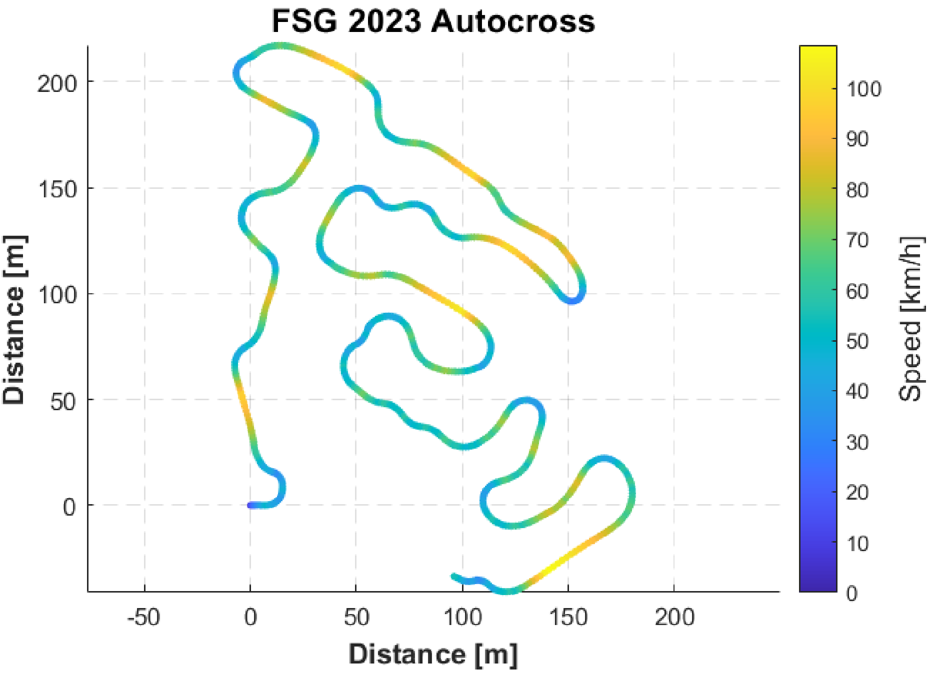
CHALMERS



Lap Time Simulation For A Formula Student Car

Bachelor Thesis in Mechanical Engineering

Tim Albertsson
Isak Nilsson
Alexander Wigh

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS TECHNICAL UNIVERSITY
Gothenburg, Sweden 2024
www.chalmers.se

Bachelor Thesis in Mechanical Engineering

Lap Time Simulation For A Formula Student Car

Tim Albertsson
Isak Nilsson
Alexander Wigh

CHALMERS

Department of Mechanics and Maritime Sciences
Chalmers University of Technology
Gothenburg 2024

Lap Time Simulation For A Formula Student Car
Tim Albertsson
Isak Nilsson
Alexander Wigh

© Tim Albertsson, Isak Nilsson, Alexander Wigh, 2024.

Supervisor:
Björn Pålsson, Associate Professor, Mechanics and Maritime Sciences
Christian Svensson, PhD Student, Mechanics and Maritime Sciences
Examiner:
Håkan Johansson, Professor, Mechanics and Maritime Sciences

Bachelor Thesis 2024
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Gothenburg

Cover Picture: Speed heat map for simulated lap of the 2023 Formula Student Germany Autocross Track.

Lap Time Simulation For A Formula Student Car
Tim Albertsson, Isak Nilsson, Alexander Wigh,
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Formula Student is a competition where universities all over the world compete in who can design and build the best Formula-style car. The car is judged based on design and performance in both static and dynamic events. Simulating the car's performance prior to these events is essential in order to build a car that is competitive. The purpose of this thesis is to build a lap time simulator that satisfies the needs of Chalmers Formula Student in ways that already existing commercial simulators can not. The result of this paper is a lap time simulator that has an easy to use user interface that enables a wide option of different simulations. The user can simulate all the dynamic events at the same time and also analyze, in multiple ways, how changes on the car affects the results in these events in both lap time and subsequent points scored. This report describes the theories behind the implementation of the simulator as well as the methods used to attain the result. The initial version of the lap time simulator uses a quasi steady state point-mass model, as it is a simple, quick to simulate, and simple to develop while still keeping good accuracy to real world data. A bicycle quasi steady state model was also developed to increase accuracy at the expense of simulation time. This model includes front and rear axles and therefore also longitudinal load transfer. As the simulation time for the bicycle model is about 10 times that of the point-mass, both are options in the user interface. As the Formula Student competitions include several dynamic events to test the car's on-track performance, it is also possible in the developed lap time simulator to simulate these events, either one by one, or multiple at once. Simulations featuring parameter sweeps were also developed to facilitate easy quantification of changes in parameters.

Key words: Lap Time Simulator, Simulation, Solver, Chalmers Formula Student, Telemetry visualization, GGV, lap time accuracy, parameter sweep

Acknowledgments

This bachelors thesis is not only a result of the hard work by the members of the project group, but also of the commitment from the rest of Chalmers Formula Student and our supervisors. Firstly, we would like to show our sincere gratitude towards all the members in Chalmers Formula Student that has guided us and helped us thorough the project. We would also like to thank our supervisors, Björn Pålsson and Christian Svensson, for their constant help and guidance during the whole thesis. Without them constantly being open to provide feedback, the project would not have been as successful.

Tim Albertsson, Isak Nilsson and Alexander Wigh
Göteborg, May, 2024

List of Acronyms

CFD Computational Fluid Dynamics
CFS Chalmers Formula Student
CoG Center of Gravity
CoP Center of Pressure
DC Driverless Cup
DoF Degree of Freedom
DRS Drag Reduction System
DV Driverless Vehicle
EV Electric Vehicle
FS Formula Student
FSG Formula Student Germany
GNSS Global Navigation Satellite Systems
GUI Graphical User Interface
IMU Inertial Measurement Unit
GGV Performance Envelope
LTS Lap Time Simulator
QSS Quasi Steady State
RPM Revolutions Per Minute
SS Steady State

Symbols

Parameters

ω_{motor}	Motor angular velocity $[\frac{rad}{s}]$
v	Vehicle speed $[\frac{m}{s}]$
$i_{\text{gear ratio}}$	Gear ratio [-]
β_{torque}	Torque bias [-]
β_{regen}	Regenerative braking bias [-]
β_{brake}	Brake bias [-]
$\beta_{\text{total brake}}$	Total brake bias, including mechanical and regenerative braking [-]
T	Torque $[Nm]$
P	Power $[W]$
p	Brake line pressure $[Pa]$
$F_{x,\text{mechanical brakes}}$	Longitudinal force generated by the mechanical brakes $[N]$
r_l	Loaded radius of the tyre $[m]$
r_{eff}	Effective brake disc radius $[m]$
A_c	Total caliper piston area $[m^2]$
D_c	Caliper piston diameter $[m]$
n_{pistons}	Number of caliper pistons [-]
μ_{pad}	Brake pad coefficient of friction [-]
μ_{lat}	Lateral coefficient of friction [-]
μ_{lon}	Longitudinal coefficient of friction [-]
μ_{drop}	Coefficient drop with normal force [-]
N	Normal force $[N]$
$N_{\text{load transfer}}$	Load transfer $[N]$
m_{total}	Total mass $[kg]$
a_y	Lateral acceleration $[\frac{m}{s^2}]$
a_x	Longitudinal acceleration $[\frac{m}{s^2}]$

C_D	Coefficient of Drag [-]
C_L	Coefficient of Lift [-]
ρ	Density of air $[\frac{kg}{m^3}]$
A_{frontal}	Frontal area of the car $[m^2]$
F_r	Rolling resistance force $[N]$
C_r	Rolling resistance coefficient [-]
$l_{\text{wheelbase}}$	Wheelbase $[m]$
h_{CoG}	Center of gravity height $[m]$
θ	Pitch angle $[rad]$
z	Ride height $[m]$
Θ	Angle between contact patch and the instantaneous center for pitch $[rad]$
k_t	Tyre stiffness $[\frac{N}{m}]$
k	Spring stiffness $[\frac{N}{m}]$
R	Corner radius $[m]$
δx	Corner segment length $[m]$
$F_{\text{downforce}}$	Downforce $[N]$
$E_{\text{motor losses}}$	Motor losses $[W]$
$E_{\text{inverter losses}}$	Inverter losses $[W]$

Contents

List of Figures

List of Tables

1

Introduction

Formula Student (FS) is an engineering competition where university students design, produce and compete with their own formula-style racing cars [1]. The teams compete in design, manufacturing, cost, business, and multiple dynamics events to test the car's performance on track. Around the world there are over 1000 Formula Student teams [2]. One of these teams is Chalmers Formula Student (CFS), which was founded in 2002 and has had great successes throughout the years, including winning the 2023 Formula Student Germany (FSG) competition in the driverless category [3]. Today CFS competes with a four-wheel drive electric car, that can be driven manually by a driver in the Electric Vehicle (EV) class and autonomously in the Driverless Vehicle (DV) class and Driverless Cup (DC). CFS is divided into eight subgroups, each developing a specific area of the car. These areas are Aerodynamics, Electronics, Frame, Suspension, Software, Autonomous, Mechanical Powertrain, and Business & Management.

During the development of an FS car, the choice of design concept and resource management are among the most critical factors for enhancing the performance of the car. Therefore, it is important to make data-driven decisions. The Formula Student project takes place over the course of a year, primarily involving completely new students with little prior knowledge. In just a few months, these students need to understand previous concepts in order to develop both existing and new components. During this process, it is crucial that information, knowledge, and tools for developing the car are readily accessible.

The FS competitions are divided into static and dynamic events, as outlined in the rules [4]. The static events include the Business Plan Presentation, Cost and Manufacturing, and Engineering Design. Additionally, there are eight dynamic events, split evenly between EV and DV/DC categories, to assess the car's on-track performance.

Specifically, the Skidpad event evaluates the car's lateral acceleration on a constant radius course, while the Acceleration event measures straight-line acceleration from a standstill. The Autocross event requires navigating a course shorter than 1.5 km for EV and between 200-500 m for DV. Exclusive to the EV class, the Endurance event tests performance over approximately 22 km on a track similar to the Autocross layout, with scores based on time and energy consumption. For the DV class, the Trackdrive event involves 10 laps on the DV autocross track, with scoring based on time and laps completed.

Given the variety of events in the Formula Student competitions, each with unique demands, the team needs to strategically allocate resources to optimize total points, as outlined in Table 1.1. Balancing performance across events is crucial to maximising overall score.

	EV	DC
Static Events:		
Business Plan Presentation	75 points	-
Cost and Manufacturing	100 points	-
Engineering Design	150 points	150 points
Dynamic Events:		
DV Skidpad	75 points	75 points
Acceleration	50 points	-
DV Acceleration	75 points	75 points
Autocross	100 points	-
DV Autocross	-	100 points
Endurance	250 points	-
Efficiency	75 points	-
Trackdrive	-	200 points
Overall	1000 points	600 points

Table 1.1: Maximum points awarded according to the 2024 FSG rules [4].

Over the past decades, lap time simulation has become an invaluable tool for optimizing a vehicle's performance around a track as it can be used to analyze measurement data, optimize the vehicle, strategy, and more [5]. A Lap Time Simulator (LTS) can show trends, predict, and quantify the performance of different car parameters for various tracks using a simplified mathematical model of the car. Lap time simulation models vary in complexity, from point-mass models to full-scale multi-body representations with nonlinear equations.

One of the competition elements in FS is Engineering Design, where the students' engineering skills and design work on the car are evaluated. During this event, it is important to demonstrate why decisions were made regarding the design of the car's components. This is an area where lap time simulation is a useful tool.

Within CFS, various vehicle representations and software solutions have been used over the years to simulate the different subsystems of the car. However, many of these tools have only been created for specific subgroups and used for a few years before being forgotten or lacking knowledge for continued use. The limited features and modularity of commercial LTS software created the need for an in-house lap time simulator in CFS. A custom-developed lap time simulation can be adapted to specifically suit the purposes of CFS and several of the subgroups.

2

Benchmarking

In an LTS, various assumptions must be made to construct models of the real world that can be simulated quickly while maintaining acceptable accuracy. This chapter will discuss these assumptions and models along with their strengths and weaknesses. Chapter 2 will also inform about already existing lap time simulators and present the requirements that CFS have on the LTS that has been developed in this project.

2.1 Vehicle Model

In the field of lap time simulation, there are several vehicle models of varying complexity. An increase in complexity also lead to higher simulation and development times [6]. A visualisation of the car with its coordinate system and rotations can be seen in Figures 2.1 - 2.3.

The simplest model of the car is a point-mass, which only has one degree of freedom: speed. In this model, the car is represented as a point in space with all its parameters. The point-mass model can include mass, aerodynamics, drivetrain, brakes, and a tyre model [7]. Figure 2.4 shows that point-mass models are of low complexity in terms of development and simulation time, but they also have lower usability compared to other models. As the model only includes one Degree of Freedom (DoF), the simulation time is short, making it the perfect choice when simulation time is of the essence, e.g. large parameter sweeps or optimizations.

The next step in complexity is a bicycle model, also known as single-track, which models the front and rear axles independently. Since the model includes two axles, behaviours such as longitudinal load transfer, and pitch (θ) during acceleration and braking, as well as tyre slip and self aligning torque can be considered. This increases the fidelity over the point-mass model [8]. Figure 2.4 shows that the bicycle model is more complex, but also more useful than the pointmass model.

The most advanced models include all four wheels, and one of them is called four-wheel model or two-track model. These models are more complex and can include effects such as lateral load transfers which makes the models more useful compared to the other mentioned, as parameters for each individual wheel can be evaluated [8]. There are multiple ways to create four-wheel models, but for Figure 2.4 these have all been combined as they are of the highest complexity, but also fidelity and usefulness. Generally, the more complex and advanced a vehicle model is, the more information about the track and vehicle is required. This information can sometimes be unknown och difficult to estimate.

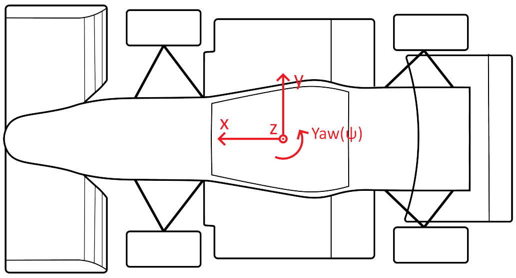


Figure 2.1: Top view of a FS car with coordinate system.

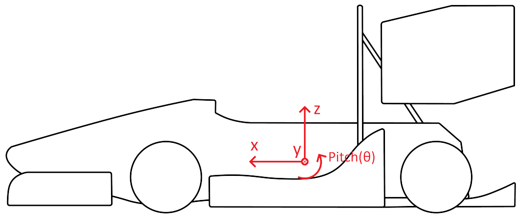


Figure 2.2: Side view of FS car with coordinate system.

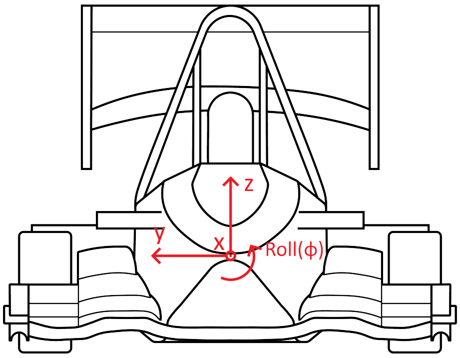


Figure 2.3: Front view of FS car with coordinate system.

2.2 Lap Time Solver

In addition to the vehicle model, there are also different levels of complexity for the simulation of a lap around the track, which can be seen in Figure 2.4. The simplest is called Steady State (SS), which considers variables as constant at each time step and thus piecewise constant around a lap. Due to this, only lateral acceleration through corners is considered, making it less useful compared to other types of simulation. The next level in complexity is Quasi Steady State (QSS), which means that both lateral and longitudinal accelerations occur in the system, but the car is still in steady state for each time step. With SS and QSS simulations, a Performance Envelope (GGV)-diagram can be used in the solver since the vehicle will always be in steady-state. A GGV-diagram describes the steady state longitudinal and lateral acceleration capabilities of the vehicle for different speeds which can be modelled from the chosen vehicle model.

Transient models are the most complicated, where the system is not considered to be in a steady state at every time step. This model is more complex than the two previously mentioned models, but also more useful as it better represents reality with more degrees of freedom and uses less assumptions. As the transient model is not in steady state for each time step, a GGV diagram is not applicable. Instead, calculations for the vehicles performance is required for every time step in the simulation, taking the current state of the car into consideration.

Incorporating an optimal path solver also called optimal control in the solver allows the simulation to determine the best possible driving line around the track [5]. By using this approach, the model can calculate the most efficient trajectory that minimizes lap times while adhering to the track's constraints and the vehicle's dynamics. Optimal control adjusts the car's path by considering various factors such as speed, acceleration, braking points, and cornering strategies. This optimization is typically achieved through advanced mathematical algorithms that solve for the optimal trajectory at each point on the track and will lead to higher simulation times.

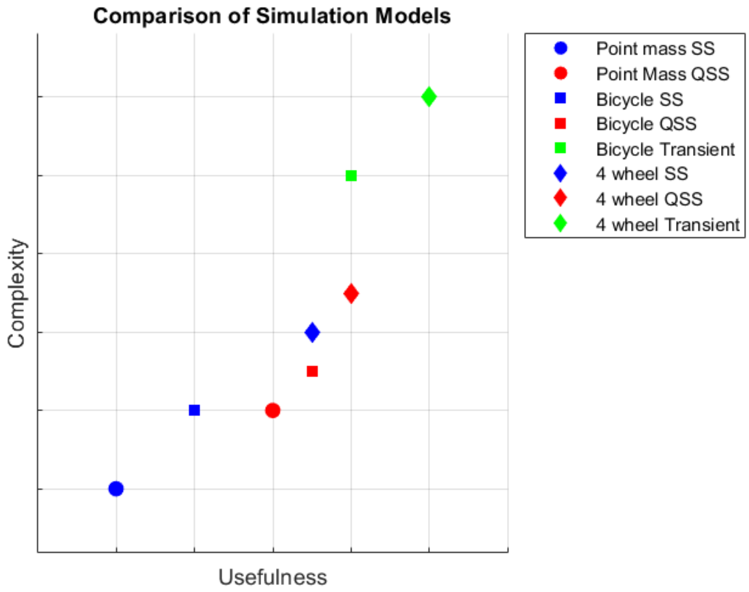


Figure 2.4: Comparison of LTS models complexity and usefulness. Adapted from [9].

2.3 Tyre Model

Tyres on a racing car are one of the most crucial components as they are the contact points between the car and the ground, making tyre modelling an important part of simulation. The simplest model for tyres is a constant coefficient of friction, which means that the tractive force linearly increases with normal load. For a better correlation with reality, a semi-empirical model like Pacejka, also known as the magic formula, can be used [8]. This model is built from tyre tests and includes several parameters to represent the tyre's performance . These parameters include, but are not limited to, normal force, slip angle, slip ratio and temperature. If only one parameter is desired, a simpler model that only includes that specific parameter can be constructed. This will be discussed further in Section 4.1.

2.4 Track Model

For the track, models can be created in different ways. The most intuitive way is to create a model of the real track, including geometry, track width, and potentially elevation [5]. This type of model is required if the simulation should also find the optimal path around the track, often using optimal control [5].

A simpler way to create a track is to follow a predefined path. This path can be created from Global Navigation Satellite Systems (GNSS) data, or lateral acceleration and speed from test data of a lap around the desired track [7]. As the model does not include any track width, optimal control can not be included in the simulation. However, this approach makes it easier to compare the simulated data with real telemetry data, as same path will be used. The path is defined by dividing the track into multiple sections, with section length and radius.

2.5 Available Lap Time Simulators

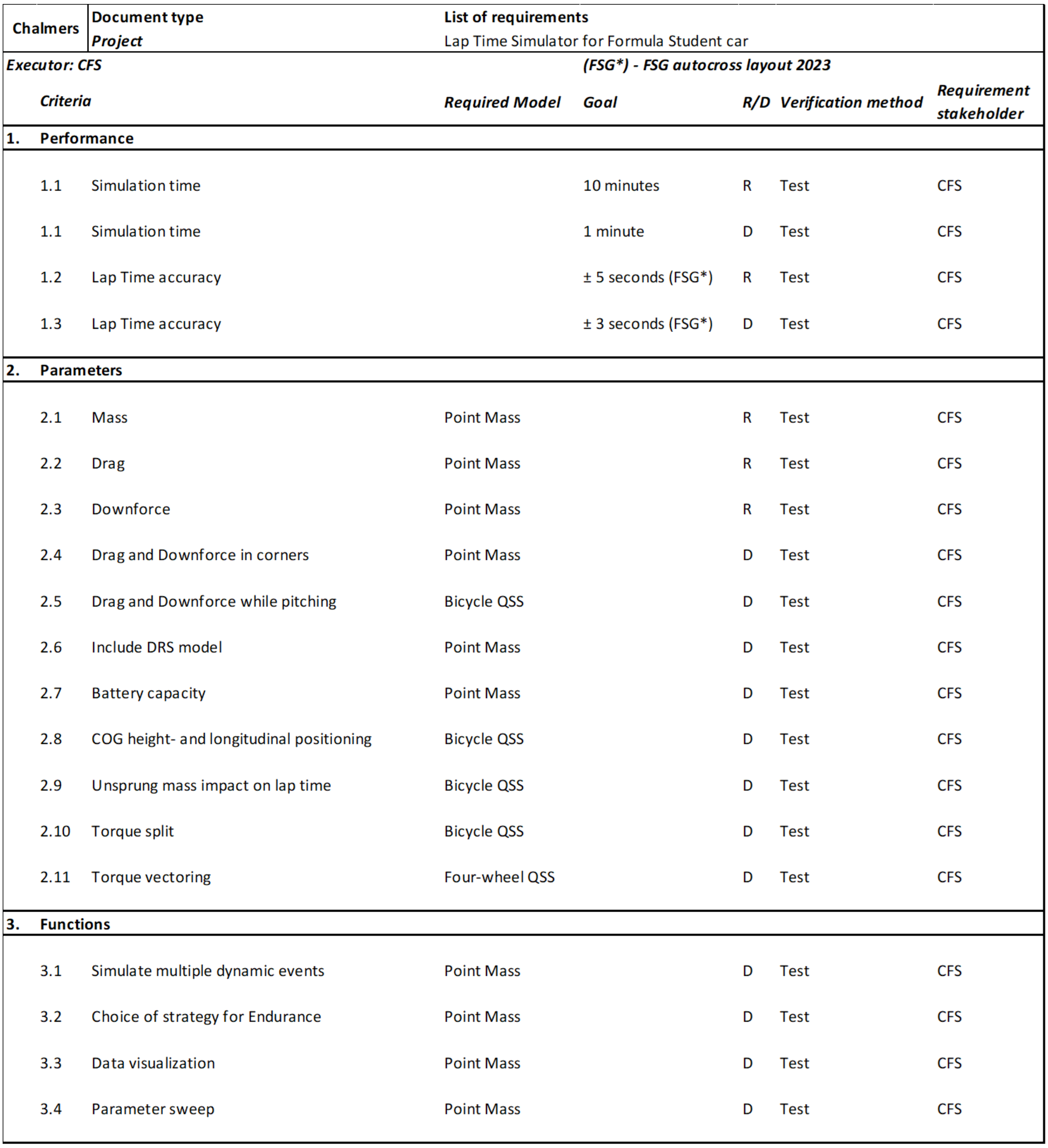
There are currently several commercial Lap Time Simulators on the market and CFS use a few of them. Optimum Lap, a simulator used by the aerodynamics department in CFS, is a point-mass QSS simulator [10]. It uses a predefined track path, as explained in Section 2.4 and car parameters as input, such as total weight, powertrain parameters, aerodynamics coefficients, and static tyre friction values. As the car's aerodynamic properties can be evaluated, it is useful for the aerodynamic department as they can test the impacts of aerodynamic coefficients on lap time but not considering changes in ride height, pitch, yaw, cornering or roll.

IPG CarMaker is a Lap Time Simulator used by CFS to evaluate controls parameters. CarMaker is according to the Controls subgroup quite complex and extensive in its features, but the documentation is difficult to understand. As a result, CFS use only a limited selection of the available features, while potential features that could be incorporated are often excluded due to their complexity and the time required for implementation.

The foundation for this thesis is that the commercial simulators do not satisfy CFS. Points calculations, multi event simulations and more advanced vehicle models are some of the things that the team desires beyond what existing simulators already provides. The teams wishes and requirements will be presented in Section 2.6.

2.6 Requirements

The aim of the project is to develop an LTS to CFS' requirements. To fulfill this, the preferences and needs of different groups and people within CFS for an in-house simulator were investigated. Interviews took place with managers and team members from the different subgroups in CFS. These interviews became the foundation for creating the list of requirements found in Figure 2.5 which consists of both required and desired functions.



Chalmers	**Document type** ***Project***		**List of requirements** Lap Time Simulator for Formula Student car			
Executor: CFS			***(FSG*) - FSG autocross layout 2023***			
	Criteria	***Required Model***	***Goal***	***R/D***	***Verification method***	***Requirement stakeholder***
1.	**Performance**					
1.1	Simulation time		10 minutes	R	Test	CFS
1.1	Simulation time		1 minute	D	Test	CFS
1.2	Lap Time accuracy		± 5 seconds (FSG*)	R	Test	CFS
1.3	Lap Time accuracy		± 3 seconds (FSG*)	D	Test	CFS
2.	**Parameters**					
2.1	Mass	Point Mass		R	Test	CFS
2.2	Drag	Point Mass		R	Test	CFS
2.3	Downforce	Point Mass		R	Test	CFS
2.4	Drag and Downforce in corners	Point Mass		D	Test	CFS
2.5	Drag and Downforce while pitching	Bicycle QSS		D	Test	CFS
2.6	Include DRS model	Point Mass		D	Test	CFS
2.7	Battery capacity	Point Mass		D	Test	CFS
2.8	COG height- and longitudinal positioning	Bicycle QSS		D	Test	CFS
2.9	Unsprung mass impact on lap time	Bicycle QSS		D	Test	CFS
2.10	Torque split	Bicycle QSS		D	Test	CFS
2.11	Torque vectoring	Four-wheel QSS		D	Test	CFS
3.	**Functions**					
3.1	Simulate multiple dynamic events	Point Mass		D	Test	CFS
3.2	Choice of strategy for Endurance	Point Mass		D	Test	CFS
3.3	Data visualization	Point Mass		D	Test	CFS
3.4	Parameter sweep	Point Mass		D	Test	CFS

Figure 2.5: The requirements stakeholders had on the simulator. R - Required functions. D - Desired functions

3

Lap Time Simulator Presentation

Chapter 3 will provide a short explanation on how the developed LTS works from a user point of view. It will show the optional simulation variants and functions that can be reached through the Graphical User Interface (GUI) and explain how the user can manage them.

3.1 Choice of Vehicle Model

A conclusion was drawn from the list of requirements and the benchmarking, a QSS model should be used with a point-mass and bicycle vehicle model. The LTS feature the option to choose between the models as they have different advantages and disadvantages. For the time frame given for the project, a point-mass and bicycle model was suitable to focus the project on the modularity and documentation of the code but also its features.

In all the different kinds of simulations that the user can choose from, the simulation model needs to be specified. They are different from each other in complexity and usefulness, as seen in Figure 2.4, and thus the user is given the opportunity to choose between them depending on what is going to be simulated.

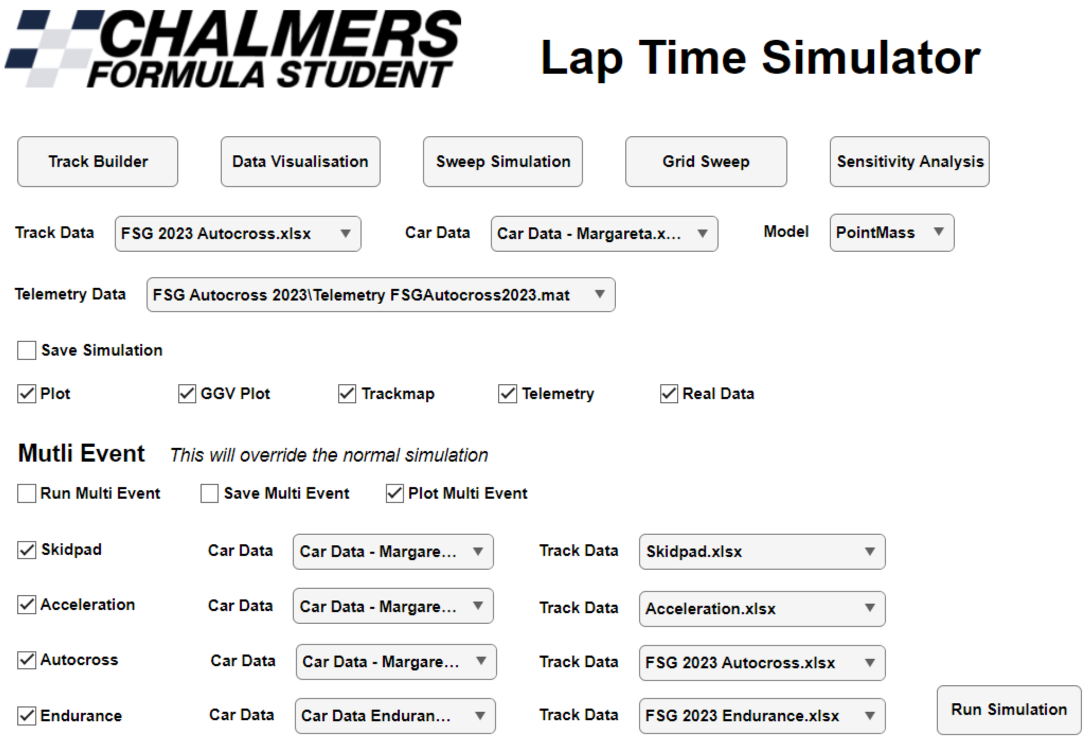
3.2 Input Data

All information and properties of the vehicle is stored in Excel files. Depending on what event is simulated there are different car data files. Should the user want to simulate a change in one parameter it is most suitable to do this by a sweep study. However, if CFS would like to test next years car, a new car data file will be created. The selection of input is done entirely by the user in the GUI.

For every simulation a track needs to be specified. Every event is driven on different track layouts and therefore the program gives the user the option to choose track before every simulation however this does not apply for the multi event simulation where all tracks are used.

3.3 User Interface

For ease of use, a GUI was created using MATLAB's Design App. The interface is split to multiple different windows, all controlling a specific part of the LTS. The most common simulations and plots have been included for the user, but if a specific simulation or output is desired, the data is also stored in the MATLAB workspace or as a file for later use. The main window in the GUI can be seen in Figure 3.1. It includes controls for a general lap simulation and multi event simulation, but also buttons to open the other GUI windows. These windows can be found in Appendix B.



Track Data FSG 2023 Autocross.xlsx ▼ Car Data Car Data - Margareta.x... ▼ Model PointMass ▼

Telemetry Data FSG Autocross 2023\Telemetry FSGAutocross2023.mat ▼

☐ Save Simulation

☑ Plot ☑ GGV Plot ☑ Trackmap ☑ Telemetry ☑ Real Data

Mutli Event *This will override the normal simulation*

☐ Run Multi Event ☐ Save Multi Event ☑ Plot Multi Event

☑ Skidpad Car Data Car Data - Margare... ▼ Track Data Skidpad.xlsx ▼

☑ Acceleration Car Data Car Data - Margare... ▼ Track Data Acceleration.xlsx ▼

☑ Autocross Car Data Car Data - Margare... ▼ Track Data FSG 2023 Autocross.xlsx ▼

☑ Endurance Car Data Car Data Enduran... ▼ Track Data FSG 2023 Endurance.xlsx ▼

Figure 3.1: GUI interface created in MATLAB's Design App. This is the main window that appears when you initiate the GUI.

3.4 Single and Multi Event Simulations

The most basic simulation involves simulating a single FS event. Users can select the track, event, car data file, choice of plots, and simulation model through the GUI. Figure A.2 in Appendix A illustrates how the simulation results could look like.

A multi event runs simulations for multiple FS events at the same time. This is useful in cases where changes of car parameters that affect all or multiple events are evaluated. An example of a normal multi event simulation can be seen in Figure 3.2, but it is also possible to do multi event simulations for sweeps. This concept will be explained further in the next sections.

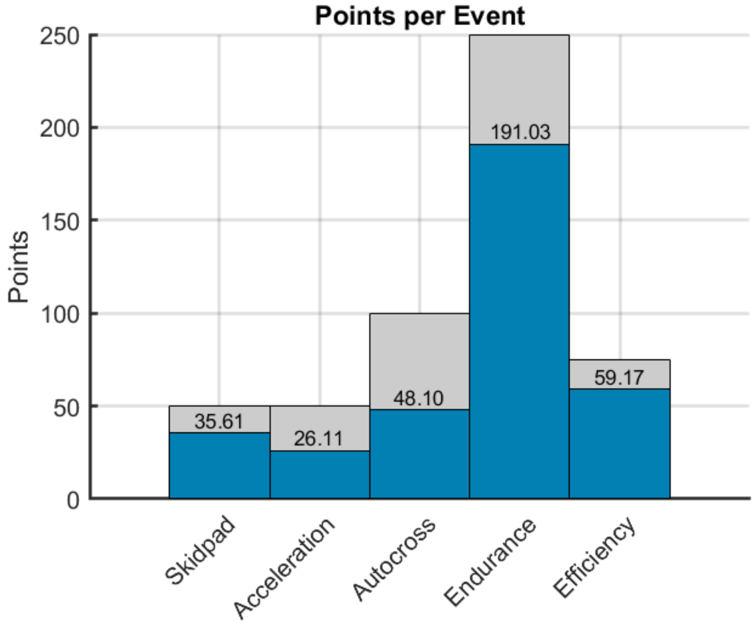


Figure 3.2: Multi Event Simulation. The grey areas show maximum points for that event [4].

3.5 Parameter Sweep

A parameter sweep is an analysis on how the change of a parameter affects the score and lap time. The simulator includes three different kinds of parameter sweeps that the user can attain from the GUI.

A single parameter sweep studies one parameter at a time. The user specifies the range over which the parameter should vary and the number of points within this range. The program then runs a simulation for each different value of the parameter. For instance, a parameter sweep could be used to assess how an increase in mass affects the results by varying the car's mass from 0-10% above its base value. An example sweep is shown in Figure 3.3.

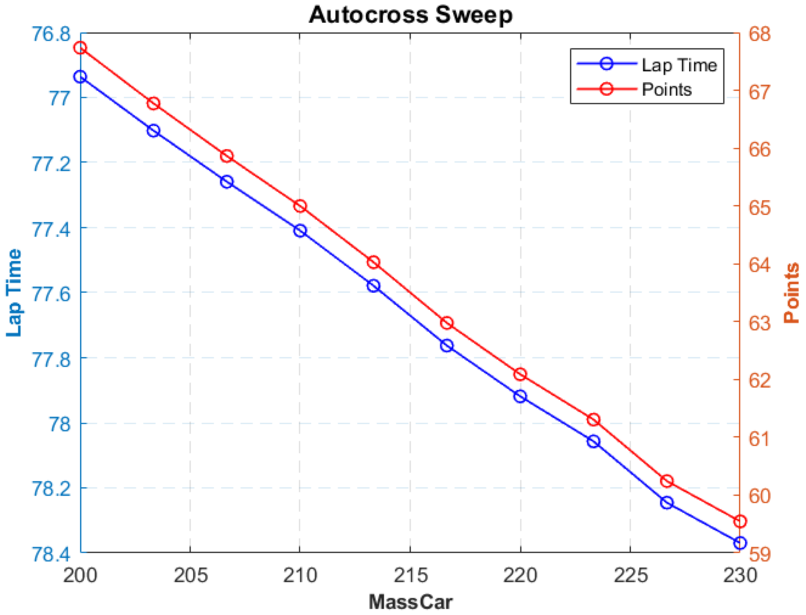


Figure 3.3: Sweep simulation of the 2023 FSG Autocross. Mass swept from 200 to 230 kg. Plot includes lap time and points scored.

A grid sweep investigates two parameters, in the same way as a single parameter sweep, against each other and finds the best combination in order to get the best results. If the user chooses to have ten data points in each range, the grid sweep will generate $10 \cdot 10 = 100$ combinations. This is applicable if there are two parameters, for example maximum torque and maximum RPM, that affect the result in different ways and the best combination needs to be found. An example of a grid sweep can be seen in Figure 3.4.

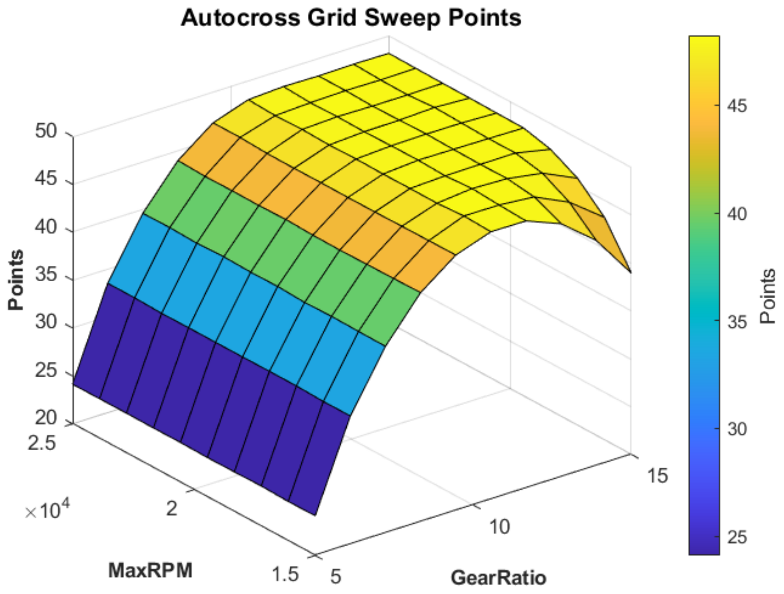


Figure 3.4: Grid Sweep between gear ratio and maximum RPM for an Autocross event.

A sensitivity analysis is performed by looking at e.g. a 10% improvement for a couple of parameters. The aim is to visualize what parameter and area of the car that is most profitable to work on in terms of points and lap time during the design phase. An example can be seen in Figure 3.5.

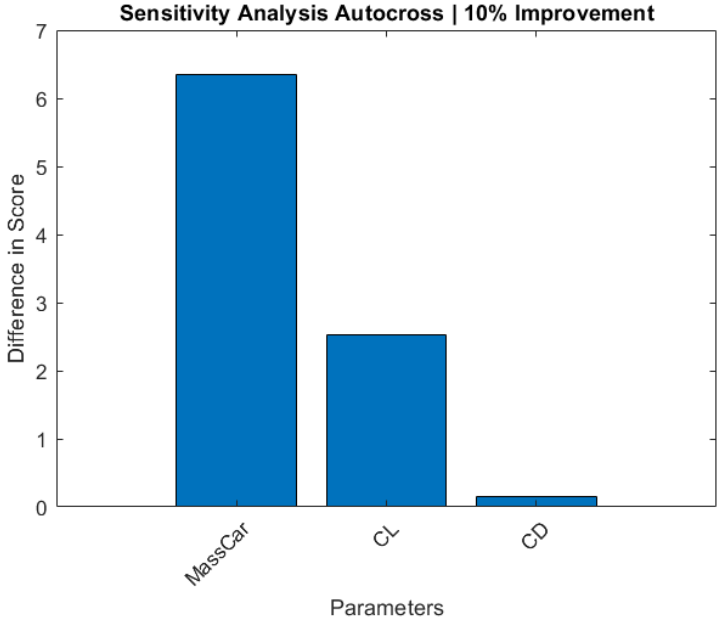


Figure 3.5: Sensitivity analysis that studies 10% decrease in the car's mass, 10% increase in C_L and 10% decrease in C_D at the 2023 FSG Autocross. Points scored visualised.

4

Lap Time Simulator Implementation

This chapter will describe the areas of science that were used to achieve the development of the LTS. Chapter 4 will also explain the theory behind the most crucial parts of the simulator such as the solver, GGV-diagram and vehicle models. The reader will get to know how these components are associated with each other in the LTS.

4.1 Tyre Model

A simple tyre model with a constant friction coefficient was ruled out. It did not meet all requirements for an accurate simulator as the normal load dependent friction was found to be quite substantial according to tyre tests preformed by Calspan for the Hoosier LC0 tyre used by CFS [11]. Another common option, as stated in Section 2.3, is the Pacejka Magic Tyre formula. This model comes in many different configurations, but they all model a tyre from multiple input parameters that govern the tyre's performance. This is a good option in a lot of cases, but was ruled out for this project as the added complexity and increased number of DoFs would not result in any significant improvement over the final model that was chosen. Other parameters found in the Pacejka model, such as slip angle and self aligning torque, were not used in the LTS and therefore deemed unnecessary. Further discussion regarding tyre modeling is found in 4.3.

For this LTS, a model was chosen that includes the friction drop-off with normal load, but excludes other parameters typically found in the Pacejka model. One model for lateral and one for longitudinal friction was created as the grip level of the tyre differs for lateral and longitudinal cases. This model is a good option in between the previously two stated as it only includes the desired input parameter, normal load. In the simulations, normal load was found to be in the range of 500 to 1500 N per wheel. Calspan's tyre test procedure typically involves mounting tyres onto a test rig and subjecting them to varying loads and rotations while measuring friction, normal force, slip angle, and tyre deformation across different operating conditions such as speeds, road surfaces, and temperatures [11]. The tyre tests conducted by Calspan, as illustrated in Figures 4.1 and 4.2, show a linear drop-off in friction when the normal force exceeds 500 N. Consequently, a linear model was selected to represent the friction drop-off. Equations for these models can be seen below.

$$\mu_{\text{lat}} = \mu_{\text{lat, 0}} - \mu_{\text{drop, lat}} \cdot N_{\text{tyre}} \tag{4.1}$$

$$\mu_{\text{lon}} = \mu_{\text{lon, 0}} - \mu_{\text{drop, lon}} \cdot N_{\text{tyre}} \tag{4.2}$$

where μ_{lat} is the final lateral friction coefficient, $\mu_{\text{lat, 0}}$ is the friction coefficient with no normal load, $\mu_{\text{drop, lon}}$ is the drop of in friction per newton normal load applied, and N_{tyre} is the normal load applied to the tyre. The same variables apply in the longitudinal case. The model can be seen as a blue line in the figures. The initial coefficients of friction, $\mu_{\text{lat, 0}}$ and $\mu_{\text{lon, 0}}$, was calibrated with CFS test data and the load dependency behaviour was adjusted to match Calspan test data.

The rolling resistance of the tyres is also a contributing factor to the longitudinal acceleration capabilities of the car. The rolling resistance force F_r is directly proportional to the rolling resistance coefficient C_r and the normal force N, as described by the following equation [12].

$$F_r = C_r \cdot N \tag{4.3}$$

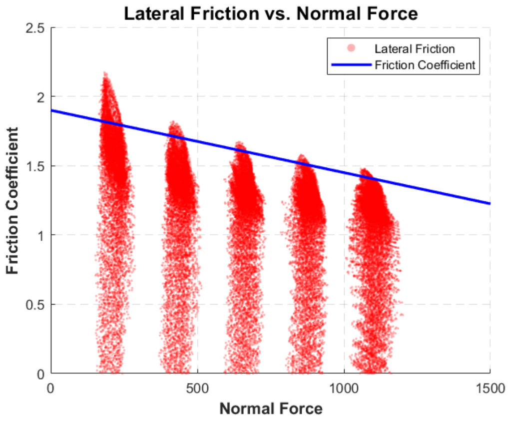


Figure 4.1: Lateral friction model with friction drop-off compared to Calspan test data

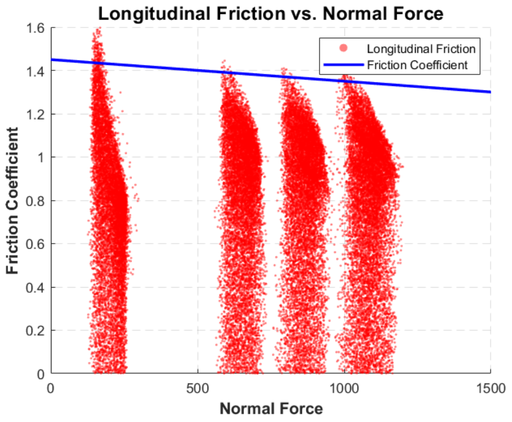


Figure 4.2: Longitudinal friction model with friction drop-off compared to Calspan test data

4.2 Point-Mass Model

The first vehicle model in the LTS is a point-mass model. A point-mass model assumes that the car is a single point moving along the track. This “point” has the same mass, tyre friction coefficients and aerodynamic coefficients as the real car. What is not included in this model is any type of load transfer, yaw, pitch, roll.

The aerodynamic modelling in the LTS relies on coefficients derived from CFS Computational Fluid Dynamics (CFD) simulations and testing. To effectively assess the aerodynamics of a race car, these coefficients are essential as they provide an accurate representation of how downforce and drag scale with the square of velocity (v) [8]. The downforce $F_{\text{downforce}}$ and drag F_{drag} equations are presented in the equations below.

$$F_{\text{downforce}} = \frac{1}{2} C_L \cdot \rho \cdot A_{\text{frontal}} \cdot v^2 \tag{4.4}$$

$$F_{\text{drag}} = \frac{1}{2} C_D \cdot \rho \cdot A_{\text{frontal}} \cdot v^2 \tag{4.5}$$

Here, C_L denotes the coefficient of lift, C_D is the coefficient of drag, ρ represents the density of air, and A_{Frontal} is the frontal area of the car.

According to CFS' CFD-simulations, the coefficients $C_L, C_D, A_{\text{frontal}}$ sometimes changes with velocity, ride height, pitch, corner radius and yaw angle which makes it important to include the effects in the LTS. For the point-mass model, the change of coefficient of lift and drag with velocity is modelled with an interpolation from the simulation values. With the downforce values known, the normalforce N can be calculated with the following equation, where $F_{\text{downforce}}$ is the generated downforce and m_{total} is the total car mass including driver.

$$N = F_{\text{downforce}} + m_{\text{total}} \cdot g \tag{4.6}$$

4.2.1 Torque Vectoring

The 2023 CFS car has electric in-hub motors which means that each motor can be individually controlled, this is called torque vectoring. CFS currently has three models of torque vectoring, TV1, TV2 and TV3. For the lap time simulator, TV2 is simplified and implemented with a torque and regenerative bias for the front and rear axle. The parameter torque bias value states the % of rear torque on the front axle and regenerative bias states the % of front torque on the rear axle.

TV2 was chosen as the torque vectoring model for the lap time simulator since it was the main model used by CFS in 2023 and because TV3 is still under development.

4.2.2 Torque Curve

The torque curve for the electric motors on the CFS car can be modelled from the max torque and power limit of 80 kW or lower, as stated in the FS rules [4]. Equation 4.8 can be derived from the power (P) equation and modified to consider the vehicle speed (v), gear ratio ($i_{\text{gear ratio}}$) and rolling radius (r_l).

$$P = T \cdot \omega_{\text{motor}} \tag{4.7}$$

$$P = \frac{T \cdot i_{\text{gear ratio}} \cdot v}{r_l} \tag{4.8}$$

This can now be used for calculating the allowed maximum torque per front ($T_{F,\text{max}}$) and rear motor ($T_{R,\text{max}}$) in Equations 4.10-4.11. Additionally, torque bias (β_{torque})

is factored into these equations due to its notable impact on power output (P_{max}).

$$T_{R,\text{power limit}} = \frac{P_{\text{max}} \cdot r_l}{2 \cdot (1 + \beta_{\text{torque}}) \cdot v \cdot i_{\text{gear ratio}}} \tag{4.9}$$

$$T_{R,\text{max}} = min(T_{R,\text{torque limit}}, T_{R,\text{power limit}}) \tag{4.10}$$

$$T_{F,\text{max}} = T_{R,\text{max}} \cdot \beta_{\text{torque}} \tag{4.11}$$

As seen in Figure 4.3 and 4.4, the modelled torque curve approximates the torque curve generated from CFS test data.

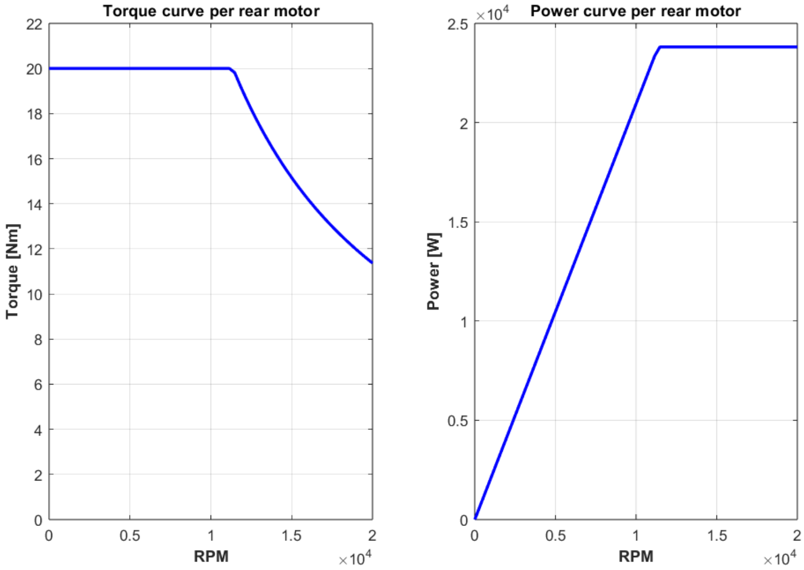


Figure 4.3: Modelled torque and power curve per rear motor.

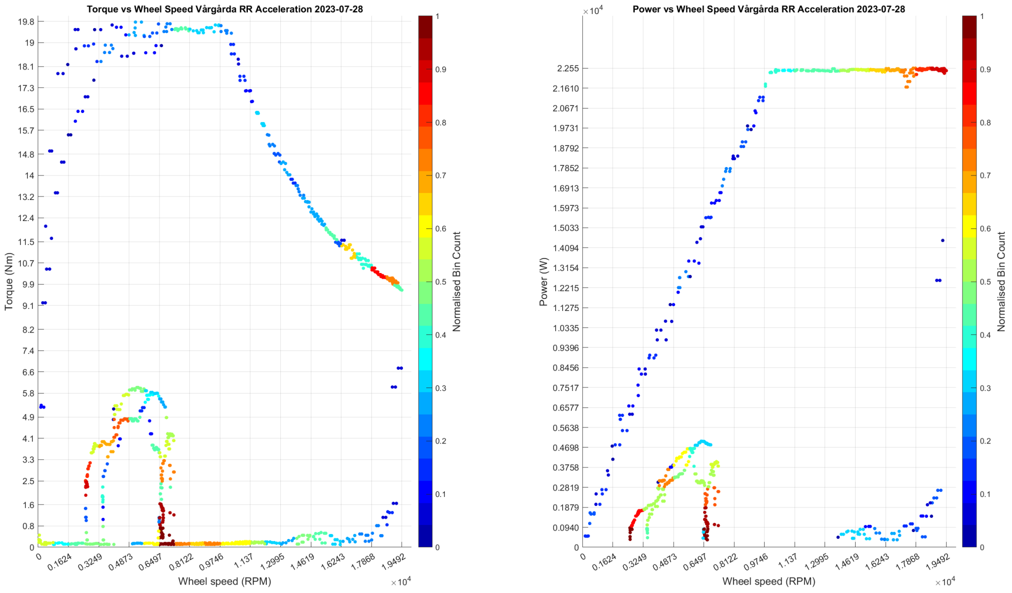


Figure 4.4: Torque and power curve per rear motor from CFS acceleration test data, manual driving.

4.2.3 Braking

The case of braking can be divided into two parts in the CFS car, regenerative and mechanical braking. The mechanical braking force can be calculated using equations from [13]. By using the maximum pressure the drivers are capable of pushing (p_{driver}) on the pedal according to CFS test data, the maximum braking force generated on the ground plane by the wheel $F_{x,\text{mechanical brakes}}$ can be calculated as:

$$F_{x,\text{mechanical brakes}} = \frac{p \cdot A_c \cdot \mu_{\text{pad}}}{R_l / r_{\text{eff}}} \tag{4.12}$$

Here, A_c represents the caliper piston area, μ_{pad} denotes the brake pad friction, and r_{eff} signifies the effective radius of the brake pad calculated from the mean radius of outer (r_o) and inner (r_i) radius of the brake disc. Next, calculating the total caliper piston area using the piston diameter (D_c) and number of pistons ($n_{pistons}$) is needed for calculating the mechanical braking force ($F_{x,\text{mechanical brakes}}$).

$$A_{c,F} = n_{\text{pistons},F} \cdot \frac{\pi \cdot D_{c,F}^2}{4} \tag{4.13}$$

$$A_{c,R} = n_{\text{pistons},R} \cdot \frac{\pi \cdot D_{c,R}^2}{4} \tag{4.14}$$

The equations from [13] do not consider a balance bar. To integrate brake balance (β_{brake}), pressure bias data from CFS is used in the following equations.

$$F_{x,\text{mechanical brakes},F} = 2 \cdot \frac{p_{\text{driver}} \cdot \beta_{\text{brake}} \cdot A_{c,F} \cdot \mu_{\text{pad}}}{R_l / r_{\text{eff}}} \tag{4.15}$$

$$F_{x,\text{mechanical brakes},R} = 2 \cdot \frac{p_{\text{driver}} \cdot (1 - \beta_{\text{brake}}) \cdot A_{c,R} \cdot \mu_{\text{pad}}}{R_l / r_{\text{eff}}} \tag{4.16}$$

$$F_{x,\text{mechanical brakes}} = F_{x,\text{mechanical brakes},F} + F_{x,\text{mechanical brakes},R} \tag{4.17}$$

The 2023 CFS car also uses regenerative braking from the electric motors on all four wheels. To get the maximum deceleration available, the regenerative braking contribution need to be added in the deceleration equations. The regenerative capabilities of the motors has a torque curve with a the same power limit as for the acceleration [4]. In 2023, CFS limited the regenerative braking torque to 15 Nm. This was primarily influenced by driver feedback and confidence, coupled with the performance of the mechanical brakes. The amount of regenerative braking torque is based on the mechanical brake pressure and is linear with a maximum pressure ($p_{\text{max regen}}$). The regenerative torque curve calculated in Equations 4.18-4.20 was generated using the same method as the calculation of the motor torque in Equations 4.8-4.11.

$$T_{\text{regen power limit},F} = \frac{P_{\max} \cdot r_l}{2 \cdot (1 + \beta_{\text{regen}}) \cdot v \cdot i_{\text{gear ratio}}} \tag{4.18}$$

$$T_{\text{regen},F} = -min(T_{\text{regen torque limit},F}, T_{\text{regen power limit},F}) \tag{4.19}$$

$$T_{\text{regen},R} = T_{\text{regen},F} \cdot \beta_{\text{regen}} \tag{4.20}$$

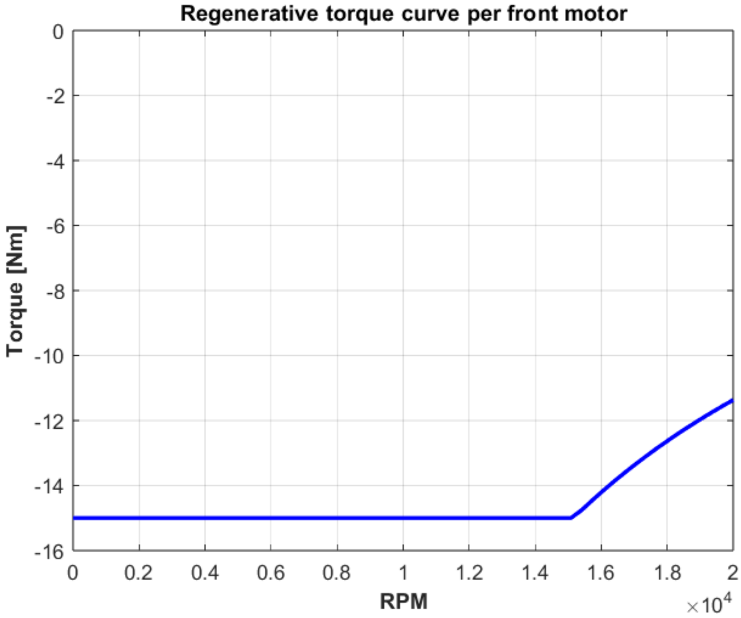


Figure 4.5: Modelled regenerative braking torque curve per front motor.

4.2.4 Performance Envelope Diagram for Point-Mass

The GGV-diagram plays an important role in the LTS as it models the vehicles capabilities. One graph of the maximum acceleration is created per velocity in the diagram. For the point-mass model, maximum lateral (a_y) and longitudinal (a_x) accelerations are calculated with the following equations:

$$a_{y,\text{max}} = \mu_{\text{lat}} \cdot \frac{N}{m_{\text{total}}} \tag{4.21}$$

$$a_{x,\text{max}} = \mu_{\text{lon}} \cdot \frac{N}{m_{\text{total}}} \tag{4.22}$$

Points are then spaced out between the maximum lateral acceleration in both directions to create a mesh. For each point, the remaining available grip in longitudinal acceleration is calculated with the following equation.

$$a_{x,\text{point}} = a_{x,\text{max}} \cdot \sqrt{1 - \frac{a_{y,\text{point}}}{a_{y,\text{max}}}} \tag{4.23}$$

Figure 4.6 shows what the finished ellipse, known as a G-G diagram, looks like. However, this only accounts for the maximum grip of the tyres. Finalizing the G-G diagram involves considering resistive forces such as rolling resistance and drag, as well as motor and braking capabilities. For motor parameters, the torque curve calculated in Section 4.2.2 is applied. If the available torque limits the acceleration more than the tyre's grip, the maximum acceleration from the motors is used instead, as it represents the limiting factor on the acceleration. The same logic is applied for the braking capabilities of the car, taking mechanical and regenerative braking in to account. The resulting G-G diagram can be seen in Figure 4.7. It is only limited in acceleration since the braking capabilities are greater than the tyre's grip in this case.

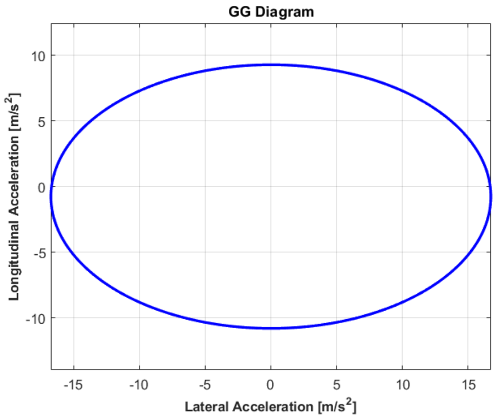


Figure 4.6: G-G diagram at 40 km/h

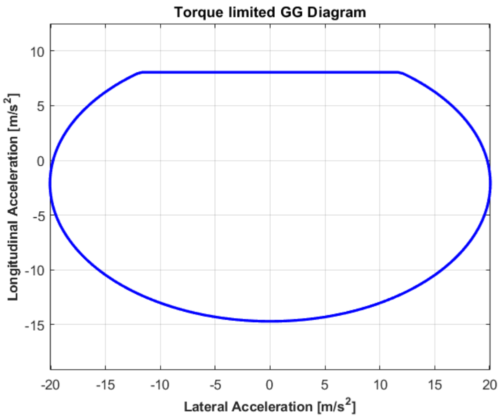


Figure 4.7: Torque limited G-G diagram at 80 km/h

Combining multiple velocities spaced evenly between 0 and the vehicles top speed, generates the performance envelope. The resulting figure, shown in Figure 4.8, serves as the basis for the solver when solving the simulation. Further explanation will be provided in Section 4.5.

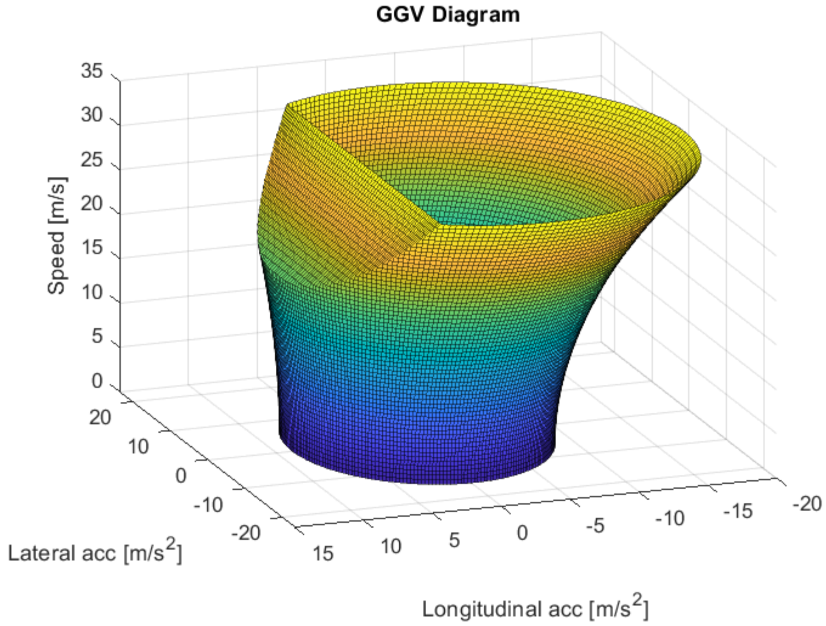


Figure 4.8: Performance envelope for point-mass model. It shows the available accelerations of the car at different speeds.

4.3 Bicycle Model

The next step in vehicle modelling was to create a bicycle model, which offers a more sophisticated representation of the vehicle's dynamics. Compared to the point-mass model, this approach also considers longitudinal weight distribution and load transfer. The steady state longitudinal load transfer refers to normal force change between the front and rear axles and is a function of the wheelbase ($l_{\text{wheelbase}}$), Center of Gravity (CoG) height (h_{CoG}), and the braking or acceleration force shown in the following equation [13].

$$\Delta N_{\text{load transfer}} = \frac{m_{\text{total}} \cdot a_x \cdot h_{\text{CoG}}}{g \cdot l_{\text{wheelbase}}} \tag{4.24}$$

Furthermore, this shift leads to changes in the vehicle's pitch angle and influences its aerodynamic performance. By incorporating these dynamics, the bicycle model provides a more accurate simulation of vehicle behavior, particularly during acceleration and braking maneuvers.

Implementing a bicycle model into the LTS involved updating the equations for the vehicle model in the GGV-diagram and adding the required parameters. With the front and rear axles split, the aero load is distributed on the axles by taking the load transfer caused by drag into account. An important aspect of this distribution is the aerodynamic balance or downforce distribution across the axles. This is determined by a moment balance, as the drag force influences where on the vehicle the force acts. While the resultant of the aerodynamic forces, F, can be described as the square root of the sum of the squares of lift and drag $F = \sqrt{F_{\text{downforce}}^2 + F_{\text{drag}}^2}$, the distribution of these forces and moments across the vehicle determines the Center of Pressure (CoP).

The choice was made to not include any slip angle or slip ratio for the tyre model. This is the reason why a Pacejka tyre model would not result in any significant improvement over the chosen tyre model. The model instead assumes the tyres are at optimum slip angle and ratio at all time steps.

4.3.1 Performance Envelope Diagram for Bicycle Model

As mentioned above in Equation 4.24, steady state longitudinal load transfer alters the normal force on the front and rear axle, thus changing the vehicle's acceleration capabilities. This implies that the load transfer depends on the longitudinal acceleration and also influences it. In other words the problem is that the convergence of the equilibrium need to be found through iterative solving where CasADi [14] and its *ipopt* optimization routine was used. CasADi is a symbolic framework designed for automatic differentiation and numerical optimization. It is useful in optimization and optimal control, where complex mathematical models need to be efficiently optimized or analyzed.

Similarly to the point-mass model, a performance envelope can be created for the bicycle model. First, maximum longitudinal acceleration need to be calculated. This, as mentioned, is done by the iterative optimization solver ipopt. The solver maximises longitudinal acceleration and finds one value for longitudinal acceleration and one for deceleration. Points are then spaced out between these values as explained in Section 4.2.4 to create a mesh. To find the maximum lateral acceleration for each point the following equations are used.

$$a_{y,\text{max}} = \frac{N_F \cdot \mu_{\text{lat},F} + N_R \cdot \mu_{\text{lat},R}}{m_{\text{total}}} \tag{4.25}$$

$$a_{y,\text{point}} = a_{y,\text{max}} \cdot \sqrt{1 - \left(\frac{a_{x,\text{point}}}{a_{x,\text{max}}}\right)^2} \tag{4.26}$$

The torque limit and the braking capabilities are then applied to the model at every velocity in the mesh, creating the performance envelope.

4.3.2 Pitch implementation

The pitch angle implementation adds another DoF and more equations are necessary. Calculating the pitch angle θ can be based on the front ride height z_F, rear ride height z_R and wheelbase in the following equation.

$$\theta = \arcsin\left(\frac{z_R - z_F}{WB}\right) \tag{4.27}$$

When calculating the front and rear axle ride heights, suspension geometry and stiffness need to be considered. Suspension geometry is the design of how the wheels and chassis are linked together. Suspension geometry can introduce anti-effects, which means that parts of the load transfer goes into the chassis instead of the springs and dampers, thus decreasing the pitching attitude of the vehicle [13]. However, suspension antis do not affect the steady state load transfer at the contact patch. As the Formula Student car has outboard brakes and electric in-hub motors that produces a reaction force on the uprights, the deceleration case will give the anti-effects as described by the equations below.

$$\% \text{ anti-dive} = \frac{\beta_{\text{total brake}} \cdot \tan \Theta_F \cdot l_{\text{wheelbase}}}{h_{\text{CoG}}} \tag{4.28}$$

$$\% \text{ anti-lift rear} = \frac{(1 - \beta_{\text{total brake}}) \cdot \tan \Theta_R \cdot l_{\text{wheelbase}}}{h_{\text{CoG}}} \tag{4.29}$$

With $\beta_{\text{total brake}}$ being the brake bias taking account for both regenerative and mechanical braking and Θ being the angle between the ground plane and the instantaneous center for pitch [13]. The front and rear ride height for the deceleration case is calculated in the following equations.

$$z_{\text{dec},F} = z_0 - \frac{F_{\text{aero},F}(\theta) - N_{\text{load transfer}} \cdot (1 - \% \text{ anti-dive})}{2k_F} - \frac{F_{\text{aero},F}(\theta) - N_{\text{load transfer}}}{2k_{t,F}} \tag{4.30}$$

$$z_{\text{dec},R} = z_0 - \frac{F_{\text{aero},R}(\theta) + N_{\text{load transfer}} \cdot (1 - \% \text{ anti-lift rear})}{2k_R} - \frac{F_{\text{aero},R}(\theta) + N_{\text{load transfer}}}{2k_{t,R}} \tag{4.31}$$

Here, k and k_t is the spring and tyre stiffness. For the acceleration case, the antis is instead based on torque bias.

$$\% \text{ anti-lift front} = \frac{\beta_{\text{torque}} \cdot \tan \Theta_F \cdot l_{\text{wheelbase}}}{h_{\text{CoG}}} \tag{4.32}$$

$$\% \text{ anti-squat} = \frac{(1 - \beta_{\text{torque}}) \cdot \tan \Theta_R \cdot l_{\text{wheelbase}}}{h_{\text{CoG}}} \tag{4.33}$$

Giving the front and rear ride heights:

$$z_{acc,F} = z_0 - \frac{F_{\text{aero},F}(\theta) - N_{\text{load transfer}} \cdot (1 - \% \text{ anti-lift front})}{2k_F} - \frac{F_{\text{aero},F}(\theta) - N_{\text{load transfer}}}{2k_{t,F}} \tag{4.34}$$

$$z_{\text{acc},R} = z_0 - \frac{F_{\text{aero},R}(\theta) + N_{\text{load transfer}} \cdot (1 - \% \text{ anti-squat})}{2k_R} - \frac{F_{\text{aero},R}(\theta) + N_{\text{load transfer}}}{2k_{t,R}} \tag{4.35}$$

The aeromap from CFS has simulation results for different pitch angles which is used with the interpolate function from CasADi in the pitch implementation. Unfortunately the implementation did not operate successfully due to complications in the solver, resulting in an inconsistent mesh. The choice was made to not include pitch in the final LTS as a consequence of this error. Future development will be discussed in the Section 7.

4.4 Track Model

For the track, a predefined path was chosen as it satisfied the requirements best. As the tracks in FS are only around 3 meters wide, an optimal line solver was deemed unnecessary [4]. Having the path predefined from test data also makes it easier to correlate test data and therefore also easier fit the simulation parameters to the real data.

As the sample frequency of CFS' GNSS data is only 1 Hz, the data is too coarse and could therefore not be used for recreating the track. Instead the track is created using the vehicle velocity estimated from wheel speed sensors and lateral acceleration data from the Inertial Measurement Unit (IMU), which both has a frequency of 100 Hz. The track model divides the track into sections, which are characterized by two variables: segment length and radius. The radius (R) is determined by the formula $R = \frac{v^2}{a_x}$. Additionally, the corner direction is stored for each section

Using MATLAB's *resample* function, it is possible to scale down the number of sections for the track at the expense of detail. This is done to decrease simulation time as more sections also leads to more calculations in the solver. An example of a generated track can be seen in Figure 4.9. The track is not connected because an Autocross track in FS has separate points for start and finish, but also because of slight inaccuracies in the measurements over time, known as sensor drift.

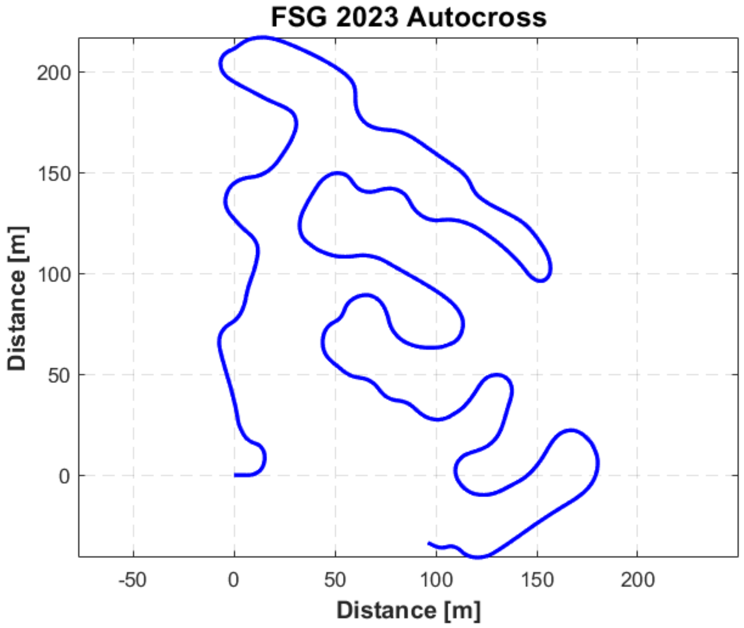


Figure 4.9: The 2023 FSG Autocross track generated from CFS telemetry data.

4.5 Solver

For the solver, a forwards-backwards QSS model was chosen. This solver includes lateral and longitudinal acceleration, but keeps the car in steady state in each time step [15]. Therefore no transient behavior can be evaluated with this solver. A simplified flowchart of the process can be seen in Figure 4.11.

The solver starts by calculating the maximum velocity in each corner segment of the track based on the maximum feasible lateral acceleration.

$$v = \sqrt{a_x \cdot R} \tag{4.36}$$

To find the maximum velocity, the solver loops through the GGV diagram until it finds the maximum speed for the given radius that the vehicle model can handle. This gives us the steady state velocity profile and is the only part required if an SS model is desired.

For the QSS part of the solver, longitudinal acceleration need to be considered. From each segment of the track, both braking and acceleration is calculated and compared to the previously calculated segments. The solver takes the calculated maximum velocity for the corner segment and then looks at the next segment of the track. If that segment radius is larger than the previous, an acceleration can be calculated. From the previous segments, velocity and radius of the next segment, a maximum available longitudinal acceleration is found and the new velocity is calculated with the following formula.

$$v_{i+1} = \sqrt{v_i^2 + 2a_x \cdot \delta x} \tag{4.37}$$

Where a_x is the available longitudinal acceleration and δx is the segment length. The solver continues to calculate the next segment until a velocity higher than that segments previously calculated velocity is found. This is done for every segment of the track, resulting in the velocity profile for the acceleration case. The same is then applied for deceleration, now looking at the negative longitudinal acceleration in the GGV diagram and calculating backwards from each segment which gives the deceleration velocity profile.

To find the final velocity in each segment, the minimum velocity of the previously calculated profiles is chosen. A velocity profile can be seen in Figure 4.10 and the lap time is calculated by taking the sum of every segment time with the following formula.

$$\text{Lap Time} = \sum t_i = \sum \frac{\delta x_i}{v_i} \tag{4.38}$$

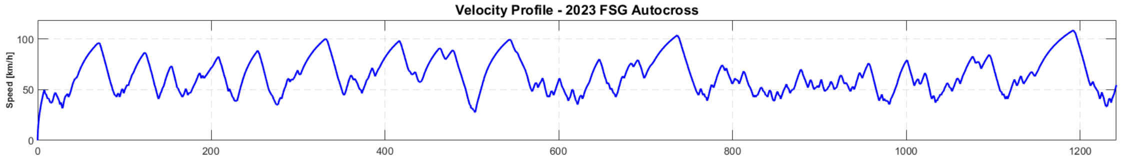


Figure 4.10: Calculated velocity profile from the 2023 FSG autocross track.

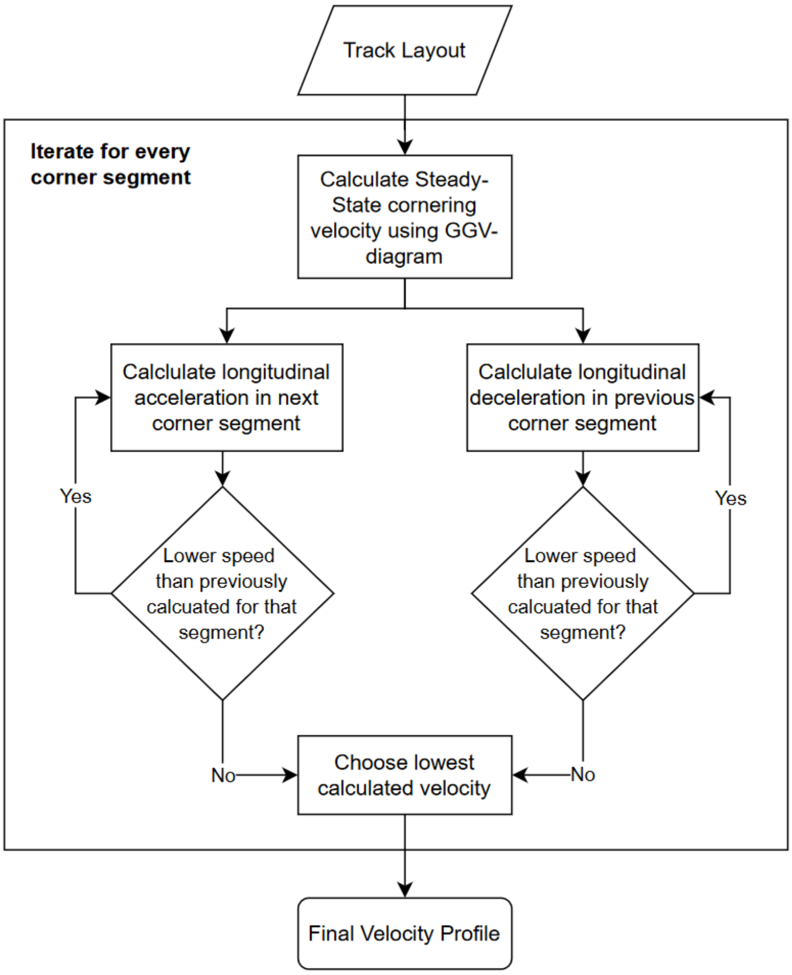


Figure 4.11: Flowchart of the QSS solver

4.5.1 Post-Processing

With the velocity profile calculated, post-processing is necessary to get more output data. A crucial part of this process involves comparing the simulation data with telemetry data for a more accurate correlation. In the context of the telemetry data, CFS uses braking pressure which was implemented in the simulator with the equations below.

$$T_{\text{brake}} = \frac{a_x \cdot m_{\text{total}} \cdot r_l}{1 + 2 \cdot (1 + \beta_{\text{regen}}) \cdot \frac{T_{\text{regen max}}}{T_{\text{braking,max regen pressure}}} \cdot i_{\text{gear ratio}}} \tag{4.39}$$

$$P_{\text{brakes}} = \frac{T_{\text{brake}}}{2 \cdot A_{c,F} \cdot \mu_{\text{pad}} \cdot r_{\text{eff}} + 2 \cdot (1 - \beta_{\text{brake}}) \cdot A_{c,R} \cdot \mu_{\text{pad}} \cdot r_{\text{eff}}} \cdot 10^{-5} \tag{4.40}$$

The regenerative braking has a power limit described in Section 4.2.3. Equation 4.41 calculates regenerative braking, which is then compared to the power limit, taking the smallest of the two values.

$$T_{\text{regen}} = \frac{T_{\text{brake}} \cdot 2 \cdot (1 + \beta_{\text{regen}}) \cdot T_{\text{max regen}}}{T_{\text{braking,max regen pressure}}} \tag{4.41}$$

Understanding the energy consumption in the simulation is vital for making informed design decisions. Efficiency calculations for electric motors and inverters can be derived from data obtained during CFS dyno-testing. These losses calculated in Equation 4.42-4.43 are based on torque and constants a, b, c, d.

$$E_{\text{motor losses}} = a \cdot x^2 + b \cdot T \tag{4.42}$$

$$E_{\text{inverter losses}} = c \cdot T + d \tag{4.43}$$

To calculate the scored points for each event, the FSG rule books points calculations are used with the simulated lap time [4]. Maximum points according to Table 1.1 together with the equations found in Appendix D. The equations necessitate a high-scoring team for each event. Therefore, these simulations employ a theoretically quick car, modeled on the parameters of the top teams from the 2023 FSG competition. It is possible to adjust this to reflect the actual FSG results. However, since the simulated times are slightly quicker than the real results, using an optimal simulated car as the baseline for points was deemed more appropriate.

4.5.2 Calibration of Data

As an LTS with good correlation to the real world is desired, calibration of parameters is required. Initially, parameters are set to what is known from previous data collection, calculations, simulations, and given specifications. To get the LTS within the desired accuracy, these parameters need to be adjusted. This done by comparing simulated data to real data and adjusting the parameters until an acceptable value is found. The most accurate, if available, is to compare telemetry data to simulated data. Multiple telemetry data channels can be looked at and compared to analyse what parameters should be adjusted. Velocity, lateral acceleration and longitudinal accelerations can be compared to find if parameters such as tyre coefficients, drivetrain parameters and resistances should be modified in order to have the LTS correlate better.

If no telemetry data is available for a specific event, it is possible to correlate final lap time. This is possible in case of the Skidpad event, as only lateral acceleration affects lap time in the LTS, and to some extent the Acceleration event as only longitudinal acceleration affects lap time. For more complex events such as autocross and endurance, lap time alone is not a sufficient reference, as multiple parameters affect it.

5

Simulation Results

This chapter and following sections will illustrate the results of the different simulations. It will explain what information is available and what conclusions that can be made following different simulations. Chapter 5 will also present areas of use where the LTS could bring value to CFS.

5.1 Plots

The majority of the results are displayed as plots in MATLAB. Some of the more essential plots, in addition to those presented in this section, are provided in Appendix A. This section includes a brief description of what each plot represents.

During a standard simulation execution, three default plots are generated: one for the GGV-diagram, one for the track map, and the last one containing multiple graphs representing velocities, accelerations, throttle position, and brake pressure. In the case of a multi-event simulation, a default plot showing only points gained during the different events is generated. If more detailed data on all the events is desired, this can be specified in the GUI.

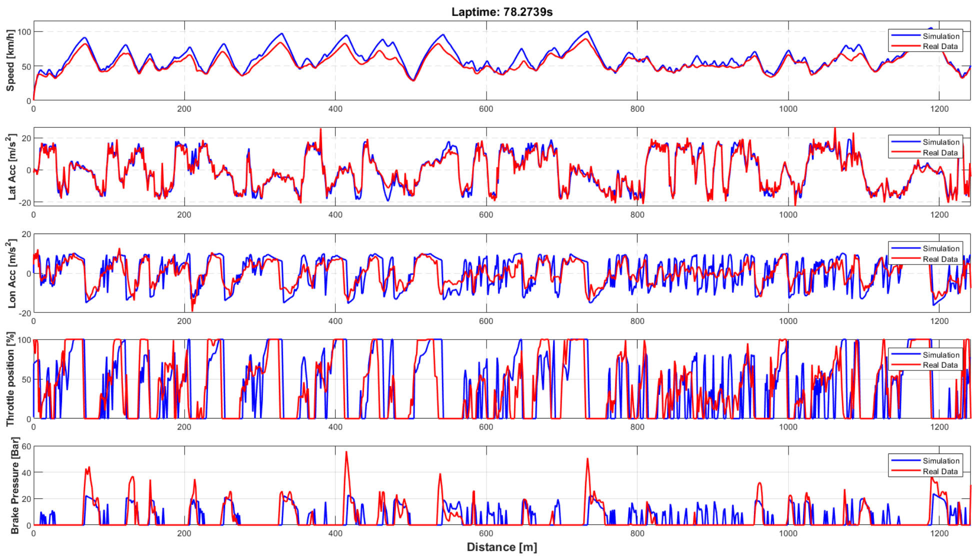


Figure 5.1: This plot shows, other than lap time, different graphs describing how velocity, accelerations, throttle position and brake pressure varies during an autocross lap.

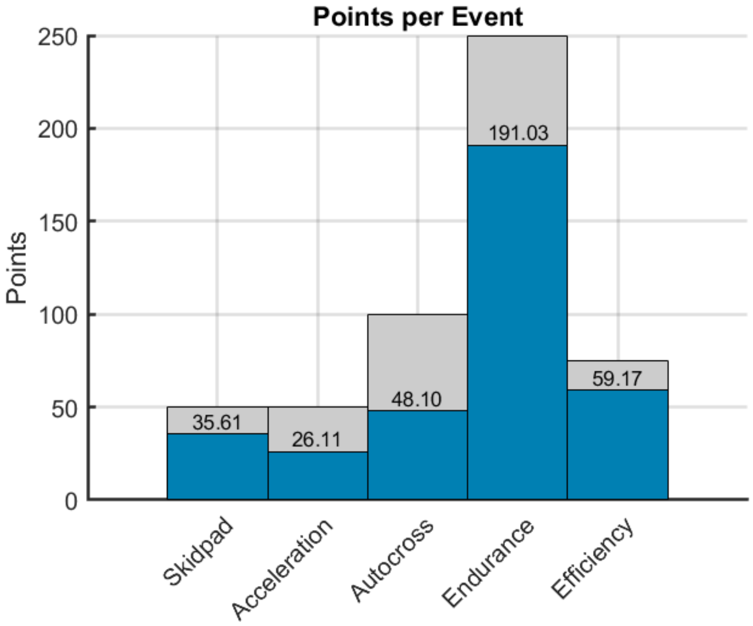


Figure 5.2: Points scored in every event. The grey areas illustrates maximum achievable points per event according to the FSG 2024 rules [4].

By using the Data Visualization GUI, the user can access previous simulations with its data and related plots. One example is two graphs showing how drag and downforce varies during the event.

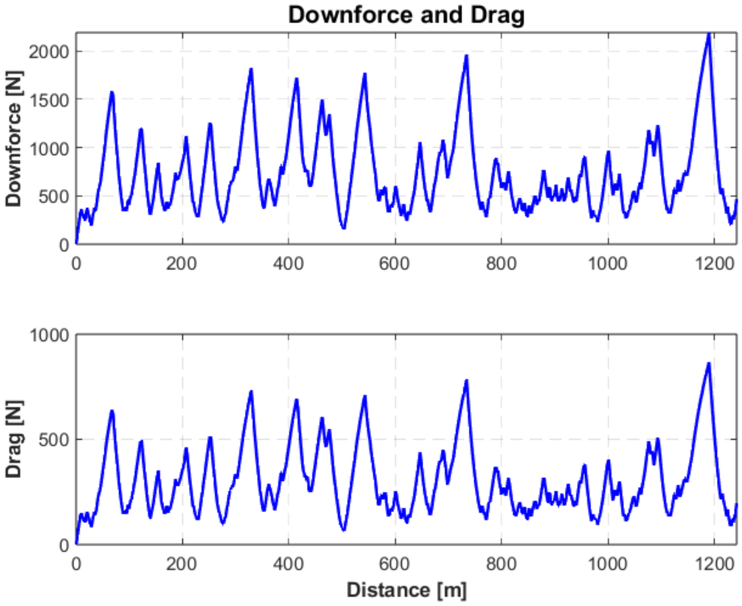


Figure 5.3: This plot illustrates how the drag force and down force varies during an autocross event with a point-mass simulation

Finally, for single parameter sweep, grid sweep and sensitivity analysis similar plots to Figures 3.3, 3.4 and 3.5 will appear when simulating a single event. For multi event the program makes one plot for every chosen event.

5.2 Simulation Output

The simulator saves all the data to the MATLAB workspace and, if desired, as a MATLAB file for later use. This data can be used to find values that are not visualised in the predefined plots, or to create new plots or used in other ways to understand the results of the simulation. It is important that this data is easily accessible to a new user and therefore everything is labeled in a structured way.

Following every simulation there is an option to automatically create a report in PDF format with all necessary information from the simulation. The reports can be used to compare different simulations to each other or work as presentation material on meetings. Reports are also an important tool to find saved simulations for later use or comparison. An example of a report can be found in Appendix C.

For easier correlation comparison, real telemetry data can be layered on top of the simulated data in the plots if that is available for the specific track. This is important to see differences and similarities between the simulation and real world. In Figure 5.1, data is compared for velocity, lateral and longitudinal acceleration, throttle position, and brake pressure. Information gathered from this can be used to calibrate simulation parameters to real world data, but also to see where improvements can be made for the real car.

5.3 Interesting Areas of Use

Many fields of study within CFS will benefit from this LTS. This section will present examples on how the result of this project can be used.

From the usual CFD simulations that the aerodynamic department uses to simulate changes of the aero package, only values such as lift and drag coefficients, as well as total downforce can be found. In order to quantify this to on-track performance and subsequent points, an LTS is a good tool. An LTS can be used to quantify how much downforce is required for a specific lap time or points gain. It can also be used to compare the loss of lap time due to weight and drag of aerodynamic parts to gain from downforce. For example, by studying Figure 5.3 this kind of data can be obtained.

For the car to be successful in the Endurance event, the team needs to find the balance between finding good lap times and reduce energy consumption as teams are scored for both elapsed time and efficiency. Equations for points in endurance and efficiency can be found in Appendix D. One way to optimise points is to find the best combination between the maximum torque from the motors and the maximum Revolutions Per Minute (RPM). This is done by pursuing a grid sweep in the GUI with the parameters. The result can be seen in Figure 5.4, a plot that illustrate how much points different combinations of maximum torque and maximum RPM would score. According to these simulations the maximum torque should be set to 15.6 Nm and maximum motor speed to 13300 RPM in order to score the most points in the Endurance event.

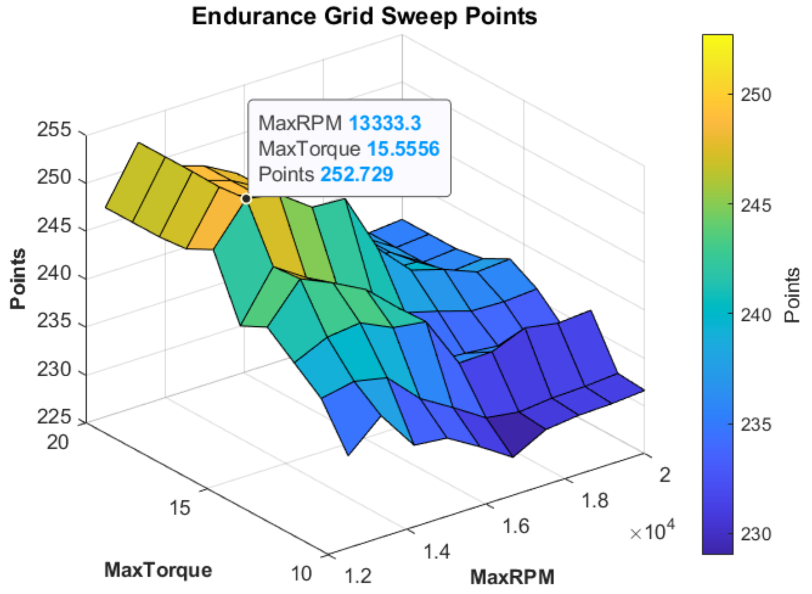


Figure 5.4: Grid sweep simulation of the 2023 FSG Endurance. Max RPM and max torque swept to find optimal strategy.

6

Analysis

Chapter 6 provides an analysis on how well the result of this project satisfies CFS. This chapter will describe whether all the requirements have been fulfilled and evaluate the predictive capability of the LTS.

6.1 Requirements

The performance requirements in the list of requirements, Figure 2.5, has for most parts been fulfilled. The simulation time varies depending on what kind of simulation is executed. The simulation time for a standard simulation around the 2023 FSG autocross track with the point-mass model is 0.25 seconds on a laptop with an Intel i7 11800H processor. A simulation with the bicycle model with the same track takes around 5 seconds. Doing a grid sweep simulation with 10^2 simulations therefore takes around 25 seconds with the point-mass and 10 minutes for the bicycle. However, any kind of sweep executes multiple simulations and the criteria in the list of requirements regards only one simulation. Both the point-mass model and the bicycle model are kept in the program as it is sometimes valuable to keep simulation times low at the expense of accuracy. During early development, it might be more important to run many different simulations and try big changes instead of fewer more accurate simulations. Giving the user the option to choose vehicle model will satisfy CFS more.

At the 2023 FSG Autocross event CFS performed their best lap time of 83.1 seconds. The lap time on the same track in the LTS with the point-mass model is 78.3 seconds and for the bicycle model with load transfer effects is 79.3 seconds. This means that only the required lap time accuracy of ±5 seconds is fulfilled, while the desired accuracy of ±3 seconds is not. That could be a consequence of several reasons. The simplest reason is that the simulator is a perfect driver that is always in steady state, while the real driver is not and will therefore not reach the optimal time. Additionally, the LTS also does not account for how the track changes in z-direction. Elevation, banking or bumps in the race track affects the performance and lap time, but this is not accounted for in the LTS. Moreover, the creation of the track itself may introduce slight inaccuracies due to sensor drift over time, further affecting the lap time calculations. Lastly, the car in the simulator is merely a simplified mathematical model of the actual vehicle, which inherently limits the correlation to real-world performance.

In the subsequent sections of the requirement list, all parameters and functions specified in the LTS have been integrated, with the exception of criterion 2.4, 2.5, 2.6, 2.9, and 2.11. The decision to leave these criteria unaddressed was primarily driven by the limited time frame of the project. Fulfilling these criteria would have required creating numerous new GGV diagrams, which in turn would have extended simulation time. Therefore, the decision was made not to implement them.

6.2 Accuracy of lap time prediction

Comparing the real telemetry data to the simulated data in Figure 5.1 shows that the data correlates well. Lateral acceleration correlates very well, with only small deviations. Longitudinal data also correlates well, but with a few bigger differences. Maximum longitudinal acceleration is slightly higher for the simulated data and there is also bigger changes of acceleration in the simulated data. The higher maximum is likely due to the fact that it is the drivers first lap around the track, so he is not at the maximum limit of the car, especially during braking. The simulated car is also perfect and always drives with perfect grip. This is not true to reality and is one other reason why the simulated car is so much quicker in the longitudinal accelerations. As for the bigger fluctuations and spikes in the longitudinal data, that is due to the simulation being in steady state for all time steps, neglecting any transient behavior that would slow down these big fluctuations in acceleration. In the speed trace, we can see the results of these differences. The car follows the same trend very well, matching the minimum speed almost perfectly. During acceleration and braking is where we see the big differences, and maximum velocity being slightly higher for the simulation.

All of this is expected for a simulator, as it only creates a model of the real world. The primary purpose of an LTS is to evaluate changes in car parameters and the relative performance gain or loss, rather than comparing them with an actual vehicle and therefore minor differences are not a concern. These discrepancies are typical and do not detract from the utility of the simulation for its intended analytical purposes.

7

Discussion

The following chapter will discuss the results and future use of the LTS. It will go into interpretations of the outcomes, highlighting the factors contributing to the successes achieved. Additionally, it will explore the reasons behind any discrepancies between expected and actual results. Insights into potential applications and further development of the LTS will also be presented, providing a foundation for ongoing research and improvement.

7.1 Simulation Results

During this project, different members of CFS have given their input and feedback on the simulator and in the end, the project group have experienced that people within CFS are eager to use the program to improve next years car. However, it is not until something is used for a longer period that the real flaws begins to appear. The members of CFS might find the LTS interesting and useful right now, but perhaps after working with it during next year, errors or areas where it does not fully satisfy the user will be detected. However, since the code is written in an easy to understand way, split in to multiple parts, and well documented, it is easy to add or alter the code to fit specific needs in the future.

A potential factor contributing to skepticism about the simulator might be concerns regarding the accuracy of lap times. There is a risk that team members may hesitate to depend on the simulation results if there is uncertainty about their validity. To enhance confidence in the LTS, it is essential for members of CFS to conduct further comparisons with actual lap times from various tracks. Such comparisons are expected to increase as the team participates in more competitions and conducts additional testing. Additionally, the LTS can be further validated by incorporating new tracks based on historical data and creating new car data files for previous models. If the simulator consistently demonstrates accurate lap times across different tracks and cars, it is likely to be more widely accepted by all members. Since the LTS has been positively received during its development phase, members of CFS appear interested and eager to utilize the simulator.

7.2 Future Development and Use

This LTS will hopefully aid in CFS' development of future cars and the code structure is a good foundation for future improvements, new vehicle models and additional functions. The first improvement is likely to be the implementation of pitch. As stated before, the implementation of pitch angle did not work as intended and was therefore left out in the final version of the LTS. Since it only was a minor error, it can be corrected beyond this project within the near future.

During the year, CFS tests the car on track and sometimes in a wind tunnel. Important data is collected such as validation for different aerodynamics coefficients. This LTS is dependant on that data which means that more testing could lead to higher lap time accuracy. Further testing sessions on how e.g C_L and C_D varies in pitch and with increasing velocity will provide more accurate data and improve the simulator.

As mentioned earlier, this LTS is meant to be kept working on by the team. Unfortunately, similar projects tends to be put on hold because of several reasons. The main problem seen in past years of CFS has been that the program code and documentation has not been structured enough for someone else to take over. However, apart from the already modular structured program code, that consists of multiple independent functions, a user guide is developed in order to increase the teams general knowledge about how the simulator works. The more people that fully understands how the simulator is built and operates, the more people will have the ability to improve the LTS in the future and increase the likelihood of maintaining the simulator in CFS for a longer period.

Bibliography

[1] F. S. Germany. “Formula student concept.” (2024), [Online]. Available: `https://www.formulastudent.de/about/concept/` (visited on 2024-04-18).

[2] “Fsg: All known universities and teams.” (2024), [Online]. Available: `https://www.formulastudent.de/teams/all-universities/` (visited on 2024-02-01).

[3] C. F. Student. “Chalmers formula student garage.” (2024), [Online]. Available: `https://www.chalmersformulastudent.se/garage` (visited on 2024-04-18).

[4] F. S. Germany. “Formula student rules 2024.” (2024), [Online]. Available: `https://www.formulastudent.de/fileadmin/user_upload/all/2024/rules/FS-Rules_2024_v1.1.pdf` (visited on 2024-04-18).

[5] M. Massaro and D. J. N. Limebeer, “Minimum-lap-time optimisation and simulation,” *Vehicle System Dynamics*, vol. 59, no. 7, pp. 1069–1113, 2021. DOI: `10.1080/00423114.2021.1910718`. eprint: `https://doi.org/10.1080/00423114.2021.1910718`. [Online]. Available: `https://doi.org/10.1080/00423114.2021.1910718`.

[6] B. Siegler, A. Deakin, and D. Crolla, “Lap time simulation: Comparison of steady state, quasi- static and transient racing car cornering strategies,” *SAE Transactions*, vol. 109, pp. 2575–2581, 2000, ISSN: 0096736X, 25771531. [Online]. Available: `http://www.jstor.org/stable/44687095` (visited on 2024-02-03).

[7] V. Cortes. “Laptime simulator: Mass-point quasi-steady state.” (2021), [Online]. Available: `https://www.linkedin.com/pulse/laptime-simulator-mass-point-quasi-steady-state-victor-cort%C3%A9s-abad/` (visited on 2024-02-02).

[8] D. Nowlan, *The Dynamics of the Race Car*, en. Chassisim, 2010, ISBN: 978-0-9870593-0-7.

[9] C. Rouelle, “Applied vehicle dynamics seminar,” Optimum G, 2018.

[10] “Optimumlap.” (), [Online]. Available: `https://optimumg.com/product/optimumlap/` (visited on 2024-04-27).

[11] bseater. “Data from round 6 of fsae ttc tire testing at calspan.” (2015), [Online]. Available: `https://www.fsaettc.org/viewtopic.php?t=151` (visited on 2024-04-18).

[12] H. B. Pacejka, *Tire and Vehicle Dynamics*. Elsevier Ltd, 2011.

[13] W. F. Milliken, *Race car vehicle dynamics*, en. SAE International, 1995, ISBN: 1-56091-526-9.

[14] "Casadi." (), [Online]. Available: `https://web.casadi.org/` (visited on 2024-05-01).

[15] R. P. Costa and R. Bortolussi, "Lap time simulation of formula sae vehicle with quasi-steady state model," *SAE Technical Paper Series*, vol. Not available, Not available, Oct. 2016, ISSN: 0148-7191. DOI: `10.4271/2016-36-0164`. [Online]. Available: `https://dx.doi.org/10.4271/2016-36-0164`.

A

Plots

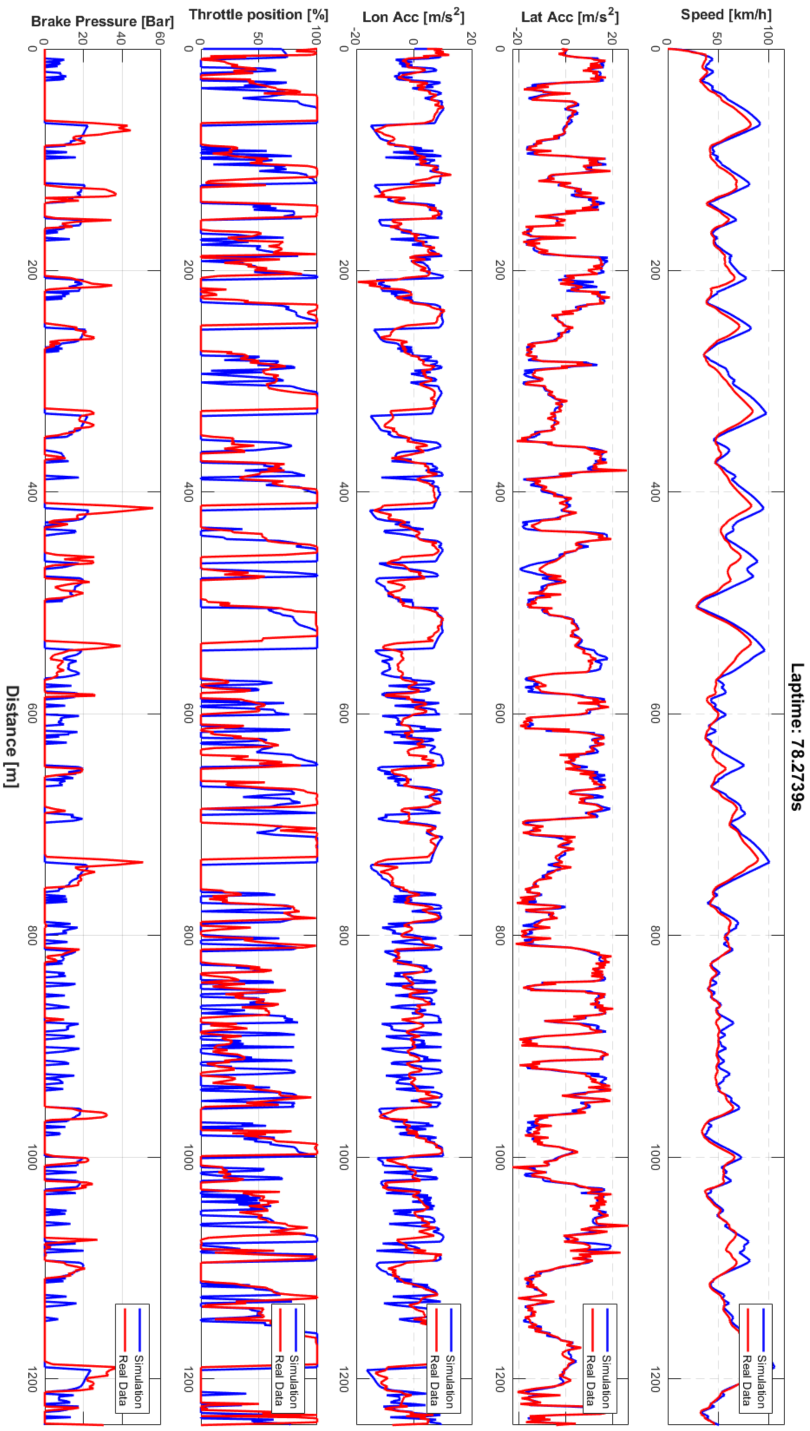


Figure A.1: Standard telemetry simulation graphs with real telemetry overlayed.

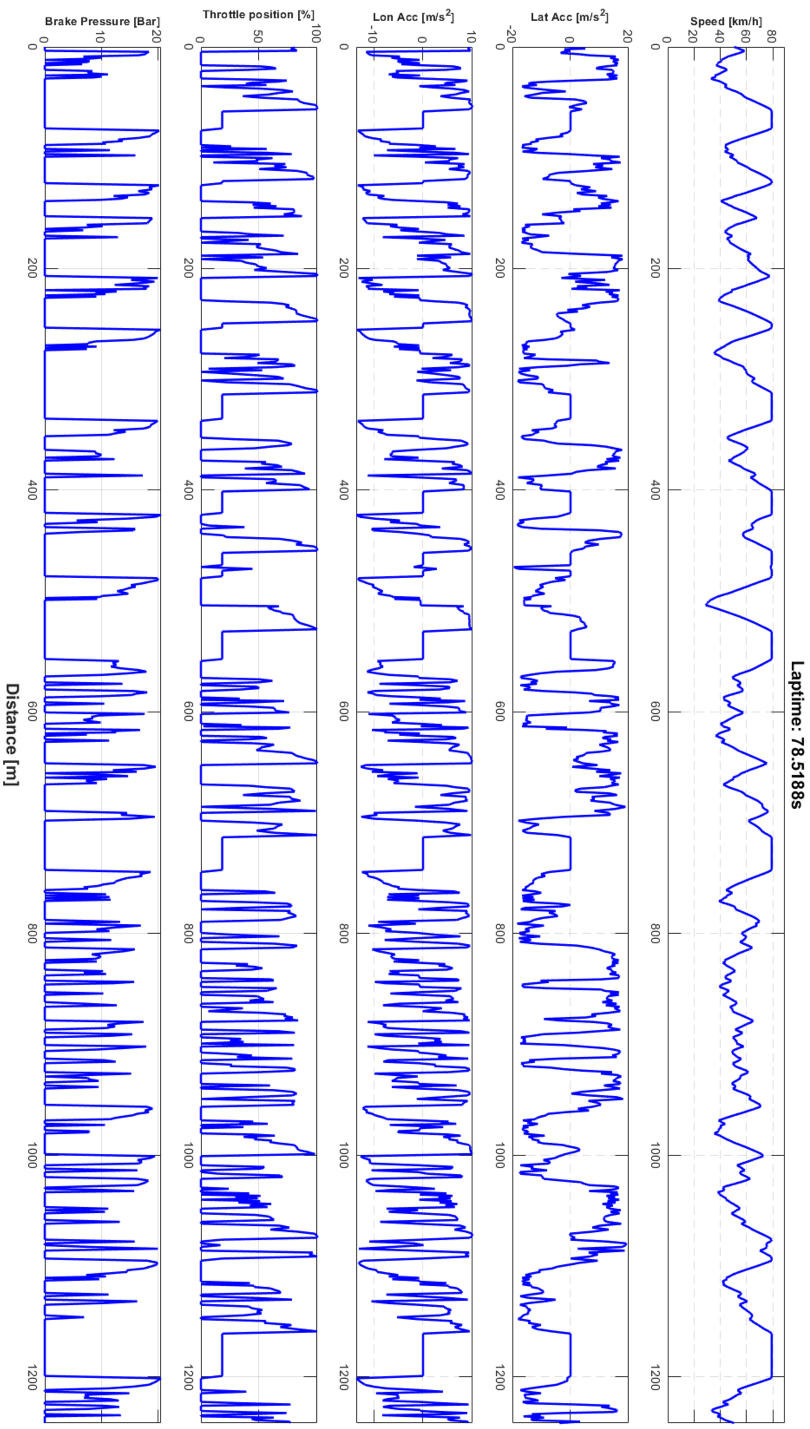


Figure A.2: The figure shows a plot with graphs from a standard Endurance simulation.

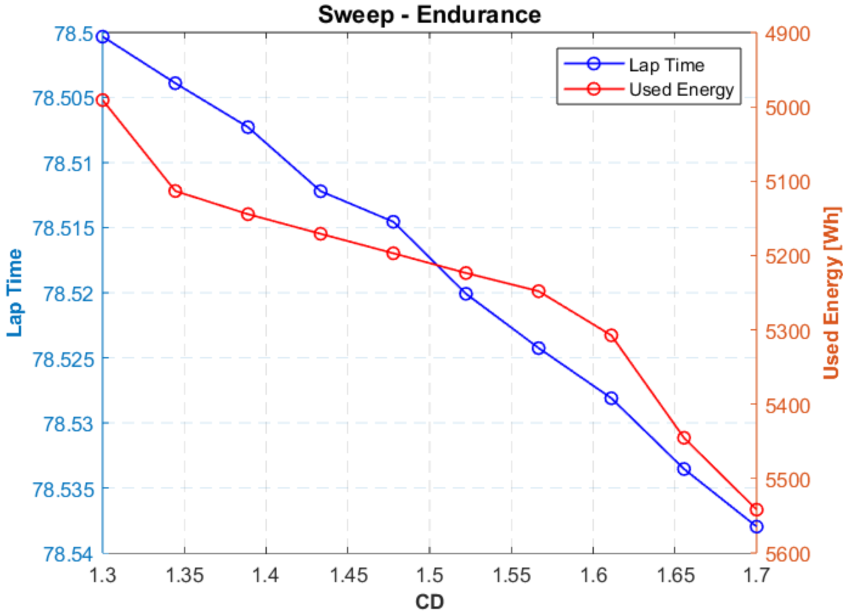


Figure A.3: This is a plot of a CD parameter sweep for the endurance event. It shows how much energy is used as well as the lap time

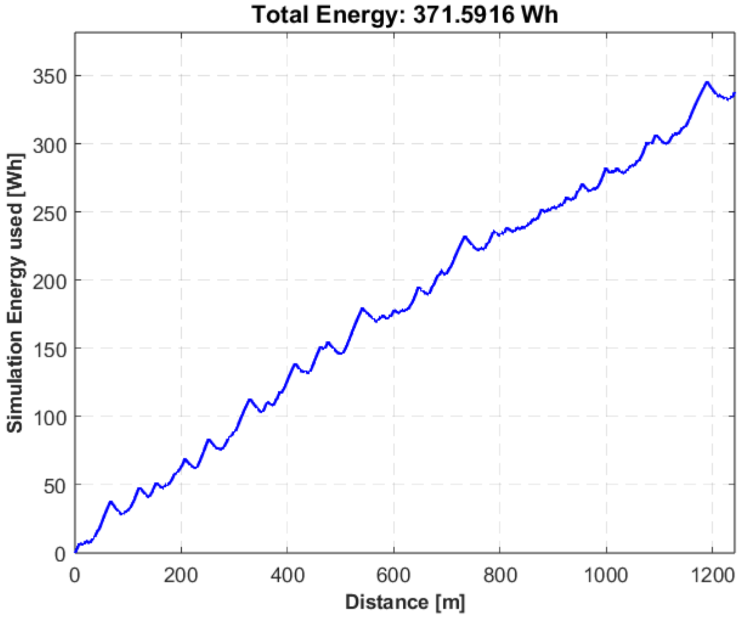


Figure A.4: This is a plot recreated from a simulation using the data visualization GUI. It illustrates how much energy that is used during an autocross event. As the car has regenerative braking, it recuperates energy while braking.

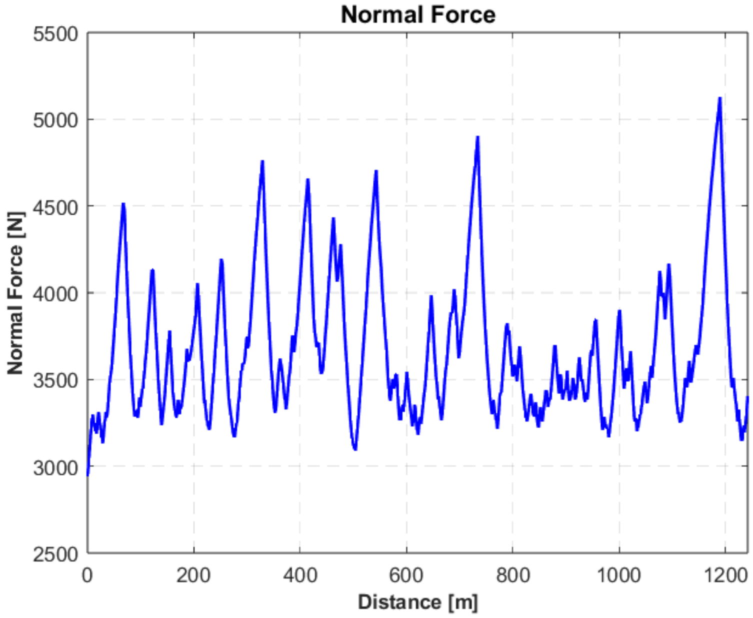
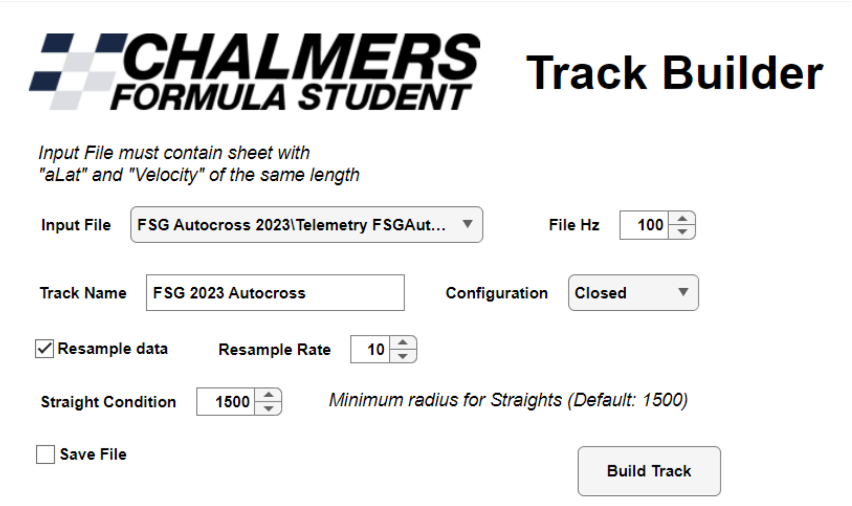


Figure A.5: A plot of how normal force on tyres varies during an Autocross event.

B

User Interfaces



Input File must contain sheet with "aLat" and "Velocity" of the same length

Input File: FSG Autocross 2023\Telemetry FSGAut...

File Hz: 100

Track Name: FSG 2023 Autocross

Configuration: Closed

☑ Resample data

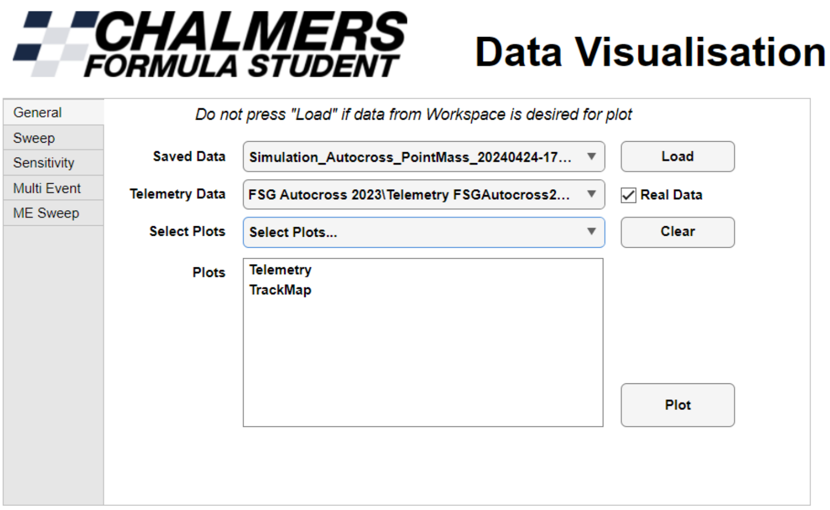
Resample Rate: 10

Straight Condition: 1500

Minimum radius for Straights (Default: 1500)

☐ Save File

Figure B.1: Track Builder GUI



Do not press "Load" if data from Workspace is desired for plot

Saved Data: Simulation_Autocross_PointMass_20240424-17...

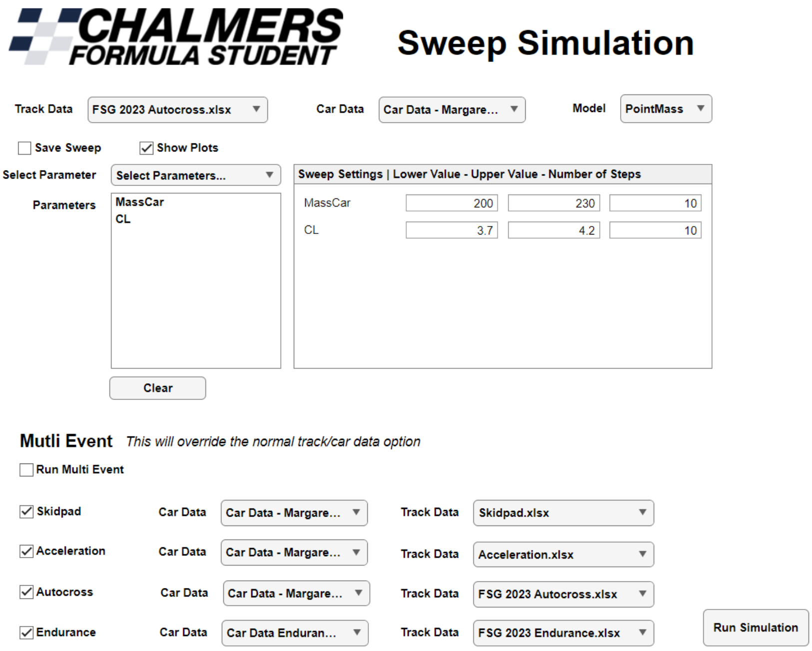
Telemetry Data: FSG Autocross 2023\Telemetry FSGAutocross2...

☑ Real Data

Select Plots: Select Plots...

Plots: Telemetry
TrackMap

Figure B.2: Data Visualisation GUI



Sweep Simulation

Track Data FSG 2023 Autocross.xlsx ▼
Car Data Car Data - Margare... ▼
Model PointMass ▼

☐ Save Sweep
☑ Show Plots

Select Parameter Select Parameters... ▼

Parameters
MassCar
CL

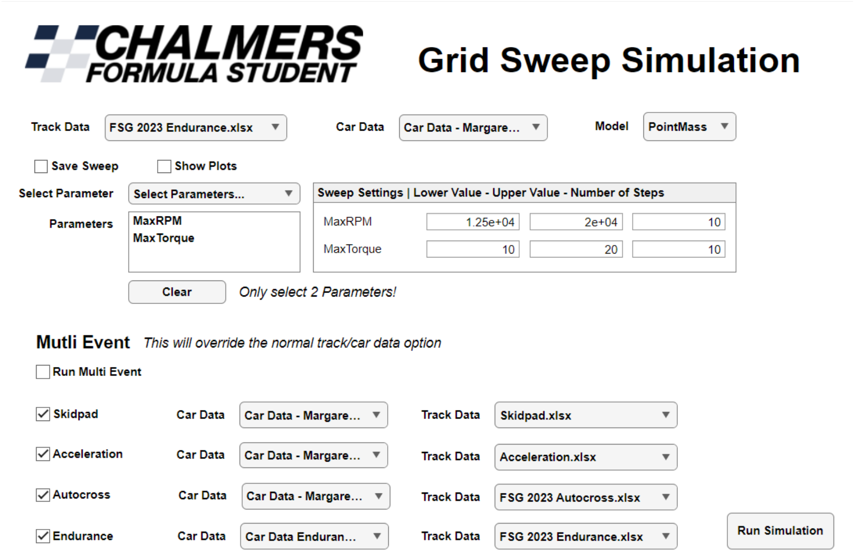
Sweep Settings | Lower Value - Upper Value - Number of Steps			
MassCar	200	230	10
CL	3.7	4.2	10

Mutli Event *This will override the normal track/car data option*

☐ Run Multi Event

☑ Skidpad	Car Data	Car Data - Margare... ▼	Track Data	Skidpad.xlsx ▼
☑ Acceleration	Car Data	Car Data - Margare... ▼	Track Data	Acceleration.xlsx ▼
☑ Autocross	Car Data	Car Data - Margare... ▼	Track Data	FSG 2023 Autocross.xlsx ▼
☑ Endurance	Car Data	Car Data Enduran... ▼	Track Data	FSG 2023 Endurance.xlsx ▼

Figure B.3: Sweep Analysis GUI



Grid Sweep Simulation

Track Data FSG 2023 Endurance.xlsx ▼
Car Data Car Data - Margare... ▼
Model PointMass ▼

☐ Save Sweep
☐ Show Plots

Select Parameter Select Parameters... ▼

Parameters
MaxRPM
MaxTorque

Sweep Settings | Lower Value - Upper Value - Number of Steps			
MaxRPM	1.25e+04	2e+04	10
MaxTorque	10	20	10

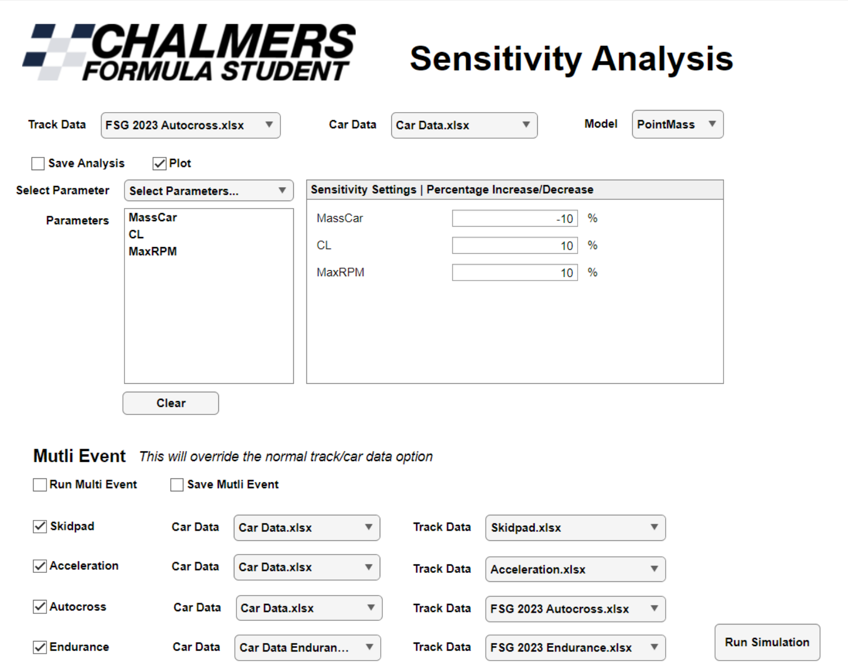
Only select 2 Parameters!

Mutli Event *This will override the normal track/car data option*

☐ Run Multi Event

☑ Skidpad	Car Data	Car Data - Margare... ▼	Track Data	Skidpad.xlsx ▼
☑ Acceleration	Car Data	Car Data - Margare... ▼	Track Data	Acceleration.xlsx ▼
☑ Autocross	Car Data	Car Data - Margare... ▼	Track Data	FSG 2023 Autocross.xlsx ▼
☑ Endurance	Car Data	Car Data Enduran... ▼	Track Data	FSG 2023 Endurance.xlsx ▼

Figure B.4: Grid Sweep GUI



Track Data: FSG 2023 Autocross.xlsx
Car Data: Car Data.xlsx
Model: PointMass

☐ Save Analysis ☑ Plot

Select Parameter: Select Parameters...

Parameters:
MassCar
CL
MaxRPM

Sensitivity Settings | Percentage Increase/Decrease

MassCar -10 %
CL 10 %
MaxRPM 10 %

Mutli Event *This will override the normal track/car data option*

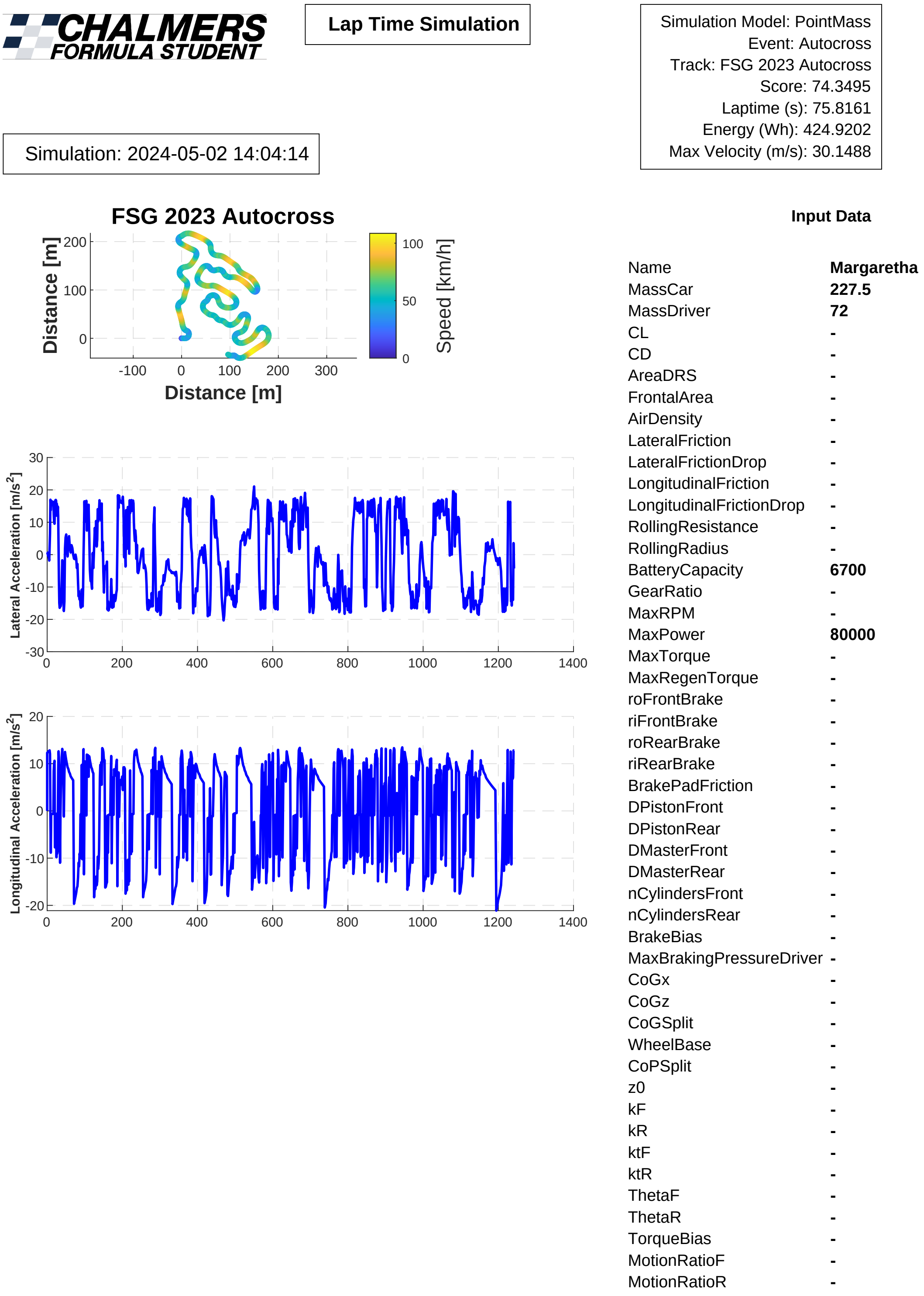
☐ Run Multi Event ☐ Save Mutli Event

☑ Skidpad — Car Data: Car Data.xlsx — Track Data: Skidpad.xlsx
☑ Acceleration — Car Data: Car Data.xlsx — Track Data: Acceleration.xlsx
☑ Autocross — Car Data: Car Data.xlsx — Track Data: FSG 2023 Autocross.xlsx
☑ Endurance — Car Data: Car Data Enduran... — Track Data: FSG 2023 Endurance.xlsx

Figure B.5: Sensitivity Analysis GUI

C

PDF report document



CHALMERS FORMULA STUDENT

Lap Time Simulation

Simulation Model: PointMass
Event: Autocross
Track: FSG 2023 Autocross
Score: 74.3495
Laptime (s): 75.8161
Energy (Wh): 424.9202
Max Velocity (m/s): 30.1488

Simulation: 2024-05-02 14:04:14

Input Data

Name	**Margaretha**
MassCar	**227.5**
MassDriver	**72**
CL	**-**
CD	**-**
AreaDRS	**-**
FrontalArea	**-**
AirDensity	**-**
LateralFriction	**-**
LateralFrictionDrop	**-**
LongitudinalFriction	**-**
LongitudinalFrictionDrop	**-**
RollingResistance	**-**
RollingRadius	**-**
BatteryCapacity	**6700**
GearRatio	**-**
MaxRPM	**-**
MaxPower	**80000**
MaxTorque	**-**
MaxRegenTorque	**-**
roFrontBrake	**-**
riFrontBrake	**-**
roRearBrake	**-**
riRearBrake	**-**
BrakePadFriction	**-**
DPistonFront	**-**
DPistonRear	**-**
DMasterFront	**-**
DMasterRear	**-**
nCylindersFront	**-**
nCylindersRear	**-**
BrakeBias	**-**
MaxBrakingPressureDriver	**-**
CoGx	**-**
CoGz	**-**
CoGSplit	**-**
WheelBase	**-**
CoPSplit	**-**
z0	**-**
kF	**-**
kR	**-**
ktF	**-**
ktR	**-**
ThetaF	**-**
ThetaR	**-**
TorqueBias	**-**
MotionRatioF	**-**
MotionRatioR	**-**

D

Points Equations

Equations for points scored per event according to the 2024 FSG rules [4].

D.1 Manual Skidpad Scoring

The run time is the average time of the timed left and the timed right circle plus penalties which are added after the averaging.

5 % of the maximum points for the event according to Table 1.1 are awarded to every team that finishes at least one run without DNF or DQ.

If a team's best manual mode time including penalties is below T_{max}, additional points based on the following formula are given:

$$\text{M\_SKIDPAD\_SCORE} = 0.95 P_{max} \left(\frac{(\frac{T_{max}}{T_{team}})^2 - 1}{0.5625} \right) \tag{D.1}$$

P_{max} is the maximum points for the event according to Table 1.1.
T_{team} is the team's best manual mode time including penalties.
T_{max} is 1.25 times the time of the fastest manual mode vehicle including penalties.

D.2 Manual Acceleration Scoring

5 % of the maximum points for the event according to Table 1.1 are awarded to every team that finishes at least one manual mode run without a DNF or DQ.

If a team's best manual mode time including penalties is below Tmax, additional points based on the following formula are given:

$$\text{M\_ACCLERATION\_SCORE} = 0.95 P_{max} \left(\frac{\frac{T_{max}}{T_{team}} - 1}{0.5} \right) \tag{D.2}$$

P_{max} is the maximum points for the event according to table 1.1.
T_{team} is the team's best manual mode time including penalties.
T_{max} is 1.5 times the time of the fastest manual mode vehicle including penalties.

D.3 Autocross Scoring

5 % of the maximum points for the event according to Table 1.1 are awarded to every team that finishes at least one run without DNF or DQ.

If a team's corrected elapsed time is below Tmax, points based on the following formula are given:

$$\mathbf{AUTOCROSS\_SCORE} = 0.95 P_{max} \left(\frac{\frac{T_{max}}{T_{team}} - 1}{0.25} \right) \tag{D.3}$$

P_{max} is the maximum points for the event according to Table 1.1.
T_{team} is the team's best time including penalties.
T_{max} is 1.25 times the time of the fastest vehicle including penalties.

D.4 Endurance Scoring

Each lap of the endurance event is individually timed. The corrected elapsed time is determined by subtracting the extra-long lap for the driver change from the total time and adding any penalty times.

10 % of the maximum points for the event according to Table 1.1 are awarded to every team that finishes endurance without DNF or DQ.

If a team's corrected elapsed time is below Tmax, additional points based on the following formula are given:

$$\mathrm{ENDURANCE\_SCORE} = 0.9 P_{max} \left(\frac{\frac{T_{max}}{T_{team}} - 1}{0.333} \right) \tag{D.4}$$

P_{max} is the maximum points for the event according to table 1.1.
T_{team} is the team's corrected elapsed time.
T_{max} is 1.333 times of the corrected elapsed time of the fastest vehicle.

D.5 Efficiency Scoring EV

The endurance energy is calculated as the time integrated value of the measured voltage multiplied by the measured current logged by the data logger, see EV 4.6. Regenerated energy is multiplied by 0.9 and subtracted from the used energy.

Efficiency points based on the following formula are given:

$$\mathrm{EFFICIENCY\_SCORE} = P_{max} = \left(\frac{EF_{max} - EF_{team}}{EF_{max} - EF_{min}} \right) \tag{D.5}$$

with
P_{max} is the maximum points for the event according to table 1.1.

EF_{team} the team's efficiency factor.
EF_{min} the lowest efficiency factor of all teams which were considered for efficiency.
EF_{max} is defined as $1.5 \cdot EF_{min}$.

The efficiency factor is calculated based on the following formula:

$$EF = T^2 \cdot E \tag{D.6}$$

with
T uncorrected elapsed driving time
E used energy

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

CHALMERS