

Politecnico di Torino
Master's degree course in Automotive engineering



**Politecnico
di Torino**

Lap time simulator for a single-seater electric car

Master's Thesis
Academic year 2020/2021

Professor:
Andrea Tonoli

Candidate:
Alberto Reinerio

Ai miei nonni

Abstract

This thesis aims to develop a quasi steady-state lap time simulator for a Formula Student electric prototype developed by Squadra Corse PoliTO.

The idea is to create a tool that helps the team to evaluate the influence of the main parameters of the vehicle on the performance and define the race strategy in terms of a trade-off between lap time and energy consumption.

In the following chapters, a point-mass quasi steady-state simulator is developed. The vehicle is modelled as a point-mass with longitudinal and lateral friction coefficients, longitudinal and vertical aerodynamics, and with an electric power-train composed of motors, inverters, and a battery pack.

This model is adopted in the early stages of the design to find the balance between the primary input values of the car, such as mass, aerodynamic coefficients, and battery pack capacity. The simulation process is analyzed in detail phase by phase.

The last part of this thesis is dedicated to the development of a simulation code used to obtain yaw moment diagrams, a useful tool to analyze the vehicle dynamics behavior and to create GGV diagrams that in future developments could be integrated into the simulator code.

Acknowledgments

Giunto alla conclusione del mio percorso di laurea, desidero ringraziare le persone che mi hanno aiutato maggiormente nel corso di questi anni. Inizialmente vorrei ringraziare il Professor Tonoli, tutore del team Squadra Corse e relatore per questa tesi. Grazie per avermi permesso di far parte di questo team per le stagioni 2018/19 e 2019/20. Desidero inoltre ringraziare i colleghi di Squadra Corse, con i quali abbiamo lavorato duramente nonostante il difficile periodo di pandemia. In particolare un ringraziamento a Mario, con cui ho condiviso gran parte del mio percorso universitario e con cui ho collaborato nella divisione di Vehicle Dynamics. Grazie inoltre a Jack, con cui ho collaborato nello sviluppo della parte dedicata agli Yaw Moment Diagrams.

Grazie alla mia famiglia che mi ha sostenuto in questi anni. Un particolare ringraziamento a mia sorella Eleonora, che con la sua simpatia ha reso i momenti di studio più leggeri.

Infine grazie a Francesca che, al mio fianco nell'ultima parte di questo percorso al Politecnico, mi ha donato la forza e la tenacia necessaria per concludere al meglio questo capitolo della mia vita.

Contents

1	Introduction	13
1.1	Formula SAE/Formula Student	13
1.1.1	History	13
1.1.2	The competition	13
1.2	Squadra Corse PoliTO	17
2	Lap time simulation basics	19
2.1	Classification	19
2.1.1	Steady-state simulations	19
2.1.2	Quasi steady-state simulations	21
2.1.3	Transient simulations	24
3	The vehicle	27
3.0.1	Powetrain	30
3.0.2	Battery Pack	34
3.0.3	Aerodynamics	37
4	Development of a point-mass quasi steady-state simulator	39
4.1	Target setting	39
4.2	Inputs required	39
4.2.1	Track	39
4.2.2	Vehicle model	40
4.3	Main structure	42
4.3.1	Definition of the GGV diagram of the vehicle	43
4.3.2	Evaluation of the maximum cornering speed	46
4.3.3	Evaluation of the combined cornering-braking speed profile .	49
4.3.4	Evaluation of the combined cornering-braking-acceleration speed profile	52
4.3.5	Evaluation of auxiliary data	63
5	Yaw moment diagrams	66
5.1	Introduction	66

5.2	Evaluation of constant velocity yaw moment diagrams	70
5.2.1	Structure of the script	70
5.2.2	Analysis of the results	76
5.3	GGV diagram through yaw moment diagrams	80
5.3.1	Results	81
5.3.2	Simulation 1	81
5.3.3	Simulation 2	85
6	Validations	90
6.1	Point-mass lap time simulator	90
6.1.1	Comparison with the real car	90
6.1.2	Comparison with the transient simulator	95
6.2	GGV diagrams results	102
6.2.1	Comparison with the transient simulator	102
7	Conclusions	106
	Appendix	108
	References	112

List of Tables

1	Points awarded in FSG static events	14
2	Points awarded in FSG dynamic events	16
3	Points awarded in FSAE Italy static events	16
4	Points awarded in FSAE Italy dynamic events	17
5	Comparison between real car and simulator	95
6	Comparison between transient simulator and quasi steady-state simulator	101

List of Figures

1	Skidpad test layout	15
2	2019 team after the first place at FSAE Italy	18
3	Example of a racecar GGV	21
4	Curvature radius of each segment	22
5	Intersection of the three speed profiles	23
6	Simulated velocity profile	24
7	Effect of different damping ratios on the roll angle	25
8	SC21 at the FSS event	27
9	SC21 dimensions	28
10	dSpace MicroAutobox II	29
11	Simulink model for the VCU	30
12	Dimensions of the motors	31
13	CAD model of the inverters	31
14	Torque-speed map of AMK motors	32
15	Efficiency map of AMK motors	33
16	Transmission layout	34
17	Battery pack	35
18	Open circuit voltage of the battery pack as a function of the SOC	36
19	Internal resistance of the battery pack as a function of the SOC	37
20	Example of track used	40
21	Half GGV diagram used to simulate the grip of tires	45
22	Maximum lateral acceleration for different velocities	47
23	Maximum cornering speed for each segment	48
24	Intersection between cornering and braking profiles	51
25	Comparison between first and second loop of profile 2	52
26	Intersection between cornering, braking and acceleration profiles	57
27	Battery pack electrical scheme	59
28	State of charge of the battery pack	61
29	Comparison between first and second loop of profile 3	62
30	Example of final velocity profile	63
31	Absolute value of the lateral accelerations	64

32	Bi-dimensional representation of the vehicle with characteristic angles	66
33	Typical shape of a yaw moment diagram	68
34	ISO vehicle axis system	70
35	Structure of the script	72
36	Structure of the load transfer convergence loop	73
37	Structure of the two iterative loops	75
38	Pure rolling YMD evaluated at 50 km/h	76
39	Pure rolling YMD evaluated at 83 km/h	77
40	Pure rolling YMD evaluated at 126 km/h	78
41	YMD evaluated during an acceleration phase at 126 km/h	79
42	YMD evaluated during a braking phase at 126 km/h	80
43	Approach used to determine the GGV diagram	81
44	Top view of the GGV obtained with simulation 1	82
45	Level curves defining the GGV diagram obtained with simulation 1	83
46	External surface of the GGV obtained with simulation 1	84
47	Cloud of points obtained from yaw moment diagrams in simulation 1	85
48	Top view of the GGV obtained with simulation 2	86
49	Level curves defining the GGV diagram obtained with simulation 2	87
50	External surface of the GGV obtained with simulation 2	88
51	Cloud of points obtained from yaw moment diagrams in simulation 2	89
52	Velocity-distance plot comparison with the real car	90
53	Electric energy-distance plot comparison with the real car	91
54	Longitudinal acceleration-distance plot comparison with the real car	92
55	Lateral acceleration-distance plot comparison with the real car . . .	93
56	Velocity-time plot comparison with the real car	94
57	Electric energy-time plot comparison with the real car	95
58	Velocity-distance plot comparison with a transient simulator	96
59	Electric energy-distance plot comparison with a transient simulator	97
60	Longitudinal acceleration-distance plot comparison with a transient simulator	98
61	Lateral acceleration-distance plot comparison with a transient sim- ulator	99

62	Velocity-time plot comparison with a transient simulator	100
63	Electric energy-time plot comparison with a transient simulator . .	101
64	Comparison between GGV and simulations data	102
65	Comparison between GGV and transient simulation data	103
66	Front view of the GGV surface with respect to the transient simulation data	104
67	Top view of the GGV surface with respect to the transient simulation data	105
68	3D plot of the endurance lap time as a function of the mass and aerodynamic parameters	108
69	2D plot of the endurance lap time as a function of the mass and aerodynamic parameters	109
70	3D plot of the endurance electric energy consumed as a function of the mass and aerodynamic parameters	110
71	2D plot of the endurance electric energy consumed as a function of the mass and aerodynamic parameters	111

1 Introduction

1.1 Formula SAE/Formula Student

1.1.1 History

Formula SAE was created in 1981 in the USA by the Society of Automotive Engineers (SAE, nowadays known as SAE International). Using their definition, the *Formula SAE challenges students to conceive, design, fabricate, and compete with small formula-style racing cars. Teams spend 8-12 months designing, building, and preparing their vehicles for competition. These cars are judged in a series of static and dynamic events, including technical inspection, cost, presentation, engineering design, solo performance trials, and high-performance endurance.*^[1]

These events arrived in Europe in 1998 with the Formula Student UK and in 2006 with the Formula Student Germany.^[2] In the following years, a lot of competitions have been created and nowadays a lot of them are held in Europe, such as Formula SAE Italy, Formula Student Spain, Formula Student Austria and many others.

The European competitions are based on the rules of Formula Student Germany^[3] and of FSAE^[4].

1.1.2 The competition

According to FSG rules, the competition is composed of university teams and each team member must be enrolled as a degree-seeking undergraduate or graduate student. The vehicles must be conceived, designed, and maintained by team members. The direct involvement of professional figures is not allowed.

The event is divided into 3 different classes:

- Internal combustion engine vehicles;
- Electric vehicles;
- Driverless vehicles;

Before the start of the competition, a series of technical inspections are executed

for safety reasons and to control that the vehicle fulfills the rules. If all these inspections succeed, the vehicle can enter the competition.

Teams are evaluated in both static and dynamic events. Static events are composed of:

1. Business plan presentation event, where the team has to develop a business model in which the race car is used to create monetary profit.
2. Cost and manufacturing event, where the team has to evaluate different manufacturing processes and the related cost to build the race car.
3. Engineering design event, where the team has to explain and justify the engineering process behind the development of the race car.

Event	Points
Business plan presentation	75
Cost and manufacturing	100
Engineering design	150
Total	325

Table 1: Points awarded in FSG static events

The final part of the competition is related to dynamic events:

1. Acceleration event, where the performance of the car is evaluated on a 75 meters long straight line.
2. Skidpad event, where the vehicle has to drive on right-hand-side and left-hand-side circles with an inner diameter of 15.25 meters.
3. Autocross event, where the vehicle has to run on a track with a maximum length of 1.5 kilometers.

4. Endurance and efficiency event, where the vehicle has to run on a closed circuit for 22 kilometers, with a driver change in the middle. During the event, the efficiency, in terms of energy consumption with respect to other teams, is also measured.

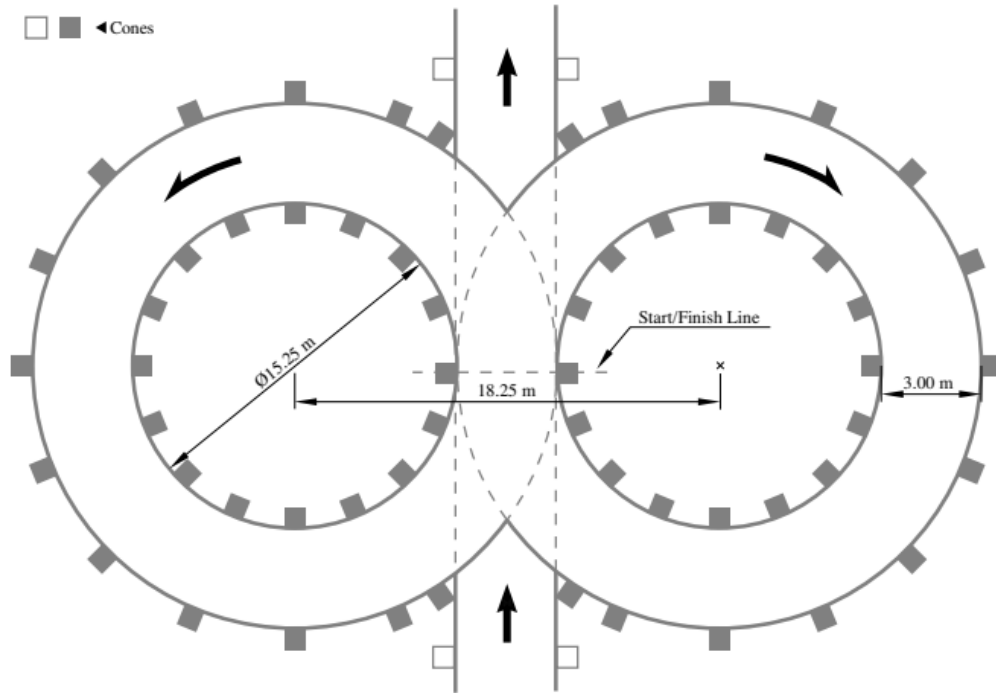


Figure 1: Skidpad test layout

Event	Points
Acceleration	75
Skidpad	75
Autocross	100
Endurance + Efficiency	325 + 100
Total	675

Table 2: Points awarded in FSG dynamic events

For what concerns our main event, Formula SAE Italy, the structure of the competition is the same, but the points awarded for each event are distributed differently.

Event	Points
Business plan presentation	75
Cost and manufacturing	100
Engineering design	150
Total	325

Table 3: Points awarded in FSAE Italy static events

Event	Points
Acceleration	100
Skidpad	75
Autocross	125
Endurance + Efficiency	275 + 100
Total	675

Table 4: Points awarded in FSAE Italy dynamic events

1.2 Squadra Corse PoliTO

Squadra Corse PoliTO is the Formula Student team of the Polytechnic University of Turin. The team was founded in 2004, together with the first cycle of the Automotive engineering degree course.

So far, the team has designed, built, and raced 15 prototypes, divided into 7 internal combustion engine vehicles, 1 hybrid vehicle, and 7 electric vehicles^[5].

According to the world ranking, updated in 2019, Squadra Corse PoliTO is in the 32nd place in the electric category.^[6]

In 2012, the team was the first Italian team to enter the electric category and in 2015, it was the first Italian team to adopt a 4 wheel-drive layout. So far, the team was successful in four events, that are:

- 2009 Formula EHI (Electric Hybrid Italy)
- 2010 Formula EHI
- 2010 Formula Hybrid US
- 2019 Formula SAE Italy



Figure 2: 2019 team after the first place at FSAE Italy

2 Lap time simulation basics

Following the development of digital technology, also automotive engineering, in particular in the motorsport field, has had the same evolution.

Nowadays, every racecar is equipped with a lot of sensors and control units that communicate, through digitalized signals, in controlled area networks (CAN). With a simple data logger, it is possible to collect all these signals and retrieve them to analyze the vehicle dynamics behavior with objective data. In this way, the setup of the car can be improved using both unbiased data coming from the sensors and the driver's feeling.

In this scenario, together with the increased computational power of computers, vehicle dynamics simulations have been developed.

In particular, lap time simulators with different degrees of complexity are extensively used in both the design and the development phases of a racecar. Then, when the track data of the actual vehicle is available, it is possible to compare that with the output of the simulators to check if the level of correlation is sufficient.

2.1 Classification

Lap time simulators can be classified according to their degree of complexity in the solution of the simulation.

Usually they are divided into:

1. **Steady-state simulations**
2. **Quasi steady-state simulations**
3. **Transient simulations**

2.1.1 Steady-state simulations

The simplest approach in terms of computational complexity and data management is the steady-state method. In these simulations, the evolution of the vehicle speed on the track is computed with a distance-based solver. The track is defined as the trajectory that the vehicle has to follow and it is discretized in straights and

corners. In every section, the acceleration to which the vehicle is subjected is kept constant and without interactions between longitudinal and lateral performances. The result is that the vehicle has to brake or accelerate in the straight segments and turn at constant speed in the corners. Each corner is described with an arc of circumference. Knowing the radius of curvature of the segment and the maximum lateral acceleration with the centripetal formula, the speed of the segment can be evaluated.

$$V(i) = \sqrt{Ay_{max} * R(i)} \quad (1)$$

Once all the cornering speeds are evaluated, it is possible to proceed with the straight speed computation. Knowing the initial speed of the straight, which is equal to the previous cornering speed, and the final velocity of the braking phase, equal to the following cornering segment, the equilibrium velocity, when the transition between acceleration and deceleration occurs, can be found.

These simulations are extremely simple in terms of computational complexity, but, on the other side, they are quite inaccurate. The two main simplifications, which are the constant acceleration along the whole segment and the impossibility of combining both longitudinal and lateral accelerations, led to an over-simplified result, especially if the tool is used for motorsport applications. This is mainly due to the strong dependency of the grip capabilities on the velocity of the car, as represented in GGV diagrams.

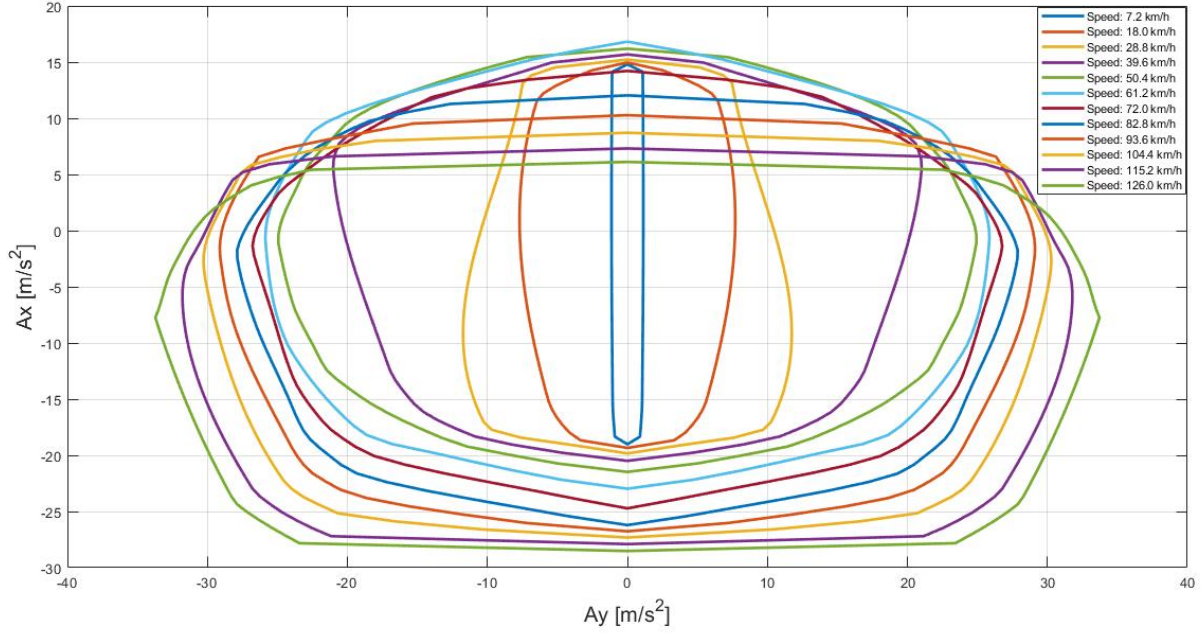


Figure 3: Example of a racecar GGV

On the other hand, this tool can give an acceptable correlation with passenger cars performing very simple maneuvers, where aerodynamic effects are negligible and the velocity range of the vehicles is smaller when compared to race cars. Due to the availability of computational power from PCs nowadays they are replaced by more complex and accurate simulators.

2.1.2 Quasi steady-state simulations

The quasi steady-state simulation approach is an evolution of the previously seen simulation technique, with the aim of keeping the simulation strategy quite simple but, at the same time, to obtain a more realistic and correlated result with reality. The simulation is always computed with a distance-based solver and the track is described with a trajectory.

To achieve this goal, the vehicle is also simulated with combined longitudinal/lateral accelerations through a GGV. With this approach, it is possible to include the

effects that depend on the velocity on the performance of the car, such as the aerodynamics.

To have a more realistic speed profile, the track is discretized in small segments with a length that is usually included in a range between 0.5 to 5 meters. Each segment is characterized by a curvature radius.

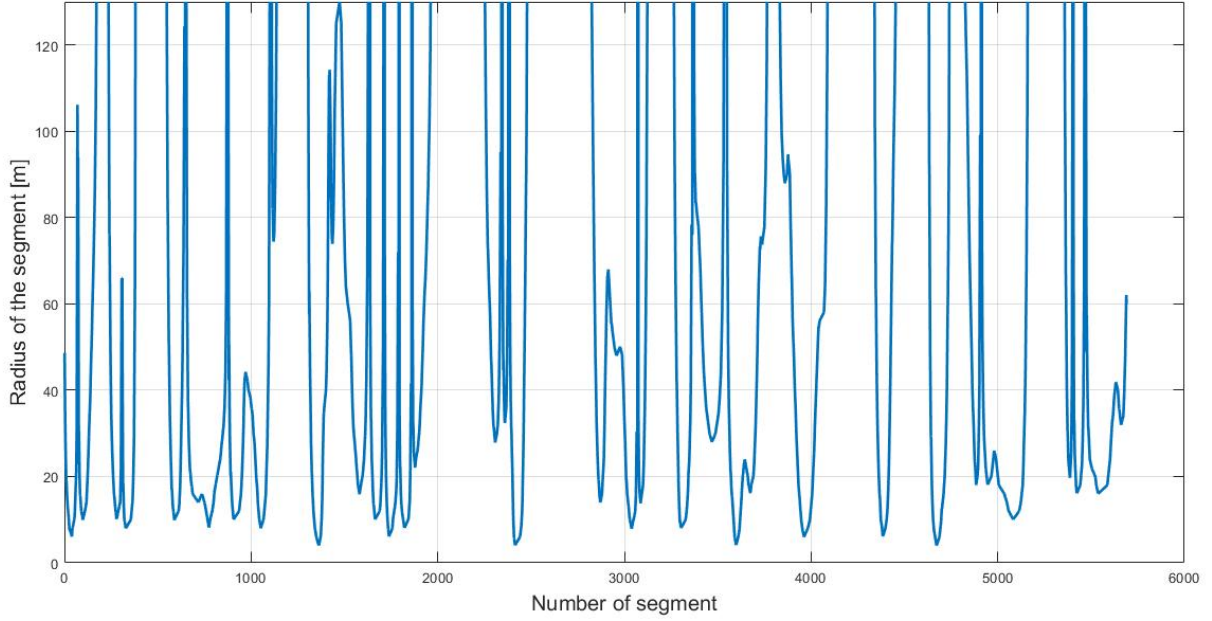


Figure 4: Curvature radius of each segment

Knowing the radii evolution, it is possible to determine the apex of each corner, which is the segment corresponding to the local minimum of the radii plot.

The first step to solving the velocity profile is to evaluate the maximum cornering speed on each apex.

Once the velocity in those segments is known, it is possible to proceed with the computation of the speed profile between corners. This approach relies on the increasing turning radius of segments, before and after the apex, to accelerate and brake the car. Having the starting speed, equal to the apex (i), and knowing that the following segment has a bigger turn radius, the velocity profile considering the

maximum combined longitudinal and lateral acceleration is evaluated.

At the same time, the braking profile is computed similarly to the acceleration one, starting from the apex ($i + 1$) and going in a backward sequence with respect to the acceleration profile.

By comparing the three profiles for each segment and taking the minimum value of velocity, it is easy to find the resulting velocity profile.

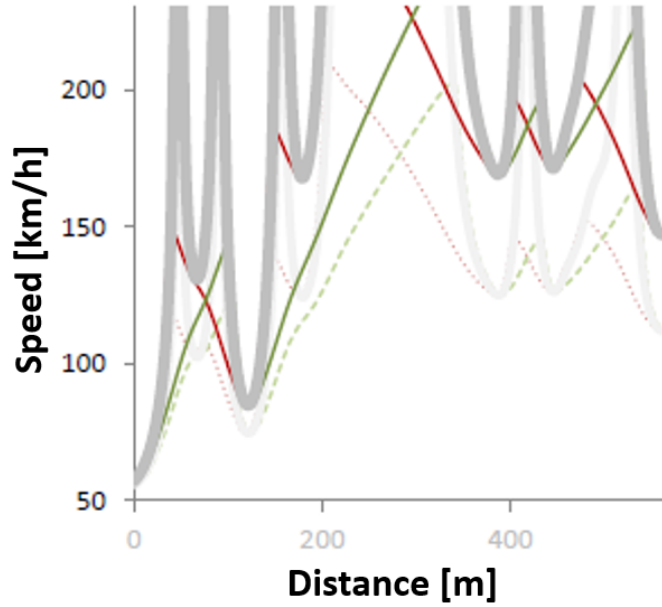


Figure 5: Intersection of the three speed profiles

This simulation technique has a medium complexity and, depending on the selected vehicle model, can be used for different purposes. For example, if a simple point mass vehicle is adopted, the simulator can be used for the preliminary phase in the design stage, also known as target setting, where the influence of the most important parameters (such as mass, aerodynamics, friction coefficients of tires), can be investigated on the lap time and also on the energy/fuel consumption.

If a more complex model is used, such as a bicycle model or a 4 wheel model, it is also possible to simulate load transfers and the grip of the vehicle with a more detailed tire model.

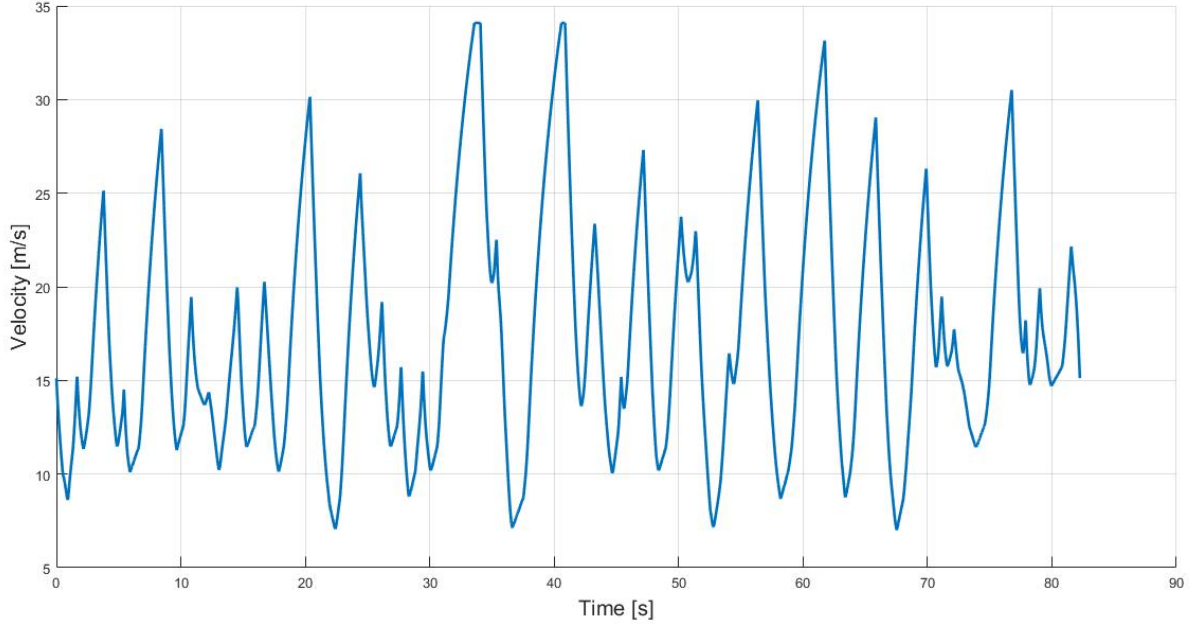


Figure 6: Simulated velocity profile

In conclusion, this type of simulator has an intermediate level of complexity that allows the teams to develop their own code with the required level of accuracy, having also the possibility to modify and update the code for any vehicle without the constraints of a commercial package.

2.1.3 Transient simulations

Transient simulations are the most accurate and complex techniques of lap time simulations. As their name suggests, they consider all the time-varying phenomena of the vehicle without constraining the car to be in steady-state; as a result, the vehicle can have unbalanced forces and moments that create linear and angular accelerations that influence the vehicle dynamics. For example, a complete transient model can simulate all the 6 degrees of freedom of the body with respect to the ground (longitudinal, lateral, and vertical motion together with pitch, roll, and yaw angular displacements) and the two additional degrees of freedom for each

wheel (rotation around the hub axis and vertical travel of the suspension).

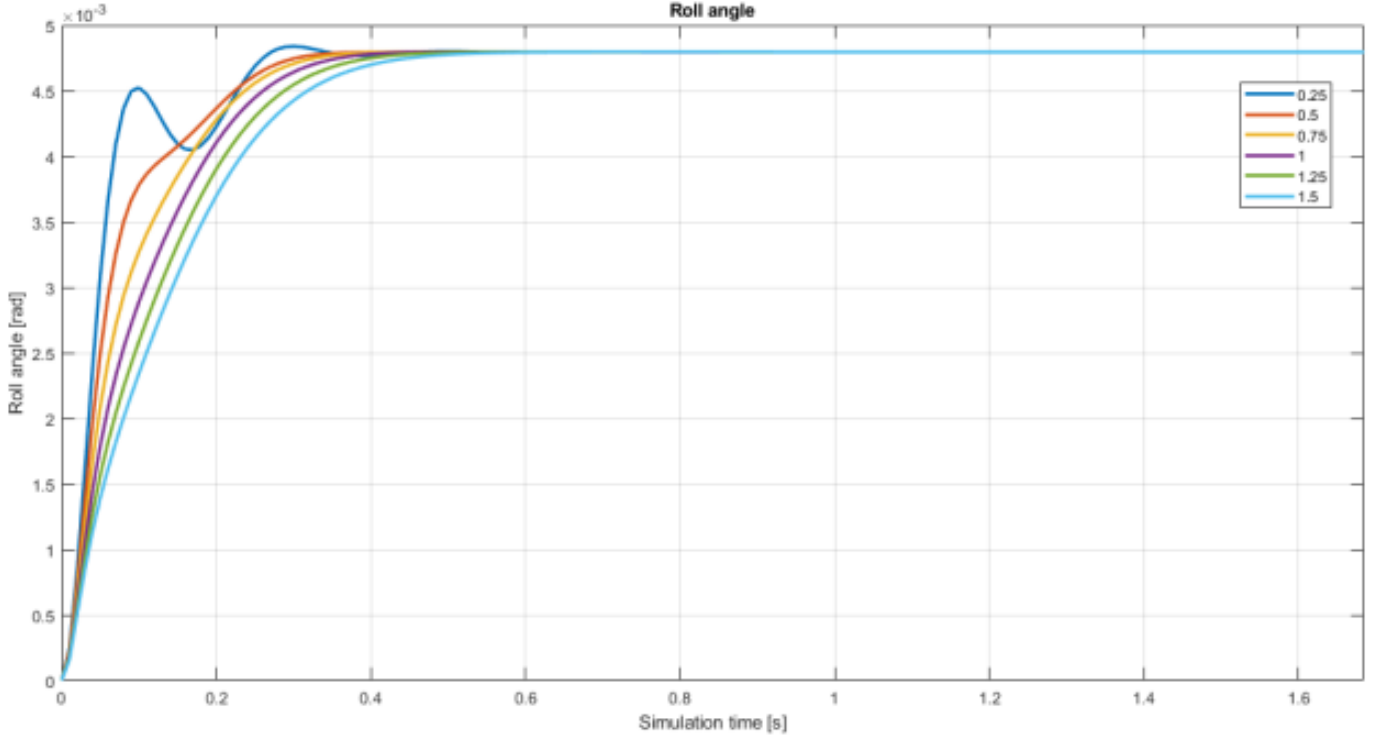


Figure 7: Effect of different damping ratios on the roll angle

The vehicle evolution along the track is simulated through differential equations and, to have a more realistic result, also the driver is modeled with controllers, both on steering and pedals, with a target optimal trajectory.

The car is usually modeled with 5 rigid bodies, one for the sprung mass and the others for the unsprung masses. Simulating the relative positions of the bodies, it is possible to compute all the velocities and accelerations of the vehicle around its reference frame. Modeling suspension elements, such as dampers and springs, tires through the property file, aerodynamics with maps as a function of the sprung mass position and angles, it is possible to update continuously all the forces and moments acting on the vehicle that influence the vehicle dynamics.

Nowadays, in the automotive field, there are three main categories of simula-

tions that are:

1. **Software in the loop:** These simulations allow for integrating the real software of the control unit as part of the simulation. They are also known as co-simulations and they are very important because they can integrate and test the effect of control systems present in the vehicle.
2. **Hardware in the loop:** They are an evolution of the previously explained software in the loop category. The fundamental difference here is that, when available, the physical control unit is installed in series with the simulator, allowing to test both the software and the hardware.
3. **Driver in the loop:** In these simulations, the internal driver of the simulator, modeled with some controllers, is substituted with a proper driver. In this way, simulations are more accurate and drivers can be trained and return their subjective feeling. They are usually adopted for racing applications or, with opportunely developed simulators, to evaluate the ride quality in terms of noise, harshness, and vibration comfort of passenger cars.

3 The vehicle

SC21 is the most recent prototype produced by Squadra Corse to compete in the electric vehicle class of Formula Student competitions. It is designed according to FSG rules. This is the vehicle used for the validation of the simulator on track.



Figure 8: SC21 at the FSS event

The car features a full carbon monocoque, full aero-package composed of front and rear wings, sidepods, undertray and diffuser, double wishbone pushrod actuated front and rear suspensions and a 13 inch wheelset.

For what concerns the dimensions, the car has a wheelbase of 1525 millimeters and a track of 1200 millimeters.

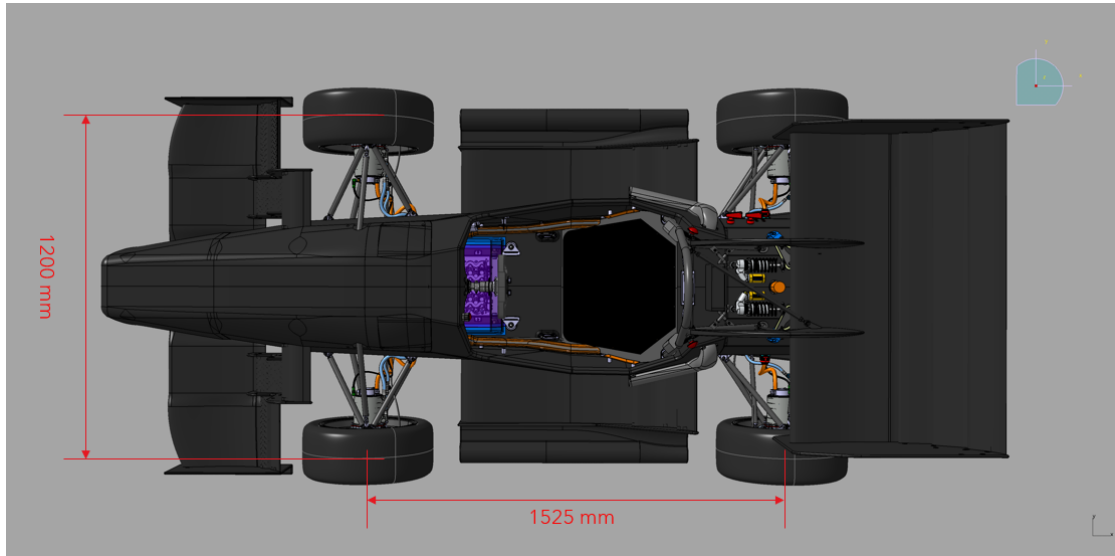


Figure 9: SC21 dimensions

It is powered by the AMK RACING KIT “Formula Student Electric”, which includes 4 inverters and 4 electric motors, installed outboards on the uprights. The battery pack is designed and developed by the team, specifically for this vehicle. For what concerns the electronics, it features a set of self-developed electronic boards for both safety and sensors acquisition. These boards are linked to the AMK powertrain and battery management system through the vehicle control unit, which is a dSpace Microautobox II.



Figure 10: dSpace MicroAutobox II

The communication between all these electronic boards and the VCU occurs through 4 different CAN bus lines, two dedicated to the powertrain and two for sensors acquisition and data logging.

The software used on the vehicle control unit is developed in a Simulink environment and it is divided into 4 main subsystems:

1. CAN setup: block necessary to manage all the parameters of the CAN lines used.
2. CAN LV: it is the part where all the signals acquired from the electronic boards and the low-voltage circuit are pre-processed with their factors and gains.
3. Control system: in this section, signals are combined with different strategies in order to obtain torque requests to the motors that will have the best impact on the performance of the car.

4. CAN HV: it is the final part that manages the communication with the powertrain and receives back information on the status of inverters and motors.

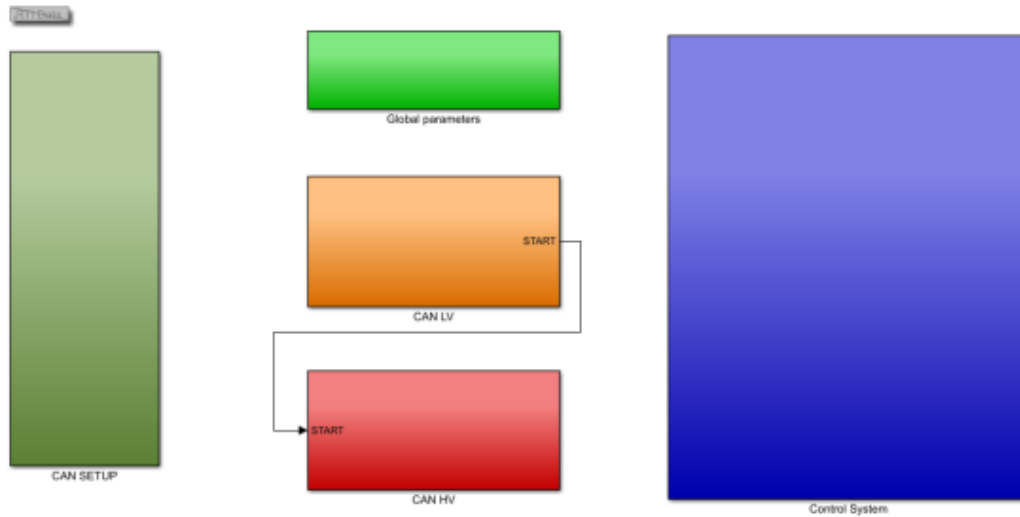


Figure 11: Simulink model for the VCU

3.0.1 Powetrain

As introduced before, the powertrain adopted is the AMK RACING KIT composed of 4 permanent magnet synchronous motors with a peak power of 35 kW each and a maximum torque of 21 Nm, with a maximum angular speed of 20000 rpm.

Each motor is driven by its own inverter and the maximum voltage of the system, because of the Formula Student rules, is 600 V.

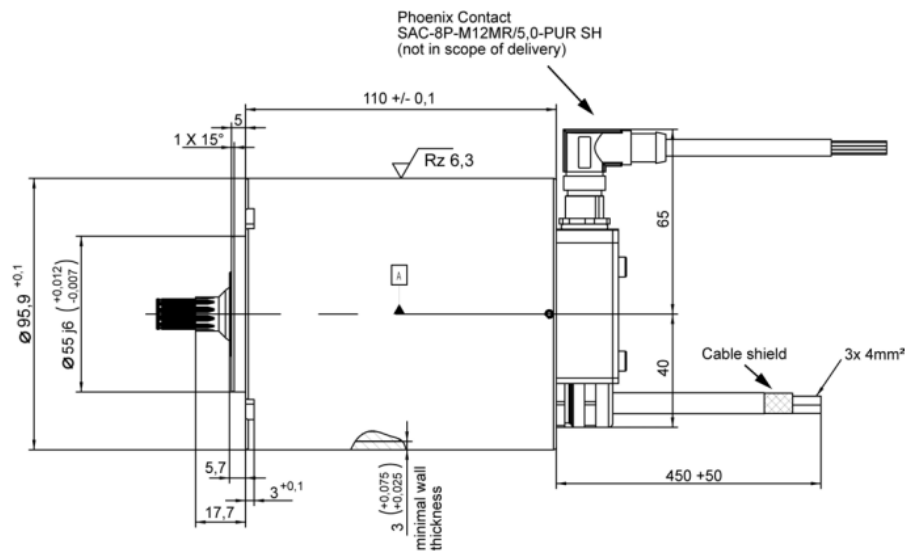


Figure 12: Dimensions of the motors

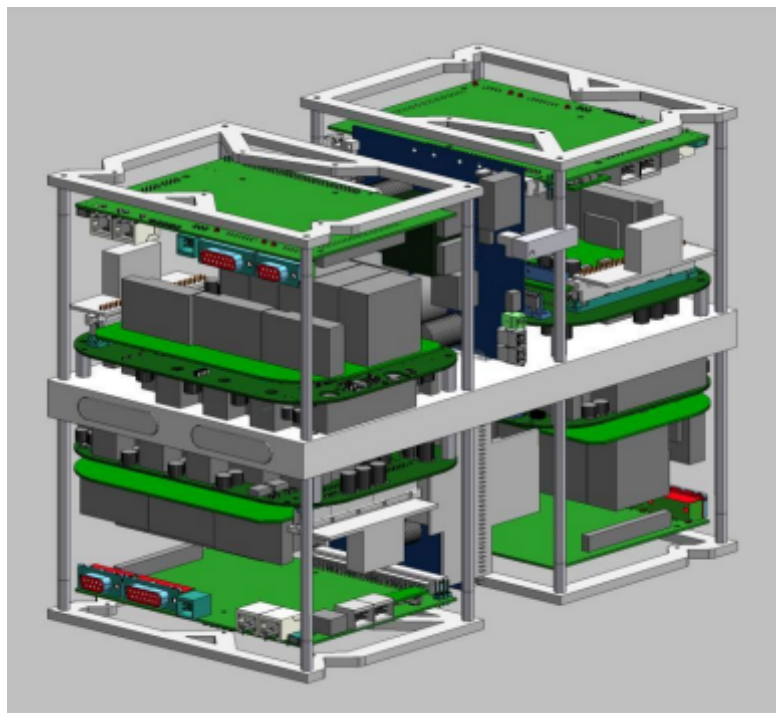


Figure 13: CAD model of the inverters

The supplied voltage is an important parameter for the powertrain performance because of the flux weakening.

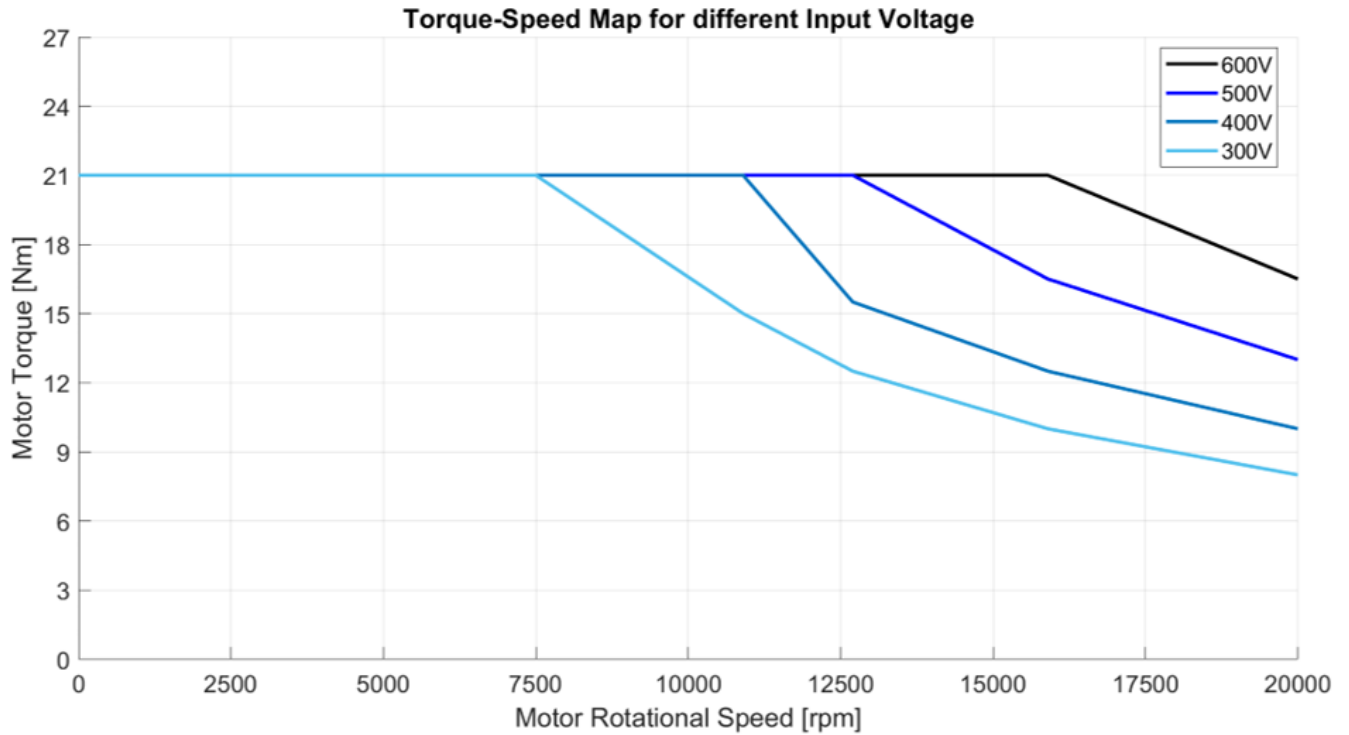


Figure 14: Torque-speed map of AMK motors

Because of their working principle, to reach higher rotational velocity, these motors have to work in flux weakening conditions and the result is a reduced deliverable torque. As reported in the plot, this phenomenon depends on the voltage and, for that reason, during the design of the battery pack, it is important to consider the voltage range.

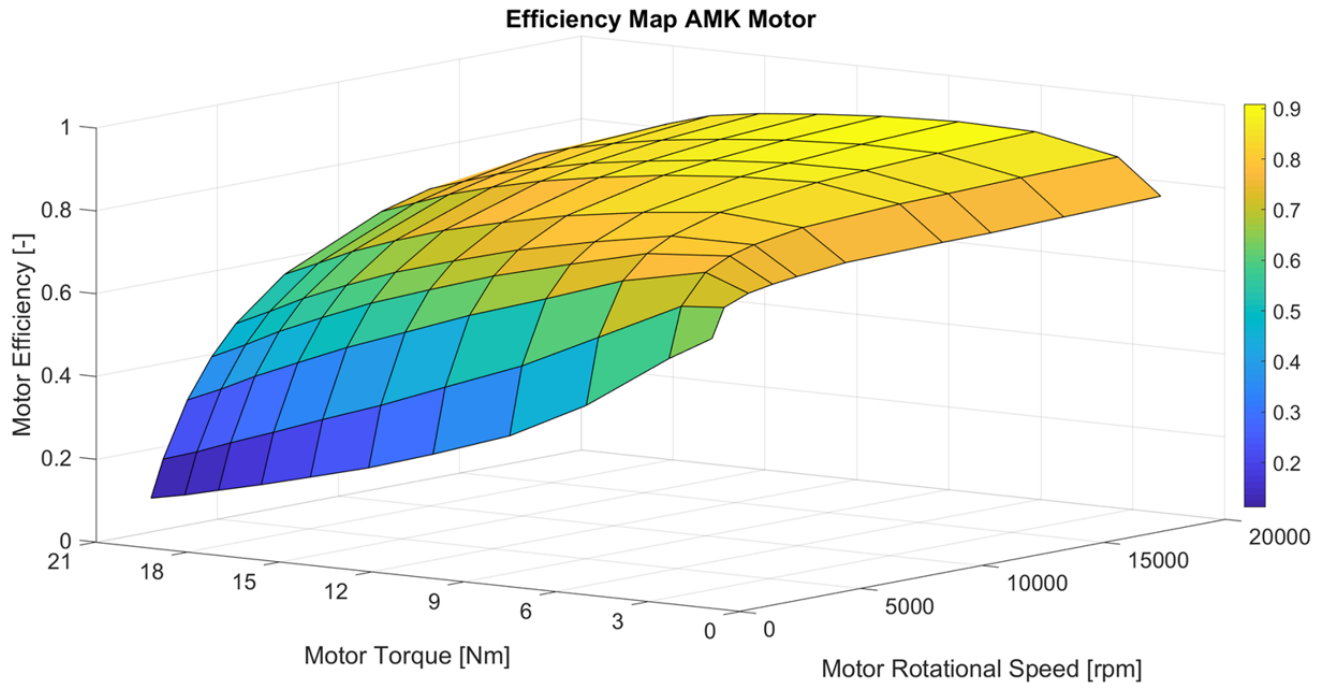


Figure 15: Efficiency map of AMK motors

Another important parameter for the performance and energy consumption of the vehicle is the reduction ratio of the transmissions installed on each motor. These ratios influence the average requested torque and angular speed of each motor, since they are the two most important parameters when defining the electrical efficiency of the motors.

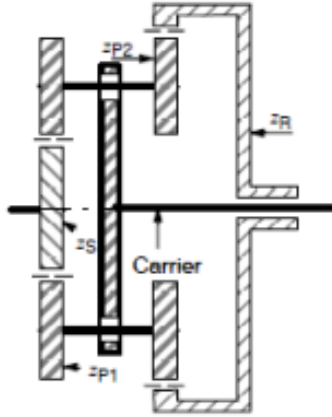


Figure 16: Transmission layout

In the design phase, it is important to evaluate the working ranges of motors on typical formula student tracks and decide the transmission ratio, considering the trade-off between lap times, energy consumed, and performances in the acceleration event. The value chosen for this vehicle is 14.69 and it is the same for the 4 wheels.

3.0.2 Battery Pack

The battery pack is the key subsystem to be designed and has a significant influence on the overall performance of the vehicle. It is the assembly that has the highest impact on the weight of the vehicle, with a mass of around 50 kg. Also the volume necessary for the installation in the monocoque is quite consistent. This component has, together with the driver, the greatest impact on the weight repartition and on the height of the center of gravity. For this reason, its position must be fixed in the earliest phases of the design.

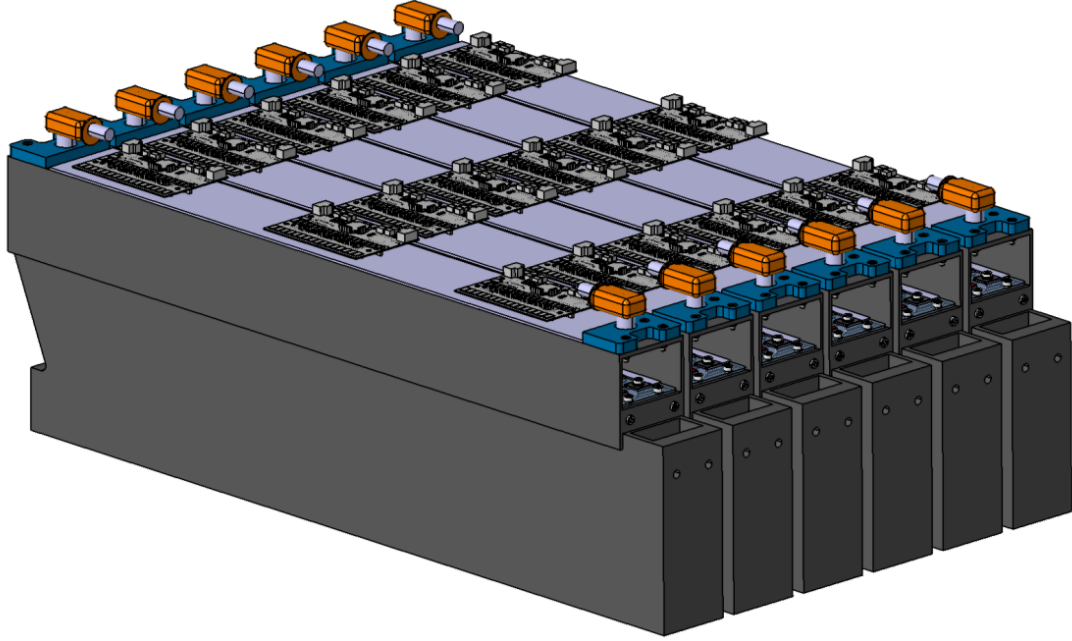


Figure 17: Battery pack

The battery pack is composed of a self-laminated carbon fiber external case and is internally divided into two vertical levels. On the lower one, six cell modules are located, physically divided by carbon and glass fiber panels to ensure proper insulation between them. The modules are then connected in series through high voltage cables. On the upper part, the electronic devices necessary to monitor the cells with the battery management system are installed.

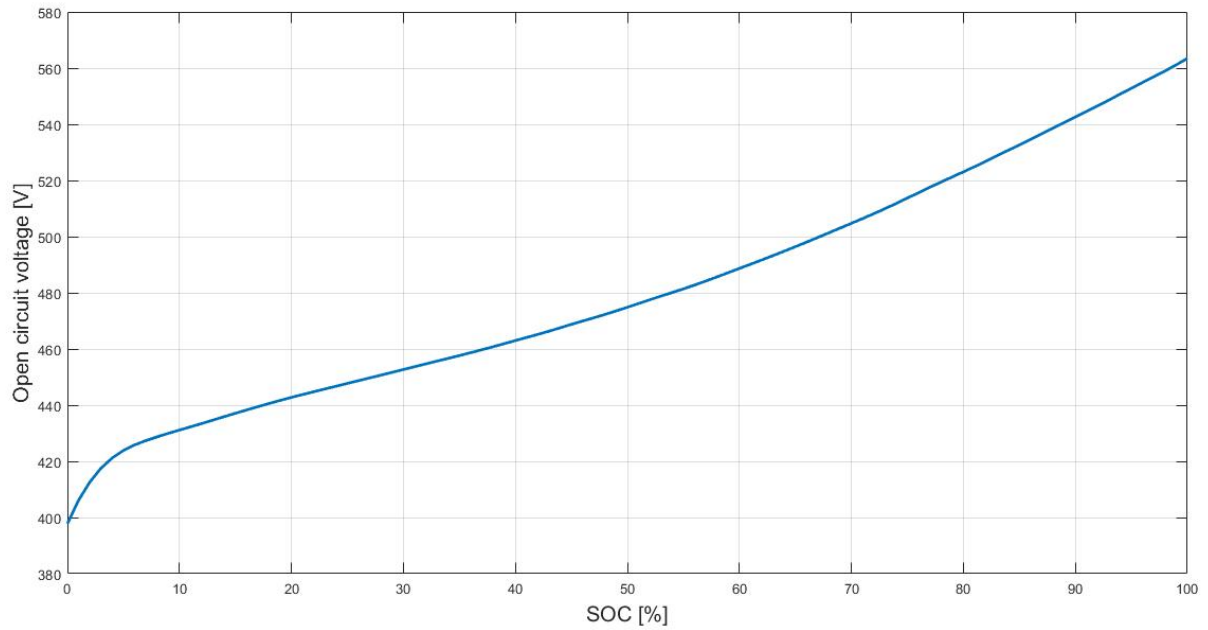


Figure 18: Open circuit voltage of the battery pack as a function of the SOC

The battery cells are responsible for a percentage of around 90% of the battery pack overall mass, which directly depends on the voltage and electrical energy storage needed. For these reasons, simulations are fundamental and have a big impact on the performance of the car; the knowledge of the energy consumption with respect to the lap time has primary importance in the development of the simulator.

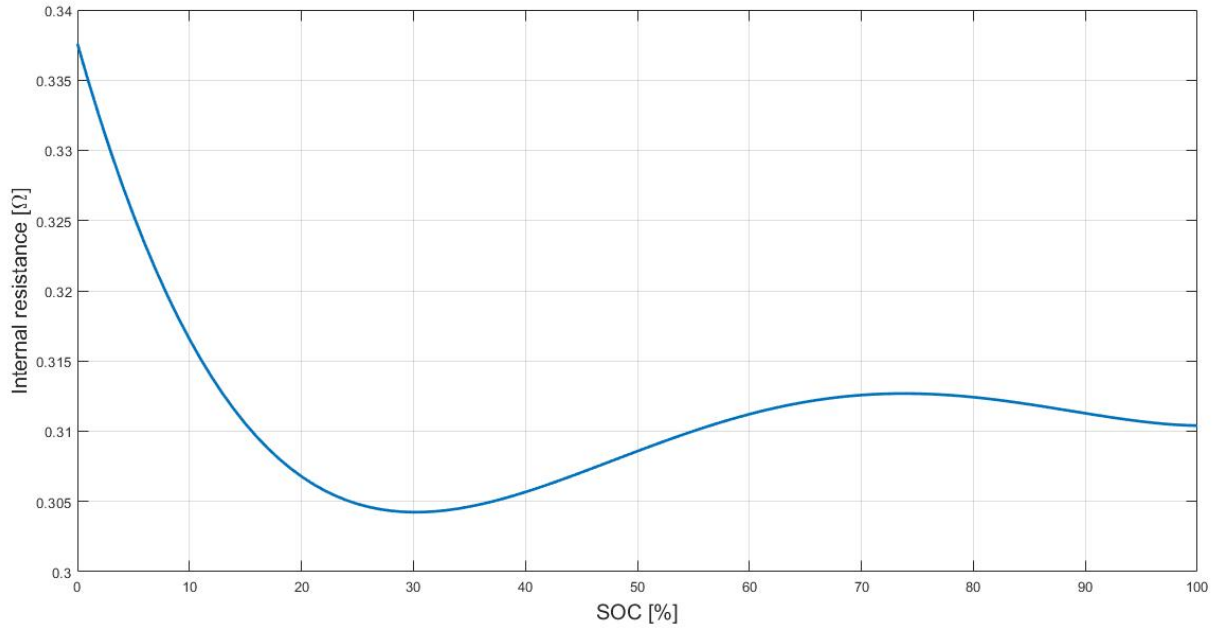


Figure 19: Internal resistance of the battery pack as a function of the SOC

The battery pack of the SC21 features pouch LiPo cells, with an architecture of 132 series and 2 parallels, divided into 6 independent modules.

The total capacity is 7700 Wh, with a nominal voltage of 500 V. Each cell has a nominal voltage of 3.8 V and a capacity of 7700 mAh.

3.0.3 Aerodynamics

The aerodynamic package is composed of front and rear wings, sidepods, undertray, and diffuser. Even though the average velocity in a typical formula student track is quite low when compared to other formula-style cars, around 50 - 60 km/h, the effect of the aeropack is fundamental and must be carefully considered.

In the earliest phases of the design, especially in the target setting phase, it is fundamental to establish the best trade-off for the aerodynamic coefficients to achieve the targeted pace without exceeding the available energy for each lap.

In particular, it is important to consider the efficiency, defined as the ratio between

aerodynamic downforce and aerodynamic drag.

$$\eta_{aero} = \frac{CzA}{Cx A} \quad (2)$$

As is well known, downforce helps the car through an additional vertical load, which allows the tires to have higher grip and develop higher lateral accelerations in the corners. At the same time it is fundamental to control the amount of drag generated because it is one of the main contributions for the power needed for the motion of the vehicle and the resulting electric energy consumption.

Another important parameter that must be considered is the mass of the whole aeropackage and its center of mass. This is fundamental because the weight is not constant but, at least a fraction of it, depends on the amount of downforce and drag, because of the number of wing profiles required and the structural requirements that vary as a function of the aerodynamic forces.

For what concerns the SC21, it has a CzA coefficient of 4.5 and a CxA coefficient of 1.5, for a total efficiency of 3.

4 Development of a point-mass quasi steady-state simulator

4.1 Target setting

This project aims to create and develop a point-mass quasi steady-state simulator for Squadra Corse PoliTO. This simulator will be used in the target setting phase, so the main requirements are a good correlation of the results with the data logged on track but also a relative simplicity to model and simulate the vehicle, without requiring too many inputs, which are not defined in that early phase of the design. This approach will allow simulating the vehicle using the most affecting parameters, performing sensitivity analysis on each one of those, and helping the designers in choosing the best trade-off between those values.

Particular attention is devoted to the modeling of the electric powertrain because of its impact on both laptime and energy consumption. This part is also modeled with a quite simple approach, based on lookup tables, which guarantees the possibility of simulating the vehicle with few parameters.

The simulator is designed with the same approach introduced in the general classification, therefore, it will use the evaluation and intersection of the three speed profiles to determine the final velocity profile. The code is specifically developed for the endurance event, so the lap will be computed as a flying lap. It means that the car starts with a proper speed that is reachable at the end of the previous lap and ends with a velocity compatible with the continuation of the following lap.

For what concerns the programming environment, the script is written in MATLAB[®], since it is the best-known programming language inside the team for future developments.

4.2 Inputs required

4.2.1 Track

The input data necessary to evaluate the performance of the vehicle on the track is the trajectory. The track is discretized in a defined number of elements N , depending on the segment length.

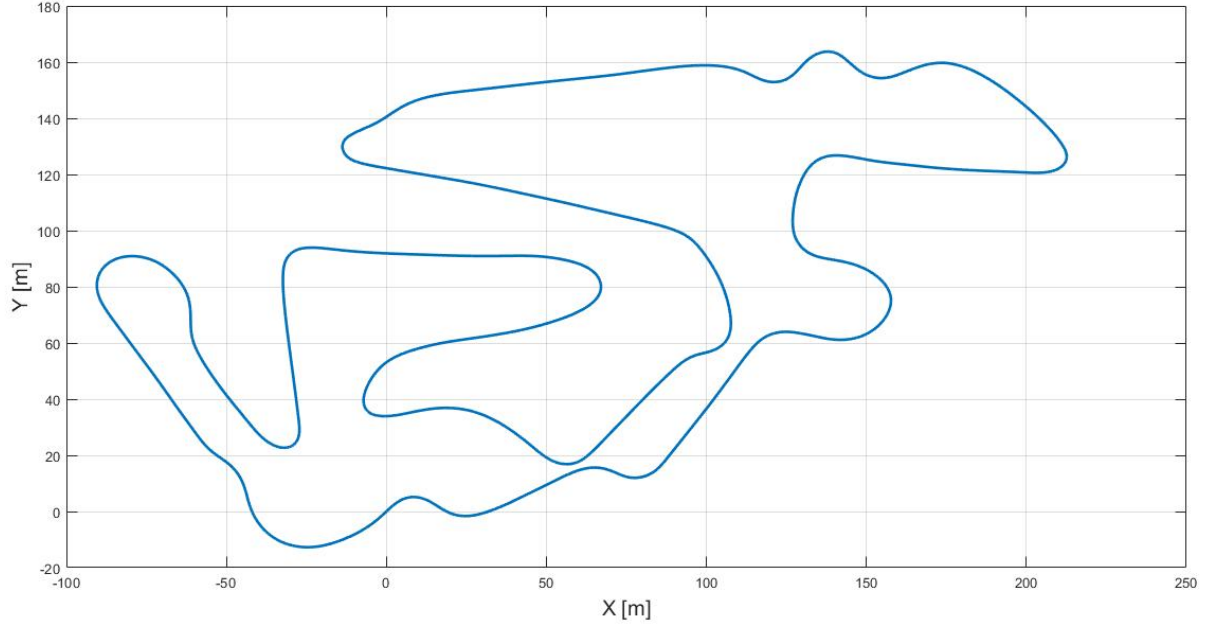


Figure 20: Example of track used

Each trajectory data is composed of a $N * 2$ *TRACK* matrix with the X and Y coordinates expressed in meters, a vector R that contains the curvature radius, evaluated considering three consecutive points, and finally the length of each segment ds . Each element $ds(i)$ contains the distance between the points $(i - 1)$ and (i) .

4.2.2 Vehicle model

The model of the vehicle starts with the definition of the masses of the vehicle and the driver. For what concerns longitudinal accelerations, the equivalent mass is considered. As a result, the inertia of rotating elements, such as wheels and rotors of motors, is introduced.

Using this approach, it is possible to get more realistic accelerations, considering not only the overall mass but also increasing this last one with the effect of the inertia during these acceleration phases.

$$m_{eq} = m + 4 * \left(\frac{J_{wheel}}{R_w^2} \right) + 4 * \left(\frac{J_{motor} * \tau^2}{R_w^2} \right) \quad (3)$$

To model the forces exchanged between the vehicle and the ground through the tires, two friction coefficients μ_x , μ_y are defined at a nominal vehicle vertical force Fz_{nom} , together with two load sensitivity coefficients LS_x and LS_y when the vertical force differs from the nominal one.

The rolling resistance of the vehicle is considered with the parameter f , as a function of the vertical load.

$$f = \frac{F_{RR}}{Fz} \quad (4)$$

The aerodynamic forces are considered with the downforce coefficient CzA and the drag coefficient CxA , together with the air density ρ .

$$Fz_{aero} = \frac{1}{2} * \rho * CzA * V^2 \quad (5)$$

$$Fx_{aero} = \frac{1}{2} * \rho * CxA * V^2 \quad (6)$$

To model the battery pack, the inputs required are the starting SOC of the battery pack, the total energy capacity of the battery, and two lookup tables that define the open-circuit voltage and its internal resistance, both as a function of the state of charge.

The powertrain requires two bi-dimensional lookup tables. The first one defines the maximum torque that each motor can develop, because of the flux weakening, as a function of its angular speed and the supplied voltage from the battery pack. The second one defines the efficiency of each motor as a function of its angular speed and torque delivered.

To simulate the power control system present in the actual car, a power threshold is also introduced, together with a speed limit that acts on the maximum angular velocity of the motors.

Finally, the transmission is described with the transmission ratio τ , the mechanical efficiency, and the wheel radius. These parameters are necessary to correlate the linear speed of the vehicle with the angular speed of the wheels and the motors.

$$\tau = \frac{\omega_{motor}}{\omega_{wheel}} \quad (7)$$

$$\eta_T = \frac{P_{wheel}}{P_{motor}} \quad (8)$$

With these parameters, it is also possible to evaluate the maximum linear speed that the vehicle can reach, not considering any power and space limitations during the simulation, according to the maximum *rpm* defined.

$$V_{max} = \frac{RPM_{max} * \pi * R_w}{\tau * 30} \quad (9)$$

4.3 Main structure

The quasi steady-state simulator is developed through 5 main phases:

1. Definition of the GGV diagram of the vehicle;
2. Evaluation of the maximum cornering speed;
3. Evaluation of the combined cornering-braking speed profile;
4. Evaluation of the combined cornering-braking-acceleration speed profile;

5. Evaluation of auxiliary data.

In the following part of this chapter, a series of plots coming from different simulations are shown. For clarity, to highlight the strategy used for the simulator, different results belonging to various lap simulations are displayed.

4.3.1 Definition of the GGV diagram of the vehicle

The purpose of this diagram in the simulator is to define the grip capabilities of the vehicle without considering the limits introduced by the electric powertrain. The portion of the script used to create the GGV is defined in an external function, to be easily changed and updated without having to modify excessively the main script. As inputs, it takes all the parameters except for the ones of the powertrain. After the pre-allocation of the arrays and matrices, the function starts with the discretization of the velocity range in N_{vel} steps, from 0 to V_{max} . The N_{vel} parameter defines the number of GG diagrams computed to define the GGV diagram. Knowing the velocities, it is possible to define the vertical forces acting between vehicle and ground, which are due to the weight of the car and the aerodynamic downforce.

$$Fz = m * g + \frac{1}{2} * \rho * CzA * V^2 \quad (10)$$

The peak longitudinal and lateral friction coefficients are now determined for each Fz with the load sensitivity coefficients, which describe, in absolute values, the slope of the μ_x and μ_y curves as a function of the vertical force. In other terms, they define the change of friction coefficient value per newton of vertical force variation, with respect to the nominal value. These values are stored in a $3 * N_{vel}$ matrix called *V_muxp-muyp*, with the velocity V in the first column, μ_x in the second one, and μ_y in the third one.

Once the friction coefficients are known for each velocity, the function proceeds with the evaluation of the interactions between longitudinal and lateral friction coefficients at each velocity. This relation is managed by the script through a geometrical ellipse that has the two radii equal to μ_x and μ_y . The ellipse is computed

by another external function and, for each velocity, the μ_y is split into different parts using the *linspace* command, starting from 0 to μ_y peak, in N_{ell} points.

$$\mu_x = \sqrt{\left(1 - \left(\frac{\mu_y^2}{\mu_{yp}^2}\right)\right) * \mu_{xp}^2} \quad (11)$$

Finally, the function returns a number of points, equal to N_{ell} , on the ellipse, with pure longitudinal and lateral friction coefficients at the extremities. The ellipses are computed for both positive and negative longitudinal friction coefficients.

When each ellipse is computed in its function, it comes back to the GGV function and, at last, these points are used to compute the longitudinal and lateral accelerations at that velocity.

$$A_x = \frac{Fz * \mu_x - Fx_{aero} - F_{RR}}{m_{eq}} \quad (12)$$

$$A_y = \frac{Fz * \mu_y}{m} \quad (13)$$

The accelerations are computed according to Newton's second law, and for what concerns the longitudinal direction, the sum of forces is evaluated by considering the aerodynamic drag and the rolling resistance, while, as mass, the equivalent one is considered.

The GGV is stored in a matrix with $(N_{vel} * N_{ell} * 2)$ rows and three columns. The first column contains the velocity, the second column the longitudinal acceleration, and the third column is the lateral acceleration.

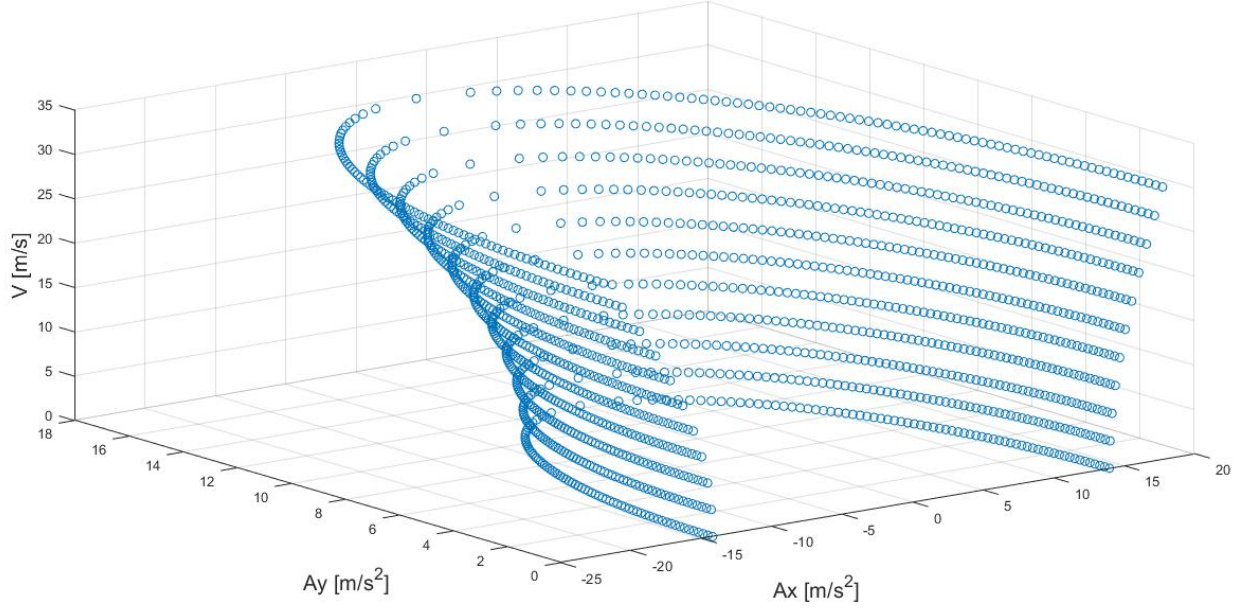


Figure 21: Half GGV diagram used to simulate the grip of tires

At this stage, the external surface of the GGV is defined with the points inside the matrix. To interpolate the surface of the GGV with a given longitudinal acceleration and velocity or lateral acceleration and velocity, the command *scatteredInterpolant* is used. To avoid the double solution, with both positive and negative longitudinal acceleration when interpolating the GGV with A_y and the velocity, the GGV is split into positive and negative matrices.

The outputs of this part are the four *scatteredInterpolant* functions, two for the positive GGV and two for the negative one.

$$A_{y_{pos}} = f(Ax_{pos}, V) \quad (14)$$

$$Ax_{pos} = f(Ay_{pos}, V) \quad (15)$$

$$Ay_{neg} = f(Ax_{neg}, V) \quad (16)$$

$$Ax_{neg} = f(Ay_{neg}, V) \quad (17)$$

Once the grip of the vehicle is modeled through the GGv, it is possible to proceed with the script and start with the evaluation of the velocity profile.

4.3.2 Evaluation of the maximum cornering speed

The simulation of the speed profile starts with the evaluation of the maximum cornering velocity. It is defined as the speed that the vehicle could reach on a segment characterized by a path curvature R , independently of the segments that are present before and after the one considered. The vehicle is simulated in pure lateral acceleration condition, so the car has no longitudinal accelerations.

$$V_{profile1} = \sqrt{Ay * R} \quad (18)$$

The key problem in computing this velocity profile is that the lateral acceleration depends on the velocity, because of aerodynamics and load sensitivity. This implies that, to evaluate these velocities, it is necessary to resort to an iterative procedure.

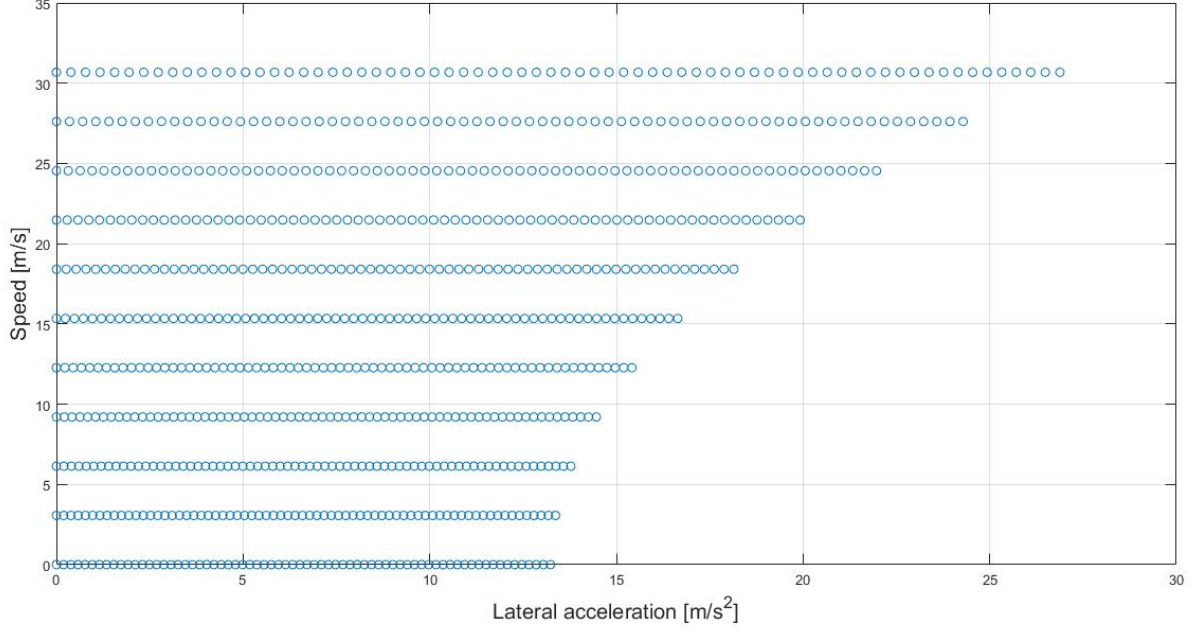


Figure 22: Maximum lateral acceleration for different velocities

Knowing that the lateral acceleration profile has an increasing trend with respect to the velocity, the speed profile is evaluated with a trial-and-error technique. The computation is performed in an external function, which takes as input the curvature radius, the lateral interpolating function, and the maximum velocity of the vehicle.

Inside the function, the velocity range is discretized in N steps and, starting from the maximum speed, the test lateral acceleration is evaluated.

$$Ay_{test} = \frac{V_{test}^2}{R} \quad (19)$$

As soon as Ay_{test} is lower than the lateral acceleration that the vehicle can develop in pure lateral condition, with the same V_{test} , the iterative procedure is stopped and the function returns the last V_{test} as maximum cornering speed for that segment.

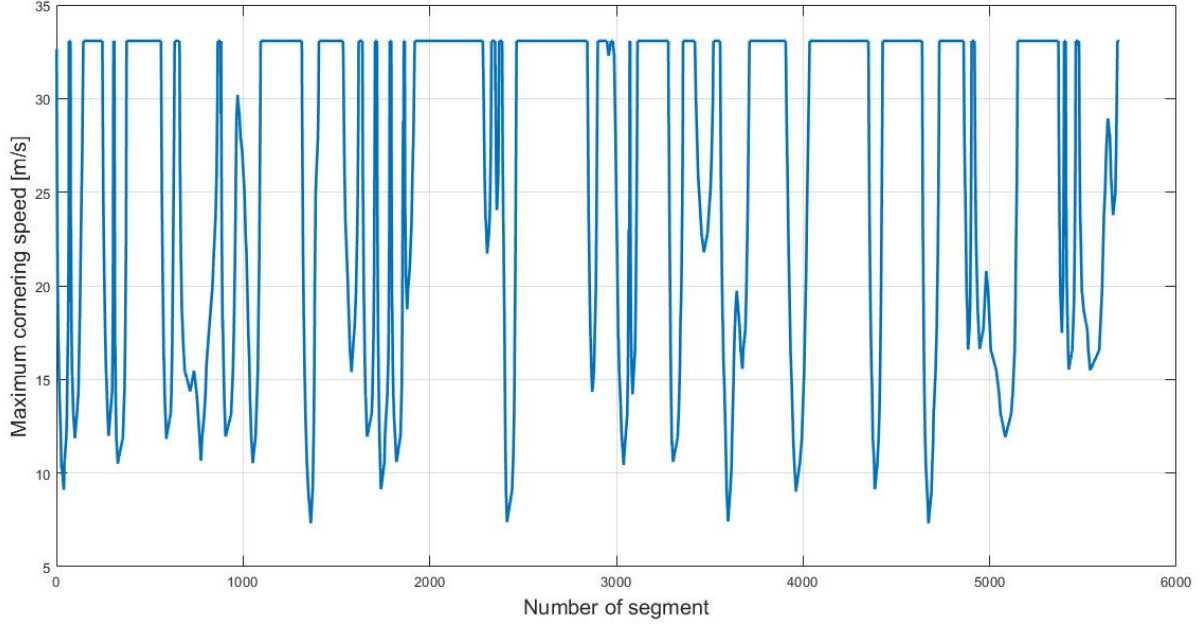


Figure 23: Maximum cornering speed for each segment

As reported in the figure above, the velocity profile obtained has quite a similar shape with respect to a real one. The major difference is that the rate of change of the velocity between segments is significantly high, so the longitudinal accelerations are severe and not practicable for the vehicle.

This profile is useful as a starting point and reference for setting a threshold velocity on each segment. In the final velocity plot, there will be at least a series of points in which the speed is equal to the one evaluated in this profile. These points are the apexes of each corner, where the vehicle runs across the segment in pure lateral condition. Once the speed of these points is computed, it is possible to continue with the two other profiles, one which considers the braking phase and another one that considers the acceleration of the car.

4.3.3 Evaluation of the combined cornering-braking speed profile

Having previously computed the maximum cornering velocity on each segment, it is possible to continue with the speed profile evaluation determining the interaction between lateral and braking accelerations.

The computation occurs in a backward *for* cycle, starting from the segment N up to the first one. Inside each cycle, the velocity of the segment $(i - 1)$ is computed knowing the velocity (i) ; reversing the cycle, the braking phase can be seen as an acceleration phase. The main issue is that, to brake and stay on the required trajectory at the point (i) , the vehicle has to develop certain accelerations, which depend on the velocity at the point $(i - 1)$ and not on the one at (i) . In the simulator, the script is solved with an iterative procedure. As a first attempt, it is possible to evaluate the required lateral acceleration at the point $(i - 1)$ with the velocity $V(i)$.

$$Ay(i - 1) = \frac{V(i)^2}{R(i - 1)} \quad (20)$$

Considering starting from the apex (i) , the radius $R(i - 1)$ is going to be slightly bigger and the resulting lateral acceleration will be lower than the one at point (i) . Therefore, interpolating the GGV diagram with $Ay(i - 1)$ and $V(i)$ will give a residual longitudinal acceleration that the vehicle can use to brake and reduce the velocity between points $(i - 1)$ and (i) .

$$Ax(i - 1) = f(Ay(i - 1), V(i)) \quad (21)$$

$$V(i - 1) = \sqrt{V(i)^2 + 2 * Ax(i - 1) * ds(i)} \quad (22)$$

As explained before, this method assumes that $V(i - 1)$ is equal to $V(i)$, which

leads to an underestimated lateral acceleration $Ay(i - 1)$, because the real velocity $V(i - 1)$ will be higher. The result is that, depending on the conditions, the interpolation to find the residual longitudinal acceleration can be overestimated and therefore the car cannot sustain that acceleration.

This assumption of equal velocity introduces a sort of inaccuracy in the profile but, if the segments are short enough to have minor differences between velocities, this method can provide a satisfactory result.

To get a successful outcome, even with a different discretization length, this simulator evaluates the correct $V(i - 1)$ with an iterative procedure. When the velocity is computed with the first attempt, the parameters $V(i - 1)$, $V(i)$, $R(i - 1)$, $ds(i)$, and the interpolating functions of the GGV are passed to an external function.

In this function, the velocity found on the first try, here known as estimated velocity, $V_{estimated} = V(i - 1)$, is used to check if the deceleration can be performed by the car.

$$Ay_{estimated} = \frac{V_{estimated}^2}{R(i - 1)} \quad (23)$$

$$Ax_{estimated} = \frac{V_{estimated}^2 - V(i)^2}{2 * ds(i)} \quad (24)$$

If the estimated longitudinal acceleration is lower or equal to the one found with the interpolating function, the velocity $V(i - 1)$ computed with the first attempt is correct and the function returns the same value. If it is not, the function reduces the estimated velocity by a determined quantity until a solution that satisfies the requirements can be found.

$$Ax_{estimated} \leq Ax(Ay_{estimated}, V_{estimated}) \quad (25)$$

When the correct velocity $V(i - 1)$ returns to the simulator, it is compared with the maximum cornering speed $V_{profile1}(i - 1)$; the minimum between these two values becomes the new value of $V_{profile2}(i - 1)$.

$$V_{profile2}(i - 1) = \min(V(i - 1), V_{profile1}(i - 1)) \quad (26)$$

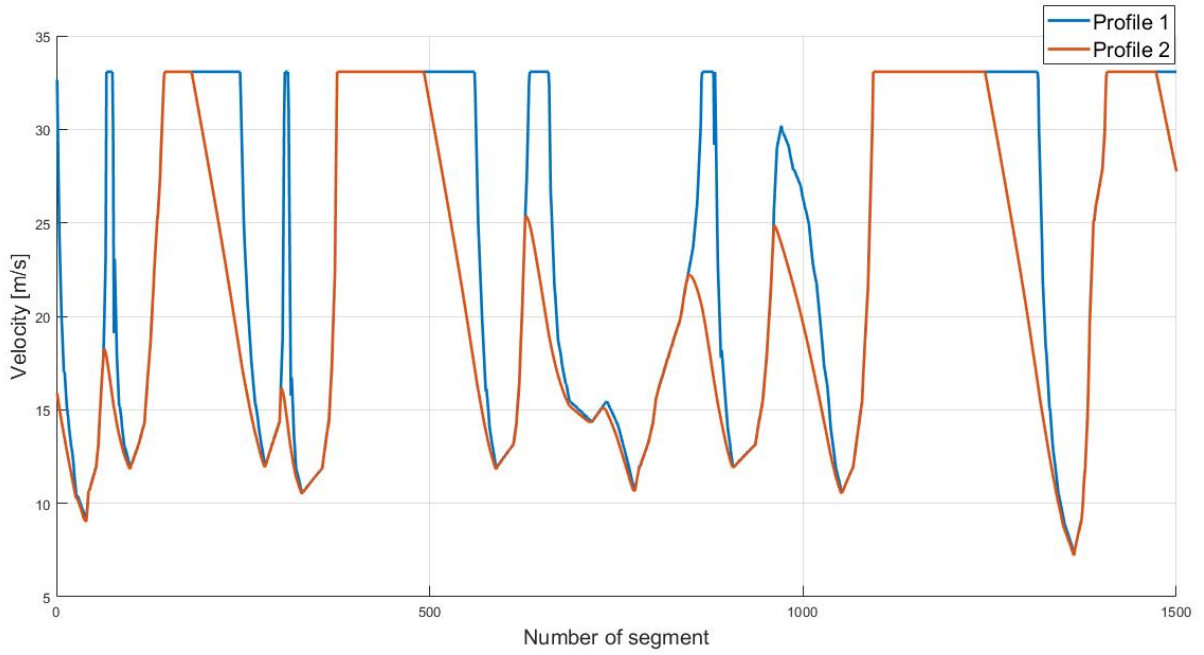


Figure 24: Intersection between cornering and braking profiles

Because of the flying lap requirement, it is necessary to simulate the braking profile into 2 different loops.

The first one is only used to have the continuity of the braking phase in case the simulation starts quite close to a corner. In this scenario, the vehicle needs to start the lap with a braking phase and knowledge of the previous speed is necessary.

Hence, the target of the first computation loop is to evaluate the velocity of the combined profile in the last segment $V_{profile2}(N)$. To reduce the computational time, it is also possible to shorten the first loop by starting from the first apex with the backward speed evaluation.

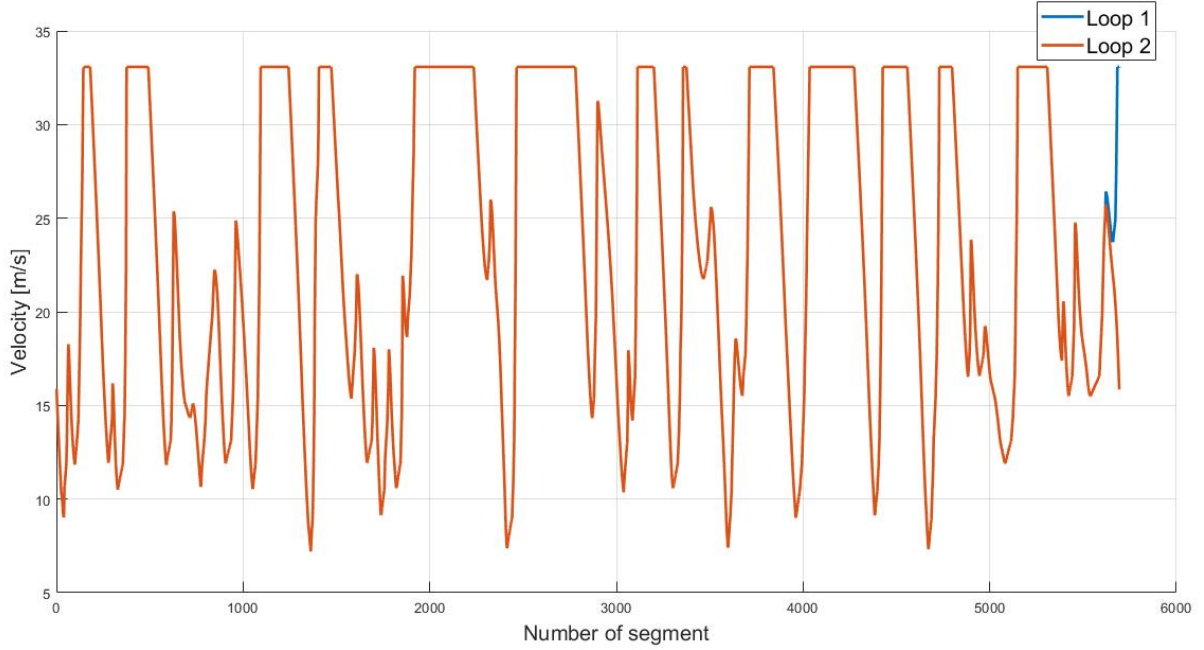


Figure 25: Comparison between first and second loop of profile 2

Once the velocity in (N) is known, the second loop is implemented to compute all the velocities until $V_{profile2}(1)$.

4.3.4 Evaluation of the combined cornering-braking-acceleration speed profile

After the evaluation of the combined cornering and braking velocity profile, it is necessary to evaluate the final result, also considering the acceleration phase. For what concerns the computational complexity, this phase is quite simple because it does not require the application of iterative techniques, as the velocity in point (i)

is already known and it is possible to interpolate directly to compute the velocity in the next segment ($i + 1$).

The complex part of this phase is that, together with the interpolations, it is necessary to simulate the electric powertrain and compare the possible acceleration due to the grip of the vehicle with the one allowed by the motors.

Similarly to the cornering-braking velocity profile, during this phase, it is also necessary to compute the acceleration in two different loops, to simulate the flying lap. The target of the first loop is to evaluate the initial velocity $V(1)$ of the second lap.

The first loop starts with the interpolation of the open circuit voltage-SOC and internal resistance-SOC curves, with the initial value of the state of charge, which is defined by the user as an input. These values, *OC_voltage* and *R_in*, are kept constant for the whole first cycle because, in reality, it is not performed by the vehicle, hence the electric energy is not consumed. The loop starts with a velocity $V(1)$ equal to zero and, knowing the actual velocity in point (i), the first elements that can be evaluated are the required lateral acceleration, the vertical force applied by the vehicle to the ground, and the angular speed of the motors.

$$Ay(i) = \frac{V(i)^2}{R(i)} \quad (27)$$

$$Fz(i) = m * g + \frac{1}{2} * \rho * CzA * V(i)^2 \quad (28)$$

$$RPM_{motor}(i) = \frac{V(i) * \tau * 30}{R_w * \pi} \quad (29)$$

Knowing the angular speed of the motors, it is necessary to find the maximum torque that can be requested, which, because of the flux weakening phenomenon,

can be lower than the maximum one. This value is found with a *scattered Interpolant* command, interpolating the two-dimensional map, defined as input, as a function of angular speed and battery voltage. The voltage used as a reference is the previous one, called $V_{BP}(i - 1)$. This value is the dynamic voltage, which is computed as the sum of the open-circuit voltage and of the voltage drop due to the current drawn from the battery pack. For the first step, the open circuit value is used, since the vehicle starts from a standstill position.

$$T_{lim}(i) = f(V_{BP}(i - 1), RPM_{motor}(i)) \quad (30)$$

Afterward, the power control function is used to check, and eventually rescale, that value of torque, to stay below the power threshold limit imposed as input. It is implemented in an external function and it takes as inputs the maximum torque requestable, the angular speed of motors, the motors efficiency map, and the maximum power allowed. From this point on, the simulator is developed with equal torque repartition between axles. If the repartition is not equal between the axles, it is quite easy to introduce a rescaling coefficient that decreases the torque on one axle.

$$T_{PC}(i) = f(RPM_{motor}(i), T_{lim}(i), \eta_m, P_{lim}) \quad (31)$$

Inside this function, the overall electrical power is computed through the mechanical one and through the interpolation of the efficiency map. The efficiency is also defined with a *scatteredInterpolant* command and it is evaluated as a function of the requested torque and the angular speed.

$$\eta_m = f(RPM_{motor}, T_{required}) \quad (32)$$

$$P_{mech} = 4 * T_{required} * \frac{\pi * RPM_{motor}}{30} \quad (33)$$

$$P_{elt} = \frac{P_{mech}}{\eta_m} \quad (34)$$

If the evaluated electrical power is below or equal to the threshold power, the function returns the value of torque $T_{required}$, which was given as an input to the main script. If the condition is not satisfied, the power control function starts reducing the requested torque by a user-defined quantity until the exit condition is respected.

$$P_{elt} \leq P_{lim} \quad (35)$$

Going back to the main script, when the maximum torque $T_{PC}(i)$ that each motor can develop is known, it is possible to compute the maximum longitudinal acceleration that motors could express. This value is computed considering the maximum longitudinal force developed by motors on the ground, reduced by the aerodynamic drag and rolling resistance, and, finally, divided by the equivalent mass.

$$Ax_{motors}(i) = \frac{1}{m_{eq}} * \left(\frac{4 * T_{PC}(i) * \tau * \eta_t}{R_w} - \frac{1}{2} * \rho * Cx A * V(i)^2 - f * Fz(i) \right) \quad (36)$$

Interpolating the GGV with the previously computed required lateral acceleration and velocity, the maximum longitudinal acceleration that the vehicle can sustain, because of the grip of the tires, is found.

$$Ax_{tires}(i) = f(Ay(i), V(i)) \quad (37)$$

Comparing these two longitudinal accelerations and selecting the lower one, the maximum acceleration that the vehicle can perform is determined, and the following velocity can be computed.

$$Ax(i) = \min(Ax_{motors}(i), Ax_{tires}(i)) \quad (38)$$

$$V(i+1) = \sqrt{V(i)^2 + 2 * Ax(i) * ds(i+1)} \quad (39)$$

Finally, the velocity of profile 3 is determined by choosing the minimum value between the 3 profiles.

$$V_{profile3}(i+1) = \min(V_{profile1}(i+1), V_{profile2}(i+1), V(i+1)) \quad (40)$$

Once the correct velocity in point $(i+1)$ is known, the actual longitudinal acceleration $Ax(i)$ that the vehicle has to develop is evaluated. This acceleration differs from the one evaluated in formula (38) because that one was considered as the maximum value for the acceleration profile, while this one considers the final velocity profile and can be positive, negative, or equal to zero.

$$Ax_{profile3}(i) = \frac{V_{profile3}(i+1)^2 - V_{profile3}(i)^2}{2 * ds(i+1)} \quad (41)$$

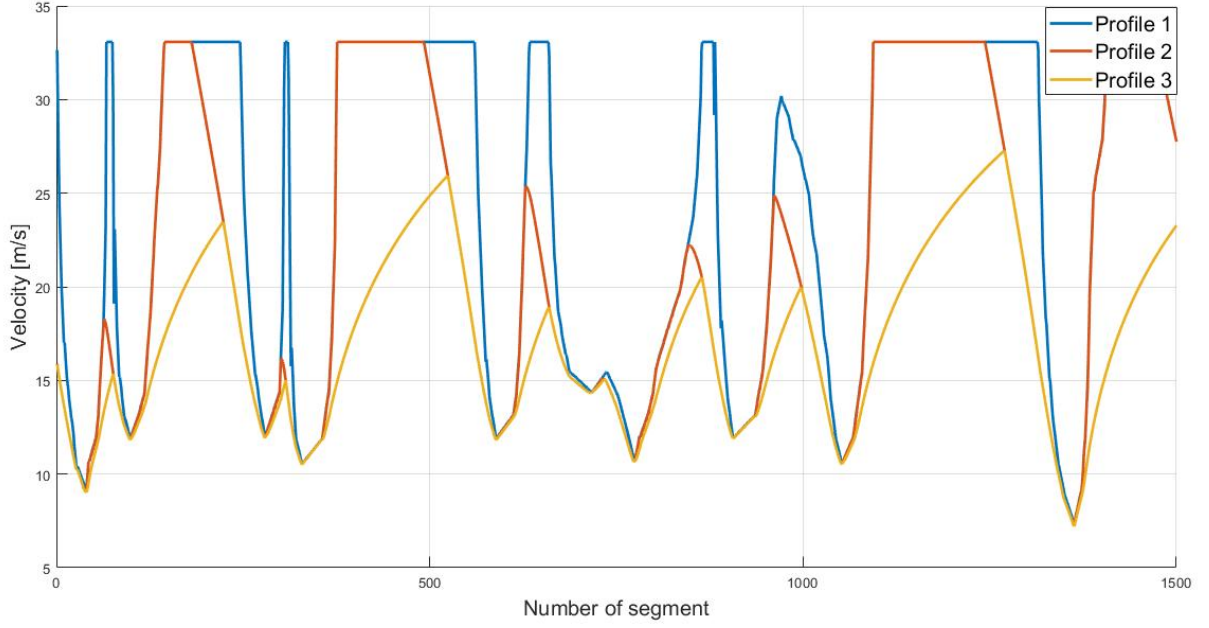


Figure 26: Intersection between cornering, braking and acceleration profiles

Now, the last aspect that has to be evaluated is the voltage of the battery pack $V_{BP}(i)$, necessary for the next cycle.

With the knowledge of the real longitudinal acceleration developed by the vehicle, the total amount of torque requested from motors, together with the mechanical power, is determined. If the acceleration is negative, both torque and power are set equal to zero.

$$Torque(i) = \left(\frac{1}{2} \rho * Cx A * V_{profile3}(i)^2 + f F z(i) + m_{eq} A x_{profile3}(i) \right) \frac{R_w}{\tau * \eta_t} \quad (42)$$

$$P_{mech}(i) = Torque(i) * RPM_{motor}(i) * \frac{\pi}{30} \quad (43)$$

Knowing the overall mechanical power and torque, the electrical power required for this cycle, drawn from the battery pack, is finally evaluated. To compute the efficiency, it is necessary to go back to the torque and mechanical power developed by each motor and, after that, interpolate the efficiency map.

$$\eta_m = f(RPM_{motor}(i), Torque(i)/4) \quad (44)$$

$$P_{eltmotor}(i) = \frac{P_{mech}(i)}{4 * \eta_m} \quad (45)$$

$$P_{elt}(i) = 4 * P_{eltmotor}(i) \quad (46)$$

This requested electrical power is then sent to the battery pack model, managed in an external function, together with the constant open circuit voltage and internal resistance, evaluated at the start of this first loop of simulation.

$$[V_{BP}(i), i_{BP}(i)] = f(P_{elt}, OC_voltage, R_{in}) \quad (47)$$

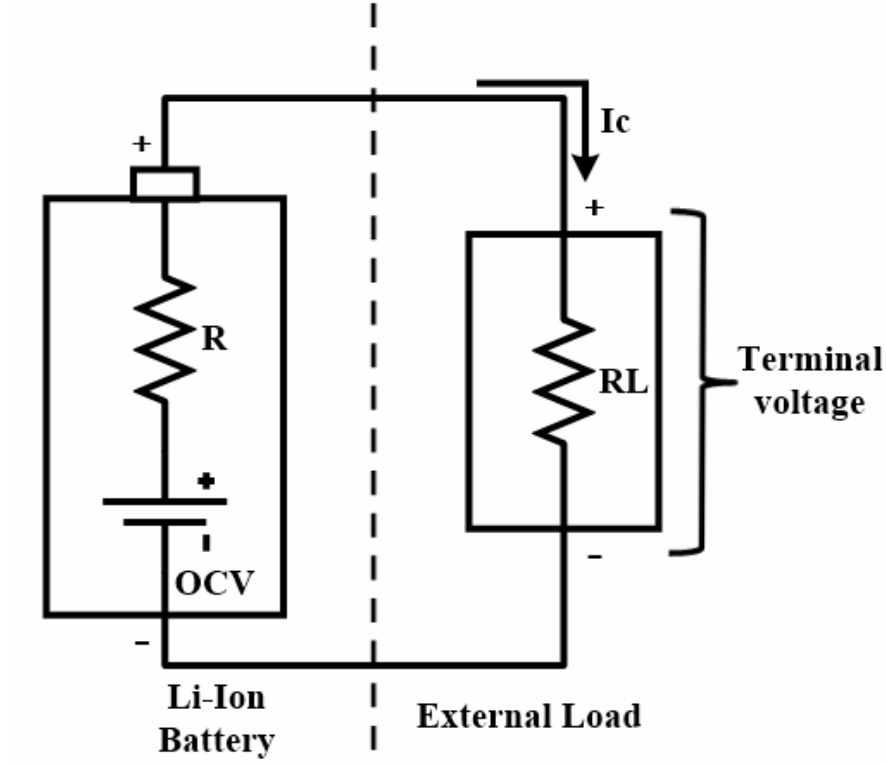


Figure 27: Battery pack electrical scheme

The battery pack function simply models the battery as an ideal voltage source connected in series with the internal resistance and the load. Knowing the electric power dissipated by the load and applying Kirchhoff's voltage law on this simple circuit, it is possible to write the second-order equation that defines the current drawn from the battery pack.

$$i_{BP} = \frac{OC_voltage - \sqrt{OC_voltage^2 - 4 * R_in * P_{elt}}}{2 * R_in} \quad (48)$$

$$V_{BP} = OC_voltage - R_in * i_{BP} \quad (49)$$

At this point of the script, the first loop is completed, and the initial velocity $V(1)$ of the second loop is available. To reduce the computation time, also for this profile the first loop can be shorter, starting the evaluation of the velocity from the last apex of the track.

The major difference for the second *for* loop is that, in this one, the open-circuit voltage and internal resistance are no longer constant, but they need to be updated every cycle, as the electrical energy of the battery pack is consumed.

Basically, the structure is the same from equation (27) to equation (46). Together with the evaluation of the electrical power $P_{elt}(i)$, the time dt necessary to travel from point (i) to point $(i + 1)$ is also found. It is evaluated by computing the area below the curve of the speed as a function of the distance for each segment. If the average speed is considered, this shape is simply a rectangle.

$$V_{average} = \frac{V_{profile3}(i)^2 + V_{profile3}(i + 1)^2}{2} \quad (50)$$

$$dt(i + 1) = \frac{ds(i + 1)}{V_{average}} \quad (51)$$

The knowledge of the time dt is necessary to determine the electric energy consumption for each segment and to update the state of charge of the battery pack.

$$Elt_{energy}(i) = Elt_{energy}(i - 1) + dt(i) * P_{elt}(i - 1) \quad (52)$$

Essentially, the electric energy evaluated in point (i) is the electric energy consumed up to this covered distance and, for this reason, the electric power considered is the one saved in $(i - 1)$, which represents the acceleration between $(i - 1)$ and (i) . Moreover, $dt(i)$ defines the time necessary between $(i - 1)$ and (i) .

If the thermal power dissipated because of the Joule effect is not negligible, it is

also possible to consider that contribution.

$$Elt_{energy}(i) = Elt_{energy}(i-1) + dt(i) * (P_{elt}(i-1) + R_{in}(i-1) * i_{BP}(i-1)^2) \quad (53)$$

With the electric energy, the state of charge of the battery pack is updated cycle by cycle.

$$SOC(i) = \frac{\left(E_{total} * \frac{SOC_{start}}{100} - Elt_{energy}(i)\right)}{E_{total}} * 100 \quad (54)$$

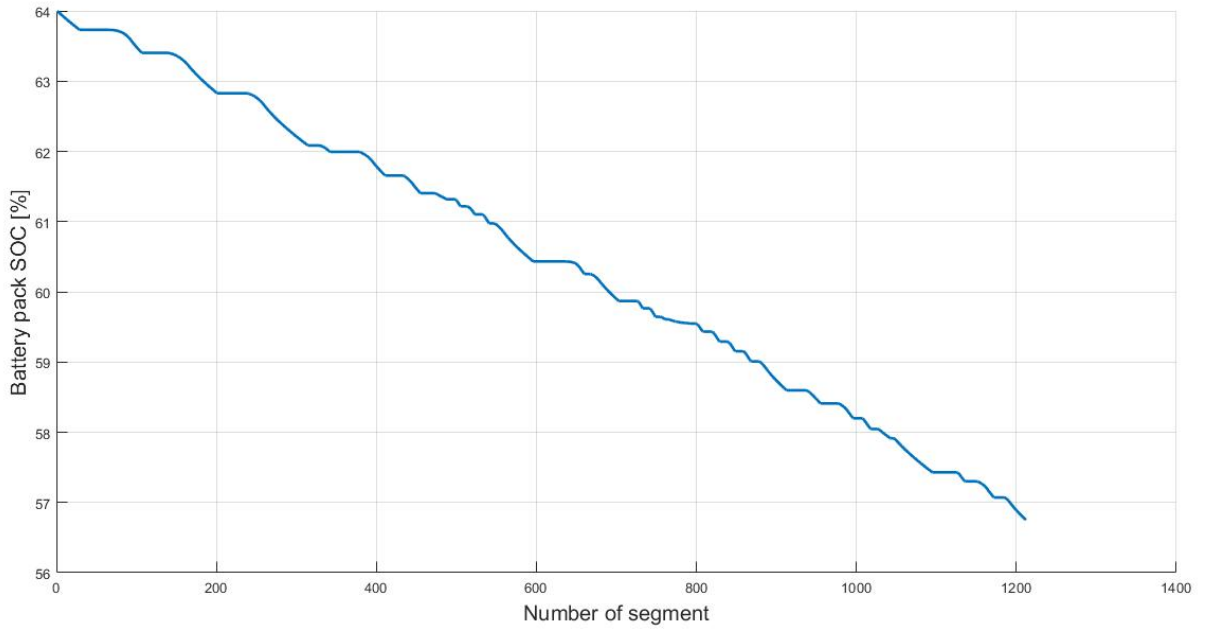


Figure 28: State of charge of the battery pack

Knowing the value of SOC, both internal resistance and open-circuit voltage are interpolated with this new value.

$$R_{in}(i) = f(SOC(i)) \quad (55)$$

$$OC_voltage(i) = f(SOC(i)) \quad (56)$$

Finally, this second loop ends with the evaluation of voltage and current using the battery pack model.

$$[V_{BP}(i), i_{BP}(i)] = f(P_{elt}, OC_voltage(i), R_{in}(i)) \quad (57)$$

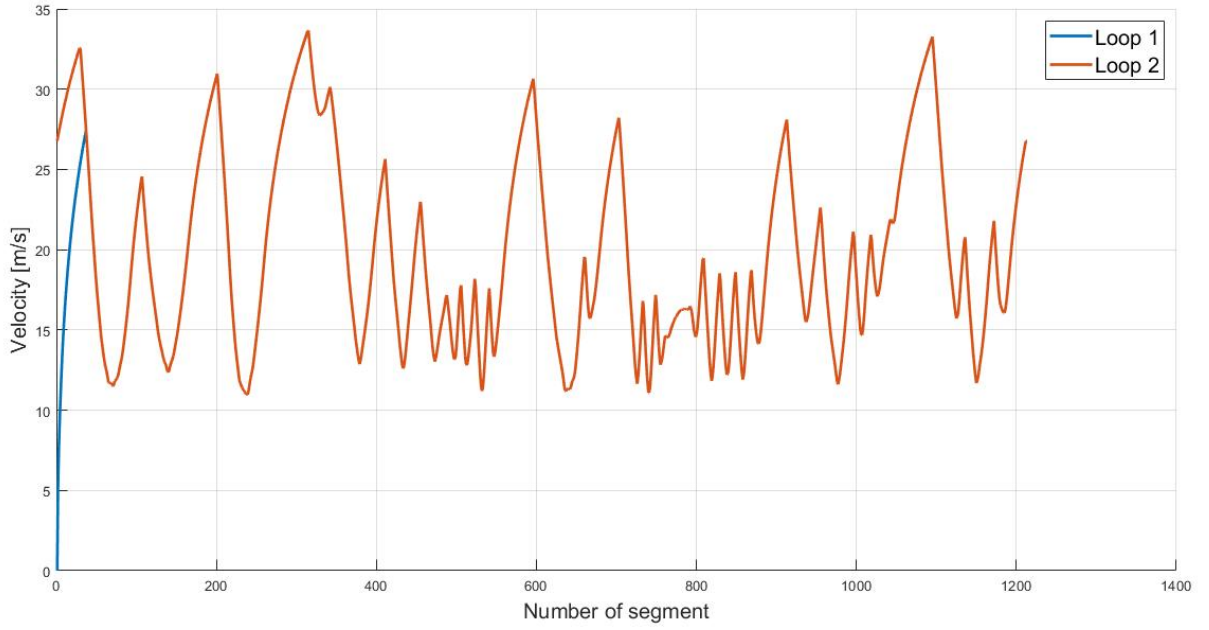


Figure 29: Comparison between first and second loop of profile 3

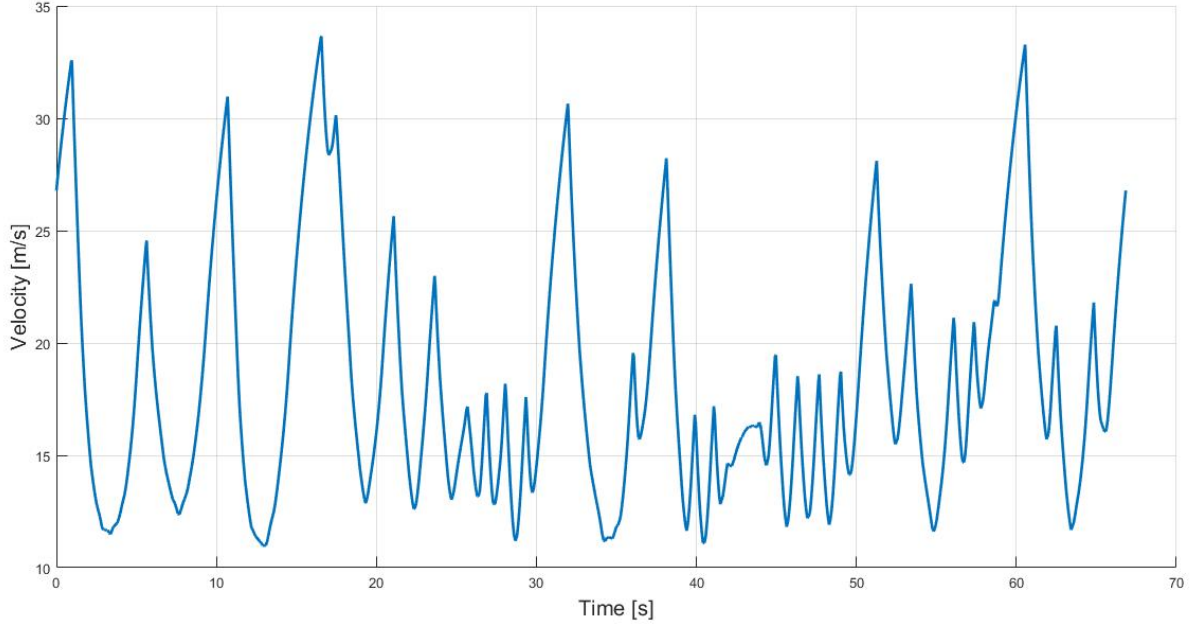


Figure 30: Example of final velocity profile

4.3.5 Evaluation of auxiliary data

The last section of the script is used to compute auxiliary data that was not strictly necessary in the previous steps and to check the eventual presence of bugs that allow the vehicle to exit from the GGV diagram.

The first quantity that is computed is the resulting lateral acceleration Ay .

$$Ay_{profile3}(i) = \frac{V_{profile3}(i)^2}{R(i)} \quad (58)$$

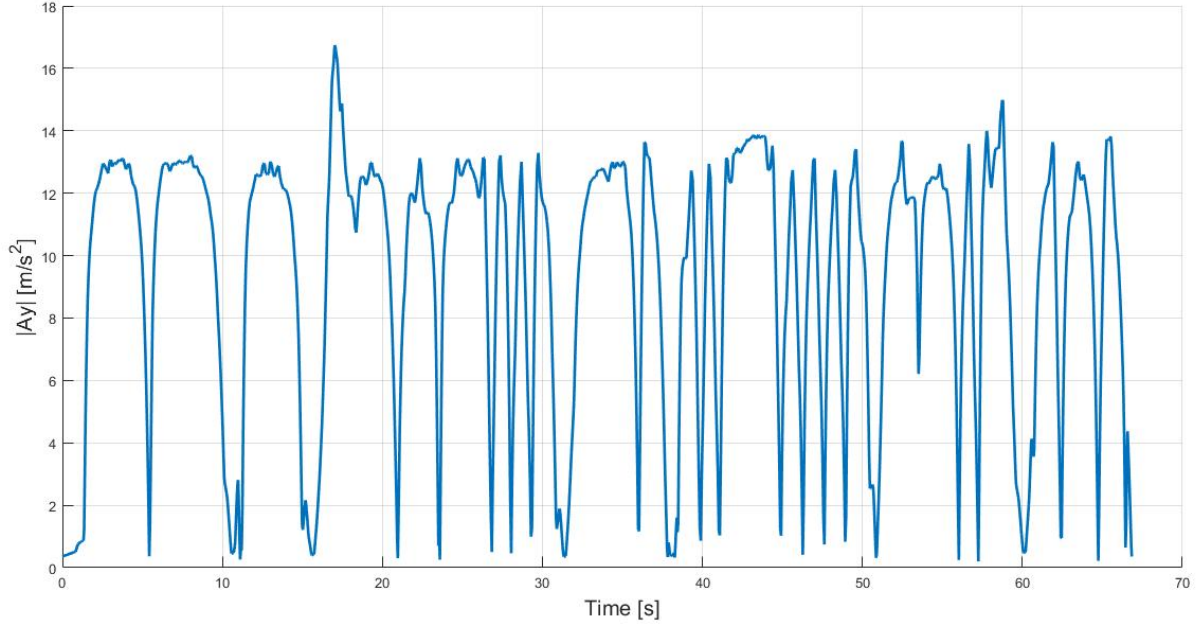


Figure 31: Absolute value of the lateral accelerations

Other parameters that are evaluated are the total laptime and the total covered distance.

$$Laptime(i) = Laptime(i - 1) + dt(i) \quad (59)$$

$$Distance(i) = Distance(i - 1) + ds(i) \quad (60)$$

The mechanical energy delivered by motors can also be useful, especially if compared to the electrical one, to underline the differences due to efficiencies.

$$Mech_{energy}(i) = Mech_{energy}(i - 1) + dt(i) * P_{mech}(i - 1) \quad (61)$$

If more data is necessary and needs to be evaluated, in this part of the script, it is possible to add lines of code and compute the requested quantities without modifying the structure of the simulator solver.

The very last part of the script is used to check if some errors in the evaluation of the speed profile are present. The control of the eventual presence of inaccuracies is done by interpolating the GGV function with the lateral acceleration and velocity. If the result is greater or equal to the longitudinal acceleration, the simulation is acceptable. If not, the script plots an error message in the command window.

$$Ax_{profile3}(i) \leq Ax(Ay_{profile3}(i), V_{profile3}(i)) \quad (62)$$

The simulator code ends with a series of plots, which are useful to display the results of the simulation.

5 Yaw moment diagrams

5.1 Introduction

The analysis with yaw moment diagrams is a tool that allows studying the vehicle dynamics behavior of a car. The concept of this method was originally introduced by Bill Milliken in 1952 and developed through the following years. The goal of these diagrams is to find the overall lateral force or lateral acceleration and the corresponding yaw moment that the vehicle can develop under different conditions.

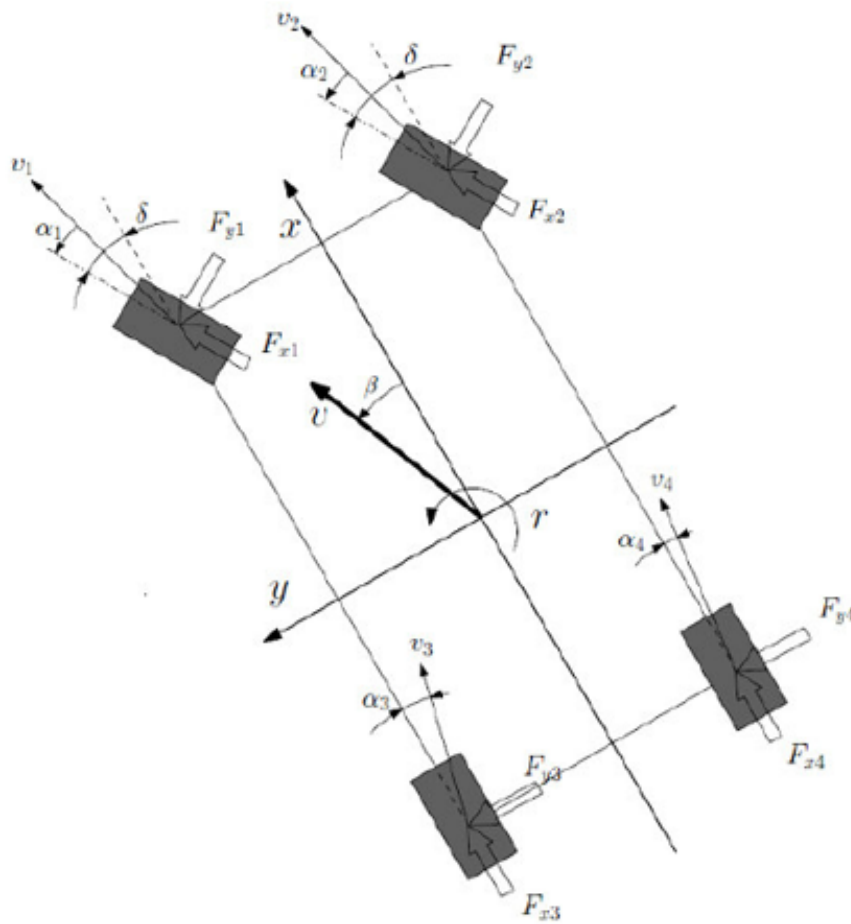


Figure 32: Bi-dimensional representation of the vehicle with characteristic angles

The simplest approach explained in Milliken & Milliken Race car vehicle dynamics is the so-called “constrained testing”. The vehicle is placed on a flat, moving belt and it is constrained to the ground by three steel wire ropes, two to fix the lateral position of the vehicle and one to constrain the longitudinal one. Inclining the vehicle with respect to the longitudinal direction it is possible to impose a sideslip angle of the vehicle β , while the steering effect is simulated by locking the wheels at an angle δ .

The diagram is created by testing the vehicle at a fixed belt speed and by trying different combinations of sideslip and steering angle. These combinations set different sideslip angles α_{front} and α_{rear} on tires, and, plotting the measured Mz and Fy , the MMM diagram is obtained. The diagrams usually are completed with iso-lines, which connect the points with equal values of β and δ .

$$\alpha_{front} = \beta + \delta \quad (63)$$

$$\alpha_{rear} = \beta \quad (64)$$

The key problem of the constrained testing is that the effect that the yaw rate has on the sideslip angles of tires is not considered.

Nowadays, with the availability of computational power from PCs, these diagrams are completely obtained through simulations of the vehicle at constant velocity or at a constant radius. With this approach, the effect of the yaw rate is considered and accurate results can be achieved.

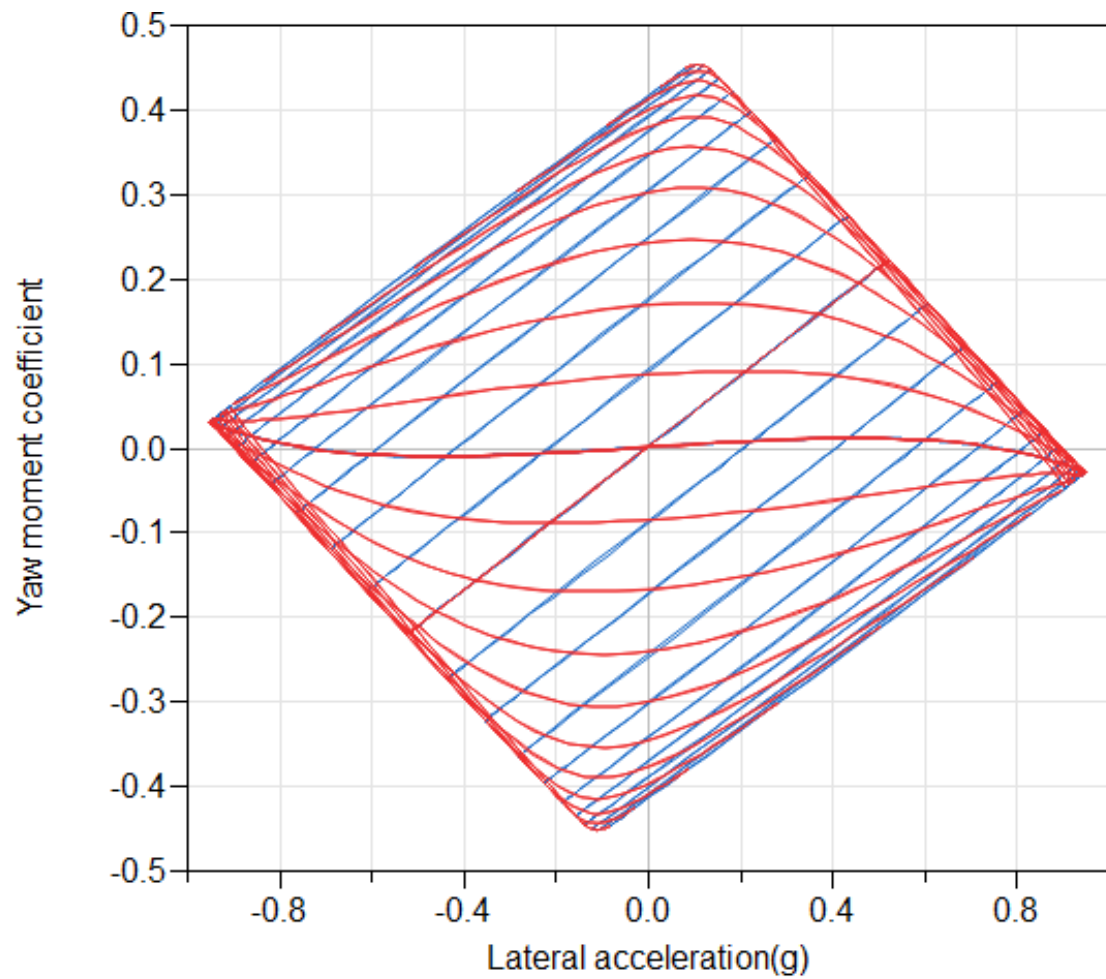


Figure 33: Typical shape of a yaw moment diagram

The tool can be very helpful in the design phase, because of the amount of information that is displayed within the plot. Usually, four different quantities are monitored with this approach:

1. **Stability:** It is measured as the rate of change of yaw moment with respect to the change of sideslip angle. The lower this indicator is, the higher the stability of the car. Mathematically, it is defined as the derivative of the yaw moment with respect to the sideslip angle.

$$Stability = \frac{dMz}{d\beta} \quad (65)$$

2. **Control:** It is evaluated as the rate of change of the yaw moment with respect to the change of steering angle. The higher this indicator is, the higher the controllability of the car. Similarly to stability, it can be defined as a derivative.

$$Controllability = \frac{dMz}{d\delta} \quad (66)$$

3. **Grip:** The grip indicator is related to the limit cornering behavior. It is defined as the maximum lateral acceleration that the vehicle can develop, even if not in steady-state.
4. **Balance:** The balance is related to the limit cornering behavior of the vehicle and is associated to the position of the peak acceleration with respect to the zero yaw moment line ($Mz = 0$). Assuming to evaluate the diagram using the ISO vehicle axis system, and for positive lateral acceleration (left-hand turn), if the peak acceleration is above the zero moment line, then the vehicle is defined as oversteering. It means that the car can develop that maximum acceleration with an associated oversteering moment. On the opposite, if the maximum lateral acceleration is below that line, the vehicle is said to be understeering.

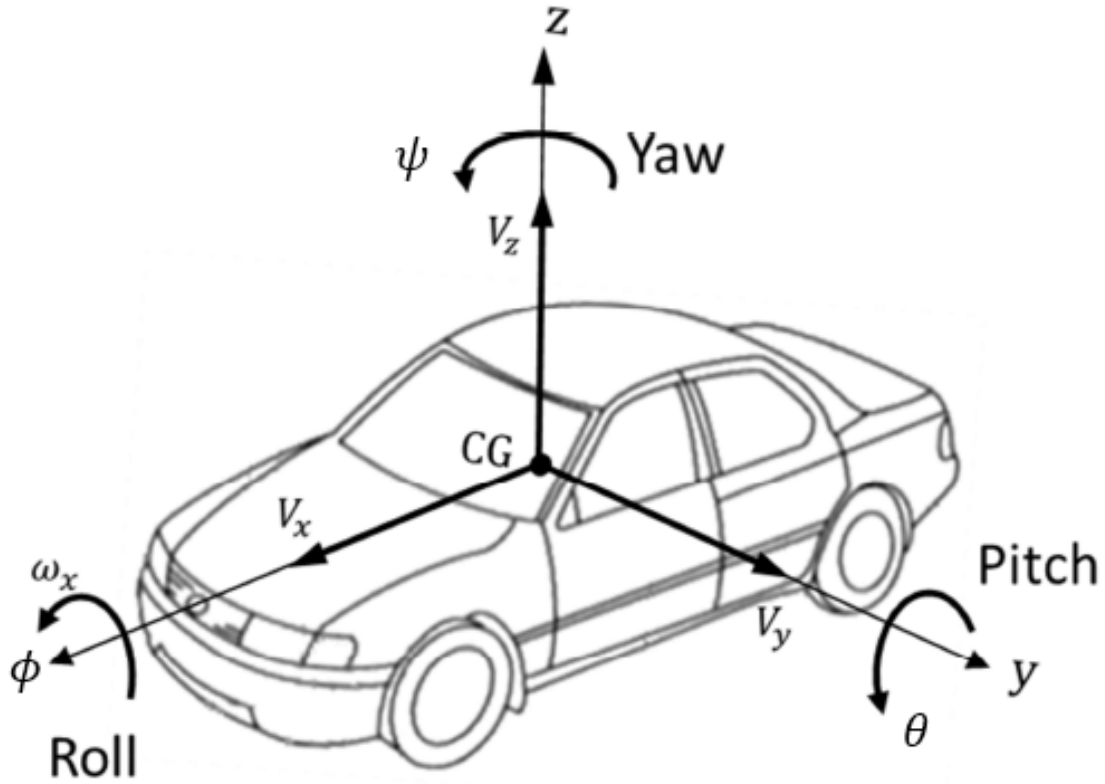


Figure 34: ISO vehicle axis system

5.2 Evaluation of constant velocity yaw moment diagrams

5.2.1 Structure of the script

The script developed to create the constant velocity yaw moment diagram (also indicated with YMD) is written in MATLAB[®] and starts with the definition of the inputs. The following quantities are required:

1. Tires: forces and moments are considered using a Pacejka model (.TIR file);
2. Suspension: the stiffnesses of the suspension are taken into account to define the roll stiffness between axles and the lateral load transfer distribution;
3. Aerodynamics: the aerodynamic effect is considered with downforce, drag, aerobalance, but also with the side aerodynamic force Fy_{aero} , and the aero-

dynamic yaw moment $M_{z_{aero}}$;

4. Body: the body of the vehicle is considered with the mass, the weight repartition, and the height of the center of gravity;
5. Steering: the steering geometry is considered with the Ackermann percentage;

The inputs defined by the user are:

1. Velocity of the vehicle for the simulation;
2. Range of sideslip angles of the vehicle considered;
3. Range of steering angles of the vehicle considered;
4. Longitudinal force required by each tire due to brakes or motors;

Once the inputs are defined, the script starts with the evaluation of the diagram. The target is to compute yaw moment, lateral and longitudinal accelerations, knowing the steering angle and sideslip angle of the vehicle.

Because of the dependence of the sideslip angle of each tire on the yaw rate, the entire process is managed with an iterative procedure.

Also, for what concerns the evaluation of the load transfer equilibrium of the vehicle, an iterative procedure is necessary to obtain a result. This is because of the over-constraints due to the presence of 4 wheels and of the non-linearity of friction coefficients in the tire model. The resulting script is composed of a *for* loop used to update, for every cycle, the values of δ and β , and two nested iterative cycles.

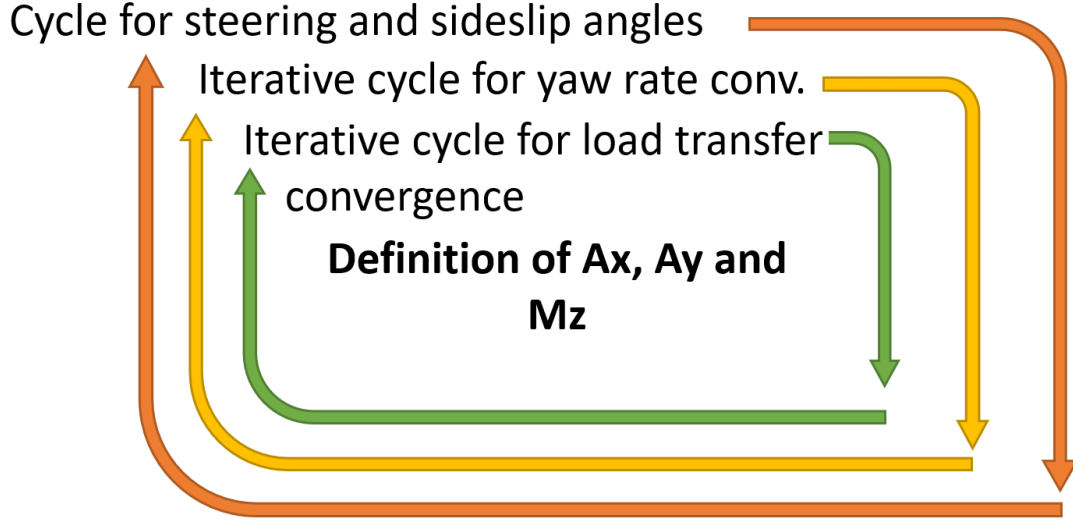


Figure 35: Structure of the script

As soon as β and δ are defined, the script enters the iterative cycle for the yaw rate, assuming zero radians per second as a first value.

$$r_{test} = 0 \quad (67)$$

With these three quantities, it is possible to compute the sideslip angle experienced by each tire.

$$\alpha_{front} = \text{atan} \left(\frac{V * \cos(\beta) + \dot{r} * a}{V * \sin(\beta) \pm \dot{r} * \frac{t}{2}} \right) - \delta \quad (68)$$

$$\alpha_{rear} = \text{atan} \left(\frac{V * \cos(\beta) - \dot{r} * b}{V * \sin(\beta) \pm \dot{r} * \frac{t}{2}} \right) \quad (69)$$

These angles are sent to the internal loop, used for the load transfer convergence.

Here the second iterative cycle starts. As a first assumption, the vertical force on the tires are equal to the static one with the addition of aerodynamic effects.

Having assumed to know the four vertical forces, the tire property file is used to determine if the longitudinal forces requested are available (if not, the peak value is taken) and the residual lateral forces are also computed with these sideslip angles. The output from the tire model is constituted of four F_y and four F_x , which are then applied to the vehicle, causing a load transfer and a variation on the vertical forces of tires F_z .

Now, these vertical forces are compared to the ones assumed before and, if the error is lower than a threshold, for example of 1%, the assumption is valid. However, if the error is bigger, the loop restarts with another cycle, considering as input the newer values of F_z .

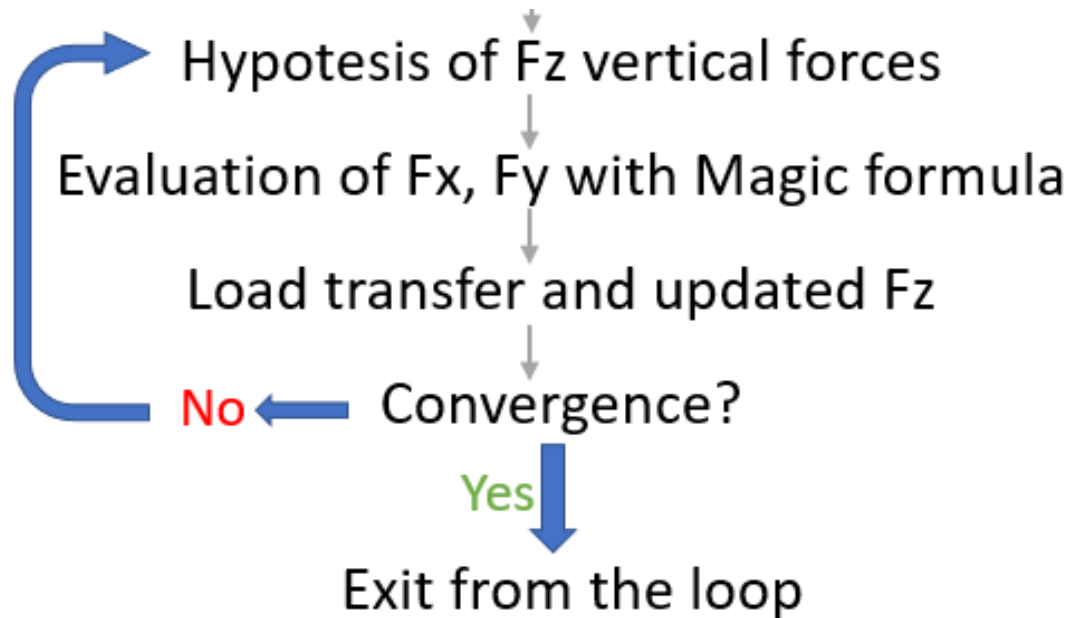


Figure 36: Structure of the load transfer convergence loop

As soon as the exit condition is satisfied, the loop is concluded with the evaluation of the corresponding lateral acceleration A_y and yaw moment M_z , using the longitudinal forces, lateral forces, and self-aligning torques of tires, together with the aerodynamic forces.

$$\left| 1 - \frac{Fz_i}{Fz_{i-1}} \right| \leq 0.01 \quad (70)$$

These values are sent back to the external loop, as they are necessary for the yaw rate convergence. With these outputs, the yaw rate has to be computed. The challenge, in this case, is that, in transient conditions, the yaw rate depends also on the derivative of the lateral velocity v . Since these yaw moment diagrams are developed at a constant speed, the yaw rate depends on the derivative of β .

$$Ay = V * r + \dot{v} \quad (71)$$

$$Ay = V * r + V * \dot{\beta} \quad (72)$$

To avoid this issue, the diagram is simulated under quasi steady-state conditions. This approximation is defined as a simplified transient condition, in which the vehicle can have unbalanced forces and moments, but the angular velocity of the sideslip angle $\dot{\beta}$ is assumed to be equal to zero.

$$r_{evaluated} = \frac{V}{R} = \frac{Ay}{V} \quad (73)$$

Comparing this result to the initial hypothesis r_{test} , if the error is small enough, the loop can be concluded, and the iteration is considered valid. If not, a new value of r_{test} has to be chosen until the convergence is reached.

$$\left| 1 - \frac{r_{test}}{r_{evaluated}} \right| \leq 0.01 \quad (74)$$

The choice of a good iterative procedure is fundamental to reach the result within

a reasonable computational time. In this script, the successive under-relaxation method is used to assume a new value of r_{test} , now defined with the name r_{n+1} .

$$r_{n+1} = r_{evaluated} * \omega_r + r_n * (1 - \omega_r) \quad (75)$$

The coefficient ω_r is known as the relaxation parameter and it defines the following assumption for the iterative cycle as the weighted average of the previously discarded assumption and its result.

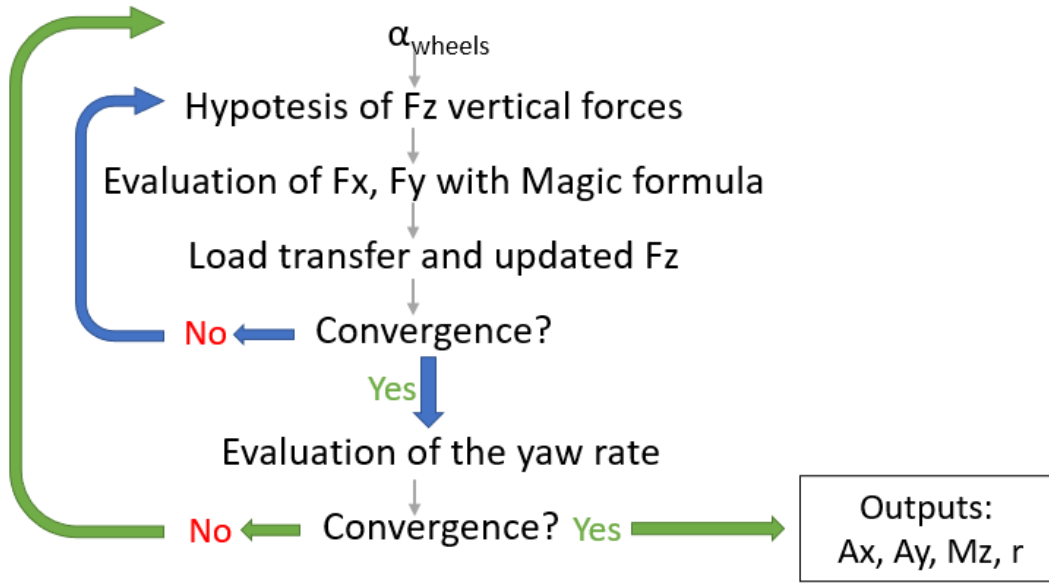


Figure 37: Structure of the two iterative loops

To get a rather fast convergence, a ω_r of 0.6/0.7 is used at high speed, while for lower velocities, this value can be also smaller than 0.1.

Finally, the outputs of the loop Ay and Mz are plotted in the yaw moment diagram. Once the figure is completed with all the points, the script connects through iso-lines all the points with equal steering angle and equal sideslip angle.

5.2.2 Analysis of the results

The following diagrams are evaluated with steering angles ranging from -15 to +15 degrees, equally spaced into 25 steps, and the sideslip angle β is simulated varying from -12 to +12 degrees, discretized into 13 steps.

Blue lines identify constant steering angle, while green lines represent constant sideslip angle.

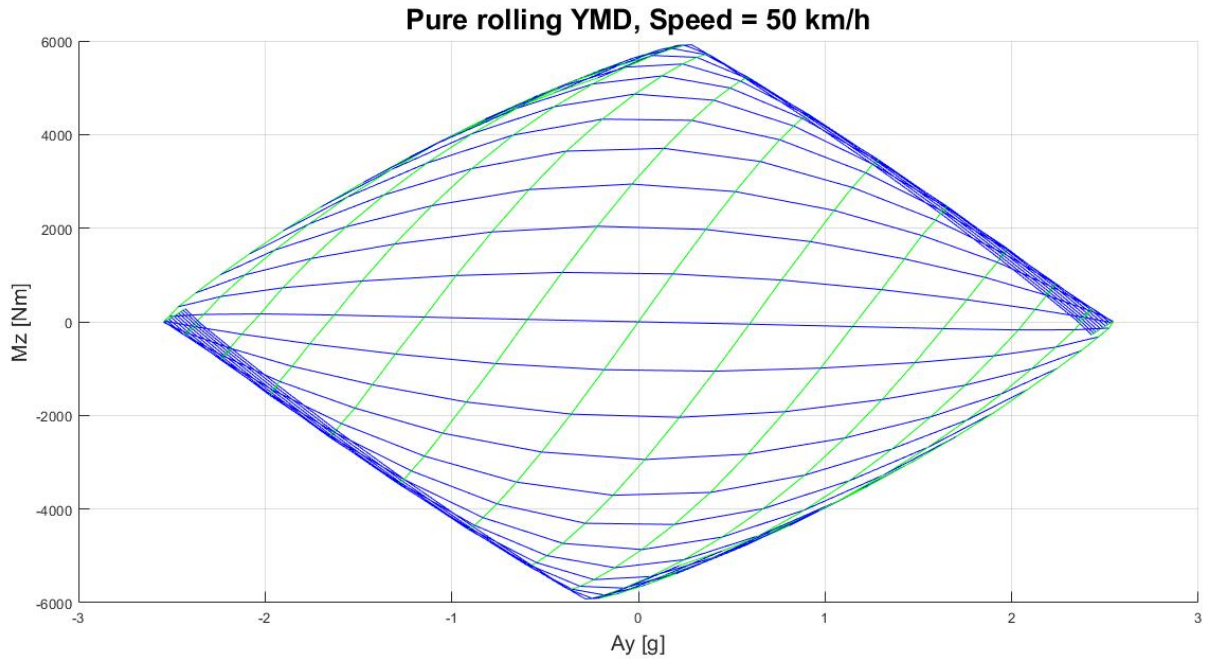


Figure 38: Pure rolling YMD evaluated at 50 km/h

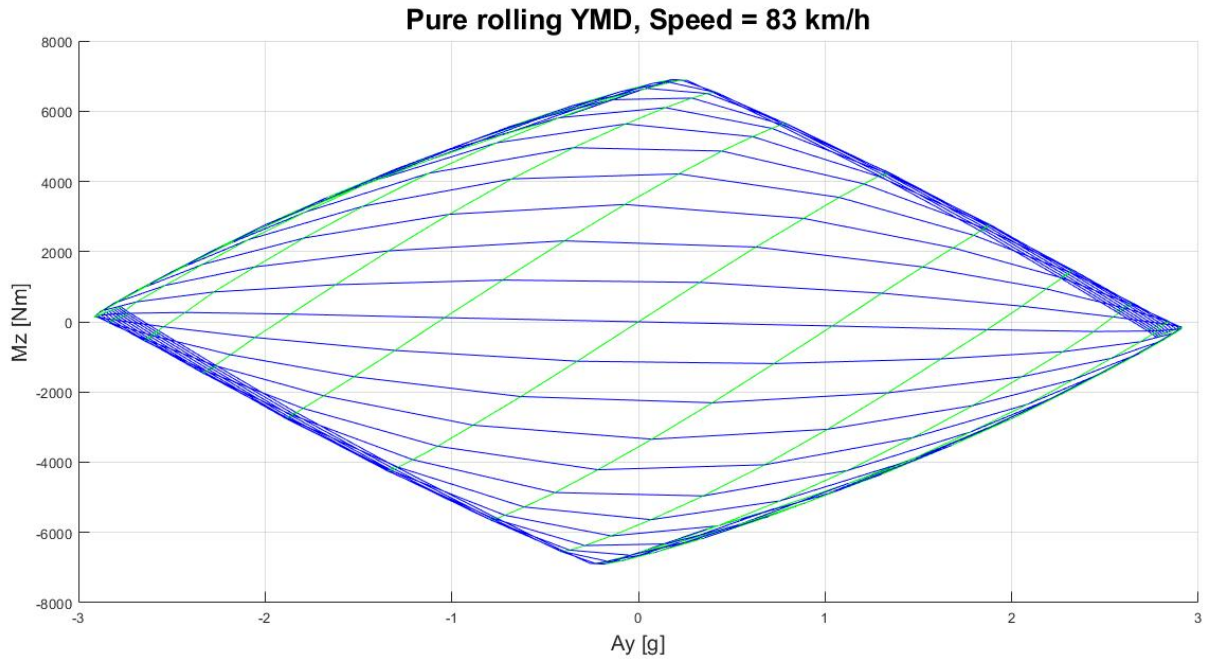


Figure 39: Pure rolling YMD evaluated at 83 km/h

Comparing the two diagrams obtained in pure rolling conditions, which means that the tires are not applying any longitudinal forces, it is noticeable that the behavior at maximum lateral acceleration changes from a slightly oversteering characteristic to an understeering one. At the same time, because of the aerodynamic effects, the magnitude of the peak lateral acceleration A_y is increased.

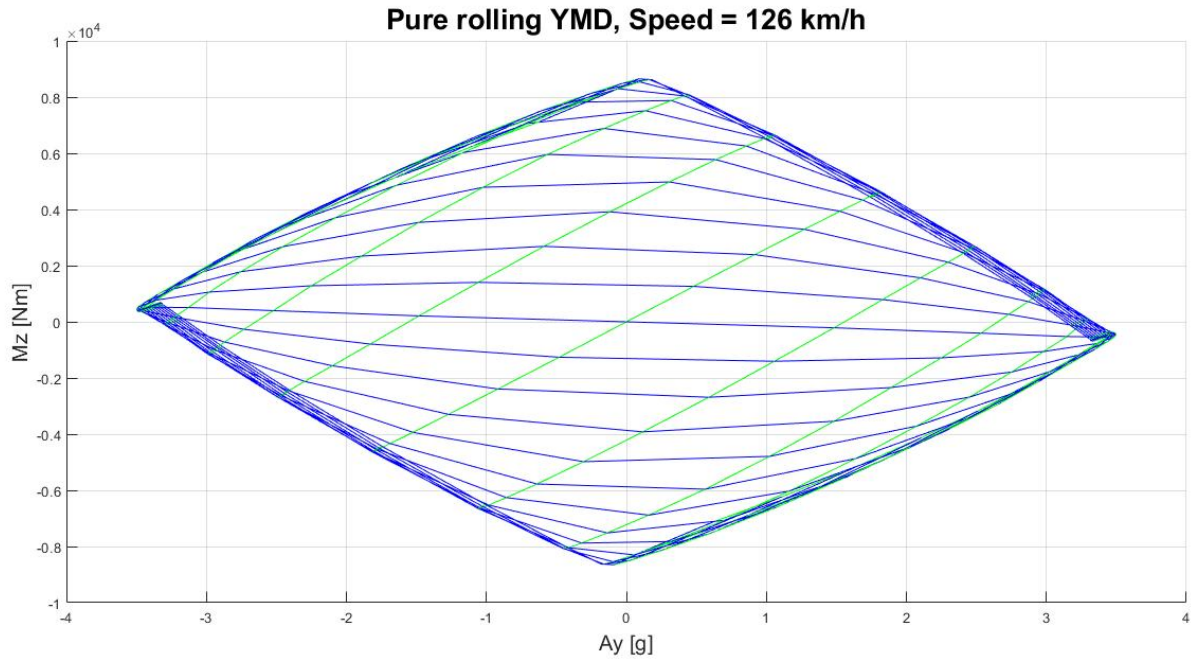


Figure 40: Pure rolling YMD evaluated at 126 km/h

At higher speeds, the trend is the same, so that the understeering behavior at peak lateral acceleration is increased because of the aerodynamic load repartition shifted towards the rear.

For what concerns the handling of the vehicle, the slope of the steering iso-lines is almost constant, meaning that the stability index is almost equal in this velocity range.

On the opposite, analyzing the slope of sideslip angle β iso-lines, the controllability is reduced as the velocity of the vehicle increases.

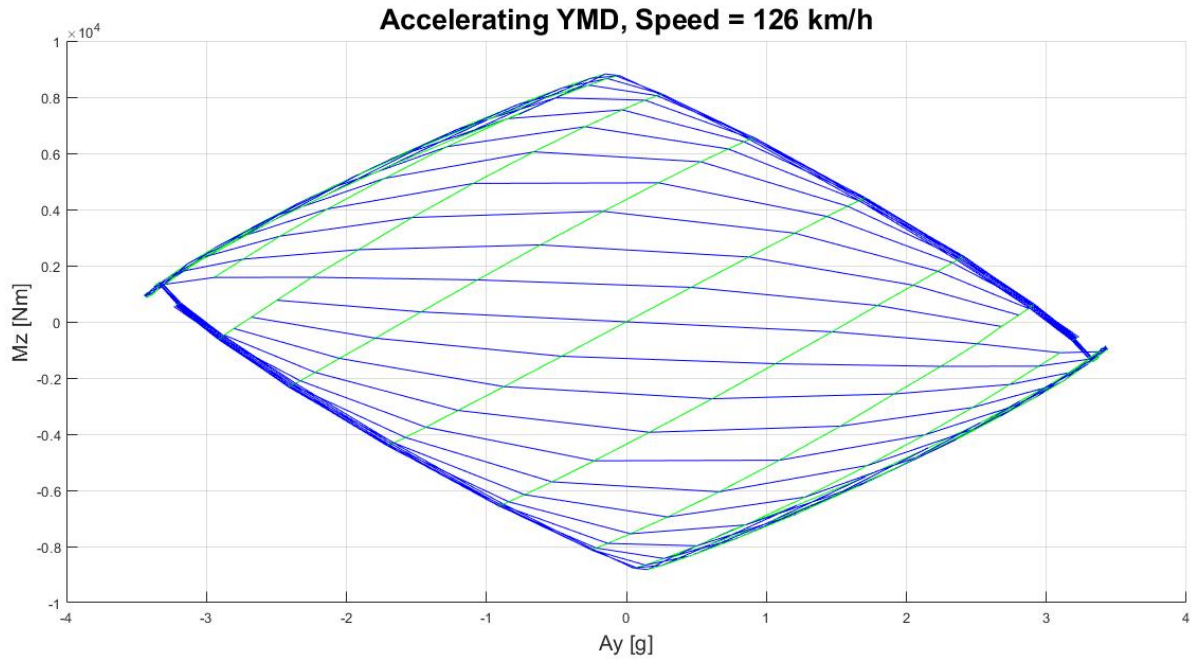


Figure 41: YMD evaluated during an acceleration phase at 126 km/h

The yaw moment diagram above is evaluated with the vehicle in accelerating condition, and it is noticeable the absence of some points around the limit lateral conditions, because of the unfeasibility of this condition at such a high speed with these angles.

The last diagram displays the braking phase, and similar to the accelerating one, the extremities are missing due to the impossibility of reaching the equilibrium in those conditions.

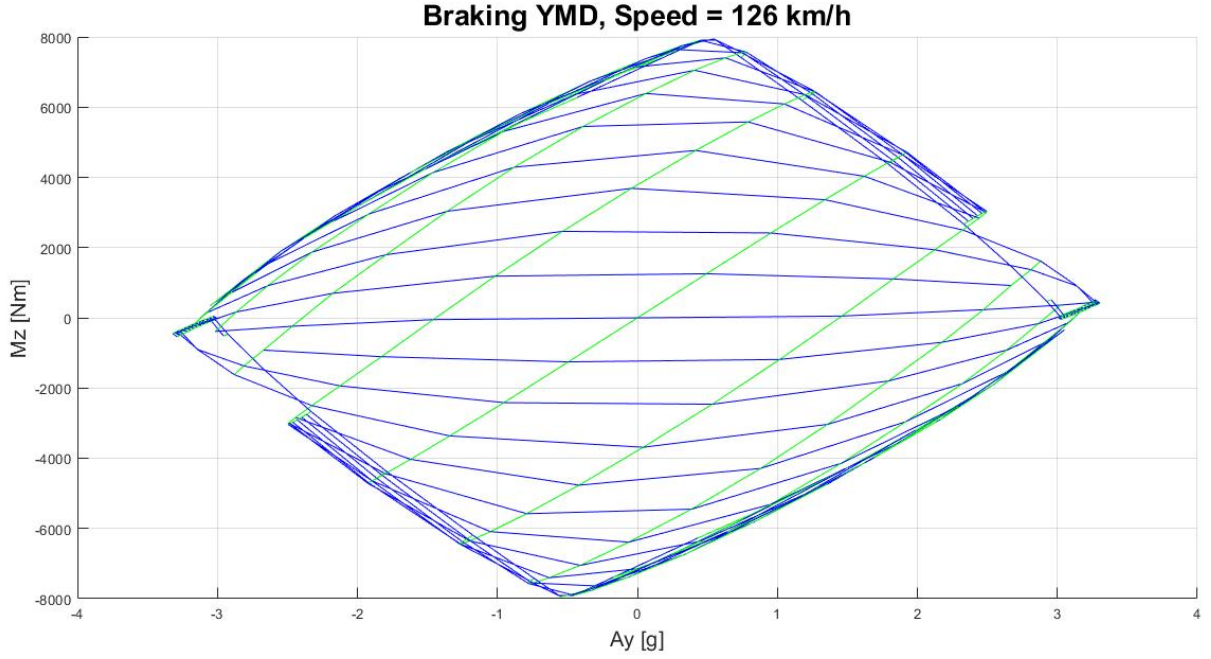


Figure 42: YMD evaluated during a braking phase at 126 km/h

5.3 GGv diagram through yaw moment diagrams

The purpose of this part of the script is to define the performance envelope of the vehicle using the yaw moment diagram method.

This approach allows modeling the vehicle in a more accurate way, considering many important phenomena that influence the vehicle dynamics, such as load transfers, tire models, and lateral aerodynamics.

The intention is to create a defined number of yaw moment diagrams for different longitudinal forces requested on the tires. These forces depend directly on the pedal inputs applied by the driver. Basically, a set of pedal combinations N_{pedals} are defined, from the full-braking condition to the full-throttle condition. For each one of them, the corresponding longitudinal force Fx requested from the tire is computed.

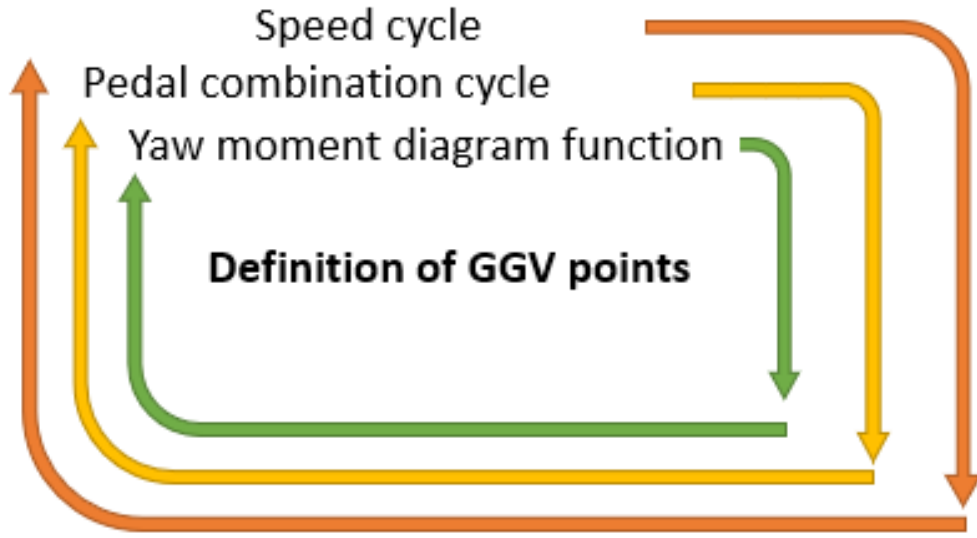


Figure 43: Approach used to determine the GGV diagram

This procedure is repeated for a user-defined number of velocities $N_{velocity}$, up to the maximum speed. If in each diagram the steering iso-lines and the β iso-lines are interpolated with a value of Mz equal to zero, a set of points in which the vehicle is in steady-state conditions is found.

Knowing the three coordinates, Ax , Ay , and V , the GGV diagram is defined.

5.3.1 Results

In this part, two previously computed GGV diagrams are presented, with a different number of discretized points and time required for the simulations.

5.3.2 Simulation 1

The first simulation is discretized with 12 levels of velocity and 11 values of pedal combinations, for a total time required for the simulation of around 1 hour and 30 minutes.

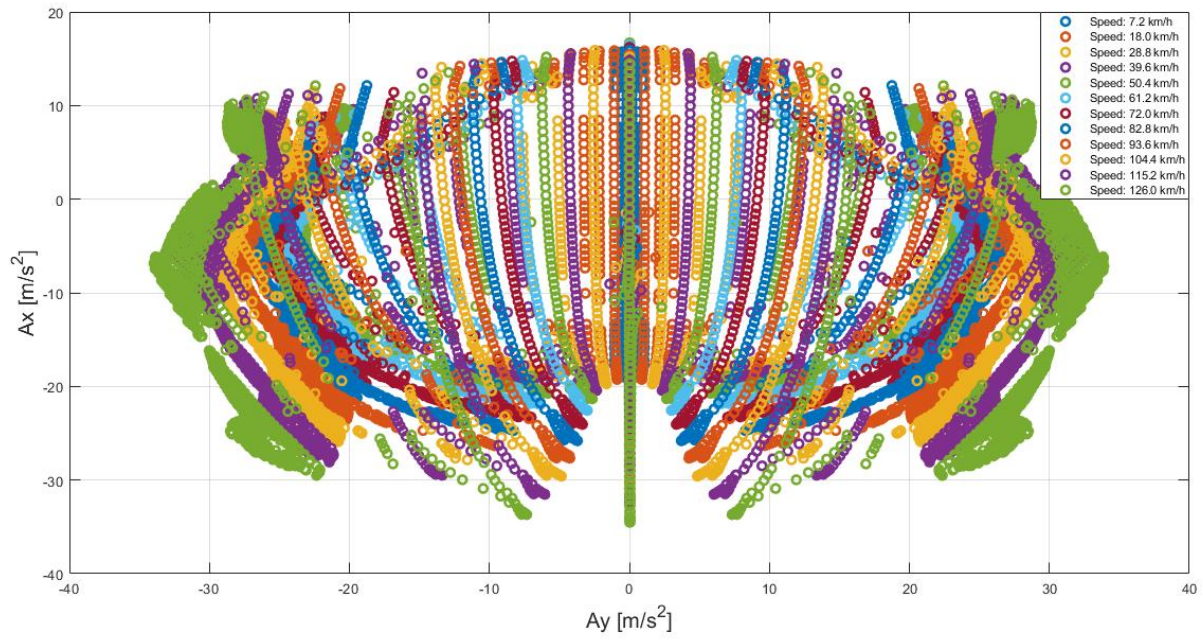


Figure 44: Top view of the GGv obtained with simulation 1

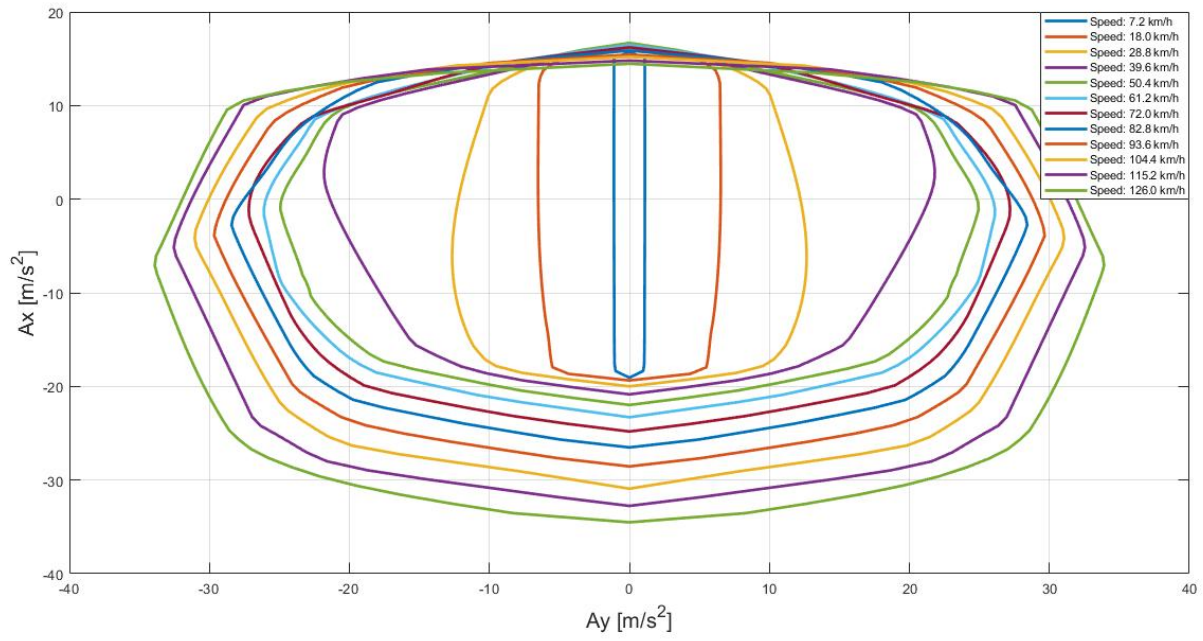


Figure 45: Level curves defining the GGv diagram obtained with simulation 1

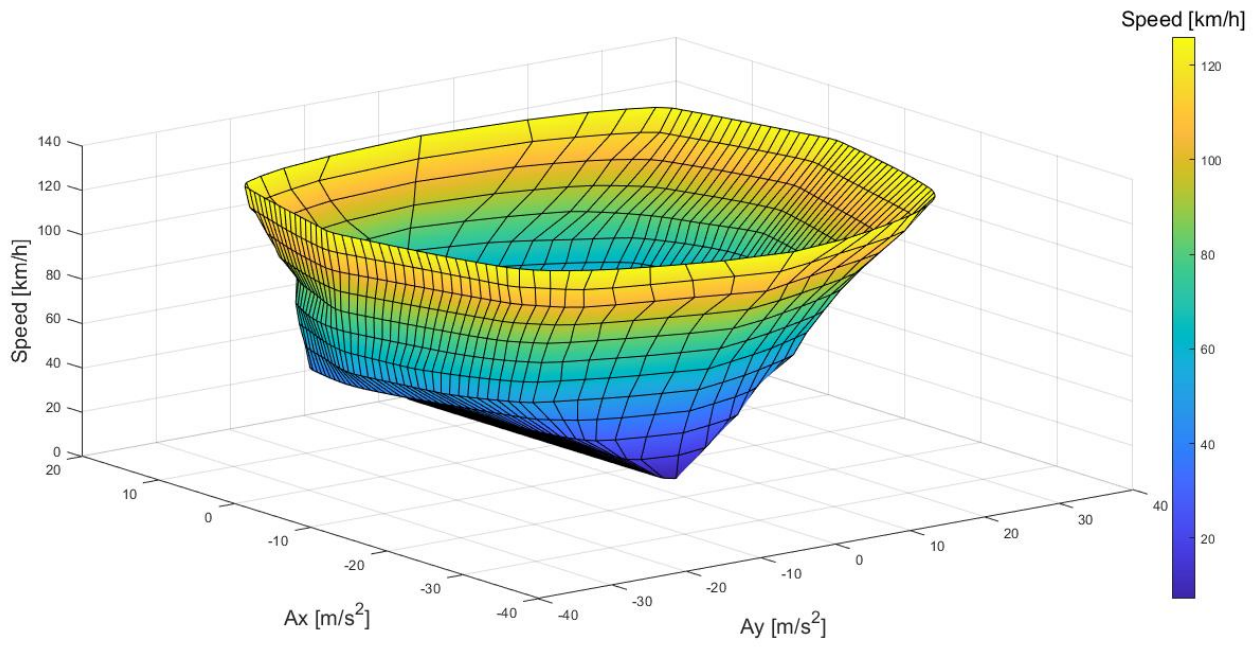


Figure 46: External surface of the GGV obtained with simulation 1

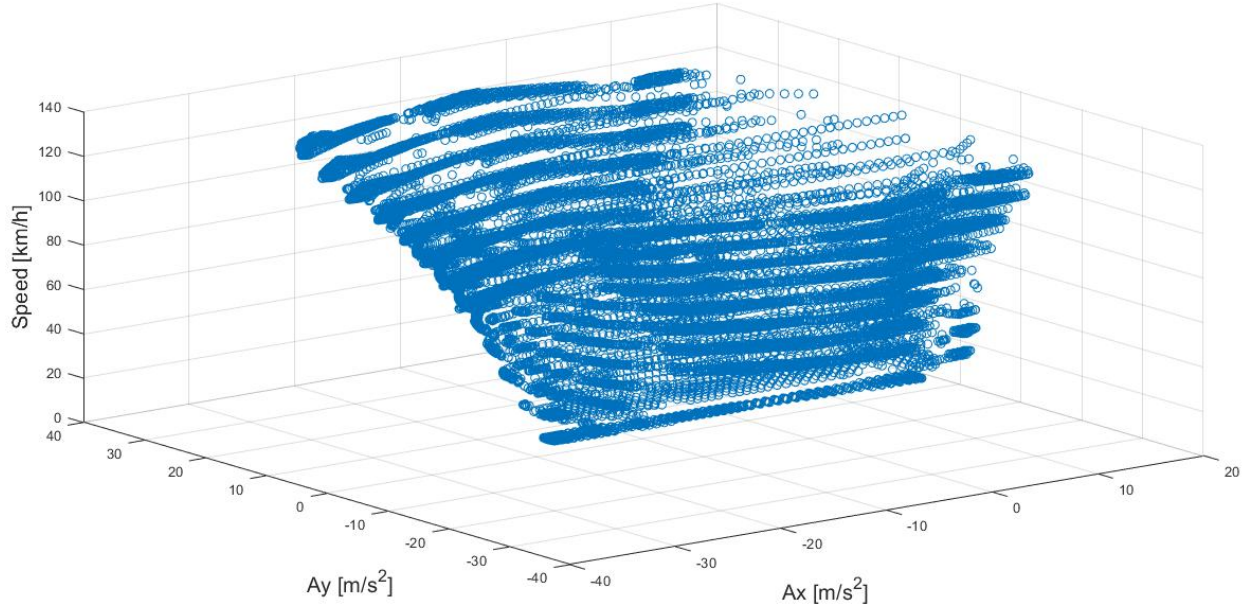


Figure 47: Cloud of points obtained from yaw moment diagrams in simulation 1

5.3.3 Simulation 2

The second simulation is discretized with 12 levels of velocity and 71 values of pedal combinations, for a total time required for the simulation of around 14 hours.

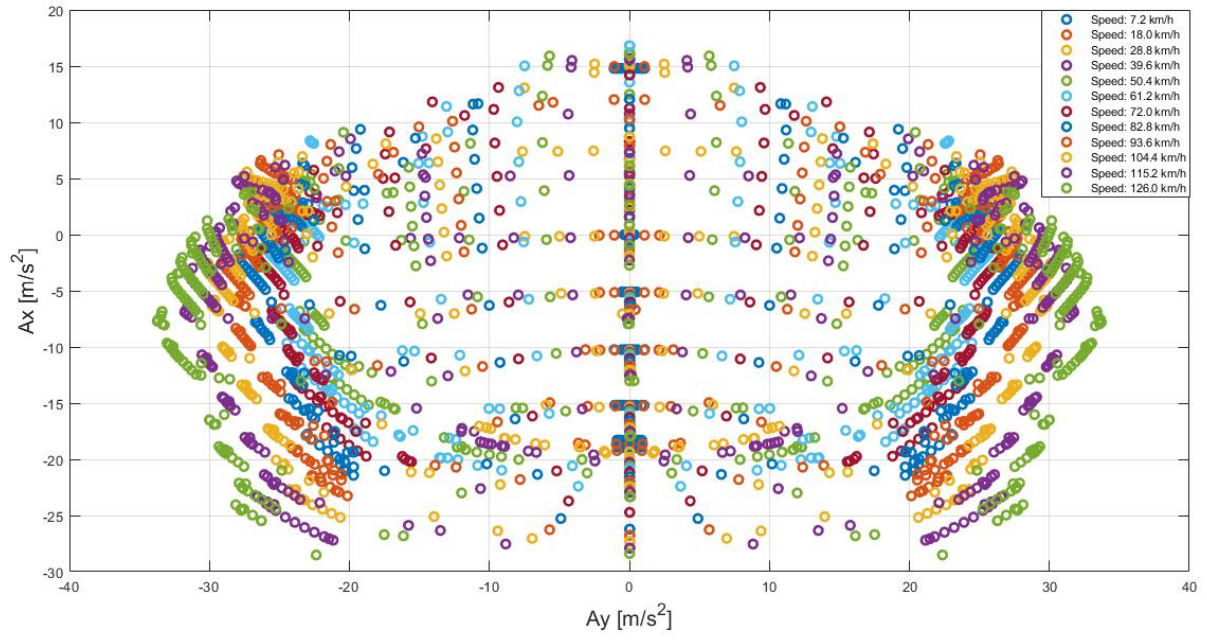


Figure 48: Top view of the GGV obtained with simulation 2

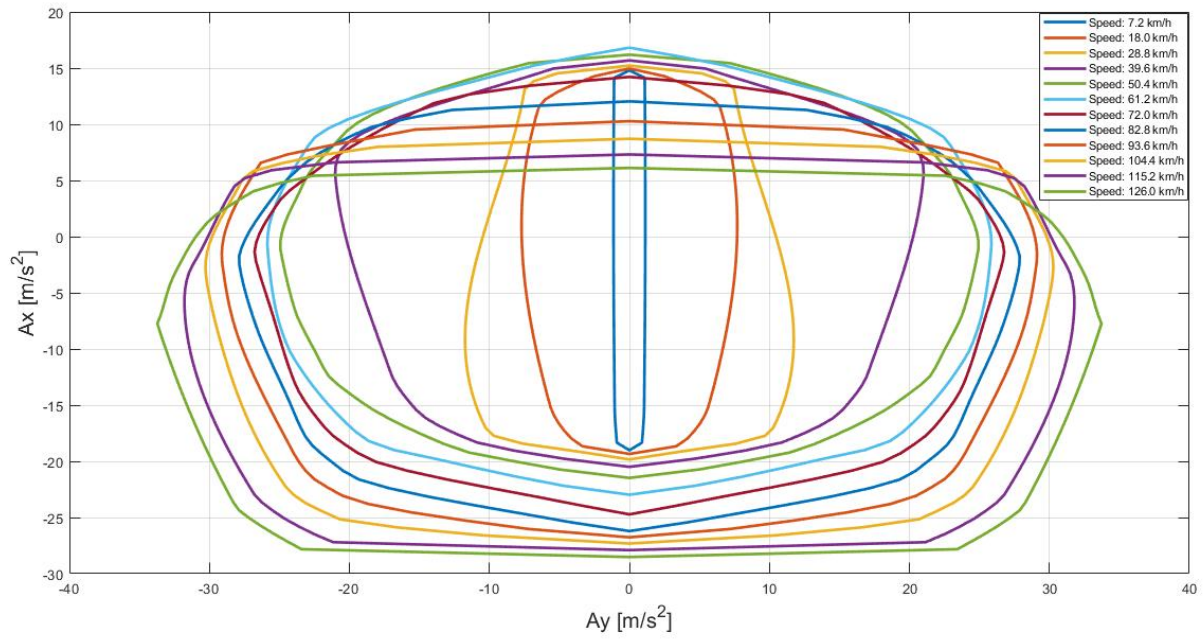


Figure 49: Level curves defining the GGv diagram obtained with simulation 2

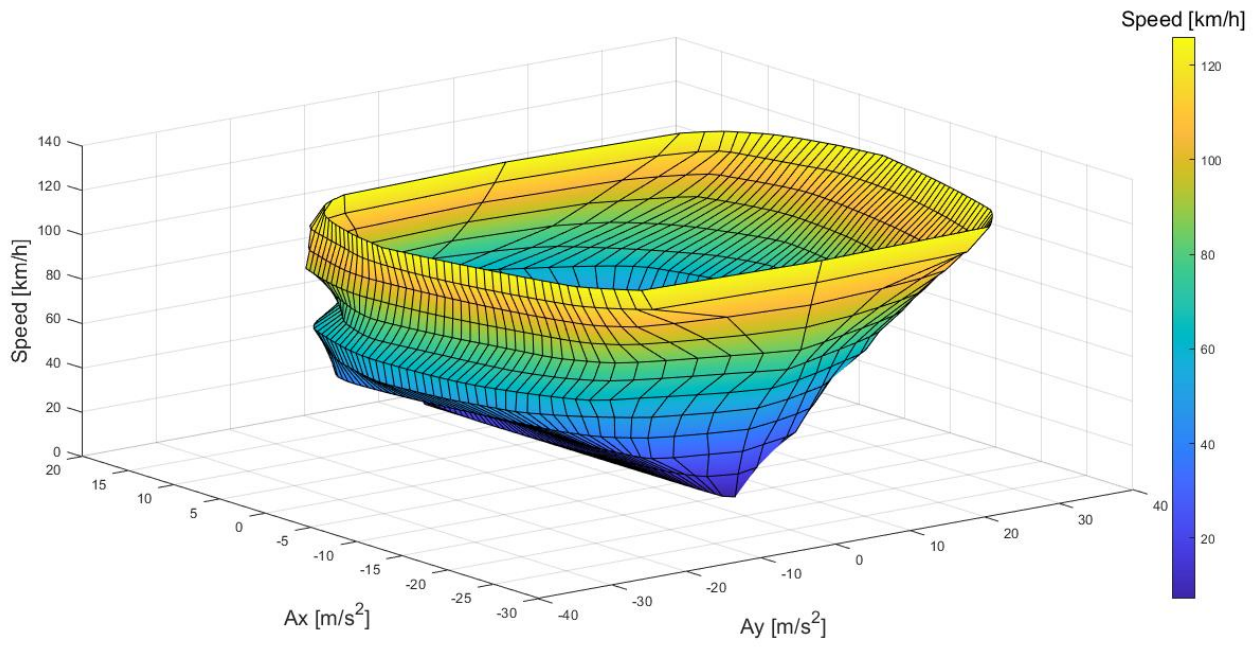


Figure 50: External surface of the GGV obtained with simulation 2

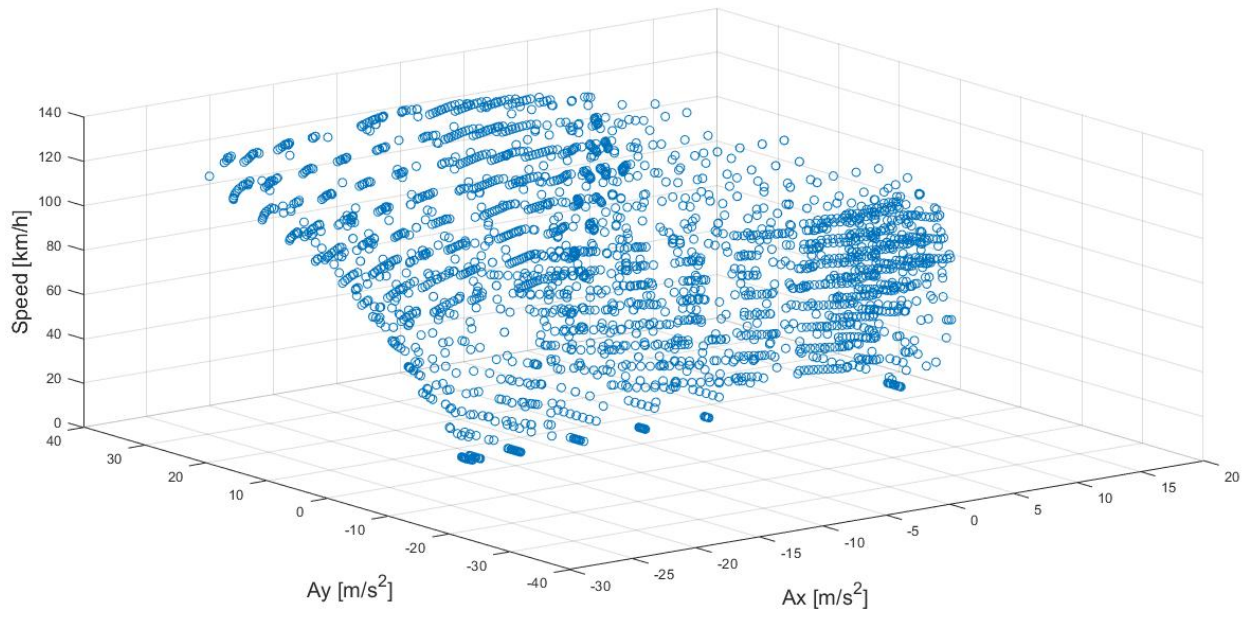


Figure 51: Cloud of points obtained from yaw moment diagrams in simulation 2

6 Validations

6.1 Point-mass lap time simulator

To validate the point-mass lap time simulator, the outputs of the simulator are compared to the logged data from the actual vehicle on track, and also to the outputs of a commercial transient simulator.

6.1.1 Comparison with the real car

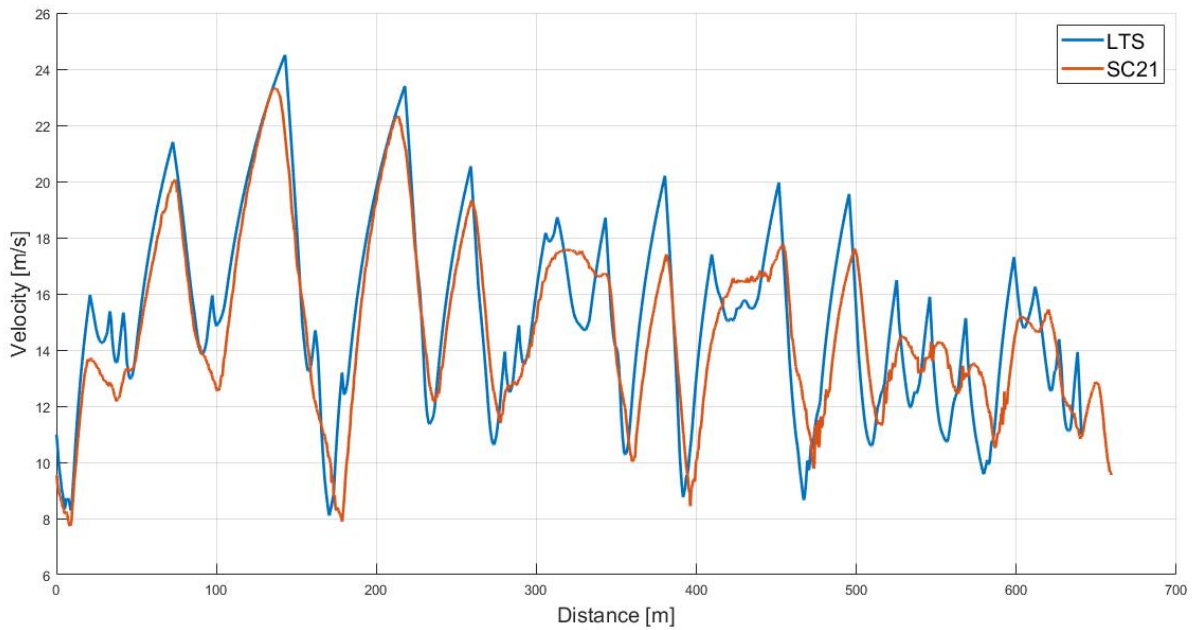


Figure 52: Velocity-distance plot comparison with the real car

The plot above displays the comparative analysis between the simulated speed profile and the logged one, as a function of distance. Since the driver is ideal, bringing the vehicle to its limits in any segment, and since the simulation approach used exploits a point-mass vehicle model, the simulated laptime is always faster than reality.

The real driver has freedom in selecting the fastest trajectory that allows the vehicle to keep a more constant turning radius and consequently a less varying speed. On the opposite, the simulator has a more efficient driver that exploits at a maximum extent the variation of turning radius. For the same reason also the two cumulative distances have some small differences.

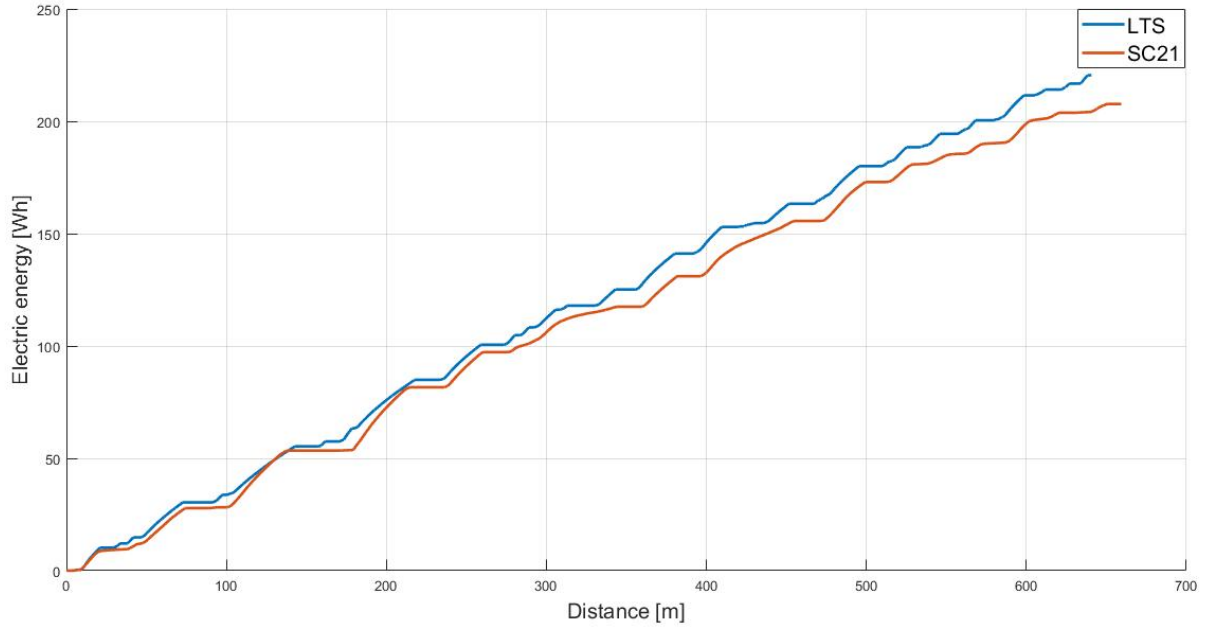


Figure 53: Electric energy-distance plot comparison with the real car

For what concerns the electric energy consumption, the two trends are quite similar, with the simulation that has a higher consumption because of the lower laptime and higher brake and accelerator requests from the driver.

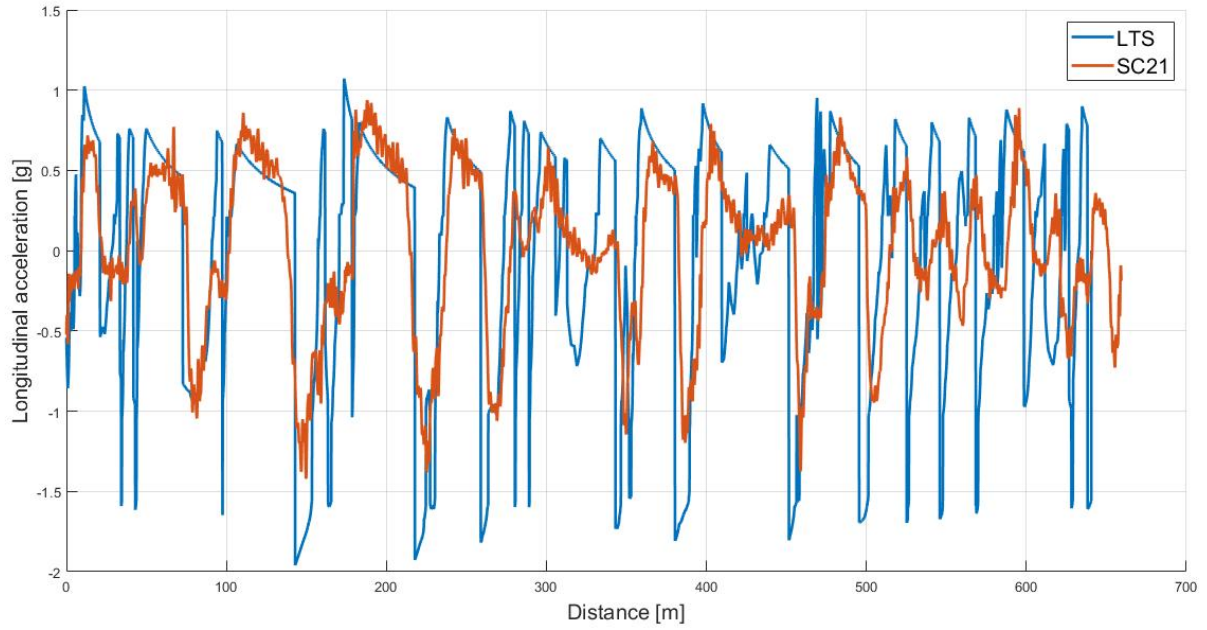


Figure 54: Longitudinal acceleration-distance plot comparison with the real car

Regarding the longitudinal and lateral accelerations, the trends are comparable to the logged data, especially in high-speed turns, while the difference between the plots is larger in the miscellaneous turns sections, because of the differences in the trajectories, as explained above.

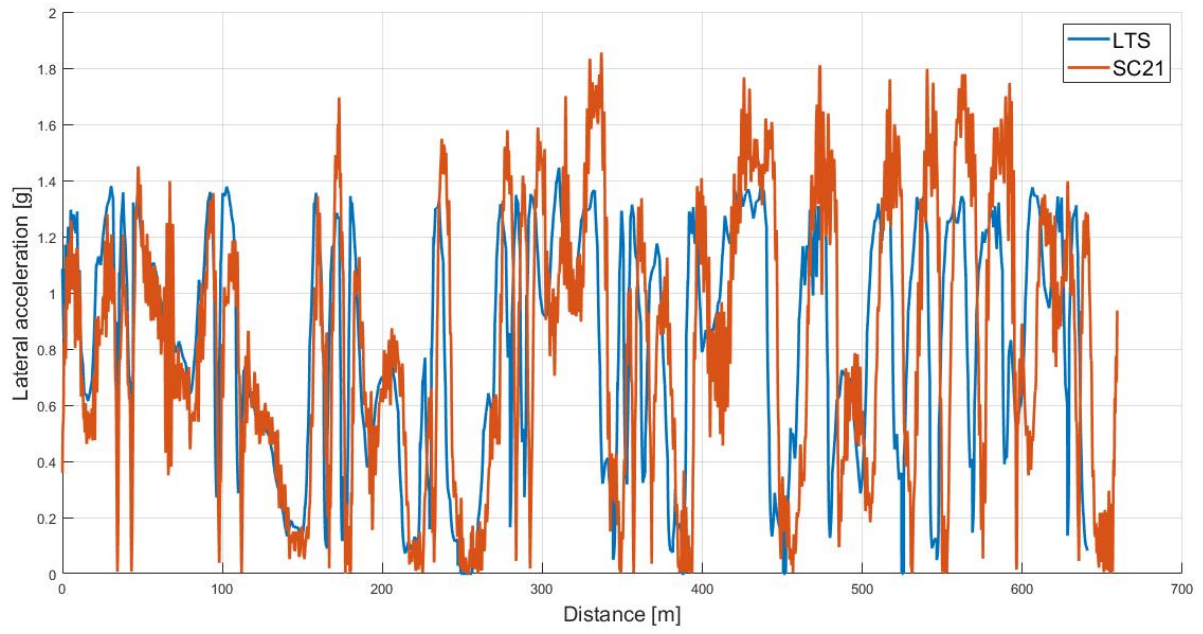


Figure 55: Lateral acceleration-distance plot comparison with the real car

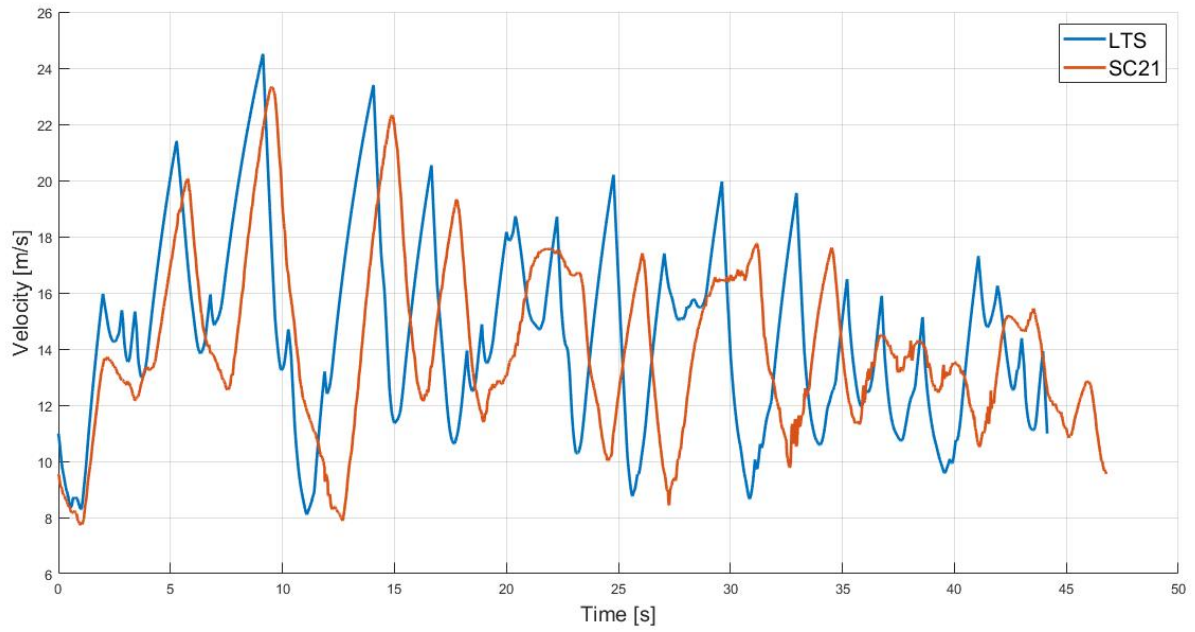


Figure 56: Velocity-time plot comparison with the real car

These two final plots represent the same data displayed previously but, in this case, speed and electrical energy are plotted as a function of time instead of space.

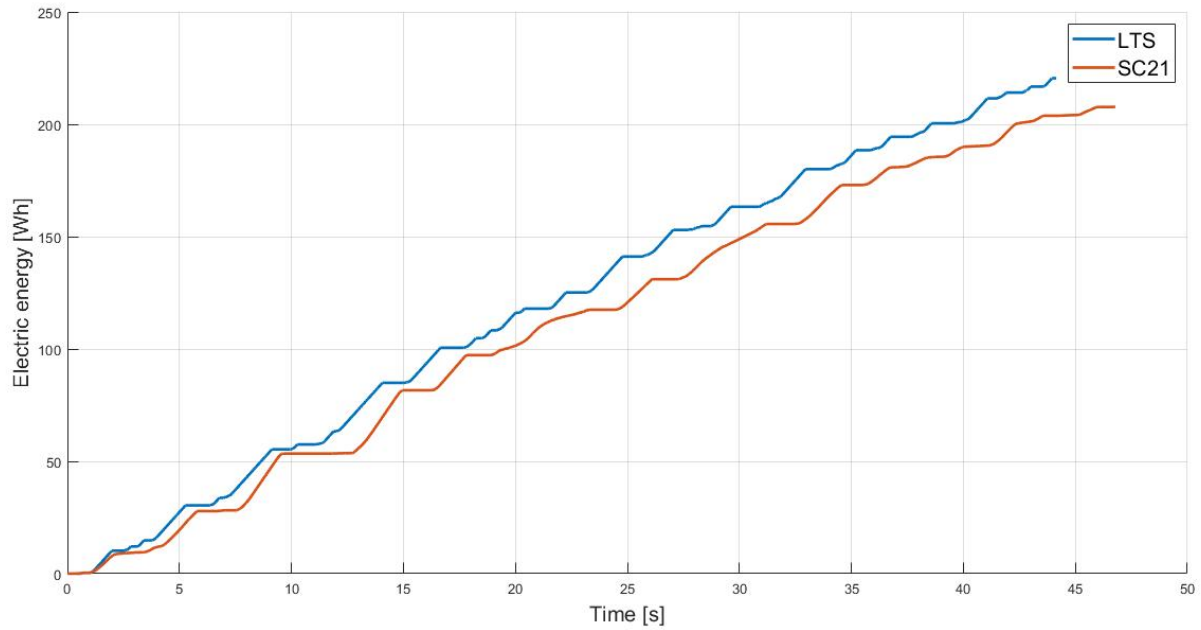


Figure 57: Electric energy-time plot comparison with the real car

	Lap-time [s]	Energy [Wh]
SC21	46.6	207.7
LapTimeSim	44.1	220.5
Error	5.7%	5.8%

Table 5: Comparison between real car and simulator

6.1.2 Comparison with the transient simulator

The following plot displays the comparative analysis between the simulated quasi steady-state speed profile and the simulated transient speed profile, as a function

of distance. The two trends are more similar than the actual vehicle logged data. This is mainly because of the driver controller featured in the commercial transient simulator, which is not ideal but has a higher precision when compared to a human driver.

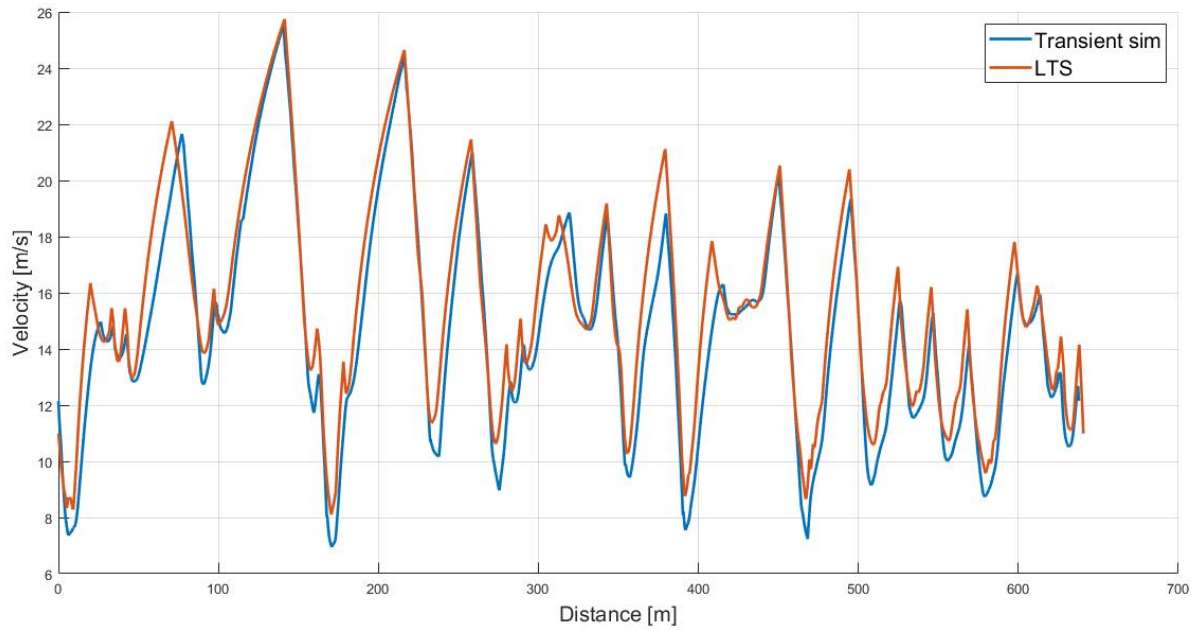


Figure 58: Velocity-distance plot comparison with a transient simulator

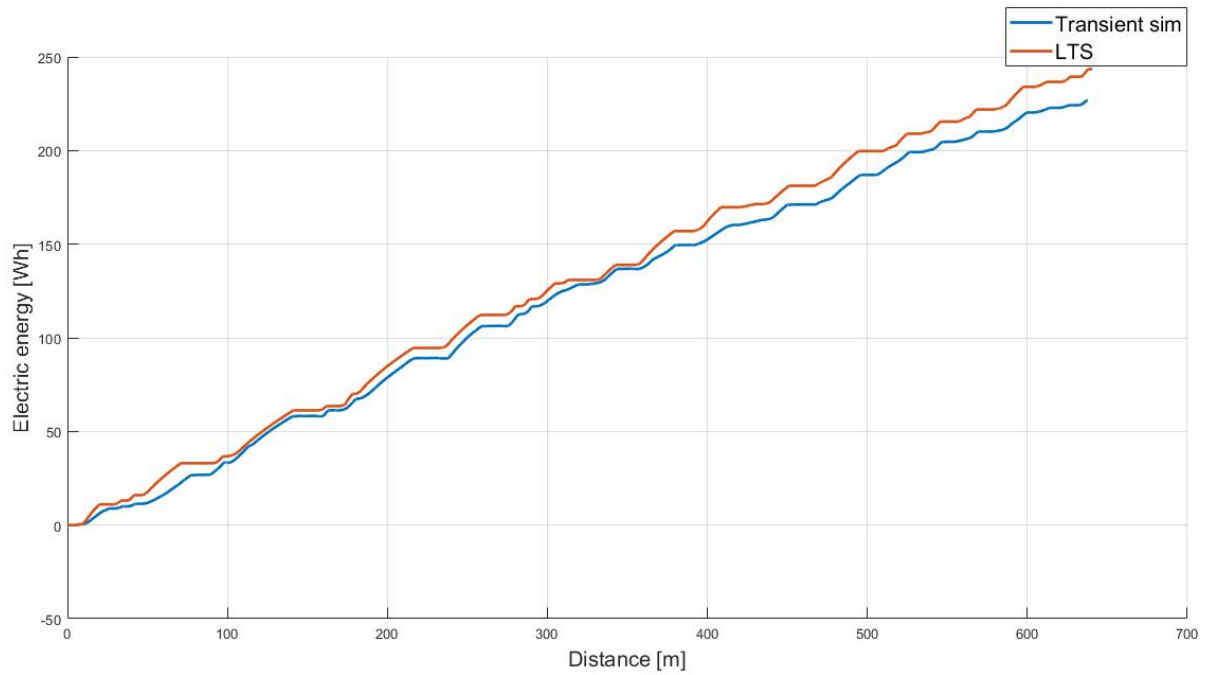


Figure 59: Electric energy-distance plot comparison with a transient simulator

The powertrain in the two simulators is similarly modeled, with the same motor efficiency lookup tables, and with the same battery pack data as inputs. As a result, the energy consumption is quite similar, with the quasi steady-state simulator that has a slightly higher energy consumption.

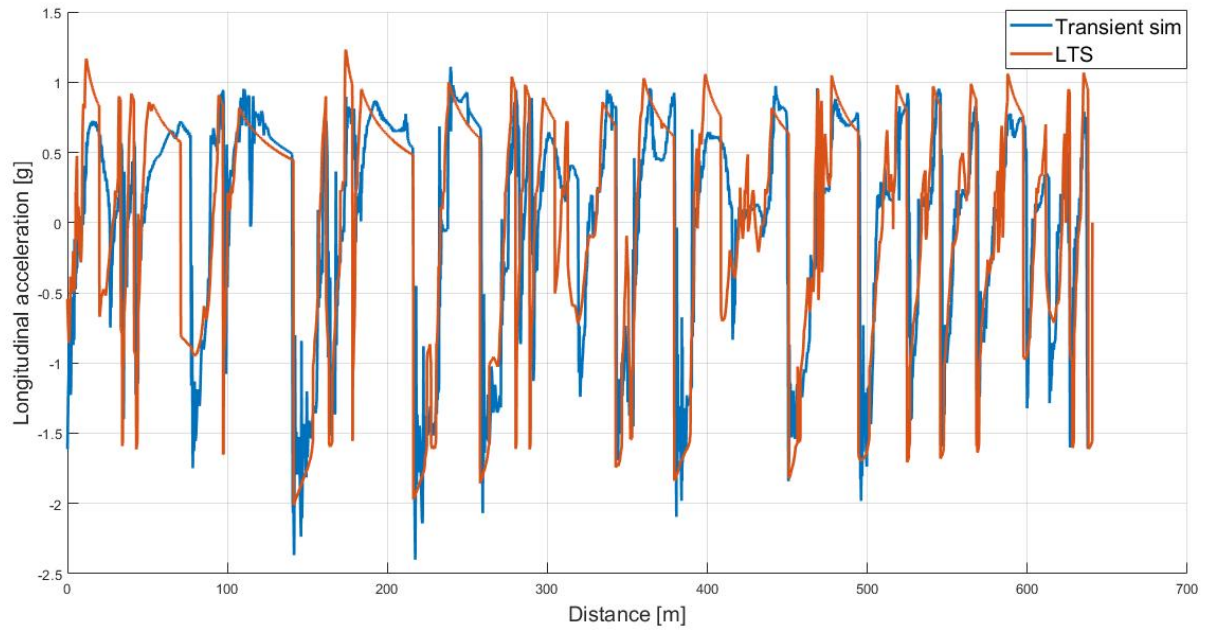


Figure 60: Longitudinal acceleration-distance plot comparison with a transient simulator

In this comparison, the acceleration values are more similar than in the previous one. The data comparative is particularly improved in the miscellaneous turns sections: this is mainly because of the higher precision of the driver in exploiting the performances of the car.

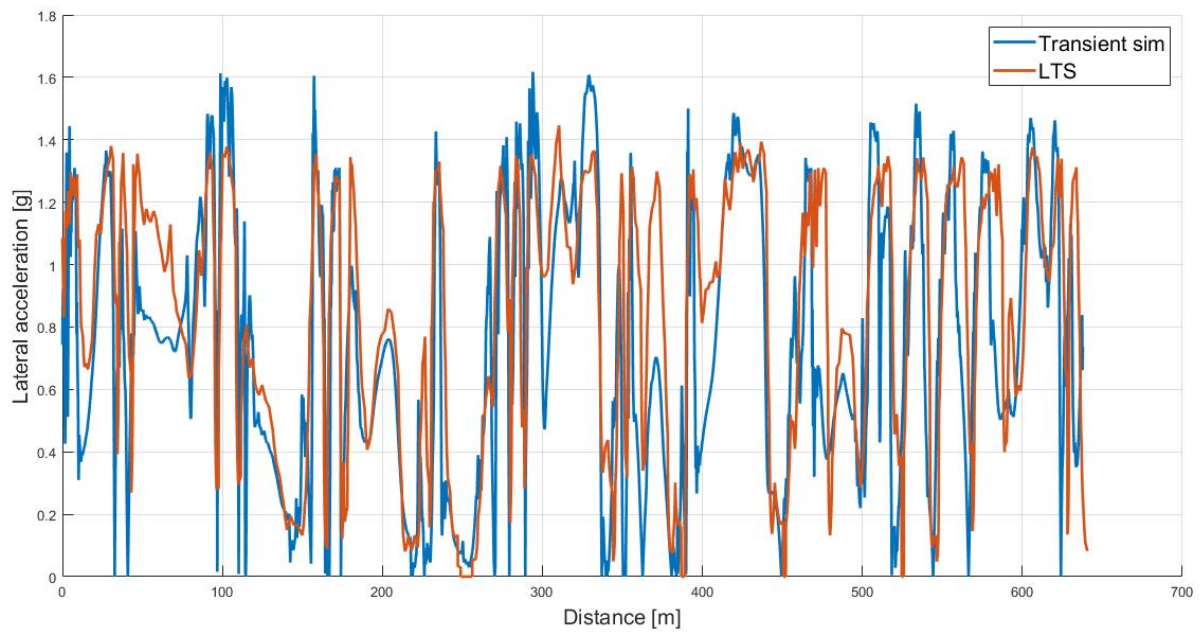


Figure 61: Lateral acceleration-distance plot comparison with a transient simulator

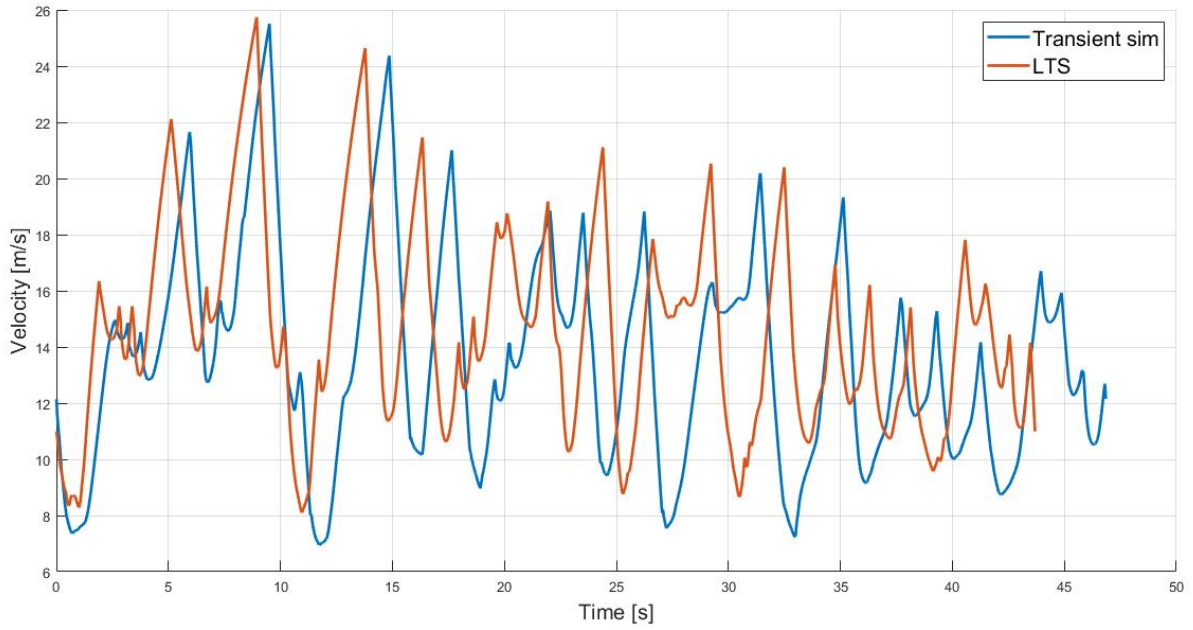


Figure 62: Velocity-time plot comparison with a transient simulator

Finally, these two last plots present the velocity and energy consumption comparisons as a function of time. As expected also here, because of the lower complexity and higher level of simplification for the vehicle model, the quasi steady-state simulator completes the lap with a lower time. As a result, the energy consumed by the quasi steady-state lap time simulator is slightly higher.

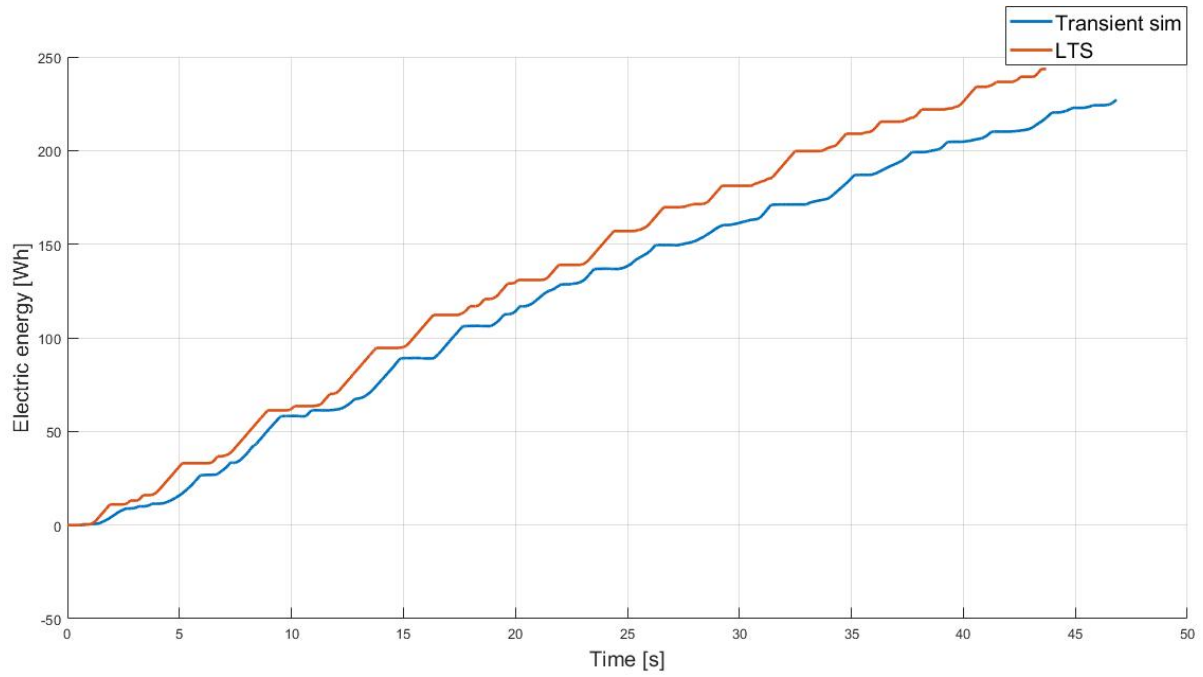


Figure 63: Electric energy-time plot comparison with a transient simulator

	Lap-time [s]	Energy [Wh]
Transient sim	46.8	226.9
LapTimeSim	43.7	243.5
Error	7%	6.8%

Table 6: Comparison between transient simulator and quasi steady-state simulator

6.2 GGV diagrams results

6.2.1 Comparison with the transient simulator

In this last section, the GGV diagram is compared with the data obtained from two lap time simulations using a transient lap time simulator modeled on the same vehicle.

In particular, the GGV is superimposed with the simulations coming from two different tracks: the first one, slower and with a lot of hairpin turns, and a second, with more rounded turns and longer straights.

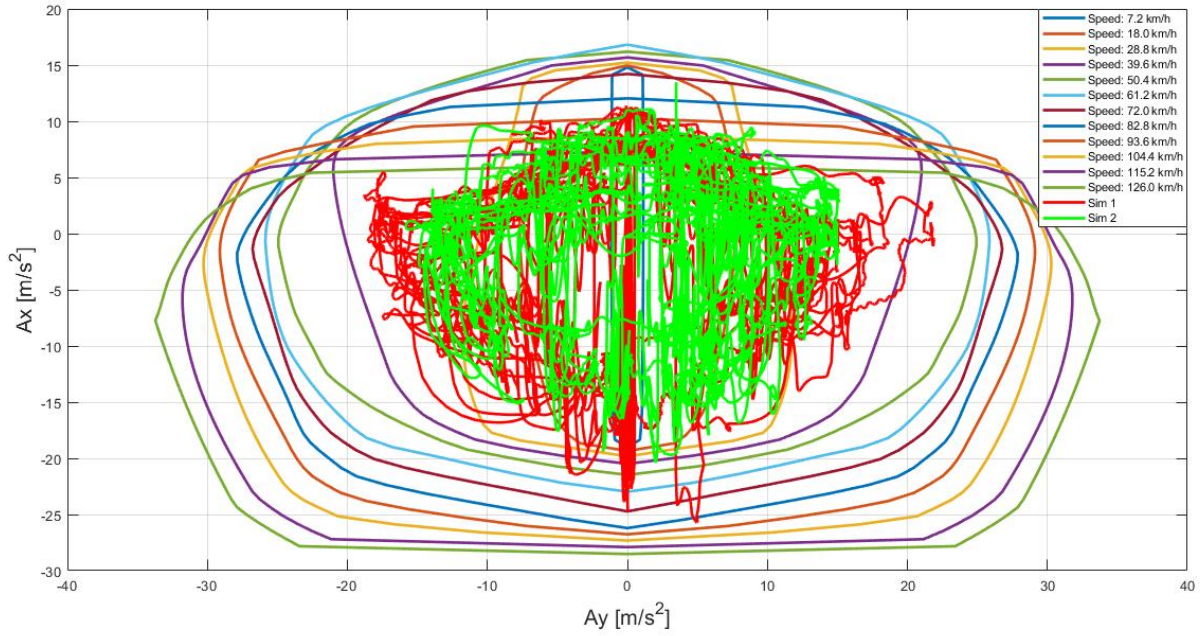


Figure 64: Comparison between GGV and simulations data

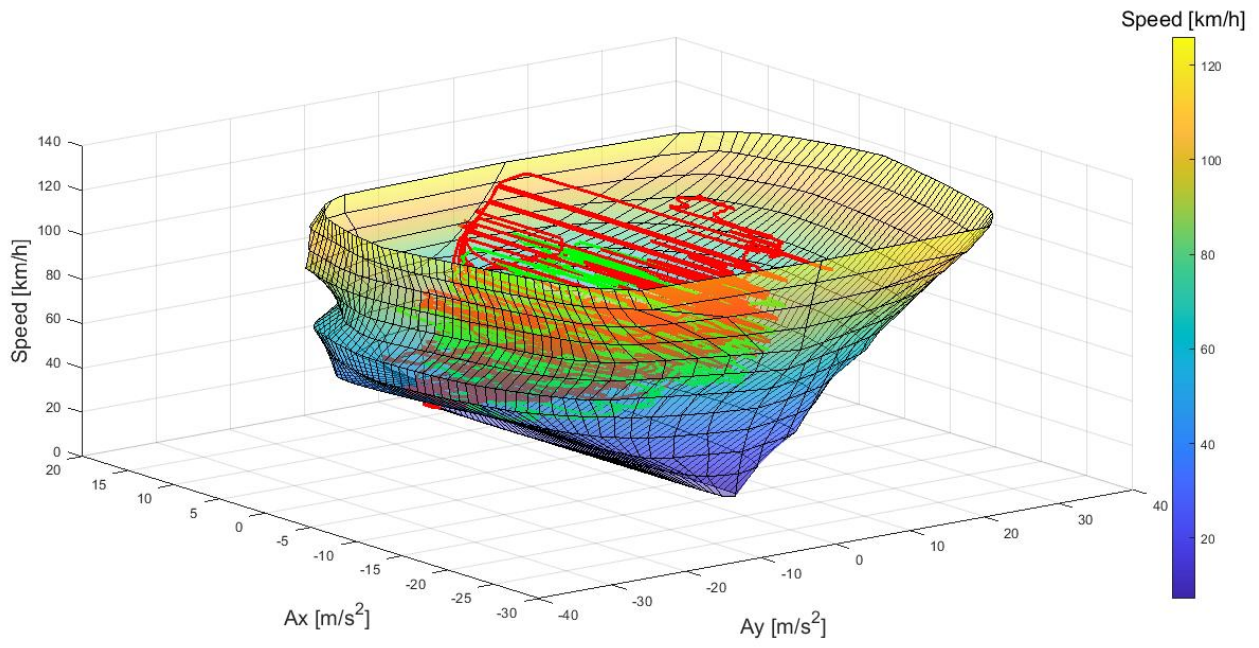


Figure 65: Comparison between GGV and transient simulation data

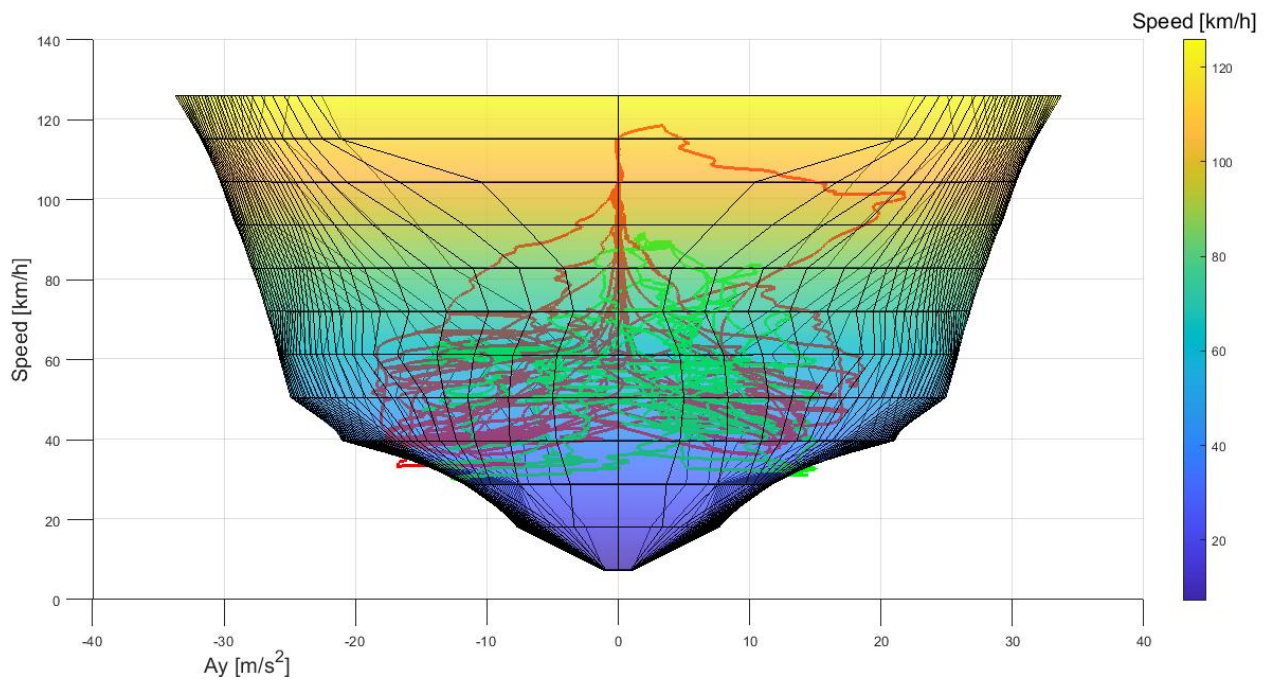


Figure 66: Front view of the GGV surface with respect to the transient simulation data

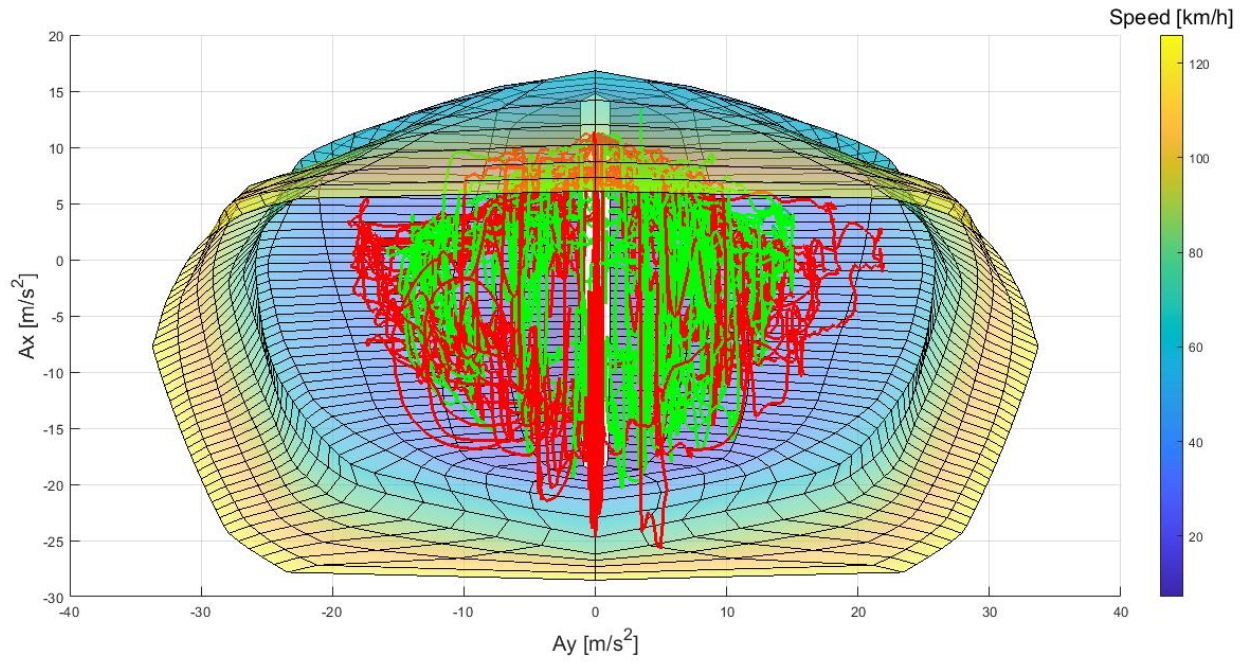


Figure 67: Top view of the GGK surface with respect to the transient simulation data

As displayed in the plots, most of the acceleration points created by the simulator are inside the performance envelope obtained through yaw moment diagrams. At low speed, there are some points outside of the GGK. This is probably because of the presence of very narrow hairpin turns, where the vehicle reaches the peak lateral acceleration in transient conditions. However, the GGK diagram is created considering only the points at which the vehicle is in steady-state cornering equilibrium.

7 Conclusions

The importance of vehicle dynamics simulations is a well-known topic since it is a widely used approach in the motorsport field. These simulations are a very useful instrument to provide reliable data to engineers in the design phase, predicting the resulting behavior of the vehicle, and also on track, to help race engineers in both the setup of the vehicle and in the development of a race-winning strategy. These tools have particular importance inside Formula Student teams, where all the engineering choices are evaluated in the design event. For these reasons, it is necessary to have a validated and trustable lap time simulator, which can be customized according to the vehicle needs and that can be updated throughout the years.

From the results of the point-mass simulator, it is possible to see that the overall correlation with the actual car is quite good, especially considering the high level of simplification in the simulation approach and the low quantity of inputs required. As expected, the comparisons show that the simulator is always faster than the actual vehicle and the transient simulator. This is mainly because of the quasi steady-state approach with a point-mass vehicle model and because of the driver, which is considered perfect and can bring the car to its limit in any condition. As a consequence, even the resulting consumed energy is slightly higher than the real one.

Furthermore, during the season, some issues with the vehicle arose, which compromised the possibility of gaining more data to refine the vehicle model of the simulator. Another problem was the change of tire compound throughout the seasons, which reduced the data available for tuning grip parameters of the vehicle. In conclusion, the results of the simulator are considered trustable, with differences in time and energy with respect to the actual car of around 5%.

In the final part, which is dedicated to yaw moment diagrams, a starting point for the evolution of this simulator is presented.

The idea is to introduce this innovative approach, that was never used before inside the team, to both evaluate and characterize the vehicle dynamics behavior and to create a GGV diagram for the simulator. This method allows the team to simulate a two-track model of the vehicle instead of a point-mass one, considering

more inputs, such as the tire model (.tir file), lateral aerodynamics, and suspension effects. The purpose of this simulator is not to replace the point-mass one, but to have a more accurate simulator for the setup of the vehicle on track.

Considering the premises, the validation of the simulator with respect to a transient simulator can be considered satisfactory. The main issue with the usage of the tire model is the above-mentioned compound change, which prevented the .tir file from being accurately rescaled, and thus precluding the possibility of comparing the simulated data with the data logged on track.

Appendix

Example of sensitivity analysis

This appendix shows an example of a sensitivity analysis, varying both the mass and the aerodynamic coefficients of the vehicle. As a first approximation, the aerodynamic efficiency is kept constant with a value of 3.

By simulating all the combinations of mass and $Cz * A$, it is possible to find the following results. The analysis is performed on the entire endurance event, which has a total length of 22 kilometers. The lap time output is the estimated overall time to complete the event, without considering the driver change in the middle, while the total energy required does not take into account the regenerated energy. By estimating the regenerated energy and removing it from the total amount of energy required, it is possible to find the battery capacity needed to conclude the event.

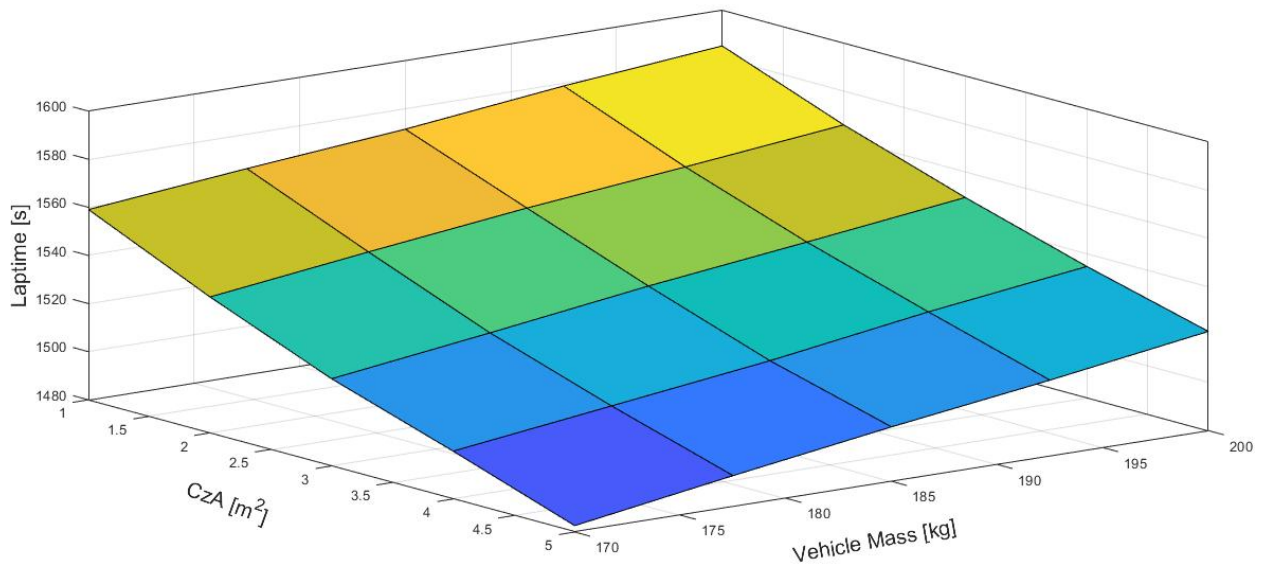


Figure 68: 3D plot of the endurance lap time as a function of the mass and aerodynamic parameters

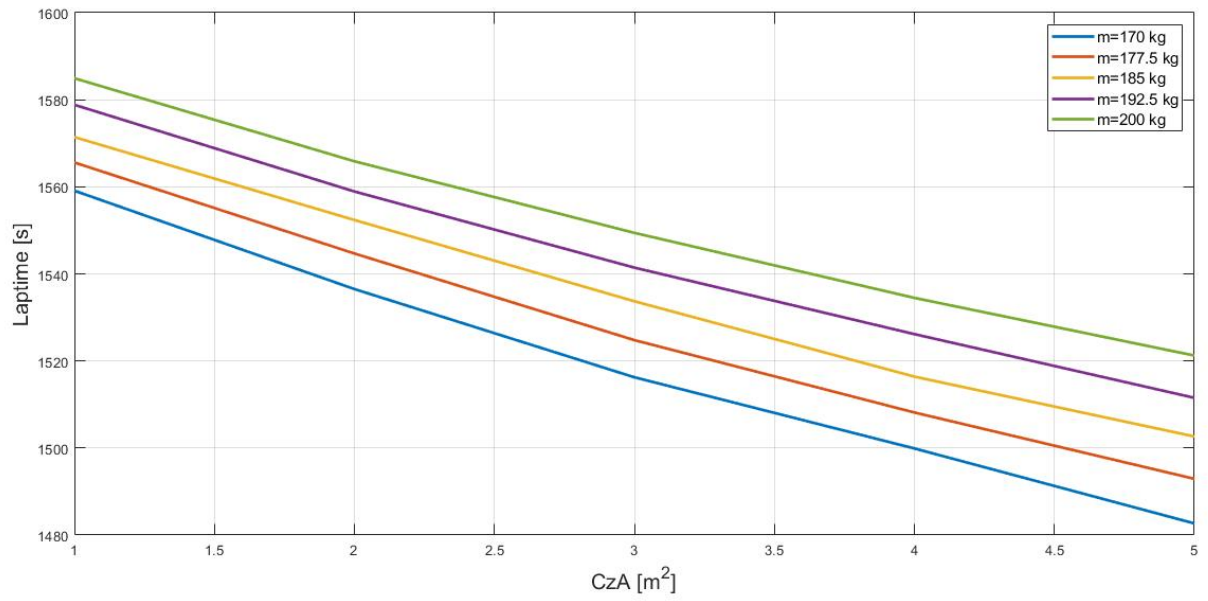


Figure 69: 2D plot of the endurance lap time as a function of the mass and aerodynamic parameters

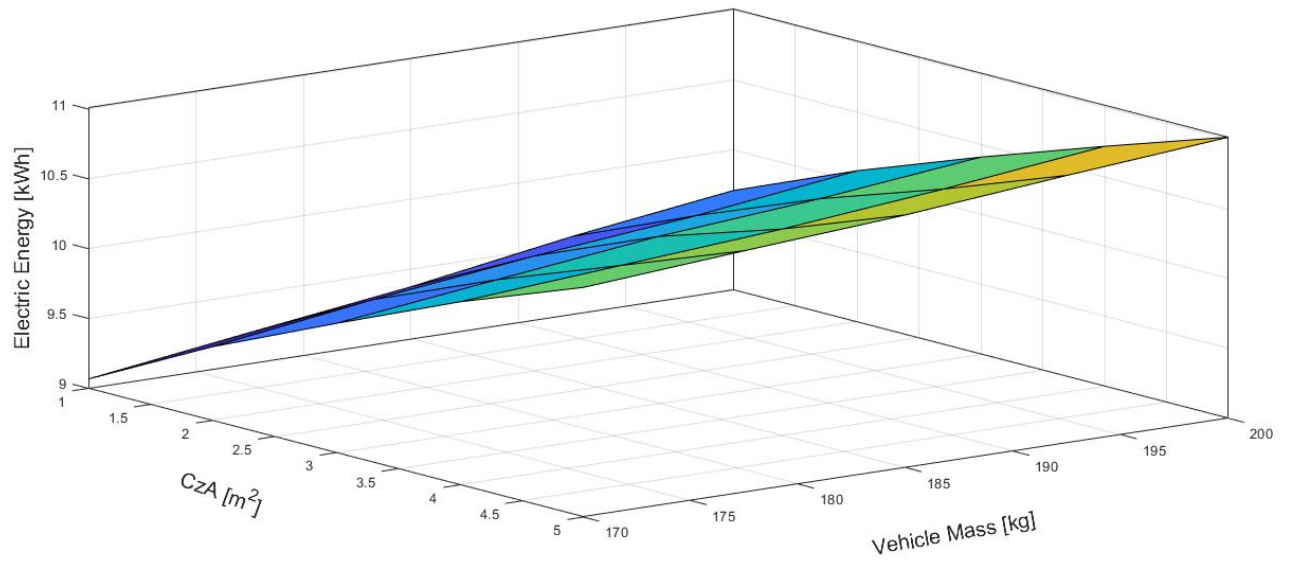


Figure 70: 3D plot of the endurance electric energy consumed as a function of the mass and aerodynamic parameters

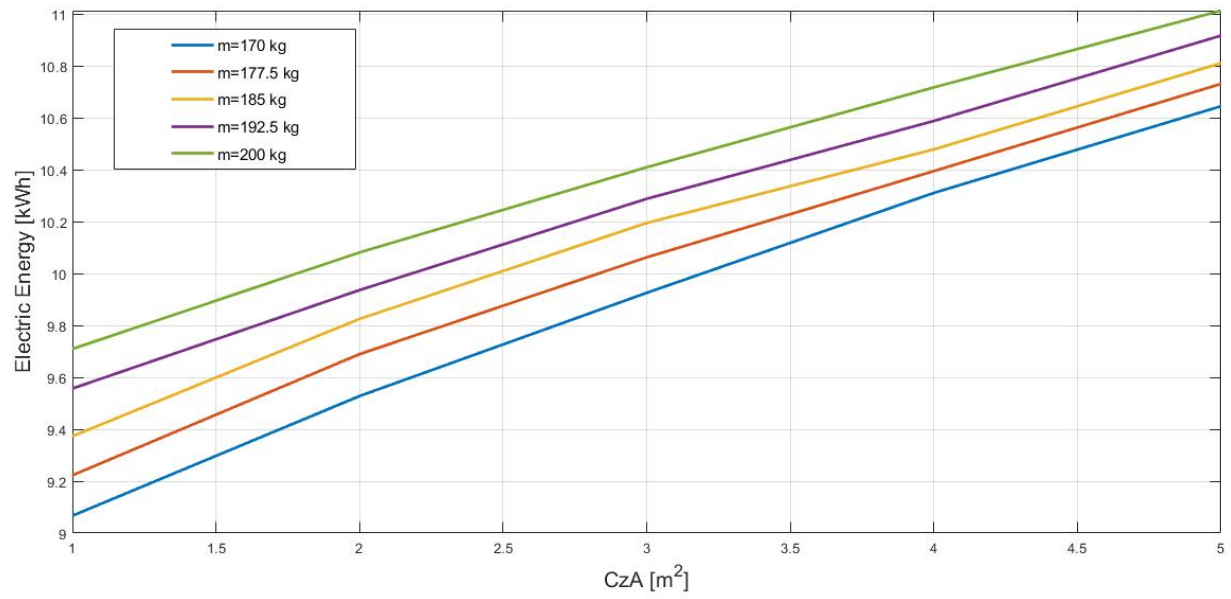


Figure 71: 2D plot of the endurance electric energy consumed as a function of the mass and aerodynamic parameters

References

1. www.fsaeonline.com consulted on 10/06/2021
2. www.formulastudent.de/about/chronicle consulted on 10/06/2021
3. www.formulastudent.de/fsg/rules consulted on 10/06/2021
4. www.formula-ata.it/wp-content/uploads/2021/01/Information-_Rules_2020_2021_rev4.pdf consulted on 10/06/2021
5. www.fsaeonline.com/cdsweb/gen/DocumentResources.aspx consulted on 10/06/2021
6. www.squadracorsepolito.com consulted on 10/06/2021
7. fs-world.org/E consulted on 11/06/2021
8. **“Race Car Vehicle Dynamics”**, Milliken & Milliken
9. **“The Automotive Chassis”**, Vol.2, Genta & Morello
10. **“Lap Time Simulation”**, Tomas Novotny, 2017
11. **“Development of Vehicle Dynamics Tools for Motorsports”**, Chris Patton, 2013
12. **“Generating MMM diagram for defining the safety margin of self driving cars”**, G Szucs and G Bári 2018 IOP Conf. Ser.: Mater. Sci. Eng. 393 012128
13. **“Optimum G seminar slides”**