IPG Documentation

FormulaCarMaker®

SOLUTIONS FOR VIRTUAL TEST DRIVING

Table of Contents

Abstract

How to start? A question many Academic Teams must have asked themselves before facing a new season - or even the before their first. With the IPG Automotive GmbH as a new team sponsor you receive two licenses for IPGKinematics and CarMaker for free.

But having made the first glances on the new programs many don't know how to get started. Questions emerge like "How does the whole thing work?", "What program am I supposed to start with?" or "What are the benefits of using IPGKinematics and CarMaker in the development process of my Formula Student racing car?".

Fortunately the company IPG Automotive GmbH offers to all registered teams along with their software further technical support like a User's Guide, Manuals, Formula CarMaker workshops, video tutorials and the last but not the least email-support

FormulaCarMaker@ipg-automotive.com

to answer any kind of questions concerning CarMaker, the CarMaker Toolset and IPGKinematics.

This document should also play another auxiliary role. It will show you step by step how to integrate both tools IPGKinematics and CarMaker most efficiently into the designing and optimization process of your Formula Student racing car to assure an outstanding performance in any competition your car participates in. But always remember, this document is only an addition to the handbooks. The CarMaker User's Guide and Reference Manual available in the Help menu of the CarMaker main window should always be used besides this document.

The general approach is explained based on several thesis and internship projects students from different Formula SAE teams made at IPG.

Our current Formula CarMaker racing car model ("FS_RaceCar_6.0") is mainly based on a Formula Student Combustion racing car from season 2011. The example car as well as the showcase test runs are supposed to provide a basis for your simulation model. This document will guide you how to adapt the existing model to best reproduce your car so that you can start simulating as soon as possible. Furthermore, it is dwelled on certain testing procedures and their analysis to validate the model and finally optimize your Formula Student racing car.

Chapter 1

Installation

1.1 Installation and Licensing

1.1.1 How to install CarMaker and other IPG software?

After successfully registering for a Formula CarMaker sponsorship, you‘ll be send a ftp-link. Download the containing files. To be able to install CarMaker on your PC you have to dispose of administrating rights. The .zip archive on the ftp server contains a installation guide which offers you several information about the installation process.

- 1. Please read the enclosed license agreement carefully. By installing the software you accept the terms of the license agreement.
- 2. Log in as user with administrative privileges and extract the downloaded installation pacckage (.zip archive). Open the executable file "ipg-install.exe" via double-click.
- 3. Follow the instructions of IPGInstall. The destination directory you have chosen at the installation (recommended: C:\IPG) will be further referred as "<InstallationDirectory>".
- 4. You will find a start menu entry under "Start > Programs > IPG > <Current Version >".

You should choose "C:\IPG" as installation library. It is not compulsory but suggested as otherwise some changes must be done in the data sets of IPGKinematics. Further information can be found on the IPG website as well as on the information paper which was downloaded together with the installation files. If this information is not sufficient please feel free to contact the Formula CarMaker team under **FormulaCarMaker@ipg-automotive.com**.

The license file

Once you have installed the software, you have to request a license file suitable to your machine. As long as you have not installed a valid license file, IPG products like CarMaker will not be able to start. To increase the intelligibility, this document uses "CarMaker" as a synonym for all IPG products. To order a license file and register the node, please visit our website and use the license key request form for FormulaCarMaker in the Support area:

https://ipg-automotive.com/support/licenses/license-formula-carmaker/

As part of the sponsorship each team receives the license file with two registered nodes. Each node is created directly on some specific PC information and thus is bound to this very Computer. Thus make sure that the workstation you choose is really the best one for CarMaker since you will not be able to change it easily once you have received the license file.

Each license is valid until the end of the current season (October 31st). After expiry each team can register once again to get a new license file. Please contact the FormulaCarMaker support team via **FormulaCarMaker@ipg-automotive.com** in case of questions regarding the license.

The Cockpit Package

The ftp link you received also contains an installation package of our tool "Cockpit Package standard". It is an interface to connect CarMaker with a game controller such as a Logitech (G27/G29) or Thrustmaster (RS500) steering wheel. Please find further information about this tool in the Cockpit Package Reference Manual which is part of the zip package. To request a license for the interface please contact the Formula CarMaker team via **FormulaCarMaker@ipg-automotive.com**

Chapter 2
Formula CarMaker and IPG Toolset

2.1 Introduction

Chapter 2 will give you a short overview of the general handling of the programs. At first there is a description of Formula CarMaker's structure and functionality. Following to that a Formula CarMaker project is defined. After that an example shows how to build up and interpret a Formula CarMaker TestRun.

In this manner it will be made easier for the user to get used to the program so that one won't get problems in the handling of the program while generating an own model later on This section is followed by another example that gives you an introduction to IPGKinematics. Finally, the interaction of both tools is demonstrated by importing kinematic and compliance characteristics which were generated using IPGKinematics to CarMaker.

Please bear in mind that you always have access to the "User's Guide" and the "Reference Manual" which contain plenty of information about the two programs. This document is only supposed to be a supplement to these guidebooks. The reader will be reminded incessantly through references and cross-references to use these manuals. Many of the emerging questions and mistakes can be avoided and solved by the help of these documents.

Those documents are found via "Help > User's Guide" or "Help > Reference Manual". The "Help" - menu contains further documentation regarding the main IPG tools e.g. IPGMovie or Control.

2.1.1 CarMaker

Before starting to simulate with CarMaker, we should explain the general structure of CarMaker, the meaning of the virtual vehicle environment (VVE) and the CarMaker Interface Toolbox (CIT).

CarMaker is a tool that is used to simulate the vehicle dynamics of fourwheeled cars. It's one of IPG's products and it was designed for the simulation of passenger cars. By means of mathematical models a Virtual Vehicle Environment (VVE) is created. The VVE simulates vehicle, driver and road including wind, obstacles, traffic signs etc. These make up all the parts neccessary to evaluate a controller or to test the dynamics of a vehicle or a vehicle subsystem.

Using the slightly modified tool CarMaker/HIL with an identical functionality, database and appliance have the possibility to test real ECUs in a virtual vehicle environment (= hardware in the loop).

The second part of the CarMaker system includes all the tools that are used to manage the VVE. These tools will do things like: start and stop a simulation, select vehicle parameter data, define a vehicle maneuver, display results, show the progress graphically or as an animation, send and receive messages from the VVE, etc. We can call these tools the CarMaker Interface ToolBox (CIT)

The tools can be classified as:

- Control and Direct Access Tools - control the actions performed by the simulation (e.g. start, stop, etc.) and also allow certain parts of the simulation to be directly controlled by the user (e.g. direct variable access).
- Parameterization Tools - serve to specify the parameters that will be used in the VVE.
- Analysis and Visualization Tools - allow the data to be viewed and analyzed either during or after a simulation.
  - IPGMovie for visualisation of the virtual vehicle environment
  - IPGControl to plot data for analysis
  - IPGInstruments to provide important information regarding the state of the car

Controlling and parameterization is done via the GUI (= Graphic User Interface). The first step using the GUI would be to edit all the parameter fields to define your model. Then you can perform a simulation (i.e. perform loop calculations) so as to predict the behavior of the car and check the results.

Please refer to the "User’s Guide" and the "Reference Manual" in order to prevent problems and questions. Both are to be found via the "Help" - menu in the main GUI of CarMaker.

2.1.2 IPGKinematics

Basically, IPGKinematics is a calculator that describes the movement of the wheels in space (it describes its kinematics tables). It calculates the corresponding forces necessary to trigger the movement and thus determines the entire kinematics, steering kinematics and compliance of all suspension types. It can be seen as a virtual axle kinematics test bench.

For data processing and analysis there is a tool to generate diagrams and plot various curves. Another option allows editing the gained data so that it can be exported to several vehicle simulation tools, such as IPG CarMaker.

Controlling and parameterization is also performed via the graphic user interface. Using IPGKinematics, at first you have to define all the parameters for the simulation control. Secondly enter all the values needed to parametrize your axle ("Input Data"). In a third step perform a simulation to calculate the wheel positions and forces. Finally you can analyze and export the results.

In order to guard against problems and questions, please note that the manuals contain a lot of information about the tool.

Chapter 3

Getting Started with Formula CarMaker

3.0.1 Opening CarMaker

After installing CarMaker on your computer and having ordered the license file, the tool can be started for the first time.

To start CarMaker under Windows press the "Start" button and select "Programs > IPG > CarMaker".

To start CarMaker under Linux, open a shell like xterm. Type in the command

```
CM &
```

The CarMaker main GUI pops up and automatically loads the project folder you were working in before you shut down the program the last time. You can easily switch the project folder by selecting "File > Project Folder" and select the one you would like to work in.

Exercise

- Start Formula CarMaker:
  Follow the description above for your operating system.

3.0.2 CarMaker Main GUI

Once CarMaker has started, the main window opens (Figure 3.1). It is the central platform where you can define and edit all input quantities, set the simulation speed and change whether the results should be saved or not. From now on this window will be referred to as the main Graphical User Interface (main GUI).

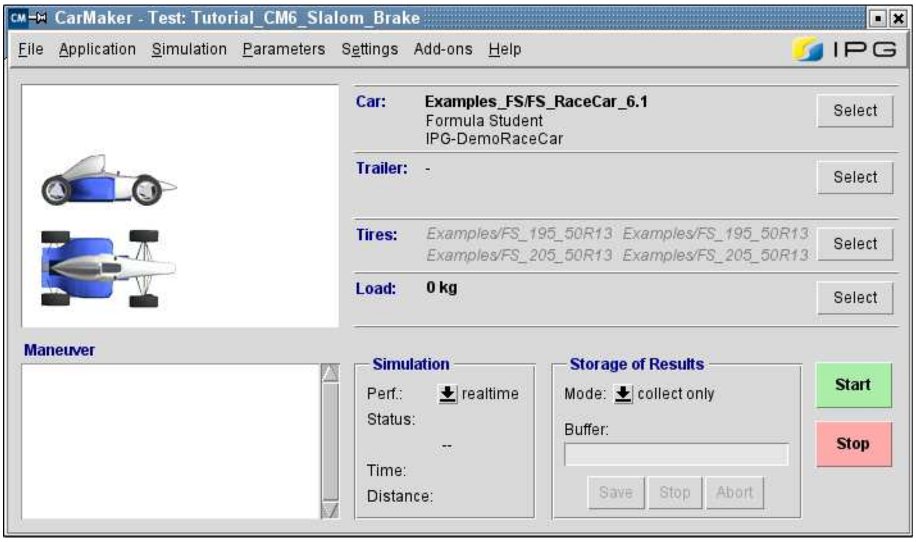


Figure 3.1: The CarMaker main GUI with tutorial TestRun:
Car: Chosen test car
Trailer: Chosen trailer model
Tires: Sorted by position. Gray is default by test car; black is assigned by the TestRun or user
Load: Assigned load in specific positions
Maneuver: List of maneuver steps in the TestRun

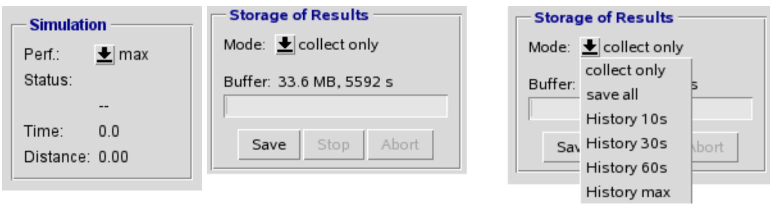


Figure 3.2: Setting the speed of simulation and storage behavior in the main GUI

Speed of Simulation

There is plenty of information and many buttons in the main GUI. One is the simulation speed (see the simulation selection area in the main GUI). In general, you can choose any simulation speed since it is a purely computational simulation (i.e. the computer performs the calculations at any speed since there is no real component in the simulation loop). You can also change this speed during the simulation.

Note that this speed depends on the power of the processor and graphic card of your system (if IPGMovie is running, it may slow down calculations). The box also displays the time from the start and the distance covered.

Storage of Results

In most of cases it should be sufficient to choose "collect only". Indeed, with this option results are only saved in a buffer, a temporary memory (and you have the option to save them to a file at the end if you need the results later again). If you use another option the results will be directly saved in a file and concerning the numerous simulations you will perform, the size of the folder containing the results files will become enormous!

File System

Formula CarMaker works within a project directory. In this project all the folders used by Formula CarMaker are stored according to a certain hierarchy that is identical for every project. Hence, CarMaker always knows where to find common files such as tire data, vehicle description and so on. The data itself, of course, differs from project to project.

Thus, you have to choose a project folder after opening CarMaker, because you have to tell the software in which project folder you would like to work in. Of course, you can also create new project directories containing your own data, provided that the structure of the directory remains the same. Therefore choose "File > Project Folder > Create Project". In the pop-up window you can enter the path where the new folder should be created. This new project folder contains the folder hierarchy needed by Formula CarMaker to find common files.

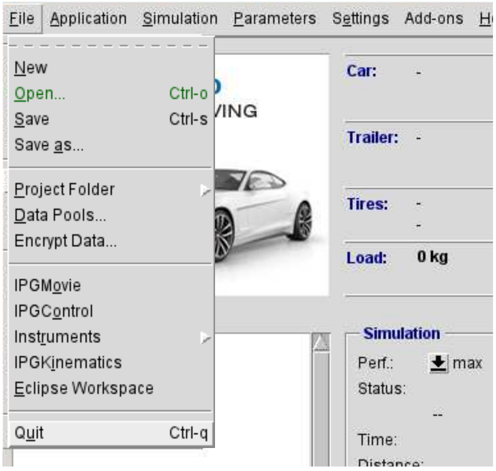


Figure 3.3: Selecting a project folder

We highly recommend **not** to create a variety of project folder, but to create subfolders in the specific folder inside one big project folder. For example you could create one project folder each season and inside the project folder you create a subfolder which contains all closed race circuits and another one which contains the scenarios for braking and acceleration.Folders can be add via the *Select Project Folder* dialog or in your explorer.

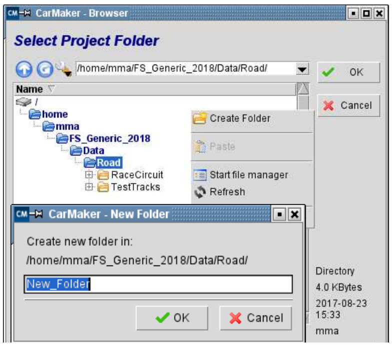


Figure 3.4: Creating subfolders inside the project folder

Please refer to the chapter "Directory Structure" in the User's Guide ("main GUI > Help > User's Guide") for further information about the directory and e.g. shared data pools.

Exercise

- Create project folder:
  - Download the "FS_Generic_2018.zip" and extract the archive to your desired work directory.
  - In the main GUI choose "File > Project Folder > Select" and select the previously extracted project folder

3.1 CarMaker TestRun

CarMaker is based on fixed models (vehicle, suspension, tires, etc.) whose properties (e.g. values for the mass of each body or spring stiffness) can be varied. This means that the number of bodies and the DOF between them are already defined and the user doesn?t need to model these on his own. If the user wishes to extend a model, CM4SL is recommended. For further information on this topic, please refer to section "CarMaker for Simulink"

The models themselves are already defined, but they still need to be parameterized according to their environment. For this, a so called data set is manually implemented or loaded for each of the models. Parameterizing includes selecting a vehicle, selecting or designing a road, defining a type of driver and defining a maneuver. After all of these components are set, CarMaker has the information necessary to control the virtual vehicle environment (VVE) and simulate.

All of these settings are stored in a file used by the VVE during simulation. Said file, whichcan be saved, loaded or edited, is what we call the TestRun definition.

In summary: a TestRun represents a test scenario in which all parameters of the virtual environment (vehicle, driver, tires, etc.) are sufficiently defined. As a minimum requirement to be able to simulate, the following modules have to be parameterized within the TestRun:

- Vehicle plus tire’s - definition of the vehicle data set used.
- Road - Parameterization of the test track.
- Maneuver - mainly to specify the driver’s task.
- Driver - set driver behavior (defensive, normal, aggressive or racing driver)

In CarMaker the preparation for a vehicle test is very similar to how it would be done in thereal world.

- 1) Select a vehicle .
- 2) Choose or define a test track with obstacles and the desired conditions.
- 3) Specify the tires and loads.
- 4) Define the Maneuver(s).
- 5) Specify the type of driver or use a simple closed loop controller.
- 6) Save & Run the TestRun.
- 7) Analyze the TestRun

Going ahead, we will create one TestRun to simulate and analyze the braking and slalom ability of our race car.

For further information regarding TestRuns in general see the User’s Guide ("main GUI > Help > User’s Guide").

3.1.1 Vehicle Model

Instead of individually setting every parameter for all CarMaker subwindows, we will load a predefined model. In this chapter the "FS_RaceCar_6.0" vehicle data set will be used. This dataset represents a common Formula Student racing car from season 2011.

To use a different vehicle, click on the "Select" - button and choose a different one out of the given product examples. Under "Project > Data > Vehicle > Examples_FS" you'll find a variety of predefined race cars. In section "Preparing a vehicle dataset in CarMaker", pg. 76 it is explained how to create an individual vehicle model.

Exercise

- Open a new TestRun via "File > New".
- Choose the test car:
  - Press "Select" in the *Car* - section of the main GUI. Browse through the files and choose "FS_RaceCar_6.0".

3.1.2 Tire Model

In this TestRun we will use a pre-defined tire dataset, in section 5.8, "Creating a tire dataset using IPGTire" it is explained how to create your own tire dataset using IPGTire. The FS_RaceCar_6.0 contains a preconfigured data set for tires which is a global setting. This means, unless you change it, the selected vehicle will always contain the same tires. The gray font in the *Tire* section in Figure 3.5: is the indicator that you use the predefined data set from the vehicle.

But if you want specific tires for a particular TestRun only than the best way to do it is via the main GUI. In Figure 3.5: you can see the main GUI and the "Select" - button in the *Tire* section. By clicking this button you are able to choose the same tire for each wheel. To select different tires for front and rear wheels, click directly on the name of the tire in the main GUI and make your choice (in this case **"**FS_205_50_R13**"** for the rear).

By hovering the mouse over the four tires shown in the tire selection area of the GUI an a ballon help will appear, describing which tire (front right, front left, rear right, rear left) will be modified.

In this example, the tire "FS_195_50R13" will be chosen for the front and rear tires.

Exercise

- Assign the Tire:
  - By using the "Select"-button assign the **"**FS_195_50R13**"** - tire from the available example models to all four wheels together.
  - Save your first TestRun as "Tutorial_CM6_Slalom_Brake" by selecting "File > Save as" in the main GUI.

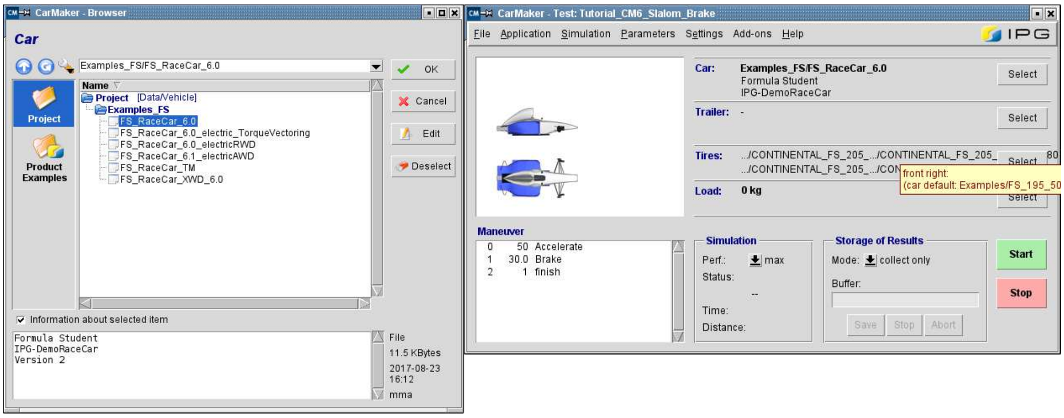


Figure 3.5: Left: Selecting a FSAE vehicle example
Right: Selecting a single tire in the main GUI

3.1.3 Scenario Editor

In CarMaker, the Scenario Editor is the graphical user interface that enables the creation of highly complex road networks for vehicle and driving simulations. It is accessible via the CarMaker GUI by clicking the "Parameters" menu and selecting "Scenario/Road".

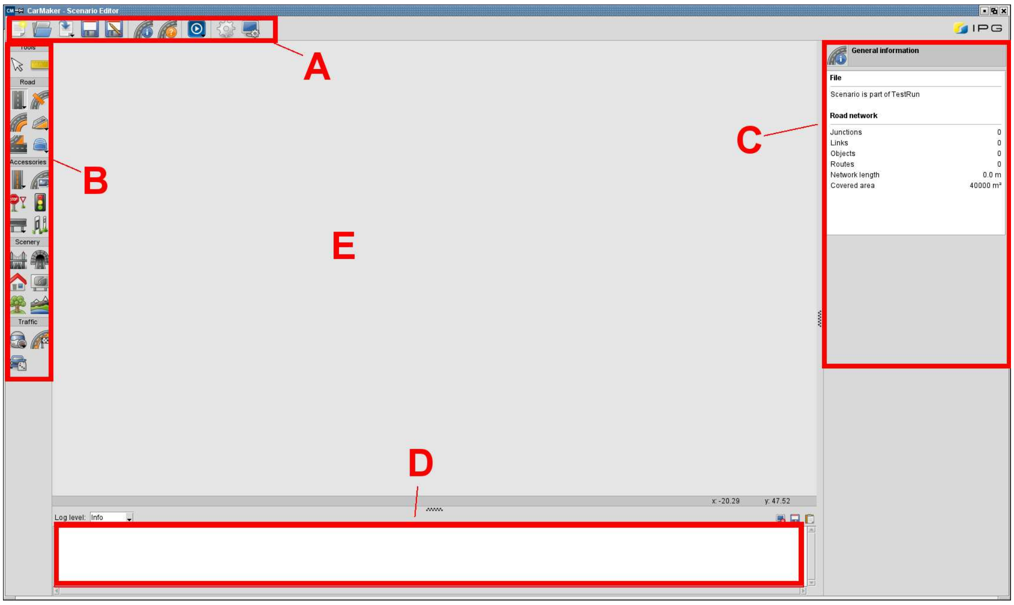


Figure 3.6: Scenario Editor - GUI:
A - Menu tab for accessing, saving and viewing information about the scenario
B - Tool panel containing the tools to create the road network and environment
C - Definition dialog lets you view and edit the data
D - General information section about the state of Scenario Editor
E - Drawing area

Exercise

- Open the Scenario Editor:
  In the main GUI, click on "Parameters > Scenario / Road" or press "Ctrl+r".

Useful Tips in Scenario Editor

Before we start, please take note of the controls in the Scenario Editor. There is a heavy use of both the mouse and shift button. i.e. the road icon has a context menu which you can call by hovering over the icon and holding the left mouse button (LMB) or by pressing the right button (RMB) in the drawing field.

If you want to place a straight or curve segment you can almost always press "shift" on your keyboard to lock certain values to a constant step size.

With the "Selection"-tool active the LMB is also used to select a drawn segment. Simply hover over the desired area and click the LMB. A context menu will open where you can choose between the different objects. These are sorted and tabbed according to their child-parent-relationship.

A very useful tip is to get familiar with the use of the different manuals we offer. E.g the Scenario Editor has a comprehensive and useful chapter in the User's Guide ("main GUI > Help > User's Guide"). Please take a look at chapter "Scenario Editor" for detailed Information about all icons, tools and procedure.

In the following chapters you will learn to build a course with two turns, a slalom parcours and braking test area. The course will give the opportunity to experience the scenario editor, test manager and use of DVA (direct variable access).

The decimal separator in all CarMaker and IPGKinematics applications is dot (.) instead of comma (,). If there is any error message related to inconsistent metrics most of the time it is due to wrong decimal separator.

Table 3.1: Basic terminology of IPGRoad

Name	Description
Junction	Road element used to join different Links
Link	A section of the road network, usually one road, and can be further specified using one or many Lane Segments.
Reference Line	The course of a Link is defined by its Reference Line.
Lane	Lateral extension of a Link
Lane Section	Sections of a Link with constant number of lanes
Elevation Profile	Lateral, longitudinal and camber profile changes
Route	Reference line defining the course of the vehicle travel on the road
Path	Actual course of the vehicle travel on the road

Table 3.1: shows you the terminology behind the theory of the IPGRoad. For further information please take a look in the chapter "Theory of IPGRoad" in the User's Guide ("main GUI > Help > User's Guide").

Menu tab - Field A

The tab on the top most part of the Scenario Editor GUI offers general file operations such as loading road files, saving them and modifying general settings for the GUI. All icons are explained in section "Building Roads with Scenario Editor", in User's Guide ("main GUI > Help > User's Guide").

- Import road definition - With this option, a Road network can be imported from either an external Road InfoFile or from a TestRun.
- Save Save TestRun - This option has two functionalities. When a road network, that is defined as part of a TestRun, is being worked on, the entire TestRun can be saved. However, if the road network that is being worked on, is a Road InfoFile with the file format *.rd5 or *.road, then this button saves the changes made to the Road InfoFile. Scenario Editor automatically identifies whether the road network is part of a TestRun or a separate Road InfoFile.
- Save road file as - This option is a "Save as" function. It always saves the created road network as a CarMaker Road InfoFile.
- 3D Preview - With this option, the 3D view of the road network can be seen in IPGMovie.
- Scenario Settings - In this option general settings for the scenario can be set. The parameters that can be defined here are described in the User's Guide ("main GUI > Help > User's Guide").

Tool Panel - Field B

The tool panel itself contains a variety of tools which are gathered in sections:

- Tools - different attributes of the selected road section can be activated and measured .

- Road - create the basic road layout , modify lanes and add road bumps .
- Accessoirces - road markings , guide posts , road paintings traffic signs - lights , -barriers .
- Scenery - enable beautification of the simulations environment . All objects inserted here have no effect on the simulation itself, are ignored by IPGDriver and cannot be detected by Object Sensors.
- Traffic - The features in the Traffic panel generally have an effect on simulation participants, such as the test car or traffic objects. A few markers affect solely the test car , while others also have an effect on traffic participants . The option Route is used to generate multiple routes on a road network, defining the path on which the vehicle will drive.
- All icons are explained in section "Building Roads with Scenario Editor", in User's Guide ("main GUI > Help > User's Guide").

Markers

A useful feature are the Markers. Markers influence the behavior of IPGDriver, the driving maneuver and the vehicle.

Driver Stop This marker places a STOP sign on the road.

Pylon Alley This type of marker inserts pylons on the road. These are always created in pairs. The Driver always tries to drive between them.

DrivMan command Includes a minimaneuver command using the road distance as a trigger. All available Minimaneuver Command Language commands can be used. A description of all commands can be found in section "Minimaneuver Command Language" in the User's Guide ("main GUI > Help > User's Guide").

DrivMan jump This gives the opportunity to enter a specified maneuver with a start offset of the current road segment as the trigger point.

Trigger point Using the road marker TriggerPoint, different actions can be executed at a given position sRoad on the road.

User - defined User defined markers can be used to place a marker with a list of user defined parameters along the road.

Building a Testtrack

The next seven exercise steps will create a closed circuit. It will provide us with a braking area, slalom course and 75m acceleration track.

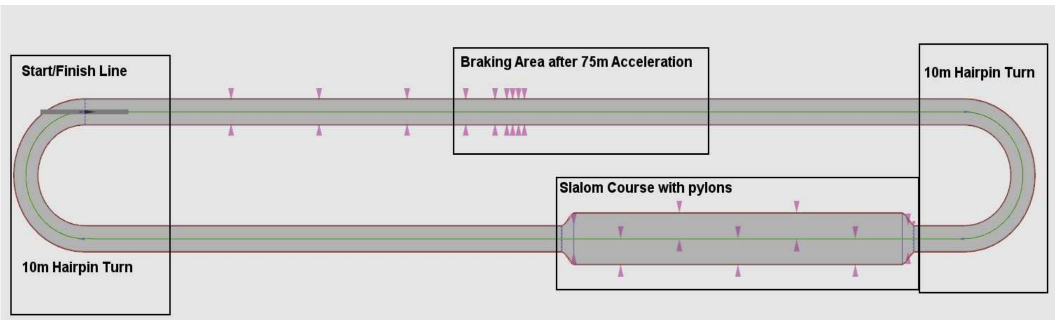


Figure 3.7: Closed Test Track with Braking Area and Slalom course

Lane Configuration

The Formula Student rules say that the road has to be at least 3.5m wide. To get familiar with the icons and to create a default configuration which fits the requirements we have to edit the lane configuration before we start. With this option, different template lane configurations can be created and saved.

Before creating a new road segment, one of these lane configurations can be selected and the road segment will have this lane definition. However, selection of the lane type has to be done before generation of the road segment. If a lane configuration has not been selected, then the top most lane definition is automatically selected to create the road segment. The different lane configurations that are created by the user is saved as a part of the Project folder. Hence, if a new project folder is created, the lane configurations have to be created anew. Later, we'll learn how to change the width to single segments. In Figure 3.8 you see the lane configuration dialog.

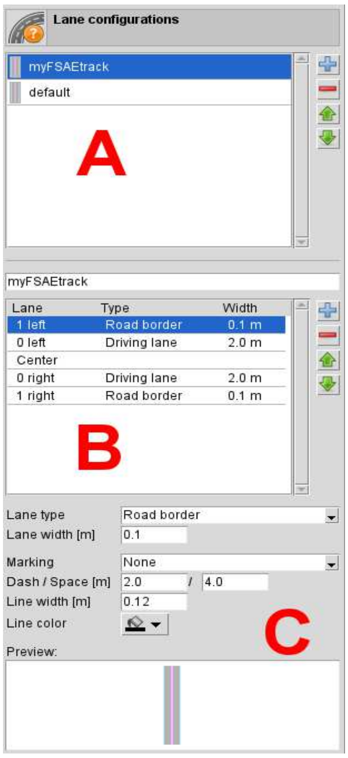


Figure 3.8: Road model to fit FSAE track rules
A: List of existing configurations
B: List of lanes in selected model
C: Preferences for each lane

Exercise - Step 1

- Add a new lane configuration:
  - Open the dialog via "Lane configuration" . Press the - button (Figure 3.8 - section A) on the right side to add a new configuration and rename it to "myFSA-Etrack".
  - Press the - button (Figure 3.8 - section B) to add five lanes in total (Two left, One center, Two right).
  - Set 2.0m to the inner left and inner right driving lane. Set both outer lanes as a road boarder with 0.1m.

- Remove the marking in the center of the road by selecting the "center" - lane and set "none" in the dropdown menu next to *Marking* (Figure 3.8 - section C).

As mentioned before the new configuration will not overwrite already existing segments, but it will be the default configuration of every new segement within this project.

Building Road Segments

The next step is to create the road itself. We'll learn how to get straight and turns and how to connect them. In Figure 3.9 you can see the result of this exercise step. While building roads with Scenario Editor there are a few hints which you should mention:

1) The first point that is set is the origin of the global coordinate system.

2) If you did not hit the required length at first try, you can edit it afterwards simply by activating the drawn road segment with the "Selection" - Tool.

3) There are two possible situations of connecting road segments. The first one occurs when you want to place a road segment connected to a previously drawn segment one (e.g. a turn after a straight). Then you have to select the tool (e.g. "Turn" - tool) from the tool panel and move the cursor to the middle line at the end of the first segment. An orange circle (see Figure 3.11) appears. Click the LMB and the start point of the new segment is connected to the first segment.

The second situation is mostly used in case of a closed loop circuit (race track). The "Connect" tool is to connect two segments which are close together. This tool tries to find the shortest possible path between two segments.

Exercise - Step 2

Figure 3.9: Simple closed loop circuit with 362m track length

- Place the Straight:
  - Left click and hold the icon called "Road segment" (see Figure 3.10) until a dropdown menu opens. Select the road segment tool: "Straight" .
  - Use the cursor to navigate into the design area of the editor (the middle window, see Figure 3.6 - section E) and click once to define the starting point of the straight segment.

- While drawing a line, hold shift and click again to create a horizontal, straight road segment that is 150m long.

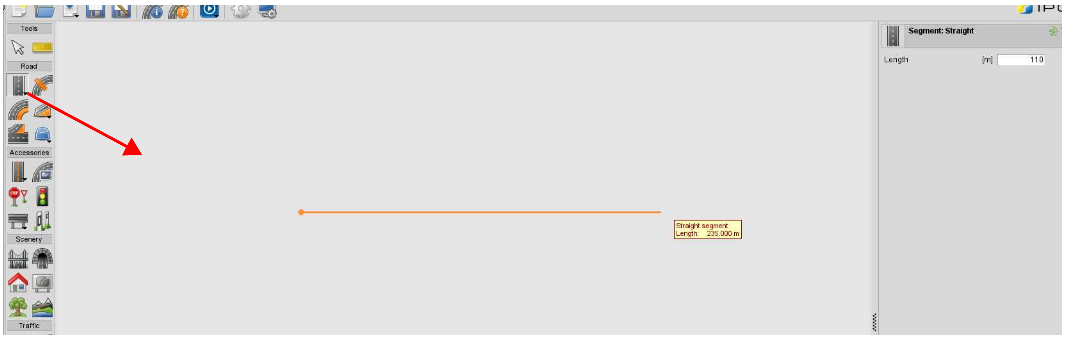


Figure 3.10: Defining a straight road

- Place a 10m radius hairpin turn:
  - Select the "Turn" - Tool  by right click in the design area and choosing from "Road Segment". Make sure to connect it to the end of the first segment (orange dot inside the green marking).

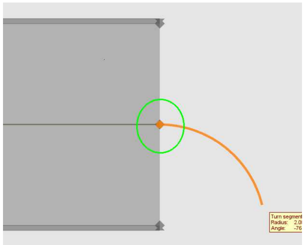


Figure 3.11: The orange circle is the feedback for a correct connection of two road segments (The green marking is only for visualization in this tutorial)

- Place the second straight (150m):
  - Pick the "Straight" - Tool and connect (orange circle) to the previously drawn turn to place another horizontal straight (parallel to the first one).

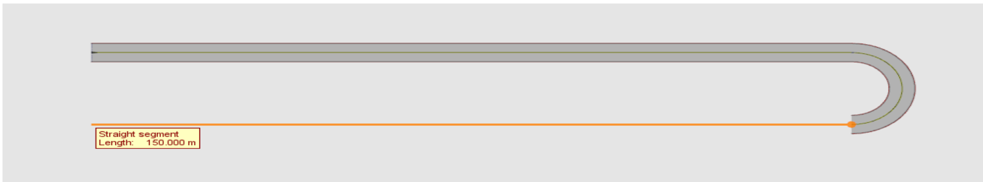


Figure 3.12: Placing the second straight

- Place the second and last hairpin turn (10m) but with <u>175 degrees.</u>

Figure 3.13: Placing the second turn, connecting the end of the second straight

- Connect the last turn to the beginning of the very first straight:
  - Choose "Connect" via the "Road Segment" - drop-down menu and connect the last turn with the first straight. Wait for the two orange circles to appear
  .

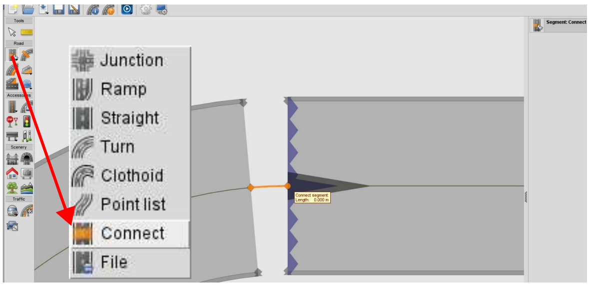


Figure 3.14: Connecting the road segments

Path and Route

Now, in order for the vehicle to be able to drive on the road, a route and corresponding path are necessary. The "Route" - tool generates a path given by the user and the road network.

The path has a variety of settings which can be edited (e.g. driving side - left, right or center or in the center of the road) via the "general settings" - icon . Those settings are saved in the TestRun - file. The next step will generate a path in the center of our circuit by commanding the route which the test car has to follow. The generation is done by selecting the corresponding Links. Please take note of the following hints for route generation:

1) Figure 3.15: shows the steps of route generation and the corresponding overlay color.

2) Renaming the route is possible, after selection, in the tab on the right side of the Scenario Editor.

3) The balloon help will show you how many Links are added and how long the path is.

4) Available reference lines are marked in dashed, orange lines and need to be clicked once. To set and complete the route definition, double click anywhere until the reference line becomes yellow.

5) You can check the results by clicking the "Preview" - icon .If your route is correct you should be able to move the yellow marker along the path using the slider in IPGMovie.

6) The Road is saved inside the TestRun and you can save it seperately via .

Exercise - Step 3

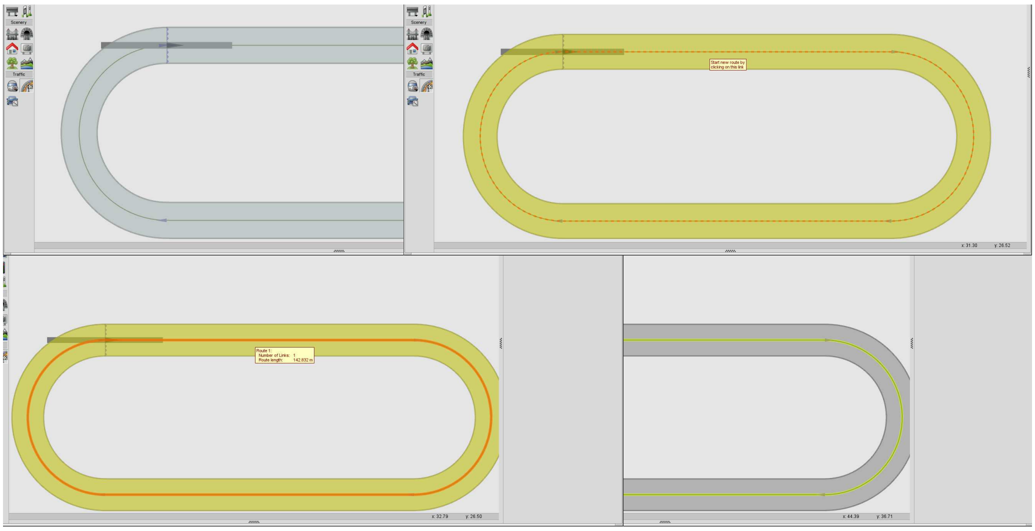


Figure 3.15: Route generation step by step:
Top left: Activation by hovering over road (grayish overlay)
Top right: Left click on the road to assign this link to the route (orange, dashed line)
Bottom left: Assigned route around the circuit (orange, full line)
Bottom right: Finished route generation by double clicking anywhere with LMB (yellow Line)

- Change to center driving:
  - Click the "Settings" - icon in the menu and make sure that the *Driving side* is set to "Center (Full width)". If this isn't already the case, change the driving side.
- Add a route to the road:
  - In tool panel > *Traffic* section, select the "Route" - icon . using the cursor and hover over the link which will be highlighted with a grayish layer.
  - Now the appendant *reference line* needs to be chosen. Click LMB on the highlighted Road and another click on the orange, dashed line will assign the path to the route.
  - To finish the route generation simply double click LMB anywhere outside the drawn segments!
- Save the road via under "myFSAE_Brakes_Slalom".

Brake Test Area

This is an extract from the Formula Student Rules 2017:

> "Brake Test Objective - The brake system will be dynamically tested and must demonstrate the capability of locking all four wheels and stopping the vehicle in a straight line."

For the braking area we'll add eight pylons. The braking will begin as soon the vehicle hits the 75m mark. The corresponding maneuvers are configured in section "Maneuver", pg. 31. Some hints for placing pylons:

1) If you want to create a pylon alley with constant width, you should set the parameters right after creating the first pair. This will ensure that the following pylon pairs have the same width.

2) At 70-75m the higher density of pylons helps to make sure that the car is straight and perpendicular on the road. The maneuver steps are going to be defined in the later exercises.

3) The pylons only work in one chosen direction. The direction is shown as a small arrow at the beginning of the Link (see Figure 3.14:).

Exercise - Step 4

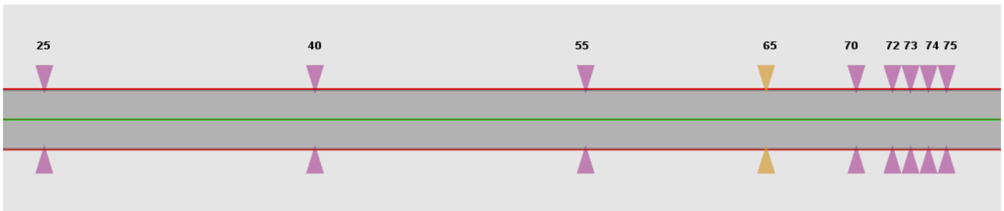


Figure 3.16: Braking test pylon placement

- We'll start with the "Pylon alley" - tool :
  - Select them via the "Markers" dropdown-menu (click LMB and hold for context menu) in the *Traffic* section.
  - Place the first alley in the center of the road, 25m away from link start. Define 3.5m as width and place seven more pylons as shown in Figure 3.16.

Slalom Course

The slalom course is a must have on any FSAE Autocross race track. In Hockenheim, known as the Pendulum, we try to create something similar to check the lateral accleration. In the Pendulum there are sets of straight pylons which one has to drive by, not through.

In our case, we want to keep it simple and we'll define the slalom as a pylon alley. The next step is to get the space we need to place the pylon alley. The wider area is 60m long and we'll place seven pylon pairs.

1) The end of the lane section is defined by the start of the following lane section.

2) With every major change to the road network, CarMaker needs to regenerate the route.

Exercise - Step 5

The intermediate result of this exercise is a circuit with three seperated sections of the circuit:

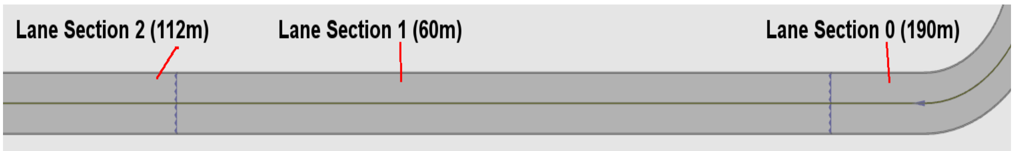


Figure 3.17: Lane Section 0: From Start (0)m to 190m
Lane Section 1: From 190m to 250m
Lane section 2: From 250m to End (362m)

- Create the lane section:
  - Pick the "Lane section "- tool, go to the second straight, right after the turn, and press the LMB to select the road segment. Another click on the road with the LMB defines the start of the lane section.
  - Choose the "Selection" - Tool to activate the new Lane Section and define the starting point of it. Set the beginning to 190m (*Start offset*) and define the end of the section with an offset of 250m to the start. The result should be a 60m long individual road section.

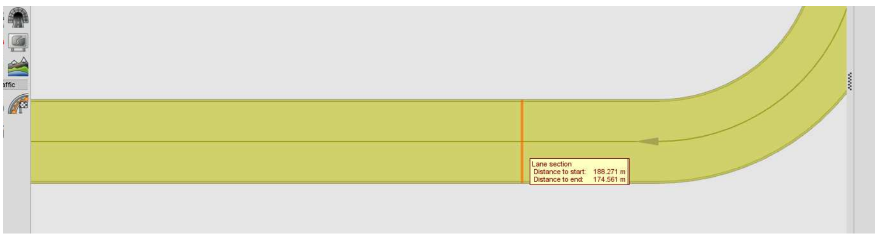


Figure 3.18: Orange line appears when intersecting a lane with "Lane Section" Tool

Lane Width

The next step is to change the widht of the newly configured 60m long "Lane section 1". The goal is to have a 8m broad track and a smooth transition from narrow to wide. This is done in the definition dialog, after selecting, on the right side of the Scenario Editor.

1) The "Lane" - tool is capable of adding new lanes to the section. This is not what we want, we just want to edit the parameter of the already existing lanes.

2) To achieve a wider road plus a smooth transition one has to edit the parameters on the right side after activation. For further information, please refer to User's Guide ("main GUI > Help > User's Guide"), section "Parametrization: TestRun > Scenario Editor > Lane"

3) With every major change to the road network, CarMaker needs to regenerate the route.

Exercise - Step 6

This step will provide us a wider road segment within the TestTrack:

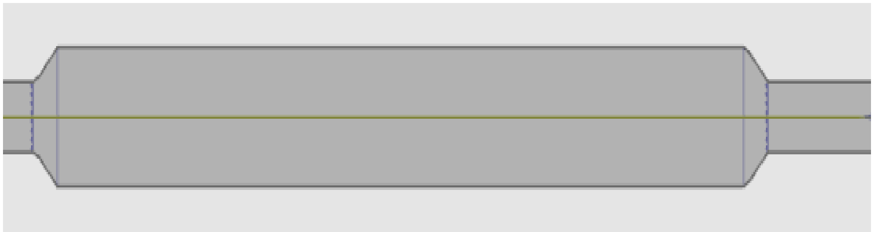


Figure 3.19: Wider road to have space for the car and pylons

- Create a wider road:
  - Pick the "Lane" - tool from the "Road" section and activate one lane from the previously created "Lane Section 1" (see Figure 3.17).
  - In the defintion dialog (Figure 3.6 - section C) you can edit the *Point list for lane width (optional)* and follow the settings shown in Figure 3.20.
  - Repeat the steps for the other lane in the section.

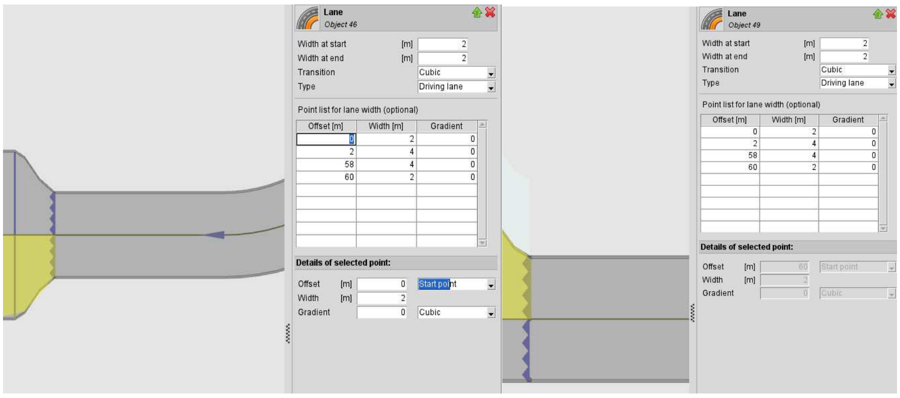


Figure 3.20: Configuring both lanes to fit the requirements

Pylon Placement

A slalom course is a good way to test the dynamics of the race car.

> "The autocross track layout is a handling course built with the following guidelines: Slaloms: Cones in a straight line with 7.5 m to 12 m spacing" - FSG Rules 2017

To keep it simple we'll place the pylon pairs with a width of 3.5m for a length of 10m along "Lane section 1".

1) The placement of the pylon can vary. Either you chose center and add an offset to the position or you chose the lane on left or right side. For this step we will place two (first and last) in the center (Zero offset) and the rest will alternate between Lane Right/Left.

2) If you want to create a pylon alley with constant width you should set the parameters right after creating the first pair. This will make sure that the following pylon pairs have the same width.

3) As mentioned in previous tips: The pylons are only working in one chosen direction. The link direction is shown as a small arrow in the beginning of the link.(see Figure 3.14:).

4) Third time is a charm: With every major change to the road network, CarMaker needs to regenerate the route.

Exercise - Step 7

In step 7 the pylon pairs for the slalom course are placed:

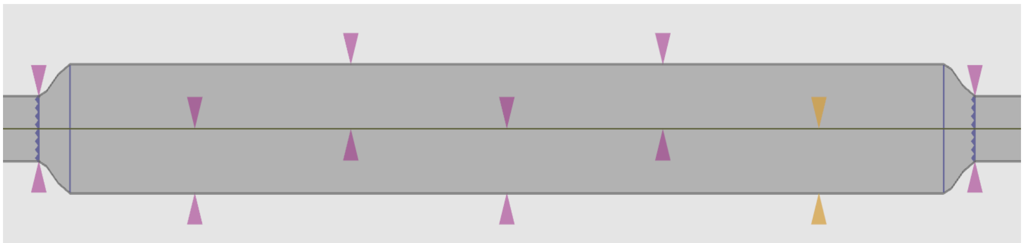


Figure 3.21: Finished Slalom course

- Place the Pylons:
  - Pick the "Pylonalley" tool from "Markers" in the *Traffic* - section of the tool panel.
  - Put the first pair of pylons in the center right after the beginning of the "Lane Section 1" (190m from Link Start). Set the width to 4m for the pylon pair, and make sure the *Validity* is set to "With link direction".
  - Add six additional pylon pairs with 10m spacing between each pair. The lateral offset of the second pair has to be set to 0.0m on the *Lane left 0* from the dropdown-menu. Accordingly the third pylon pair has to be placed on *Lane right 0.* Alternate the lane for the next pylon pairs. Set the last pair to the center with zero lateral offset.
- Regenerate the Route:
  - Pick the "Route" - tool and right-click on the old route and select "Generate new path". Save the roadfile and TestRun.
- Preview in 3D:
  In the menu tab press the - icon to load the new scenario in IPGMovie. If the route is correct, you're able to move the camera along the route. Use the slider in IPGMovie

to do so. More information about IPGMovie is to find in User's Guide ("main GUI > Help > User's Guide").

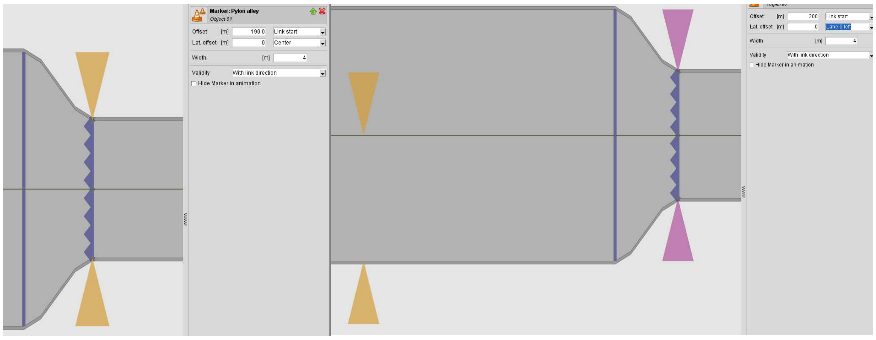


Figure 3.22: Placement of the pylons for slalom course in step 7
Left: First Pylon pair configuration and placement
Right: Placement of the second pair on "Lane Left 0"

This was the last step to create a test track. After saving, feel free to close Scenario Editor.

3.1.4 Maneuver

A maneuver describes the drivers actions. You can define several successive maneuver steps (mini maneuvers) along the track. The braking test requires a special manuever which we will define in this chapter. The path through the slalom course will be commanded by the pylon alley. Ergo, no special maneuver is required, since the driver will know what to do. You can open the maneuver window in the main GUI by clicking on "Parameters > Maneuver" or "Ctrl+m".

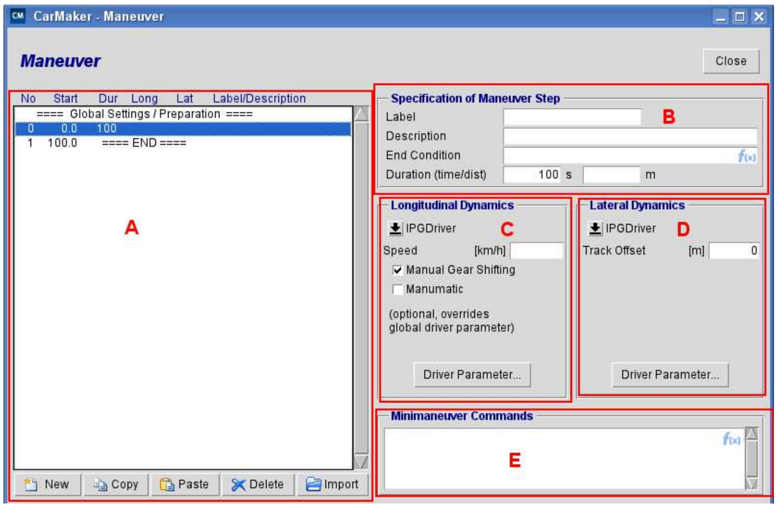


Figure 3.23: CarMaker Maneuver Window
A: Overview of the steps
B: Duration of the single steps
C & D: Dynamics definiton with different controller
E: Minimaneuver commands and Real Time Expressions

Maneuver Definition

A new maneuver step is added via the "New" button below the *Global Settings / Preparation* (section A) field.

Different values for the maneuver can be specified in the box called: *Specification of Maneuver step* (section B). A *Label* is an alias to the maneuver step.

Please bear in mind that the label must not contain numbers at the beginning, since it would lead to errors in the simulation. In the *Description* Field you're free to write a brief description of the maneuver step.

Two simple ways to define the duration are by time or length. If both the parameters are specified, the simulation stops when one condition is fulfilled. If only one field is initialized the second one will be disregarded.
The third way to end a maneuver step is via user defined end conditions. Therefore, you have to click RMB in the *End Condition* field. Now you have access to all CarMaker variables (see section "Data Access", pg. 41).

Once you select one, you can tie it with a logical operator to create a condition. If this condition becomes true, the maneuver step is terminated and the next in order is started.

At the bottom, *Minimaneuver Commands* enables you to specify several maneuver commands such as maneuver jumps, creation of new quantities and variables, calculations, etc (see appendix "Real Time Expressions" in User's Guide).

To find out more about the maneuver definition, please read the User's Guide ("main GUI > Help > User's Guide"), section "Parameterization: TestRun > Maneuver".

Dynamic Controller

The maneuver parameterization is divided into two parts: *Longitudinal Dynamics* and *Lateral Dynamics*, which respectively control the (throttle, brake, clutch) pedals and the steering wheel. To learn more about this, please read the User's Guide ("main GUI > Help > User's Guide") chapter "Parametrization: TestRun > Driver" or the following chapter in this tutorial.

The goal of the next exercise is to define several maneuver steps. The first step #0 will tell the driver to accelerate the car for 75m (see (EQ 1)). At this point on the track our last pylon pair of the braking area is set. Exactly at this point we want to start the braking test which is defined by the following maneuver step #1.

For step #1 we'll not use the IPGDriver, but the GBCP (Gas, Brake, Clutch and Gear-Number) - controller which offers the option to predefine percentage of pedal actuation. We need to set the clutch (won't kill the engine) and brake (standstill and block) both to 1.0, and gas has to be 0.0. Since the clutch is fully pressed, we don't need to set the gears.
(EQ 2) describes the end condition, via Real Time Expressions, for the brake step and fullfills part of the rule:

> "The brake system will be dynamically tested and must demonstrate the capability of locking all four wheels and stopping the vehicle in a straight line" - FS Rules 2017

Later (chapter 3.2), you'll learn how to check the vehicle offset to the centerline and rotation around z-axis (yaw) to determine whether the vehicle is in a straight line.

Step #2 will accelerate the car to a certain value (user driver) or to maximum possible velocity (racing driver). It will be active until the end of the first lap (see (EQ 3)). Please note that this quantity (EQ 3) is an integer and starts with 0 (zero).

1) The racing driver tries to stay in the center of the road but due to the car's perfomance he could break out. Later, you will be able to check the steering wheel torque in IPGControl which is an interesting value to check.

2) Please keep in mind that the CarMaker sampling rate is locked at 1000hz. There is no guarantee that a variable can have an exact value (e.g. 75.0000000m). Therefore we have to use "<=,>=" - logical operators, instead of "==".

Exercise

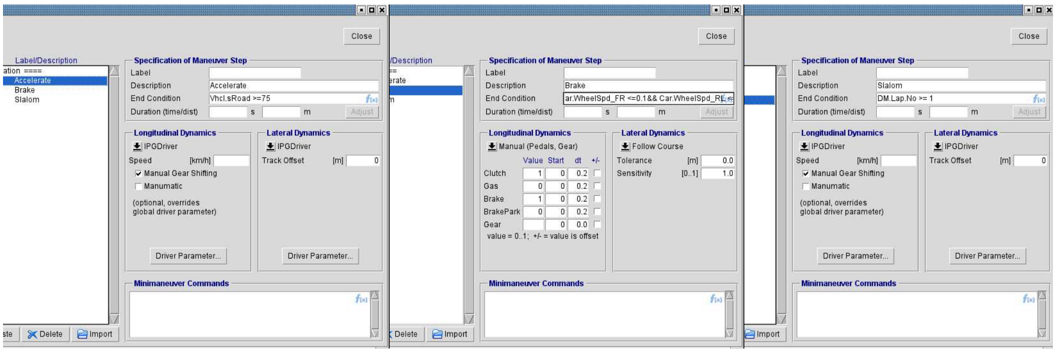


Figure 3.24: Maneuver dialog with successive maneuver steps
Left: Accelerate maneuver
Middle: Braking maneuver
Right: Slalom maneuver

- Create Maneuver #0:
  - Open the maneuver window in the main GUI by clicking on "Parameters > Maneuver". Add a new maneuver with the "New" button and write "Acceleration" in the *Description* Field.
  - Select the "IPGDriver" for both dynamic fields from the dropdown - menu. The end condition is:

$$\mathrm{Vhcl.sRoad} >= 75 \quad \textbf{(EQ 1)}$$

- Create Maneuver Step #1:
  - Add a new maneuver with the "New" button. Write "Brake" in the *Description* Field.
  - Select "Manual" controller from the dropdown - menu in the *Longitudinal Dynamics* field. The clutch and brakes have to be fully pressed, set the values as shown in Figure 3.24: - (Middle).
  - The lateral controller is set to "IPGDriver". The end condition is:

$$\mathrm{Vhcl.sRoad} >= 77 \;\&\&\; \mathrm{Car.WheelSpd\_FR} <= 0.1 \;\&\&\; \mathrm{Car.WheelSpd\_FL} <= 0.1 \;\&\& \\ \mathrm{Car.WheelSpd\_FR} <= 0.1 \;\&\&\; \mathrm{Car.WheelSpd\_RL} <= 0.1 \quad \textbf{(EQ 2)}$$

- Create Maneuver Step #2:
  - The third maneuver step is a simple acceleration maneuver with the lap number as end condition which quits the step and ends the simulation right after crossing the start/finish line:

$$\mathrm{DM.Lap.No} >= 1 \quad \textbf{(EQ 3)}$$

3.1.5 Driver Model

IPGDriver is a controller for following a course with a speed controller on a given track. It can be activated in the Maneuver dialog. IPGDriver enables you to add the control actions of a human driver to your complete vehicle simulation. These actions include steering, braking, throttle position, gear shifting and clutch operation. But you can decide to use IPGDriver only to control the course and not the speed or vice versa.

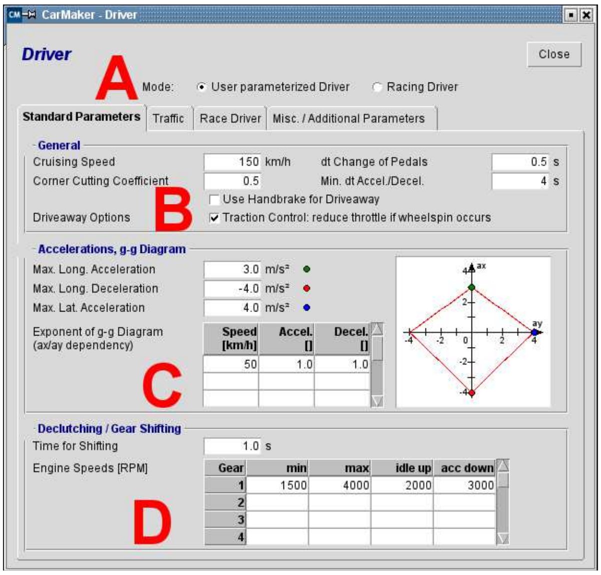


Figure 3.25: Driver dialog - Standard Parameter tab:
A: Driver mode selection in CarMaker
B: General settings such as target speed or corner cutting coefficient
C: Accelerations settings and g,g diagram
D: Shifting point definition

User Parameterized Driver

You have two different driver configurations to choose from. The first option is the "User parameterized Driver" which can be customized by a lot of parameters.

The *Cruising Speed* (see Figure 3.25 - section B) is the target speed which the driver will try to reach. In Figure 3.25 - section C) you find the g-g diagram. It describes the maximal allowed (purely longitudinal and lateral, or combined) accelerations as plots.

The panel in section D is to set the gear shifting values. For example, the shift time can be varied (the shorter, the faster). A common error is to set values for idle up/acc down which are smaller/higher than the engine minimum/maximum speed. For further information please read section "User parameterized Driver > Declutching/Gear Shifting" in User’s Guide ("main GUI > Help > User’s Guide").

This type of driver simulates the behavior of a real driver who can steer, accelerate, brake, slow down, shift gears, cut corners, adapt the driving style according to the track.

Racing Driver

The second type of driver is the "Racing Driver". He drives the car up to its physical limits and thus attains maximum power and speed. However, it is essential that the driver knows the car’s limits.The driver determines the limits through specific driving maneuvers. This procedure is called *Driver Adaption* in CarMaker and can be started via "Simulation > Driver Adaption > Basic Knowledge".

In the *Basic Knowledge* tab four different adaptions are proposed:

- Vehicle Limits - This module calculates the maximum longitudinal acceleration, deceleration and lateral acceleration of the vehicle.
- Controller Dynamics - This module calculates the time preview used by the driver to predict his actions during a TestRun.
- Engine speeds for shifting - This module calculates the optimal engine speeds for shifting.
- Friction - The friction used by the dynamic area to train the driver.

Figure 3.26: Driver adaption in CarMaker

Useful Tips

1) The racing driver model however, should only be used on race circuits (closed loop). It is optimized to make the fastest lap time on a closed race track. So this is the right choice for Autocross and Endurance events.

For other competitions (e.g. Acceleration and SkidPad), the User parameterized Driver is the better choice. This driver can be tuned for the special task, which means it can be focused on pure longitudinal dynamics (for the Acceleration event) or lateral dynamics (SkidPad). The focus is set by defining a very high max. long./lat. acceleration. It can be even an unrealistically high value such as 50 m/s^2 to find the limits of the car. Check the driver settings in our FS_Example TestRuns to find the proper driver selection for each event.

2) Racing Driver has to be adapted to car and course.

3) A right-click in the Driver dialog, gives you access to predefined driver characteristics, such as a defensive, normal or aggressive driver.

With the maneuver finished we want to see the difference between different drivers. As said, the user parametrized driver is the one to chose for maneuvers and specialization. On the other hand. the racing driver is perfect for closed loop circuits. In our case we have both, a closed loop and maneuvers. But since we can tune the driver for this TestRun we choose the User parameterized Driver.

To show the difference between the driver characteristics, three TestRuns will be performed which only differ in the choice of the selected type of driver. Later in the chapter we'll learn how to check the lap time, maximum velocity and lots of other interesting data.

Exercise

- Create TestRun with normal driver:
  - Open the Driver window in the main GUI by clicking: "Parameters > Driver" or "Ctrl+d". Activate the *User parameterized Driver* in the Driver’s dialog and set the normal characteristics by right-click.
  - Close the Window and save the TestRun as "Tutorial_CM6_SlalomBrake".
- Create TestRun with aggressive driver:
  - Repeat the steps for an aggressive driver.
  - Save the TestRun to "Tutorial_CM6_SlalomBrake_Aggressive".

3.1.6 Environment

The Environment module under "Parameters > Environment", enables the definition of environmental conditions like the temperature, time of day or wind velocity for your simulation. If your model takes these parameters into account, they will influence the results of your simulation. Please read the User’s Guide ("main GUI > Help > User’s Guide") section "Parameterization:TestRun > Environment" for further information

3.1.7 Model Check

The last step before starting a simulation is to check the equilibrium state of the car using the "Model Parameter Check" (Model Check). This utility is quite helpful to see, whether the changes made have the desired effect on your car. To do so open "Simulation > Model Check". Select a field of interest and then click "Start". Please read the User’s Guide ("main GUI > Help > User’s Guide"), section "Simulation > ModelCheck" for further Information.

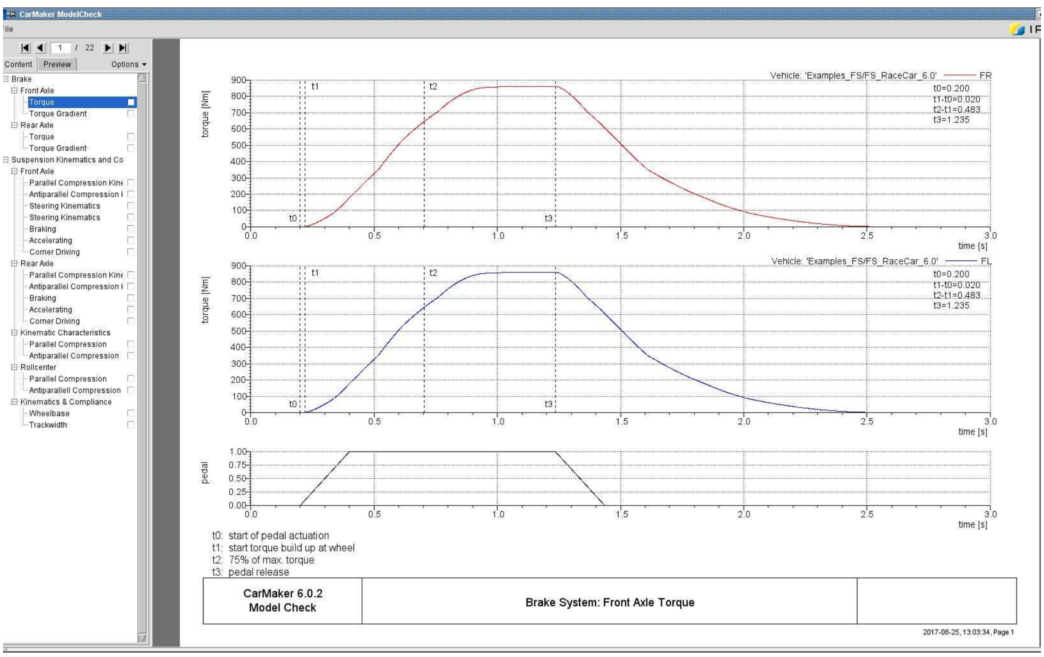


Figure 3.27: Results of simulated brake system actuation in the Model Check

3.2 CarMaker Analysis and Manipulating Methods

Introduction

Postprocessing of simulation results can be used for different reasons. For example, you may like to compare the results in a diagram, build simulation tables containing several quantities or create illustrative material for presentations. CarMaker offers the possibility to get close to all of these occasions.
Precondition for postprocessing is to dispose of simulation results. Therefore, you have to simulate TestRuns defined by the:

- parameterization of a vehicle including the tires,
- definition of a virtual road,
- selection of a driver and
- declaration of maneuvers

3.2.1 Visualization

IPGMovie

IPGMovie is a very powerful visualization tool that offers a lot of possibilities. To not expand this tutorial unnecessarily, please read IPGMovie's manual ("Main Gui > Help > IPGMovie").

Either during a simulation or afterwards, you can change the view direction (activate IPG-Movie window). Click and hold LMB, then move the pointer to the left or to the right. Do the same upwards and downwards. Click and hold the middle mouse button, then move the pointer upwards and downwards or use the mouse wheel to zoom in/out on the car.

In IPGMovie all the changes to the road are visible: the differences in friction coefficient (different colors), road signs and, of course, the route itself. If your graphic card is powerful enough you can export the movie as an .avi, .mpg or .wmv file (in IPGMovie window select "File > Export Video/Images"). However, you'll need to install a video codec such as DivX or XVid first.

Furthermore, using the "Overlay" option in IPGMovie that can be found "View > Overlay Left / Right" you can add additional displays to the IPGMovie window. In the top left/right corner you can add a route view, speedometer, gg-diagram, driver's steering wheel movements and much more.

Under "View > Quality Settings" you can change IPGMovie's quality settings. For maximum perfomance we recommend you use the "Performance" mode.

Another interesting feature of IPGMovie is the possibility to compare two cars in the same movie window. Once your primary simulation is finished you can select "Scene > Reference Vehicle > Copy current motion data" (IPGMovie menu panel). Then, activate "Primary & Reference Vehicle" in the "Scene" menu.

The goal of the next exercise is to view a primary and reference vehicle in IPGMovie. Therefore, the formerly created TestRuns with agressive and normal driver will be compared. While the second simulation is running you will see two cars in the IPGMovie window: the currently running aggressive TestRun and the former TestRun with the normal driver.

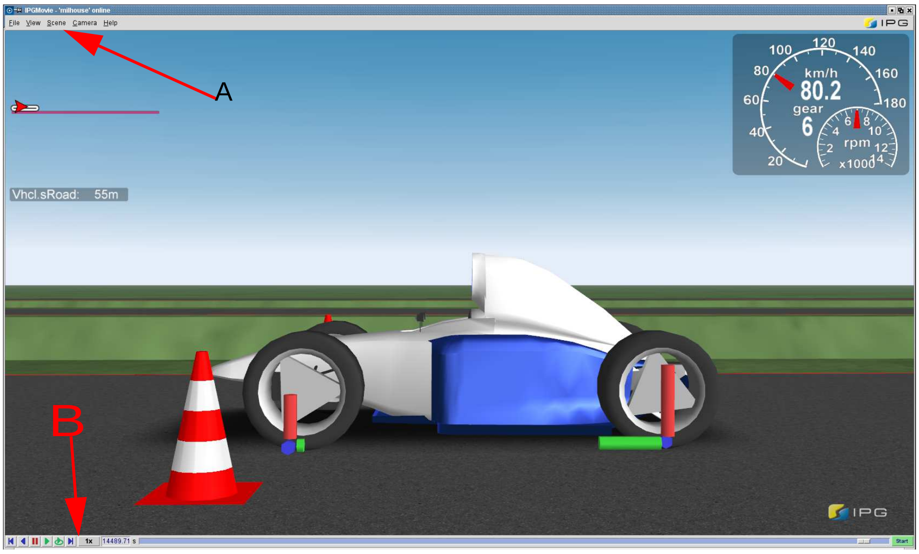


Figure 3.28: IPGMovie side view with overlay:
A: Menu panel to access settings
B: Control panel with buttons and slider

Exercise

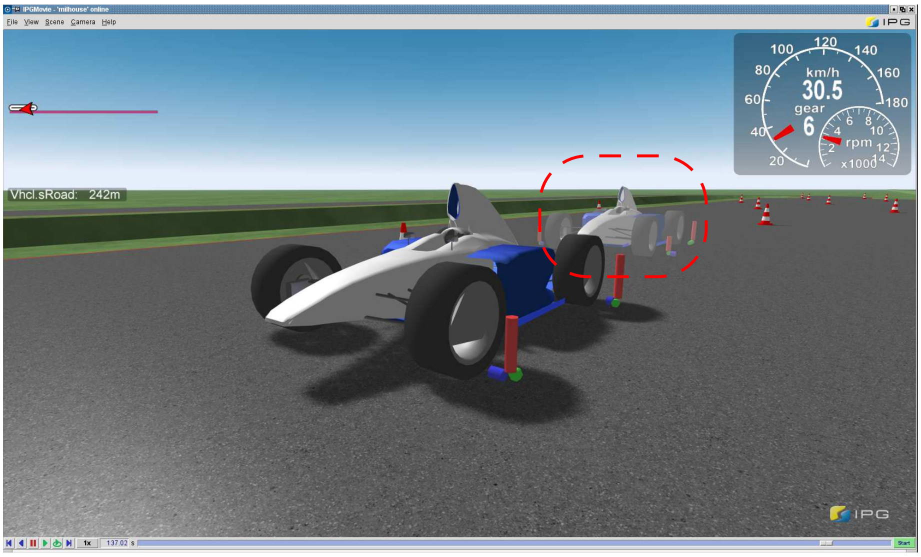


Figure 3.29: IPGMovie in "Quality" mode with reference vehicle (red frame), route view (left) and speedometer (right)

- Start TestRun with normal driver :
  - Open the first TestRun "Tutorial_CM6_SlalomBrake". Set the simulation speed to "max" and the results to "collect only".
  - Start the simulation and wait until it is complete. Save the results by hitting "Save" in *Storage of Results* (main GUI).
  - In IPGMovie ("main GUI > File > IPGMovie") select "Scene > Reference Vehicle > Copy current Motion Data" and then "Scene > Primary & Reference Vehicle".

- Start TestRun with aggressive driver:
  - Now open the second TestRun "Tutorial_CM6_SlalomBrake_Aggressive" which was prepared before and start another simulation with "collect only" settings as above.
  - Set the simulation speed to "real time" and start the simulation via the "Start" button in the main GUI.

Instruments

Another feature for observing the simulation is the "Instruments" window. You can choose between the ordinary one and a special Formula Student version. Both can be opened in the main GUI via "File > Instruments".
In both cases the "Instruments" window contains a speedometer, a rev counter, four histograms indicating the actual steering angle, clutch, brake and gas pedal position and another graph showing the active gear.

After opening the "Instruments_FS" version, at the left end of the window, sector times and lap times are recorded once you use the Racing Driver and a closed circuit.

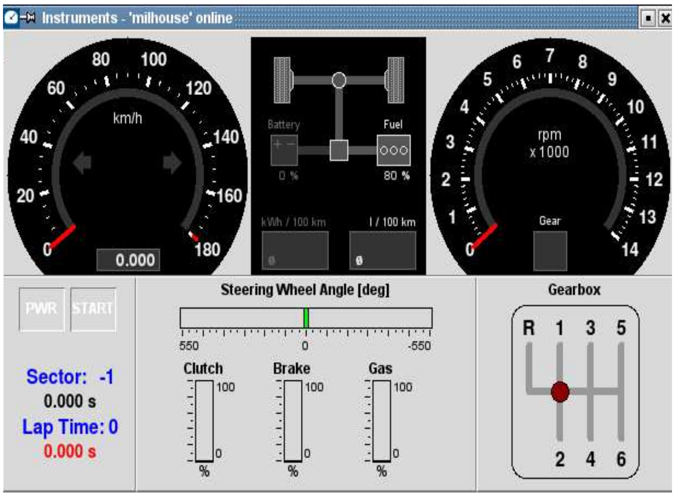


Figure 3.30: The "Instruments FS" window

Having finished both simulations and saved the results, we can now start to analyze both the TestRuns.

3.3 IPGControl

IPGControl is the CarMaker analysis tool. Various simulation result files can be loaded (Figure 3.31 - section A) consecutively to plot and compare signals. Multiple diagrams can be generated and shown in one window - one below the other (Figure 3.32).

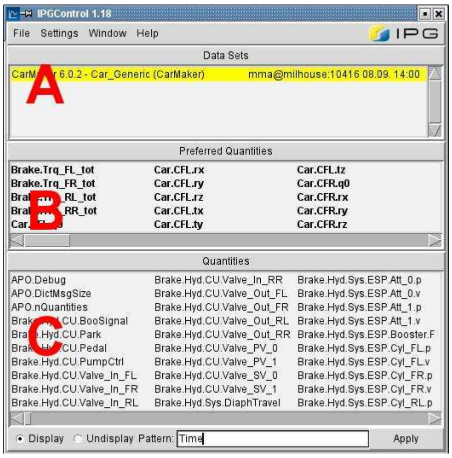


Figure 3.31: IPGControl GUI to choose quantities to be plotted in Data Window
A: List of data sets
B: List of preferred quantities
C: All available UAQ's

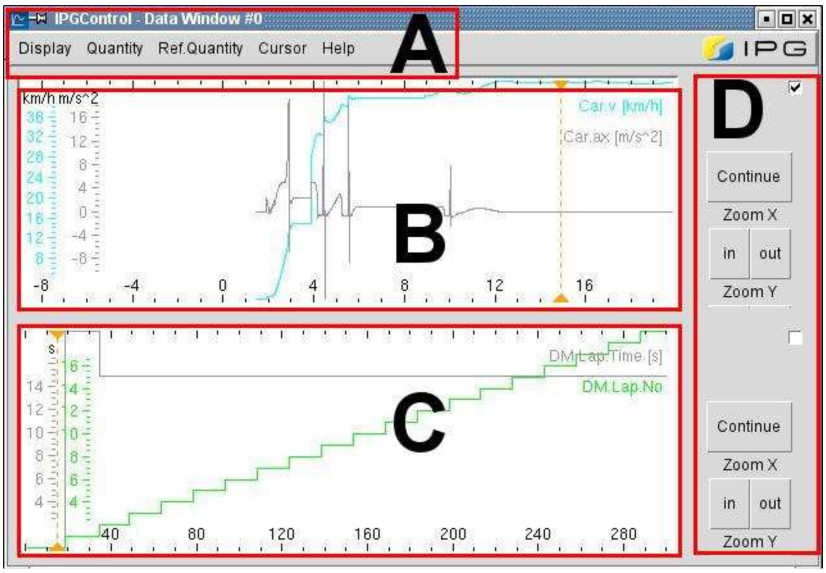


Figure 3.32: Data Window:
A: Menu panel
B: Diagram #0 (velocity and acceleration of the car is plotted)
C: Diagram #1 (lap number and time is plotted)
D: Control buttons

3.3.1 Data Access

To analyze the results we must understand how the internal data in CarMaker is acessed. During a CarMaker simulation some hundred variables are calculated that are also available for analysis. Some of them are read-only, others one can edit with tools like Direct Variable Access (DVA) or MATLAB/Simulink .

The public variables, so called *User accessible quantities* (UAQ) are gathered in logical subsystems. The naming convention is explained in detail in the CarMaker Reference Manual, in section "Introduction > Naming Convention".
The UAQ contains all public variables, both the read-only and the readable/writable ones. We already had some contact with UAQs during the maneuver (see chapter 3.1.4) definition. The readable data is plottable in IPGControl which is described in the later chapter.

To get access to the writable variable we have to open the DVA dialog. The dialog can be found in the menu "Application > Direct Variable Access". The DVA enables to influencing the simulation once it is already running.You can manipulate various quantities and assign a new value online (during the simulation). There are several ways to use the DVA mechanism:

- DVA dialog (this chapter)
- Minimaneuver commands in section "Direct Variable Access Commands" in User's Guide ("main GUI > Help > User's Guide")
- ScriptControl ("main GUI > Help > Programmer's Guide)

Using the Direct Variable Access dialog, a new value can be assigned to a quantity in a very fast and easy manner. Furthermore, the DVA dialog can be used to display the current value of the selected quantity.

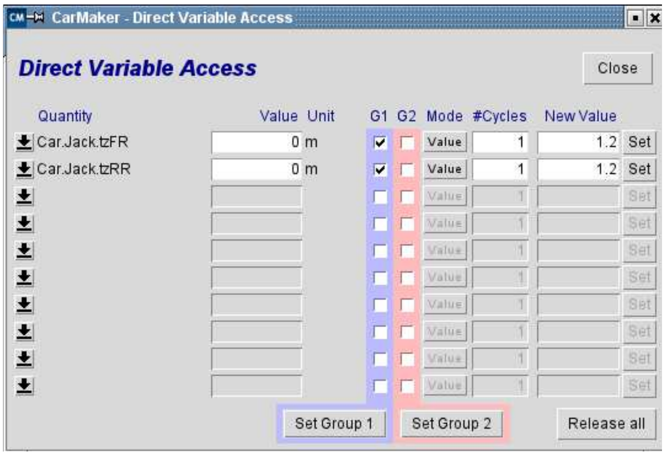


Figure 3.33: Direct Variable Access dialog

In Figure 3.33: you'll see the chosen variable with it's current and target value. You can browse trough the variables and choose the one you want to manipulate.

Quantity On the left side you can choose the quantity you wish to modify.

Value Current value of the UAQ.

Unit Reference unit for this UAQ.

G1/G2 To assign UAQ to a group.

Mode You can either choose a certain value, an offset or a factor for the UAQ .

#Cycles The number of cycles defines for how many simulation cycles the quantity should be overwritten.

NewValue An absolute value, offset or factor with which to manipulate the selected quantity.

Set Group Pressing this button, the quantity is overwritten for the duration specified in #Cycles.

Release all This button releases all quantities that were overwritten.

Some of the quantities may be clones of others e.g. the *Car.Road* and *Sensor.Road* are exactly the same values, but for accesibility and due to downwards support those quantities are still available. The *Vhcl* subsystem is a subset of *Car* and contains frequently used quantities.

To get familiar with the use of UAQ's, IPGControl and real time expressions, we will generate a new TestRun. The goal is to simulate the effectivness of rear axle steering. Since rear axle steering is best to evaluate under constant conditions, we want to create a simple left handed turn with 30m radius. As the simulation runs, we'll overwrite two UAQ's which are used to change the rotation around the z-axis of the rear carrier. The rotation is calculated in radians. A good starting point will be around -0,0523599 radian (~3 degrees) for each side. To manipulate the UAQ we'll overwrite by the DVA - dialog.

Exercise

- Create a new TestRun:
  - Save your current TestRun and open a new one. Pick FS_RaceCar_6.0 as your vehicle, choose the racing driver (Corner Cutting Coefficient = 1.0) and create a 360 degrees turn with 30m radius in the Scenario Editor (see Figure 3.34:).

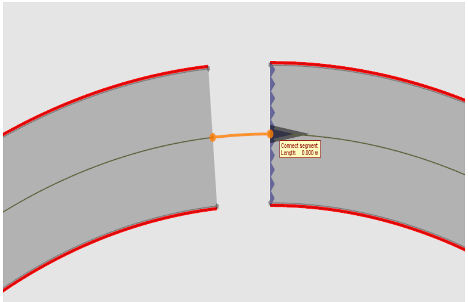


Figure 3.34: 360 Turn and connection of the segment to a closed loop

- Create a route:
  - Generate the route by using the LMB to select the desired Link. A dashed line will occur. Click the dashed line and end the definition with a double click in anywhere in the drawing area. In the - menu choose the active route (e.g. route 0) and driving side (center). Remember, when changing the driving side to generate a new path by using RMB on the orange line.

-

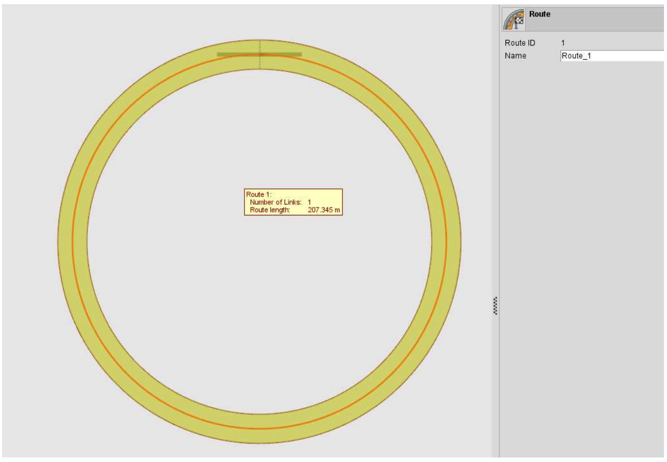


Figure 3.35: Route generation

- Define the maneuver:
  - The maneuver should contain only one step with a duration of 300s.
  - In Figure 3.36: there are two commented lines in the minimaneuver commands section. Those real time expressions (RTE) are not active (i.e. "#") right now. Before we start using RTE we should take a short look at the Direct Variable Access window.

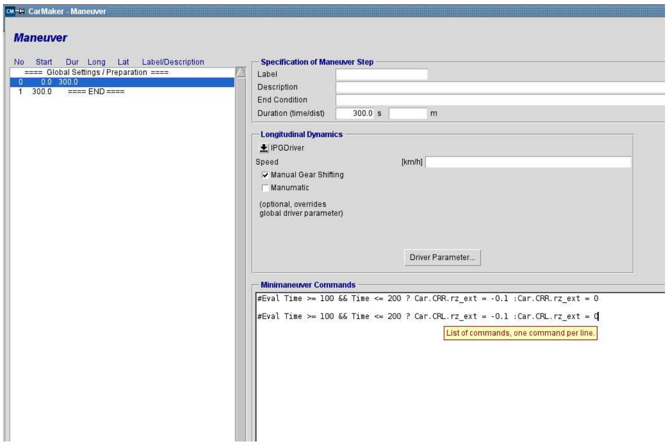


Figure 3.36: Maneuver dialog

- Choose the two UAQ's:
  - In the DVA - dialog choose "Car.CRR.rz_ext" and "Car.CRL.rz_ext" from the drop-down menu.
  - Set the UAQ to "Group 1" by marking the corresponding checkbox.

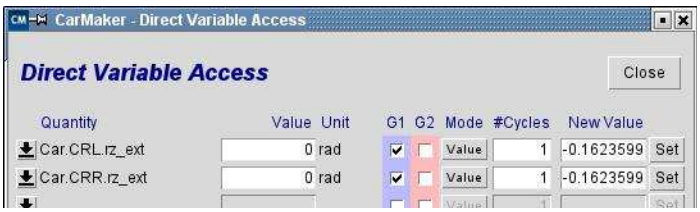


Figure 3.37: DVA dialog - manipulation tool for the writable UAQ's

- Create a second diagram in IPGControl Data Window #0:
  - Go to "Display > Add Diagram" in the Data Window menu panel. To set the white background as in Figure 3.38 go to the IPGControl GU and choose from "Settings > White Diagram Background".

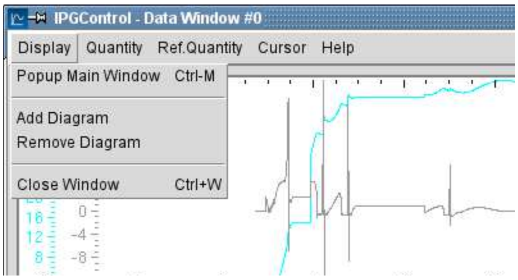


Figure 3.38: Add Diagrams to the active Data Window

- Add three UAQ to the upper diagram by using the quantity browser:
  - Cicking in the the upper diagram activates it.
  - Go to "Data Window#0 > Quantity > Browse" to open the browser dialog for UAQ.
  - Search for the following UAQ and add them to the upper diagram:
    DM.Lap.Time - The measured lap time
    Car.v - the velocity of the car
    Car.ax - Translational acceleration of vehicle connected body

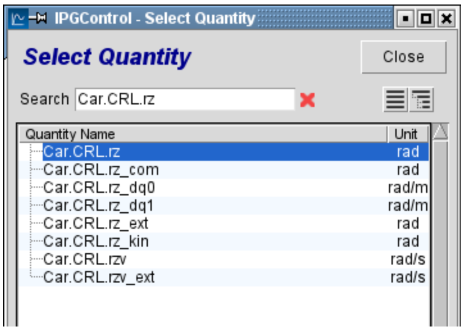


Figure 3.39: Quantity tab - here you can search for the quantities to be plotted

- Add two UAQ to the lower diagram by using the GUI:
  - Click in the lower diagram and go to the GUI. You can either search in section B/C (see Figure 3.31) and select the UAQ with LMB or you can copy-paste the name of the UAQ to the box in the bottom of the GUI and press enter.
  - Please add the following two UAQ to the lower diagram:
    Car.CRR.rz_ext - The rotation around rear right carrier z-axis
    Car.CRL.rz_ext - The rotation around rear left carrier z-axis
- Change the units to degrees:
  - As you can see the unit of the value is radians. To change it to a different unit, you have to activate the UAQ by left clicking it in the legend of Datawindow.
  - Then click RMB on the active UAQ (see Figure 3.40:). Once you've clicked the context menu opens and offers you a variety of options. Choose from "Unit of Selected Quantities > deg" for both UAQ and "fit the data totally" in the drawing field.

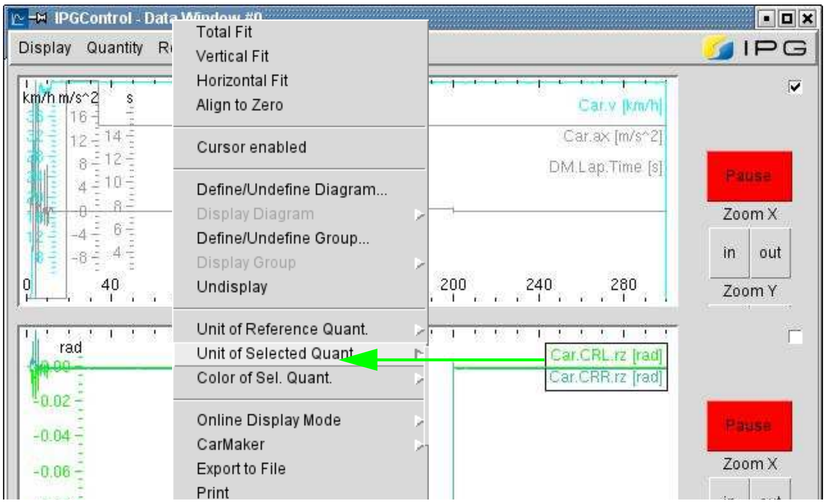


Figure 3.40: Context menu opens with after clicking RMB on plotted value

- Start the simulation:
  - Press the green "Start" button in main GUI
  - After some laps, click the "Set Group 1" - button in the DVA - dialog.
    Over the next cycles the UAQ will be manipulated until the target value is hit. In IPG-Movie and in IPGControl aswell you'll see a shift in the rear carriers. The visible shift in angle depends on the target value.

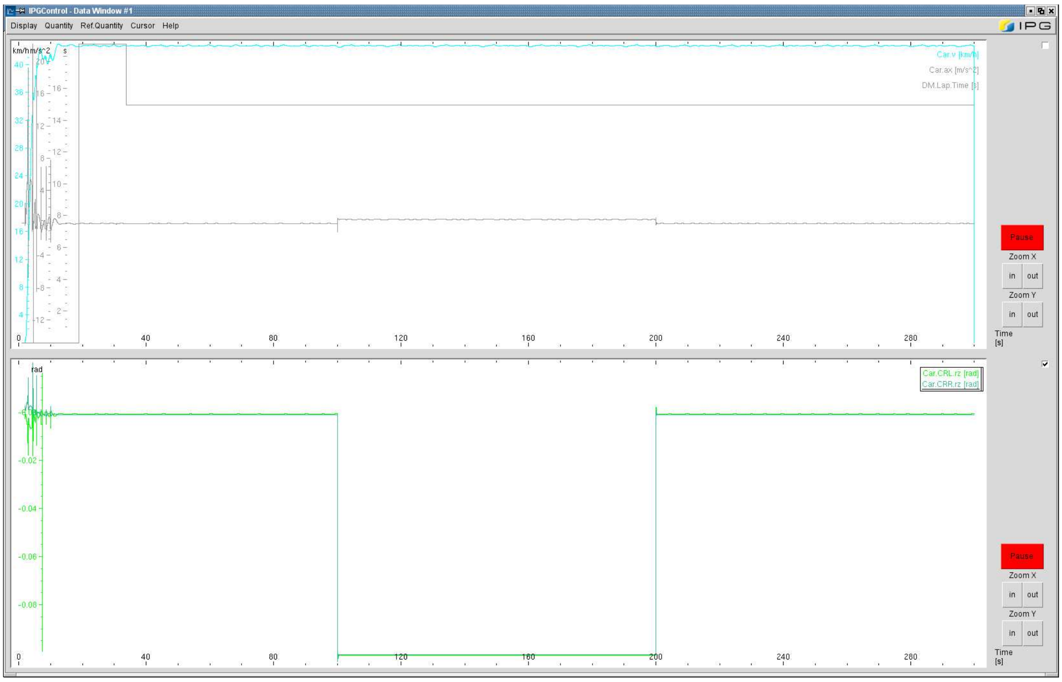


Figure 3.41: The simulation with a shift in rotation angle at 100 to 200s

Real Time Expressions

Since we want to compare the results of different steering angles we need a exact manipulation of the UAQ's. The best way to do this is real time expressions (RTE) in the minimaneuver commands. There are several ways to use RTE as embedded commands in the minimaneuver command language. In that case, the RTE is indicated by the prefix "Eval".

In Figure 3.36: we already saw the commented RTE. In the minimaneuver dialog we write:

- "Eval" to assign this line of code as an RTE.

- "?" - if (...) operator

- " : " - the column operator indicates the else segment of the logic.

Eval Time >= 100 && Time <= 200 ? Car.CRR.rz_ext = -0.1 :Car.CRR.rz_ext = 0 **(EQ 4)**

Eval Time >= 100 && Time <= 200 ? Car.CRL.rz_ext = -0.1 :Car.CRL.rz_ext = 0 **(EQ 5)**

(EQ 4) and (EQ 5) contains a simple if-else logic which is active from 100s to 200s.
The C-code snippet contains the same logic:

```
if (Time >= 100 && Time <= 200)
{Car.CRR.rz_ext = -0.1; Car.CRL.rz_ext = -0.1;}
else{ Car.CRR.rz_ext = 0 ; Car.CRL.rz_ext = 0;}
```
(EQ 6)

Please read the Appendix - section C of the User's Guide to learn more about RTE.

Exercise

- Try a variety of values:
  - Observe the changes in lap time, accleration and velocity.
  - Open Instruments to see the actuation and reaction of the driver in the moment the UAQ changes.
  - You can change the accelerations to a more aggressive behavior on your own or via a right-click in the drivers menu.
  - Save the different driver characters (normal and aggressive) as different TestRuns.
- Create a result file:
  - Load normal driver and after completing a simulation, click the "save" button in main GUI. Repeat for the aggressive driver.

Data Storage

If you save the results of your simulation, all values are not stored due to constraint of memory capacity. Only the most common ones are saved, but you can change the pre-defined settings and decide on your own, which quantities are important and which are not. The stored values can be checked and changed via "Application > Output Quantities".

To be able to compare the lap time, lateral acceleration and velocity of the car obtained by the two different drivers, please ensure that the lateral acceleration "Car.ay" and the velocity "Car.v" are activated in the "Storage of Results/OutputQuantities" dialog.

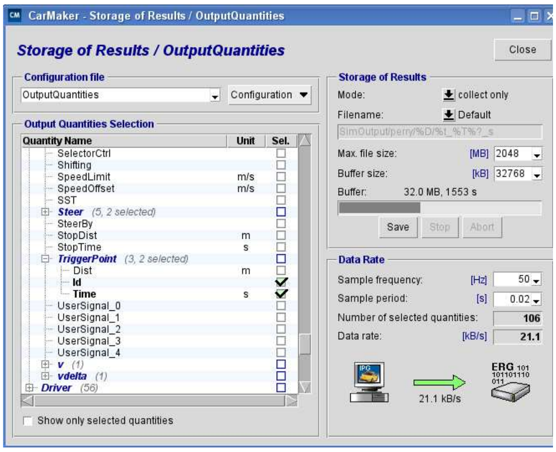


Figure 3.42: CarMaker OutputQuantities

Please also have a look at the expression "Vhcl.sRoad" for further analysis. To see the results (lap time) activate quantities "DM.Lap.Time" and "DM.Lap.No".

In this dialog you can also save or load your settings for the result storage behavior, not necessarily by the usage of one file. Keep in mind that these settings will be applied to the whole project folder.

3.3.2 Compare Results in IPGControl

To compare simulation results you have to perform several TestRun simulations and save the results of each TestRun in the *Storage of Results* box. Pay attention to the "OutputQuantities" configuration (Application > OutputQuantities), otherwise quantities you like to be saved will be missing in the result file. You can find the simulation result files of CarMaker in the following file structure of your project folder:

```
<yourprojectfolder>/SimOutput/<HostNameOfYourPC>/<Date>/NameOfYourTestRun-Time.erg
```

With IPGControl you can compare the results using different diagram windows. You can choose between two different possibilities:

- compare the last simulated TestRun with a new one
- compare the last simulated TestRun with a saved simulation results

IPGControl offers the possibility to add datasets as file or online. To compare several TestRuns without saving the performed simulation results, we add online data sets to IPGControl. Therefore, you have to run&plot a TestRun, disconnect the dataset and connect to online application. Follow the steps in the exercise below to do so.

Exercise

- Add the first Dataset:
  - Open IPGControl from the main GUI.
    Choose a TestRun, run it and plot the vehicle velocity ("Car.v") to the DataWindow #0.
- Add the second Dataset:
  - Disconnect the last performed TestRun in the main window of IPGControl (Figure 3.43).

- Vary the TestRun (e.g. change the tires of your active vehicle).
- Reconnect the simulation by the use of "File > Connect to Application". Select your application which will be listed in the data set box.
- Add "Car.v" from the second TestRun to same diagram.
- Start the simulation.

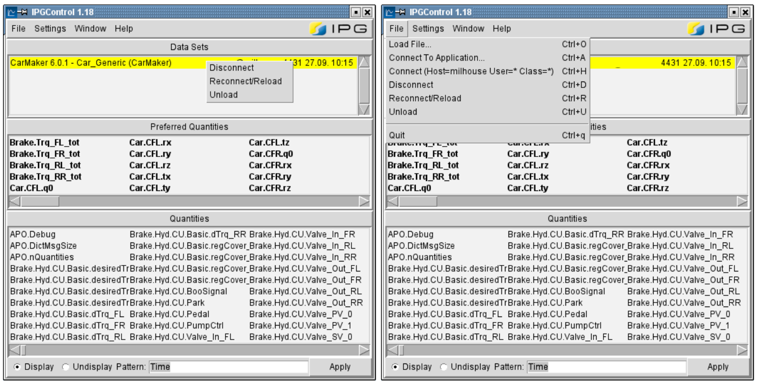


Figure 3.43: Left: Disconnect application
Right: Reconnect application

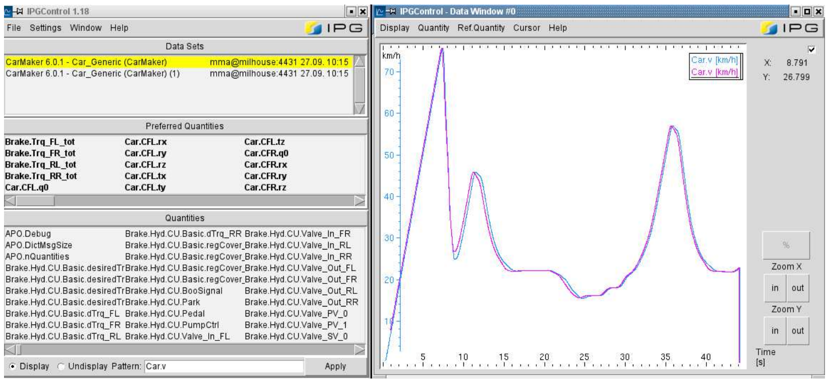


Figure 3.44: Compare the results

The second way to compare the simulation results is to save the result files first. The saved files are then loaded in IPGControl and used as same as an online application (previous example).

Exercise

- Gather Datasets:
  - Run a TestRun and save the results by "save" button in the main GUI.
    Now, vary the TestRun, start the simulation and save the dataset ("save" button in main GUI).
- Add both Datasets:
  - In IPGControl unload the last performed simulation.
    Choose "File > Load File" and load all simulation results you like to compare. These results will be listed in the data set box in IPGControl.
- Plot the Datasets:

- Select the first simulation result and add "Car.v". Do the same with the second data set and change the color by right clicking on the legend in Datawindow and choosing from "Color of selected Quantitiy".
- Feel free to add another diagrams to the Datawindow ("Display > Add diagram").

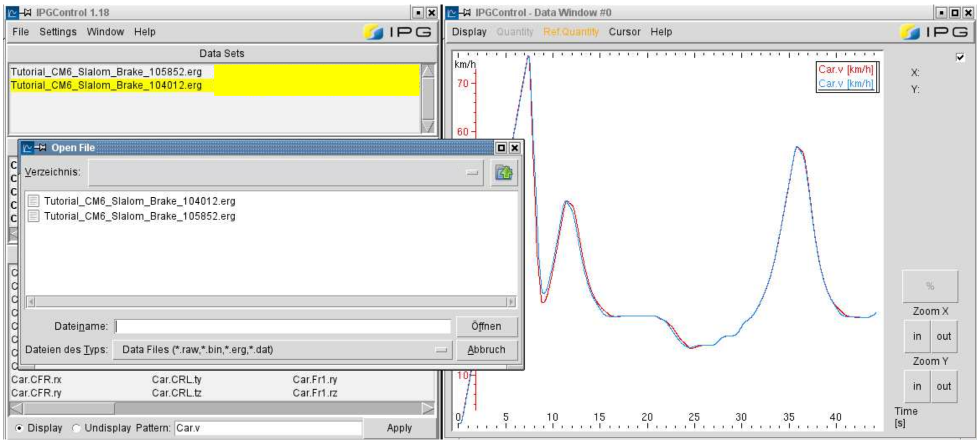


Figure 3.45: Load saved simulation results and compare them. The color of the graphs is changed to blue and red.

To get more experience with it here is another exercise:

Exercise

- Add the TestRuns to IPGControl:
  - Load the result files of the TestRuns via "File > Load File" (in IPGControl GUI) and browse to "FS_Generic_2018/SimOutput/<hostname>/<date>/<TestRunRe-sults.erg>.
- Create two Data Windows with two diagrams:
  - Open a new Data Window by "Window > New Data Window" in IPGControl GUI and create a second diagram ("Display > Add Diagram" in Data Window).
- Plot the normal driver velocity, acceleration and lap time to DataWindow#1:
  - Click in the upper diagram of . Activate the normal driver - TestRun results by clicking on it in the *Data Sets* section (IPGControl GUI) and plot "Car.v" for the car velocity and "Car.ay" in the upper diagram. Click in the lower diagram and add "DM.Lap.Time".
- Plot the aggressive driver velocity, acceleration and lap time to DataWindow#2:
- Change the colors in both windows:
  - Activate the plotted UAQ by using LMB and open the context menu by right-clicking the variables name in the diagram. Assign "Color of selected Unit > Red" for velocity and "Color of selected Unit > Blue" to acceleration.

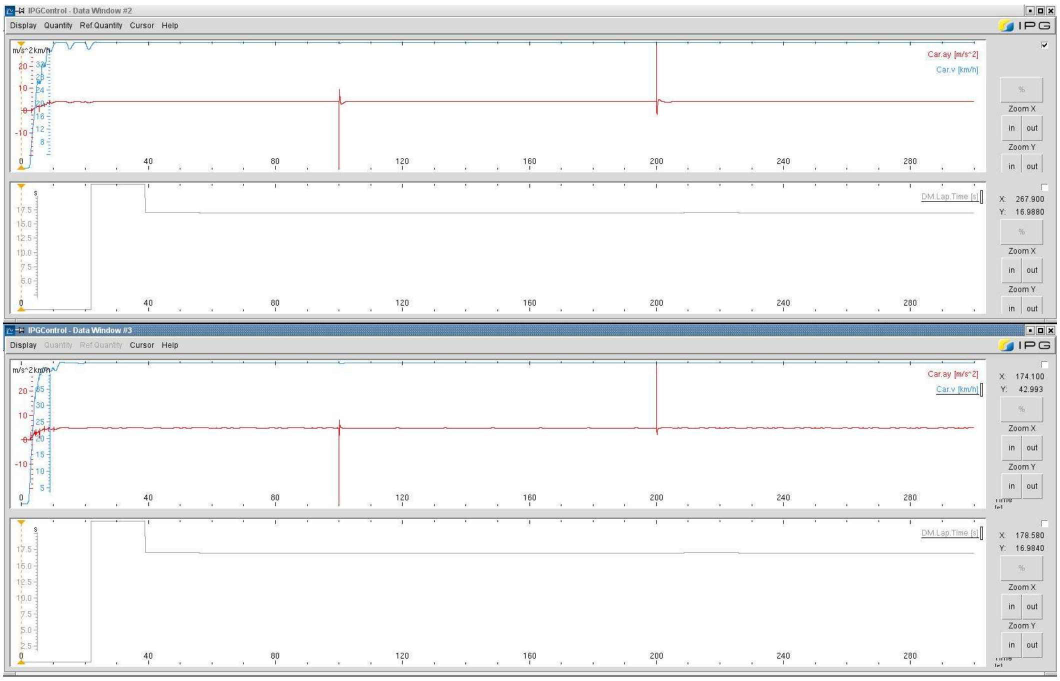


Figure 3.46: Comparison between normal driver (top) aggressive driver (bottom)

3.3.3 Exporting Simulation Results

CarMaker also offers different file formats to export the simulation results. To do so, wait until the TestRun is finished and follow the steps listed below.

- Right click anywhere in the data window and choose "Export to File".
- In the popup window select a file format, choose a file path and enter a feasible file name.
- Confirm with "Save".

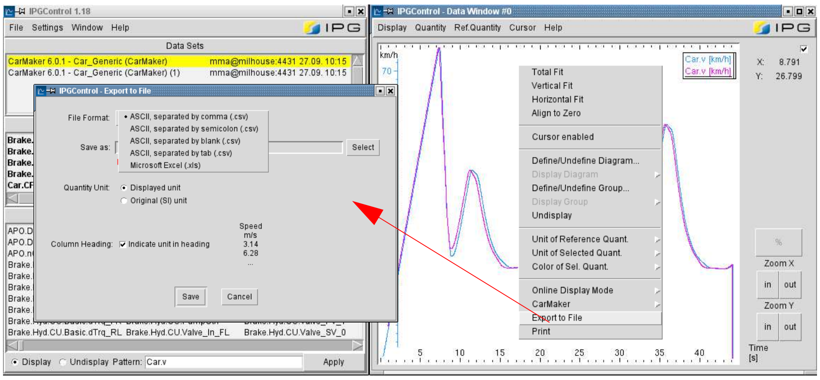


Figure 3.47: Export simulation results by right click in Datawindow

3.3.4 Print Diagrams

In IPGControl you can print plotted diagrams by using right mouse button (RMB) features. In the popping up window you can enter a diagram title and confirm with "Print". IPGControl creates a preview of the diagram which can be sent to a local printer by "File > Print".

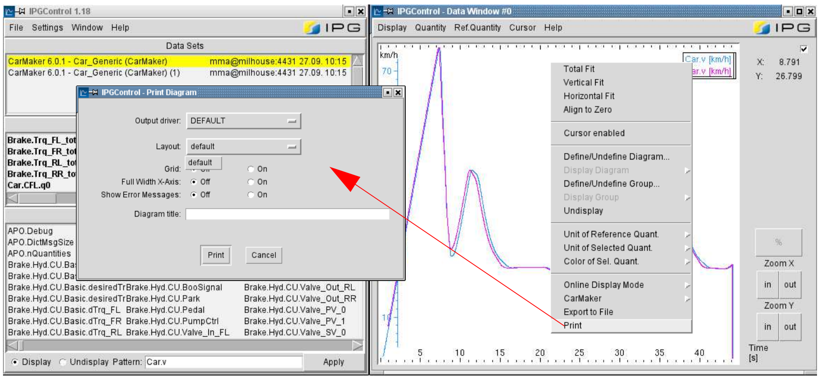


Figure 3.48: Print diagrams

3.3.5 Postprocessing with MATLAB

To work with simulation results of CarMaker in MATLAB a command is given to convert the data of the .erg files to vectors used in MATLAB:

```
variabel1=cmread('.../SimOutput/<NameOfYourPC>/<Date>/<NameOfYourTestRun-Time.erg')
```

Instead of typing the path to your result file you can simply hit "Enter" after "cmread". With this, an explorer pops up which lets you easily select the result file of interest.

The following script shows the procedure how to use the .erg files in MATLAB which should be opened in the "src_cm4sl" folder of your project folder:

```
variabel1=cmread('...\SimOutput\<NameOfYourPC>\<Date>\<NameOfYourTestRun-Time.erg')
% Plot
figure
% subplot (2,1,1)
hold on;grid;
plot(variable.Time.data, variable1.Car_v.data, 'b');
legend('Vehicle Speed');
ylabel('Time');
title('Speed vs. Time')
```

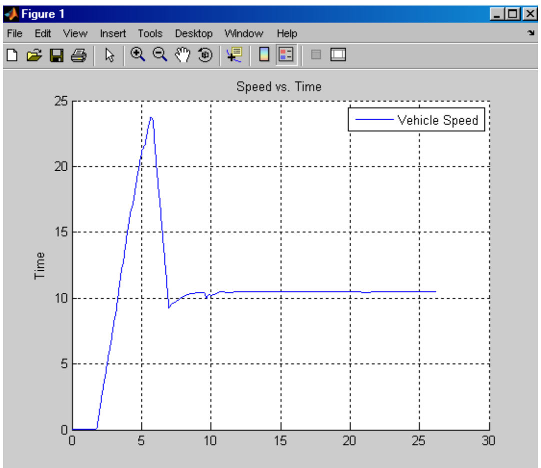


Figure 3.49: Plotting in MATLAB

If the command "cmread" will not be found by MATLAB please make sure that the script "cmenv.m" from the src_cm4sl folder of your CarMaker project directory was run successfully before. If not, you might have to adapt the path to your CarMaker installation folder in the cmenv.m script.

Chapter 4

Getting Started with IPGKinematics

IPGKinematics is a program designed to simulate a vehicle axle on an axle kinematics test bench and is used to calculate the kinematics, steering kinematics and elastokinematics of all types of suspension.
Each suspension has been modeled as multi body system built with MESA-VERDE and then converted as a library. A customized Graphical User Interface (GUI) suits each library. If desired, IPGKinematics will generate all necessary parameter files to be used by a vehicle simulation software like e.g. CarMaker.

IPGKinematics has a Reference Manual where you can find further information. In the tool bar you can click the blue question mark or via "Help > Reference Manual".

4.0.1 Opening IPGKinematics

Since IPGKinematics is a stand alone application, it can be opened using the Windows start menu. Alternatively, it can be started from the "File" menu of the CarMaker main GUI.

It is advisable to create a new project directory since an IPGKinematics simulation will generate plenty of files. Our recommendation is to use the FS_Generic_2018 folder for CarMaker TestRuns and a standalone "Kinematics" folder for IPGKinematics. The "Kinematics" folder can and should contain subfolders.

The settings of the vehicle and simulation are saved in editable *.kin files. Results are saved in different file formats and some of them are used by CarMaker (e.g. *.skc - files).

For helpful file management, we create a new directory **outside** the FS_Generic_2018 project folder called "Kinematics".

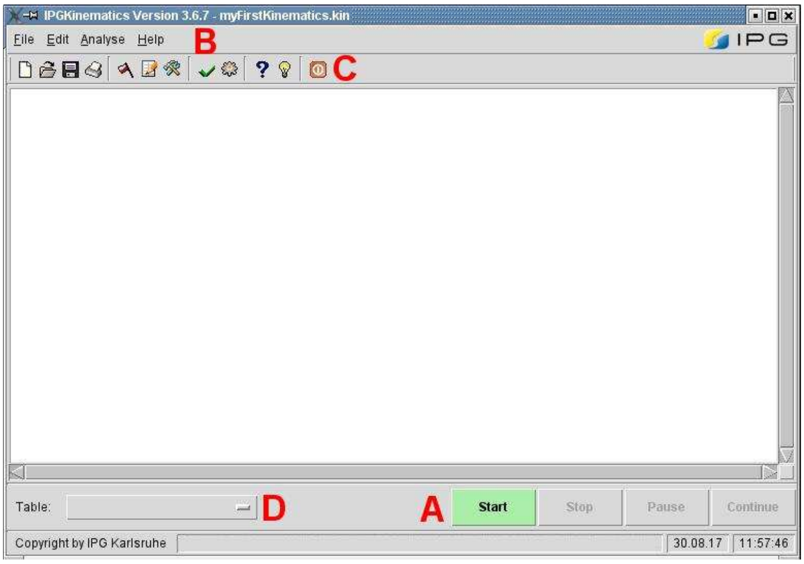


Figure 4.1: IPGKinematics GUI
A: Control buttons for computation
B: Menu bar for editing parameters, accessing help
C: Tool bar for easier access to certain dialogs
D: Drop-down table to choose the presentation mode

Exercise

- Create a new working directory for IPGKinematics:
  - Before starting IPGKinematics we want to create a folder hierarchy

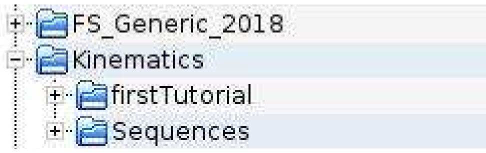


Figure 4.2: Simulation models in IPGKinematics

- Open IPGKinematics via "Main GUI > File > IPGKinematics".

4.0.2 Configuration of a Double Wishbone Axle

Before starting with a new model you have to select the type of axle you want to model. As Formula Student cars usually use a double wishbone axle this type will be chosen. The parametrization of the axle is done in the submenu *Vehicle Data*. Here all general specifications for kinematics, buffers, masses and geometric are made.

The following example aims at giving a short insight into the several submenus and analysis tool. The single parameters will be discussed more intensively in section 4.2.

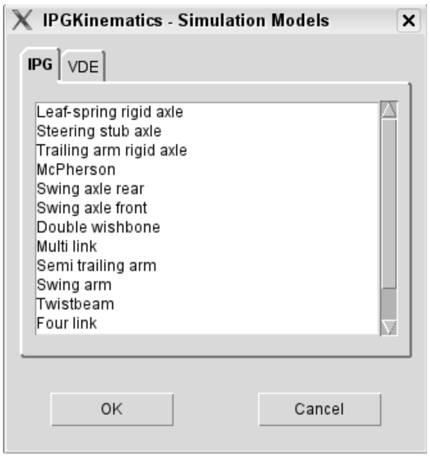


Figure 4.3: Simulation models in IPGKinematics

Exercise

- Create a new double wish bone model:
  - In the IPGKinematics GUI: "File > New Model > Double Wishbone".
- Save the model in the created folder "firstTutorial" as "doubleWishBone.kin"

4.1 Coordinate System

To improve the dynamics and performance of a car many time and cost consuming test runs are necessary. Vehicle simulation enables the user to do a bulk of these test runs in a virtual vehicle environment on a PC which saves a lot of time and money. However, this is only reasonable if the results of the simulation and those of the real car fit together as well as possible. For this reason, it is essential to achieve a very high degree of affinity between the virtual model and the vehicle itself and to this claim the parametrization of the vehicle has to be implemented carefully. After finishing a first concept the model has to be validated by means of test runs and measurements which is a complex and iterative process.

The following chapter shows an approach on how to parametrize a model efficiently. Moreover, the parameters will be explained in detail and a few examples will be given on how to determine these values. As the beginning of the development of every vehicle is the suspension, the models for front and rear axle will be built first.

In the beginning of the development process of every car there is a choice of the coordinate system. In IPGKinematics two different options for an coordinate system are available:

- Coordinate system in accordance to DIN 70000:

  The x-axis is defined in the forward direction of travel with the y-axis to the left and the z-axis upwards.

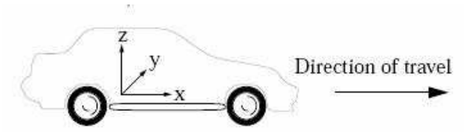


Figure 4.4: Coordinate system in accordance to DIN 70000 [KIN07]

- Coordinate system frequently used in the automobile industry:

  The x-axis is orientated opposite to the direction of travel, towards the back, with the y-axis to the right and the z-axis up.

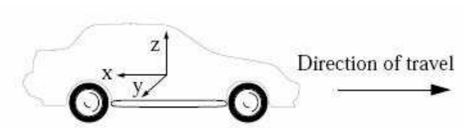


Figure 4.5: Coordinate system used in the automobile industry [KIN07]

In IPGKinematics both coordinate systems are available, but this applies only for the data input as for all calculations, the data is translated into the coordinate system according to DIN 70000. The output of the results refers to the DIN 70000 coordinate system, too. To avoid interpreting the results in the wrong coordinate system, it is recommended to work with DIN 70000 right from the start. Another advantage is that you can use this coordinate system in CarMaker later on. CarMaker is based exclusively on this axle system.

It is left to the user where he wants the origin to be placed. The only crucial thing is that all of the input data must refer to the same origin. However, you shouldn't choose the center of gravity as origin. The exact location of the center of gravity is mostly unknown and thus the results are falsified. It also makes sense to use the same origin for both tools, IPGKinematics and CarMaker, so delays needn't be considered later on. Furthermore, it is important to place the origin in the y-center of the vehicle as the geometrical input is only done for the left half of the axle and the right side is mirrored automatically by the program.

Due to the reasons mentioned above the following convention applies to both IPGKinematics and CarMaker in the course of this document:
Coordinate system: DIN 70000

- Origin in x direction: 500 mm behind the rear axle
- Origin in y direction: in the center of the vehicle
- Origin in z direction: on the road surface

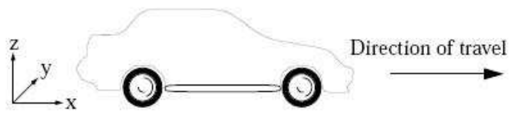


Figure 4.6: Coordinate system used in the following [KIN07]

4.2 Input Data

4.2.1 Simulation Control

This dialog controls the simulation parameters.

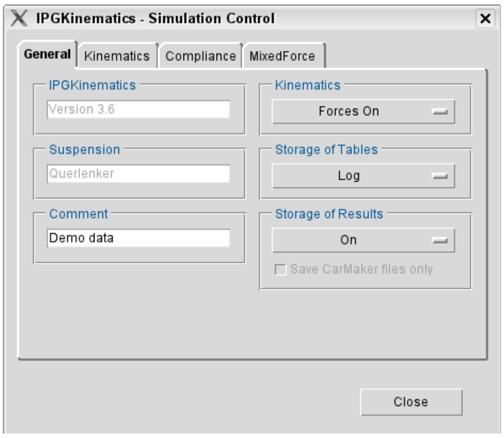


Figure 4.7: The Simulation Control window

General

Kinematics Here you should select "Forces On" from the *Kinematics* drop-down menu. "Forces Off" means that there are neither inertial, nor spring forces, purely kinematics without any effort.

Storage of Tables The control parameters for selecting the control print output for the calculated value tables are set. "Off" means there's no output of tables to file. Selecting "On", all tables are put out to file and with "Log", all tables are put out to file and on screen.

Storage of Results The result files required for the graphical output are stored when selecting "On" in the *Storage of Results* section. This is the option you should choose when defining the axle. Otherwise, for each simulation the whole compliance model for the use in CarMaker is calculated additionally, which increases the simulation duration needlessly. To export the results to CarMaker you must choose one of the "CarMaker Interface" options. Whereas

- "CarMaker Interface (1)" provides linear compliance,
- "CarMaker Interface (2)" uses a non-linear compliance model.

However, the second option increases the simulation duration. At the same time, the results don't improve much, as Formula Student cars usually don't use buffers. So, you should prefer the option "CarMaker Interface (1)".

Exercise

- Edit the simulation settings:
  - Open the Simulation Control window in the menu "Edit" in the main GUI or by clicking on the corresponding button.
  - *Kinematics* = "Forces On".
  - *Storage of Tables* = "Log".
  - *Storage of Results* = "CarMaker Interface (1)".

Kinematics

This tab defines the protocol according to which the movement of the wheels is calculated.

Parallel Kinematics This means that both wheels of an axle will move in the same direction on compression if activated (On).

Reciprocal Kinematics This stands for wheels that travel in opposite directions. To get to know the meaning of the different choices, please have a look at the Reference Manual, chapter "Input Data".

Steering Kinematics is only necessary for the front axle for front axle steering. It simulates rack-and-pinion steering which steers the front wheels correspondingly.
Choose "On (2)", as "On (1)" does not consider the interaction of steering and reciprocal wheels travel. "Steering Forces" is used in certain situations such as a tire hitting a pavement.

Compression and Distance Steering Rack The maximal values for Compression and Distance Steering Rack should be kept small, not to increase the simulation duration needlessly. That means, they should lie only insignificantly above the values possible in your car.

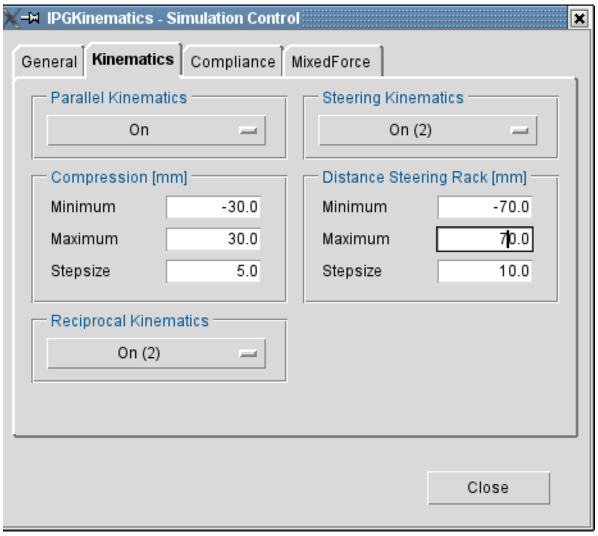


Figure 4.8: The Simulation Control Kinematics tab

Exercise

- In the *Kinematics* tab set:
  - *Parallel Kinematics* = On.
  - *Steering Kinematics* = On (2).
  - *Reciprocal Kinematics* = On (2).
  - *Compression* = +/- 30 mm; *Stepsize* = 5.0.
  - *Distance Steering Rack* = default values.

Compliance and MixedForce

The last two tabs are used to apply external forces to the wheels in longitudinal and lateral direction. In case you are interested in the stresses of the single suspension components for FEM analysis, these options can be activated. For pure kinematics, external forces are deactivated, which is why we leave this option turned off. Further information can be found in the IPGKinematics Reference Manual ("Help > Reference Manual), chapter "Input Data" .

Exercise

- In both, *Compliance* and *MixedForce* tab, turn everything "Off"
- Save the model in the folder created earlier. Beware that you choose "Input Data (.kin)" in the filter selection area.

4.2.2 Geometrical Control

Before starting the simulation you should check your inputs. For geometrical data there is *Geometrical Control* which gives an overview of all the positions of the points entered. You find it in the menu "Analyse > Geometrical Control" or use the corresponding icon.

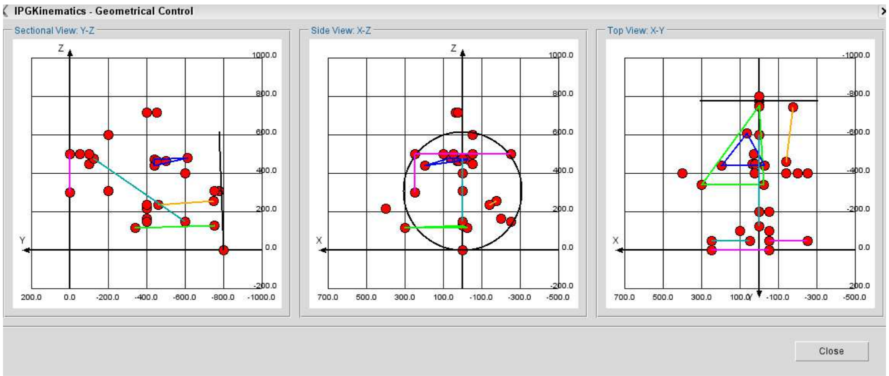


Figure 4.9: The Geometrical Control Window

We will define the geometrical settings in the following chapter. Nonetheless as a trial you can check the pre-defined geometry. Follow the path described above and when everything is ready push the "Start" button. According to the "Simulation Control" parameters you can see several arrays of calculations being performed. The calculation process can take quite a while, depending on your input data. Once the simulation is finished (the "Start" button turns green) you should save the results straight away. In the later chapter "Output Data" you can take a look at how to analyze the data.

Exercise

- Perform and save the simulation:
  - Start the simulation
  - Hit "File > Save" and overwrite the existing file to save your model with the output data of the simulation. Saving generates numerous files depending on the parameters set in Simulation Control window.

4.2.3 Vehicle Data

This subchapter describes the specific vehicle input parameters that are available for processing in the "Edit > Vehicle Data" menu. In the various tabs you have to set general terms of the car, define spring rates, bushings and the masses of several components and their positions.

General

The *General* Tab contains some general settings.

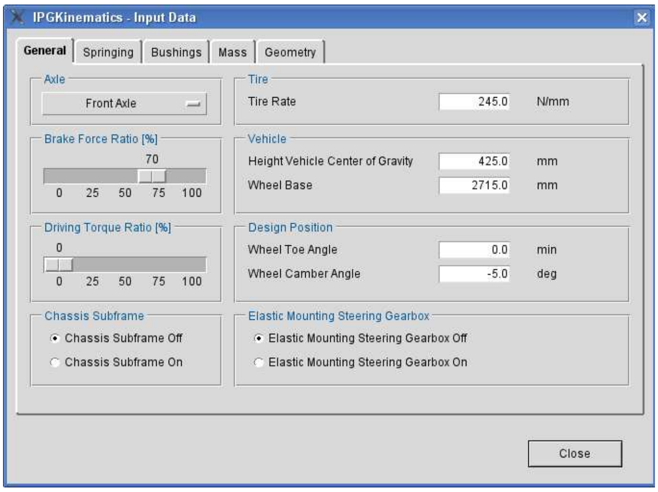


Figure 4.10: The Input Data General tab

Axle The options here are front and rear axle, one of those needs to be steered, whereas the pitching pole position is defined

Note: If you choose “Rear Axle” no steering kinematics will be calculated even if this was set in the simulation control.

Brake Force Ratio In this field the brake force ratio of the considered axle is defined. 0% means that the axle you’re modeling receives no brake power. This information is required for calculations of the braking torque compensation angle and anti-dive.

Driving Torque Ratio Here the driving torque ratio of the axle is defined. 0% means that the axle is not powered and 100% means that only this axle is powered. It is necessary to enable anti-squat and starting torque compensation angle calculations.

Model configuration (1) In the case of a Formula Student racing car you should tick "Chassis Subframe off" since there is only a chassis without any subframe supporting the engine or lower suspension components.

Tire This point is to define the vertical stiffness of the tire in [N/mm]. As the tire is modeled as a linear spring, a tire spring coefficient must be entered. This spring coefficient is a function of:

- tire pressure
- tire design
- tire tread
- tires height to width ratio
- driving speed
- rolling characteristics of the tire

The spring coefficient can be determined by a simple static test. All you need is a spring scale and two plain plates. The tire is clamped between the two plates and put on the spring balance. Then you apply a defined force F_R. Through measuring the tire deflection Δs the spring coefficient can be determined as follows:

$$c_1 = \frac{F_R}{\Delta s} \quad \textbf{(EQ 7)}$$

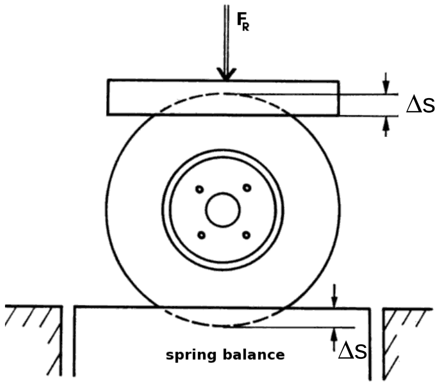


Figure 4.11: Measuring the tire deflection [Rei82]

If you are a member of the Tire Test Consortium you are in charge of detailed spring rate data of your tires. For this information, see the FromCalspan.zip archive from your RoundX tire testing directory in the official TTC forum. The archive contains a spreadsheet titled SummaryTables.xls. There you will find measured data of cornering stiffness as well as the vertical stiffness as a function of the inclination angle, tire pressure and vertical load.

If you don't have reasonable values available there is another possibility to get an approximation. For this you need the tire's diameter D, static radius r_{stat}, load in accordance with the actual tire pressure F_R and a correction factor k_B for the tire design, tread and velocity. If these values are available, a good approximation for the spring coefficient can be found according to

$$c_1 = k_B \cdot \frac{F_R}{D/2 - r_{stat}} \quad \textbf{(EQ 8)}$$

Vehicle

In this area the height of the vehicle's center of gravity and its wheelbase must be entered. The wheelbase is needed to calculate the turning track diameter and turning circle. The height of the center of gravity is crucial to compute the lever of the roll axis. To determine the center of gravity for your car, please take a look at section "Vehicle Body", pg. 77.

Design position

Here, the constructive wheel toe angle and wheel camber angle have to be defined. The design position is the position of the car with only the axle load on it. This accords to a wheel travel of 0 mm. Thus, it must be kept in mind if the weight of the driver was included in the design position or not. If not, attention must be paid to various settings such as location of the center of gravity, camber and toe angle etc. All of them have to be measured or calculated without the driver's weight.

Model configuration (2)

This offers the possibility to support the rack-and-pinion steering elastically. Regarding FS racing cars, this isn't commonly used.

Exercise

- Choose a rear axle configuration and set:
  - "Tire Rate" = 200 N/mm
  - "Height Vehicle Center of Gravity" = 300 mm
  - "Wheelbase" = 1600 mm
  - "Wheel Toe Angle" = 0 min
  - "Wheel Camber Angle" = -3 deg
  - "Chassis Subframe" = off
  - "Driving Torque Ratio" = 100%
  - "Brake Force Ratio" = 30%
  -

Table 4.1:

Variable	Value
Tire Rate	200 N/mm
Height Vehicle Center of Gravity	300 mm
Wheelbase	1600 mm
Wheel Toe Angle	0 min
Wheel Camber Angle	-3 deg
Chassis Subframe	off
Driving Torque Ratio	100 %
Brake Force Ratio	30 %

Springing

Here are the parameters for axle spring and stabilizer bar, as well as the parameters required for wheel suspensions featuring pull rod/push rod springing.

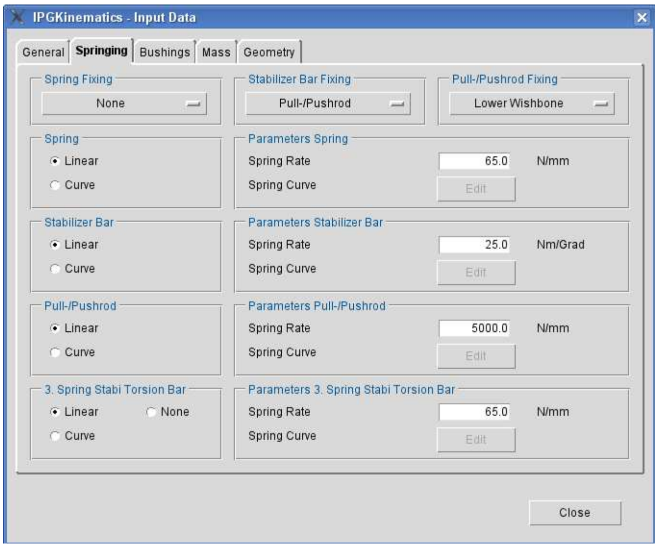


Figure 4.12: The Springing tab

Spring Fixing

It defines which part the spring is fastened to. It is assumed that the "Spring Fixing" is the one on the suspension side. The option "None" is chosen in case of a pull-/pushrod.

Stabilizer Fixing

In IPGKinematics you have the possibility to mount the stabilizer fixing to the lower wishbone, wheel carrier or the pull-/pushrod. If you have anti-roll-bars which are attached to the rocker, you use the option "pull-/pushrod". By using this option, IPGKinematics will consider a T-Stabilizer, otherwise a U-Stabilizer is used.

To properly parameterize the stabilizer in IPGKinematics you can use the option "pull-/pushrod", if you do not have a T-Stabilizer in your car, calculate the stiffness of the T-Stabilizer that matches the configuration.

The second possibility is to convert any stabilizer into a U-bar. You should use wheel carrier as stabilizer fixing due to the motion ratio from wheel travel to connection point travel of 1. This method should work well if in reality you are using a constant motion ratio, too.

There is a variety of possibilities to design a stabilizer template, here are a few suggestions:

1. Choose the wheel center as connection point for the stabilizer link to the wheel.

2. The connection point of the stabilizer link and U-bar should be placed 100mm vertically to the former point.

3. The lever arm of the U-bar should be located 200mm horizontally in x-direction to stabilizer mounting point on the chassis.

This stabilizer parametrization will work properly in IPGKinematics as long as you use an adapted stiffness, too, so that the whole substituted component has the same properties as a real one.

Parameter Stabilizer

To parameterize the stabilizer template you need to calculate the total stiffness c_S of the real stabilizer. In case of a U-bar this dependents on the linearized torsional stiffness of the bar c_{Ba} and the linearized bending stiffness c_{Bl} of the lever arms also called blades. This linear total stiffness is transferred into the rolling stiffness of the front-/rear axle $c_{Ro,S}$. Recalculating with the motion ratio and track of the axle, you get the torsional stiffness of the stabilizer template $c_{S,\theta}$, which is to be entered in IPGKinematics.

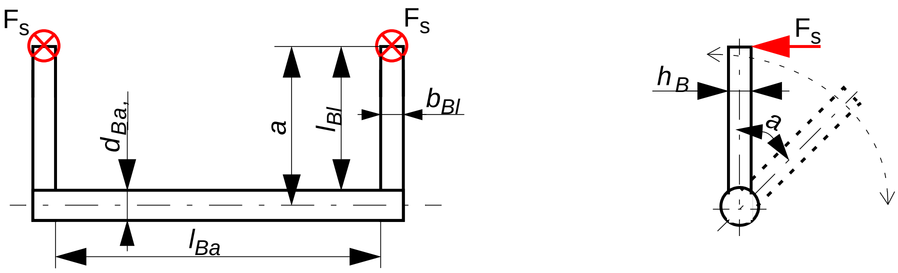


Figure 4.13: Geometrical notations for the calculation of the substituted stabilizer stiffness

In order to calculate the linearized torsional stiffness of the bar c_{Ba} you need to calculate the torsional rigidity of the cross section $I_{p,Ba}$, with $d_{Ba,o}$ as outer diameter and $d_{Ba,i}$ as inner diameter.

$$I_{p,\,Ba} = \frac{\pi \cdot (d_{Ba,o}^4 - d_{Ba,i}^4)}{32} \qquad \text{(EQ 9)}$$

With the stabilizer bar's shear modulus G, the torsional rigidity of the cross section $I_{p,Ba}$, the length of the lever arm a (with forces acting at a), and the length of the bar l_{Ba}, you get c_{Ba}.

$$c_{Ba} = \frac{G \cdot I_{p,\,Ba}}{a^2 \cdot l_{Ba}} \qquad \text{(EQ 10)}$$

In order to calculate the blades' bending stiffness you need to calculate the cross sections' bending rigidity. Many teams use tunable blades which means that the cross section on which the forces are acting changes. To take that into account you have to calculate the rigidity of the blade in 0deg and in 90deg position. Using (EQ 13) you get the combined cross sections rigidity for any bending angle α_{Bl}.

$$I_{Bl,0°} = \frac{b_{Bl} \cdot h_{Bl}^3}{12} \qquad \text{(EQ 11)}$$

$$l_{Bl,90°} = \frac{b_{Bl}^3 \cdot h_{Bl}}{12} \quad \text{(EQ 12)}$$

$$l_{Bl} = l_{Bl,0°} - \frac{\alpha_{Bl} \cdot l_{Bl,0°}}{90°} + \frac{\alpha_{Bl} \cdot l_{Bl,90°}}{90°} \quad \text{(EQ 13)}$$

In order to calculate the bending stiffness of the blade, you need the E modulus of the blade material and the springing length of the blade l_{Bl}.

$$c_{Bl} = \frac{3 \cdot E \cdot I_{Bl}}{2 \cdot l^3_{Bl}} \quad \text{(EQ 14)}$$

By adding the linearized torsional stiffness of the bar and the stiffness of the blade you get the total stiffness of the real stabilizer c_s for a force acting at the end of one blade.

$$c_s = \frac{c_{Ba} \cdot c_{Bl}}{c_{Ba} + c_{Bl}} \quad \text{(EQ 15)}$$

To transfer the overall stiffness of the real stabilizer into overall stiffness of the template U-bar the roll stiffness of the axle, the stabilizer is associated with, has to be calculated. s/s_f is the motion ratio of wheel travel s and stabilizer blade-end travel s_f. b is the track width of the axle.

$$c_{Ro,s} = \frac{c_s}{(s/s_f)^2} \cdot b^2 \cdot \frac{\pi}{180°} \quad \text{(EQ 16)}$$

Using (EQ 17) it is possible to calculate the linearized stiffness of the template U-bar $c_{s,x}$. This value is needed in the parametrization of the stabilizer in CarMaker. Right now it is only needed to calculate the torsional stiffness in (EQ 18) with a' =200 mm as the blade length of the template U-bar.

The value of $c_{s,\theta}$ has to be entered as the torsional spring rate in IPGKinematics among "Parameters Stabilizer Bar".

$$c_{s,x} = \frac{c_{Ro,s}}{2 \cdot b^2} \cdot \frac{180°}{\pi} \quad \text{(EQ 17)}$$

$$c_{s,\Theta} = c_{s,x} \cdot a'^2 \cdot \frac{\pi}{180°} \quad \text{(EQ 18)}$$

Spring/ Parameters Spring

There are two possibilities to parametrize the spring: linear and non-linear characteristics. For a linear progression the only needed value is the spring rate. It can be determined by a simple tension test. With this test, the spring rate results from the quotient of applied force ΔF and deflection Δs:

$$c = \frac{\Delta F}{\Delta s} = \frac{m_1 \cdot g}{\Delta s} \quad \text{(EQ 19)}$$

A non-linear spring rate should only be selected if the spring rate is truly non-linear!

Parameter Pull-/ Pushrod

With this parameter (also called spring rate) the elastic deflection of the pull-/pushrod is taken into account. Therefore the pull-/pushrod is seen as a virtual spring. Accordingly the unit is N/mm, like in section "Parametrization Spring". By knowing the material and the geometry of the pull-/pushrod you can assume that the stresses lie within the elastic region and thus apply Hooke's law. Requested is the force triggering a deflection of 1mm of the pull-/pushrod:

$$F = \frac{\sigma}{A} = \frac{\sigma}{\pi \cdot r^2} = \frac{E \cdot \varepsilon}{\pi \cdot r^2} = \frac{E \cdot \Delta l}{(\pi \cdot r^2) \cdot l_0} = \frac{E \cdot 1mm}{(\pi \cdot r^2) \cdot l_0} \qquad \text{(EQ 20)}$$

For a radius of r = 20 mm, a length of l = 430 mm and an E modulus of E = 70 GPa (Aluminum 7075) a force of F = 55 kN is required. Hence, this pull-/pushrod would have a spring rate of 55 kN/mm.

However, in a Formula Student racing car such high forces will never be applied on a pull-/pushrod. You can assume that a pull-/pushrod won't be deformed elastically in a Formula Student car. To achieve this with IPGKinematics you needn't put such high values for the spring rate. At the same time, this would increase the simulation duration needlessly. Using an exact spring characteristic as explained in case of the axle spring, better results won't be achieved. Moreover there is the danger of running the pull-/pushrod during the tension test through a ductile deflection.

Parameter 3. Spring Stabi Torsion Bar

With this option you can add a 3rd spring to your axle, which is actuated due to parallel bump of the left and right wheel.

Exercise

- Set the fixings as follows:

Table 4.2:

Variable	Value
Spring Fixing	None
Stabilizer Bar Fixing	Pull-/Pushrod
Pull-/Pushrod Fixing	Lower wishbone
3rd Spring Stabi Torsion Bar	None
Spring Rate	23.9 N/mm
Torsional Spring Rate	40 Nm/deg
Parameters Pull-/Pushrod	5000 N/mm

Bushings

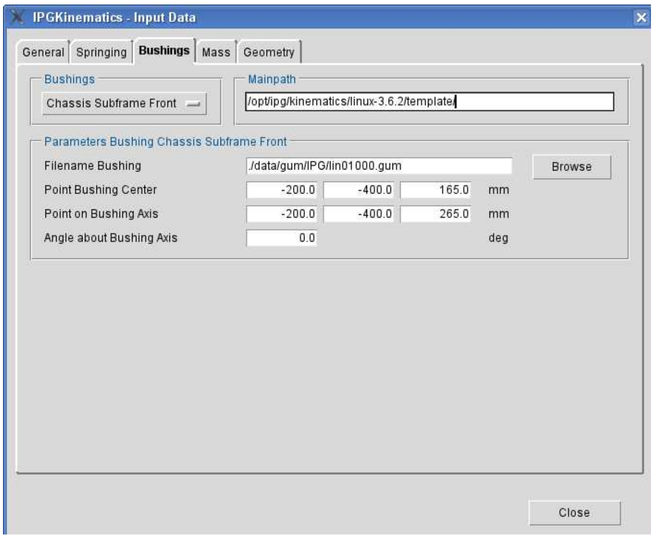


Figure 4.14: The Bushings tab

In Formula Student cars travelling comfort is not of high priority and elastic bushings would soften the suspension, therefore they are not used.

Bushings This drop-down menu lets you choose from the bushing according to the selected suspension model. When the "Bushings" menu button is pressed, you can switch between the individual bushings of the wheel suspension. The contents of the tab are then updated accordingly and the parameters for the selected bushing are displayed. In case of double wishbone model there should be 9 different bushing models.

Mainpath Specification of the main path for bushing files. The purpose of the main path is to eliminate the need to specify the complete path and the file name for each bushing. The path must end with "/". The main path and the file name are then used by the program to construct the path to the input files for the bushings.

Parameters Bushing .. Control parameters for selecting the bushing parameters. All parameters for a bushing are entered in this field.

To simulate the inexistent bushings in IPGKinematics they are defined as extremely rigid ($E = 5000\ N/mm^2 = 5\ GPa$). This stiffens the bushings in all direction. Therefore the bushings can always be orientated parallel to the stationary coordinate system. That means, you do not have to enter any value for "Angle at bushing axle".

Further information can be found in IPGKinematics Reference Manual.

Exercise

- Set the path for each Bushing:
  - Since the suspension geometry should be defined first we have to make sure the path, set in *Mainpath*, is set to the corresponding directory on your workstation (e.g. C:\IPG\kinematics\version\template - This is not your project folder "Kinematics").

Mass

Note that all masses are in kilograms and the decimal point must not be a comma!

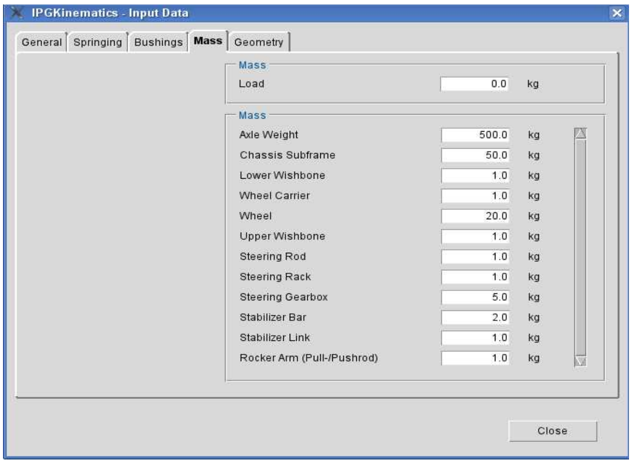


Figure 4.15: The Mass tab

Load The mass referred as "Load" is only used if you want to add a specific load that does not belong to the vehicle itself. It is usually set to 0, as you do not wish to simulate additional load in racing cars!

Axle Weight This is the load on the front/rear axle minus the single masses listed below in Figure 4.15: .

Chassis Subframe The chassis subframe mass can be set to any value, except 0 (we have chosen the option "Chassis Subframe Off" in the "General" tab so the mass is not considered in the calculation). However, setting the mass to 0 will generate a failure.

Other Masses These depend on the chosen type of axle. More information can be found in IPGKinematics Reference Manual ("Help > Reference Manual"), chapter "Vehicle Data".

The GUI does not differ between front axle and rear axle. Please leave the default values for parts which do not occur. In case of entering 0 arithmetic errors could occur.

Exercise

- Set the following values for the masses:

Table 4.3:

Part	Mass in kg
Load	0
Axle Weight	140
Chassis Subframe	10
Lower Wishbone	0.5
Wheel Carrier	0.45
Wheel	10.5
Upper Wishbone	0.4
Steering Rod	0.25
Steering Rack	0.2

Table 4.3:

Part	Mass in kg
Stabilizer Bar	0.4
Stabilizer Link	0.08
Rocker Arm	0.08

Geometry

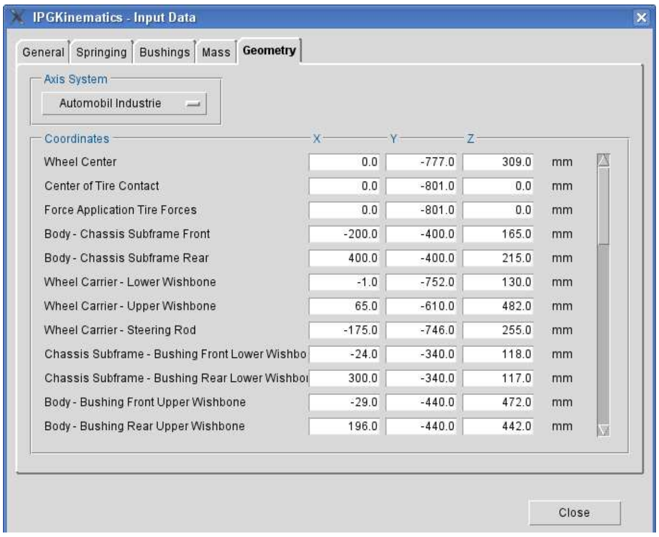


Figure 4.16: The Geometry tab

Axis System As explained in section "Coordinate System", pg. 54, the coordinate system in accordance to DIN 70000 is used.

Coordinates The coordinates defined in this section are the most important of all. The calculations of the kinematics are based on these values. For this reason the values of these points should be done carefully in order to generate a model as accurate as possible.

In the table you only define those points for the left half of the axle. IPGKinematics automatically mirrors the values for the other half so as to model the complete axle.

Here, only some of the crucial points are explained. You will find a detailed description of all entries in the IPGKinematics Reference Manual, chapter 4.3.

Table 4.4:

Coordinates	Description
Force Application Tire Forces	Usually the same values as "Center of Tire Contact"
Body - Chassis Subframe Front/ Rear	As mentioned before Formula Student cars don’t usually use subframes. To give feasible values to the program enter the mounting points of the lower wishbones on the chassis.
Subframe - Bushing Front/Rear Lower Wishbone	As the subframe is deactivated it corresponds with the body mounting points of the lower wishbones

Table 4.4:

Coordinates	Description
Spring/Damper Body	Although these points are not used in case of a pull-/pushrod actuated suspension feasible values are needed as IPGKinematics will generate an error if you leave them out. Thus, a good value to choose is the same as "Wheel Carrier - Lower Wishbone" + 500 mm in height
Spring/Damper Wheel suspension	Although these points are not used in case of a pull-/pushrod actuated suspension feasible values are needed as IPGKinematics will generate an error if you leave them out. Thus, a good value to choose is the same as "Wheel Carrier - Lower Wishbone"
Axle Drive Shaft - Differential/Wheel	Even if you simulate a non-driven shaft, plausible values are necessary. A good value to choose is the same as "Wheel Center" and the middle of the axle
Rotation Axis - Rocker Arm	Can be any point on the rotation axis of the rocker arm
Stabilizer Link - Wheel Carrier	Equal to point Wheel Center
Stabilizer Bar - Stabilizer Link	Add 100 mm in height to the former point
Stabilizer Bar - Chassis Subframe	Add 200 mm in x direction to the former point.
Stabilizer Bar - Chassis Subframe	Add 200 mm in x direction to the former point.
3. Spring Stabi Torsion Bar - Torsion Bar	This point defines the connection point of the third spring, the stabilizer and the rotation axis of the torsion bar. Therefore it needs to be placed correctly, even if there is no third spring used.
"Marker Damper Rocker Arm":	It does not effect the simulation results and is not needed in Formula Student cars. The only requirement for this point is, that the y-coordinate must not be 0 since this leads to an error during the simulation.
Car Body (Turning Circle Diameter)	Defines the space required by the car for turning. It describes the difference between turning track diameter and turning circle diameter. So, it is the point next to the outer skin of the car body, input values are the x- and y-coordinates and the radius to the outer skin. In Formula Student cars this value is of minor priority as the minimum turning track diameter is more important. Hence, you can set any values.

In case of using a T-Stabilizer by choosing push/pull rod as the stabilizer bar fixing, the stabilizer points shown in Figure 4.17 above will remain unused during the simulation. Nevertheless, the coordinates for the U-Stabilizer need feasible values and should be parameterized as explained above.

To illustrate the meaning of the points of the double wishbone model with the U-Stabilizer explained above, compare Figure 4.17: with Table 4.5: .

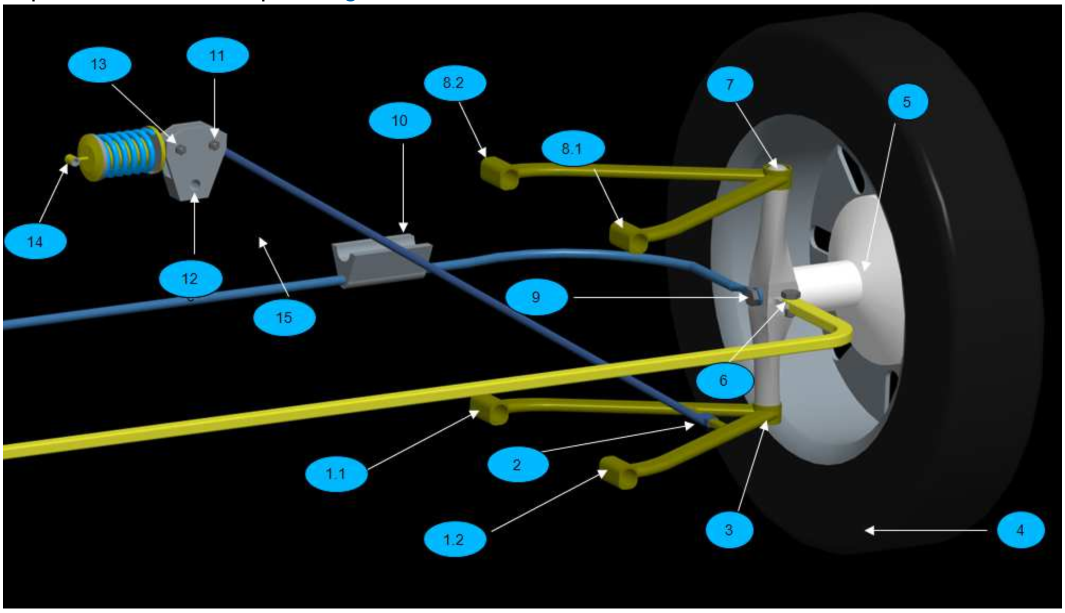


Figure 4.17: Overview of the kinematics points

Table 4.5: Listing of the geometry points in Figure 4.17

Number	Item
1.1/1.2	Chassis Subframe - Bushing Front/Rear Lower Wishbone
2	Pull/Pushrod - Wheel Suspension
3	Wheel Carrier - Lower Wishbone
4	Center of Tire Contact
5	Wheel Center
6	Wheel Carrier - Steering Rod
7	Wheel Carrier - Upper Wishbone
8.1/8.2	Body - Bushing Front/Rear Upper Wishbone
9	Wheel Carrier - Stabilizer Link (Stabilizer Link - Stabilizer Bar)
10	Chassis Subframe - Stabilizer Bar
11	Pull-/Pushrod - Rocker Arm
12	Body - Rocker Arm
13	Spring Element - Rocker Arm
14	Spring Element - Body
15	Rotation Axis Rocker Arm

The points shown in Figure 4.18 describe the T-Stabilizer. In case of using a U-Stabilizer, these coodinates should keep the default values.

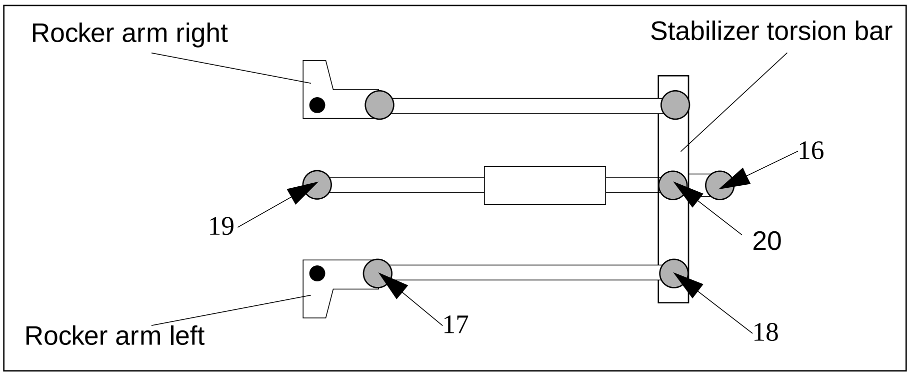


Figure 4.18: Geometry of Stabilizer via push/pull rod (top view)

Table 4.6:

Number	Item
16	Joint Stabi Torsion Bar - Body
17	Rod Stabi Torsion Bar - Rocker Arm
18	Rod Stabi Torsion Bar - Torsion Bar
19	3. Spring Stabi Torsion Bar - Body
20	3. Spring Stabi Torsion Bar - Torsion Bar

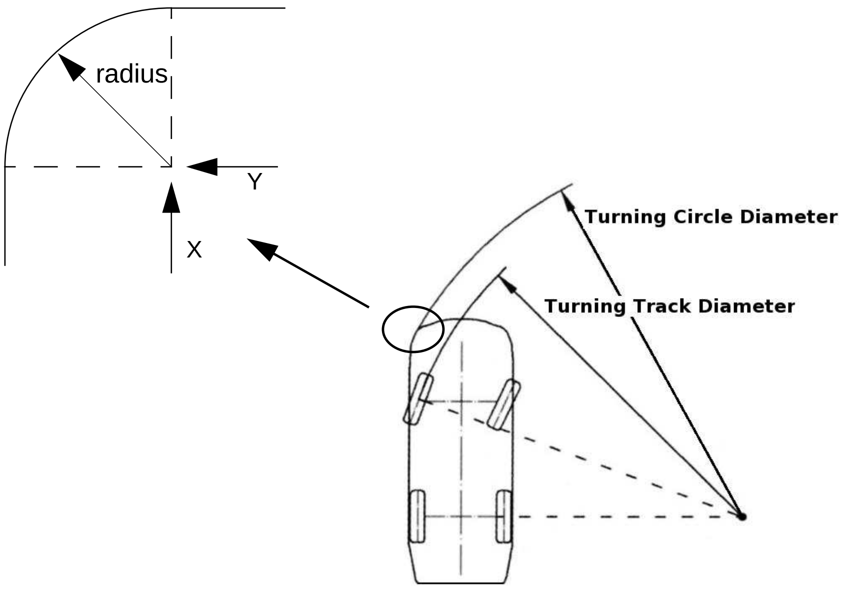


Figure 4.19: Turning circle and turning track diameter [HE07]

4.3 Output Data

4.3.1 IPGGraph

The graphical analysis is based on the created .erg files and the tool IPGGraph. It lets you plot any kind of diagram containing the calculated values. Open IPGGraph by clicking in the main GUI "Analyse > Graphical Analysis" or by hitting the corresponding icon.

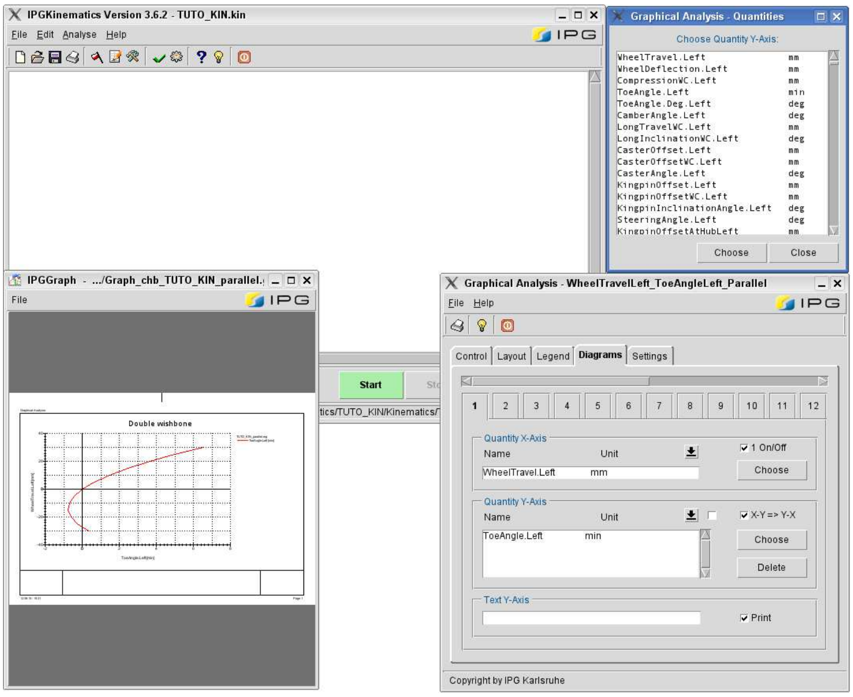


Figure 4.20: Graphical analysis in IPGKinematics

Control

In this tab you can load the results you want to analyze.

Choose Results This field is for loading the file. It will usually be `NameOfYourKinFile_parallel.erg` for the parallel compression part.

Choose Sequence The *Choose Sequence* field lets you selecet a predefined diagram layout that will load preset diagram parameters for different result files. If you are using IPGGraph for the first time there will be no sequences.

Layout

This tab lets you set the orientation of the page, number of the created plots and other settings regarding appearance of the diagram.

Legend

Here you can set up the information regarding the data fields in title blocks and document headers. You can add a logo aswell.

Diagrams

In this tab you can define which parameters you’d like to plot. You can set up to 24 diagrams in one sequence. To switch to other diagrams you have to left click in the number panel (1,2,3,..).

Quantity X-Axis Here you choose the reference variable.

Quantity Y-Axis Here you choose one or more corresponding signals to plot it against the reference variable.

 # On/Off Lets you leave out a particular diagram.

X-Y =>Y-X For inverting the X and Y axis you have to mark the checkbox.

Settings

This tab contains general graph settings. Both, the path to results and sequence should be set to their corrosponding path in your work folder.

IPGGraph will create the plots and open a viewer. If there are several pages, use the arrows on your keyboard to view them. Once the diagrams are displayed you can print them either with a printer or to a file (.ps or .pdf format): click on "File > Print".

To save the layout of your diagrams generate a sequence hitting the "Save" button in the "Control" tab in the *Choose Sequence* area and give it a suitable name. When you perform another simulation you can apply the diagram layout settings by selecting this sequence.

The next exercise will give a short example on how to create a plot.

Exercise

- Create a plot from a data set:
  - Open IPGGraph via "IPGKinematics > Analyse > Graphical Analysis"
  - Make sure that *Path to Results* and *Path to Sequences* lead to your project files created earlier ("Kinematics/firstTutorial" and "Kinematics/Sequences"). If not, edit the correct path.
  - In the "Control" tab load the file "doubleWishBone_parallel.erg". Do not define any sequences yet.
  - Parametrize the options in "Layout" to have a single diagram in landscape orientation. Leave the other options to default.
  - Feel free to add information to "Legend".
  - In the "Diagrams" tab choose quantities y(x) for:
    Tab 1) ToeAngle.Left (WheelTravel.Left)
    Tab 2) CamberAngle.Left (WheelTravel.Left)
- Print the plot and save the sequence:
  - Once all settings are complete you are ready to plot your diagram. To do so, click "File > Print" or the corresponding button. Your result should look like this:

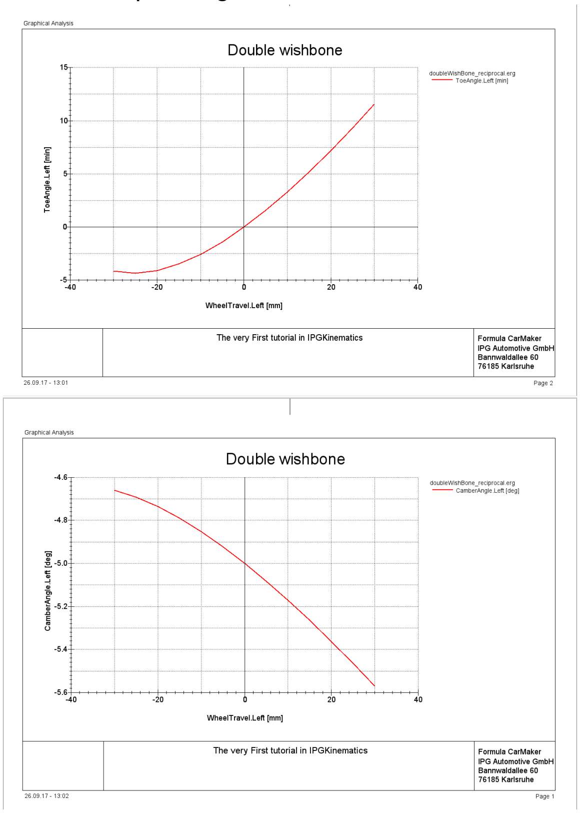


Figure 4.21: Graphical analysis in IPGGraph

  - If your plot is sufficient, save the sequence in the "Control" tab under *Choose Sequence* via "Save".

4.3.2 Exporting Results to CarMaker

As explained before, to export results to CarMaker you have to choose a special option: In "Simulation Control > Global > Storage of Results" select "CarMaker Interface (1)".

With this option among of the numerous data files generated by the simulation, one is called

- `NameOfResults_front.skc` (for a front axle) or
- `NameOfResults_rear.skc` (for a rear axle).

To use the IPGKinematics results in CarMaker you have to copy those skc-files and paste it to the "<yourProjectDirectory>/Data/Chassis" folder of your CarMaker project directory, for instance:

- "FS_Generic_2018/Data/Chassis".

We will see how to use this file in the following chapter.

Exercise

- Copy the skc files to the "Data/Chassis":
  - Look into your "Kinematics/firstTutorial" folder for the *.skc files.
  - Copy-paste them to "FS_Generic_2018/Data/Chassis".

Chapter 5

Preparing a vehicle dataset in CarMaker

Introduction

There are two ways to prepare a completely new vehicle dataset.

The first one is to adapt a pre-existing problem-free running dataset step by step. The main advantage of this method is, that the entered data can be checked easily. Knowing that the former test run worked without any problems, a driver adaption or a new simulation run tells you if there are any mistakes in the changes made.

The second opportunity is to use an empty dataset. This method saves a lot of time during the parametrization process. However, the time gained can be easily lost during a complex debugging process as the user doesn't know where exactly the mistake is located. A few common error messages are described in section 6.3.1.

For electric cars there are basically two different possibilities to create an electric drivetrain. Firstly, you can parameterize your electric powertrain directly in the CarMaker GUI. Again you have the choice to create your own powertrain or to modify one of the example vehicles (Rear Wheel Drive (RWD) or All Wheel Drive (AWD)) from the "FS_Generic_2018" project folder.The electric powertrain model in CarMaker gives the possibility to simulate one to four electric motors. Beside the motor and driveline characteristics, the operation strategy and the power supply are defined directly in CarMaker.
In order to test your own controllers, it is possible to substitute or expand the predefined control units with MATLAB/Simulink models.

Secondly, you can use the CarMaker interface to MATLAB/Simulink. You can create your own electric drivetrain model in MATLAB/Simulink and deactivate the IPG powertrain model in the vehicle data set as shown in our example "FS_Generic_XWD_6.0". The basic idea of the FSE_RaceCar is that the generic powertrain is substituted with a modified electric powertrain modeled in Simulink. You are free to design a model with one, two or even more electric motors.
Furthermore, the Simulink environment is a very convenient way to develop driver assistance systems like a simple traction control or an advanced Torque Vectoring model. Of course, this can be combined with the first approach of parameterizing the electric powertrain in the CarMaker vehicle data set, too.

5.1 General Vehicle Data Set

In case of preparing a vehicle dataset of a Formula Student racing car it is recommendable to choose the first option as a CarMaker model of a race car (FS_RaceCar_6.0) which already exists. It was designed on that purpose so that a Formula Student Team has guided values and needn’t start from stratch. Thus, you have an operating vehicle model with values of proper range. All you need to do is to adjust the pre-defined values to fit your car. To open the "Vehicle Data Set" window click: "Parameters > Car".

Note: Except if specified, all units are set in the International System of Units (SI). Accordingly, lengths are in meter and angles in radian (pay attention, not degree!).

Exercise

- Select the "FS_RaceCar_6.0" in the Main GUI.

5.1.1 Vehicle Body

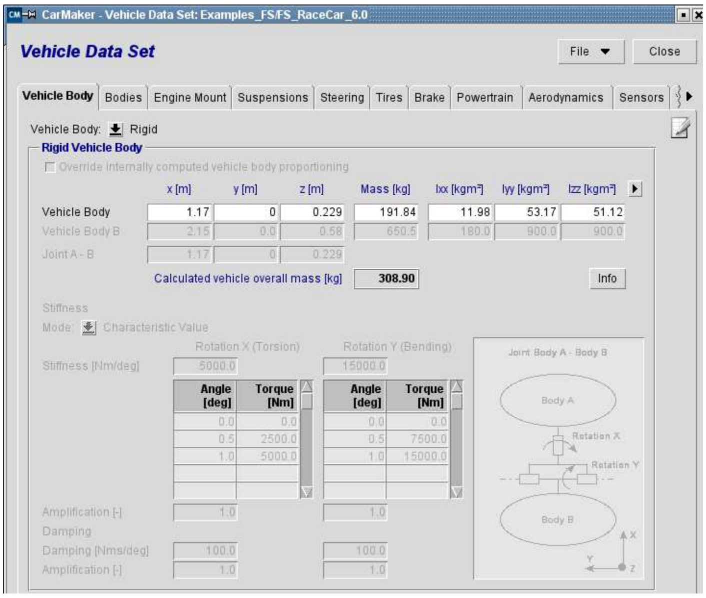


Figure 5.1: Vehicle Data Set - Vehicle Body

In this tab you can choose your body to be flexible if you select "Vehicle Body > Flexible". This option allows to regard bending and torsion effects of the main body. Otherwise the car is considered as infinitely stiff. By activating the option "Flexible", the hidden input boxes turn active. In the upper part you can separate your body into two bodies called A and B - A for the front and B for the rear part. Both parts are defined by their own mass, center of mass and inertias.

Below coefficients of torsional body stiffness in x- and y-direction can be defined. When choosing the mode "1D Look-Up Table" instead of "Characteristic value" a non-linear characteristic can be edited.

Please keep in mind, that only very exact (measured) values for a flexible body make sense, otherwise the use of the rigid body is recommended.

Calculating Center of Gravity

The calculation of the vehicle center of gravity is very complex and challenging. Even with the help of CAD and CAE tools, it is only possible when every single body is defined in the model with all its physical properties.

A much easier possibility is to measure the total center of gravity. Therefore, the car must be brought to an absolutely horizontal position. To measure the axle load, one axle at time is put on the scales. Setting up the balance of moments at the front axle the position of the center of gravity can be determined as follows:

$$l_f = \frac{m_{V,r}}{m_{V,t}} \cdot l \quad \text{(EQ 21)}$$

$$l_r = \frac{m_{V,f}}{m_{V,t}} \cdot l \quad \text{(EQ 22)}$$

With

- l = wheelbase
- l_f = distance between the center of gravity and the front axle
- l_r = distance between the center of gravity and the rear axle
- $m_{V,f}$ = axle load at the front
- $m_{V,r}$ = axle load at the rear
- $m_{V,t}$ = total weight

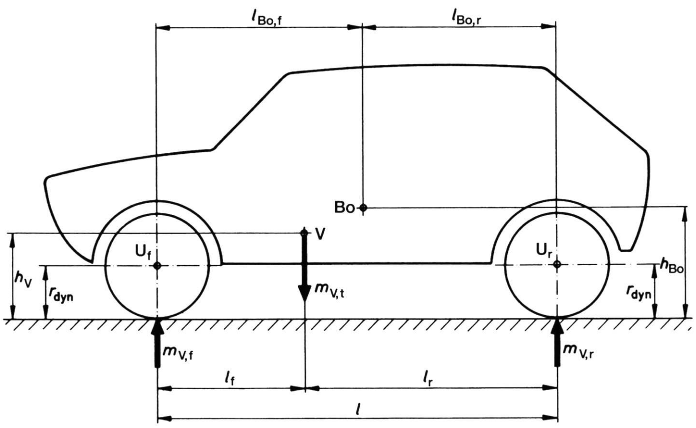


Figure 5.2: Locating of the vehicle center of gravity [RB00]

To determine the height of the center of gravity a second measuring setup is required. In this, one axle is jacked up by the height h. It is necessary to jack the vehicle as high as possible to avoid measuring faults. Moreover, the following issues should be taken into account:

- Rolling of the car should be prevented using wedges. Brakes should be released and the engine should idle. If this is not the case, bracing can occur which are likely to affect the exactitude of measurement.
- The wheels should be placed in the middle of the scales to prevent inexact measures due to different force initiation points.

- The vehicle should be roadworthy which means refuelled including the driver with his equipment.
- The axle springs should be blocked to prevent deflection. Otherwise this could lead to measuring faults.
- To prevent tire deflection a high pressure should be attained.

The car’s angle of slope is defined as follows:

$$\sin\alpha = \frac{h}{l} \qquad \text{(EQ 23)}$$

To determine the height h_V of the center of gravity Δl_r (see Figure 5.3) is required because of

$$h_V = h'_V + r_{dyn} \qquad \text{(EQ 24)}$$

$$h'_V = \frac{\Delta l_r}{\tan\alpha} \qquad \text{(EQ 25)}$$

Δl_r can be calculated setting up the balance of moments for the front axle. Then, the height h_V can be estimated with the following equation:

$$h_V = \frac{l}{m_{V,t}} \cdot \frac{\Delta m}{h} \cdot \sqrt{l^2 - h^2} + r_{dyn} \qquad \text{(EQ 26)}$$

The meaning of the single parameters is shown in the figure below:

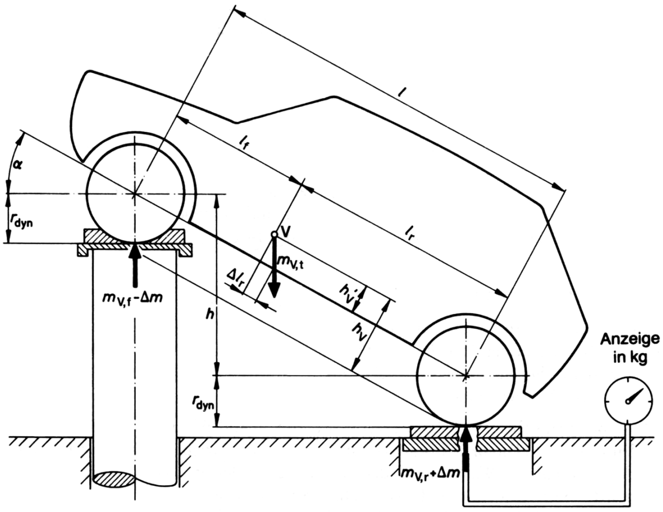


Figure 5.3: Determining the height of the center of gravity [RB00]

With this procedure, however, only the center of gravity of the complete vehicle can be measured. The many centers of gravity of single components being required by CarMaker must be determined in a different manner. To locate the driver’s center of gravity the measurements explained above are carried out with and without the driver. Afterwards the center of gravity can be calculated setting up the balance of moments.

The same procedure enables the location of the engine’s center of gravity. The determination of the other masses required can be achieved by weighing the single parts. Beware, the half-sprung parts such as wishbones, tie rods and driving torques are added one half each to the body and to the wheel carrier.

Moments of Inertia

One attempt to estimate the moments of inertia is to replace each assembly by a simple geometric body and then calculate the moments of inertia with the known formulas. These approximations deliver acceptable results, though.

More exact values can only be gained with complicated and cost-intensive experiments. Here, the moments of inertia are determined by the oscillation period of the considered part on a torsion pendulum in relation to the oscillation period of a reference body. For this reference body a simple geometry is chosen so that its moments of inertia can be easily calculated for comparison. An even better possibility offers a test bench especially designed to determine the inertial tensor of fourwheeled vehicles.

5.1.2 Bodies

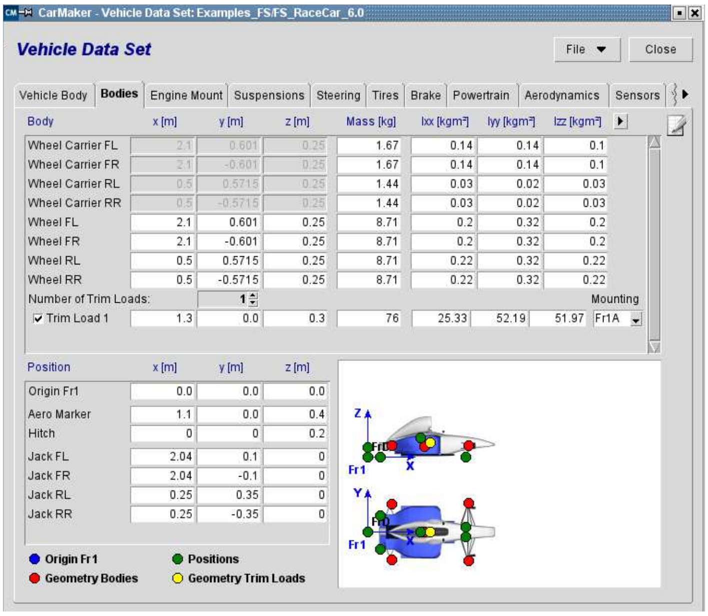


Figure 5.4: Vehicle Data Set - Bodies

Each simulation model is based on the masses of the single bodies and their moments of inertia. Concerning vehicles these masses are the suspension (unsprung masses) and the body (body masses). In CarMaker these two groups are divided into further sections. The unsprung masses are classified into spinning (e.g. wheel, rim, wheel-hub) and non-spinning masses (e.g. wheel carrier, half the wishbone or brake caliper). So, do not forget to add the weight of the disk brake to the wheel masses. Furthermore, note that the unsprung masses which have a link with the body should only be considered by half of their weight as the rest isn’t unsprung.
The driver can additionally be defined with the "Trim Load" entries.

Positions CarMaker calculates the car’s movements in an axis system referred to as "Fr1". The origin should be placed at the rearmost end of the car (see section "Coordinate System", pg. 54 or the CarMaker Reference Manual). However, you may have the values of the "Masses" positions in an axis system whose origin is forward the rear axle. If so, in the box "Origin Fr1" you can define a translation between Fr1 and your design axis system and you will be able to write the positions directly with the values you initially have.

5.1.3 Engine Mount

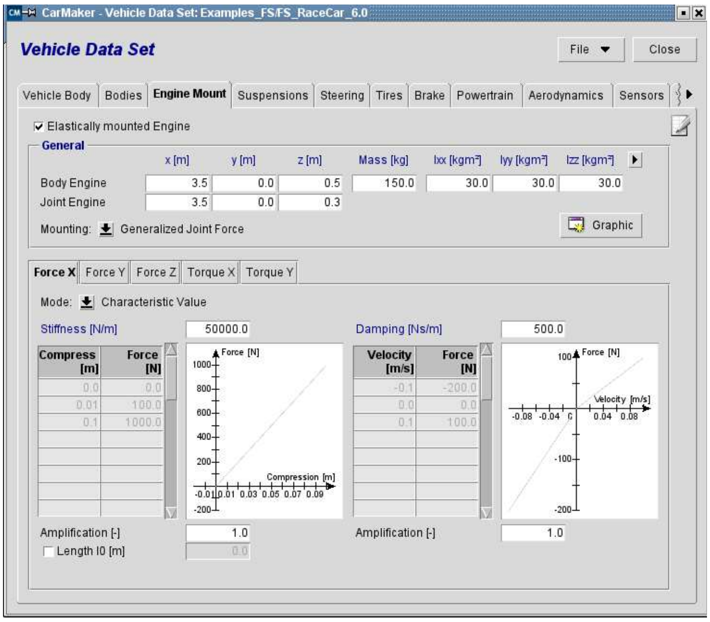


Figure 5.5: Vehicle Data Set - Engine

There are three possibilities to get the engine's mass into the vehicle data set. The easiest way is to add the mass to the *Vehicle Body* masses, especially when you do not have any detailed information about it. When you know the position of the engine's center of gravity and maybe the moments of inertia, you should define a *Trim Load* under the "Bodies" tab.

Lastly, you can activate the option *Elastically mounted Engine* under the "Engine" tab and enter all the needed parameters. This helps you to tune your race car model. More information about the elastical mounting can be found in the Reference Manual at section "Engine".

Exercise

- Edit the masses:
  - Increase the vehicle body weight to 220 kg and its center of gravity to 0.5 m above the ground (RigidBody, Vehicle Body A).
  - Then subtract 0.2 kg from the spinning disc brakes which are included in the wheels.
  - Keep the moments of inertia values same.

5.1.4 Suspensions

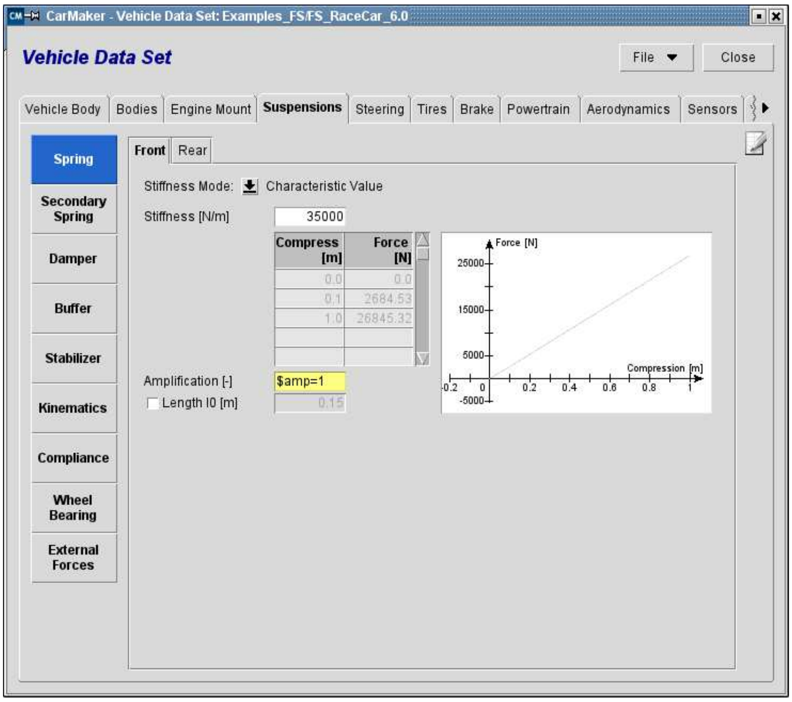


Figure 5.6: Vehicle Data Set - Suspensions

Spring

The spring rate of the axle spring has already been defined in IPGKinematics. The same values should be entered here. The parameter l_0 is the actual free length of the spring in its unmounted state. You can let this value be calculated automatically by CarMaker which does some iterations during the determination of the car's static equilibrium position in the preparation phase ahead of every simulation. If the car's position on the road does not match with your real car, this value serves as an adjustment parameter.

Exercise

- Edit the spring:
  - Increase the front stiffness up to 45 000 N/m and the rear stiffness up to 52 300 N/m. Leave the length l_0 unticked.

Damper

The final damper settings are mostly determined directly on the vehicle through empiric test runs. On that reason not even the average damper force for the final damper setup is known. Accordingly a significant damper characteristic can only be determined via test runs on the real vehicle or using special test benches. If both opportunities are available you should always choose the test bench, as measurements on the real vehicle can be distorted by friction on the suspensions. Moreover, test benches enable a wide range of speed (up to n = 200 min^{-1}) and travel (5 - 150 mm) to be tested. They also deliver directly the required characteristic "damper force via speed".

Exercise

- Set the damper values to:

Tabelle 5.1: Push values

Damper Front - Push:		Damper Rear - Push:	
velocity [m/s]	force [N]	velocity [m/s]	force [N]
0.0	0.0	0.0	0.0
0.125	525.80	0.125	566.25
0.25	788.70	0.25	849.37

Tabelle 5.2: Pull values

Damper Front - Pull:		Damper Rear - Pull:	
velocity [m/s]	force [N]	velocity [m/s]	force [N]
0.0	0.0	0.0	0.0
0.125	1183.06	0.125	1274.06
0.25	1774.58	0.25	1911.09

Buffer

The springs used in Formula Student cars are so stiff that only little spring travels are possible. However, construction-conditioned much bigger travels are feasible. For that reason and to prevent resiliences making the car softer, buffers are commonly not used in those cars.

To implement that in CarMaker, two steps are required. On one hand, a large stiffness is attached which reduces the elasticity considerably. On the other hand, the positions of the buffers are set very high (respectively low) so that they aren't activated in general. The length tz0 defines the limit above which the wheels are prevented to deflect. For a clear illustration see also Reference Manual, chapter "Suspension Force Elements".

Stabilizer

The stabilizer parameters have already been explained in section "Parameter Stabilizer", pg. 63. In CarMaker the same stiffness must be used. Beware: CarMaker uses a linear spring rate [N/m] as opposed to IPGKinematics which uses a torsional spring rate [N/deg]. The following will show you how to convert the spring rate.

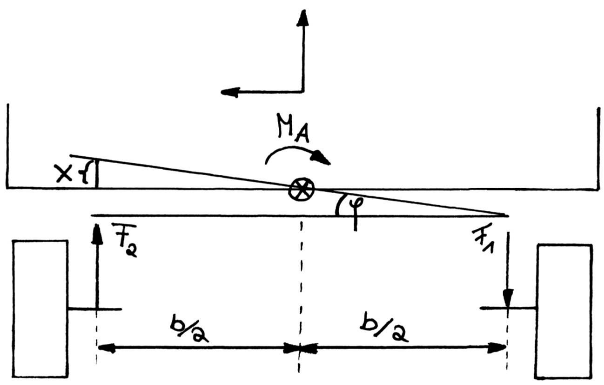


Figure 5.7: Converting the stabilizer bar spring rate

The translation is based on the balance of moments around the pivotal point and on the balance of forces in z-direction:

$$F_1 \cdot \frac{b}{2} + F_2 \cdot \frac{b}{2} + M_A = 0 \quad \text{(EQ 27)}$$

$$F_1 = F_2 \quad \text{(EQ 28)}$$

Solving and inserting into each other provides:

$$F_1 = \frac{M_A}{b} \quad \text{(EQ 29)}$$

Further the rolling moment M_A is known:

$$M_A = c_\varphi \cdot \varphi \quad \text{(EQ 30)}$$

With M_A being inserted:

$$F_1 \cdot b = c_\varphi \cdot \varphi \quad \text{(EQ 31)}$$

In consideration of:

$$F = c_x \cdot x \quad \text{(EQ 32)}$$

and

$$\tan\varphi = \frac{2x}{b} \cong \varphi \quad \text{(EQ 33)}$$

you finally get the stabilizer bar spring rate:

$$c_x = c_\varphi \cdot \frac{2}{b^2} \quad \text{(EQ 34)}$$

Kinematics, Compliance and External Forces

All these fundamental parameters were calculated using IPGKinematics. To import the data in CarMaker load both of the skc-files for front and rear axle (see section 4.3.2).

Exercise

- Load the generated skc-files:
  - If you have generated a IPGKinematics dataset for the front axle as well as for the rear axle, then load both via "Vehicle Data Set > Suspension > Kinematics > SKC-File".

5.1.5 Steering

The steering system consists of a mechanical module which defines the ratio between the steering wheel angle (or steering wheel torque) and the steering rack displacement. Optionally, a power steering unit can be added by selecting the Pfeffer Steering Model which also regards frictions losses in the steering system.

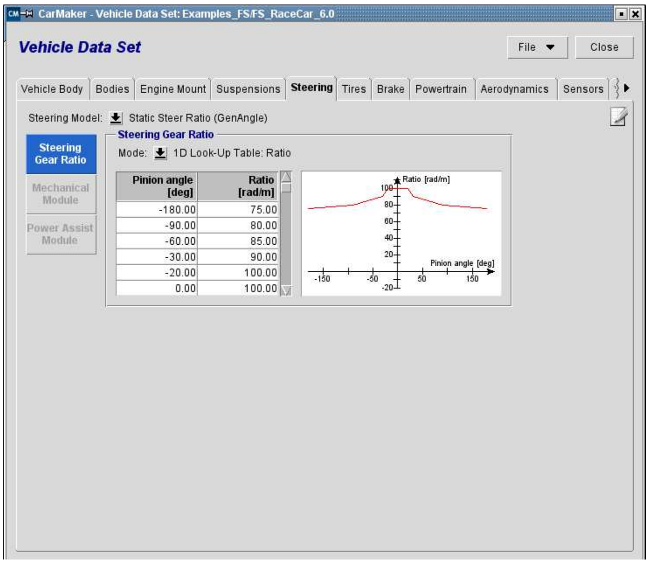


Figure 5.8: Vehicle Data Set - Steering

Steering Model

Static and Dynamic Steer Ratio

To simulate a steered axle CarMaker only needs the ratio between steering wheel angle (in radians) and steering rack travel (in meters). The ratio does not have to be static: non-linear maps are available under the "Mode" selection menu.
Note: The ratio between steering rack travel and wheel angle is a part of the kinematic data generated in IPGKinematics and will be transferred to CarMaker using the skc files.

Pfeffer with Power Steering

You can also use the Pfeffer Power Steering model which is a very detailed steering model including friction loss effects, elasticities and power steering.

Mechanical Module This model consists of a mechanical module which includes all mechanical components (transferring torque from the steering wheel to the tie rods).

Power Assist Module Also a power assistance module e.g. hydraulic (HPS), electrical (EPSc and EPSapa) can be parameterized.

In the User’s Guide ("main GUI > Help > User’s Guide"), section "Parameterization: Vehicle Model > Steering System" you find more information about the different models.

5.1.6 Tires

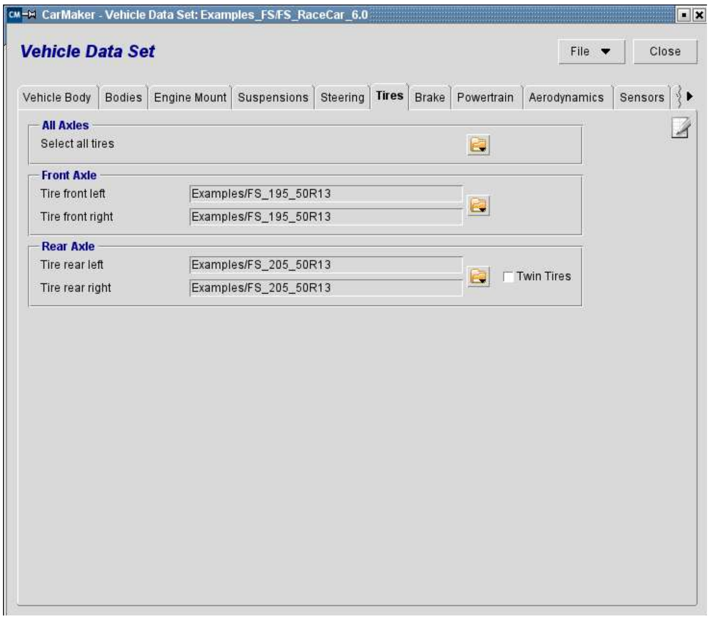


Figure 5.9: Vehicle Data Set - Tires

Here the tires for the race car can be selected. The tire attributes belong to the vehicle parameters and will be loaded automatically by selecting a car. Additionally you have the opportunity to overrule these tire files by selecting another one in the CarMaker main GUI. Learn more about how to import own tire data in section "Creating a Tire Dataset Using IPGTire", pg. 121.

5.1.7 Brake

The brake system in CarMaker is divided into two parts: The Brake Control unit and the Brake System. Whereas the first is the interface for controllers the brake system represents the physical brake unit. The pre-implemented brake unit is a hydraulic brake system with a control unit that supports e.g. regenerative braking.

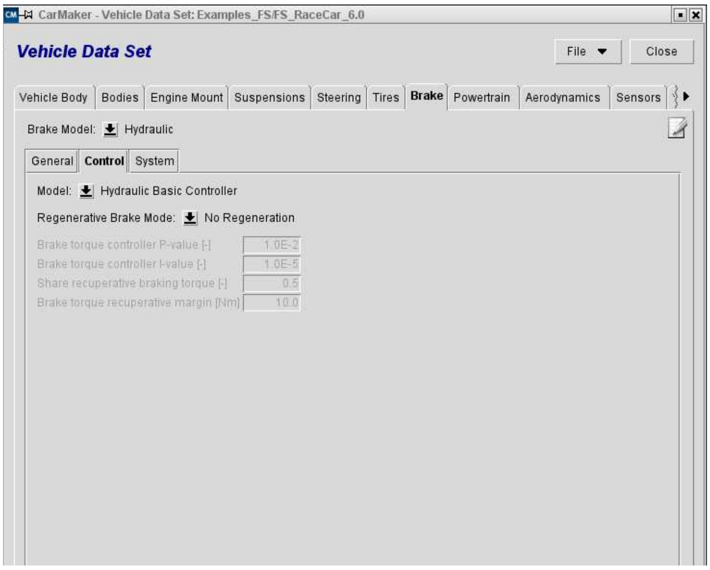


Figure 5.10: Vehicle Data Set - Brake Control

The Hydraulic Brake Control model "HydBasic" gives the possibility to use regenerative wheel brake torque in a electrical or hybrid powertrain. The regenerative brake torque can either be applied by a parallel or a serial strategy as show in Figure 5.11.

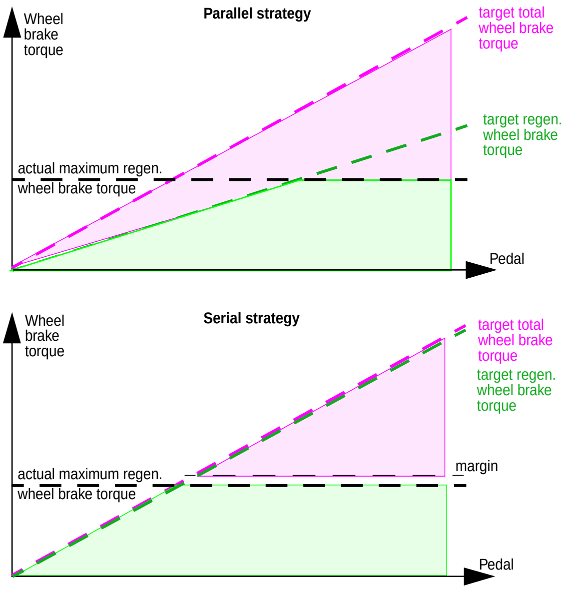


Figure 5.11: Brake torque repartition strategy

The actual hydraulic brake system is parameterized in the "System" tab, shown in Figure 5.12.

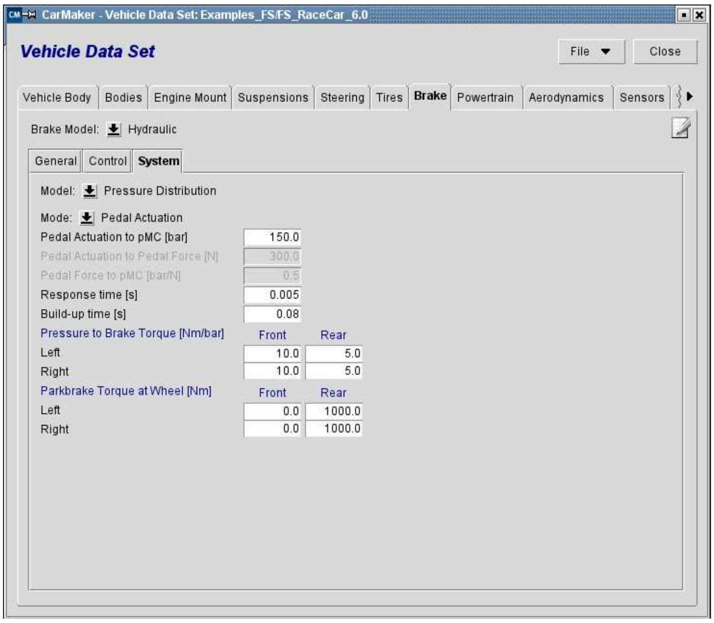


Figure 5.12: Vehicle Data Set - Brake System

It takes several steps to transform the brake pedal force applied by the driver into a brake torque at the wheels:

- Applying the pedal force by the driver.
- Enforcing the pedal force through the pedal transmission.
- Transforming this force into a pressure at the main brake cylinder.
- Dividing the pressure into two brake cycles.
- Transforming the pressure into a brake torque.

These five steps are implemented in CarMaker by the definition of the maximum pedal force and two transmissions. The meaning of the two transmissions "Pedal Force to p_{MC}" and "Pressure to Brake Torque" will be explained in detail in the following.

Table 5.3: Required parameters for the calculation of the brake system

Explanation	Variable Name
Pedal to Pedal Force	F_P
Transmission Braking Pedal	i_{bp}
Diameter Main Brake Cylinder	d_{mbc}
Diameter Brake Piston Front	$d_{bp,f}$
Diameter Brake Piston Rear	$d_{bp,r}$
Adhesion Coefficient of the Brake Pads	μ_b
Effective Radius Braking Disk Front	$r_{bd,f}$
Effective Radius Braking Disk Rear	$r_{bd,r}$
Brake Portion Front	b_f
Brake Portion Rear	b_r

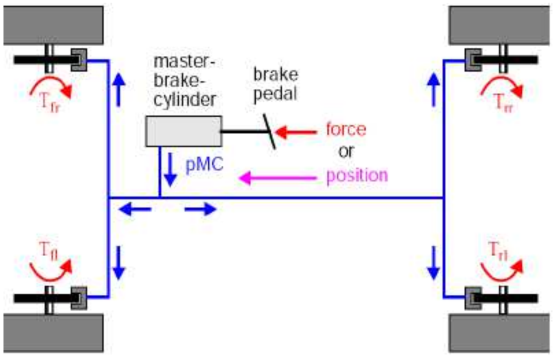


Figure 5.13: Schematic layout of the brake system parameters

The first transmission "Pedal force to p_{MC}" is made up by the quotient of the pressure at the main brake cylinder p_{MC} and the pedal force F_P. Therefore the pressure p_{MC} must be determined first:

$$p_{MC} = \frac{F_{mbc}}{\pi \cdot \frac{d_{mbc}^2}{4}} \tag{EQ 35}$$

The force F_{mbc} at the main brake cylinder results from the pedal force reinforced by the pedal transmission:

$$F_{mbc} = F_P \cdot i_{bp} \tag{EQ 36}$$

Thus follows:

$$\frac{p_{MC}}{F_P} = \frac{i_{bp}}{\pi \cdot \frac{d_{mbc}^2}{4}} \tag{EQ 37}$$

In general the ratio "Pressure to Brake Torque" is different at front and rear axle. This results from an unequal brake balance and the used brake types which are mostly different at front and rear axle. Thus, this ratio must be calculated for each axle separately.

The pressure at the front brake piston is:

$$p_f = b_f \cdot p_{MC} \tag{EQ 38}$$

The contact pressure at the front brake disc constitutes to:

$$F_{ap, bd, f} = p_{VA} \cdot \pi \cdot \frac{d_{bp, f}^2}{4} \tag{EQ 39}$$

With the brake force at the front disks

$$F_{bd, f} = F_{ap, bd, f} \cdot \mu_b \tag{EQ 40}$$

the brake torque at the front wheels can be estimated to:

$$T_f = 2 \cdot r_{bd,f} \cdot F_{bd,f} \qquad \text{(EQ 41)}$$

Using equation (EQ 35) - (EQ 38) the ratio "Pressure to Brake Torque Front" can be determined:

$$\frac{T_f}{p_{MC}} = 2 \cdot r_{bd,f} \cdot \mu_b \cdot \pi \cdot \frac{d_{bp,f}^{\ 2}}{4} \cdot b_f \qquad \text{(EQ 42)}$$

The ratio "Pressure to Brake Torque Rear" is calculated in the same way:

$$\frac{T_r}{p_{MC}} = 2 \cdot r_{bd,r} \cdot \mu_b \cdot \pi \cdot \frac{d_{bp,r}^{\ 2}}{4} \cdot b_r \qquad \text{(EQ 43)}$$

The formulas guide you through the parametrization of the linear brake model "Pressure Distribution".

If you have detailed information about your brake system, you can activate the more precise brake model "Hyd_ESP_FS_RaceCar_6.0" by selecting "External File" model ("Vehicle data set > Brake > System > Model"). This model regards characteristics of the single volumes, pipes and the hydraulic fluid. Moreover, it covers the brake booster, as well as the suction and pilot values as an interface for ABS and ESP controllers.

For more information about this very detailed model please refer to the Reference Manual or open the template "HydESP_FS_RaceCar_6.0". Each line starting with a hash is a comment.

Exercise

- Increase the braking ratio up to 75% without changing the available input braking force:
  - Set the "Pressure to Brake Torque" front to 12 at each wheel and the "Pressure to Brake Torque" rear to 4 at each wheel.

5.1.8 Powertrain

Please refer to section "Powertrain: Combustion Race Car", pg. 93 or section "Powertrain: Electric Race Car", pg. 99.

5.1.9 Aerodynamics

As aerodynamics become ever more important in the Formula SAE competition, you can use the detailed aerodynamics model in CarMaker to simulate its effects. In dependence on the air flow angle tau, the coefficients for the resistance forces and torques in all three directions of space have to be specified. For this, a wind tunnel test or at least a very comprehensive CFL model is required.

Please find further information about the measurement procedure of the required coefficients in the Reference Manual.

General The "Reference Point" determines the point of attack for the combined wind forces. The "Reference Area" usually describes the front area of your car and the "Reference Length" corresponds to the wheel base of your car.

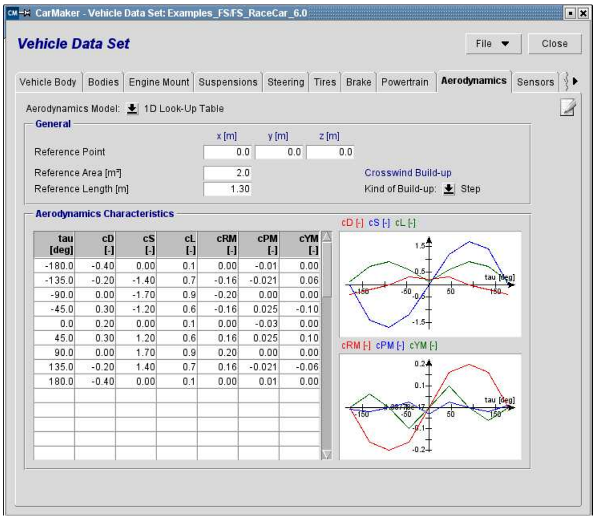


Figure 5.14: Vehicle Data Set - Aerodynamics

Sensors

In the Sensor tab you have the opportunity to create different kinds of sensors.

Slip Angle This sensor is bound to the center of gravity normally. If you like to place it anywhere else just change the parameters of the sensor's position.

Intertial A Body Sensor is an inertial sensor that can be placed anywhere on the vehicle to measure the movements, velocity, rotational velocity, acceleration and rotational acceleration. All corresponding output quantities will be attached to IPGControl automatically. To get to know more about sensors please read the Reference Manual.

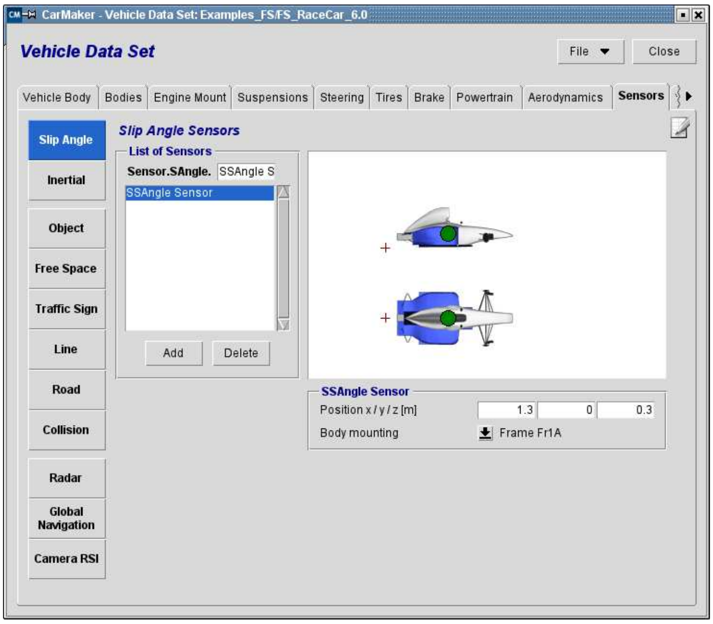


Figure 5.15: Vehicle Data Set - Sensors

5.1.10 Vehicle Control

This tab is only relevant to integrate controller systems, mainly for driver assistance systems.

The integration of controller models relevant for Formula Student cars is demonstrated by several CarMaker for Simulink examples such as "FSE_TorqueVectoring.mdl", "HydBrakeCU_ESP.mdl".

5.1.11 Misc.

The object file used by IPGMovie can be selected under *Movie Geometry*. This file can be created by the user himself with the help of a CAD software, or one can be bought online. The file must be in .obj, .kmz, .dae or .3ds format.

File definition guidelines:

- ONLY polygons and triangles, NO splines or free forms.
- No more than 50,000 nods.
- If this option is available while generating the file: generate the normal for all points
- Colors: Generate the corresponding material file.

Under Vehicle Graphics, a picture in the PNG format can be chosen to be displayed in the CarMaker main GUI and in the vehicle editor as preview to the 3D object. The user can also create an own picture by taking snapshots of the 3D-object by the help of the button next to the file browser. The field Additional Parameters gives room for additional, optional parameters.

Please find further information on the various options for obj files in the User's Guide ("main GUI > Help > User's Guide"), section "Parametrization: Vehicle Model > Miscellaneous".

Vehicle Outer Skin

If an obj. file as described above is not available, there is another possibility to display your car in IPGMovie: an Abraxas model. This model is basically a simple, rectangular box which refers to the maximal dimensions of your car. To determine these maximal dimensions two points are required which face each other diagonally. It's these very points which have to be entered here.

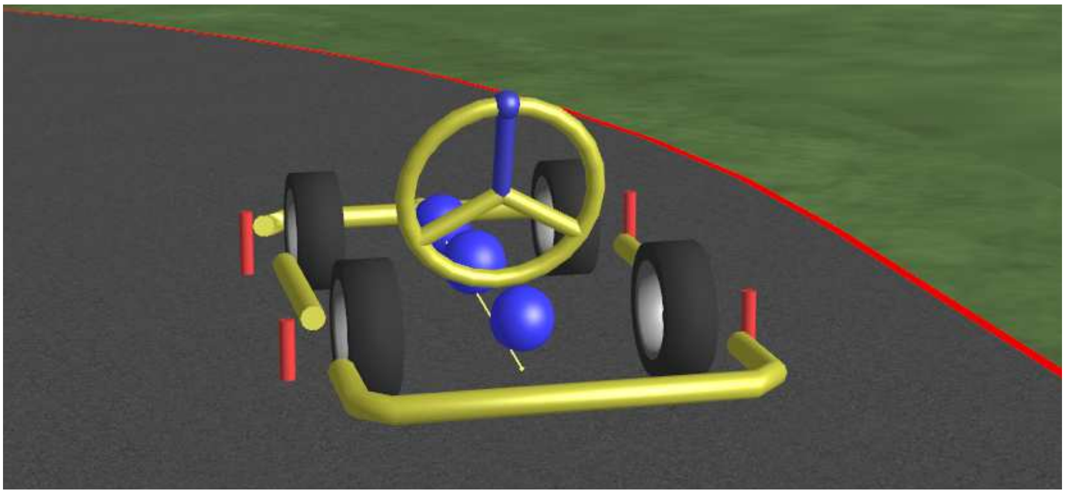


Figure 5.16: Outer dimensions of the FS_RaceCar shown in the Abraxas modes

5.2 Powertrain: Combustion Race Car

5.2.1 General

CarMaker gives you the possibility to choose between several powertrain models such as pure combustion, hybrid and electric vehicles. You can start the parameterization of any powertrain model by choosing predefined settings. To do so, simply right click in the powertrain window or select one from the "Predefined" button (see Figure 5.17:)

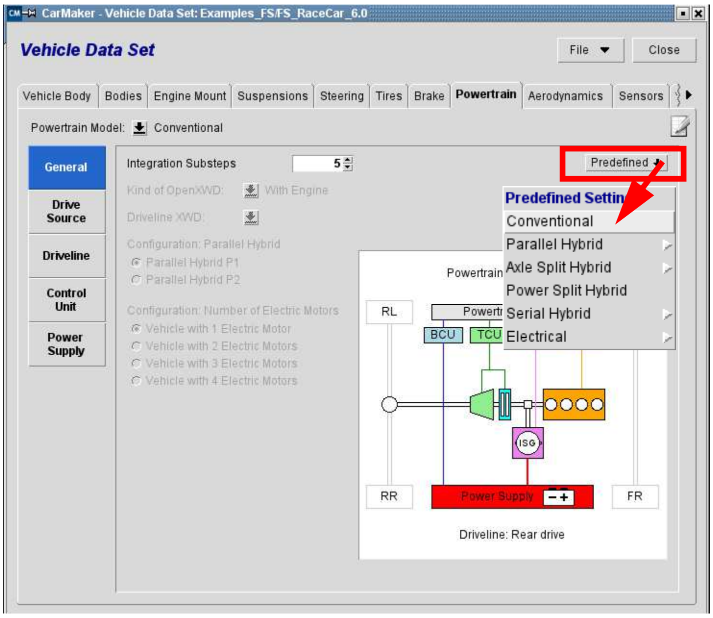


Figure 5.17: Vehicle Data Set - Combustion Engine. Selecting predefined models via drop-down menu.

Depending on the selected powertrain model, one to four drive sources (numeration: 0-3) appear in the powertrain window. The drive source characteristics, as well as the gearboxes and clutches can be parameterized in the corresponding drive source tab.
The connection between the drive sources and the wheels is defined by the driveline model.
The required control units for hybrid or electrical vehicles can be parameterized in the corresponding tab.
Finally, the power supply is presented as a LV and, if necessary, a HV battery model.

The first thing to do in order to parameterize a combustion powertrain is selection the conventional powertrain model with a right click in the powertrain window. "Conventional" is a common combustion engine model including clutch, gearbox and differential (e.g. the FS_RaceCar_6.0). All quantities and parameters from the engine right up to the wheels can be set by the GUI.

5.2.2 Drive Source

Engine

General The engine used in your Formula Student car can be defined by several ways. One possibility is to use the Engine Model "Characteristic Value" what assumes a linear distribution between gas pedal and engine torque. You can also choose the option "Look-Up Table".

Torque With the option "Look-Up Table" you have the opportunity to define a full load power curve (1D Look-Up Table) by inserting single points. This option is a lot more precise, but you have to do measurements on an engine test bench.
Another exponent defines the transition from the full load to the part load power curve. Along with the full load characteristic some other engine parameters like the drag torque characteristic, the range of speeds and the idle speed are required. Note that the entered full load power curve must start and end at zero.

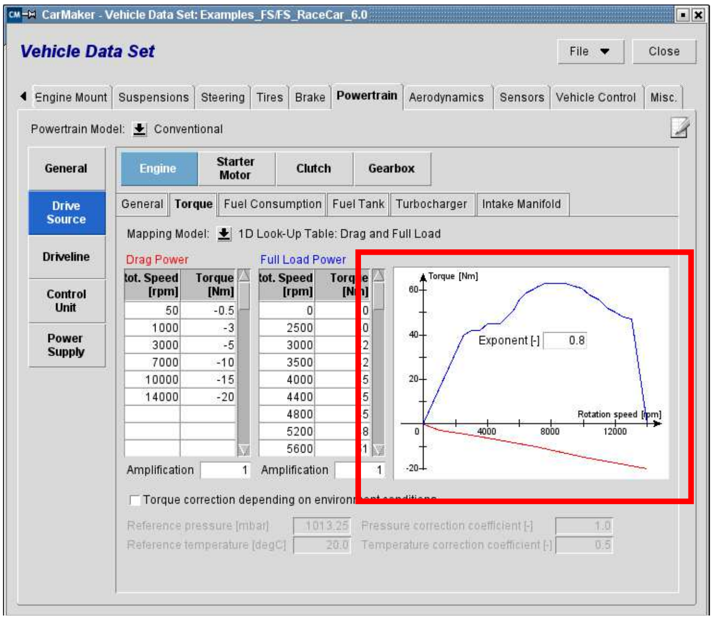


Figure 5.18: Vehicle Data Set - Combustion Engine

A further possibility is to define a 2D Look-Up Table with the dependence between the gas pedal, the engine speed and the output torque.

Fuel Consumption This feature enables a prediction of the fuel consumption. Activate the tick box and you have the possibility to enter values of specific fuel consumption. Therefore, you have to do measurements which reflect these points.

Fuel Tank, Turbocharger, Intake Manifold These values can be left by their default values. For further informations, please take a look at the Reference Manual.

Starter Motor

It is possible to define the power and torque of the starter motor as well as the power supply source. For Formula Student cars, the predefined values can be used.

Clutch

The pre-defined values can normally be used for a Formula Student car. Choose the mode "Friction".

Gearbox

Here, the transmission ratio of each gear has to be specified. Furthermore, there is an opportunity to define the time for synchronization. Although Formula Student cars usually don't have any reverse gear, it is necessary to define one as otherwise the program generates an error message.

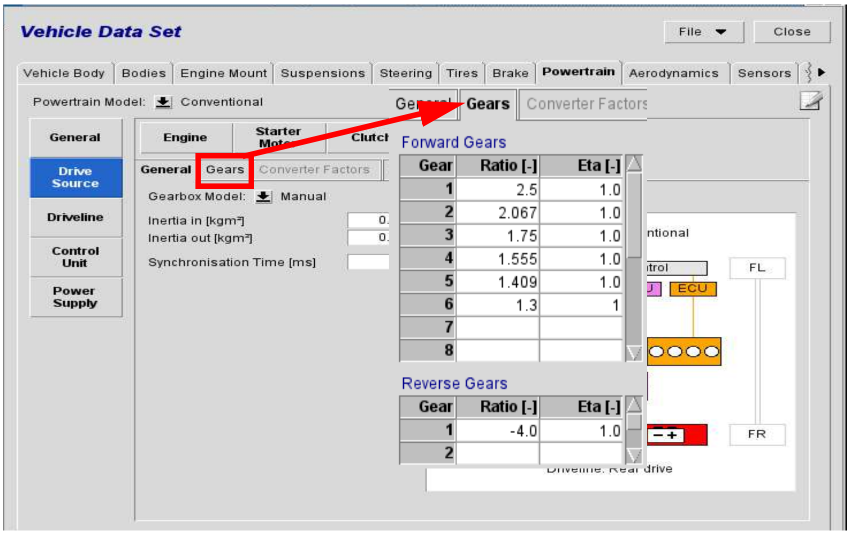


Figure 5.19: Powertrain - Gearbox

5.2.3 Driveline

General

The *Driveline Model* determines the connection between the Drive Source and the wheels. For most Formula Student combustion cars, the "Rear Drive" model should be suitable.

Rear Axle

For a Formula Student car choose the model "Torque Sensing" and the mounting mode "Left to Right".To parameterize the differential, the three values *Torque Bias Ratio Driven, Torque Bias Ratio Dragged,* and *Transmission* should be adjusted.

In case you use a motorcycle powertrain, you have a primary ratio between the engine and the gear box. You should consider this primary ratio together with the final ratio. Multiply the final transmission ratio by the primary transmission ratio for the *Transmission* value.

The torque bias (TB) is used when a wheel is spinning (e.g. on ice) while the other one has more grip. In this situation some differentials are able to limit the torque transmitted to the spinning wheel (T_{low}) so as to favor the wheel that has more grip (it receives T_{high}). TB is the factor between T_{low} and T_{high}. For instance: if TB = 2, the differential will limit the torque to the spinning wheel to 33.3% and give 66.6% to the other wheel (twice as much as at the spinning wheel).

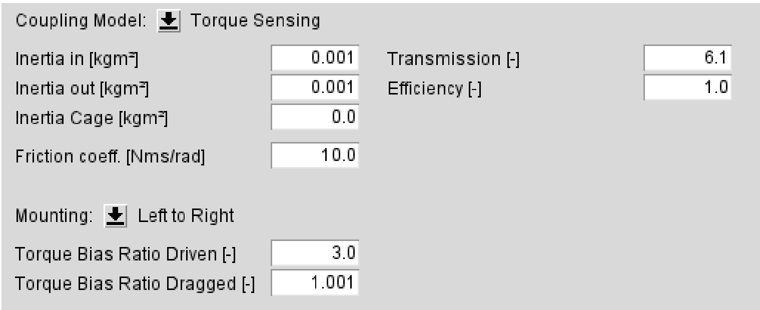


Figure 5.20: Powertrain - Differential Rear

5.2.4 Control Unit

The powertrain model contains four Control Units:

- ECU (Engine Control Unit) - Combustion Engine
- TCU (Transmission Control Unit) - gearboxes and clutches
- MCU (Motor Control Unit) - electric motors including the integrated starten motor
- BCU (Battery Control Unit) - power supply including the batteries

PT Control

The PT Control is a global control unit whose main task is to provide target values for the controler algorithms of the other control units

.

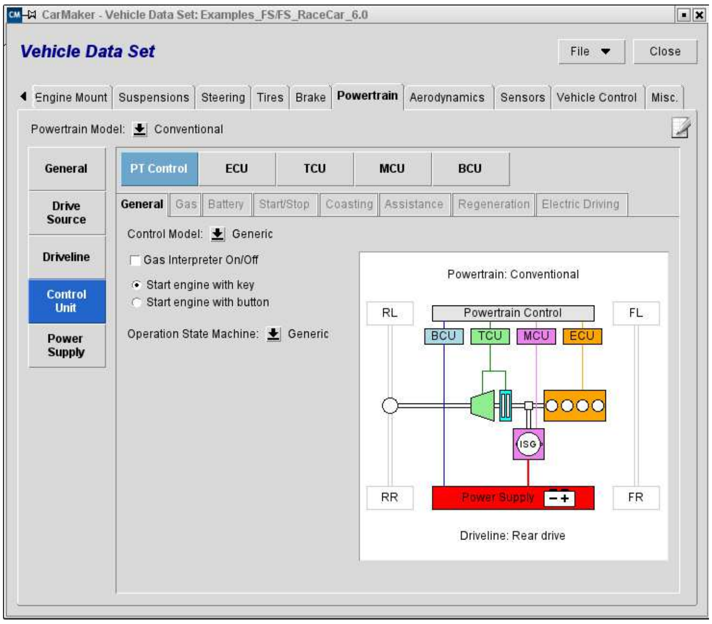


Figure 5.21: Powertrain - Control Units

For a FS combustion car, the control model "Generic" is most suitable. This model allows driving with the combustion engine only, without any use of the starter motor or other electric motors during driving.
The *Operation State Machine* can be set to "Generic" as well.

ECU

The Basic ECU Model contains a PI controller for the idle speed control and the load controler. The predefined values can be used for most FS cars.
For more information on the P/I values, please take a look in the Reference Manual.

TCU

There are five different TCU Models in CarMaker. For a automatic gearbox, the logic for gearshifting and the shifting limits need to be parameterized.
In case of a manual gearbox that is shifted by the driver, select "- not specified -" as TCU Model.

MCU

The Basic MCU Model consists of a PI controller for the load control of all electric motors. For a pure combustion car, this is only the starter motor. The predefined settings should be suitable for most applications. For more informations on the controller, please take a look in the Reference Manual.

BCU

The BCU calculates characteristic battery values such as the soc. If the battery is not of interest, the BCU model can be declared as not specified.

5.2.5 Power Supply

The Power Supply consists of the electric circuits, the battery models and the auxiliary consumers. There are four different power supply configurations available. A LV circuit can be expanded with up to two HV circuits. The batteries are modeled as a chen model with two RC-circuits and a single resistance that are connected in series.
If the power supply is not of interest, it can also be declared as not specified.

Exercise

- Beat the clock:
  - Keep the rear driven transmission configuration (Generic > Rear drive).
  - Increase the engine maximal torque up to 65 Nm and smooth the curve.
  - Reduce the final transmission ratio to 3.5 (in "Differential Rear > Transmission") and increase the driven Torque Bias to 3 (in "Differential Rear > Torque Bias Ratio Driven").
  - Start a TestRun (e.g. SlalomBrake) and compare the pace of FS_RaceCar_6.0 to your parameterized one.

5.3 Powertrain: Electric Race Car

5.3.1 General

The powertrain model in CarMaker gives the possibility to create an electric powertrain with one to four electric motors. The best way to start the parameterization of your electric powertrain in the CarMaker GUI is to adapt the FS_RaceCar_6.0_electric*" example models as they come with all elements required for a FSE race car.

The project folder FS_Generic_2018 contains two example vehicles with an electric powertrain model in CarMaker. The data set "FS_RaceCar_6.0_electricRWD" is rear wheel driven and vehicle called "FS_RaceCar_6.0_electricAWD" comes with an all wheel drive. The selected settings for the electric powertrain are described in the following.

If you want to create an own electric powertrain from scratch, we highly recommend to carefully read the Powertrain section in the CarMaker Reference Manual (accessible via CarMaker main GUI > Help > Reference Manual). In this case, you should start with selecting a predefined template with the "Predefined" button in the powertrain dialog. For a pure electric powertrain, select "Electrical" and the number of electric motors. The number of drive sources corresponds to the number of electric motors and the parameterization of the control units is adapted to an electric powertrain.

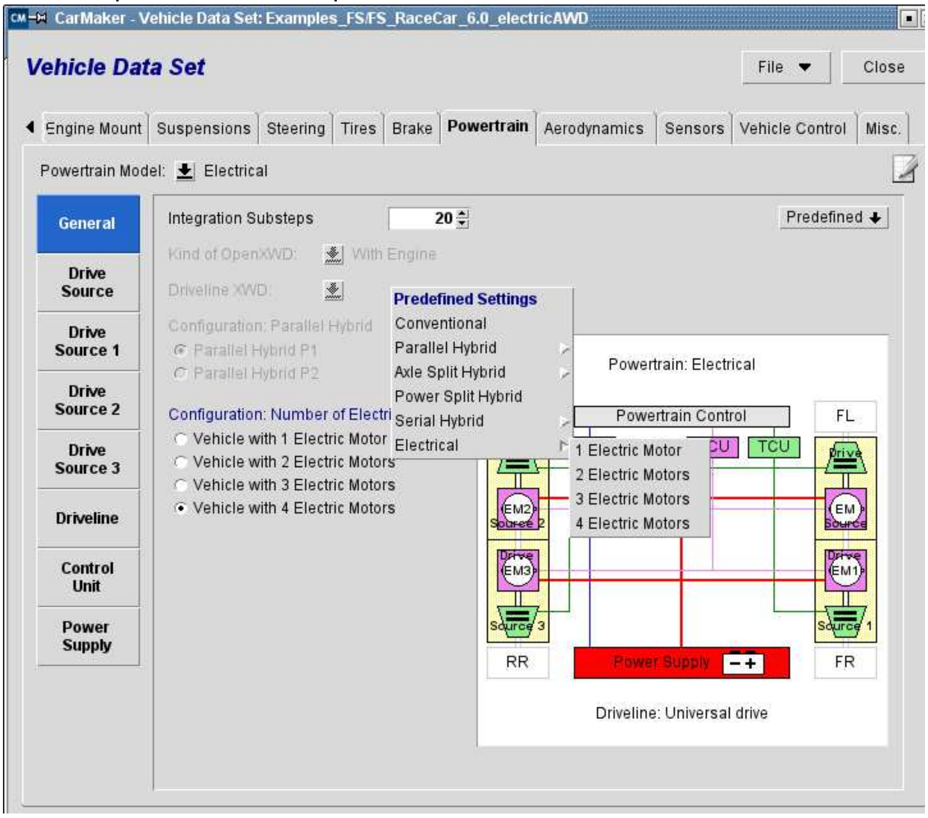


Figure 5.22: Selecting a AWD electric powertrain template with right-click

5.3.2 Drive Sources

Each electric motor is parameterized as one so called "Drive Source". Besides the characteristics of the motor, each Drive Source contains a gearbox and a clutch model. The gearbox and the clutch are modeled the same way as for a combustion engine described in section "Drive Source", pg. 94. If the powertrain does not include a gearbox and a clutch for each motor, but has only one fixed transmission as it is common in FS cars, it is possible to define only one forward gear in the gearbox (without reverse gear!).

The specification of only one forward gear has a special treatment in CarMaker: The clutch is disabled internally and CarMaker considers a rigid connection between the motor shaft and the gearbox.

The electric motor requires some general information like inertia or the voltage level, which describes to the circuit used for the motor (high voltage or low voltage).

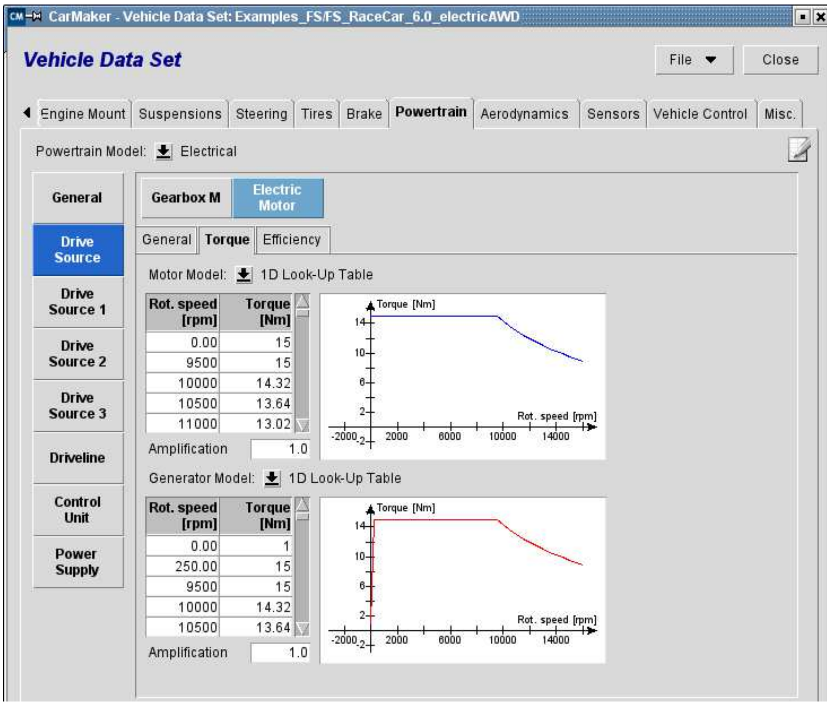


Figure 5.23: Powertrain - Electric Motor Mapping via look-up table

There are two ways to define the torque mapping. The maximum torque curve can be either parameterized by characteristic values or as a look-up table as shown in Figure 5.23. The meaning of this characteristic values is shown in Figure 5.24.

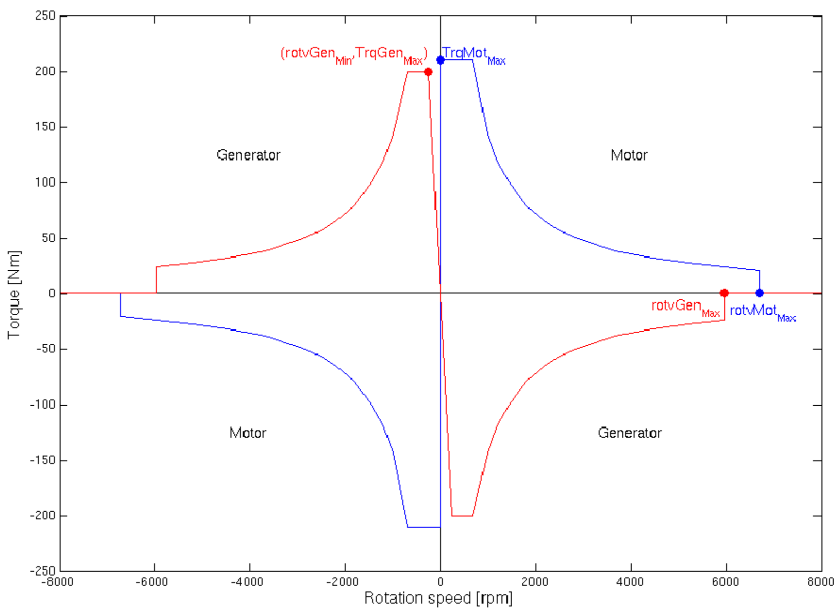


Figure 5.24: Powertrain - Electric Motor Mapping via characteristic values

In order to prevent an unstable situation when switching from motor mode to generator mode, the generator torque curve should start at the point 0 rpm/0 Nm.

Both, the motor and generator characteristics should be specified with absolute values. CarMaker internally handles the sign.

Finally, each motor's efficiency can be defined as a single value or a look-up table in the "Efficiency" tab.

5.3.3 Driveline

There are several Driveline models available in CarMaker. For an electric powertrain with more then one drive source, the Driveline model "Universal drive" should be selected.

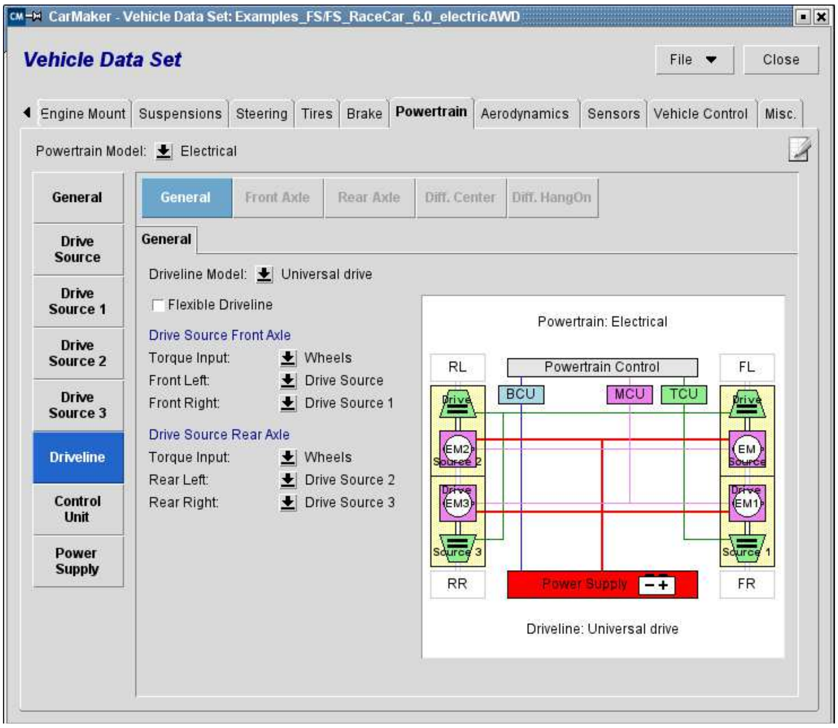


Figure 5.25: Powertrain - Universal drive

As shown in Figure 5.25, the "Universal drive" model provides several options to apply the drive source torque at:

Torque Input: Wheels The Drive Source is directly connected to the wheels, the driving torque is supported by the wheel carrier. This is the recommended mode for Formula Student Electric cars.

Torque Input: Driveshafts The Drive Source is connected to the wheel by a shaft, the riving torque is supported by the vehicle body.

Torque Input: Differential The Drive Source is connected to the axle's differential.

Furthermore, the different Drive Sources (electric motors) can be assigned to a wheel.

5.3.4 Control Units

Control Units are responsible to manage the Drive Sources and the Power Supply. There is only one control unit for all components of the same kind:

- Motor Control Unit (MCU) --> electric motors
- Transmission Control Unit (TCU) --> gearboxes and clutches
- Battery Control Unit (BCU) --> batteries of power supply

The Control Units have two main tasks: The evaluation of the current state of the component and the regulation of the electro mechanic components in order to reach the target values (e.g. target torque for motor) provided by PTControl.

The PT Control unit is the superordinated powertrain control strategy.

The powertrain model "Electrical" comes with a typical control strategy for an electric powertrain. This operation strategy is described in the following. If further functions are required, it is also possible to replace the control strategy by a model extension via Matlab/ Simulink, an FMU or C-code.

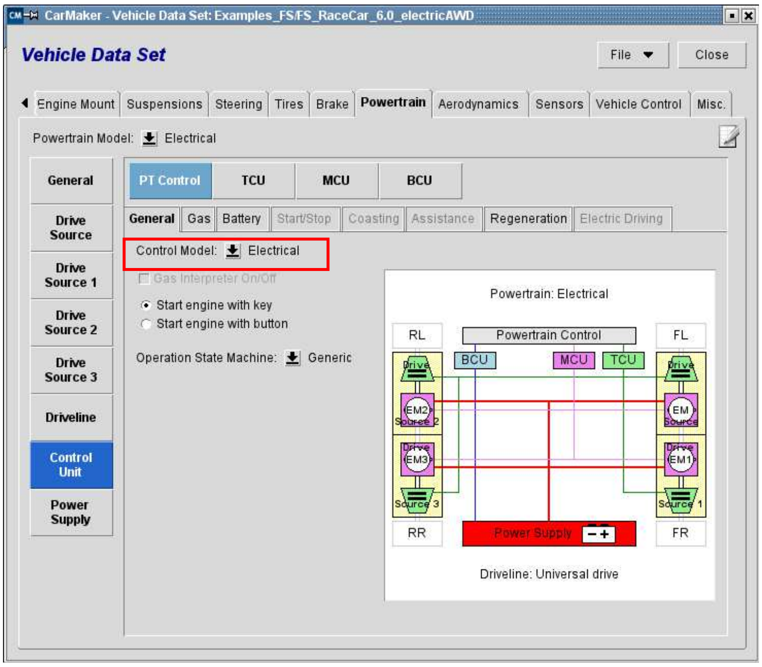


Figure 5.26: Control Units in the Powertrain dialog

PT Control

The PT Control is the main control model of the powertrain. It manages the vehicle operation machine, the battery, the interpretation of the gas pedal and the regeneration.

In case of an electric vehicle, you should always use the control model "Electrical". It automatically deactivates the PT Control features, which are not needed for an electric drivetrain.

General The battery management in tab "General" controls the batteries state of charge by assigning torques to the electric motors and by managing the energy transfer between the electric circuits. It can be deactivated if not needed and is an optional feature. The "Operation State Machine" handles the vehicle operation state based on pedal and key position/start-stop-button activation. IPGDriver is adapted to the "Generic" Operation State Machine and thus this setting should be left untouched. For detailed information on the controller parameters, take a look at the Reference Manual at chapter "PTControl".

Gas The interpretation of the gas pedal position is defined here. It can either be linear relation defined by characteristic values or nonlinear with a look-up table.

Regeneration This tab provides the two most common operation strategy models for electric vehicles: Regenerative braking and regenerative drag. For both strategy modes, the limits for entering or leaving the strategy mode must be defined. The meaning of this limits are shown in Figure 5.27 and Figure 5.28. The regenerative brake mode itself is defined in the brake model (section "Brake", pg. 87).

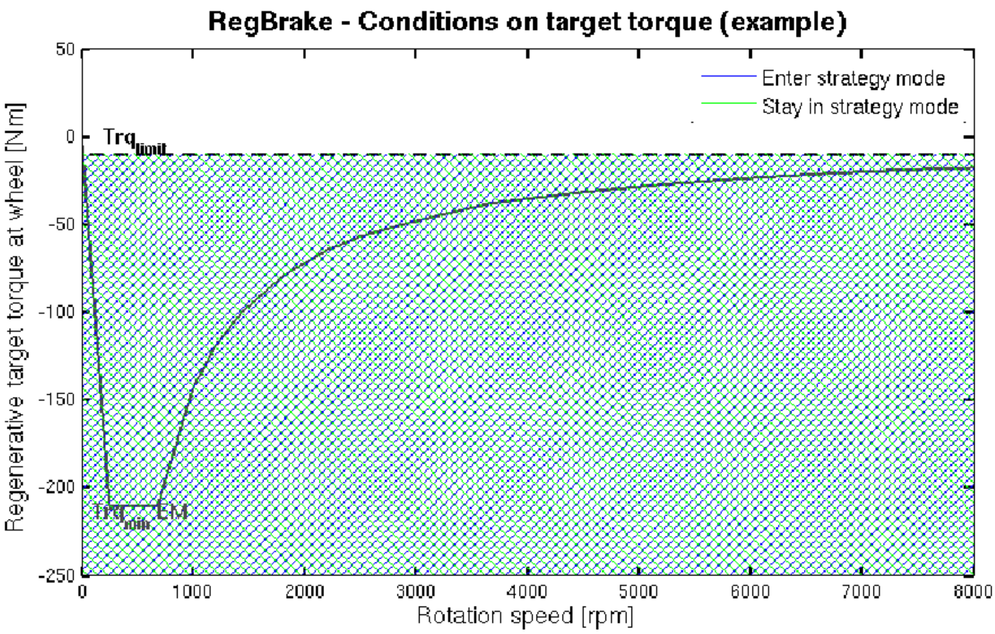


Figure 5.27: Powertrain - Conditions on target torque

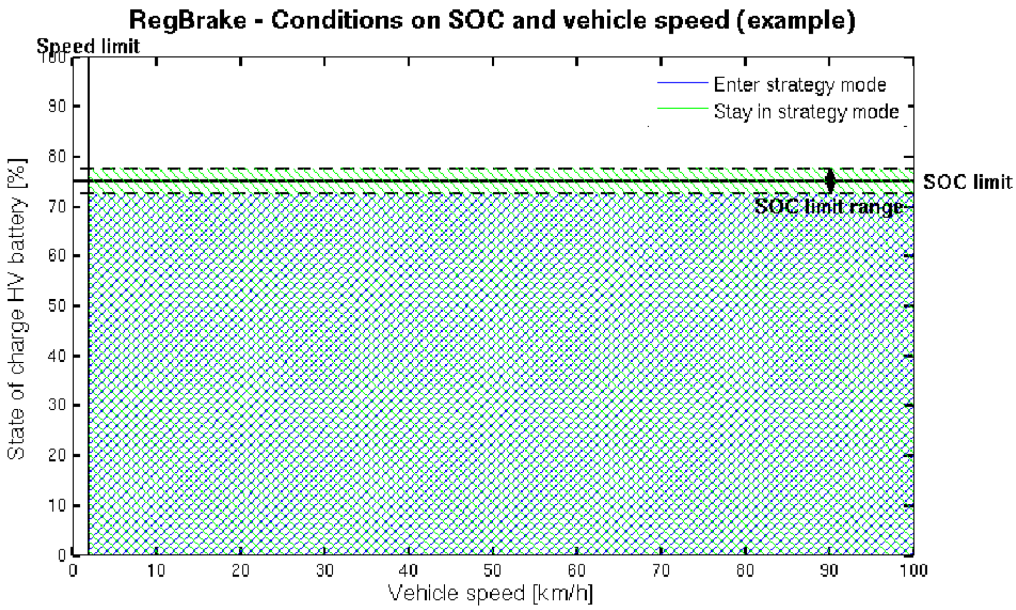


Figure 5.28: Powertrain - Conditions on SOC and vehicle speed

Transmission Control Unit (TCU)

The TCU functionality is similar to the explanation for the combustion engine in section "Control Unit", pg. 97. If there is only one static gear defined in the gearbox, the TCU can be deactivated by selecting "-not specified-".

Motor Control Unit (MCU)

The MCU model "Basic" sets each motor's load and determines its maximum motor and generator torque at the actual rotation speed. For more information on the P/I values of the torque and speed controller, please take a look at the Reference Manual at chapter "Control Units > Engine Control Unit (ECU)".

Battery Control Unit (BCU)

The BCU calculates the state of charge (SOC) and state of health (SOH) of the batteries. It tries to keep the batteries state of charge at the target value by applying additional generator torques to electric motors if engine is on or by controlling the energy transfer between low voltage and high voltage 1 electric circuit. The energy can be transferred between a maximum of three circuits: A low voltage circuit and up to two high voltage circuits.
This control unit is optional. Choose "BCU Model: Low Voltage + High Voltage 1" for your race car.

5.3.5 Power Supply

The main task of the power supply model is the accounting of the electrical power flow to and between the electric circuits. It contains the auxiliary consumers and optionally converters between the low voltage (LV) and high voltage (HV) circuits and the battery models.
Since most Formula Student cars do not use a converter, it can be deactivated by entering 0 kW as the maximum power.

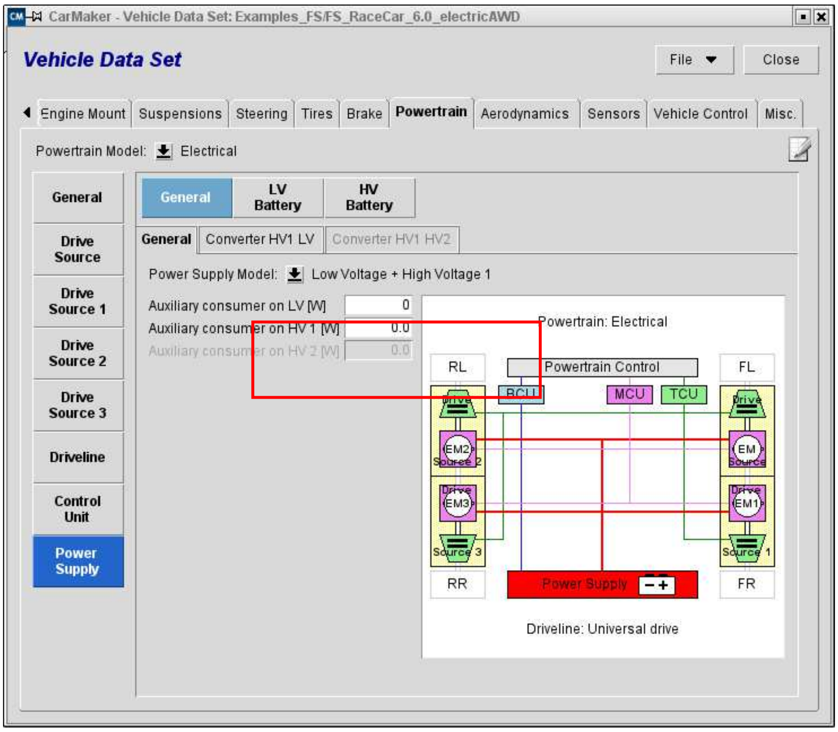


Figure 5.29: Settings of the Power Supply module

Battery Model

The Battery model is based on an electric battery model combined with a characteristic curve to introduce the influence of state of charge (SOC) on the idle voltage. The HV and LV batteries are modeled with a Chen model. It consists of two RC-circuits and a single resistance that are connected in series as shown in Figure 5.30.

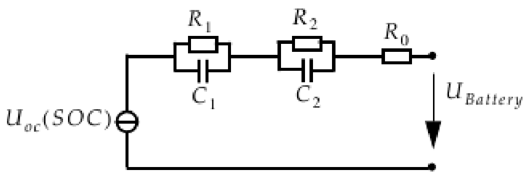


Figure 5.30: Powertrain - Chen Model

The dependency of an ideal voltage source on the batterie’s SOC can be parameterized via a look-up table. The battery’s SOC is not allowed to drop below the minimum state of charge or to exceed the maximum state of charge

You can find more information on the battery model in the Reference Model in chapter "Power Supply > Battery".

5.4 Powertrain: "OpenXWD" Model

5.4.1 Prolog

To get started with CarMaker and Simulink you should read the section "CarMaker for Simulink" in our Quick Start Guide (to be found in the CarMaker main GUI under *Help*) beforehand.
Do not forget to choose "src_cm4sl" from the project folder "FS_Generic_2018" as your "Current Directory" in MATLAB. There you find some files which are related to the FSE race car example:

FSE_OpenXWD_StandAlone.mdl The whole generic CarMaker powertrain model is replaced. Instead the powertrain kind 'OpenXWD Stand Alone' is activated. The driving torque is generated by the Simulink model which includes two electric motors and a traction control.

FSE_Parameter.m The m-file initializes all the needed parameters and characteristics for all Simulink models.

FSE_Define_Motor Characteristic.xls This is an Excel sheet (version 2003 or newer) to generate motor characteristics.

FSE_M85.txt This text file includes a motor characteristic with a maximum torque of 85 Nm.

FSE_M100.txt It is a motor with a maximum torque of 100 Nm.

To run the Simulink model, the OpenXWD powertrain model need to be chosen in the vehicle data set. The example vehicle "FS_RaceCar_XWD_6.0" already includes the necessary settings.

To start CarMaker for Simulink please make sure that the MATLAB script "cmenv.m" was run successfully. It adds several paths to the current MATLAB session, which give access to the CarMaker toolbox. In case you did not install CarMaker in the default installation folder C:\IPG, the execution of the cmenv.m script will result in an error. Please adapt the CarMaker installation path in the cmenv.m script. In line 6 the path to your IPG installation folder needs to be specified.

```
function cmenv (varargin)         % -*- Mode: Fundamental -*-
% CMENV - Add CarMaker directories to the MATLAB search path.
%

    % CarMaker installation directory.
    if isempty(which('cmlocaldir'))
        cminstdir = 'C:\IPG\hil\win32-6.0.2';
    else
        cminstdir = cmlocaldir; % for mat: CM-6.0
    end

    disp(['CarMaker directory: ', cminstdir]);
    if ~exist(cminstdir, 'dir')
        error('Unable to find specified CarMaker installation directory.');
    end

%    if is_64_bit_version
%        unsupported_version_error(cminstdir);
%    end
```

Figure 5.31: This path has to be set to the correct path

Exercise - Step 1

- Prepare for CarMaker for Simulink:
  Read the section "CarMaker for Simulink" in CarMaker's Quickstart guide ("Help > Quickstart Guide"). Open MathWorks MATLAB and change your current directory to "<pathto>/FS_Generic_2018/src_cm4sl".
  Check "cmenv.m" path and execute the script.

5.4.2 OpenXWD Example

The "FS_Generic_2018" project folder includes a Simulink model for the Formula Student Electric competition ("FSE_OpenXWD_StandAlone.mdl"). It uses the OpenXWD interface and has two electric motors at the rear axle combined with a traction control.

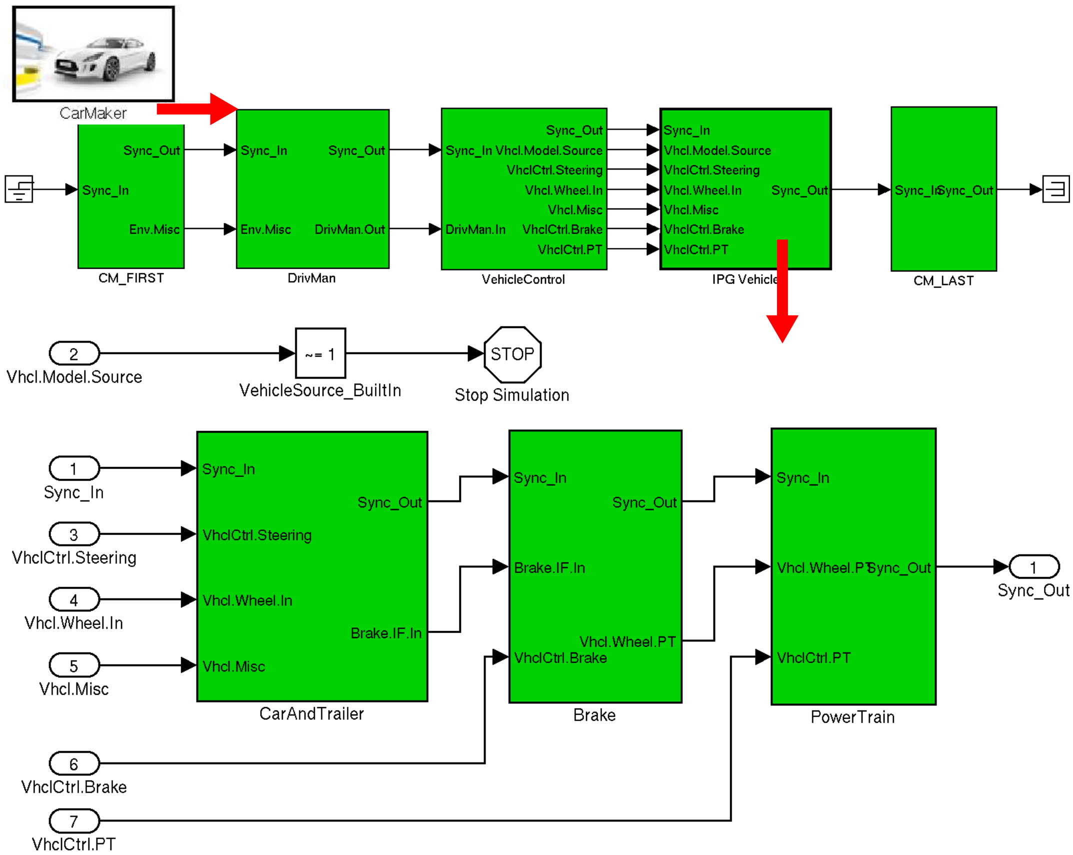


Figure 5.32: Simulink with OpenXWD StandAlone Model

With the OpenXWD interface you are very flexible in designing your own powertrain model, because every pre-defined component (e.g. clutch, gearbox, differential...) is deactivated. The basic concept of this model is actually pretty simple: CarMaker expects a drive torque at the wheels. It is up to the user's model extension, how this torque is generated. The wheel integration is done on CarMaker side and it returns the current wheel speed.

To activate the OpenXWD driveline model, select the "Powertrain Model: OpenXWD" in the vehicle data set editor (see Figure 5.33) or select our example vehicle data set "Examples_FS/FS_RaceCar_XWD_6.0". Without any model extension, the vehicle will not drive in this configuration. The interface variables required to transmit a drive torque from your model extension to CarMaker are called "PT.W<position>.Trq_Drive". For this, the Simulink model extension is used.

Exercise - Step 2

- TestRun with MATLAB:
  Open the "FSE_OpenXWD_StandAlone.mdl" in MATLAB-Simulink. Double-click the to open "CarMaker for Simulink". In "CarMaker for Simulink" - GUI, create a new TestRun, add the "Examples_FS/FS_RaceCar_XWD_6.0" race car to the TestRun.

5.4.3 Powertrain data set based on "OpenXWD" model

With the powertrain model "OpenXWD" and the option "With Engine" or "With Motor" it is nearly the same like "Generic", but the connection from the gearbox output to the wheels is cut and has to be modeled by the user (e.g. in Simulink). The option "Stand Alone" deactivates every component of the CarMaker powertrain. All required quantities and parameters (from generating torque right up transmitting it to the wheels) need to be calculated by an external program (e.g. Simulink).

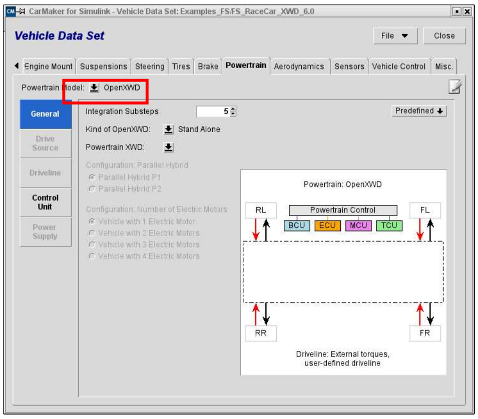


Figure 5.33: Vehicle Data Set > Powertrain > OpenXWD > Stand Alone

More detailed information about the powertrain models can be found in the Reference Manual, section "Parameterization: Vehicle model > Powertrain".

5.4.4 General Remarks to the Simulink Models

Generally you can manipulate or overwrite signal in CarMaker from Simulink. With using the pre-defined CarMaker interface blockset (available in the "CarMaker4SL" library) you have easy access to manipulate/overwrite quantities. Thereto use the "Read CM Dict" block to read the current value and the "Write CM Dict" block to overwrite the variable you like. But remember, when you overwrite a quantity calculated in CarMaker (also called UAQ, see section "Data Access", pg. 41) you influence the variable in the whole CarMaker and MATLAB/Simulink workspace. To get an overview about the quantities look at the Reference Manual, chapter "User Accessible Quantities".
In addition, you can also create your own variables. Thereto use the "Define CM Dict" block. For integrating your self-defined quantities in a Simulink model, please use the "Read/Write CM Dict" blocks again. More information can be found in the Programmer's Guide, section "CarMaker for Simulink > The CarMaker Interface Blockset".

Figure 5.34: Simulink Dictionary

Additional Parameters in the Vehicle data set

Please note, that the Open_XWD prowertrain model only replaces the electrical and mechanical components of the drivetrain. It still uses the preimplemented powertrain controller models (see section 5.2.4).

To make an Open_XWD model run, you first need to select a suitable PTControl unit. For an FSE race car we recommend the PTControl unit "Electrical". This unit needs some information about the powertrain model. This parameters can be provided in the vehicle data set under "Misc > Additional Parameters". In our example we want to avoid any interference by PTControl unit which is why we reduced the information provided to the PTC. The minimum set of required parameters is defined in the example "FS_RaceCar_XWD_6.0:

Table 5.4: Additional Paramaters for PTControl with Open XWD model

Parameter	Value	Description
PowerTrain.PTKind	BEV	Definition of the powertrain kind (electric)
PowerTrain.nGearBoxM	1	Number of gearboxes used
PowerTrain.nMotor	1	Number of electric motors used
PowerTrain.Gear-BoxM.ClKind	Closed	No gearbox used
PowerTrain.Gear-BoxM.GBKind	NoGearBox	Only fix transmission (no shifting by IPGDriver)
PowerTrain.Gear-BoxM.iBackwardGears	1	Transmission ration of reverse gear
PowerTrain.Gear-BoxM.iForwardGears	01	Transmission ration of forward gears
PowerTrain.Motor.Gen.Trq-Map	0 0 0 0	Torque map of electric motor (set to zero as defined in the Simulink model separately)

For further information about the parameters please check "Powertrain > Powertrain models > Powertrain model "OpenXWD" in the Reference Manual ("MainGUI > Help > Reference Manual").

5.4.5 OpenXWD:Powertrain in Simulink

In comparison to the "Generic.mdl" the "FSE_OpenXWD_StandAlone.mdl" has a modified powertrain with two terminators, which kill "PT.OperationState" and "PT.Engine.rotv". With the blue submodel a way to generate those UAQ is provided. Note that there is a saturation block which only allows a range from 100 - 16000 rpm at the engine output shaft.

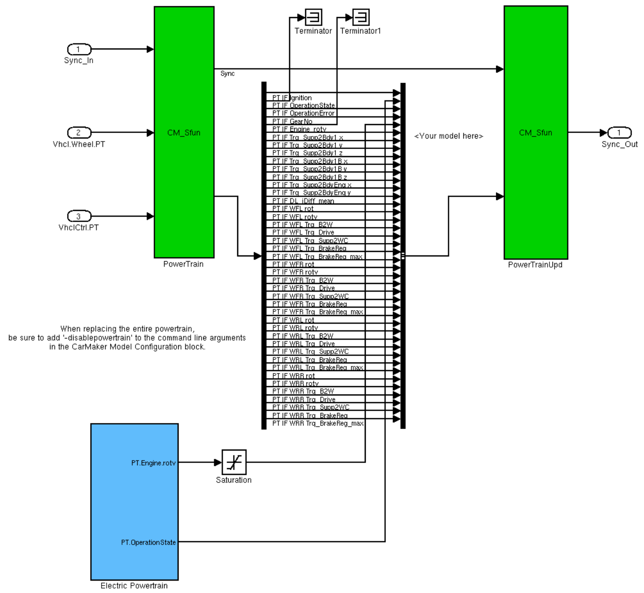


Figure 5.35: Overview of "IPGVehicle > Powertrain" of "FSE_OpenXWD_StandAlone.mdl"

An inside look in the blue submodel shows a predefined possibilty to generate the required UAQ's.
I

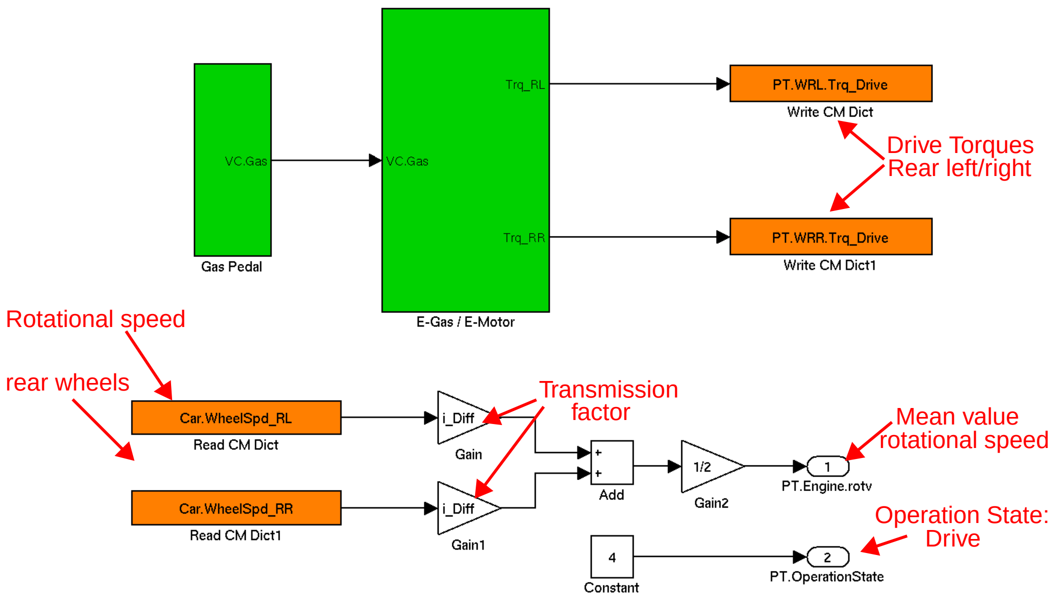


Figure 5.36: Simulink Powertrain of FSE_OpenXWD_Standalone model

Gas Pedal

The function of this subsystem is both to initialize the signal "Vhcl.Ignition" (enables to switch off the ignition during simulation) and to read out gas pedal position for the motor characteristic.

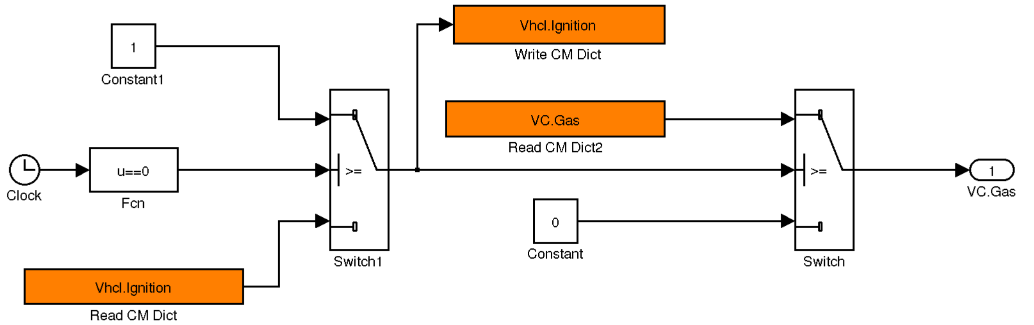


Figure 5.37: Electric Powertrain > Gas Pedal

E-Gas / E-Motor

Firstly, this block includes with the "E-Motor Mapping" the heart of the electric powertrain. There you can load your own motor characteristics (more details in section "Manipulating the OpenXWD Example Model", pg. 117) into the system. The PT1 transfer functions should give the system a plausible step response. Here you can also change the fixed transmission factor but do not forget to adapt these factors in every block.

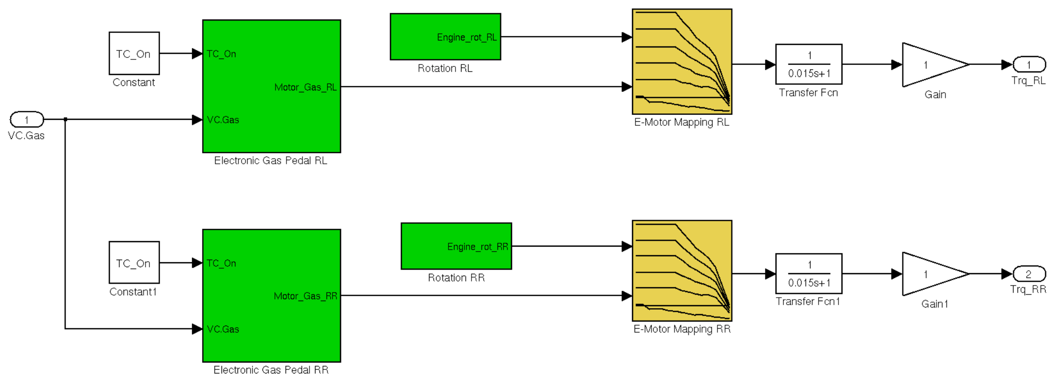


Figure 5.38: Electric Powertrain > E-Gas / E-Motor model

Secondly, the calculation of the rotational speed (blocks: Rotation RL/RR) from each of the motor shafts is done. The calculation includes a transmission factor and a unit change from revolutions per second in rpm.

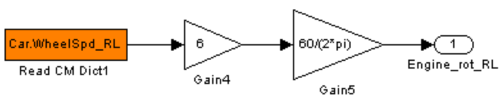


Figure 5.39: Electric Powertrain > E-Gas / E-Motor model > Rotation <Pos>: Calculation of engine speed

Lastly, you will find the "Electronic Gas Pedal" block. This subsystem determines whether the gas pedal value from the driver or a control system (e.g. traction control) will be used. It can serve as an interface for your own control system by simply exchanging the model.

The model includes a very simple traction control for each of the driven wheels. When the variable "TC_On" equals one in the parameter file (FSE_Parameters.m) the traction control is activated, it can be deactivated by entering zero.
The traction control (short: TC) monitors the current slip in longitudinal direction and the gas pedal values. When the slip increases too fast the TC sets the electronic gas pedal into a waiting position (mode 1). When in the following time the slip oversteps a threshold, mode 2 will be activated and the gas pedal will automatically be decreased (linear).
In mode 0 the TC simply decides whether an increasing gas pedal value comes from the driver or from a linear curve to accelerate the car.

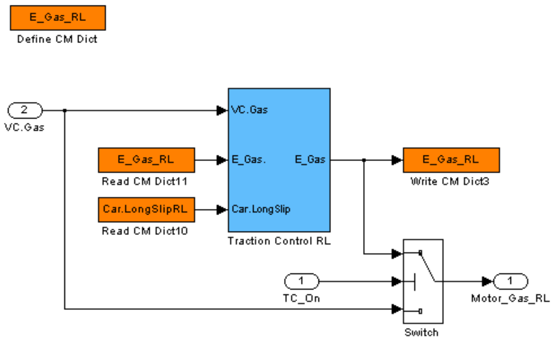


Figure 5.40: Simulink Electronic Gas Pedal model

5.4.6 Parameter File

The parameter file "FSE_Parameters.m" should be loaded into the MATLAB workspace once before you start a simulation. On the one hand the file initializes the necessary motor characteristics and on the other hand the parameters for the traction control (OpenXWD model) are defined. It also contains the parameters for the recuperation- and torque vectoring model.

Table 5.5: Parameters in FSE_Parameters.m

Parameter	Unit	Description
i_Diff	[-]	factor for a mechanical transmission
TC_On	[-]	turn traction control on or off by setting this parameter to 1 or 0
TC_Dericative	[-]	treshold traction control mode 1
TC_Relay_On	[-]	maximum allowed longslip
TC_Relay_Off	[-]	minimum allowed longslip
TC_Incr_Gas	[-]	increasing electronic gas pedal (mode 0]
TC_Decr_Gas	[-]	decreasing electronic gas pedal (mode 2)
TV_On	[-]	turn torque vectoring on of off by setting this parameter to 1 or 0
TV_Override	[-]	increase the torque applied to outer wheels in corners
eta_Gen	[-]	efficiency of the electric motor in generator mode.
Gen_Loss	[-]	braking torque loss due to time needed to build up strator load

Table 5.5: Parameters in FSE_Parameters.m

Parameter	Unit	Description
Brake_Pedal_Travel	[-]	defines how much of the brake pedal travel is allowed for pure recuperational braking
Brake_Mech_Rec_Ratio	[-]	defines how much of the braking torque is recuperational after reaching "Brake_Pedal_Travel"
SOC_Init	[%]	initial state of charge
C_Batt	[kWh]	battery’s capacity
U0_Batt	[V]	No-load voltage of battery
R0_Batt	[Ohm]	resistance of battery
C1_Batt	[F]	capacity
R1_Batt	[Ohm]	resistance
C2_Batt	[F]	capacity
R2_Batt	[Ohm]	resistance

Every change in the file requires a reload of the complete parameter file to the workspace.

5.5 Adaption of the Example Models

5.5.1 User Defined Powertrain Control Models

In case you are using the pre-implemented electric powertrain model in CarMaker as explained in section "Drive Sources", pg. 99, the control strategy for PT Control can be replaced by a user defined control strategy. CarMaker offers different ways to implement user models such as the C-code interface, FMUs or MATLAB/Simulink. As the latter is the most popular interface among most Formula Student teams, we will explain this approach in the following.

The FSE_TorqueVectoring.mdl in the FS_Generic_2018 project folder is an example for a self-developed controller containing a traction control and a yaw control.
In contrast to the OpenXWD example presented in section "OpenXWD Example", pg. 107, this model only replaces the PTControl unit. Therefore, it is not necessary to model the entire powertrain such as the motor or the battery model. As shown in Figure 5.41, the controller calculates a load for each motor, which is passed on to the MCU.

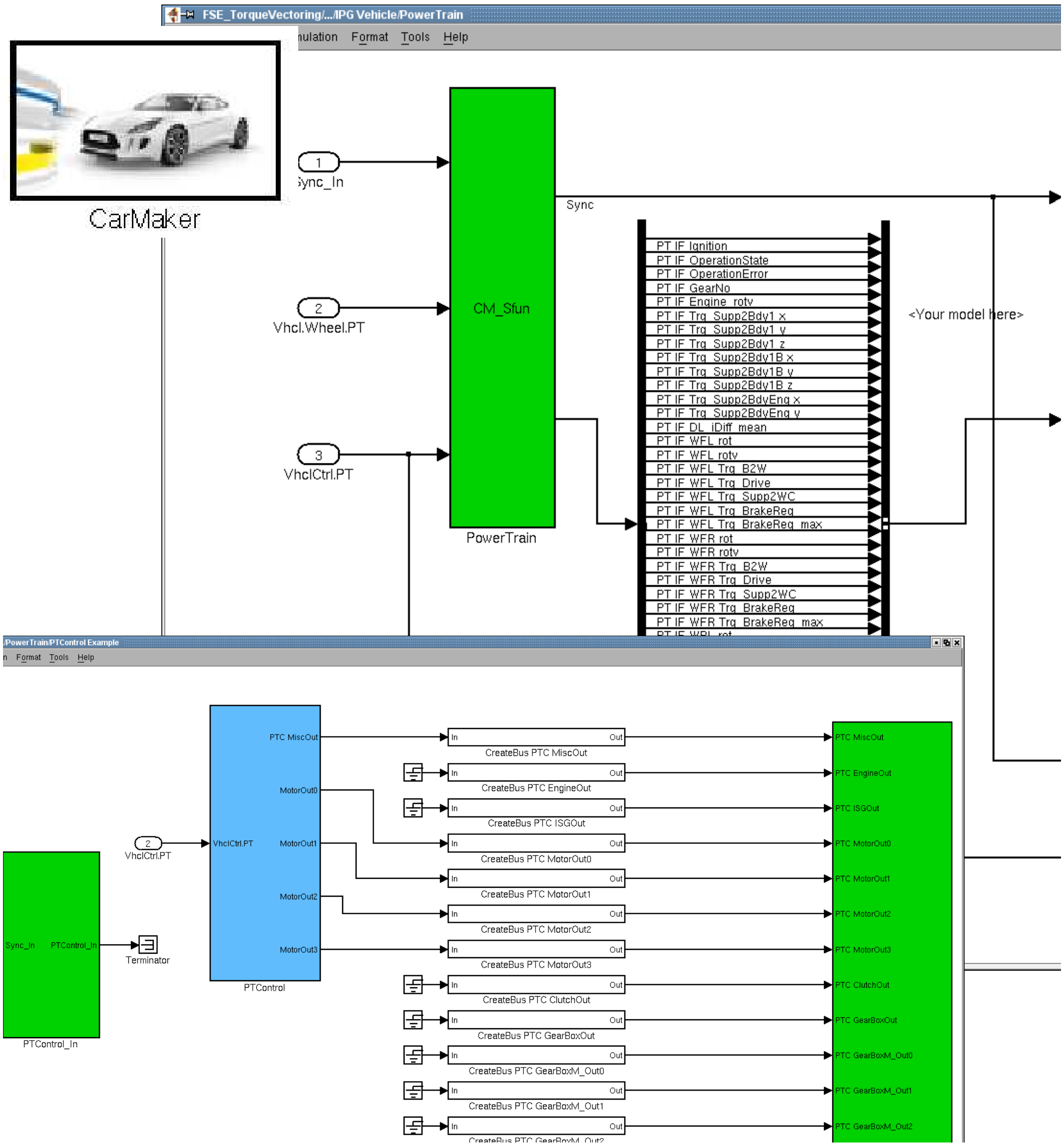


Figure 5.41: PTControl - Simulink interface

There are four torque vectoring controller, one for teach wheel. In this example of a Torque Vectoring model a higher torque is applied on the outer wheel in a corner. The yaw momentum is increased and the vehicle gets more agile.

In this model the maximum possible yaw rate is calculated by multiplying the friction coefficient with the gravity coefficient. This value is then divided by the current velocity of the car. The difference between actual and maximum yaw rate is the potential for a torque incrementent on the outer wheel in a corner. The parameter is then added up to the motor gas input.

With the TV_Override parameter (in FSE_Parameters.m) you are able to artificially increase the maximum yaw momentum in order to increase the parameter added up to the motor gas. But this can lead to an unstable cornering behavior depending on the amount.

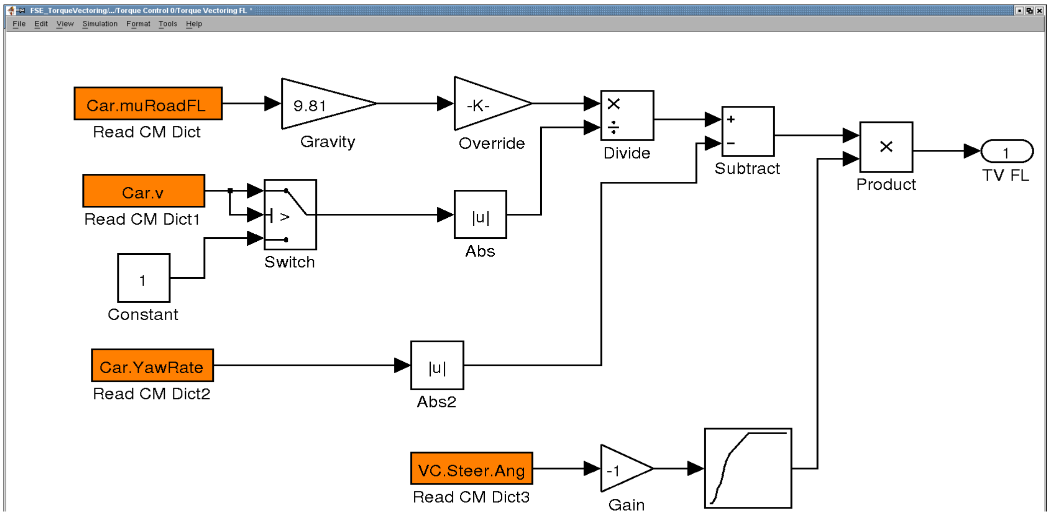


Figure 5.42: Torque Vectoring parameter calculation for the rear left wheel

In order to replace the PT Control model provided by CarMaker with a MATLAB/Simulink model, CarMaker for Simulink has to be started as explained above. Just select the example vehicle "Examples_FS/FS_RaceCar_6.0_electric_TorqueVectoring". With any other vehicle data set, "CM4SL User Model" must be selected as the control model in the PTControl tab.

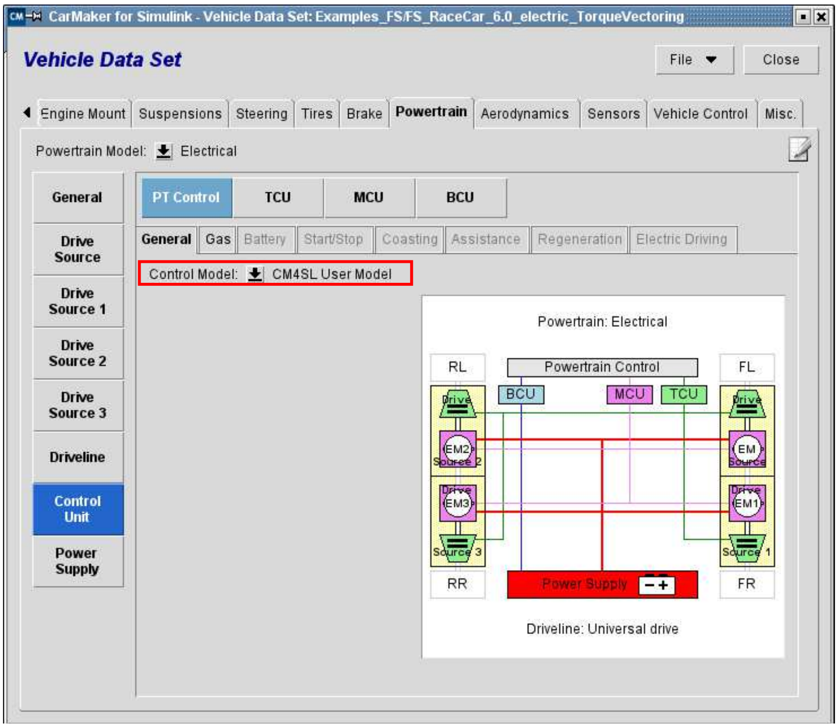


Figure 5.43: Activating a Simulink based PT Control strategy

5.5.2 Manipulating the OpenXWD Example Model

This chapter explains how to implement your own motor characteristics in the OpenXWD example model. The file "FSE_Define_MotorCharacteristic.xls" is an Excel sheet to define motor characteristics. Only the green highlighted fields should be adjusted.

1

% rotational speed in rpm

99999	0	500	1000	1500	2000	2500	3000	3500	4000	4500	5000	5500	6000	7000

% gas pedal | motor torque in Nm

0	0	0	-5	-5	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14
0.2	17	[illegible]	17	17	17	17	17	15	14	12	11	10	8	-8
0.4	34	[illegible]	34	34	34	34	34	30	28	24	22	20	16	-6
0.6	[illegible]	51	51	51	51	51	51	46	41	37	33	29	24	-4
0.8	[illegible]	68	68	68	68	68	68	61	55	49	44	39	32	-2
1	85	85	85	85	85	85	85	76	69	61	55	49	40	0

2

3

Figure 5.44: Table Motor Characteristic
Box 1: Rotational speed range
Box 2: Drag torque
Box 3: Full load

Exercise

- Please do the following:
    - Enter the rotational speed range (in 1/min) in box 1
    - Enter the drag torque curve (in Nm) in box 2
    - Enter the full load curve (in Nm) in box 3
    - Check your entries with the torque-speed-diagram
    - Save the file as a text file (tap-stop-separated) and ignore the warnings, e.g. FSE_M120.txt
- To initialize the model with your motor characteristic please load the generated text file with the command
    - load('FSE_M120.txt')

  into the MATLAB workspace (you also can extend the parameter file!). Do not forget to change the matrix names in the 2D-Look-up-Table for the "E-Motor Mapping" (e.g. to FSE_M120).

5.5.3 Driver Model

It is highly recommended to use the User parameterized Driver when simulating with FS cars. The simulations are reproducible and the driver can be adapted to needs of each discipline.

However, if there is the need to simulate with the Racing Driver there are several things which have to be taken into account to successfully perform a Driver Adaption with a FSE Powertrain Model. The following steps have to be completed:

Exercise - Step 1:

- Teach the driver:
  After opening one of the Simulink models out of MATLAB and opening the CarMaker GUI, a TestRun can be created or loaded.
- You can now start a Basic Knowledge Driver Adaption. After completing the adaption and starting the simulation, you might encounter that the vehicle drives very slowly along the defined course. In this case continue with Step 2.

Exercise - Step 2:

- Fix it if slow:
  To fix this, open the ASCII-file of your TestRun with a text editor (e.g. Notepad++, Kate). Almost at the end of the file there is a line called "Driver.Learn.vIdle" which defines the vehicle’s idle velocity. **Set there a value greater than 0**.
  Save the text file.

```
Driver.Learn.nEngine.Standard:
     0 0
     0 0
     0 0
     0 0
     0 0
Driver.Learn.vIdle = 1.000
Driver.Learn.vMax = 103.956
Driver.Learn.vG2nEng025 = 25.989
```

Listing 5.1: Excerpt of an ASCII-file of a TestRun

Exercise - Step 3.1:

- Driving with the User parameterized Driver:
  There is also the possibility to select the User parameterized Driver instead of the Race Driver, which is also recommended. Load the parameter set of an aggressive driver with right-click anywhere in the window.

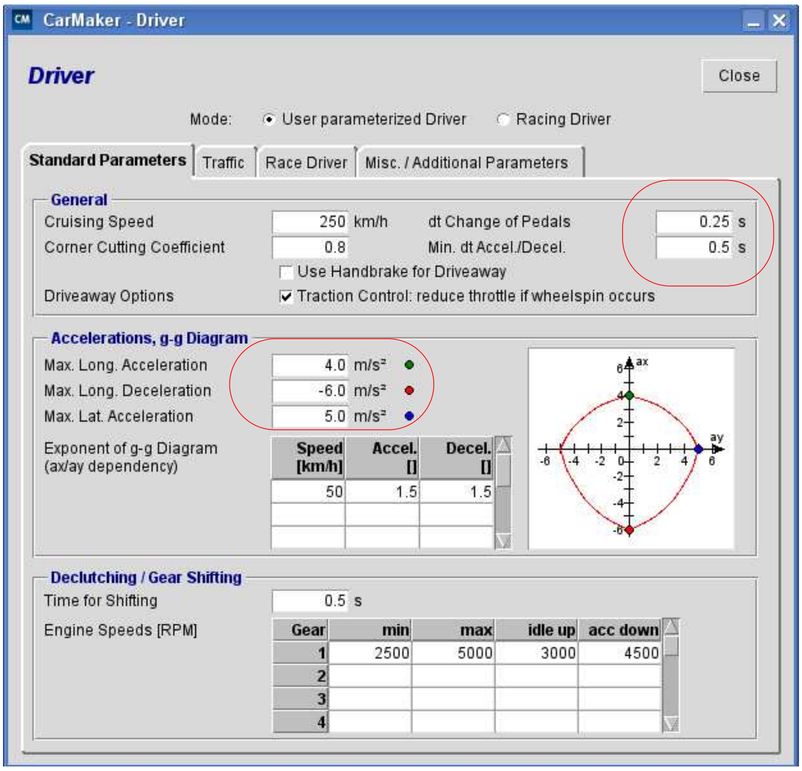


Figure 5.45: User parameterized Driver parameters

First it is recommended to adjust the parameters in the *General* area:

- d*t. Change of Pedals* to a lower value
- *Min. dt Accel/Decel* = 0.
- *Accelerations, g-g-Diagram* area double the values.

- You can now start your TestRun. If the vehicle leaves the road or rolls over, decrease the Acceleration parameters step by step.

Exercise - Step 3.2:

- Driving with the Race Driver:

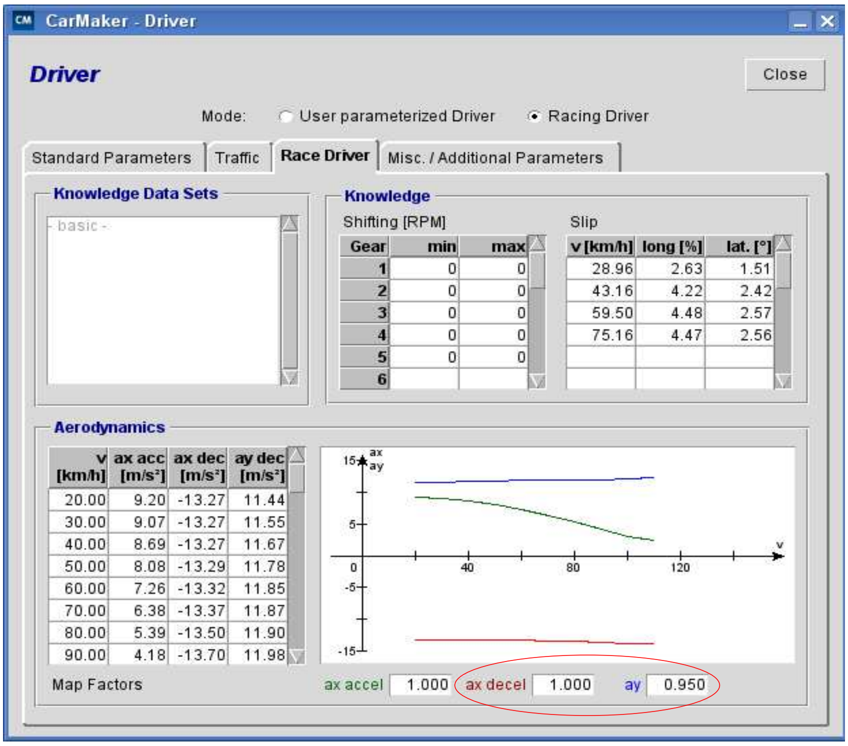


Figure 5.46: Racing Driver Parameters

If you select the Race Driver after completing a Basic Knowledge Driver Adaption and completing step 1, in some cases you will encounter that the vehicle eventually leaves the road or rolls over. But there is a way to avoid this: Decrease the acceleration factors *ax deccel* and *ay* step by step and observe the results. Optimize these factors so that the vehicle stays on track, but still accomplishes an acceptable lap time.

Exercise - Step 4:

- Perform a TestRun with TC:
  Load the parameter file "FSE_Parameters.m" into the MATLAB workspace.
  Then start the "FSE_OpenXWD_StandAlone_2Motors_TC.mdl" Simulink file and open the CarMaker GUI by double clicking on the "Open GUI" icon. Choose the TestRun "FSE_Acceleration_XWD_TC" via "File > Open" the. This TestRun is similar to the FS Acceleration competition.
- Perform a simulation and monitor the following quantities with IPGControl:
  - Car.LongSlipRL
  - DM.TriggerPoint.Time
  - VC.Gas
  - PT.WRL.Trq_Ext2W
- Perform a TestRun without TC:
  Now switch off the traction control. Thereto open the parameter file and set the variable "TC_On" to zero. Re-load the file into the workspace, perform a second TestRun and compare the results.

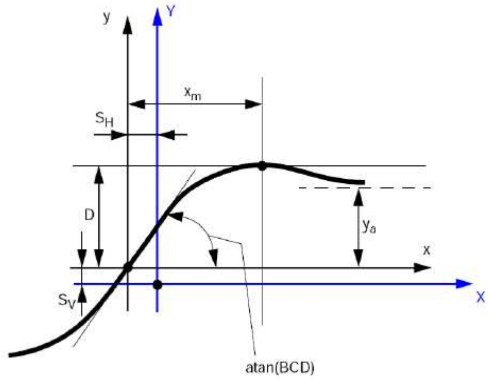
5.6 Creating a Tire Dataset Using IPGTire

The tire is one of the most important components of a vehicle. All forces and torques are transferred from the road to the car through the tire. Hence, the tires provide the basis for all lateral and longitudinal dynamics of a vehicle. Thus, the tire model used in a simulation must be a detailed one. Therefore, CarMaker offers different opportunities:

- Pacejka Magic Formula
- IPGTire
- TameTire
- Tire Data Set Generator

5.6.1 Pacejka Magic Formula

The Pacejka Model is based on a complex mathematical formula. Its parameters, called "Magic Parameters", do not have any physical meaning. They are calculated with suitable programs using test readings. If you'd like to have more information about the Magic Formula, see the CarMaker Reference Manual.



$$y(x) = D \cdot \sin[C \cdot \mathrm{atan}\{Bx - E(Bx - \mathrm{atan}(Bx))\}]$$

Figure 5.47: The Magic Formula

In the following there will be given some advice on how to generate a Magic Parameters dataset based on measured raw data from a flat belt tire testrig. The used test data is from the database of FSAE TTC - Tire Test Consortium.

"The Formula SAE Tire Test Consortium (FSAE TTC) is a volunteer-managed organization of Formula SAE teams who pool their financial resources to obtain high quality tire force and moment data. The FSAE TTC's role is to gather funds from participating FSAE teams, organize and conduct tire force and moment tests and distribute the data to all participating teams." [2]

To be able to generate a set of parameters from the tire data you need to fit the Pacejka tire model to the tires' measured raw data. The fitting process of the Magic Formula curves is an iterative process. It is either feasible to program a script in e.g. MATLAB to generate the parameter set or to use commercial software. In the FSAE many teams use the software OptimumTire from OptimumG.

"OptimumTire is a convenient and intuitive software package that allows users to perform advanced tire data analysis, visualization, and model fitting. The model fitting procedure is very fast and efficient partially due to the data processing tools incorporated into the software." [3]

The software is available as free trial version at optimumg.com. The preprocessing and fitting process to generate the Magic Formula parameter set of a tire is described in a tutorial also available at OptimumG's website.

In addition to the tutorial a few hints regarding the preprocessing and optimization:

- Take care of the coordinate system you use while importing the raw data into OptimumTire. The coordinate system the data was measured is in SAE coordinate system.
- The raw data including warming phase and beginning and end of SA sweeps looks like this:

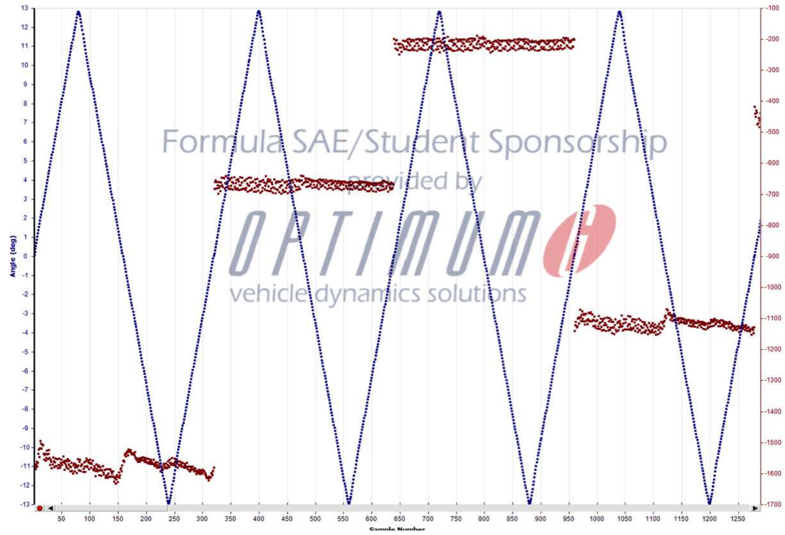


Figure 5.48: Raw tire data in OptimumG

- You should remove the warming phase and spring stiffness measurements as well as the first and last 3˚ of SA sweeps.

  The cropped data looks way smoother and periodical. The fitting process works way better based on this kind of data.

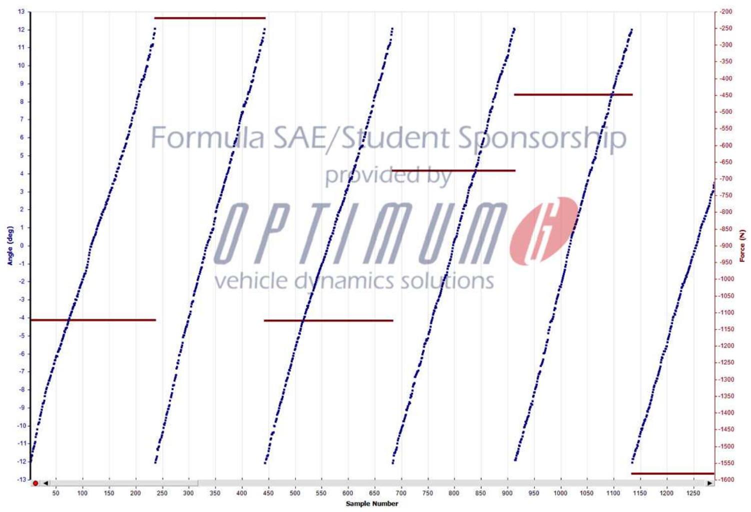


Figure 5.49: Cropped tire data in OptimumG

Import Model into CarMaker

Before you export your data you should choose ISO as coordinate system for your fitted Pacejka MF5.2 model. The question pops up whether you want to "convert" or "interpret as new coordinate system". Choose convert!

In OptimumTire there are several ways of exporting the generated data. Use the export to TIR function.

Exercise

- Copy the generated tire data and paste it in an ASCII File:
  Rename the file something like DD.MM.JJJJ_TireType_TyreSize.tir and put it into / Data/Tire/Examples/Pacejka in your CarMaker project directory.
- Use the CarMaker Tire Data Set Editor to convert the .tir file to a CarMaker Infofile:
  For this, go to "Parameters > Vehicle > Tires" in the CarMaker main GUI. Keep the folder button pressed and select "Edit" from the dropdown menu. The Tire Data Set Editor opens.

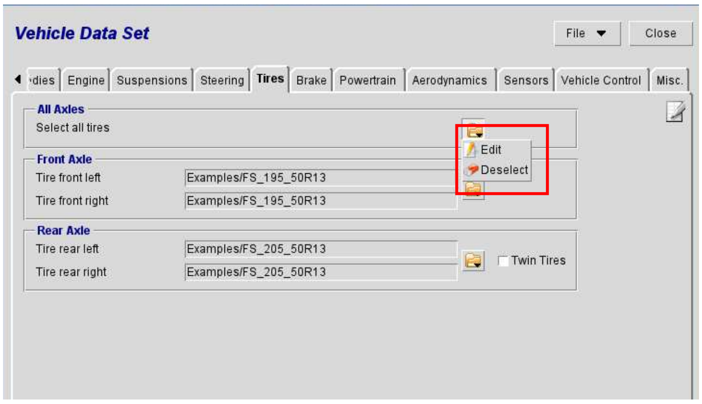


Figure 5.50: The Tire Data Set Edito in CarMaker

- In the File menu of the new window, select "New > Magic Formula 5.2".

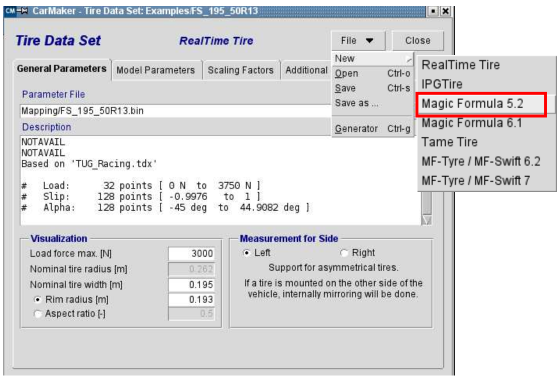


Figure 5.51: Generating a new Pacejka tire data set

- In the first tab "General Parameters" go to the section *Visualization* and adapt the tire size according to your data set. This information is used by IPGMovie for visualization only, it does not effect the physical tire characteristics itself. In the next tab "Model Parameters" import the generated .tir file and with it all the Magic Parameters to define the

Pacejka model. Select the button labeled *Import Adams Property File* and choose your tire data from the Examples/Pacejka folder.

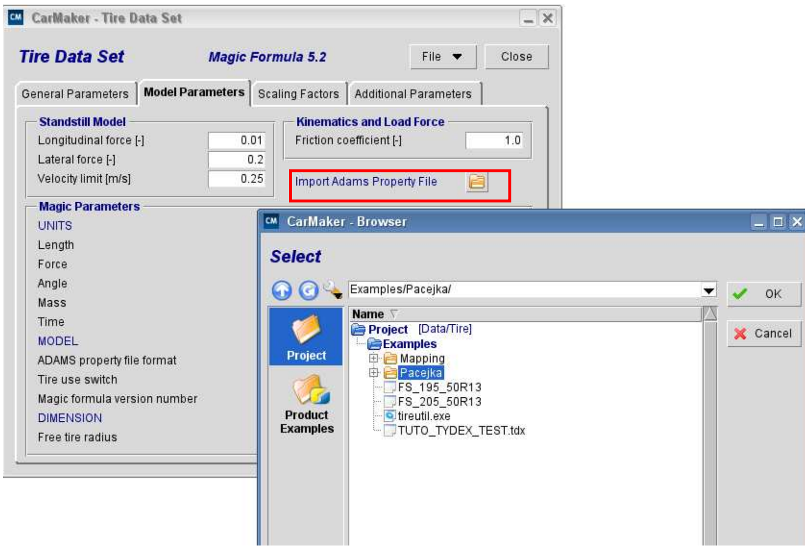


Figure 5.52: Import of an Adams property file

- Make sure that the following parameters are set correctly:
  - section "MODEL"

```
PROPERTY_FILE_FORMAT= MF_05  # ADAMS property file format
USE_Mode= 14         # Tire use switch
```

  - section "ALIGNING"

```
Align.SSZ2= -2.2371e-09  # variation of distance s/R0 with Fy/FzNom
Align.SSZ3=3.6029e-08   # variation of distance s/R0 with camber
Align.SSZ4=3.9826e-08   # variation of distance s/R0 with camber and load
Align.QTZ1=0.3      # Gyroscopic torque constant
Align.MBELT=7.5      # belt mass of wheel
```

  - section "LONGITUDINAL"

```
PTX1= 1   # Relaxation length SigKappa/Fz at FzNom
PTX2= 0.2  # Variation of SigKappa/Fz with load
PTX3= -0.15 # Variation of SigKappa/Fz with exponent of load
```

Attention: The decimal separator in all CarMaker and IPGKinematics applications is dot (.) instead of comma (,). If there is any error message related to inconsistent metrics most of the time it is due to wrong decimal separator.

- Save your tire data set via "File > Save As" in the Tire Data Set dialog.

5.6.2 IPGTire

The IPGTire model is also based on measurements. Although, the results aren't taken to implement any mathematical formulas but the discrete values can be directly used. The missing values are estimated via interpolation. The basis for the tire data provides the TYDEX format. The required files for CarMaker are generated out of a TYDEX file using the program "tireutil.exe" which is also a tool of the IPG Automotive GmbH. Thus, all you have to do to create a tire dataset is to prepare a TYDEX file and transform it.

What is a TYDEX File?

TYDEX is an abbreviation for "Tyre Data Exchange". A TYDEX file is a special format widely used in the automotive industry to export tire measurement data. It is a ASCII-file with the extension .tdx that can be opened and read by anyone interested at any time. The TYDEX format aims at making it easier for participating companies and institutes to exchange tire data.It is usually generated directly by the tire testbed measurement software and should not be written by hand.

You can find an example TYDEX file in the FS_Generic_2018 project folder under "Data/ Tire/Examples/TUTO_TYDEX_TEST.tdx"

Content of a TYDEX file

Coordinate System

The TYDEX format considers three different coordinate systems in which the values are measured:

TYDEX-C Its origin lies in the center of the wheel. The x-axis is in the central plane of the wheel and is parallel to the ground, the y-axis is identical with the spin axis of the wheel (thus it may not be parallel to the ground in case of non-zero camber angle) and the z-axis points upwards and is perpendicular to the x-y-plane. So, this coordinate system moves with slip and camber.

TYDEX-H The origin lies also at the center of the wheel. The x-axis is in the central plane of the wheel and is parallel to the ground, the y-axis is perpendicular to the x-axis and is together with the x-axis in a plane parallel to the ground and the z-axis points upwards and is perpendicular to the track surface. So, it moves with slip, but keeps perpendicular to the road in case of camber.

TYDEX-W This coordinate system is similar to the H-coordinate system, but its origin lies at the center of the footprint on the track surface. All axes are oriented like in the H-coordinate system. So, it moves with slip, but keeps perpendicular to the road in case of camber.

TYDEX Structure

Each TYDEX file is separated in single paragraphs. These paragraphs serve to structure the tire data. In total there are 14 paragraphs, but not each one is needed. However, the more information is available the better the resulting tire model is. Each single paragraph is introduced by one of the following key words:

```
**HEADER
**COMMENTS
**CONSTANTS
**MEASURCHANNELS
**MEASURDATA
**MODELDEFINITION
**MODELPARAMETERS
**MODELCOEFFICIENTS
**MODELCHANNELS
```

```
**MODELOUTPUTS
**MODELEND
**END
```

The two most important blocks are "**MEASURCHANNELS" and "**MEASURDATA". The first one defines the variables, values are available for. In the second block these values are listed. While editing you must be aware of the order: Every line of the "**MEASURCHANNELS" paragraph belongs to the corresponding column in the "**MEASURDATA" block. The first row of the "**MEASURCHANNELS" paragraph deals for instance with the lateral force F_y, the first column in the "**MEASURDATA" block has to contain the measure data of the lateral force. All acting forces (tangential force F_x, lateral force F_y, tire load F_z) and torques (pitching moment M_x, driving/braking torque M_y, self-aligning moment M_z) on the tire can be entered. Moreover there are several parameters like slip and inclination angle. You can find a complete list of all possible parameters on this website:

http://www.fast.kit.edu/download/DownloadsFahrzeugtechnik/TY100531_TYDEX_V1_3.pdf

Data with Camber Angle

The lateral force applied on the footprint has three components (the same applies for the self aligning torque):

- one resulting from the slip angle (F_{slip}),
- another (constant) caused by the camber angle (F_{camb}),
- and a third due to the distortion of the contact surface area (F_{dis}).

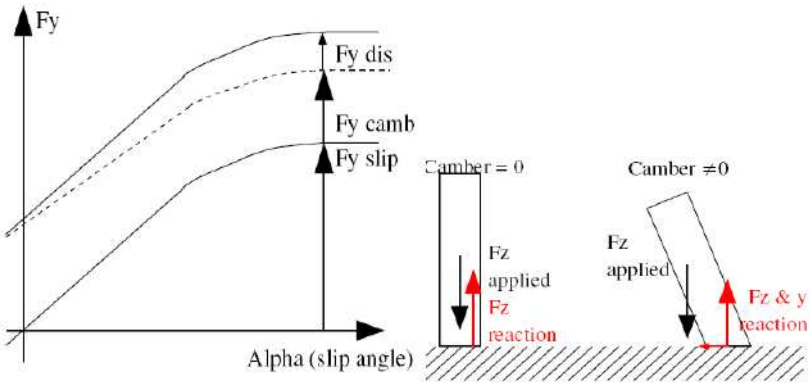


Figure 5.53: Components of the lateral force [TIR07]

But in case of an automobile tire, the last component can be neglected - as opposed to a motorcycle tire. Then the component due to the camber angle can be easily found by a state equilibrium calculation. This calculation is done by CarMaker on its own.

Nor does CarMaker need two curves with the same term. For instance: if you have two measures with F_y vs. slip angle with the same vertical load but different camber angle values, it is gratuitous to store both of them in the TYDEX file as only one camber angle measurement is needed by CarMaker.

Offset coefficients

You can specify offset and scaling factors to be applied to the values. These factors are specified for all values in the "**MEASURCHANNELS" block.

Syntax of a TYDEX file

In order to enter your data correctly, you have to follow a few advises. All characters placed after the symbol "!" are comments that are ignored.

Measurchannel

The syntax of the Measurchannel is the following:

```
**MEASURCHANNELS
PARAMETER_KEY1  comment1  unit1  correction factors value1
PARAMETER_KEY2  comment2  unit2  correction factors value2
PARAMETER_KEY3  comment3  unit3  correction factors value3
```

and so on...

Between each part of the line you should leave a certain number of blank spaces; pay attention to not use Tab-spaces:

- the first character of the parameter key starts at the first character of the line
- the first character of the comment is the 11th character of the line
- the first character of the units is the 42nd character of the line
- the first character of the first factor is the 52nd character of the line
- the first character of the second factor is the 62nd character of the line
- the first character of the third factor is the 72nd character of the line

The parameter key must be written according to the list available on the mentioned website. The parameter key also defines which axis system is used for the values: for example, "FZH" defines a vertical load in the TYDEX-H axis system and "FZY" defines the same vertical load in the TYDEX-C axis system.

Measurdata

In this block, the values on each line describe a particular point of the curve. A new line defines a new point. Each value must be separated from the other by at least one blank space.

For instance, hereunder 3 points are defined. In each line the first value is the vertical force, the second the slip angle, the third the camber angle, the fourth the lateral resulting force and the fifth the self-aligning torque. In the hereunder example the "**MEASURCHANNELS" block is displayed again for reminding:

```
**MEASURCHANNELS
FZH    Vertical force         N     1    0     0
SLIPANGL Slip angle          deg    1     0     0
INCLANGL Inclination angle        deg    1     0     0
FYH    Lateral force         N     1     0     0
MZH    Aligning moment        Nm    1     0     0
**MEASURDATA
1211.00  -0.50  0.00  211.21  -2.27
3021.70  -0.50  0.00  460.01  -8.08
4828.00  -0.50  0.00  641.08  -15.50
```

If, for example, you don’t have the values of the self-aligning torque, you will write:

```
**MEASURCHANNELS
FZH    Vertical force         N     1    0     0
SLIPANGL Slip angle          deg    1     0     0
INCLANGL Inclination angle        deg    1     0     0
FYH    Lateral force         N     1     0     0
**MEASURDATA
1211.00  -0.50  0.00  211.21
3021.70  -0.50  0.00  460.01
4828.00  -0.50  0.00  641.08
```

Furthermore, it is possible to define several couples of "**MEASURCHANNELS" and "**MEASURDATA" blocks in the same file. Thus, it is a good idea to define each curve in a special pair of blocks.

However, the former examples are only excerpted from a file. Copying and pasting the examples as they are in a file will generate an error message because there is only one value in the entire column for the inclination angle. In case you only have one value for a certain parameter you must leave it out.

Coverting a TYDEX file for CarMaker

Although a TYDEX file contains all data required by CarMaker, it can't be simply embedded in CarMaker. The data must be transformed first. Therefore, the IPG Automotive GmbH offers a tool called "tireutil.exe". Executing that tool two files are generated, both needed to model a tire in CarMaker.

To execute this tool it must be located in the same folders as your TYDEX file. You can find it in the installation folder "IPG", usually under: `C:\IPG\hil\win32-versionNumber\bin`. Copy it in your current project directory in the folder `FS_Generic_2018\Data\Tire` where you saved the `TUTO_TYDEX_TEST.tdx`.

Now, hit "Windows Start button > Execute". To open the command line window type "cmd". To use "tireutil.exe" you have to browse to your project folder, let's say `C:\CM_Projects\FS_Generic_2018\Data\Tire`. In the command line window type "`C:`" and hit "enter". To open the respective folder type the following line and hit "enter":

"`C:\>cd CM_Projects\FS_Generic_2018\Data\Tire`".

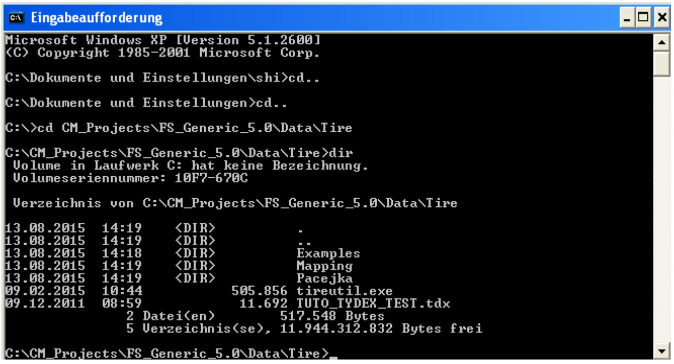


Figure 5.54: Windows command line

Once you are located in the right directory you have to call the "tireutil.exe" command. You have to clarify the input file (your TYDEX file, e.g. "TUTO_TYDEX_TEST.tdx") and the output files (a file without extension and a file with the .tir extension). The name of the output files can be different from the input file, and must be written here without extension:

```
C:\CM_Projects\FS_Generic_2018\Data\Tire > tireutil -if TUTO_TYDEX_TEST.tdx -of
TUTO_TYDEX_TEST -ofbin Mapping/TUTO_TYDEX_TEST.bin
```

The files generation starts (according to the tydex file weight, it can last several seconds). Once the generation is finished the command line displays:

```
C:\CM_Projects\FS_Generic_2018\Data\Tire>_
```

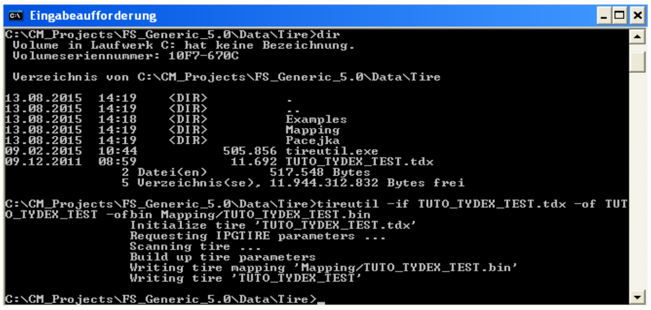


Figure 5.55: Initializing Tire

Now open your Windows Explorer and browse to the folder TIRE: you can now see in addition to the `TUTO_TYDEX_TEST.tdx` a new file respectively named TUTO_TYDEX_TEST (control file). The generated file TUTO_TYDEX_TEST.bin (data file) directly was attached to the Mapping folder. You can close the command line window.

Exercise

- Do the changes described before with your own files.
- Now you have created your own tire file successfully! Open CarMaker and test the new tire: in the "Tires" selection area you will find your new tire under the name "TUTO_TYDEX_TEST". Select it and perform a simulation.

5.6.3 TameTire

Tame Tire is a physical tire model developed by the Michelin R&D Team. The model aims at an accurate description of the transient mechanical and thermal behaviour of the tire. It is a part of the standard CarMaker license but it requires an additional license as well.

For detailed information about the TameTire model please refer to the TameTire documentation and the section Tire Model "Tame Tire" in the Reference Manual.

5.6.4 Tire Data Set Generator

The Tire Data Set Generator is a tool to create generic tire data sets for a specific vehicle class and tire dimensions. The algorithm is based on the Magic Formula approach and on the tire characteristics defined by the European Standard Tire and Rim Technical Organization (Abbreviated as ETRTO).

Based on the user inputs and the ETRTO references, the constructive parameters are determined analytically. Measurement curves defining the tire force and torque characteristics are generated with the help of Pacejka's Magic formula. This results are saved as measurement curves in the TYDEX format. (See section 5.6.2 for more information about TYDEX). Once the TYDEX file has been created, the Tire Data Set Generator calls the tireutil application to convert the TYDEX file into an infofile and a binary file that can be used by the CarMaker.

The Tire Data Set Generator is available in the file menu of the Tire Data Set Editor. Figure 5.56 describes the path for accessing the Tire Data Set Generator.

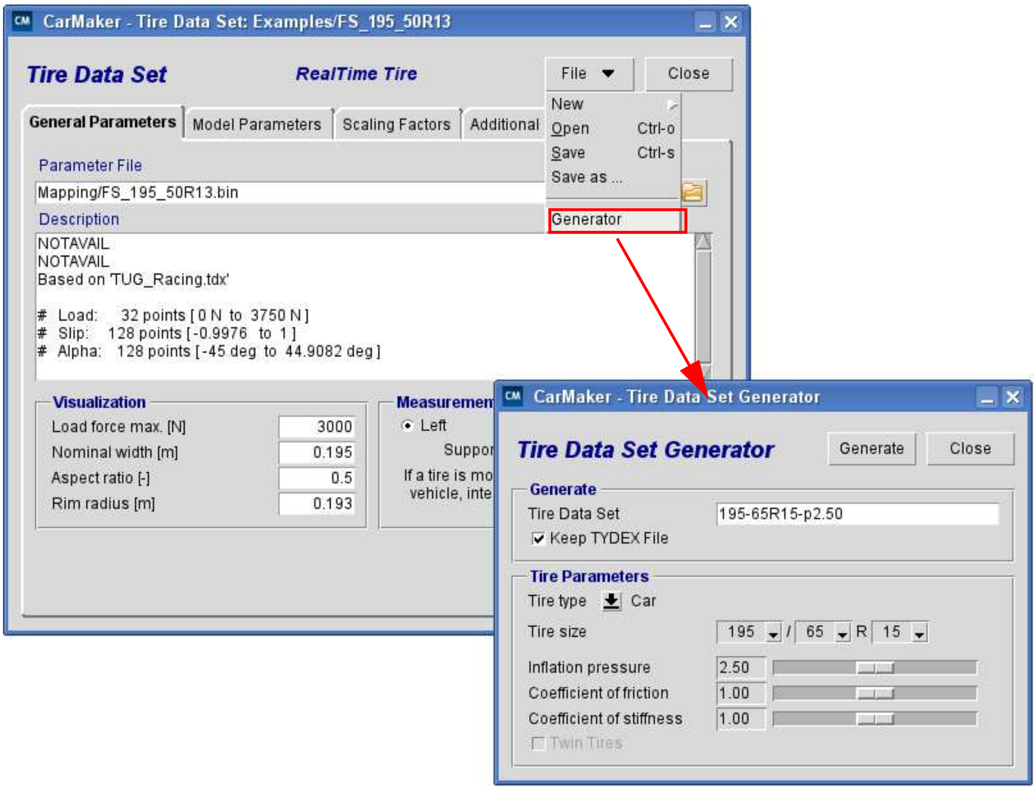


Figure 5.56: Opening the Tire Data Set Generator

For further information please refer Tire Model "Tire Data Set Generator" in the User's Guide.

Chapter 6

Simulation and Model Validation

Introduction

Any preexisting or self-generated vehicle can't serve as reference as long as it hasn't been validated. This is due to the many estimated assumptions and potential mistakes probably made in the parametrization process. Before starting with the real model validation a few plausibility checks should be performed to eliminate fundamental input errors. In that context, the suspension models and kinematics results should be checked as well.

Once the plausibility checks are completed the validation process can be started. To which extent it is practiced, mainly depends on the available time, equipment and budget. If the just mentioned resources can't be provided in a sufficient degree, the accuracy of measurements should have a higher priority than the quantity of measurements. Moreover, in that case, even more attention should be paid to the plausibility checks.

6.1 Plausibility Checks of Axle Models

6.1.1 Variation of Camber and Inclination vs. Wheel Travel

Constructively, the variation of the camber of an axle can be easily determined via the function of wheel travel vs. angle of inclination change ($\Delta\sigma$). The existing resilience's are neglected though. Concerning a double wishbone axle, both control arms (lengths e and f) move on circular paths around the junctures on the body (C and D). If the lower juncture between wishbone and suspension is lifted or lowered by the distance s_1, the actual inclination change (respectively the actual angle of inclination change) can be determined. In case of a double wishbone axle, the angle of inclination change is the same as the angle of camber change and thus the camber angle can be estimated at the same test.

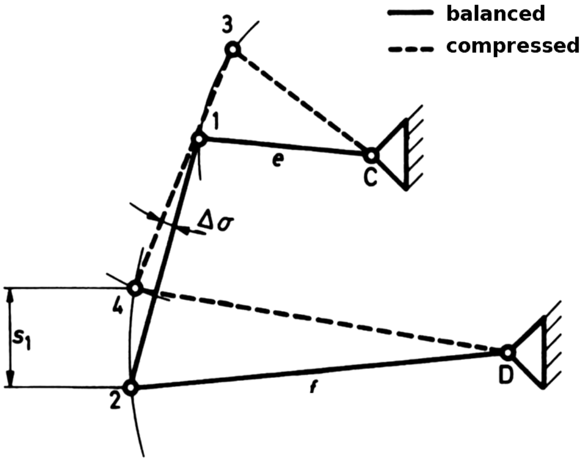


Figure 6.1: Determination of the angle of inclination and camber change [RB00]

The process explained above needn't be done with drawings but is a lot more precise using CAD programs. Therefore, all kinematic points are inserted into a sketch. Then, the lengths of the connecting lines and the two junctures of wishbone and chassis are fixed. Modifying the point "Lower Wishbone - Suspension" the compression can be simulated.

Figure 6.2: CAD sketches of a suspension

Table 6.1: Camber and toe change.

wheel travel s_1	camber measured	camber calculated	inclination measured	inclination calculated	relative error
30 mm	-1.61 deg	-1.46 deg	16.26 deg	16.26 deg	0.00%
25 mm	-1.07 deg	-1.20 deg	15.99 deg	15.86 deg	0.82%
20 mm	-1.61 deg	-0.95 deg	15.71 deg	15.60 deg	0.71%
15 mm	-0.79 deg	-0.71 deg	15.44 deg	15.36 deg	0.52%
10 mm	-0.52 deg	-0.47 deg	15.17 deg	15.12 deg	0.33%
5 mm	-0.26 deg	-0.23 deg	14.91 deg	14.88 deg	0.20%
0 mm	0 deg	0 deg	14.65 deg	14.65 deg	0.00%
-5 mm	0.25 deg	0.23 deg	14.40 deg	14.43 deg	0.21%
-10 mm	0.51 deg	0.45 deg	14.14 deg	14.20 deg	0.42%
-15 mm	0.96 deg	0.67 deg	13.89 deg	13.98 deg	0.65%
-20 mm	1.01 deg	0.89 deg	13.64 deg	13.77 deg	0.95%
-25 mm	1.25 deg	1.10 deg	13.40 deg	13.55 deg	1.11%
-30 mm	1.50 deg	1.31 deg	13.15 deg	13.34 deg	1.44%

The relative error is always less than 2%. It results from the existing resilience's in the bearing which are not considered in the measurements as opposed to IPGKinematics.

6.1.2 Track Change

The compression of the wheels causes a track change at almost every suspension, so do the double wishbone axles. Thus, this track change should be checked as well. It can be done using the sketch below. You just have to add the wheel contact point and fix the connecting line of the two junctures between wheel carrier and body.

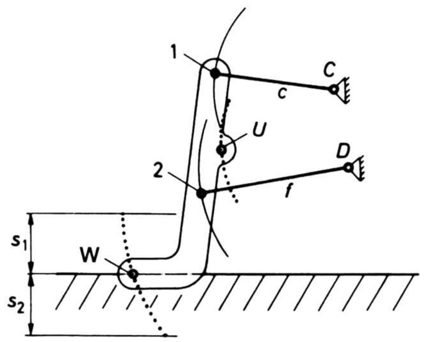


Figure 6.3: Determination of the track change [RB00]

6.1.3 Toe Change

A change in toe angle due to the compression of the wheels can be used to achieve certain driving characteristics. In general, the toe change should be in the range of minutes and never above, as this leads to an unforeseeable handling of the car.

If IPGKinematics releases a nearly constant toe change vs. wheel travel this result is very likely correct and a check-up isn't compulsory. But if it's varying or even exceeds the minutes-range the, model should be checked in any case.

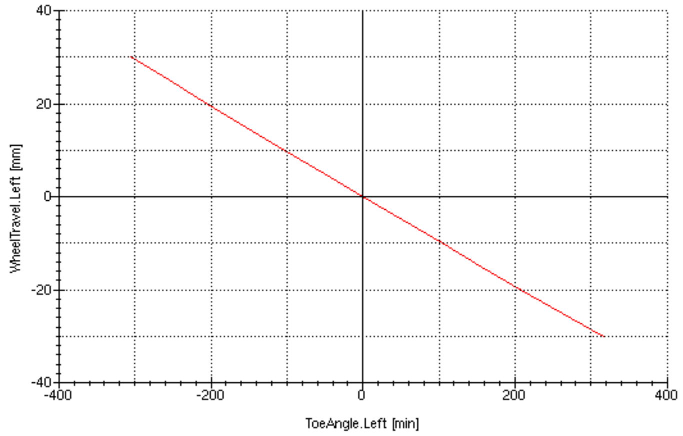


Figure 6.4: Toe change vs. wheel travel

If you have such a remarkable toe change as shown in the diagram above the idea suggests itself that there is a mistake in the model parametrization. Of course, this is not unlikely and you should check your model carefully. Pay attention in particular to the coordinates of the wishbones, the axle journal and the tie rod. If you can't find any mistake in your model, you should ask the question, if the calculations may be incorrect. The result is absolutely unsatisfactory, but constructively possible! To check this out, a measurement should be performed directly on the car.

To avoid the compression inducing forces, the car is jacked up and the wheels are disconnected from the spring/damper unit. Before doing so, the wheels should be put on a vertically adjustable platform thus the wishbone bearings needn't carry the whole self-weight. With the use of an angle plate it is now possible to measure the true toe angle.

As the tire deflects at the footprint it is wider in that area. For that reason the angle plate should be attached to the rim and never to the tire. But therefore, the angle plate must feature distance plates to equate the distance between rim and tire. The following figure shows such an angle plate.

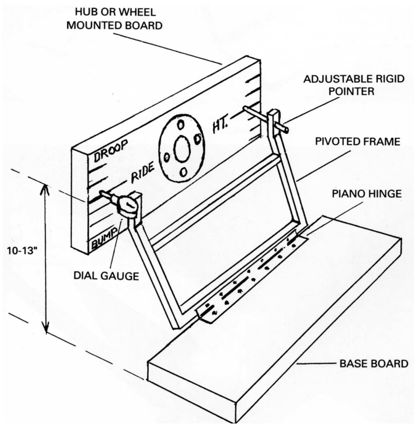
 

Figure 6.5: Angle plate used to measure the toe angle.

Via lifting and lowering the plate under the tire, the compression can be simulated and the corresponding toe angle can be measured using the angle plate. During the measurements it is important to fix the steering to prevent influences of steering angle.

After measuring the toe angle directly on the car the results can be compared with the calculations. If both values are the same your model is correct. However, it should be modified urgently. This can be done by adjusting the steering system, respectively the position of the tie rod. Further explanations on this can be found in the following paragraph.

6.1.4 Steering Angle/Steering Ratio

After checking the toe change, the steering system should be looked closely. To measure the steering angle the same configuration as described above can be used. Another possibility is to put the roadworthy car on the ground and do the measurements right there. If the driver was included in the former calculations of the design position, this second variant should be chosen in any case to prevent errors.

Once the car is on the ground, firstly the steering gear ratio should be verified. Therefore, the best way is to give a certain steering angle and measure the corresponding rack travel. To verify the result, the test can be performed in the other direction once again.

After checking the steering ratio, the steering angle is next. For this purpose you can avail yourself of the angle plate once more. It is attached to the rim again. Once you marked the reference line, the steering angle (or the rack travel) can be increased step by step while recording the corresponding steering angles at the right and left side.

6.1.5 Turning Track Diameter

The turning track diameter is the circular arc being followed by the outer wheel contact point at fixed steering angle. It is quite common to refer to the smallest possible arc as turning track diameter. It is attained by applying the maximum steering angle. Hence, measuring the turning track diameter means in general checking the steering angles. For this procedure, a driver isn't required. It is sufficient to fix a certain steering wheel angle or a steering rack travel and then pushing the car on that circular path. In this manner it is a lot easier to maintain a given steering angle than it was in a driven car.

Table 6.2: Checking the turning track diameter.

steering rack travel	turning track diameter measured	turning track diameter calculated	relative error
0 mm	infinite	infinite	0.00%
5 mm	69.00 m	68.12 m	1.29%
10 mm	34.50 m	34.06 m	1.29%
15 mm	23.00 m	22.71 m	1.28%
20 mm	17.20 m	17.05 m	0.88%
25 mm	13.80 m	13.65 m	1.10%
30 mm	11.60 m	11.39 m	1.84%
35 mm	10.00 m	9.78 m	2.25%
40 mm	8.80 m	8.58 m	2.56%

6.1.6 Anti-Dive and Anti-Squat

While braking, the front axle compresses by the distance Δs_f and the rear axle by the distance Δs_r. The pitch angle φ [rad] can be calculated out of the two distances and the wheelbase l:

$$\varphi = \frac{\Delta s_f + \Delta s_r}{l} \tag{EQ 44}$$

The skewing of the rods due to compression can be reduced by a double wishbone axle. In case the brake is located outside the wheel, the wishbones must be twisted to each other to countervail the emerging forces. However if the brake is located centrally at the differential, the wishbones must be skewed in the same rotational direction. If the wishbones are orientated parallel there is no anti-dive or anti-squat.

6.2 Plausibility Checks of the Vehicle Data Set

6.2.1 Center of Gravity and Moments of Inertia

In section "Vehicle Body", pg. 77 the determination of the vehicle's center of gravity was treated in detail. If the process of determination wasn't performed as carefully as described in that chapter, the results should be investigated once again. The generated tcl-chart can serve as a rough approximation. Here, the entered points for the single centers of gravity should be placed at proper positions. To check the dimensions of the moments of inertia, each part should be replaced by a similar geometric body whose moment of inertia can be determined exactly. Therefore, simple bodies like rectangles, cylinders, hollow cylinders or balls are mostly sufficient.

6.2.2 Comparability of IPGKinematics and CarMaker

A few parameters are required by both programs, IPGKinematics and CarMaker, including amongst the other parameters like stabilizer bar front/rear, spring rates and spring characteristics front/rear, masses and axle loads. Each of these parameters should have the same value in both programs. Most of the parameters are declared equally in both programs. The only exception is the stabilizer bar. It is defined in different terms whose translation was explained in section "Suspensions", pg. 82.

6.2.3 Assignation of the Spring Length l_0 at Front and Rear Axle

In section "Suspensions", pg. 82 the spring rates for front and rear axle were entered but the spring lengths l0 have still been left out for the following reason. If you chose a linear kinematic model the length l0 serves as adjustment parameter. Using non-linear models, the length l0 is the actual free length of the spring. The determination of this parameter is done via the static equilibrium configuration of the vehicle. To perform an equilibrium calculation you can use the CarMaker "Model Check" (CarMaker main GUI: "Simulation > Model Check", see section "Model Check", pg. 36). With this tool the user can create and analyze all car specific diagrams. One amongst the others is the analysis of the static equilibrium configuration.
To activate this tool, deselect all options except "Vehicle Characteristics". With having activated both options in this section the tool can be started hitting the "Generate Diagrams" button. After completing the calculations a text file opens containing the results. At the beginning of this file all important input parameters are listed. At the end of this file you can find the results of the static equilibrium calculation. The following shows an excerpt of such a file which contains the results of the static equilibrium calculation.

```
### Geometry (equilibrium or start-off configuration)

               :            x             y             z
VhclPoI        :        1.383 m       0.000 m       0.529 m  FrD
               :        1.383 m       0.000 m       0.529 m  Fr1
               :        2.386 m      -0.000 m       0.506 m  Fr0

Fr1 Origin     :        1.000 m       0.000 m      -0.018 m  Fr0
Fr1 Roll       :        0.000 deg  Fr0.X
Fr1 Pitch      :        0.229 deg  Fr0.Y
Fr1 Yaw        :        0.000 deg  Fr0.Z


GenBdy1        :        1.314 m      -0.000 m       0.316 m  FrD
               :        1.314 m      -0.000 m       0.316 m  Fr1
               :        2.315 m      -0.000 m       0.293 m  Fr0

ConBdy1        :        1.334 m       0.000 m       0.328 m  FrD
               :        1.334 m       0.000 m       0.328 m  Fr1
               :        2.335 m      -0.000 m       0.305 m  Fr0

               :      FL              FR              RL              RR
carrier WC tx : 2.2667206 m     2.2667206 m     0.4999997 m     0.4999997 m  FrD
           ty : 0.6049116 m    -0.6049116 m     0.6310333 m    -0.6310333 m  FrD
           tz : 0.2678416 m     0.2678416 m     0.2616688 m     0.2616688 m  FrD

carrier WC tx : 2.2667206 m     2.2667206 m     0.4999997 m     0.4999997 m  Fr1
           ty : 0.6049116 m    -0.6049116 m     0.6310333 m    -0.6310333 m  Fr1
           tz : 0.2678416 m     0.2678416 m     0.2616688 m     0.2616688 m  Fr1

carrier WC tx : 3.2677741 m     3.2677741 m     1.5010427 m     1.5010427 m  Fr0
           ty : 0.6049116 m    -0.6049116 m     0.6310333 m    -0.6310333 m  Fr0
           tz : 0.2409331 m     0.2409331 m     0.2418293 m     0.2418293 m  Fr0

carrier Mnt tx: -0.0002794 m    -0.0002794 m    -0.0000003 m    -0.0000003 m  Fr1
           ty : -0.0000884 m     0.0000884 m     0.0000333 m    -0.0000333 m  Fr1
           tz : 0.0008458 m     0.0008458 m    -0.0003271 m    -0.0003271 m  Fr1


compression q0  : 0.0009741 -    0.0009741 -   -0.0003483 -   -0.0003483 -
coordinate q1   : -0.0000000 -   -0.0000000 -   -0.0003483 -   -0.0003483 -
spring coord    : 0.1751192 m    0.1751192 m    0.1426429 m    0.1426429 m
damper coord    : 0.0989997 m    0.0989997 m    0.1003696 m    0.1003696 m
buffer coord    : -0.0008458 m   -0.0008458 m    0.0003271 m    0.0003271 m
stabi coord     : 0.0000000 -    0.0000000 -    0.0000000 -    0.0000000 -

camber          : -0.0140927 deg   -0.0140928 deg    0.0159658 deg    0.0159656 deg
                : -0.8455619 min   -0.8455700 min    0.9579481 min    0.9579378 min

toe             : -0.1664832 deg   -0.1664846 deg   -0.0000944 deg   -0.0000944 deg
                : -9.9889940 min   -9.9890780 min   -0.0056652 min   -0.0056651 min

caster          : -0.0390493 deg   -0.0390497 deg    0.0000424 deg    0.0000424 deg
                : -2.3429591 min   -2.3429791 min    0.0025417 min    0.0025417 min

wheel center Fx : -2.7001 N       -2.7001 N       -3.8067 N       -3.8067 N  Fr1
             Fy : 0.2095 N       -0.2095 N       -0.2650 N        0.2651 N
             Fz : 813.3719 N      813.3721 N      951.2038 N      951.2054 N

wheel road   Fx : 0.0000 N        0.0000 N        0.0000 N        0.0000 N  FrH
             Fy : -0.0000 N        0.0000 N        0.0000 N       -0.0000 N
             Fz : 813.3764 N      813.3767 N      951.2114 N      951.2131 N

--------------------------------------------------------------------------------
```

Listing 6.1: Result file of static equilibrium calculation

In the middle of the last paragraph of the "Model Check" result file, there is a point called "carrier Mnt". Using a non-linear model the variable "tz" in the "carrier Mnt" section should be approximately zero for all four wheels. To achieve this the front and rear spring lengths must be modified. After each change of l0 a new "Model Check" can be performed until tz is adequately small.

If the parameter was not adjusted, the whole static equilibrium configuration of the car is wrong. It expresses whether the car is running under the track surface or well above it. It can be easily visualized with IPGMovie. The following picture shows a vehicle without any adjustment of the spring length and a car with a correct value for l0.

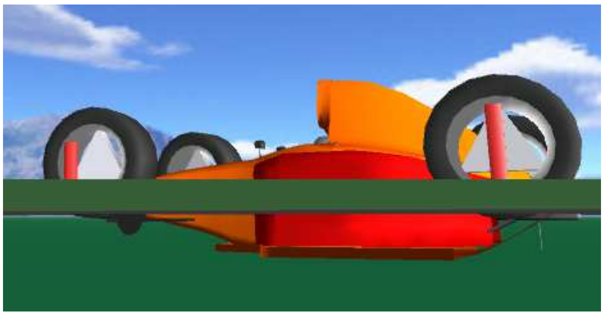


Figure 6.6: Vehicle in a wrong equilibrium state

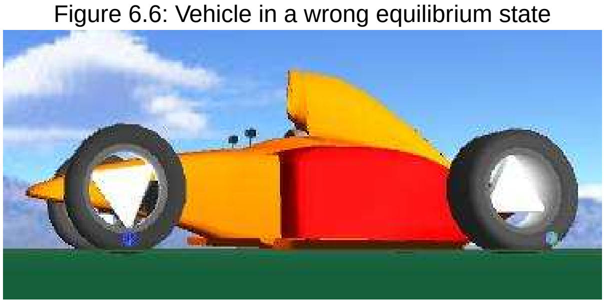


Figure 6.7: Vehicle with a correct spring length l0

6.3 Initial Start-up of the Model

6.3.1 Common Error Messages

Once you finished the plausibility checks and the spring length configuration, the actual model validation can be started. The primary step should consist of a first general simulation. The performed TestRun should be as simple as possible to emphasize the car itself.

This test run's target is to make the car drive straight along the road. If this is the case right at first go, the remaining part of this chapter can be skipped. However, in the most cases an error message will occur. In the following topics, the most common error messages and their solutions are exemplified.

For the following example it is assumed you're already familiar with the previous exercises.

Exercise

- For the first TestRun the following settings should be chosen:
  - Road: a long straight with a track width of 10 m.
  - Driver: standard driver set to "normal".
  - Maneuver: a maneuver duration of 100 s. Assign IPGDriver to both controllers.
  - Tire: Formula Student tire "FS_195_50R13" (front) and "FS_205_50R13" (rear).

"'Idle speed' bigger than 'Minimal engine speed'"

Either the "idle up/acc down" (Driver dialog) is too big/small or the engine idle speed (Vehicle Data > Powertrain > Engine) is too small/big. Change one of those to fit the settings.

"Suspension front left: Wheel and carrier must have the same position"

As explained in section "Vehicle Body", pg. 77 the center of gravity for the wheel must be the same as that for the wheel carrier. If all four centers of gravity are different, a simulation can't be executed. If the centers of gravity of only three of the four wheels are different, the simulation starts but delivers three error messages. This can be checked out in the session logbook ("Simulation > Session Log"). Here, all warnings and error messages of the current session are stored. Once CarMaker is restarted, a new session opens. After correcting the positions of centers of gravity of the wheels this error won't occur again.

"Wrong number of elements or syntax error in 'PowerTrain.GearBox.iBackwardGears'..."

Again, this is a simple faulty insertion. As mentioned in section "Drive Source", pg. 94 "Powertrain: Gearbox" a reverse gear must be defined, although Formula Student cars usually don't have one.

"Can't get parameters for body sensor 'Jack.fr'"

The actual problem is similar to the non-existing reverse gear. The program needs one point for all four jack sensors. It is possible to place all four points in the origin.

"Suspension KnC front left: No kinematics selected, model number 0"

Here, the existing model is correct in the most cases. The only problem is that the selected front axle kinematics is actually a rear axle kinematics. This mistake can easily happen when selecting the skc-files and later on it is read over. If you choose the right skc-file this problem will be solved.

"Suspension KnC 'MapNL' rear left: damper length is not decreasing monotonously (0.071741 (0,6) .. 0.0758616 (1,6))"

With this error in the session logbook a wrong value was entered in the kinematics spec. The indicator "rear" (respectively "front") shows the user at once in which skc file he has to search. As the damper length doesn't increase constantly, the mistake must be at the coordinates of the damper. Along with this error, similar error messages often exist which tell you, the spring length doesn't increase constantly, either. Here, the problem lies in the definition of the spring/damper elements. In Formula Student cars push-/pullrods are mainly used. On that reason the parameters "Damper - Body", "Damper - Lower Wishbone", "Spring - Body" and "Spring - Wheel Suspension" might not have been considered or set to "zero". In section "Geometry", pg. 68 it was pointed out that these parameters must have sensible values, nonetheless. If this wasn't taken into account an error emerges.

Fault analysis and fault isolation

If you receive error messages different to those described above, a closer fault analysis must be performed. If the error message tells you neither which parameter is affected nor in which section the problem lies, a closer examination is required. For it, each and every variable should be checked for a missing point or a wrong factor.

If you still don't find any mistake, another method should be applied. CarMaker offers almost everywhere the opportunity to import data sets from other TestRuns. In CarMaker all tabs - tantamount to partial models - and even their sub cases, like e.g. dampers in the suspensions tab, can be imported from other vehicle data sets ("File > Import").

Thus, there is the possibility to import the data of your own car step by step into an existing, operating model until the problem is found. However, this reference model should be similar to your car (e.g. the Formula Student car "FS_RaceCar_6.0" introduced in this document) to avoid new problems. In any case, different masses between the reference car and the newly generated skc-files must be evicted.

Figure 6.8: Submenu to import vehicle data sets

6.3.2 Driver Adaption

As soon as a simulation exists, that doesn't produce any errors or warnings, you can proceed to the next step: performing and analyzing initial TestRuns with your car. But before doing so, a driver adaption should be implemented ("Simulation > Driver Adaption"). This is not compulsory as the controller learns its limits by and large, but in case of racing drivers such as used in FS cars an adaption is sensible.

During the driver adaption the user can observe if the car acts in a convenient way. If this is not the case or if the adaption can't even be finished, another parameter check up should be performed. For this see the previous chapter.

Once the driver adaption is completed successfully you can start to simulate TestRuns.

.

Figure 6.9: Driver Adaption menu

6.4 TestManager

TestManager is the new update of Test Series. With this tool, you can automatically start several TestRuns one after another, and possibly simulate a TestRun in a loop with a variation of an unlimited number of variables.

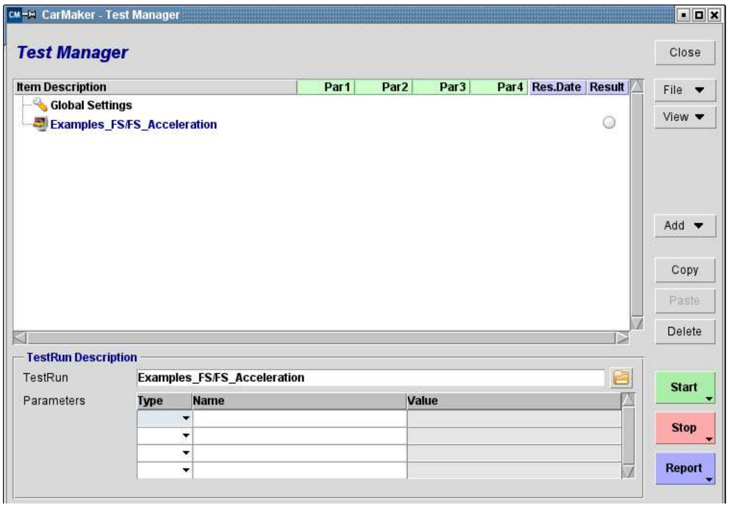


Figure 6.11: TestManager main window

One possibility to do parameter variations is to define several TestRuns with different conditions, save them and perform one simulation after the other manually. The TestManager makes this work a lot easier for you. Just select the TestRuns you have prepared by choosing "Add > TestRun". Afterwards, you have to select the TestRun from the library.

The loaded files are displayed in the TestManager window. Press the start button and you will see the simulations will be performed automatically in a row. Furthermore, the TestManager provides the flexibility in selecting which TestRuns to simulate.

Don't forget to choose the option "save all" in the *Storage of Results* box at each TestRun. Otherwise the simulation will be performed, but the results will be only saved to a buffer and thus will be overwritten at the next simulation!

6.4.1 Variations and Variable kinds

Instead of creating a new TestRun for a single variation of a quantity, you can also insert a loop to the TestManager. This can be done with the help of inserting Variations for the Predefined quantities in the TestRuns. Just create the Variation by choosing Add > Variation in the selected TestRun. This means, the same TestRun is been run for several times but each time with different User defined Variations e.g. Longitudinal acceleration. In the end, once you built an automated TestRun you can save and load it as a normal TestRun.

Variations can be defined with the help of different variables e.g. NValue or KValue. NValue abbr. Named Values (NV) are used to change the parameters in the CarMaker GUI e.g. Coordinates of CoG. On the other hand KValue abbr. Key Values (KV) are used to change the info-file parameters which do not have an editable parameter in the CarMaker GUI e.g. Gear Ratio.

.

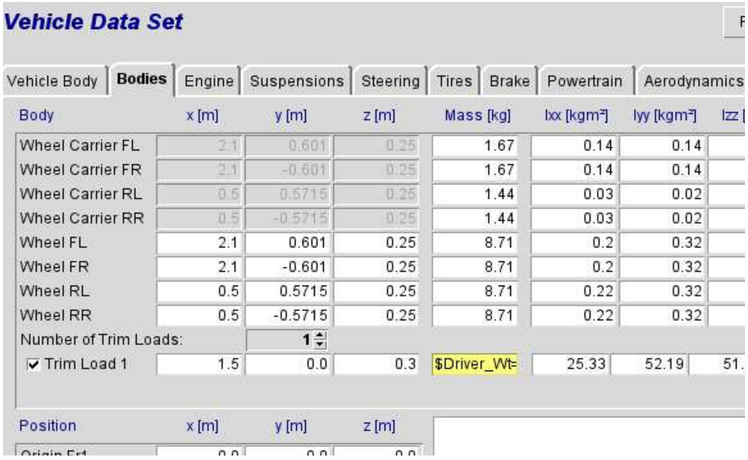


Figure 6.12: Set the variable in Vehicle Data Set dialog

Note: While defining the NValues in the CarMaker GUI, it is important the User gives a default value to the parameter to avoid getting errors while simulating TestRuns without the TestManager.

In the following example we want to show how to prepare an automated TestRun using the acceleration event and the driver weight as a variable. After having generated the TestSeries, you can start a simulation. If you want to save the TestSeries' layout, choose "File > Save As". To compare the results open IPGControl, load the different files as mentioned in exercise on page 46. Feasible values could be "Car.Pitch" or "Car.ax".

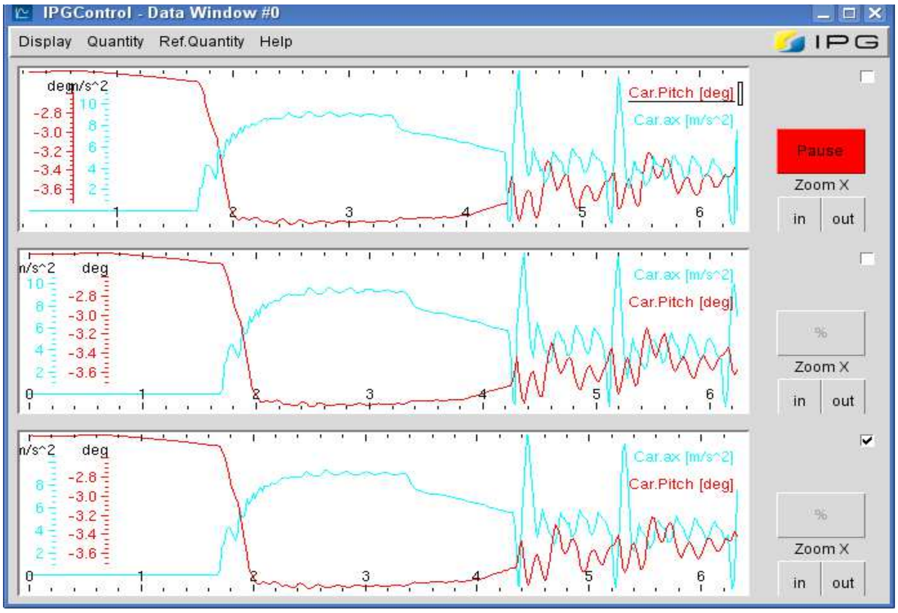


Figure 6.13: Comparison of the saved results

Exercise

- Please save your current work.
- Add a TestRun to TestManager:
  - Open "Examples_FS/FS_Acceleration".
  - Change the car to FS_RaceCar_TM.
  - Set the Trim Load1 in Bodies tab of Car to "$Driver_Wt=70" (notice that "=70" sets the variable to the default value 70) and select "save all" for the storage of results.
  - Save the new file as "FS_Acceleration_TUTO".
  - Now start the TestManager (Simulation > TestManager) and add the previous created TestRun "FS_Acceleration_TUTO". Select "Add" ->"TestRun", then load your previous created TestRun in the field "TestRun Description".
- Add variations:
  - Type "NValue", named "Driver_Wt" with the values "70", "85" and "100". A
  - As you will recognize the type and name will be transfered to next variations. You only have to insert the values.

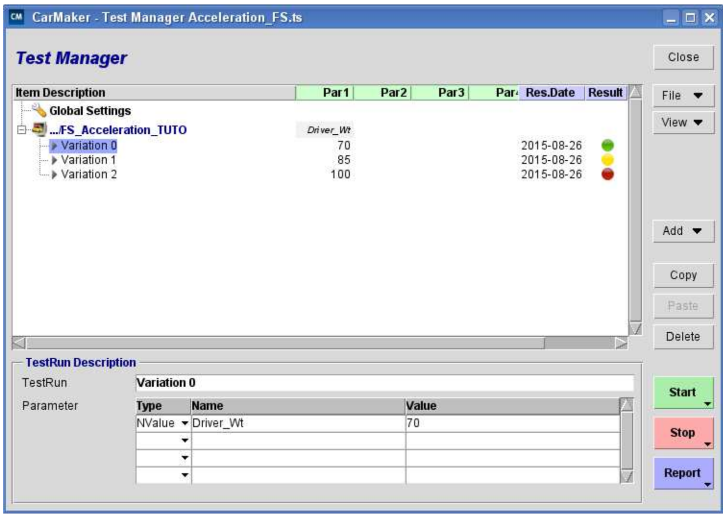


Figure 6.14: The declaration of variations

6.4.2 Criteria and Analysis

In addition to the execution of TestRun loops including parameter variations, the TestManager also provides an interface to quantify the simulations directly with the help of user defined criteria. These are basically used to gauge the Vehicle performance with respect to set targets i.e. criteria. You can create it by choosing "Add > Criterion" in the selected TestRun (Figure 6.15). It is also possible to add multiple criteria for the same TestRun.

Criteria are defined by using syntax "[get Parameter]" followed by condition. Example TestRun is considered good when the yaw angle is less than 6 degrees [get Car.Yaw] <= 6*pi/180

Note: The UAQ always follow their default units but the TestManager uses SI system. So it is important to convert the units into SI units whenever required. If you are not sure about the default unit of a quantity, please have a look at the Reference Manual, section User Accessible Quantities.

The Parameter called in the criteria expression always returns the value which it has taken at the end of simulation. Many parameters are triggered to 0 at the end of the simulation and therefore can't be called in the criteria function. In such cases new variable creation is advisable which can be used to take the value of such parameters and will not return back to 0.

You can create this variable in the Maneuver tab of the TestRun using Mini-maneuver Commands in the CarMaker GUI or by defining "Characteristic" ("Add > Characteristic", see Figure 6.17) in the respective TestRun of the TestManager.

Please note that the commands in the Characteristics tab of the TestManager override the real time expressions mentioned in the CarMaker GUI.

With the help of criteria, the TestManager evaluates if a test was successful or not and provides the User with a Visual signal which comprises of 6 different signals, each having their own meaning. For eg. Green for successfully satisfying the said criteria or Red for signifying that the criteria is not met. For more information please refer the User's Guide ("main GUI > Help > User's Guide"), section Test Automation.

In addition to setting up the Criteria, the TestManager can define diagrams in a test series. These diagrams are plotted after the simulation has come to an end. There are three different diagram modes which are supported:

- Quantity(s) vs. Time
- Quantity(s) vs. Quantity
- Characteristic value vs. Variation

It is possible to plot one diagram for each variation or to summarize all variations in a single diagram for a better overview. You can create diagrams for the said TestRun by choosing "Add > Diagram" (Figure 6.16). Here, the user will have the flexibility to name respective axis labels and parameters. To get the Diagram right, it is necessary to insert a UAQ with correct syntax.

The TestManager also includes a report functionality which automatically generates a report of the performed test series. This report can be called up either when a test series is completed or aborted by pressing the "Report" button in the lower right corner of the TestManager window. It provides detailed results of every single TestRun which are prepared and clearly arranged fully automatically. The report starts with an overview of all the TestRuns and their respective results. It also provides details of each and every TestRuns which comprises of diagrams, variations and TestRun maneuvers (Figure 6.18). The Report can be found under <yourProjectDirectory>/SimOutput/>HostName>/Report.

For more information see the User's Guide ("main GUI > Help > User's Guide"), section"Test Automation".

In continuation with the previous exercise, we will now define the Criteria, Diagrams and Characteristic in the same TestRun in the TestManager followed by Report generation.

Exercise

- Create a Criteria:
  - Select TestRun "FS_Acceleration_TUTO" in TestManager and "Add > Criterion".
  - Set the Criteria for Green, Yellow and Red as shown in figure given below.

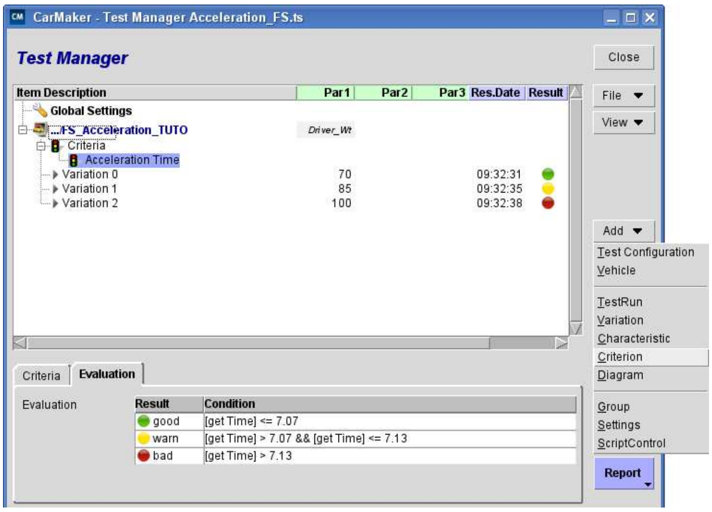


Figure 6.15: Defining Criteria

- Create a Diagram:
  - "Add > Diagram". Give diagram title as "Longitudinal Acceleration Vs Time". Tick the check-box, *Show all variations in single graph*.
  - Provide axis labels and User Accessible quantity "Car.ax" for Y-Axis.

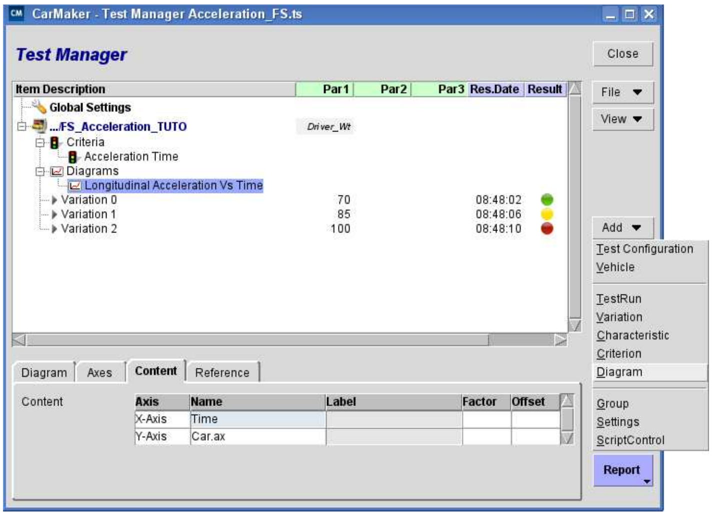


Figure 6.16: Creating Diagrams

- Define Characteristics:
  - Using "Add > Characteristic" to add condition for 80% gas after 4 seconds as shown in Figure 6.17.
- After assigning Criteria, Diagrams and Characteristics, simulate the TestRun from Test-Manager and generate the Report from TestManager GUI.

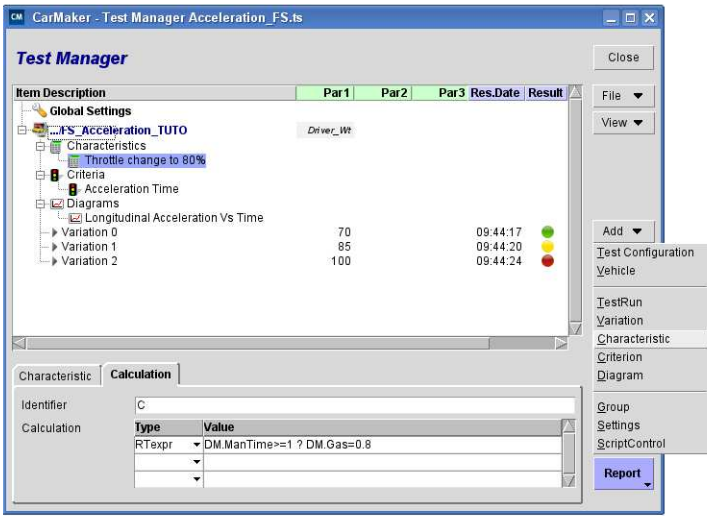


Figure 6.17: Defining Characteristic

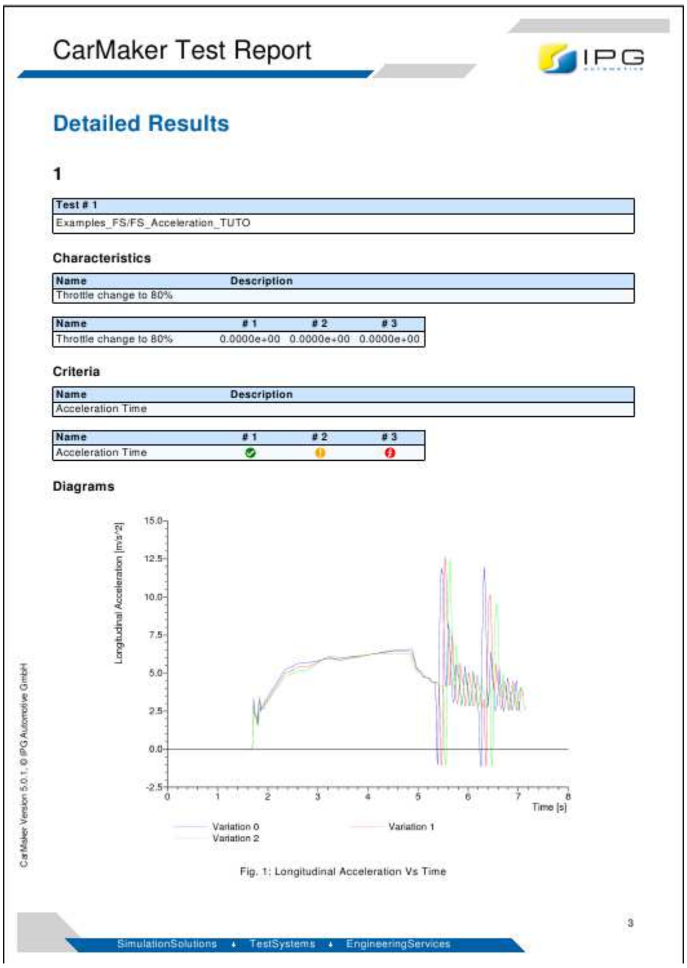


Figure 6.18: Sample Report page

6.4.3 Example Test Series

To help Formula Student teams for testing of their Vehicle Models, IPG Team has created a special TestSeries named "FS_RaceCar_Validation.ts". This TestSeries serves to support students understand their Car better using various static and dynamic TestRuns and eventually optimize their Car by analyzing the results of the TestRuns given in this TestSeries. The TestSeries is made up of various TestRuns which simulate the cars of Formula Student teams by replicating the Race environment in best possible ways.

This TestSeries comprises of major 3 groups having their own TestRuns. All these TestRuns are explained below.

Equilibrium

1. Standstill

Objective: To verify the static equilibrium of the vehicle with different loading conditions. This TestRun is a good way for students to start analyzing their vehicle behavior before going into dynamic trails. For analysis students can opt for "Toe Angle", "Camber Angle" and "Loading" on each wheel of the car.

Table 6.3: Standstill variables

Type	Variable Name	Variable Description	Parameter Location
NV	Driver_Wt	Driver Weight	Parameters > Car > Bodies > Trim Load 1 > Mass
NV	Fuel_Wt	Fuel Weight	Parameters > Car > Bodies > Trim Load 2 > Mass
NV	Position_X	X Location of Fuel Weight	Parameters > Car > Bodies > Trim Load 2 > x
NV	Position_Y	Y Location of Fuel Weight	Parameters > Car > Bodies > Trim Load 2 > y
NV	Position_Z	Z Location of Fuel Weight	Parameters > Car > Bodies > Trim Load 2 > z

Table 6.4: Quantities to be evaluated in Standstill Test

Validation Output	User Accessible Quantities
Toe Angle	Car.ToeFL, Car.ToeRR
Camber Angle	Car.CamberFL, Car.CamberRR
Vertical Loads on all 4 Wheels	Car.FzFL, Car.FzRL, Car.FzFR, Car.FzRR

Longitudinal Dynamics

1. Pull Test

Objective To check the straight line stability of the vehicle for no steering input. This TestRun is important for Driver safety and Vehicle performance. Good result for this test depends on CoG location, suspension and steering kinematics and overall symmetry. For analysis students can opt for Lateral Deviation of car, Toe Angle and Camber Angle on each wheel of the car.

Table 6.5: Variables for Pull Test

Type	Variable Name	Variable Description	Parameter Location
NV	Speed	IPG Driver Speed	Parameters > Maneuver > Longitudinal Dynamics > IPG Driver > Speed
NV	CoG_x	X Location of CoG	Parameters > Car > Vehicle Body > Rigid Vehicle Body > Vehicle Body > x
NV	CoG_y	Y Location of CoG	Parameters > Car > Vehicle Body > Rigid Vehicle Body > Vehicle Body > y
NV	CoG_z	Z Location of CoG	Parameters > Car > Vehicle Body > Rigid Vehicle Body > Vehicle Body > z

Table 6.6: Quantities to be evaluated in Pull Test

Validation Output	User Accessible Quantities
Lateral Deviation	Driver.Lat.dy
Toe Angle	Car.ToeFL, Car.ToeRR
Camber Angle	Car.CamberFL, Car.CamberRR

2. Acceleration

Objective This TestRun checks vehicle capability to reach 75m in the shortest possible time when starting from standstill. To maximize the acceleration, students can change Gear Shifting RPM and Gear ratios. For analysis students can opt for max. vehicle speed, max. longitudinal acceleration, max. engine RPM and Pitch angle of the car.

Table 6.7: Variables for Acceleration Test

Type	Variable Name	Variable Description	Parameter Location
NV	Gas	Throttle Value (0-1)	Parameters > Maneuver > Maneuver 0 > Minimaneuver Commands
NV	Shiftup	Shifting RPM	Parameters > Driver > User Parameterized Driver > Standard Parameters > Declutching / Gear Shifting > Engine Speeds [RPM] > max
KV	PowerTrain.GearBox.iForwardGears	Gear Ratio change for all gears	Path to File /Data/Vehicle/Examples_FS/ FS_RaceCar_TM

Table 6.8: Quantities to be evaluated in Acceleration Test

Validation Output	User Accessible Quantities
Vertical Loads on all 4 Wheels	Car.FzFL, Car.FzRL, Car.FzFR, Car.FzRR
Toe Angle	Car.ToeFL, Car.ToeRR

Table 6.8: Quantities to be evaluated in Acceleration Test

Validation Output	User Accessible Quantities
Camber Angle	Car.CamberFL, Car.CamberRR
Pitch Velocity	Car.PitchVel
Max Vehicle Speed	Car.v
Max Vehicle Acceleration	Car.ax
Max Engine RPM	PT.Engine.rotv
Max Engine Power	PT.Engine.PwrO
Pitch	Car.Pitch
Time	Time

3. Top Speed

Objective This TestRun checks vehicle capability to reach its Top Speed in the shortest possible time when starting from standstill. To maximize the acceleration, students can change Gear Shifting RPM and Gear ratios. For analysis students can opt for max. vehicle speed, max. longitudinal acceleration, max. engine RPM and Pitch angle of the car.

Table 6.9: Variables for Top Speed Test

Type	Variable Name	Variable Description	Parameter Location
NV	Gas	Throttle Value (0-1)	Parameters > Maneuver > Maneuver 0 > Minimaneuver Commands
NV	Shiftup	Shifting RPM	Parameters > Driver > User Parameterized Driver > Standard Parameters > Declutching / Gear Shifting > Engine Speeds [RPM] > max
KV	PowerTrain.GearBox.iForwardGears	Gear Ratio change for all gears	Path to File /Data/Vehicle/Examples_FS/FS_RaceCar_TM

Table 6.10: Quantities to be evaluated in Top Speed Test

Validation Output	User Accessible Quantities
Vertical Loads on all 4 Wheels	Car.FzFL, Car.FzRL, Car.FzFR, Car.FzRR
Toe Angle	Car.ToeFL, Car.ToeRR
Camber Angle	Car.CamberFL, Car.CamberRR
Pitch Velocity	Car.PitchVel
Max Vehicle Speed	Car.v
Max Vehicle Acceleration	Car.ax
Max Engine RPM	PT.Engine.rotv
Max Engine Power	PT.Engine.PwrO
Pitch	Car.Pitch
Time	Time

4. Braking_FS

Objective This TestRun aims to check the capability of brake system to lock all four (4) wheels and stop the vehicle in a straight line. Vehicle accelerates from standstill to 50m followed by full braking. Good result for this test depends on appropriate brake pedal ratio, CoG location, suspension and steering kinematics. For analysis students can opt for braking distance, car velocity and Yaw angle of the car.

Table 6.11: Variable for Braking_FS Test

Type	Variable Name	Variable Description	Parameter Location
NV	CoG_x	X Location of CoG	Parameters > Car > Vehicle Body > Rigid Vehicle Body > Vehicle Body > x
NV	CoG_y	Y Location of CoG	Parameters > Car > Vehicle Body > Rigid Vehicle Body > Vehicle Body > y
NV	CoG_z	Z Location of CoG	Parameters > Car > Vehicle Body > Rigid Vehicle Body > Vehicle Body > z
KV	Pedal.ratio	Brake Pedal Ratio	Path to File /Data/Misc/ HydESP_FS_RaceCar_4.0

Table 6.12: Quantities to be evaluated in Braking_FS Test

Validation Output	User Accessible Quantities
Vertical Loads on all 4 Wheels	Car.FzFL, Car.FzRL, Car.FzFR, Car.FzRR
Toe Angle	Car.ToeFL, Car.ToeRR
Camber Angle	Car.CamberFL, Car.CamberRR
Pitch Velocity	Car.PitchVel
Max Vehicle Deceleration	Car.ax
Pitch	Car.Pitch
Pedal force	Brake.Hyd.Sys.PedFrc
Pressure at master cylinder	Brake.Hyd.Sys.pMC
Pressure at each wheel	Brake.Hyd.Sys.pWB_FL, Brake.Hyd.Sys.pWB_FR, Brake.Hyd.Sys.pWB_RL, Brake.Hyd.Sys.pWB_RR
Yaw Angle	Car.Yaw
Max Vehicle Speed	Car.v
Braking Distance	To be calculated using Real Time Expressions
Braking Time	To be calculated using Real Time Expressions

5. Braking

Objective

This TestRun aims to check the capability of brake system to lock all four (4) wheels and stop the vehicle in a straight line. Vehicle accelerates from standstill to 100 km/h followed by full braking. Good result for this test depends on appropriate brake pedal ratio, CoG location, suspension and steering kinematics. For analysis students can opt for braking distance, car velocity and Yaw angle of the car. t

Table 6.13: Variables for Braking Test

Type	Variable Name	Variable Description	Parameter Location
NV	CoG_x	X Location of CoG	Parameters > Car > Vehicle Body > Rigid Vehicle Body > Vehicle Body > x
NV	CoG_y	Y Location of CoG	Parameters > Car > Vehicle Body > Rigid Vehicle Body > Vehicle Body > y
NV	CoG_z	Z Location of CoG	Parameters > Car > Vehicle Body > Rigid Vehicle Body > Vehicle Body > z
NV	Brake	Brake Value (0-1)	Parameters > Maneuver > Maneuver 1 > Longitudinal Dynamics > Manual (Pedals Gears) > Brake > Value

Table 6.14: Quantities to be evaluated in Braking Test

Validation Output	User Accessible Quantities
Vertical Loads on all 4 Wheels	Car.FzFL, Car.FzRL, Car.FzFR, Car.FzRR
Toe Angle	Car.ToeFL, Car.ToeRR
Camber Angle	Car.CamberFL, Car.CamberRR
Pitch Velocity	Car.PitchVel
Max Vehicle Deceleration	Car.ax
Pitch	Car.Pitch
Pedal force	Brake.Hyd.Sys.PedFrc
Pressure at master cylinder	Brake.Hyd.Sys.pMC
Pressure at all 4 wheels	Brake.Hyd.Sys.pWB_FL, Brake.Hyd.Sys.pWB_FR, Brake.Hyd.Sys.pWB_RL, Brake.Hyd.Sys.pWB_RR
Yaw Angle	Car.Yaw
Max Vehicle Speed	Car.v
Braking Distance	To be calculated using Real Time Expressions
Braking Time	To be calculated using Real Time Expressions

Lateral Dynamics

1. Slalom

Objective This TestRun aims to check Vehicle's response to Steering input by following continuous Left-Right turns which are spaced at a distance of 15 and 18m at varying speeds. In order to increase the cornering efficiency students can alter suspension and steering parameters so as to optimize body roll. For analysis students can opt for Body roll, Lateral acceleration and steering wheel angle of the car.

Table 6.15: Variables for Slalom Test

Type	Variable Name	Variable Description	Parameter Location
NV	CSpeed	Cruising Speed	Parameters > Driver > User Parameterized Driver > Standard Parameters > General > Cruising Speed

Table 6.16: Quantities to be evaluated in Slalom Test

Validation Output	User Accessible Quantities
Lateral Acceleration	Car.ay
Max Roll angle	Car.Roll
Roll angle gradient	Car.RollAcc
Steering wheel angle	DM.Steer.Ang
Understeer gradient	To be calculated using Real Time Expressions
Slip Angle (all wheels)	Car.SlipAngleFL, Car.SlipAngleFR, Car.SlipAngleRL, Car.SlipAngleRR,
Steering Turning Torque	DM.Steer.Trq
Fz (Vertical forces on all tyres)	Car.FzFL, Car.FzRL, Car.FzFR, Car.FzRR
Yaw	Car.Yaw
Yaw rate	Car.YawRate
Time	Time

2. Steer Step

Objective This TestRun aims to check Vehicle's response to sudden Steering input at varying speeds and steering angles. In order to increase the cornering efficiency students can alter suspension and steering parameters so as to optimize body roll. For analysis students can opt for Yaw rate, Lateral acceleration and steering wheel angle of the car.

Table 6.17: Variables for Steer Step Tes

Type	Variable Name	Variable Description	Parameter Location
NV	SteerAng	IPG Driver Steer Step Angle	Parameters > Maneuver > Maneuver 1,2 > Lateral Dynamics > Steer Step > Amplitude
NV	SpeedCtrl	Control Speed Long. Dynamics	Parameters > Maneuver > Maneuver 1 > Longitudinal Dynamics > Speed Control > Speed

Table 6.18: Quantities to be evaluated in Steer Step Test

Validation Output	User Accessible Quantities
Lateral Acceleration	Car.ay
Max Roll angle	Car.Roll
Roll angle gradient	Car.RollAcc
Steering wheel angle	DM.Steer.Ang
Understeer gradient	To be calculated using Real Time Expressions
Slip Angle (all wheels)	Car.SlipAngleFL, Car.SlipAngleFR, Car.SlipAngleRL, Car.SlipAngleRR,
Steering Turning Torque	DM.Steer.Trq
Fz (Vertical forces on all tyres)	Car.FzFL, Car.FzRL, Car.FzFR, Car.FzRR
Yaw	Car.Yaw
Yaw rate	Car.YawRate
Time	Time

3. Steady State Circle

Objective This TestRun aims to check limiting lateral stability of Vehicle by rotating in steady-state circle of 42m radius at varying speeds and suspension parameters. In order to increase the cornering efficiency students can alter suspension and steering parameters so as to optimize body roll. For analysis students can opt for Lateral deviation, Lateral acceleration and steering wheel angle of the car. t

Table 6.19: Variables for Steady State Circle Test

Type	Variable Name	Variable Description	Parameter Location
NV	CSpeed	Cruising Speed	Parameters > Driver > User Parameterized Driver > Standard Parameters > General > Cruising Speed
NV	StabAmp	Stabilizer Stiffness Amplification	Parameters > Car > Suspension > Stabilizer > Front, Rear > Amplification

Table 6.20: Quantities to be evaluated in Steady State Circle Test

Validation Output	User Accessible Quantities
Lateral Acceleration	Car.ay
Max Roll angle	Car.Roll
Roll angle gradient	Car.RollAcc
Steering wheel angle	DM.Steer.Ang
Understeer gradient	To be calculated using Real Time Expressions
Slip Angle for all 4 wheels	Car.SlipAngleFL, Car.SlipAngleFR, Car.SlipAngleRL, Car.SlipAngleRR,
Steering Turning Torque	DM.Steer.Trq
Fz (Vertical forces on all tyres)	Car.FzFL, Car.FzRL, Car.FzFR, Car.FzRR
Yaw	Car.Yaw
Yaw rate	Car.YawRate

Table 6.20: Quantities to be evaluated in Steady State Circle Test

Validation Output	User Accessible Quantities
Lateral Deviation	Driver.Lat.dy
Time	Time

For majority of the TestRuns respective acceptance criteria were defined and their meanings are expressed in the below Table.

Table 6.21: List of Criteria and Meanings

TestRun Name	Criteria	Description
Pull Test	G - [get LatShift] < abs(0.35)	Lateral Deviation of Car on either side from centre line is less than 0.35m
	Y - [get LatShift] >= abs(0.35) && [get LatShift] < abs(0.7)	Lateral Deviation of Car on either side from centre line is between 0.35m and 0.7m
	R - [get LatShift] >= abs(0.7)	Lateral Deviation of Car on either side from centre line is more than 0.7m
Acceleration	G - [get DM.ManTime] < 5	Acceleration Maneuver Time is less than 5sec
	Y - [get DM.ManTime] >= 5 && [get DM.ManTime] <= 6	Acceleration Maneuver Time is in between 5sec and 6 sec
	R - [get DM.ManTime] > 6	Acceleration Maneuver Time is more than 6sec
Top Speed	G - [get Time] < 16	Time to reach Top Speed is less than 16sec
	Y - [get Time] >= 16 && [get Time] <= 20	Time to reach Top Speed is in between 16sec and 20 sec
	R - [get Time] > 20	Time to reach Top Speed is more than 20sec
Top Speed	G - [get TopSpeed] > 44	Top Speed is more than 44m/s
	Y - [get TopSpeed] >=41 && [get TopSpeed] <=44	Top Speed is in between 41m/s and 44m/s
	R - [get TopSpeed] < 41	Top Speed is less than 41m/s
Braking_FS & Braking	G - [get Car.Yaw] < 0.087 || [get Car.Yaw] > -0.087	Car Yaw angle is less than 0.087 rad (5 deg) in both the direction
	Y - [get Car.Yaw] >= 0.087 && [get Car.Yaw] <= 0.174 || [get Car.Yaw] <= -0.087 && [get Car.Yaw] >= -0.174	Car Yaw angle is between 0.087 rad (5 deg) and 0.174 rad (10 deg) in both the direction
	R - [get Car.Yaw] > 0.174 || [get Car.Yaw] < -0.174	Car Yaw angle is more than 0.174 rad (10 deg) in both the direction

Table 6.21: List of Criteria and Meanings

TestRun Name	Criteria	Description
ZigZag	G - [get Rollmax] < 0.0104 && [get Rollmin] > -0.0104	Car Roll Angle is less than 0.0104 rad (0.6 deg) on either side
	Y - [get Rollmax] >= 0.0104 && [get Rollmax] <= 0.0122 || [get Rollmin] <= -0.0122 && [get Rollmin] >= -0.0104	Car Roll Angle is between 0.0104 rad (0.6 deg) and 0.0122 rad (0.7 deg) on either side
	R - [get Rollmax] > 0.0122 && [get Rollmin] < -0.0122	Car Roll Angle is more than 0.0122 rad (0.7 deg) on either side
Steady State Circle	G - [get LatDyMax] <=0.35 && [get LatDyMin] >=-0.35	Lateral Deviation of Car on either side from centre line is less than 0.35m
	Y - [get LatDyMax] >0.35 && [get LatDyMax] <=0.7 || [get LatDyMin] <-0.35 && [get LatDyMin] >=-0.7	Lateral Deviation of Car on either side from centre line is between 0.35m and 0.7m
	R - [get LatDyMax] >0.7 || [get LatDyMin] <-0.7	Lateral Deviation of Car on either side from centre line is more than 0.7m

Exercise

- Run your own car:
  After running these TestRuns in the TestManager with the FS_RaceCar_TM, now you can import your own car and perform these TestRuns. Select Global settings in the TestManager and Add > Vehicle. After going to the Configuration tab, select your own car from your Project Folder. After importing, save the TestSeries as "MyTestSeries".

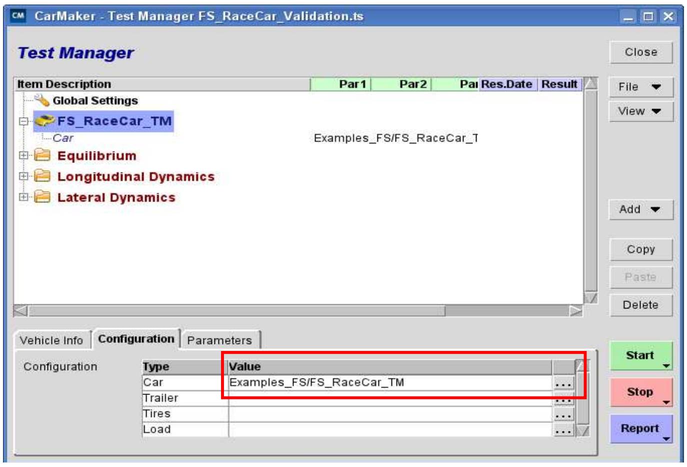


Figure 6.19: Importing the student car model

Chapter 7

Helpful suggestions

Recommended literature

If you are developing a Formula Student racing car for the first time, you might need some more detailed background information. You can find below a list of recommended literature which we think is very valuable to know.

- Gillespie, Thomas:
  Fundamentals of Vehicle Dynamics, SAE International, 1992
- Heißing, Bernd; Ersoy, Metin:
  Fahrwerkhandbuch, Vierweg-Verlag, 2007
- Matschinsky, Wolfgang:
  Die Radaufhängungen der Straßenfahrzeuge, Springer Verlag, 1998
- Mitschke, Manfred:
  Dynamik der Kraftfahrzeuge, Band A-C, Springer Verlag, 1982
- Milliken, William F.:
  Race Car Vehicle Dynamics, SAE International, 1995
- Reimpell, Jörnsen; Betzler, Jürgen:
  Fahrwerktechnik: Grundlagen, Vogel-Verlag, 2000

The company IPG Automotive GmbH observes with enthusiasm, how committed the members of each Formula Student team deal with such a huge project. From this enthusiasm the idea to support these projects arose. This is done by the supply of software licenses as well as technical support. Every team can contribute to achieve a high efficiency in this. In the following paragraph a few incitements to it are given.

Email contact

The email support is available for every team at no charge. Please address every question to **FormulaCarMaker@ipg.de**. Behind this email contact lies a contributor which forwards the incoming mails to every member of the Formula CarMaker Support Team immediately. In this way all mails are sent to the person in charge in each case. Moreover, in special cases such as an employee being on holiday or out of town all questions can be handled nonetheless.

Meanwhile the IPG Automotive GmbH supports over 200 teams worldwide. This means at two registered members per team a total of over 400 students to service. A mental association with every 400 persons is impossible. Accordingly, this must be done with the help of your email address. Many Formula Student teams do not dispose of an own email address and all that's left from your mail is your text with signature. Thus, the Formula CarMaker Support Team is very grateful for every inquiry that includes the name of your team and university.

You make our life easier, when...

- .. you add the license number to the email inquiry.
- ..you add a informative signature.
- ..you add at least university and team name .

In addition to the email contact the Formula CarMaker support features various tutorials and handbooks. It is recommendable to browse through these documents as they provide many answers of fequently asked questions. Furthermore they are available round the clock and might solve your problem earlier than a contact via email can do.

Further development of the programs

To enhance the development of CarMaker, IPGKinematics and other IPG products, the IPG Automotive GmbH is looking forward to any information about errors and potential improvements. The Formula CarMaker Support Team truly appreciates any kind of suggestion and incitement!

Remark to the used versions and datasets

This document refers to the following programs and versions: CarMaker 6.0.2, IPGKinematics 3.6.7 and Matlab R2012a.
All used datasets such as the TestRun "FS_CM6_BrakeSlalom" ran without problems in these versions and can be downloaded from the member area (FS_Generic_2018) on the IPG website.

Appendix A

Bibliography

Literature

[CMR07] IPG Automotive GmbH: CarMaker Reference Manual. IPG Automotive GmbH, 2014

[CMU07] IPG Automotive GmbH: CarMaker User's Guide. IPG Automotive GmbH, 2014

[HE07] Heißing, E.; Estoy, M.: Fahrwerkhandbuch. Wiesbaden: Vierweg-Verlag, 2007

[KIN07] IPG Automotive GmbH: IPGKinematics Handbuch. IPG Automotive GmbH, 2014

[RB00] Reimpell, J.; Betzler, J..: Fahrwerktechnik: Grundlagen. Würzburg: Vogel-Verlag, 2000

[Rei82] Reimpell, J.: Fahrwerktechnik 1. Würzburg: Vogel-Verlag, 1982

[Rei84] Reimpell, J.: Fahrwerktechnik: Lenkung. Würzburg: Vogel-Verlag, 1984

[RH84] Rompe, K.; Heißing, B.: Objektive Testfahrten für die Fahreigenschaften von Kraftfahrzeugen. Köln: Verlag TÜV Rheinland GmbH, 1984

[SAE07] Society of Automotive Engineers: Formula SAE Rules. Society of Automotive Engineers, 2007

[TIR07] IPG Automotive GmbH: IPGTire Tutorial. IPG Automotive GmbH, 2007

Websites

[1] http://www.formulastudent.de

[2] http://www.millikenresearch.com/fsaettc.html

[3] www.optimumg.com

Appendix B

Data files

B.1 Example of a TYDEX code

```
!# $Id: TUTO_TYDEX_TEST.tdx,v 1.2 2005/06/30 14:02:50 cr Exp $
**HEADER
!---------!-----------------------------!---------!---------!
! Key ! Comment ! Unit ! value !
!---------!-----------------------------!---------!---------!
RELEASE Release of TYDEX-Format 1.3
SUPPLIER Data supplier DEMO
DATE Date 07/98
CLCKTIME Clocktime 12:05
**COMMENTS
! Demo with TIME measurement procedure
! New tyre used
**CONSTANTS
!---------!-----------------------------!---------!---------!
! Key ! Comment ! Unit ! value !
!---------!-----------------------------!---------!---------!
TESTRIG Test rig DEMO
LOCATION Location DEMO
MANUFACT Manufacturer of the tyre DEMO
IDENTITY Identity of the tyre DEMO
```

NOMWIDTH Nominal section width of tyre mm 195

ASPRATIO Nominal aspect ratio % 65

RIMDIAME Nominal rim diameter inch 15

RIMWIDTH Rim width inch 6.5

RIMPROF Rim profile J

REFSIDEW Reference sidewall DOT

TYWHASSB Tyre-wheel assembly REFSWLEFT

TRDDEPB tread depth before mm 8

TRDDEPA tread depth after mm 5.4

TRCKSURF Surface of track SAFETYWALK

AMBITEMP Ambient temperature deg C 25

PATHRADC Path radius m INFINITY

INFLPRES Inflation pressure bar 2.5

TRCKCOND Track condition DRY

OVALLDIA Overall diameter mm 580

KROLRAD kinematic roll radius mm 280

**MODELPARAMETERS

!---------!-----------------------------!---------!---------!----!

! Key ! Comment ! Unit ! value !factor!

!---------!-----------------------------!---------!---------!----!

ITVS Vertikale Reifenfeder 350000.

ITVD Vertikaler Reifendaempfer 1000.

ITRLLO Relaxationslaenge laengs 0.05

ITRLLA Relaxationsl. quer 0.1

ITSSCLO StillStandskoeffizient long 0.01

ITSSCLA StillStandskoeffizient quer 0.01

ITRORETL roll resistance Trq/Load 0.01

**MEASURCHANNELS

!---------!-------------------------!---------!--------!-------!--

! Key ! Comment ! Unit ! factor ! offset!offset

!---------!-------------------------!---------!--------!-------!--

!

!physical value [Unit]= factor*(measured value + offset1) + offset2!

FZH Vertical force N 1 0 0

SLIPANGL Slip angle deg 1 0 0

INCLANGL Inclination angle deg 1 0 0

FYH Lateral force N 1 0 0

MZH Aligning moment Nm 1 0 0

**MEASURDATA

1211.00 -0.50 0.00 211.21 -2.27

3021.70 -0.50 0.00 460.01 -8.08

4828.00 -0.50 0.00 641.08 -15.50

6635.91 -0.50 0.00 748.59 -23.11

8444.54 -0.50 0.00 782.56 -29.70

8451.64 -1.00 0.00 1650.01 -74.74

6640.98 -1.00 0.00 1549.56 -55.52

4835.30 -1.00 0.00 1296.28 -34.95

3023.70 -1.00 0.00 883.88 -16.36

1217.11 -1.00 0.00 359.97 -3.51

1210.54 0.01 0.00 15.75 0.29

3022.89 0.00 0.00 -31.05 2.83

4828.88 0.00 0.00 -80.67 7.00

6641.35 0.00 0.00 -106.23 11.94

8447.86 0.01 0.00 -130.66 18.29

8447.03 0.51 0.00 -1014.16 64.33

6643.04 0.50 0.00 -927.98 46.09

4837.24 0.50 0.00 -763.21 28.49

3021.81 0.50 0.00 -478.76 13.10

1211.60 0.50 0.00 -143.84 2.69

1213.37 1.00 0.00 -305.18 4.09

3024.83 1.00 0.00 -890.32 20.23

4830.25 1.00 0.00 -1383.04 45.36

6639.96 1.00 0.00 -1677.46 75.28

8446.77 1.00 -0.01 -1816.43 105.83

8439.81 0.00 -3.04 323.53 28.80

4827.96 0.00 -3.03 216.27 17.45

1209.10 0.00 -3.02 99.88 6.40

1213.58 0.00 -1.00 72.70 2.71

4829.41 0.00 -1.00 45.71 10.10

8438.02 0.01 -1.00 39.30 21.39

8454.52 0.00 1.01 -213.57 13.88

4838.03 0.00 1.01 -108.97 3.08

1213.67 0.00 1.01 32.59 -2.27

1216.10 0.00 3.03 -7.26 -5.38

4843.10 0.00 3.02 -259.95 -4.09

8447.27 0.01 3.02 -467.54 6.22

6650.47 0.00 5.03 -561.46 -6.26

4839.59 -0.50 3.00 470.52 -28.39

3027.08 -0.50 3.00 355.43 -18.89

1210.09 -1.00 0.98 377.33 -6.01

1211.40 1.01 -1.04 -294.24 6.04

3013.27 0.51 -3.05 -314.68 23.15

4821.91 0.50 -3.04 -496.45 40.38

6632.49 0.00 -5.05 466.31 30.36

6204.67 -1.11 1.42 1596.27 -65.11

4671.47 1.11 -1.42 -1385.77 52.51

4676.80 -1.11 1.42 1345.09 -44.21

6192.29 1.11 -1.43 -1665.13 79.80

8480.00 -9.68 5.56 6653.04 -70.43

2398.14 9.69 -5.58 -2400.98 0.98

2387.44 -9.69 5.57 2326.81 -5.20

8475.57 9.70 -5.57 -6594.03 67.88

6950.80 -2.12 2.82 2646.00 -128.43

3882.44 2.12 -2.83 -1856.83 57.17

3895.17 -2.12 2.82 1863.24 -56.02

6938.60 2.12 -2.83 -2774.20 140.19

9285.47 -12.11 5.91 7232.49 -48.49

1640.53 12.12 -5.93 -1672.14 -0.80

1638.45 -12.12 5.92 1657.14 -0.70

9281.01 12.13 -5.92 -7192.70 49.90

7713.80 -5.05 4.22 5267.92 -193.88

3121.37 5.04 -4.23 -2620.77 38.67

3122.04 -5.04 4.22 2648.05 -41.97

7693.48 5.05 -4.23 -5236.87 186.65

4367.24 -2.32 0.61 2292.76 -57.70

1663.58 2.32 -0.62 -834.64 11.48

1662.51 -2.32 0.61 904.85 -11.56

4362.46 2.32 -0.62 -2380.05 61.50

6174.07 -9.34 1.39 5559.15 -26.00

773.02 9.35 -1.41 -915.13 0.99

798.32 -9.34 1.40 926.42 -2.22

6179.77 9.35 -1.40 -5566.71 16.28

4819.49 -4.12 0.81 3842.98 -81.60

1213.00 4.12 -0.82 -1007.44 7.96

1212.24 -4.12 0.81 1009.89 -7.99

4821.89 4.13 -0.82 -3861.91 74.69

5271.42 -5.83 1.00 4830.02 -72.66

770.04 5.83 -1.01 -804.12 4.11

763.44 -5.83 1.00 767.93 -3.31

5278.38 5.84 -1.01 -4780.28 58.53

8458.68 -12.04 -0.01 6912.70 -18.90

1199.51 12.05 0.00 -1330.46 -1.45

1198.00 -12.05 0.00 1343.88 1.13

8459.97 12.06 0.00 -6950.92 10.93

6649.84 -5.83 -0.01 5764.30 -102.48

1808.09 5.83 0.00 -1829.74 8.55

1806.19 -5.83 0.00 1823.33 -9.80

6649.88 5.84 0.01 -5683.13 86.77

7257.76 -8.34 -0.01 6394.02 -51.62

1198.32 8.34 0.00 -1353.07 1.28

1198.65 -8.34 0.00 1366.09 -1.91

7259.76 8.35 0.01 -6374.97 42.24

1207.40 -12.06 -1.98 1370.53 3.10

7237.09 12.06 1.98 -6209.88 -1.17

5444.26 -1.71 1.61 2181.08 -78.76

3012.59 1.72 -1.63 -1390.39 34.41

3011.54 -1.71 1.61 1422.13 -33.04

5438.58 1.72 -1.63 -2243.23 85.30

7262.97 -9.76 3.97 6109.24 -40.52

1209.37 9.78 -3.99 -1327.95 2.74

1198.86 -9.77 3.98 1268.81 -1.82

7269.73 9.77 -3.98 -6200.20 42.36

6035.69 -3.62 2.41 4085.05 -136.36

2396.46 3.63 -2.43 -1914.02 29.52

2396.28 -3.62 2.42 1937.77 -29.37

6025.61 3.63 -2.43 -4096.58 130.20

6655.31 -6.74 3.20 5750.83 -91.56

1809.13 6.75 -3.22 -1887.68 10.09

```
1801.47 -6.74 3.21 1851.49 -9.41
6646.61 6.75 -3.21 -5712.57 87.20
**END
```