KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Development of an efficient trajectory optimization approach for a Formula Student electric race car

Gilles Bodart

Thesis submitted for the degree of Master of Science in Mechanical Engineering

Thesis supervisor:
Prof. dr. ir. W. Desmet

Assessors:
Dr. ir. J. Croes,
Prof. dr. ir. D. Moens

Mentors:
Dr. ir. F. Naets
Ing. G.-J.Vanderzypen

Academic year 2017 – 2018

Preface

I would like to thank everybody who helped and supported me in making this thesis. First of all I want to thank my promotor prof. dr. ir. Wim Desmet for giving me this unique opportunity. I also want to thank my mentor dr. ir. Frank Naets for his great help and feedback throughout the year. Next I want to say a big thank you to dr. ir. Joris Gillis for his help with the code implementation in CasADi. Further I would like to thank the jury for reading the text. I also want to thank Formula Electric Belgium for welcoming me in their team of motivated students and for their help. Finally, my sincere gratitude goes to my family and friends for supporting me in every way.

Gilles Bodart

Contents

Abstract

In this thesis, an optimal trajectory is calculated (the optimal racing line) to get the driving line which takes the least amount of time to accomplish a given race track. The idea is to apply these learnings to the Formula Student race car, Umicore Pulse, which has been built by the Formula Electric Belgium (FEB) team.

This calculated optimal racing line is used to assist the racing driver. It is important for FEB that the calculated optimal racing line can be repeated for different race tracks, that the computational time is limited, that the calculation is reliable, and that the complexity of an eventual implemented vehicle model is expandable.

To match these objectives, four different methods are discussed. The first method is the Euler spiral method, which is a heuristic method that uses an Euler spiral curve to fit the correct racing line through each corner. The second method assumes that the race car is a point mass and solves a non-linear optimization problem where an objective function, that consists of the curvature and the traveled distance on different moments along the race track, is minimized. This problem takes only the lateral acceleration into account. The third method solves again a non-linear optimization problem but this time formulated as an optimal control problem. It again assumes that the race car is a point mass but does take into consideration the total acceleration applied onto the race car (longitudinal and lateral). In this case, the controls are the longitudinal accelerations. The fourth and last method is again an optimal control problem but uses a vehicle model of the race car as well as a tyre model to better reflect the reality. In this case, the controls are the steering angle and the applied torque.

Matlab and CasADi/Opti have been used for the implementation of these methods.

The results show that the fourth method triggers the best results but it has a computational time which is currently too high to be useful for FEB. The third method is the best alternative but does not generate reliable speed and acceleration results. The first and second methods are less suitable.

Samenvatting

In deze masterproef wordt het optimale traject (de optimale racelijn) berekend om in zo weinig mogelijk tijd een ronde op een bepaalde racebaan te rijden. Het idee is om deze optimale racelijn te berekenen voor de Formula Student racewagen, Umicore Pulse, die gebouwd wordt door het Formula Electric Belgium (FEB) team.

De berekende optimale racelijn wordt gebruikt om de racepiloot te helpen. Het is belangrijk voor FEB dat de optimale racelijn voor verschillende racebanen kan berekend worden, dat de tijd die de computer nodig heeft voor het maken van de berekeningen beperkt is, dat de berekening betrouwbaar is en dat de complexiteit van een eventueel geïmplementeerd voertuigmodel uit te breiden is.

Om een methode te vinden die zo goed mogelijk aan deze doelstellingen voldoet, worden vier verschillende methodes besproken en vergeleken. De eerste methode is de Euler spiraal methode. Dit is een heuristische methode die een Euler spiraal curve gebruikt om de juiste racelijn door elke bocht te construeren. De tweede methode beschouwt de racewagen als een puntmassa en lost een niet-lineair optimalisatieprobleem op, waarbij een doelfunctie, die bestaat uit de kromming en afgelegde afstand op verschillende momenten langsheen de racebaan, wordt geminimaliseerd. Dit probleem houdt enkel rekening met de laterale versnelling. De derde besproken methode lost ook een niet-lineair optimalisatieprobleem op, maar deze keer geformuleerd als een optimaal controleprobleem. Het beschouwt de racewagen ook als puntmassa, maar de totale versnelling wordt in rekening gebracht (longitudinaal en lateraal). In dit geval zijn de controle variabelen de longitudinale versnellingen. De vierde en laatste methode is ook een optimaal controleprobleem, maar hierbij wordt een voertuig- en bandenmodel gebruikt, wat de realiteit beter weerspiegelt. In deze methode zijn de controle variabelen de stuurhoek en het aangelegde koppel.

Matlab en CasADi/Opti worden gebruikt voor de implementatie van bovenstaande methodes.

De resultaten tonen aan dat de vierde methode het beste resultaat geeft, maar de tijd die nodig is voor de berekening is voorlopig te lang om gebruikt te kunnen worden door FEB. De derde methode biedt het beste alternatief, maar deze genereert geen betrouwbare snelheids- en versnellingsresultaten. De eerste en tweede methode zijn minder geschikt.

List of Figures

List of Tables

List of Abbreviations

OCP	Optimal Control Problem
FEB	Formula Electric Belgium
MTM	Minimum time maneuvering
MPC	Model predictive control
NLP	Non-linear programming problem
QSS	Quasi-steady-state
COG	Center of gravity
SLAM	Simultaneous localization and mapping
2D	Two dimensional
IPOP	Interior point optimizer
DOF	Degree of freedom
FSG	Formula Student Germany

Chapter 1

Introduction

In motor sport racing the goal is to be the fastest. Many of the racing drivers trust on their practical experience and on their feeling for the sport. The calculation of the optimal racing line is an interesting assistance tool that can help the racing driver to set a better lap time.

The reason why this optimal racing line is so important is because "statistics show that successful car racing champions are always following the optimal racing line while other racers are constantly not getting the optimal line"[24]. At first sight, staying on the inside of a corner seems the best thing to do. In this way the traveled distance is minimized. However, the speed throughout this racing line is much smaller than when the racing driver would stay at the outside of the corner. This has to do with the curvature of the race line to be followed. The lateral acceleration for a point mass is given by following formula: $a_y = v^2/r$ with v the longitudinal speed and r the radius of the corner. The curvature is given by k and is equal to: $k = \frac{1}{r}$. When the inside line is chosen, r is smaller and thus for a constant a_y, the speed is smaller, but so is the traveled distance. The other way goes around for the outside line. The truth is that the optimal racing line will be a trade off between the speed and the length of the racing line taken. [24]

Formula Electric Belgium (FEB) wants to gather the optimal trajectory which takes the least amount of time to accomplish the given race track, in this case a Formula Student race track. The race car is the Formula student race car, Umicore Pulse, built by FEB. The main objective is to calculate this optimal trajectory, or in other words the optimal racing line. Further, it is important for FEB that the calculation of the racing line can be repeated for different race tracks, that the computational time is limited, that the calculation is reliable and that the complexity of an eventual implemented vehicle model is expandable.

To calculate this optimal racing line, also commonly referred to as minimum-time maneuvering (MTM) problem, a lot of methods are available in the literature. "Optimal control calculations for road cars go back to the 1950s, when simple heuristic

arguments were employed by Mercedes Benz to estimate lap times and optimize set-up parameters" [19][18]. Nowadays, the main methods used for minimum-time maneuvering are the quasi-steady-state (QSS) methods [16], the linear time-varying model predictive control (MPC) methods [20] and the large non-linear programming problem (NLP) methods [16][18]. The QSS method is mainly used for tuning the parameters of the race car to set the best lap time given the optimal racing line. Because of the fact that the main purpose of this thesis is to calculate the optimal racing line, this method is not used. The MPC method its linear approximation "allows one to define the problem of finding a suboptimal racing line and speed profile as a sequence of convex optimization problems. The computed trajectory is sub-optimal, because linear models are used (as opposed to non-linear ones) in conjunction with control problems that are defined over a sequence of receding track segments (as opposed to the whole track)" [18]. In contrast to the MPC method, the NLP method has the advantage to simultaneously calculate the controls (steering angle δ and torque T) of the race car, "one or more set-up parameters, such as the location of the car its center of mass and/or its center of pressure" [18] and of course the optimal racing line coordinates. This is a very valuable tool that FEB can use in the future to further optimize their race cars and to assist the driver. Although the NLP method is the slowest of the previous discussed methods, it offers the most possibilities for FEB and matches their objectives best. This is why this method is chosen to be elaborated further in the text.

Before the elaboration of this NLP method in this thesis, a heuristic method is chosen to be worked out, namely the Euler spiral method. This is the first discussed method in the text and mainly based on practical experience and information. It gives a good initial understanding of the MTM problem. It is also a fast computational method and proven to be applicable for a complete (simple) race track in [24]. Next, the NLP method is divided in three non-linear optimization problems in order of increasing complexity. The second method assumes that the race car is a point mass and solves a non-linear optimization problem where an objective function, that consists of the curvature and the traveled distance on different moments along the race track, is minimized. It only takes the lateral acceleration into consideration. The third method also solves a non-linear optimization problem but this time formulated as an optimal control problem (OCP). The race car is again assumed to be a point mass but this time the total acceleration (longitudinal and lateral) is taken into account. The longitudinal accelerations are the controls for the OCP. The fourth method is again an OCP, but this time a vehicle model and tyre model are used to better reflect the reality. The controls are the steering angle and applied torque on different moments along the race track. Matlab and Opti [2] in CasADi [4] are used for the implementation of these methods.

Chapter 2 describes the problem statement and the objective in more detail and tells something about the coherence between this thesis and two other theses written out by the KU Leuven [9] [6]. Chapter 3 goes in more detail concerning the four different methods and why they are chosen to calculate the optimal racing line. It further

describes how these methods can be used to calculate the optimal racing line and how they are implemented in Matlab. Chapter 4 discusses the results of the four discussed methods and compares them with each other. The comparison is done by projecting the results on each other and discussing the differences in position, velocity, acceleration, average curvature, computational time, number of variables and how good the different methods fit the given objectives of FEB relative to each other. Chapter 5 concludes this work and explains what could be added in future work relating to the objectives of FEB.

Chapter 2

Problem statement and objective

To calculate the optimal racing line, some very important information is necessary.

First of all the race track information is necessary. This includes the coefficient of friction between the race track and the tyres of the race car but also the (x, y) coordinates of the inside and outside boundaries of the race track.

Next, it is important to have the characteristics of the race car. These include the maximum steering angle, speed and torque, but also the position of the center of gravity (COG), the length and width of the car, etc.

Another important aspect is the need for a reliable vehicle model. The characteristics of the race car function as an input to set up the reliable vehicle model. For some methods discussed in this thesis, where the race car is assumed to be a point mass, the need for this reliable vehicle model is of course not true. However, to fully understand the dynamics of the race car, a reliable vehicle model will always be needed.

Finally, it is important to localize the race car during the race over the race track. In this way, visual assistance can be given to the racing driver to help him or her to stay on the optimal racing line.

Figure 2.1 gives an overview between the different information sources.

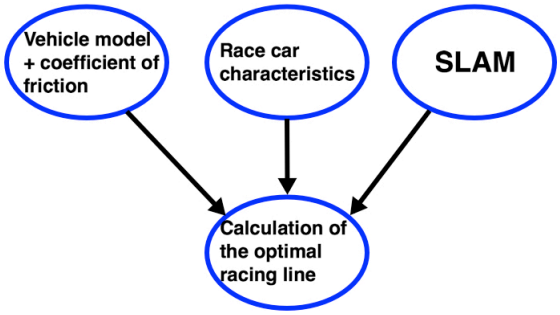

Figure 2.1: Coherence between different information sources.

In the figure above, SLAM is the abbreviation for Simultaneous Localization And Mapping. The mapping is used to obtain the information of the race track or in other words to obtain the (x, y) coordinates of the inside and outside boundaries. The localization is the real time localization of the race car on the race track, which is necessary for the racing driver's visual assistance.

The left upper balloon and the right upper balloon are the subject of two other theses at the KU Leuven that give input to the calculation of the optimal racing line [9] [6]. The third input, the race car characteristics, is data provided by Formula Electric Belgium (FEB).

2.1 Problem statement

FEB would like to participate in autonomous electric car racing in the next few years. This is why this work is very important for them. For now, the calculation of the optimal racing line will function as an input for an assistance system that tells the driver how he or she should drive to get the best lap time around the race track. The goal in racing is of course to set the fastest lap time. The calculation of the optimal racing line has to work for different race tracks where FEB drives on. Most of the time these are not official race tracks, but race tracks that are built by the Formula Student organization. The races take place around Europe. Each Formula Student competition consists of different races, spread over a race weekend. One of the races is an endurance race of 22 kilometer where one lap is one kilometer. A day before the endurance race, the Formula Student teams get the chance to walk for 30 minutes over the constructed endurance race track. This is the chance for FEB to gather the track information. Therefore, the algorithm that calculates the optimal race line has a computational time constraint of about 24 hours.

2.2 Objective

The main objective is of course to generate the optimal racing line for the Formula Student race car "Umicore Pulse". It is further important that this can be repeated for different race tracks, that the computational time is limited, that the calculation is reliable and that the complexity of an eventual implemented vehicle model is expandable.

To make sure that an optimal racing line can be generated for different race tracks, a more general approach is needed. This is done by introducing three non-linear optimization methods that can be used for every 2D race track as long as the constraints on the race car and on the track are satisfied.

To limit the computational time for the calculation of the optimal racing line, the four listed methods are discussed in order of increasing complexity (number of variables). This increase in complexity is directly proportional to the computational time.

To get a reliable optimal racing line, the outcomes of the four discussed methods are compared with each other in chapter 4.

The fourth method gives the chance to FEB to calculate the optimal racing line for a specific vehicle model. The outputs of this method are the torque and steering angle on different moments along the race track. These outputs can then function as input for a controller that makes the race car driverless. This is in line with their vision of the future. A more complex vehicle model can also be easily implemented in the fourth method.

Chapter 3

Different methods

In this chapter, the different methods to calculate the optimal racing line which were already pointed out in chapter 1 are discussed . It starts with the first method, the Euler spiral method. Next, in the second method, there is a non-linear optimization problem, that only focuses on the lateral acceleration of the race car, which is assumed to be a point mass. Further, in the third method, there is an optimal control problem (OCP) which focuses on the total acceleration (longitudinal and lateral) where the race car is again assumed to be a point mass. The controls are the longitudinal accelerations on different moments along the race track. The fourth and last method is again an OCP which uses a vehicle model and tyre model to better reflect the handling of the race car. The controls are the steering angle and applied torque on different moments along the race track.

3.1 Euler spiral method

The Euler spiral method uses an Euler spiral to fit through each of the corners of the race track to find the optimal racing line. The Euler spiral that leads for a given corner to the optimal racing line segment is based on some assumptions and constraints [24]. According to Adam Brouillard's book [8] [22] it is possible to use this Euler spiral method to generate the optimal racing line and it is already successfully applied in [24].

"An Euler spiral is a curve whose curvature changes linearly with its curve length" [22]. It is a curve generated by a parametric plot of S(s) against C(s), with S(s) and C(s) the Fresnel integrals [23]:

$$\begin{cases} C(s) = \int_0^s \cos(s^2) ds \\ S(s) = \int_0^s \sin(s^2) ds \end{cases} \quad s \in (0, s_{end})$$

with s the length measured along the spiral curve from its initial position and s_{end} the total length of the spiral curve. For an Euler spiral the following applies [22]:

$$2Rs = 2R_c s_{end} = \frac{1}{a^2}$$

or

$$\frac{1}{R} = \frac{s}{R_c s_{end}} = 2a^2 s$$

The (x, y) coordinates are determined as follows [22]:

$$\begin{cases} x(s) = \frac{1}{a}\int_0^{s'} \cos(s^2)ds \\ y(s) = \frac{1}{a}\int_0^{s'} \sin(s^2)ds \end{cases} \quad s \in (0, s_{end})$$

with $s' = as$ and $a = \frac{1}{\sqrt{2s_{end}R_c}}$, R is the radius of curvature on different moments along the Euler spiral and R_c is the radius of curvature at the end of the Euler spiral. For a normalized Euler spiral $a = 1$ and holds thus that $x(s) = C(s)$ and $y(s) = S(s)$. Figure 3.1 shows the graph of the normal Fresnel integrals, with argument $\frac{\pi}{2}s^2$ instead of s^2 [23]. Figure 3.2 shows a double-end Euler spiral. [22]

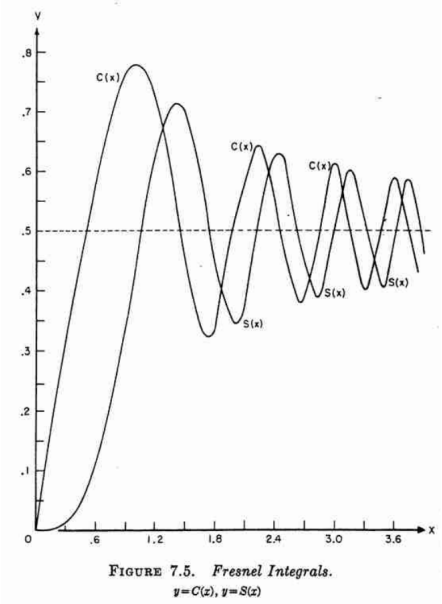

FIGURE 3.1: Fresnel integrals [5].

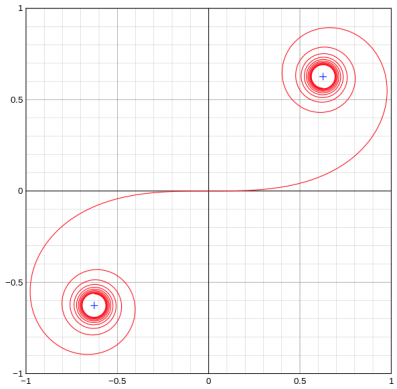

FIGURE 3.2: Double-end Euler spiral [22].

3.1.1 Reference points

According to race driver and author Ross Bentley, there are three important reference points in racing. The first one is the turn-in point, the second one is the apex and the third and last one is the exit point. Figure 3.3 gives an overview. [7]

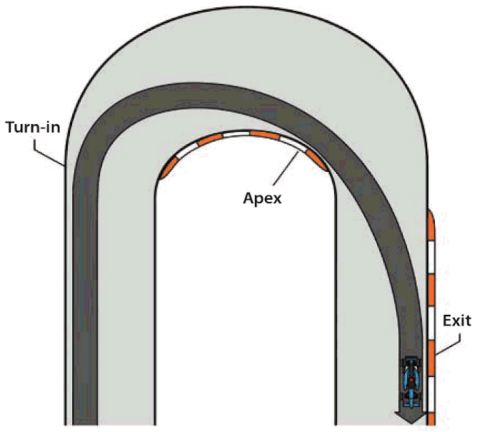

FIGURE 3.3: Typical 180 degree corner with three reference points [7].

The turn-in point is the point where the racing driver begins to turn the steering wheel to get through the corner. The apex is the point on the race track where the optimal race line touches the inside part of the corner. The exit point is the point where the race car runs closest to the outside part of the race track for the given corner. [7]

It is important to mention that more than one apex is possible for a certain corner, dependent on the turn-in point. If the turn-in point changes, the apex changes and

so does the exit point. "An apex can be early in a turn, in the middle of it, or late in a turn" [7]. The positioning of the ideal apex, the apex that belongs to the optimal racing line, depends on many factors. [7]

One of these factors are the cornering combinations over the race track. A late apex, and thus a late turn-in point ensures a high exit speed. This latter is very useful when there is a long straight after the corner. An early apex is useful in corners that have a very wide exit because an early apex ensures high cornering speeds, thus high lateral accelerations for a long time. This leads the race car to the outside part of the race track and because of the high lateral accelerations for a long time, the race driver cannot accelerate as early out of the corner in comparison to a late apex. An early apex results in lower corner exit speeds but higher cornering speeds. [7]

Another factor where the position of the apex depends on, are the properties of the race car. A race car with a lot of power is in some cases better off with a late apex. The race car will have a lower speed in the corner but can start to accelerate faster which in turn gives a higher speed at the end of a straight for example. An early apex is in some cases thus better for lower powered race cars. [7]

3.1.2 Constraints

With the Euler spiral method, an Euler curve will be fitted through the given corner(s). It is thus important that this fit satisfies some constraints. These constraints consist of [24]:

- The position of the turn-in point.
- The position of the apex.
- The position of the exit point.
- The fitted Euler spiral through a given corner does not cut that given corner.
- The angle difference of the tangents of the start and end point of the Euler spiral segment is equal to the number of degrees of the given corner.

3.1.3 Implementation

The method is applied on a 90 degree corner. The first step is to construct this 90 degree corner. The inside and outside boundaries of the corner consist of a high density of points that are connected by a cubic spline curve.

The next step is fitting different Euler spiral segments through the 90 degree constructed corner [17]. The reason why different Euler spiral segments fit through for example a 90 degree corner is explained on the basis of figure 3.4. What can be seen, is that the difference in angle between the blue tangents, the green tangents and the purple tangents is always 90 degrees. Yet, the length of each of the Euler

spiral segments gets smaller. This is explained by the definition of the Euler spiral: "an Euler spiral is a curve whose curvature changes linearly with its curve length" [22]. Thus fitting of the different Euler spiral segments through the 90 degree corner works by changing the starting points of the Fresnel integrals for the Euler spiral and by ending the integration at the point where the tangent angle difference makes 90 degrees. Figure 3.5 shows different Euler spiral segments through the 90 degree constructed corner. [24]

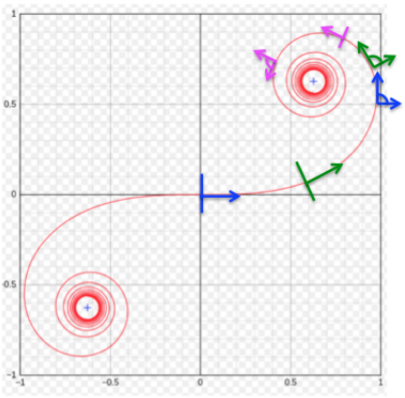

FIGURE 3.4: Euler spiral with different 90 degree sections [22].

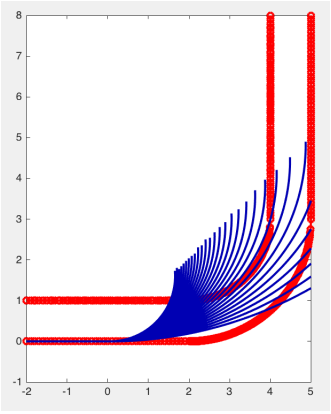

FIGURE 3.5: 90 degree corner with different Euler spiral segments.

To fit the optimal Euler spiral segment through a given corner, the five previously discussed constraints need to be implemented.

The first constraint is on the turn-in point. Where this turn-in point should be located to get the optimal Euler spiral segment is discussed in section 3.1.1. In the algorithm for selecting the optimal Euler spiral curve, the turn-in point is not a variable.

The second constraint is on the apex. The optimal racing line has always an apex, a point on the optimal racing line that touches the most inside part of the corner. This can be mathematically expressed as follows: $f'_{inside}(x_r) = g'_{optimal}(x_r)$, whereas x_r is the x coordinate of the apex, f'_{inside} and $g'_{optimal}$ are respectively the derivatives of the inside part of the corner and of the optimal racing line.

The third constraint is on the exit point. Where this exit point should be located to get the optimal Euler spiral segment is discussed in section 3.1.1. Just like the turn-in point, the exit point is not a variable in the algorithm for selecting the optimal Euler spiral curve. However, the exit point is chosen to be at the most outside of the race track. As discussed earlier in chapter 1, the speed through the corner increases if the curvature decreases. This is the reason to choose for an exit point close to the outside of the race track. In this way, the average curvature of the optimal Euler spiral segment decreases and a higher velocity can be carried throughout the corner.

The fourth constraint says that the angle difference of the tangents of the start and end point of the Euler spiral segment is equal to the number of degrees of a certain given corner. Thus in this case, it is 90 degrees.

The fifth and last constraint says that the fitted Euler spiral through a given corner does not cut that given corner.

Implementing all of these constraints in Matlab, provides the optimal Euler spiral segment for the 90 degree corner. This optimal Euler spiral segment can be seen in figure 3.6.

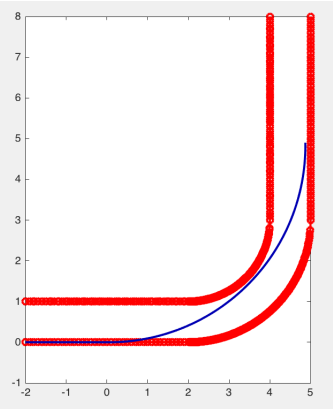

Figure 3.6: Optimal Euler spiral segment

3.1.4 Curvature and velocity calculation

For the calculation of the velocity in the corner, only the lateral acceleration is taken into consideration. In racing, it is most of the time the case that the racing driver brakes before the corner, steers in and accelerates out of the corner [7]. In that case, the racing driver is thus generating more speed at the final part of the corner. This is why the Euler spiral is fitted through the given corner in reverse order. The curvature decreases linearly along the reversed Euler spiral. If only the lateral acceleration is taken into consideration, the velocity will thus increase along the given corner. Mathematically this can be expressed as follows [24]:

$$ma_{y,i} \leq F_w \tag{3.1}$$

$$m\frac{v_i^2}{r_i} \leq \mu mg \tag{3.2}$$

$$v_i \leq \sqrt{\frac{\mu g}{k_i}} \tag{3.3}$$

with $a_{y,i}$ the lateral acceleration on different moments along the corner, with i = 1,2,...,n and n equal to the total number of points where the optimal Euler spiral consists of, F_w the friction force and equal to μmg. Further, m is the mass of the race car, v_i the longitudinal speed of the race car on different moments, with i = 1,2,...,n, r_i the radius of the corner at different moments, with i = 1,2,...,n, g the gravitational constant and μ the friction coefficient between the race track surface and the tyres of the race car. Because in equation 3.3 μ and g are constants, the speed v_i is obtained by knowing the curvature of the corner k_i at different moments.

The calculation of the curvature k_i at different moments is done by using figure 3.7. Here, $(x_i, y_i), (x_{i+1}, y_{i+1})$ and (x_{i+2}, y_{i+2}) are the coordinates of three discrete points that represent a segment of the optimal racing line. The curvature is given by [21]:

$$k = \frac{d\theta}{ds} \tag{3.4}$$

or in discrete form:

$$k_i = \frac{\Delta\theta_i}{\Delta s_i} = \frac{\theta_{i+1} - \theta_i}{s_{i+1} + s_i} \quad i = 1, 2, ..., n-2 \tag{3.5}$$

with

$$\Delta\theta_i = cos^{-1}\left(\frac{\vec{a_i} \cdot \vec{b_i}}{|\vec{a_i}| \cdot |\vec{b_i}|}\right) \quad i = 1, 2, ..., n-2 \tag{3.6}$$

where

$$\vec{a_i} = \begin{bmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{bmatrix} \quad i = 1, 2, ..., n-1 \tag{3.7}$$

$$\vec{b_i} = \begin{bmatrix} x_{i+2} - x_{i+1} \\ y_{i+2} - y_{i+1} \end{bmatrix} \quad i = 1, 2, ..., n-2 \tag{3.8}$$

and

$$s_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad i = 1, 2, ..., n-1 \tag{3.9}$$

$$s_{i+1} = \sqrt{(x_{i+2} - x_{i+1})^2 + (y_{i+2} - y_{i+1})^2} \quad i = 1, 2, ..., n-2 \tag{3.10}$$

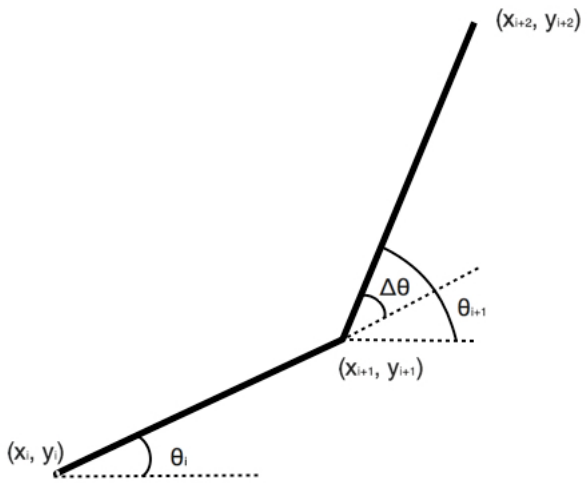

Figure 3.7: Calculation of the curvature on different moments along the corner

The last two points lie on the same circle as point i = n-2, so it holds that $k_n = k_{n-1} = k_{n-2}$.

3.2 Lateral acceleration focused non-linear optimization problem

The second method for the calculation of the optimal racing line is described in this section. It is done by introducing a non-linear optimization problem with certain constraints and an objective function. This non-linear optimization problem is solved using fmincon as commercial non-linear solver in Matlab. Of course, other commercial non-linear solvers could have been used too. The race car is assumed to be a point mass and in the generation of the solution, only the lateral acceleration is taken into account.

What is important for the optimal racing line is that it ensures the racing driver to accomplish the given race track in the shortest amount of time. The total time t can be written as: $t = \int dt = \int \frac{ds}{v}$, with dt the infinitesimal time, ds the infinitesimal distance and v the longitudinal speed of the race car. The general non-linear optimization problem looks like this [24]:

$$min \int dt = min \int \frac{ds}{v}$$
$$s.t. \quad kv^2 - \mu g \leq 0$$
$$\int ds = s_{lap}$$
$$k \leq k_{max}$$
$$0 < v \leq v_{max}$$
$$a_{min} \leq a \leq a_{max}$$

with k_{max} the maximum curvature possible that depends on the maximum steering angle δ_{max} of the race car, s_{lap} the total distance needed to accomplish one lap around the race track, v_{max} the maximum longitudinal speed possible by the race car, a_{min} the minimum longitudinal acceleration due to braking and a_{max} the maximum longitudinal acceleration provided by the engine.

The objective function $\int \frac{ds}{v}$ can be written differently. Taking only the lateral acceleration into account, the longitudinal speed v can be replaced by equation 3.3. Next, $\frac{1}{v}$ can be written as [24]:

$$\frac{1}{v} \geq \sqrt{\frac{k}{\mu g}}$$

The objective function $\int dt$ is then equal to [24]:

$$\int dt = t \geq \frac{1}{\sqrt{\mu g}} \int_0^{s_{lap}} \sqrt{k} ds$$

The goal is thus to minimize $\frac{1}{\sqrt{\mu g}} \int_0^{s_{lap}} \sqrt{k} ds$. [24]

3.2.1 Implementation

The curvature k(i), with i = 1,2,...,n-2, and the distance step $\Delta s(i)$, with i = 1,2,...,n-2, can be written in function of the coordinates (x_i, y_i), with i = 1,2,...,n, of the optimal racing line with the use of equations 3.5, 3.6, 3.7, 3.8, 3.9 and 3.10. Here, n is equal to the number of points of the optimal racing line. The coordinates (x_i, y_i) are the variables that can change during the optimization. In the end, the optimization algorithm needs to return the final (x_i, y_i) coordinates that minimizes $\frac{1}{\sqrt{\mu g}} \int_0^{s_{lap}} \sqrt{k} ds$.

The discrete non-linear optimization problem can be formulated as follows [24]:

$$min \sum_{i=1}^{n-2} \frac{1}{\sqrt{\mu g}} \sqrt{k(i)} \frac{\Delta s(i)}{2} \tag{3.11}$$

$$\begin{aligned} s.t \quad & k(i) - k_{max} && \leq 0, \quad i = 1, 2, ..., n-2 \\ & (x(i) - x_c(i))^2 + (y(i) - y_c(i))^2 - (\frac{1}{2} w(i))^2 && \leq 0, \quad i = 1, 2, ..., n \\ & \begin{bmatrix} x(n) \\ y(n) \end{bmatrix} - \begin{bmatrix} x(1) \\ y(1) \end{bmatrix} && = 0 \end{aligned} \tag{3.12}$$

with $\Delta s(i)$ equal to:
$\Delta s(i) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} + \sqrt{(x_{i+2} - x_{i+1})^2 + (y_{i+2} - y_{i+1})^2}$, with i = 1,2...,n-2. Here, $\frac{\Delta s(i)}{2}$ is the average distance where the curvature k_i holds. Further, $(x_c(i), y_c(i))$ are the centerline coordinates, with i = 1,2,...,n, w(i) is the width of the race track at different moments, with i = 1,2,...,n, μ is the friction coefficient between the race car its tyres and the road, k_{max} is the maximum curvature allowed and calculated in appendix A.

To get a fast and reliable solution from the algorithm that calculates the non-linear optimization problem, it is important that a good initial guess is given for the optimization variables. In this case, the centerline coordinates (x_c, y_c) are given as initial guess for the (x, y) coordinates.

3.2.2 Track width calculation

The constraints of the non-linear optimization problem require the centerline coordinates $x_c(i)$ and $y_c(i)$ and the width of the race track $w(i)$ at different moments along the race track. Figure 3.8, 3.9, 3.10 and 3.11 show how these inputs are calculated.

The inside and outside boundaries of the race track consist of given points (x_i, y_i) that are connected with each other by using splines. The width $w(i)$ on different moments along the race track is calculated by selecting each time five points of the inside boundary of the race track: $(x_{in,i}, y_{in,i})$, $(x_{in,i+1}, y_{in,i+1})$, $(x_{in,i+2}, y_{in,i+2})$, $(x_{in,i+3}, y_{in,i+3})$ and $(x_{in,i+4}, y_{in,i+4})$. The first point $(x_{in,i}, y_{in,i})$ and the fifth point $(x_{in,i+4}, y_{in,i+4})$ are connected by a straight line and the slope n_k of the normal on this straight line is calculated by using following formula: $n_{k,i} = \frac{-(x_{in,i+4} - x_{in,i})}{y_{in,i+4} - y_{in,i}}$. The next step is to draw the normal with previously calculated slope through the third point $(x_{in,i+2}, y_{in,i+2})$. This normal has following equation: $y = n_{k,i}(x - x_{in,i+2}) + y_{in,i+2}$. The next step is to find the point of the outside boundary of the race track associated with the third point $(x_{in,i+2}, y_{in,i+2})$ for calculating the width of the race track. This is done by searching the minimum perpendicular distance from the normal through the third point to the points of the outside boundary of the race track. This can be formulated in mathematical formulation as follows: search the point at the outside

boundary of the race track that minimizes the distance $d = \frac{|ax_{out,j}+by_{out,j}+c|}{\sqrt{a^2+b^2}}$. Here $a = -n_k$, $b = 1$, $c = n_k x_{in,i+2} - y_{in,i+2}$ and $(x_{out,j}, y_{out,j})$ are the coordinates of the points of the outside boundary of the race track. Once the corresponding point has been found, the width can be calculated by computing the distance between the selected point on the outside boundary of the race track and the third point on the inside boundary of the race track. Mathematically this can be formulated as follows: $w(i) = \sqrt{(x_{out,j} - x_{in,i+2})^2 + (y_{out,j} - y_{in,i+2})^2}$.

After the calculation of the widths $w(i)$ at different moments along the race track, the centerline coordinates $(x_{c,i}, y_{c,i})$ can be easily found. This is displayed in figure 3.10 and figure 3.11.

If $\theta_i = tan^{-1}(n_{k,i})$, then is $y_{c,i} = y_{in,i+2} - sin(\theta_i)\frac{w_i}{2}$ and
$x_{c,i} = (y_{c,i} - \frac{y_{in,i+2}}{n_{k,i}}) + x_{in,i+2}$ with $(x_{c,i}, y_{c,i})$ the centerline coordinates.

All of these calculations are implemented in the Matlab function *generate_track_22.m*.

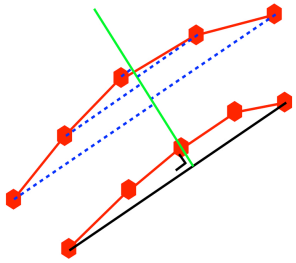

FIGURE 3.8: Selection of the corresponding point

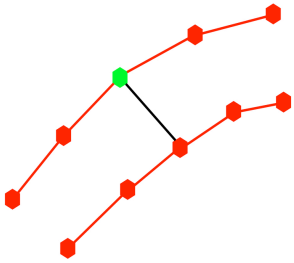

FIGURE 3.9: Measurement of the width between the two corresponding points

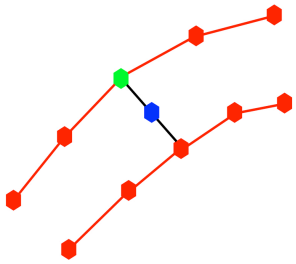

FIGURE 3.10: Selection of the centerline point

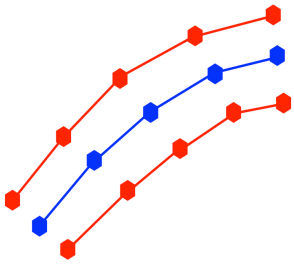

FIGURE 3.11: Generation of the centerline coordinates

3.2.3 Velocity calculation

The longitudinal velocity is calculated based on the flow chart below for finding this velocity along the calculated optimal racing line [24].

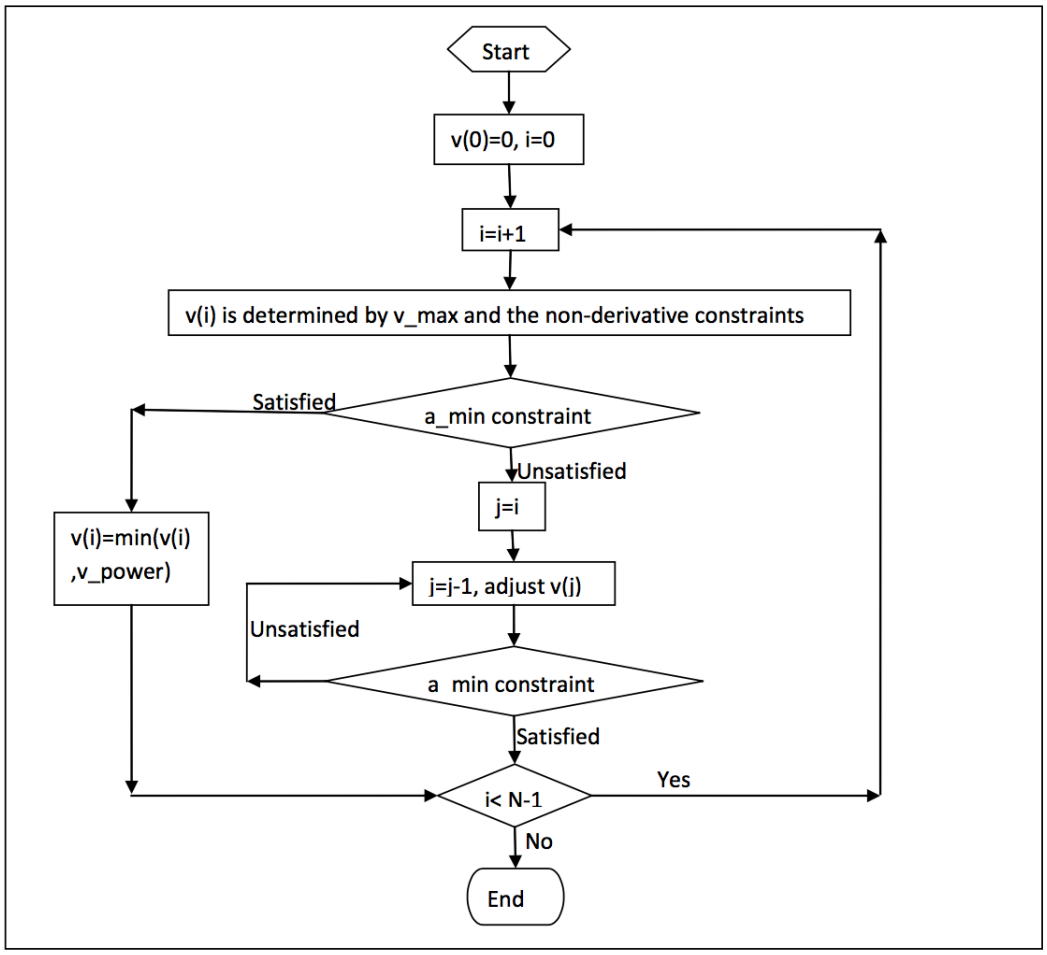

Figure 3.12: Flow chart for the calculation of the velocity of the previously calculated optimal racing line [24].

At $i = 0$, the velocity $v = 0$. For the next points, $v(i)$ is calculated based on following formula: $v(i) = \sqrt{\frac{\mu g}{k(i)}}$, taking into account following non-derivative constraint: $0 \leq v(i) \leq v_{max}$. [24]

After the calculation of the different velocities $v(i)$, a_{min} and the power P constraints are introduced. Here, P is the maximum power generated by the race car its engine. The longitudinal acceleration $a(i)$, on different moments along the race track, can be approached by linear interpolation, assuming a constant acceleration between two consecutive points. Taking into account two points along the race track with

speeds $v(i)$ and $v(i+1)$, then $a(i) = \frac{v(i+1)-v(i)}{\Delta t}$. Because of linear interpolation, the average speed between two consecutive points is equal to: $v_{avg} = \frac{v(i+1)+v(i)}{2}$. Thus $\Delta t = \frac{\Delta s}{v_{avg}}$ with $\Delta s = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$. Substituting this in the previous equation for the acceleration $a(i)$, this gives: $a(i) = \frac{(v(i+1)+v(i))(v(i+1)-v(i))}{2\Delta s}$. In this way, the previously calculated $v(i+1)$ can be validated by the addition of following constraints: $\frac{(v(i+1)+v(i))(v(i+1)-v(i))}{2\Delta s} \geq a_{min}$ and $\frac{(v(i+1)+v(i))(v(i+1)-v(i))}{2\Delta s} \leq a_{max}$. Here, a_{max} is determined by the maximum power P of the race car its engine and a_{min} is determined by the maximum negative braking acceleration. If the a_{min} constraint is violated, backwards adjustment steps are done and the velocity v is adapted until the a_{min} constraint is satisfied. [24]

All of these calculations are implemented in the Matlab function *velocity_n_curvature.m*

3.3 Optimal control problem for a point mass with lateral and longitudinal accelerations

In this section, an optimal control problem (OCP) is introduced to calculate the optimal racing line. The Formula Student race car is assumed to be a point mass with mass m, and the controls are the longitudinal accelerations a_x and a_y. These accelerations are expressed in the absolute x, y direction, with $a_{net} = \sqrt{a_x^2 + a_y^2}$ which is the total net longitudinal acceleration of the point mass. The state vectors X and dX are:

$$X = \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}, dX = \begin{bmatrix} v_x \\ v_y \\ a_x \\ a_y \end{bmatrix} \tag{3.13}$$

with X and dX expressed in the absolute coordinate system.

The reason why this OCP problem is introduced is because in this case a distinction can be made for high and low powered accelerating race cars. As discussed in section 3.1.1, the optimal racing line depends on the characteristics of the race car. This method gives the possibility to take this acceleration constraint into account to find the corresponding optimal racing line.

3.3.1 Implementation

The implementation of the algorithm is done in Matlab, using Opti [2] in CasADi [4] as optimization solver. Again, other commercial optimization solvers could have been used.

First the centerline coordinates (x_c, y_c) and the widths $w(i)$ are generated with the previously discussed Matlab function *generate_track_22.m*. Here i = 1,2,...,n

and n is the total number of points of the optimal racing line.

Next the (x_c, y_c) coordinates and widths $w(i)$ are linearly interpolated generating a denser group of centerline coordinate points and widths. This is done using the function *interparc.m* [10] out of the Matlab function library.

During the update process of the states, a changing update step is used. As in [18], the independent variable dt, the infinitesimal time, is changed in ds, the infinitesimal distance expressed in the absolute x,y coordinate system. This can only be done if there is a direct relationship between these two variables. The variable dt can be expressed as: $dt = \frac{ds}{v}$ with v the longitudinal speed of the race car expressed in the absolute x,y coordinate system. The reason for this change in independent variable is because it has the advantage of "maintaining an explicit connection with the track position" [18]. It is important to note that the longitudinal speed v of the race car should always be strictly positive.

Further the CasADi optimization environment is created and the optimization variables and controls are introduced. The optimization variables are the states x, y, v_x, v_y and Δs, the displacement step. The optimization controls are the longitudinal accelerations a_x and a_y.

The optimal control problem can be formulated as follows [18]:

$$min \int_{s_0}^{s_{lap}} l(s, x(s), u(s))ds \tag{3.14}$$

$$\begin{aligned} s.t \quad & \frac{dx}{ds} - f(s, x(s), u(s)) && = 0 \\ & g(s, x(s), u(s)) && = 0 \\ & h(s, x(s), u(s)) && \leq 0 \\ & g_b(x(s_0), x(s_{lap})) && = 0 \end{aligned} \tag{3.15}$$

where s is the distance traveled by the race car, x(s) is the state vector and u(s) is the control vector. Here, s_0 is the traveled distance at the start of the race track, or equal to 0 and s_{lap} is the total traveled distance by the race car to accomplish one lap around the race track. The vector function $f(\cdot)$ normally describes the vehicle dynamics but in this case it is a vector that consists of the speeds v_x and v_y, and also the controls a_x and a_y, divided by the average speed $v_{abs}(i) = \frac{\sqrt{v_x(i)^2+v_y(i)^2}+\sqrt{v_x(i+1)^2+v_y(i+1)^2}}{2}$ between two consecutive points (i = 1,2,...,n-1). "The vector functions g(·) and $h(\cdot)$ define the equality and inequality constraints for the system (e.g. the track topology and box constraints on the states and controls); the subscript b refers to the boundary constraints with $g_b(\cdot)$. The scalar function $l(\cdot)$ is the stage cost that is a function of the state and the controls" [18].

For the method presented here, the discrete optimal control problem can be written as follows:

$$min \frac{\sum_{i=1}^{n-1} \Delta s(i)}{\sum_{i=1}^{n} \frac{\sqrt{v_x(i)^2+v_y(i)^2}}{n}} \tag{3.16}$$

$$\begin{aligned}
s.t \quad & \begin{bmatrix} \Delta x(i+1) \\ \Delta y(i+1) \\ \Delta v_x(i+1) \\ \Delta v_y(i+1) \end{bmatrix} - \frac{\Delta s(i)}{\frac{\sqrt{v_x(i)^2+v_y(i)^2}+\sqrt{v_x(i+1)^2+v_y(i+1)^2}}{2}} \begin{bmatrix} v_x(i) \\ v_y(i) \\ a_x(i) \\ a_y(i) \end{bmatrix} && = 0, \quad i = 1,2,...,n-1 \\
& k(i) - k_{max} && \leq 0, \quad i = 1,2,...,n-2 \\
& a_x(i)^2 + a_y(i)^2 - a_{brake,max}^2 && \leq 0, \quad i = i_{brake,1}, i_{brake,2}, ..., n_{brake} \\
& a_x(i)^2 + a_y(i)^2 - a_{acc,max}^2 && \leq 0, \quad i = i_{acc,1}, i_{acc,2}, ..., n_{acc} \\
& a_x(i)^2 + a_y(i)^2 + a_{lat}(i)^2 - (\mu g)^2 && \leq 0, \quad i = 1,2,...,n-1 \\
& (x(i) - x_c(i))^2 + (y(i) - y_c(i))^2 - (\frac{1}{2}w(i))^2 && \leq 0, \quad i = 1,2,...,n \\
& 0 < v_x(i)^2 + v_y(i)^2 && \leq v_{max}^2, \quad i = 1,2,...,n \\
& x_{min} \leq x(i) && \leq x_{max}, \quad i = 1,2,...,n \\
& y_{min} \leq y(i) && \leq y_{max}, \quad i = 1,2,...,n \\
& \Delta s_{min} \leq \Delta s(i) && \leq \Delta s_{max}, \quad i = 1,2,...,n-1 \\
& \begin{bmatrix} x(n) \\ y(n) \\ v_x(n) \\ v_y(n) \end{bmatrix} - \begin{bmatrix} x(1) \\ y(1) \\ v_x(1) \\ v_y(1) \end{bmatrix} && = 0
\end{aligned} \tag{3.17}$$

with Δs the displacement step and equal to: $\Delta s(i) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$, with i = 1,2,...,n-1. The small time step used for the update process is equal to: $\Delta t(i) = \frac{\Delta s(i)}{\frac{\sqrt{v_x(i)^2+v_y(i)^2}+\sqrt{v_x(i+1)^2+v_y(i+1)^2}}{2}}$, with i = 1,2,...,n-1. This is an implicit time step and also the time needed for the race car to drive $\Delta s(i)$ in distance with a speed equal to $v_{abs}(i)$. Notice that Δs, and thus also Δt, is a variable step for the update process so that it updates the states in such a way that the constraints are satisfied. The main problem if a constant time step would have been used, is that the constraint in the discrete OCP which covers the boundaries for the race track, would be violated in the update process. In [18] this problem is solved by writing the equation for the speed of the race car along the centerline and thus by using $(\frac{ds_c}{dt})$ as speed for the update process, $\Delta t(i) = \frac{\Delta s_c(i)}{s_c(i)}$ is then the time step for the update process. It is another way of approaching the problem.

Further $a_{lat}(i) = v_{abs}(i)^2 k(i)$, with i = 1,2,...,n-2 and $a_{lat}(n-1) = a_{lat}(n-2)$,

which is the lateral acceleration working on the race car on different moments along the race track. A distinction is made in the constraints between the longitudinal acceleration due to braking ($a_{brake}(i)$) and the longitudinal acceleration provided by the engine ($a_{acc}(i)$) of the race car. For $a_{brake}(i)$, 'i' is equal to the points where the race car brakes (negative acceleration) and for $a_{acc}(i)$, 'i' is equal to the points where the race car undergoes a positive acceleration. Next, k(i), with i = 1,2,...,n-2, is the curvature on different moments along the race track and calculated the same way as in equation 3.5.

To get a fast and reliable solution out of the optimal control problem, it is important that a good initial guess is given for the optimization variables. In this case, the centerline coordinates (x_c, y_c) are given as initial guess for the (x, y) coordinates of the OCP. For the small displacement Δs, the initial guess is given by the Δs_c of the centerline coordinates with $\Delta s_c(i) = \sqrt{(x_{c,i+1} - x_{c,i})^2 + (y_{c,i+1} - y_{c,i})^2}$, i = 1,2,...,n-1.

The non-linear OCP is solved by using the interior point optimizer, IPOPT.

3.4 Optimal control problem using a vehicle model

In this section, another optimal control problem (OCP) is introduced to calculate the optimal racing line. This time the Formula Student race car is modeled by using a five degree of freedom (DOF) bicycle model and a Dugoff-tyre model. In this case the controls are the torque T and the steering anlge δ. The state vectors X and dX are:

$$X = \begin{bmatrix} x \\ y \\ \phi \\ \phi_f \\ \phi_r \\ v_x \\ v_y \\ \dot{\phi} \\ \dot{\phi}_f \\ \dot{\phi}_r \end{bmatrix}, dX = \begin{bmatrix} v_x \\ v_y \\ \dot{\phi} \\ \dot{\phi}_f \\ \dot{\phi}_r \\ a_x \\ a_y \\ \ddot{\phi} \\ \ddot{\phi}_f \\ \ddot{\phi}_r \end{bmatrix} \tag{3.18}$$

with ϕ_f and ϕ_r the angular displacement of the front and the rear wheels and ϕ the yaw rate of the race car. The first or second dot above the displayed symbols in X and dX describe the first or second time derivative of that symbol. All of these states are expressed in the absolute axial system (x, y).

The reason why this other OCP problem is introduced, is because in this case the racing line is calculated for a very specific modeled vehicle with a certain tyre model which ensures that the racing line better reflects the reality. In the ideal case, with a perfect modeled vehicle, it should be possible to get the solution (T,δ) out of

the OCP problem and use that solution as an input for a controller that controls the torque T and steering angle δ of the race car to drive the fastest possible lap time over the race track. This last part lies within the ambition of Formula Electric Belgium (FEB) for the upcoming years. Their goal is to develop an autonomous driving Formula Student race car in the future.

3.4.1 Vehicle model

As discussed earlier, the used vehicle model is a five DOF bicycle model. One of the reasons for this is that it limits the variables and thus also the computational time to solve the OCP problem. A more complex vehicle model can be easily implemented in the code to generate the solution. The same is true for the Duggoff-tyre model, this is a simplified approach but can again be easily expanded to a more realistic tyre model such as a Pacejka Magic Formula tyre model.

"The bicycle model can be used to study the mass distribution front-rear where F_{nf} is the load on the front tyre and F_{nr} the load on the rear tyre. These normal forces are important as they are an input to the Dugoff-tyre model that calculates the tyre forces. These tyre forces are then needed in order to obtain the accelerations of dX through Newton's Second Law. Next to these normal forces, the Dugoff-tyre model also uses the slip angles α_f and α_r, the longitudinal slips s_f and s_r (together with the sign of the generated force) and some other parameters like μ and the tyre stiffness parameters C_x and C_y" [13]. These other parameters are given by FEB.

Axial system

Figure 3.13 shows the different axial systems used in the bicycle model for the elaboration of the OCP. The axial system (x, y) is the absolute axial system, (x', y') is the relative vehicle axial system and (x'', y'') is the relative tyre axial system.

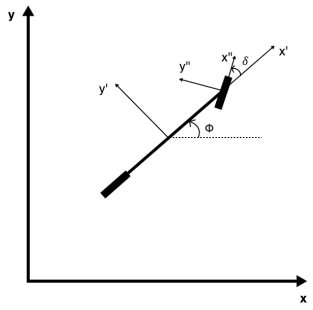

FIGURE 3.13: Different axial systems used

Normal forces

First the normal forces on the tyres need to be calculated. This is done by using figure 3.14 below:

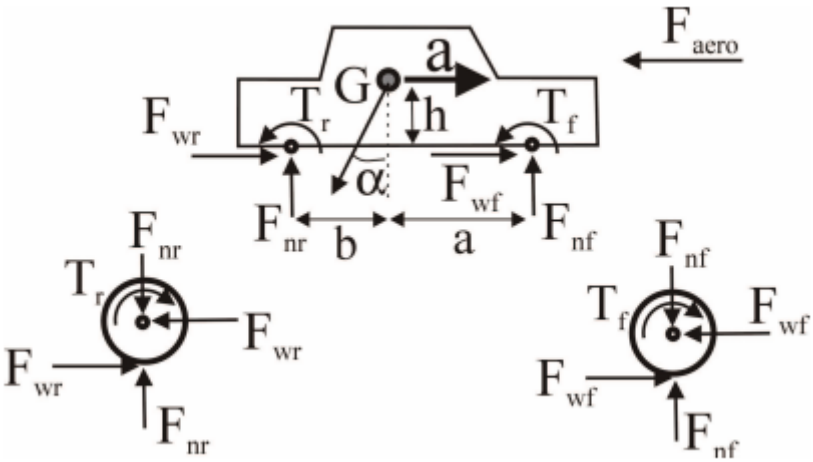

Figure 3.14: Free body diagram of the race car and its wheels [13].

The equations for the normal forces are:

$$F_{nf} = \frac{-T_r - T_f \cos(\delta) - h_{aero} F_{aero} - hma'_{i-1} - hmg \sin(\alpha) + bmg \cos(\alpha)}{a + b} \tag{3.19}$$

$$F_{nr} = mg \cos(\alpha) - F_{nf} \tag{3.20}$$

Here, T_f and T_r are the torques on the front and rear wheels, δ is the steering angle, F_{aero} is the aerodynamic force on the car, h_{aero} is the height of the aerodynamic center where the aerodynamic force F_{aero} engages, h is the height of the center of gravity (COG), m is the total weight of the car, a'_{i-1} is the longitudinal acceleration at the previous discrete position (i-1) expressed in the relative vehicle axial system, g is the gravitational constant, a and b are the distances between the front axle and the COG, and between the rear axle and the COG and α is the inclination angle of the road. The expression for F_{aero} is:

$$F_{aero} = \frac{v'^2_x \rho_{air} C_v A_{front}}{2} \tag{3.21}$$

Here v'_x is the longitudinal speed of the race car expressed in the relative vehicle axial system, ρ_{air} is the density of the air, A_{front} is the frontal surface area and C_v is the aerodynamic drag coefficient.

Longitudinal and lateral forces

The calculated normal forces now function as input for the calculation of the longitudinal (F_{longx}) and lateral (F_{laty}) tyre forces. This is done by using the Dugoff-tyre model in the Matlab function $tyre\_model\_Dugoff.m$.

The Dugoff tyre model is a physics inspired longitudinal and lateral force model that uses a friction circle to make a longitudinal and lateral force trade-off. [15]

The longitudinal and lateral forces are calculated using following equations [15][11]:

$$\begin{cases} F_{longx} = \frac{-C_x s}{(1-s)} f(\lambda) \\ F_{laty} = \frac{-C_y tan\alpha}{(1-s)} f(\lambda) \end{cases} \tag{3.22}$$

with

$$\lambda = \frac{\mu F_n (1-s)}{2\sqrt{((C_x s)^2 + (C_y tan\alpha)^2)}} \tag{3.23}$$

and

$$f(\lambda) = \begin{cases} (2-\lambda)\lambda & \lambda < 1 \\ 1 & \lambda \geq 1 \end{cases} \tag{3.24}$$

The longitudinal slips of the front (s_f) and rear wheel (s_r) are equal to:

$$\begin{aligned} s_f &= \frac{|\dot{\phi_f} r_w - v''_{x,f}|}{v''_{x,f}} \\ s_r &= \frac{|\dot{\phi_r} r_w - v''_{x,r}|}{v''_{x,r}} \end{aligned} \tag{3.25}$$

The slip angles of the front (α_f) and rear wheel (α_r) are equal to:

$$\begin{aligned} \alpha_f &= tan^{-1}(\frac{v''_{y,f}}{v''_{x,f}}) \\ \alpha_r &= tan^{-1}(\frac{v''_{y,r}}{v''_{x,r}}) \end{aligned} \tag{3.26}$$

Here, r_w is the radius of the wheels and μ is the friction coefficient between the road and the tyres of the race car.

State vector dX

The calculated longitudinal and lateral forces are after transformation to the absolute axial system, which is omitted here, used to calculate the different accelerations in the state vector dX. The equations are [13]:

$$a'_x = \frac{-F_{aero} + F_{longx,f}\cos(\delta) - F_{laty,f}\sin(\delta) + F_{longx,r} - mg\sin(\alpha)}{m} \tag{3.27}$$

$$a'_y = \frac{F_{longx,f}\sin(\delta) + F_{laty,f}\cos(\delta) + F_{laty,r}}{m} \tag{3.28}$$

$$\ddot{\phi} = \frac{-F_{laty,r}b + (F_{longx,f}\sin(\delta) + F_{laty,f}\cos(\delta))a}{I_{zz}} \tag{3.29}$$

$$\ddot{\phi}_f = \frac{T_f - F_{longx,f}r_w}{I_w} \tag{3.30}$$

$$\ddot{\phi}_r = \frac{T_r - F_{longx,r}r_w}{I_w} \tag{3.31}$$

with I_{zz} the moment of inertia of the vehicle and I_w the moment of inertia of the wheels.

3.4.2 Implementation

The implementation of this method is done roughly the same as in section 3.3.1.

The implementation is done in Matlab, using Opti [2] in CasADi [4] as optimization solver. Again, other commercial solvers could have been used.

First the centerline coordinates (x_c, y_c) and the widths $w(i)$ are generated with the previously discussed Matlab function *generate_track_22.m*. Here i = 1,2,...,n and n the total number of points of the optimal racing line.

During the update process, again a changing update step is used where also the independent variable *dt* is again changed in *ds*. This is discussed in more detail later in the text.

Next, the different vehicle parameters of the Formula Student race car are introduced.

Further, the (x_c, y_c) coordinates and widths $w(i)$ are linearly interpolated generating a denser group of centerline coordinate points and widths. This again by using the function *interparc.m* [10] out of the Matlab function library.

The next step is the creation of the CasADi optimization environment and the introduction of the different optimization variables and controls. In this case, the optimization variables are the states x, y, ϕ, ϕ_f, ϕ_r, v_x, v_y, $\dot{\phi}$, $\dot{\phi}_f$, $\dot{\phi}_r$ and the displacement step Δs. The optimization controls are the torque T and the steering angle δ.

The general OCP can be formulated as in section 3.3.1. In this case, the vector $f(\cdot)$ describes the vehicle dynamics of the Formula Student race car given by the

equations of motion described in section 3.4.1.

For this method, the discrete optimal control problem can be written as follows:

$$min \frac{\sum_{i=1}^{n-1} \Delta s(i)}{\sum_{i=1}^{n} \frac{v'_x(i)}{n}} \tag{3.32}$$

$$\begin{aligned}
s.t \quad & dX(i+1) - f(X(i), U(i)) && = 0, \quad i = 1, 2, ..., n-1 \\
& a_x(i)^2 + a_y(i)^2 - (\mu g)^2 && \leq 0, \quad i = 1, 2, ..., n-1 \\
& (x(i) - x_c(i))^2 + (y(i) - y_c(i))^2 - (\frac{1}{2} w(i))^2 && \leq 0, \quad i = 1, 2, ..., n \\
& x_{min} \leq x(i) && \leq x_{max}, \quad i = 1, 2, ..., n \\
& y_{min} \leq y(i) && \leq y_{max}, \quad i = 1, 2, ..., n \\
& 0 < v'_x(i) && \leq v_{max}, \quad i = 1, 2, ..., n-1 \\
& T_{min} \leq T(i) && \leq T_{max}, \quad i = 1, 2, ..., n \\
& \delta_{min} \leq \delta(i) && \leq \delta_{max}, \quad i = 1, 2, ..., n \\
& \Delta s_{min} \leq \Delta s(i) && \leq \Delta s_{max}, \quad i = 1, 2, ..., n-1 \\
& \begin{bmatrix} x(n) \\ y(n) \\ v_x(n) \\ v_y(n) \\ \phi(n) \\ \dot{\phi}_f(n) \\ \dot{\phi}_r(n) \end{bmatrix} - \begin{bmatrix} x(1) \\ y(1) \\ v_x(1) \\ v_y(1) \\ \phi(1) \\ \dot{\phi}_f(1) \\ \dot{\phi}_r(1) \end{bmatrix} && = 0
\end{aligned} \tag{3.33}$$

where U consists of the controls T and δ, v'_x is again the longitudinal speed of the race car expressed in the relative vehicle axial system (x', y') and equal to: $v'_x = v_x cos(\phi) + v_y sin(\phi)$.

The implicit time step used for the update process is equal to: $\Delta t(i) = \frac{\Delta s(i)}{\frac{v'_x(i) + v'_x(i+1)}{2}}$, with i = 1,2,...,n-1. This is the time needed for the race car to drive a distance equal to $\Delta s(i) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$, with an average speed equal to $\frac{v'_x(i) + v'_x(i+1)}{2}$, with i = 1,2,...,n-1 in both equations.

Again, it is important to give a good initial guess for the different variables in the OCP. The same initial guess is given for the (x, y) coordinates and the displacement step Δs as in section 3.3.1.

The non-linear OCP is also solved by using the interior point optimizer, IPOPT.

Chapter 4

Results and discussion

In this chapter the results of the different methods are discussed and compared with each other.

4.1 Euler spiral method

4.1.1 Results

This method, further referred to as the first method, is only applied on the 90 degree corner discussed in section 3.1. Because of the fact that this is a heuristic method and difficult to apply to different sorts of race tracks, this method is not further elaborated in more detail. Still, it gives interesting insight in how an optimal racing line should be constructed for certain types of corners.

The longitudinal velocity of the race car for the optimal racing line in figure 3.6 is plotted in figure 4.1. It is the longitudinal speed plotted from the turn-in point until the exit point. Because of the fact that only lateral acceleration is taken into account and that it is assumed that the race car drives the complete corner on its limits, the longitudinal speed is calculated according to following formula: $v(i) = \sqrt{\frac{\mu g}{k(i)}}$, with i = 1,2,...,n-2 and n equal to the total number of points where the Euler spiral consists of. Further, $v(i)$ and $k(i)$ are the longitudinal speed and curvature calculated on different moments along the Euler spiral. This formula follows directly from equation 3.3. Thus $v \sim \frac{1}{\sqrt{k}}$, which can also be seen in figure 4.1. The friction coefficient μ is equal to 1.4 and is predicted by another thesis at the KU Leuven [6] discussed in chapter 2.

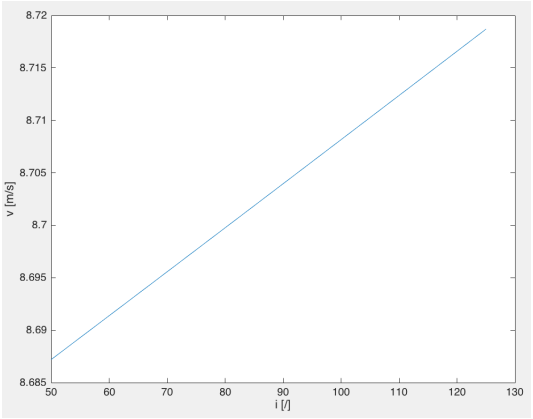

Figure 4.1: Longitudinal speed v [m/s] in function of the number of points i = 50,51,...,125.

4.1.2 Discussion

The main advantages and disadvantages of the Euler spiral method are listed and discussed in [24]. Some of them are repeated here.

The main advantages are:

- "Guarantee of near-optimum in cornering part" [24].
- "Natural achievement of an apex" [24].
- Small computational time for a simple race track. [24]

The main disadvantages are:

- Not easy applicable to irregular shapes of race tracks and for straight lines. [24]
- "The model is not guaranteed to be the best when more factors other than the lateral friction are considered" [24].
- The calculated optimal racing line is independent of the used race car.

4.2 Lateral acceleration focused non-linear optimization problem

4.2.1 Results

This method, further referred to as the second method, is applied on the race track of Formula Student Germany (FSG) [14]. The different values for the boundaries, the value of k_{max} and μ etc. are given in appendix B. The maximum curvature

$k_{max} = 0.1609$ [rad/m] is calculated, based on the maximum steering angle δ_{max} and the fact that Ackermann steering geometry is used. The calculation is given in appendix A. The friction coefficient μ is the same as in the previous method and equal to 1.4.

A solution for the optimal racing line is found after 1215 iterations. The total time needed for this calculation was less than 1 hour on a 2.6 GHz Intel Core i5 processor. The minimum value of the objective function is 25.47 [s]. The total number of variables is 1200, the total number of equality constraints is 2 and the total number of inequality constraints is 1198.

Figure 4.2 shows the calculated optimal racing line for the FSG race track. The black lines are the boundaries of the race track, the green line is the centerline and the red line is the calculated optimal racing line.

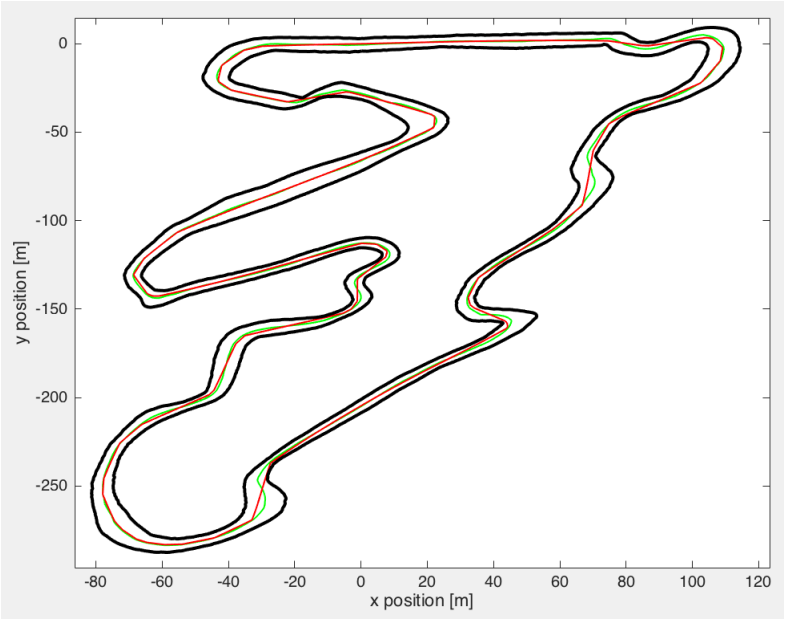

FIGURE 4.2: Red line is the calculated optimal racing line, green line is the center line and the black lines are the boundaries of the FSG race track.

Figure 4.3 and figure 4.4 display two zoomed in segments of the FSG race track. The color of the lines has the same meaning as in figure 4.2.

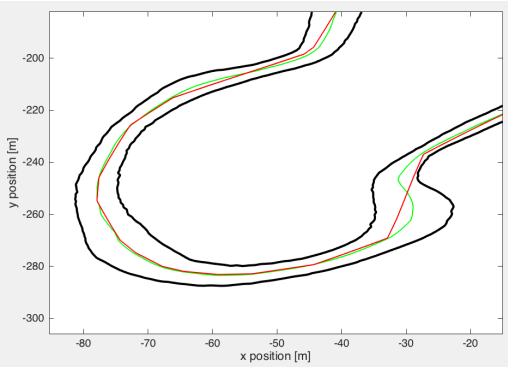

Figure 4.3: Race track segment 1

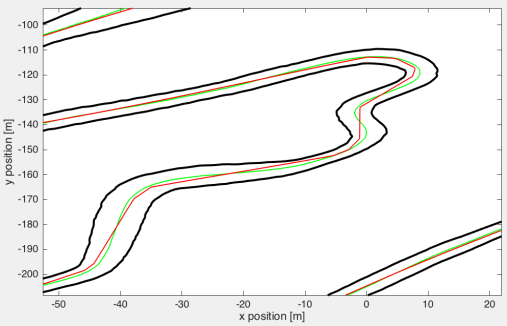

Figure 4.4: Race track segment 2

Figure 4.5 shows the calculated longitudinal velocity, using figure 3.12, for the optimal racing line in function of the number of points i = 1,2,...,n and n = 600. The maximum longitudinal acceleration is approximated to be 0.95g. This is done by taking into account the fact that on average the Formula Student race car accelerates from standstill 75 meters in approximately 4 seconds [3], assuming a constant longitudinal acceleration. The minimum longitudinal acceleration due to braking is approximated to be -1.2g. This is done by examining GoPro data of the previous FEB race car, Umicore Nova, driving the autocross event in Formula Student Czech Republic [12]. The maximum longitudinal speed is assumed to be 30 [m/s]. This is a good assumption because the real top speed on maximum power (120 [kW]) of Umicore Nova is 33 [m/s]. Within the competition of Formula Student, the maximum power is limited to 80 [kW]. Figure 4.6 shows the braking zones (red lines), the accelerating zones (green lines) and the constant speed zones (blue lines) of the calculated optimal racing line for the FSG race track.

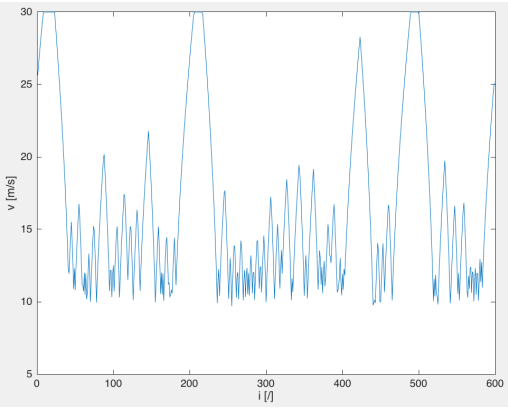

Figure 4.5: Longitudinal speed v [m/s] in function of the number of points i = 1,2,...,n.

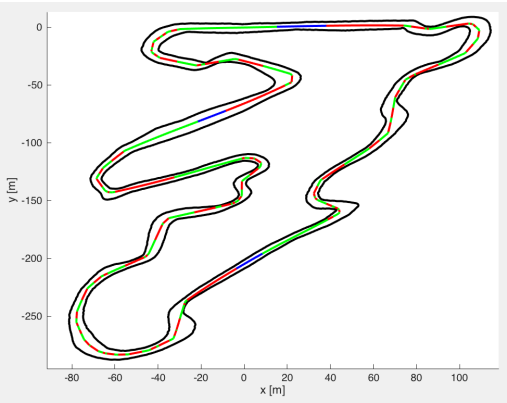

Figure 4.6: Optimal racing line for the FSG race track with braking zones (red lines), accelerating zones (green lines) and constant speed zones (blue lines).

The total time needed for the race car to accomplish one lap is: $t_{tot} = 70.5119$ [s]. This time can be used as a rough indication but certainly not for the comparison of the different methods. The way the speed is calculated is not very accurate because it does not take into account the limits on the total net acceleration which is equal to: $a_{net} = \sqrt{a_{long}^2 + a_{lat}^2}$.

The average curvature $k_{avg} = \frac{\sum_{i=1}^{n-1} k(i)}{n-1}$, is equal to: $k_{avg} = 0.0219$ [rad/m].

4.2.2 Discussion

The results with regard to the shape of the calculated optimal racing line are as expected. Minimizing the curvatures means cutting corners and trying to get a lot of straight lines. Because of the fact that this method is a geometrical method, it does not take the dynamics of the race car into account while optimizing. This results in some unrealistic parts of the calculated optimal racing line. Only the maximum curvature k_{max} is implemented in the constraints, but nothing is said about the difference in curvature over time. This means that for every possible speed of the race car, k_{max} can be achieved. Of course, this is not true. Figure 4.4 shows some of these abrupt changes in curvature.

Advantages and disadvantages

The main advantages are:

- Small computational time.
- Reliable in a way that the solution minimizes a function that consists of the curvature and the traveled distance.
- Easy applicable for different race tracks.

The main disadvantages are:

- Optimal racing line independent of the dynamics.
- Optimal racing line independent of the race car characteristics.
- Speed calculated without taking limits on the total net acceleration a_{net} into account.

4.3 Optimal control problem for a point mass with lateral and longitudinal accelerations

4.3.1 Results

This method, further referred to as the third method, is again applied on the race track of FSG but this time for two different situations. The first situation is for a

high accelerating race car, the second situation is for a low accelerating race car. The different values for the boundaries, k_{max}, μ, the total number of points n and the mass m etc. are given in appendix B. Here is $k_{max} = 0.1609$ [rad/m] and $\mu = 1.4$ and thus the same as in previous method.

A solution for the optimal racing lines is found after 3000 iterations due to a maximum iteration cut off. The total time needed for these calculations was approximately 11 hours and 15 minutes, again on a 2.6 GHz Intel Core i5 processor. The minimum value of the objective function for the high accelerating race car is 49.415 [s], for the low accelerating race car it is 78.982 [s]. The total number of variables is 4197, the total number of equality constraints is 3000 and the total number of inequality constraints is 5393.

Figure 4.7 shows the optimal racing lines for the FSG race track for the high and low accelerating race car. The black lines are the boundaries of the race track, the green line is the centerline, the red line is the calculated optimal racing line for the high accelerating race car and the blue line is for the low accelerating race car.

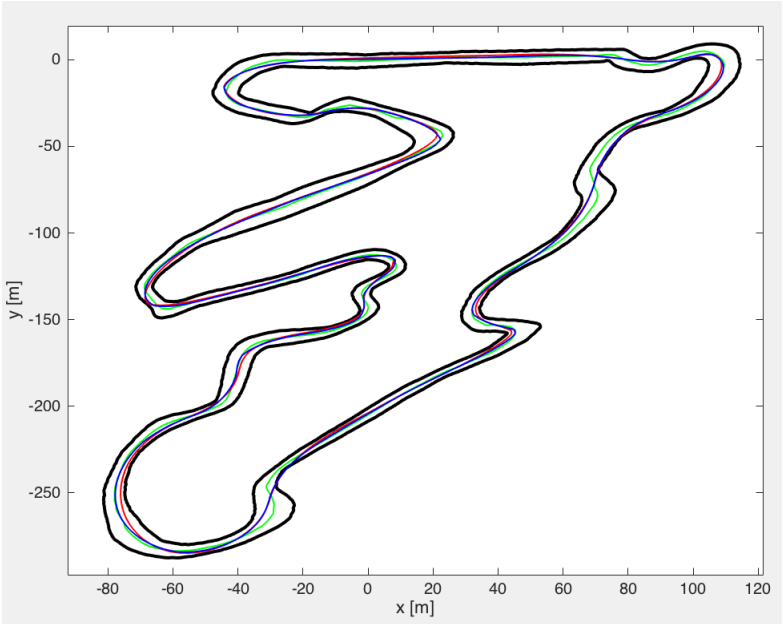

Figure 4.7: Red line is the calculated optimal racing line for the high accelerating race car, blue line is the optimal racing line for the low accelerating race car, green line is the center line and the black lines are the boundaries of the FSG race track.

Figure 4.8 and figure 4.9 display again the two zoomed in segments of the FSG race track. The color of the lines has the same meaning as in figure 4.7.

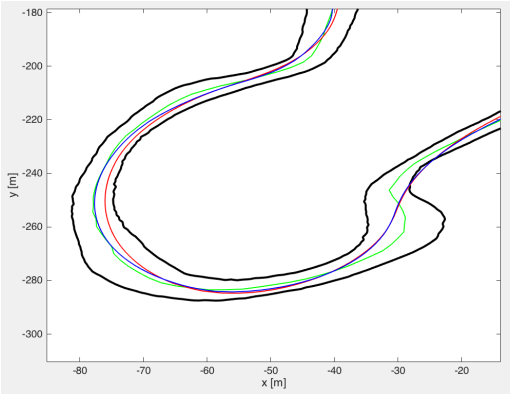

FIGURE 4.8: Race track segment 1

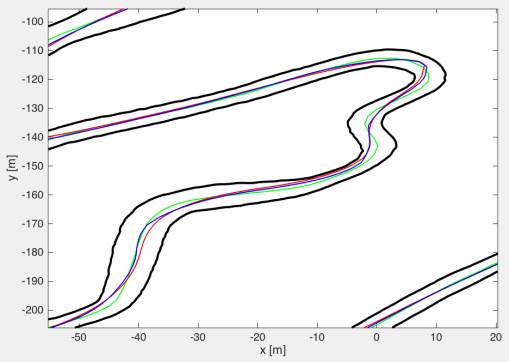

FIGURE 4.9: Race track segment 2

Figure 4.10 shows the velocity for the optimal racing line of the high (red line) and low accelerating (blue line) race car in function of the number of points i = 1,2,...,n and n = 600. The maximum longitudinal acceleration of the high accelerating race car is approximated to be 0.95g (as in the previous method), and 0.3g for the low accelerating race car. The minimum longitudinal acceleration (due to braking) is -1.2g (as in the previous method) for the high accelerating race car and -0.8g for the low accelerating race car. Figure 4.11 and figure 4.12 show the braking zones (red lines), the accelerating zones (green lines) and the constant speed zones (blue lines) for the high and low accelerating race cars on the FSG race track.

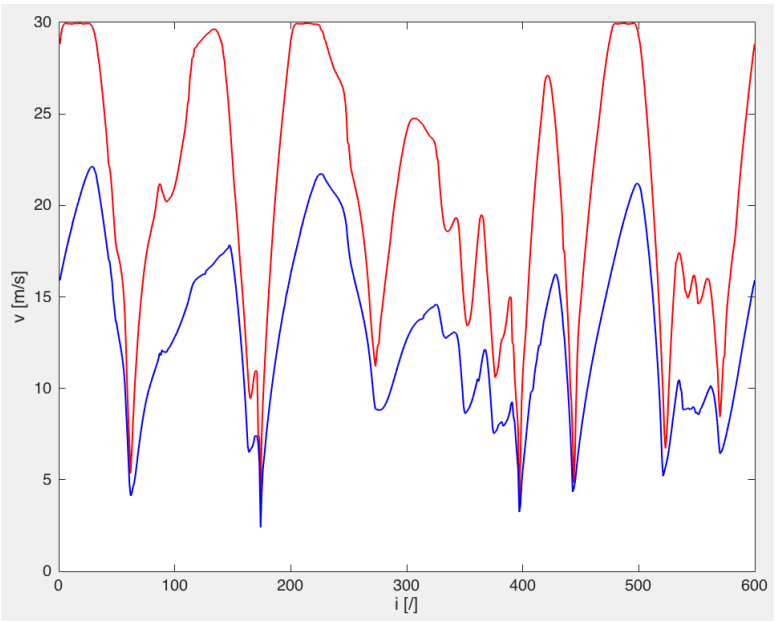

FIGURE 4.10: Longitudinal speed v [m/s] of the high (red line) and low (blue line) accelerating race car in function of the number of points i = 1,2,...,n.

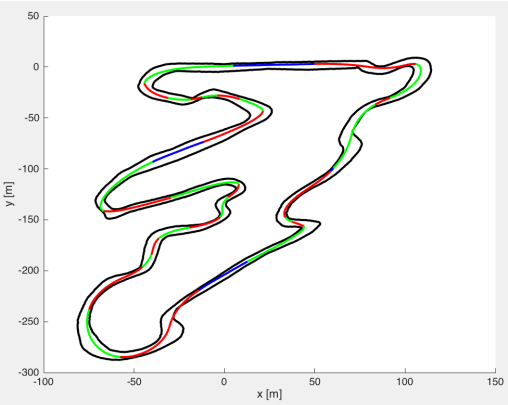

Figure 4.11: Optimal racing line for the high accelerating race car on the FSG race track with braking zones (red lines), accelerating zones (green lines) and constant speed zones (blue lines).

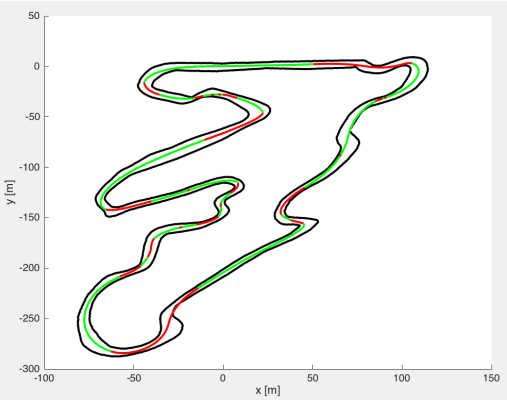

Figure 4.12: Optimal racing line for the low accelerating race car on the FSG race track with braking zones (red lines), accelerating zones (green lines) and constant speed zones (blue lines).

The total time needed for the high accelerating race car to accomplish one lap is: $t_{tot,high} = 58.2276$ [s]. For the low accelerating race car the total time is: $t_{tot,low} = 92.2871$ [s].

The average curvature $k_{avg} = \frac{\sum_{i=1}^{n-1} k(i)}{n-1}$ for the high accelerating race car is equal to: $k_{avg,high} = 0.0140$ [rad/m]. For the low accelerating race car this is equal to: $k_{avg,low} = 0.0153$ [rad/m].

4.3.2 Discussion

Low versus high accelerating race car

Most of the results for the optimal racing line of the low and high accelerating race car are as expected. As explained in section 3.1, a lower powered race car will most of the time aim for an earlier apex than a higher powered race car. In this way, the lower powered race car generates higher corner speeds, but also needs to wait longer to accelerate out of the corner again in comparison with the optimal racing line the higher powered race car takes. This phenomenon can be observed by looking at figure 4.8 and figure 4.13. The first positive peak (i = 272) for the lateral acceleration of the low accelerating race car (blue line), happens before the lateral acceleration peak (i = 275) of the high accelerating car (red line). This is the moment the race car 'steers in'. After this peak, the race car starts to accelerate out of the corner, thus generating higher longitudinal accelerations, and the longitudinal acceleration of the high accelerating race car (black line) shows that this happens earlier than that for the low accelerating race car (green line). Zooming in at figure 4.8 at coordinate (-50,-280) shows this difference in turn-in point too, although the

difference is small. Figure 4.8 also shows the wider line the lower accelerating race car takes, this probably to retain more speed through the corner but still the expectation was to see higher lateral accelerations in figure 4.13 because the friction coefficient μ and the mass M_{total} are for both cars the same.

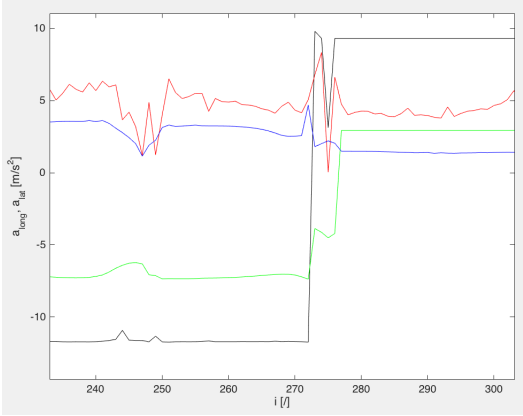

FIGURE 4.13: Longitudinal a_{long} and lateral $|a_{lat}|$ acceleration $[m/s^2]$ of the high (black and red) and low (green and blue) accelerating race car in function of the number of points i = 235,236,...,305. (segment 1)

Method 3 versus method 2

As discussed earlier in section 4.2.2 and what can also be seen when comparing figure 4.3 and figure 4.8 with each other, is that method 2 generates an optimal racing line with some abrupt changes in curvature. This is because method 2 does not take the dynamics of the race car into consideration while generating the optimal racing line in contrast to method 3. Although method 2 generates the optimal racing line very fast ($<$ 1hour), method 3 matches the different objectives better. It calculates the optimal racing line also within 24 hours and generates a more reliable optimal racing line than method 2.

Advantages and disadvantages

The main advantages are:

- A reliable solution within 24 hours.
- The limits on the total net acceleration a_{net} are taken into account while generating the solution.
- Easy applicable for different race tracks.

The main disadvantages are:

- Optimal racing line independent of an eventual vehicle model of the race car.
- Applied accelerations are also the real accelerations, without taking slip through a tyre model into account.

These advantages and disadvantages are further explained in the discussion of the fourth method.

4.4 Optimal control problem using a vehicle model

4.4.1 Results

This method, further referred to as the fourth method, is again applied on the race track of FSG. The different values for the boundaries, k_{max}, μ, the total number of points n and the mass m etc. are given in appendix B. Here $k_{max} = 0.1609$ [rad/m] and $\mu = 1.4$ and thus again the same as in the previous method.

A solution for the optimal racing line is found after 2000 iterations due to a maximum iteration cut off. The total time needed for these calculations was approximately 25 hours, again on a 2.6 GHz Intel Core i5 processor. The minimum value of the objective function is 55.230 [s]. The total number of variables is 7797, the total number of equality constraints is 5999 and the total number of inequality constraints is 4795.

Figure 4.14 shows the optimal racing line for the FSG race track. The colors of the lines have the same meaning as in figure 4.2.

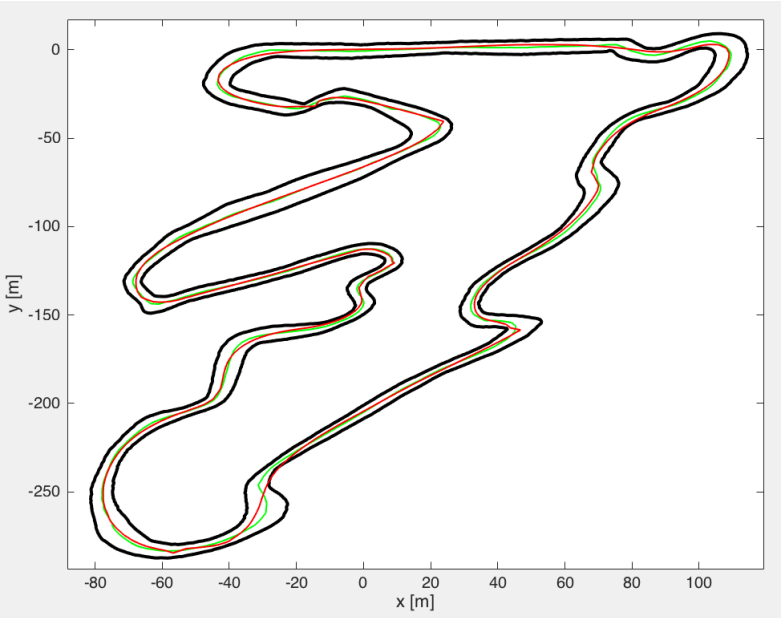

FIGURE 4.14: Red line is the calculated optimal racing line, green line is the center line and the black lines are the boundaries of the FSG race track.

Figure 4.15 and figure 4.16 display again the two zoomed in segments of the FSG race track. The color of the lines has the same meaning as in figure 4.14.

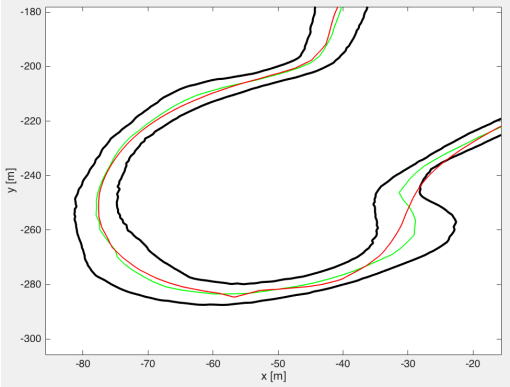

FIGURE 4.15: Race track segment 1

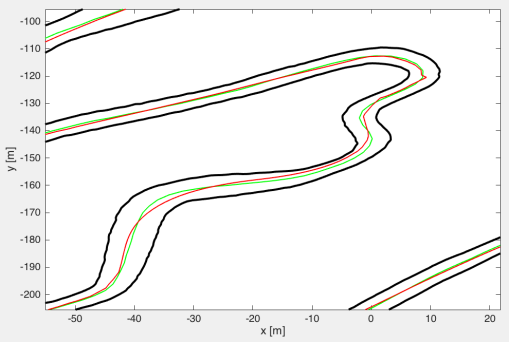

FIGURE 4.16: Race track segment 2

Figure 4.17 shows the longitudinal velocity v'_x for the optimal racing line in function of the number of points i = 1,2,...,n and n = 600. The maximal torque is based on the data sheet of the used electric motors in appendix C. What can be seen, is that the maximum torque at the output of the electric motor is approximately 21 [Nm]. With a gear ratio of 15.12, and 4 electric motors (each wheel has one electric motor), this makes a total maximum torque of $T_{max,real} = 1270.1$ [Nm]. Within the simulation, the maximum torque is expressed per axle and furthermore divided by

two to limit the tyre saturation. The bicycle model is implemented in such a way that the torque on the front wheel (T_f) is the same as the torque on the rear wheel (T_r). Thus the implemented and used maximum torque in the simulation is equal to: $T_{max} = 317.52$ [Nm] per axle. The minimum torque is based on the minimum longitudinal acceleration of the previous method (-1.2g). The minimum torque per axle is then calculated as: $T_{min} = \frac{-1.2gM_{total}r_w}{2} = -406.58$ [Nm]. Figure 4.18 shows the braking zones (red lines), the accelerating zones (green lines) and the constant speed zones (blue lines) of the calculated optimal racing line for the FSG race track.

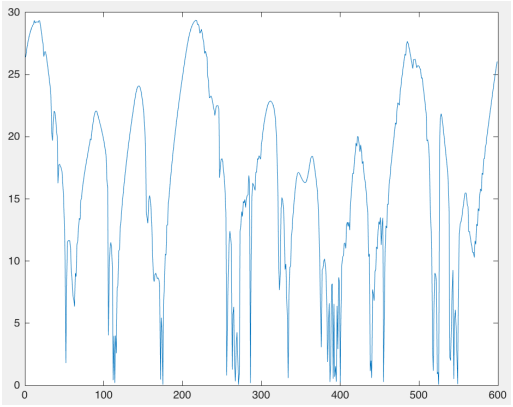

Figure 4.17: Longitudinal speed v'_x [m/s] in function of the number of points i = 1,2,...,n.

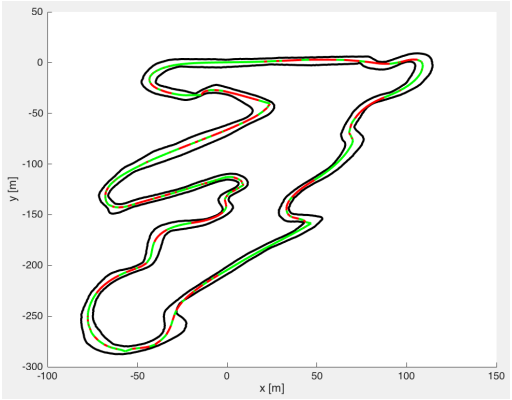

Figure 4.18: Optimal racing line for the FSG race track with braking zones (red lines), accelerating zones (green lines) and constant speed zones (blue lines).

4.4.2 Discussion

The calculation of the optimal racing line has not fully converged yet. This can be clearly seen in figure 4.15 and figure 4.16 where there are some abrupt changes in curvature which lead to the longitudinal speed graph in figure 4.17. This calculation includes all the values for the state vectors X and dX, but also the values for the controls δ and T, on different moments along the track. Thus comparing the current results of this method with the other methods is difficult. The calculation has not fully converged yet because of two reasons. The first reason is because the time steps for the update of the state X are getting too small in the slow parts of the race track ($\Delta t(i) = \frac{\Delta s(i)}{\frac{v'_x(i)+v'_x(i+1)}{2}}$, with i = 1,2,...,n-1). This can be solved by implementing a constraint on v'_x for a higher minimal speed or by using more discrete points (Δs gets smaller). This latter increases of course the computational time. The second reason is because a higher number of iterations were needed to converge to an acceptable solution.

Although the results for method 4 are not converged yet, the results of this method, after a long enough computational time and with the right update step, will represent

a more realistic behavior of the race car than the other discussed methods [18]. Thus it seems that the computational time of this method is the biggest problem.

Method 4 versus method 2

The fourth method takes a lot of the dynamics of the race car into consideration. This means that the abrupt changes in curvature, that are observed at the results for method 2, are not the case here. Still, some of these abrupt changes are shown in figure 4.14 but the reason for this is that the solution is not converged yet. Method 4 has the most possibilities in terms of what FEB is looking for in the future, namely switching to the competition of autonomous driving race cars. For now, because one of the biggest constraints is the maximal computational time, method 2 is a good alternative to assist the racing driver in the race.

Method 4 versus method 3

Both method 3 and method 4 do take some of the dynamics of the race car into consideration. The main difference is that method 3 does not take slip into account, which makes that the applied acceleration is also the real acceleration working on that race car, which is assumed to be a point mass. In method 4, the forces on the tyres, which are represented by a tyre model, lead to the accelerations on the COG of the race car. In this case the longitudinal slips s and slip angles α cause the race car to generate longitudinal and lateral accelerations on the COG by means of the applied torque and steering angles on different moments along the race track. This latter leads to more realistic values for the accelerations on the race car. Figure 4.19 shows the absolute value of the lateral accelerations for segment 1 of the FSG race track for method 3 (blue) and method 4 (red). The black rectangle shows the turn-in point for the corner of segment 1, the arrows show the increase in lateral acceleration through the corner. Although method 4 is not fully converged yet, it shows that the lateral accelerations in the corners are much higher and more realistic than the ones for method 3.

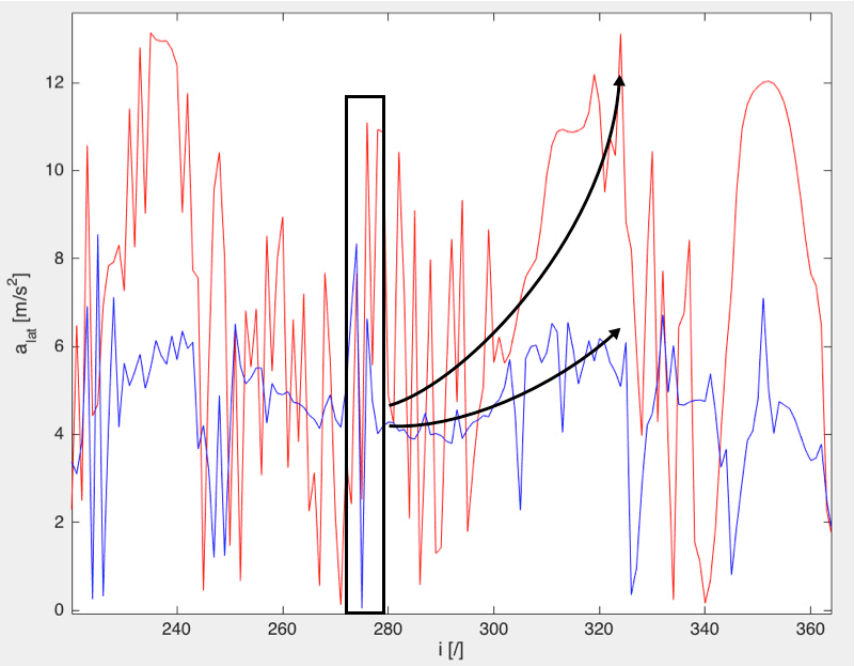

Figure 4.19: Absolute value of the lateral acceleration a_{lat} for method 3 (blue) and method 4 (red) for segment 1 of the FSG race track.

Method 4 generates a very reliable and race car specific optimal racing line but at this moment, the computational time is too long to be used in the competitions of Formula Student. Therefore, method 3 is the best alternative of the four discussed methods at this moment. Method 3 matches all the different objectives except one, namely the need for a method where the implemented vehicle model is expandable. This last objective is especially important when FEB wants to join the autonomous racing competition. As an assistance tool for the driver, the calculated x,y coordinates of the optimal racing line from method 3 will certainly help the driver to drive on the right 'line'. The calculated velocities and accelerations are not very realistic, as discussed earlier. As discussed in chapter 1, the realistic values for these velocities and accelerations can be calculated by using methods such as the quasi-steady-state method. The combination of these two methods is an alternative for method 4 with the important difference that the x,y coordinates of the optimal racing line are in that case not calculated for a specific vehicle model of the race car.

Advantages and disadvantages

The main advantages are:

- Optimal racing line calculated for a specific vehicle model of the race car.
- Implemented vehicle model is easy expandable.
- Realistic accelerations on the COG of the race car due to the consideration of slip through a tyre model.
- Torque (T) and steering angle (δ) directly calculated on different moments along the race track.
- Easy applicable for different race tracks.

The main disadvantages are:

- High computational time (>24hours).

4.5 Recap

The first method is a fast heuristic method but not easy to repeat for different race tracks, it is reliable in a way that the solution is suboptimal and it does not take the dynamics of the race car into account. It is thus not expandable in terms of vehicle model complexity. The second method its computational time is low (<1hour), it is easy applicable for different race tracks, the calculation outcome is reliable but less than the third and the fourth method because it again does not take the dynamics of the race car into account. It is thus also not expandable in terms of vehicle model complexity. The third method its computational time is higher than the previous two methods but still smaller than the constraint of 24 hours. The calculation is more reliable than in the first and second method because in this case both the longitudinal and lateral accelerations are taken into account when calculating the solution. A difference can also be made in terms of slower and faster accelerating cars, it is easy applicable for different race tracks but again for the calculations, a vehicle model is not used. The fourth method its computational time is too long on this moment (>24h). The calculation is the most reliable because it uses a vehicle model and tyre model to reflect the reality better than the previously discussed methods. The implemented vehicle model can be easily expanded and the method can also be applied for different race tracks.

Chapter 5

Conclusion and future work

The goal of this thesis is to propose a method to FEB that calculates the optimal racing line that meets their given requirements as well as possible. Four methods were discussed with each of them their advantages and disadvantages relative to the given objectives.

For now, FEB wants to obtain the optimal racing line to function as an assistance system for the racing driver to help him or her to set the best possible lap time for a given Formula Student race track. Although method 3 does not generate the most realistic accelerations and speeds on different moments, it does give the best first understanding, given the four discussed methods and the different objectives, of which path the racing driver should follow to set the best lap time.

In the future, FEB wants to join the autonomous racing competition, Formula Student driverless. Out of the four discussed methods, the fourth method gives the best alternative to help FEB achieve this goal. It directly calculates the steering angles and torque inputs for a specific Formula student race car on different moments along a given race track, in order to drive this race car as fast as possible around the given race track. These calculated controls can then function as input for a controller that makes the Formula student race car driverless.

Regarding future work, it is important that the different information sources discussed in chapter 2 are successfully combined to give the racing driver the real time feedback he or she needs. Finally, there should also be thought about how the real time data should be sent to the racing driver and how the visualization of the optimal racing line should be done.

Appendices

Appendix A

Maximum curvature calculation

The maximum possible curvature is calculated using figure A.1 where Ackermann steering geometry is used. The values for δ_{max}, a and b are given in appendix B.

$$R_1 = \frac{a+b}{tan(\delta_{max})} \tag{A.1}$$

$$R = \sqrt{b^2 + R_1^2} \tag{A.2}$$

$$k_{max} = \frac{1}{R} \tag{A.3}$$

Filling in all the values gives $k_{max} = 0.1609$ [rad/m].

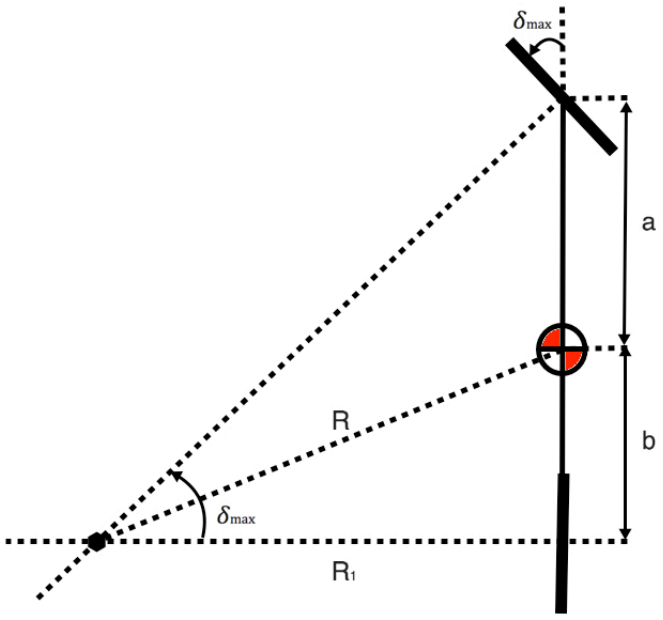

FIGURE A.1: Calculation of the maximum curvature k_{max}

Appendix B

Vehicle parameters and optimization values

Symbol	Description	Value
M_{car}	Mass of the race car	205 kg
M_{driver}	Mass of the driver	75 kg
M_{total}	Total mass of the race car + driver	280 kg
I_{zz}	Moment of inertia of the race car	48.8298 kgm^2
L	Length of the race car	1.54 m
h	height of the center of gravity	0.25342 m
a	Distance between front axle and COG	0.84 m
b	Distance between rear axle and COG	0.7 m
r_w	Wheel radius	0.2467 m
I_w	Moment of inertia of the wheel	0.2650 kgm^2
Cx_f	Longitudinal tyre-stiffness front wheels	44508 N/m
Cx_r	Longitudinal tyre-stiffness rear wheels	44508 N/m
Cy_f	Lateral tyre-stiffness front wheels	23000 N/rad
Cy_r	Lateral tyre-stiffness rear wheels	23000 N/rad
A_{front}	Frontal surface area	1 m^2
h_{aero}	Height of the aerodynamic center	0.3 m
ρ_{air}	Density of the air	1.225 kg/m^3
μ	Friction coefficient between wheel-road	1.4

TABLE B.1: Parameters for the Formula Student race car 'Umicore Pulse' from Formula Electric Belgium

Symbol	Value
n	600
k_{max}	0.1609 rad/m
μ	1.4
g	$9.81 m/s^2$

Table B.2: Optimization values for the results of method 2

Symbol	Value
n	600
k_{max}	0.1609 rad/m
μ	1.4
g	9.81 m/s^2
M_{total}	280 kg
v_{max}	30 m/s
x_{min}	-100 m
x_{max}	150 m
y_{min}	-300 m
y_{max}	50 m
Δs_{min}	0.2 m
Δs_{max}	3.5 m
$a_{brake,max,h}$	-1.2g m/s^2
$a_{acc,max,h}$	0.95g m/s^2
$a_{brake,max,l}$	-0.8g m/s^2
$a_{acc,max,l}$	0.3g m/s^2

Table B.3: Optimization values for the results of method 3

Here, $a_{brake,max,h}$ and $a_{acc,max,h}$ are the constraints on the accelerations for the high accelerating race car and $a_{brake,max,l}$, $a_{acc,max,l}$ for the low accelerating race car.

Symbol	**Value**
n	600
k_{max}	0.1609 rad/m
μ	1.4
g	9.81 m/s^2
M_{total}	280 kg
v_{max}	30 m/s
x_{min}	-100 m
x_{max}	150 m
y_{min}	-300 m
y_{max}	50 m
Δs_{min}	0.2 m
Δs_{max}	3.5 m
T_{min}	-406.5813 Nm
T_{max}	317.52 Nm
δ_{min}	$\frac{-14\pi}{180}$ rad
δ_{max}	$\frac{14\pi}{180}$ rad

TABLE B.4: Optimization values for the results of method 4

Appendix C

Data sheet electric motor

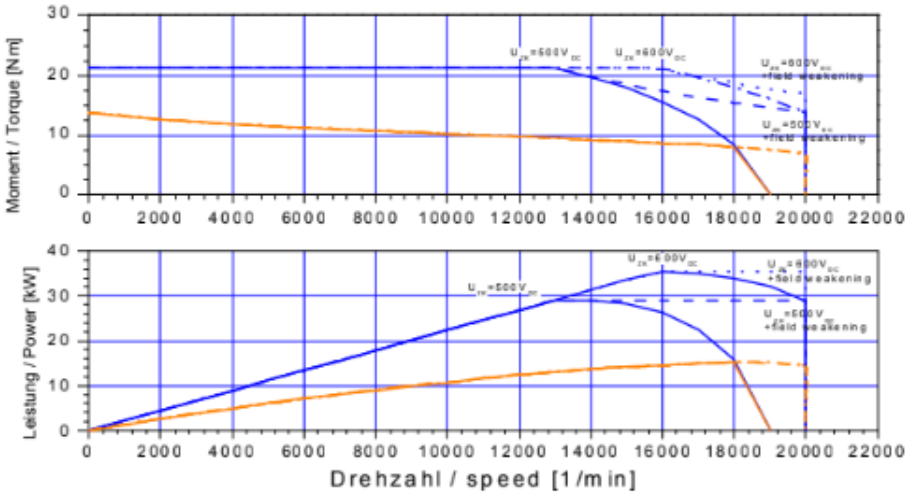

FIGURE C.1: Data sheet for the electric motors of the Formula Student race car 'Umicore Pulse'. [1]

Bibliography

[1] AMK Antriebs- und Steuerungstechnik GmbH Co. KG. Motor data sheet:DD5-14-10-POW - 18600-B5 Formula Student, March 2017.

[2] Gillis J. (2018). Effortless NLP modeling with CasADi's Opti stack. Benelux Meeting on Systems and Control. Soesterberg, Netherlands, 27-29 March 2018.

[3] Disciplines. URL:https://www.formulastudent.de/about/disciplines/. Formula Student Germany.

[4] J. A. E. Andersson, J. Gillis, G. Horn, J. B Rawlings, and M. Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, In Press, 2018.

[5] M. Abramowitz and S. A. Irene, editors. *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, pages 300 – 301. U.S. Department of Commerce; National Bureau of Standards, December 1972.

[6] S. Arend and D. Janssens. Development of a coupled state and parameter estimator for torque vectoring in a formula student electric race car. KU Leuven, 2018.

[7] R. Bentley. *Ultimate Speed Secrets: The Complete Guide to High-Performance and Race Driving*. Motorbooks, Augustus 2011. ISBN-13: 978-0760340509.

[8] A. Brouillard. *The Perfect Corner: A Driver's Step-By-Step Guide to Finding Their Own Optimal Line Through the Physics of Racing*, volume 1 of *The Science of Speed*. Paradigm Shift Driver Development, March 2016. ISBN-13: 978-0997382426.

[9] J. Bruin and T. Van Hemelen. Simultaneous localization and track mapping for a formula student vehicle. KU Leuven, 2018.

[10] J. D'Errico. interparc. URL: https://nl.mathworks.com/matlabcentral/fileexchange/34874-interparc, August 2012.

[11] H. Dugoff, S. P. Fancher, and L. Segel. Tyre performance charecteristics affecting vehicle response to steering and braking control inputs. *Final Report, Contract CST-460*, 1969.

[12] FEB. Fscz 2017 - autocross. URL: https://www.youtube.com/watch?v=05MJrqQOuQ0. Youtube.

[13] P. Gepts and G. Bodart. Final assignment: Vehicle dynamics an 8dof vehicle model (matlab) in comparison with a multibody model (virtual lab). January 2017.

[14] R. Kötke. Track layout. URL:https://www.formulastudent.de/pr/news/details/article/track-layout/, March 2006.

[15] F. Naets. Lecture 3: Tyres. October 2016.

[16] D. Patrick Kelly. *Lap Time Simulation with Transient Vehicle and Tyre Dynamics*. PhD thesis, Cranfield University School of Engineering, May 2008.

[17] A. Pawar. Euler spiral. URL: https://nl.mathworks.com/matlabcentral/fileexchange/24922-euler-spiral, May 2016.

[18] G. Perantoni and D. J. Limebeer. Optimal control for a formula one car with variable parameters. *Vehicle System Dynamics*, 52(5):653–678, 2014. URL: https://doi.org/10.1080/00423114.2014.889315.

[19] H. Scherenberg. Mercedes-benz racing design and cars experience, 1958. SAE Technical Paper 580042.

[20] J. P. Timings and D. J. Cole. Minimum maneuver time calculation using convex optimization. *Journal of Dynamic Systems, Measurement and Control*, 135, 2013.

[21] Wikipedia. Curvature. URL: https://en.wikipedia.org/wiki/Curvature, last checked on 28-05-2018.

[22] Wikipedia. Euler spiral. URL: https://en.wikipedia.org/wiki/Euler_spiral, last checked on 28-05-2018.

[23] Wikipedia. Fresnel integral. URL: https://en.wikipedia.org/wiki/Fresnel_integral, last checked on 28-05-2018.

[24] Y. Xiong. Racing line optimization. Master's thesis, Massachusetts Institute of Technology, September 2010.