Date:	2019-06-12
Author:	Jasvir S. Dhillon
Release No.:	CM 7.x – 8.x

Modelling Ride Height Effects with User-defined Aero C-Code Model

Here, we discuss how a User-defined Aero model can be implemented using the C-code interface with CarMaker. The main objective being to model the effects of changes in ride height on the lift and drag force of the vehicle. The built example uses generic aero-coefficients and force calculation equations which provide a base for you to build on your own model.

Technical Background

The front and rear ride heights of the vehicle strongly affect the aerodynamic lift forces, as well as the net drag force acting on the vehicle to a certain extent. As ride height changes at front and rear, it changes the distance and profile between the underbody of the vehicle and the ground surface, thus affecting the airflow underneath. This airflow is a major contributor to the net lift/downforce on the vehicle. Since, ride height is dynamic the lift and drag forces change dynamically too, when the vehicle is on the move. This makes the modelling of ride height effects on aerodynamics of a vehicle important and vital, especially in cases of high-performance and race vehicles.

Solution

The necessary code for the User Aero model needs to be used inside the build environment of CarMaker.

As the current model comprises of a large number of variables and equations, the entire code is built inside a separate new C-file ‘ExtendedAero.c’ and its corresponding header file ‘ExtendedAero.h’ is used to declare the register function.

The ‘ExtendedAero.c’ file itself describes the various steps involved in building the code in good detail. The following provides an overall summary of the process followed to create the Aero model and integrate it with CarMaker. (Kindly cross-refer to the provided c-file as you proceed with the explanation below)

A. Header files and new variables

A set of header files existing inside the CarMaker installation directory required for the execution of the User code are included at the start in the ‘ExtendedAero.c’ file. The set of variables required to build the code are then defined next.

B. Defining CarMaker UAQs

In order to view and analyse the C-variables inside the CarMaker environment (for eg. in IPGControl), the variables need to be assigned to new User Accessible Quantities (UAQs) using the ‘DDef’ function. It is up to the user to choose whether these UAQs get Direct Variable Access (DVA) or not. If this is activated, the user would be able to modify the values of the UAQs online during the simulation. In this example, there is no access point given to any of the created UAQs (DVA_NONE).

C. Adding aero coefficient maps inside vehicle infofile and accessing them in C-code

Depending upon the aerodynamic build of the vehicle under consideration, the corresponding drag and lift coefficients at the front and rear would change with changing ride height. The existing example uses three sets of coefficients: Static, dependent on ride height of each axle, interdependent on ride heights of each axle.

Since, these are specific to the vehicle model they have to be mentioned inside the vehicle infofile. This is done inside the 'Additional Parameters' of the vehicle infofile.

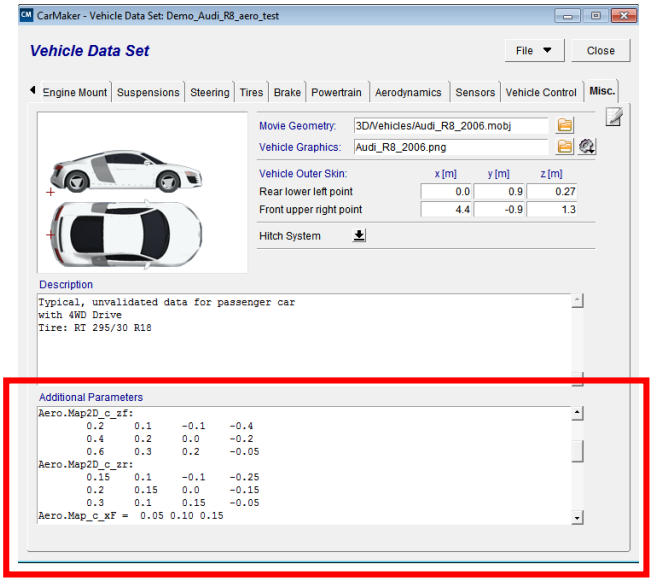

Fig 1: Additional Parameters required for User-Aero Model as also seen in the GUI

In our example, the 2D maps of coefficients have been created by a combination of front ride height vector ('Aero.vector_rh_f') and the rear ride height vector ('Aero_vector_rh_r').

Since, during simulation the ride height will have values which are not directly described by the input points on the maps, the in-between values are interpolated using a linear function. CM C-functions called 'LMEval' and 'LM2DEval' are used for this purpose.

D. Calculation of Aero Forces

Assuming that the tyres stays in contact with the ground at all times, it is considered that the wheel centre will move up by the same amount the vehicle body will go closer to the ground surface. Additionally, there is a pre-existing static offset at the start ('Aero.rh_offset_f' and 'Aero.rh_offset_r') between the wheel centre and the point on the vehicle body where the ride height is measured.

The ride height at each axle is calculated by taking the mean values the ride heights at the left and the right tracks of that particular axle. The general formula put simply is as follows:

$$\begin{aligned} \boldsymbol{RideHeight_{axle}} \\ &= \mathbf{0.5} \\ &* \left[(\boldsymbol{WheelRadius} - \boldsymbol{BodyTranslation_z})_{\boldsymbol{left}} \right. \\ &\left. + (\boldsymbol{WheelRadius} - \boldsymbol{BodyTranslation_z})_{\boldsymbol{right}} \right] - \boldsymbol{RideHeight\_offset_{axle}} \end{aligned}$$

Once the ride heights for front and rear are calculated, the corresponding lift and drag coefficients are obtained by linear interpolation as mentioned in the previous step.

The Aerodynamic forces are then calculated using the following standard equation:

$$\boldsymbol{AeroForce = 0.5 * rho * v^2_{realtiveWind} * Area_{frontal} * C}$$

Where,

rho – air density ; C – Coefficient of lift/drag ; v_relativeWind – true relative wind speed wrt vehicle

The net lift and drag forces calculated are then applied on the POA (Point of Attack) which can be defined by the user using the vehicle infofile parameter 'Aero.pos_CarMaker'.

E. Function to register ExtendedAero model

The last step in the code is used to define the register function in the correct Model Class 'Aero'. This is important since this will make the compiled aero model available for selection inside the Aerodynamics section.

F. Building the executable

Make sure that the 'ExtendedAero.c' and 'ExtendedAero.h' are placed inside the 'src' folder of the CM project directory. Inside the 'Makefile', mention the object file name 'ExtendedAero.o' for its inclusion. Please refer to section 5.2.3 'Integrating C-Code Models/Demonstration Examples / Preparation' of the Reference Manual for more details.

G. Simulation

Please refer to section 5.2.4 'Integrating C-Code Models/Demonstration Examples/Running the Model' of the Reference Manual for the instructions.

Run the simulation! You will see that a new set of signals (as defined in the 'ExtendedAero.c' file) is created which can be viewed and analysed inside IPGControl.

H. Further suggested work

1. Since in this example, the ride height is mathematically calculated, it could alternatively be done by adding a sensor on the vehicle body similar to a real world test-vehicle. Then by continuously reading the distance to the ground (ride height) at points of interest, this could be used as the input for the aerodynamic forces calculation.

2. The current example uses a map of lift and drag coefficients without considering the different possible approach angles (approach angle is '0'). Further steps could also involve calculating aerodynamic forces by including this dependency.