Politecnico di Torino

Creation and Validation of a Formula SAE Vehicle Model

Automotive Engineering Master Degree Thesis

Supervisor

Prof. Andrea Tonoli

Candidate

Luca Raimondo Marongiu

Academic year 2019-2020

Abstract

The following discussion concerns the creation of a dynamic vehicle model for the 2018 Vehicle of Formula SAE team "Squadra Corse PoliTo" and its subsequent validation to obtain more relevant results from simulations. Topics discussed in detail range from the basic model creation, to track testing and data logging to obtain a reference behavior for the vehicle and finally to the comparison of the model behavior with the real vehicle and subsequent optimizations of the model.

Summary

Chapter 1

Squadra Corse Polito – Formula SAE

1.1 Formula SAE

Formula SAE is an international student competition where teams from universities all over the world are challenged with the task of creating a race car that is then evaluated based on various performance metrics and the merits of the design choices taken by the teams. The competition was established in 1980 and the main objective behind it is to give engineering students a way to apply in a practical way what they learned in university courses, using tools and techniques that are common in the industry and preparing them for their future careers.

Every year there are multiple official events around the world, usually starting in May (Formula SAE Michigan) events and ending in September (Formula SAE Japan). Each car is eligible to participate to the same event for only two years in a row, leading to frequent redesigns in order to have a fresh experience for all team members. This rapid environment and not so stringent regulation (for non-safety critical aspects) also leads to rapid development of innovative technologies and ideas and the cars in the competition, while usually following some trends, are very diverse

The typical race is divided into static and dynamic events. The statics evaluate aspects such as the merits of the design of each car, the cost considerations of the design and a business case study on the car while the dynamics evaluate various performance metrics of the car, such as acceleration, lateral adherence, full lap time and endurance of the vehicle.

In detail, the events are:

STATIC EVENTS

- **Design event**: At the start of the engineering design competition, the students must hand in an eight-page technical description of their car. The documents must show both their design and how the design will be applied to the ir chosen construction. On the basis of these documents, the members o f the jury will evaluate the layout, technical design, construction and implementation of the production of the actual vehicle. Then, there will be a discussion where the teams are questioned by the judges. These discussions focus on clarifying tech nical details, exploring the thinking behind the chosen design, as well a s the corresponding technical understanding of the students. The evaluation will not only assess the quality of the technical solution in question but also the reasons behind it.
- **Cost Event**:
  In the cost analysis event, the teams must grapple with the calculative size of the vehicle, its components, and the necessary manufacturing steps and record all of this in a written cost report. The students must then answer questions from the judges relating to the cost report on their prototype. In addition to considering the thoroughness of the written report, the students' understanding of the manufacturing process and the total cost calculation will be assessed.
- **Business Plan Presentation Event**: Each team presents their business plan for the constructed prototype to a fictitious company represented by judges. During a ten minute presentation, the team must demonstrate why their design best fulfils the demands of their target group and show how their design can be successfully marketed. The presentation will be followed by a five minute discussion and question round with the judges. In this event the content, structure, and editing of the presentation, as well as the team‘s performance in delivering it, will be evaluated alongside their answers to the panel‘s questions.

DYNAMIC EVENTS

- **Acceleration Event**: The vehicle‘s acceleration from a standing start is measured over a 75 metre straight. In addition to traction, the correct engine design is especially important, either in terms of greater power or for the highest possible torque. The fastest cars cross the line in less than four seconds and can reach speeds of over 100 km/h by the end of the stretch.
- **Skidpad Event**: During the Skid Pad event, the cars must drive a figure of 8 circuit lined with track cones, performing two laps of each circle. In each case, the second lap will be measured. The lap time gives a comparative value for the maximum possible lateral acceleration of the car. Most of the cars use aerodynamics to raise the contact pressure and thus, increase lateral acceleration. As with all the dynamic events, knocking over any of the cones results in a time penalty.
- **Autocross Event**: In the autocross event, the cars traverse a 1 km long track with straights, curves, and chicanes. A fast lap time is a sign of high driving dynamics, precise handling and good acceleration and braking ability. Once again, time penalties occur for those who knock over any cones. The autocross rankings decide the starting positions for the endurance competition that follows.
- **Endurance Event**: Providing the highest number of points, the Endurance is the main discipline. Over a distance of 22 kilometers the cars have to prove their durability under long-term conditions. Acceleration, speed, handling, dynamics, fuel economy, reliability – the cars have to prove it all. The Endurance also demands handling skills of the driver because there can be up to four cars on the track at the same time. Each team has only one attempt, the drivers change after 11 kilometers.
- **Efficiency Event**: During the endurance race, fuel consumption (combustion cars) or energy consumption (electric cars) is precisely recorded. However, the absolute fuel and energy consumption is not what is used to calculate the efficiency score, but rather the consumption relative to speed. This is to prevent teams from driving particularly slow in the endurance competition in order to obtain a high score in the efficiency category.

The scores for all disciplines in both the Italian and Spanish event are reported in the table below:

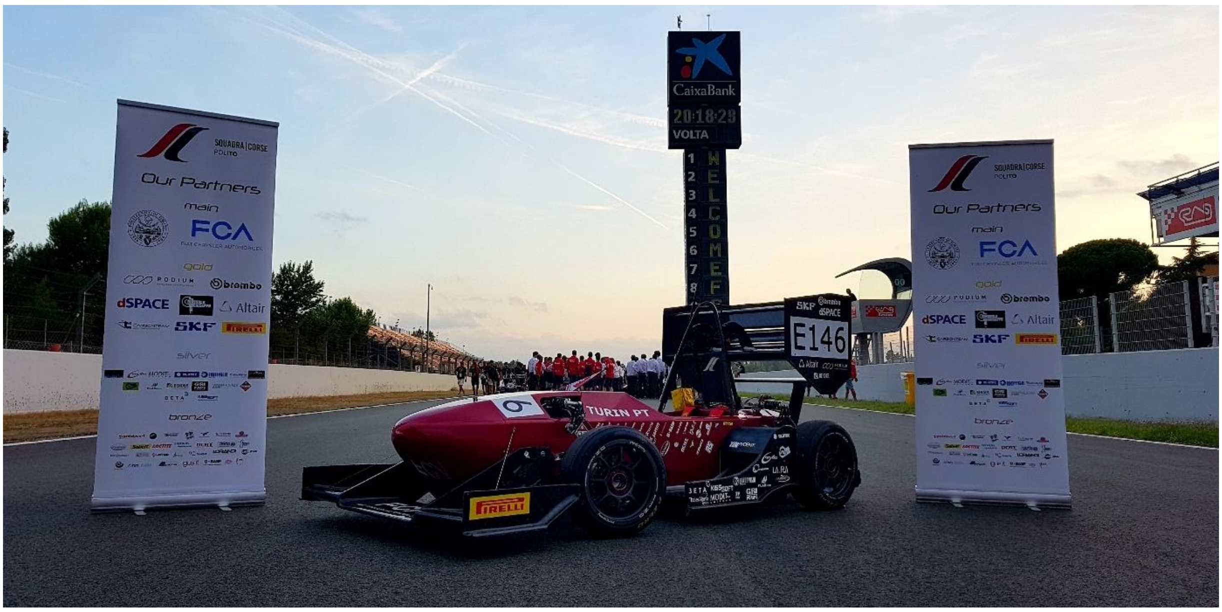
Event	Max Score (Formula SAE Italy)	Max Score (Formula Student)
Design	150	150
Cost	100	100
Business Plan	75	75
Acceleration	100	75
Skidpad	75	75
Autocross	125	100
Endurance	275	325
Efficiency	100	100
Total	1000	1000

1.2 Squadra Corse – Polito

Squadra Corse is the Formula SAE team of the Polythecnic of Turin. The team was created in 2003 and took part in a Formula SAE event for the first time in 2005. It participated in the Combustion category until 2012, when it switched to electric propulsion with the SC12e. The team also participated to Formula Hybrid in 2010, winning the competition. Another important step was the adoption of a 4wd configuration with independent motors for the 2015 model, the SCXV. This is the same configuration used by the latest vehicles of Squadra Corse, the SC18 (object of this discussion) and the SC19.

The SC18, the vehicle discussed in the following thesis, participated in Formula SAE Italy and Formula Student Spain, securing third place overall in Italy and achieving a world ranking of 23 out of 188 teams in the Electric category at the end of the season. It achieved very important milestones in the team's development: compared to the SC17, the mass was lowered of 20 kg (from a starting weight of 220 kg, almost a 10% improvement), the CG height was lowered from 260mm to 245mm (6% improvement) mainly due to the choice of a low profile tire that also has markedly better consumption at the expense of a negligible decrease of peak friction coefficient. Other improvements were the increase in battery pack capacity at lower weight due to the choice of better battery cells.

The team continued to improve season after season and is currently currently ranked 10th out of 188 teams in the Electric category after achieving the first place in Formula SAE Italy with the new SC19.



Chapter 2

Vehicle architecture

2.1 Main Parameters and subsystems

In this chapter a brief introduction to the vehicle's parameters will be presented, to give a better idea of the type of vehicle that has been modeled. Furthermore, the Sensors and data acquisition systems used for the validation will be discussed in more detail.

General dimensions and masses

Typical of a Formula SAE car, the overall weight is very low at only 200 kg. This is due to the lack of a minimum weight regulation, thus making weight reduction of all components of the utmost importance. The wheelbase is 1525mm, the minimum allowed by the rules to have better agility, as the tracks are narrow and twisty, so agility is more important than stability. The track is of 1200 mm, as this was found to be the ideal compromise between agility and reduction of load transfer. The weight distribution is 56% to the rear. The target was 53%, however late changes to the location of some components shifted the COG backwards. Finally, the height of the CG is 245mm, a reduction of 15mm over the previous year's car, mainly due to packaging changes and the reduction of tire radius.

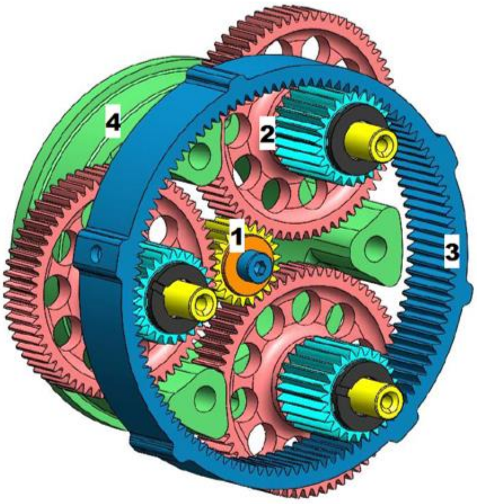
Monocoque

The SC18 chassis is a carbon fiber monocoque constructed entirely by the team members. The overall weight of the monocoque is of only 20 kg, including the steel roll hoop and aluminum front hoops mandatory due to safety regulations. The firewall, separating the driver from the battery pack, is made of Kevlar with fire retardant resin.

Suspensions

The suspension adopted by the SC18 is typical of a Formula SAE car: the architecture is SLA double wishbone at both front and rear. However, there are two peculiarities to the design: the anti-roll bar is a completely original design consisting of fiberglass “knives” to achieve the very low stiffnesses needed with such low weight. There is also an hydraulic third element that acts only in pitch and helps in decoupling pitch and roll motions. This too is an original design by the team. The Anti-roll bar and third element can be seen in the image below:

Powertrain



The vehicle has a 4WD configuration with four independent motors mounted on wheel. The motors and inverter package are AMK Formula SAE models capable of delivering a maximum torque of 21 Nm and a peak power of 35 kW for limited periods of time. However, the power limit for the whole car is of 80 kW, so such peaks are almost never reached. The maximum rotational speed is 20000 rpm. The motors are coupled to the wheel through a two-stage planetary redactor with a ratio of 14.82, thus giving the car a maximum speed of 120.6 km/h.

Battery pack

The Battery pack adopted for the SC18 is a single box accumulator container with a total capacity of 7,78 kWh composed by 720 Li-Ion cells in configuration 144s5p with a maximum voltage of 600 VDC and maximum power of 94 kW. The cells are divided in 6 modules each composed by 120 cells, with 24s5p configuration. Each module has a nominal voltage of 100 V, with a total fully charged rated energy of 5.44 MJ. The total weight of the battery pack is 44 kg leading to an energy density of 177.4 Wh/kg.

Aerodynamics

The car has a full aerodynamic package consisting of front wing, sidepods, diffuser and rear wing. The focus of the design was finding the best compromise between light weight, overall downforce and efficiency of the aeropack. The diffuser especially also had a big effect on the vertical CG position of the car, due to the battery pack, the single heaviest component of the car, being mounted on a slanted surface. However, the increased downforce was found be worth the 5mm estimated increase in CoG height after simulation and analysis. The overall Cx and , obtained through CFD simulation, resulted to be 1.03 and 3.7 respectively, giving an efficiency of 3.6.

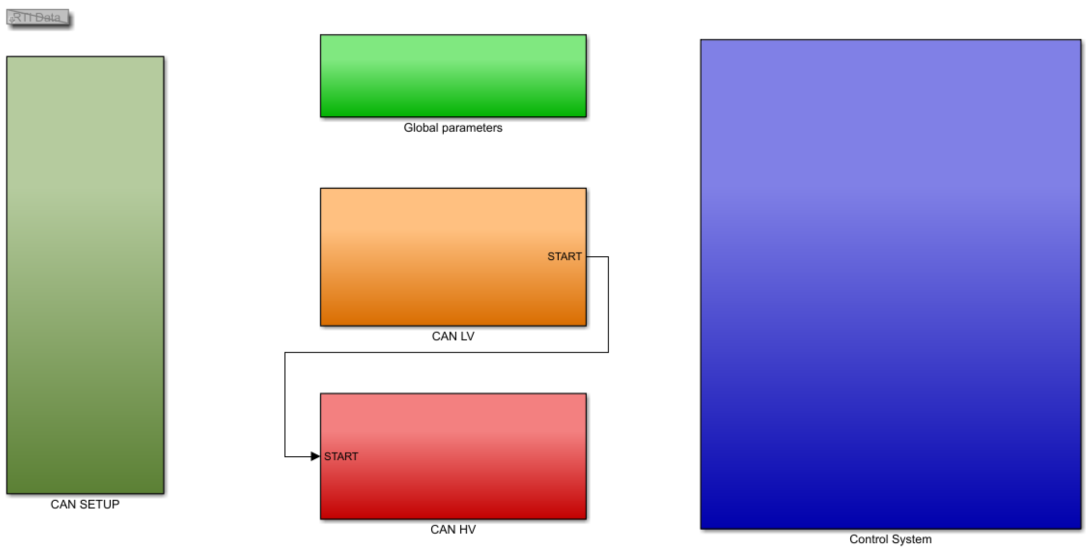
2.2 Electronics, Sensors and Data Logging

ECU

The Electronic Control Unit is a dSpace MicroAutobox II experimental ECU. It was chosen despite its significant weight because it allows to design the control systems of the vehicle in Matlab – Simulink environment and to translate it directly into code for the ECU utilizing the Matlab Coder function, leading to faster development time and easier debug. Furthermore, telemetry software is readily available and easy to configure, which is ideal for track testing. The ease of use, rapid prototyping, testing and reliability of this commercial model were thus considered valuable enough to justify its weight and cost over a custom built ECU.

As can be seen from the scheme above, the ECU is the most important electronic component in the vehicle: its main function is to acquire sensor and auxiliary board signals, utilize them in its control logic to determine various outputs. The main output of the control system is the torque request, determined

The main subsystems of the ECU control logic are shown below:



The function of each block is the following:

- **CAN setup:** provides all parameters for correct communication on the CAN line

- **CAN LV:** the acquisition of signals from the low voltage components such as sensors and other boards and to the transmission of internal signals to the datalogger

- **CAN HV:** reception and transmission of signals to the inverters and powertrain

- **Control Systems:** Contains all systems that determine the torque request in each instant, including the Power limiter, Torque vectoring, Traction control and Launch control subsystems

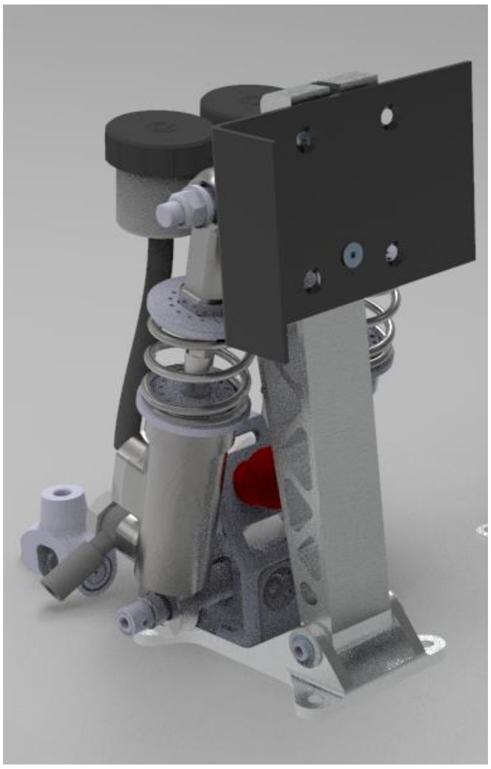
- **Global Parametrs:** The main variables of the control systems are stored here so their values can be changed rapidly when needed

Sensors

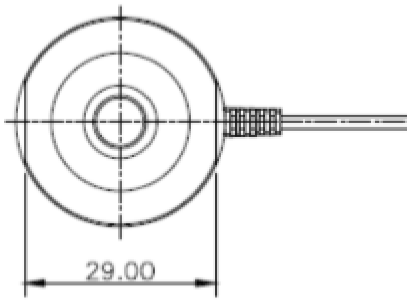
IMU

The Inertial Measuring Unit is without doubt the most important sensor in the vehicle. In this case the model is a *Bosch Motorsport MM5.10* 5-axis IMU, with 3 axes for linear acceleration (X, Y, Z) and 2 axes for rotation rate (Yaw and Roll). The sensor includes an internal low-pass filter with cutoff frequency of 15 Hz, resulting in smooth readings without post processing. The measurement range is of ± 4.2 g for the linear accelerations and of ± 163 °/s for the rotation speeds. The sensor was mounted behind the firewall, over the battery pack. This location was chosen because it is as close as possible to the location of the car’s CG, in order to avoid the reading of accelerations due to rotational motions of the vehicle that could sum to the linear components if mounted with eccentricity.

Brake Pedal sensors

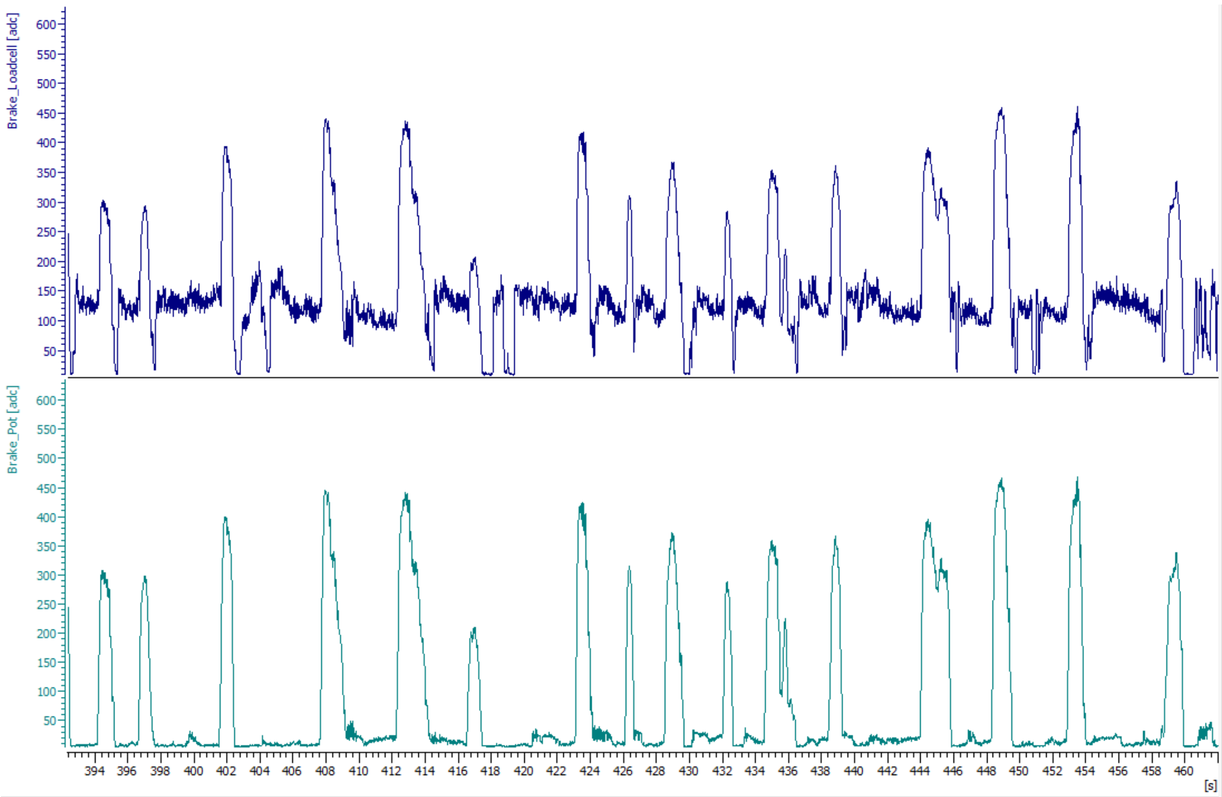


It was chosen for the 2018 car to have three different sensors for the brake pedal, this is due to different requirements for regenerative braking and hydraulic braking. In particular, the desired behavior was to have no movement of the pedal until a threshold value of applied force is reached: in this first phase the force applied on the pedal is registered by a load cell and by a strain gage, this value is used to determine the amount of regenerative braking needed. After the threshold value is reached the maximum value of regenerative braking is achieved, the pedal starts moving and the hydraulic braking is activated by the master cylinder. At this point the pressure in both front and rear brake lines is measured through pressure sensors.

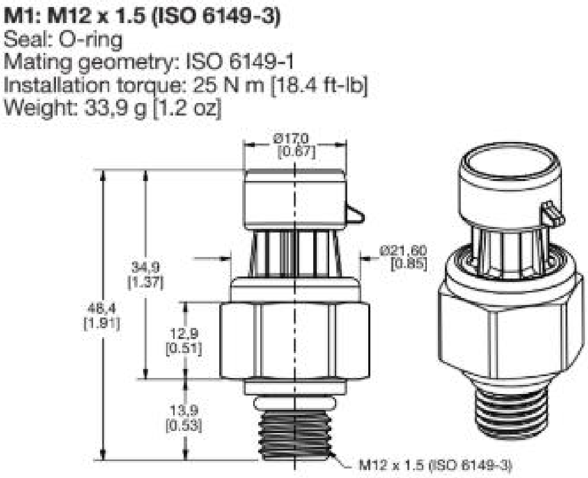


Compression Load Cell**:** The load cell was installed in the brake pedal through the use of a properly designed hollow portion of the brake pedal that would fit the diameter of the load cell and allow most of it to sit flush with the carbon fiber cover plate. The sensing potion was slightly protruding from this cavity and was installed facing the front of the vehicle. The model used is the TE connectivity FC23.

Strain Gage: the second sensor used to determine the regenerative braking request is a strain gage mounted on the rod end connecting the brake pedal to the master cylinder and spring assembly. This solution was chosen because in the previous year's car the load cell signal proved to be sensible to small shifts in position. The problem was solved by redesigning the pedal assembly, however it was chosen to also add another sensor for better reliability and redundancy. The strain gage proved to be more accurate than the load cell, as shown in the image below

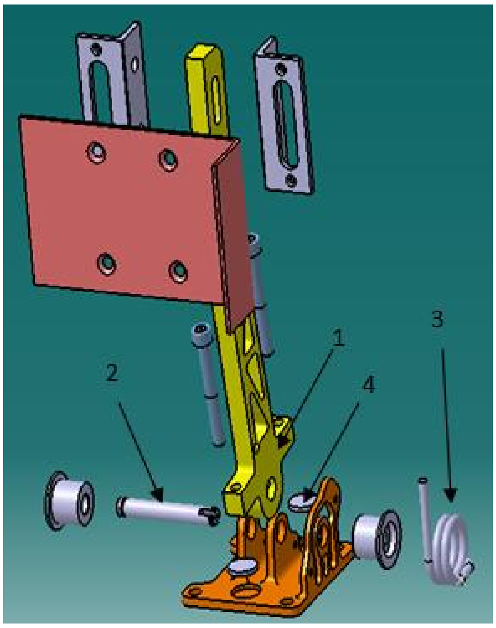


Pressure sensor: Finally, to accurately measure the pressure in the hydraulic braking system, two pressure sensors were installed directly on the braking circuit, one for the front braking line and one for the rear. In this case, the sensor used is an Honeywell's PX3 Pressure Transducer using piezoresistive sensing technology.



Throttle Pedal sensor

Since the vehicle is completely electric, throttle by wire is the only possible solution to implement. The angular position is converted into a voltage range by the sensor and then the software converts this voltage in a throttle request ranging from 0 to 100 (full throttle). The chosen sensor was a of the Magnetic Hall effect type and was installed on the center of rotation of the pedal assembly. Rules require the sensor to be redundant for safety reasons, so two values are registered for each instant and a plausibility check is performed: if the difference between the two vaues is higher than a threshold the tractive system is disabled.



The model used by the team was a Vishay 981 HE, installed in the center of rotation of the pedal.

Steering angular position sensor

The steering rack for the 2018 car was a commercially bought Zedaro zRack including a position sensor. The position sensor is of the non-contact Hall effect type that has the advantage of being very reliable due to the lack of friction and wear. Its main characteristics are reported in the table below:

Description	Unit
Power Supply Vdd	5 V ± 5 % (Red Wire)
Output Voltage Vout	0 V to Vdd (Green Wire)
Ground	(Blue Wire)
Power Consumption (Typ.)	26 mA
Output Load (Max.)	2 mA
Nonlinearity	1%
Accuracy	0.35 Degrees
Operating Temp.	–40 °C to +85 °C
Max. speed	30,000 RPM

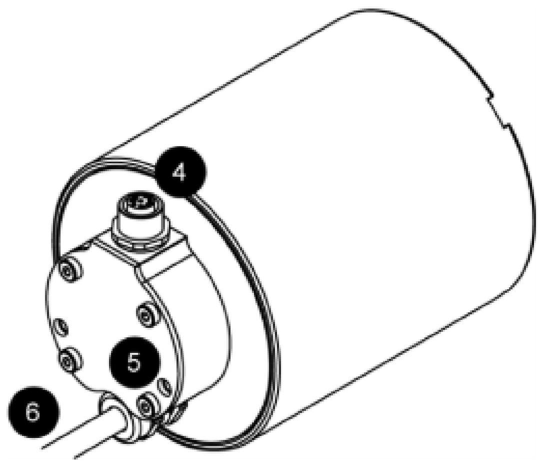
Energy meter

The energy meter is a custom made board provided by our sponsor FLAG-MS. It is used for the measurement of instantaneous current, voltage and power being drawn from the battery pack. It provides a much faster and more accurate reading of these values compared to the current transducer and BMS signals, so it is the main input used for the power limiter control of the car.

BMS and Battery pack current transducer

Other measurements about the voltage and current being drawn from the battery pack are available from the BMS and current transducer respectively. The BMS is custom made by our sponsor Podium Advanced Technologies and provides information about the battery pack voltage, the voltage of every single cell and their respective temperatures. It is also needed for the charging of the battery pack. The current transducer reads the instantaneous current being drawn from the battery pack.

Motor speed and current sensors

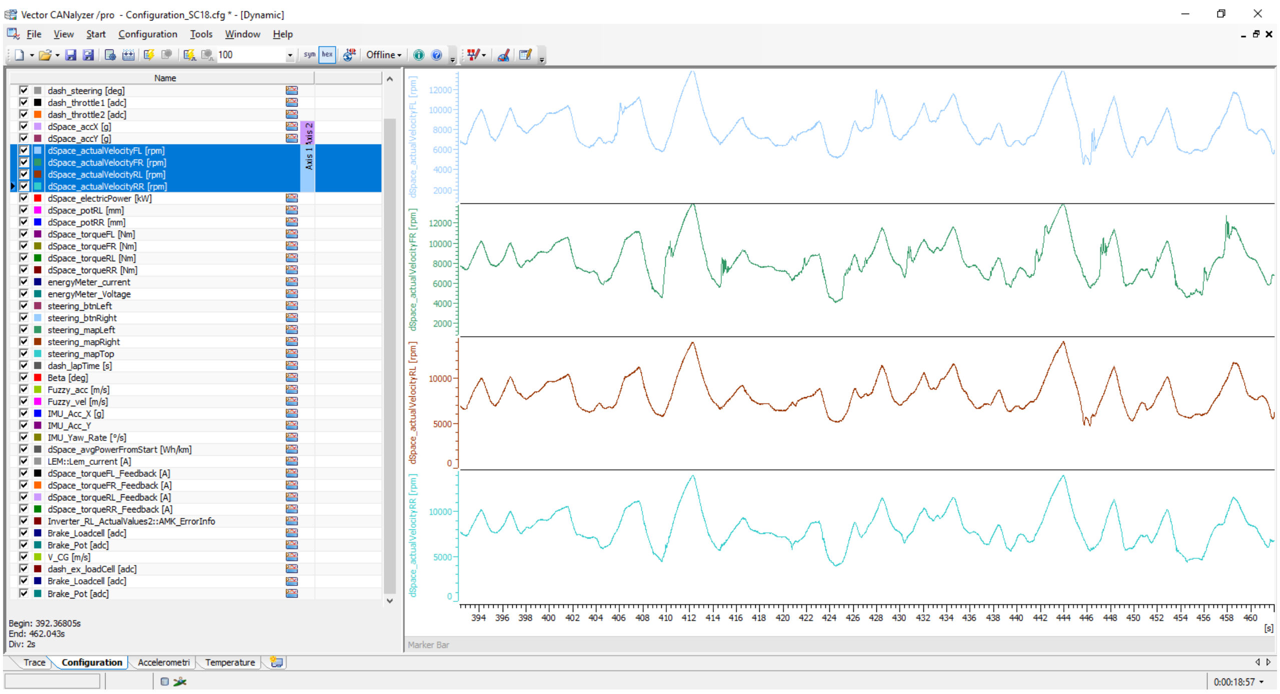


The motor speed is measured by the encoder included in the AMK motor and inverter package. The reading is very accurate as it is the basis of the speed control of the motors. Furthermore, sensors on the inverters measure the instantaneous current being supplied to each motor. This value can be used in turn to calculate the instantaneous torque, as the two values are related by a constant found in the datasheet.

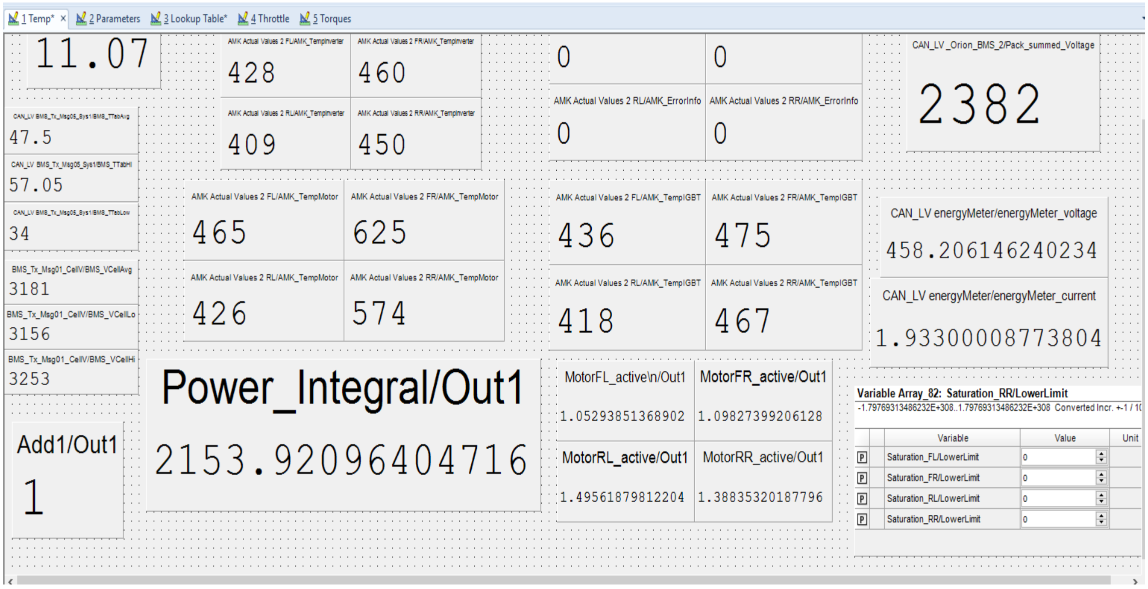
Data Logging and Telemetry

Data logging is a fundamental process in the development of any vehicle: without it, it is impossible to determine if the vehicle is performing as specified. In fact, data logging provides a way to diagnose possible problems and allows to use the data saved to perform analysis aimed at improving the performance of monitored systems. It also has a very important role in this dissertation: data logging is fundamental to acquire the input signals to be fed to the model for validation.

The datalogger utilized in the vehicle was a Vector CANcaseXL connected to the vehicle CAN line. The relevant signals, defined in a .dbc file, are logged on an SD card or alternatively monitored live by connecting a PC through USB for static tests. The software used to read the data logs is Vector CANalyzer, with the possibility to export data in various formats for post processing.



Furthermore, the previously mentioned ECU is connected through ethernet to a telemetry modem inside the car. This modem (slave) communicates with the master modem in the telemetry station through 2.4 GHz Wi-Fi. Finally, the master modem is simply connected to the telemetry station through ethernet.



This allowed the team to perform basic diagnostics without the need of stopping the car to read the data logs. Additionally, this type of telemetry is able to change the control system parameters in real time. This feature was used both for rapid calibration of control system parameters during track tests and also as a failsafe way to change parameters during a race in case of steering wheel boar malfunctions.

Chapter 3

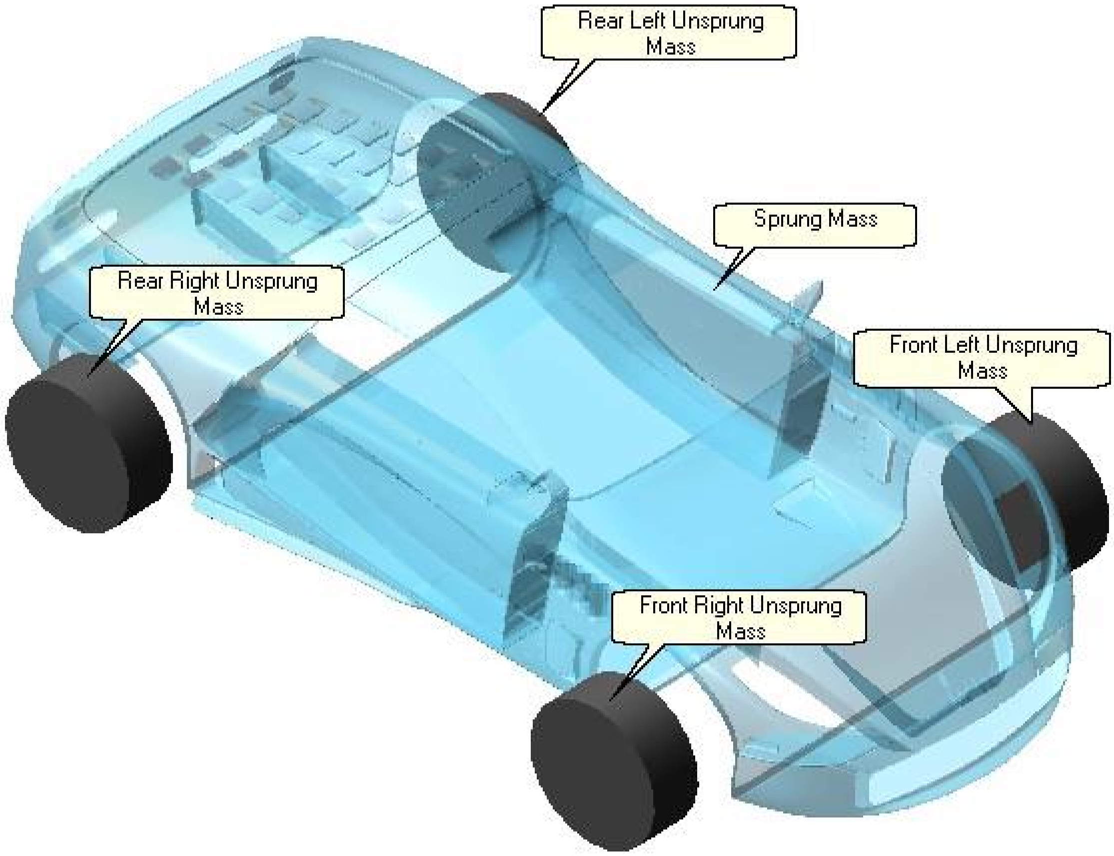
Vehicle modeling and Analysis

3.1 Introduction to Vi-CarRealTime

Nowadays, simulations are becoming more and more fundamental in the design phase of any product, including vehicles. Through simulation, ideas can be tested without the need of creating expensive and time consuming physical prototypes. However, while they provide many advantages, it is important to know the limitations of any software package and to test the effectiveness of the models used, in order to understand the degree to which results achieved to simulation can be relied upon. That is why the second part of this discussion will concern the validation of the model that is being created.

The software package chosen by the team to model the vehicle was Vi-CarRealTime, due to previous positive experience and a strong partnership with Vi-Grade. The software package allows to create vehicle models to aid in the design phase and test various options at the subsystem level and the influence of various parameters on vehicle performance.

Vi-CarRealTime is a parametric real time simulation environment that utilizes a simplified vehicle model and an advanced driver logic to perform a variety of user defined maneuvers, from simple straight line acceleration or braking to full lap-time simulation. The vehicle's subsystems are described in a parametric manner to allow rapid changes and ease of use. The vehicle model has 14 DOF, 6 of which are body rotations and translations, 4 are for the vertical motion of each wheel and the remaining four are for the longitudinal slip of each wheel.



Suspension and steering system characteristics are modeled not through linkages, but through the use of lookup tables that describe all relevant behaviors in each time step. Braking and powertrain are described completely by algebraic and differential equation without the need of adding extra parts.

In the full package, other utilities are included to aid in the creation of the model, these are:

- VI-Road: Tool for generating roads and driver paths.
- VI-Animator: Post-processing tool for plots and animations
- VI-SuspensionGen: elestokinematic analysis of suspension, generates lookup tables for suspension subsystem
- VI-TireLimits: Tool for the evaluation of tire forces using .tir property files

The typical procedure to correctly model a Vehicle is to gather all relevant data for each subsystem, input the values in the program and then input simulation parameters.

3.2 Subsystem modeling

As previously mentioned, each subsystem must be modeled individually. The various subsystems to be modeled are:

- Body
- Front Suspension
- Rear Suspension
- Front Wheels
- Rear Wheels
- Brakes
- Steering
- Powertrain

Body

The body subsystem is divided in two smaller categories: **suspended mass** and **aerodynamic forces.** The first contains all information about the mass and inertia properties of the suspended mass, while the second is for aerodynamic forces acting on the vehicle.

Suspended mass:

The relevant data required to model the suspended mass and the method used to obtain it is reported in the table below:

Data	Value	Unit	Obtained through:
Wheelbase	1525	mm	Design value, measured
CG distance from front axle	854	mm	From weight distribution
CG lateral position	0	mm	From weight distribution
CG height	245	mm	Lifting test
Mass	170,31	kg	W
I_{xx}	11945000	$kg * mm^2$	Inertia calculation (shown below)
I_{yy}	55240000	$kg * mm^2$	Inertia calculation
I_{zz}	54306000	$kg * mm^2$	Inertia calculation
I_{xy}	74000	$kg * mm^2$	Inertia calculation
I_{xz}	-36000	$kg * mm^2$	Inertia calculation
I_{yz}	436000	$kg * mm^2$	Inertia calculation

Mass

The suspended mass of the vehicle was obtained by weighting the whole vehicle and then subtracting the weight of the unsprung masses. The masses of the various unsprung components were either weighted individually, obtained from CAD models with proper volume and material or taken from datasheets.

CG position, longitudinal

In order to obtain the sprung mass CG location, first the total vehicle's CG location must be known. The vehicle's CG position in the longitudinal direction was determined simply by weighting the vehicle on 4 scales, one for each wheel. Once the weight distribution between front axle and rear axle was obtained, the CG distance from the front axle can simply be calculated as:

$$CG_{x_{vehicle}} = m_r * l$$

Where m_r is the mass percentage on the rear axle, l is the wheelbase and CG_x is the distance of the vehicle $CG_{x_{vehicle}}$ from the front axle.

Once this value was obtained, the position of the CG of the unsprung masses was calculated. Knowing both the Full vehicle's CG position and unsprung mass CG location the sprung mass CG was calculated using the formula:

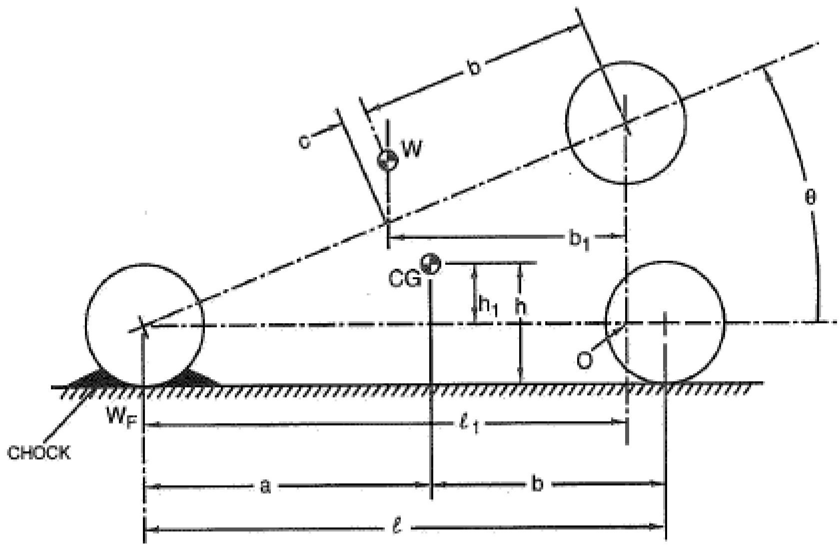
$$CG_{x_{sprung}} = \frac{CG_{x_{vehicle}} * m_{vehicle} - CG_{unsprung} * m_{sprung}}{m_{sprung}}$$

CG position, lateral

Again, by utilizing the 4 scales the percentage of mass to the left of the vehicle could be determined. However, the difference was negligible, so a position in the symmetry plane was considered.

CG position, vertical

Again, before the sprung mass's CG position could be calculated, the entire vehicle's CG position should be known. This was achieved through a lifting test of the vehicle, as illustrated in the figure below:



The procedure is the following:

The vehicle's weight on the four corners is measured on flat ground, then the front wheels are secured in place and the rear wheels are lifted, rotating the vehicle of angle θ. The vehicle's suspensions must be locked (by substituting dampers with rigid elements) to prevent motion that would alter the static geometry due to the shift in weight.

The parameters needed for the calculation are then:

M	200	Total mass of the vehicle [kg]
M_{front}	88	Mass on the front axle with rear elevated [kg]
b	671	Horizontal distance from rear axle to CG [mm]
l	1525	Wheelbase [mm]
R_l	237	Loaded radius of the wheels [mm]
θ	20	Angle by which the vehicle is raised [deg]

Parameters not indicated on the table are obtained from the image above. Then, the CG height is calculated with the following formulas:

$$l_1 = l * \cos(\theta)$$

$$M_f * l_1 = M * b_1$$

$$\frac{b_1}{b + c} = \cos(\theta)$$

$$h_1 = \left(\frac{M_f}{M} * l\right) - b$$

And finally:

$$h = R_l + h_1$$

Inertia Calculation

The inertia of all major components was evaluated by considering their real mass and their position in the CAD assembly and approximating their shape to that of either a cylinder or a parallelepiped. Then the inertia of the component was evaluated with respect to a local reference frame in the COG of the object and parallel to that of the car. For cylindrical shaped bodies we used the formulas:

$$I_{xx} = \frac{1}{4}mr^2 + \frac{1}{12}ml^2$$

$$I_{yy} = \frac{mr^2}{2}$$

$$I_{zz} = \frac{1}{2}mr^2$$

While for parallelepipedal bodies the formulas were:

$$I_{xx} = \frac{m(a^2 + l^2)}{12}$$

$$I_{yy} = \frac{m(b^2 + l^2)}{12}$$

$$I_{zz} = \frac{m(a^2 + b^2)}{12}$$

The inertias obtained this way were then translated first to the intersection between the firewall, midplane and monocoque floor, then to the car's COG utilizing the Huygens-Steiner inertia translation formula for parallel axes:

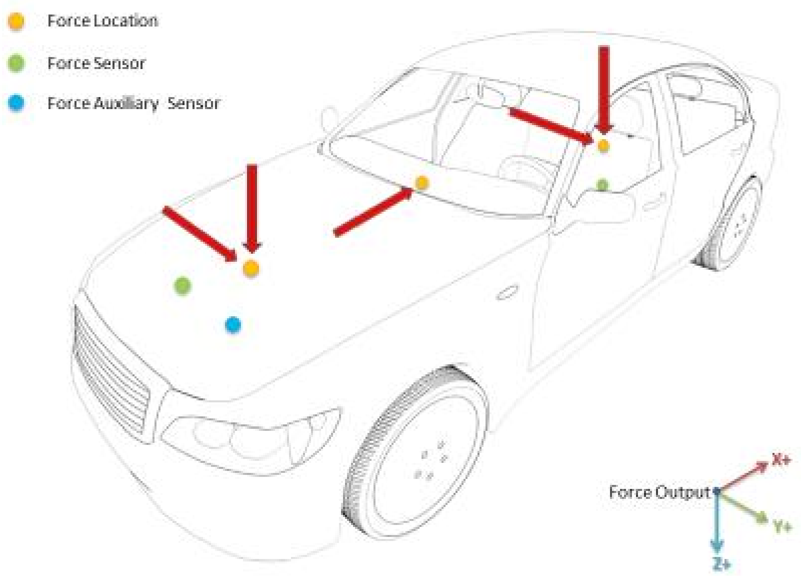
$$I_{xxCOG} = I_{xx} * m * x^2$$
$$I_{yyCOG} = I_{yy} * m * y^2$$
$$I_{zzCOG} = I_{zz} * m * z^2$$

Aerodynamic Forces:

To determine overall aerodynamic forces acting on the vehicle, Vi-CarRealTime uses a simplified approach where the total downforce is divided in front and rear contributions, while the drag force is applied in a single point. Each of those forces is then calculated using a property file, where the force at a reference speed and at various ride heights is indicated. During the simulation the forces in each moment can then be calculated simply by using the aerodynamic force formula:

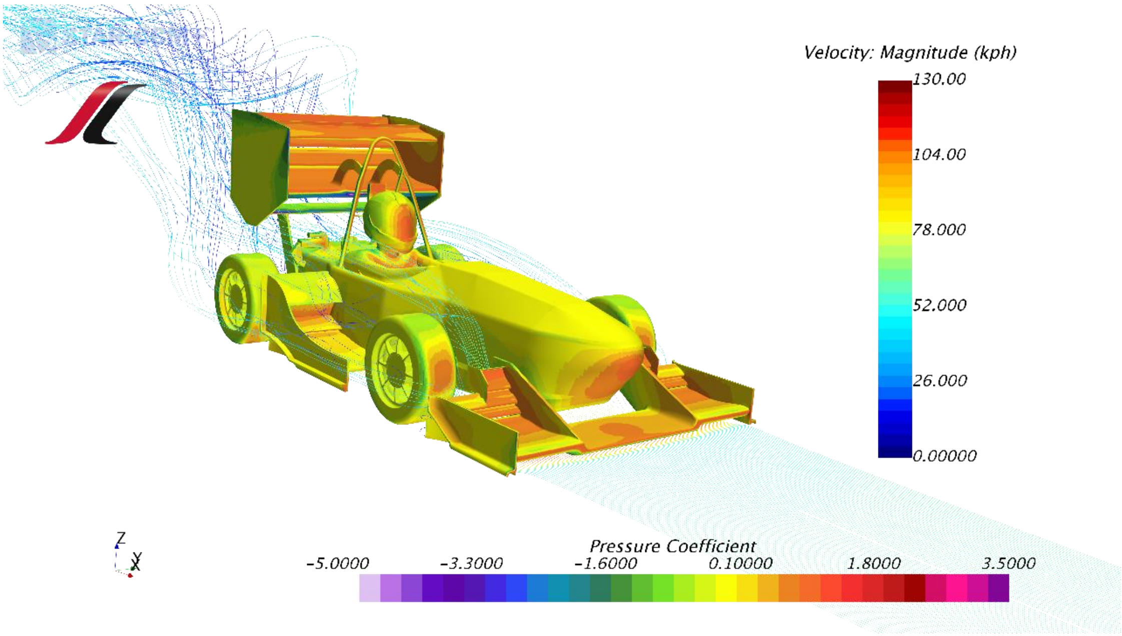
$$F_{aero} = F_{table} * \left(\frac{V}{V_{ref}}\right)^2$$

Where F_{table} is the force extrapolated from the property file and V_{ref} is the reference speed used in the property file.



The point of application of the front downforce was considered as the center of application of the front wing's downforce, while that of the rear downforce was considered as that of the rear wing. The downforce contribution due to sidepods was divided in front and rear based on the distance from the two points. The drag was instead applied at the center of mass of the vehicle.

The parameters that must be defined and their values are indicated in the table below, while the .aer property file is reported in the appendix.



All the relevant parameters were obtained through CFD simulation of the vehicle. Since the 2018 model could not be tested in the wind tunnel, but the 2017 model had been tested, the difference From CFD results and wind tunnel testing was considered: The 2017 car was found to produce on average 25% less downforce than the CFD values, so the values obtained from simulation for the 2018 car were reduced of the same amount.

@60kph	L[N]	D[N]	E	Lift vs. SC17 [%]
Rear wing	210	113	1.86	+21.4%
Front Wing	181	40	4.53	+23.1%
Sidepods	80	18	4.44	-15.8%
Body	165	40	4.13	+85.4%
Total	636	220	2.89	+28.0%

Brakes

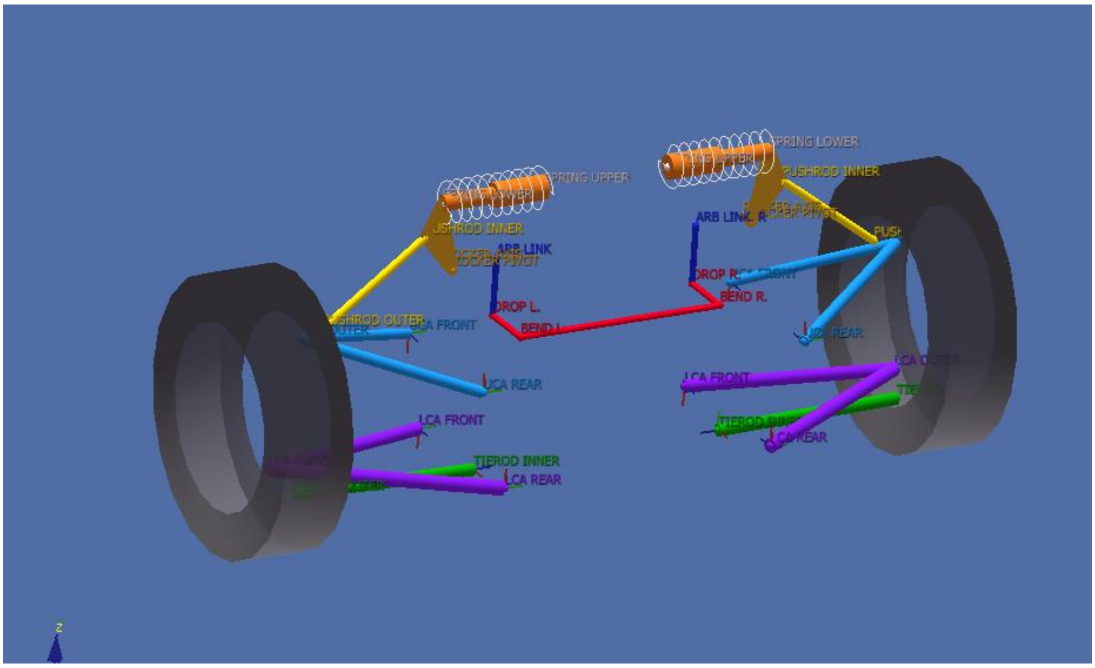
The software allows the modeling of a four-wheel disk brake configuration, that is the same found in our vehicle. A limited number of parameters allows to describe a complex subsystem with good accuracy. Those parameters and their values are reported below. As can be seen from the variables below, lockup is modeled by a 1 DOF spring-damper system. All values were either design values or available from the manufacturer's datasheet, thus easily obtainable.

- **Front Bias:** Percentage of braking system pressure going to front wheels.
- **Master cylinder pressure gain:** Constant relating the driver brake demand to master cylinder pressure.
- **μ:** Brake pad-brake disk friction coefficient
- **Effective piston radius:** Radius where braking force is applied
- **Piston area:** Area of caliper piston
- **lockup_natural_frequency:** Natural frequency of the spring-damper model used for lockup
- **lockup_damping_ratio:** Damping of the model used for lockup
- **lockup_speed:** Speed for the lockup model

Variable	Value	Unit
Front bias	0.7	%
Master cylinder pressure gain	0.1	N/mm^2
μ	0.4	/
Effective piston radius	180.0	mm
Piston Area	2300.0	mm^2
Lockup damping ratio	1	/
Lockup natural frequency	10.0	Hz
lockup speed	139	mm/s

Front and Rear Suspension

For the modeling of suspension components, the integrated Vi-SuspensionGen was utilized. It fundamentally works like a simplified ADAMS car suspension kinematics simulator, the hardpoints of the suspension are entered in a table and a simulation is performed, yielding characteristic kinematic curves that are used by the model. The front suspension model also includes the tierod hardpoints, so it is possible to also extract curves related to the steering subsystem in this phase. There is also the possibility to perform elasto-kinematic simulations taking into account the compliance of joints and bushings. Only the kinematic curves were considered because all joints in the car are considered rigid as there are no elastomeric element (not needed in a race car).



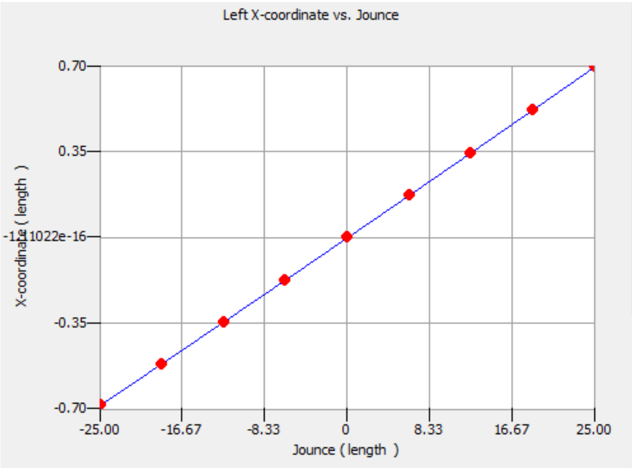
After the kinematic curves have been extracted, they must be inserted in the model together with spring, damper and anti-roll bar characteristics.

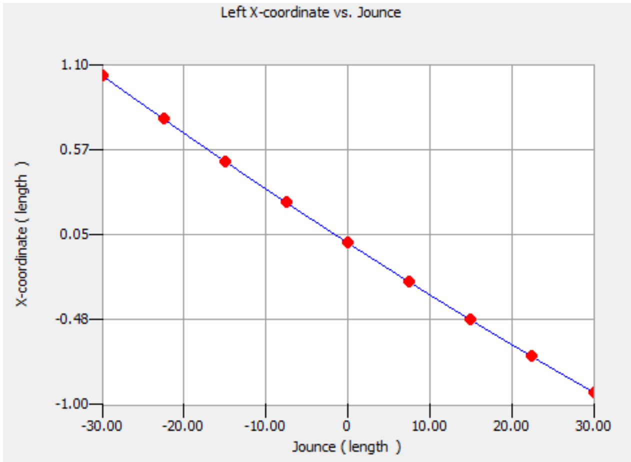
Curves will be shown for every category. Only curves regarding the left wheel will be reported for the sake of brevity, as the curves are symmetric. The left image will regard the front suspension and the right one the rear suspension.

Wheel location

The relevant data for this category is:

1. **Wheelbase:** The design value in static trim was considered. The value is 1200 mm
2. **X coordinate variation with vertical motion:** The previously obtained curve is represented in the graph below:

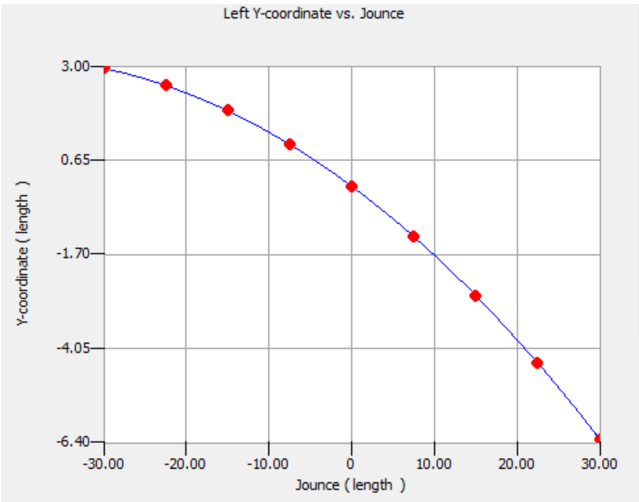


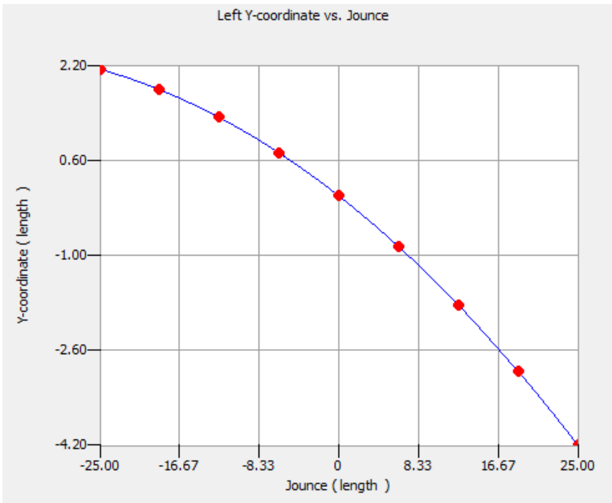


Front Suspension

Rear Suspension

3. **Y coordinate variation with vertical motion:** The previously obtained curve is represented in the graph below:





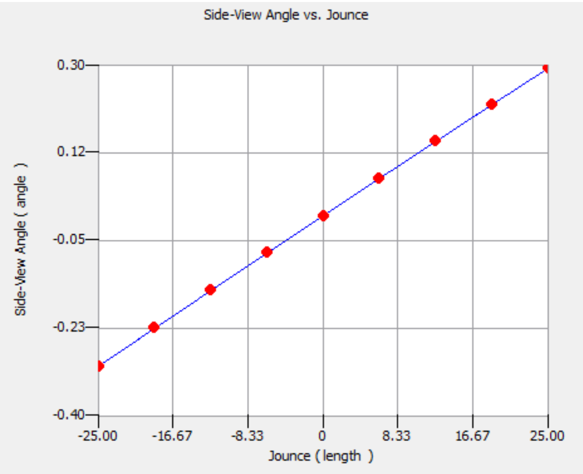
Front Suspension

Rear Suspension

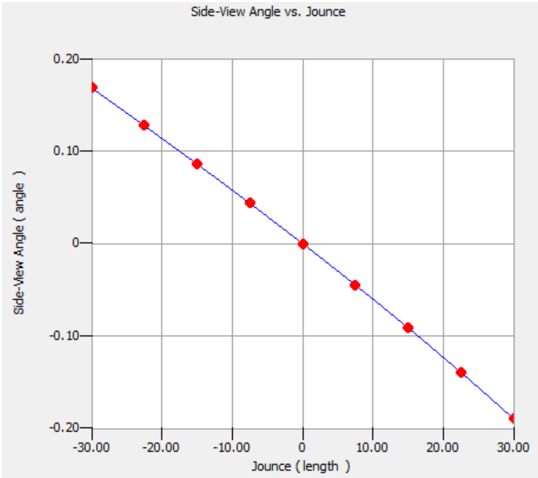
Wheel orientation

In this section the curves regarding the characteristic angles of the wheel, obtained through kinematic analysis, are reported. In particular, the data is:

1. **Side view angle vs vertical motion of the wheel:**

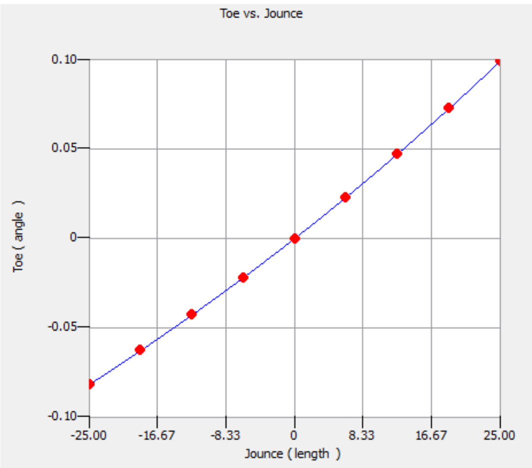


Front Suspension

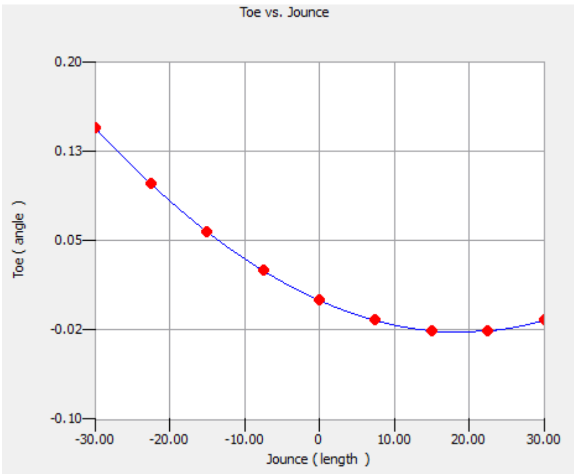


Rear Suspension

2. **Toe vs vertical motion of the wheel:**

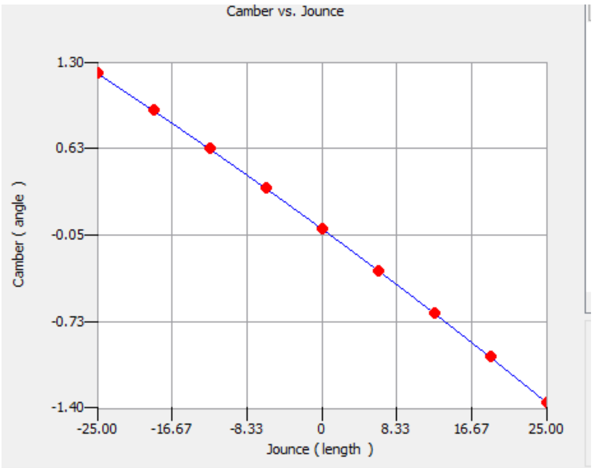


Front Suspension

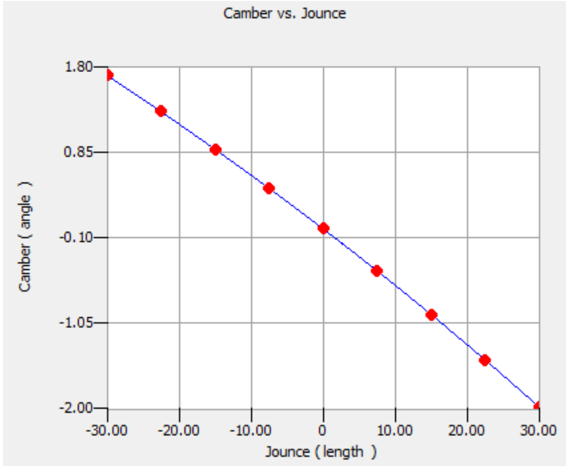


Rear Suspension

3. **Camber vs vertical motion of the wheel:**



Front Suspension



Rear Suspension

Springs

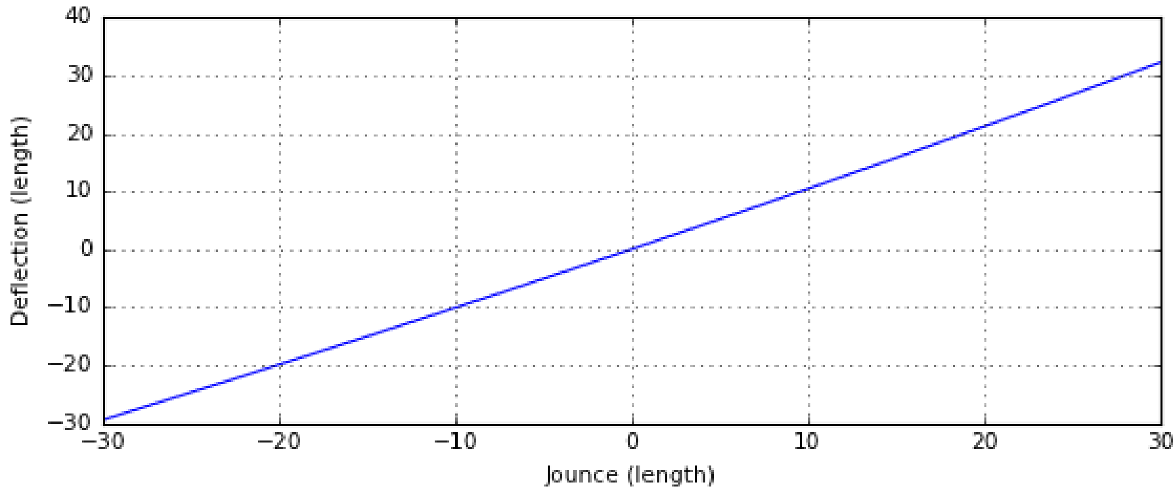
This section contains all parameters of the springs and

1. **Spring Data:** Springs can be modeled in the program by utilizing property files. A property file for our type of springs was not readily available, however it is easily created by knowing the stiffness and free length of the spring, data that can easily be measured or found in the datasheet. The data, along with the installation length required for preload and static height calculation and is reported in the table below:

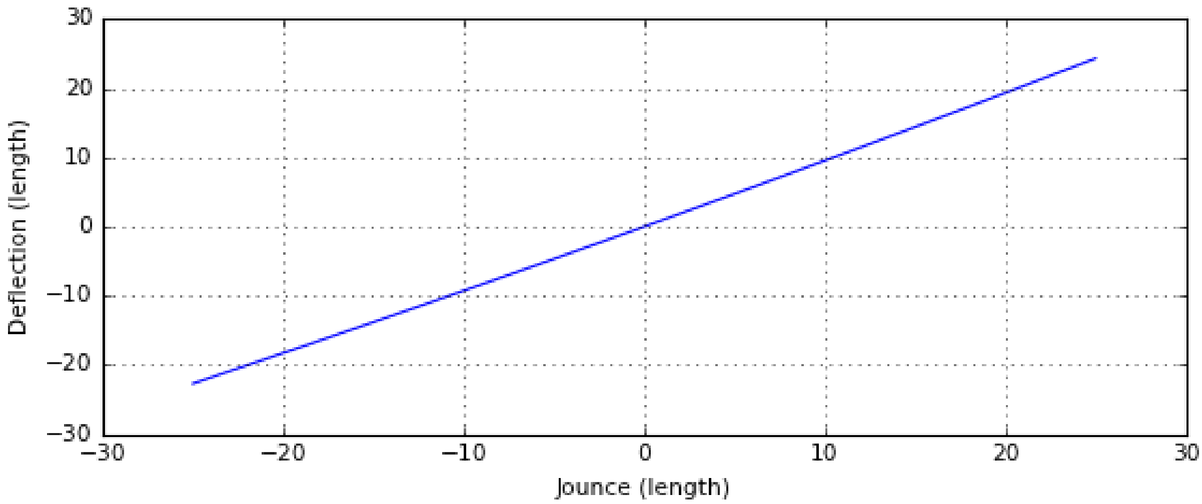
Data	Front	Rear	Unit
Stiffness	35	35	N/mm
Free length	125	125	mm
Installed length	112.8	115	mm

2. **Compression Ratio:** The motion ratio between wheel motion and spring deflection is reported in this section. The graph, obtained from kinematic analysis, can be found below:

- **Spring deflection vs jounce:**

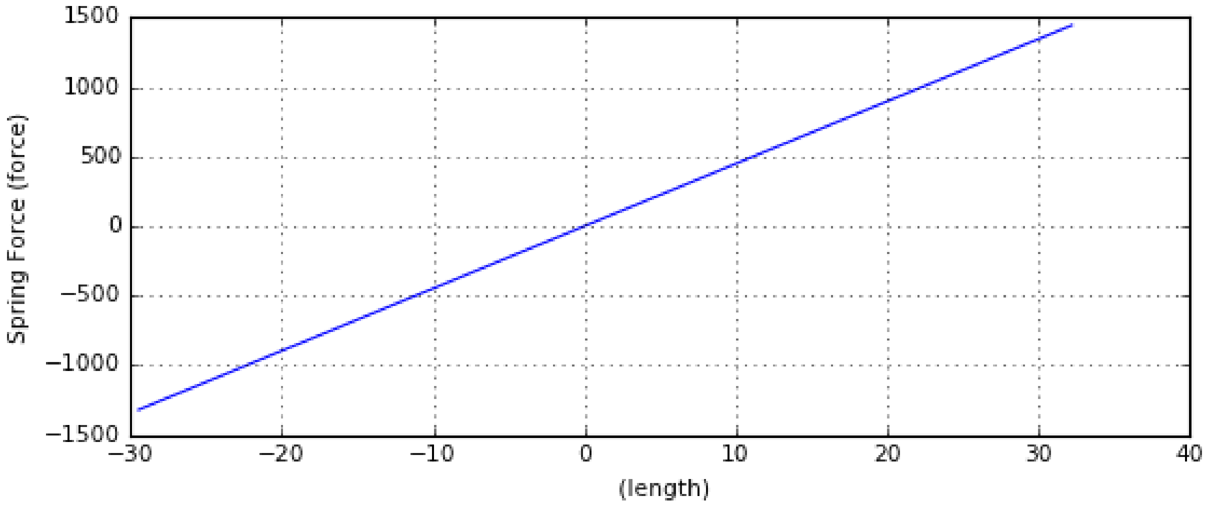


Front suspension

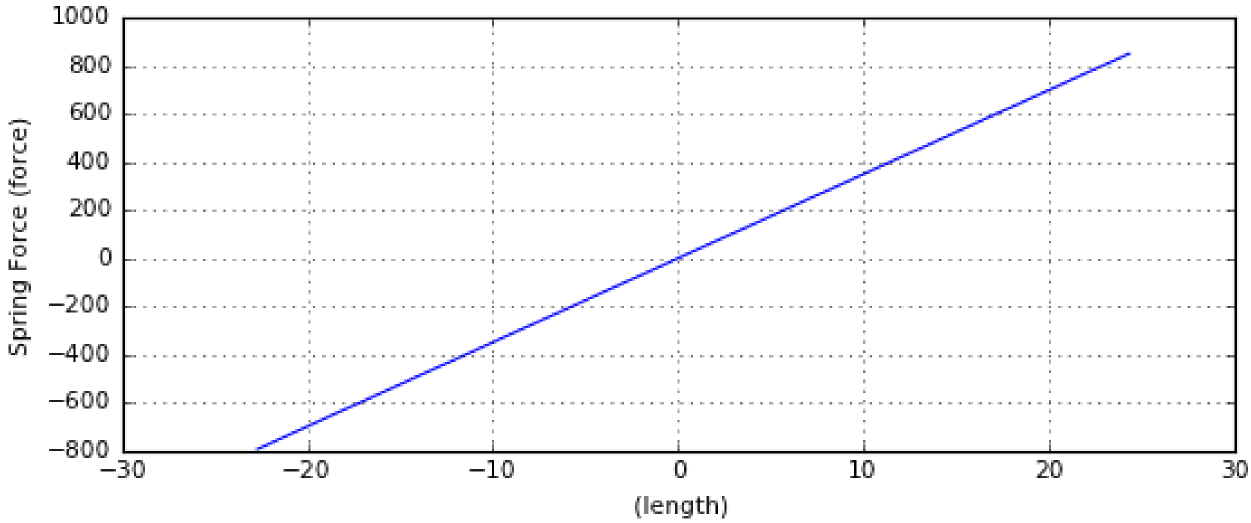


Rear suspension

- **Spring force vs jounce:**



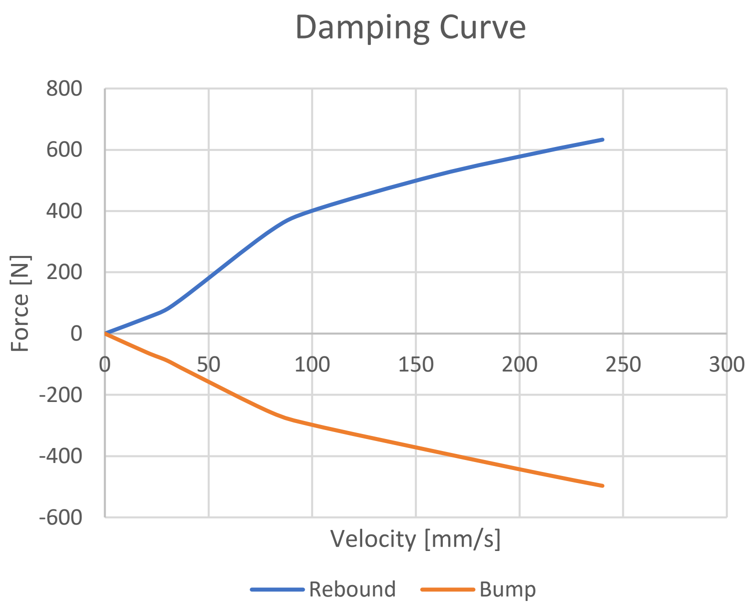
Front Suspension

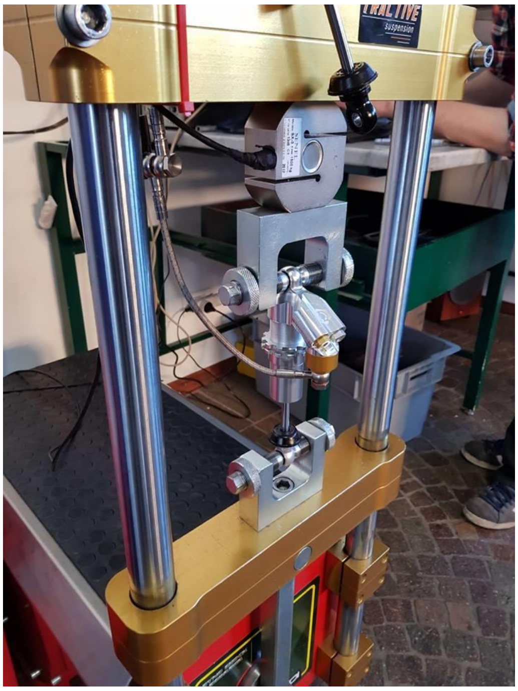


Rear Suspension

Dampers

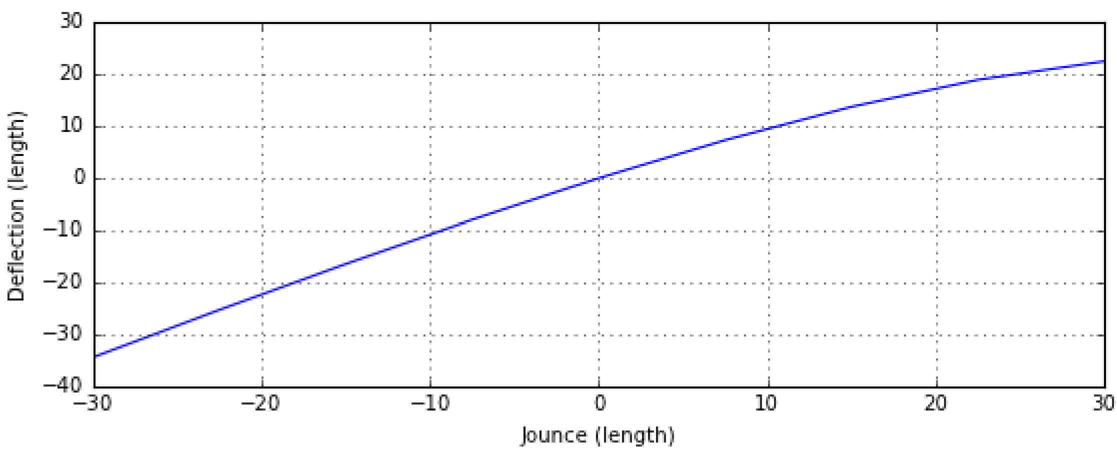
The dampers equipped on the vehicle are Ohlins FSAE TTX 25 with adjustable valves for bump and rebound at high and low speed. The damping curves are available in datasheets, however, to obtain more accurate results the dampers were tested on a rig. The resulting curve from these tests can is shown below:

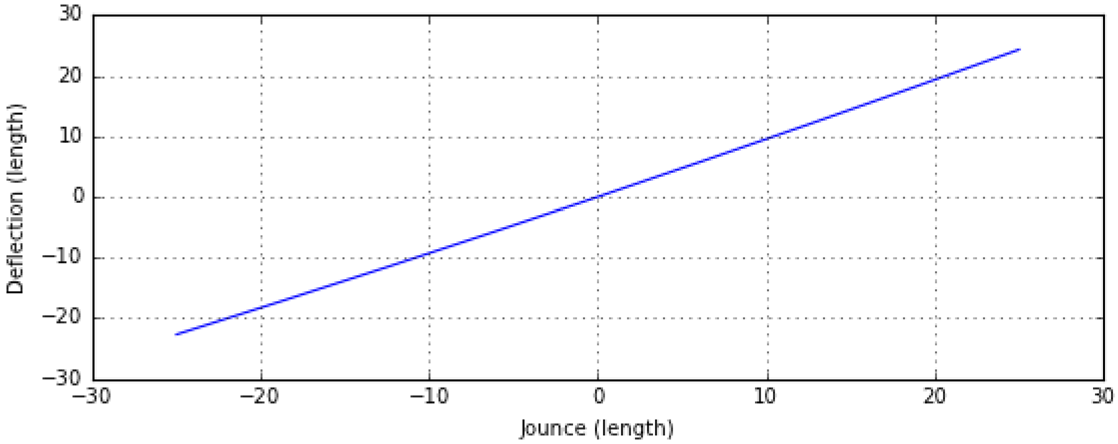




- **Damper deflection vs jounce:**

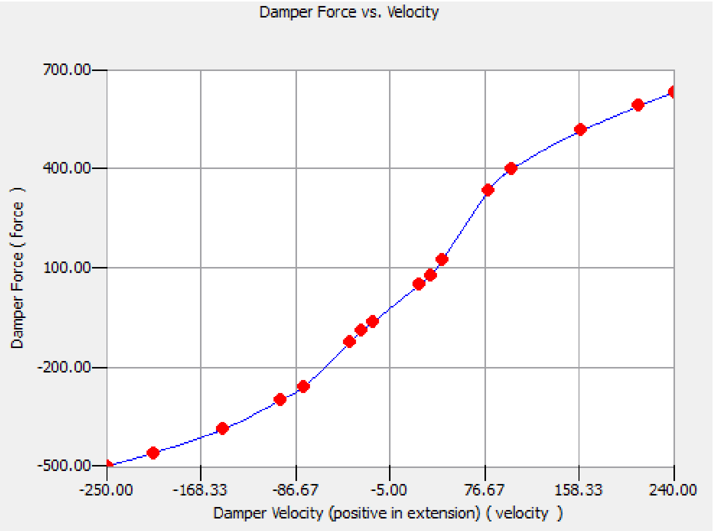
Front Suspension



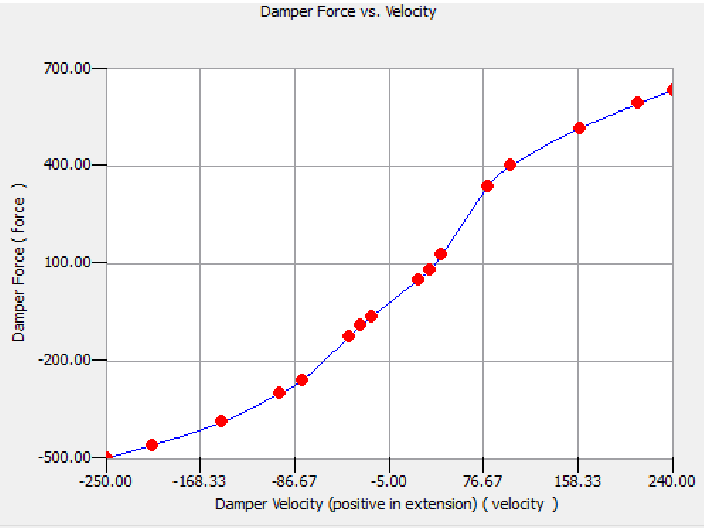


Rear Suspension

- **Damper force vs jounce velocity:**



Front Suspension



Rear Suspension

Anti-Roll bar

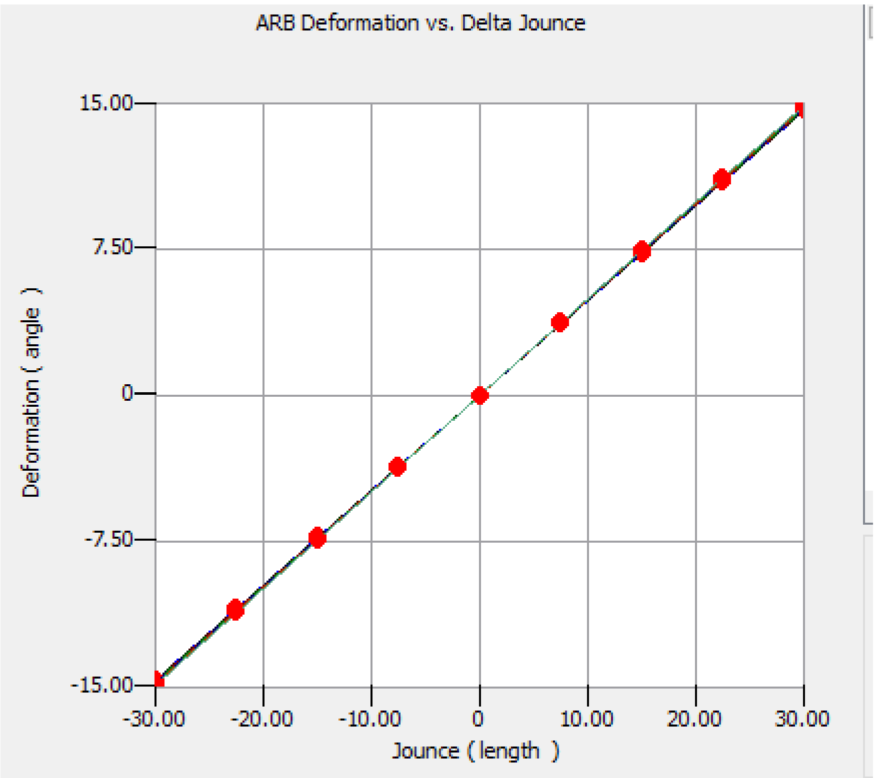
The vehicle's anti-roll bar is an original design of the team: to achieve the very low stiffnesses needed for a car weighing only 200 kg it uses small fiberglass "knives" that are loaded in bending. The system is illustrated in the figure below.

The "knives" of various thicknesses were analyzed through FEM and then bench tested to obtain stiffness values. To increase the stiffness two options are available: first, the "knives" can be turned to change the bending inertia of the system. If this is not enough, thicker "knives" can be used. The stiffnesses obtained are not linear but vary with displacement, however considering that the suspension movement is quite limited and the transmission ratio is also quite low (around 0.6), the stiffness could be considered linear for this range.

The Anti-roll bar was modeled in the program through the use of the Vi-SuspensionGen tool: while this design of antiroll bar is not available from the preset it was substituted with a torsion bar with the same effective stiffness and motion ratio. The curves are then extrapolated as for the rest of the subsystem.

The relevant parameters are:

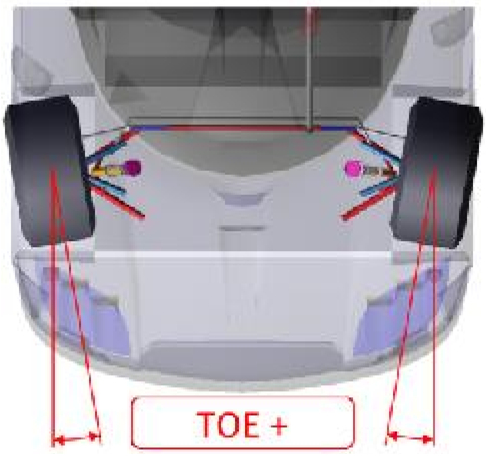
- **Deformation vs Vertical motion of the wheel:**

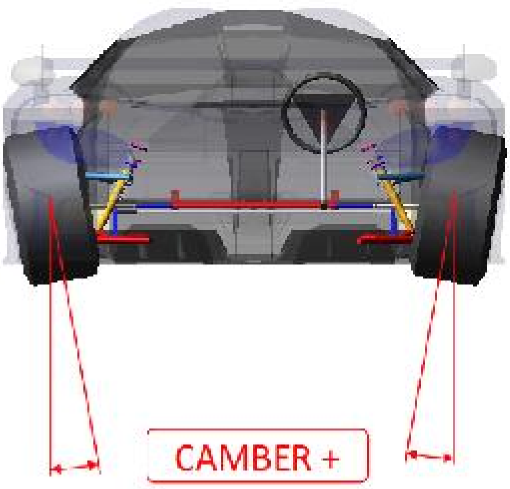


- **ARB Stiffness:** a value of 2000 Nm/rad was chosen to obtain an effective stiffness at the wheel of 12 N/mm, equal to that of the real ARB

Suspension set-up data:

In this category, all setup parameters of the suspension subsystem are reported. The most important ones are the static angle setup, reported below:





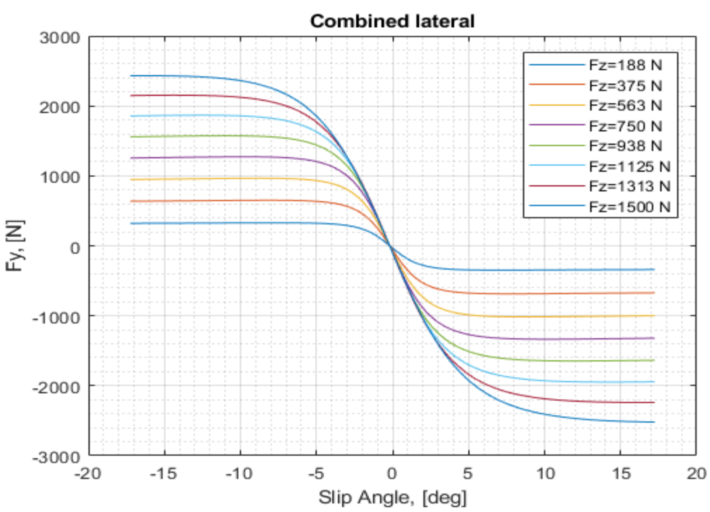
Data	Front	Rear	Unit
Toe	-0.55	0.55	°
Camber	-1.5	-1.75	°

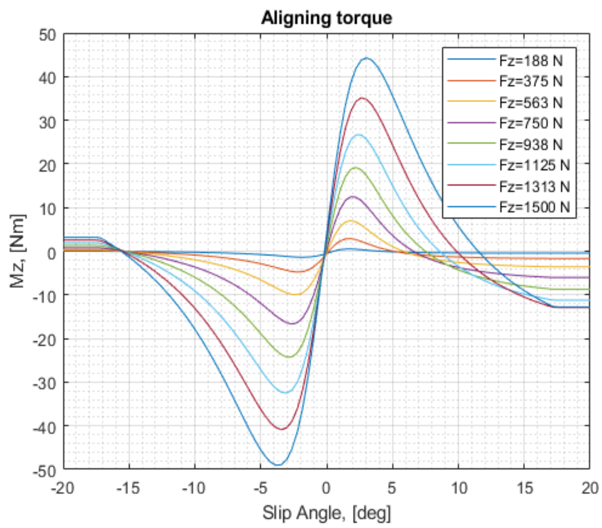
Wheels

The wheels subsystem, divided in front and rear, contains information about the unsprung masses, inertias and tire properties of the wheels. The main parameters are:

Tire property file

For accurate results, a complete tire property file with pacejka parameters must be used. In our case, the tire in use was a Formula SAE 13" low profile tire made by Pirelli. The tire is the same for front and rear wheels. The .tir file was provided by Pirelli and provides accurate estimation of tire forces for most driving conditions. Plots of the lateral force vs slip and Aligning torque vs slip characteristics of the tire at various vertical loads can be found below:





Masses and Inertias

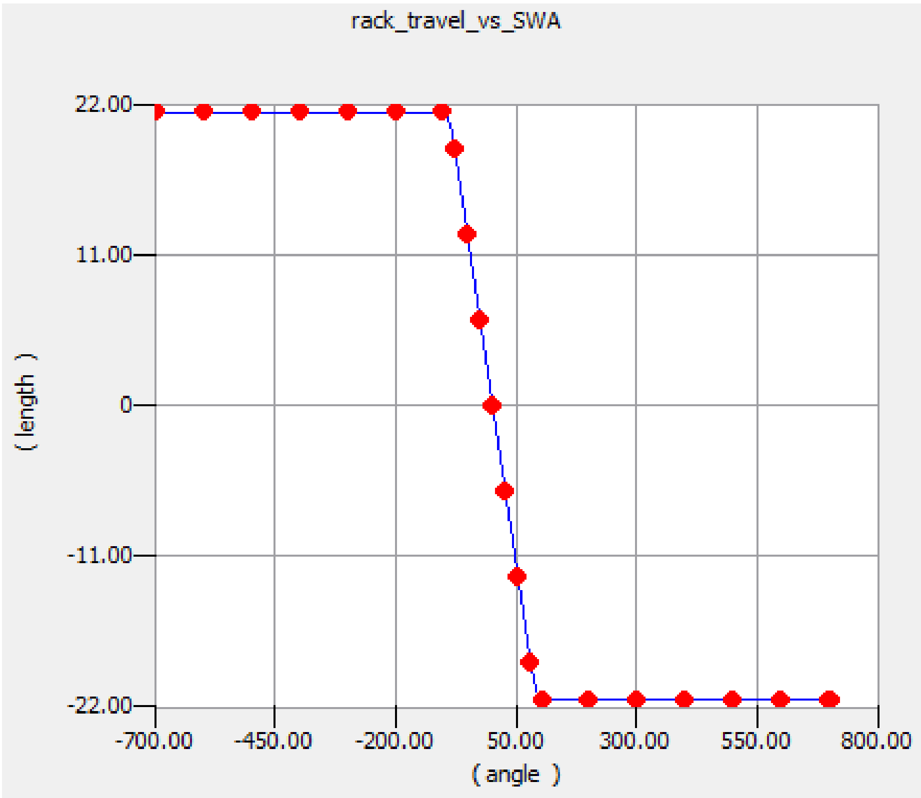
The various masses and inertias of the wheel subsystem are reported in the table below:

Data	Front	Rear	Unit
Spin Inertia	187500	mm	$kg * mm^2$
I_{xx}	228000	mm	$kg * mm^2$
I_{yy}	230000	mm	$kg * mm^2$
I_{zz}	368000	mm	$kg * mm^2$
Unsprung mass	16.24	kg	kg
Wheel center height	237	$kg * mm^2$	mm

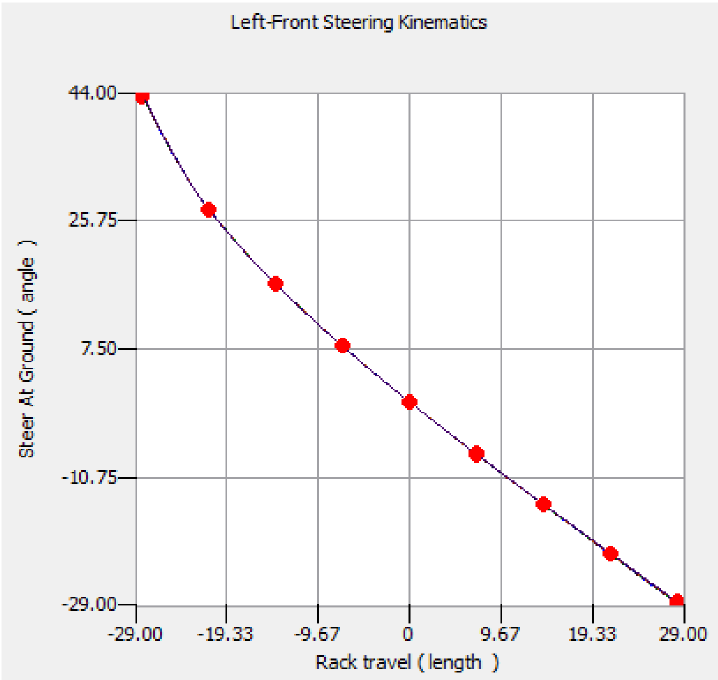
Steering

As for the suspension subsystem, there is no physical representation of the subsystem, instead, curves extracted from a kinematic simulation with the SuspensionGen tool are used as lookup tables during simulation. The relevant curves are:

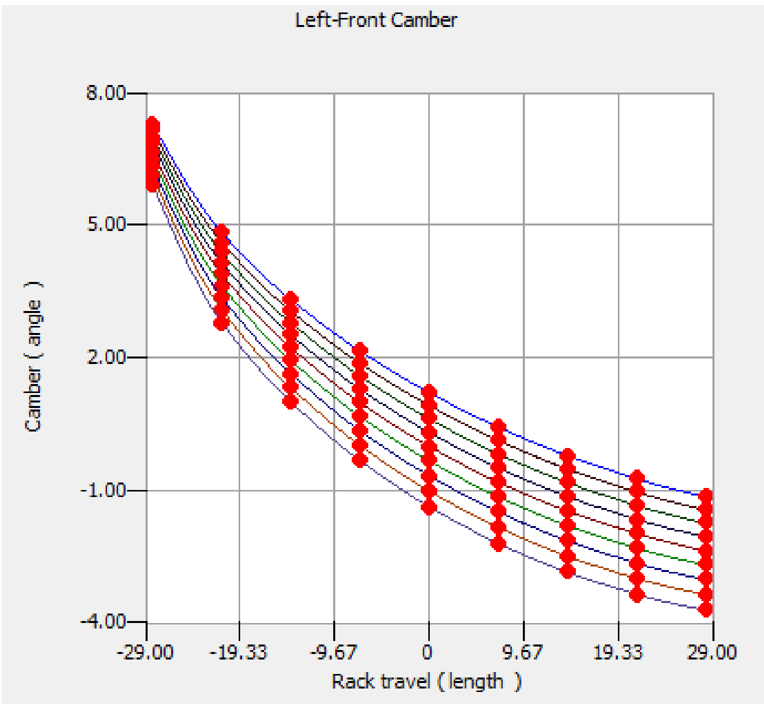
- ***Rack travel vs Steering wheel angle:*** This curve reports the rack travel for each positon of the steering wheel. Once the rack travel is knwown, the kinematic curves for each wheel can be calculated



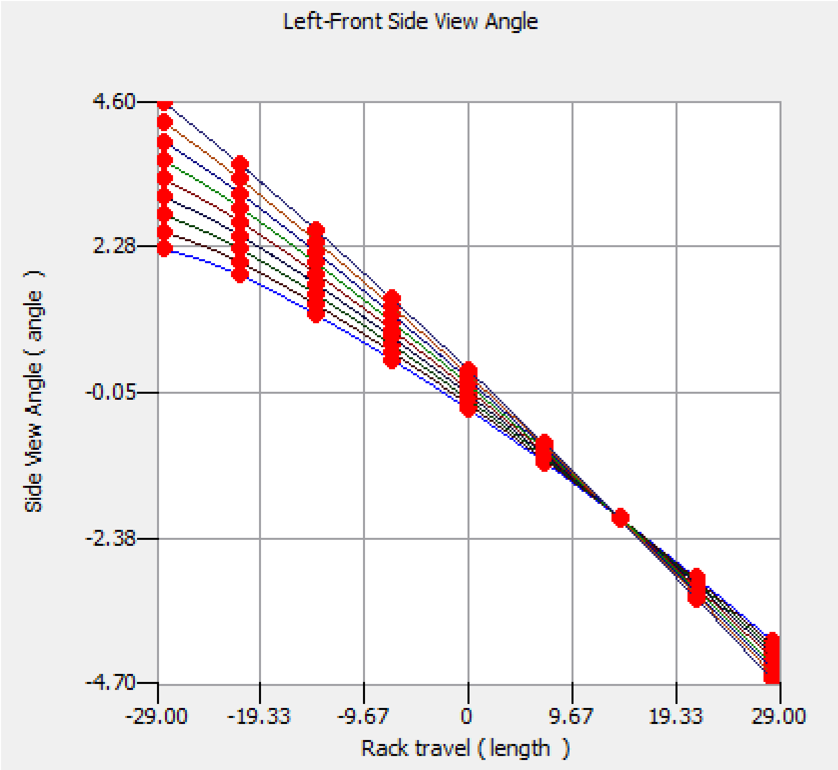
- ***Steer at ground vs rack travel***: Once the rack travel has been established from the previous curve, this lookup table is used to determine the steering angle of each wheel.



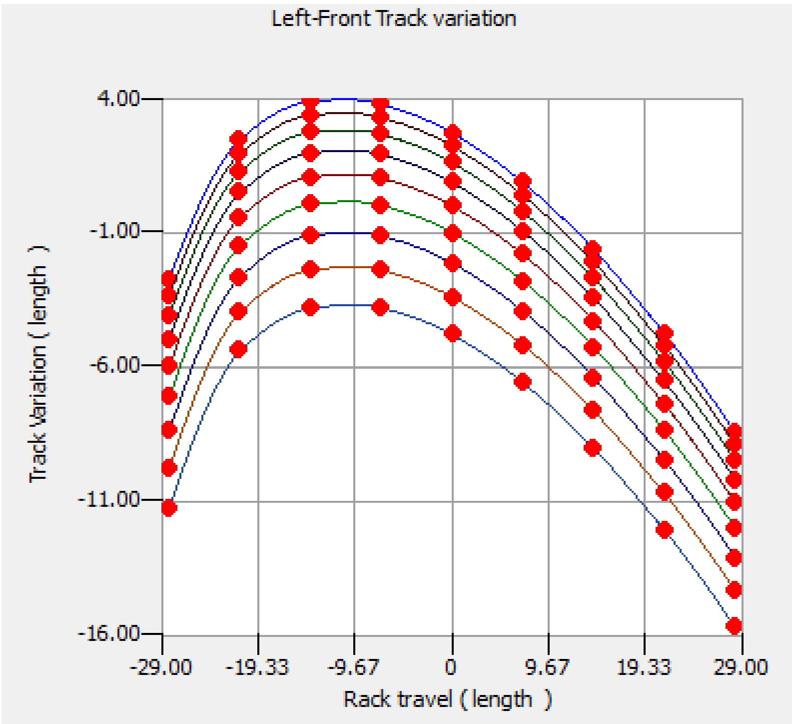
- ***Camber angle vs input steer and Jounce:*** These curves are used to calculate the variation of camber angle for every steering condition, also taking into account the vertical motion of the wheel. Each curve is for a different value of vertical motion of the wheel.



- ***Side view angle vs input steer and Jounce***: This graph contains the curves relating the side view angle to input steer and vertical motion of the wheel. Again, each curve is for a certain value of vertical displacement, from -25mm to 25mm.

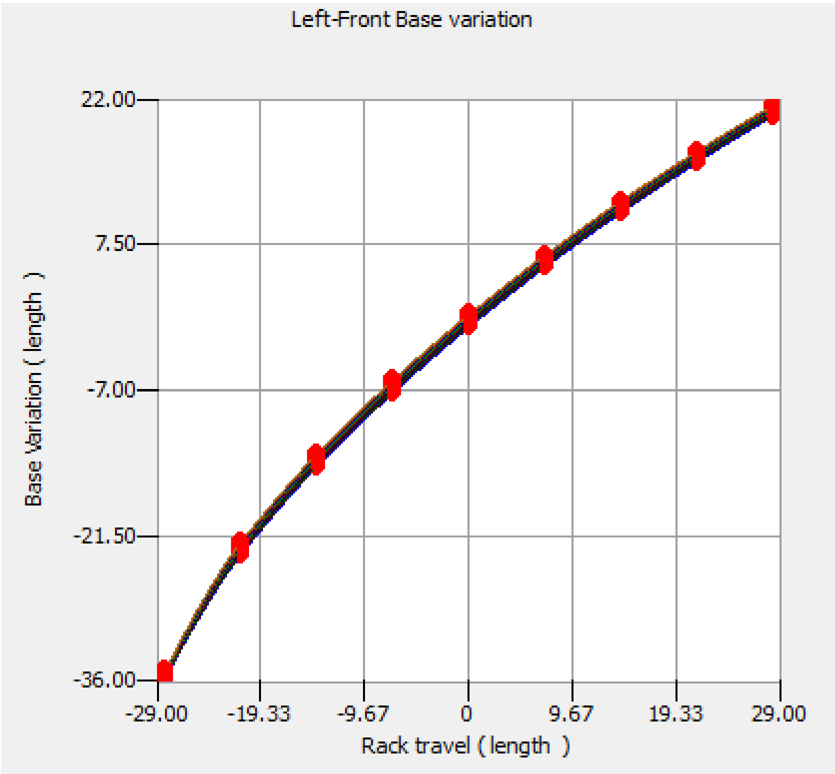


- ***Track variation vs input steer and Jounce***: This graph contains the curves relating the track variation (displacement of the wheel along y)to input

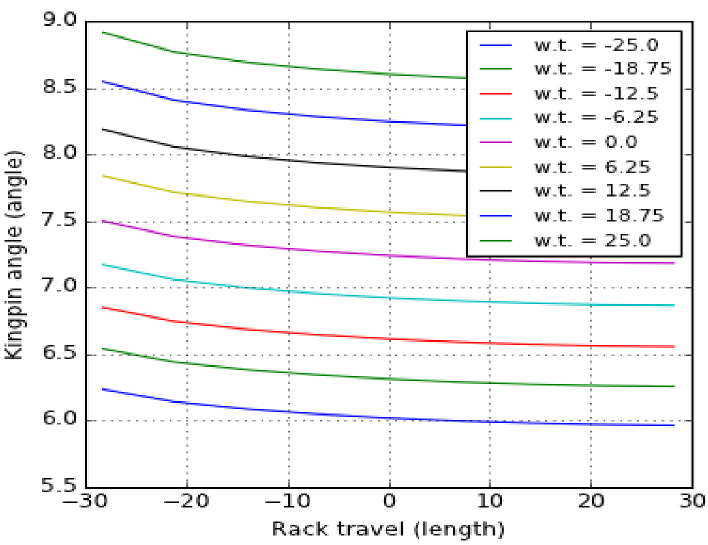


steer and vertical motion of the wheel.

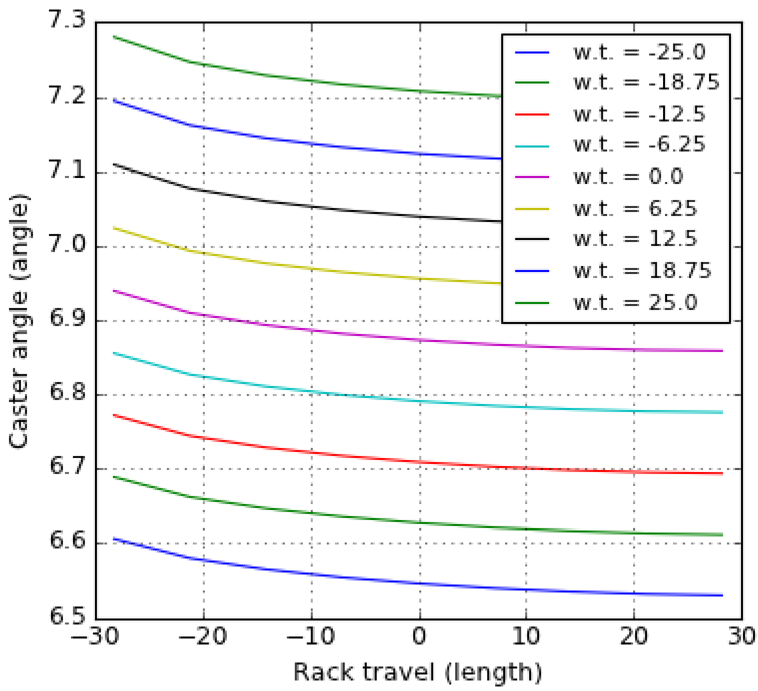
- ***Wheelbase vs input steer and Jounce***: This graph contains the curves relating the track variation (displacement of the wheel along x) to input steer and vertical motion of the wheel. Again, each curve is for a certain value of vertical displacement, from -25mm to 25mm.



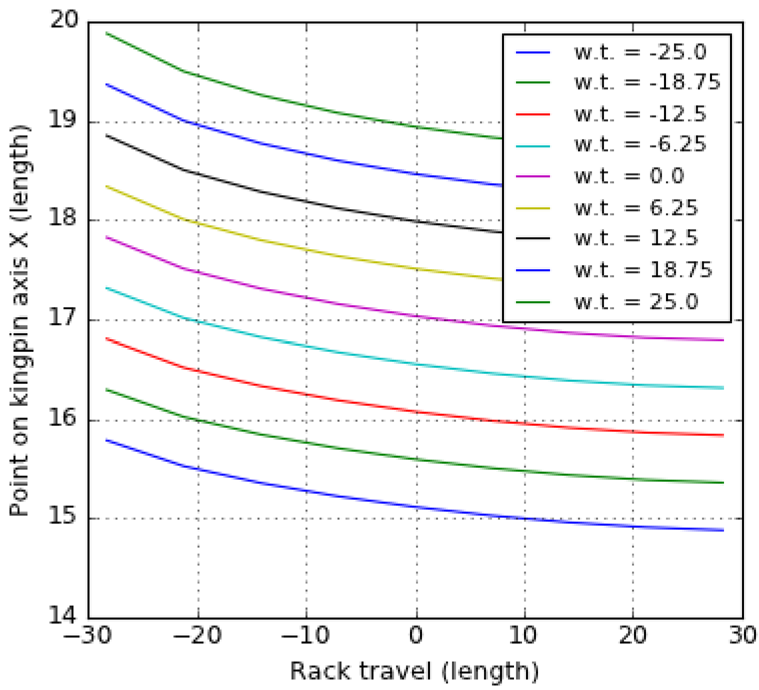
- ***Kingpin angle vs rack travel vs wheel travel***: This graph contains the curves relating the kingpin angle variation to input steer and vertical motion of the wheel. Again, each curve is for a certain value of vertical displacement, from -25mm to 25mm



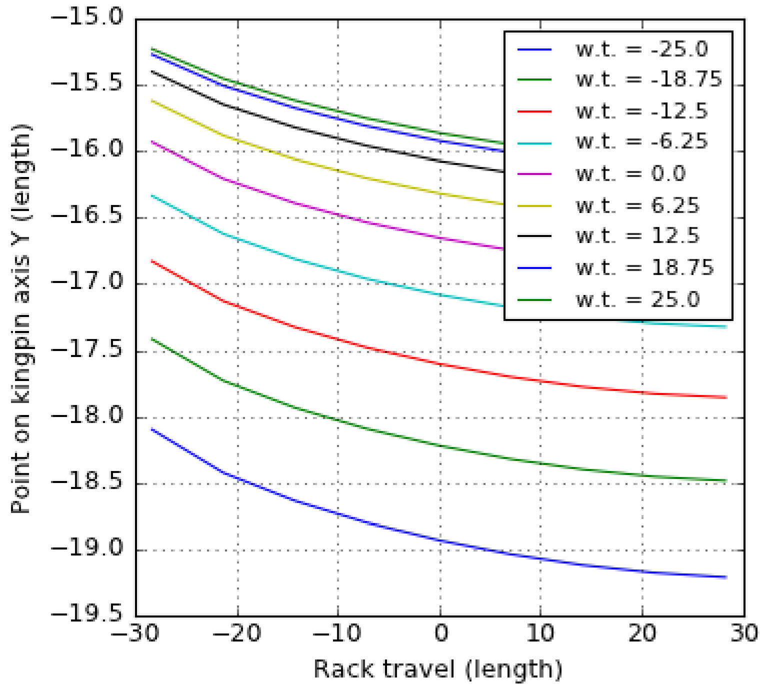
- ***Caster vs rack travel vs wheel travel***: This graph contains the curves relating the caster angle variation to input steer and vertical motion of the wheel.



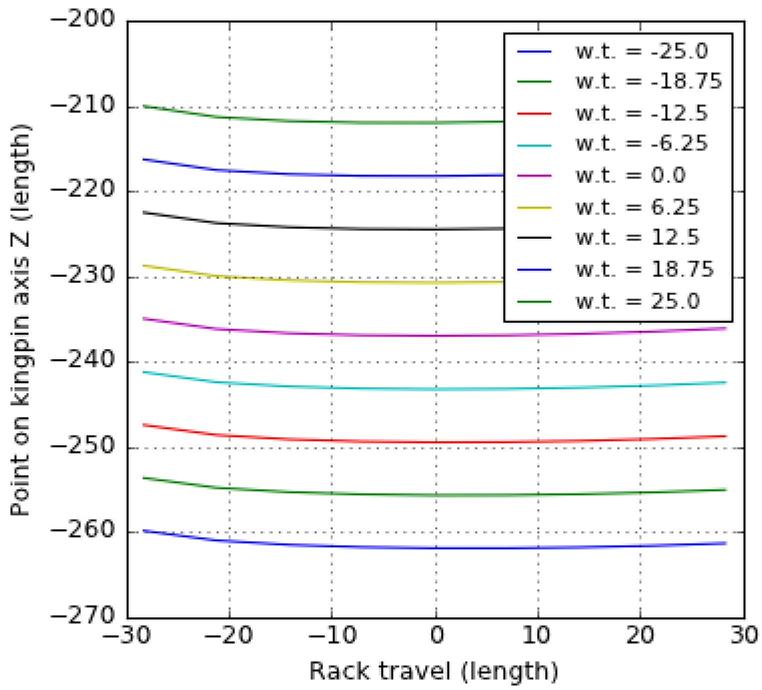
- ***Kingpin axis X position vs rack travel vs wheel travel***: This graph contains the curves relating the variation of the x coordinate of the kingpin axis to input steer and vertical motion of the wheel.



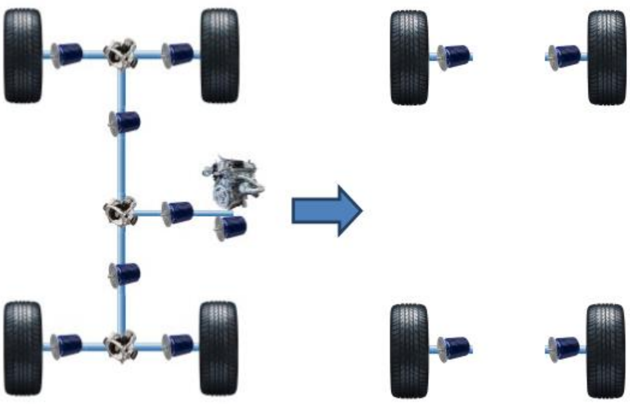
- ***Kingpin axis Y position vs rack travel vs wheel travel***: This graph contains the curves relating the variation of the y coordinate of the kingpin axis to input steer and vertical motion of the wheel.



- ***Kingpin axis Z position vs rack travel vs wheel travel***: This graph contains the curves relating the variation of the z coordinate of the kingpin axis to input steer and vertical motion of the wheel.



Powertrain



Lastly, the powertrain should be modeled. The configuration used by our car is 4WD with 4 independent motors mounted on-wheel. This configuration can be modeled in the software simply by removing the central engine, gearbox, differential and transmission, typical of a traditional combustion engine vehicle, and inserting the on- wheel motors. Next, the torque map and other data must be inserted into the model. The four engines are all identical, so for the sake of brevity only the data about the front left motor will be presented.

General data

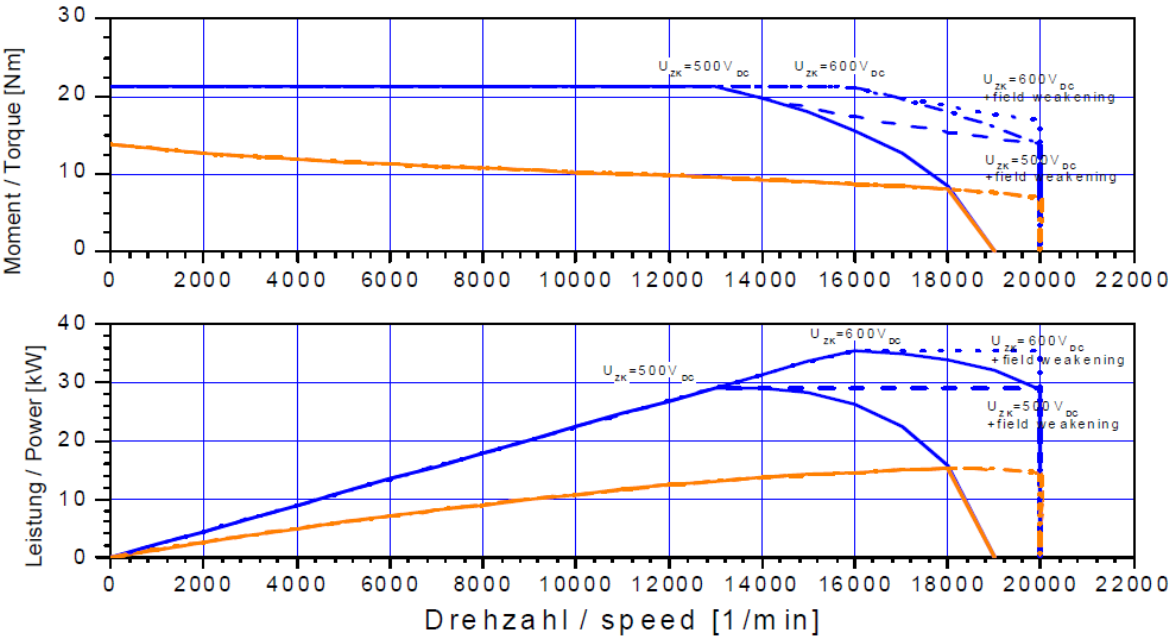
The general data about the motors can be found in the table below:

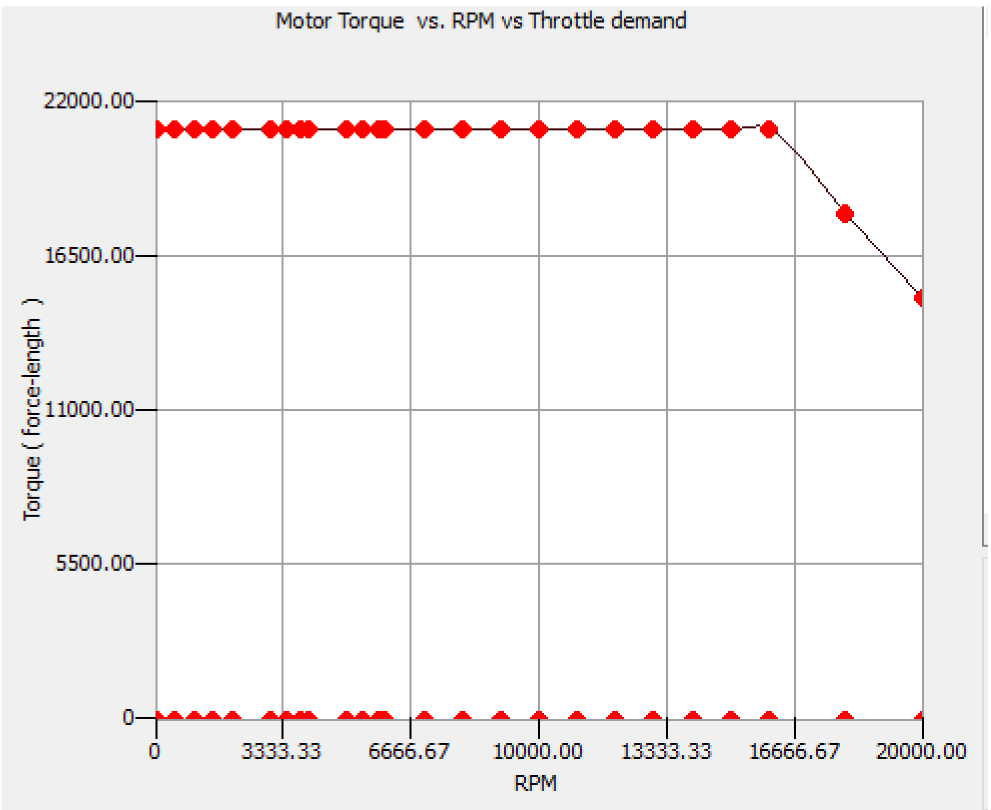
Data	Value	Unit
Inertia	274	$kg * mm^2$
Efficiency	1	
Transmission ratio	14.82	
Maximum RPM	20000	RPM

The inertia and maximum RPM of the motors was available from the datasheet of, the efficiency is considered equal to 1 because the provided torque map is already corrected for efficiency while the transmission ratio is the design value of the redactor gear.

Torque map

The torque map was taken from the datasheet of the motors. Both the map from the datasheet and the model map can be found below:



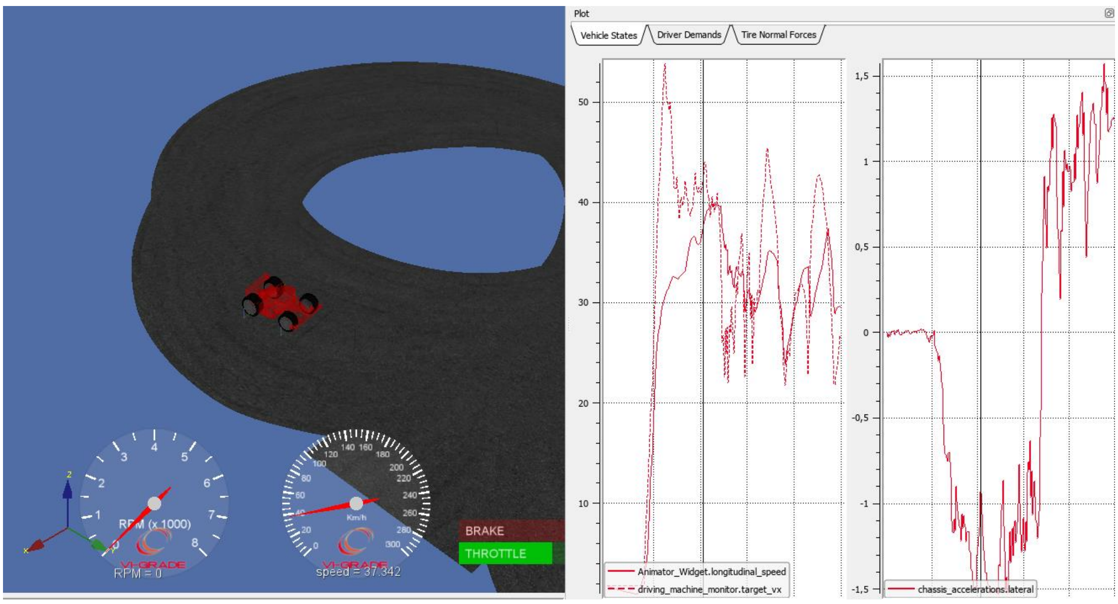


Model Torque map

One consideration has to be made about the torque map: the control system uses a variable torque distribution depending on the current load on each wheel, usually the fraction of the torque going to the front wheels is less than that going to the rear ones, the only way to make the model behave this way would be to use different torque maps for front and rear motors. Furthermore, the torque map would have to be modified to comply with the power limit of 80 kW, with each power distribution couple having different cut points and overall curves. Since it would still be difficult to approximate a variable torque distribution with a fixed curve, so it was chosen to use only the physical limits of the motors as torque maps and to implement control systems in Co-Simulations to have more accurate approximations of the real limits of the car.

3.3 Co-Simulation environmen

In addition to simulations performed completely in the CarRealTime environment, the program can be interfaced with other softaware for Co-Simulations. Among those, it is possible to use the CarRealTime solver in Simulink environment in order to perform Co-Simulations. In this case the CarRealTime solver receives various signals as input from Simulink at each time step and uses them to calculate outputs that are again passed to Simulink for Post processing or calculations. This Co-Simulation environment is very flexible and allows to model additional subsystems not included in the base model, deactivate internal subsystems and use user-modeled ones and, as is the case for this dissertation, to feed data logged from the real vehicle to the model in order to validate the behavior of the vehicle.



Chapter 4

Preparatory work for Validation

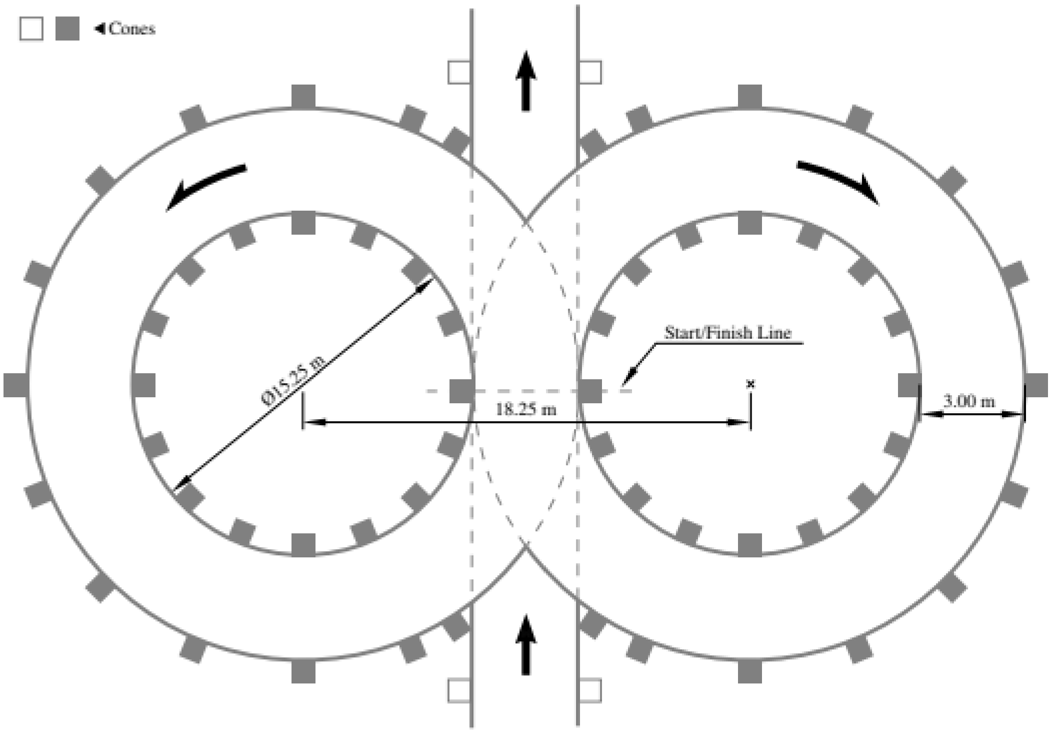
4.1 Choice of maneuvers for validation

Simulation models are fundamental tools in the design of any product, but any model has its limits, and these must be known to make the most out of every simulation. Without knowing these limits, only a qualitative assessment of the performance of a given solution can be obtained through simulation. Meanwhile, if a model is validated and found to be accurate in various situations simulations can become a much more powerful tool, giving adequate quantitative results.

The first step for the validation process is to decide what maneuvers to use when validating it. Since using complex maneuvers from the start could lead to trouble in finding the cause of the lack of fit between model and vehicle it was chosen to proceed in three steps: first the longitudinal behavior of the vehicle should be validated, then the lateral behavior and finally an event with both longitudinal and lateral maneuvers should be used to stress-test the model.

The first chosen maneuver was an acceleration event with strong braking at the end, testing the modeling of the longitudinal behavior of the vehicle in isolation. The acceleration course used had the same characteristics as the official one of any Formula SAE event, namely the course is a straight line with a length of 75 m from starting line to finish line. The course is at least 5 m wide.

The next step would be to achieve a good fit in lateral behavior of the model, so a skidpad simulation was chosen as the benchmark. Again, the track used had the same characteristics as that of an official Formula SAE event, namely: the course consists of two pairs of concentric circles in a figure of eight pattern, the centers of these circles are 18.25 meters apart , the inner circles are 15.25 meters in diameter, while the outer ones have a diameter of 21.25 meters. This gives an average turn radius of 9.125 meters and a track width of 3 meters.



Finally, an Autocross event simulation was chosen to validate the combined behavior of the model. However, such a maneuver could prove difficult to validate using only logged data as input, due to the length and complexity of the maneuver. So, this simulation will be run using the software's driver model with the trajectory as the only input.

Once again, the track used followed Formula SAE rules for Autocross events, whose requirements are:

- Straights: No longer than 80 m
- Constant Turns: up to 50 m diameter
- Hairpin Turns: Minimum of 9 m outside diameter (of the turn)
- Slaloms: Cones in a straight line with 7.5 m to 12 m spacing
- Miscellaneous: Chicanes, multiple turns, decreasing radius turns, etc.

- The minimum track width is 3 m.
- The length of the autocross track is less than 1.5 km.

4.2 Choice and creation of reference signals

Once the type of events for validation was decided, track tests were organized to record input signals to be fed to the model. The sensors and logging equipment used have been briefly presented in chapter 2, in this chapter the choice of signals and their extraction method will be explained.

The signals to be registered for the validation through Co-Simulation are divided in Inputs for the model and Outputs to be compared with the model's response.

Inputs

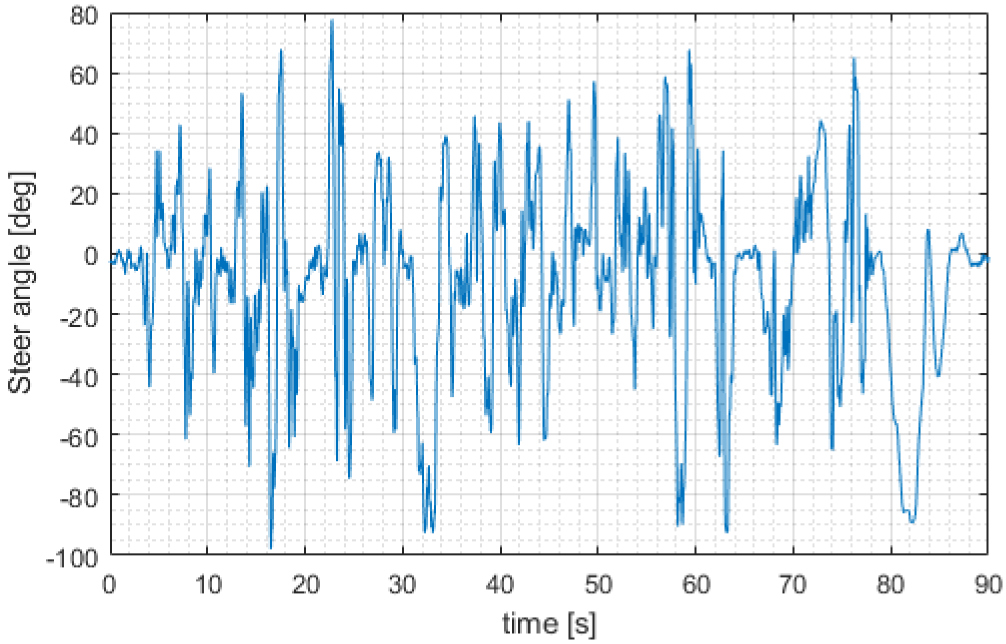
Steering wheel angle

Steering wheel angle was recorded from the previously mentioned rotation angle sensor integrated in the rack. As the model requires a rack displacement in mm as input, the rotation angle was converted into a linear displacement using the motion ratio of the rack. The conversion in this case is:

$$Delta_{rack}[mm] = SteeringAngle[deg] * \tau \, [\frac{\mathrm{mm}}{\mathrm{deg}}]$$

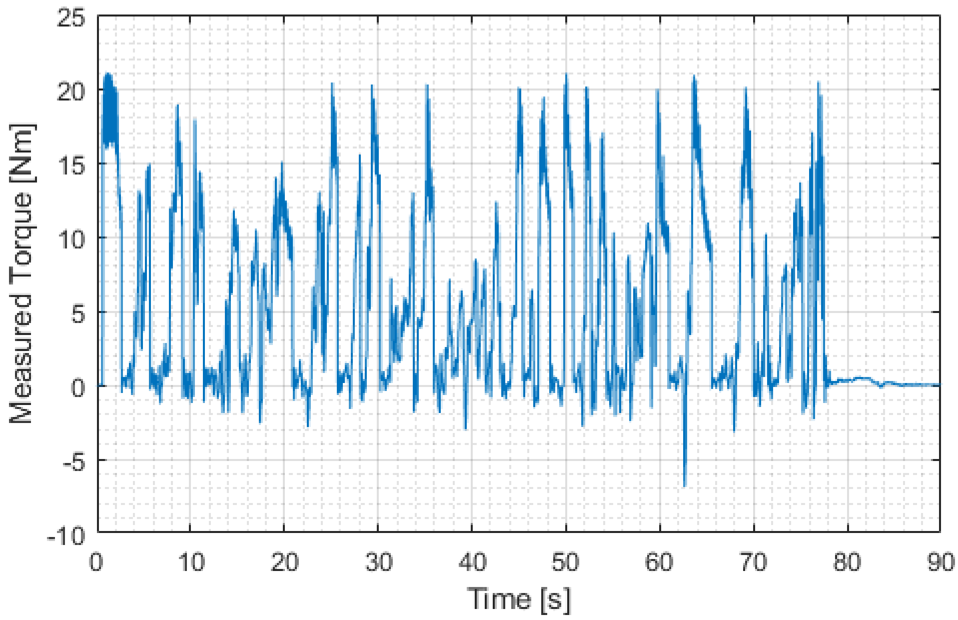
Where $\tau = 0.2383$ is the ratio between steering angle input and rack linear displacement.

A sample of the signal recorded during the Autocross simulation can be found in the image below.



Torque at the motors

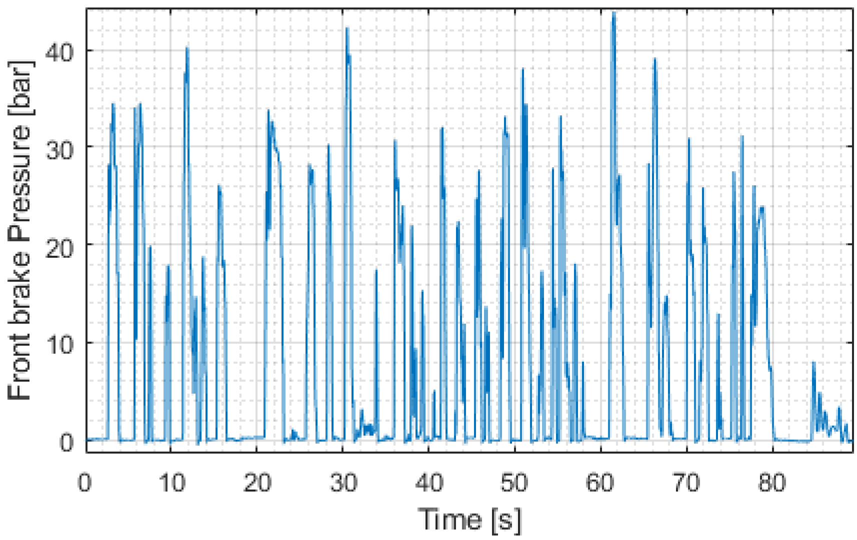
Since this thesis is focused on the validation of the overall dynamic behavior of the whole vehicle, the torque at each motor was chosen as input for the longitudinal behavior of the vehicle instead of the throttle pedal signal, as using the latter would also imply a validation of the powertrain of the model that is outside the scope of this thesis. To have a correct longitudinal behavior of the vehicle an accurate measurement of the torque provided by each motor in every moment must be known. Two signals were available to determine this: the torque requested in each moment by the ECU control logic and the feedback current registered by the inverters. The second signal was chosen as it was a more correct measurement: the requested torque could not be completely accurate because of lag between request and internal control achieving the required value, furthermore in some instants the requested value could not be reached due to physical and control constraints. Meanwhile it is easy to calculate the actual supplied torque at the motor by knowing the current: the datasheet provides a conversion factor for this purpose with correction factors for various operating conditions.



Brake line pressure

The next signal needed as input is the torque due to braking action. For this purpose, the signals from the front and rear brake line pressure sensors were used. There was no need to use data from the brake load cell and strain gage as regenerative braking torque was already taken into account by considering the motor torque. The needed input for the model was the Master cylinder pressure, this value was easily obtainable considering that the brake bias is fixed at 70% at the front. The torque was thus obtained with the equation:

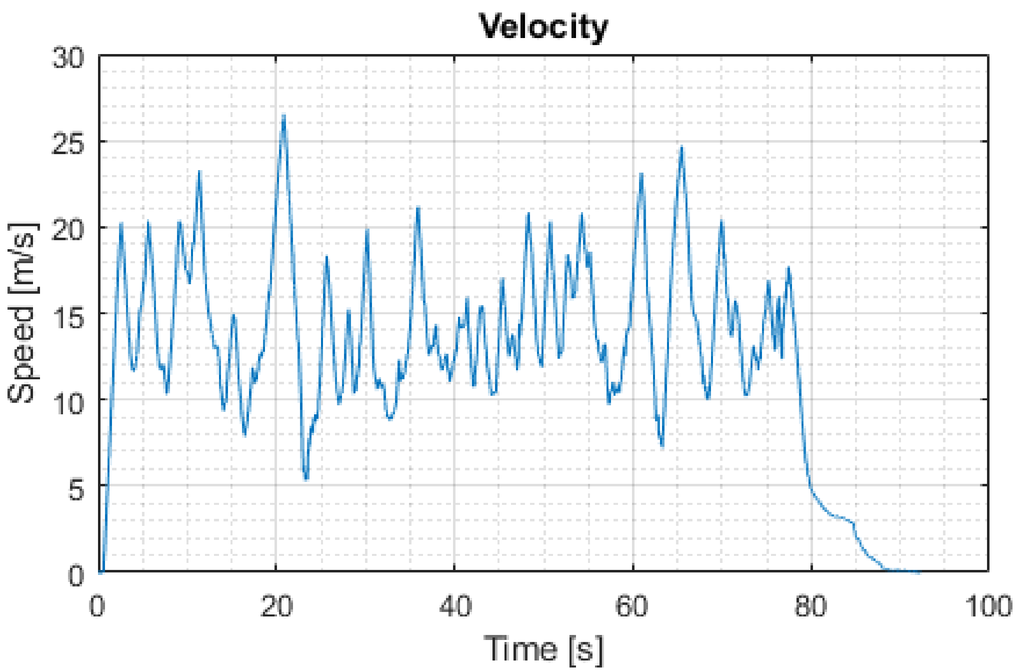
$$p_{master} = p_{front} * \frac{100}{70}$$



Outputs

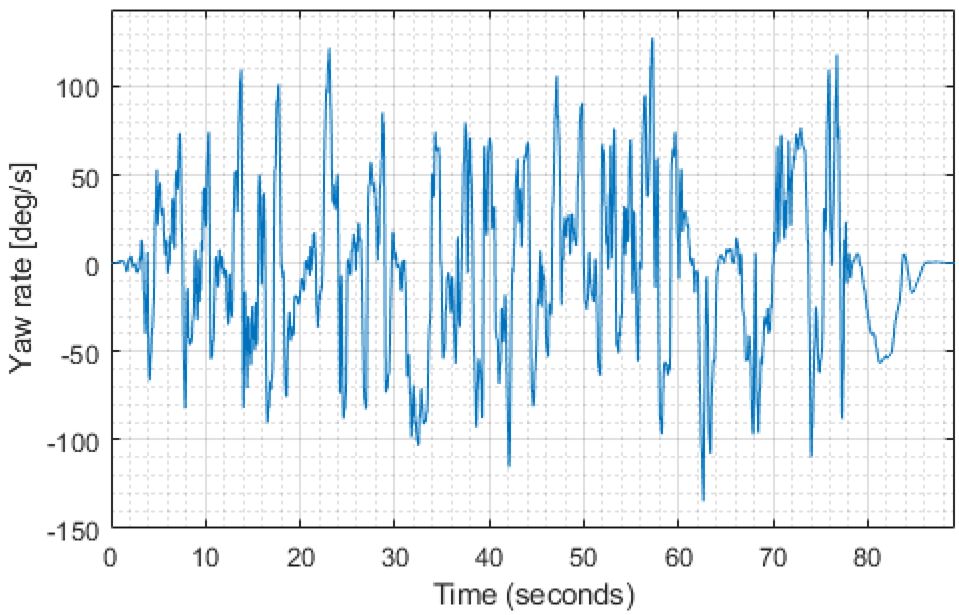
Vehicle Speed

For the vehicle speed two signals were available: the rotational speed of each motor registered by the motor's encoder and the results form the velocity estimator subsystem of the vehicle control. The main difference between them is that the encoder signals, while accurate, are affected by the slip condition of each individual tire, so establishing a correct velocity from them, especially in critical conditions where all tires are sliding may prove difficult. Meanwhile, the Velocity estimator implemented by the team is of a Fuzzy logic type and uses the previously mentioned encoder signals and the accelerometer signals to determine accurately the vehicle speed in every situation. The signal can be considered accurate enough as the design of the estimator and its accuracy have been proven in another dissertation. However, the wheel speed signals from the motor encoders cold prove to be useful in troubleshooting the model's faults, so even though they are not among the monitored outputs. they have been included in the data exported in Matlab environment.



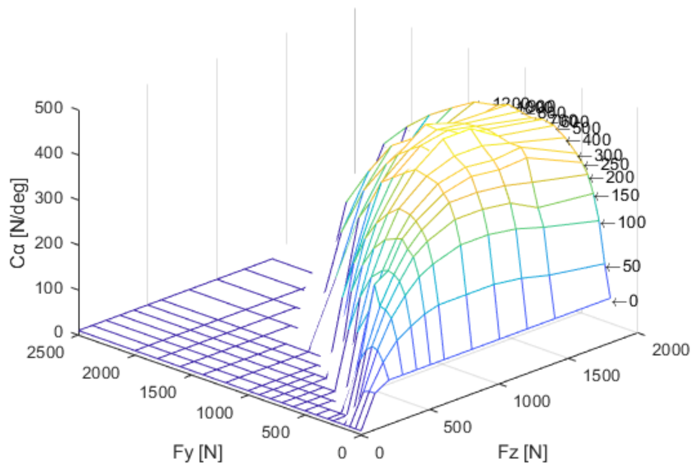
Vehicle Accelerations and Yaw rate

The acceleration signals in both X and Y direction were simply registered through data logging. The IMU includes an internal lowpass filter, leading to good quality of signal even without postprocessing. The same holds true for the Yaw Rate signal, A sample of the Yaw rate registered during an autocross event is presented in the image below:

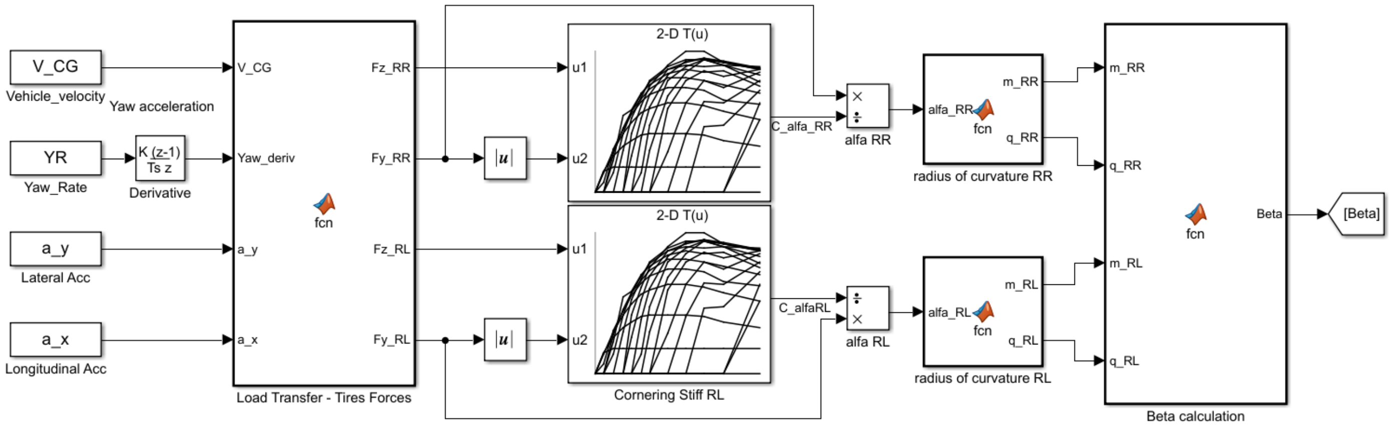


Sideslip Angle

Sideslip angle measurement is fundamental to have a good validation of the vehicle model, however, sensors capable of measuring it are very expensive and can’t be mounted permanently on the vehicle. In order to have a good reference value with the available means, the slip angle

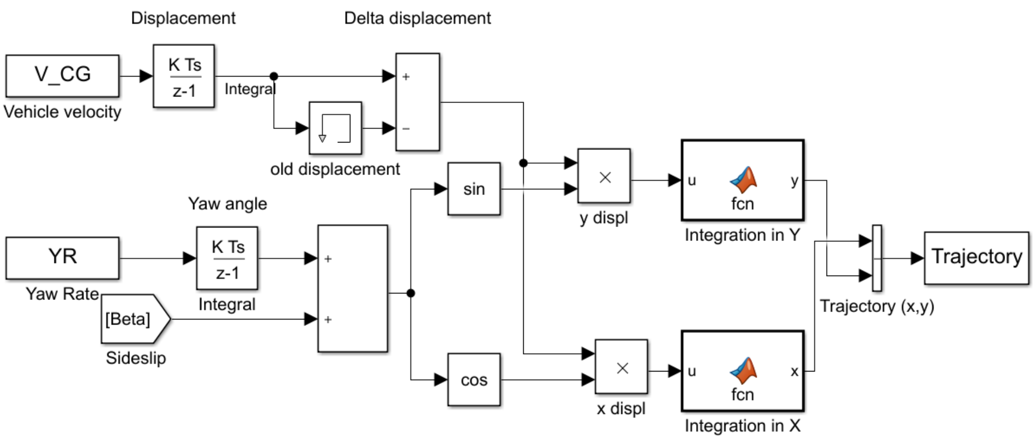


was calculate utilizing the .tir files provided by the tire manufacturer. The signals used for the calculations are the longitudinal and lateral acceleration, used to calculate the load transfer in each instant, the yaw rate, derived to obtain the yaw acceleration and used to determine the lateral force balance between axles in each instant and finally a 3d map of the cornering stiffness of the tire , obtained from the previously mentioned .tir file. In the picture below the scheme of the signal processing done through Simulink is shown:



Vehicle trajectory

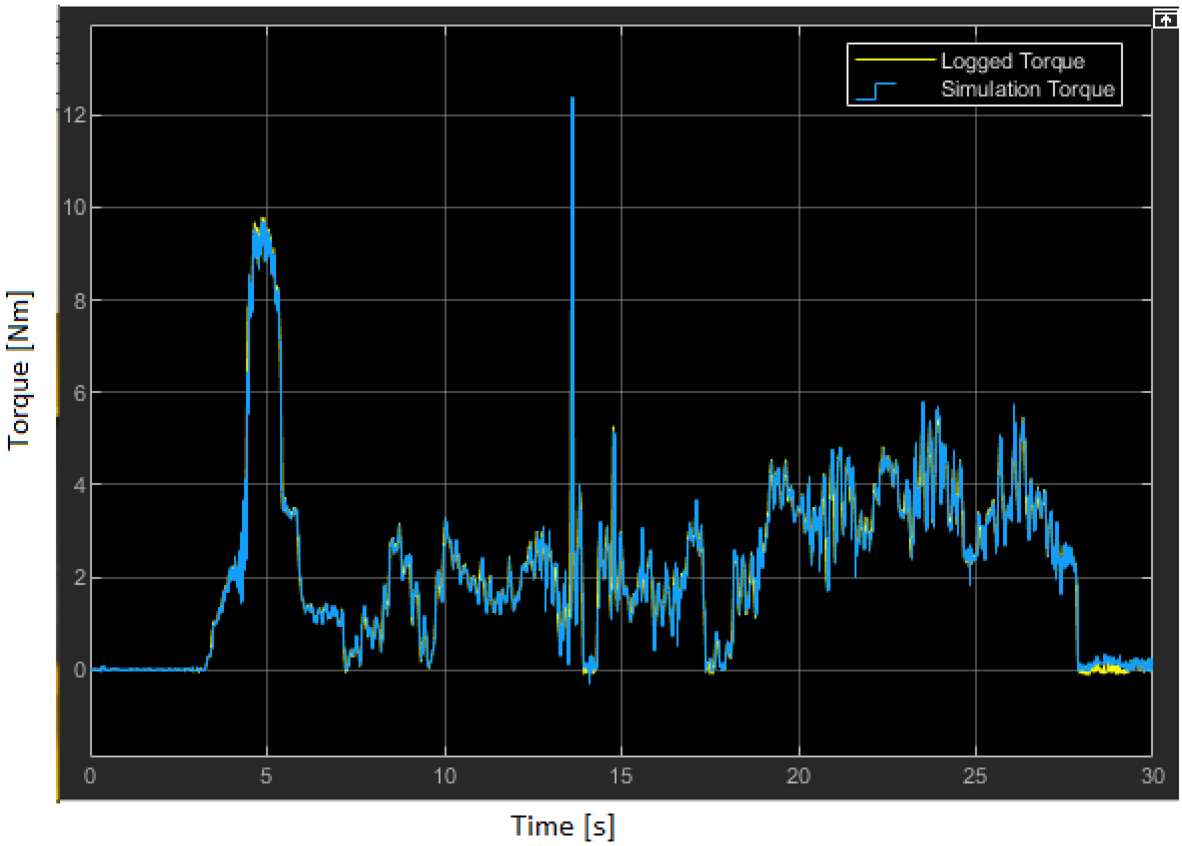
Again, the trajectory of the vehicle must be obtained through calculations and post processing of available signals rather than simply recorded using a sensor. This is mainly because commonly available GPS systems don't provide enough accuracy to be reliably used for these purposes and those that do have the same drawbacks as optical sensors for sideslip angle measurement. The trajectory was thus obtained utilizing the previously calculated Sideslip angle. The signals used for the calculations are the vehicle's velocity, the Yaw rate and the previously calculated sideslip angle. The block diagram used to perform these calculations



is shown below:

Input consistency

The various inputs provided to the model were checked against the vale at every step of the simulation to verify that all values were being received correctly. The figure below shows the logged torque request and the simulated one:



The signals are fundamentally the same, though there are some small inconsistencies in certain points. The nature of these inconsistencies is difficult to determine without in-depth knowledge of the source code of the solver, however, they are rare and small in magnitude, so the signal will be considered consistent with the input. This process was repeated for all inputs with similar results.

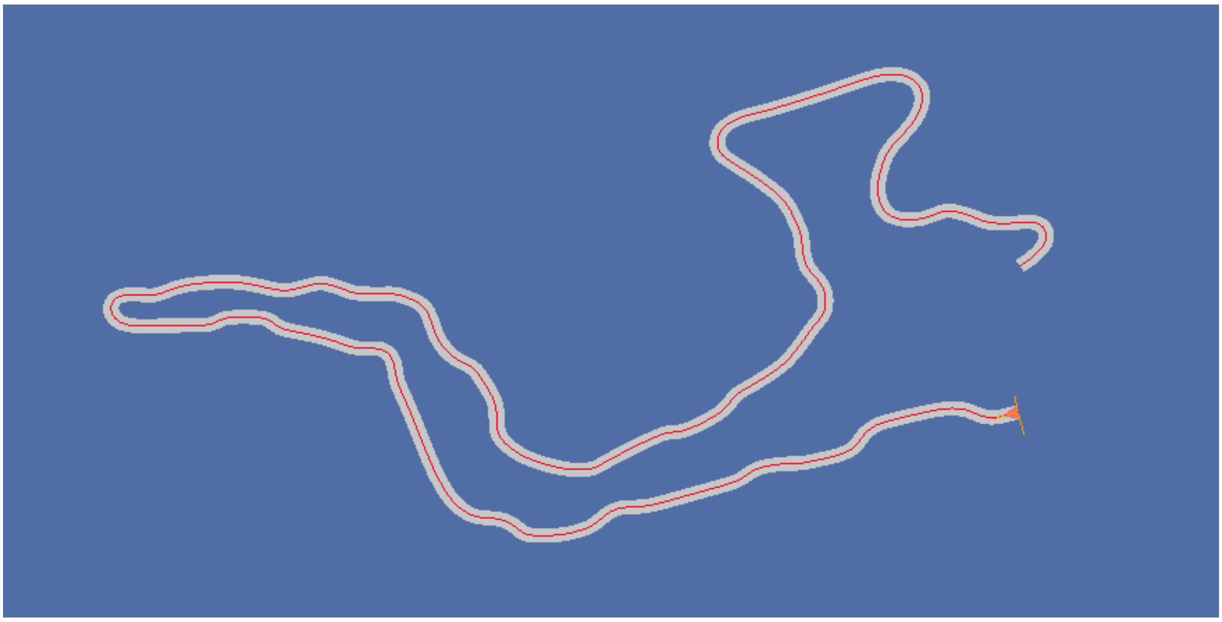
4.3 Creation of virtual track

Once the trajectory of the vehicle in the various events has been defined, a virtual track for use in the simulations can be created. This is not strictly necessary for the first two maneuvers, as the track is only used to determine driver inputs, but in this case the driver model is not enabled and the inputs are defined by the logged signals. However, the Autocross simulation will need a working pilot model, so a track must be created. Tracks will be created also for the skidpad and acceleration simulation, as virtual track also gives a graphical representation that can aid in discovering eventual trouble with the model.

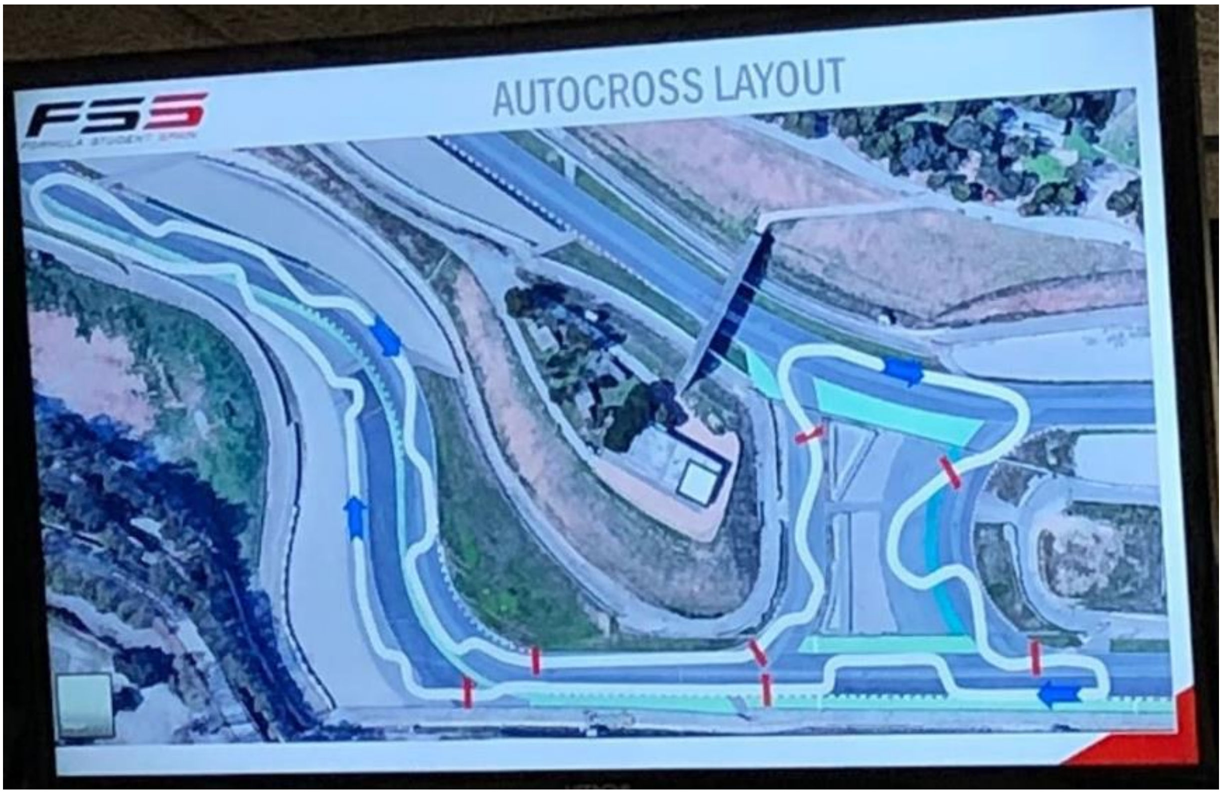
For the track creation, a .drd file must be created. The file is simply a text file with x-coordinate, y-coordinate, z-coordinate, lateral inclination and width of the track for each point. This data was simply written into a file using a matlab script.

However, the .drd file contains only information about the trajectory to be followed by the driver, in order to have a graphical representation a .rdf file must be created through the Vi-road tool. This is accomplished by importing the previously created .drd file in Vi-road, using it as centerline for a graphical representation of the road and then assigning a width to each section.

The autocross tracjk obtained this way is shown in the image below:



By comparing it to the layout presented during the event it is clear that while there is some drift in the signal, probably caused by a bias in one of the integrated variables, the track obtained is significantly similar to the real one.



Chapter 5

Validation Procedure

5.1 Preparation of Co-Simulation Environment

As previously mentioned, CarRealTime offers the possibility to perform Co-Simulations with Simulink. This functionality can be used to model additional subsystems other than the basic ones, substitute internal modules with Simulink models or direct feeding of input signals to the solver. This last functionality proved to be very useful for the validation of the model: the data obtained in the previous chapter was imported into the Matlab workspace and fed to the system, then the outputs of the model were checked against those of the real vehicle to see how good the fit was.

In normal simulations CarRealTime uses an advanced driver model to determine various inputs like steering angle, throttle and brake requests. This driver model must be bypassed in order to feed the previously logged data to the model. This is accomplished by simply deactivating the internal subsystems that must receive an external input. So, additionally to each individual signal, an activation flag for each subsystem must be provided.

For the set-up of the Co-simuation environment, a Simulink model containing the CRT solver block must be created by importing it from the CRT Simulink library. Then, the CRT solver block's parameters must be set up according to the desired maneuver. The main parameters are:

- **Input File:** Every simulation requires an .xml file to be run. This file is generated by performing a static lap time simulation in “files only” mode and contains information about the vehicle model, active subsystems, track and road characteristics and general set up parameters.
- **Input Signals:** In this setup window the list of input signals to the model are chosen.
- **Output Signals:** The list of desired output signals to the Simulink environment is chosen here. Both signals needed for further calculation and post-processing must be selected.

Input File

The input file is generated by performing a Static lap time simulation in “files only” mode. This generates an .xml file containing all information regarding the vehicle model, active subsystems, track and road characteristics, driver requirements and general set up parameters. These informations are used for the setup of the Co-Simulation.

Input Signals

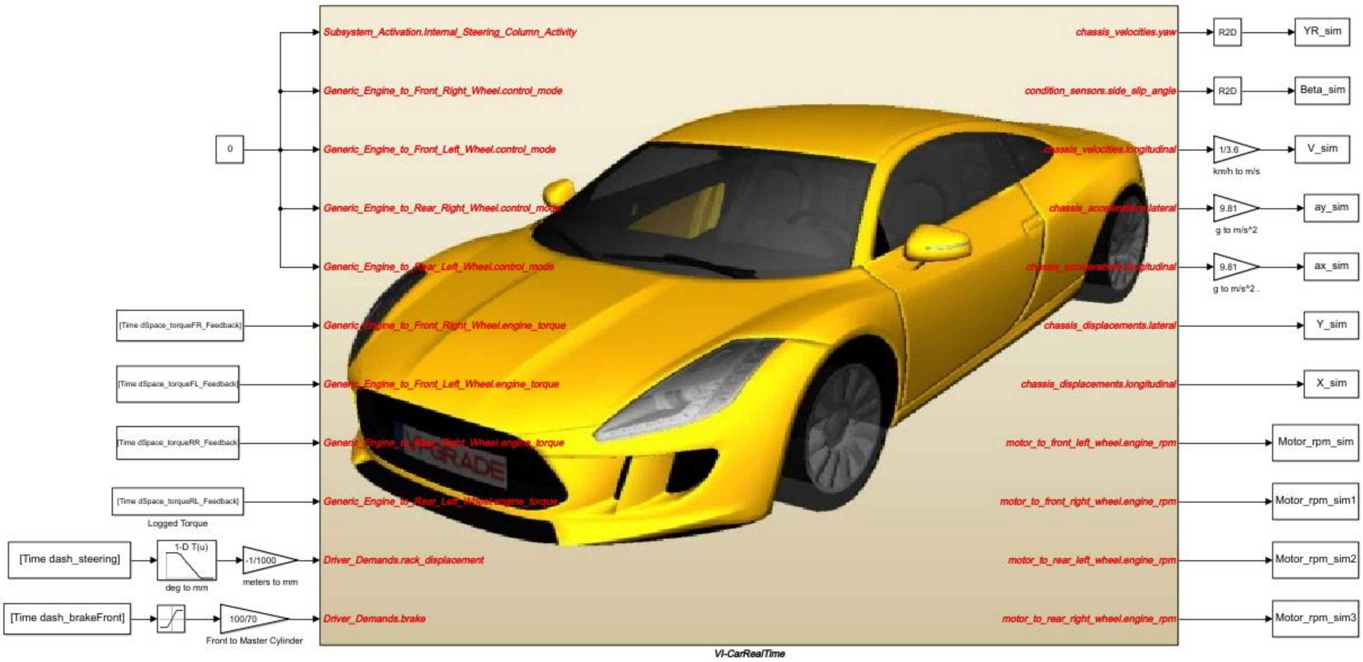
The required input signals are:

- **Internal Steering Subsystem activation:** The value is set to 0 to deactivate the internal steering model. This “disconnects” the previously mentioned driver model from the steering input, allowing a previously recorded signal to be fed to the model.
- **Steering Rack displacement:** The registered value of steering angle from the data logs is converted into a linear rack displacement and fed as an input to the model. The value is in mm.
- **Generic engine to FL (FR, RL, RR) wheel control mode:** This variable is set to 0 to allow the on-wheel motors to be controlled with a torque input. This is repeated for all 4 wheels.
- **Generic engine to FL (FR, RL, RR) wheel torque:** The registered torque values from the data logs in timeseries format are used as input for the torque value at each wheel. The value is in Nm.
- **Brake System Master Cylinder pressure:** The pressure registered in the brake line is converted in pressure at the master cylinder and fed as input to the model. The value is in Pa.

Output Signals

The output signals used for the validation are:

- **Vehicle velocity:** The vehicle's simulation velocity is available as a direct output channel and is useful to validate the longitudinal behavior of the vehicle.
- **Vehicle trajectory:** The trajectory is obtained from the x and y displacement available as output channels of the model.
- **Longitudinal acceleration:** The vehicle's longitudinal acceleration is available as a direct output channel and is useful to validate the longitudinal behavior of the vehicle.
- **Vehicle Yaw Rate:** The yaw rate is a useful output for the validation of the lateral behavior of the vehicle.
- **Sideslip angle:** The sideslip angle is useful to compare the lateral behavior of the vehicle.
- **Motor to FL (FR, RL, RR) wheel RPM:** The instantaneous speed of each motor is useful to control slip condition of each wheel.



VI-CarRealTime

5.2 Signal Comparison and results

The results of the various simulations will be discussed in this chapter. The performance metrics used to measure the curves fit are the maximum error, average error, maximum percentage error and the average percentage error. The error percentage is calculated at each instant using the following formula:

$$Delta_{\%} = abs\left(\frac{output_{log} - output_{sim}}{output_{log}}\right) * 100$$

Furthermore, the curve of punctual percentage error will be provided for the more important signals.

The curves analyzed for each event are:

Acceleration event:

- Vehicle Velocity
- Longitudinal acceleration
- Motor speed of rotation (to better determine problems due to slip)

Lateral acceleration, Trajectory, Yaw Rate, Side slip and won't be compared due to the nature of the event, for the sake of brevity,

Skidpad event:

- Vehicle Velocity
- Trajectory
- Lateral acceleration
- Yaw rate
- Side slip

Longitudinal acceleration won't be compared due to the nature of the event, for the sake of brevity,

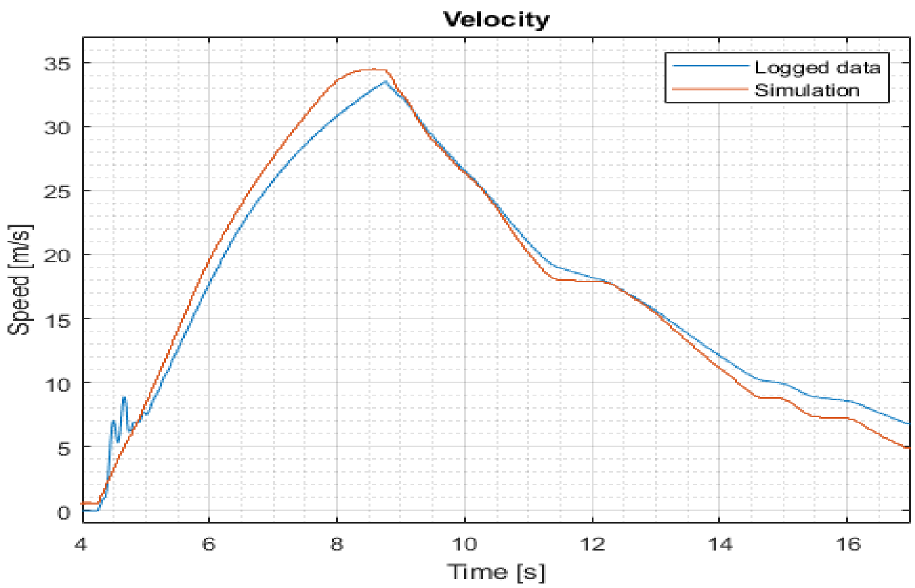
Autocross event:

- Vehicle Velocity
- Trajectory
- Longitudinal acceleration
- Lateral acceleration
- Yaw rate
- Side slip

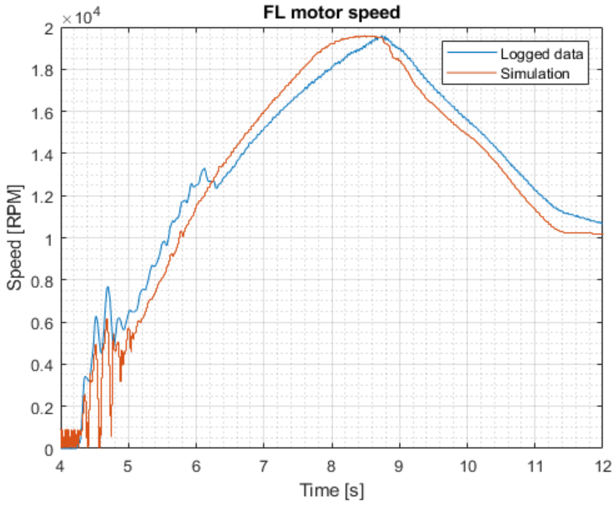
Acceleration event

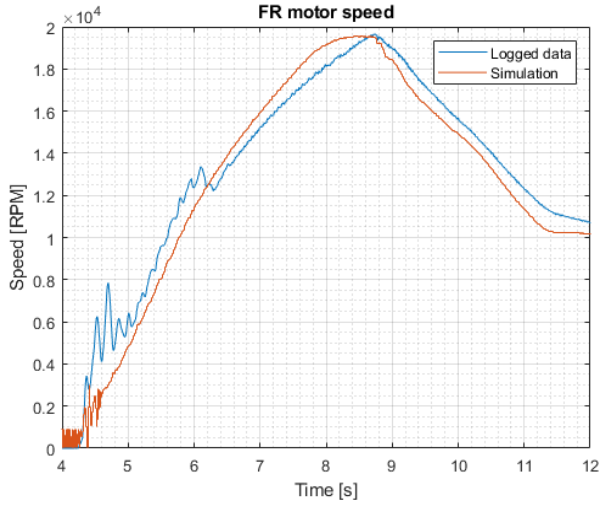
Vehicle speed

The first signal analyzed for the acceleration event is the vehicle speed:

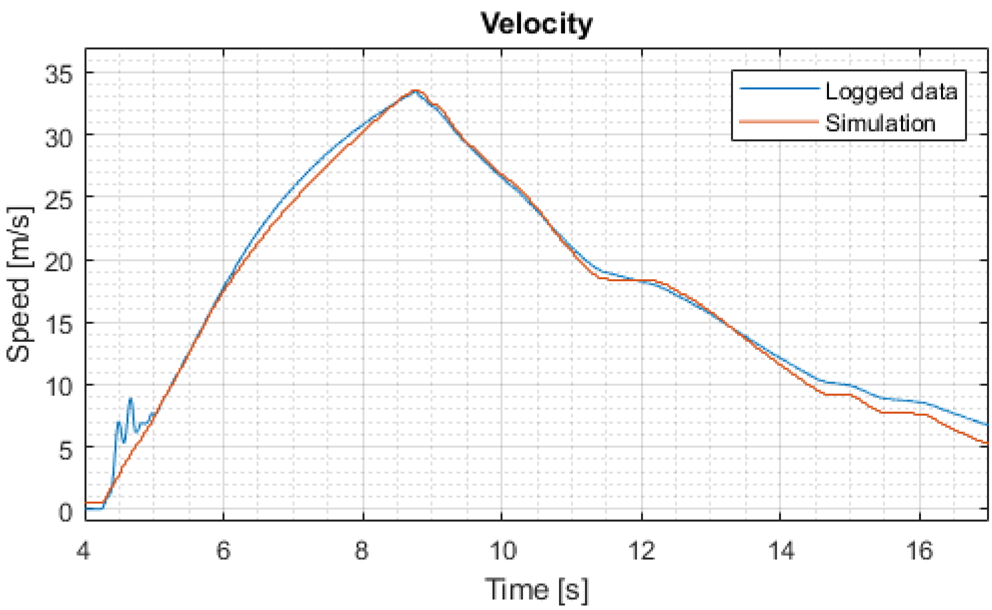


As can be seen from the plot, there is quite some difference between the logged vehicle speed and the simulation one. However, the cause for this difference is quite easily found: the logged signal shows some spikes near the beginning, since this signal is derived from wheel speeds it is a clear sign that there is significant wheel slip in the real vehicle. By comparing the logged motor speed signals to those of the model, we can obtain a clearer picture:

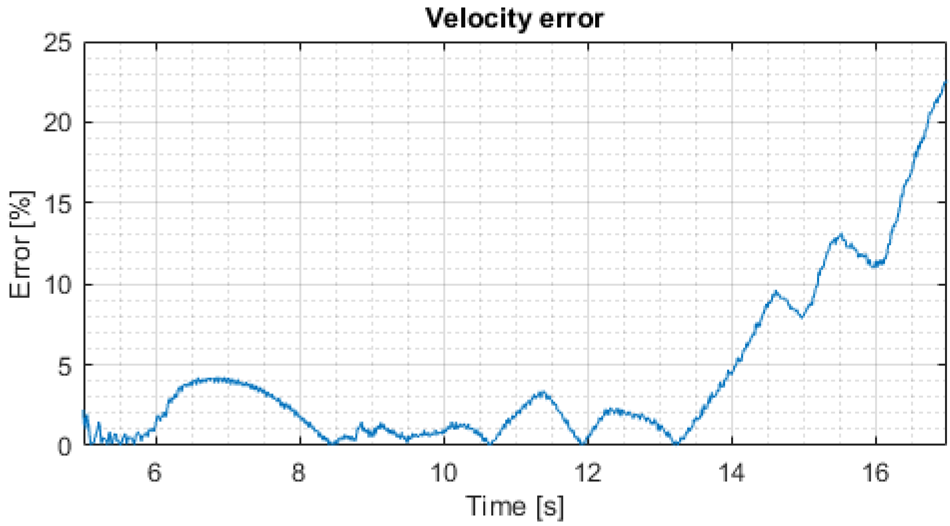




The slip experienced in the real event is much higher than that of the simulation, especially at the front wheels. This is clearly caused by inconsistencies between the real tire's behavior and the data extrapolated from the .tir file. This can be corrected quite easily by changing the scaling factor for the peak longitudinal force and longitudinal slip stiffness of the tire. After various tries, the ideal values were found to be 0.85 for the peak longitudinal force and 0.8 for the longitudinal slip stiffness of the tire. After the values were corrected to better approximate the behavior of the real tire, there was a great improvement in the simulation results, as can be seen from the speed graphs of the second simulation:



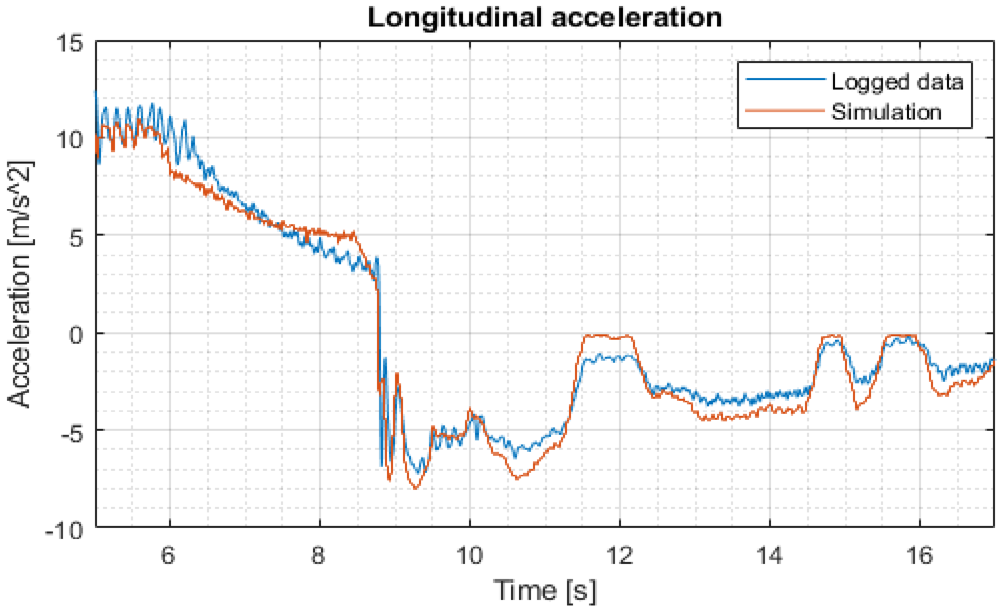
The fit is good: the mean error is 0.53 m/s, the maximum 1.53, while in percentage the mean is 4.25% and the maximum is 22.5% (during the last braking portion). The graph of the error percentage is the following:



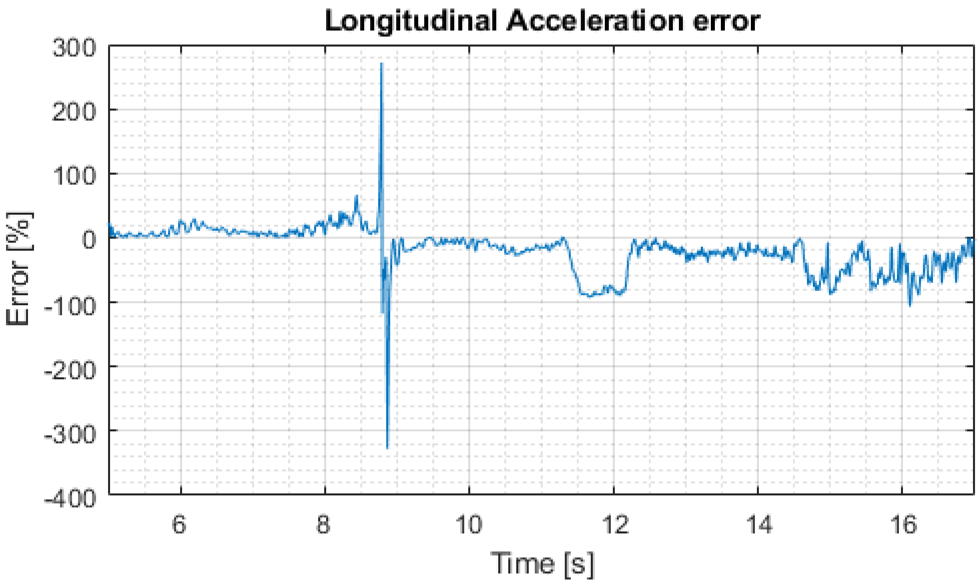
It seems from this graph that the behavior of the braking system is less robust than that of the motors. This is easily explained by the fact that the torques are supplied as registered by the data logger, without having to use the internal model of the powertrain. This is not possible for the brakes: the only way to model the braking action with the available signals is to supply a master cylinder pressure to the internal brake model. Furthermore, the increase in percentage at the end is most likely due to the decreasing magnitude of the velocity

Longitudinal acceleration

The graph of the longitudinal acceleration is the following:



It was obtained using the corrected .tir parameters found through the previous simulation. The fit is generally good, however there are some spikes in the error caused mainly by the small magnitude of the acceleration in the second half of the maneuver and due to possible inaccuracies in the braking system behavior. The mean error is 0.76 m/s^2, the maximum is 6.25, while the mean percentage of error is 18.7% and the maximum is 270%. The graph of the error percentage is presented below:

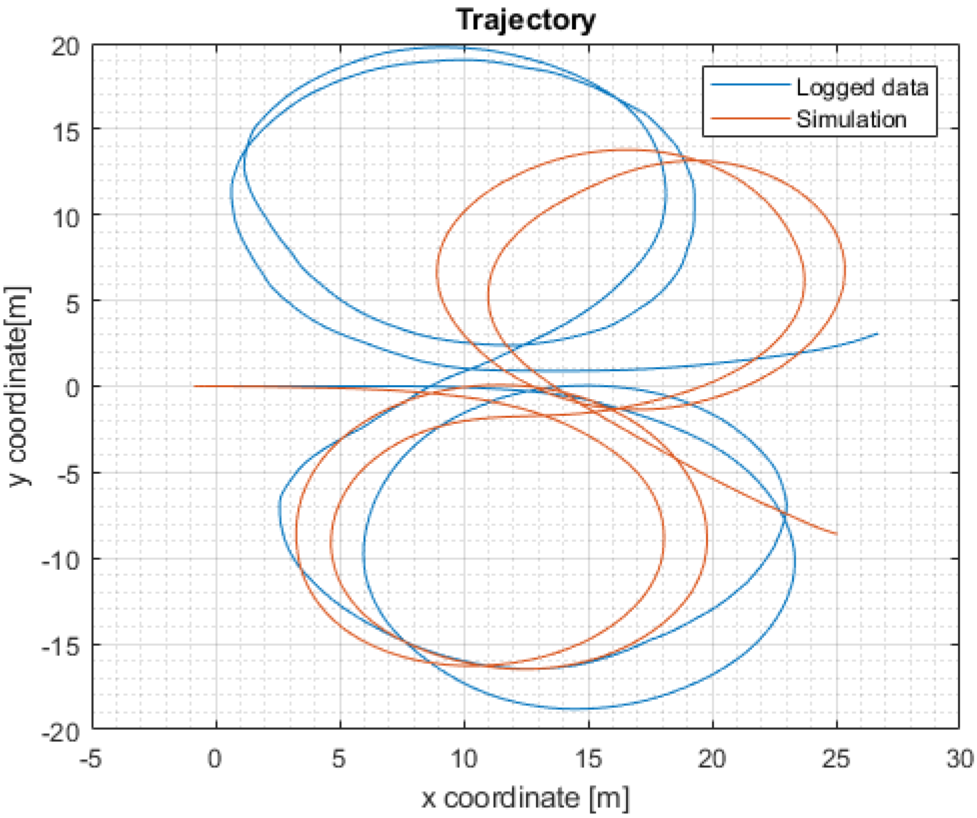


Though there are some significant inconsistencies, it is difficult to establish the cause at the moment. These results are satisfactory enough for now to pass to the next step of the validation: the Skidpad simulation.

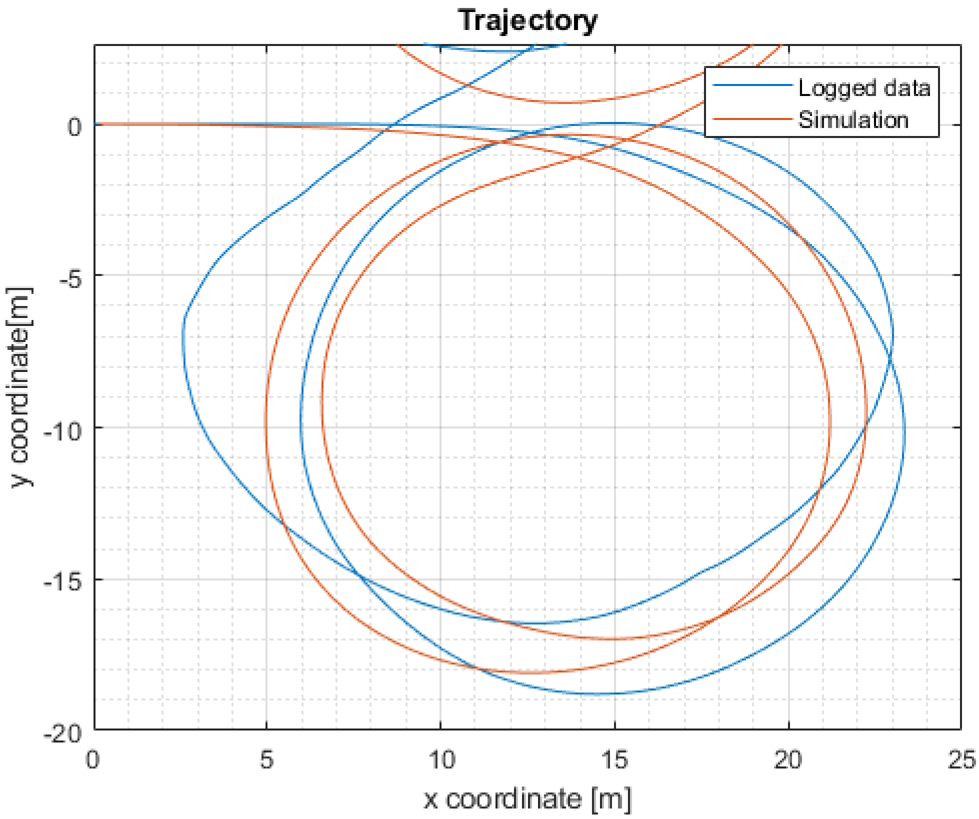
Skidpad event

Trajectory

Again, the result of the first simulation proved to be not satisfactory. The trajectory obtained is illustrated in the figure below:



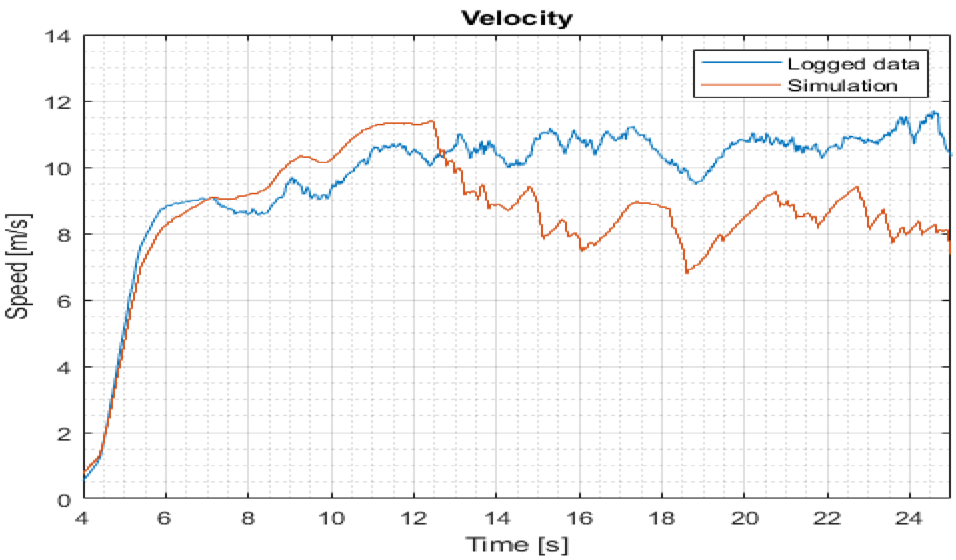
There is a noticeable drift in the signal: there is almost certainly an integration error adding up during the simulation. Indeed, the steering angle input was found to be biased by 1.45 degrees while at rest, a seemingly small amount, but enough to generate a large error during a longer (30 seconds) simulation. Furthermore, the turn radius is lower than the actual one with the same steering input, this again hints at higher tire forces than those generated by the real tire. Again, different scaling factors for the side slip stiffness and peak Fy value of the tire were tried and finally, with a value of 0.8 for the peak Fy and 0.85 for the Side slip stiffness, The following plot was obtained:



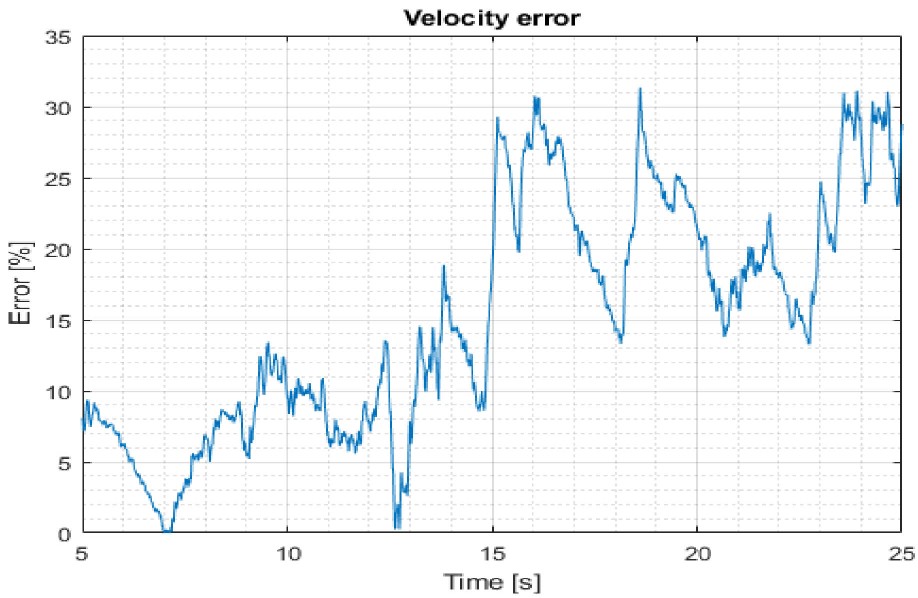
There is still some drift in the signal, though determining the cause proves difficult. It is also evident that the peak friction coefficient is still too high, as the radius of curvature is still smaller than the logged one for the same input. However, lowering the scaling factor further does not yield good results. It must be noted that following an exact trajectory (obtained through analytical methods and not logged, so prone to error itself) for a longer event such as this is a very difficult task.

Velocity

The graph of the velocities, obtained after the previous corrections, is the following:

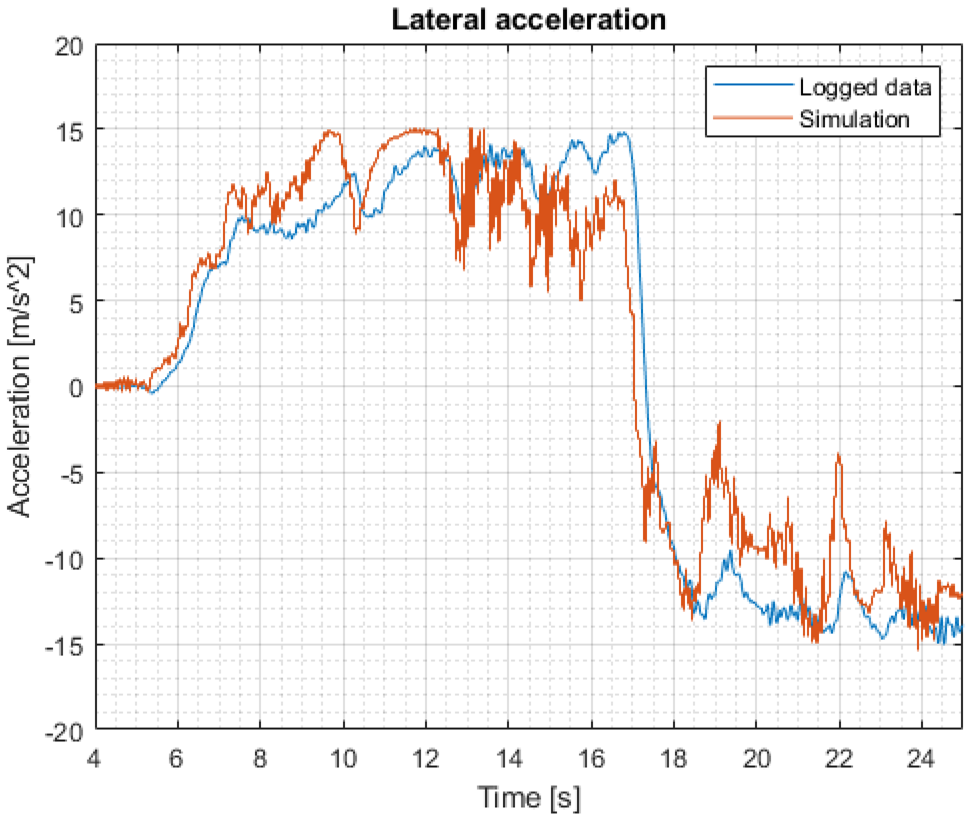


The initial fit is good, however during the change in direction the signal starts to diverge. This could suggest a discrepancy between the real tire's longitudinal force sensitivity to slip angle and the modeled one. The fit parameters are: maximum difference 3.5 m/s, mean difference 1.35 m/s, maximum percentage difference 33.5% and mean percentage difference 13%. The curve of the punctual difference percentage is shown below:

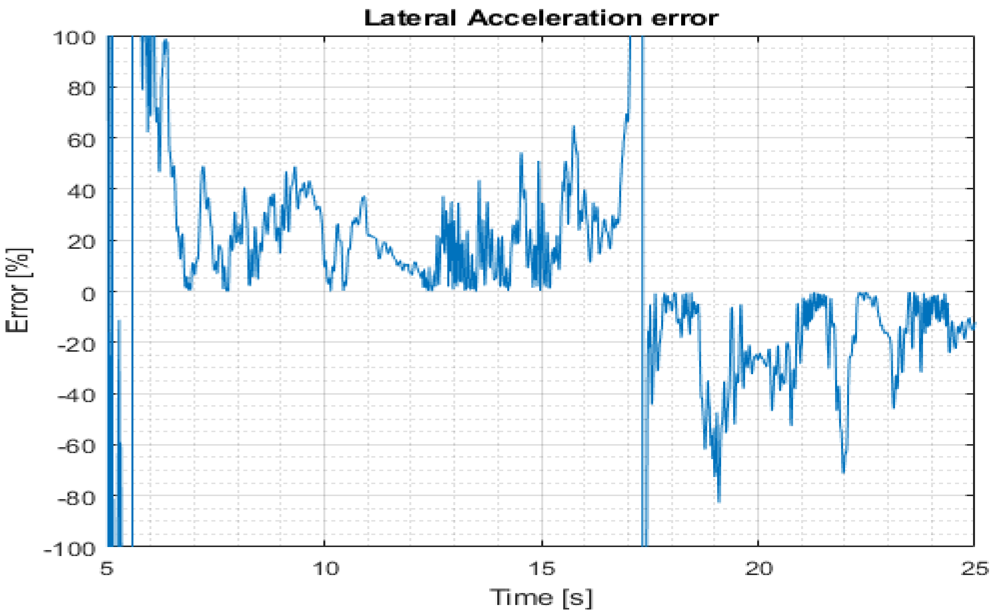


Lateral Acceleration

The graph of the lateral acceleration is the following:



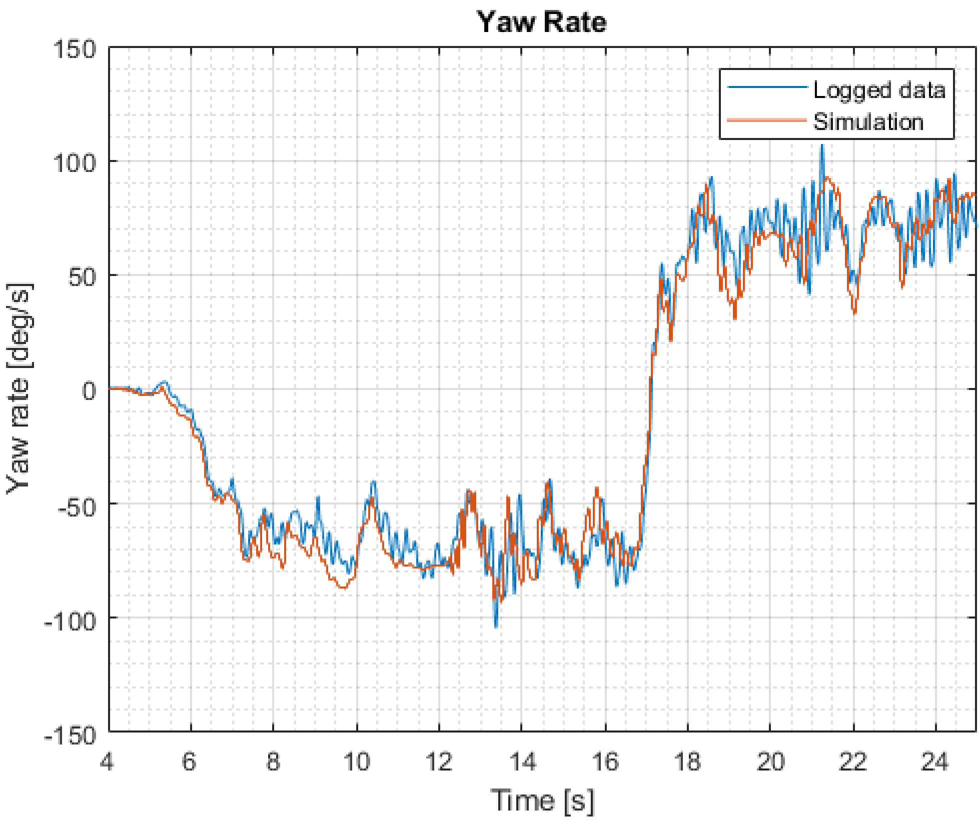
The fit is good to start with, though in the later portion of the simulation the accelerations are lower than the logged ones, probably due to the lower speed, as seen in the previous section. The maximum difference is 11 m/s^, the mean difference 1.32 m/s^2, maximum percentage difference 33.5% and mean percentage difference 40%. The curve of the punctual difference percentage is shown below:



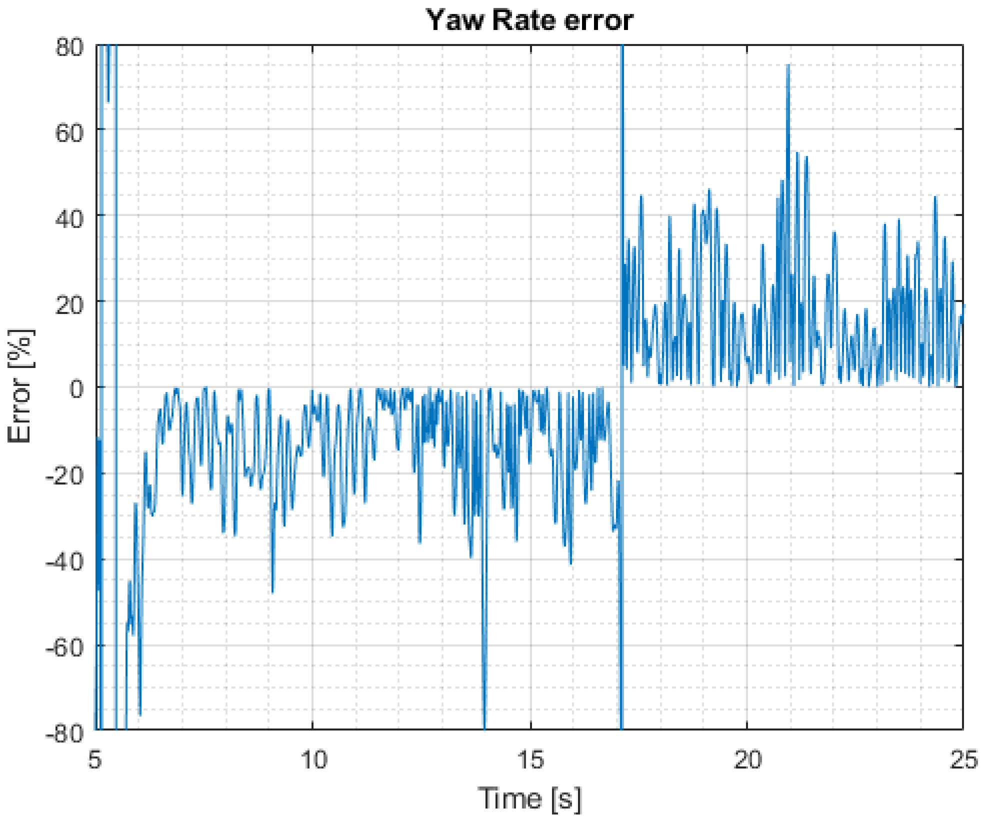
From this curve it seems clear that the modeled lateral acceleration cannot be trusted when the vehicle speed is low and during inversions of direction. Otherwise, the error varies in magnitude

Yaw Rate

The graph of the Yaw rate is the following:

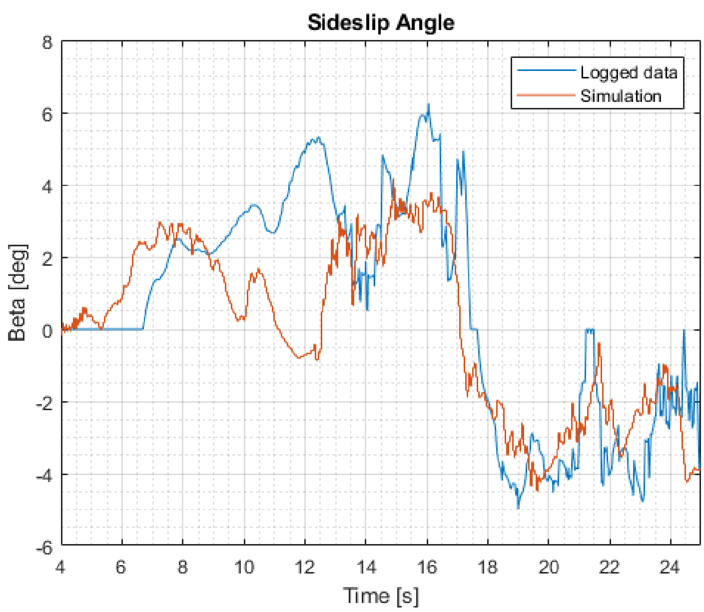


In this case the fit seems to be good during the entire simulation. The maximum error is 17 deg/s, the mean is 9 deg/s, the maximum percentage is 9.64%, the maximum (outside of the areas nearing division by zero) is 76%. The plot of the error percentage is the following:



Sideslip angle

The plot of the sideslip angle is shown below:

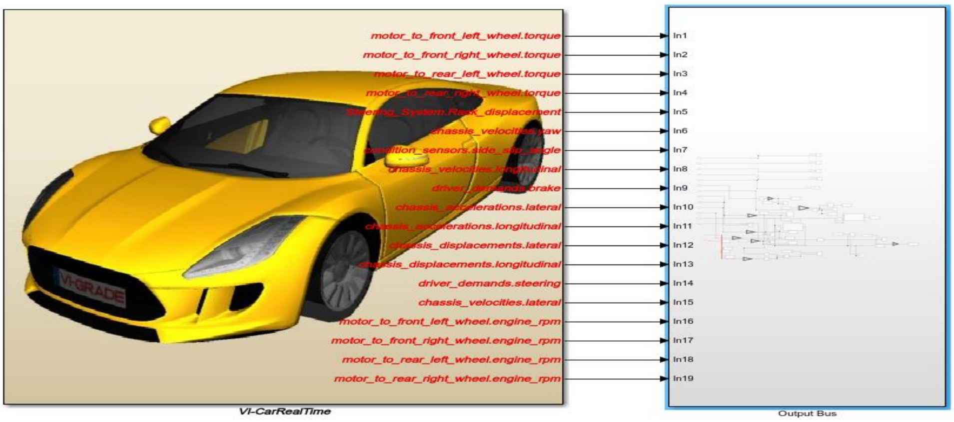


There are a few considerations to be made about this plot: the signal used for validation has been obtained analytically and not logged through a sensor and is not validated, as this would require the aforementioned sensor and the point would be moot. This simulation then is more of a test of the methodology used to extract the sideslip angle than an actual validation of the model. In fact, neither signal can be validated through the other, however the comparison can still be interesting. The vehicle model signal actually starts before the logged one, the reason for this is unclear. The first ascending part of the curve shows similar slope but with a delay. Then there is an almost mirrored behavior between the two signals. The cause is again uncertain. After this however, the signals converge to similar values for the remainder of the simulation , even after the change in direction.

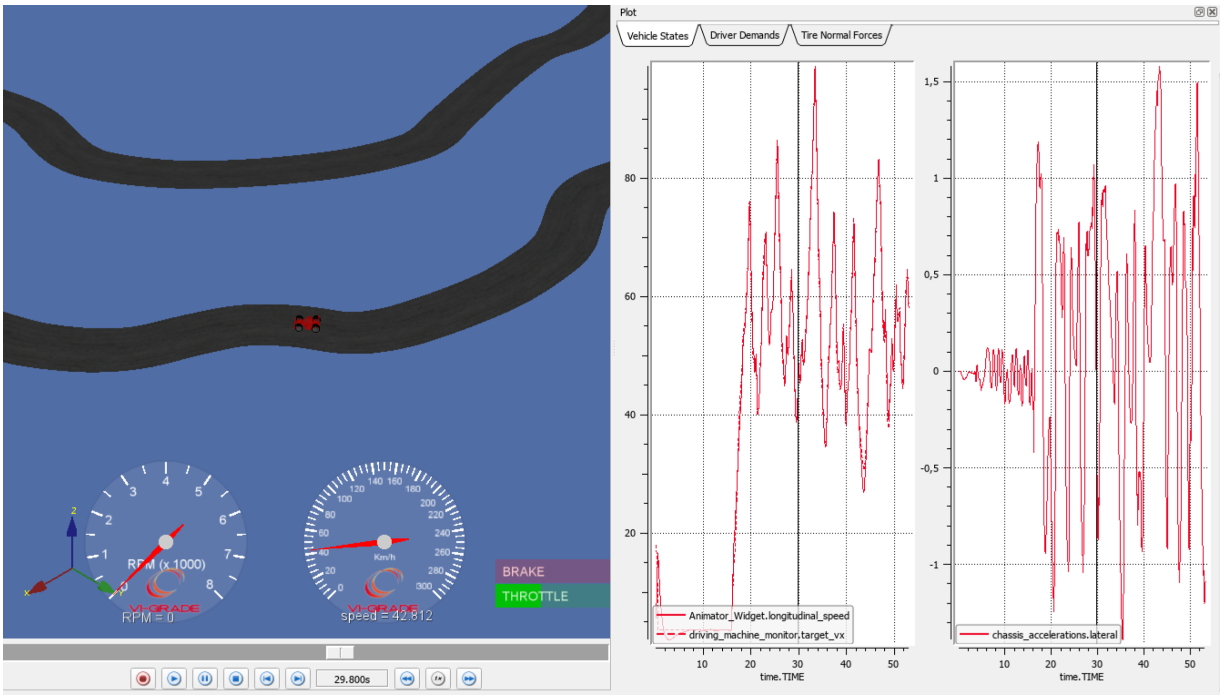
It is unclear from this simulation which of these two signals would be closer to the real behavior of the car, as such, the following simulations won't compare the sideslip angles as a proper reference could not be established.

Autocross event

As mentioned previously, this event was run using the software's driver model. The needed output signals were still exported in Matlab environment: the simulation was run as a CoSimulation without inputs, only with the needed outputs. This is because with such a complex event it would be hard to achieve good results with the logged inputs. Also, this event is much longer than the previous ones, and even a small integration error could lead to vastly different, diverging signals.



However, this mode of testing also has its advantages: the inputs to the models are not strictly the same as in the previous case, however, most simulations with the software will be using the driver model. This step thus validates the ability of



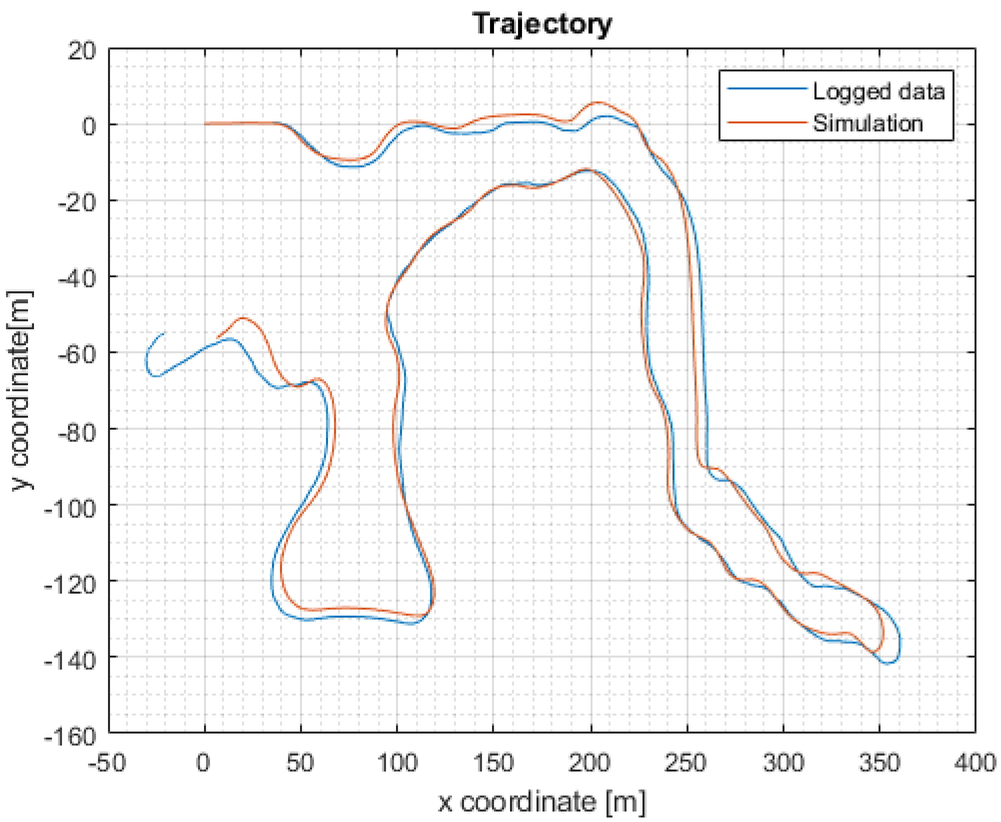
a "normal" simulation with the driver model to provide reliable and realistic results.

Lap time

The simulated lap time is 77.55 seconds, while the registered time is 77.67. This is a difference of 1.5%.

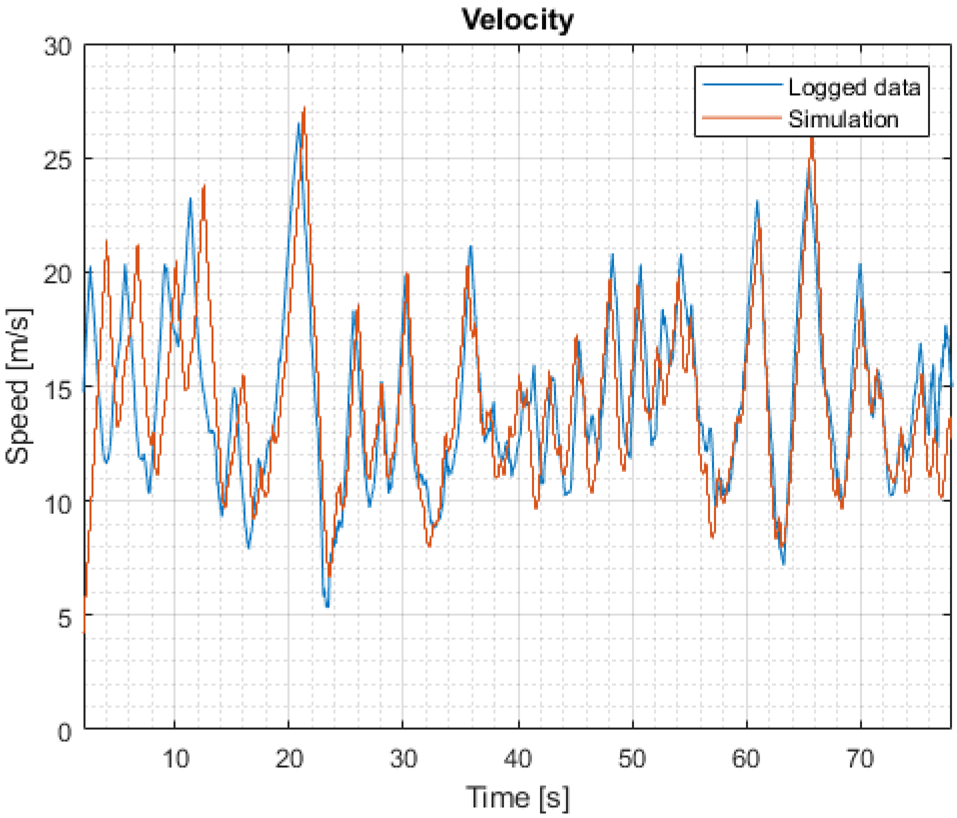
Trajectory

The trajectory is very similar to the input one. That is to be expected, ad the driver model was controlling the vehicle to achieve this trajectory. Deviations are mostly due to the fact that the defined road has a width of 5 meters, as such the driver can deviate from the centerline to reduce lap-time. However, since the width is small, the usually negligible.



Velocity

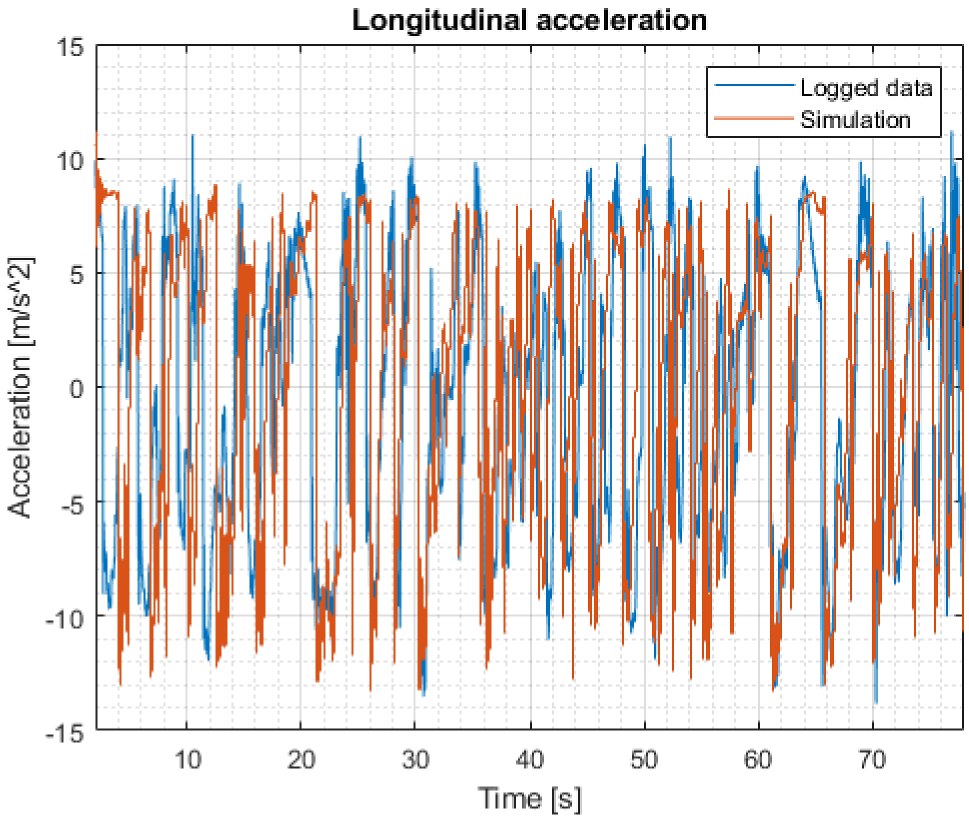
Again, the velocity plot shows a good fit between the model and the real vehicle's behavior, with the model reaching slightly higher speeds and having a small delay at the start.



The maximum error is 9 m/s, the mean error is 1.82 m/s, the maximum error percentage is % and the mean error percentage is 12.8%.

Longitudinal acceleration

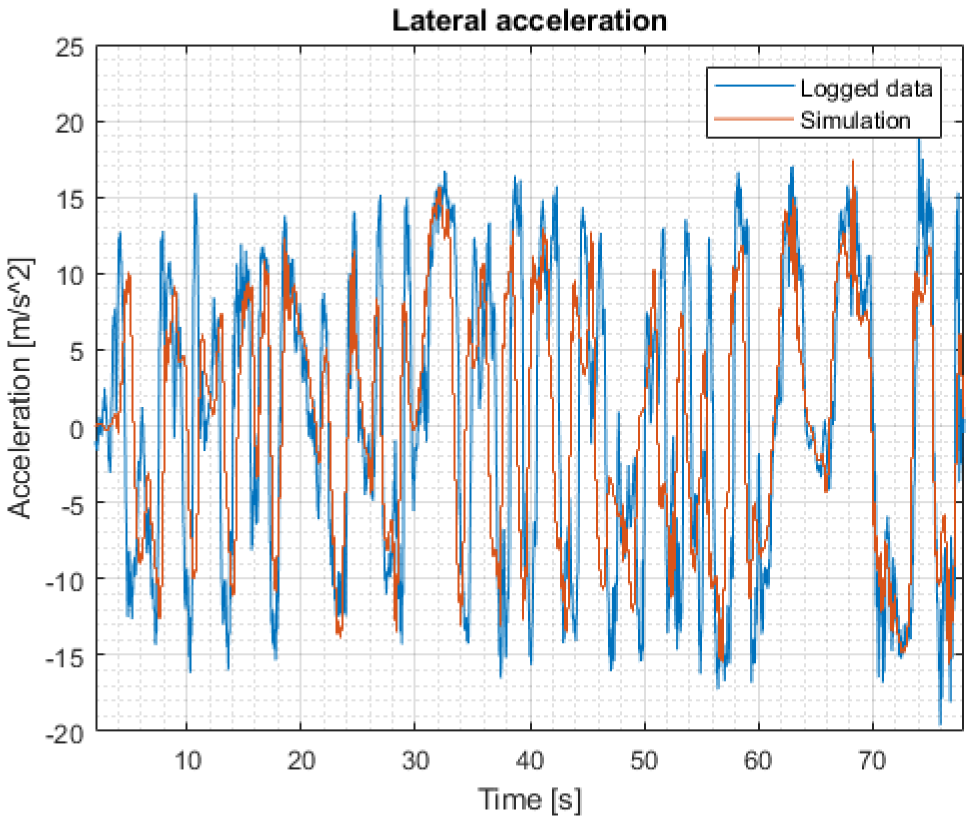
The longitudinal acceleration plot shows a slightly worse fit than that of the previous signals, mainly due to noisy signals.



The maximum error is 15 m/s^2, the mean error is 4.78 m/s^2, the mean error percentage is 52%. This is skewed heavily by the many points where the denominator is small and the difference, though small in magnitude, reaches very high peaks. This condition is caused by the many intersections with the x axis of the curve (many situation where the vehicle switches from acceleration to deceleration or vice versa).

Longitudinal acceleration

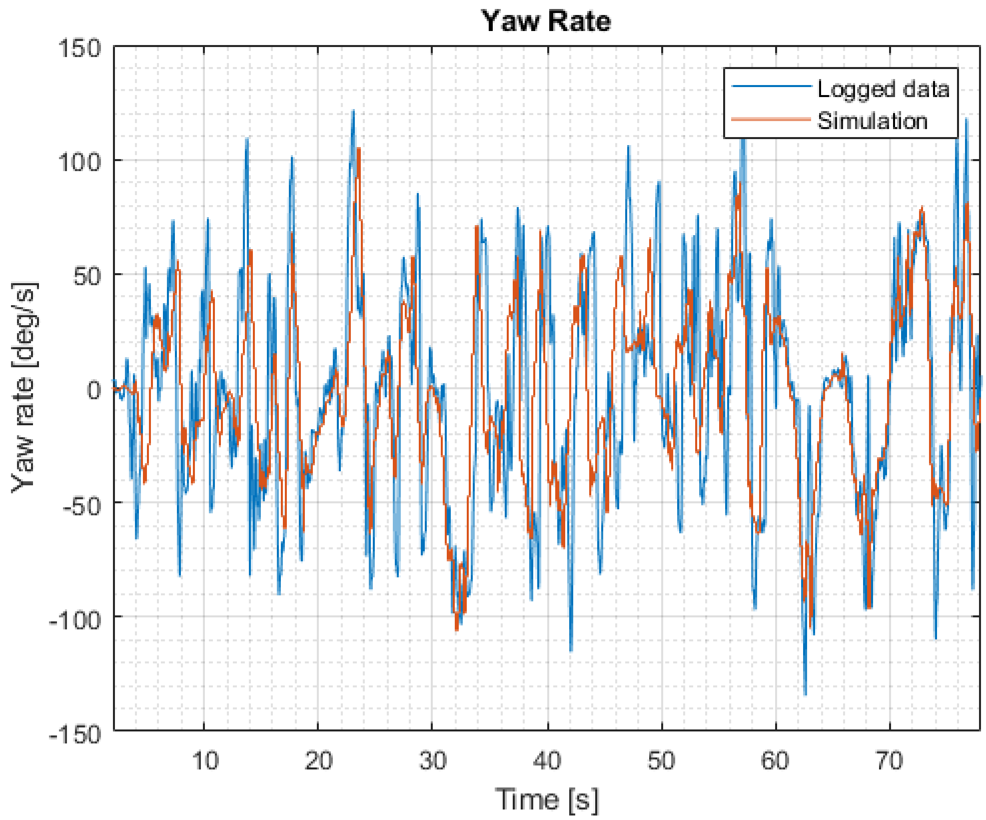
The lateral acceleration plot shows a similar fit to the longitudinal one:



The maximum error is 17 m/s^2, the mean error is 4.72 m/s^2, and the mean error percentage is 12.6%. The same considerations hold true as for the longitudinal acceleration, but to a lesser degree in this case.

Yaw rate

Out of all the analyzed signals, the yaw rate is the one that shows the worst fit: the simulation values clearly show lower peaks than the logged ones. This may be caused by the change in trajectory: the driver tends to reduce the radius of curvature and so at similar speeds the yaw rate is reduced (more so than the lateral acceleration, that on first approximation depends on the square of the speed while the yaw rate is directly proportional)



The maximum error is 127 deg/s, the mean error is 29.6 deg/s, and the mean error percentage is 12.6%.

Sideslip angle

The sideslip angle will not be compared as the logged signal, even if not without its merits, proved to be fundamentally flawed. Thus, comparing a simulation value to a signal that is known to be wrong cannot yield good results.

Chapter 6

6.1 Conclusions

The importance of having a reliable simulation model for the design and testing phases of any product cannot be overstated: that is why this discussion is focused on the validation of a vehicle model to improve future efforts.

There are some notes to make about the procedure used: some signals could not be properly logged due to budget restrictions: accurate Sideslip angle sensors can cost tens of thousands of euros, and as such could not be sourced for this research.

The procedure used to define a substitute for this signal, while theoretically correct, relies heavily on the correct modeling of the tire behavior, as the 3d map of cornering stiffness vs Fy and Fz has been obtained through the tire property file. Modeling of a tire's behavior is a very complex field, and while the Pacejka formulas do provide a good approximation of the tire forces, the method used to create such parameters has a great influence. In this case we know that this is a tire with an unusually low vertical load on average (50 kg), and our sponsor Pirelli told us that they had to use a new test rig for this reason and that there could be some inaccuracies due to this.

Furthermore, even a "perfect" .tir file would need to be validated to find good scaling factors for each road surface, as there is bound to be significant difference between each road and the test rig's surface.

This leads to an output signal that cannot be considered fit as a reference behavior for the vehicle. The trajectory is also affected by this, but to a lesser extent, as sideslip angle values are usually low and the value of the yaw rate has much higher effect on the trajectory.

Furthermore, incorrect modeling of the tire can be even worse for the determination of correct simulation outputs. It is in fact no coincidence that most of the errors between the logged behavior and the simulated one could be explained by inconsistency in tire behavior,

It can then be surmised that having a good tire model is of the utmost importance if accurate simulation outputs are to be achieved.

However, the tire property file’s validation is outside of the scope of this discussion and some of the results achieved can be considered good if the limitations of this approach are kept in mind.

Concerning maneuvers that have been performed with input from data logs, it is clear that the magnitude of most signals cannot be trusted completely, the main exception being the vehicle’s velocity during a straight line acceleration event. However, the fit is somewhat good for most signals, showing that the general trend of the signals can be trusted.

The best result was achieved with the autocross simulation with driver inputs: this goes to show that while the response of the model to the same inputs od the real vehicle diverges slightly, the overall capabilities of the two are well matched when at least the peak force deliverable by the tire has been corrected.

Thus, general maneuvers performed with the driver can approximate the vehicle’s behavior well. This is mainly because even small errors that are integrated during longer periods of time can lead to large drift in the signals, while the driver can correct this by simply changing input.

Acknowledgements

First of all, I would like to thank Professor Tonoli for the support and freedom he gave me in the development of this thesis and for the amazing opportunity of participating to a Formula SAE team. This experience taught me a lot both in technical and human terms and gave me the opportunity to meet extraordinary people with which I shared some of the most memorable moments of my life.

It is those people that I would like to thank next: every team member from the 2017 and 2018 season. Among them special thanks go to Francesco and Marco, who shared all the best and worst moments of this experience with me and Alessandro and Federico, with which I had a lot of fun.

I should not forget to thank all my friends at home who I got to spend more time with the last few months, from old ones like Alessio, Alessandro, Giacomo, Davide, Enrico and Daniele to new ones like Fernando.

Finally and most importantly, I would like to thank my beloved parents for giving me the possibility to study and achieve higher education, for believing in me and supporting me in my hardest times: I also thank my brother for giving me an example to follow with his previous achievements.

Torino, 23/03/2018

Luca Raimondo Marongiu

Bibliography

[1] William F. Milliken, Douglas L. Milliken, 1994*, Race Car Vehicle Dynamics,* SAE International

[2] Vi-grade, 2018, "*Vi-CarRealTime 18.2 Documentation",* Vi-grade Gmbh

[3] Genta Giancarlo, 1997, *Motor Vehicle Dynamics: Modeling and Simulation,* Series on Advances in Mathematics for Applied Sciences

[4] Genta Giancarlo, Morello Lorenzo, 2008, *The Automotive Chassis part 1: Components Design,* Springer

[5] Genta Giancarlo, Morello Lorenzo, 2008, *The Automotive Chassis part 2: Components Design,* Springer

[6] Hans B. Pacejka, J. M. Besselink, 2012, *Tire and Vehicle Dynamics,* Butterworth-Heinemann

[7] **Guiggiani** Massimo, 2014, *The Science of Vehicle Dynamics: Handling, Braking, and Ride of Road and Race Cars,* Springer

[8] **Kiencke** Uwe**, Nielsen** Lars, 2005 *Automotive Control Systems For Engine, Driveline, and Vehicle,* Springer

[9] **Marongiu Luca Raimondo, 2018, Design and Simulation of a Formula SAE Suspension (Inertia calculation)**

Appendix

Aerodynamic properties (.aer file)

```
$--------------------------------------------------------------------MDI_HEADER
[MDI_HEADER]
FILE_TYPE  =  'aer'
FILE_VERSION  =  5.00
FILE_FORMAT  =  'ASCII'
(COMMENTS)
{comment_string}
'Sample Aero Data'
'NOTE: the user-defined unit tags are not supported within tables'
'     They are accepted for single value parameters only.'
$-----------------------------------------------------------------units
[UNITS]
(BASE)
{length  force     angle      mass    time}
'inch'   'pound_force'   'degree'   'pound_mass'    'sec'
(USER)
{unit_type    length  force  angle  mass  time  conversion}
'mph'          1      0      0      0    -1    17.6
$--------------------------------------------------------------test_conditions
[TEST_CONDITIONS]
reference_velocity <mph>       = 37.3
reference_density            = 1.2
front_ride_height_min  = 0.2
front_ride_height_max  = 2.2
rear_ride_height_min   = 0.2
rear_ride_height_max   = 2.2
DRAG_ARM_HEIGHT_MIN = 0
DRAG_ARM_HEIGHT_MAX = 0
$--------------------------------------------------------------front_downforce
[FRONT_DOWNFORCE]
(Z_DATA)
{rear_ride_height }
0.20
0.60
1.20
1.80
2.20
(XY_DATA)
{font_ride_height   downforce }
0.20 53 53 53 53 53
0.60 53 53 53 53 53
1.20 53 53 53 53 53
1.80 53 53 53 53 53
2.20 53 53 53 53 53
$--------------------------------------------------------------rear_downforce
[REAR_DOWNFORCE]
(Z_DATA)
{rear_ride_height }
0.20
0.60
```

```
1.20
1.80
2.20
(XY_DATA)
{front_ride_height   downforce }
0.20 60 60 60 60 60
0.60 60 60 60 60 60
1.20 60 60 60 60 60
1.80 60 60 60 60 60
2.20 60 60 60 60 60
$---------------------------------------------------------------------drag
[DRAG]
(Z_DATA)
{rear_ride_height }
0.20
0.60
1.20
1.80
2.20
(XY_DATA)
{front_ride_height   drag }
0.20 0 0 0 0 0
0.60 0 0 0 0 0
1.20 0 0 0 0 0
1.80 0 0 0 0 0
2.20 0 0 0 0 0
```