

Torbjørn Smith

Lap time simulation: The use of Optimal Control to find the on-limit behaviour of a Formula Student racecar

Master's thesis in Industrial Cybernetics

Supervisor: Sebastien Gros

June 2021

Torbjørn Smith

Lap time simulation: The use of Optimal Control to find the on-limit behaviour of a Formula Student racecar

Master's thesis in Industrial Cybernetics
Supervisor: Sebastien Gros
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics



Norwegian University of
Science and Technology

Abstract

Automotive design and engineering is a complex endeavor. Especially in racing. Engineers need to deploy a wide range of software tools to build the fastest racecar possible. One such tool is the lap time simulator. The lap time simulator allows a holistic simulation of a vehicle design, meaning the entire vehicle is simulated driving around a track, producing a simulated lap time. The goal is to simulate the vehicle and find the limit of performance, producing the minimum lap time the vehicle is capable of around a representative track.

In this thesis, a lap time simulator is implemented for use at the Formula Student team Revolve NTNU. A free trajectory, minimum time maneuvering problem is defined as an optimal control problem, that is transcribed into a nonlinear programming problem using a direct orthogonal collocation method. The transcription is done in MATLAB using the open-source software framework CasADi, and the resulting NLP is solved using the interior point method with the solver Ipopt.

A planar, double-track vehicle model is deployed for proper tire slip modeling, utilized in a nonlinear tire model. The vehicle model also has load transfer, aerodynamics, and the powertrain modeled. The track model is generated from interpolating GNSS coordinates recorded at Formula Student competitions, creating a curvilinear abscissa. An optimal control scheme is deployed to generate a smooth abscissa curvature for the minimum time maneuvering problem. The results of the simulator are presented and compared to telemetry data, a different lap time simulator using the quasi steady state methodology, and two vehicle designs are compared.

The simulator was shown to be an accurate representation of the real vehicle, and at a higher computational cost, it yields a more accurate description of the vehicle dynamics of a real vehicle than the QSS simulator.

Sammendrag

Bildesign er en kompleks jobb, spesielt innen racing. Ingeniører bruker et bredt spekter av programvare for å bygge en racerbil som er så rask som mulig. Et slikt verktøy er rundetidssimulatoren. Rundetidssimulatoren gjør det ulig å simulere et helhetlig kjøretøydesign, noe som betyr at man simulerer at kjøretøyet kjører gjennom en bane og produserer en simulert rundetid. Målet er å simulere kjøretøyet, finne ytterpunktet av ytelsen, og produsere den minimale rundetiden kjøretøyet kan ha rundt en representativ bane.

I denne masteroppgaven blir en rundetidssimulator implementert for bruk hos Formula Student-laget Revolve NTNU. En fri bane, minimalt manøvreringsproblem blir definert som et optimal control problem, som skrives om til et ikke-lineært optimaliseringsproblem ved hjelp av en direkte ortogonal kollokasjonsmetode. Dette gjøres i MATLAB ved hjelp av rammeverket CasADi, og det resulterende ikke-lineære optimaliseringsproblemet løses ved hjelp av interior point-metoden med solveren Ipopt.

En plan, dobbeltsporet kjøretøymodell blir brukt for å oppnå god dekkmodellering, og det blir brukt sammen med en ikke-lineær dekkmodell. Kjøretøymodellen har også lastoverføring, aerodynamikk og drivlinjen modellert. Modellen av banen er generert ved å interpolere GNSS-koordinater som er målt under kjøring i Formula Student-konkurranser. Et optimal control problem blir brukt for å generere en jevn banekurvatur for simulatoren. Resultatene av simulatoren blir presentert og sammenlignet med telemetridata, og en annen rundetidssimulator som bygger på en quasi steady state metode. To kjøretøydesign blir også sammenlignet.

Simulatoren ble vist til å være en god representasjon av det virkelige kjøretøyet, og med en høyere beregningskostnad enn QSS-simulatoren, gir den en mer nøyaktig beskrivelse av dynamikken til et ekte kjøretøy.

Preface

This thesis concludes my Master of Science degree in Industrial Cybernetics, and two years at the Norwegian University of Science and Technology (NTNU), in Trondheim, Norway. The thesis was written in collaboration with the Formula Student team Revolve NTNU, which I joined back in the fall of 2019. I am grateful for my time in Revolve NTNU, as it has given me a highly technical and hard-working, engineering community to belong to.

During my first year at Revolve, I had the position *Vehicle Dynamics* on Team 2020. I got to investigate potential performance improvements, design suspension geometry, and aid in the design of suspension parts. I was lucky to be mentored by several great multi-year members as part of the group *Vehicle Dynamics & Control Systems*. My second, and now last year at Revolve, I had the position *Lap Time Simulation* on Team 2021. Again I was lucky to be mentored by several great multi-year members as part of the group *Vehicle Dynamics*, as I undertook my first software development project. It has been a great experience, as I have gotten to develop Revolve NTNU's first and second lap time simulator with my Master of Science project [1] and this thesis. I hope the two simulators serve Thomas well, and that they become a new technology for Revolve to explore and advance in future teams.

I would like to thank Professor Sebastien Gros, at the Department of Engineering Cybernetics, for being my supervisor on this project. His experience and technical expertise were of great service to me and Revolve. I would also like to thank Revolve NTNU and its members, past and present, for this great opportunity and experience.

Torbjørn Smith
Trondheim, June 27, 2021

Contents

Abstract	i
Sammendrag	ii
Preface	iii
Table of Contents	v
List of Tables	ix
List of Figures	x
Acronyms	xii
1 Introduction	1
1.1 Lap time simulation	1
1.2 Formula Student	2
1.3 Revolve NTNU	2
1.4 Thesis structure	4
2 Initial assessment and motivation	5
2.1 Motivation	5
2.2 Previous work and contributions	6
2.2.1 Transient simulation methods	6
2.2.2 Optimal Control	6

2.3	Aim of the thesis	8
3	Theory	9
3.1	Vehicle dynamics	9
3.1.1	Tires	10
3.1.2	Tire vertical load	13
3.1.3	Planar vehicle models	15
3.2	Numerical Optimization	17
3.2.1	Optimization	17
3.2.2	Ipopt	18
3.3	Optimal Control	19
3.3.1	General Optimal Control formulation	19
3.3.2	Numerical methods for solving Optimal Control Problems	19
3.4	Algorithmic Differentiation and CasADi	22
4	Method	23
4.1	Track modeling	23
4.1.1	Generating a track racing line	25
4.1.2	Optimal Control Scheme for smoothing curvature	26
4.2	Initial small scale implementation	30
4.3	Shift from time to spacial dimension	30
4.4	Vehicle modeling	31
4.4.1	Tire velocities	31
4.4.2	Tire slip angles	32
4.4.3	Vertical tire forces	33
4.4.4	Longitudinal and lateral tire forces	33
4.4.5	Dynamic model	34
4.4.6	Powertrain	35
4.5	Optimal Control Problem formulation	37
4.5.1	General Optimal Control Formulation	37
4.5.2	System states and inputs	37
4.5.3	Performance index	39
4.5.4	Constraints	40
4.6	Transcribing the OCP into an NLP	41
4.6.1	Collocation method	41
4.6.2	Cost function	43
4.6.3	Constraints	43

4.6.4	Measures for more stable convergence	43
4.7	NLP formulation and solving	44
5	Results	45
5.1	Step length investigation	45
5.2	Track models and smoothing track curvature	46
5.3	Preliminary simulation results	49
5.4	Full simulation results: 2019 FSG Autox	50
5.5	OC simulator compared to telemetry	53
5.5.1	Laptimes	53
5.5.2	Velocity and acceleration profiles	53
5.5.3	GG diagrams	56
5.6	OC simulator compared to QSS simulator	56
5.6.1	Laptimes	56
5.6.2	Velocity and acceleration profiles	57
5.7	Comparing overall vehicle designs - Nova vs Atmos	59
5.7.1	Laptimes	59
5.7.2	Velocity and acceleration profiles	59
6	Discussion	61
6.1	Data collection	61
6.1.1	Accuracy of the telemetry	61
6.1.2	Package loss	62
6.2	Track model	62
6.3	Preliminary Study - The Optimal Trajectory	63
6.3.1	Challenges with the chosen methodology	63
6.4	Full simulator	64
6.4.1	Vehicle Modeling	64
6.5	OC compared to QSS	68
7	Conclusion and Future Work	69
7.1	Conclusion	69
7.2	Future Work	70
7.2.1	Vehicle modeling	70
7.2.2	Autonomous planning	71
7.2.3	Parameter Optimization	72
7.2.4	Graphical user interface	72

Bibliography	73
Appendix A	79
A.1 Vehicle Parameters - R19 Nova	79
A.2 Vehicle Parameters - R18 Atmos	80
Appendix B	81
B.1 Magic Formula 5.2	81
B.1.1 Inputs and Outputs	81
B.1.2 Normalized vertical load increment	81
B.1.3 Pure Lateral Slip	82
B.1.4 Pure Longitudinal Slip	84
B.1.5 Lateral Slip (Combined Slip)	86
B.1.6 Longitudinal Slip (Combined Slip)	88

List of Tables

4.1	Track snippets used in preliminary study	30
4.2	States and inputs used in the OCP	39
5.1	Laptimes compared: OC simulator vs recorded	53
5.2	Laptimes compared: OC simulator vs QSS simulator	56
5.3	Laptimes compared: Nova vs Atmos	59
A.1	Vehicle parameters for Revolve NTNU's 2019 car: Nova	79
A.2	Vehicle parameters for Revolve NTNU's 2018 car: Atmos	80
B.1	Inputs and outputs from the Magic Formula 5.2	81
B.2	Coefficients – Pure Lateral Slip	83
B.3	Coefficients – Pure Longitudinal Slip	85
B.4	Coefficients – Lateral Slip (Combined Slip)	87
B.5	Coefficients – Longitudinal Slip (Combined Slip)	88

List of Figures

1.1	Revolve NTNU's 2021 racecar: <i>Luna</i>	3
3.1	Vehicle sign convention displayed. Vehicle frame defined by CoG in x- and y-direction, and ground level in z-direction	10
3.2	Tire frame: Tire longitudinal, lateral, and angular velocity shown.	11
3.3	Magic Formula pure longitudinal and lateral version for different F_z	13
3.4	The coefficient of friction decreasing with vertical load F_z	14
3.5	Single track vehicle model	16
3.6	Double track vehicle model	17
4.1	The curvilinear track model, adopted from Perantoni and Limebeer [2]	23
5.1	Step length investigation: The resulting lap time, and CPU-time for a set of step lengths is shown on a semi-log scale. The approximate CPU-time function is also shown.	46
5.2	2019 FSG AutoX: Results from track interpolation and smoothing scheme	47
5.3	2019 FSA AutoX: Results from track interpolation and smoothing scheme	48
5.4	Left hand turn: Results from the preliminary study	49
5.5	Full simulator: 2019 FSG AutoX results. Velocity and acceleration profiles	51
5.6	Full simulator: 2019 FSG AutoX results. Vehicle inputs and powertrain	52
5.7	2019 FSG AutoX: OC simulator vs telemetry. Velocity and acceleration profiles	54
5.8	2019 FSA AutoX: OC simulator vs telemetry. Velocity and acceleration profiles	55

5.9	OC simulator vs telemetry. GG diagrams compared for 2019 FSG and FSA AutoX	56
5.10	2019 FSG AutoX: OC simulator vs QSS simulator. Velocity and acceleration profiles	57
5.11	2019 FSA AutoX: OC simulator vs QSS simulator. Velocity and acceleration profiles	58
5.12	2019 FSG AutoX: Nova vs Atmos. Velocity and acceleration profiles . . .	60
6.1	2016 FSG AutoX: INS vs Kistler Optical sensor Longitudinal velocity u .	61
6.2	2019 FSG AutoX: Tire vertical loads $F_{z,ij}$	65
6.3	Performance envelope comparison showing longitudinal and lateral acceleration	66
6.4	2019 FSG AutoX: Longitudinal and lateral velocities u and v compared .	67

Acronyms

AD Algorithmic Differentiation. 22

AutoX Autocross. 2

BPP Business Plan Presentation. 2

CAD Computer Aided Design. 1, 3

CAM Computer Aided Manufacturing. 1

CAS Computer Algebra Systems. 22

CFD Computational Fluid Dynamics. 1, 15, 65

CoG Center of Gravity. 6, 9, 13, 14, 24, 32, 34

CoP Center of Pressure. 15

DAE Differential-Algebraic system of Equations. 21, 37

DOF Degrees Of Freedom. 71

ED Engineering Design. 2

FEM Finite Element Method. 1, 3

FS Formula Student. 2, 3

FSA Formula Student Austria. 8

FSG Formula Student Germany. 8

GUI Graphical User Interface. 72

INS Inertial Navigation System. 25, 61

IP Interior-point. 7, 8, 18, 29, 44

Ipoint Interior Point Optimizer. 18

LTS Lap Time Simulation. 1–8, 26, 30, 69

MF Magic Formula. x, 13

MLTP Minimum Lap Time Problem. 1, 4

MTM Minimal Time Manoeuvring. 6, 7, 26, 69, 70

MVP Minimum Viable Product. 30

NED North-East-Down. 26

NLP Non Linear Programming. 4, 6–8, 18, 20, 28, 29, 37–44, 69

OC Optimal Control. 4, 6–9, 26, 30, 53, 56, 58, 59, 62, 63, 66–72

OCP Optimal Control Problem. ix, 4, 6–8, 19–22, 26–31, 33, 35, 37–41, 43, 69, 70

ODE Ordinary Differential Equation. 19, 20

QSS Quasi Steady State. 4–8, 56, 58, 62, 65, 68, 70

SNOPT Sparse Nonlinear OPTimizer. 18

SQP Sequential Quadratic Programming. 7, 18

Introduction

1.1 Lap time simulation

The final judge in a competition between racecars is the time on the stopwatch. The time a vehicle uses to complete a given number of laps around a course or track decides the winners and the losers. The goal of a team building and running a racecar is therefore fixed on making its vehicle as fast as possible, where fast is defined as able to minimize the time spent traveling through a given track. Notice the two aspects of vehicle design and vehicle control embedded in this process. To achieve minimum lap times the two aspects need to be optimized. The vehicle must be designed and built to minimize the lap time, and the driver must be able to control the vehicle to minimize the lap time when running. This dual problem creates the need for a tool to quantify how a vehicle would perform in these two aspects.

Engineers utilize many computer-based tools to achieve their tasks. Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), Computational Fluid Dynamics (CFD), Finite Element Method (FEM), are all examples of this, and are common at an engineering firm or establishment. In the world of vehicle design and racing, one such tool is Lap Time Simulation (LTS). LTS software vary widely, but one major part of LTS is to solve what is referred to in the literature as the Minimum Lap Time Problem (MLTP) [3]. The objective is to simulate a vehicle driving around a representative track, and taking note of how fast the vehicle was able to do that.

1.2 Formula Student

Formula Student (FS) is the world's largest engineering competition for students. It has a large presence in Europe, but is not as recognized in Norway. FS is about designing and racing an open-wheel formula-style racecar, but the competition is about engineering. The vehicle is just a good platform to build an engineering completion on. Building a racecar is interdisciplinary, both in engineering, and academia at large. The bulk of teams are made up of mechanical, and electrical engineering students, broadly speaking, but leadership and economics also find their space on the teams.

The engineering focus becomes apparent when looking closer at the competition structure. The FS competitions are divided into two main parts: Dynamic, and static events. The dynamic events are driving events designed to push the vehicle and driver to the limit of performance, and the time on the stopwatch is the judge. The dynamic events are Skidpad, Acceleration, Autocross (AutoX), and Endurance. All performance aspects of the vehicle are tested, on top of the efficiency and reliability. The static events are Business Plan Presentation, Cost and Manufacturing, and Engineering Design (ED). In these events, the economics and financial management of the vehicle design are evaluated, and the vehicle design and its designers are evaluated and tested by judges from industry leaders, such as major car manufacturers and racing teams. In short, the vehicle design must perform well on the track, but the engineering choices made in the design of the vehicle are just as important. The formula student engineers, therefore, need a keen eye on the stopwatch, as well as the reasoning behind the vehicle design. LTS software fits into this narrative, providing a holistic perspective on the vehicle design, with information about the lap time and the vehicle dynamics that produced the result, without the need for physical testing.

1.3 Revolve NTNU

Revolve NTNU is an independent student organization from Trondheim that competes in FS. Revolve was started in 2010 and completed its first racecar in 2012. Every year a new racecar is designed and produced by Revolve. The racecar is conceptualized, designed, built, tested, and raced, all in one year. With the exception of the 2020 vehicle R20 that was canceled due to the COVID-19 pandemic, Revolve has been able to join other FS teams in competition's every year, achieving a top 10 world ranking.

This achievement has come as a result of dedicated, innovative, and ambitious members, with the help of great sponsors. To be in the front of the pack, cutting-edge technology is needed. The racecar gained an aerodynamic package in 2013. The central chassis, or

monocoque, has been made from carbon fiber and aluminum honeycomb since 2014. The powertrain was made electric in 2014. In 2016 the one motor, rear-wheel drive powertrain, was replaced by four-wheel drive, with four hub mounted motors. This allowed the introduction of a torque vectoring algorithm, to extract all available grip and control in the vehicle, placing it on the edge of performance. In 2019 the in-house constructed motors were driven by an in-house developed inverter and controller. For 2021 Revolve has produced its lightest racecar ever, with the adoption of a smaller tire than what has been on our racecars for the last four years. This gives weight cuts in the unsprung, rotating mass of the vehicle, less aerodynamic interference, and gives our drivers a more forgiving vehicle to drive on the edge of grip and control.

The cutting-edge designs listed are backed by the cutting-edge software tools written about earlier in the introduction. However, one area lacking has been the capabilities in LTS at Revolve. No dedicated effort has been made to establish tools and procedures for LTS at Revolve. Specific studies have been made, but not taken further and made as established as our CAD or FEM software. For Team 2021 this changed, and a dedicated member was hired to research and develop LTS tools at Revolve NTNU. The aim is to be able to simulate vehicle designs to find potential performance increases, but also to better justify design decisions made, as both are needed to achieve a good result in a FS competition.



Figure 1.1: Revolve NTNU's 2021 racecar: *Luna*

1.4 Thesis structure

Chapter 1: Introduction gives an overview of the origin of the project and establishes the needed background for the utility of the work.

Chapter 2: Initial assessment and motivation is an audit of the relevant related work. The research goals for the project are established, as a choice of methodology is made.

Chapter 3: Theory gives an overview of the relevant theory. Basic vehicle dynamics are introduced to give the background for the modeling in the simulator. An introduction to relevant Optimal Control (OC) theory is given, with a focus on the direct transcription methods.

Chapter 4: Method presents the methodology and process used when building the simulator. The vehicle modeling and track generation are presented, and the formulation of the MLTP as an Optimal Control Problem (OCP) transcribed into an Non Linear Programming (NLP) is covered.

Chapter 5: Results presents the resulting track maps from generating the tracks. The resulting lap times, velocity, and acceleration profiles from the simulator are also presented, and compared to both telemetry and the QSS LTS produced in the project thesis by [1]. Two different vehicle designs are also compared.

Chapter 6: Discussion evaluates the results presented in chapter 5, with special emphasis on identifying limitations in the modeling and simulation methodology chosen. A discussion over the pros and cons of an OC based simulator compared to QSS based simulator is also made.

Chapter 7: Conclusion and Future Work summarizes what has been achieved through this MAster of Science project, answering the research goals proposed in chapter 2. A comprehensive list of possible future extensions to the work presented in this thesis is also presented.

Initial assessment and motivation

2.1 Motivation

In the authors project thesis [1], a QSS LTS was produced. The simulator was built on methodology well established in the relevant literature [4–10]. The methodology used is based on a steady-state approximation of the vehicle dynamics, used to solve for the longitudinal acceleration on each segment of a discretized track. In [1] a double-track vehicle model with an aerodynamic package and Pacejka tire model was used. The vehicle model was used in a numerical optimization step, to build a performance envelope containing the steady-state, longitudinal, and lateral acceleration information for a range of velocities. The performance envelope was used to build a velocity profile for a given track centerline, consisting of short segments of constant radius turns. It was shown to be a good approximation to the real vehicle, though with some important caveats. The QSS method was chosen as it represented an optimization approach to finding the minimum lap time, the on-limit performance, at a relatively low computational cost. The QSS method does lack actual vehicle dynamics, or the transient effects as a result of accelerating a rigid body. As a result, a LTS capable of modeling transient effects is desired.

2.2 Previous work and contributions

2.2.1 Transient simulation methods

Moving beyond the methodology of the QSS LTS, there are several relevant simulation methods outlined in the literature, that take into account the transient effects of a vehicle. The QSS method was expanded by Patton [9] to include transient effects.

A large subcategory of LTS is the Driver Model method of simulation [11–14]. There are several different methods, but the commonality is that the methods try to emulate the behavior of a driver. As in [1], these methods are deemed suboptimal, as the need of Revolve NTNU is LTS software that finds the on-limit handling performance of the vehicle under study. This is not the case with the driver model, as the behavior of a driver is not trivial to emulate through automatic control. Vehicle performance analysis is therefore often limited to just the vehicle [15, Chapter 5].

2.2.2 Optimal Control

The Optimal Control (OC) based LTS is the final large category of lap time simulation software. The OC based LTS formulates the simulation as a large optimization problem, where the aim is to solve for the optimal vehicle inputs that minimize the lap time. The trajectory can be fixed or free, where the latter is commonly referred to as a Minimal Time Manoeuvring (MTM) problem where the trajectory and control inputs are optimized simultaneously. This produces the optimal racing line for a track that minimizes the lap time. Much like the QSS method, this method finds the on-limit handling of the vehicle, taking the driver out of the analysis. Unlike the QSS method, as the entire lap is solved as one large optimization problem, dynamics and transient effects are included. This expands the possibility for modeling in the LTS, as a more complete vehicle model can be used.

Using OC for LTS is well established in the relevant literature. One of the earliest contributions was made by Casanova [16], as he applied a direct multiple shooting method to transcribe the MTM OCP into an NLP. A full lap simulation used hours of CPU time. In the simulation, he used a double track, planar vehicle model, with a nonlinear tire model, and aerodynamic package. The effects of moving the Center of Gravity (CoG) and changing the yaw inertia of the vehicle were demonstrated on the MTM problem, validating the underlying vehicle dynamics. Kelly [17] used a direct single shooting method, with stability constraints applied on the maneuver. It was successful at modeling the nonlinear model, including suspension dynamics. Much like Casanova, a CPU-time measured in hours was reported.

Perantoni and Limebeer [2] used a direct collocation method to transcribe the MTM into an NLP. The paper also outlines a method for generating a track centerline, formulating a separate OCP to create a smooth track centerline as an input to the OC LTS. The methodology for vehicle parameter optimization was also shown. The CPU time was reported to be less than 15 minutes. The reduction in CPU-time was due to the use of a curvilinear track description, eliminating stiff dynamics in the vehicle model, model non-dimensionalization and scaling, approximating non-smooth model features with differentiable functions, and the use of computer-generated analytic derivatives in combination with an open-source sparse large-scale NLP solver. In two subsequent papers [18, 19] Perantoni and Limebeer expanded their work and modeled the track in three dimensions using a ribbon. This furthers the work done in [2], generating a smooth track centerline from GPS data, but now including changes in elevation and attitude. The effects of the change in elevation and vehicle attitude were demonstrated on a lap of Circuit de Catalunya in Barcelona, Spain.

Van Koutrik [20] used a direct collocation approach to simulate a double-track vehicle model, with load transfer, a basic aerodynamic package, and limited-slip differential, through short maneuvers. The effects of the different model aspects were evaluated, showing the importance of each aspect of the model. The removal of the fast wheel dynamics from the model was shown to have little effect on the simulations, and the tire slip ratios were used as model inputs instead of the more common wheel angular velocities. A comparison between the Sequential Quadratic Programming (SQP) method and Interior-point method for solving the resulting NLP was shown, and the Interior-point was shown to be advantageous as the size of the NLP grew.

Veneri and Massaro [7] produced a different type of LTS deploying an OCP formulation using a performance envelope based on a QSS vehicle model much like in [1], but with a free trajectory optimization step as is usually seen in OC based LTS. The method was used to simulate both a four-wheel car model and a two-wheel motorcycle model. The solution of a free trajectory problem was compared to solving the problem using a fixed trajectory and also compared to a fixed trajectory apex-finding method like the one deployed in [1]. The results showed an increased CPU runtime for the free trajectory simulation compared to the fixed trajectory and apex-finding methods. Their work opens the possibility to utilize experimental performance envelopes from test data, as opposed to numerically derived envelopes.

Christ et al. [3] have produced a series of papers on their work in the autonomous racing series Roborace [21]. In [3] Christ et al. used a free trajectory OCP formulation solved with a direct orthogonal collocation method to plan the time-optimal racing line for an

autonomous racecar. Reduced CPU-runtimes were achieved using a curvilinear track description, algorithmic differentiation with the open-source software framework CasADi [22], and track centerline smoothing by approximate spline regression. The NLP was solved using the Interior-point method with Ipopt [23]. Both a single-track and a double-track vehicle model were simulated. The effects on the final trajectory from variable road friction coefficients were demonstrated, showing yet another example of the capabilities in the modeling alternatives of the OCP simulation method.

2.3 Aim of the thesis

The aim of this thesis is to produce a LTS software much like in [1], but using a free trajectory OCP formulation solved using a direct orthogonal collocation method and NLP solver. This will give Revolve NTNU capabilities in LTS with two separate simulators using two different, leading methodologies. The QSS apex-finding method in [1] provides a strong modeling capability, simulating the vehicle at on-limit performance, at relatively low computational cost. The work presented in this thesis on the other hand, provides a methodology for a more complete description of the vehicle dynamics, with a dynamic model including transient effects, while still simulating the vehicle at on-limit performance.

The vehicle model used in this thesis is similar to the one used in [1]. It is a double track, planar vehicle model, with load transfer and a nonlinear tire model. A basic aerodynamic package is included, and a basic description of the powertrain is used. The wheel rotations will be omitted in favor of using the tire longitudinal slips as control inputs, together with steering and longitudinal and lateral load transfer. The track model used in [1] is discarded, and a track model is produced from GNSS coordinates recorded at the 2019 Formula Student Germany (FSG) [24] and Formula Student Austria (FSA) [25] competitions. A track smoothing scheme will be implemented, using a direct multiple-shooting method to create a smooth curvilinear abscissa for the OC LTS. The simulation results will be compared to both telemetry from the stated competitions, and to the QSS simulator in [1]. Finally, two different vehicle designs will be compared.

Chapter 3

Theory

The chosen methodology requires a description of the underlying theory. In this section, an introduction to vehicle dynamics is presented. This is necessary for justifying the chosen aspects of the model, and for the analysis of the results of the simulator. An introduction to the relevant theory in OC is also presented, as a distinct choice is made regarding the method, and as several techniques are utilized.

3.1 Vehicle dynamics

Vehicle dynamics is the study of vehicle design. At Revolve NTNU every team member must have some knowledge about vehicle dynamics, as the field is the starting point for all downstream decisions made regarding the vehicle design. A thing as trivial as minimizing the center of gravity height of the vehicle has a clear basis in vehicle dynamics. To build a good racecar, a good understanding of vehicle dynamics is a prerequisite.

A choice is made regarding the sign convention in this thesis. In the literature there are several sign conventions presented [26, Appendix 1], and this thesis will use the ISO standard sign convention. A Cartesian coordinate system is defined in the vehicle Center of Gravity (CoG) at the ground level. The x-axis points forward on the car, the z-axis points up in the opposite direction of gravity, and the y-axis points out to the left of the vehicle completing the right-hand coordinate system. This sign convention is shown in Figure 3.1. The movement of the vehicle is defined by the vehicle frame. Surge, sway, and heave are the linear motion along the x-, y-, and z-axis. Roll, pitch, and yaw are the rotations about

the x -, y -, and z -axis [15, Chapter 4].

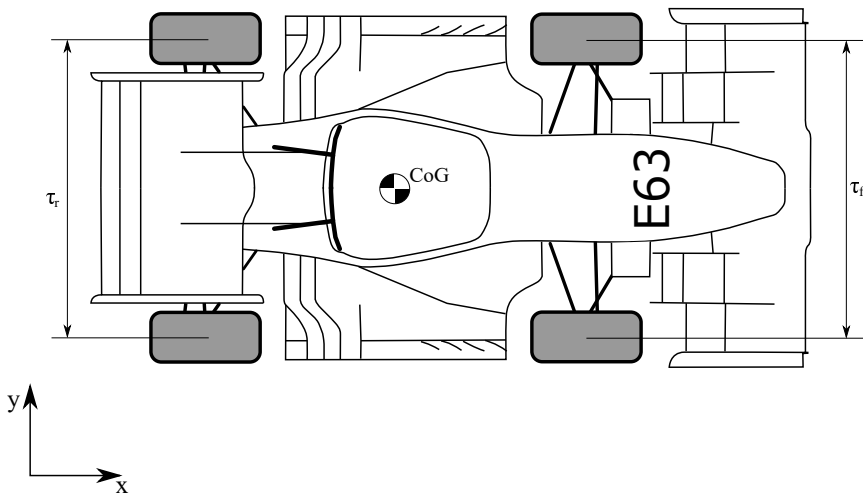
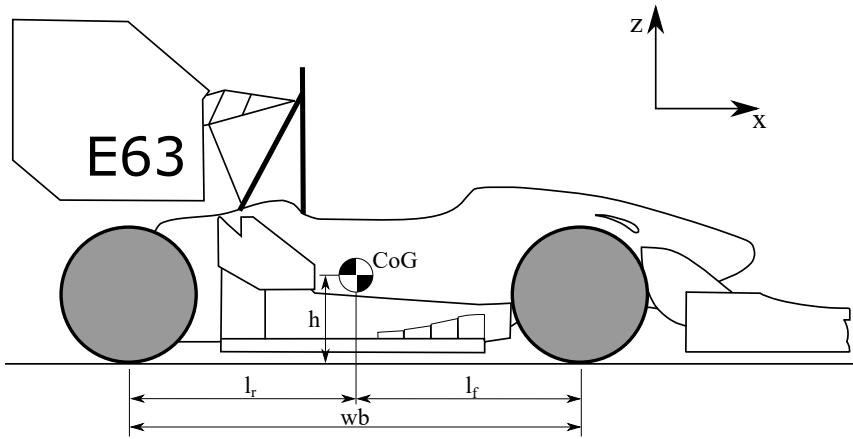


Figure 3.1: Vehicle sign convention displayed. Vehicle frame defined by CoG in x - and y -direction, and ground level in z -direction

3.1.1 Tires

Apart from gravitational and aerodynamic forces, a racecar mainly interacts with its surroundings through its tires. The tires are therefore the main decider of the performance of a racecar [27, Chapter 3][28, Chapter 2]. The rest of the vehicle is in many ways designed

to serve the dynamics of the chosen tire. As in [1], this thesis will not go into detail about how a tire is constructed or how it produces grip. This work will focus primarily on how the grip produced by the tire is modeled. For this work, the tire is viewed as a finite amount of friction. A vertical load is put on the tire, and it produces a finite amount of longitudinal and lateral grip that is used to control the racecar. This grip is produced in the tire contact patch, the area of the tire that is in constant with the ground as a result of the soft rubber tire deforming under the vertical load placed on it.

The tire is defined by the tire frame, a cartesian coordinate frame located at the center of the contact patch, with the same orientation as the vehicle frame. The tire axis is parallel to the tire frame y-axis, and the tire rotation or angular velocity is defined by the right-hand rule about the tire axis. The tire angle and the slip angle are both measured in the ground plane about the tire z-axis. The tire inclination angle is defined by the right-hand rule about the tire frame x-axis.

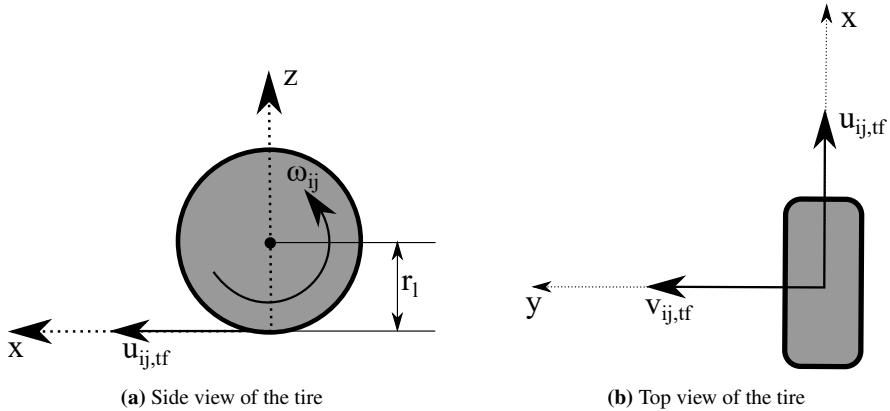


Figure 3.2: Tire frame: Tire longitudinal, lateral, and angular velocity shown.

Slip ratio

A tire produces force through vertical load and tire slip: Slip ratio and slip angle. When the moment is applied to a tire, from engine torque or braking, longitudinal slip is induced [26, Chapter 1], and longitudinal force is produced. Longitudinal slip or slip ratio is defined as the ratio between tire rotation for a driven tire and tire rotation for free-rolling tire:

$$\kappa = \frac{\omega}{\omega_0} - 1 \quad (3.1)$$

where κ is the slip ratio, ω is the angular velocity of a driven tire, and ω_0 is the angular

velocity of a free-rolling tire.

Equation (3.1) is rewritten using the loaded radius of a vertically loaded, free rolling tire, and the longitudinal velocity of the tire in the tire frame:

$$\kappa_{ij} = \frac{\omega_{ij} r_l - u_{ij,tf}}{u_{ij,tf}} \quad (3.2)$$

where r_l is the loaded radius, and $u_{ij,tf}$ the longitudinal velocity in the tire frame, where the subscript ij designates the corner of the racer where the tire is located.

Slip angle

Lateral slip is the ratio between the longitudinal and the lateral velocity of the tire in the vehicle frame [26, Chapter 1]. The slip angle for a given tire is defined by the relation:

$$\alpha_{ij} = -\arctan \frac{v_{ij,vf}}{u_{ij,vf}} \quad (3.3)$$

where α is the slip angle, and $u_{ij,vf}$ and $v_{ij,vf}$ are the longitudinal and lateral velocity of the given tire in the vehicle frame.

Tire models and the Pacejka Magic Formula

With vertical load, slip ratio, slip angle, and inclination angle defined, the longitudinal and lateral forces $F_{x,ij}$ and $F_{y,ij}$ produced by the tire are defined as shown in [26, Chapter 1]:

$$F_{x,ij} = F_x(\kappa_{ij}, \alpha_{ij}, \gamma_{ij}, F_{z,ij}) \quad (3.4)$$

$$F_{y,ij} = F_y(\kappa_{ij}, \alpha_{ij}, \gamma_{ij}, F_{z,ij}) \quad (3.5)$$

where γ_{ij} is the inclination angle, and $F_{z,ij}$ is the vertical load on the tire. This definition is for the combined case, as slip ratio and slip angle are inputs to both functions. Again, the tires are a finite amount of friction force, so it is trivial to point out that a tire that produces a high amount of lateral force close to peak grip, is not able to produce an equally high amount of longitudinal force.

There are many tire models published in the literature, spanning from friction coefficients to large, transient models. As tires are complex constructions made from fibers, steel wire, and vulcanized polymer, it is important to note that the way they produce grip, is not fully understood [15, Chapter 2].

One of the most well-established tire models in the literature is the Pacejka Magic Formula (MF) [26]. It is based on a $\sin(\arctan(\cdot))$ formula, with empirically derived coefficients. It provides a good model of the steady-state forces and moments produced by the tire [26, Chapter 2]. The model is non-linear and includes non-smooth and discontinuous functions. The empirical nature of the formula, based on a finite amount of specific tire testing, puts limits on its use, and good engineering judgment is required when drawing a conclusion based on it.

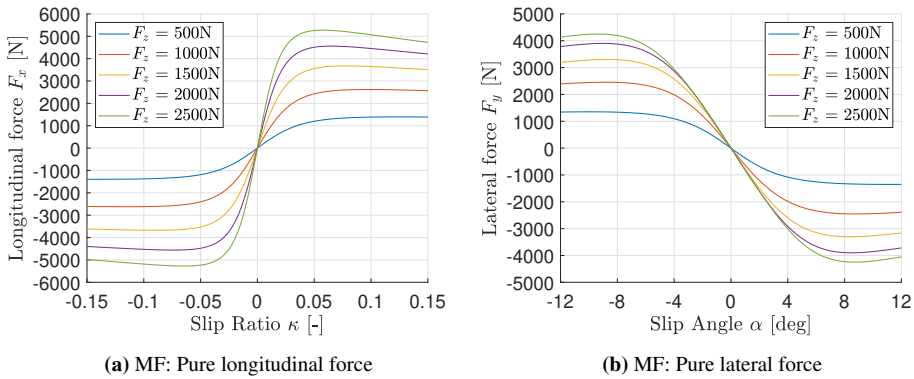


Figure 3.3: Magic Formula pure longitudinal and lateral version for different F_z

3.1.2 Tire vertical load

The final input to the tire model used in this work is the vertical load placed on the tire. Several possible contributors to this load are omitted, and the focus is placed on the vehicle mass, its distribution, the acceleration of the vehicle, and the aerodynamic loads on the vehicle.

Center of Gravity and Weight Distribution

In this work, the vehicle mass is idealized to be located in an infinitesimally small point called the CoG. All forces and moments attack in and about this point. The vertical force on the tires from the gravitational weight of this mass is distributed between the four tires based on the location of the CoG in the XY-plane of the vehicle coordinate system. This distribution of weight is often referred to as the weight distribution of the vehicle. It is often given as a percentage of the total vehicle mass that is on the front or rear axle or distributed between the left and right tires.

Load Transfer

It is observed that peak force produced by a tire longitudinally and laterally, relative to the vertical load, decreases with an increase in vertical load. In short, the coefficient of friction decreases as the vertical load increases. This effect is shown in Figure 3.4 as peak F_x and peak F_y , are shown for the input F_z .

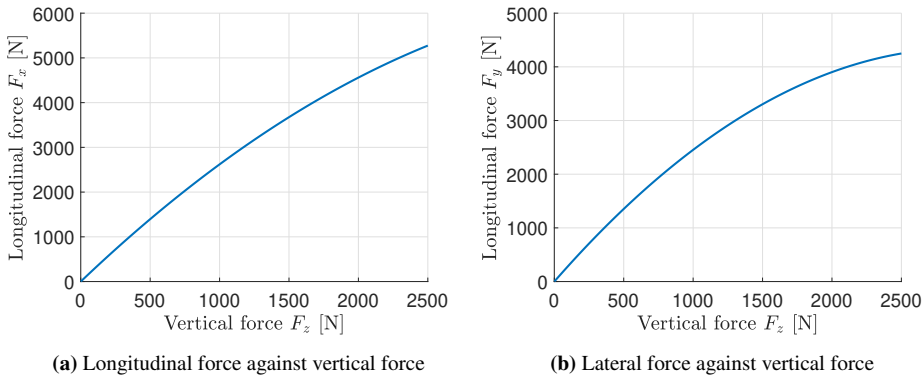


Figure 3.4: The coefficient of friction decreasing with vertical load F_z

This effect is an important characteristic of vehicle dynamics, as it means that the grip produced by equal tires that are equally loaded, is greater than the grip produced by unequally loaded tires, at the same total vertical load. As the CoG is above ground level, a moment is produced as the vehicle mass is accelerated, causing a shift of the load from one tire pair to the other. This longitudinal and lateral shift of load is called load transfer, and it is the reason for designing a racecar with a CoG that is as low as possible. The lower the CoG, the less longitudinal and lateral load transfer, the more grip is produced for a given vertical loading of the tires [28, Chapter 3].

Aerodynamic Forces

Just adding mass to get more grip is no good because of the effects shown in Figure 3.4, but there is a way to add vertical load on the tires without increasing vehicle mass. This is the aerodynamic component of the vertical loading of the tires. An aerodynamic package adds several aspects to a vehicle. First, the aerodynamic package adds downforce (negative lift), increasing grip through increasing the vertical loading of the tires. The aerodynamic package can also reduce aerodynamic drag, the main decider of the efficiency and the top speed of the vehicle. Also, depending on the relative significance of the aerodynamic package, it can play a major role in vehicle handling and stability.

Unlike the vehicle weight, the tire vertical loading from the aerodynamic package is dependant on the velocity of the vehicle. Much like the vehicle mass, aerodynamic forces can be idealized to attack in an infinitesimally small point called the Center of Pressure (CoP). The aerodynamic lift and drag forces can be modeled as:

$$F_L = \frac{1}{2} C_L A \rho u^2 \quad (3.6)$$

$$F_D = \frac{1}{2} C_D A \rho u^2 \quad (3.7)$$

Where F_L and F_D are the aerodynamic lift and drag forces respectively, C_L and C_D are the coefficients of lift and drag respectively, A is the frontal area of the vehicle, ρ is the air density, and u is the vehicle longitudinal velocity. The coefficients of lift and drag are usually derived from CFD or wind tunnel testing. It is also normal to include sweeps of the different modes of the vehicle, producing the coefficients for different velocities, ride heights, yaw, pitch, and roll angles.

3.1.3 Planar vehicle models

To build the equations of motion, a vehicle model is needed. In vehicle dynamics, planar vehicle models are often deployed. These models simplify the vehicle dynamics by removing the vertical motions of the vehicle, such as suspension kinematics and unsprung masses like the tires, and remove most of the vehicle motions defined in the introduction to Section 3.1. This simplification holds as racecars usually have very little suspension motion [15, Chapter 5], due to stiff springs and aggressive suspension kinematics, sacrificing comfort and mechanical grip in favor of little body movement and a stable aerodynamic platform.

Single track model

The single-track vehicle model often referred to as the *Bicycle model*, is an elementary vehicle model found in many publications. It is applied to steady-state and transient analysis of vehicle motion, in both linear and nonlinear descriptions. The basic concept is to explain the planar dynamics of the vehicle using an approximation of the vehicle by combining the front and rear tire pairs, as single front and rear tires, neglecting the track width of the vehicle. Milliken and Milliken [15, Chapter 5] outlines the simplifications that are often made. The benefit of the single-track model is the low barrier to entry for modeling proper yaw dynamics, and proper modeling of the slip conditions at the front and rear tires. Figure 3.5 shows a the single track modeled.

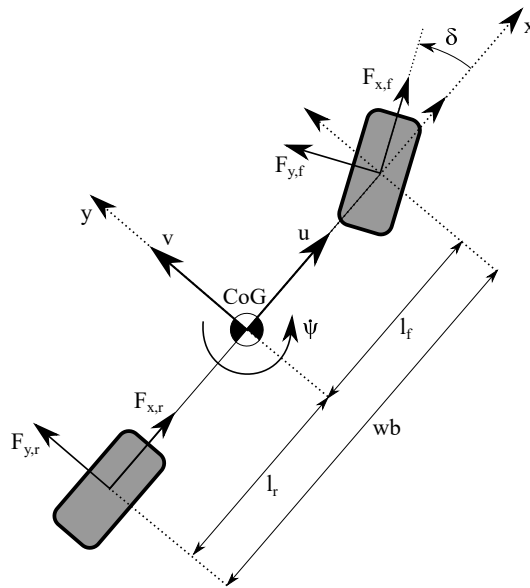


Figure 3.5: Single track vehicle model

Double track model

A small step up from the single-track model is the double-track model. The double-track model can be viewed as two single-track models separated by the track width of the vehicle. It increases the complexity of the model but is still a planar model. The step up from the single track is the fact that all four tires are modeled independently, more accurately modeling the slip conditions of the individual tires. Both longitudinal and lateral load transfer is usually included, and the kinetics of the model is more complete, as the rigid body models all four tires producing forces in their proper locations on the vehicle. Furthermore, differential tire spin and torque can be modeled, as is the case when using a mechanical differential or using independently driven tires. Van Koutrik [20, Chapter 2] showed the importance of the load transfer and differential tire dynamics using a steady-state analysis of the acceleration limits of the vehicle. Figure 3.6 shows a the double-track modeled.

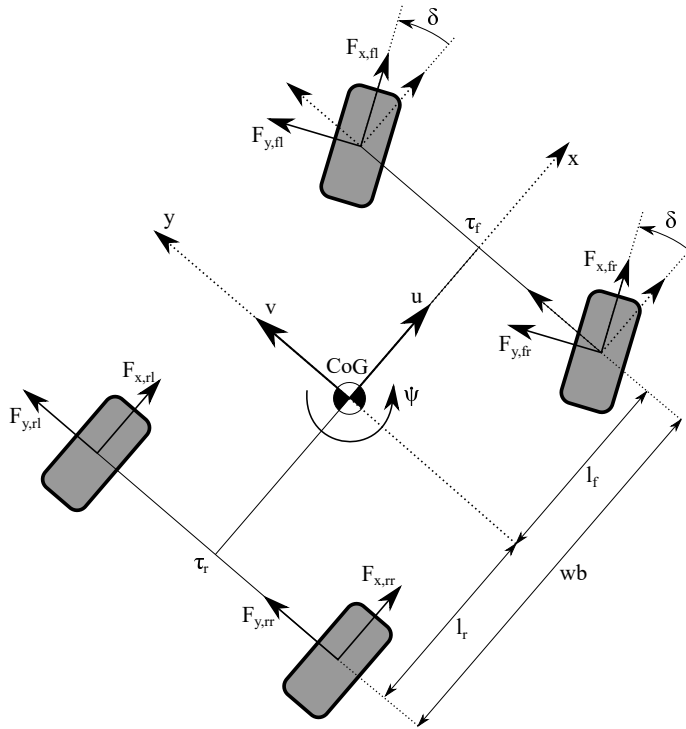


Figure 3.6: Double track vehicle model

3.2 Numerical Optimization

3.2.1 Optimization

In mathematics, optimization is the maximisation or the minimization of a function, subject to constraints on its variables. A optimization problem is defined by its variables \mathbf{x} , also known as unknowns, its objective function f , a scalar function of \mathbf{x} , to be minimized or maximized, and constraint functions of \mathbf{x} : g and h , that define equalities and inequalities that \mathbf{x} is subject to (*s.t.*) and must satisfy. The constraints define the *feasible region*, defined as the set of points that satisfy all the constraints [29, Chapter 1].

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{x}) = 0 \\ & \mathbf{h}(\mathbf{x}) \leq 0 \end{aligned}$$

Where \mathbf{x} is the variables (scalar or vector), f is the objective function, \mathbf{g} is the equality constraint, and \mathbf{h} is the inequality constraint. This minimization problem is trivial to turn into a maximisation problem as:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \max_{-\mathbf{x}} f(\mathbf{x})$$

Constraints do not always apply, and unconstrained optimization is a large field of study. For the purposes of this work, as physical systems are modeled, the constrained optimization is most relevant.

If the cost function and constraints are both linear functions, the optimization problem is a Linear Programming problem. If the cost function or constraints contain at least one nonlinear function, the problem is a Non Linear Programming (NLP) problem. This is often the case in physical science and engineering [29, Chapter 1].

In optimization, particularly with NLP problems, the difference between a local and a global optimum is important. A local minimum is a feasible point smaller than all feasible points close by. A global minimum is a feasible point smaller than all feasible points. [29]. In general it is not possible to prove a solution to an NLP is a global solution, and it comes down to the engineer or scientist who built and ran the optimization to evaluate the solution. There is a large research field dedicated to finding global optimums, known as *global optimization*, but this is beyond the scope of this thesis.

3.2.2 Ipopt

The nonlinear problems produced in this thesis are all solved using the Interior-point (IP) method. The open-source software package Ipopt [23] is a solver for large-scale nonlinear optimization. It uses the interior point line search method to find a local solution of an NLP. Van Koutrik [20] showed Ipopt to outperform SQP methods implemented in Sparse Nonlinear OPTimizer (SNOPT) [30, 31], as the size of the NLP problem got larger. The interested reader may read more about the Interior-point method in [29, Chapter 19].

3.3 Optimal Control

3.3.1 General Optimal Control formulation

A general formulation for a controlled system is given as:

$$\dot{x}(t) = f(x(t), u(t), t) \quad x(0) = x_0 \quad t > 0 \quad (3.8)$$

The state x_n will depend on the system dynamics f , the initial condition x_0 , and the input $u(t)$. A potential task may be to find the best input. Defining the best input is done with a performance index or cost function given as:

$$J(x(t), u(t), t) = m(x(T), T) + \int_0^T l(x(t), u(t), t) dt \quad (3.9)$$

where J is the cost, m is the terminal cost or Mayer term, and l is the running cost or Lagrange term. The terminal time $T > 0$ may be predefined or free. The purpose of optimal control is to find the optimum control input to a dynamic system $u^*(t)$ that minimizes the cost function J . The input $u^*(t)$ results in the state trajectory $x^*(t)$ [32, Chapter 8].

A general formulation of an OCP constrained by a continuous Ordinary Differential Equation (ODE) may be:

$$\begin{aligned} \min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \quad & \phi(\mathbf{x}(\cdot), \mathbf{u}(\cdot)) \\ \text{s.t.} \quad & \dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}, t) \\ & \mathbf{g}(\mathbf{x}(t_0), \mathbf{x}(t_f), t) = 0 \\ & \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0 \end{aligned}$$

where ϕ is the cost, \mathbf{x} is the system state, \mathbf{u} is the system inputs, \mathbf{F} are the system dynamics, \mathbf{g} is the equality constraint, and \mathbf{h} is the inequality constraint. The formulation is very similar to a general optimization problem, but with the addition of time-continuous dynamics.

3.3.2 Numerical methods for solving Optimal Control Problems

The numerical methods for solving OCPs are divided into *indirect* and *direct* methods [32, Chapter 8]. Indirect methods find a solution using the first-order necessary conditions,

building a large set of algebraic equations that are solved using a root-finding scheme. Direct methods use a discrete approximation of the OCP and solve a finite optimization problem as an approximation to the original OCP formulation. These methods are often described depending on the optimization and discretization of the system, where indirect methods are first optimized then discretized, and direct methods are first discretized and then optimized [33]. Direct methods are used in this thesis as their use is well documented in relevant literature summarized in Section 2.2.2.

Direct shooting methods

A continuous ODE-constrained OCP is given:

$$\begin{aligned} \min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \quad & \phi(\mathbf{x}(\cdot), \mathbf{u}(\cdot)) \\ \text{s.t.} \quad & \dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}, t) \\ & \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0 \\ & \mathbf{x}(t_0) = \mathbf{x}_0 \end{aligned}$$

In the direct *single-shooting* method the infinite-dimensional (continuous) input \mathbf{u} is discretized, often using piecewise constant inputs. Given the initial conditions of the OCP, the system dynamics are simulated using an integration scheme, the cost evaluated, and the NLP is solved as the numerical solver iterates on the input finding the optimum. This method is simple and the problem size is small. However, the method struggles as the time horizon is extended and the integration method is applied over a longer time horizon, possibly failing to find a solution [32, Chapter 8].

In the direct *multiple-shooting* method the problem is handled by dividing the time horizon into small local horizons $[t_k, t_{k+1}]$, and the states included in the optimization parameters. This produces a simultaneous approach where both the simulation and optimization is produced at the same time. This method effectively builds a structure consisting of multiple, small simulations, hence the name. Continuity between each local trajectory is handle by enforcing the constraints given as:

$$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) - \mathbf{x}_{k+1} = 0, \quad k = 0, \dots, N - 1 \quad (3.10)$$

Where \mathbf{f} is the integration function producing the end state for a local trajectory, and \mathbf{x}_{k+1} is the initial starting point for the next local trajectory [33]. This constraint handles what is referred to as *shooting gaps*. The structure of a multiple-shooting derived problem is

larger than that of an equivalent single-shooting problem, but the sparse structure of the problem, and the fact that the error is spread out between the many local trajectories, yields a problem that solves in comparable CPU-time to a single-shooting problem. The method shows superior local convergence given unstable systems. The direct multiple-shooting method also provides the opportunity to initialize the state trajectory, as it is a part of the optimization variables [33, Chapter 14].

Direct collocation method

The shooting methods discussed above, both use embedded integrators to simulate the state trajectory, often an explicit Runge-Kutta scheme. These methods can suffer when the different modes of the dynamic system have significantly different time constants, meaning some modes have very fast dynamics and some modes have very slow dynamics. This is called stiff dynamics [33, Chapter 11].

A direct transcription method that handles this well, and also Differential-Algebraic system of Equations (DAE), is the *direct collocation method*. Gros and Diehl [33] outlines how a direct collocation method is applied. The OCP is discretized in both states and inputs, much like when using the direct multiple-shooting method, but each local trajectory is approximated by an interpolation polynomial $P_k(t, \theta_k)$. In the case of orthogonal collocation methods, these are often made from Lagrange basis polynomials. On each time interval $[t_k, t_{k+1}]$ on the total time horizon t a set of collocation times are chosen: $t_{k,0}, \dots, t_{k,d}$, and the integration polynomial is constructed:

$$P_k(t, \theta_k) = \sum_{i=0}^d \theta_{k,i} p_{k,i}(t) \quad (3.11)$$

Where $\theta_{k,i}$ are coefficients that become optimization variables in the optimization step. The Lagrange basis polynomials of order d are defined as:

$$p_{k,i}(t) = \prod_{j=0, j \neq i}^d \frac{t - t_{k,j}}{t_{k,i} - t_{k,j}} \in \mathbb{R} \quad (3.12)$$

The collocation points on the time interval $[t_k, t_{k+1}]$ are chosen specifically to obtain a high integration order. The integration polynomial is made to represent the dynamics of the system by enforcing the starting point of the polynomial and by forcing the derivative of the polynomial to equal the system dynamics. These *collocation equations* are passed to the optimization solver as equality constraints [33]. The shooting gaps between the local

trajectories are also handled in the optimization solver.

3.4 Algorithmic Differentiation and CasADi

Algorithmic Differentiation

Algorithmic Differentiation (AD) [33, Chapter 5] is a method deployed when computing the derivative of an arbitrary function in a computer. AD uses a computers ability to evaluate elementary arithmetic and functions, and applies the chain rule to compute the derivative of a function. It is distinct from both symbolic differentiation and numerical differentiation. The former can yield inefficient symbolic derivatives as the functions grow in size and amount of variables. The latter can suffer from round-off errors as the derivatives as numerically derived.

CasADi

CasADi [22, 34] is an open source software tool built primarily for numerical optimization and optimal control. The syntax is borrowed from Computer Algebra Systems (CAS) and the tool was originally intended for performing AD, hence the name. CasADi has become a software tool that provides a simple syntax to implement and solve numerical optimization problem and OCPs. The solver used are not supplied in the framework, but rather allows the use of solvers specified and provided by the user like IPOPT [23]. The framework is available for MATLAB, Octave, Python, and C++.

Method

4.1 Track modeling

The track is modeled using a curvilinear coordinate system, where the vehicle will follow a centerline as the abscissa. An important simplification is to assume a flat track, without changes in elevation or banking. This methodology is well established in the relevant literature [2] and yields a compact notation to describe the vehicle's progression along the track.

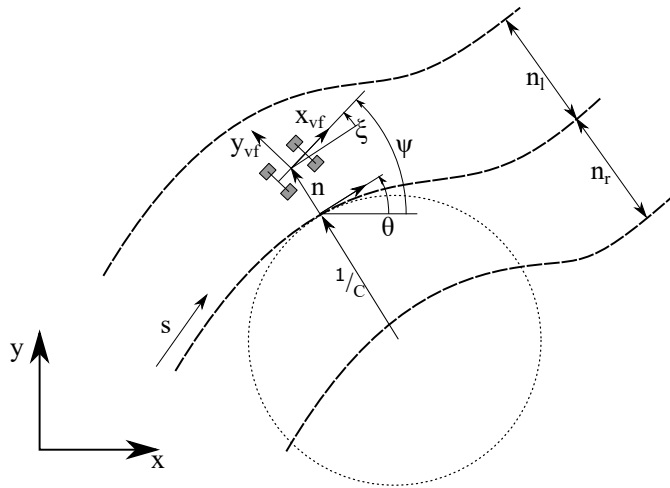


Figure 4.1: The curvilinear track model, adopted from Perantoni and Limebeer [2]

The track model used in this thesis is shown in Figure 4.1 and is adopted from Perantoni and Limebeer [2]. The track centerline length is measured from the track start line with $s(t)$. For each point $s(t)$ the track curvature is known, denoted as \mathcal{C} . The track orientation angle, meaning the orientation of the track segment tangent, is described by the angle θ .

The progress along the curvilinear description of the track ds can be converted to a change in a cartesian coordinate system as shown:

$$dx = ds \cos(\theta) \quad (4.1)$$

$$dy = ds \sin(\theta) \quad (4.2)$$

This description of the change in position on the track yields the following:

$$\frac{dy}{dx} = \tan(\theta) \quad (4.3)$$

Finally the track curvature can be derived from cartesian coordinates as given:

$$\mathcal{C} = \frac{d\theta}{ds} = \frac{d}{ds} \left(\arctan \left(\frac{dy}{dx} \right) \right) \quad (4.4)$$

The vehicle motion in the track model is defined by the vehicle's CoG, where the distance traveled along the centerline is denoted as s , the normal distance from the centerline is denoted as n , and finally ξ defines the angle between the vehicle frame x-axis and the track centerline tangent. The global yaw angle of the vehicle is denoted as ψ , and is defined by the following equation:

$$\psi = \theta + \xi \quad (4.5)$$

The vehicle velocity along the track center line is derived using the curvilinear track model, and the longitudinal velocity u and lateral velocity v of the vehicle:

$$\dot{s} - n\dot{\theta} = u \cos(\xi) - v \sin(\xi) \quad (4.6)$$

Using the relationship between the track curvature \mathcal{C} and θ in Equation (4.4), Equation (4.6) is rewritten giving the change in distance traveled along the track center line, using just vehicle velocities, vehicle deviation from the track center line, and the track curvature:

$$\dot{s} = \frac{u \cos(\xi) - v \sin(\xi)}{1 - n\mathcal{C}} \quad (4.7)$$

To complete the dynamic modeling of the vehicle moving in the track model, equations for \dot{n} and $\dot{\xi}$ are given:

$$\dot{n} = u \sin(\xi) + v \cos(\xi) \quad (4.8)$$

$$\dot{\xi} = \dot{\psi} - \dot{s}\mathcal{C} \quad (4.9)$$

Where \dot{n} is derived geometrically, and $\dot{\xi}$ is derived from differentiating Equation (4.5), and using the aforementioned relationship between track curvature \mathcal{C} and θ .

4.1.1 Generating a track racing line

Casanova [16] outlined how a racing line can be reconstructed using telemetry from a track run and a basic kinematic model of a vehicle. The basic model used longitudinal velocity and lateral acceleration to produce the curvature information for the racing line. The time used to drive the track allowed the integration of the longitudinal velocity to get the distance traveled. The method proved sensitive to the quality of the telemetry gathered, as noise and drift diluted the accuracy of the result. The basic kinematic model also proved limiting.

In Smith [1] a version of this method was used through the online tool OptimumLap [35], where telemetry data containing vehicle longitudinal velocity and lateral acceleration, given for distance traveled, was uploaded to generate the needed tracks. In [1, Chapter 6] a comparison was made between a track made using telemetry, and a track made using GNSS data. An obvious difference was observed, and for this thesis, the tracks are generated using GNSS data.

It is important to note that this is still an approximation of the real track centerline, as both telemetry and GNSS are measured during a competition run, and not a run made to map the track in question. It is therefore important to note that the tracks used in this simulator are close to what the racing line should be. This fact will determine how the constraints are defined in the free trajectory simulator.

Revolve NTNU's 2019 racecar Nova was equipped with the VectorNav VN-300 GNSS-Aided Inertial Navigation System (INS) with Integrated GNSS-Compass [36]. Through the onboard telemetry system, the GNSS data was recorded during every test run and

competition. The recorded GNSS data was extracted, and the geodetic coordinates were transformed to cartesian coordinates in a local North-East-Down (NED) frame using the MathWorks Mapping Toolbox™ in MATLAB [37]. The cartesian points were interpolated using spline interpolation in MATLAB with regards to distance traveled s to achieve the wanted step length Δs .

From the interpolated GNSS derived cartesian points the signed track curvature was derived with following formula:

$$\mathcal{C}(s) = \frac{x'(s)y''(s) - x''(s)y'(s)}{(x'(s)^2 + y'(s)^2)^{\frac{3}{2}}} \quad (4.10)$$

Where x and y refers to the interpolated transformed GNSS points and their prime refer to their first and second-order derivatives evaluated for distance traveled at the given step length. The derivatives were produced using MATLAB to create cubic splines from the points and differentiated using a differentiated function in the MathWorks Curve Fitting Toolbox™ [38].

The track orientation angle θ was approximated using cumulative trapezoidal numerical integration in MATLAB, integrating the track curvature \mathcal{C} with respect to the chosen step length. To get the same global orientation as in the NED-frame, an offset was applied, found using Equation (4.3) on the first track segment:

$$\theta_{off} = \arctan\left(\frac{y'(s_1)}{x'(s_1)}\right) \quad (4.11)$$

where θ_{off} is a scalar offset applied to the rest of the track orientation.

4.1.2 Optimal Control Scheme for smoothing curvature

An important note in the relevant literature is the need for a smooth track curvature profile for an efficient convergence of the MTM OCP. Christ et al. made this points in [3, Chapter 3], and used an approximate spline regression to generate a smooth center line from cartesian coordinates collected using a 2D LiDAR. The approach was outlined in [7]. In [2] and [18], Perantoni and Limebeer outlined a method using a separate OCP in order to smooth the track curvature profile. The method was further discussed in [32, Chapter 8]. This method was used to generate a smooth track curvature profile for the OC LTS presented in this thesis.

An OCP was defined using the change in curvature with regards to the distance traveled along the track center line as an input u :

$$\frac{d\mathcal{C}}{ds} = u \quad (4.12)$$

By integrating Equations (4.4), (4.1), and (4.2), the track orientation and track location in cartesian coordinates is found:

$$\frac{d\theta}{ds} = \mathcal{C} \quad (4.13)$$

$$\frac{dx}{ds} = \cos(\theta) \quad (4.14)$$

$$\frac{dy}{ds} = \sin(\theta) \quad (4.15)$$

A performance index is defined as:

$$\int_{s_0}^{s_f} ((x_c - x)^2 + (y_c - y)^2 + cu^2) ds \quad (4.16)$$

where s_0 and s_f are the start and finish line segments respectively, x_c and y_c are the original, constant track cartesian coordinates, x and y are states in the OCP and resulting track cartesian coordinates, and u is the input. By minimizing the given performance index a smooth track curvature profile is generated. The constant c in the performance index weights the penalty for using the input, effectively regulating how much or how little smoothing of the original track curvature profile occurs. It is chosen by balancing the smoothness of the curvature profile, with the characteristics of a Formula Student AutoX track as defined in the rules [39, Section D6].

In [2, Chapter 2], Perantoni and Limebeer also defined a closure condition for the track and boundary conditions to be enforced in the OCP. This is not needed for the purposes of this work, as the AutoX tracks used in Formula Student competitions are not closed.

Track Smoothing Optimal control problem formulation

The following OCP is defined:

$$\min_{\mathbf{x}(\cdot), u(\cdot)} \int_{s_0}^{s_f} L(\mathbf{x}(s), u(s)) ds \quad (4.17)$$

$$s.t. \quad \frac{d\mathbf{x}}{ds}(s) = \mathbf{F}(\mathbf{x}(s), u(s)), \quad s \in [s_0, s_f] \quad (4.18)$$

$$-0.5 \leq \mathcal{C} \leq 0.5 \quad (4.19)$$

$$-3\pi \leq \theta \leq 3\pi \quad (4.20)$$

$$\begin{bmatrix} x(s_0) - x_c(s_0) \\ y(s_0) - y_c(s_0) \\ x(s_f) - x_c(s_f) \\ y(s_f) - y_c(s_f) \end{bmatrix} = 0 \quad (4.21)$$

Where the performance index is defined as in Equation (4.16), the system dynamics are defined by Equations (4.12), (4.13), (4.14), and (4.15). Two constraints were defined for track curvature \mathcal{C} and track orientation θ to stop the track looping in on itself. The start and end points of the track were both constrained to the same location as the GNSS points defined.

Track Smoothing Transcription

A direct multiple shooting method was deployed to solve the track smoothing OCP. The problem was deemed fairly simple, but direct access to the state trajectory for a good initial guess was desired.

The trajectory was discretized for the chosen step length, creating the distance vector with N steps: s_0, \dots, s_N , where $s_N = s_f$. The states and inputs were discretized for the given distance grid, and the input was set as piecewise constant:

$$\mathbf{x}_k = [\mathcal{C}_k, \theta_k, x_k, y_k]^T \quad (4.22)$$

$$u_k = u(s \in [s_k, s_{k+1}]) \quad (4.23)$$

In order to structure the NLP the optimization variables \mathbf{w} were ordered as follows:

$$\mathbf{w} = \{\mathbf{x}_0, u_0, \dots, \mathbf{x}_{N-1}, u_{N-1}, \mathbf{x}_N\}^T \quad (4.24)$$

The performance index was approximated as the sum of parts as defined by the distance

vector:

$$\int_{s_0}^{s_f} L(\mathbf{x}(s), u(s)) ds \approx \sum_{k=0}^{N-1} (x_{c,k} - x_k)^2 + (y_{c,k} - y_k)^2 + cu_k^2 = \Phi(\mathbf{w}) \quad (4.25)$$

Giving the cost function for the NLP:

$$\min_{\mathbf{w}} \Phi(\mathbf{w}) \quad (4.26)$$

An explicit Runge-Kutta 4 integration scheme was used to simulate the dynamics of the system, integrating the dynamics $\mathbf{F}(\mathbf{x}(s), u(s))$ with the integrator function $\mathbf{f}(\mathbf{x}_k, u_k)$ for the distance interval $[s_k, s_{k+1}]$. The shooting gaps and the initial and terminal conditions of the OCP were handled by the equality constraint as follow:

$$\mathbf{g}(\mathbf{w}) = \begin{bmatrix} x_0 - x_{c,0} \\ y_0 - y_{c,0} \\ \mathbf{f}(\mathbf{x}_0, u_0) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}(\mathbf{x}_{N-1}, u_{N-1}) - \mathbf{x}_N \\ x_N - x_{c,N} \\ y_N - y_{c,N} \end{bmatrix} = 0 \quad (4.27)$$

The final NLP was formulated as follows:

$$\min_{\mathbf{w}} \Phi(\mathbf{w}) \quad (4.28)$$

$$s.t. \quad \mathbf{g}(\mathbf{w}) = 0 \quad (4.29)$$

$$-0.5 \leq \mathcal{C} \leq 0.5 \quad (4.30)$$

$$-3\pi \leq \theta \leq 3\pi \quad (4.31)$$

The NLP was formulated in MATLAB, using CasADi, and solved using the Interior-point method with Ipopt. Tracks from both the 2019 FSG AutoX and the 2019 FSA AutoX were generated and ran through the curvature smoothing scheme. The resulting tracks are shown in Section 5.2.

4.2 Initial small scale implementation

A preliminary study was conducted to create a Minimum Viable Product (MVP) of the simulator as outlined in the problem statement. A single-track vehicle model, as shown in Sub-section 3.1.3, was deployed, with a simplified Pacejka tire model, no load transfer, and without an aerodynamic package. The track model was modeled using the curvilinear description, but only small track segments, or "snippets" were used, designed to test the free trajectory OC based LTS, and to keep the problem size small.

Identifier	Description	Figure
Left hand turn	30m straight, 90deg left hand turn, 30m straight	1
Hairpin turn	30m straight, 180deg left hand turn, 30m straight	2
Lane-shift maneuver	30m straight, 90deg left turn, 90deg right turn, 30m straight	2

Table 4.1: Track snippets used in preliminary study

The MVP used a direct multiple shooting transcription method, and was implemented in MATLAB, using CasADi, and solved with Ipopt. The implementation is not outlined in this thesis, as the vehicle modeling was a simplified version of the one presented in this thesis, and the transcription was very similar to the track smoothing scheme presented in Subsection 4.1.2. The main objective was to get familiar with the CasADi syntax, the transcription of the LTS OCP, and to experiment with the performance index and constraints. This preliminary work came as a consequence of the struggles experienced when implementing the simulator as described in [1]. An important point made by the author was to build complexity slowly, making sure to produce a working implementation every time before adding more complexity. The free trajectory optimization results from this preliminary study are presented in Section 5.3.

4.3 Shift from time to spacial dimension

The switch from using time t as the independent variable to using progression along the track centerline s as the independent variable was made. This switch in the independent variable is well established in the relevant literature. Using s and not t as the independent variable reduces the problem size as s is removed from the state vector, and it allows the lap time, that is to be minimized, to be a dependant variable. This switch holds if the vehicle only travels in the positive s direction, which yields a unique relationship between s and t . The following conversion factor, derived from the inverse of Equation (4.7), allows the

transformation of time dependant dynamics to spacial dependant dynamics:

$$S_f = \left(\frac{ds}{dt} \right)^{-1} = \frac{1 - n\mathcal{C}}{u \cos(\xi) - v \sin(\xi)} \quad (4.32)$$

Using the above conversion factor the vehicles spacial dynamics, Equations (4.8) and (4.9), are rewritten with s as the independent variable:

$$\frac{dn}{ds} = S_f(u \sin(\xi) + v \cos(\xi)) \quad (4.33)$$

$$\frac{d\xi}{ds} = S_f \dot{\psi} - \mathcal{C} \quad (4.34)$$

4.4 Vehicle modeling

A double-track vehicle model was implemented as presented in Sub-section 3.1.3 and shown in Figure 3.6. The model uses wheel angle and tire slip ratios as inputs. Slip ratios were favored over wheel rotation, as it allowed a more direct path to the longitudinal forces produced by the tires, and it eliminates the need for modeling the wheel rations. The vehicle model has the Pacejka Magic Formula 5.2 tire model supplied by Continental (Appendix B.1), longitudinal and lateral load transfer, and a basic aerodynamic model. The limitations of the powertrain are handled in the constraints of the final OCP formulation.

4.4.1 Tire velocities

The longitudinal and lateral velocities of the four tires were modeled. The vehicle wheel-base and track width, with the vehicle longitudinal, lateral, and angular (yaw rate) velocities yields:

$$\mathbf{u}_{t,vf} = \mathbf{u} + 0.5 \begin{bmatrix} -\tau_f \\ \tau_f \\ -\tau_r \\ \tau_r \end{bmatrix} \dot{\psi} \quad (4.35)$$

$$\mathbf{v}_{t,vf} = \mathbf{v} + \begin{bmatrix} l_f \\ l_f \\ -l_r \\ -l_r \end{bmatrix} \dot{\psi} \quad (4.36)$$

Where $\mathbf{u}_{t,vf}$ and $\mathbf{v}_{t,vf}$ are 4×1 vectors containing the longitudinal and lateral velocities of each of the four tires in the vehicle frame, \mathbf{u} and \mathbf{v} are 4×1 vectors with the longitudinal and lateral velocity contribution from the vehicle CoG for each tire, $\dot{\psi}$ is the vehicle yaw rate, τ_f and τ_r are the front and rear track widths, and finally l_f and l_r are the distance from the vehicle CoG to the front and rear axles.

The longitudinal velocity of each tire in the tire frame was needed for the powertrain constraints. The front wheels steer, giving the following tire longitudinal velocity in the tire frame:

$$\mathbf{u}_{t,tf} = \begin{bmatrix} u_{fl,vf} \cos(\delta) \\ u_{fr,vf} \cos(\delta) \\ u_{rl,vf} \\ u_{rr,vf} \end{bmatrix} + \begin{bmatrix} v_{fl,vf} \sin(\delta) \\ v_{fr,vf} \sin(\delta) \\ 0 \\ 0 \end{bmatrix} \quad (4.37)$$

Where $\mathbf{u}_{t,tf}$ is a 4×1 vector containing the longitudinal velocity of each of the four tires in the tire frame, δ is the wheel angle from steering, and the subscripts *fl*, *fr*, *rl*, and *rr*, denote the specific wheel on the vehicle: Front left, front right, rear left, rear right. Notice that only the front tires are steered, hence only the tire longitudinal velocity in the vehicle frame is used for the rear tires.

4.4.2 Tire slip angles

The slip angles were computed for each tire using the longitudinal and lateral tire velocities in the vehicle frame. The front tires are steered, giving the contribution of the wheel angle:

$$\alpha_{fl} = \delta - \arctan\left(\frac{v_{fl,vf}}{u_{fl,vf}}\right) \quad (4.38)$$

$$\alpha_{fr} = \delta - \arctan\left(\frac{v_{fr,vf}}{u_{fr,vf}}\right) \quad (4.39)$$

$$\alpha_{rl} = -\arctan\left(\frac{v_{rl,vf}}{u_{rl,vf}}\right) \quad (4.40)$$

$$\alpha_{rr} = -\arctan\left(\frac{v_{rr,vf}}{u_{rr,vf}}\right) \quad (4.41)$$

Where α_{ij} is the slip angle for the tire defined by the subscript *ij*: *fl*, *fr*, *rl*, or *rr*.

4.4.3 Vertical tire forces

The tire vertical force has three main contributions: Vehicle mass, aerodynamic downforce, and load transfer. The vehicle mass is distributed across the four tires, longitudinally and laterally. An even lateral distribution is assumed, but the front to rear, longitudinal distribution is modeled. It is denoted through the mass fraction on the rear axle, commonly referred to as weight distribution rear wd_r . The same is assumed for the aerodynamic downforce. Its distribution is denoted ad_r . The longitudinal and lateral load transfer are both inputs to the model. The algebraic loop that arises from modeling load transfer is broken in the constraints of the OCP.

The tire vertical forces are modeled as follows:

$$\hat{F}_{z,fl} = \frac{1}{2}(1 - wd_r)mg + \frac{1}{2}(1 - ad_r)F_L - \frac{1}{2}\Gamma_x - \frac{1}{2}\Gamma_y \quad (4.42)$$

$$\hat{F}_{z,fr} = \frac{1}{2}(1 - wd_r)mg + \frac{1}{2}(1 - ad_r)F_L - \frac{1}{2}\Gamma_x + \frac{1}{2}\Gamma_y \quad (4.43)$$

$$\hat{F}_{z,rl} = \frac{1}{2}wd_rmg + \frac{1}{2}ad_rF_L + \frac{1}{2}\Gamma_x - \frac{1}{2}\Gamma_y \quad (4.44)$$

$$\hat{F}_{z,rr} = \frac{1}{2}wd_rmg + \frac{1}{2}ad_rF_L + \frac{1}{2}\Gamma_x + \frac{1}{2}\Gamma_y \quad (4.45)$$

Where $\hat{F}_{z,ij}$ is the sum of vertical load for the tire defined by the subscript ij : fl , fr , rl , or rr . m is the vehicle mass, g is the gravitational constant, and Γ_x and Γ_y are the longitudinal and lateral load transfers respectively. The aerodynamic downforce F_L is defined in Equation (3.6).

The vertical forces are never less than zero on the real vehicle. To model this, the following redefinition of the vertical forces is made:

$$\mathbf{F}_z = \max(\hat{\mathbf{F}}_z, 0) \quad (4.46)$$

Where \mathbf{F}_z is a 4×1 vector containing the four tire vertical forces, and $\hat{\mathbf{F}}_z$ is a 4×1 vector containing the four sums defined in Equations (4.42)-(4.45).

4.4.4 Longitudinal and lateral tire forces

The longitudinal and lateral tire forces are derived from the Pacejka Magic Formula 5.2 (Appendix B.1). The model is supplied by the tire manufacturer and provides the combined longitudinal and lateral forces produced by the tires. The tire models are used as

shown below:

$$F_{x,ij} = c_x \text{magicCombLong}(\alpha_{ij}, \kappa_{ij}, \gamma_{ij}, F_{z,ij}) \quad (4.47)$$

$$F_{y,ij} = c_y \text{magicCombLat}(\alpha_{ij}, \kappa_{ij}, \gamma_{ij}, F_{z,ij}) \quad (4.48)$$

Both the longitudinal and the lateral tire model take slip angle α , slip ratio κ , inclination angle γ , and tire vertical load F_z as inputs. As these are experimentally derived models, using data fitted to the model derived from physical testing, they overestimate the grip produced by the tires. This is handled by scaling the output of the model by comparing the longitudinal and lateral acceleration of the vehicle simulated, to test data from the actual racecar. The constants c_x and c_y are the longitudinal and lateral scaling factors respectively. The same approach was used in the preliminary project thesis for this project [1].

4.4.5 Dynamic model

The vehicle is modeled as a rigid body, with four tires producing longitudinal and lateral forces. Using the force balance in the longitudinal and lateral direction, and the moment balance about the vertical axis, the equations of motion for the vehicle are produced. The sum of forces and moments on the vehicle CoG are as follows:

$$F_x = (F_{x,fl} + F_{x,fr}) \cos(\delta) - (F_{y,fl} + F_{y,fr}) \sin(\delta) + F_{x,rl} + F_{x,rr} \quad (4.49)$$

$$F_y = (F_{x,fl} + F_{x,fr}) \sin(\delta) + (F_{y,fl} + F_{y,fr}) \cos(\delta) + F_{y,rl} + F_{y,rr} \quad (4.50)$$

$$\begin{aligned} M_z = & l_f((F_{x,fl} + F_{x,fr}) \sin(\delta) + (F_{y,fl} + F_{y,fr}) \cos(\delta)) \\ & + \frac{\tau_f}{2}((F_{x,fr} - F_{x,fl}) \cos(\delta) + (F_{y,fl} - F_{y,fr}) \sin(\delta)) \\ & + \frac{\tau_r}{2}(F_{x,rr} - F_{x,rl}) - l_r(F_{y,rl} + F_{y,rr}) \end{aligned} \quad (4.51)$$

Where F_x is the sum of longitudinal forces produced by the tires on the vehicle CoG, F_y is the sum of lateral forces produced by the tires on the vehicle CoG, and M_z is the sum of moments about the vehicle vertical axis as defined by the vehicle CoG.

The vehicle equations of motion are produced:

$$\dot{u} = \frac{1}{m} (F_x - F_D) + v \dot{\psi} \quad (4.52)$$

$$\dot{v} = \frac{1}{m} F_y - u \dot{\psi} \quad (4.53)$$

$$\ddot{\psi} = \frac{1}{I_z} M_z \quad (4.54)$$

Where \dot{u} is the change in longitudinal velocity, \dot{v} is the change in lateral velocity, $\ddot{\psi}$ is the change in vehicle yaw rate, and I_z is the vehicle yaw moment of inertia. The aerodynamic drag F_D is modeled as defined in Equation (3.7).

The equations of motion are dependant on time t as the independent variable. Using the conversion factor defined in Equation (4.32), they are expressed in terms of progression along the track centerline:

$$\frac{du}{ds} = S_f \left(\frac{1}{m} (F_x - F_D) + v \dot{\psi} \right) \quad (4.55)$$

$$\frac{dv}{ds} = S_f \left(\frac{1}{m} F_y - u \dot{\psi} \right) \quad (4.56)$$

$$\frac{d\dot{\psi}}{ds} = S_f \frac{1}{I_z} M_z \quad (4.57)$$

Including Equations (4.33) and (4.34), this completes the dynamic modeling part of the simulator.

4.4.6 Powertrain

The four hub-mounted motors on the vehicle modeled, have torque and rotation limits, and the Formula Student rules defined a maximum allowed power consumption for the powertrain. These modeling characteristics were defined as constraints in the final OCP formulation.

Power use

The Formula student rules specify a maximum power use limitation by that powertrain of $80kW$ [39, Rule EV 2.2.1]. This limitation is met through an inequality constraint applied to an approximation of the power use. The approximation uses the definition of slip ratio as defined in Equation (3.2) to approximate the power used. The power used by a torque

applied to a rotating object is given as:

$$P = T \omega \quad (4.58)$$

Where P is the power used, T is the torque doing the work, and ω is the angular velocity. For a given tire ij on the vehicle, the torque is the longitudinal force produced by the tire multiplied by the radius of the tire:

$$P_{ij} = F_{x,ij} r_l \omega_{ij} \quad (4.59)$$

By combining Equations (4.59) and (3.2) the following approximation of the power used for a given tire ij on the vehicle is modeled:

$$P_{ij} = F_{x,ij} (1 + \kappa_{ij}) u_{ij,tf} \quad (4.60)$$

Taking the sum of this term for all four tires yields the following inequality constraint:

$$P_{fl} + P_{fr} + P_{rl} + P_{rr} - P_{max} = P_t - P_m \leq 0 \quad (4.61)$$

Where P_t is the total power used by the four tires, and P_m is the maximum allowed power consumption. Only an upper bound is needed as it is assumed that braking with the mechanical brakes is limited by the tires' maximum grip.

Torque limit

The manufacturer of the motors specify a maximum torque T_m that the motors are able to produce. This is handled by the following inequality constraint using the tire longitudinal force $F_{x,ij}$, the tire radius r_l , and powertrain gear ratio n :

$$\frac{F_{x,ij} r_l}{n} - T_m \leq 0 \quad (4.62)$$

The maximum grip of the tires are the lower limit for the max torque in braking ($F_{x,ij} < 0$), as the vehicle is equipped with mechanical brakes.

Rotation limit

Much like for torque, the motors can only spin so fast. Using the definition of slip ratio κ , the tire longitudinal velocity in the tire frame $u_{ij,tf}$, and gear ratio n a maximum rotation

limit is defined:

$$0 \leq \frac{u_{ij,t_f}}{r_l} (1 + \kappa_{ij}) n \leq \omega_m \quad (4.63)$$

Where ω_m is the upper rotation limit of the motors. The lower limit is there to stop the optimization from spinning the tires in reverse.

4.5 Optimal Control Problem formulation

The OCP was defined for the vehicle and track model as defined. The goal is to minimize the time used to drive the racecar through the track. The dynamics of the system are defined in the Differential-Algebraic system of Equations (DAE) as defined by Equations (4.33), (4.34), (4.55), (4.56), and (4.57). The OCP is subject to the constraints as defined.

4.5.1 General Optimal Control Formulation

A general OCP formulation is defined:

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_{s_0}^{s_f} L(s, \mathbf{x}(s), \mathbf{u}(s)) ds \quad (4.64)$$

$$s.t. \quad \frac{d\mathbf{x}}{ds}(s) = \mathbf{F}(s, \mathbf{x}(s), \mathbf{u}(s)), \quad s \in [s_0, s_f] \quad (4.65)$$

$$g((s, \mathbf{x}(s), \mathbf{u}(s))) = 0 \quad (4.66)$$

$$h((s, \mathbf{x}(s), \mathbf{u}(s))) \leq 0 \quad (4.67)$$

The Lagrange cost L defines the time used to travel through the track, the system dynamics are defined by \mathbf{F} , and g and h are the equality and inequality constraints respectively. s is the distance traveled along the track centerline between the track start s_0 and track finish s_f , \mathbf{x} is the state vector, and \mathbf{u} is the input vector to the system.

4.5.2 System states and inputs

During the preliminary study, oscillatory tendencies were observed in the vehicle inputs, and as a result, the vehicle states. A cost was placed on the inputs and the change of the inputs, but this did not give a satisfactory result, nor a stable OCP formulation as tracks were varied. This method also ruins the advantageous structure of the NLP that is possible to achieve using a direct transcription method [33].

Instead, an additional set of dynamic equations were introduced, as the inputs to the vehicle

model were made part of the state vector, and their rates were used as inputs in the OCP formulation. This gives direct access to the rate of change of each vehicle input, as they are possible to box with the constraints in the OCP inputs. It also keeps the aforementioned advantageous structure of the NLP intact. The downside is a significantly larger number of optimization variables, which hurts the CPU time. The following additional dynamic equations, with the change in the independent variable, were included in the vehicle model:

$$\frac{d\delta}{ds} = S_f \dot{\delta} \quad (4.68)$$

$$\frac{d\kappa_{fl}}{ds} = S_f \dot{\kappa}_{fl} \quad (4.69)$$

$$\frac{d\kappa_{fr}}{ds} = S_f \dot{\kappa}_{fr} \quad (4.70)$$

$$\frac{d\kappa_{rl}}{ds} = S_f \dot{\kappa}_{rl} \quad (4.71)$$

$$\frac{d\kappa_{rr}}{ds} = S_f \dot{\kappa}_{rr} \quad (4.72)$$

$$\frac{d\Gamma_x}{ds} = S_f \dot{\Gamma}_x \quad (4.73)$$

$$\frac{d\Gamma_y}{ds} = S_f \dot{\Gamma}_y \quad (4.74)$$

Scaling

An effort was made to keep the order of magnitude of the different optimization variables similar. This was done to keep the significance of each variable similar and help the NLP solver. Load transfer and load transfer rate were both significantly larger than the rest of the variables in the problem. A scaling factor was introduced, effectively changing the unit from *Newton (N)* to *Kilo Newton (kN)*. This ensured a similar magnitude across the different optimization variables.

The states and inputs as used in the OCP formulation are listed below in Table 4.2:

Identifier	Type	Description	Symbol	Unit
x_1	State	Normal deviation from track center line	n	m
x_2	State	Angle deviation from track center line	ξ	rad
x_3	State	Longitudinal velocity	u	m/s
x_4	State	Lateral velocity	v	m/s
x_5	State	Yaw rate	$\dot{\psi}$	rad/s
x_6	State	Wheel angle	δ	rad
x_7	State	Front left tire slip ratio	κ_{fl}	-
x_8	State	Front right tire slip ratio	κ_{fr}	-
x_9	State	Rear left tire slip ratio	κ_{rl}	-
x_{10}	State	Rear right tire slip ratio	κ_{rr}	-
x_{11}	State	Longitudinal load transfer	Γ_x	kN
x_{12}	State	Lateral load transfer	Γ_y	kN
u_1	Input	Wheel angle rate	$\dot{\delta}$	rad/s
u_2	Input	Front left tire slip ratio rate	$\dot{\kappa}_{fl}$	1/s
u_3	Input	Front right tire slip ratio rate	$\dot{\kappa}_{fr}$	1/s
u_4	Input	Rear left tire slip ratio rate	$\dot{\kappa}_{rl}$	1/s
u_5	Input	Rear right tire slip ratio rate	$\dot{\kappa}_{rr}$	1/s
u_6	Input	Longitudinal load transfer rate	$\dot{\Gamma}_x$	kN/s
u_7	Input	Lateral load transfer rate	$\dot{\Gamma}_y$	kN/s

Table 4.2: States and inputs used in the OCP

4.5.3 Performance index

The performance index is the lap time used to travel through the track. As time t is removed from the problem formulation, an equivalence is made between minimizing the lap time and maximizing the progression along the track centerline. As the NLP solver used is a minimizer, the inverse of track progression becomes the performance index. This is written as follow:

$$\min \int_{t_0}^{t_f} dt = \min \int_{s_0}^{s_f} \left(\frac{ds}{dt} \right)^{-1} ds = \min \int_{s_0}^{s_f} S_f ds \quad (4.75)$$

The Lagrange cost is written as follows:

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_{s_0}^{s_f} L(s, \mathbf{x}(s), \mathbf{u}(s)) ds = \min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_{s_0}^{s_f} S_f ds \quad (4.76)$$

4.5.4 Constraints

Load transfer

An algebraic loop is present in the vehicle model. Longitudinal and lateral load transfer affects the vertical load on the tires, which in turn affects the grip produced by the tires. The grip is what accelerates the vehicle, and load transfer is a result of that acceleration. This was discussed in the theory in Sub-section 3.1.2. Two equality constraints are defined and passed to the NLP solver to break the aforementioned algebraic loop:

$$\Gamma_x - F_x \frac{h_{cog}}{wb} = 0 \quad (4.77)$$

$$\Gamma_y - F_y \frac{2h_{cog}}{\tau_f + \tau_r} = 0 \quad (4.78)$$

Powertrain

The powertrain constraints are written as inequality constraints. The rotation limit of the motors is a boxed constraint as the motors are only allowed to spin in the positive direction, but are also upper bounded by the maximum allowed rotation.

$$P_{tot} - P_m \leq 0 \quad (4.79)$$

$$\frac{F_{x,ij} r_l}{n} - T_{max} \leq 0 \quad (4.80)$$

$$0 \leq \frac{u_{ij,tj}}{r_l} (1 + \kappa_{ij}) n \leq \omega_m \quad (4.81)$$

Note that the torque and rotation limits are applied individually to all four motors.

Slip angle

The tire model used defines a maximum slip angle for which it is accurate. This is handled with a box constraint on the calculated slip angle:

$$-\alpha_m \leq \alpha_{ij} \leq \alpha_m \quad (4.82)$$

State and input bounds

There are several bound placed on the states and inputs of the OCP. There are limits on how much the vehicle is allowed to deviate from the centerline, defining the left and right

sides of the modeled track. To make the conversion from t to s hold, the vehicle is only allowed to deviate $\pm 90deg$ from the track centerline, and the longitudinal velocity is limited to be positive. The steering is mechanically limited, and like the slip angle, the slip ratio is upper and lower bound due to the tire model working range. The upper and lower bounds on the inputs were set through trial and error, balancing the authority of the vehicle controls with smooth vehicle states, i.e. limiting the aforementioned oscillations observed during the preliminary study. These bounds are summarized below:

$$n_l \leq n \leq n_r \quad (4.83)$$

$$-\frac{\pi}{2} \leq \xi \leq \frac{\pi}{2} \quad (4.84)$$

$$1 \leq u \quad (4.85)$$

$$-\delta_m \leq n \leq \delta_m \quad (4.86)$$

$$\kappa_{ij,lb} \leq \kappa_{ij} \leq \kappa_{ij,ub} \quad (4.87)$$

$$\dot{\kappa}_{ij,lb} \leq \dot{\kappa}_{ij} \leq \dot{\kappa}_{ij,ub} \quad (4.88)$$

$$\dot{\Gamma}_{x,lb} \leq \dot{\Gamma}_x \leq \dot{\Gamma}_{x,ub} \quad (4.89)$$

$$\dot{\Gamma}_{y,lb} \leq \dot{\Gamma}_y \leq \dot{\Gamma}_{y,ub} \quad (4.90)$$

4.6 Transcribing the OCP into an NLP

A direct orthogonal collocation method was used to transcribe the OCP into an NLP. In this transcription, s and not t is used as the independent variable. The transcription method was adopted from the work done by Christ et al. [3].

4.6.1 Collocation method

The track center line was discretized for the chosen uniform step length Δs , creating the distance vector with N step: s_0, \dots, s_N , where $s_N = s_f$. The states \mathbf{x} and inputs \mathbf{u} were discretized on the distance vector as \mathbf{x}_k and \mathbf{u}_k , using piece wise constant inputs:

$$\mathbf{u}_k = \mathbf{u}(s \in [s_k, s_{k+1}]) \quad (4.91)$$

In the direct collocation method every local state trajectory $\mathbf{x}(s)$ on the interval $[s_k, s_{k+1}]$ are approximated by a polynomial $\mathbf{P}_k(s, \boldsymbol{\theta}_k)$, built from Lagrange polynomials $p_{k,i}$. The polynomial is fitted to the trajectory with the multiplier $\boldsymbol{\theta}_{k,i}$, giving the definition of the

polynomial $\mathbf{P}_k(s, \boldsymbol{\theta}_k)$ as follows:

$$\mathbf{P}_k(s, \boldsymbol{\theta}_k) = \sum_{i=0}^d \boldsymbol{\theta}_{k,i} p_{k,i}(s) \quad (4.92)$$

Where the order of the polynomial is given by d . The Lagrange polynomials $p_{k,i}(s)$ are calculated for the collocation points $s_{k,j}$ on the local trajectory $[s_k, s_{k+1}]$. They are chosen corresponding to the zeros of the orthogonal Legendre polynomials. This gives a Legendre collocation method:

$$p_{k,i}(s) = \prod_{j=0, j \neq i}^d \frac{s - s_{k,j}}{s_{k,i} - s_{k,j}} \quad (4.93)$$

The order of the Lagrange polynomial was chosen to be $d = 3$, giving an accurate integration of the state trajectory, while keeping the problem size as small as possible. The multipliers $\boldsymbol{\theta}_{k,i}$ are passed to the NLP solver, using the following constraints to enforce the polynomial approximation of the system dynamics:

$$\mathbf{P}_k(s_k, \boldsymbol{\theta}_k) - \mathbf{x}_k = 0, \quad k = 0, \dots, N \quad (4.94)$$

$$\frac{\partial}{\partial t} \mathbf{P}_k(s_{k,i}, \boldsymbol{\theta}_k) - \mathbf{F}(\mathbf{P}_k(s_{k,i}, \boldsymbol{\theta}_k), \mathbf{u}_k) = 0, \quad k = 0, \dots, N, i = 1, \dots, d \quad (4.95)$$

As with the multiple shooting method, the shooting gaps between the local trajectories are managed with the constraint between the end of the Lagrange polynomial and the starting point of the next local trajectory:

$$\mathbf{P}_k(s_{k+1}, \boldsymbol{\theta}_k) - \mathbf{x}_{k+1} = 0, \quad k = 0, \dots, N - 1 \quad (4.96)$$

The three equality constraint defined in Equations (4.94), (4.95), and (4.96), are passed to the NLP solver and solved for $\boldsymbol{\theta}_k$. The optimization variables for the resulting NLP are added to a single, large vector \mathbf{w} as shown:

$$\mathbf{w} = \{\boldsymbol{\theta}_{0,0}, \dots, \boldsymbol{\theta}_{0,d}, \mathbf{u}_0, \dots, \boldsymbol{\theta}_{N-1,0}, \dots, \boldsymbol{\theta}_{N-1,d}, \mathbf{u}_{N-1}\}^T \quad (4.97)$$

Where the states extracted from the simulation are:

$$\mathbf{x}_k = \boldsymbol{\theta}_{k,0}, \quad k = 0, \dots, N - 1 \quad (4.98)$$

4.6.2 Cost function

The Lagrange cost of the original OCP was given in Equation (4.76):

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_{s_0}^{s_f} L(s, \mathbf{x}(s), \mathbf{u}(s)) ds$$

Christ et al. [3] used a numerical integration scheme to approximate the cost of their OCP. This approximation is deployed to define the cost for the NLP:

$$\int_{s_0}^{s_f} L(s, \mathbf{x}(s), \mathbf{u}(s)) ds \approx \sum_{k=0}^{N-1} \Delta s \sum_{r=0}^d B_r L(\boldsymbol{\theta}_{k,r}, \mathbf{u}_k)$$

Where Δs is the step length, L is the performance index, and $B_r = \int_0^1 p_r(\tau) d\tau$ is the integral of the Lagrange basis polynomials. The cost is extracted as the resulting lap time for the simulation.

4.6.3 Constraints

The constraints defined in the OCP formulation in Sub-section 4.5.4, are enforced for the start point of each local trajectory $[s_k, s_{k+1}]$, meaning for $\boldsymbol{\theta}_{k,0}$ and \mathbf{u}_k . These constraint contain all the path constraints, the physical limitations of the vehicle, and the bounds on the states and inputs:

$$g(\boldsymbol{\theta}_{k,0}, \mathbf{u}_k) = 0 \quad (4.99)$$

$$h(\boldsymbol{\theta}_{k,0}, \mathbf{u}_k) \leq 0 \quad (4.100)$$

Where g are the equality constraints and h are the inequality constraints.

4.6.4 Measures for more stable convergence

Interpolation of the track curvature

The track curvature information is piecewise constant. As discussed the solver performs best if the curvature trajectory is smooth. Linear interpolation is deployed on the collocation points, to give a smoother curvature input to the NLP solver. On every collocation

point the curvature is interpolated as shown below:

$$C_{k,i} = C_{k,0} + \frac{C_{k+1,0} - C_{k,0}}{\Delta s} (s_{k,i} - s_{k,0}), \quad (4.101)$$

$$\text{where } k = 0, \dots, N-1, i = 1, \dots, d$$

Smoothing functions

The vehicle model contains several non-smooth and discontinuous functions. To improve the performance and stability of the NLP solver, these functions were approximated using smoothing functions. The smoothing functions that were applied are listed below:

$$\max(x, 0) \approx \frac{x + \sqrt{x^2 + \epsilon}}{2} \quad (4.102)$$

$$\min(x, 0) \approx -\frac{-x + \sqrt{x^2 + \epsilon}}{2} \quad (4.103)$$

$$\text{sign}(x) \approx \frac{x}{\sqrt{x^2 + \epsilon}} \quad (4.104)$$

$$\text{abs}(x) \approx \sqrt{x^2 + \epsilon} \quad (4.105)$$

These functions are defined in $x = 0$ and are differentiable. The approximation accuracy is dependant on the magnitude of the constant ϵ . The smaller it is, the more accurate, but the less smooth the approximation becomes. Perantoni and Limebeer [2] listed the constant to be: $10^{-4} \leq \epsilon \leq 10^{-2}$, so $\epsilon = 10^{-3}$ was chosen.

4.7 NLP formulation and solving

The NLP was formulated in MATLAB, using CasADi, and solved using the Interior-point method with Ipopt. CasADi provides the solver Ipopt with all first and second-order derivatives in analytic form, enhancing the convergence of the NLP. The simulation results are presented in Chapter 5.

Results

5.1 Step length investigation

Perantoni and Limebeer [2] showed the CPU-time, or solution time, to increase with the step length Δs with the approximate formula:

$$t_{NLP} \approx 5 \times 10^{-3} N^{1.5} \quad (5.1)$$

Where N is the number of steps in the distance vector $[s_0, s_f]$.

A similar analysis was conducted using the 2019 FSG AutoX track [24], to determine what step length to use in the simulations presented in this thesis. The results from this analysis are presented in Figure 5.1. The analysis showed the CPU-time to be higher than what Perantoni and Limebeer [2] predicted, and the revised approximate CPU-time function became:

$$t_{NLP} \approx 4.8 \times 10^{-2} N^{1.5} \quad (5.2)$$

The analysis also showed that a step length of $\Delta s = 1\text{m}$ gave a good balance between computational cost and accuracy in the simulated lap time. This step length is used in the simulations presented in this thesis.

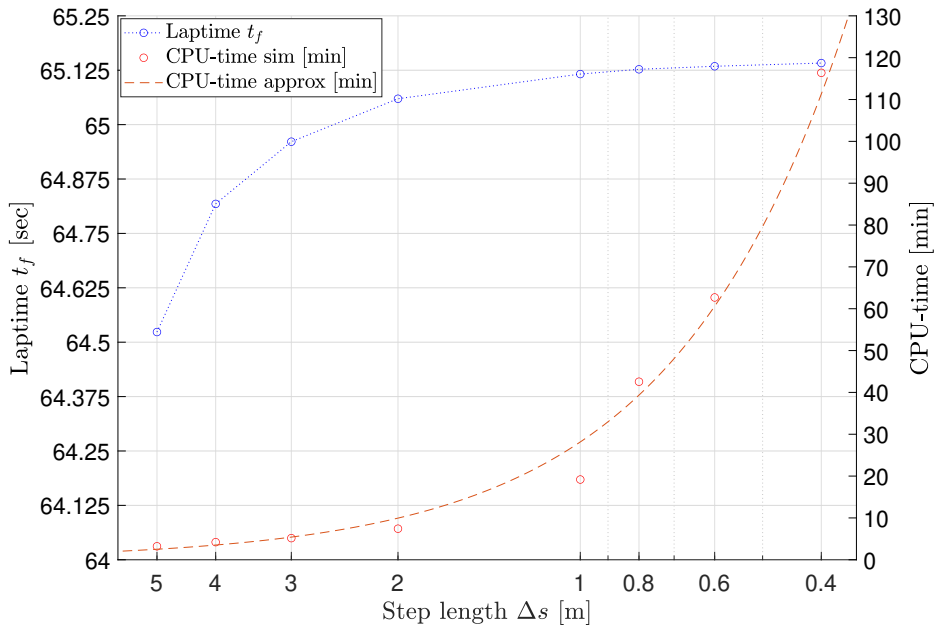
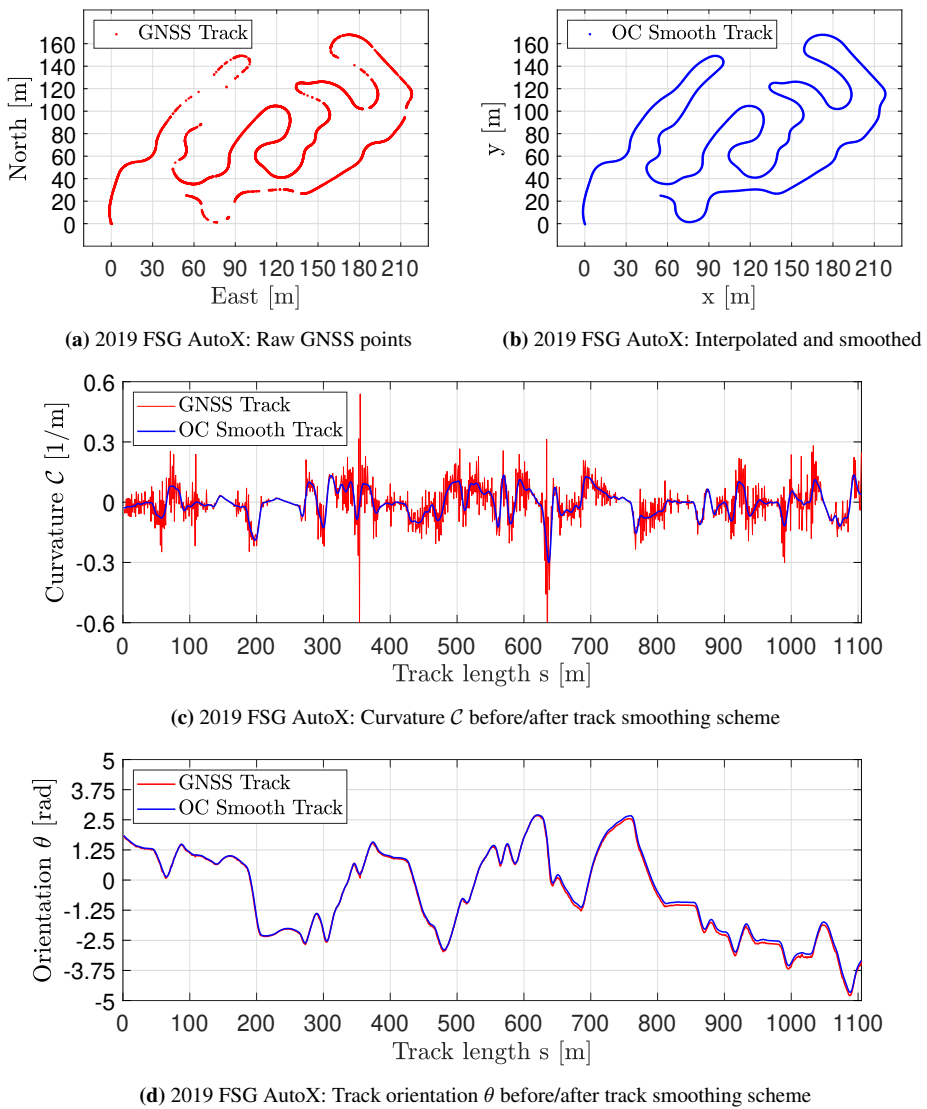


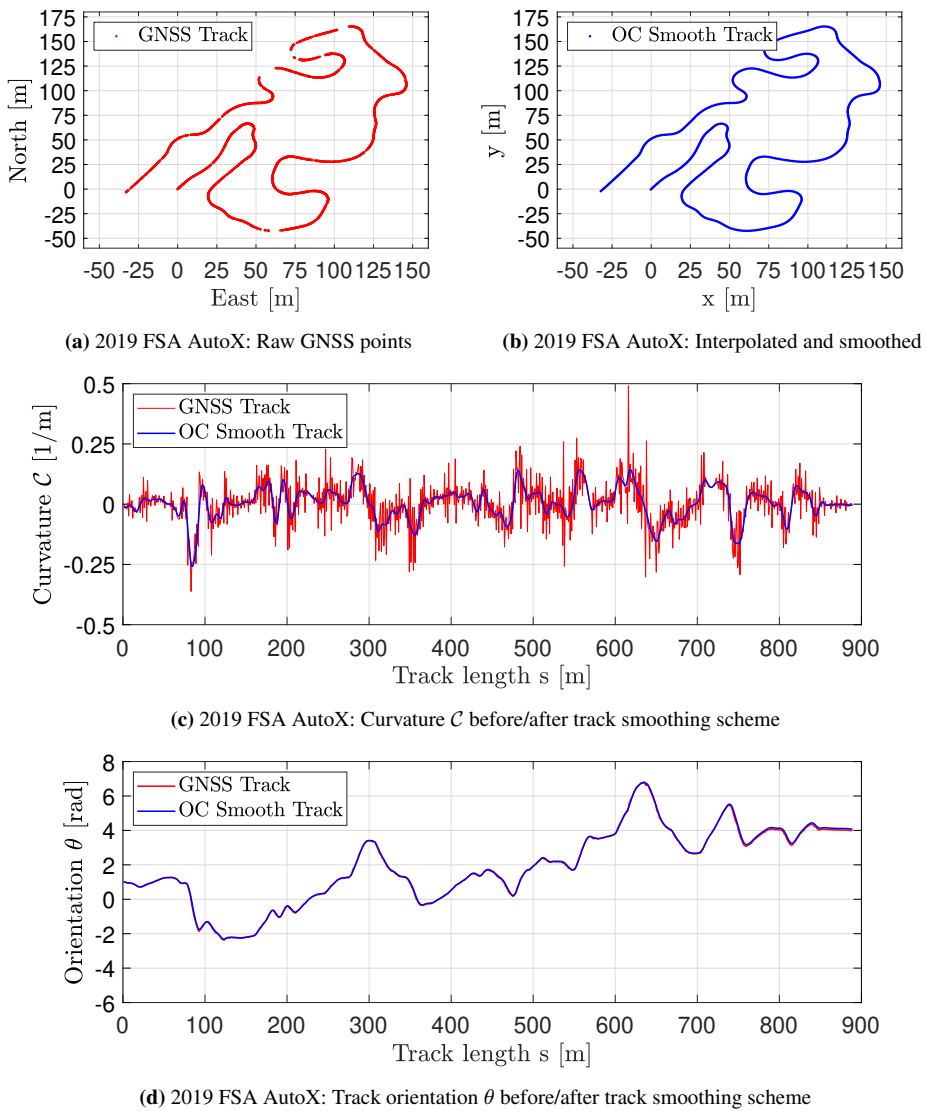
Figure 5.1: Step length investigation: The resulting lap time, and CPU-time for a set of step lengths is shown on a semi-log scale. The approximate CPU-time function is also shown.

5.2 Track models and smoothing track curvature

The track models used in this thesis are presented. The GNSS points from telemetry taken during the 2019 FSG AutoX [24], and the 2019 FSA AutoX [25], were transformed, interpolated, and the final track center line was produced with the track smoothing scheme presented in Subsection 4.1.2. The CPU-time for the whole scheme was $t_{CPU} < 5$ sec.

The constant $c = 20$ was used in the smoothing scheme. This was determined by looking at the maximum track curvature after smoothing, and comparing it to the minimum track corner radius listed in the Formula Student regulations [39, Section D6]: $C_{max} \approx 0.3$. The gaps in the raw GNSS data are due to package loss in the telemetry system that occurred during the competition. Notice that the curvature is significantly smoother after the track smoothing scheme, but that the position data and track orientation is preserved.

Track model: 2019 FSG Autox**Figure 5.2:** 2019 FSG AutoX: Results from track interpolation and smoothing scheme

Track model: 2019 FSA Autox**Figure 5.3:** 2019 FSA AutoX: Results from track interpolation and smoothing scheme

5.3 Preliminary simulation results

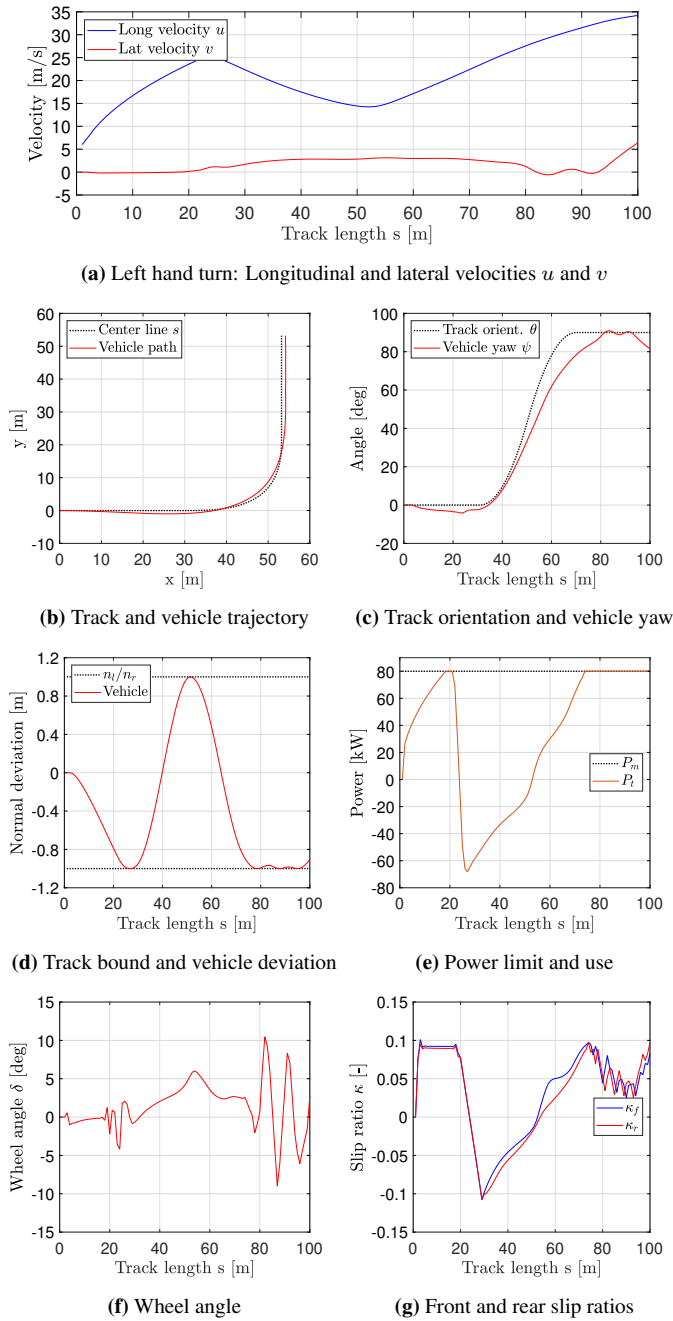


Figure 5.4: Left hand turn: Results from the preliminary study

Out of the three manoeuvres listed in Table 4.1, only the left hand turn is presented. This is to save on space in this thesis, and because the important aspects are all shown in the results presented. The allowable normal deviation from the track center line is set to $n = 1\text{m}$. The power limitation is also enforced at $P_m = 80000\text{W}$. The step length was: $\Delta s = 1\text{m}$, giving 3473 optimization variables. The CPU-time for the simulation was $t_{CPU} = 13\text{sec}$.

The longitudinal and lateral velocity profiles are shown in Sub-figure 5.4a. It shows the starting longitudinal velocity $u = 6\text{m/s}$, the lateral velocity that occurs in turns, and an increase in lateral velocity as the vehicle approaches maximum velocity v_m . The latter point is further discussed later in this thesis.

The vehicle uses all of the available track, as can be observed in Sub-figures 5.4b and 5.4d, as the vehicle hits the track outer bounds n_l/n_r . This is to maximize the velocity taken through the turn. This is in accordance with the concept of a "racing line". The power limitation P_m is also enforced.

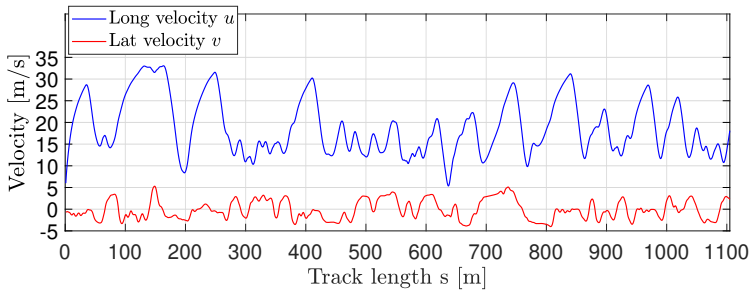
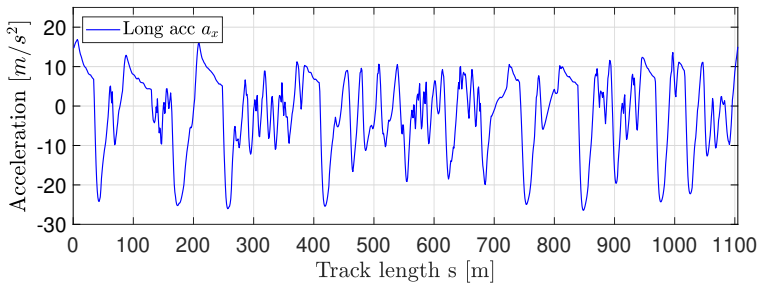
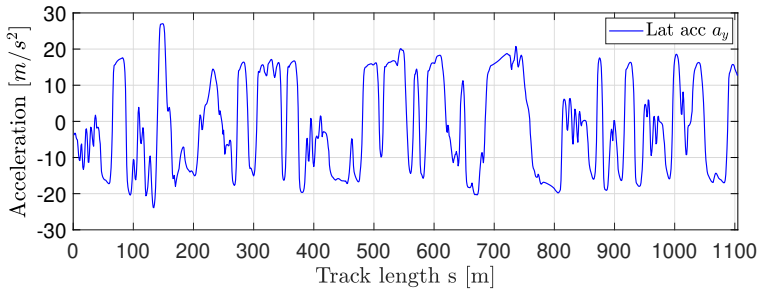
The vehicle inputs: Wheel angle δ and slip ratios κ_f/κ_r are shown. Only f/r as a single-track vehicle model is deployed. The oscillatory behavior of these inputs is shown, and this was limited carefully in the full simulator as discussed in Chapter 4.

5.4 Full simulation results: 2019 FSG Autox

The full simulator is used to simulate Revolves 2019 car *Nova* (parameters in A.1), around the 2019 FSG AutoX track [24]. The allowable normal deviation from the track center line is set to $n = 0.01\text{m}$. The step length was: $\Delta s = 1\text{m}$, giving 60732 optimization variables. The CPU-time for the simulation was: $t_{CPU} = 17.4\text{min}$. The simulated laptime was: $t_f = 64.8615\text{sec}$.

The longitudinal and lateral velocity profiles are shown in Sub-figure 5.5a. Like the preliminary simulation, the figure shows the starting longitudinal velocity $u = 6\text{m/s}$, the lateral velocity that occurs in turns, and an increase in lateral velocity as the vehicle approaches maximum velocity v_m . At around $s = 150\text{m}$ the vehicle hits the maximum speed, but the model combats this limitation as it turns the car, adding in some lateral velocity. This is discussed further in Subsection 6.4.1.

The longitudinal and lateral acceleration profiles are presented. The maximum longitudinal acceleration in the forward case is: $a_{x,m}^+ \approx 16.87\text{m/s}^2$, and for the braking case: $a_{x,m}^- \approx -26.46\text{m/s}^2$. The maximum lateral acceleration is: $a_{y,m} \approx 27.02\text{m/s}^2$. Some oscillations are observed in the acceleration data.

(a) Full sim. 2019 FSG AutoX: Longitudinal and lateral velocities u and v (b) Longitudinal acceleration a_x (c) Lateral acceleration a_y **Figure 5.5:** Full simulator: 2019 FSG AutoX results. Velocity and acceleration profiles

The vehicle inputs: Wheel angle δ and slip ratios $\kappa_{i,j}$ are shown. For all four tires this time. Not that many oscillations are present in the data, but the bounds on the input rates are observed in the figures, as the vehicle inputs appear to be piecewise linear.

The limits on the powertrain are shown with both power used P_t and motor angular velocities $\omega_{i,j}$. The upper bounds on these two quantities are enforced. It is also visible in the angular velocity data that the vehicle uses differential motor velocities to steer the vehicle in low-speed corners. This is clearly visible in one of the slowest corners at $s \approx 200\text{m}$.

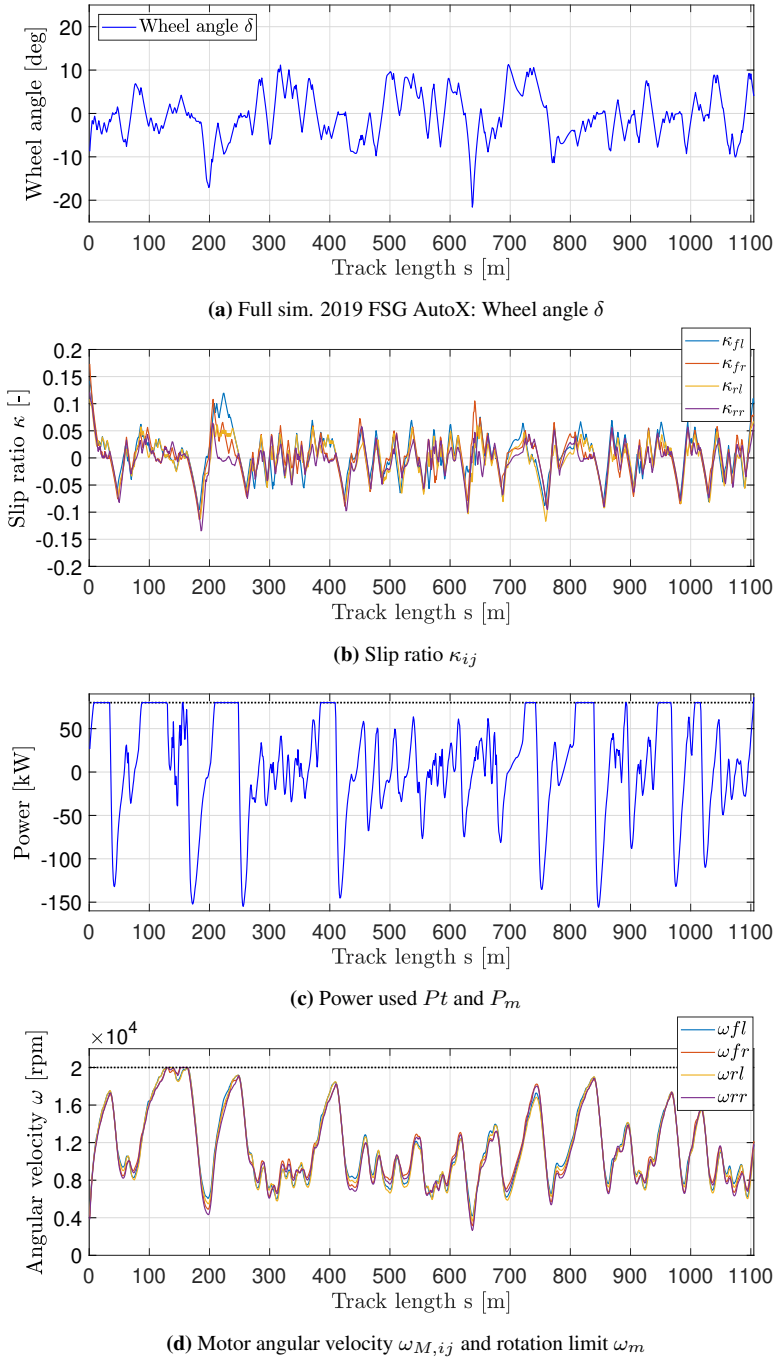


Figure 5.6: Full simulator: 2019 FSG AutoX results. Vehicle inputs and powertrain

5.5 OC simulator compared to telemetry

The OC simulator was used to simulate Revolves 2019 car *Nova* (parameters in A.1), around the 2019 FSG AutoX track [24], and the 2019 FSA AutoX track [25]. The resulting velocity and acceleration profiles are compared to telemetry from the given competition. The GG-diagrams for the simulator and the telemetry are also presented. The allowable normal deviation from the track center line was set to $n = 0.01\text{m}$. The step length was: $\Delta s = 1\text{m}$. This gave 60732 optimization variables and a CPU-time: $t_{CPU} = 17.4\text{min}$ for the 2019 FSG AutoX, and 48852 optimization variables and a CPU-time: $t_{CPU} = 7.9\text{min}$ for the 2019 FSA AutoX.

5.5.1 Laptimes

The simulated lap times from the OC simulator and the lap times recorded during the actual driven competitions are compared. The recorded lap times are taken from their respective competition website: FSG [40, FSE19 Scoring Results Autocross] and FSA [41, E-Autocross-Times-2019].

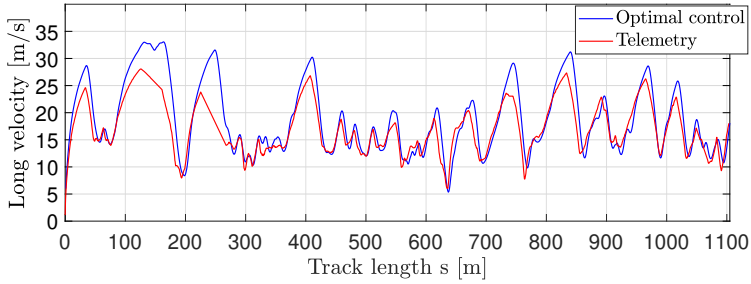
Competition	Sim. laptime [s]	Rec. laptime [s]	Percentage diff. [%]
2019 FSG AutoX	64.862	68.074	4.72%
2019 FSA AutoX	50.163	54.685	7.4%

Table 5.1: Laptimes compared: OC simulator vs recorded

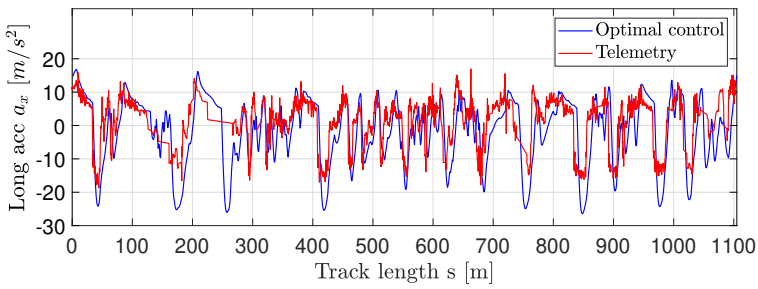
5.5.2 Velocity and acceleration profiles

The comparison between the OC simulator and the telemetry shows good agreement between the simulator and the real vehicle. The velocity and acceleration profiles for both competitions are similar, with package loss in mind.

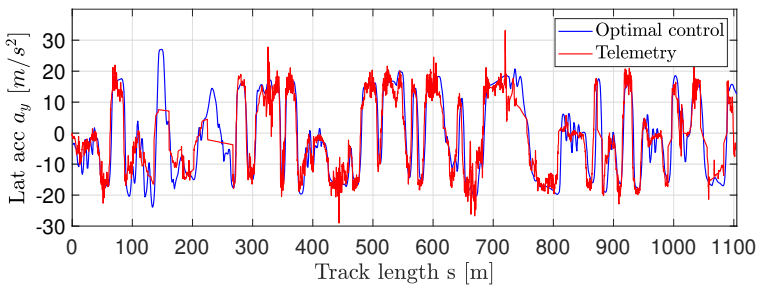
It is important to note the difference in peak velocity and peak braking ($a_x < 0$). The simulator reaches higher velocities and is able to brake harder. This fact is further emphasized in the GG-diagrams for the two competitions shown in Figure 5.9: Good agreement between simulator and the real car for positive longitudinal acceleration a_x^+ and lateral acceleration a_y , but clear difference between the negative (braking) longitudinal acceleration a_x^- .



(a) 2019 FSG AutoX: OC sim vs telemetry: Longitudinal velocity u

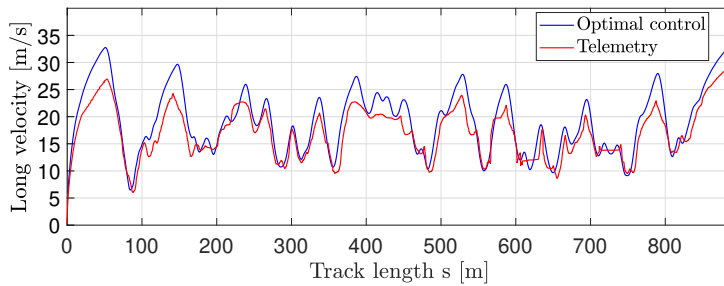


(b) Longitudinal acceleration a_x

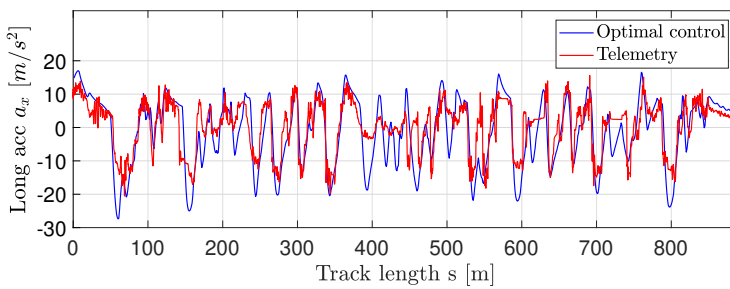


(c) Lateral acceleration a_y

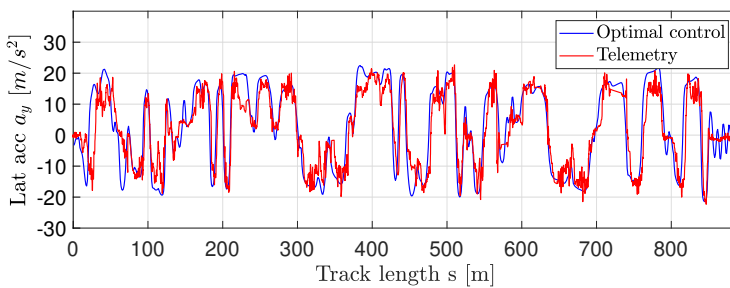
Figure 5.7: 2019 FSG AutoX: OC simulator vs telemetry. Velocity and acceleration profiles



(a) 2019 FSA AutoX: OC sim vs telemetry: Longitudinal velocity u



(b) Longitudinal acceleration a_x



(c) Lateral acceleration a_y

Figure 5.8: 2019 FSA AutoX: OC simulator vs telemetry. Velocity and acceleration profiles

5.5.3 GG diagrams

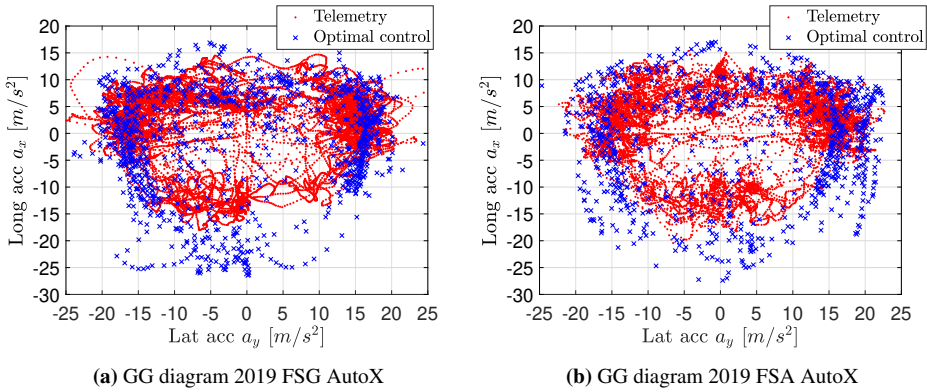


Figure 5.9: OC simulator vs telemetry. GG diagrams compared for 2019 FSG and FSA AutoX

5.6 OC simulator compared to QSS simulator

To compare the two methodologies, the OC simulator was compared to the QSS simulator produced by Smith [1]. The same OC simulation results used in Section 5.5 was used in the comparison, and the same vehicle parameters (*Nova A.1*) and step length was used in the QSS simulator. The CPU-times for the QSS simulator were: $t_{CPU} = 53.2\text{sec}$ for the FSG AutoX, and $t_{CPU} = 52.0\text{sec}$ for the FSA AutoX. Note that it would only be a few second on top of one of those CPU-times to run both at the same time, as the time consuming part is generating the performance envelope for the vehicle model. Running the solver for a given track takes just a few seconds.

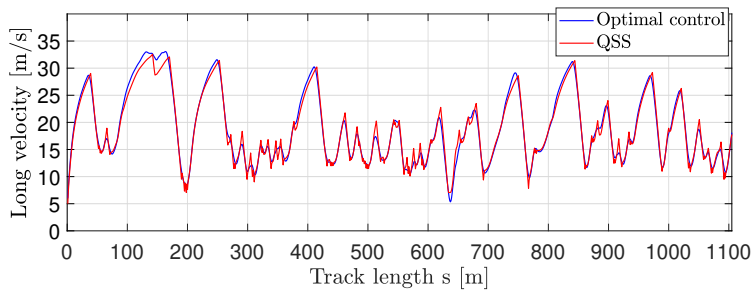
5.6.1 Laptimes

The lap times simulated by the OC simulator and the QSS simulator are compared below.

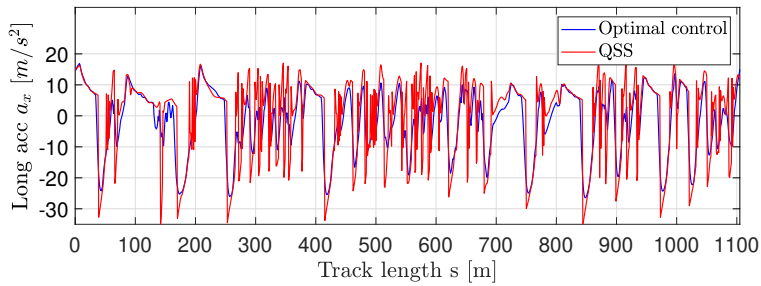
Competition	OC laptime [s]	QSS laptime [s]	Percentage diff. [%]
2019 FSG AutoX	64.862	65.90	2.86%
2019 FSA AutoX	50.163	49.049	2.27%

Table 5.2: Laptimes compared: OC simulator vs QSS simulator

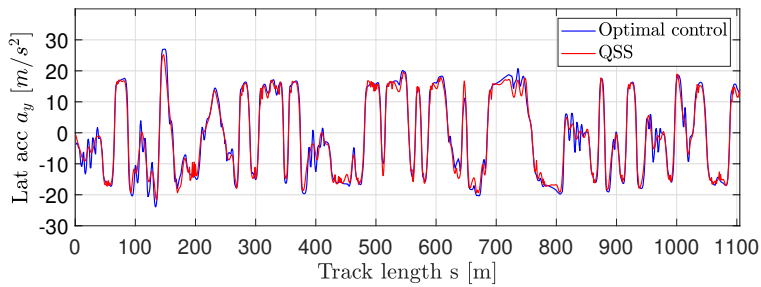
5.6.2 Velocity and acceleration profiles



(a) 2019 FSG AutoX: OC sim vs QSS sim: Longitudinal velocity u

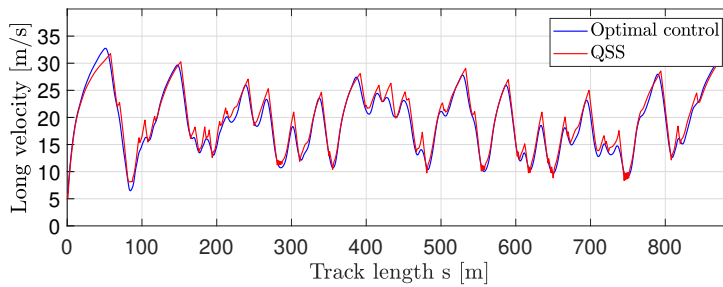
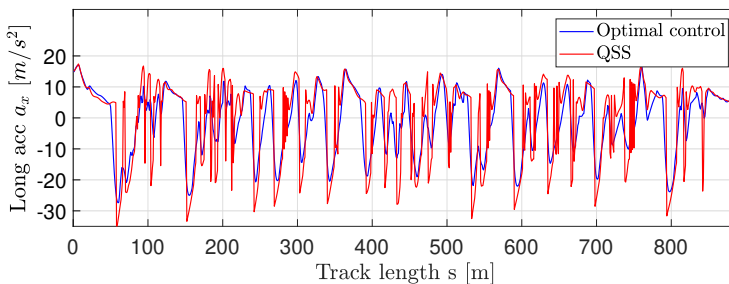
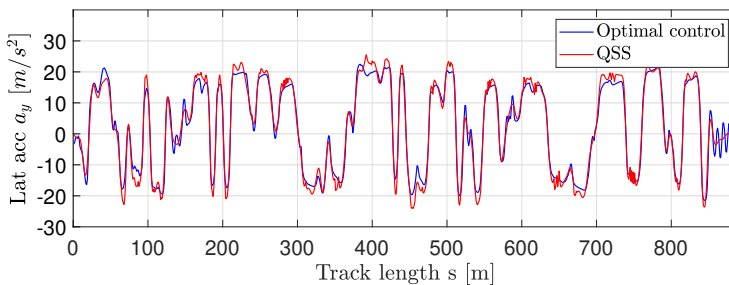


(b) Longitudinal acceleration a_x



(c) Lateral acceleration a_y

Figure 5.10: 2019 FSG AutoX: OC simulator vs QSS simulator. Velocity and acceleration profiles

(a) 2019 FSA AutoX: OC sim vs QSS sim: Longitudinal velocity u (b) Longitudinal acceleration a_x (c) Lateral acceleration a_y **Figure 5.11:** 2019 FSA AutoX: OC simulator vs QSS simulator. Velocity and acceleration profiles

The comparison shows good agreement between the two simulators. The QSS simulator achieves slightly higher lateral and positive longitudinal acceleration, but the major difference between the two is observed in the negative longitudinal acceleration. There is also a difference in the peaks of the velocity profiles. The continuous nature of the OC simulator is obvious compared to the QSS simulator when looking at the longitudinal acceleration profile. The transitions between acceleration and deceleration are also distinct for the two different simulators.

5.7 Comparing overall vehicle designs - Nova vs Atmos

A comparison was made between two different vehicle designs: Revolve NTNU's 2019 car *Nova* (Parameters: A.1), and Revolve NTNU's 2018 car *Atmos* (Parameters: A.2). They were simulated on the 2019 FSG autoX track, with the same settings as with the other simulations done with *Nova*. The CPU-time were similar for the two cars, at $t_{CPU} \approx 17$ min.

5.7.1 Laptimes

The lap times simulated by the OC simulator for the two cars are compared below. For reference and further discussion, the lap times recorded at competitions are also listed below.

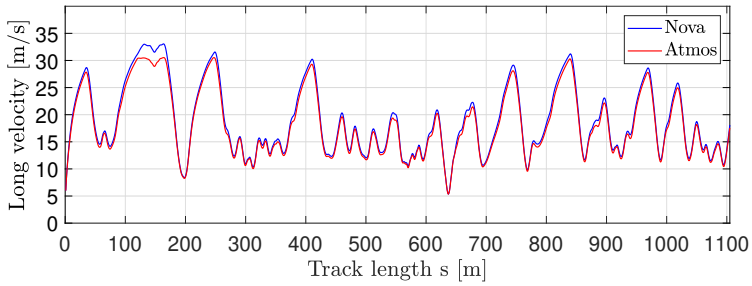
Vehicle	Sim. laptime [s]	Rec. laptime [s]	Percentage diff. [%]
Nova	64.862	68.074	4.72%
Atmos	67.039	67.331	0.43%

Table 5.3: Laptimes compared: Nova vs Atmos

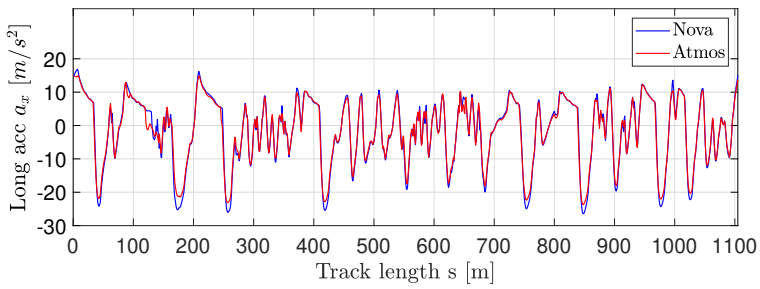
5.7.2 Velocity and acceleration profiles

The velocity and acceleration profiles are compared. Nova and Atmos are very similar cars, but Nova has Atmos beat. The lap times and profiles plotted, show that Nova can accelerate faster, both longitudinally and laterally, giving a faster overall lap, and resulting in lower lap times.

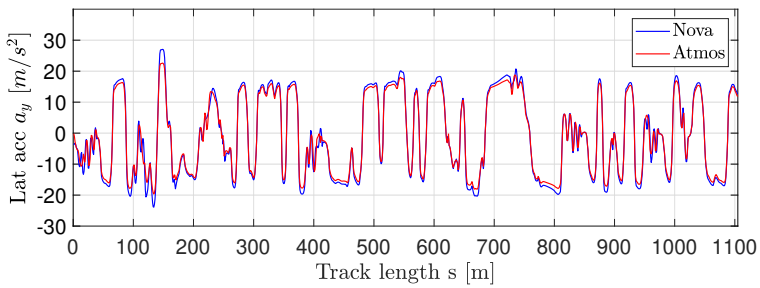
The difference in lap time recorded during competitions is an important point that is further discussed in Sub-section 6.4.1. Note that the run for which Atmos' lap time is presented, received a penalty due to hitting a cone during the run, giving an actual best time of $T = 69.331$. For this comparison, this fact is ignored.



(a) 2019 FSG AutoX: Nova vs Atmos Longitudinal velocity u



(b) Longitudinal acceleration a_x



(c) Lateral acceleration a_y

Figure 5.12: 2019 FSG AutoX: Nova vs Atmos. Velocity and acceleration profiles

Discussion

6.1 Data collection

6.1.1 Accuracy of the telemetry

The VectorNav VN-300 GNSS-Aided Inertial Navigation System (INS) with Integrated GNSS-Compass [36] was used to produce the recorded data presented in this thesis. The velocity profiles presented are not measured, but estimates produced with a Kalman filter in the INS. In 2016 Revolve NTNU verified the velocity estimates from the INS, by taking separate ground speed measurements with the Kistler Correvit SFII: 2-axis optical sensor [42]. The comparison showed the INS to give an accurate velocity estimate for the vehicle when compared to the optical sensor. A comparison of the two sensors from the 2016 FSG AutoX is shown in Figure 6.1.

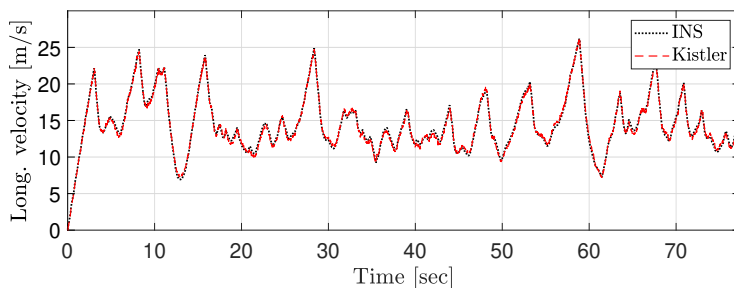


Figure 6.1: 2016 FSG AutoX: INS vs Kistler Optical sensor Longitudinal velocity u

6.1.2 Package loss

The telemetry data suffers from package loss that occurred during the competition, as the telemetry system failed to transmit all data packages from the vehicle to the base station. In the track model, this loss of information is handled by the spline interpolation, but this leads to a diluted final result. In the telemetry velocity and acceleration profiles, this package loss makes a detailed comparison of the OC simulator and the real car difficult, as the position data is inaccurate. An analysis of braking points, for example, is difficult. The data is deemed decent enough for a comparison between the velocity and acceleration magnitudes.

6.2 Track model

The trackmaker, consisting of spline interpolation of GNSS coordinates and OC based curvature smoothing, makes it possible to take telemetry from a test or competition run with a vehicle, extract the GNSS points, and create tracks for both the OC simulator and the QSS simulator presented in [1]. That includes varying the step length as needed. The trackmaker is simple and fast to deploy as a track is ready for the simulator after just a few seconds.

There are however some important considerations with the trackmaker. The tracks that are produced are not the actual tracks driven by the vehicle. The track centerline produced is the trajectory taken by the driver when the telemetry was recorded. For this reason, the simulations, except for the preliminary study, are produced with small boundaries n_l/n_r . It is assumed that the track centerlines used in this thesis are the racing line. The Formula Student tracks are cone tracks, where a large flat or parking lot is made into a track by placing cones out on the flat, specifying the track boundaries. This makes reconstruction of the track in question difficult without accurate imagining. An alternative to the track model used in this thesis could be a manually generated track, where it would be possible to ease the boundaries n_l/n_r , and allow the OC simulator to optimize the trajectory taken by the vehicle. Note that it is limited how much a 1.4m wide car can maneuver on a track that is only 3m wide [39, Rule D6.1.1.], so the cost-benefit analysis for this manual approach is questionable. The spline interpolation and curvature smoothing scheme dilutes the track further. The package loss is also an issue, and a better final result could be achieved, meaning a more accurate representation of the real track, if the quality of the telemetry was improved.

The discrepancy between the track model and the real track is not deemed a fatal flaw,

however, as the simulator made in this thesis is meant as a tool for automotive engineers to evaluate vehicle designs. The comparison between the simulator and the telemetry is just an indicator of how well the simulation models the vehicle under study.

6.3 Preliminary Study - The Optimal Trajectory

One of the maneuvers simulated in the preliminary study was presented in Section 5.3. The simple left-hand turn maneuver displays some of the important aspects of the OC simulator built in this thesis. The simulator is a free trajectory simulator, where both the vehicle inputs and the trajectory taken by the vehicle are determined by the optimization. The way the track is modeled in this thesis (already the approximate optimal path), this capability is limited as shown, but it is present in the simulator as a capability for future work.

This optimal trajectory is displayed in Sub-figure 5.4b. The vehicle moves to the right side of the track before turning into the corner. In the corner, the vehicle moves across the track centerline to the left side of the track, as it hits the *apex* (the point of maximum curvature) of the corner. As the vehicle accelerates out of the corner it again moves across the center line to the right side of the track. This line is commonly referred to as the *racing line*. The racing line is distinct from the shortest path, as it is longer in distance, but maximizes the progression of the vehicle through the corner. This method of driving is used by racing drivers to minimize lap times, and the preliminary study captures and displays this aspect of the OC simulator.

6.3.1 Challenges with the chosen methodology

The results from the preliminary study also display several challenges with the chosen methodology.

The vehicle inputs show oscillatory behavior. The wheel angle shown in Sub-figure 5.4f displays this behaviour after $s = 80\text{m}$. It is unclear why the optimization gives this input as the vehicle travels in a straight line, as the intuitive input is no steering input: $\delta = 0$. The oscillatory behavior was mitigated, but not eliminated, by using the rates of the vehicle inputs as inputs, and letting the integration produce the actual vehicle inputs. This increased the problem size significantly, and a need to determine the upper and lower bounds for the rates. Further work is needed to analyze the oscillatory behavior, and possibly additional remedies for it.

After $s \approx 92\text{m}$ the vehicle begins to yaw and increase the lateral velocity v . This happens as the vehicle approaches the maximum longitudinal velocity. This behavior displays a shortcoming in the vehicle modeling, as the vehicle progression is increased by doing this, and not decreased as would be the case for an actual vehicle. The vehicle model in the preliminary study does not have an aerodynamic package, so the aerodynamic forces present on a real vehicle at high speed, are not present. This effect is further analyzed in Sub-section 6.4.1.

6.4 Full simulator

6.4.1 Vehicle Modeling

When dealing with optimization, it is important to watch for modeling errors, as these can be exploited by the optimization solver, producing skewed results. The vehicle model presented in this thesis is comprehensive, and the most important modeling aspects are included: Tire model, load transfer, tire slip, and aerodynamics. All on a double-track vehicle model. It becomes clear when comparing the simulation result to the telemetry in Section 5.5 that the vehicle simulator captures a significant part of the real vehicle dynamics. The lateral acceleration profile is very similar between the simulator and the telemetry. The velocity profile is very similar, and so is the longitudinal acceleration profile. There is a significant difference in braking performance between the simulated vehicle and the real vehicle. It is believed that this in turn creates the opportunity for higher peak velocities, as the simulated vehicle can accelerate for longer before braking. Potential reasons for this are discussed.

The tire model

The tire model used in the vehicle model is complex. The Pajecka Magic Formula 5.2 provided by Revolve NTNU's tire manufacturer Continental, is shown in the Appendix B.1. The tire model adds significant complexity to the vehicle model, and this could be reason for the increase in CPU-time observed in Section 5.1, when compared to Perantoni and Limebeer [2] and Christ et al. [3] that used similar methodologies. The tire model is accurate, but needs to be scaled to produce a realistic result. The scaling constant $c = 0.55$ applied to both the longitudinal and the lateral tire model, is simplistic, and could be insufficient in dealing with the high vertical tire loads at high velocities. Looking at Sub-figures 5.7a and 5.7b, peak braking (a_x^-) occurs at points of high velocity. At $s \approx 850\text{m}$ this is shown. It is possible that the high vertical loads on the tires at high velocities

(from aerodynamic downforce), extrapolates the tire model past an accurate range. In their documentation, Continental lists a range of vertical loads for which their tire model is valid: $F_z \in [230, 1600]$ N. The tire vertical loads for the 2019 FSG AutoX simulation are presented below:

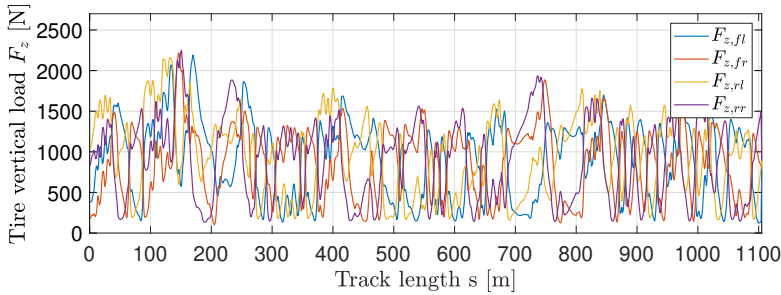


Figure 6.2: 2019 FSG AutoX: Tire vertical loads $F_{z,ij}$

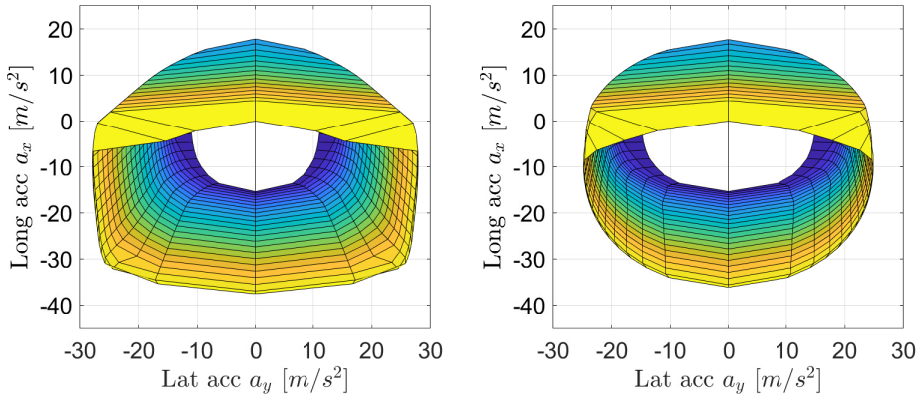
Figure 6.2 shows that the tire vertical loads do move outside the maximum, valid vertical tire load. More specifically this is observed for the section of track described above, at $s \approx 850$ m. The peak sits at $F_z \approx 1750$ N. Further work is needed to determine if this is significant enough to overestimate the braking acceleration as observed in Section 5.5.

The aerodynamic modeling

The aerodynamic model used in the vehicle model is also simplistic. The coefficient of lift and drag C_L and C_D are derived from CFD simulations, by simulating the vehicle at $u = 60$ km/h. This could lead to modeling error at higher velocities. The aerodynamic package is also idealized, as the vehicle has no suspension and therefore no movement in the aerodynamic package. It is unclear if this increases or decreases the aerodynamic downforce, as braking tends to make the vehicle pitch forward giving more aerodynamic downforce, but the vehicle is also under a lot of downforce, so the ride height is lowered, limiting the airflow to the underbody of the car, limiting downforce.

Through expanding the QSS simulator, outlined in [1], an attempt was made to include variable aerodynamic coefficients. The attitude of the vehicle was estimated using a "meta-suspension system": The calculated spring stiffnesses of the suspension were combined with velocity, for vertical load on the vehicle body through down force, longitudinal and lateral acceleration for pitch and roll, and lateral velocity for vehicle side slip. Using a lookup table, vehicle ride height, pitch, roll, and, side slip was used to find aerodynamic coefficients derived from parameter sweeps conducted in CFD simulations. The perfor-

mance envelope was generated, and compared to a performance envelope for a vehicle model without the expanded aerodynamic model. The resulting performance envelopes (GGV-diagrams), generated for *Nova* (parameters in A.1) are presented below:



(a) GGV top view: Constant aerodynamic coefficients (b) GGV top view: Variable aerodynamic coefficients

Figure 6.3: Performance envelope comparison showing longitudinal and lateral acceleration

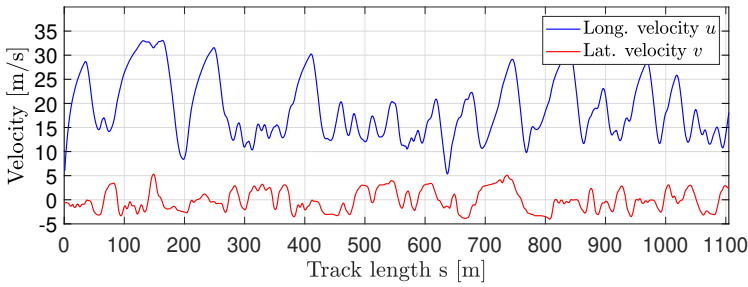
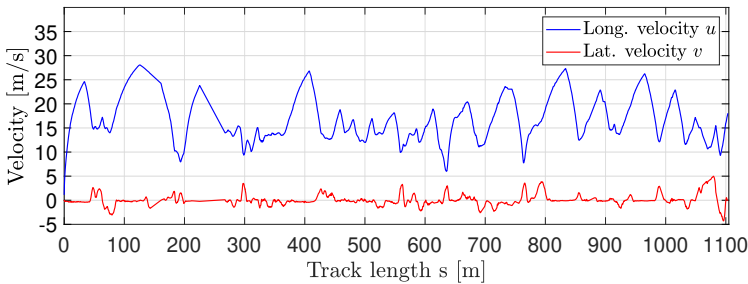
The variable aerodynamic coefficients have a clear impact on the maximum longitudinal and lateral acceleration a vehicle can produce, especially in combined longitudinal and lateral acceleration, as is often the case on an AutoX track. Further work could determine the impact of the simplification of the aerodynamic model used in the OC simulator.

Lateral velocity at high longitudinal velocity

In both the preliminary study in Section 5.3, and the full simulation study 5.4, a strange increase in lateral velocity v was observed at high longitudinal velocities u . The longitudinal and lateral velocity profiles are presented for both the OC simulator and telemetry, from FSG 2019 AutoX:

This increase in lateral velocity is clearly observed at $s \approx 740\text{m}$. The simulator increases lateral velocity almost in tandem with the longitudinal velocity, while the telemetry shows the lateral velocity decreasing as the local peak in longitudinal velocity is reached.

The lateral velocity observed could be a result of a lack of lateral sensitivity in the aerodynamic package. As the vehicle turns, side slip is introduced, which hurts aerodynamic performance. The aerodynamic model only uses longitudinal velocity u to calculate aerodynamic downforce and drag, ignoring lateral velocity. The powertrain also ignores the lateral velocity, as only the tire longitudinal velocity in the tire frame is used to calculate

(a) Longitudinal and lateral velocities u and v from simulator(b) Longitudinal and lateral velocities u and v from telemetry**Figure 6.4:** 2019 FSG AutoX: Longitudinal and lateral velocities u and v compared

power used P_i , torque produced by the motors T_{ij} , and motor angular velocity $\omega_{i,j}$. Sensitivity to lateral velocity should be included in future work, to remove this error in the model.

Driver influence

The telemetry used in the comparison in Section 5.5 is recorded on a vehicle driven by a human driver. The effects of the driver's abilities and confidence in the car cannot be overstated. It is telling that the lateral and positive longitudinal acceleration data for the OC simulator agrees well with the telemetry, while the negative longitudinal acceleration data does not. It is possible that the driver recorded was less confident in the stopping power of the vehicle than in the lateral grip. The effects a driver has on the lap time of a vehicle is observed in the comparison between *Nova* and *Atmos* in Section 5.7. Even though *Nova* is the faster car in the simulator, *Atmos* achieved the fastest lap time at the competition. A possible reason for the discrepancy between the simulated vehicle and the recorded vehicle is the driver that produced the telemetry data.

This raises an important potential use case for the OC simulator produced: Driver evalua-

tion. It is possible to use the simulator as a benchmark when evaluating where a driver is doing well, and where a driver might be leaving performance on the table. Good judgment is needed when doing this, as the simulator gives a theoretical optimum, and it is not given that a driver can reach this.

6.5 OC compared to QSS

In Section 5.6 the OC simulator produced in this thesis was compared to the QSS simulator produced in [1]. The comparison is interesting as both simulators bring distinct advantages.

The OC simulator provides a more complete description of the vehicle dynamics. Even though the vehicle and track modeling are the same for the two simulators, the OC simulator provides a close approximation to the continuous dynamics of the vehicle. The inclusion of yaw dynamics, where the change in yaw rate is produced through a yaw moment is important, as the acceleration of the vehicle in yaw can have a significant impact on the final solution, especially considering the AutoX type tracks where this is an important part of the vehicle's performance. The modeling of transient effects is valuable as it allows modeling suspension systems with springs and dampers. The two main drawbacks are the CPU time and the stability of the optimization problem as the model complexity is increased.

The QSS simulator on the other hand is shown to be a good approximation of the dynamics modeled in the OC simulator. The velocity and acceleration profiles are very similar, but the quasi steady-state approximation of the vehicle dynamics and lack of transient effects, are visible in the longitudinal acceleration profile shown in Sub-figure 5.10b. The maximum possible positive or negative longitudinal acceleration is applied for each step, without continuity. A big advantage of the QSS simulator is the CPU-time, as it allows quick investigation and iteration of vehicle designs, with good accuracy in the solution.

Conclusion and Future Work

7.1 Conclusion

An Optimal Control (OC) based Lap Time Simulation (LTS) has been implemented and presented. The simulator uses a double-track vehicle model, with a nonlinear tire model, load transfer, aerodynamic package, and basic powertrain modeling. The MTM problem was formulated as a free trajectory Optimal Control Problem (OCP), and solved with a Non Linear Programming (NLP) solver, using a direct orthogonal collocation method to transcribe the problem.

The important vehicle modeling aspects have been presented and implemented in the MTM problem. The track model was created using a track maker, implemented using spline interpolation and OC to transform GNSS coordinates into a smooth track center-line. The trackmaker was used to produce track center lines from telemetry data, creating track center lines for the 2019 FSG AutoX track, and the 2019 FSA AutoX track.

A preliminary study, using simplified vehicle and track models, was presented, showcasing the free trajectory optimization capabilities of the chosen methodology.

The full simulator was presented, simulating Revolve NTNU's 2019 racecar *Nova* around the 2019 FSG AutoX track. Simulations on both the 2019 FSG AutoX track and the 2019 FSA AutoX tracks were compared to telemetry recorded at the given competition. The comparison showed good agreement between the simulator and the real vehicle, except for the braking performance. The simulator was able to brake much harder than what the

telemetry showed the real car to do, producing a discrepancy in the negative longitudinal acceleration and as a result faster lap times. Several potential reasons for this were identified, including the tire model, the aerodynamic modeling, and the driver influence on the telemetry data.

The OC simulator was also compared to a QSS based simulator, using the same vehicle and track models. The comparison help identify the advantages of the two different methodologies.

Lastly, a comparison between the two vehicle designs was made. This showcased the capabilities to use the simulator for its intended purpose: Evaluating vehicle designs. The lighter, more agile, car *Nova* beat Revolve NTNU's 2018 racecar *Atmos* around a lap of the 2019 FSG AutoX.

7.2 Future Work

Several areas to improve and expand the work presented in this thesis have been identified.

7.2.1 Vehicle modeling

With the methodology for solving the MTM established, the modeling in the simulator should be considered.

Tire modeling

The tire model used in the vehicle model is complex, and a potential reason for the high computation cost of the MTM problem. There is potential for scaling back the complexity of the model while retaining the important modeling aspects. A possible simplification to make would be to use a simpler form of the semi-empirical Magic Formula, or use a Tire Brush model [26, Chapter 3].

If increased modeling accuracy is desired, the tire model provided by Continental does contain several tire modeling aspects that are not used in this thesis. Effects from rolling resistance, aligning torque about the tire vertical axis, and overturning moment about the tire longitudinal axis, are all presented in the Pacejka Magic Formula 5.2. By including these modeling aspects in the vehicle model, an even more comprehensive tire model could be achieved, at the expense of computational cost and less stable convergence of the simulator OCP.

Aerodynamic modeling

The potential modeling error in the aerodynamic model of the vehicle model has been discussed. Sensitivity to vehicle side slip should be included. A more comprehensive aerodynamic model could be implemented, similar to what was presented in Sub-section 6.4.1. If a multi-body vehicle model was implemented, then a dynamic aerodynamic model is a natural extension.

Multi-body vehicle model

Lot and Dal Bianco [43] showed the significance of implementing a multi-body vehicle model. A 14 Degrees Of Freedom (DOF) vehicle model was simulated using OC. 10 DOF and 7 DOF models were also implemented and simulated. The latter being a model similar to the one implemented in this thesis. The multi-body models were shown to have a significant impact on the final result. The 14 DOF model and the 10 DOF were very similar, but the 10 DOF required much less computation time. A 10 DOF vehicle model, like then one implemented by Lot and Dal Bianco [43], could be implemented in the OC simulator presented in this thesis.

Powertrain model

The powertrain model implemented in this thesis is limited. It covers the basic limitations of the powertrain, but a lot more can be modeled. The electric power train starts at the battery pack that supplies high voltage current to the motor controllers and inverters. The inverters run the hub-mounted motors, which produce torque through the gearboxes to drive the wheels and tires. In all these steps, losses and efficiencies could be modeled. Revolve NTNU's vehicles use energy recovery through the motors in braking, to regain some of the spent energy during acceleration. This can also be modeled for a more accurate model for the power consumption. The motors do not produce the same torque across the whole RPM range, as they drop in torque towards the end of the range. This has a significant impact on the vehicle dynamics and could be implemented and studied.

7.2.2 Autonomous planning

The simulator implemented can be reconfigured for use in an autonomous racecar, as the one Revolve NTNU has. The free trajectory optimization could be used to optimize a reference line on an already mapped track. This could help the autonomous racecar achieve faster lap times. To improve the computation time of the free trajectory optimization, a more simplistic model could be deployed. A single-track model could be an alternative.

7.2.3 Parameter Optimization

The direct transcription method deployed in the OC simulator is trivial to reconfigure for simultaneous simulation and parameter optimization. This methodology is well established and has been outlined in detail by Perantoni and Limebeer [2]. Note that this might lead to a less stable convergence of the optimization problem.

7.2.4 Graphical user interface

As this simulator is meant as a tool for automotive engineers to evaluate and iterate on vehicle designs, a Graphical User Interface (GUI) could be produced for the OC simulator to make its use more accessible. As the simulator is implemented in MATLAB, the MATLAB App Designer [44] could be used to implement the GUI.

Bibliography

- [1] Torbjørn Smith. Development of a Quasi Steady State Lap Time Simulator for a Formula Student Racing Team. Technical report, Norwegian University of Science and Technology, Trondheim, 1 2021.
- [2] Giacomo Perantoni and David J.N. Limebeer. Optimal control for a Formula One car with variable parameters. *Vehicle System Dynamics*, 52(5):653–678, 5 2014. ISSN 17445159. doi: 10.1080/00423114.2014.889315.
- [3] Fabian Christ, Alexander Wischnewski, Alexander Heilmeier, and Boris Lohmann. Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients. *Vehicle System Dynamics*, 59(4):588–612, 2021. ISSN 17445159. doi: 10.1080/00423114.2019.1704804.
- [4] Damien Brayshaw. *Use of numerical optimisation to determine on-limit handling behaviour of race cars*. PhD thesis, Cranfield University, Cranfield, 8 2004.
- [5] D. L. Brayshaw and M. F. Harrison. A quasi steady state approach to race car lap simulation in order to understand the effects of racing line and centre of gravity location. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 219(6):725–739, 6 2005. ISSN 09544070. doi: 10.1243/095440705X11211.
- [6] A. J. Tremlett, F. Assadian, D. J. Purdy, N. Vaughan, A. P. Moore, and M. Halley. Quasi-steady-state linearisation of the racing vehicle acceleration envelope: A limited slip differential example. *Vehicle System Dynamics*, 52(11):1416–1442, 11 2014. ISSN 17445159. doi: 10.1080/00423114.2014.943927.

- [7] M. Veneri and M. Massaro. A free-trajectory quasi-steady-state optimal-control method for minimum lap-time of race vehicles. *Vehicle System Dynamics*, 58(6): 933–954, 6 2020. ISSN 17445159. doi: 10.1080/00423114.2019.1608364.
- [8] Blake Siegler, Andrew Deakin, and David Crolla. Lap Time Simulation: Comparison of Steady State, Quasi-Static and Transient Racing Car Cornering Strategies. *SAE TECHNICAL PAPER SERIES*, 2000.
- [9] Chris Patton. *Development of Vehicle Dynamics Tools for Motorsports*. PhD thesis, Oregon State University, 2 2013.
- [10] Rodrigo Pasiani, Costa Roberto Bortolussi, Rodrigo Pasiani Costa, and Roberto Bortolussi. LAP TIME SIMULATION OF FORMULA SAE VEHICLE WITH QUASI-STEADY STATE MODEL. *SAE TECHNICAL PAPER SERIES*, 2016.
- [11] K. GUO and H. GUAN. Modelling of driver/vehicle directional control system. *Vehicle System Dynamics*, 22(3-4):141–184, 1993. doi: 10.1080/00423119308969025.
- [12] Wenjuan Jiang, Carlos Canudas de Wit, Olivier Sename, and Jonathan Dumon. A new mathematical model for car drivers with spatial preview. *IFAC Proceedings Volumes*, 44(1):1139–1144, 2011. ISSN 1474-6670. doi: <https://doi.org/10.3182/20110828-6-IT-1002.02634>. URL <https://www.sciencedirect.com/science/article/pii/S1474667016437626>. 18th IFAC World Congress.
- [13] Charles C. MacAdam. Application of an optimal preview control for simulation of closed-loop automobile driving. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):393–399, 1981. doi: 10.1109/TSMC.1981.4308705.
- [14] Christoforos Chatzikomis and Konstantinos Spentzas. A path-following driver model with longitudinal and lateral control of vehicle’s motion. *Forschung Im Ingenieurwesen-engineering Research - FORSCH INGENIEURW*, 73:257–266, 12 2009. doi: 10.1007/s10010-009-0112-5.
- [15] William F. Milliken and Douglas L. Milliken. *Race Car Vehicle Dynamics*. SAE International, Warrendale, PA, 12 1994. ISBN 978-1-56091-526-3.
- [16] D Casanova. *On Minimum Time Vehicle Manoeuvring: The Theoretical Optimal Lap*. PhD thesis, Cranfield University, 11 2000.
- [17] Daniel Patrick Kelly. *Lap Time Simulation with Transient Vehicle and Tyre Dynamics*. PhD thesis, Cranfield University, 5 2008.

- [18] Giacomo Perantoni and David J.N. Limebeer. Optimal Control of a Formula One Car on a Three-Dimensional Track-Part 1: Track Modeling and Identification. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 137(5), 5 2015. ISSN 15289028. doi: 10.1115/1.4028253.
- [19] D. J.N. Limebeer and G. Perantoni. Optimal Control of a Formula One Car on a Three-Dimensional Track - Part 2: Optimal Control. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 137(5), 5 2015. ISSN 15289028. doi: 10.1115/1.4029466.
- [20] S Van Koutrik. *Optimal Control for Race Car Minimum Time Maneuvering*. Master's thesis, Delft University of Technology, 3 2015.
- [21] Roborace. Roborace. <https://roborace.com/>, 2021. Accessed: 2021.06.06.
- [22] Joel A.E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 3 2019. ISSN 18672957. doi: 10.1007/s12532-018-0139-4.
- [23] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 5 2006. ISSN 00255610. doi: 10.1007/s10107-004-0559-y.
- [24] Formula Student Germany. Formula Student Germany: FSG. <https://www.formulastudent.de/fsg/>, 2021. Accessed: 2021.05.13.
- [25] Formula Student Austria. Formula Student Austria. <https://fsaustria.at/>, 2021. Accessed: 2021.05.13.
- [26] Hans B. Pacejka. *Tyre and Vehicle Dynamics*. Butterworth-Heinemann - Elsevier, Oxford, 2 edition, 2006. ISBN 978-0-7506-6918-4. doi: <https://doi.org/10.1016/B978-0-7506-6918-4.X5000-X>.
- [27] Reza N. Jazar. *Vehicle Dynamics: Theory and Application*. Springer, Boston, MA, 1 edition, 2008. ISBN 978-0-387-74243-4,978-0-387-74244-1. doi: 10.1007/978-0-387-74244-1.
- [28] James Balkwill. *Performance Vehicle Dynamics*. Butterworth-Heinemann - Elsevier, 1 edition, 8 2017. ISBN 9780128126936,9780128126943.

- [29] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer, New York, NY, 2 edition, 2006. ISBN 978-0-387-30303-1. doi: <https://doi.org/10.1007/978-0-387-40065-5>.
- [30] Philip E. Gill, Walter Murray, Michael A. Saunders, and Elizabeth Wong. User's guide for SNOPT 7.7: Software for large-scale nonlinear programming. Center for Computational Mathematics Report CCoM 18-1, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2018.
- [31] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47:99–131, 2005.
- [32] David J. N. Limbeer and Matteo Massaro. *Dynamics and Optimal Control of Road Vehicles*. Oxford University Press, Oxford, first edition, 2018. ISBN 9780198825715. doi: 10.1093/oso/9780198825715.001.0001.
- [33] Sébastien Gros and Moritz Diehl. *Numerical Optimal Control (preliminary and incomplete draft)*. 2021.
- [34] Joel Andersson, Joris Gillis, and Moritz Diehl. User Documentation for CasADi v3.4.4. Technical report, 2018.
- [35] OptimumG. OptimumLap. <https://optimumg.com/product/optimumlap/>, 2021. Accessed: 2021.02.04.
- [36] VectorNav. VN-300. <https://www.vectornav.com/products/detail/vn-300>, 2021. Accessed: 2021.06.20.
- [37] The MathWorks Inc. *Mapping Toolbox*, 2021. URL <https://www.mathworks.com/products/mapping.html>.
- [38] The MathWorks Inc. *Curve Fitting Toolbox*, 2021. URL <https://www.mathworks.com/products/curvefitting.html>.
- [39] Formula Student Germany. *FS-Rules 2020 V1.0.pdf*, 2021. URL <https://www.formulastudent.de/fsg/rules/>. Accessed: 2021.06.20.
- [40] Formula Student Germany. Results FSG 2019. <https://www.formulastudent.de/fsg/results/2019/>, 2021. Accessed: 2021.06.25.
- [41] Formula Student Austria. 2019 Results. <https://fsaustria.at/past-events/2019-2/>, 2021. Accessed: 2021.06.25.

- [42] Kistler Group. Correvit SFII: 2-axis optical sensors. <https://www.kistler.com/en/product/type-csf2a/>, 2021. Accessed: 2021.06.26.
- [43] Roberto Lot and Nicola Dal Bianco. The significance of high-order dynamics in lap time simulations. In *IAVSD 2015: 24th Symposium of the International Association for Vehicle System Dynamics*. CRC Press, 5 2016. ISBN 978-1-138-02885-2. doi: 10.1201/b21185-60. URL www.multibody.net/mbsymba/vehicles/gtcar14dof.
- [44] The MathWorks Inc. *App Designer*, 2021. URL <https://www.mathworks.com/products/matlab/app-designer.html>.

Appendix A

A.1 Vehicle Parameters - R19 Nova

Symbol	Description	Value	Unit
m	Vehicle mass (car and driver)	234.5	kg
I_z	Vehicle moment of inertia (car and driver)	82	kgm^2
wd_r	Mass distribution rear	0.514	-
h_{cog}	Center of mass height	0.273	m
wb	Vehicle wheel base	1.530	m
$\tau_{f/r}$	Vehicle track front/rear	1.20/1.20	m
δ_m	Max wheel angle	33	deg
P_m	Max power	80000	W
T_m	Max torque	29.2	Nm
R_l	Loaded wheel radius	0.228	m
n	Gear ratio	14.38	-
ω_m	Motor rotation limit	20000	rpm
ClA	Coeff. of lift (and frontal area)	5.60	m^2
CdA	Coeff. of drag (and frontal area)	1.82	m^2
ad_r	Aerodynamic distribution rear	0.5	-
κ_m	Max slip ratio	0.2	-
α_m	Max slip angle	10	deg
$\gamma_{f/r}$	Inclination angle front/rear	0.0/0.0	deg
g	Gravitational constant	9.81	m/s^2
ρ	Air density	1.184	kg/m^3
c_x	Longitudinal tire scaling factor	0.55	-
c_y	Lateral tire scaling factor	0.55	-
ϵ	Coefficient used in smoothing functions	10^{-2}	-

Table A.1: Vehicle parameters for Revolve NTNU's 2019 car: Nova

A.2 Vehicle Parameters - R18 Atmos

Symbol	Description	Value	Unit
m	Vehicle mass (car and driver)	252	kg
I_z	Vehicle moment of inertia (car and driver)	110	kgm^2
wd_r	Mass distribution rear	0.505	-
h_{cog}	Center of mass height	0.288	m
wb	Vehicle wheel base	1.530	m
$\tau_{f/r}$	Vehicle track front/rear	1.20/1.18	m
δ_m	Max wheel angle	33	deg
P_m	Max power	80000	W
T_m	Max torque	21	Nm
R_l	Loaded wheel radius	0.228	m
n	Gear ratio	15.58	-
ω_m	Motor rotation limit	20000	rpm
ClA	Coeff. of lift (and frontal area)	4.814	m^2
CdA	Coeff. of drag (and frontal area)	1.786	m^2
ad_r	Aerodynamic distribution rear	0.5	-
κ_m	Max slip ratio	0.2	-
α_m	Max slip angle	10	deg
$\gamma_{f/r}$	Inclination angle front/rear	0.0/0.0	deg
g	Gravitational constant	9.81	m/s^2
ρ	Air density	1.184	kg/m^3
c_x	Longitudinal tire scaling factor	0.55	-
c_y	Lateral tire scaling factor	0.55	-
ϵ	Coefficient used in smoothing functions	10^{-2}	-

Table A.2: Vehicle parameters for Revolve NTNU's 2018 car: Atmos

Appendix **B**

B.1 Magic Formula 5.2

The tire model used in this thesis is the Pacejka Magic Formula 5.2. The manufacturer of Revolve NTNU's tires: Continental, provides the model equations with fitted parameters. The parameter file used in this thesis is for the C19 Continental Formula Student 205/470 R13 tire, at $80kPa$. The model, as provided by Continental, is referenced below.

B.1.1 Inputs and Outputs

	Description	Symbol	Unit
Inputs	Nominal (rated) load	F_{z0}	[N]
	Unloaded tire radius	R_0	[m]
	Longitudinal slip	κ	[-]
	Slip angle	α	[rad]
	Camber angle	γ	[rad]
	Normal wheel load	F_z	[N]
Outputs	Longitudinal force	F_x	[N]
	Lateral force	F_y	[N]

Table B.1: Inputs and outputs from the Magic Formula 5.2

B.1.2 Normalized vertical load increment

$$df_z = \frac{F_z - F'_{z0}}{F'_{z0}}, \quad F'_{z0} = F_{z0} \cdot \lambda_{F_{z0}} \quad (\text{B.1})$$

B.1.3 Pure Lateral Slip

$$F_y = F_{y0}(\alpha, \gamma, F_z) \quad (\text{B.2})$$

$$F_{y0} = D_y \sin [C_y \arctan \{B_y \alpha_y - E_y (B_y \alpha_y - \arctan (B_y \alpha_y))\}] + S_{V_y} \quad (\text{B.3})$$

$$\alpha_y = \alpha + S_{H_y} \quad (\text{B.4})$$

$$\gamma_y = \gamma \cdot \lambda_{\gamma_y} \quad (\text{B.5})$$

Coefficients

$$B_y = K_y / (C_y D_y), \quad (\text{B.6})$$

$$C_y = p_{C_y1} \cdot \lambda_{C_y} \quad (\text{B.7})$$

$$D_y = \mu_y \cdot F_z \quad (\text{B.8})$$

$$E_y = (p_{E_y1} + p_{E_y2} df_z) \cdot \{1 - (p_{E_y3} + p_{E_y4} \gamma_y) \operatorname{sgn}(\alpha_y)\} \cdot \lambda_{E_y} (\leq 1) \quad (\text{B.9})$$

$$S_{H_y} = (P_{H_y1} + P_{H_y2} df_z) \cdot \lambda_{H_y} + P_{H_y3} \gamma_y \quad (\text{B.10})$$

$$S_{V_y} = F_z \cdot \{(p_{V_y1} + p_{V_y2} df_z) \cdot \lambda_{V_y} + (p_{V_y3} + p_{V_y4} \cdot df_z) \cdot \gamma_y\} \cdot \lambda_{\mu_y} \quad (\text{B.11})$$

Where:

$$K_y = p_{K_y1} F_{z0} \sin [2 \arctan \{F_z / (p_{k_y2} F_{z0} \lambda_{F_{z0}})\}] \cdot (1 - p_{K_y3} |\gamma_y|) \cdot \lambda_{F_{z0}} \cdot \lambda_{K_y} \quad (\text{B.12})$$

$$\left(\left(= B_y C_y D_y = \frac{\partial y_y}{\partial \alpha_y} \text{ at } \alpha_y = 0 \right) \right) \quad (\text{B.13})$$

$$\mu_y = (p_{D_y1} + p_{D_y2} df_z) \cdot (1 - p_{D_y3} \gamma_y^2) \cdot \lambda_{\mu_y} \quad (\text{B.14})$$

Coefficients – Pure Lateral Slip

Name	Name used in tire property file	Explanation
pCy1	PCY1	Shape factor C_{fy} for lateral forces
pDy1	PDY1	Lateral friction M_{uy}
pDy2	PDY2	Variation of friction M_{uy} with load
pDy3	PDY3	Variation of friction M_{uy} with squared camber
pEy1	PEY1	Lateral curvature E_{fy} at F_{znom}
pEy2	PEY2	Variation of curvature E_{fy} with load
pEy3	PEY3	Zero order camber dependency of curvature E_{fy}
pEy4	PEY4	Variation of curvature E_{fy} with camber
pKy1	PKY1	Maximum value of stiffness K_{fy}/F_{znom}
pKy2	PKY2	Load at which K_{fy} reaches maximum value
pKy3	PKY3	Variation of K_{fy}/F_{znom} with camber
pHy1	PHY1	Horizontal shift S_{hy} at F_{znom}
pHy2	PHY2	Variation of shift S_{hy} with load
pHy3	PHY3	Variation of shift S_{hy} with camber
pVy1	PVY1	Vertical shift in S_{vy}/F_z at F_{znom}
pVy2	PVY2	Variation of shift S_{vy}/F_z with load
pVy3	PVY3	Variation of shift S_{vy}/F_z with camber
pVy4	PVY4	Variation of shift S_{vy}/F_z with camber and load

Table B.2: Coefficients – Pure Lateral Slip

B.1.4 Pure Longitudinal Slip

$$F_x = F_{x0}(\kappa, F_z) \quad (\text{B.15})$$

$$F_{x0} = D_x \sin [C_x \arctan \{B_x \kappa_x - E_x (B_x \kappa_x - \arctan (B_x \kappa_x))\}] + S_{Vx} \quad (\text{B.16})$$

$$\kappa_x = \kappa + S_{Hx} \quad (\text{B.17})$$

$$\gamma_x = \gamma \cdot \lambda_{\gamma x} \quad (\text{B.18})$$

Coefficients

$$B_x = K_x / (C_x D_x) \quad (\text{B.19})$$

$$C_x = p_{Cx1} \cdot \lambda_{Cx} \quad (\text{B.20})$$

$$D_x = \mu_x \cdot F_z \quad (\text{B.21})$$

$$E_x = (p_{Ex1} + p_{Ex2} df_z + p_{Ex3} df_z^2) \cdot \{1 - p_{Ex4} \operatorname{sgn}(\kappa_x)\} \cdot \lambda_{Ex} (\leq 1) \quad (\text{B.22})$$

$$S_{Hx} = (p_{Hx1} + p_{Hx2} \cdot df_z) \lambda_{Hx} \quad (\text{B.23})$$

$$S_{Vx} = F_z \cdot (p_{Vx1} + p_{Vx2} df_z) \cdot \lambda_{Vx} \cdot \lambda_{\mu x} \quad (\text{B.24})$$

Where:

$$K_x = F_z \cdot (p_{Kx1} + p_{Kx2} df_z) \cdot \exp(p_{Kx3} df_z) \cdot \lambda_{Kx} \quad (\text{B.25})$$

$$\left(K_x = B_x C_x D_x = \frac{\partial F_{x0}}{\partial \kappa_x} \text{ at } \kappa_x = 0 \right) \quad (\text{B.26})$$

$$\mu_x = (p_{Dx1} + p_{Dx2} df_z) \cdot (1 - p_{Dx3} \cdot \gamma_x^2) \lambda_{\mu x} \quad (\text{B.27})$$

Coefficients – Pure Longitudinal Slip

Name	Name used in tire property file	Explanation
pCx1	PCX1	Shape factor C_{fx} for longitudinal force
pDx1	PDX1	Longitudinal friction M_{ux} at F_{znom}
pDx2	PDX2	Variation of friction M_{ux} with load
pDx3	PDX3	Variation of friction M_{ux} with camber
pEx1	PEX1	Longitudinal curvature E_{fx} at F_{znom}
pEx2	PEX2	Variation of curvature E_{fx} with load
pEx3	PEX3	Variation of curvature E_{fx} with load squared
pEx4	PEX4	Factor in curvature E_{fx} while driving
pKx1	PKX1	Longitudinal slip stiffness K_{fx}/F_z at F_{znom}
pKx2	PKX2	Variation of slip stiffness K_{fx}/F_z with load
pKx3	PKX3	Exponent in slip stiffness K_{fx}/F_z with load
pHx1	PHX1	Horizontal shift S_{hx} at F_{znom}
pHx2	PHX2	Variation of shift S_{hx} with load
pVx1	PVX1	Vertical shift S_{vx}/F_z at F_{znom}
pVx2	PVX2	Variation of shift S_{vx}/F_z with load

Table B.3: Coefficients – Pure Longitudinal Slip

B.1.5 Lateral Slip (Combined Slip)

$$F_y = F_{y0} \cdot G_{y\kappa}(\alpha, \kappa, \gamma, F_z) + S_{V_{yK}} \quad (\text{B.28})$$

$$F_y = D_{y\kappa} \cos [C_{y\kappa} \arctan \{B_{y\kappa}\kappa_s - E_{y\kappa} (B_{yK}\kappa_s - \arctan (B_{yK}\kappa_s))\}] + S_{V_{y\kappa}} \quad (\text{B.29})$$

$$\kappa_s = \kappa + S_{H_{yK}} \quad (\text{B.30})$$

with weighting function:

$$G_{y\kappa} = \frac{\cos [C_{y\kappa} \arctan \{B_{y\kappa}\kappa_s - E_{y\kappa} (B_{yK}\kappa_s - \arctan (B_{yK}\kappa_s))\}]}{\cos [C_{y\kappa} \arctan \{B_{yK}S_{H_{yK}} - E_{y\kappa} (B_{yK}S_{H_{yK}} - \arctan (B_{yK}S_{H_{yK}}))\}]} \quad (\text{B.31})$$

Coefficients

$$B_{y\kappa} = r_{B_{y1}} \cos [\arctan \{r_{B_{y2}} (\alpha - r_{B_{y3}})\}] \cdot \lambda_{y\kappa} \quad (\text{B.32})$$

$$C_{y\kappa} = r_{C_{y1}} \quad (\text{B.33})$$

$$D_{y\kappa} = \frac{F_{y0}}{\cos [C_{y\kappa} \arctan \{B_{yK}S_{H_{y\kappa}} - E_{y\kappa} (B_{yK}S_{H_{y\kappa}} - \arctan (B_{yK}S_{H_{y\kappa}}))\}]} \quad (\text{B.34})$$

$$D_{V_{y\kappa}} = \mu_y F_z \cdot (r_{V_{y1}} + r_{V_{y2}} df_z + r_{V_{y3}} \gamma) \cdot \cos [\arctan (r_{V_{y4}} \alpha)] \quad (\text{B.35})$$

$$E_{y\kappa} = r_{E_{y1}} + r_{E_{y2}} df_z \quad (\text{B.36})$$

$$S_{H_{y\kappa}} = r_{H_{y1}} + r_{H_{y2}} df_z \quad (\text{B.37})$$

$$S_{V_{y\kappa}} = D_{V_{y\kappa}} \sin [r_{V_{y5}} \arctan (r_{V_{y6}} K)] \cdot \lambda_{V_{y\kappa}} \quad (\text{B.38})$$

Coefficients – Lateral Slip (Combined Slip)

Name	Name used in tire property file	Explanation
rBy1	RBV1	Slope factor for combined F_y reduction
rBy2	RBV2	Variation of slope F_y reduction with α
rBy3	RBV3	Shift term for α in slope F_y reduction
rCy1	RCY1	Shape factor for combined F_y reduction
rEy1	REY1	Curvature factor of combined F_y
rEy2	REY2	Curvature factor of combined F_y with load
rHy1	RHY1	Shift factor for combined F_y reduction
rHy2	RHY2	Shift factor for combined F_y reduction with load
rVy1	RVY1	Kappa induced side force $S_{vyk}/\mu_y * F_z$ at F_{znom}
rVy2	RVY2	Variation of $S_{vyk}/\mu_y * F_z$ with load
rVy3	RVY3	Variation of $S_{vyk}/\mu_y * F_z$ with camber
rVy4	RVY4	Variation of $S_{vyk}/\mu_y * F_z$ with α
rVy5	RVY5	Variation of $S_{vyk}/\mu_y * F_z$ with κ
rVy6	RVY6	Variation of $S_{vyk}/\mu_y * F_z$ with $\tan(\kappa)$

Table B.4: Coefficients – Lateral Slip (Combined Slip)

B.1.6 Longitudinal Slip (Combined Slip)

$$F_x = F_{x0} \cdot G_{X\alpha}(\alpha, \kappa, F_z) \quad (\text{B.39})$$

$$F_x = D_{x\alpha} \cos [C_{x\alpha} \arctan \{B_{x\alpha} \alpha_s - E_{x\alpha} (B_{x\alpha} \alpha_s - \arctan (B_{x\alpha} \alpha_s))\}] \quad (\text{B.40})$$

$$\alpha_s = \alpha + S_{Hx\alpha} \quad (\text{B.41})$$

with weighting function:

$$G_{x\alpha} = \frac{\cos [C_{x\alpha} \arctan \{B_{x\alpha} \alpha_s - E_{x\alpha} (B_{x\alpha} \alpha_s - \arctan (B_{x\alpha} \alpha_s))\}]}{\cos [C_{x\alpha} \arctan [B_{x\alpha} S_{Hx\alpha} - E_{x\alpha} (B_{x\alpha} S_{Hx\alpha} - \arctan (B_{x\alpha} S_{Hx\alpha}))]]} \quad (\text{B.42})$$

Coefficients

$$B_{x\alpha} = r_{Bx1} \cos [\arctan \{r_{Bx2} K\}] \cdot \lambda_{x\alpha} \quad (\text{B.43})$$

$$C_{x\alpha} = r_{Cx1} \quad (\text{B.44})$$

$$D_{x\alpha} = \frac{F_{x0}}{\cos [C_{x\alpha} \arctan \{B_{x\alpha} S_{Hx\alpha} - E_{x\alpha} (B_{x\alpha} S_{Hx\alpha} - \arctan (B_{x\alpha} S_{Hx\alpha}))\}]} \quad (\text{B.45})$$

$$E_{x\alpha} = r_{Ex1} + r_{Ex2} df_z \quad (\text{B.46})$$

$$S_{Hx\alpha} = r_{Hx1} \quad (\text{B.47})$$

Coefficients – Longitudinal Slip (Combined Slip)

Name	Name used in tire property file	Explanation
rBx1	RBX1	Slope factor for combined slip Fx reduction
rBx2	RBX2	Variation of slope Fx reduction with kappa
rCx1	RCX1	Shape factor for combined slip Fx reduction
rEx1	REX1	Curvature factor of combined Fx
rEx2	REX2	Curvature factor of combined Fx with load
rHx1	RHX1	Shift factor for combined slip Fx reduction

Table B.5: Coefficients – Longitudinal Slip (Combined Slip)

