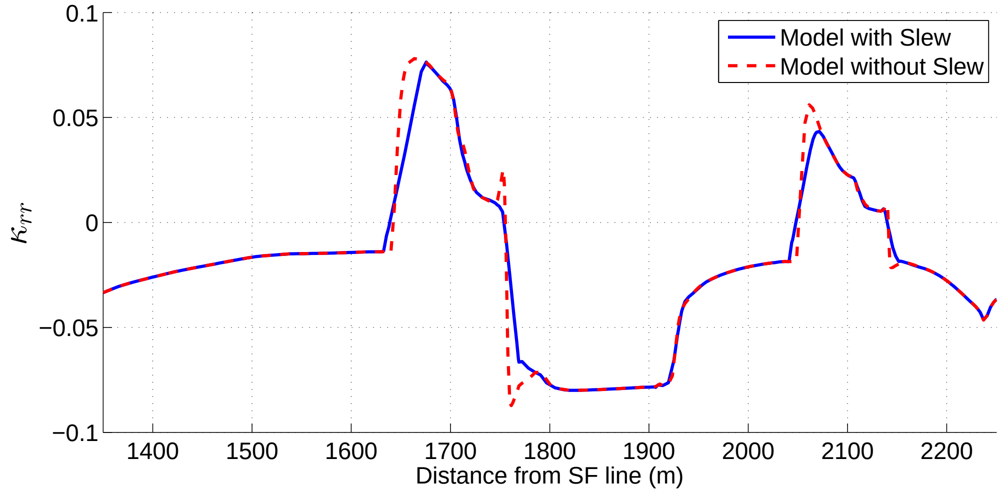

Figure 5.19: Comparison of the steering angle, δ, between the vehicle model with and without slew rates imposed.

simulations were carried out with and without race-line optimisation to see whether the driver having to 'drive around' the perturbation would manifest itself as a need for him/her to pre-empt and prepare for it in a way that is damaging to lap time. As can be seen in Figure 5.20, imposing the slew rate limits increases the lap time by around 6 ms. However, as shown in Figure 5.21, the actual shape of the aero-perturbation lap time sensitivity plot remains almost the same. Overall, we can conclude that the slew rate limits do not have a large influence on the aero-perturbation simulation.

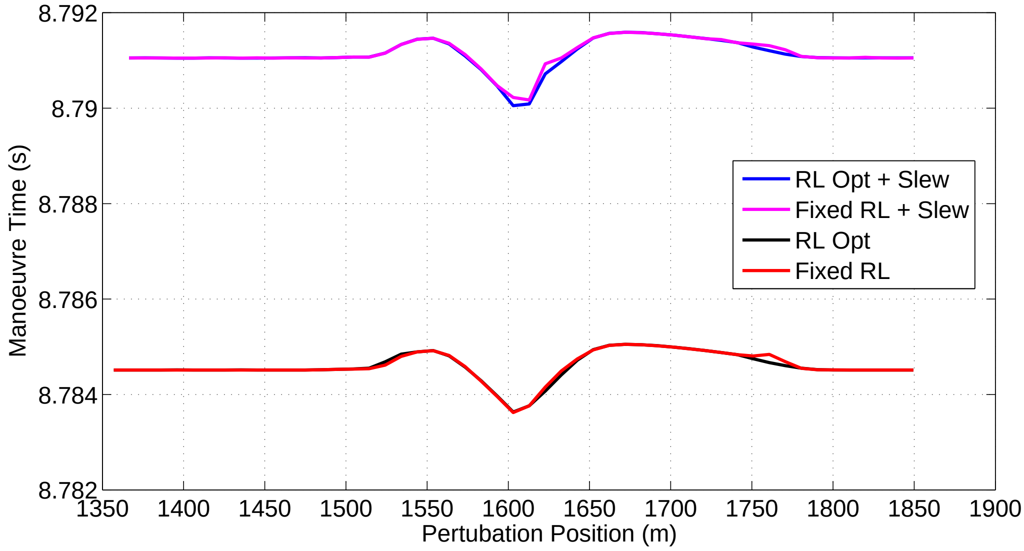

Figure 5.20: Manoeuvre times for aero-balance perturbation simulation for the cases with and without slew rates imposed.

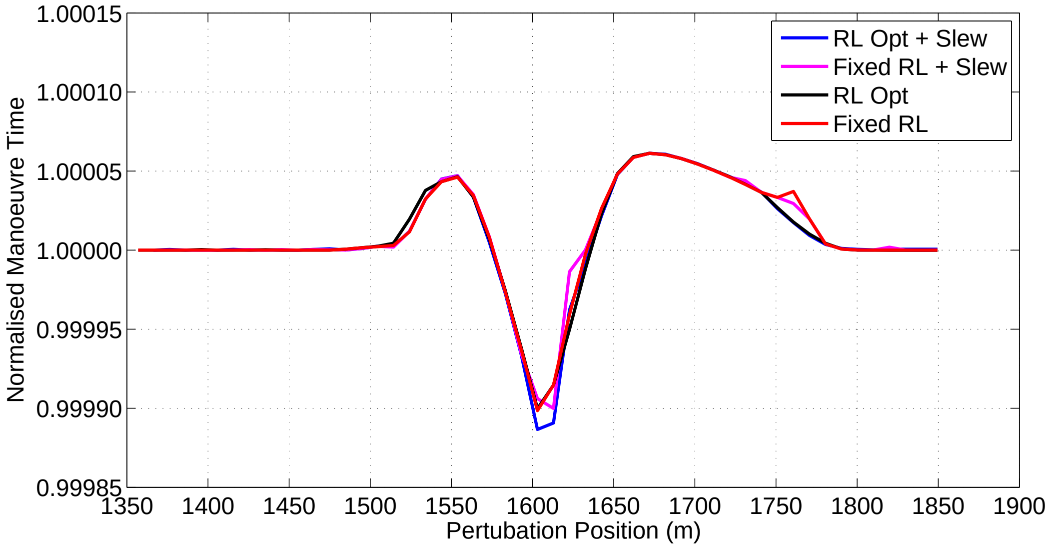

Figure 5.21: Normalised manoeuvre times for aero-balance perturbation simulation for the cases with and without slew rates imposed.

5.7 Conclusions

In this study, we investigated the influence of aerodynamic balance perturbations on lap time. After calibrating the vehicle against the simulator data, a baseline was generated for a cornering manoeuvre extracted from the Barcelona circuit. The aerodynamic balance of the car was then perturbed along a short distance of the track. The lap times were calculated for different positions of perturbation along the manoeuvre. It was shown that an aero-balance perturbation towards the rear axle is more beneficial at corner entry, while one towards the front axle is more beneficial mid-corner. The optimal control problem formulation was then modified to ignore yaw dynamics. It was found that the influence of yaw dynamics on the aero-balance perturbation simulation is small. Finally, slew rates limits were identified and imposed on the controller. It was shown that although such driver limits can increase the overall lap time, the response to the aero-balance perturbation remains unchanged.

Chapter 6

Minimum Fuel Problem

6.1 Introduction

Fuel efficiency and the reduction of CO_2 emissions from road vehicles are problems of ever increasing importance. Energy storage systems based on lithium-ion batteries can be used to recover and then redeploy vehicular kinetic energy. This type of electrical storage has already proved its worth in achieving significant improvements in fuel efficiency in a number of commercially available road cars. In comparison with current battery technologies, flywheel-based energy storage systems have a high power density, high maximum power delivery limits, but a relatively low energy-storage capacity. This makes flywheels especially suitable for kinetic energy recovery systems in high-performance closed-circuit racing cars. In this chapter, we will consider a vehicle that utilises a combined battery-flywheel energy-storage system that exploits the complementary strengths of each energy-storage medium, with the fuel consumption performance of a vehicle subsequently optimised.

Vehicular fuel consumption minimisation has attracted a great deal of recent attention. In the majority of cases driving cycles are used to optimise the drivetrain and control the vehicle in order to reduce fuel consumption. Power management strategies and component sizing for a fuel cell hybrid vehicle are considered in [79]. The driving scenario was modelled as a stochastic process and the power management strategy was calculated using stochastic dynamic programming (SDP). A 'Pseudo-SDP' controller was designed to reduce the computational burden and near optimal results were obtained. SDP was also used in [80] to control a power-split hybrid vehicle. An Equivalent Consumption Minimisation Strategy (ECMS), which is an instantaneous optimisation algorithm, was also designed. The results from both the SDP and the ECMS approach were compared to DP and were shown to be near optimal. This work was then extended in [81], where the hybrid vehicle was optimised by SDP over a distribution of driving cycles to obtain a strategy that is optimal in an 'average' sense.

In [82] the fuel consumption of a hybrid electric vehicle was computed using dynamic programming (DP) on a specific driving cycle. A simple vehicle model was used and the simulation was run for different battery and electric motors sizes in order to optimise these parameters. To control the energy sources in real driving conditions, when the future is unknown, an ECMS algorithm was developed for real-time control. This sub-optimal control law was derived using Pontryagin's principle.

In [83] optimal control was used to synthesise a controller for a parallel hybrid vehicle for a given drive cycle. The controller has the required wheel torque and the vehicle speed as inputs, and the controller's task is to find the gear and power drain from battery in order that the fuel consumption is minimised. The optimal control was found by discretising the cost function (fuel consumption) and computing controls that satisfy the first- and second-order optimality conditions. Much faster results were obtained using this method as compared with DP or simulated annealing.

Optimising vehicle parameters over one drive cycle does not necessarily mean that the vehicle will perform well over other drive cycles; this issue was recognised in [84]. A method was proposed that optimised the HEV over a combined range of drive cycles, with different levels of driving aggressiveness and traffic conditions, in order to reduce the fuel economy variability with respect to drive cycle changes. Nevertheless, this method is still restricted by the set of drive cycle combination used and there were cases where a single driving cycle resulted in a lower fuel variability compared with the proposed multi-cycle method.

Establishing an absolute lower bound on fuel consumption requires controlling the vehicle speed. An alternative method of minimising fuel consumption over different driving conditions, without using driving cycles, is proposed in this chapter. The method involves optimisation over a given travelled route, which is possible using GPS-based navigation systems. Once the driven route has been established, one then chooses a time of arrival constraint, which acts as an ‘aggressiveness’ surrogate. The optimal driving algorithm then seeks to minimise the fuel consumption within the time of arrival constraint. A shorter travel time corresponds to a more aggressive driving style and a higher consumption. A realistic vehicle model with three degrees of freedom and non-linear tyres is employed. It is shown that flywheels are an excellent means of reducing fuel consumption in manoeuvres where high levels of braking are involved such as extra-urban driving on fast roads.

Another distinguishing feature of this study over the majority of the existing literature is that unlike rule-based methods, system dynamics and non-linear constraints can be included as part of the optimal control problem without further modifications and optimal rather than near optimal results are obtained. A pseudo-spectral method will be used to solve the optimal control problem, which makes use of first- and second-order gradient information for fast convergence. Compared to DP/SDP, models of much higher complexity can be solved using this approach.

In this study, we will take the road geometric profile into consideration. In mechanical systems with comparable potential and kinetic energies, the optimal control strategy is typically highly dependent on trade-offs between the two sources of energy, as was the case in the famous "Minimum Time to Climb" problem of the jet-powered aircraft [85]. A simple motivating example is hence provided in Section 6.2 to highlight the importance of road gradient influences. In Section 6.3 the vehicle and track models employed in the study are described. In Section 6.4 the optimal control problem is formulated and in Section 6.5 the numerical solution to the problem is presented and discussed. The conclusions are given in Section 6.6 and the vehicle parameters used in the study are provided in the Appendix C.2.

6.2 Illustrative Example

Imagine a path-constrained point mass (bead) that is required to reach a given destination within a pre-specified time. Suppose the trajectory's starting point is $(0,0)$ in absolute Cartesian coordinates and that the path is parabolic and defined by $y = ax^2 + bx$. If the destination point (x_f, y_f) is given, there holds $y_f = ax_f^2 + bx_f$ and so the path is specified by a single free parameter. We would like to minimise the energy supplied to the bead in order that it arrives at the destination 'just in time'. If the effects of air resistance and friction forces are neglected, the bead's height is indicative of its speed. If the bead has unity mass, conservation of energy dictates

$$\frac{1}{2}v^2 = E_{fuel} - gy. \tag{6.1}$$

The bead's speed is v, its instantaneous height above the origin is y and E_{fuel} is the amount of external energy supplied. The speed of the bead is thus

$$v = \sqrt{2E_{fuel} - 2gy}. \tag{6.2}$$

An infinitesimal path segment can be described as

$$\begin{aligned} ds &= \sqrt{dx^2 + dy^2} = dx\sqrt{(1 + \left(\frac{dy}{dx}\right)^2} \\ &= dx\sqrt{1 + (2ax+b)^2}. \end{aligned} \tag{6.3}$$

Using (6.2) and (6.3) the manoeuvre time is given by

$$T = \int_0^T dt = \int_0^L \frac{1}{v} ds \tag{6.4}$$

$$= \int_0^{x_f} \sqrt{\frac{1 + (2ax+b)^2}{2E_{fuel} - 2g(ax^2+bx)}}\, dx. \tag{6.5}$$

The thrust programme that minimises the total external energy supplied is the optimal control problem of minimising

$$E_{fuel} = \int_{s_0}^{s_f} F(s) \cdot ds = \int_0^{t_f} F(t) \cdot v(t) dt \tag{6.6}$$

subject to state dynamics

$$\begin{cases} \dot{x} = v\dfrac{dx}{ds} = \dfrac{v}{\sqrt{1+\left(\frac{dy}{dx}\right)^2}} \\ \dot{v} = F - \dfrac{g\frac{dy}{dx}}{\sqrt{1+\left(\frac{dy}{dx}\right)^2}} \end{cases} \tag{6.7}$$

where F is the externally applied force. The boundary conditions are

$$\begin{cases} x(t_0) = 0 \quad x(t_f) = 5 \\ v(t_0) = 0 \quad v(t_f) = \text{free}. \end{cases} \tag{6.8}$$

The control Hamiltonian of this system is

$$\mathcal{H} = \lambda_1\dot{x} + \lambda_2\dot{v} + Fv. \tag{6.9}$$

Since $v(t_f)$ is free, $\lambda_2(t_f) = 0$. For the quadratic path the control Hamiltonian becomes

$$\mathcal{H} = F(\lambda_2 + v) + \frac{\lambda_1 v}{\sqrt{1+(2ax+b)^2}} + \frac{\lambda_2 g(2ax+b)}{\sqrt{1+(2ax+b)^2}} \tag{6.10}$$

with co-state equations

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial\mathcal{H}}{\partial\mathbf{x}} = \begin{pmatrix} \partial\mathcal{H}/\partial x \\ -F - \lambda_1/\sqrt{1+(2ax+b)^2} \end{pmatrix} \tag{6.11}$$

where

$$\begin{aligned} \partial\mathcal{H}/\partial x = {} & \frac{2\lambda_2 ag}{\sqrt{1+(2ax+b)^2}} - \frac{2\lambda_2 ag(2ax+b)^2}{(1+(2ax+b)^2)^{3/2}} \\ & + \frac{2\lambda_1 av(2ax+b)}{(1+(2ax+b)^2)^{3/2}}. \end{aligned} \tag{6.12}$$

Since the drive force is bounded, Pontryagin's minimum principle determines that

$$\begin{cases} F = F_{max} \quad \text{if} \quad \lambda_2 < -v \\ F = 0 \quad \text{if} \quad \lambda_2 > -v \\ F = singular \quad \text{if} \quad \lambda_2 = -v. \end{cases} \tag{6.13}$$

In a manner reminiscent of simple variants of the Goddard rocket problem, the optimal thrust programme turns out to be impulsive [86], with all the external energy

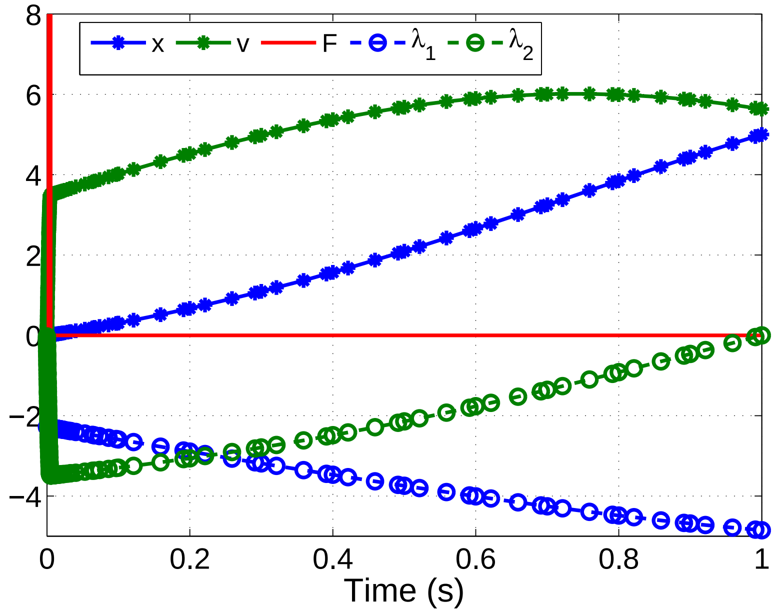

Figure 6.1: Solution of the optimal control problem for $a = 0.1$ and $y_f = -1$.

injected into the system at the beginning of the manoeuvre. To illustrate this, the described optimal control problem was solved with GPOPS-II in the case that $y_f = -1$ and $a = 0.1$; the states, control and co-states are shown in Figure 6.1.

Nine cases are considered that highlight the significance of interactions between the kinetic and potential energy of the bead. These paths correspond to three values for the a parameter and three values for the terminal point; the resulting paths are shown in Figure 6.2.

For each of the nine cases, Table 6.1 shows the path lengths L, the zero-fuel times T_0 (in the cases that a zero-fuel journey is possible), and the external energy E_{fuel} required to complete the journey under a 1 s time of arrival constraint.

The first column in Table 6.1 shows the elevation of the terminal point. To calculate the arrival time, (6.5) is solved numerically by running a bisection on E_{fuel} until the journey time constraint is met.

Superior fuel consumption performance is achieved when the initial part of the journey is downhill, because the early conversion of potential energy into kinetic energy results in higher speeds throughout the journey. Shorter journey paths do

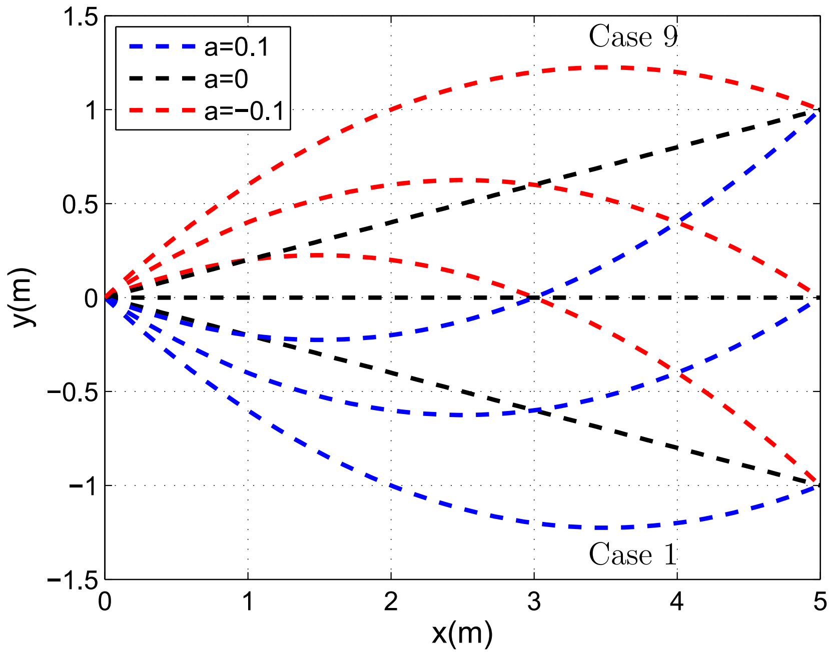

Figure 6.2: Quadratic paths taken by the bead from the starting point $(0, 0)$, to the terminal points $(5, -1)$, $(5, 0)$ and $(5, 1)$.

not necessarily result in lower fuel consumption (e.g. compare Case 1 and 2). As one would expect, the journeys in Cases 7 to 9 are the most arduous due to the high terminal point. This example demonstrates that the path elevation curvature can have a significant influence on fuel usage and sometimes in a counter-intuitive manner.

6.3 The Mathematical Model

A bicycle model of the car is used that has yaw, lateral and longitudinal freedoms, and nonlinear tyres. The road is assumed three-dimensional and is represented by a geometric construct called a 'ribbon', which is describable in terms of three curvature variables. The three-dimensional mathematical model used here was previously described in detail in [4,74]. We make use of the same track description and the equations governing the movement of the point mass on the track. However, the vehicle

Case	y_f	a	b	$L(m)$	$T_0(s)$	$E_{fuel}(J)$
1	-1	0.1	-0.7	5.29	1.57	6.02
2	-1	0	-0.2	5.10	2.30	8.56
3	-1	-0.1	0.3	5.29	-	13.51
4	0	0.1	-0.5	5.20	2.38	9.70
5	0	0	0	5.00		12.50
6	0	-0.1	0.5	5.20	-	17.73
7	+1	0.1	-0.3	5.29	-	15.83
8	+1	0	-0.2	5.10	-	18.37
9	+1	-0.1	0.7	5.29	-	23.31

Table 6.1: Path lengths (L), zero-fuel arrival times (T_0) and fuel requirements (E_{fuel}) for a 1 s arrival time.

model differs by being single track and corresponds to a hybrid passenger vehicle with a battery and flywheel.

6.3.1 Track Model

A moving coordinate system (called a Darboux frame) is used to describe the track. As shown in Figure 6.3 the origin of this moving system is 'dragged along' by the car. The independent variable in the track description is s, which is the distance travelled by the car projected onto the track spine. The spine could be, but need not be, the track centre line.

The x-axis of this coordinate system is the track tangent $\boldsymbol{t}$, the z-axis is perpendicular to the road surface, while the y-axis is orthogonal to x and z-axis and lies on the road surface and is given by $\boldsymbol{n} = \boldsymbol{t} \times \boldsymbol{m}$. The road is represented locally by the $\boldsymbol{t}$-$\boldsymbol{n}$ plane, which 'travels' with the car down the road.

The orientation of this local 'patch' is described in terms of three Euler angles that are all functions of s [74]. If the track orientation is described by the roll, yaw and pitch angles, respectively ϕ, μ and θ, then the track curvatures are given by

$$\boldsymbol{\Omega} = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \frac{1}{\dot{s}} \begin{bmatrix} \dot{\phi} - \sin(\mu)\dot{\theta} \\ \cos(\phi)\dot{\mu} + \cos(\mu)\sin(\phi)\dot{\theta} \\ -\sin(\phi)\dot{\mu} + \cos(\mu)\cos(\phi)\dot{\theta} \end{bmatrix}. \tag{6.14}$$

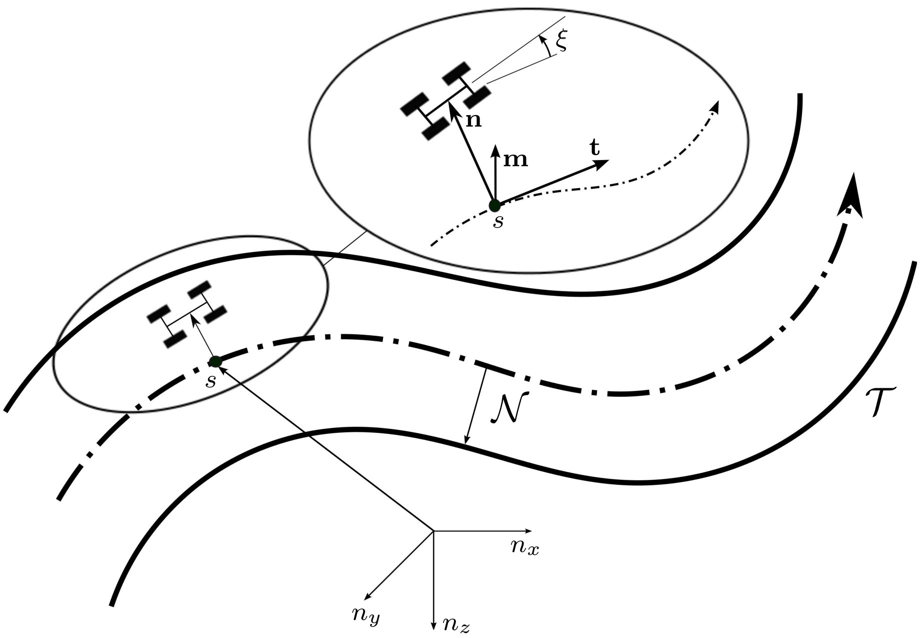

Figure 6.3: Differential-geometric description of a track segment $\mathcal{T}$. The independent variable s represents the distance travelled. The track half-width is $\mathcal{N}$, with ξ the car's yaw angle relative to the track spine's tangent direction. The inertial reference frame is given by n_x, n_y and n_z [4].

The angular velocity of the Darboux frame is given by

$$\boldsymbol{\omega} = \begin{bmatrix}\omega_x & \omega_y & \omega_z\end{bmatrix}^T = \dot{s}\boldsymbol{\Omega}. \tag{6.15}$$

The next kinematic relationships we will require relates to the way in which the car progresses down the road. Suppose that the absolute velocity of the car in its body-fixed coordinate system is $[u\ v\ w]^T$; the longitudinal velocity component u is determined by the throttle/brakes, the lateral component v is determined by the steering, while the vertical component w is determined by the track characteristics. The car's geometric centre (the car's mass centre projected down on to the road) is given by $\boldsymbol{n} = [0\ n\ 0]^T$ in the Darboux frame. The absolute velocity of the car in the Darboux frame is:

$$\begin{bmatrix}\dot{s}\\ \dot{n}\\ 0\end{bmatrix} = \boldsymbol{n} \times \boldsymbol{\omega} + R_z(\xi)\boldsymbol{v} = \begin{bmatrix} n\omega_z + u\cos\xi - v\sin\xi \\ v\cos\xi + u\cos\xi \\ w - n\omega_x \end{bmatrix} \tag{6.16}$$

where

$$R_z(\xi) = R(\mathbf{e}_z, \xi) = \begin{bmatrix} \cos\xi & -\sin\xi & 0 \\ \sin\xi & \cos\xi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.17}$$

represents the yawing of the car relative to the spine of the track. The first term in (6.16) derives from the angular velocity of the Darboux frame, while the second is the velocity of the car's geometric centre expressed in Darboux coordinates. The first row of (6.16) gives the speed of the origin of the Darboux frame in its tangent direction and can be re-written as

$$\dot{s} = \frac{u\cos\xi - v\sin\xi}{1 - n\Omega_z}, \tag{6.18}$$

since $\omega_z = \dot{s}\Omega_z$. The function

$$S_f(s) = \frac{dt}{ds} \tag{6.19}$$

transforms 'time' as the independent variable into the 'elapsed distance' as the independent variable. It is assumed that $S_f(s)$ and its inverse remain non-zero everywhere along the spine. The second row of (6.16):

$$\dot{n} = u\sin\xi + v\cos\xi \tag{6.20}$$

describes the way in which the vehicle moves normal to the spine.

Transforming (6.20) into the distance-travelled domain gives

$$\frac{dn}{ds} = S_f(s)\left(u\sin\xi + v\cos\xi\right). \tag{6.21}$$

Expressing the absolute angular velocity of the car in its body-fixed reference frame gives

$$\bar{\boldsymbol{\omega}} = \begin{bmatrix} \bar{\omega}_x \\ \bar{\omega}_y \\ \bar{\omega}_z \end{bmatrix} = \begin{bmatrix} \cos\xi\omega_x + \sin\xi\omega_y \\ \cos\xi\omega_y - \sin\xi\omega_x \\ \omega_z + \dot{\xi} \end{bmatrix}. \tag{6.22}$$

This will be used in the next section to derive the vehicle's equations of motion. The car's yaw angle in Darboux frame ξ expressed in distance domain is deduced from the third row of (6.22) by integrating

$$\frac{d\xi}{ds} = S_f(s)\bar{\omega}_z - \Omega_z. \tag{6.23}$$

6.3.2 Dynamics

The equations describing the dynamics of the car are derived using standard vectorial methods. The absolute velocity of the car's mass centre (expressed on the vehicle's coordinate system) can be written as [4]

$$\boldsymbol{v}_B = \boldsymbol{v} + \bar{\boldsymbol{\omega}} \times \boldsymbol{h}$$

$$= \begin{bmatrix} u \\ v \\ n\omega_x \end{bmatrix} + \begin{bmatrix} \bar{\omega}_x \\ \bar{\omega}_y \\ \bar{\omega}_z \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ -h \end{bmatrix} = \begin{bmatrix} u - h\bar{\omega}_y \\ v + h\bar{\omega}_x \\ n\omega_x \end{bmatrix} \tag{6.24}$$

where $\times$ denotes the cross product. The Newton-Euler equations for this system are given by

$$M\left(\dot{\boldsymbol{v}}_B + \bar{\boldsymbol{\omega}} \times \boldsymbol{v}_B\right) = \boldsymbol{F}_B + MgR^T\boldsymbol{e}_z \tag{6.25}$$

$$I_B\dot{\bar{\boldsymbol{\omega}}} + \bar{\boldsymbol{\omega}} \times (I_B\bar{\boldsymbol{\omega}}) = \boldsymbol{M}_B, \tag{6.26}$$

where the car's inertia matrix is assumed to be diagonal and is given by $I_B = \mathrm{diag}(I_x\ I_y\ I_z)$, with $\boldsymbol{F}_B = [F_x\ F_y\ F_z]^T$ and $\boldsymbol{M}_B = [M_x\ M_y\ M_z]^T$ being the external force and moment. The last term in (6.25) is due to the gravitational acceleration of the car's mass centre and can be expressed as

$$gR^T\boldsymbol{e}_z = R_z^T(\xi)R_x^T(\phi)R_y^T(\mu)\begin{bmatrix}0 & 0 & g\end{bmatrix}^T$$

$$= g\begin{bmatrix} \sin\xi\sin\phi\cos\mu - \cos\xi\sin\mu \\ \sin\xi\sin\mu + \cos\xi\sin\phi\cos\mu \\ \cos\phi\cos\mu \end{bmatrix}. \tag{6.27}$$

The car's equations of motion can now be assembled from (6.25), (6.26) and (6.27) as follows:

$$\dot{u} = (v + h\bar{\omega}_x)\bar{\omega}_z - n\omega_x\bar{\omega}_y + h\dot{\bar{\omega}}_y + g\left(\sin\xi\sin\phi\cos\mu - \cos\xi\sin\mu\right) + F_x/M \tag{6.28}$$

$$\dot{v} = n\omega_x\bar{\omega}_x - (u - h\bar{\omega}_y)\bar{\omega}_z - h\dot{\bar{\omega}}_x + g\left(\sin\xi\sin\mu + \cos\xi\sin\phi\cos\mu\right) + F_y/M \tag{6.29}$$

$$\dot{\bar{\omega}}_z = \left((I_x - I_y)\bar{\omega}_x\bar{\omega}_y + M_z\right)/I_z, \tag{6.30}$$

in which F_x, F_y are the resultant longitudinal and lateral forces, and M_z is the z-axis tyre moment acting on the car. These quantities are given by

$$F_x = F_{fx}\cos\delta - F_{fy}\sin\delta + F_{rx} + F_{ax} + (F_{fz}\cos\delta + F_{rz})C_r \tag{6.31}$$

$$F_y = F_{fy}\cos\delta + F_{fx}\sin\delta + F_{ry} + F_{fz}\sin\delta C_r \tag{6.32}$$

$$M_z = a\,(F_{fy}\cos\delta + F_{fx}\sin\delta) - bF_{ry}. \tag{6.33}$$

The tyre force system is illustrated in Figure 6.4 and is discussed in Section 6.3.4. The coefficient C_r is the tyre rolling resistance coefficient and the last two terms in (6.31) and (6.32) represent the rolling resistance forces. The aerodynamic drag force F_{ax} acts in negative x-axis direction and is given by

$$F_{ax} = -\frac{1}{2}\,C_D\,\rho\,A\,u^2 \tag{6.34}$$

where C_D is the drag coefficient. Expressing the equations of motion (6.28), (6.29) and (4.7) in terms of s gives

$$\frac{du}{ds} = S_f\dot{u} \tag{6.35}$$

$$\frac{dv}{ds} = S_f\dot{v} \tag{6.36}$$

$$\frac{d\bar{\omega}_z}{ds} = S_f\dot{\bar{\omega}}_z. \tag{6.37}$$

6.3.3 Load Transfer

The vertical tyre loads can be calculated by balancing the forces acting on the car normal to the road, and then balancing the moments around the body-fixed y_b axis (see Figure 6.4). The contributions from the gravitational, inertial, centripetal and aerodynamic forces acting on the car as well as the road-related gyroscopic moments must all be recognised in these calculations. Balancing the forces in road normal direction gives

$$(F_{rz} + F_{fz})/M - n\dot{\omega}_x + g\cos\phi\cos\mu + (u - h\omega_y)\bar{\omega}_y - (v + h\omega_x)\bar{\omega}_x = 0, \tag{6.38}$$

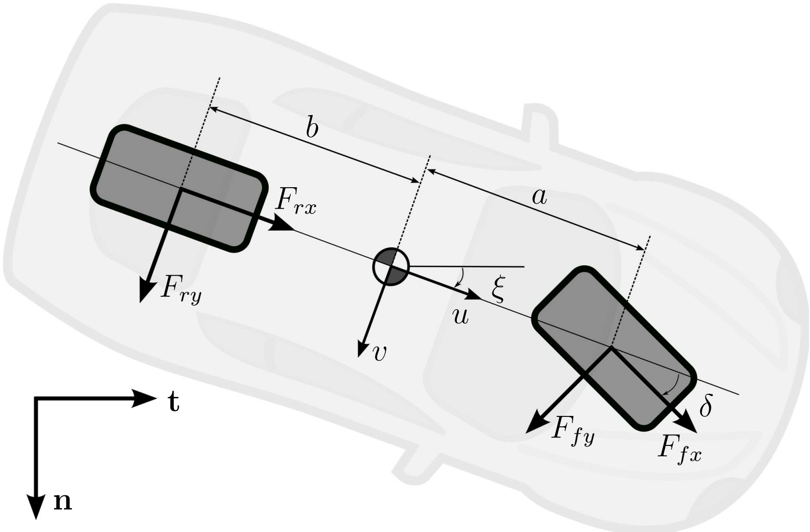

Figure 6.4: Tyre force system. The car's yaw angle with respect to the Darboux frame, which is defined in terms of the vectors $\mathbf{t}$ and $\mathbf{n}$, is ξ and δ is the steering angle.

in which the F_{fz} and F_{rz} are the front and rear tyre normal loads, the second term is the acceleration due to inertial moment around x_b axis, the third term is the acceleration due to gravity, while the last two terms are centripetal accelerations.

Balancing pitching moments around the car's mass centre results in

$$bF_{rz} - aF_{fz} + hF_x - I_y\dot{\bar{\omega}}_y + (I_z - I_x)\bar{\omega}_z\bar{\omega}_x = 0. \tag{6.39}$$

The first two terms represent the pitching moments produced by the normal tyre forces, the third term is the pitching moment produced by the longitudinal force F_x given in (6.31), the fourth is the inertia moment around the y_b axis, whilst the fifth term is a gyroscopic moment acting in the y_b direction.

6.3.4 Tyre Forces

Similar to previous chapters, we make use of the well-known Magic Formula model [87], where tyre forces are a function of the normal load, the tyre's longitudinal slip coefficient κ, and the tyre's lateral slip angle α. The tyre equations are explained in details in Appendix B. The tyre parameters are the same as those described in Table B.1, with the exception that the peak longitudinal and lateral friction coefficients are scaled down by 30 % to reflect the fact that the considered vehicle is a road car rather than a racing car. For the bicycle model employed here, the front and rear tyre lateral slip angles are given by

$$\alpha_r = \arctan\left(\frac{v - \dot{\psi} b}{u}\right), \tag{6.40}$$

$$\alpha_f = \arctan\left(\frac{\cos\delta(\dot{\psi} a + v) - \sin\delta u}{\cos\delta u + \sin\delta(\dot{\psi} a + v)}\right). \tag{6.41}$$

6.3.5 Battery Model

We will use a simple 'voltage behind output resistance' model for the battery, as shown in Figure 6.5. The terminal voltage is

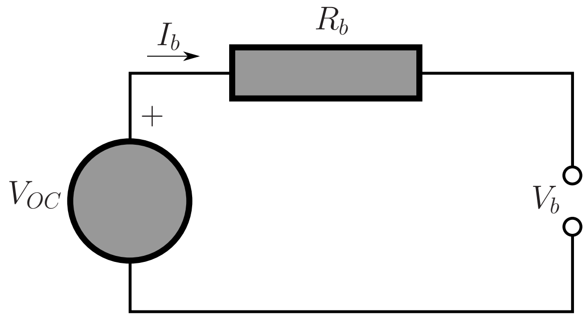

Figure 6.5: Battery equivalent circuit model.

$$V_b = V_{OC} - I_b R_b, \tag{6.42}$$

where V_{OC} is the battery open-circuit voltage, I_b is the battery current and R_b is the internal resistor. In this convention positive I_b corresponds to discharging the battery, while negative I_b corresponds to charging. The unloaded battery voltage V_{OC} has a dependency on the state of the charge SoC as follows

$$V_{OC} = V_{OC}^{min} + (V_{OC}^{max} - V_{OC}^{min})SoC, \tag{6.43}$$

where the SoC is defined as

$$SoC = \frac{Q_b}{Q_b^{max}}. \tag{6.44}$$

The maximum battery charge Q_b^{max} is given by

$$Q_b^{max} = \frac{2E_b^{max}}{V_{OC}^{max} + V_{OC}^{min}} \tag{6.45}$$

where E_b^{max} represents the maximum energy storage capacity of the battery. Using (6.42) and (6.43) the power delivered, or drawn from the battery is

$$P_b = (V_{OC}^{min} + (V_{OC}^{max} - V_{OC}^{min})SoC - I_b R_b)I_b. \tag{6.46}$$

Again negative P_b implies charging and positive P_b implies discharging.

The battery charge can be modelled by the dynamic equation

$$\dot{Q}_b = -I_b. \tag{6.47}$$

The power transmission between the battery and the rear wheel requires an electric motor, which is assumed to have an efficiency factor μ_{em}. The electric motor output power is thus

$$P_{em} = \mu_{em}^{sign(P_b)} P_b. \tag{6.48}$$

6.3.6 Engine Map

The engine used in this study is the 1.5 L Prius engine with maximum power output of 43 kW. The fuel consumption map (see Figure 6.6) for this engine was obtained from the ADVISOR software [88]. One measure of fuel efficiency is brake specific fuel consumption (BSFC), which represents the fuel mass needed to release one unit of energy. A BSFC map can be calculated from the fuel consumption map using

$$BSFC = \frac{\dot{m}_f(\omega_e, T_e)}{\omega_e T_e}, \tag{6.49}$$

where $\dot{m}_f$ is the fuel-mass consumption rate and T_e and ω_e are the engine torque and rotational speed respectively. We will use the BSFC to evaluate the efficiency performance of the engine. The internal combustion engine power is represented by P_{ICE} and is given by

$$P_{ICE} = T_e \omega_e. \tag{6.50}$$

In the optimal control calculations a quadratic multivariate polynomial was fitted

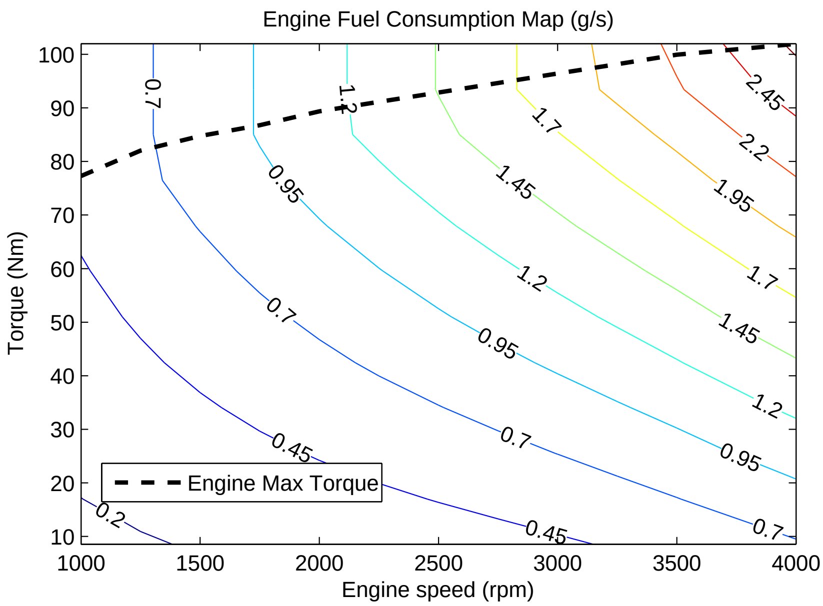

Figure 6.6: Engine fuel consumption map. The consumption rate in g/s is shown on the contours/level sets. The black dashed line is the maximum engine torque available as a function of speed.

to the engine map using a Linear Least Squares algorithm to speed up the fuel consumption calculation. The polynomial captures the shape of the map quite well and has an average absolute error of 2.6% over the entire engine operating range.

6.3.7 Flywheel

A high-speed flywheel is included in the drivetrain to provide a high-power energy re-deployment capability, which complements the low-power high-energy storage capability of the battery. While batteries can store energy for a relatively long time due to their low inherent losses, flywheel storage systems suffer from high losses especially when running at high speeds. A basic flywheel storage system can be represented by a spinning inertia with kinetic energy

$$E_{fly} = \frac{1}{2} J_f \omega_f^2, \tag{6.51}$$

where J_f is the moment of inertia and ω_f is the flywheel's angular velocity. The dominant losses in the flywheel come from the friction in the bearings. These losses are modelled using the empirical relationship given on pg. 147 of [89]

$$P_{loss} = 2 \times 10^{-7}\omega_f^2 + 0.0151\omega_f + 4.0577, \tag{6.52}$$

where ω_f is given in rpm. The flywheel dynamics are described by

$$\dot{E}_{fly} = -P_{fly} - P_{loss}. \tag{6.53}$$

There are also losses in the continuous variable transmission (CVT), which can be lumped into an efficiency factor μ_{CVT}.

6.3.8 Power Transmission

The power transmitted to the rear wheel can be modelled by the constraint

$$P_{ICE} + P_{em} + \mu_{CVT}^{sign(P_{fly})} P_{fly} - (F_{rrx} + F_{rlx})u \geq 0, \tag{6.54}$$

which ensures that the power delivered to the back wheels never exceeds the combined power delivery capability of the internal combustion engine, the battery and the flywheel. If the rear-wheel tyre force is positive, the vehicle is being driven, and the sum of powers delivered by the engine, the flywheel CVT and electric motor will match the mechanical power delivered to the rear wheel. Under braking, rear-wheel tyre force is negative, and the mechanical power at the back wheels is used to charge the battery and/or the flywheel, or else it is dissipated as heat.

At $P_{fly} = 0$, $\mu_{CVT}^{sign(P_{fly})}$ is discontinuous and hence must be approximated using a smooth function. The approximation used in this study is

$$\mu^{sign(P)} \approx 0.5\mu\,(1 + \tanh(\varrho P)) + \frac{0.5}{\mu}(1 + \tanh(-\varrho P)), \tag{6.55}$$

in which ϱ is a constant. As ϱ is increased, the approximation (6.55) approaches $\mu^{sign(P)}$.

6.4 Optimal Control

The minimum fuel problem can be formulated as an optimal control problem that is now described.

The system is described by a set of equations of the form

$$\frac{d\boldsymbol{x}(s)}{ds} = \boldsymbol{f}(\boldsymbol{x}(s), \boldsymbol{u}(s)), \tag{6.56}$$

in which the state-vector is given by

$$\boldsymbol{x} = \begin{bmatrix} n & \xi & u & v & \omega & E_{fly} & Q_b \end{bmatrix}^T. \tag{6.57}$$

The associated differential equations are given by (6.21), (6.23), (6.28), (6.29), (4.7), (6.47) and (6.53), respectively. The control vector is given by

$$\boldsymbol{u} = [\delta \quad k_f \quad k_r \quad F_{fz} \quad F_{rz} \quad \omega_e \quad T_e \quad P_{fly} \quad I_b]^T. \tag{6.58}$$

The problem is also subject to the path constraints (6.38), (6.39), (6.54) and bounds on states and controls. There is also a constraint on maximum engine torque available as shown in Figure 6.6.

The minimum-fuel performance index to be minimised is given by

$$J = \int_{s_0}^{s_f} S_f(s)\dot{m}_f \, ds. \tag{6.59}$$

The arrival time constraint can be written as an integral constraint as below

$$\int_{s_0}^{s_f} S_f(s) ds \, \leq \, T, \tag{6.60}$$

where T is the arrival time.

In practice however, to avoid jerky controls and singular arcs, we actually control $\dot{\boldsymbol{u}}$ and minimise

$$J_{mod} = \int_{s_0}^{s_f} S_f(s)(\dot{m}_f + \dot{\boldsymbol{u}}^T \boldsymbol{R} \dot{\boldsymbol{u}}) \, ds \tag{6.61}$$

in which $\boldsymbol{R}$ is an appropriate weighting matrix. We also impose slew-rate limits on controls by placing hard constraints on $|\dot{\boldsymbol{u}}|$.

6.5 Results

In order to illustrate the concepts described, the motor-racing Circuit de Spa Francorchamps will be used as an exemplar track. The path is restricted to the neighbourhood of the track centre line in order to make comparisons easier; this was achieved by constraining the state n, which is the perpendicular distance from the car mass centre to the track centre line, to be 'small'. The three-dimensional track, as well as a two-dimensional projection on to a ground plane are shown in Figure 6.7.

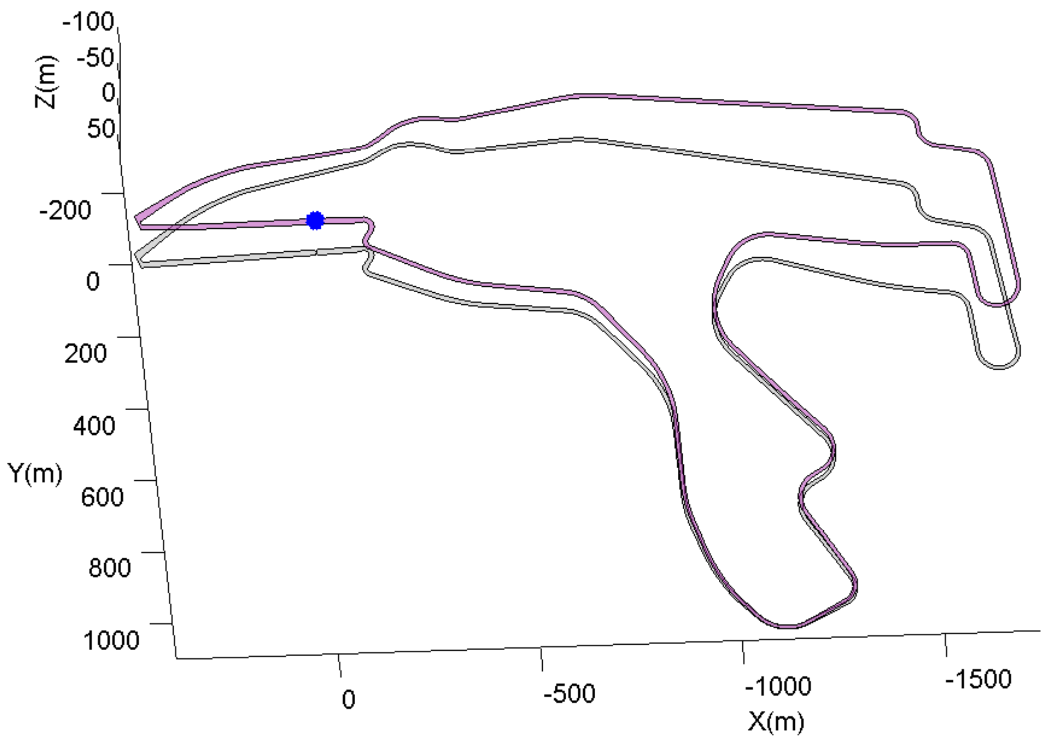

Figure 6.7: Circuit de Spa-Francorchamps used in the presented study. Start/Finish line (SF) is marked by the blue dot.

The route that the car is constrained to take together with the corner distances are shown in Figure 6.8. This figure will be useful in analysing the results to be presented later. In all the simulations described here, the vehicle will start at rest (in practice the vehicle is set to start from a negligibly small velocity to keep the reciprocal of the speed well defined) at the SF line, and will complete the circuit such that fuel usage is minimised. The car will come to a standstill at the end of the route.

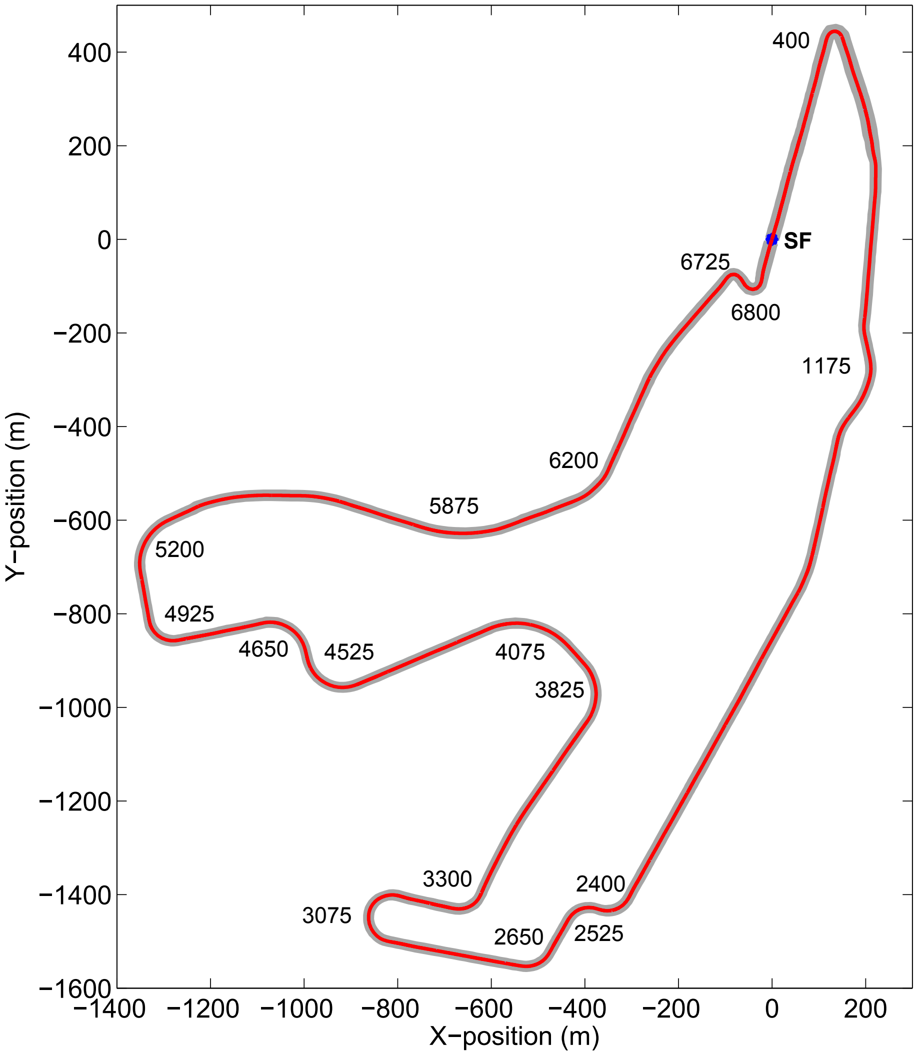

Figure 6.8: The path that vehicle is set to follow with distances travelled marked on the plot in meters.

The minimum fuel problem was solved for a journey time of 240 s on the three-dimensional and two-dimensional track; this calculation is used to quantify the significance of the road undulations. The optimal speed profile of the full hybrid vehicle for both the two and three-dimensional tracks, along with the track elevation changes, are depicted in Figure 6.9. One can see that the track is on a slight incline when the car starts its journey; this results in the car's initial speed being slightly higher on the 2D track. At approximately 400 m the track elevation falls away and the speed on the 3D track overtakes that in the 2D track. The 3D track speed advantage is then 'given back' as the car enters the rising gradient at approximately 1100 m. At

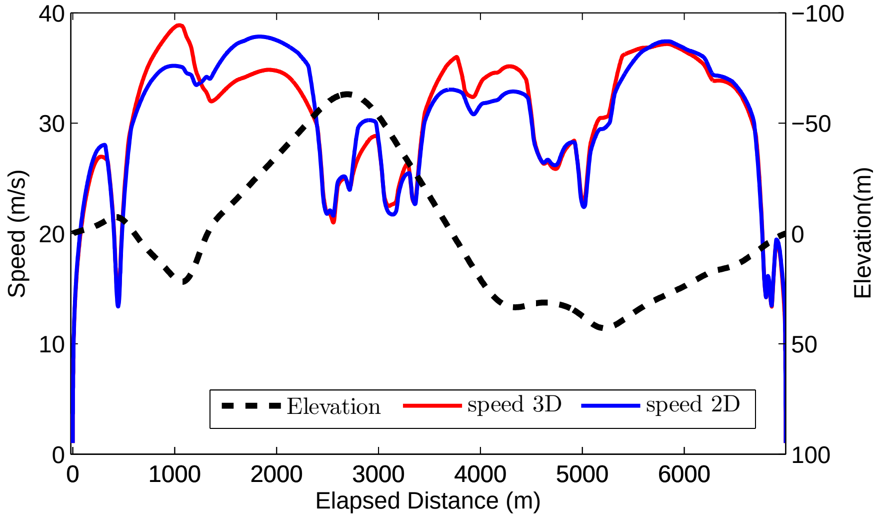

Figure 6.9: Speed profile for the three-dimensional (red solid line) and two-dimensional (blue solid line) track for the case when arrival time was set to 240 s. The black dashed line shows the road elevation heights.

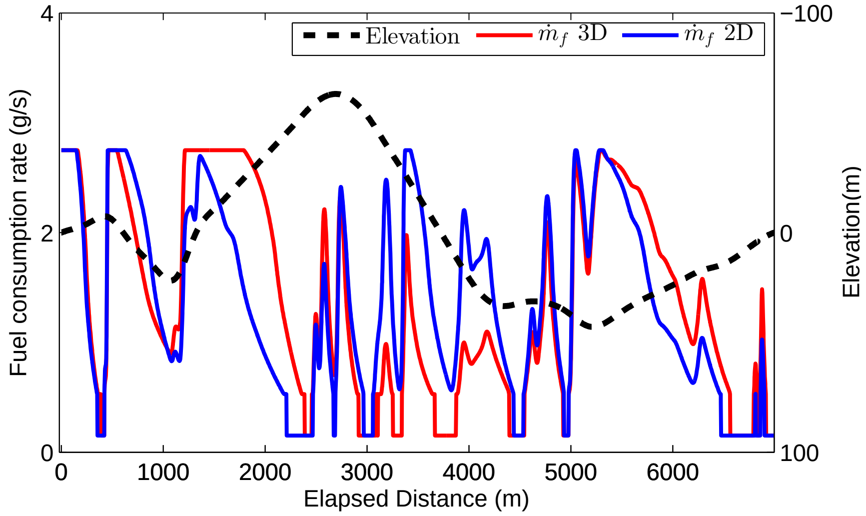

Figure 6.10: Fuel consumption rate in g/s for the three-dimensional (red solid line) and two-dimensional (blue solid line) track for the case when arrival time was set to 240 s. The black dashed line shows the road elevation profile.

the start of the incline at 1100 m, one also sees an increase in the fuel consumption rate as shown in Figure 6.10.

The vehicle's fuel consumption rate is high initially in order for it to accelerate from rest. As the vehicle approaches the first turn at 400 m, the throttle is eased off at approximately 150 m, with the brake applied at approximately 200 m (see also

Figure 6.11). The full-lap fuel usage for the 2D case was 321.2 g, while that in the 3D case was 320.5 g. Although this difference is small, Figure 6.10 shows that the fuel consumption strategy is different in the two cases; more fuel is used ascending hills, while fuel is saved coming down them.

Figure 6.11 shows the power management strategy of the vehicle. An inspection of this plot brings a number of issues to light: (i) the battery power delivery and absorption is relatively small compared to the flywheel; (ii) the controller tries to utilise the stored flywheel energy as fast as possible in order to avoid energy dissipation resulting from the rotational losses; (iii) The battery stored energy is only used when that in the flywheel has been depleted; (iv) often times the power delivery from the flywheel exceeds that of the engine; (v) the battery is discharged at relatively few isolated points on the circuit, and only when the flywheel energy has been depleted.

The power losses associated with various dissipation mechanisms are shown in Figure 6.12. Predominant are the aerodynamic drag power losses that are proportional to the cube of the forward speed; see (6.34). The rolling resistance loss is given by the product of forward speed and the rolling resistance forces. The flywheel losses are the sum of internal rotational losses (6.52) and the CVT losses. The peaks that appear in the flywheel loss plot are due to transmission losses. The low power loss regions such as the one that appears around 4000 m and the gradients on the peaks, are due to the flywheel self-discharging. The battery and electric motor losses come from the battery's output resistance and motor efficiency. There are also losses due to the longitudinal and lateral tyre slips but are not shown in the figure.

The brake specific fuel consumption map (6.49) is shown in Figure 6.13. It can be seen that the optimal controller tries to utilise the engine such that it operates as close as possible to the optimum efficiency line; this implies that the controller chooses the engine speed and torque such that maximum mechanical energy is extracted per unit of fuel mass burnt.

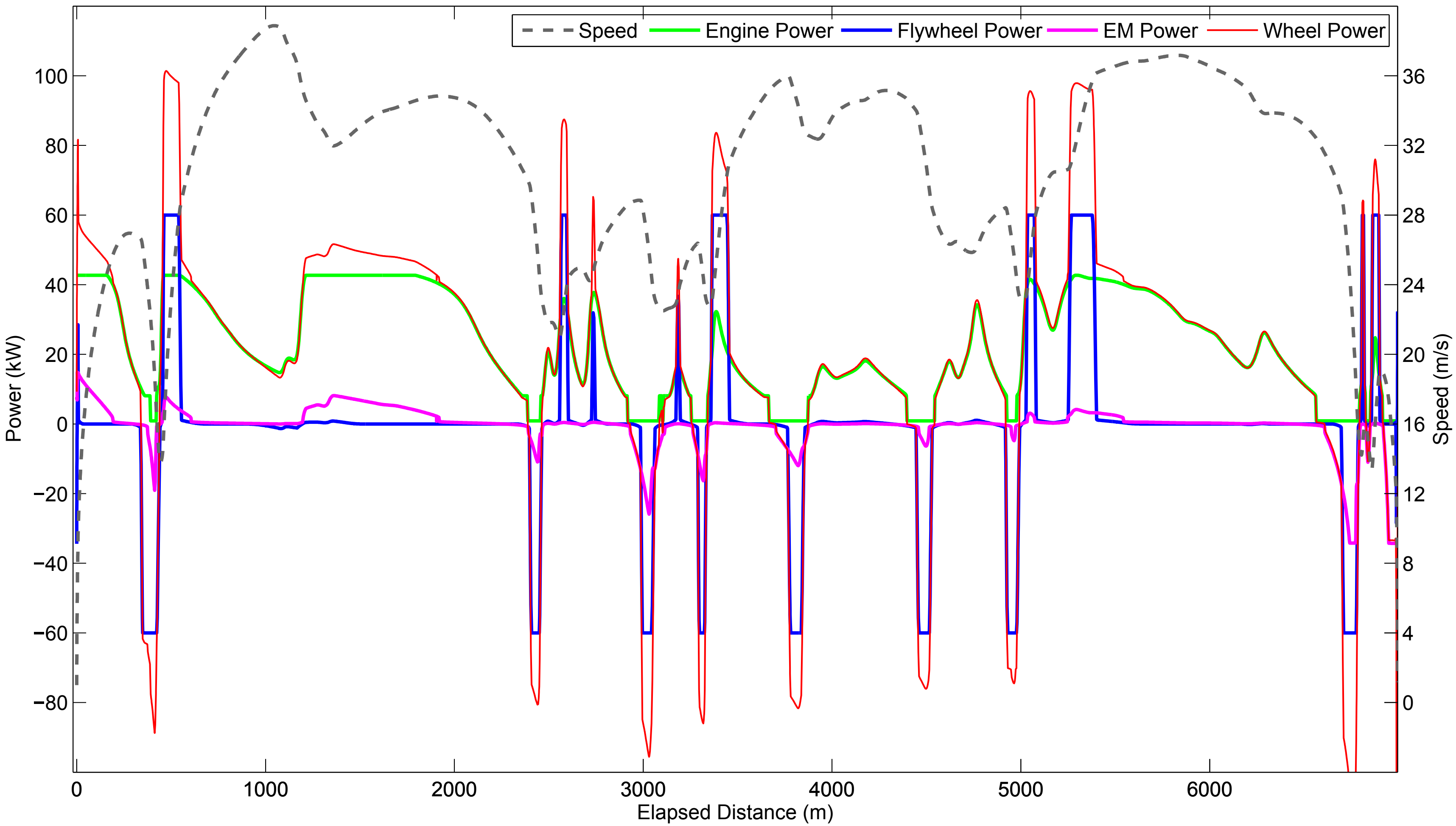

Figure 6.11: Power strategy plot for the vehicle with flywheel and battery when the arrival time was set to 240 s. The power produced by the engine is shown in green, flywheel power in blue, electric motor power in magenta and the rear wheel mechanical power is shown in red. The grey dashed line represents the speed profile.

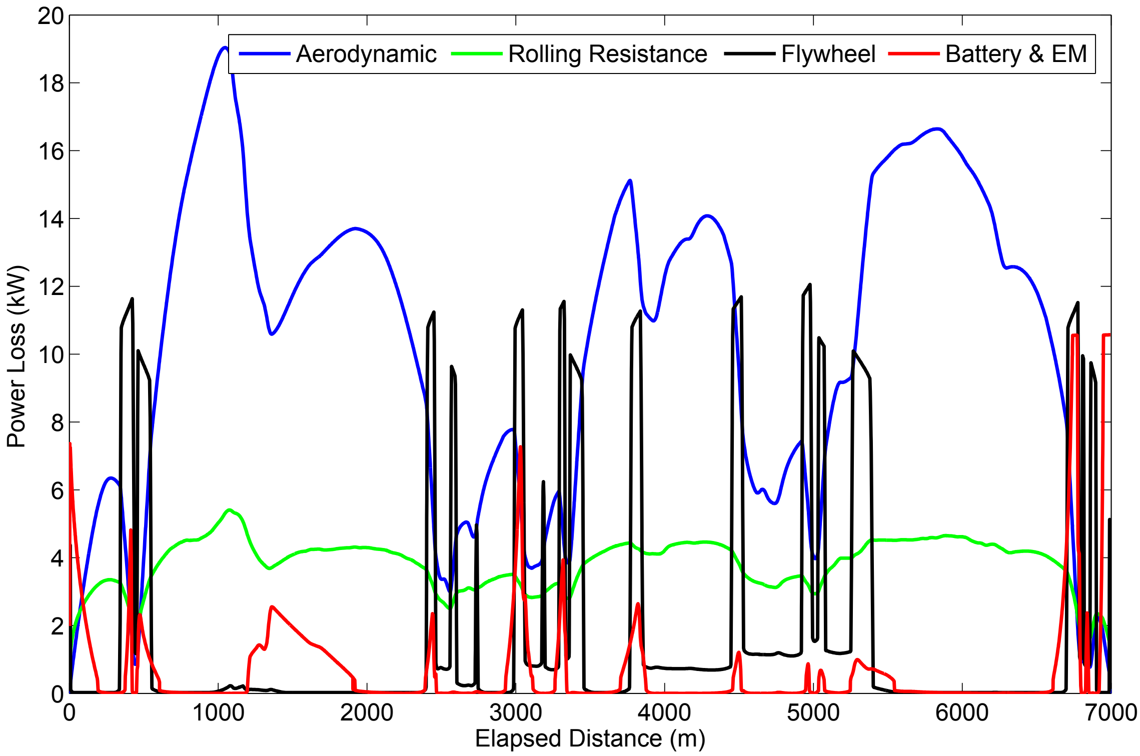

Figure 6.12: Power losses as a function of elapsed distance from the start-finish line. The aerodynamic losses are shown in blue, the rolling resistance losses are shown in green, the flywheel losses are shown in black, and the battery and motor losses in red.

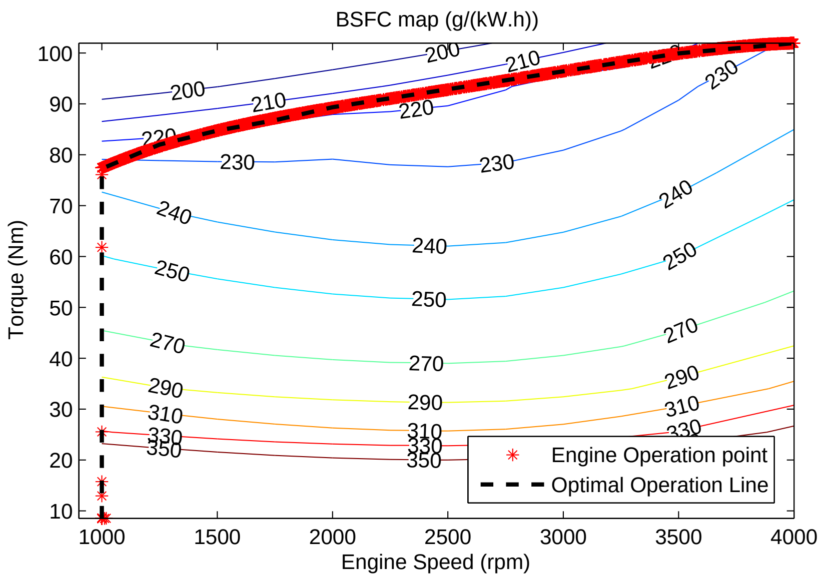

Figure 6.13: Engine operation points on the BSFC map. The optimum operation line is shown in dashed black line.

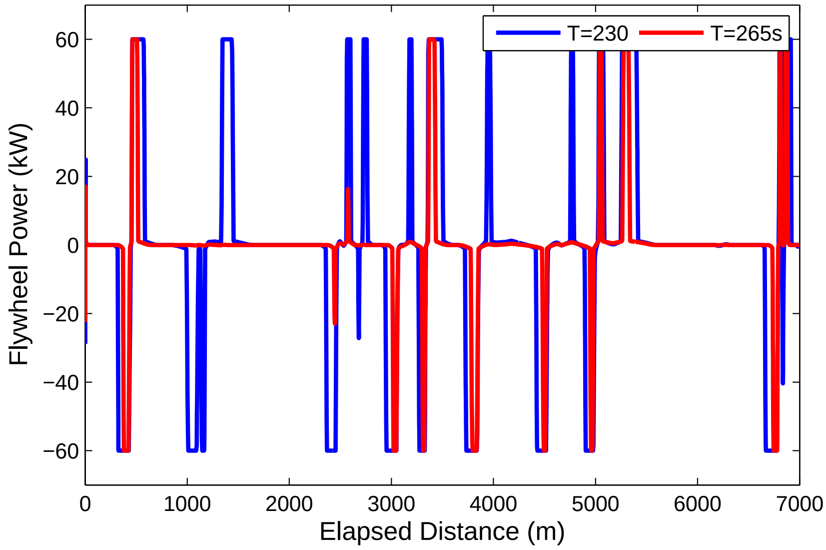

Figure 6.14: Flywheel power for different journey times for the vehicle with no battery.

It will be shown that the flywheel can provide a significant amount of fuel savings, especially when the journey times are low, and when the vehicle is required to complete the route aggressively. In this case, there will be many braking regions that will regenerate energy back into the flywheel that can then be quickly redeployed to then speed up the vehicle. For longer journey times the vehicle can essentially complete the circuit without braking; in this case the vehicle can go around the corners with a low enough speed that tyres will have enough lateral grip to accommodate the centripetal forces involved. These ideas are illustrated in Figure 6.14 and Figure 6.15, which show the power and energy stored in the flywheel respectively for a vehicle with engine and flywheel only. For a journey time of 230 s, the flywheel consistently reaches high levels of stored energy. However when the journey time is increased to 265 s the braking regions become less frequent, and lighter, resulting in fewer opportunities to scavenge energy into the flywheel. The flywheel self-discharging power losses are evident in Figure 6.15. These are low-gradient energy dissipation regions when there is no power being delivered from the flywheel; see the flywheel power at approximately 4500 m for example.

In order to analyse the battery management strategy, the vehicle with ICE and

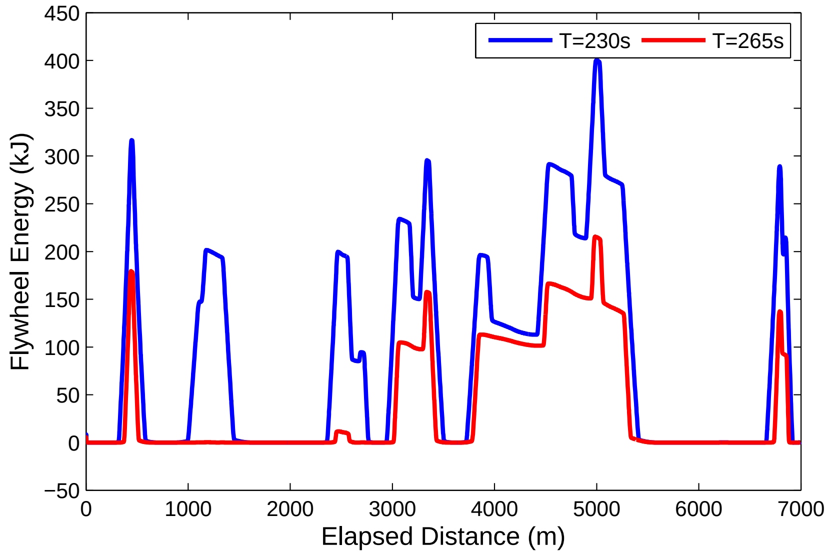

Figure 6.15: Flywheel energy for different journey times for the vehicle with no battery.

battery was considered. The optimal strategy for a journey time of 250 s is illustrated in Figure 6.16.

The state of charge of the battery at the start and end of the lap is constrained to be 60 %. The battery is discharged in the acceleration regions for power assist and is charged in braking phases by electric motor energy regeneration. It is evident that unlike the flywheel, the discharge does not happen in a *bang-bang* fashion. This is because the power losses in the battery are proportional to square of the current. This results in the optimal controller spreading the power delivery over a longer time period in order to maintain high efficiency. In the charging mode the maximum power available is absorbed as the braking phase has a short span so for maximum energy regeneration, all the power available from the wheels is extracted. The electric motor efficiency losses result in the power delivered to the battery being smaller than the power from the wheels in regeneration phase. This efficiency loss is also evident in the power assist mode when part of battery power is lost on its way to the wheels and explains why the electric motor power is higher in braking and lower in acceleration.

In a final study, the minimum fuel problem was solved for a range of arrival times with the full combination of auxiliary storage systems available; the results

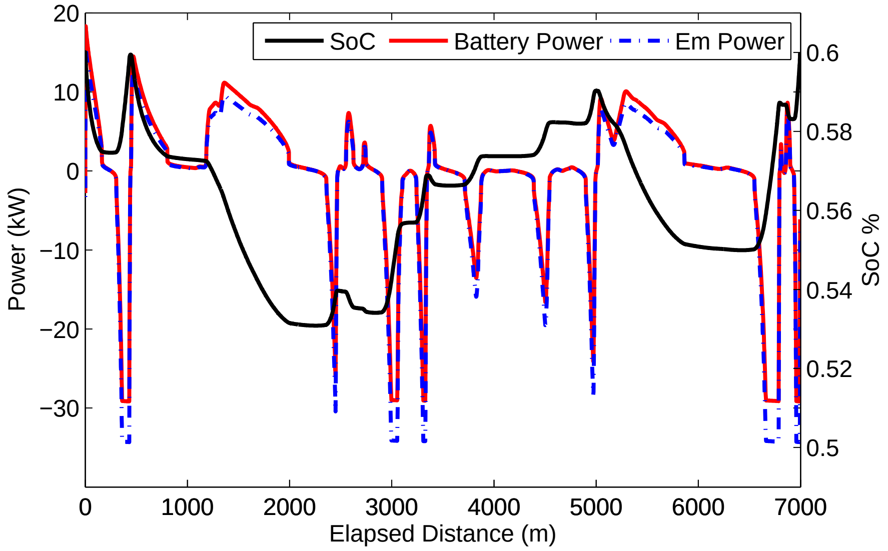

Figure 6.16: The battery power and state of charge along with the power at the electric motor shaft. The journey time was 250 s.

are summarised in Figure 6.17. As one would expect, at lower journey times, when the vehicle has to be driven more aggressively, the fuel consumption increases, since at higher engine powers more fuel is consumed. Furthermore, the main vehicle-related losses come from aerodynamic drag, which is proportional to the square of the vehicle speed. The vehicle with both the flywheel and battery performs best at very low journey times. However, as the journey time increases, the vehicle with engine and flywheel performs the best as it does not have to carry the additional 100 kg of battery load. With increasing journey times the fuel usage drops significantly and the benefit of using an axillary storage system also diminishes as the amount of energy regeneration decreases. The minimum journey times possible for the vehicle with engine only, battery only, flywheel only and both battery and flywheel were 230.09 s, 225.76 s, 219.67 s and 217.35 s respectively. This highlights the power boost capability of the flywheel in aggressive manoeuvrings.

6.5.1 Some computational details

In this study the optimal control solver was initialised with 100 mesh segments. The number of collocation points was allowed to vary between 4 to 10 per mesh segment.

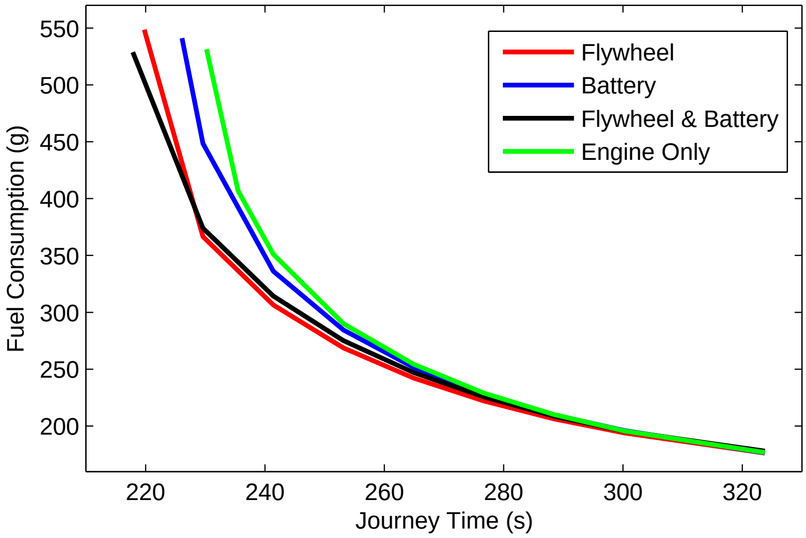

Figure 6.17: Fuel usage against journey time for the case with engine and flywheel (red), engine and battery (blue), full combination (black) and engine only (green).

The tolerance for error across all meshes was at 10^{-3} and IPOPT tolerance was set to 10^{-7}. For the vehicle with both battery and flywheel, when arrival time was set to 240 s, the error tolerance was reached after 18 mesh adaptation iterations. The final mesh had a total of 367 mesh segments and 2861 collocation points. The whole problem took under 2 hours to solve on an 8 core 3.5 GHz computer. For the engine-only case the error tolerance was reached with a similar number of collocations and meshes. However, the total solution time was only 20 minutes as only 10 mesh adaptation iterations were required and each iteration was faster to complete. The solution time is sensitive to the problem set-up. For example, increasing the size of ϱ in (6.55) slows down convergence as the problem becomes 'less continuous'. Avoiding the use of look-up tables in the cost function speeds up the algorithm significantly. In general, even though the direct pseudo-spectral method shows great robustness, careful attention is required in problem set-up when fast solution speeds are needed.

6.5.2 Optimality

As with any gradient-based optimisation method, the optimal solution obtained from the algorithm might be a local rather than the global minimum. It is therefore

Initial Guess	Fuel (g)	Initial Guess	Fuel (g)
Linear 10%	DNC	Linear 20%	DNC
Linear 30%	320.53	Linear 40%	320.52
Linear 50%	320.53	Linear 60%	320.59
Linear 70%	320.52	Linear 80%	DNC
Random 1	320.51	Random 2	320.59
Random 3	320.59	Random 4	320.60
Random 5	320.59	Random 5	320.59
Random 6	320.59	Random 7	320.59
Random 8	320.58	Random 9	320.57
Battery min	320.56	Battery max	320.53

Table 6.2: Optimal fuel usage for different initial guesses. DNC stands for 'Did not converge'.

important to quantify the sensitivity of the optimal solution to the initial guess. In this study, the problem was initialised using a set of sensible constant values across the entire solution space. A series of cases were then considered to examine whether the solution obtained can be trusted to be a global optimum. In all the cases, the vehicle was equipped with a battery and flywheel and the arrival time was set to 240 s.

Firstly, the initial guesses were chosen as constant numbers by linearly slicing the states and controls bounds such that $\boldsymbol{x}_0 = \boldsymbol{x}_{min} + (\boldsymbol{x}_{max} - \boldsymbol{x}_{min})w$ and $\boldsymbol{u}_0 = \boldsymbol{u}_{min} + (\boldsymbol{u}_{max} - \boldsymbol{u}_{min})w$, where w is the weighting percentage. The optimal fuel usage for these cases are shown in Table 6.2 as 'Linear x%'. For the 10%, 20% and 80% cases, the solution did not converge (marked as DNC). However, for all the other cases the solutions were essentially identical with the differences being less than the mesh tolerance error of 10^{-3}. Even though the set of initial guesses were vastly different, the optimiser showed great robustness and managed to converge to the same optimal solution.

Rather than slicing the bounds linearly, a random value between 20% to 80% of the total bound of each state and control was then chosen as the initial guess. The experiment was repeated 10 times and the results are shown as 'Random n' in Table 6.2. All the runs converged successfully to the same solution. Finally, the problem was initialised with lowest charging current and SOC and then with highest discharge current and SOC for the battery. The results are shown as 'Battery min' and 'Battery max'. Identical solutions were obtained once more, demonstrating that the optimal control problem has a great radius of convergence when solved using a direct pseudo-spectral method.

To ensure the same minimal cost was not obtained using different trajectories the cases with the highest cost difference were selected (Case Random 1 and Random 2). The state and controls for the two cases were then compared. The worst case difference was found to be in the Engine Torque, T_e, as plotted in Figure 6.18. As we can see, the trajectories are nearly identical making us confident that the obtained solution is indeed globally optimal.

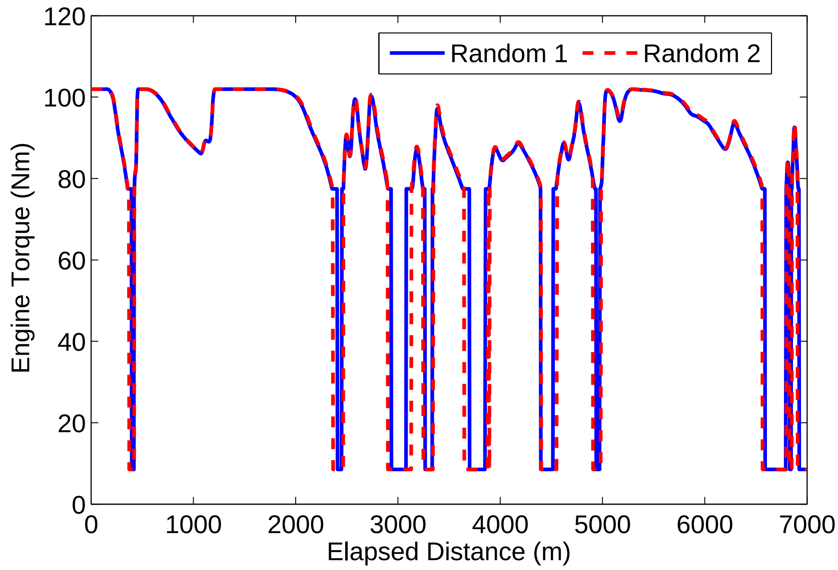

Figure 6.18: Optimal engine torques for the Random 1(sold blue) and Random 2(dashed red) initial guesses.

6.6 Conclusions

We have presented a method of evaluating the fuel consumption performance of hybrid vehicles. In our approach, rather than utilising standardised driving cycles, we make use of a specific route and then drive the vehicle so that we reach the destination within a given time, while controlling the vehicle in order to minimise the fuel consumption. In this framework driving aggressiveness can be systematically controlled by changing the arrival time. Another thrust of this work was to quantify the effectiveness of flywheels and batteries in reducing fuel consumption. Finally, the minimum fuel usage strategy for the hybrid vehicle over a three-dimensional route was found and the effects of different combinations of auxiliary storage systems were studied. It was demonstrated that flywheels can help to achieve significant fuel reductions in aggressive driving.

Chapter 7

Nonlinear Stability

7.1 Introduction

STUDIES of vehicular lateral dynamics allow us to determine its directional stability. A vehicle is directionally stable if it is able to return to its prior steady-state condition after experiencing a disturbance. The disturbance might be a wind gust, external forces from the road due to an uneven surface, or a sudden small change to the steering [90]. Yaw stability is extremely important for vehicle safety; a lack of yaw stability increases the propensity to roll-over [91]. In racing, where the vehicle is often under grip limited conditions, a loss of yaw stability will result in the vehicle spinning off the track. It is therefore of value to understand the factors that have the most impact on vehicular stability and performance. In this chapter, we will propose a method of finding the direction and extent to which a vehicle is vulnerable to instability. This will be done using a region of attraction (RoA) analysis, which finds the set of initial conditions for which the vehicle can return to its stable equilibrium.

As with many other applications, linear stability theory has been utilised extensively in vehicle dynamics [60, 87] and the eigenvalues of the linearised system can be used to determine the stability of an equilibrium operating condition. This method is simple and can be used to deduce many important system properties, but linear models can only be used to determine local performance and stability properties. For example, it is shown in [87] that a linearised model results in a straight handling line, whereas the nonlinear model shows a curved line. Linear theory predicts that the gradient of the steering against lateral acceleration plot, known as the understeer gradient, is constant [87]. If this gradient is positive, the car exhibits understeering and can never become unstable. However, if it is positive, it will oversteer and will become unstable above a certain value of forward velocity. A nonlinear model, on the other hand, predicts that the steering behaviour can change as the lateral acceleration increases. It was therefore suggested that handling diagrams should be used to predict the steering behaviour, with isoclines employed as a means of approximating the system trajectories and thereby estimating the domain of attraction.

Lyapunov's second method was first used in [5] to estimate the RoA. Two candidate functions were considered: the Lyapunov function coming from solving the Lyapunov equation, and a modified kinetic energy function. The RoA estimate was then found by finding manually the largest level set of the candidate Lyapunov function that fits within the region, where the time derivative of the Lyapunov function is negative. The estimated RoA was then compared with the results of the numerical integration of the dynamics equations from a chosen set of initial conditions. The RoA estimates were reasonable, but conservative. This method is extended in [92], where the use of tangency points is suggested as a way of finding the largest level sets of Lyapunov functions automatically. The use of trajectory reversal is also suggested, which relies on forward and backward integration to find the simulation stability boundary. A better RoA estimate is found in [6], using an examination of the phase plots in combination with heuristic arguments. This estimate is still conservative and so the use of Lyapunov exponents is proposed as a way of finding an accurate estimate of the RoA [93].

Regions of attraction can be estimated with a Lyapunov function determined by sum-of-squares (SOS) programming techniques. Before the advent of SOS programming, the determination of a Lyapunov function was neither trivial nor systematic. It is shown in [10] that a Lyapunov function search can be written as SOS constraints that can then be cast as a convex semi-definite program (SDP). Many algorithms have been proposed subsequently that estimate the RoA using SOS programming [10, 94–97].

In [14] we demonstrated how SOS programming can be used to improve the RoA estimates published in [5, 6, 92]; one issue with the models used is the employment of a cubic polynomial to describe the nonlinear tyre behaviour as a function of the slip angle. In this research, we will show that rational functions can capture the nonlinearities of the tyre characteristic more effectively than polynomials. We will then extend the algorithm proposed in [94] to utilise systems over rational vector fields. Heuristics that can improve the performance of the RoA algorithm are also presented. The influence of driving conditions and vehicle parameters on the region of stability are investigated and compared with the predictions of linear theory. Using this method, we can find regions of attraction without resorting to numerical integration and mathematical guarantees can be obtained that can be used for safety verification and validation.

The rest of this chapter is structured as follows. In Section 7.2 the mathematical model of the vehicle and the tyre model approximation are presented. SOS programming is briefly introduced in Section 7.3 with the RoA estimation algorithm presented in Section 7.4. Finally, the results are presented in Section 7.5 with the conclusions given in Section 7.6.

7.2 The Mathematical Model

The single track 'bicycle' model is employed here to describe the lateral dynamics of the vehicle. This model benefits from its simplicity, whilst also capturing important vehicular dynamic properties. This simplicity makes it suitable for the stability analysis to follow.

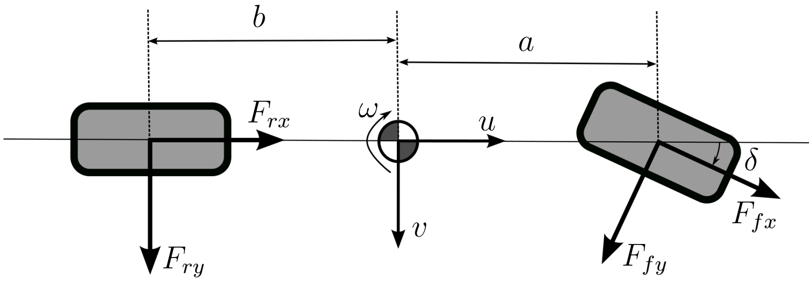

Figure 7.1: Vehicle bicycle model. The F_{ij}'s represent the tyre forces, δ is the steering angle, u and v are the longitudinal and lateral velocity, and a and b are the distances between the mass centre and the front and rear axles respectively.

7.2.1 Dynamics

The standard and well known 'bicycle' model has longitudinal, lateral and yaw freedoms and is given by

$$m(\dot{u} - \omega v) = F_{rx} + F_{fx} \cos\delta - F_{fy} \sin\delta \tag{7.1}$$

$$m(\dot{v} + \omega u) = F_{ry} - F_{fx} \sin\delta + F_{fy} \cos\delta \tag{7.2}$$

$$I\dot{\omega} = a(F_{fy} \cos\delta + F_{fx} \sin\delta) - bF_{ry}, \tag{7.3}$$

where m is the vehicle mass, I its yaw moment of inertia and ω its yaw rate. The remainder of the quantities are shown in Figure 7.1. In this study we assume that the vehicle has a constant longitudinal velocity, which can be achieved by cruise control. With this assumption, there will be no longitudinal acceleration and (7.1) can be ignored. Assuming the steering angle is small and fixed and ignoring the effects of longitudinal tyre forces, the lateral dynamics can be described by (see also [87])

$$m(\dot{v} + \omega u_0) = F_{fy} + F_{ry} \tag{7.4}$$

$$I\dot{\omega} = aF_{fy} - bF_{ry}, \tag{7.5}$$

where u_0 is the (constant) forward speed.

7.2.2 Tyre Forces

The lateral tyre slip angle α is given by

$$\tan\alpha = -\frac{v_w}{u_w}, \tag{7.6}$$

where u_w and v_w are the absolute speed components of the wheel centre in a wheel-fixed coordinate system. The front and rear tyre lateral slip angles are given by

$$\alpha_f = \arctan\left(\frac{\cos\delta(\omega a + v) - \sin\delta u}{\cos\delta u + \sin\delta(\omega a + v)}\right) \tag{7.7}$$

$$\alpha_r = \arctan\left(\frac{v - \omega b}{u}\right). \tag{7.8}$$

Expression (7.7) can be simplified by using the identity $\arctan(\frac{x+y}{1-xy}) = \arctan(x) + \arctan(y)$ to give

$$\alpha_f = \arctan\left(\frac{\omega a + v}{u}\right) - \delta. \tag{7.9}$$

In this study we assume that the vehicle side-slip angle is small, i.e. the longitudinal velocity is large compared to the lateral velocity. Therefore, we can approximate the tyre slip angles by linearising (7.7) and (7.8) to obtain

$$\alpha_f = \frac{\omega a + v}{u} - \delta \tag{7.10}$$

$$\alpha_r = \frac{v - \omega b}{u}. \tag{7.11}$$

Tyre longitudinal and lateral forces are typically nonlinear functions of the wheel loads and camber, and the longitudinal and lateral tyre slips. We make use of the simplified 'Magic Formula' [87] tyre model described in [98] with the parameters for the front and rear tyre corresponding to that of a dry road. The tyre model needs to be approximated using either polynomials or rational functions for use in the SOS programs described later. The front tyre curve from [98] is shown in dots in Figure 7.2. The polynomial and rational approximations are shown on the same plot. The polynomial approximation is given by

$$F_{iy} = \sum_{k=0}^{m} p_{ik}\alpha_i^k, \quad i = f, r. \tag{7.12}$$

It is evident in Figure 7.2 that the fitting error decreases as the order of the polynomial is increased. However, even with a high-degree polynomials, it is difficult to approximate the cornering force accurately. On the other hand, the rational approximation given below fits the 'Magic Formula' model quite accurately.

$$F_{iy} = \frac{p_{i1}\alpha_i^3 + p_{i2}\alpha_i^2 + p_{i3}\alpha_i}{\alpha_i^4 + q_{i1}\alpha_i^2 + q_{i2}}, \quad i = f, r. \tag{7.13}$$

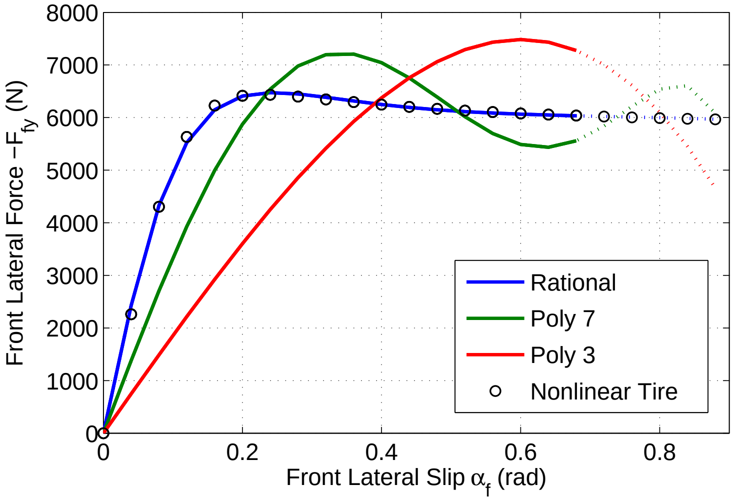

Figure 7.2: Front tyre cornering force approximation. The 'Magic Formula' data points are shown in black circles and the rational function fit of type (7.13) is shown in blue. The 3rd and 7th order polynomial fits (7.12) are given in red and green respectively.

Also, increasing the polynomial order leads to over fitting complications, where large errors can appear between the sample points. Furthermore, polynomial fitting is sensitive to the range of the data points chosen and extrapolating outside the data range degrades badly; this is because any $p(x)$ tends to $\pm\infty$ as x goes to $\pm\infty$. In contrast, rational functions can have asymptotes and this makes them more suitable for representing tyre saturation.

The lateral dynamics model used in the study is assembled by substituting (7.10) and (7.11) into the tyre approximation fits (7.12) or (7.13), which can then substituted in (7.4) and (7.5) to obtain the two ordinary differential equations (ODE) that describe the lateral dynamics. To validate the approximations made in the slip angles and tyre forces we compare the phase portrait of the approximated systems against the full nonlinear model. The phase portrait for the model with 7th order polynomial tyre model approximation is shown in Figure 7.3. The full nonlinear model, which has the slip angles described by (7.9) and (7.7), and uses the 'Magic Formula' tyre model is shown in the dotted lines. We can see in Figure 7.3 that the approximation shifts the equilibrium points (shown in magenta) away from the actual equilibriums

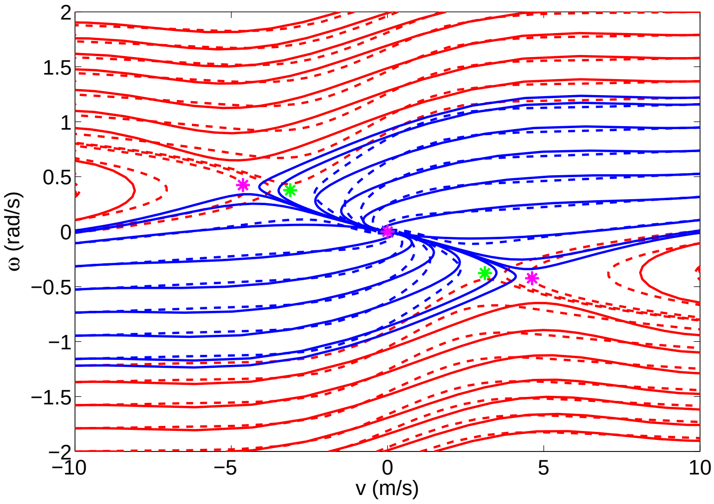

Figure 7.3: Phase portrait for the vehicle model with a 7th order polynomial tyre force approximation. The dashed lines show the fully nonlinear vehicle trajectories. Equilibrium points for the original system are shown in green, while those for the 7th order polynomial tyre are shown in purple.

(shown in green). This causes the region of stability to appear larger. The trajectory paths also become distorted, because of fluctuations in the polynomial tyre model.

The vehicle model with the rational tyre force approximation of type (7.13) shows much better agreement with the fully nonlinear model as demonstrated in Figure 7.4. The saddle equilibriums are now much closer to that of original system and the trajectories are almost identical. We will therefore make use of the rational function approximation in the rest of this study for estimating RoA. The rational function fitting can be done using a nonlinear least squares algorithm with an appropriate initial guess. The *lsqcurvefit* function in MATLAB$^{\mathrm{TM}}$ was used here to find the numerator and denominator polynomial coefficients p_i's and q_i's that resulted in the minimum fitting error.

7.3 Sum-of-Squares Programming

The sum-of-squares techniques required here are summarized in the next two sections (Sections 7.3 and 7.4). Our summary makes extensive use of [99] and the references therein.

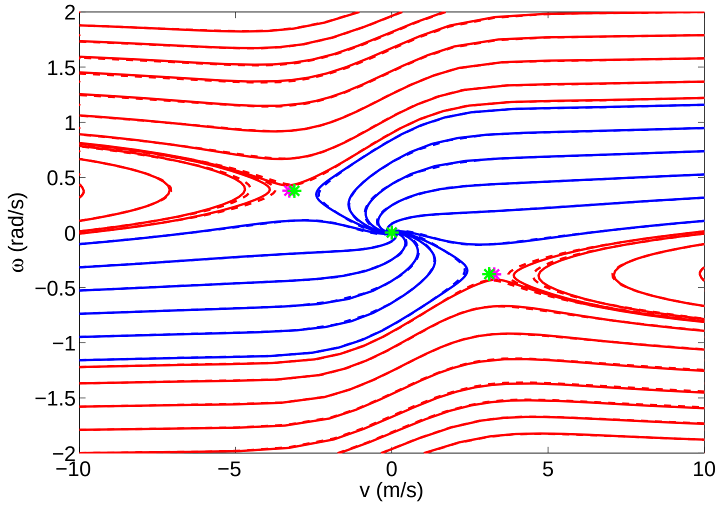

Figure 7.4: Phase portrait for the vehicle model with rational function tyre force approximation. The dashed lines show the fully nonlinear vehicle trajectories. Equilibrium points for the original system are shown in green, with the model with the rational function tyre approximation shown in purple.

Sum of squares (SOS) theory relates to multivariate polynomials $p(\boldsymbol{x})$ in n real variables, in which $\boldsymbol{x}$ is a vector with components $x_1, \cdots, x_n$. A multivariate polynomial $p(\boldsymbol{x})$ is said to be sum of squares (SOS) if there exist polynomials $f_i(\boldsymbol{x})$, such that

$$p(\boldsymbol{x}) = \sum_{i=1}^{M} f_i^2(\boldsymbol{x}). \tag{7.14}$$

The set of SOS polynomials is denoted $\Sigma[x]$. If $p(\boldsymbol{x})$ is SOS, then it is also positive semi-definite, i.e. $p(\boldsymbol{x}) \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$. If $p(\boldsymbol{x})$ is expressed as a linear combination of monomials, then its degree is the degree of the highest order monomial. For example

$$p(\boldsymbol{x}) = x_1^2 + 3x_2^4 + 7x_1x_2^5 - x_1^3x_2^2 + 5x_6^2$$

has degree 6. If $p(\boldsymbol{x})$ is a SOS polynomial then there exists a decomposition such that $p(\boldsymbol{x})$ can be written as

$$p(\boldsymbol{x}) = Z^T(\boldsymbol{x}) Q_0 Z(\boldsymbol{x}) \tag{7.15}$$

where $Z(\boldsymbol{x})$ is a vector of monomials in $\boldsymbol{x}$ up to half the degree of $p(\boldsymbol{x})$ and $Q_0 \geq 0$ is a *Gram* matrix [10]; this representation is not unique. If matrices M_i, $i = 1, \cdots, N$, span the null space of the representation of $p(\boldsymbol{x})$ in (7.15), then

$$p(\boldsymbol{x}) = Z^T(\boldsymbol{x})(Q_0 + \sum_{i=1}^{N} \lambda_i M_i) Z(\boldsymbol{x}). \tag{7.16}$$

Thus $p(\boldsymbol{x})$ is a SOS if and only if there exist λ_i, $i = 1, \cdots, N$, such that

$$Q = Q_0 + \sum_{i=1}^{N} \lambda_i M_i \geq 0,$$

which is a linear matrix inequality (LMI) feasibility problem. Not all non-negative polynomials are SOS, but checking if a SOS decomposition exists is computationally tractable with polynomial complexity. It is this feature of SOS techniques that assist us in finding Lyapunov functions for non-linear systems.

7.3.1 Generalised $\mathcal{S}$-procedure

In Lyapunov stability theory one is concerned with the sign definiteness of scalar functions and their derivatives. To this end we consider $p_0(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ such that

$$p_0(\boldsymbol{x}) \geq 0 \tag{7.17}$$

for $\boldsymbol{x} \in \mathcal{D}$. In order to restrict the domain of $\boldsymbol{x}$, one might require

$$p_i(\boldsymbol{x}) \geq 0, \cdots, p_m(\boldsymbol{x}) \geq 0, \tag{7.18}$$

where $p_0, p_1, \cdots, p_m : \mathbb{R}^n \to \mathbb{R}$. The inequalities in (7.17) and (7.18) are equivalent to the following set-containment condition

$$\bigcap_{i=1}^{m} \{\boldsymbol{x} \in \mathbb{R}^n | p_i(x) \geq 0\} \subseteq \{\boldsymbol{x} \in \mathbb{R}^n | p_0(\boldsymbol{x}) \geq 0\}. \tag{7.19}$$

A possibly conservative test for (7.17) and (7.18) is the existence of positive semi-definite functions $q_1(\boldsymbol{x}), \cdots, q_m(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ such that

$$p_0(\boldsymbol{x}) - \sum_{i=1}^{m} q_i(\boldsymbol{x}) p_i(\boldsymbol{x}) \geq 0 \text{ for all } \boldsymbol{x} \in \mathbb{R}^n. \tag{7.20}$$

To see how this works consider a point $\boldsymbol{x}$ such that $p_i(\boldsymbol{x}) \geq 0, \cdots, p_m(\boldsymbol{x}) \geq 0$. This means that $q_i(\boldsymbol{x}) p_i(\boldsymbol{x}) \geq 0$ and therefore that $p_0(\boldsymbol{x}) \geq 0$.

7.4 Regions of Attraction Estimation

Consider an autonomous non-linear system of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)), \tag{7.21}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector. The locally Lipschitz function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ describes the system dynamics; we will assume without loss of generality that $\boldsymbol{f}(\mathbf{0}) = \mathbf{0}$ is an equilibrium point of (7.21). If $\boldsymbol{\xi}(\boldsymbol{x}_0, t)$ is a solution to (7.21) at t, then the initial condition is $\boldsymbol{\xi}(\boldsymbol{x}_0, 0) = \boldsymbol{x}_0$ and the region of attraction (RoA) is the set $\{\boldsymbol{x}_0 \in \mathbb{R}^n : \lim_{t\to\infty} \boldsymbol{\xi}(\boldsymbol{x}_0, t) = \mathbf{0}\}$. The set $\mathcal{M}$ is invariant under the flow $\boldsymbol{f}(\boldsymbol{x}(t))$ if $\boldsymbol{x}_0 \in \mathcal{M}$ implies $\boldsymbol{\xi}(\boldsymbol{x}_0, t) \in \mathcal{M}$ for all $t > 0$.

The stability of the equilibrium at $\boldsymbol{f}(\mathbf{0}) = \mathbf{0}$ can be established by constructing a Lyapunov function with the following properties [100].

Theorem 7.1 *The origin of the system* (7.21) *is locally stable if there exists a continuously differentiable function* $V : \mathbb{R}^n \to \mathbb{R}$ *such that the following two conditions hold:*

(i) If $V(\boldsymbol{x})$ *is locally positive definite i.e.* $V(\boldsymbol{x}) > 0\ \forall \boldsymbol{x} \in \mathcal{D} \setminus \{\mathbf{0}\}$ *with* $V(\mathbf{0}) = 0$

(ii) $\dot{V}(\boldsymbol{x})$ *is negative semi-definite in* $\mathcal{D}$ *i.e.* $\frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) \leq 0\ \forall x \in \mathcal{D}$

where $\mathcal{D}$ *is a neighbourhood of the origin. Furthermore, if* $\frac{\partial V}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) < 0$, *the equilibrium is locally asymptotically stable.*

Lyapunov functions for systems with polynomial vector fields can be constructed by relaxing conditions (i) and (ii) in Theorem 7.1 to SOS constraints [95].

Suppose $\varphi_i(\boldsymbol{x})$ are positive definite polynomials defined by

$$\varphi_i(\boldsymbol{x}) = \sum_{j=1}^{n} \epsilon_{ij} x_j^2 \tag{7.22}$$

in which ϵ_{ij} are small positive real numbers. Define the neighbourhood of the origin to be the set $\mathcal{D} = \{\boldsymbol{x} \in \mathbb{R}^n \mid g(\boldsymbol{x}) \leq 0\}$. Then we can establish the local asymptotic stability of the equilibrium at the origin of (7.21) using the following SOS program.

Program 7.1 *Given an ODE system* (7.21) *with polynomial vector fields, the local asymptomatic stability of the equilibrium at the origin can be checked if we could find* $V(\boldsymbol{x})$, *such that* $V(\mathbf{0}) = 0$, *and* $q(\boldsymbol{x}) \in \Sigma[x]$ *such that:*

$$V(\boldsymbol{x}) - \varphi_1(\boldsymbol{x}) \in \Sigma[x] \tag{7.23}$$

$$-\frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) - \varphi_2(\boldsymbol{x}) + q(\boldsymbol{x})g(\boldsymbol{x}) \in \Sigma[x]. \tag{7.24}$$

The first constraint enforces $V(\boldsymbol{x}) > 0 \; \forall \boldsymbol{x} \neq 0$. Constraint (7.24) enforces condition (*ii*) in Theorem 7.1, and comes from applying the generalised $\mathcal{S}$-procedure.

The system of interest in this study is of rational form

$$\dot{\boldsymbol{x}} = \frac{\boldsymbol{n}(\boldsymbol{x})}{\boldsymbol{d}(\boldsymbol{x})} \tag{7.25}$$

with $\boldsymbol{n}(\boldsymbol{x})$ and $\boldsymbol{d}(\boldsymbol{x})$ polynomials and $\boldsymbol{d}(\boldsymbol{x}) > \mathbf{0}$ in $\mathcal{D}$. The stability of the origin can be checked with Program 7.1 if constraint (7.24) is multiplied by $\boldsymbol{d}(\boldsymbol{x})$ so that it admits a SOS decomposition. The following modified program is obtained [95].

Program 7.2 *Given a rational system of form* (7.25), *then the local asymptotic stability of the origin can be established by finding* $V(\boldsymbol{x})$, *such that* $V(\mathbf{0}) = 0$, *and* $q \in \Sigma[x]$ *such that:*

$$V(\boldsymbol{x}) - \varphi_1(\boldsymbol{x}) \in \Sigma[x] \tag{7.26}$$

$$-\frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{n}(\boldsymbol{x}) - \varphi_2(\boldsymbol{x})\boldsymbol{d}(\boldsymbol{x}) + q(\boldsymbol{x})g(\boldsymbol{x})\boldsymbol{d}(\boldsymbol{x}) \in \Sigma[x]. \tag{7.27}$$

Once the stability of an equilibrium is established, we can then attempt to estimate its region of attraction. The following result characterises some invariant subsets of the RoA.

Lemma 7.1 *[99] Suppose* $\gamma > 0$ *and suppose also that there exists a continuously differentiable function* $V : \mathbb{R}^n \to \mathbb{R}$ *such that*

(i) $V(\mathbf{0}) = 0$, $V(\boldsymbol{x}) > 0$ *for all nonzero* $x \in \mathbb{R}^n$*;*

(ii) $\mathcal{M}_{V,\gamma} := \{\boldsymbol{x} \in \mathbb{R}^n : V(\boldsymbol{x}) \leq \gamma\}$ *is bounded;*

(iii) $\mathcal{M}_{V,\gamma}\backslash\{\mathbf{0}\} \subset \{x \in \mathbb{R}^n : \frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) < 0\}$.

Then, for all $\boldsymbol{x}_0 \in \mathcal{M}_{V,\gamma}$*, the solution* $\boldsymbol{\xi}(\boldsymbol{x}_0, t)$ *of (7.21) exists on* $[0, \infty)$ *with* $\boldsymbol{\xi}(\boldsymbol{x}_0, t) \in \mathcal{M}_{V,\gamma}$ *for all* $t \geq 0$*, and with* $\lim_{t\to\infty} \boldsymbol{\xi}(\boldsymbol{x}_0, t) = \mathbf{0}$.

If the conditions of Lemma 7.1 are met, $\mathcal{M}_{V,\gamma}$ is an invariant subset of the RoA for the equilibrium $\boldsymbol{f}(\mathbf{0}) = 0$. Both sets in (iii) are described by inequalities, and the generalized $\mathcal{S}$-Procedure can be used to establish containment. If $\varphi(\boldsymbol{x})$ is positive definite, $q(\boldsymbol{x})$ is positive semidefinite and

$$q(\boldsymbol{x})(V(\boldsymbol{x}) - \gamma) \geq (\varphi(\boldsymbol{x}) + \frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}))\ \forall \boldsymbol{x},$$

then (*iii*) holds. If $\boldsymbol{x} \neq \mathbf{0}$ satisfies $V(\boldsymbol{x}) \leq \gamma$, then the nonnegativity of $q(\boldsymbol{x})$ and the positivity of $\varphi(\boldsymbol{x})$ ensures $\frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x}) < 0$.

Following [94], a 'shape function' can be used to further refine the RoA estimate by varying $V(\boldsymbol{x})$. To this end a positive definite polynomial $s(\boldsymbol{x})$ is introduced with $s(\boldsymbol{x}) \leq \beta$ in the region of attraction. The aim of the optimisation process is to maximise the value of β such that containment is guaranteed. In the following program we will accommodate systems with rational vector fields of form (7.25).

Program 7.3 *The region of attraction around the equilibrium of rational system (7.25) can be estimated by finding* V*,* $q_1 \in \Sigma[x]$ *and* $q_2 \in \Sigma[x]$ *that maximise* β *such that:*

$$V(\boldsymbol{x}) - \varphi_1(\boldsymbol{x}) \in \Sigma[x], \tag{7.28}$$

$$q_1(\boldsymbol{x})(s(\boldsymbol{x}) - \beta) - (V(\boldsymbol{x}) - \gamma) \in \Sigma[x], \tag{7.29}$$

$$-\frac{\partial V}{\partial \boldsymbol{x}}\boldsymbol{n}(\boldsymbol{x}) - \varphi_2(\boldsymbol{x})\boldsymbol{d}(\boldsymbol{x}) + q_2(\boldsymbol{x})\boldsymbol{d}(\boldsymbol{x})(V(\boldsymbol{x}) - \gamma) \in \Sigma[x] \tag{7.30}$$

Condition (7.28) ensures that Lemma 7.1 (*i*) is satisfied, (7.29) ensures that

$$\mathcal{M}_{s,\beta} \subseteq \mathcal{M}_{V,\gamma},$$

while (7.30) enforces Lemma 7.1 (*iii*). The decision variables are $V(\boldsymbol{x})$, $q_1(\boldsymbol{x})$ and $q_2(\boldsymbol{x})$, which makes Program 7.3 bilinear due to the q_2V term. Program 7.3 can be solved using a local bilinear matrix inequality solver such as PENBMI [96]. It is common for PENBMI to converge to different solutions for different runs due to its built-in random initial point starter. Since PENBMI is not freely available, it is useful to have an alternative method of solving Program 7.3. Upon closer inspection one can see that this problem is quasi-convex [101] and can be solved using an iterative algorithm [94] such as the following

Algorithm 7.1 *Program 7.3 can be solved iteratively by repeating the following steps until β no longer improves.*

(i) γ step: Initialise V and maximise γ by finding q_2 such that (7.30) *holds;*

(ii) β step: Keep V, γ and q_2 fixed and maximise β by finding q_1 such that (7.29) *holds;*

(iii) V step: Keep γ, β, q_1 and q_2 fixed and find V such that (7.28)-(7.30) *hold.*

Constraint (7.30) can be solved using a bisection on γ until the difference between the lower and upper estimates of γ attains some specified threshold. The β step is treated likewise. Since $V(\boldsymbol{x}) = 0$, V is positive definite and $\boldsymbol{f}(\mathbf{0}) = \mathbf{0}$, it is clear that V, q_1 and q_2 need not contain constant terms. In the same way, it is important to not include 'unnecessary' monomials in the SOS program. In addition, if we require $s_1 - s_2 \geq 0$ where s_1 and s_2 are SOS polynomials, the degree of s_1 must be at least equal to the degree of s_2.

When implementing Program 7.3, we suggest that the following degree bounds are recognised

$$\deg(V) \geq \deg(\varphi_1) \tag{7.31}$$

$$\deg(q_1) + \deg(s) \geq \deg(V) \tag{7.32}$$

$$\deg(q_2) + \deg(d) \geq \deg(n) - 1. \tag{7.33}$$

In this application the φ_i are quadratic with 'small' coefficients. It is an interesting consequence of (7.33), that the degree of q_2 is dependent on the relative degree of the rational vector field in (7.25).

7.4.1 Scaling

When high degree monomials are used in the SOS program, the order of the polynomial coefficients can become vastly different. In order to avoid numerical ill conditioning, we need to scale the problem so that the optimisation decision variables are in a 'roughly spherical' space. This was done here by scaling state $\boldsymbol{x}$ so that $\boldsymbol{x} = \boldsymbol{x}_{sc}\bar{\boldsymbol{x}}$. The scaled dynamics are then given by $\dot{\bar{\boldsymbol{x}}} = \boldsymbol{f}(\bar{\boldsymbol{x}}/\boldsymbol{x}_{sc})$. Once the problem is solved, we can again use $\boldsymbol{x} = \boldsymbol{x}_{sc}\bar{\boldsymbol{x}}$ to go back to the original system's dimensions.

7.5 Results

As is well known, the straight-running stability of the simple bicycle model described in Section 7.2.1 can be studied with the help of an eigenvalue analysis. A linearisation of (7.4) and (7.5) around a straight-running equilibrium at speed u_0 is given by

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{C_f+C_r}{Mu_0} & \frac{aC_f-bC_r}{Mu_0} - u_0 \\ \frac{aC_f-bC_r}{I_z u_0} & \frac{a^2C_f+b^2C_r}{I_z u_0} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{7.34}$$

where $l = a + b$, with the tyre cornering stiffnesses given by

$$C_i = \frac{\partial F_i}{\partial \alpha_i}\bigg|_{\alpha_i=0}, \quad i = f, r. \tag{7.35}$$

The tyre cornering stiffnesses for the polynomial approximation are of the form $C_i = p_{i1}$, while those of a rational approximation of type (7.13) are $C_i = \frac{p_{i3}}{q_{i2}}$.

A well-known condition for the (linear) stability of the vehicle is obtained by examining the coefficients of the yaw rate differential equation and is given by [87]

$$C_f C_r l^2 - Mu_0^2(aC_f - bC_r) > 0. \tag{7.36}$$

Satisfaction of inequality (7.36) ensures that the system eigenvalues have negative real part.

The stability of the equilibrium of the nonlinear vehicle model given in (7.4) and (7.5) can also be checked using Lyapunov function based methods. We consider first the constant-speed straight-running case with the data given in Table 7.1. Program 7.2 was used to find a Lyapunov function which guarantees the local stability of the system for the neighbourhood region $\mathcal{D}$ of $x^T x \leq 1$. Writing this SOS program in SOSOPT [102] and solving the resulting SDP using SeDuMi [103] we obtain the following Lyapunov function that establishes the stability of the straight-running equilibrium.

$$V_{local} = 0.0746v^2 + 0.0715v\omega + 2.9155\omega^2.$$

We will now focus our attention on estimating the region of attraction around the stable equilibrium point; this will be done with using Algorithm 7.1. This algorithm requires an initial Lyapunov function for the γ step at the first iteration. We follow the suggestion made in [94], and initialise V with a quadratic Lyapunov function $V_{lin} = x^T P x$ derived from the linearised system and the associated Lyapunov equation

$$A^T P + PA = -Q; \tag{7.37}$$

$Q \geq 0$ was chosen to be the identity matrix. An important factor in the quality of the RoA estimate is the choice of the shaping function $s(\boldsymbol{x})$, which is used to enlarge the size of the invariant set $\mathcal{M}_{s,\beta}$. RoA estimates for different degrees of Lyapunov function when $s(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{x}$ are shown on Figure 7.5. The largest level set of $s(\boldsymbol{x})$ for Lyapunov functions of degrees two, four, six and eight are 0.639, 0.769, 0.772 and 0.774 respectively. Although the region $s(\boldsymbol{x}) \leq \beta$ is expanding with increasing degree of V, it is evident from Figure 7.5 that the RoA estimates for $\mathcal{M}_{V,\gamma}$ are not enlarging in all directions. Indeed, the size of the RoA estimate not increasing beyond V of fourth order, which motivates the use of $s(\boldsymbol{x})$ in determining the quality of the RoA estimate.

To expand V in the 'right direction', we can use $s(\boldsymbol{x}) = V_{lin}$, since it will be locally aligned with the RoA and thus might be used to obtain a better estimate. Figure 7.6 depicts the RoA estimates for $s(\boldsymbol{x}) = V_{lin}$. Once again, even though the region $s \leq \beta$ gets larger, a higher degree Lyapunov function does not necessarily guarantee an overall increase in the RoA estimate. Nonetheless, an improved estimate is obtained using $s(\boldsymbol{x}) = V_{lin}$ as compared with $s(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{x}$.

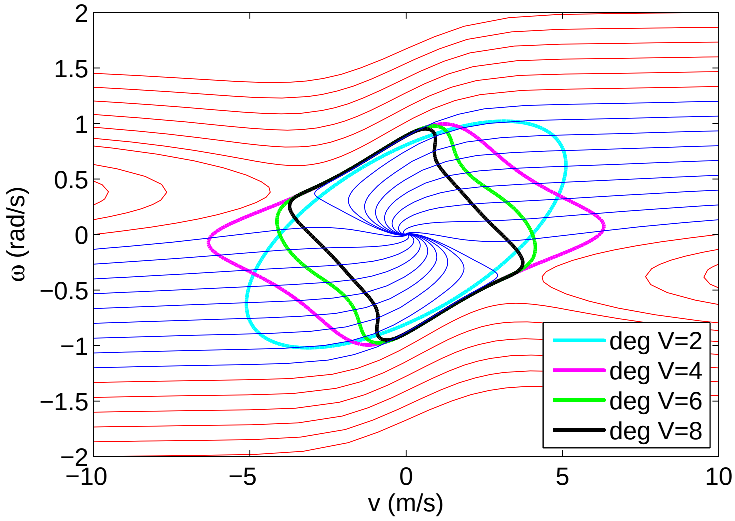

Figure 7.5: RoA estimate using $s(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{x}$ for different degree Lyapunov function. Phase portrait for the system with standard parameters are shown in blue and red.

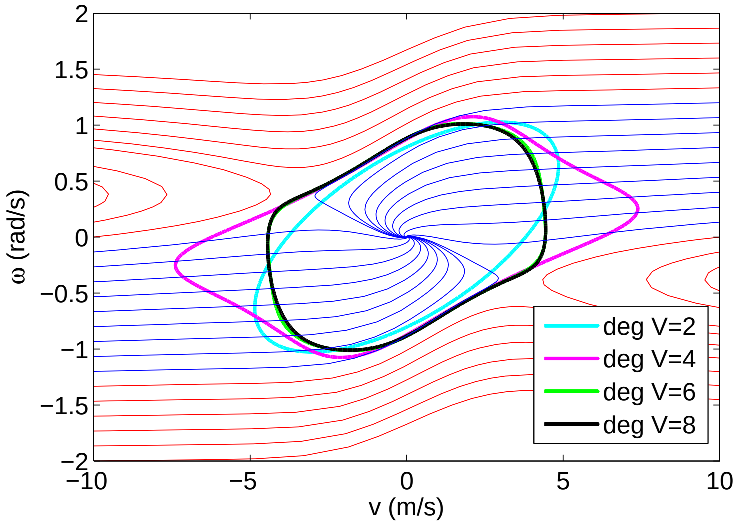

Figure 7.6: RoA estimate using $s(\boldsymbol{x}) = V_{lin}$ for different degrees of Lyapunov function.

Closer inspection of Figure 7.6 shows that as the degree of the Lyapunov function is increased, the $\mathcal{M}_{V,\gamma}$ estimate aligns more closely with the actual RoA. This leads one to the idea that Lyapunov functions of lower degree can be used as the shape function when finding higher order Lyapunov functions. This 'bootstrapping' process does indeed result in a significant improvement in the RoA estimate as shown in Figure 7.7. This technique requires running the program multiple times, but the

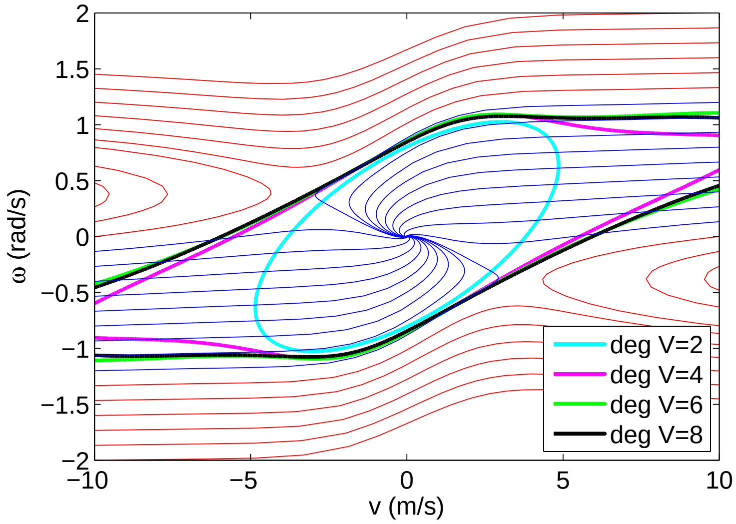

Figure 7.7: RoA estimate for the case where $s(\boldsymbol{x})$ is bootstrapped with the previous Lyapunov function.

additional iterations result in much better RoA estimates; this technique will used in the rest of this study.

Now that the RoA can be estimated with good quality, we can carry out a series of parametric studies to investigate their influence on the RoA. As one might expect, an important factor on the stability of the vehicle is the longitudinal velocity u_0. Figure 7.8 shows the RoA estimates in black along with the phase portraits for different forward velocities. As u_0 increases, the region of stability becomes smaller in yaw rate direction. This matches the linear stability results given by (7.36), where it is predicted that an increased forward speed will bring the vehicle closer to instability.

In the next study the rear tyre cornering stiffness (7.35) is varied by $\pm 10\%$ of its nominal value; see Figure 7.9. With a higher C_r value, the rear tyre will operate at reduced slip and the vehicle will tend to understeer, which has a stabilising effect on the vehicle and enlarges the region of attraction. The opposite occurs when C_r is reduced by 10%, which causes the car to exhibit oversteer behaviour that shrinks the RoA.

So far we have only considered the straight-running behaviour. Steady-state cornering requires a non-zero steering angle that causes the equilibrium of system move

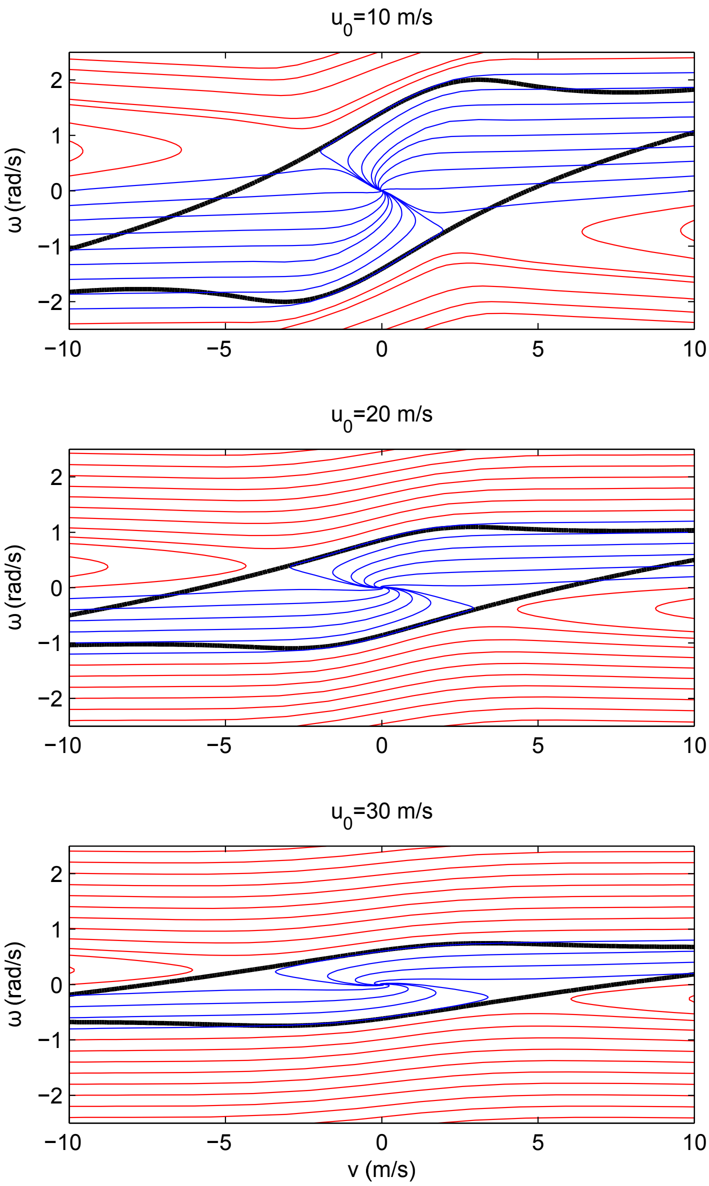

Figure 7.8: RoA estimates (black) for different values of forward velocity u_0 along with the respective phase portrait from numerical integration.

away from the origin of the phase space. For pre-specified δ and u_0, it is possible to solve (7.4) and (7.5) using a nonlinear solver. The state-vector is then translated so that the equilibria under study are at the origin of the transformed state space. Figure 7.10 shows the estimated ROAs for steering angles of ± 5 degrees. Under cor-

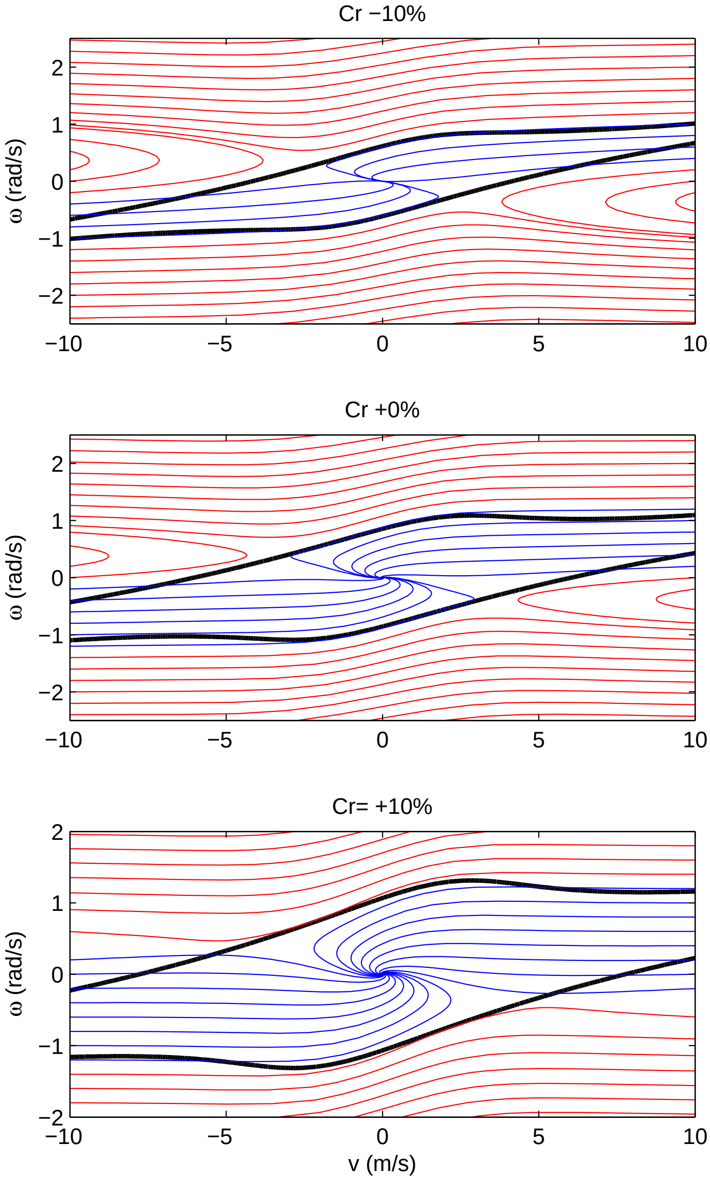

Figure 7.9: Influence of rear tyre cornering stiffness C_r on RoA. In the top plot C_r is reduced by 10% and in the bottom plot it is increased by 10%. The forward velocity is 20 m/s.

nering the phase plot is no longer symmetric. This makes it more difficult for the algorithm to approximate the RoA with the prior levels of accuracy. This is due to $s(\boldsymbol{x})$ hitting the stability boundary close to the equilibrium, thereby inhibiting the enlargement of the $\mathcal{M}_{V,\gamma}$ estimate. Figure 7.10 shows that the RoA shrinks in the

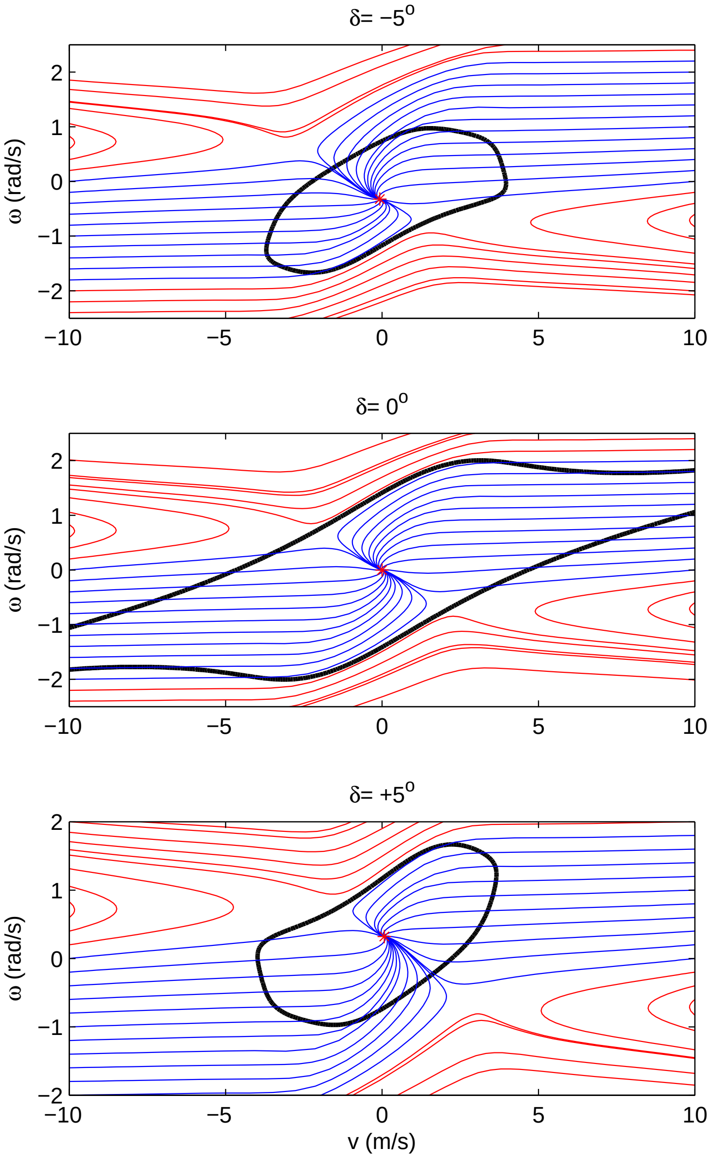

Figure 7.10: RoA for cornering equilibria at a forward speed of 10 m/s. The steering angle is at -5^o degrees in the top plot, 0^o degrees in the centre plot, and 5^o degrees in the bottom plot. The equilibrium points are shown in red stars.

yaw rate direction corresponding to the direction of steering. This is what one would expect intuitively, because a vehicle making a right-hand turn is more vulnerable to a clockwise yaw disturbance, whereas a left-turning vehicle is relatively exposed to an anti-clockwise disturbance.

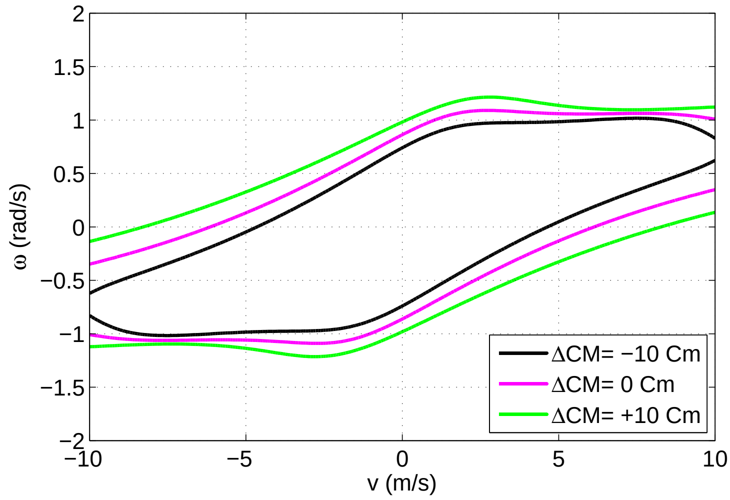

Figure 7.11: RoA estimates for the vehicle with the centre of mass shifted 10 cm backward (black) and 10 cm forward (green).

The effect of moving the centre of mass on the RoA is illustrated in Figure 7.11. As the mass is moved forwards the RoA enlarges, which coincides with the prediction made by (7.36); holding l constant and increasing a, while decreasing b, will bring the vehicle closer to instability. With the mass centre being more forwards, the front tyres have to provide larger proportion of the total lateral force. This in turn causes the slip angles on the front tyres exceed that of the rear, hence causing understeer.

Increasing the mass of the vehicle has a destabilising effect according to the linear stability criterion (7.36); the nonlinear stability region in Figure 7.12 supports this prediction.

The influence of the vehicle yaw moment of inertia I on the RoA is considered in Figure 7.13. It is interesting to note that I does affect the stability of the equilibrium, since it does not appear in (7.36), but it can have a significant impact on the RoA as shown in Figure 7.13.

In order to compare the proposed method with existing results published in the literature, we will employ the model used in [104]. This model makes use of a third order polynomial tyre model and is still useful for a RoA estimate comparisons, and has been adopted in several prior studies [5, 6, 92, 93]. Figure 7.14 compares the RoA estimates obtained by the method proposed here and the previously published results.

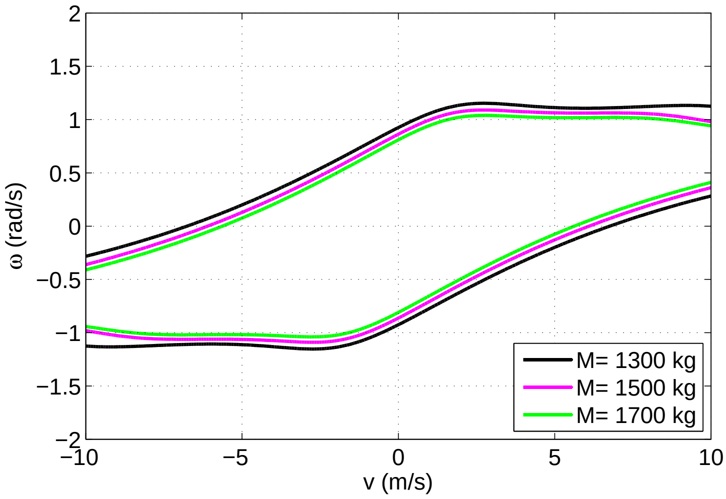

Figure 7.12: RoA estimates for the vehicle with mass increased (green) and decreased (black) by 200 kg.

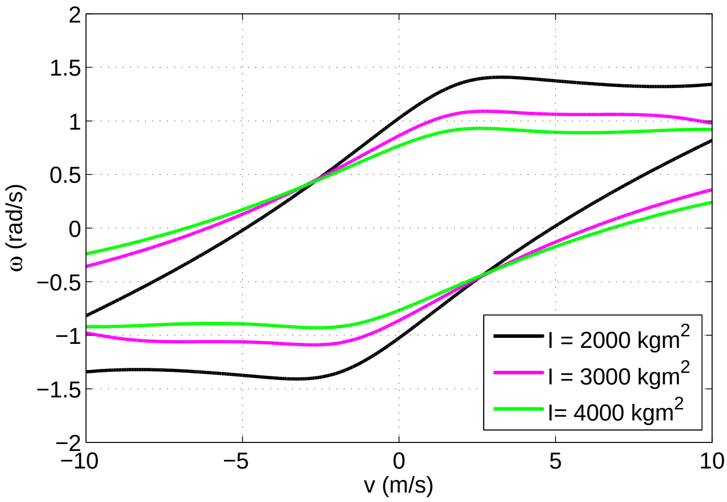

Figure 7.13: RoA estimates for the vehicle for different values of moment of inertia.

In [5] two Lyapunov functions were used for estimating the RoA, the linearised system Lyapunov function V_{lit1} and the modified energy function V_{lit2}. The estimate that was obtained in [6] by examining the phase plot, and rotating the Lyapunov function such that it aligned with the stable region, is referred to as V_{lit3}; all the three Lyapunov functions were quadratic. It is evident from Figure 7.14 that the proposed method with quadratic V gives an estimate with a similar area to V_{lit3}, and a significantly larger area when compared with V_{lit1} and V_{lit2}. The quartic V encompassed all the

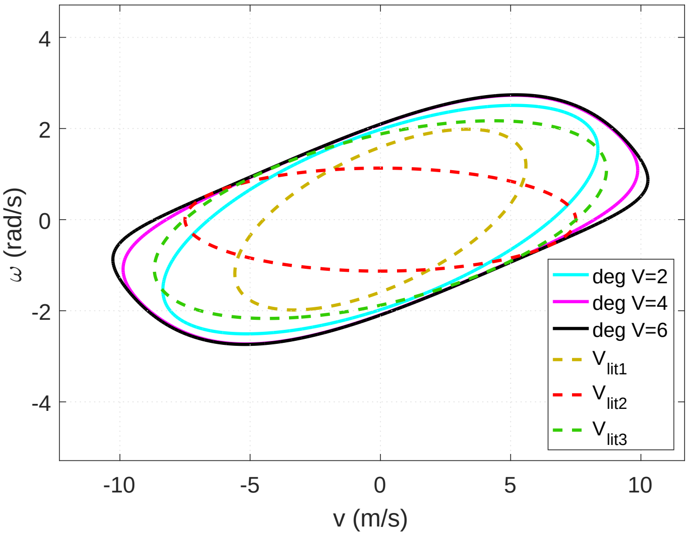

Figure 7.14: Comparison of the RoA estimate using the proposed method against the methods used in the literature. The solid lines are the estimates for various degrees of V with $s(\boldsymbol{x})$ bootstrapped. V_{lit1} and V_{lit2} are the estimates from [5] and V_{lit3} from [6].

regions covered by the quadratic Lyapunov functions, with the sixth order performing slightly better still. The phase portrait along with the RoA estimate using the sixth degree Lyapunov function is shown in Figure 7.15. The RoA estimate covers almost

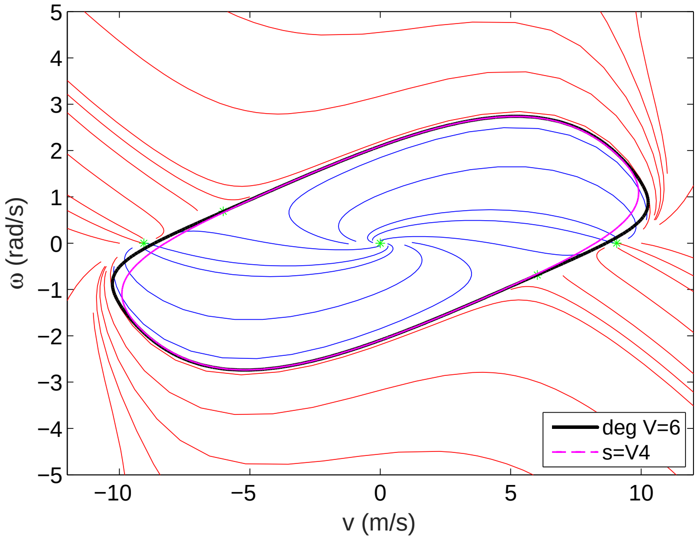

Figure 7.15: Phase plot of the vehicle model used in RoA comparison. The solid black line is the RoA estimate using a degree 6 Lyapunov function with $s(\boldsymbol{x})$ set to the quartic Lyapunov function (dashed magenta). The green stars show the equilibrium points.

all of the stable region. The variable sized region $s(\boldsymbol{x})$ was set to be the previous fourth-order Lyapunov function and is shown in dashed magenta.

7.6 Conclusions

Sum-of-squares programming has been used to estimate the region of attraction of the nonlinear lateral dynamics of a single-track car model. Existing polynomial-based SOS programs were extended to allow for rational vector fields in the system dynamics. The rational RoA algorithm performs well and is able to capture the majority of the actual RoA for the vehicle-related problems studied here. It is shown that rational approximations of the tyre friction characteristics result in significant improvements in accuracy as compared with the more conventional polynomial approximations. It is shown that the proposed method can be used to study the influence of various vehicle parameters on the stability boundary.

Symbol	Description	Value
M	Vehicle total mass	1500 kg
I_z	z moment of inertia	3000 $kg\,m^2$
a	Mass centre from front axle	1.2 m
b	Mass centre from rear axle	1.3 m
u_0	Longitudinal Speed	20 m/s

Table 7.1: Vehicle parameter values [98].

Chapter 8

Conclusions

In this thesis a series of problems related to performance, efficiency and stability of four-wheeled vehicles were studied. The first part of the thesis relies mainly on numerical optimal control so the necessary background was provided in Chapter 2. It was shown that difficult problems of great practical importance can be solved with the correct formulation and choice of algorithm. The second and smaller part focused on vehicular nonlinear stability analysis using sum-of-squares (SOS) programming. The knowledge obtained from this stability analysis can be used to predict the difficulty of the related vehicular optimal control problem.

An F1 car suspension model was developed using a first principles kinematics analysis in Chapter 3. Multivariate polynomials in normal tyre loads were employed as surrogates for the full set of nonlinear equations for the purpose of optimal control calculations. Minimum lap time problems for the F1 vehicle with the suspension meta-model in place were considered in Chapter 4. Rather than employing constant aerodynamics coefficients, an aerodynamic map that is sensitive to the configuration of the chassis was introduced. The optimal control problem was solved using a mesh and order adaptive pseudospectral method with derivatives obtained from Automatic Differentiation for faster solution times. It was shown that the car's suspension motion can have a significant impact on the aerodynamic performance of the vehicle. Parametric studies were then carried out to determine the optimal suspension set-up and find trade-offs in the aerodynamic design that can lead to lap time improvement.

The influence of local aerodynamic balance perturbations on the lap time was investigated in Chapter 5. This is particularly useful in determining whether compromises made in front and rear downforce coefficients would result in actual lap time improvements. A series of smaller issues relevant to minimum lap time problems were selected and discussed. Among these were the influence of imposing slew rate limits on the steering and tyre longitudinal slips, importance of choosing an appropriate tolerance for the NLP solver, and quantifying the effects of ignoring race-line optimisation and yaw dynamics.

A method for evaluating and optimising the fuel consumption performance of hybrid vehicles that does not require the use of driving cycles was proposed in Chapter 6. The methodology was applied to a hybrid vehicle that utilised a powertrain comprising an internal combustion engine, a flywheel and a battery. The benefits of using multiple auxiliary storage systems were thereby demonstrated. It was shown that flywheels can help in achieving significant fuel reductions in aggressive driving. Optimum fuel efficiency was sought by controlling the vehicle so that the fuel consumption is minimised over an arbitrary given three-dimensional path with a prescribed time-of-arrival constraint. The described methodology can be used to establish an absolute lower-bound estimate for the fuel consumption.

The region of attraction (RoA) estimation of the lateral dynamics of a nonlinear single-track vehicle model was considered in Chapter 7. The tyre forces were approximated using rational functions that were shown to capture the nonlinearities of tyre curves significantly better than polynomial functions. By finding a Lyapunov function for a system, one can prove the stability of its equilibrium. The level sets of this Lyapunov function are invariant sets that can be used for estimation of region of attraction. Using SOS programming, one can systematically search for a Lyapunov function by relaxing the positivity conditions to SOS conditions, which can then be solved as a semi-definite program. An existing sum-of-squares (SOS) programming algorithm for RoA estimation was extended to accommodate the use of rational vector fields. This algorithm was then used to estimate the RoA of the vehicle's lateral dynamics. The influence of vehicle parameters such as mass centre position and moment of inertia, and driving conditions such as forward speed and tyre cornering stiffness on the stability region were studied. The results were then compared and reconciled with predictions from linear stability theory. It was shown that SOS programming techniques can be used to approximate the stability region relatively accurately without resorting to numerical integration. The RoA estimates from the SOS algorithm were compared to the existing results in the literature that are based on finding invariant sets. The proposed method was shown to obtain significantly better RoA estimates.

8.1 Future Work

In this section we provide directions for future research.

- The suspension meta-model developed for the optimal lap time calculations ignores the fast dynamics of the suspension by assuming a steady-state condition. The impact of this assumption can be quantified by using a model that takes into account the dynamics of the heave spring and damper combination. This will result in much slower solution times. However, rather than solving the full optimal control problem, one can take the previous state and control histories and integrate the vehicle model with the full nonlinear suspension system to quantify the impact of using the meta-model.

- The minimum lap time results correspond to the best open loop strategy obtained from the optimiser. They take no account of solution robustness and the difficulty of implementation in real-world racing. It would be of paramount benefit to include driver characteristics in the model and develop an elegant method to appropriately exclude solutions that are considered undesirable from the racing driver perspective.

- In Chapter 5, comparing the modified optimal control problem with the QSS assumptions included against a QSS code would determine the significance of assuming the lateral acceleration to be temporarily constant.

- In the minimum fuel problem, it would be useful to study the effects of route optimisation. Adding traffic information to the model would also make the simulations more realistic. Component sizing of the combustion engine and auxiliary storage system can be formulated easily in the optimal control problem by including static parameters. This is of great practical interest and has been receiving much attention in the literature.

- An interesting extension to Chapter 7 would be an RoA study for the braking and accelerating vehicle. In this case, the system dynamics would be time-varying as the forward velocity would be a function of time and longitudinal acceleration. The load transfer in these cases has a significant impact on the steering behaviour [87]. It is therefore important to include the normal loads in the tyre approximation model. The stability of a time-varying nonlinear system can be established using Lyapunov functions [100]. The challenge would be to determine whether SOS techniques can still be applied to find the invariant sets of such a system for RoA estimation.

Appendix A

Multivariate Polynomial Fitting

The set of all multivariate polynomials in n variables up to degree d over the field of complex numbers $\mathbb{C}$, together with addition and multiplication with a scalar, form a vector space. This vector space is denoted by $\mathcal{C}_d^n$. A basis for this vector space consists of all monomials from degree 0 up to d. Since the total number of monomials in n variables from degree 0 to degree d is given by $q = \begin{pmatrix} n+d \\ n \end{pmatrix} = \frac{(n+d)!}{d!n!}$, it follows that $\dim(\mathcal{C}_d^n) = q$. The degree of a monomial $x^a = x_1^{a_1} \cdots x_n^{a_n}$ is defined as $|a| = \sum_{i=1}^n a_i$. The degree of a polynomial p is the degree of the monomial of p with highest degree. The terms of multivariate polynomials can be ordered in different ways. A formal definition of monomial orderings together with a detailed description of some orderings in computational algebraic geometry is given in [105]; we will use the *Graded Reverse Lexicographic Order*. In this ordering system, $a > b$ if $|a| = \sum_{i=1}^n a_i > |b| = \sum_{i=1}^n b_i$, or if $|a| = |b|$ and $a >_{xel} b$, if in the vector difference $a - b$ the leftmost nonzero entry is negative. The ordering is 'graded' because it first compares the degrees of the two monomials and then applies the $>_{xel}$ ordering when there is a tie.

Example $(2,0,0) > (0,0,1)$, because $|(2,0,0)| > |(0,0,1)|$, which implies that $x_1^2 > x_3$. Likewise, $(0,1,1) > (2,0,0)$, because $(0,1,1) >_{xel} (2,0,0)$ and so $x_2x_3 > x_1^2$.

A monomial ordering allows a multivariate polynomial p to be represented by a vector of its coefficients in a consistent way. To do this one orders the coefficients in a row vector that is induced by the chosen ordering system. Suppose that $p = 2 + 3x_1 - 4x_2 + x_1x_2 - 8x_1x_3 - 7x_2^2 + 3x_3^2 \in \mathcal{C}_3^2$; since $\frac{5!}{3!2!} = 10$, the associated basis is 10 dimensional. As a result p can be represented by the vector of coefficients in graded reverse lexicographic order

1	x_1	x_2	x_3	x_1^2	x_1x_2	x_1x_3	x_2^2	x_2x_3	x_3^2
2	3	-4	0	0	1	-8	-7	0	3

The least-squares fitting of multivariate polynomials to given data is an easy extension of the univariate case. Suppose that we wish to fit a multivariate polynomial of the form

$$\begin{bmatrix} 1 & x_1 & x_2 & \cdots & x_n^d \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_q \end{bmatrix} \tag{A.1}$$

to given data, in which the terms are listed in *graded reverse lexicographic order*. Since the of dimension of $\mathcal{C}_d^n$ is q, we will suppose from now on that we have $k \geq q$ independent experimental outcomes, which can be stacked as

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_k \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{bmatrix} - \begin{bmatrix} 1 & (x_1)_1 & (x_2)_1 & (x_3)_1 & \cdots & (x_n^d)_1 \\ 1 & (x_1)_2 & (x_2)_2 & (x_3)_2 & \cdots & (x_n^d)_2 \\ \vdots & & \ddots & \ddots & & \vdots \\ 1 & (x_1)_k & (x_2)_k & (x_3)_k & \cdots & (x_n^d)_k \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_q \end{bmatrix} \tag{A.2}$$

in which $\boldsymbol{e}$ is a k-vector of errors, $\boldsymbol{f}$ is a k-vector of experiment outcomes,

$$\Phi = \begin{bmatrix} 1 & (x_1)_1 & (x_2)_1 & (x_3)_1 & \cdots & (x_n^d)_1 \\ 1 & (x_1)_2 & (x_2)_2 & (x_3)_2 & \cdots & (x_n^d)_2 \\ \vdots & & \ddots & \ddots & & \vdots \\ 1 & (x_1)_k & (x_2)_k & (x_3)_k & \cdots & (x_n^d)_k \end{bmatrix}$$

and $\boldsymbol{p}$ is a q-vector of polynomial coefficients. More compactly we have

$$\boldsymbol{e} = \boldsymbol{f} - \Phi \boldsymbol{p}. \tag{A.3}$$

Optimum values for the parameter vector $\boldsymbol{p}$ can be found by completing the square

$$\begin{aligned} \boldsymbol{e}^T\boldsymbol{e} &= (\boldsymbol{p} - (\Phi^T\Phi)^{-1}\Phi^T\boldsymbol{f})^T(\Phi^T\Phi)(\boldsymbol{p} - (\Phi^T\Phi)^{-1}\Phi^T\boldsymbol{f}) \\ &\quad + \boldsymbol{f}^T(I_k - \Phi(\Phi^T\Phi)^{-1}\Phi^T)\boldsymbol{f}. \end{aligned} \tag{A.4}$$

Since the first term on the right-hand side of (A.4) is non-negative for any choice of $\boldsymbol{p}$, the optimal polynomial coefficients come from making this term zero by setting

$$\boldsymbol{p}_{opt} = (\Phi^T\Phi)^{-1}\Phi^T\boldsymbol{f} \tag{A.5}$$

with the resulting optimal estimation error

$$\|\boldsymbol{e}\|_2^2 = \boldsymbol{f}^T(I_k - \Phi(\Phi^T\Phi)^{-1}\Phi^T)\boldsymbol{f}. \tag{A.6}$$

Appendix B

Tyre Friction

As explained previously in [2, 4, 7], the tyre frictional forces are modelled using empirical formulae that are functions of the tyre normal loads and the combined slip. The tyre's longitudinal slip is described by a longitudinal slip coefficient κ, while the lateral slip is described by a slip angle α [75]. Following standard conventions we use

$$\kappa = -\left(1 + \frac{R\omega_w}{u_w}\right), \tag{B.1}$$

$$\tan\alpha = -\frac{v_w}{u_w}, \tag{B.2}$$

where R is the wheel radius and ω_w the wheel's spin angular velocity. The quantities u_w and v_w are the absolute speed components of the wheel centre in a wheel-fixed coordinate system. The following formulae give the peak values and locations of the lateral and longitudinal friction coefficients using linear interpolation [71]

$$\mu_{x\,max} = (F_z - F_{z1})\frac{\mu_{x\,max2} - \mu_{x\,max1}}{F_{z2} - F_{z1}} + \mu_{x\,max1}, \tag{B.3}$$

$$\mu_{y\,max} = (F_z - F_{z1})\frac{\mu_{y\,max2} - \mu_{y\,max1}}{F_{z2} - F_{z1}} + \mu_{y\,max1}, \tag{B.4}$$

$$\kappa_{max} = (F_z - F_{z1})\frac{\kappa_{max2} - \kappa_{max1}}{F_{z2} - F_{z1}} + \kappa_{max1}, \tag{B.5}$$

$$\alpha_{max} = (F_z - F_{z1})\frac{\alpha_{max2} - \alpha_{max1}}{F_{z2} - F_{z1}} + \alpha_{max1}, \tag{B.6}$$

where the quantities containing a '1' or a '2' in the subscript are measured tyre parameters. Normalising the slips by the peak slips gives

$$\kappa_n = \kappa/\kappa_{max}, \tag{B.7}$$

$$\alpha_n = \alpha/\alpha_{max}. \tag{B.8}$$

Following normalisation, the slip is characterised by a combined-slip coefficient

$$\rho = \sqrt{\alpha_n^2 + \kappa_n^2}. \tag{B.9}$$

The lateral and longitudinal coefficients of friction are given by

$$\mu_x = \mu_{x\,max} \sin\left(Q_x \arctan(S_x \rho)\right), \tag{B.10}$$

$$\mu_y = \mu_{y\,max} \sin\left(Q_y \arctan(S_y \rho)\right), \tag{B.11}$$

where Q_x and Q_y are the tyre longitudinal and lateral shape factors, and

$$S_x = \frac{\pi}{2\arctan(Q_x)}, \tag{B.12}$$

$$S_y = \frac{\pi}{2\arctan(Q_y)}. \tag{B.13}$$

Finally, the lateral and longitudinal tyre forces are given by

$$F_x = \mu_x F_z \frac{\kappa_n}{\rho}, \tag{B.14}$$

$$F_y = \mu_y F_z \frac{\alpha_n}{\rho}. \tag{B.15}$$

The normal loads are determined by solving the load balance equations. The nominal tyre parameters are given in Table B.1. The left-hand diagram on Figure B.1 shows the load dependence of the longitudinal tyre force under pure longitudinal slip conditions, i.e. $\alpha = 0$. As the normal load increases, the tyre is able to generate more longitudinal force. The main reason for introducing aerodynamic downforce generating systems on Formula One cars is to exploit this effect. The right-hand diagram on Figure B.1 shows how the longitudinal force is compromised when the tyre is side slipping. As the side-slip angle increases, the longitudinal peak force reduces and moves further away from the origin. The lateral tyre force exhibits broadly similar behaviours.

The combined slip characteristics of the tyre model at a fixed normal load of 2 kN is demonstrated in Figure B.2. On the left-hand diagram, the magnitude of the tyre force generated is shown against the longitudinal and lateral slips. The contours of equal force magnitude are shown on the right-hand diagram.

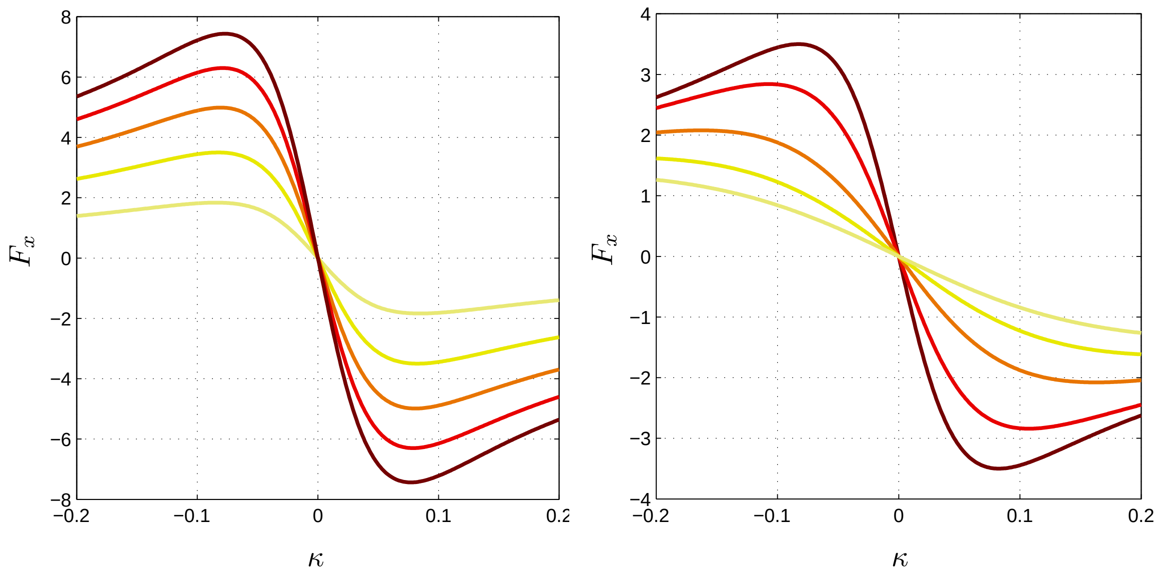

Figure B.1: Longitudinal tyre force F_x in kN against the longitudinal slip κ. The left-hand diagram considers five normal loads (at $\alpha = 0°$); the lightest curve corresponds to 1 kN with the normal load then increased in steps of 1 kN. The right-hand diagram considers combined slip for five values of side-slip angle at a normal load of 2 kN; the darkest curve corresponds to a slip angle of 0° with the slip angle increased in steps of 5° [7].

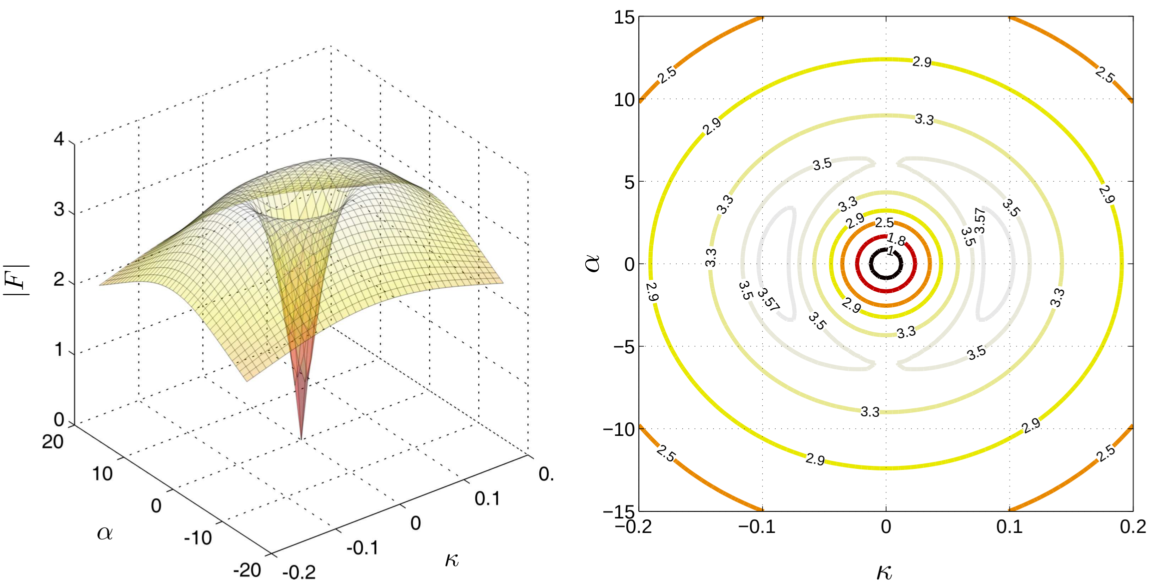

Figure B.2: Combined tyre force magnitude as a function of the longitudinal slip κ and lateral slip angle α (in degrees) at a normal load of 2 kN. The right-hand diagram shows contours of equal tyre force [7].

Symbol	Description	Value
F_{z1}	Reference load 1	2000 N
F_{z2}	Reference load 2	6000 N
μ_{x1}	Peak longitudinal friction coefficient at load 1	1.75
μ_{x2}	Peak longitudinal friction coefficient at load 2	1.40
κ_1	Slip coefficient for the friction peak at load 1	0.11
κ_2	Slip coefficient for the friction peak at load 2	0.10
μ_{y1}	Peak lateral friction coefficient at load 1	1.80
μ_{y2}	Peak lateral friction coefficient at load 2	1.45
α_1	Slip angle for the friction peak at load 1	9°
α_2	Slip angle for the friction peak at load 2	8°
Q_x	Longitudinal shape factor	1.9
Q_y	Lateral shape factor	1.9

Table B.1: Nominal tyre friction parameters.

Appendix C

Vehicle Parameters

C.1 Racing Car

The following table contains the nominal values for the vehicle parameters used in Chapter 4. The air density is assumed to be $\rho = 1.2\,kg/m^3$.

Symbol	**Description**	**Value**
P_{max}	Peak engine power	$560\ kW$
M	Vehicle mass	$660\ kg$
I_z	Moment of inertia about the $z-$axis	$450\ kg\,m^2$
w	Wheelbase	$3.4\ m$
a	Distance of the mass centre from the front axle	$1.8\ m$
b	Distance of the mass centre from the rear axle	$w - a$
h	Centre of mass height	$0.3\ m$
w_f	Front wheel to car centre line distance	$0.73\ m$
w_r	Rear wheel to car centre line distance	$0.73\ m$
R	Wheel radius	$0.33\ m$
k_d	Differential friction coefficient	$100\ N/rpm$
A	Vehicle frontal area	$1.5\ m^2$

Table C.1: Racing vehicle nominal parameters

C.2 Hybrid Vehicle

This table contains the hybrid vehicle parameters used in Chapter 6.

Symbol	**Description**	**Value**
M	Vehicle total mass	1400 kg
M_{car}	Vehicle only mass	1280 kg
M_{bat}	Battery and motor mass	100 kg
M_{fly}	Flywheel mass	20 kg
I_x	x moment of inertia	500 $kg\,m^2$
I_y	y moment of inertia	1000 $kg\,m^2$
I_z	z moment of inertia	1000 $kg\,m^2$
a	Mass centre from front axle	1.35 m
b	Mass centre from rear axle	1.35 m
h	Centre of mass height	0.5 m
C_d	Drag coefficient	0.3
A	Vehicle frontal area	1.8 m^2
C_d	Rolling resistance coefficient	0.009
E_b^{max}	Max battery energy capacity	5 MJ
P_b^{max}	Max battery power	25 kW
V_{OC}^{min}	Min battery voltage	240 V
V_{OC}^{max}	Max battery voltage	210 V
R_b	Battery internal resistance	0.5 Ω
SoC_{min}	Min state of charge	40 % V
SoC_{max}	Max state of charge	80 % V
P_{fly}^{max}	Max flywheel power	60 kW
μ_{em}	Electric motor efficiency	85%
E_{fly}^{max}	Max flywheel energy	400 kJ
J_{fly}	Flywheel spinning inertia	0.02 $kg\,m^2$
μ_{CVT}	CVT efficiency	85%

Table C.2: Hybrid vehicle parameters

Bibliography

[1] *Formula One Technical Regulations*, Fédération Internationale de l'Automobile Std., 2015. [Online]. Available: http://www.fia.com/sport/regulations

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

[3] J. R. Blasco-Figueroa, "Minimum time manoeuvre based on the g-g-speed envelope," Master's thesis, School of Mechanical Engineering, 2000.

[4] D. Limebeer and G. Perantoni, "Optimal control of a formula one car on a three-dimensional track-part 2: Optimal control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, p. 051019, 2015.

[5] D. Johnson and J. Huston, "Nonlinear lateral stability analysis of road vehicles using liapunov's second method," SAE Technical Paper, Tech. Rep., 1984.

[6] S. Sadri and C. Q. Wu, "Lateral stability analysis of on-road vehicles using lyapunov's direct method," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 821–826.

[7] 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. 19, pp. 2065–2080, 2014.

[8] D. Sperling and D. Gordon, *Two Billion Cars: Driving Toward Sustainability*. Oxford University Press, 2009.

[9] V. I. Zubov and L. F. Boron, *Methods of AM Lyapunov and their Application.* Noordhoff Groningen, 1964.

[10] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[11] M. I. Masouleh and D. J. N. Limebeer, "Heave spring and ride height optimisation of a formula one car suspension system," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on.* IEEE, 2015, pp. 165–170.

[12] ——, "Optimizing the aero-suspension interactions in a formula one car," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 3, pp. 912–927, 2016.

[13] ——, "Fuel minimisation for a vehicle equipped with a flywheel and battery on a three-dimensional track," *IEEE Transactions on Intelligent Vehicles*, under Review.

[14] ——, "Non-linear vehicle domain of attraction analysis using sum-of-squares programming," in *Control Applications (CCA), 2016 IEEE Conference on.* IEEE, 2016, pp. 1209–1214.

[15] ——, "Region of attraction analysis for nonlinear vehicle lateral dynamics using sum-of-squares programming," *Vehicle System Dynamics*, under Review.

[16] H. J. Sussmann and J. C. Willems, "300 years of optimal control: from the brachystochrone to the maximum principle," *IEEE Control Systems*, vol. 17, no. 3, pp. 32–44, 1997.

[17] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction.* Princeton, NJ, USA: Princeton University Press, 2011.

[18] J. Nocedal and S. Wright, *Numerical optimization.* Springer Science & Business Media, 2006.

[19] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 2004.

[20] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming.* SIAM, 2010.

[21] P. E. Gill, W. Murray, and M. A. Saunders, *Users guide for SNOPT version 7: Software for large-scale nonlinear programming*, 2006.

[22] 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.

[23] A. Griewank and A. Walther, *Evaluating derivatives: principles and techniques of algorithmic differentiation.* Siam, 2008.

[24] M. J. Weinstein and A. V. Rao, "A source transformation via operator overloading method for the automatic differentiation of mathematical functions in MATLAB," *ACM Trans. Math. Softw.*, vol. 42, no. 2, pp. 11:1–11:44, May 2016. [Online]. Available: http://doi.acm.org/10.1145/2699456

[25] D. E. Kirk, *Optimal Control Theory: An Introduction.* Prentice-Hall, Inc., Englewood Cliffs, N. J., 1970.

[26] A. E. Bryson and Y. C. Ho, *Applied optimal control: optimization, estimation and control.* Hemisphere Publishing Corporation, 1975.

[27] C. Lanczos, *The Variational Principles of Mechanics*, ser. Dover Books On Physics. Dover Publications, 1970.

[28] I. M. Gelfand, R. A. Silverman *et al.*, *Calculus of variations.* Courier Corporation, 2000.

[29] L. S. Pontryagin, *Mathematical theory of optimal processes.* CRC Press, 1987.

[30] M. Athans and P. L. Falb, *Optimal control: an introduction to the theory and its applications.* Courier Corporation, 2013.

[31] U. M. Ascher, R. M. M. Mattheij, and R. D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations.* Society for Industrial and Applied Mathematics, 1995.

[32] V. I. Arnol'd, *Mathematical methods of classical mechanics.* Springer Science & Business Media, 2013, vol. 60.

[33] M. R. Osborne, "On shooting methods for boundary value problems," *Journal of mathematical analysis and applications*, vol. 27, no. 2, pp. 417–433, 1969.

[34] E. Bertolazzi, F. Biral, and M. Da Lio, "Symbolic-numeric indirect method for solving optimal control problems for large multibody systems: The time-optimal racing vehicle example," *Multibody System Dynamics*, vol. 13, pp. 233–252, 2005.

[35] P. Deuflhard and G. Heindl, "Affine invariant convergence theorems for newton's method and extensions to related methods," *SIAM Journal on Numerical Analysis*, vol. 16, pp. 1–10, 1979.

[36] D. Kraft, "Algorithm 733: TOMP–Fortran modules for optimal control calculations," *ACM TransactIons on Mathematical Software*, vol. 20, no. 3, pp. 262–281, Sep. 1994. [Online]. Available: http://doi.acm.org/10.1145/192115.192124

[37] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[38] E. W. Cheney and W. A. Light, *A course in approximation theory.* American Mathematical Soc., 2009, vol. 101.

[39] K. E. Atkinson, *An introduction to numerical analysis.* John Wiley & Sons, 2008.

[40] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Dover Publications, 1965.

[41] 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.

[42] 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.

[43] 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.

[44] C. L. Darby, W. W. Hager, and A. V. Rao, "An hp-adaptive pseudospectral method for solving optimal control problems," *Optim. Control Appl. Meth.*, vol. 32, pp. 476–502, 2011.

[45] Q. Gong, F. Fahroo, and I. M. Ross, "Spectral algorithm for pseudospectral methods in optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 3, pp. 460–471, 2008.

[46] H. J. Kelley, "A second variation test for singular extremals," *AIAA J.*, vol. 2, pp. 1380–1382, 1964.

[47] D. Jacobson, S. B. Gershwin, and M. Lele, "Computation of optimal singular controls," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 67–73, Feb 1970.

[48] A. L. Schwartz, "Theory and implementation of numerical methods based on Runge-Kutta integration for solving optimal control problems," Ph.D. dissertation, University of California at Berkeley, 1996.

[49] J. Reimpell, H. Stoll, and J. W. Betzler, *The Automotive Chassis: Engineering Principles*, 2nd ed. Society of Automotive Engineers, 2001.

[50] M. Raghavan, "Number and dimensional synthesis of independent suspension mechanisms," *Mech. Mach. Theory*, vol. 31, no. 8, pp. 1141–1153, 1996.

[51] D. L. Cronin, "Macpherson strut kinematics," *Mechanism and Machine Theory*, vol. 16, no. 6, pp. 631–644, 1981.

[52] K. Chen and D. G. Beale, "Base dynamic parameter estimation of a Macpherson suspension mechanism," *Vehicle System Dynamics*, vol. 39, no. 3, pp. 227–244, 2003.

[53] D. E. Woods and B. A. Jawad, "Numerical design of racecar suspension parameters," in *SAE Technical Series Papers*, no. 1999-01-2257, 1999.

[54] H. M. Hiller and S. Frik, "Road vehicle benchmark 2 five-link suspension," *Vehicle System Dynamics*, vol. 22, no. S1, pp. 254–262, 1993.

[55] J. Knapczyk and S. Dzierżek, "Displacement and force analysis of five-rod suspension with flexible joints," *ASME J Mechanical Desgn*, vol. 117, no. 4, pp. 532–538, 1995.

[56] P. Simionescu and D. Beale, "Synthesis and analysis of the five-link rear suspension system used in automobiles," *Mechanism and Machine Theory*, vol. 37, pp. 815–832, 2002.

[57] R. Sancibrian, P. Garcia, F. Viadero, A. Fernandez, and A. De-Juan, "Kinematic design of double-wishbone suspension systems using a multiobjective optimisation approach," *Vehicle System Dynamics*, vol. 48, no. 7, pp. 793–813, 2010.

[58] K. P. Balike, S. Rakheja, and I. Stiharu, "Development of kineto-dynamic quarter-car model for synthesis of a double wishbone suspension," *Vehicle System Dynamics*, vol. 49, pp. 107–128, 2011.

[59] Y. A. Papegay, J.-P. Merlet, and D. Daney, "Exact kinematics analysis of cars suspension mechanisms using symbolic computation and interval analysis," *Mechanism and Machine Theory*, vol. 40, no. 4, pp. 395–413, 2005.

[60] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization," *ASME J Mechanical Design*, vol. 129, no. 4, pp. 370–380, 2007.

[61] H. Scherenberg, "Mercedes-Benz racing design and car experience," *SAE Transactions*, vol. no. 580042, pp. 414–420, 1958.

[62] W. F. Milliken and D. L. Milliken, *Race car Vehicle dynamics.* SAE, 1995.

[63] V. Cossalter, M. Da Lio, R. Lot, and L. Fabbri, "A general method for evaluation of vehicle manoeuverability with special empashis on motorcycles," *Vehicle System Dynamics*, vol. 31, pp. 113–135, 1999.

[64] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C.* Cambridge University Press, 1992.

[65] D. Tavernini, M. Massaro, E. Velenis, D. I. Katzourakis, , and R. Lot, "Minimum time cornering: the effect of road surface and car transmission layout," *Vehicle System Dynamics*, vol. 51, pp. 1533–1547, 2013.

[66] J. Hendrikx, T. Meijlink, and R. Kriens, "Application of optimal control theory to inverse simulation of car handling," *Vehicle System Dynamics*, vol. 26, pp. 449–461, 1996.

[67] A. P. Sage and C. C. White, *Optimum System Control.* Prentice-Hall, 1977.

[68] P. J. Enright and B. A. Conway, "Discrete approximations to optimal trajectories using direct transcription and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, vol. 15, pp. 994–1002, 1992.

[69] D.Casanova, “On minimum time vehicle manoeuvring: The theoretical optimal lap,” Ph.D. dissertation, Cranfield University, School of Mechanical Engineering, 2000.

[70] H. B. Pacejka and I. J. M. Besselink, “Magic formula tyre model with transient properties,” *Vehicle System Dynamics*, vol. 27, pp. 234–249, 1997.

[71] D. P. Kelly, “Lap time simulation with transient vehicle and tyre dynamics,” Ph.D. dissertation, Cranfield University, School of Mechanical Engineering, 2008.

[72] P. Falugi, E. Kerrigan, and E. van Wyk, *Imperial College London Optimal Control Software User Guide (ICLOCS)*, Imperial College London, May 2010.

[73] G. Perantoni, “Optimal control of vehicle systems,” Ph.D. dissertation, Deparment of Engineering Science, University of Oxford, 2013.

[74] G. Perantoni and D. J. Limebeer, “Optimal control of a formula one car on a three-dimensional track-part 1: Track modeling and identification,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 5, p. 051018, 2015.

[75] H. B. Pacejka, *Tyre and Vehicle Dynamics*, second edition ed. Butterworth-Heinemann, 2008.

[76] M. A. Patterson and A. V. Rao, *GPOPS - II Version 1.0: A General-Purpose MATLAB Toolbox for Solving Optimal Control Problems Using the Radau Pseudospectral Method*, University of Florida, 2013.

[77] L. Biegler and V. Zavala, “Large-scale nonlinear programming using IPOPT: An integrating framework for enterprise-wide dynamic optimization,” *Computers and Chemical Engineering*, vol. 33, pp. 575–582, 2008.

[78] D. L. Brayshaw and M. F. Harrison, “A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location,” *Proceedings of the Institution of Mechanical Engineers,*

Part D: Journal of Automobile Engineering, vol. 219, no. 6, pp. 725–739, 2005. [Online]. Available: http://dx.doi.org/10.1243/095440705X11211

[79] M.-J. Kim and H. Peng, "Power management and design optimization of fuel cell/battery hybrid vehicles," *Journal of Power Sources*, vol. 165, pp. 819–832, 2007.

[80] J. Liu and H. Peng, "Modeling and control of a power-split hybrid vehicle," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 6, pp. 1242–1251, Nov 2008.

[81] S. Moura, H. Fathy, D. Callaway, and J. Stein, "A stochastic optimal control approach for power management in plug-in hybrid electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 545–555, May 2011.

[82] G. Rousseau, D. Sinoquet, A. Sciarretta, and Y. Milhau, "Design optimisation and optimal control for hybrid vehicles," in *EngOpt 2008 - International Conference on Engineering Optimization*, 2008.

[83] S. Delprat, J. Lauber, T. M. Guerra, and J. Rimaux, "Control of a parallel hybrid powertrain: Optimal control," *IEEE Transactions on Vehicular Technology*, vol. 53, pp. 872–881, 2004.

[84] H. Roy, A. McGordon, and P. Jennings, "A generalized powertrain design optimization methodology to reduce fuel economy variability in hybrid electric vehicles," *Vehicular Technology, IEEE Transactions on*, vol. 63, no. 3, pp. 1055–1070, March 2014.

[85] F. Kaiser, "The climb of jet-propelled aircraft, part I. speed along the path of optimum climb," Ministry of Supply (Gt. Brit.), Tech. Rep., April 1944.

[86] H. S. Tsien and R. C. Evans, "Optimal thrust programming for a sounding rocket," *Journal of the American Rocket Society*, vol. 21, no. 3, pp. 99–107, 1951.

[87] H. B. Pacejka, *Tyre and Vehicle Dynamics, 2nd Edition.* Butterworth-Heinemann Ltd, 2006.

[88] N. R. E. Laboratory, "Advanced vechile simulator," http://adv-vehicle-sim.sourceforge.net/advisor_doc.html, 2003.

[89] R. Doucette, "The oxford vehicle model: A tool for modeling and simulating the powertrains of electric and hybrid electric vehicles," Ph.D. dissertation, University of Oxford, 2013.

[90] J. Y. Wong, *Theory of ground vehicles.* John Wiley & Sons, 2001.

[91] N. Zhang, G.-M. Dong, and H.-P. Du, "Investigation into untripped rollover of light vehicles in the modified fishhook and the sine maneuvers. part i: Vehicle modelling, roll and yaw instability," *Vehicle System Dynamics*, vol. 46, no. 4, pp. 271–293, 2008.

[92] J. Samsundar and J. C. Huston, "Estimating lateral stability region of a nonlinear 2 degree-of-freedom vehicle," SAE Technical Paper, Tech. Rep., 1998.

[93] S. Sadri and C. Wu, "Stability analysis of a nonlinear vehicle model in plane motion using the concept of lyapunov exponents," *Vehicle System Dynamics*, vol. 51, no. 6, pp. 906–924, 2013.

[94] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard, *Positive Polynomials in Control.* Springer-Verlag Berlin Heidelberg, 2005, ch. Control Applications of Sum of Squares Programming, pp. 3–22.

[95] A. Papachristodoulou, "Scalable analysis of nonlinear systems using convex optimization," Ph.D. dissertation, California Institute of Technology, 2005.

[96] W. Tan and A. Packard, "Stability region analysis using polynomial and composite polynomial lyapunov functions and sum-of-squares programming," *IEEE Transactions on Automatic Control*, vol. 53, pp. 565–571, 2008.

[97] U. Topcu, A. Packarda, and P. Seiler, "Local stability analysis using simulations and sum-of-squares programming," *Automatica*, vol. 44, pp. 2669–2675, 2008.

[98] E. Ono, S. Hosoe, H. D. Tuan, and S. Doi, "Bifurcation in vehicle dynamics and robust front wheel steering control," *IEEE Transactions on Control Systems Technology*, vol. 6, pp. 412–420, 1998.

[99] A. Packard, U. Topcu, P. Seiler, and G. Balas, "Help on SOS," *IEEE Control Systems*, vol. 30, no. 4, pp. 18–23, 2010.

[100] H. K. Khalil and J. Grizzle, *Nonlinear systems.* Prentice hall New Jersey, 1996, vol. 3.

[101] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory.* Society for Industrial and Applied Mathematics, 1994. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611970777

[102] P. Seiler, *SOSOPT: A Toolbox for Polynomial Optimization Version 2.00*, August 2013.

[103] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, 1999. [Online]. Available: http://dx.doi.org/10.1080/10556789908805766

[104] H. Sachs and C. Chou, "On the stability in the sense of liapunov of a rubber tire vehicle," *Journal of Dynamic Systems, Measurement, and Control*, vol. 98, no. 2, pp. 180–185, 1976.

[105] D. A. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry*, 2nd ed., S. Axler, F. Gehring, and K. Ribet, Eds. Springer, 2005.