Trajectory Optimization Using Model Scale Cars

Diogo André Fernandes da Silva

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof. Paulo Jorge Coelho Ramalho Oliveira
Prof. Carlos Baptista Cardeira

Examination Committee

Chairperson: Prof. Paulo Rui Alves Fernandes
Supervisor: Prof. Carlos Baptista Cardeira
Member of the Committee: Prof. Jorge Alberto Cadete Ambrósio

June 2017

Acknowledgements

First of all, I would like to thank my parents for the chance of studying in this institution.

To Mariana is difficult to mention all the things that I would like to thank, some of them were so perfectly executed that I can not even recall.

To Prof. Paulo Oliveira and Prof. Carlos Cardeira I would first like to thank for the mentoring during the courses that I had the pleasure to learn from you. Special thanks to Prof. Paulo Oliveira, which made me saw the beauty of systems control theory and application.

At last, I would like to thank to all my friends and colleagues that with me shared 3 great years during our time at Projecto FST Novabase, specially to the Suspension team.

Resumo

Nesta tese duas formulações de trajectórias de tempo mínimo usando controlo óptimo são propostas: Tempo mínimo num caminho prescrito e um método de optimização de trajectória. Primeiro, são introduzidas noções importantes de dinâmica de veículo são introduzidas, seguindo-se uma reformulação usando o método de Euler-Lagrange para transformar o problema numa forma que facilita a implementação. De seguida, a pista é parametrizada e aplica-se uma transformação de variáveis que facilita a formulação é feita. O problema de optimização é formulado no contínuo e aplica-se um método de transcrição directa para transformar o problema de controlo óptimo num de optimização de grande escala. Por fim, obtêm-se soluções para o problema em causa usando programas de soluções de optimização baseados em métodos de ponto interior, de programação sequencial quadrática(SQP) e de programação sequencial de cones de segunda ordem(SOCP).
Palavras-Chave: Optimização de Trajectória, Controlo Óptimo, Planeamento de Movimento

Abstract

This work proposes two Optimal Control formulations: Minimum time path tracking and minimum Time trajectory optimization. Firstly, the vehicle dynamics is introduced and is reformulated in a way that eases the optimization formulation. Secondly, the track is parametrized and a change of variables is performed to adapt the optimal control problem. To transform the problem in a large scale optimization problems direct transcription methods were employed. At last, the problem is solved exploiting nonlinear optimization methods, such as interior-point(IP) and sequential quadratic programming(SQP) and Second Order Cone Programming (SOCP).

Keywords: Trajectory Optimization, Optimal Control, Motion Planning

Nomenclature

Roman Symbols

w_f - Half front track

w_r - Half rear track

a_1 - Distance from front axle to the car center of gravity

a_2 - Distance from front axle to the car center of gravity

a - Acceleration in body coordinates

l - Wheelbase

F - Force

v - Velocity in body coordinates

V - Velocity in fixed frame coordinates coordinates

K - Longitudinal Resistance Coefficient

M - Moment

m - Mass

R_g - Wheel Loaded Radius

w_w - Wheel angular velocity

C_{SR} - Tyre longitudinal stiffness

C_α - Tyre cornering stiffness

I - Moment of Inertia

r - Yaw rate

s - Variable defining the distance traveled in a path

t - Time

q - Generalized coordinates

$\mathcal{L}$ - Lagrangian vector

$\mathcal{M}$ - Mass matrix

$\mathcal{C}$ - Centripetal and resistive forces matrix

$\mathbf{F}$ - Tyre friction related terms matrix

$\mathbf{u}$ - Vector of inputs

$\mathbf{m}$ - Modified mass related terms matrix

$\mathbf{c}$ - Modified Centripetal and resistive forces matrix

$\mathbf{f}$ - Modified tyre friction related terms matrix

u_p - Pedal input signal

T - Final simulated time

f - Auxiliary variable representing velocity

b - Variable representing the transversal position of the vehicle compared to a reference path

c - First derivative of b

d - Second derivative of b

n - Vector normal to the vehicle x-axis in xy plane

J - Jacobian matrix

Greek Symbols

ϕ - Roll angle

θ - Pitch angle

ψ - Yaw angle

ν - Tyre coefficient of friction

α - Tyre slip angle

γ - Camber angle δ - Steering angle

β - Relation between the lateral and longitudinal velocity

κ - Curvature

ω - Convex set

Subscripts

a_x - Quantity in direction of x-axis

a_y - Quantity in direction of y-axis

a_z - Quantity in direction of z-axis

a_f - Front wheel(s)

a_r - Rear wheel(s)

a_o - Outer wheel(s)

a_i - Inner wheel(s)

a_0 - Initial state

Superscripts

$\dot{a}$ - First time derivative

$\ddot{a}$ - Second time derivative

a' - First spatial derivative

a'' - Second spatial derivative

$\underline{a}$ - Lower bound

$\bar{a}$ - Upper bound

a^k - Position k in a path discrtized in K points

Abbreviations

CoG - Center of Gravity

SR - Wheel Slip Ratio

GA - Genetic Algorithm

SQP - Sequential Quadratic Programming

SCP - Sequential Convex Programming

SOCP - Second Order Cone Programming

IP - Interior Point

Contents

List of Tables

List of Figures

1 Introduction

1.1 Motivation

An important role of engineers and technicians within the automotive and motorsport industry is to thorough investigate the dynamic potential of a vehicle, thus improving its setup to drive as fast as possible to ultimately minimize the lap time over a circuit. The art of modelling vehicle dynamics and simulate the car behaviour has been used for decades and has helped this industry to reduce the expensive time spent on track and in test benches as *4-post rig* or *7-post rig*. In Formula 1, for example, simulation has been taken to an extreme, due to the limited test time on track imposed by Federation Internationale de l'Automobile (FIA). Within this high technological industry it is where it is speculated that lap time simulations are most advanced. It developed from one point car with acceleration and velocity, to a two track model, incorporating the four wheels, to, ultimately, lap time simulation with embedded multibody dynamics.

Nonetheless, motorsport is not the only sector exploring trajectory optimization problems. Within mobile robotics research this topic has been gaining popularity, mostly to compute optimal trajectories, either minimum time or minimum energy, over uneven terrains, or over a space with obstacles. Assembling this both examples, one easily gets to one of the most trendy subject nowadays, which is the autonomous, or *self-driving*, cars. Moreover, trajectory optimization, nowadays, incorporates very important fields such as image precessing, to identify other vehicles, obstacles, traffic signals and traffic marks. But its application it is not restricted to cars, it has also been widely used in aerospace and aeronautical, to guide spacecrafts and aeroplanes. Furthermore, recent research in aerial unmanned vehicles (UAV's) has also been investigating optimal trajectories for drones. In this area, this topic is of utmost importance, since big enterprises have been exploring drone goods delivery and also first aid assistance.

This is still a topic to mature, there are not yet solutions widely recognized as the formula to solve this problem by the scientific community and remains one of the most appealing this field. Besides, this topic explores a confessed passion for the automotive and, more specifically, motorsport industry.

1.2 Literature Review

Trajectory optimization is a specificity of the more general and commonly used term motion planning. In [2] LaValle presented motion planning as an important collision course of robotics, artificial intelligence and control theory. In robotics, motion planning is related with problems such as how to move an object from one division into another without hitting anything. The field has been maturing and nowadays is common to include uncertainties and multiple bodies. In regards of artificial intelligence, planning was typically related with the sequence of operations necessary for the transition from an initial state into a desired one. Mostly since the 1990's, there has been a growing interest within the control field to find the open-loop trajectories for non-linear systems. Motion planning is a key subject that is inherent to a great number of robotics applications, from medical applications [3, 4, 5], industrial applications [6], autonomous urban-navigation [7, 8, 9, 10] and military logistics [11].

In the 1980's and beggining of 1990's, robotics manipulators were too slow to justify their use economically. Their productivity was limited by speed, which was limited by torque actuators. Nonetheless, increasing torques was not seen as the best solution, due to the increase in cost, power consumption and inertia. The solution to justify the use of manipulators was to perform minimum time motion planning, i.e. minimize the time needed to perform a given task, subject to actuator's constraints [12]. The scientific community that study and concern motion planning and trajectory optimization, including minimum time and minimum energy problems, has three widely accepted approaches to solve this family of problems. The first method was proposed by Brobow, Dubowsky and Gibson in [12]. It is a practical method that surpassed the limited computational of that time by defining a path by a small number of control points and then using a novel algorithm, at that time, to compute the minimum time velocity profile. Next the control points position is varied in order to get the minimum time trajectory, thus it can be reckoned as an indirect method. The solution for a special family of paths was proven to be bang-bang. This method was mainly used for manipulators, but was also successfully applied to an unmanned ground vehicle (UGV) over an uneven terrain [13]. Later, the algorithm was improved to deal with manipulator dynamic singularities [14], which expanded its applicability, since it presented a method to solve non bang-bang problems. Other similar approaches were presented such as [15], which have modelled the torque

constraints as quadratic instead of a mere function of velocity and position. Shiller's later work, once again exploits this approach to introduce more efficient algorithms, for offline and online motion planning, using the branch and bound method with more careful constraints classification for a car over uneven terrain and with multiple obstacles [16]. The solution provided in [17] was also innovative since it is assumed to be one order of magnitude faster than [12, 15]. Such results were achieved by an exhaustive exploration of the switching points timing, i.e., the timing determines when the robot should accelerate or decelerate, which is the most demanding task in terms of computational time. Another strategy to solve this problem is to apply Bellman's dynamic programming principal [18, 15], but due to its high computational effort has not been widely used.

The second great method to solve this problem is by direct transcription [19, 20, 21], which is mostly, within the field of trajectory optimization, a strategy to solve Optimal Control problems. This method transforms a continuous problem into a dicrete one, reducing therefore the complexity of some problems and increasing its computational efficiency. To use direct transcription one has to formulate the problem as convex, and once it is done each variable is discretized, facilitating the solution of the problem. This approach has been successfully applied not only to a manipulator to minimize time over a fixed path [22, 23, 24] but also for minimizing time in a track subject to its boundaries and to the car dynamics [25, 26, 27, 28].

Recently, optimal lap time methods converged with multibody dynamics systems modelling to enrich and better represent vehicle dynamics [29, 30]. This formulation, although difficult to model as a convex, thus hard to solve, constitutes the most realistic approach.

1.3 Thesis Outline

In section 2 is presented an explanation of vehicle dynamics concepts used in this work. It starts by defining the reference axis and the most important geometric features. Next a simple explanation of tyre mechanics is made, with particular attention to the tyre linear model. A two track model, named full car model in this work, was deduced and linearized to be than simplified, obtaining the bicycle model.

Section 3 describes the first trajectory optimization formulation used, which is based on genetic algorithms. Starting with the track parametrization and interpolation, followed by

a genetic algorithm explanation, and, finally, an explanation of the path planning method used.

Section 4 addresses optimal control trajectory optimization methods. Two different applications are presented in this method. First, an optimal control minimum time path tracking is presented. This subsection starts with a reformulation, through the Euler-Lagrange method, of the dynamics presented in section 2, along with a variable transformation that eases the following problem formulation. Than the problem is formulated in a convex way and direct transcription methods are applied to transform the optimal control problem in a large scale optimization problem. The minimum time path tracking problem subsection ends with a problem reformulation in the form of Second Order Cone Programming problem. The second method presented is the optimal control minimum time trajectory optimization problem, which is a generalization of the minimum time path tracking. First, the new track parametrization is presented and than the dynamics are reformulated. Analogously, the problem is formulated and direct transcription methods are employed.

In section 5 the results and discussion, for all algorithms explored in this work, are presented.

In section 6 one can find the conclusions.

This work ends with section 7 in where the future work is proposed.

2 Vehicle Dynamics

In this section the vehicle model and its inherent dynamics will be addressed. Starting from the reference axis choice, followed by tyre dynamics, full car and bicycle model. The latter limitations and assumptions shall be explained. To arrange the system in a more practical form a state-space system is derived for a linearized bicycle model.

2.1 Reference Axis

The discipline of vehicle dynamics has been studied for the last century, it has moved from the study of simple classical vehicles to modelling, simulation and optimization of multi-body dynamics systems. Attention to this field has been growing mostly due to advances in positioning, sensing, decision and control.

To study the vehicle one must first define a proper reference axis, the one chosen in this work is represented in fig. 1.

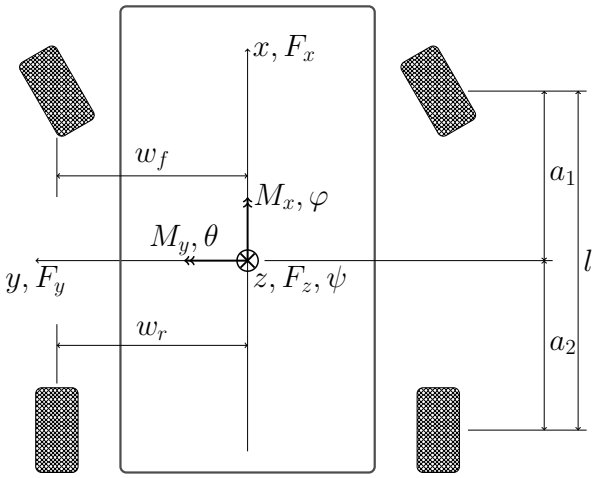

Figure 1: Representation of the reference axis used

In which, φ , θ and ψ are the roll, pitch and yaw angles, respectively. As well as F_x, F_y, F_z and M_x, M_y, M_z are, respectively, the forces and moments about the axis x, y and z. The distance from the front and rear axle to the CoG is defined as a_1 and a_2, respectively. The sum of a_1 and a_2 is the wheelbase, defined in this work as l.

2.2 Tyre Dynamics

The cornering ability of a vehicle relies on the tires friction and deformation, therefore this interaction supplies the tractive, braking and cornering forces [31]. As can be seen in fig. 2, along with the forces tires also generate moments, mostly due to non uniform load distribution in the tire print, which generate a decentralized resultant force. Such moments are M_x, overturning or roll moment, M_y, rolling resistance torque or pitch moment and M_z, self-alignment torque or yaw moment. M_y is a torque acting in the opposite way of the wheel rotation, therefore creating resistance. Such torque increases with velocity and naturally with the contact patch area. M_z is a moment that is particularly important for suspension and steering systems design, since it is one of the causes of the well known self-aligning moment on the steering wheel. In very advanced applications, considering this effect for vehicle dynamics applications may help to better understand the driver feedback and possible traction losses.

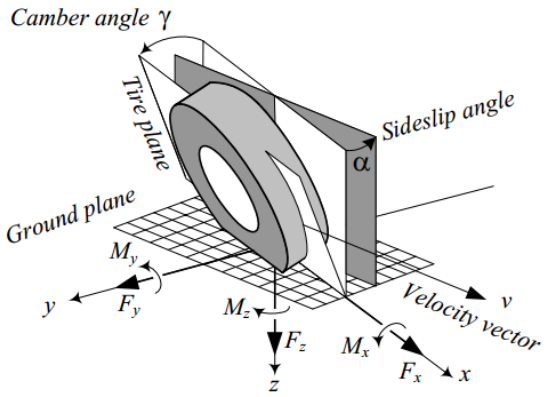

Figure 2: Tire forces and moments [1].

Tyre modelling is a complex subject, in which there are two principal types of models, the theoretical models and the ones that rely on experimental data to obtain a function that better describe such phenomenon. One of the most used tyre model is the Pacejka Magic Formula [32], which uses experimental data to adjust the form of the tyre parameters functions generated through the theoretical model.

2.2.1 Tyre Longitudinal Dynamics

To accelerate or brake a vehicle a longitudinal force must be developed between the tyre footprint and the road. When a moment is applied to the wheel spin axis, slip ratio occurs

and a longitudinal force, F_x is generated [1]. Slip ratio is defined as

$$SR = \frac{R_g w_w}{v_x} - 1 \tag{1}$$

with R_g as the tyre loaded radius, w_w is the tyre's angular velocity, v_x is the wheel forward velocity.

The longitudinal force generated by the tyre, fig. 3, is defined as

$$F_x = \mu_x(SR, \gamma, \alpha, w_w, T, F_z)F_z \approx C_{SR}SR \tag{2}$$

with μ_x defined as the tyre longitudinal coefficient of friction. The issue arises when trying to determine this coefficient, since it varies in a nonlinear fashion with the slip ratio and is also dependent on the wheel camber, γ, wheel angular velocity, operating temperature, T, tyre normal load. Furthermore, μ_x is not independent on the tyre lateral dynamics, that is why the wheel slip angle, α, appear in equation 2. To overcome such issue it is usual to merge the longitudinal and lateral slips in a combined wheel slip [33]. On equation 2, C_{SR}, which arises from a linearization about the origin, is regarded as longitudinal slip coefficient defined as

$$C_{SR} = \left| \lim_{SR \to 0} \frac{\partial F_x}{SR} \right| \tag{3}$$

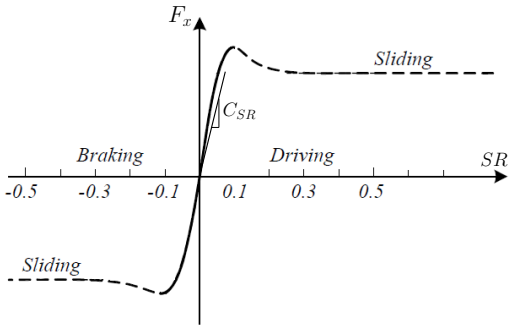

Figure 3: Longitudinal tire force, F_x, as a function of the slip ratio, SR.

2.2.2 Tyre Lateral Dynamics

Analogous to the longitudinal force generation mechanism, the tyre to create lateral force needs to deform, in this case, laterally. It causes a misalignment, between the tyre path of

motion and the tire symmetry plane, which is known as the slip angle, α. The tyre lateral force, F_y is defined as

$$F_y = C_\alpha(SR, \gamma, \alpha, w_w, T, F_z)\alpha \tag{4}$$

with C_α as the cornering stiffness, which is, as in equation 2, dependent on diverse factors. For the lateral case, factors as the camber, tyre normal load and temperature are important, due to its dynamic behaviour in cornering. Cornering stiffness, because of its nonlinearity(fig. 4), is usually considered as the derivative of the lateral force, F_y at the origin, i.e. when α tends to zero, equation 5. This operation is basically linearizing the cornering stiffness function at the origin.

$$C_\alpha = \left| \lim_{\alpha \to 0} \frac{\partial F_y}{\alpha} \right| \tag{5}$$

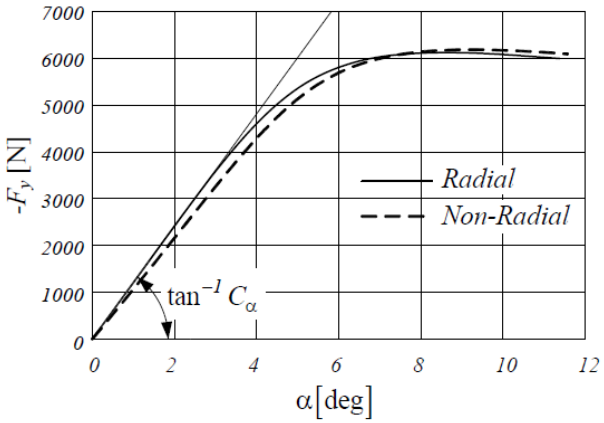

Figure 4: Lateral tire force, F_y, as a function of the slip angle, α [1].

2.3 Full Car Planar model

To obtain the full car model one must apply rigid body mechanics, arriving at a model with longitudinal, lateral and yaw degrees of freedom. The important geometric modelling quantities are represented in plain view in figure 1.

If one performs the sum of longitudinal and lateral forces at the center of gravity, CoG, according to figure 5 and accounting with the equilibrium moment about the z axis and assuming equal steering angle, δ at both front wheels, one arrives at

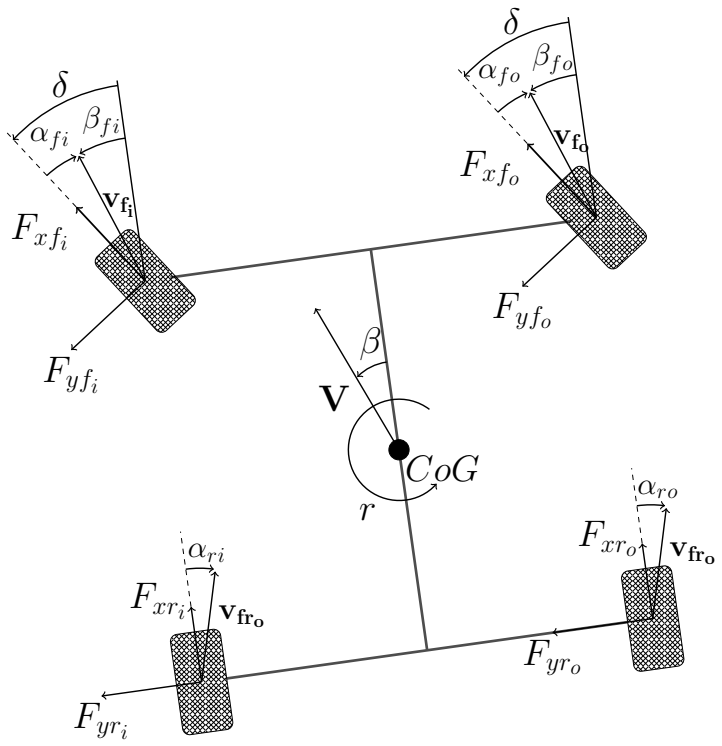

Figure 5: Representation of longitudinal and lateral forces acting on each wheel

$$\begin{aligned}
m\dot{v}_x &= mw_z v_y + F_{xf_o}\cos\delta + F_{xf_i}\cos\delta + F_{xr_o} + F_{xr_i} - F_{yf_o}\sin\delta - F_{yf_i}\sin\delta \\
m\dot{v}_y &= -mw_z v_x + F_{yf_o}\cos\delta + F_{yf_i}\cos\delta + F_{yr_o} + F_{yr_i} + F_{xf_o}\sin\delta + F_{xf_i}\sin\delta \\
I_z\dot{r} &= a_1\left[\cos\delta\left(F_{yf_o} + F_{yf_i}\right) + \sin\delta\left(F_{xf_o} + F_{xf_i}\right)\right] + w_f\left(F_{yf_i}\sin\delta - F_{xf_i}\cos\delta\right) \\
&\quad - w_r F_{xr_i} + w_f\left(F_{xf_o}\cos\delta - F_{yf_o}\sin\delta\right) + w_r F_{xr_o} - a_2\left(F_{yr_o} + F_{yr_i}\right)
\end{aligned} \tag{6}$$

Equations 6, with I_z as the body inertia concerning the z axis, describe a complete vehicle planar dynamics model, equations 4 and 2 may be used to replace F_x and F_y. To further advance on such computations, one must be able to determine each wheel slip angle, α. A closer inspection of figure 2, concludes that

$$\alpha = \delta - \beta_i \tag{7}$$

from which one infers that is crucial to compute the wheel global sideslip, β_i, with the subscript i indicating each wheel, which can be defined as

$$\beta_{f,r_{i,o}} = tan^{-1}\left(\frac{v_{y_i}}{v_{x_i}}\right) \tag{8}$$

with $v_{x_{f,r_{i,o}}}$ and $v_{y_{f,r_{i,o}}}$ each wheel longitudinal and lateral velocities, respectively. To compute both quantities one may resort to rigid body mechanics laws arriving at

$$\begin{aligned} \mathbf{v}_i &= \mathbf{V} + r \times \mathbf{r}_i \\ \begin{bmatrix} v_{x_i} \\ v_{y_i} \\ 0 \end{bmatrix} &= \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ r \end{bmatrix} \times \begin{bmatrix} x_i \\ y_i \\ 0 \end{bmatrix} = \begin{bmatrix} v_x - y_i r \\ v_y + x_i r \\ 0 \end{bmatrix} \end{aligned} \tag{9}$$

Substituting equation 9 on equation 8 , and recognizing x_i as the front or rear track and y_i as the distance of the front or rear axle to the CoG one arrives at

$$\beta_f = tan^{-1}\left(\frac{v_y + a_1 r}{v_x \pm w_f r}\right) \approx \frac{v_y + a_1 r}{v_x \pm w_f r} \tag{10a}$$

$$\beta_r = tan^{-1}\left(\frac{v_y - a_2 r}{v_x \pm w_r r}\right) \approx \frac{v_y - a_2 r}{v_x \pm w_r r} \tag{10b}$$

The assumption on equations 10a and 10b is of small angles, i.e. small absolute value of the lateral velocity in comparison with the longitudinal component, hence a linearization on this angle is accomplished. If one replaces the previous 10a and 10b on equation 7 the following linearized slip angles are obtained

$$\alpha_f = \delta - \frac{v_y + a_1 r}{v_x \pm w_f r} \tag{11a}$$

$$\alpha_r = \frac{v_y - a_2 r}{v_x \pm w_r r} \tag{11b}$$

Additionally, considering, as well, small angles for the steering angles, $sin\delta \approx 0$ and $cos\delta \approx 1$. Thus, substituting equations 2 and 4 on equation 6 it may be rewritten as

$$\begin{aligned} m\dot{v}_x &= mrv_y + C_{SR_{f_o}} SR_{f_o} + C_{SR_{f_i}} SR_{f_i} + C_{SR_{r_o}} SR_{r_o} + C_{SR_{r_i}} SR_{r_i} \\ m\dot{v}_y &= -mrv_x + C_{\alpha_{f_o}} \alpha_{f_o} + C_{\alpha_{f_i}} \alpha_{f_i} + C_{\alpha_{r_o}} \alpha_{r_o} + + C_{\alpha_{r_i}} \alpha_{r_i} \\ I_z \dot{r} &= a_1 \left(C_{\alpha_{f_o}} \alpha_{f_o} + C_{\alpha_{f_i}} \alpha_{f_i} \right) - w_f C_{SR_{f_i}} SR_{f_i} - w_r C_{SR_{r_i}} SR_{r_i} + w_f C_{SR_{f_o}} SR_{f_o} \\ &\quad + w_r C_{SR_{r_o}} SR_{r_o} - a_2 \left(C_{\alpha_{r_o}} \alpha_{r_o} + C_{\alpha_{r_i}} \alpha_{r_i} \right) \end{aligned} \tag{12}$$

Equation 12 is the final full planar model, also regarded as the two track model, which takes into account both longitudinal and lateral forces contributions for the moments around z. Although this model can be further developed to take into acount instantaneous or transient mass transfer it will not be considered in this work.

2.4 Bicycle Model

The bicycle model is a simpler solution than the full car model, in the way that it condenses in two wheels, one for each axle, the behaviour of four wheels. Therefore, both front wheels cornering stiffness are taken as a single quantity, C_{α_f}, and the same is done for the real axle, C_{α_r}. Regarding the longitudinal force, the same is done, and an important consequence is that the contribution of the longitudinal force for the moment taken about the z axis is neglected. This approximation was done to avoid modelling an electric or mechanical differential, which allows an higher rotation of the outer wheels due to travelling on a higher radius. Moreover, Ackermann geometry concerns [34] are also neglected, as before, hence each wheel steering angle is equal. Additionally, the bicycle model is a well known model for control, since it is easily described as a state-space model, accounting for tyre and rigid body dynamics, which results in more reliable control actions. As the full car model, load transfer could be added to this model, however that was not the choice, once again to simplify the problem. Nonetheless, such assumption results in a conservative solution, since the bicycle model underestimates the cornering and accelerating capabilities of the vehicle [35]. It occurs, due to the fact that load transfer increase the vertical load on the outer wheels and, consequently, its cornering stiffness. Moreover, the outer wheel, typically, has a higher slip angle which multiplies with the greater cornering stiffness, resulting in greater lateral force produced by the axle. A schematic of this model is represented in figure 6. The model equations are a special case of equations 12, in which C_α and C_{SR} values are doubled.

$$\begin{aligned} m\dot{v}_x &= mrv_y + C_{SR_f}SR_f + C_{SR_r}SR_r \\ m\dot{v}_y &= -mrv_x + C_{\alpha_f}\alpha_f + C_{\alpha_r}\alpha_r \\ I_z\dot{r} &= a_1C_{\alpha_f}\alpha_f - a_2C_{\alpha_r}\alpha_r \end{aligned} \tag{13}$$

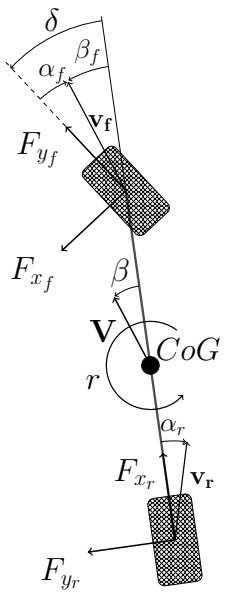

Figure 6: Representation of the longitudinal model including slip angles, $\alpha_{f,r}$, vehicle sideslip, β and the loads on each wheel, F_{x_f},F_{y_f},F_{x_r} and F_{y_r}.

The slip angles, α_f and α_r, are now defined as

$$\alpha_f = \delta - \frac{v_y + a_1 r}{v_x} \tag{14a}$$

$$\alpha_r = \frac{v_y - a_2 r}{v_x} \tag{14b}$$

And substituting equations 14a and 14b on equation 13, and rearranging it on a matrix form, one finally obtains the state space lateral dynamics system of equation 15.

$$\begin{bmatrix} \dot{v}_y \\ \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -\frac{C_{\alpha_f}+C_{\alpha_r}}{mv_x} & \frac{-a_1C_{\alpha_f}+a_2C_{\alpha_r}}{mv_x} - v_x \\ \\ -\frac{a_1C_{\alpha_f}-a_2C_{\alpha_r}}{I_z v_x} & -\frac{a_1^2C_{\alpha_f}+a_2^2C_{\alpha_r}}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ \\ r \end{bmatrix} + \begin{bmatrix} \frac{C_{\alpha_f}}{m} \\ \\ \frac{a_1C_{\alpha_f}}{Iz} \end{bmatrix} \delta \tag{15}$$

3 Trajectory Optimization

In this chapter a track parametrization is proposed along with a simple introduction to the path interpolation technique adopted. The second part of this chapter is the optimal path planning method resorting to genetic algorithm.

3.1 Track Parametrization

A reference trajectory for a minimum time trajectory in a circuit must be constrained between an outer and an inner limit. Hence, the track was modelled as a inner, r_{inner}, and outer, r_{outer}, boarders, described in Cartesian coordinates, $r = (x, y)^T \in \Re^2$, figure 7. Furthermore, the difference between outside and inner limits is defined as the track width, w_t. If one considers σ, with ($\sigma \in [0, 1]$), the position of the reference trajectory orthogonal to the inner boarder, r_{inner}, until the outer boarder, r_{outer}. The next equation fully defines a trajectory in a cartesian plane.

$$r(s) = r_{inner}(s) + \sigma(s) w_t \tag{16}$$

with s as the distance travelled, defined as

$$s(t) = \int_0^t \sqrt{v_x^2(t) + v_y^2(t)} dt \tag{17}$$

If, the distance between two consecutive points is small enough equation 17 may be approximated to

$$s(t) = \sqrt{[x(t) - x(t-1)]^2 + [y(t) - y(t-1)]^2} + s(t-1) \tag{18}$$

To compute the yaw angle, ψ, it is assumed that the car axis is tangent to the trajectory, thus

$$\psi(t) = tan^{-1} \frac{x(t) - x(t-1)}{y(t) - y(t-1)} \tag{19}$$

The curvature radius relates the yaw angle and the distance travelled between two consecutive points, if one assumes small angles

$$R(s) = \left(\frac{|x'y'' + x''y'|}{(x'^2 + y'^2)^{\frac{3}{2}}} \right)^{-1} \approx \frac{ds}{d\psi} \tag{20}$$

with x' and y' as $\frac{dx}{ds}$ and $\frac{dy}{ds}$ and x'' and y'' $\frac{dx^2}{d^2s}$ and $\frac{dy^2}{d^2s}$, i.e. $'$ is the operator representing the derivative of a vector in relation to the distance s.

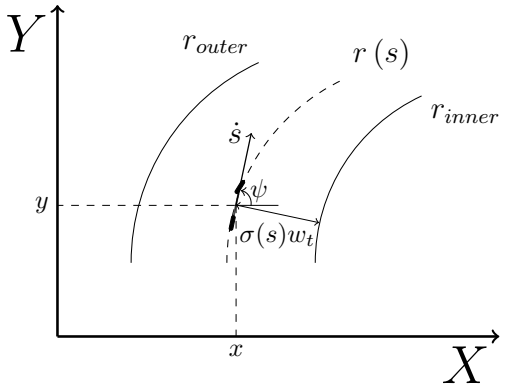

Figure 7: Track modelling and vehicle representation along a given path

Resorting to rigid body laws, a transformation from the fixed reference frame coordinates to the body reference frame coordinates, i.e. the reference frame that moves along with the body, is obtained by

$$\begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_X \\ v_Y \\ 0 \end{bmatrix} \tag{21}$$

3.2 Path Interpolation using B-Splines

If one desires to reach a minimum time trajectory or even a sub-optimal, it is important to accurately interpolate the trajectory control points into a polynomial. The issue is that second or third order polynomials are constructed resorting to 3 or 4 control points, respectively. The solution is to use piecewise polynomials. A special case of piecewise polynomials is the spline, which is defined as a piecewise polynomial parametric curve in a two-dimensional plane $Q(t) \in \Re^2$. Furthermore, when generating paths it is important to look over parametric and geometric continuity. Parametric continuity in C^n is said to exist when the n-th derivative $\frac{d^n!(t)}{dt^n}$ is continuous. Geometric continuity is broader criteria that states that the both

left-hand limit and right-hand limit may not be equal, but must have the same direction, i.e. have to be scalar multiplies of each other [36]. Additionally, it is desirable, to drivability purposes, that the heading is a continuous function on C^0, for clear reasons, and C^1 to force the heading to change continuously, i.e the yaw rate, r. Continuity in curvature is also desirable, since in the absence of such characteristics would require infinitely high accelerations to drive or steer the vehicle along the path. The choice for B-splines, can be explained by the C^2 continuity and also for the absence of any manual stitching as Bézier Curves, for example. The disadvantage of this method is the less obvious correlations between the control points and the curve shape. Nevertheless, since one is interpolating, the control points for the trajectory have to match the points belonging to the interpolant curve, thus the spline control points are adjusted in order to achieve such result.

A cubic B-spline on t, with control points $(P_i, ..., P_{i+3})$,may be defined as

$$S_i(t) = TM_{BS}G^i_{BS} = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_i \\ P_{i+1} \\ P_{i+2} \\ P_{i+3} \end{bmatrix} \tag{22}$$

The basis splines vector is

$$B_{BS} \begin{bmatrix} B_{BS,0} \\ B_{BS,1} \\ B_{BS,2} \\ B_{BS,3} \end{bmatrix}^t = \frac{1}{6} \begin{bmatrix} -t^3 + 3t^2 - 3t + 1 \\ -t^3 - 6t^2 + 4 \\ -3t^3 + 3t^2 - 3t + 1 \\ t^3 \end{bmatrix}^t \tag{23}$$

Equation 23 confirms the sufficient condition of convex hull p roperty [37], which states that $\sum_{i=0}^{3} B_{BS,i} = 1$, with $B_{BS,i} \in [0, 1], i = 0, ..., 3$.

Equation 22 may be rewritten in the polynomial form as

$$\begin{aligned} S_i(t) = &\frac{1}{6}(-P_i + 3P_{i+1} - 3P_{i+2} + P_{i+3})t^3 + \frac{1}{6}(3P_i - 6P_{i+1} + 3P_{i+2})t^2 + \\ &+ \frac{1}{6}(-3P_i + 3P_{i+2})t + \frac{1}{6}(P_i + 4P_{i+1} + P_{i+2}) \end{aligned} \tag{24}$$

3.3 Genetic Algorithm

Genetic algorithms (GA), first introduced by Holland [38], are evolutionary based algorithms, inspired by natural systems survival and adaptation mechanisms. GA starts by randomly

creating a population and from generation, i.e. iteration, to generation it scores the individuals of the current generation, called parents. After the score, GA relies on simple concepts, such as *crossover*, *mutution* and *elites*, which are the key concepts of diversity, adaptation and survival of the bests fitted individuals within a population. In a minimization problem, the lowest fitted individuals in each generation are called the *elites* and they automatically pass to the next generation, according with a used-defined variable denominated $\%elite$, the remain of the population of the current generation are the parents of the next generation. The parents suffer crossover and mutations, according with $\%crossover$ and $\%mutation$. *Crossover* represent a fine resolution change to the next individual, this is accomplished by selecting a pair of parents and combining a randomly selected vector entry of both parents. At each generation, after the fitness score, a probability is assigned to each parent, rewarding lower fitness individuals. High probability indivuduals have higher probability of being picked for crossover. The population remaing will suffer *mutations*. *Mutation* selects a parent and randomly change the vector values originating a children. Opposed to *crossovers*, *mutations* represent abrupt changes that cause the diversity and high amount of stochastic behaviour that lack in deterministic algorithms. *elites* are fundamental in GA, since it guarantees that the best fitted individuals with the best characteristics, i.e. *genes*, pass to the next generation. A generic GA pseudo-code is summarized in algorithm 1.

Algorithm 1 GA pseudo-code

```
Creation of initial population x
Computes the initial population fitness f_x
repeatfor the population of each generation
    Selects the Elite individual, x_elite based on f_x
    %_crossover of the remaining population, x − x_elite, suffers crossover
    %_mutation of the remaining population, x − x_elite, suffers mutation
    Save descendency as the current generation
until Stopping criteria condition satisfied
Save the best solution and its fitness f_x
```

3.4 Optimal path planning using minimum time criteria

The method here presented results from the decoupling of the minimum time optimization in a three stage optimization. The first stage is the determination of the limit speed for a given path. The limit speed is computed resorting not only to the maximum velocity the vehicle can achieve, but also the limit velocity imposed, during cornering, by the tyres lateral

friction. The second optimization stage is the velocity profile optimization that minimizes time, subject to dynamic and velocity limits constraints. The last stage is the refinement of the present solution. In this work, what is called the solution refinement is the use of an initial population, on a genetic algorithm, with $k - best$ trajectories, i.e. the k on a larger population resulting in lower times, in a optimization with a minimum time objective. To find this initial population another optimization is performed with the goal of minimizing the trajectory curvature by changing the position of a given number of trajectory control points. This method, although using other type of heuristics instead of genetic algorithms, was first developed for manipulator robots [12, 39] and later applied for mobile robots motion planning on uneven terrain [13, 40].

3.4.1 Determination of the limit velocities for a given path

The limit velocities considered in this work are the maximum velocity the vehicle can achieve,$v_{x,max}$, or any other desired, and the maximum velocity the vehicle can achieve during cornering without deviating from the desired path, which is derived from the centripetal acceleration, a_y.

$$v_{x,cornering}(s) = \sqrt{ay_{max}R(s)} \tag{25}$$

Equation 25 is obtained from the centripetal acceleration, $ay_{max} = \frac{v_x(s)^2}{R}$, with $R(s)$ as the radius as a function of the distance, s. Therefore, the velocity constrained by the acceleration is function of s. The maximum lateral acceleration must be identified through experiments or user defined, due to absence of tyre curves. The radius is computed as

$$R(s) = \frac{d\psi}{ds} \tag{26}$$

An alternative to define this dynamic constraint would be to define a friction coefficient, μ, and bound the maximum acceleration as

$$F = F_y + F_x \leq \mu F_z \tag{27}$$

This could be done for every every wheel or for the car itself. Nonetheless, such method would also require knowledge of tyre behaviour curves.

Another alternative method, would be to resort to equations 3 and 5 to compute F_x and F_y, however such methods are dependent on v_x and on the slip angle, α, which would turn the method under-determined. Moreover, it would also require full knowledge of the tyre curves.

To conclude, to obtain the admissible velocity as a function of the distance, i.e. as a function of each point in the path, one simply has to compute the minimum of both limit velocities at each point,

$$g(x, \dot{x}) = v_{x_{limit}} = \min_{\forall s \in S} [v_{x,max}, v_{x,cornering}] \tag{28}$$

with s as a given point of the interpolant curve S.

3.4.2 Solution of the minimum time velocity profile for a given path

The initial assumption is that the longitudinal velocity, v_x, which is influenced by the before mentioned factors, governs the optimal velocity profile. One is assuming that since the lateral dynamics influences the admissible velocity at a given position on the prescribed path, the vehicle is capable of following any trajectory as long as its longitudinal velocity verifies the next condition

$$f(x, \dot{x}) \leq g(x, \dot{x}) \tag{29}$$

with $f(x, \dot{x})$ as the equation governing the longitudinal dynamics, hence longitudinal velocity and acceleration.

The optimization problem to be solved is

$$\begin{aligned} \min_{\mathbf{u} \in \Re^n} \quad & t_f = \int_{s_0}^{s_f} \frac{ds}{\dot{s}} \\ \text{subject to} \quad & f(x, \dot{x}) \leq g(x, \dot{x}) \\ & -1 \leq \mathbf{u} \leq 1 \end{aligned} \tag{30}$$

With u as the control input that governs the longitudinal velocity,$\dot{s}$ and s_0 and s_f as the initial and final points of the function s.

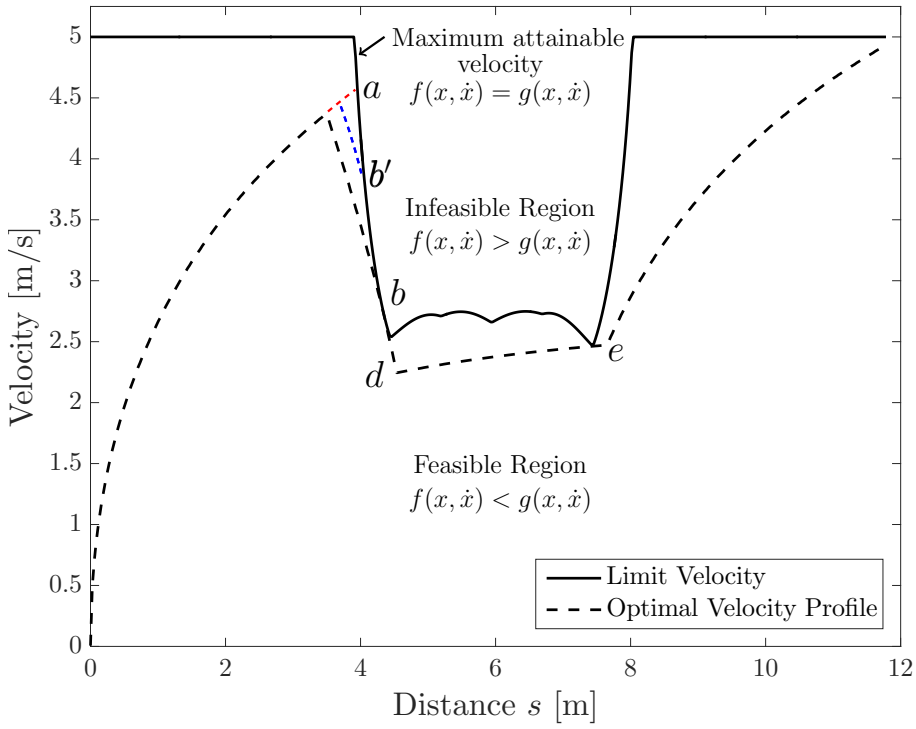

Figure 8: An example, for the path of figure 9 , showing the minimum time, or very similar, trajectory construction

This problem is solved resorting to algorithms presented in [39, 12], the solution, figure 8 for a given path, figure 8, follows the next steps, which was adapted(point 2.) to our problem to simplify implementation.

1. Forward integration accelerating, from the initial state (x_0,$\dot{x}_0$), of $f(x,\dot{x})$ until the maximum admissible velocity curve is reached, point a.

2. From point a go back to a lower value of s but still on the curve of point 1 and switch to decceleration. At this point only two things can occur: Either the trajectory intersect the limit velocity curve at b' or it will intersect the $x - axis$ in a given point, which has to be greater than s_f. (By going back in the curve of point 1, instead of following the original algorithm, which draws a vertical line from a until a defined velocity, the chance of the intersection of the curve with the x_axis can never be lower than s_f.) The goal here is to find the switching point that produces a decceleration curve that only intersects once the admissible velocity curve, point b. This process is done iteratively.

3. From the intersection point b, one most once again accelerate. A similar procedure

of point 2 must be followed, but progressing in s, until there is only one intersection between the generated curve and the limit curve. (If one finds a singularity a different approach must be taken)

4. Repeat points 1.,2. and 3. until $s = s_f$. This point follows the assumption that in any track a braking point is proceeded by an accelerating point and followed another accelerating point.

The aforementioned algorithm may be summarized in algorithm 2.

Algorithm 2 Minimum time v_x algorithm over a fixed path

```
Initialize dynamic equations
Compute the maximum attainable velocity function g(x, ẋ)
while s_k ≤ s_f do
    Forward integration of f(x, ẋ) with u = 1
    if f(x, ẋ) ≥ g(x, ẋ) then
        Compute k
        k = k − 1
        while true do
            Forward integration of f(x, ẋ) with u = −1
            if length(f(x, ẋ) = g(x, ẋ)) > 1 then
                k = k − 1
            else
                Break
            end if
        end while
    end if
end while
```

The solution for equation 30 is said to be bang-bang [12], meaning that the input u may take either the maximum value, 1, or the minimum value, -1, thus representing maximum acceleration or minimum acceleration. Such model only delivers the optimal, or near optimal, solution in special cases. When during a trajectory manipulator robot reaches a singularity this method will not present good results, in [41] was presented a solution to overcome manipulators singularities, but this solution applicability to a mobile robot must be further investigated.

The solution followed in this work was to evaluate the velocity limit values in an interval, computing its mean and then relating its value with the input u, as in equation 31. This may not be reckoned as an optimal but rather a practical solution, since it will must certainly create sub-optimal, but feasible and reasonable, longitudinal velocity profiles.

$$\bar{u} = \text{mean}\left(\sum_{s(d)}^{s(e)} \dot{s}\right) \tag{31}$$

Figure 9 is an example of the result of the interpolation over the control points. The longitudinal velocity and the velocity limits associated with this path are show in fig. 8.

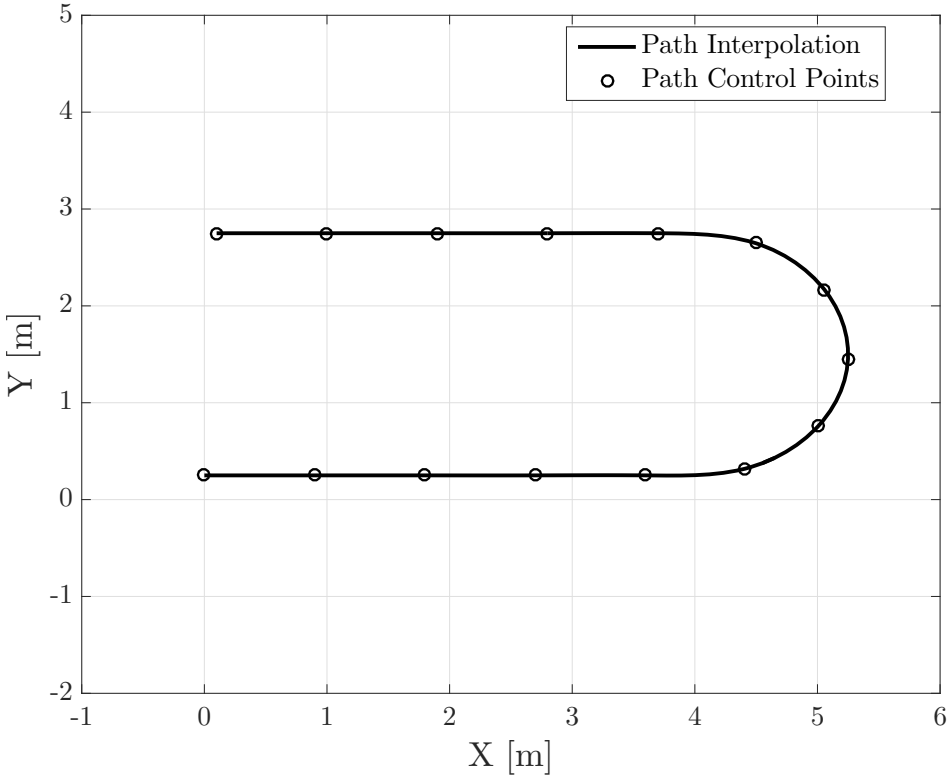

Figure 9: Representation of a given path, including the path control points and the interpolant curve

3.4.3 Minimum time trajectory solution

This is the third step of the optimization problem, which regards the time minimization for a given track. At this stage of the optimization, one is capable of not only computing the limit velocities for a given path but also the optimal, or sub-optimal, longitudinal velocity profile. To minimize time, one must change the position of the path control points and repeat the procedures of sections 3.4.1 and 3.4.2 to compute the resulting time until convergence is reached. Although, conceptually, this approach would deliver good results, such conclusion was not reached. The reason is that the change of the path control points is not obvious and, additionally, the function has plenty local minima. To avoid such problems, the approach followed was to use *a priori* knowledge, i.e. the minimum time trajectory on a race track is similar to the trajectory that minimizes the curvature, which is the inverse of equation 20. So to minimize time, two minimizations were performed, one to minimize curvature and another to minimize time. The k best results of the curvature minimization are used as the genetic algorithm initial population, thus resulting in a diverse population with individuals representing reasonable fit with characteristics near the minimum curvature solution, which is known to be related with the minimum time solutions.

The curvature minimization may be formulated as

$$\begin{aligned} \min_{\mathbf{x_c}} \quad & \sum_{i=s_0}^{s_f} \kappa_i = \sum_{i=s_0}^{s_f} \frac{1}{R_i} \\ \text{subject to} \quad & 0 < \sigma < 1 \end{aligned} \tag{32}$$

with x_c as the position of the path control points in the track, κ as the curvature, and σ as the position in the track as in equation 16.

After obtaining the results of the minimization in equation 32, the time associated with the best at each generation/iteration is presented in figure 32, for several runs of the software.

A thorough analysis of figure 10 reveals an unexpected result regarding the time, which is it increase with the reduction of the curvature. Furthermore, the stochastic/chaotic behaviour related with the lower values of curvature is also interesting and demonstrate the difficulty of minimizing time. This aspect also reveals sudden great changes that are associated with slightly different paths or trajectories. The trajectory of figure 11 appears to be very good, using a good line on the track, as it should be.

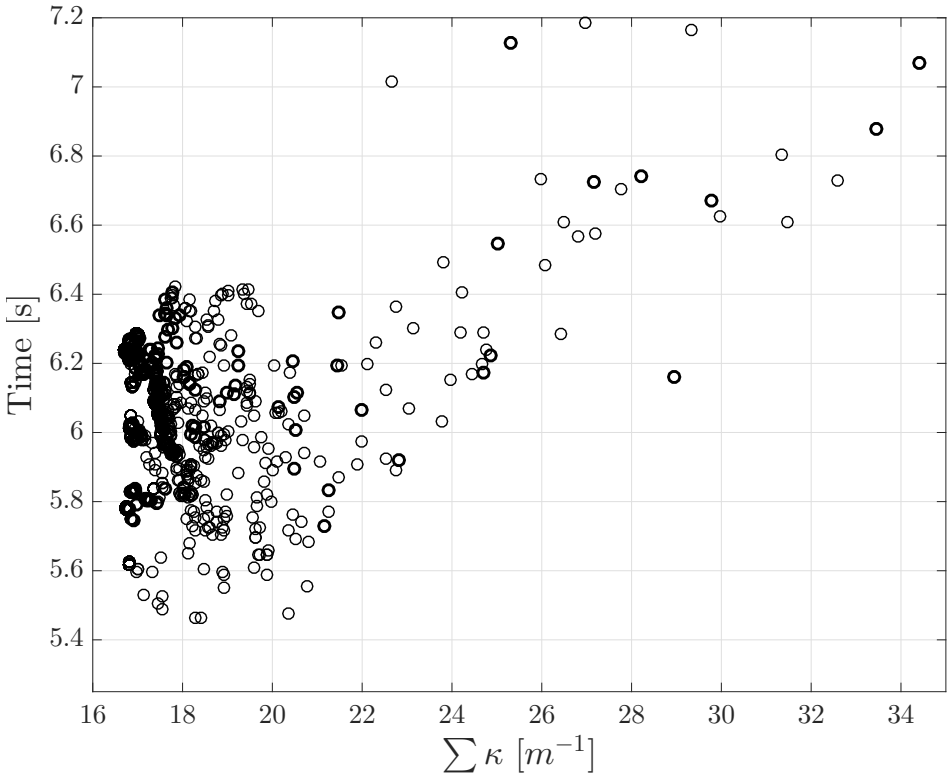

Figure 10: Each iteration best trajectory of the minimization of equation 32 and its associated time at the track, figure 11

The time minimization may be formulated as

$$\begin{aligned} \min_{\mathbf{x_c}} \quad & t_f = \int_{s_0}^{s_f} \frac{ds}{\dot{s}} \\ \text{subject to} \quad & 0 < \sigma < 1 \\ & f(x, \dot{x}) \leq g(x, \dot{x}) \end{aligned} \tag{33}$$

The optimization of equation 33 relies on changing the position of n control points, within the track limits, equation 16, to find the path and respective velocity profile that minimizes time, t_f. The pseudo-code of this algorithm can be inspected in algorithm 3.

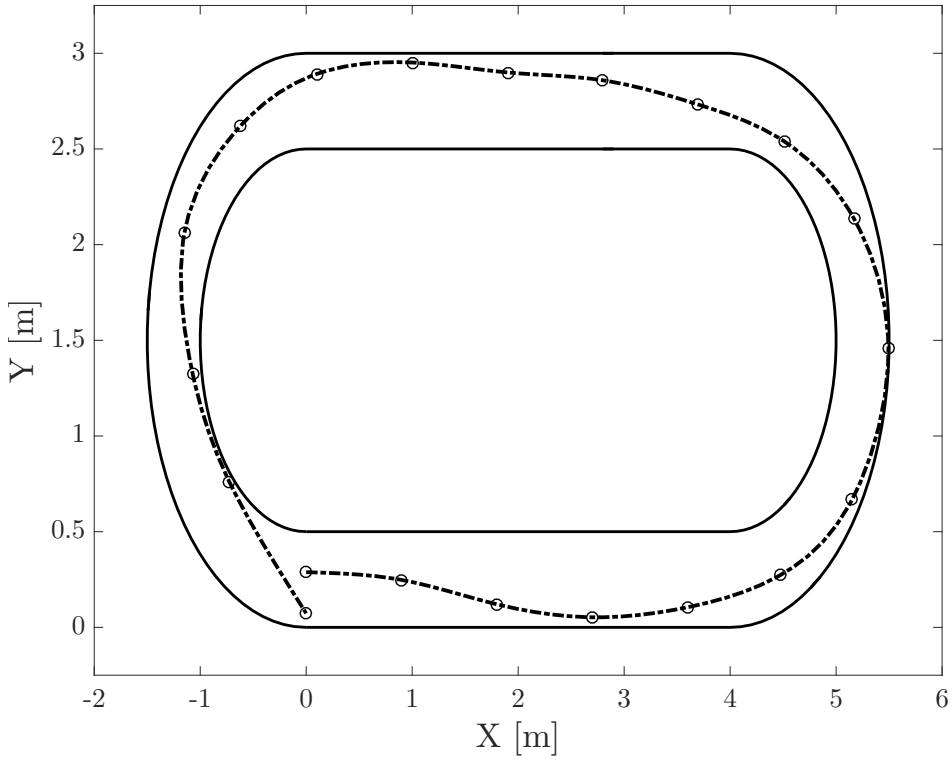

Figure 11: Representation of the track and illustration of a minimum curvature path result

Algorithm 3 Minimum time algorithm

Initialize lateral, longitudinal dynamics state space equations
Start $\sum \kappa_{best}$ as ∞
Minimize $\sum \kappa$, according to equation 32
for each individual, $\mathbf{x_c}$, **do**
    Interpolate with B-Splines
    Compute $\sum \kappa$
    if $\sum \kappa < \sum \kappa_{best}$ **then**
        Set $\sum \kappa_{best} = \sum \kappa$
        Save $\mathbf{x_c}$ in a new entry of **GA Initial Population**
    end if
end for
Start $t_{f_{best}}$ as ∞
Minimize t_f, equation 33, using **GA Initial Population**
for each individual, $\mathbf{x_c}$, **do**
    Interpolate with B-Splines
    Compute $\mathbf{v_{x_{limit}}}$
    Compute $\mathbf{v_{x_{optimal}}}$ and s, using procedure of section 3.4.2. Set $\dot{s} = \mathbf{v_{x_{optimal}}}$
    Compute $t_f = \int_{s_0}^{s_f} \frac{ds}{\dot{s}}$
    if $t_f < t_{f_{best}}$ **then**
        Set $t_{f_{best}} = t_f$
    end if
end for

4 Optimal control trajectory optimization

In this chapter a new set of coordinates is created, transforming the time dependent system in a space dependent system. The previously mentioned dynamics, transverse and longitudinal, in addition to the new set of coordinates enables the use of an optimal control formulation to solve the minimum time problem. This section is divided in three subsections: Dynamics reformulation, minimum time tracking over a fixed path and, finally, time optimal trajectory planning.

4.1 Minimum time path tracking

The minimum time path tracking problem concerns with the velocity profile optimization along a given path. This is an important problem that has been addressed mainly for robotic manipulators [18, 39, 13].

4.1.1 Reformulated dynamics

The bicycle model previously derived, equations 13, was modeled in the state-space form, equation 15. Notwithstanding, the same model can be derived by means of the Euler-Lagrange approach, using $q = (x, y, \psi)$ as generalized coordinates.

$$\frac{d}{dt}\frac{\partial \mathcal{L}^T}{\partial \dot{q}} - \frac{\partial \mathcal{L}^T}{\partial q} = \mathbf{u} \tag{34}$$

with $\mathbf{u}$ as the generalized forces and $\mathcal{L}(q, \dot{q}) = T(q, \dot{q}) - V(q)$, in which T and V as the kinetic the potential energies, respectively. Exploiting such set of equations results in

$$\mathcal{M}\ddot{q} + \mathcal{C}(\dot{q}) + \mathcal{F}(\dot{q}) = \mathbf{u} \tag{35}$$

where $\mathcal{M}$ is the mass matrix, $\mathbb{R}^{3x3}$, $\mathcal{C}$ is the matrix accounting for centripetal and resistive forces, i.e. drag, $\mathbb{R}^{3x1}$ and $\mathcal{F}$ is a matrix with the tyre friction related coefficients, $\mathbb{R}^{3x1}$.

$$
\begin{aligned}
\mathcal{M} &= \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix} \\
\mathcal{C}(\dot{q}) &= \begin{bmatrix} -m\dot{y}\dot{\psi} - K\dot{x}^2 \\ m\dot{x}\dot{\psi} \\ 0 \end{bmatrix} \\
\mathcal{F}(\dot{q}) &= \begin{bmatrix} 0 \\ C_{\alpha_f}(\frac{\dot{y}}{\dot{x}} + a_1\frac{\dot{\psi}}{\dot{x}}) + C_{\alpha_r}(\frac{\dot{y}}{\dot{x}} - a_2\frac{\dot{\psi}}{\dot{x}}) \\ a_1 C_{\alpha_f}(\frac{\dot{y}}{\dot{x}} + a_1\frac{\dot{\psi}}{\dot{x}}) - a_2 C_{\alpha_r}(\frac{\dot{y}}{\dot{x}} - a_2\frac{\dot{\psi}}{\dot{x}}) \end{bmatrix} \\
\mathbf{u} &= \begin{bmatrix} 4.6875m & 0 \\ 0 & C_{\alpha_f} \\ 0 & a_1 C_{\alpha_f} \end{bmatrix} \begin{bmatrix} u_p \\ \delta \end{bmatrix}
\end{aligned}
\tag{36}
$$

where u_p is the pedal position, $-1 \leq u_p(t) \leq 1$ and K represents a resistive force coefficient, resultant from drag and rolling resistance.

Over a fixed path in the car joint space coordinates, $q(s)$, is function of a scalar s representing the distance, which is time-dependent, $s(t)$. In this work was considered that the distance is normalized, i.e. $0 \leq s(t) \leq 1$, with the time starting at $t = 0$ and ending at $t = T$. One has to impose the constraint that $\dot{s} \geq 0$. From here on, the time dependent characteristic of the path s will be omitted when possible. Resorting to the chain rule one can rewrite the time dependent generalized coordinates as a function of the path variable s

$$\dot{q} = \frac{dq}{ds}\frac{ds}{dt} = q'\dot{s} \tag{37}$$

$$\ddot{q} = \frac{dq}{ds}\frac{d^2s}{dt^2} + \frac{d}{ds}\left(\frac{dq}{ds}\right) = q'\ddot{s} + q''\dot{s}^2 \tag{38}$$

Using equations 37 and 38 one can rewrite equation 35 as

$$\mathbf{u}(s) = \mathbf{m}(s)\ddot{s} + \mathbf{c}(s)\dot{s} + \mathbf{f}(s) \tag{39}$$

in which $\mathbf{m}(s)$, $\mathbf{c}(s)$ and $\mathbf{f}(s)$ are defined as

$$\begin{aligned}\mathbf{m}(s) &= \mathcal{M}q'(s) \\ \mathbf{c}(s) &= \mathcal{M}q''(s) + \mathcal{C}(q'(s))q'(s) \\ \mathbf{f}(s) &= \mathcal{F}(q'(s))\end{aligned} \tag{40}$$

At this point the car motion dynamics was transformed to be space dependent instead of time dependent, aspect that will be further explored and allow an optimal control formulation for the minimum time trajectory problem.

4.1.2 Problem Formulation

Making use of the dynamics governing equations previously derived one can write the general minimum time problem as

$$\min_{T,s,\mathbf{u}(\cdot)} T \tag{41}$$

$$\text{subject to } \mathbf{u}(t) = \mathbf{m}(s(t))\ddot{s}(t) + \mathbf{c}(s(t))\dot{s}^2(t) + \mathbf{f}(\dot{s}(t)) \tag{42}$$

$$s(0) = 0 \tag{43}$$

$$s(T) = 1 \tag{44}$$

$$\dot{s}(0) = \dot{s}_0 \tag{45}$$

$$\dot{s}(T) = \dot{s}_T \tag{46}$$

$$\dot{s}(t) \geq 0 \tag{47}$$

$$\dot{s}(t) \geq 0 \tag{48}$$

$$\underline{\mathbf{u}}(s(t)) \leq \mathbf{u}(\mathbf{t}) \leq \bar{\mathbf{u}}(s(t)) \tag{49}$$

$$\text{for } t \in [0, T] \tag{50}$$

in which $\underline{\mathbf{u}}$ and $\bar{\mathbf{u}}$ represent the input lower and upper bounds, respectively.

A closer inspection at equations 41 to 50 do not reveal any information about the locally and/or globally solutions of the problem, since the convexity of the objective function is not obvious, despite the constraints being linear. To conclude on such matter, one might recall to [22] and [14], in which the formulations are quite similar, with the difference that in the latter the problem is of minimizing energy and in the former is of minimizing time. In both cases,

the formulation is of an optimal control problem stating linear system dynamics, differential states $(s, \dot{s})^T$ and control input $\ddot{s}$. This set of statements are the necessary conditions to define a problem as convex, leading to the important conclusion that any local optimal is also global optimal. To accomplish the aforementioned convex problem one must start by doing a change of the integration variable of equation 41 from time, t, to space, s,

$$T = \int_0^T 1dt = \int_{s(0)}^{s(T)} \frac{ds}{\dot{s}} = \int_0^1 \frac{ds}{\dot{s}} \tag{51}$$

Second, two new optimization variables are introduced

$$a(s) = \ddot{s} \tag{52}$$

$$f(s) = \dot{s}^2 \tag{53}$$

To ensure relation between $a(s)$ and $f(s)$ an additional constraint is added

$$f(s) = 2a(s) \tag{54}$$

which comes from

$$\dot{f}(s) = f'\dot{s} \tag{55}$$

and

$$\dot{f}(s) = \frac{d(\dot{s}^2)}{dt} = 2\ddot{s}\dot{s} = 2a(s)\dot{s} \tag{56}$$

After doing this set of transformations and recognizing $a(s)$ and $f(s)$ as optimization variables, the problem described by equations 41 to 50 is rewritten as

$$\min_{a(\cdot),f(\cdot),\mathbf{u}(\cdot)} \int_0^1 \frac{ds}{\sqrt{f(s)}} \tag{57}$$

$$\text{subject to } \mathbf{u}(t) = \mathbf{m}(s)a(s) + \mathbf{c}(s)f(s) + \mathbf{f}(s) \tag{58}$$

$$f(0) = \dot{s}_0^2 \tag{59}$$

$$f(1) = \dot{s}_T^2 \tag{60}$$

$$f'(s) = 2a(s) \tag{61}$$

$$f(s) \geq 0 \tag{62}$$

$$\underline{\mathbf{u}}(s(t)) \leq \mathbf{u}(t) \leq \bar{\mathbf{u}}(s(t)) \tag{63}$$

$$\text{for } s \in [0, 1] \tag{64}$$

It is easy to prove that problem 57-64 is convex, due to the constraints linearity and convexity of its function, since a power function is convex [42]. This problem, in this form, can be considered in differential algebraic form (DAE), with pseudo-time s, control input a(s), differential state b(s), algebraic states $\mathbf{u}(s)$, linear system dynamics 61, linear state dependent contraints 58, 62 and 64, as well as initial and final conditions [22].

Furthermore, a great advantage of such an humble formulation is the simplicity of adding extra relevant objective function terms, that ensure a weighting to the main objective of minimizing time.

There are several options for additional constraints, such as thermal energy, the integral of the absolute value of the inputs rate of change, velocity constraints, acceleration constraints, rate of torque change, slip angle constraints, among much more. In this work, it will only be considered the thermal energy, the integral of the absolute value of the inputs rate of change and slip angle constraints.

The thermal energy is defined as

$$\int_0^1 u_i^2(s)dt = \int_0^1 \frac{u_i^2(s)}{\dot{s}}ds = \int_0^1 \frac{u_i^2(s)}{\sqrt{f(s)}}ds \tag{65}$$

The integral of the absolute value of the inputs rate of change is defined as

$$\int_0^T |\dot{u}_i(s)|dt = \int_0^1 \frac{|u_i'(s)\dot{s}|}{\dot{s}}ds = \int_0^1 |u_i'(s)|ds \tag{66}$$

By adding this pair of equations, 65 and 66, to the objective function, the effect is, respectively, not only a penalty on the input signal energy, which results in lower absolute values, but also a penalty on inputs rate of change, which creates input signals with smoother transitions. In this way one is not restricting energy or rate of changes but penalizing its occurrence, fact that helps the solver convergence. Moreover, both equations 65 and 66 convexity proof is quite easy and is shown in [42], thus the final objective function is convex.

An additional constraint may be added to limit the slip angle, $\alpha_{f,r}$, since the cornering stiffness was linearized, higher slip angles will devise greater lateral forces. By imposing such limit, one is deviating the tyre from working near or above the saturation limit. The slip angle is merely a function of δ and generalized coordinates derivatives, $\dot{q}$.

$$\underline{\alpha} \leq \alpha_{f,r} \leq \bar{\alpha} \Leftrightarrow \underline{\alpha} \leq g(\dot{q}, \delta) \leq \bar{\alpha} \tag{67}$$

where $g(\dot{q}, \delta)$ is the function describing the slip angle and $\underline{\alpha}$ and $\bar{\alpha}$ are the slip angle lower and upper limits, respectively.

Gathering the supplementary objective functions, equations 65 and 66, and constraints, equation 67, and adding to the general problem 57-64, outcomes a richer formulation of the problem

$$\min_{a(\cdot),f(\cdot),\mathbf{u}(\cdot)} \int_0^1 \frac{1}{\sqrt{f(s)}} + \frac{\gamma_1}{\sqrt{f(s)}} \left(\sum_{i=1}^{n} \frac{u_i^2(s)}{\bar{u}_i^2} \right) + \gamma_2 \left(\sum_{i=1}^{n} \frac{|u_i'(s)|}{|\bar{u}_i|} \right) ds \tag{68}$$

$$\text{subject to } \mathbf{u}(s) = \mathbf{m}(s)a(s) + \mathbf{c}(s)f(s) + \mathbf{f}(s) \tag{69}$$

$$f(0) = \dot{s}_0^2 \tag{70}$$

$$f(1) = \dot{s}_T^2 \tag{71}$$

$$f'(s) = 2a(s) \tag{72}$$

$$f(s) \geq 0 \tag{73}$$

$$\underline{\alpha}(s) \leq g(\dot{q}(s), \delta(s)) \leq \bar{\alpha}(s) \tag{74}$$

$$\underline{\mathbf{u}}(s(t)) \leq \mathbf{u}(t) \leq \bar{\mathbf{u}}(s(t)) \tag{75}$$

$$\text{for } s \in [0, 1] \tag{76}$$

The problem formulated in 68-76 represents the generalized and richer optimal control formulation considered in this work. As mentioned before the additional objective functions preserve convexity and the added constraint is linear. Furthermore, $\bar{u}_i$ is the maximum value in an interval, most frequently it is the upper torque bound. γ_1 and γ_2 are constants that define the desired weight on the penalties, equations 65 and 66. By increasing γ_1, the input signal tends to loose energy, deviating from higher values. By increasing γ_2 one tends to obtain smoother signals, i.e. with less abrupt changes, even though some of them are necessary. It is worth mentioning that such extra functions result in non-optimal solutions when compared with the ones obtained with 57-64.

4.1.3 Numerical Solution

To solve the generalized optimal control problem there are three well recognized methods. Dynamic programming methods [18, 15, 43], which are by nature a decision-based method instead of a gradient based, falling, however, in a group of methods that require a very careful implementation due to the sequential way this method solve the problem. Furthermore, dynamic programming is $\mathcal{O}(n)$, requiring, therefore, considerable computation time. The second method falls under the category of indirect methods, which are characterized by the several sub-steps necessary to solve the problem [13, 12]. At last, most recently, methods

such as the direct method, motivated by the work developed by [19, 20] and others, allowed the solution of the optimal control using quadratic and/or convex optimization methods. This set of solutions is also called direct transcription. The previously extensive problem formulation was fundamental to achieve a convex optimal control problem, thus enabling the use of mature solvers, such as SeDuMi [44] or CVX [45], as suggested by [22].

In this work the approach was to use direct transcription, which is the reformulation, in the form of a large sparse optimization problem, of the optimal control problem 68-76. To get to such result, one first discretize the path s, with $s \in [0, 1]$, which results in $K+1$ points, as in equation 77.

$$s^0 = 0 \leq s^k \leq s^K \text{for } k = 0, ..., k \tag{77}$$

Then, the acceleration, $a(s)$, velocity, $f(s)$, and inputs, u_i, are also discretized, with the functions evaluated on each of the path points, or, in some cases, at the middle points. The number of grid points used and the manner one evaluates the functions, either at the point or at middle points, define the direct transcription method used.

The approach chosen was to, as mentioned before, set the accelerations, $a(s)$, as the optimal control input and, therefore, to assume such function as piecewise constant, figure 12. By doing this, naturally, it is implied that the velocity, $f(s)$ is piecewise linear, figure 13, and the inputs, u_i are piecewise nonlinear, figure 14. So, following such transcription, it is useful to define f^k to the grid points s^k and one can define the continuous velocity function, due to the assumption of piecewise linear, as

$$f(s) = f^k + \frac{f^{k+1} - f^k}{s^{k+1} - s^k}(s - s^k) \quad \text{for } s \in [s^k, s^{k+1}] \tag{78}$$

From here on, it is assumed that $f(s^k) = f^k$ and, by ease of implementation, a^k and u_i^k are evaluated at the middle points, particularly at $s^{k+\frac{1}{2}}$, as can be inspected in figures 12 to 14. Hence, $a^k = a(s^{k+\frac{1}{2}})$ and $u_i^k = u_i(s^{k+\frac{1}{2}})$.

With such discretization assumptions in mind one can perform additional operations on the objective function integral first two terms, equation 68, which will simplify implementation and algorithm efficiency

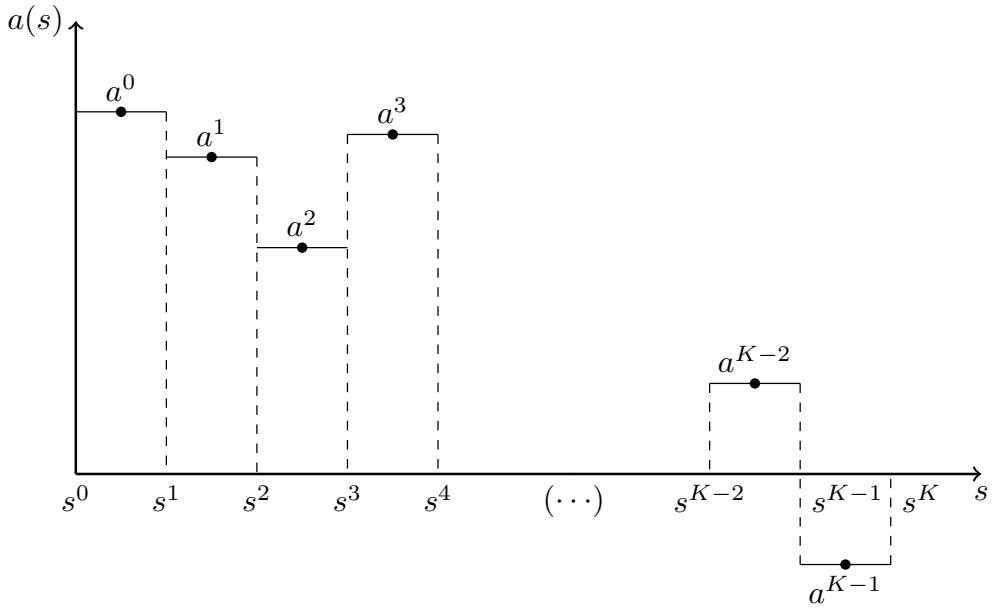

Figure 12: Acceleration, $a(s)$, represented as a piecewise constant function for $s \in [0, 1]$

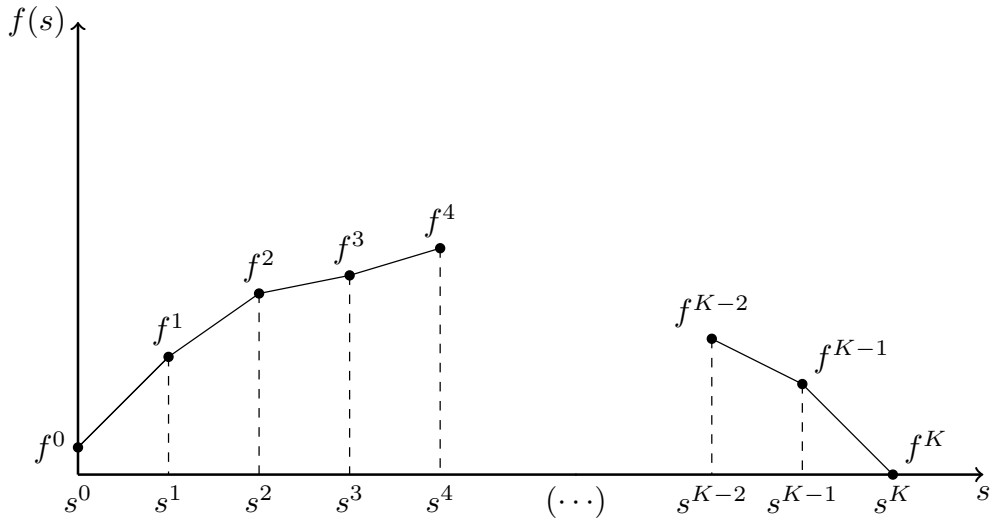

Figure 13: Velocity, $f(s)$, represented as a piecewise linear function for $s \in [0, 1]$

$$\begin{aligned}
&\int_0^1 \frac{ds}{\sqrt{f(s)}} + \frac{\gamma_1}{\sqrt{f(s)}} \left(\sum_{i=1}^n \frac{u_i^2(s)}{\bar{u}_i^2} \right) + \gamma_2 \left(\sum_{i=1}^n \frac{|u_i'(s)|}{|\bar{u}_i|} \right) \quad = \\
&= \sum_{k=0}^{K-1} \int_{s^k}^{s^{k+1}} \left[\frac{1}{\sqrt{f(s)}} + \frac{\gamma_1}{\sqrt{f(s)}} \left(\sum_{i=1}^n \frac{u_i^2(s)}{\bar{u}_i^2} \right) \right] ds = \\
&\approx \sum_{k=0}^{K-1} \left[1 + \gamma_1 \left(\sum_{i=1}^n \frac{\left(u_i^k\right)^2}{u_i^2} \right) \right] \int_{s^k}^{s^{k+1}} \frac{ds}{\sqrt{f(s)}}
\end{aligned} \tag{79}$$

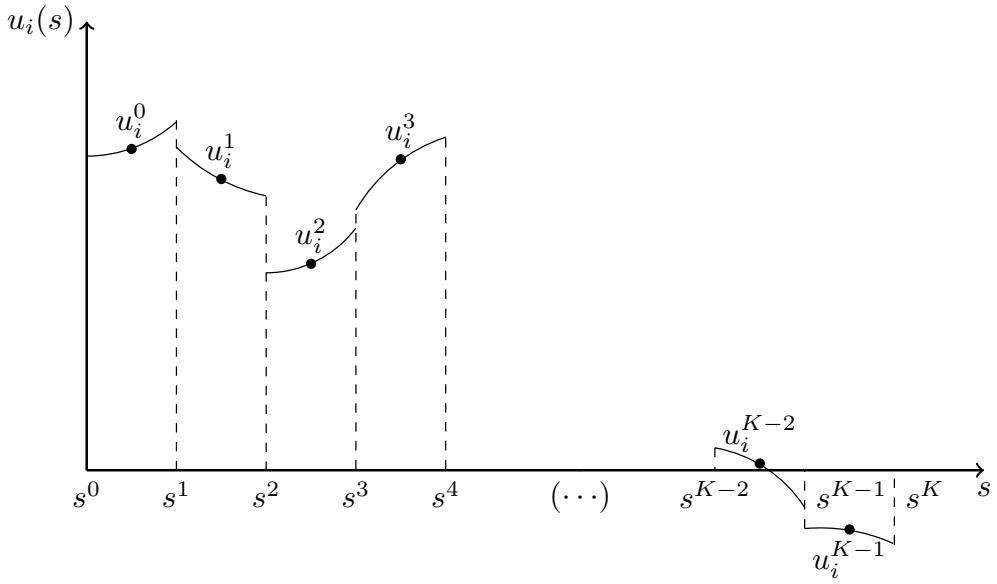

Figure 14: Inputs, $u_i(s)$, represented as piecewise nonlinear functions for $s \in [0, 1]$

To deal with the eventual singularities on the integral of the last term of equation 79, namely when $f(s) = 0$, one treat the term $\frac{1}{\sqrt{f(s)}}$ separately using equation 78, which results in

$$\sum_{k=0}^{K-1} \left[1 + \gamma_1 \left(\sum_{i=1}^{n} \frac{\left(u_i^k\right)^2}{u_i^2} \right) \right] \frac{2\Delta s^k}{\sqrt{f^k k} + \sqrt{f^k k + 1}} \tag{80}$$

with $\Delta s^k = s^{k+1} - s^k$. To conclude the discrete reformulation of the objective one still has to deal the third term of equation 68, which may be rewritten as

$$\gamma_2 \int_0^1 \frac{|u_i'(s)|}{|\bar{u}_i|} ds \approx \gamma_2 \sum_{k=0}^{K-1} \frac{|\Delta u_i^k|}{|\bar{u}_i|} \tag{81}$$

with $\Delta u_i^k = u_i^{k+1} - u_i^k$. The final optimal control problem in the discretized form obtained with the direct transcription is

$$\min_{a^k, f^k, \mathbf{u}^k} \sum_{k=0}^{K-1} \frac{2\Delta s^k \left(1 + \gamma_1 \sum_{i=1}^{n} \frac{\left(u_i^k\right)^2}{\bar{u}_i^2}\right)}{\sqrt{f^k} + \sqrt{f^{k+1}}} + \gamma_2 \sum_{k=0}^{K-1} \left(\sum_{i=1}^{n} \frac{|\Delta u_i^k|}{|\bar{u}_i|} \right) \tag{82}$$

$$\text{subject to } \mathbf{u}^k = \mathbf{m}(s^{k+\frac{1}{2}})a^k + \mathbf{c}(s^{k+\frac{1}{2}})f(s) + \mathbf{f}(s^{k+\frac{1}{2}}) \tag{83}$$

$$f(0) = \dot{s}_0^2 \tag{84}$$

$$f(K) = \dot{s}_T^2 \tag{85}$$

$$(f^{k+1} - f^k) = 2a^k\Delta^k \tag{86}$$

$$f^k \geq 0 \tag{87}$$

$$\underline{\alpha}(s^{k+\frac{1}{2}}) \leq g(\dot{q}(s^{k+\frac{1}{2}}), \delta^k) \leq \bar{\alpha}(s^{k+\frac{1}{2}}) \tag{88}$$

$$\underline{\mathbf{u}}(s^{k+\frac{1}{2}}) \leq \mathbf{u}^k \leq \bar{\mathbf{u}}(s^{k+\frac{1}{2}}) \tag{89}$$

$$\text{for } s \in [0, 1] \tag{90}$$

Once again it is easy to conclude that the operations performed on the problem formulations did not change the convexity of the final optimal control problem after the direct transcription, equations 82-90. By concluding this, one is assuring that this problem can be solved by any nonlinear optimization solver. Notwithstanding, the problem can still be changed to allow the use of more efficient solvers previously referenced, such as SeDuMi and CVX, which take advantage of an optimization technique called Second Order Cone Programming (SOCP). The final stage of this section is precisely the reformulation of equations 82-90 as SOCP.

4.1.4 Second order cone programming Formulation

The second order cone program is a type of optimization in which a linear function is minimized over the intersection of an affine set and the product of second-order (quadratic) cones [46]. Furthermore, SOCP are nonlinear convex problems that include not only linear but also quadratic programs as particular cases. One of the main advantages of SOCP are the several primal-dual interior-point methods that have been developed over the last two decades, which allow the solution of difficult problems in a very fast and efficient fashion. A wide band of applications of optimal control problems have been successfully solved using

SOCP, power electronics [47], switching problems [48], smart buildings climate control [49] and, finally, optimal path tracking for manipulator robots [22].

A second order cone programming problem is of the form

$$\min h^T x \tag{91}$$

$$\text{subject to } \parallel A_i x + b_i \parallel \leq c_i^T x + d_i,\ i = 1, ..., N \tag{92}$$

$$\tag{93}$$

with $x \in \mathbb{R}^n$ as the optimization variable, and the problem parameters are $h \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{n_i - 1}$, $b_i \in \mathbb{R}^{n_i - 1}$, $c_i \in \mathbb{R}^n$ and $d_i \in \mathbb{R}$. The operator $\|\|$ is defined as the standard Euclidean norm, $\parallel x \parallel = \left(x^T x\right)^{\frac{1}{2}}$.

The second order cone constraint of dimension n_i is the inequation

$$\parallel A_i x + b_i \parallel \leq c_i^T x + d_i \tag{94}$$

The standard cone of dimension k, which is also called Lorentz cone, is defined as

$$\mathcal{H}_k = \left\{ \begin{bmatrix} x \\ w \end{bmatrix} | x \in \mathbb{R}^{k-1}, w \in \mathbb{R}, \parallel x \parallel \leq w \right\} \tag{95}$$

The set of points defining a second order cone constraint are given by

$$\parallel A_i x + b_i \parallel \leq c_i^T x + d_i \Leftrightarrow \begin{bmatrix} A_i \\ c_i^T \end{bmatrix} x + \begin{bmatrix} c_i \\ d_i \end{bmatrix} \in \mathcal{H}_{n_i} \tag{96}$$

which is convex, thus SOCP is a convex programming problem since the objective function and the constraints both define convex sets. Several suitable SOCP objective functions and constraints can be found in [46].

To what concerns the scope of this work, one has to reformulate the objective function and constraints of problem 82-90, and to that end a new set of optimization variables must be introduced. An issue when adding new optimization variables is that frequently the physical meaning is lost, but, as mentioned before, this is done in regard of substantial increase of computational efficiency.

To rewrite equation the previous objective function 82 as linear, one has to introduce slack

variables d^k with $k = 0, ..., K-1$ and $\mathbf{e}^k \in \mathbb{R}^n$ for $k = 1, ..., K-1$, in a way that equation 82 may be rewritten as

$$\sum_{k=0}^{K-1} 2\Delta s^k d^k + \gamma_2 \sum_{k=1}^{max} \mathbf{1^T} \mathbf{e}^k \tag{97}$$

with $\mathbf{1} \in \mathbb{R}^n$ as dimension n vector with all entries equal to 1. It is now necessary to augment the problem 82-90 with the inequalities contraints [22]

$$\frac{1 + \gamma_1 \sum_n^{i=1} \frac{\left(u_i^k\right)^2}{\bar{u}_i^2}}{\sqrt{f^{k+1}} + \sqrt{f^k}} \leq d_k, \quad \text{for } k = 0, ..., K-1 \tag{98}$$

and

$$-\mathbf{e}^k \leq \begin{bmatrix} \frac{\Delta u_1^k}{|\bar{u}_1|} \\ \vdots \\ \frac{\Delta u_n^k}{|\bar{u}_n|} \end{bmatrix} \leq \mathbf{e}^k, \quad \text{for } k = 0, ..., K-1 \tag{99}$$

Additionally, to obtain a SOCP, constraints 98 must be replaced by two equivalent constraints by introducing variables c^k for $k = 0, ..., K$ as

$$\begin{aligned} \frac{1 + \gamma_1 \sum_n^{i=1} \frac{\left(u_i^k\right)^2}{\bar{u}_i^2}}{\sqrt{c^{k+1}} + \sqrt{c^k}} \leq & d_k \quad , \text{for } k = 0, ..., K-1 \\ c^k \leq & \sqrt{f^k} \quad , \text{for } k = 0, ..., K \end{aligned} \tag{100}$$

Now, conditions 100 may be rewritten as two second order cone constraints

$$\begin{aligned} & \frac{1 + \gamma_1 \sum_n^{i=1} \frac{\left(u_i^k\right)^2}{\bar{u}_i^2}}{\sqrt{c^{k+1}} + \sqrt{c^k}} \leq d_k \Leftrightarrow \\ \Leftrightarrow & \left\| \begin{matrix} 2 \\ 2\sqrt{\gamma_1} \frac{u_1^k}{\bar{u}_1} \\ \vdots \\ 2\sqrt{\gamma_1} \frac{u_n^k}{\bar{u}_n} \\ c^{k+1} + c^k - d^k \end{matrix} \right\| \leq c^{k+1} + c^k + d^k \\ & \text{for } k = 0, ..., K-1 \end{aligned} \tag{101}$$

and

$$c^k \leq \sqrt{b^k} \Leftrightarrow \left\| \begin{matrix} 2c^k \\ b^k - 1 \end{matrix} \right\| \leq b^k + 1, \quad \text{for } k = 0, ..., K \tag{102}$$

After defining the new linear objective function and the second order cone constraints one can finally arrive at the minimum time problem formulation in the classic SOCP form 91-92

$$\min_{a^k, f^k, \mathbf{u}^k, c^k, d^k, e^k} \sum_{k=0}^{K-1} 2\Delta s^k d^k + \gamma_2 \sum_{k=1}^{K-1} \mathbf{1}^T \mathbf{e}^k \tag{103}$$

$$\text{subject to } \mathbf{u}^k = \mathbf{m}(s^{k+\frac{1}{2}})a^k + \mathbf{c}(s^{k+\frac{1}{2}})f(s) + \mathbf{f}(s^{k+\frac{1}{2}}) \tag{104}$$

$$f(0) = \dot{s}_0^2 \tag{105}$$

$$f(K) = \dot{s}_T^2 \tag{106}$$

$$(f^{k+1} - f^k) = 2a^k \Delta^k \tag{107}$$

$$\underline{\alpha}(s^{k+\frac{1}{2}}) \leq g(\dot{q}(s^{k+\frac{1}{2}}), \delta^k) \leq \bar{\alpha}(s^{k+\frac{1}{2}}) \tag{108}$$

$$\underline{\mathbf{u}}(s^{k+\frac{1}{2}}) \leq \mathbf{u}^k \leq \bar{\mathbf{u}}(s^{k+\frac{1}{2}}) \tag{109}$$

$$\left\| \begin{matrix} 2 \\ 2\sqrt{\gamma_1}\frac{u_1^k}{\bar{u}_1} \\ \vdots \\ 2\sqrt{\gamma_1}\frac{u_n^k}{\bar{u}_n} \\ c^{k+1} + c^k - d^k \end{matrix} \right\| \leq c^{k+1} + c^k + d^k, \text{ for } k = 0, ..., K-1 \tag{110}$$

$$\left\| \begin{matrix} 2c^k \\ b^k - 1 \end{matrix} \right\| \leq b^k + 1 \text{ and } f^k \geq 0, \text{ for } k = 0, ..., K \tag{111}$$

$$-\mathbf{e}^k \leq \begin{bmatrix} \frac{\Delta u_1^k}{|\bar{u}_1|} \\ \vdots \\ \frac{\Delta u_n^k}{|\bar{u}_n|} \end{bmatrix} \leq \mathbf{e}^k, \text{ for } k = 1, ..., K-1 \tag{112}$$

$$\text{for } s \in [0, 1] \tag{113}$$

The formulation as a SOCP may not be as simple as 82-90, due to the introduction of auxiliary variables a^k,c^k, d^k and e^k, yet it allows the use of much efficient solvers, for further details see [22]. Using this formulation, after obtaining a solution, one uses the variable b^k to compute the minimum time over the fixed path

$$t(s) = \int_0^1 \frac{1}{\sqrt{b(s)}} ds \tag{114}$$

4.2 Minimum Time Trajectory

In the previous sections one was concerned in determining the inputs that lead to the minimum time velocity profile. By other words, for a prescribed path, a formulation that successfully minimize time was explained, computed and tested. Next, the problem of computing the path and vehicle inputs that result in minimum time trajectory is addressed. First, the definition of trajectory shall be explained. Trajectory differs from path in the way that not only that the waypoints in space are prescribed but also the velocity. This is the reason why in previous section one was minimizing time for path tracking and in this section one is fully addressing trajectory optimization.

Trajectory optimization is a complex and difficult problem. For many years, inside the scientific community, one could not find efficient and successful implementations. On the other hand, commercial software such as ADAMS [50], Milliken [51], OptimumDynamics [52], among others, have such kind of software implemented for years. It is worth mentioning, that within the afore mentioned software one can not have access to the problem formulation or methods used, which highly limit a thorough investigation. Mainly for the last two decades, with the increase of computational power, which motivated research for new optimization methods, some new formulations for minimum lap time appeared.

4.2.1 Reformulated Dynamics

The dynamics shall be reformulated to account with path change. But first must be explained how the track parametrization was done. For sake of simplicity, it is assumed that the vehicle moves along the track centerline, from the initial point s_0 until the final point s_f. To allow the car deviate from this path, a vector $\vec{n}(s)$, orthogonal to the body $x - axis$, is defined at each point s as $\mathbf{n(s)} = [\sin(\psi_0(s)), -\cos(\psi_0(s))]$. Furthermore, the car transversal car position may vary from $-b_{max}$ until b_{max} along the direction $\vec{n}(s)$, figure 15. Thus, the transversal car position is defined as $b(s) \in [-b_{max}(s), -b_{max}(s)]$. Moreover, defining $r_0(s) = (x_0(s), y_0(s))$ one arrives at

$$r(s) = r_0(s) + b(s)\mathbf{n}(\mathbf{s}) \Leftrightarrow \begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} x_0(s) \\ y_0(s) \end{bmatrix} + b(s) \begin{bmatrix} \sin(\psi_0(s)) \\ -\cos(\psi_0(s)) \end{bmatrix} \tag{115}$$

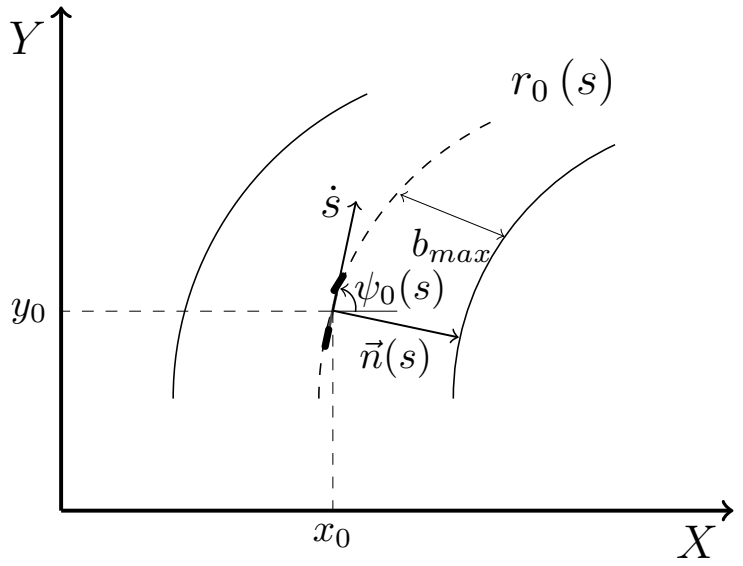

Figure 15: Representation of the track parametrization adopted

Once again one is assuming a normalized path vector, $s(0) = 0 \leq s(t) \leq s(T) = 1$, with T as the final time. As in the previous method, the chain rule was applied in order to change time dependency to spatial dependency

$$\begin{aligned} v_x = \dot{x} = x'\dot{s} \quad a_x = \ddot{x} = x'\ddot{s} + x''\dot{s}^2 \\ v_y = \dot{y} = y'\dot{s} \quad a_y = \ddot{y} = y'\ddot{s} + y''\dot{s}^2 \end{aligned} \tag{116}$$

Computing now the first and second derivative of x and y, in equation 116, with respect to s

$$\begin{aligned} x'(s) &= x_0'(s) + b'(s)\sin(\psi_0(s)) + b(s)\psi_0'\cos(\psi_0(s)) \\ y'(s) &= y_0'(s) - b'(s)\cos(\psi_0(s)) + b(s)\psi_0'\sin(\psi_0(s)) \\ x''(s) &= x_0''(s) + b''(s)\sin(\psi_0(s)) + 2b'(s)\psi_0'(s)\cos(\psi_0(s)) + b\psi_0''(s)\cos(\psi_0(s)) - b\psi_0'^2(s)\sin(\psi_0(s)) \\ y''(s) &= y_0''(s) - b''(s)\cos(\psi_0(s)) + 2b'(s)\psi_0'(s)\sin(\psi_0(s)) + b\psi_0''(s)\sin(\psi_0(s)) + b\psi_0'^2(s)\cos(\psi_0(s)) \end{aligned} \tag{117}$$

To increase readability of the afore further equations, one adds the following notation for the vector $\vec{n}(s)$ and its first and second derivatives taken for s

$$\begin{aligned}
n_x(s) &= \sin(\psi_0(s)) \\
n_y(s) &= -\cos(\psi_0(s)) \\
n'_x(s) &= \psi'_0(s)\cos(\psi_0(s)) \\
n'_y(s) &= \psi'_0(s)\sin(\psi_0(s)) \\
n''_x(s) &= \psi''_0(s)\sin(\psi_0(s)) - \psi'^2_0(s)\sin(\psi_0(s)) \\
n''_y(s) &= \psi''_0(s)\sin(\psi_0(s)) + \psi'^2_0(s)\cos(\psi_0(s))
\end{aligned} \tag{118}$$

Combining equations 117 and 118 one arrives at

$$\begin{aligned}
x'(s) &= x'_0(s) + b'(s)n_x(s) + b(s)n'_x(s) \\
y'(s) &= y'_0(s) + b'(s)n_y(s) + b(s)n'_y(s) \\
x''(s) &= x''_0(s) + b''(s)n_x(s) + 2b'(s)n'_x(s) + b(s)n''_x(s) \\
y''(s) &= y''_0(s) + b''(s)n_y(s) + 2b'(s)n'_y(s) + b(s)n''_y(s)
\end{aligned} \tag{119}$$

Substituting equations 119 on equations 116, one obtains

$$\begin{aligned}
v_x(s) &= \left[x'_0(s) + b'(s)n_x(s) + b(s)n'_x(s)\right]\dot{s} \\
v_y(s) &= \left[y'_0(s) + b'(s)n_y(s) + b(s)n'_y(s)\right]\dot{s} \\
a_x(s) &= \left[x'_0(s) + b'(s)n_x(s) + b(s)n'_x(s)\right]\ddot{s} + \left[x''_0(s) + b''(s)n_x(s) + 2b'(s)n'_x(s) + b(s)n''_x(s)\right]\dot{s}^2 \\
a_y(s) &= \left[x'_0(s) + b'(s)n_y(s) + b(s)n'_y(s)\right]\ddot{s} + \left[y''_0(s) + b''(s)n_y(s) + 2b'(s)n'_y(s) + b(s)n''_y(s)\right]\dot{s}^2
\end{aligned} \tag{120}$$

The only quantity yet to define is the yaw, ψ, which would be computed as

$$\begin{aligned}
\psi(s) &= \tan^{-1}\left(\frac{y_0(s) + b(s)n_y(s)}{x_0(s) + b(s)n_x(s)}\right) \approx \psi_0(s) \\
\psi'(s) &= \psi'_0(s) \\
\psi''(s) &= \psi''_0(s)
\end{aligned} \tag{121}$$

Basically, this approximation is a linearization, which remains valid when the track width, b_{max} is small. Such approximation is quite useful, since it avoids highly non-linear terms,

potentially harm for implementation. At this point, one is capable of rewriting equations 36, 39 and 40, which are represented in equation 157 in Appendix A. To facilitate readibility, such equation can be regarded as

$$\begin{aligned} \mathcal{M}\ddot{q}(s) &= \mathbf{m_1}(s, b(s), b'(s))\ddot{s} + \mathbf{m_2}(s, b(s), b'(s), b''(s))\dot{s}^2 \\ \mathcal{C}(\dot{q}) &= \mathbf{c}(s, b(s), b') \\ \mathcal{F}(\dot{q}) &= \mathbf{f}(s, b(s), b') \end{aligned} \tag{122}$$

To conclude this section, the modified vehicle governing dynamics can be condensed in

$$\mathbf{R}(s)\left[\mathbf{m_1}(s, b(s), b'(s))\ddot{s} + \mathbf{m_2}(s, b(s), b'(s), b''(s))\dot{s}^2 + \mathbf{f}(s, b(s), b'(s))\right] + \mathbf{c}(s, b(s), b'(s)) = \mathbf{u}(s) \tag{123}$$

where $\mathbf{R}(s)$ is defined as $\begin{bmatrix} \cos(\psi_0(s)) & -\sin(\psi_0(s)) & 0 \\ \sin(\psi_0(s)) & \cos(\psi_0(s)) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and represents the rotation from the fixed reference axis coordinates to body coordinates. The component $\mathbf{c}(s, b(s), b'(s))$ is excluded from this operation, since it is already defined in body coordinates.

4.2.2 Problem Formulation

Resorting equation 123 and defining $c(s) = b'(s)$ and $d(s) = b''(s)$, one can formulated the minimum time trajectory problem, following the previously adopted notation, as an optimal control problem:

$$\min_{a(\cdot),f(\cdot),b(\cdot),c(\cdot),d(\cdot),\mathbf{u}(\cdot)} \int_0^1 \frac{1}{\sqrt{f(s)}} + \frac{\gamma_1}{\sqrt{f(s)}} \left(\sum_{i=1}^{n} \frac{u_i^2(s)}{\bar{u}_i^2} \right) + \gamma_2 \left(\sum_{i=1}^{n} \frac{|u_i'(s)|}{|\bar{u}_i|} \right) ds \tag{124}$$

$$\text{subject to } \mathbf{u}(s) = \mathbf{R}(s) \left[\mathbf{m_1}(s, b(s), b'(s)) a(s) + \mathbf{m_2}(s, b(s), b'(s), b''(s)) f(s) + \mathbf{f}(s, b(s), b'(s)) \right] + \tag{125}$$

$$+ \mathbf{c}(s, b(s), b'(s))$$

$$f(0) = \dot{s}_0^2 \tag{126}$$

$$f(1) = \dot{s}_T^2 \tag{127}$$

$$b(0) = b_0 \tag{128}$$

$$b(1) = b_T \tag{129}$$

$$f'(s) = 2a(s) \tag{130}$$

$$b'(s) = c(s) \tag{131}$$

$$c'(s) = d(s) \tag{132}$$

$$f(s) \geq 0 \tag{133}$$

$$-b_{max} \leq b(s) \leq b_{max} \tag{134}$$

$$\underline{\alpha}(s) \leq g(\dot{q}(s), \delta(s)) \leq \bar{\alpha}(s) \tag{135}$$

$$\underline{\mathbf{u}}(s(t)) \leq \mathbf{u}(t) \leq \bar{\mathbf{u}}(s(t)) \tag{136}$$

$$\text{for } s \in [0, 1] \tag{137}$$

in which b_0 and b_T represent, respectively, the vehicle starting and finishing transversal deviation from the reference trajectory. As before, $\dot{s}_0^2$ and $\dot{s}_T^2$ represent the initial and final velocities of the vehicle, respectively.

The problem defined by equations 124-137 as many similarities with problem 68-76, and can be, as well, regarded as an optimal control problem in differential algebraic form (DAE), with pseudo time s, three differential states b, c and f, an algebraic state variable $\mathbf{u}$ and two input variables a and d. The definition of a convex has already been explored in this work. Nonetheless, it is worth recalling that the constraints must be linear, which is not verified in equation 124, which is deduced from equation 157 in Appendix A. Hence the constraints are nonlinear, thus raising the problem complexity. The complexity is enhanced, since the direction of search is not obtained directly, as in a linear function.

A short description of how nonlinear constraints issues are solved shall be given. If one considers a general non-convex problem

$$\begin{aligned} &\min_{t \in \mathbb{R}^{\mathbf{n_t}}} L(t) \\ &\text{subject to } G(t) = 0, t \in \Omega \end{aligned} \tag{138}$$

with $L : \mathbb{R}^{\mathbf{n_t}} \to \mathbb{R}$ convex, $\Omega \subset \mathbb{R}^{\mathbf{n_t}}$ nonempty closed convex and $G : \mathbb{R}^{\mathbf{n_t}} \to \mathbb{R}^m$ nonlinear and smooth.

The most common way to overcome nonlinear constraints is to linearize the constraints about a given operation point, t^0, equation 139, and then iterate.

$$t^{p+1} = t^p + \varepsilon_p \Delta t^p \tag{139}$$

with $\varepsilon_p > 0$, which is usually computed using the line search method. This approach is the foundation of Sequential Convex class of problems. Equation 139 computes the search direction Δp^i by solving a convex subproblem, obtained by means of a linearization of the nonlinear constraints $G(t) = 0$. This step of linearizing, through equation 139, preserves the objective function and constraints convexity. The convex subproblem to solve is

$$\begin{aligned} &\min_{\Delta t} L(t^p + \Delta t) \\ &\text{subject to } J_G(t^p) G((t^p + \Delta t) = 0, t^p + \Delta t \in \Omega \end{aligned} \tag{140}$$

in which $J_G(t^p) = \left(\frac{\partial G(t^p)_i}{\partial t_j} \right)_{i,j}$ is the Jacobian of the nonlinear constraints function at t^p.

The nonlinearity of the constraints forbids the use of SOCP methods in its natural state. In [27], SOCP was used in a very similar problem, but the exact formulation afore mentioned was used to linearize the problem at every single iteration. In [35], a Newton operator method was used, which is inherently a descent method, and an exploration strategy was added to diverge the solution from local minimums. In [26] the problem was solved using the Pontryagin's method and was converted into an unconstrained optimization solver. In [25], multi-objective optmization was used to compute minimum lap times for a Formula 1 car with variable parameters, employing direct transcription and nonlinear programming algorithm IPOPT [53]. The approach followed in this work was to use MATLAB *fmincon* function, which applies interior-point techniques, to solve this nonlinear problem. This function uses

Sequential Quadratic Programming (SQP) or Linear Programming (LP) to solve the convex subproblem [54].

4.2.3 Numerical Solution

As in section 4.1.3, a direct transcription method is applied to the optimal control problem. The acceleration, $a(\cdot)$, is parametrized as a piecewise constant function, 12, the squared velocity, $f(\cdot)$, is consequently a piecewise constant function, 13, and the inputs, $u_i(\cdot)$ result to be piecewise nonlinear functions. Analogously, the optimal control in DAE form input $d(\cdot)$ is parametrized as a piecewise constant function, figure 16, the first spatial derivative of $b(s)$, $c(\cdot)$, is piecewise linear, figure 17, and thereupon the path transversal position $b(\cdot)$ is piecewise quadratic, figure 18. Therefore, to d is evaluated at the middle points, $d^k = d(s^{k+\frac{1}{2}})$, c is evaluated at the end points and in the middle function, its values is $c^{k+\frac{1}{2}} = \frac{c^k + c^{k+1}}{2}$ and, finally b also has its transcribed values at the end points but can be computed at middle points by $b^{k+\frac{1}{2}} = b^k + \frac{c^k \Delta s_k}{2} + \frac{d^k \Delta s_k^2}{8}$.

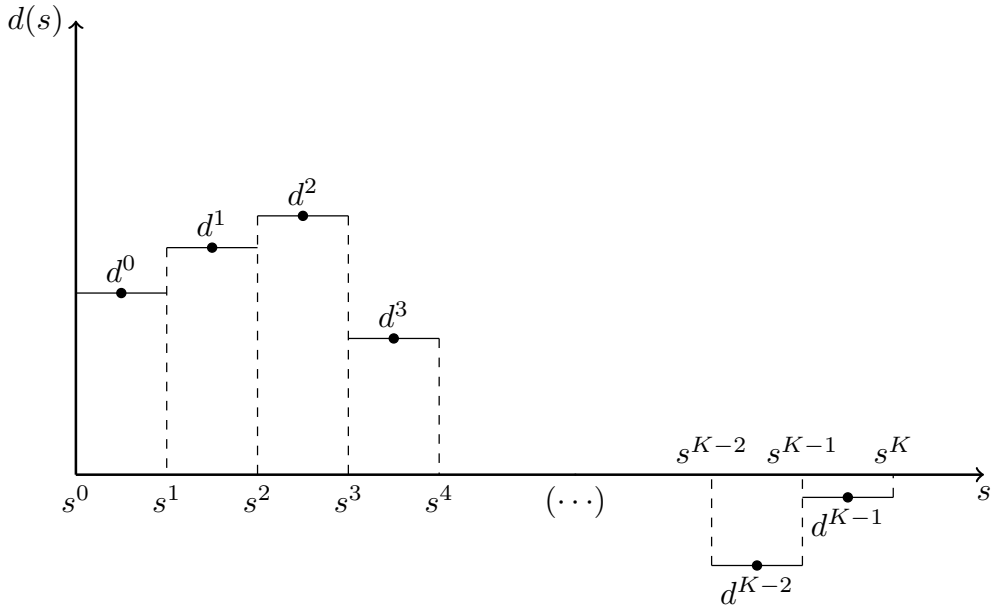

Figure 16: Second spatial derivative of $b(s)$, $d(s)$, represented as a piecewise constant function for $s \in [0, 1]$

One can easily infer that due to the discretization the relations between d, c and b are

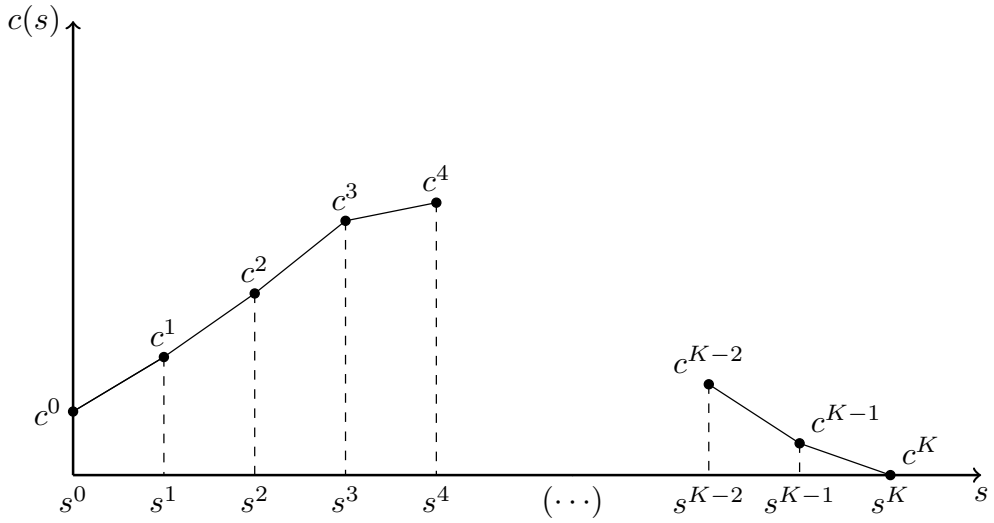

Figure 17: First spatial derivative of $b(s)$, $c(s)$, represented as a piecewise linear function for $s \in [0, 1]$

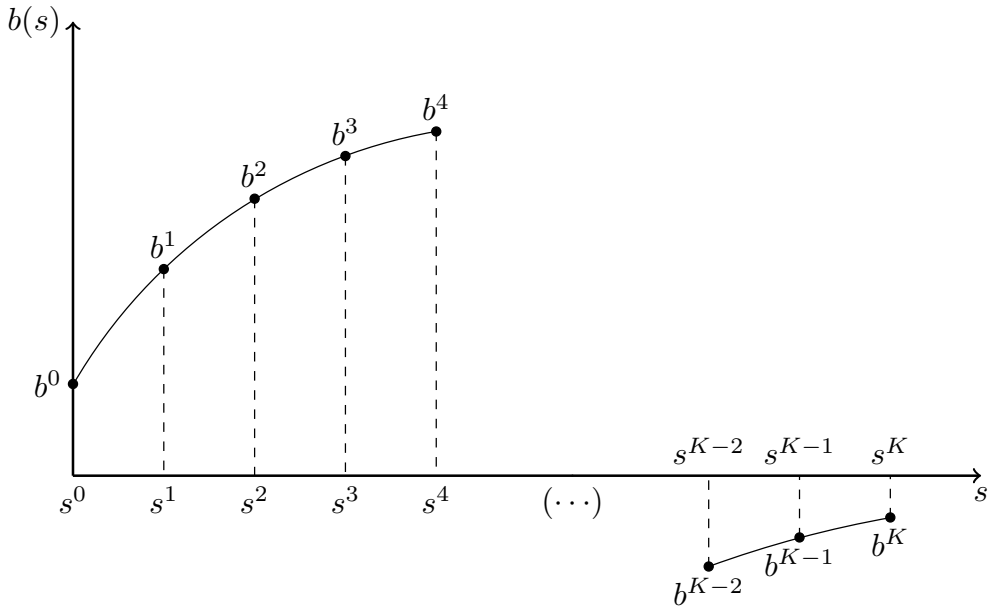

Figure 18: Car transversal position in the track, $b(s)$, represented as piecewise quadratic functions for $s \in [0, 1]$

$$\begin{aligned} b^{k+1} - b^k &= \Delta s_k c^k + \frac{1}{2}\Delta s_k^2 d^k \\ c^{k+1} - c^k &= \Delta s_k d^k \end{aligned} \tag{141}$$

It is important to mention that the evaluation of $\mathbf{R}(s)$, $x_0(s)$,$y_0(s)$, $\psi_0(s)$ and equations 118

is done in middle points and its computation is performed in pre-processing, which relieves optimization computational cost. To get to the final optimization problem in discrete form, to avoid contents repetition of section 4.1, it was chosen to just show the problem in its final form. Most steps were already explained in detail and it is only required to do not difficult mathematical derivation. Hence, the optimal control problem of finding the trajectory that minimizes time, can be regarded as

$$\min_{a^k,f^k,b^k,c^k,d^k,\mathbf{u}^k} \sum_{k=0}^{K-1} \frac{2\Delta s^k \left(1+\gamma_1 \sum_{i=1}^{n} \frac{\left(u_i^k\right)^2}{\bar{u}_i^2}\right)}{\sqrt{f^k}+\sqrt{f^{k+1}}} + \gamma_2 \sum_{k=0}^{K-1} \left(\sum_{i=1}^{n} \frac{|\Delta u_i^k|}{|\bar{u}_i|}\right) \tag{142}$$

$$\text{subject to } \mathbf{u}^k = \mathbf{R}^k \left[\mathbf{m_1}(s^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}})a^k + \mathbf{m_2}(s^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}}, d^k)f^{k+\frac{1}{2}} + \mathbf{f}^k(s^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}})\right] +$$

$$+ \mathbf{c}(s^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}}) \tag{143}$$

$$f^0 = \dot{s}_0^2 \tag{144}$$

$$f^K = \dot{s}_T^2 \tag{145}$$

$$b^0 = b_0 \tag{146}$$

$$b^K = b_T \tag{147}$$

$$(f^{k+1} - f^k) = 2a^k\Delta^k \tag{148}$$

$$b^{k+1} - b^k = \Delta s_k c^k + \frac{1}{2}\Delta s_k^2 d^k \tag{149}$$

$$c^{k+1} - c^k = \Delta s_k d^k \tag{150}$$

$$f^k \geq 0 \tag{151}$$

$$-b_{max} \leq b^k \leq b_{max} \tag{152}$$

$$\underline{\alpha}^k \leq g(q_0^k, \mathbf{n}^k, \mathbf{n}'^k, b^{k+\frac{1}{2}}, c^{k+\frac{1}{2}}, \delta^k) \leq \bar{\alpha}^k \tag{153}$$

$$\underline{\mathbf{u}}^k \leq \mathbf{u}^k \leq \bar{\mathbf{u}}^k \tag{154}$$

$$\text{for } s \in [0,1] \tag{155}$$

The nonlinear optimization problem of 142-155 , formulated in this form, appears to be simple and well formulated. The issue is that the problem is nonlinear and non-convex, due to its nonlinear constraints, and, therefore, much more difficult to solve. An important statement is that the nonlinearity comes from the variables b^k, c^k and d^k and it is easy to

prove that if c^k and d^k are zero, than the problem falls in the convex problem of 82-90. Thus, if one considers b^k, c^k and d^k to be small, which, actually, is reasonable assumption on a race track, since the racing line is supposed to be smooth and b_max is usually small, the constraints 143 may be considered slightly near linear, enabling the use, without much error, of previously mentioned SCP and SQP methods to linearize the constraints.

5 Results and Discussion

In this section the results of the afore mentioned methods are presented. First one shall present the results that arise from the algorithm of section 3. These are followed by section 4.1, which correspond to the optimal control minimum time path tracking algorithm and, at last the results of the optimal control minimum time trajectory optimization of section 4.2 are presented. The results were obtained using the car parameters of table 1.

Table 1: Car parameters

Mass, m [Kg]	0.68
Front Cornering Stiffness, C_{α_f} [N/rad]	8
Rear Cornering Stiffness, C_{α_r} [N/rad]	8
Front axis distance to CoG, a_1 [m]	0.08025
Rear axis distance to CoG, a_2 [m]	0.084975
Inertia about z-axis, I_z [m^4]	0.01
Maximum wheel steering angle, δ_{max} [degrees]	22
Rolling resistance coefficient, K [Kgm^2]	0.25

5.1 Trajectory Optimization using Genetic Algorithm

The results obtained for the trajectory optimization resorting to genetic algorithms shall be presented in this section. In section 3, the longitudinal dynamics was represented in the form of a function $f(x, \dot{x})$. Hence, one presents in 156 the linearized equation describing the longitudinal dynamics that was used in this work

$$m\ddot{x} = 4.6875mu_p - K\dot{x}_0\dot{x} \tag{156}$$

with $\dot{x}_0 = 1$ as the linearization point. It is important to mention that all the results presented here were obtained using the same GA optimzation parameters, which are presented in table 2.

One must emphasize that the optimization parameters were not optimized. Nonetheless, the criteria was to have low amount of elites in order to allow diversed trajectories caused by the mutations and fine tuning as a consequence of the crossovers. Naturally, the percentage of crossovers is much higher to force many solutions with little changes, hence resulting in

Table 2: GA optimization parameters

Number of optimization control point	20
Population Size	200
Population elite [%]	10
Population crossover [%]	75
Population mutation [%]	15
Stopping criteria	Δt<0,001s

better, although with high chance of sub-optimal solutions. The number of control points was chosen, merely to decrease the computational effort, it is expected that with higher number of control points the solution would be better. Even though, this number was chosen to have at least 4 points in each curve, which are the necessary to define the third order polynomial chosed for the parametrization. Furthermore, the population size was chosen to be large enough to reach good trajectories, although it would produce better results if it was bigger, but once again is a compromise to obtain results in useful time.

Some results of this method can be found in figures 19, 20 and 21. The first point to mention is that the three solutions would be exactly or approximately the same it they were optimal and has been proved that if the GA runs for long enough with a population large enough such assumption is true. Secondly, one would like to mention that the three trajectories correspond to good trajectories for this track. Trajectories of figures 19 and 21 open the first turn in order to increase the radius and hence increase the velocity, however in the former the apex is done the beginning of the turn and in the latter and the end. It is difficult to say which one is better, due to the influence of the entry in the exit and vice-versa. Although, a professional driver would say that the exit should control the entry, since a good exit is key because one starts accelerating first. It is also interesting that in both trajectories, if the first turn apex is done at the entry, the apex of the second turn would occur at the exit of the second turn and the opposite also occurs. The exception is the trajectory of figure 20 that has two apex in turn 1 ans does the turn 2 in a similar fashion that the trajectory of figure 21. The time difference involved between each solution is quite small, about $0.03s$ which corresponds to approximately 0.5% of the total time.

Another important point to mention is that the use of an initial population containing the 200 best paths resulting from the minimum curvature optimization greatly decreased

the simulation time and the final solution time. A representation of the pedal input, u_p, of the trajectory of figure 19 is in figure 23 and its corresponding longitudinal velocity is in figure 22. The bang-bang behaviour of the velocity is obvious with an exception at about $s = 5m$. Finally, sub-optimal solutions would occur due to the path discretization, B-splines interpolation and size of the initial population.

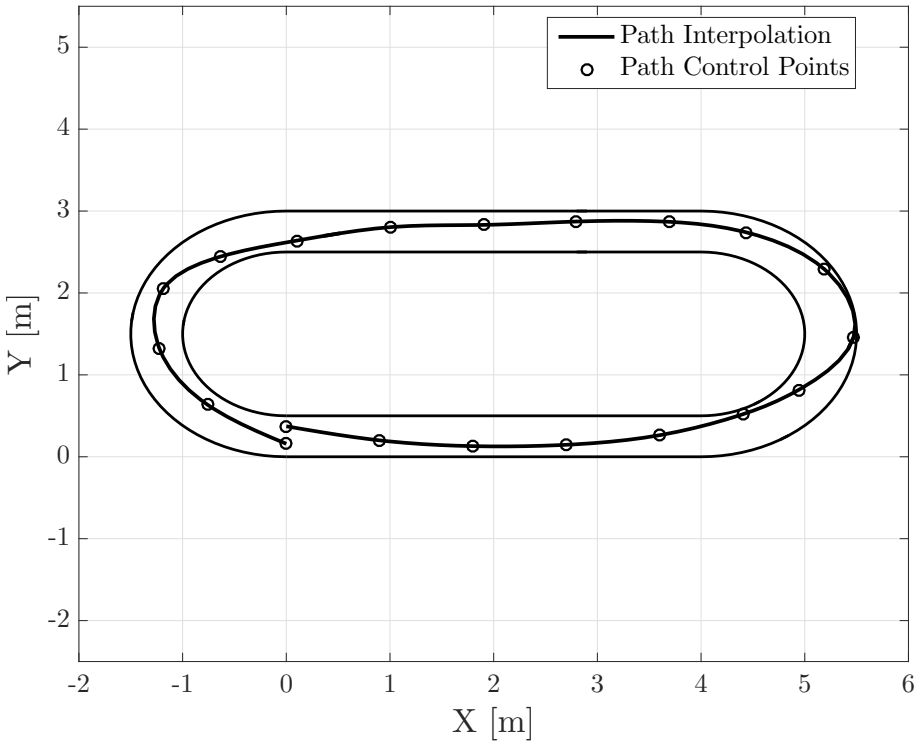

Figure 19: Result of the minimum time optimization, obtaining $t_f = 5.1741s$

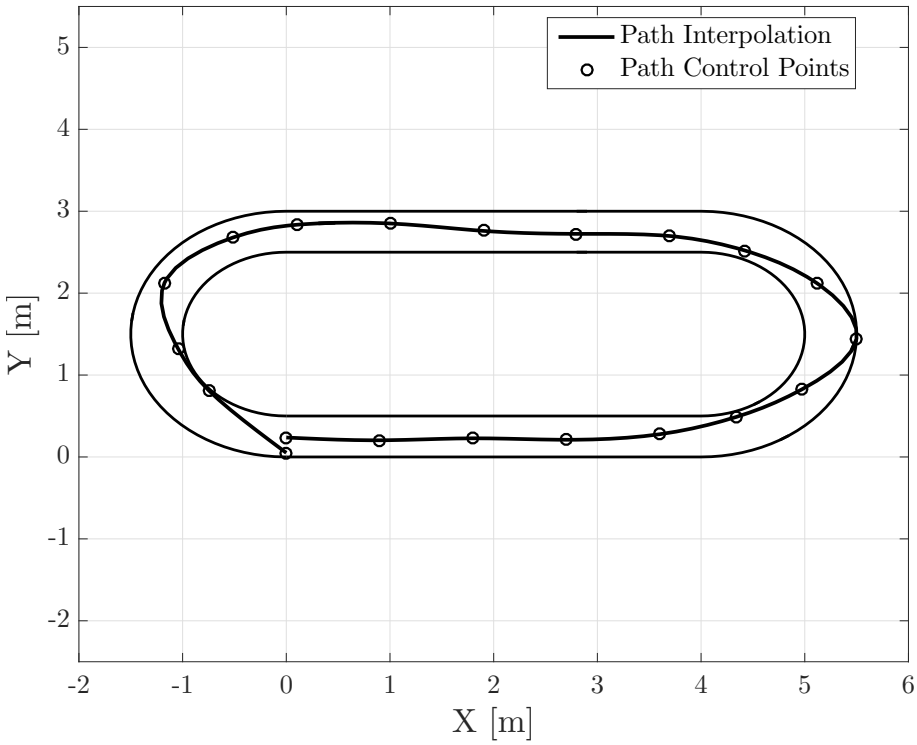

Figure 20: Result of the minimum time optimization, obtaining $t_f = 5.2075s$

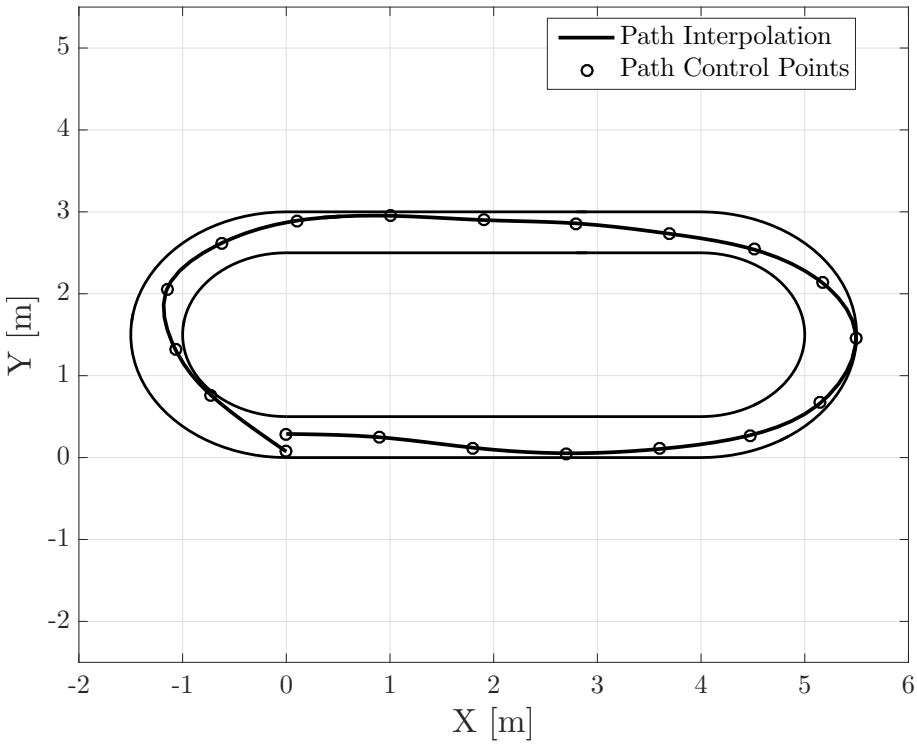

Figure 21: Result of the minimum time optimization, obtaining $t_f = 5.233s$

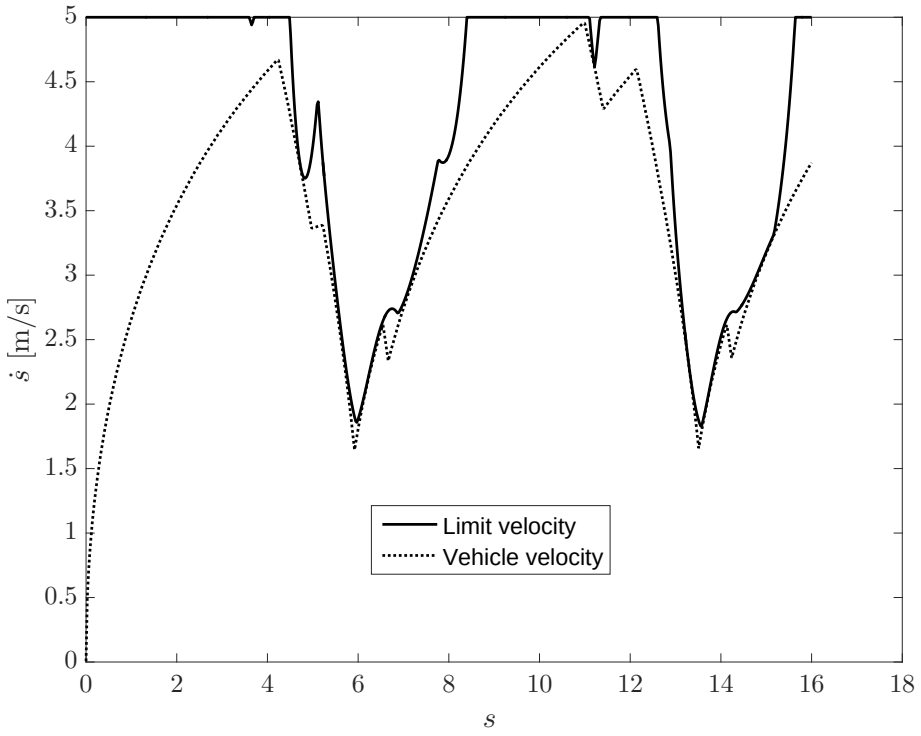

Figure 22: Longitudinal velocity corresponding to the trajectory of $t_f = 5.1741s$

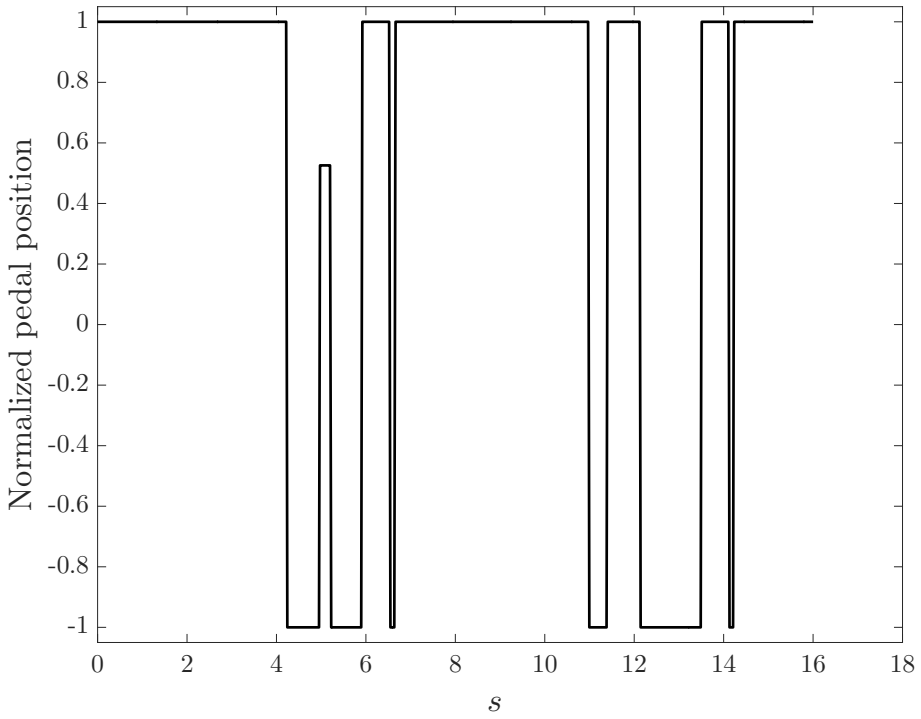

Figure 23: Pedal Position corresponding to the trajectory of $t_f = 5.1741s$

5.2 Optimal Control Minimum time path tracking

The optimal control minimum time formulation over a given path was modelled to not only include the lateral dynamics in the computation of the optimal $\dot{s}$, but also to overcome the bang-bang nature of the section 3.4.2 algorithm. Moreover, the problem was formulated in a way that one guarantees, due to the objective function convexity, that the solution is the global minimum. Therefore, in this section the results shall be presented as a comparison between the two algorithms. For the track used in section 5.1, the comparison of both methods can be inspected in figure 24. The inputs associated with this solution are represented in figure 25. For this result, the track was equally divided in 200 points and $\gamma_2 = 1\text{e}^{-2}$.

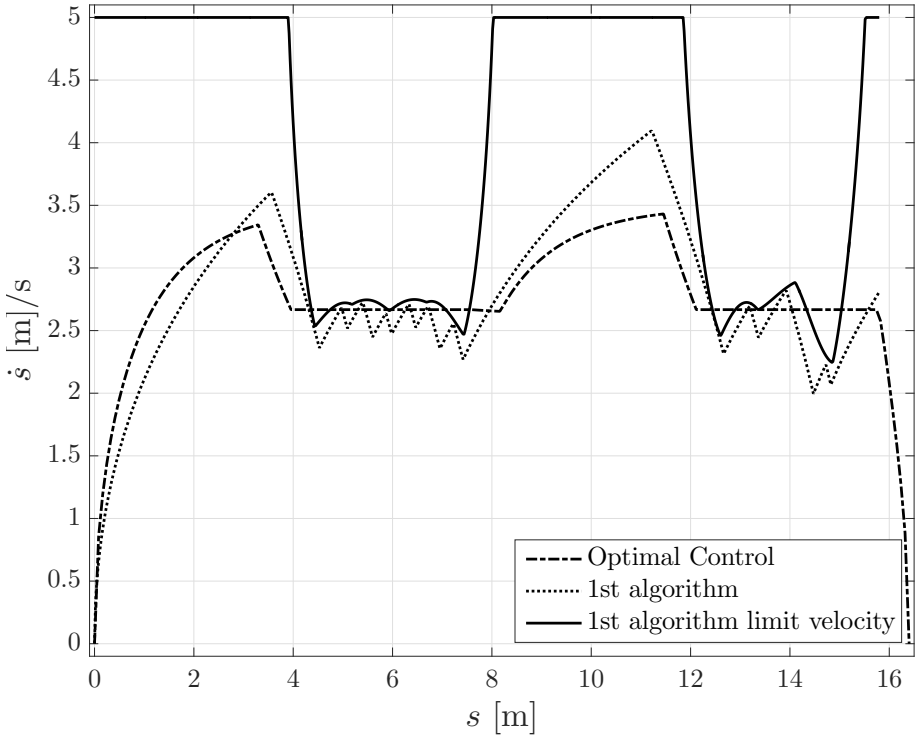

Figure 24: Comparison of the optimal control longitudinal velocity profile, dashed-dotted line, with a solution of section 3.4.2, dotted line and its corresponding limit velocity, solid line.

Regarding figure 24 it is clear that the optimal control method delivered a better result during the cornering, $s \in [4, 8]$. This is the result of accounting with the lateral dynamics within the problem formulation. The difference in the transient behaviour of both functions arise due to the nonlinear effect caused by the longitudinal vehicle resistance, which in this formulation is proportional do $\dot{x}^2$ instead of being linearized as in equation 156. The fact that $\dot{s}$, in the optimal control case, finishes in 0 is the consequence of a constraint inherent to the

formulation, which is the prescription of the final state. The lap times of both methods will not be compared, since it would not be fair. Nonetheless, the computational effort of both methods is low, with the optimal control tooking about 3.5 seconds and the method of section 3.4.2 about 1 second. Despite the latter's code is not optimized, one must recognized that the former represents a more complete problem formulated as convex, which was adapted to be efficiently used with SOCP solvers.

To demonstrate the impact of adding additional penalties to the objective function, in this case γ_2, in figure 26 one finds the resulting inputs using $\gamma_2 = 0$. In some applications, the jitter may wear and damage components and also deviate the robot from the desired path.

It is worth mentioning that the steering input in figures and 25 26 assumes the maximum value of 1, due to the absence of more mature tyre model. Tires were modeled as linear, when in fact they are not, thus with the increase of the steering angle the slip angle will also increase, producing higher lateral force. This is not representative of the reality, once the saturation behaviour is ignored, thus overestimating the tyre lateral force capability.

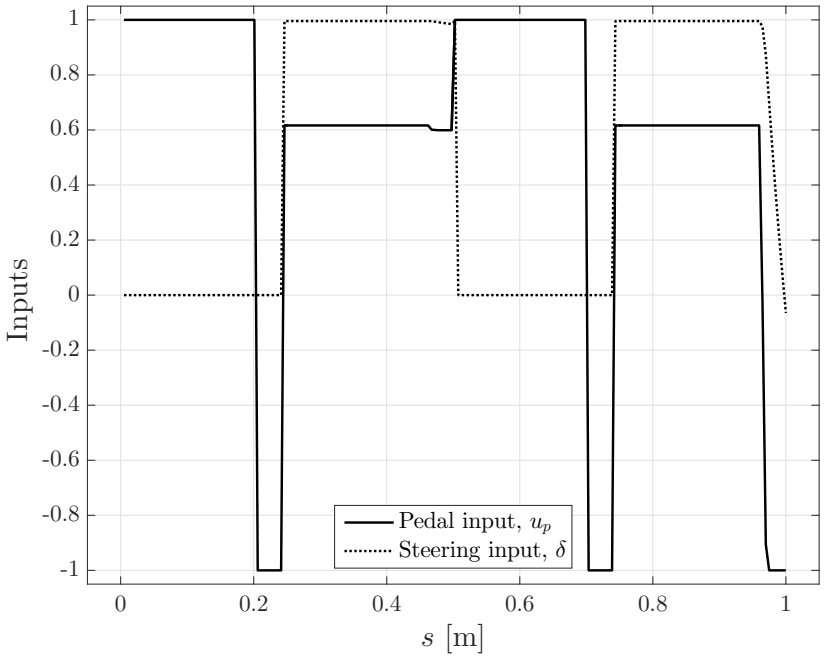

Figure 25: Pedal input and steering inputs, u_p and δ, respectively, for the solution of figure 24

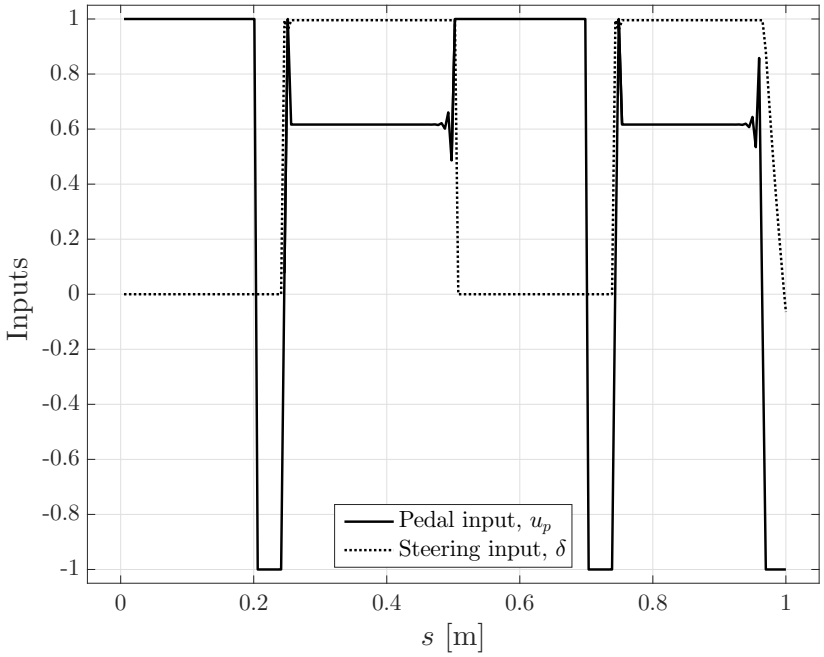

Figure 26: Pedal input and steering inputs, u_p and δ, respectively, using $gamma_2 = 0$

5.3 Optimal Control Trajectory Optimization

In this section the results for the method demonstrated in section 4.2 are presented. To make use of the more complete formulation and to surpass the naïve tyre model, a slip angle constraint was imposed. Consequently, one is forcing the slip angle absolute value, at rear and front wheels, to be lower or equal than 5 degrees, $|\alpha| \leq 5$. One would like to refer that although this constraint resembles the real tyre behaviour it does not replace a more mature model. Once again, the simulation was taken for the same track as the aforementioned methods. The trajectory obtained, using the path divided in 75 equally spaced points, can be found in figure 27, the longitudinal velocity, $\dot{s}$, in figure 28 and the pedal and steering inputs in figure 29.

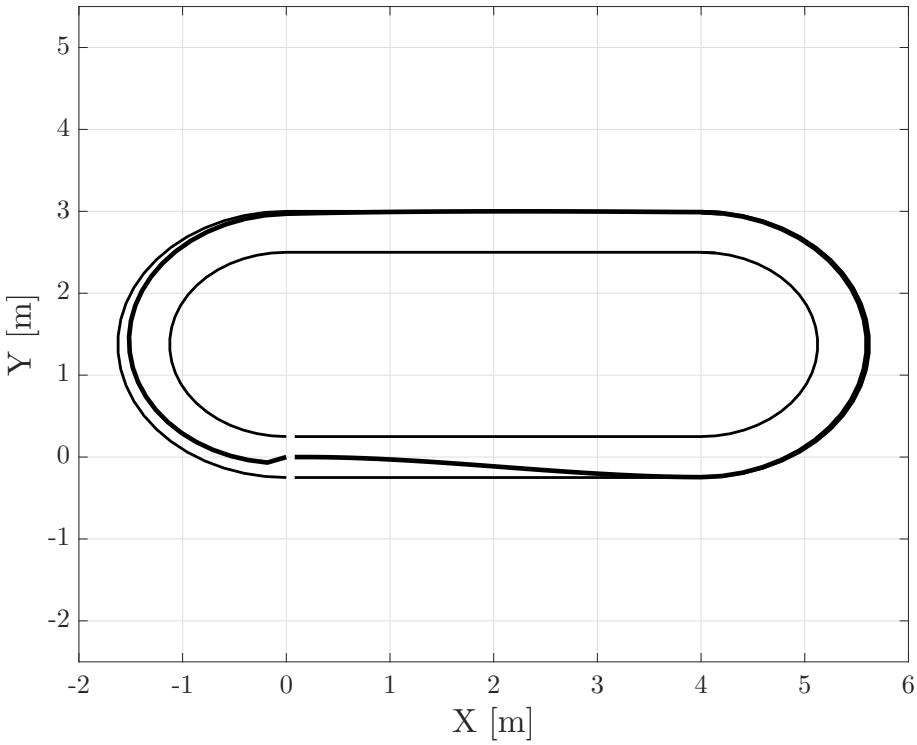

Figure 27: Optimal control minimum time trajectory

The differences between the trajectory of figure 28 and 19-21 are admirable. Nonetheless, it is not a consequence of a poorer minimum time optimal control formulation. In fact, it is a consequence of the non-convexity of the problem. Since the constraints are non-linear, the objective functions may have several minimum and depending on the initial point the function may stop at different local minimum. Despite the use of the Sequential Quadratic Programming solver of the MATLAB function *fmincon*, which linearizes the problem at each

iteration to overcome the nonlinearity, this issue has not been surpassed. The hypothesis here introduced is inline with the formulation in [27], where a constraint linearization is performed, at each iteration, using SeDuMi, thus formulating a Sequential Convex Programming algorithm. Furthermore, in [35] a Newton Projector Operator was implemented to repel the solutions from local minimum, enhancing, therefore, the probability of reaching the global minimum. Simulations were performed without the slip angle constraint and the vehicle tends to get closer to the inner track limit, or even to the center line, which apparently seems to be a local minimum solution. This behaviour may be explained by the current tyre model, which produces lateral force proportional to the slip angle, which increases linearly with the steering angle. Moreover, the current tyre model assumes small slip angles, condition that is often violated.

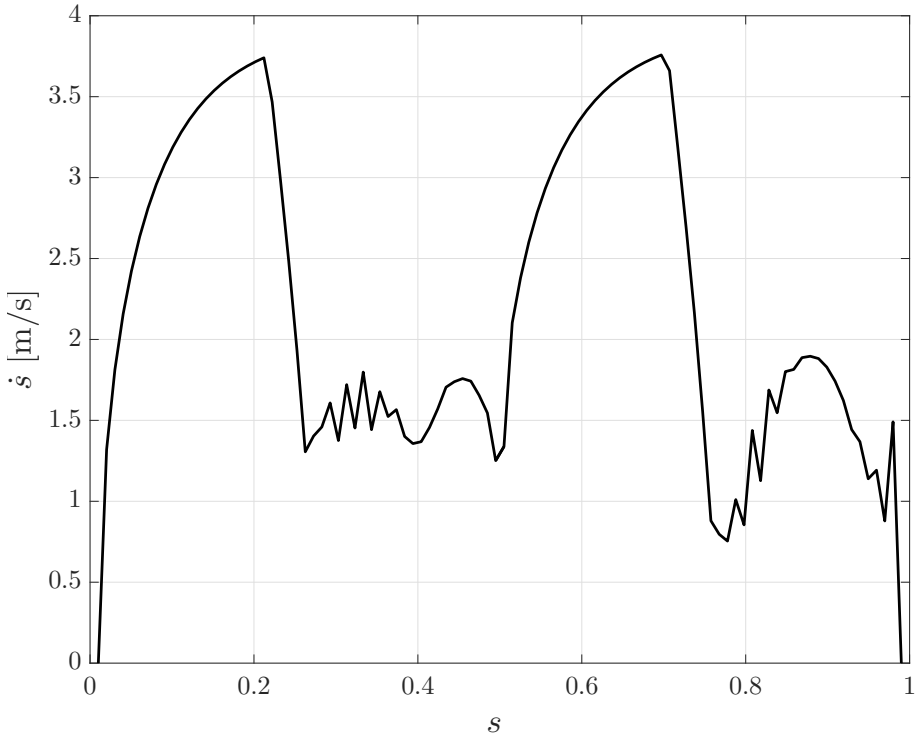

Figure 28: Optimal control minimum time trajectory longitudinal velocity in m/s

The results of figures 28 and 29 show severe numerical instability. To further explore this issue, one more simulation has been performed using 75 points recurring to Interior-Point *fmincon* solver, figures 30-32. Furthermore, the maximum steering angle was reduced to 11 degrees, in order to force the slip angle to deviate from greater values. The results reveal much less jitter in the inputs, which outcomes a much smoother and more realistic velocity

profile. This results, reveal that the tyre model not only has a high impact on the simulated car behaviour but also on numerical stability.

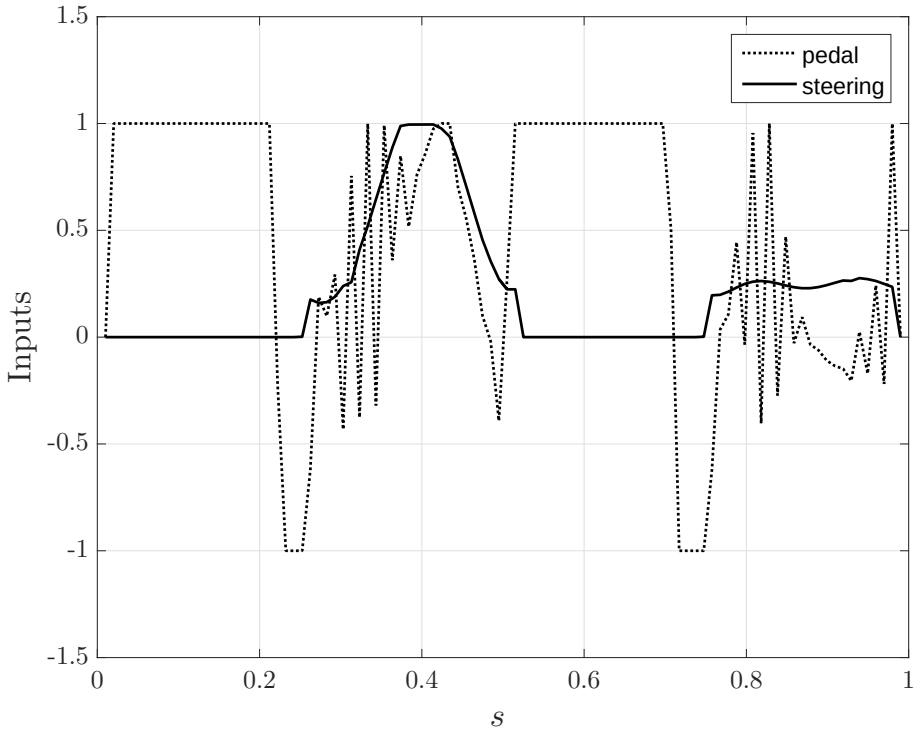

Figure 29: Optimal control minimum time trajectory longitudinal normalized inputs

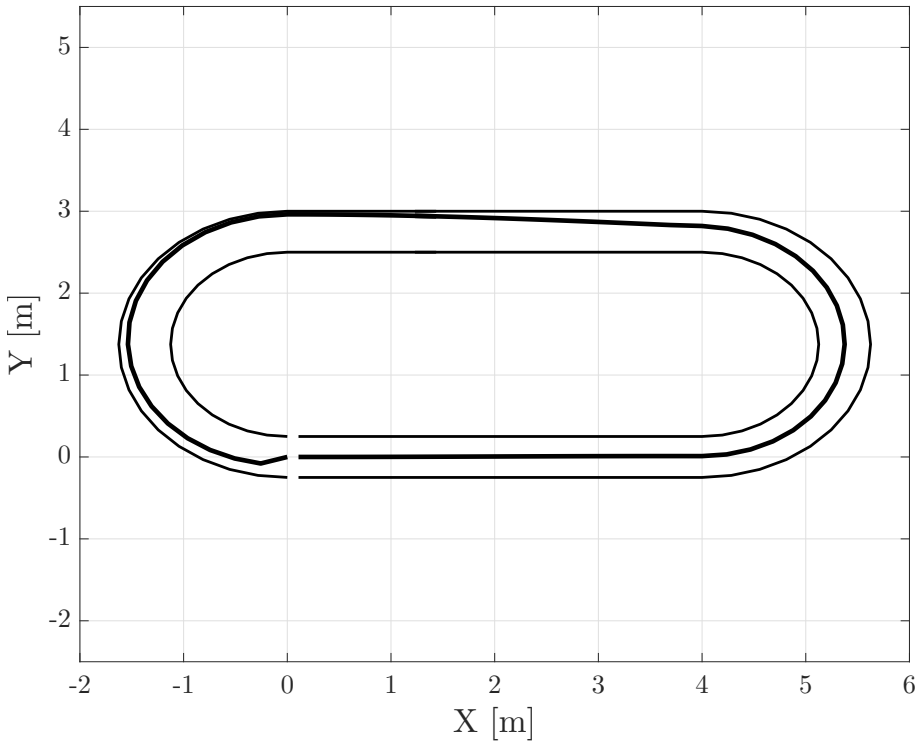

Figure 30: Optimal control minimum time trajectory using Interior-Point solver

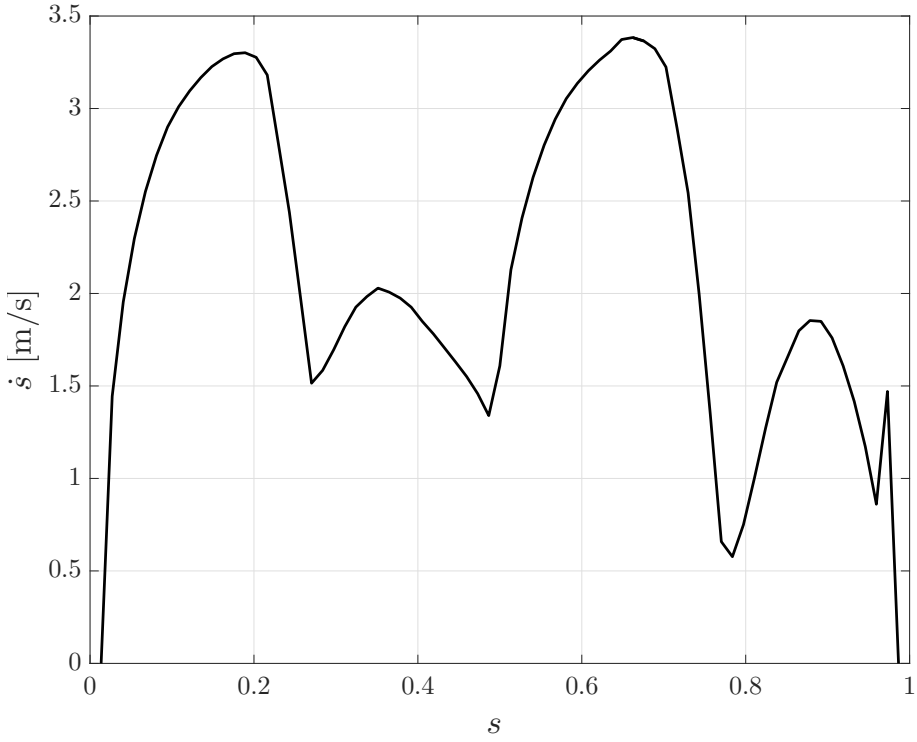

Figure 31: Optimal control minimum time trajectory longitudinal velocity in m/s using Interior-Point solver

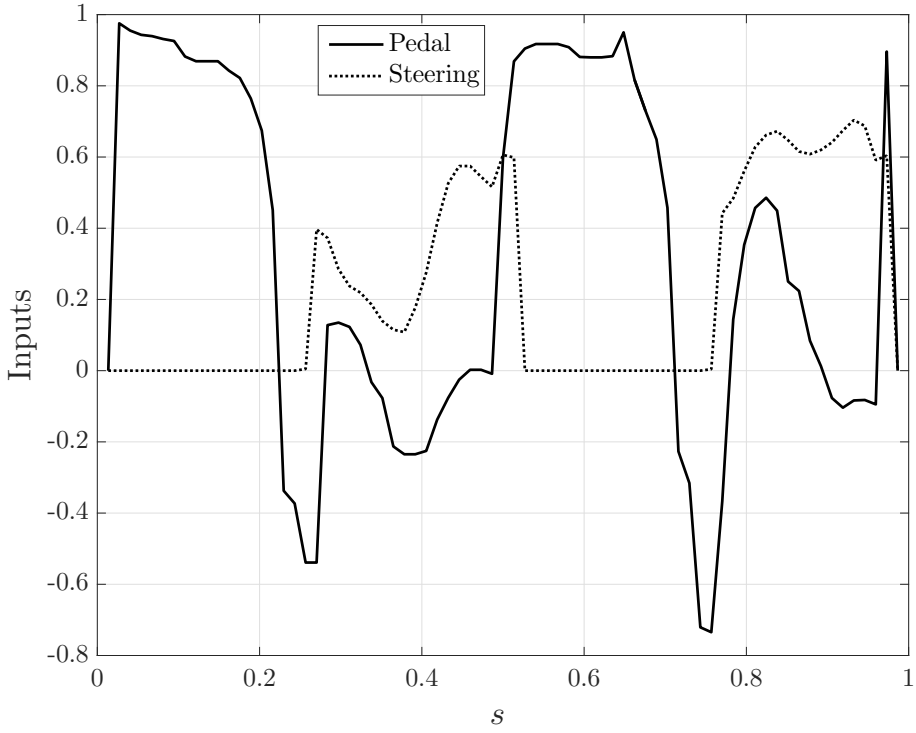

Figure 32: Optimal control minimum time trajectory longitudinal normalized inputs using Interior-Point solver

6 Conclusions

This work presents two different approaches, a stochastic and a deterministic, to solve the minimum time trajectory optimization and one robust and mature optimal control formulation to solve the minimum time path tracking problem.

The first part of the work presented starts with a simple explanation of tyre dynamics, particularly on the lateral and longitudinal force generation mechanism, culminating with a simple and naïve tyre model. Tyre dynamics leads the understanding of vehicle dynamics, which is the next topic introduced. A two track planar model was deduced from rigid body and tyre dynamics and was than linearized to obtain the bicycle model used thereupon.

The minimum time trajectory optimization problem was first approached resorting to Genetic Algorithms. The first step of this method was to parametrize the track. Next, b-splines path interpolation was suggested to achieve first and second derivatives continuity, which are fundamental to obtain smooth paths with continuous curvature. This algorithm resorts to a change of variables from time to space to facilitate the problem formulation. It results from the decoupling of the minimum time problem in three subproblems, which are the determination of the longitudinal velocity limits over a fixed path, computed through vehicle dynamics equations.The second stage is the velocity profile optimization for the prescribed path, culminating in an optimization that controls the position of n path control points that when interpolated originate a path. To reduce the computational effort of the algorithm another optimization was implemented in pre-processing with the goal of minimizing the curvature. The resultant $k - best$ paths are the initial population of the genetic algorithm used to solve the problem.

The following method was the optimal control minimum time path tracking, which delivers the optimal inputs signals over a fixed path. This formulation overcomes the lack of dynamics in the former method. The vehicle dynamics were reformulated in order to obtain more obvious generalized coordinates and than, once again, resorting to chain rule, the time dependency of the dynamic variables was replaced by path dependency, namely on the variable s. The problem was formulated as convex, than discretized using direct transcription methods. Finally, it was adapted, using convex preserving operators, to be formulated as SOCP problem, hence enabling the use of more mature solvers like SeDuMi or CVX.

At last, an optimal control minimum time trajectory optimization was proposed. The

formulation here addressed is a development of the previous method to account with the transversal degree freedom on the track, enabling a proper trajectory optimization formulation. The track was first parametrized, followed by the dynamics reformulation, ending in the a discrete optimization problem, recurring to direct transcription methods. The method here presented is non-convex, therefore representing a substantial difficulty increase to achieve the global minimum.

The results obtained with the first method revealed a good racing line, although not periodic. Periodic and more consistent results would be obtained using a greater population and a more conservative stopping criteria. Regarding the optimal control minimum time path track algorithm, one can say that it was successfully implemented outcoming good results with computational efficiency. The optimal control minimum time trajectory results, despite the elegant problem formulation, could not produce outstanding results. One could not surpass the non-convexity issue, which would require an alteration of the available solvers and is beyond the objectives of this thesis.

7 Future Work

On the light of the work presented, the future work would be a more mature tyre modelling, such as Pacejka Magic Formula. This would certainly be a relevant improvement when compared with the present model, event though one lacks tyre experimental data. An important component that requires a thorough study is the solver to be used. The fact that the minimum time optimal control trajectory optimization problem is a non-convex problem demands a modification on the solvers available at the time. One solution would be to adapt SeDuMi or CVX code in order to linearize the nonlinear constraints at every function evaluation. The adaptation of MATLAB *fmincon* function seems unlikely, since one do not have access to the function code. An interesting experiment would be to test and compare the computational results with experimental data obtained, for instance, with model scale cars. SeDuMi computational efficiency, though with some modifications, enables the use of the methods here presented for Optimal Control and Model Predictive Control in real time, for example for trajectory or path tracking optimization in an environment with moving obstacles.

Appendices

A Reformulated Dynamics for Trajectory Optimization

$$
\begin{aligned}
\mathcal{M}\ddot{q}(s) &= \mathcal{M}\left(\left[q_0'(s) + b'(s)\mathbf{n} + b(s)\mathbf{n}'\right]\ddot{s} + \left[q_0''(s) + b''(s)\mathbf{n}(s) + 2b'(s)\mathbf{n}(s) + b(s)\mathbf{n}''(s)\right]\dot{s}^2\right) \\
\mathcal{C}(\dot{q}) &= \begin{bmatrix} -m\left[y_0'(s) + b'(s)n_y(s) + b(s)n_y'(s)\right]\psi_0'(s) - K[x_0'^2(s) + 2x_0'(s)b'(s)\sin(\psi_0(s)) + 2x_0'(s)b'(s)\cos(\psi_0(s))] \\ m\left[y_0'(s) + b'(s)n_y(s) + b(s)n_y'(s)\right]\psi_0'(s) \\ 0 \end{bmatrix}\dot{s}^2 \\
\mathcal{F}(\dot{q}) &= \begin{bmatrix} 0 \\ C_{\alpha_f}\left(\frac{y_0'(s)+b'(s)n_y(s)+b(s)n_y'(s)}{x_0'(s)+b'(s)n_x(s)+b(s)n_x'(s)} + a_1\frac{\psi_0'(s)}{x_0'(s)+b'(s)n_x(s)+b(s)n_x'(s)}\right) + C_{\alpha_r}\left(\frac{y_0'(s)+b'(s)n_y(s)+b(s)n_y'(s)}{x_0'(s)+b'(s)n_x(s)+b(s)n_x'(s)} - a_2\frac{\psi_0'(s)}{x_0'(s)+b'(s)n_x(s)+b(s)n_x'(s)}\right) \\ a_1C_{\alpha_f}\left(\frac{y_0'(s)+b'(s)n_y(s)+b(s)n_y'(s)}{x_0'(s)+b'(s)n_x(s)+b(s)n_x'(s)} + a_1\frac{\psi_0'(s)}{x_0'(s)+b'(s)n_x(s)+b(s)n_x'(s)}\right) - a_2C_{\alpha_r}\left(\frac{y_0'(s)+b'(s)n_y(s)+b(s)n_y'(s)}{x_0'(s)+b'(s)n_x(s)+b(s)n_x'(s)} - a_2\frac{\psi_0'(s)}{x_0'(s)+b'(s)n_x(s)+b(s)n_x'(s)}\right) \end{bmatrix} \\
\mathbf{u} &= \begin{bmatrix} 4.6875m & 0 \\ 0 & C_{\alpha_f} \\ 0 & a_1C_{\alpha_f} \end{bmatrix}\begin{bmatrix} u_p \\ \delta \end{bmatrix}
\end{aligned}
\tag{157}
$$

References

[1] R. Jazar, *Vehicle dynamics: theory and application.* Springer, 2008.

[2] S. M. LaValle, *Planning algorithms.* Cambridge university press, 2006.

[3] J. Funda, R. H. Taylor, B. Eldridge, S. Gomory, and K. G. Gruben, "Constrained cartesian motion control for teleoperated surgical robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 453–465, 1996.

[4] P. Kazanzides, J. Zuhars, B. Mittelstadt, and R. H. Taylor, "Force sensing and control for a surgical robot," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on.* IEEE, 1992, pp. 612–617.

[5] G.-Q. Wei, K. Arbter, and G. Hirzinger, "Real-time visual servoing for laparoscopic surgery. controlling robot motion with color image segmentation," *IEEE Engineering in Medicine and Biology Magazine*, vol. 16, no. 1, pp. 40–45, 1997.

[6] J.-C. Latombe, *Robot motion planning.* Springer Science & Business Media, 2012, vol. 124.

[7] C. Chen, M. Rickert, and A. Knoll, "Kinodynamic motion planning with space-time exploration guided heuristic search for car-like robots in dynamic environments," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on.* IEEE, 2015, pp. 2666–2671.

[8] T. Gu and J. M. Dolan, "On-road motion planning for autonomous vehicles," in *International Conference on Intelligent Robotics and Applications.* Springer, 2012, pp. 588–597.

[9] T. Gu, J. Snider, J. M. Dolan, and J.-w. Lee, "Focused trajectory planning for autonomous on-road driving," in *Intelligent Vehicles Symposium (IV), 2013 IEEE.* IEEE, 2013, pp. 547–552.

[10] J. Larson, M. Bruch, R. Halterman, J. Rogers, and R. Webster, "Advances in autonomous obstacle avoidance for unmanned surface vehicles," DTIC Document, Tech. Rep., 2007.

[11] S. Teller, M. R. Walter, M. Antone, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, J. Glass, J. P. How, A. S. Huang *et al.*, "A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on.* IEEE, 2010, pp. 526–533.

[12] J. E. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The international journal of robotics research*, vol. 4, no. 3, pp. 3–17, 1985.

[13] Z. Shiller and Y.-R. Gwo, "Dynamic motion planning of autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 241–249, 1991.

[14] Z. Shiller, "Time-energy optimal control of articulated systems with geometric path constraints," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on.* IEEE, 1994, pp. 2680–2685.

[15] K. Shin and N. McKay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Transactions on Automatic Control*, vol. 31, no. 6, pp. 491–500, 1986.

[16] Z. Shiller, "Off-line and on-line trajectory planning," in *Motion and Operation Planning of Robotic Systems.* Springer, 2015, pp. 29–62.

[17] J.-J. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 118–124, 1989.

[18] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE Journal on Robotics and Automation*, vol. 3, no. 2, pp. 115–123, 1987.

[19] J. T. Betts and W. P. Huffman, "Path-constrained trajectory optimization using sparse sequential quadratic programming," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 1, pp. 59–68, 1993.

[20] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[21] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[22] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.

[23] D. Costantinescu and E. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *Journal of robotic systems*, vol. 17, no. 5, pp. 233–249, 2000.

[24] T. Ardeshiri, M. Norrlöf, J. Löfberg, and A. Hansson, "Convex optimization approach for time-optimal path tracking of robots with speed dependent constraints," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 14 648–14 653, 2011.

[25] 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.

[26] R. Lot and F. Biral, "A curvilinear abscissa approach for the lap time optimization of racing vehicles," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 7559–7565, 2014.

[27] D. Q. Tran and M. Diehl, "An application of sequential convex programming to time optimal trajectory planning for a car motion," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 4366–4371.

[28] D. Casanova, "On minimum time vehicle manoeuvring: The theoretical optimal lap," 2000.

[29] L. Marques, "Optimal lap time for a race vehicle," *Instituto Superior Técnico, Universidade de Lisboa*, 2016.

[30] J. A. Ambrósio and J. P. Gonçalves, "Complex flexible multibody systems with application to vehicle dynamics," *Multibody System Dynamics*, vol. 6, no. 2, pp. 163–182, 2001.

[31] W. F. Milliken and D. L. Milliken, *Race car vehicle dynamics.* Society of Automotive Engineers Warrendale, 1995, vol. 400.

[32] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle system dynamics*, vol. 21, no. S1, pp. 1–18, 1992.

[33] H. Pacejka and I. Besselink, "Magic formula tyre model with transient properties," *Vehicle system dynamics*, vol. 27, no. S1, pp. 234–249, 1997.

[34] W. C. Mitchell, A. Staniforth, and I. Scott, "Analysis of ackermann steering geometry," SAE Technical Paper, Tech. Rep., 2006.

[35] A. Rucco, G. Notarstefano, and J. Hauser, "Computing minimum lap-time trajectories for a single-track car with load transfer," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on.* IEEE, 2012, pp. 6321–6326.

[36] C. Sprunk, "Planning motion trajectories for mobile robots using splines," *University of Freiburg*, 2008.

[37] H. Prautzsch, W. Boehm, and M. Paluszny, *Bézier and B-spline techniques.* Springer Science & Business Media, 2013.

[38] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, 1992.

[39] J. E. Bobrow, "Optimal robot plant planning using the minimum-time criterion," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 443–450, 1988.

[40] Z. Shiller, Y. Fujita, D. Ophir, and Y. Nakamura, "Computing a set of local optimal paths through cluttered environments and over open terrain," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5. IEEE, pp. 4759–4764.

[41] H.-H. Lu, "Computation of path constrained time optimal lotions with dynamic singularities1," 1992.

[42] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[43] M. Leu and S. Singh, "Optimal trajectory generation for robotic manipulators using dynamic programming," *ASME JOURNAL OF DYNAMIC SYSTEMS, MEASUREMENT, AND CONTROL*, vol. 109, pp. 88–96, 1987.

[44] 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.

[45] M. Grant, S. Boyd, and Y. Ye, "CVX: MATLAB software for disciplined convex programming," 2008.

[46] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear algebra and its applications*, vol. 284, no. 1-3, pp. 193–228, 1998.

[47] M. Farivar, R. Neal, C. Clarke, and S. Low, "Optimal inverter VAR control in distribution systems with high pv penetration," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–7.

[48] S. C. Bengea and R. A. DeCarlo, "Optimal control of switching systems," *automatica*, vol. 41, no. 1, pp. 11–27, 2005.

[49] F. Oldewurtel, A. Parisio, C. N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth, "Energy efficient building climate control using stochastic model predictive control and weather predictions," in *American control conference (ACC), 2010*. IEEE, 2010, pp. 5100–5105.

[50] "MSC ADAMS," [online] http://www.mscsoftware.com/industry/motorsports, accessed: 2016-04-27.

[51] "Milliken Research Associates, Inc. (MRA)," [online] http://www.millikenresearch.com/vds.html, accessed: 2016-04-27.

[52] "OptimumG OptimumDynamics," [online] http://www.optimumg.com/software/optimumdynamics/, accessed: 2016-04-27.

[53] 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.

[54] "MATLAB Optimization Theory Overview - fmincon," [online] http://www.mathworks.com/help/optim/ug/optimization-theory-overview.html, accessed: 2016-04-28.