cosin
scientific software

cosin scientific software AG

Hans-Stützle-Str. 20
81249 München
GERMANY

info@cosin.eu
www.cosin.eu

FTire/sim

FTire Stand-Alone Simulation
Documentation and User's Guide

Contents

Document Revision: 2022-1-r26108

General Remarks

This documentation provides details of the stand-alone simulation environment **FTire/sim**. For more material about **FTire** and other tire simulation tools, please visit www.cosin.eu.

1 FTire and QC Stand-Alone Simulation with FTire/sim

The simulation environment **FTire/sim** allows for easy-to-run simulations with **FTire** alone or with the combination **FTire** together with a detailed nonlinear quarter-car mode (QC). Like all cosin products, **FTire/sim** is available for Windows, Linux, and MacOS. **FTire/sim** uses up to 7 input-files, and 2 output-files. Apart from the plot-data file, all files are in **cosin/io** syntax. Of course, the input files can be merged to fewer files. In the end, just one input file can contain all data:

- the cfd-file, that allocates the logical files `ISIM`, `ISRC`, `ISTR`, `IPAR`, `IPPAR`, `IQPAR`, `OPPAR`, `OPLOT` (cf. *cosin/io* User's Guide),
- the simulation data file (file identifier `ISIM`), which defines basic simulation data (cf. section 3.1),
- the source data file (file identifier `ISRC`), which defines time histories of all external input variables ('sources'),
- the road data file (file identifier `ISTR`), which defines the road profile,
- the **FTire** basic parameter file (file identifier `IPAR`), which contains **FTire** basic parameters described in section 1.4,
- the **FTire** preprocessed parameter file (file identifier `IPPAR`), which keeps preprocessed **FTire** parameters. By using this file, if **FTire** basic data haven't changed from simulation run to simulation run, some CPU-time can be saved,
- the *QC* parameter file (file identifier `IQPAR`), which contains *QC* parameters described in section 2.2,
- the **FTire** preprocessed parameter file (file identifier `OPPAR`), which is calculated during pre- processing, for use in subsequent simulation runs,
- the plot data file (file identifier `OPLOT`), to be viewed in one of the plot-programs **cosin/lean-plot** (Windows version) , **cosin/ip** (Unix version) , or in Matlab or Excel.

Normally, file with identifiers `ISIM`, `ISRC`, `ISTR`, are merged to one simulation file.

1.1 Simulation Data

In the simulation file (identifier `ISIM`), *FTsim* looks for the data block `$simulation`, and looks for the following data in this block:

Name in input file	Unit	Meaning
`preprocessing`	-	0 (default): try to read preprocessed data in `IPAR`- file. If not successful, perform preprocessing 1: always perform pre-processing, even if there are data available
`simulation_mode`	-	0 (default): only *FTsim*. Source data directly define hub-carrier position and velocity 1: **FTire** + *QC*. Source data define car-body position and velocity
`animation_mode`	-	0 (default): no animation (*FTsim* runs fastest) 1-9: animation with different viewing angles; 101-109: same as 1-9, but preparing the generation of an avi movie file (attention, this feature temporarily takes a lot of disk space!)

Name in input file	Unit	Meaning
`simulation`	s	an array of three values: 1: simulation starting time [s], normaly 0 2: time step [s], normally something like 0.002 .. 0.004 3: simulation final time [s]
`plot_output`	s	an array of three values: 1: starting time of plot output [s] 2: time step of plot output [s] 3: final time of plot output [s] If final time is less than start time, plot output is com- pletely suppressed
`plot_output_start_time`	s	starting time in plot annotation. This time may be defined independent on simulation starting time and plot output starting time. If not defined, plots will start at `plot_output(1)` or `simulation(1)`, resp.
`animation`	s	an array of three values: 1: starting time of animation [s] 2: time step of animation [s] 3: final time of animation [s] If final time is less than start time, or `anima- tion_mode = 0`, animation is completely suppressed
`animation_zoom`	-	zoom-factor for animation. You also can zoom in and out during animation, by dragging rectangles with the mouse. Default value 1.0
`plot_file_format`	-	a character-string out of `standard dsp dspd matlab matrix excel gnuplot` which defines the format of plot-output (cf. **cosin/ip** User's Guide)
`frame`	-	defines the frame to be used for plot output values of forces and torques: 0 (default): TYDEX C 1: TYDEX H 2: TYDEX W 3: ISO contact frame 9: all frames 10: TYDEX C, using kN and kNm 11: TYDEX W, using kN and kNm
`road_file`	-	file to be scanned for **cosin/ev** road definition data. Default is the simulation file
`obstacle`	-	string to be used instead of the default road data- block name `road_type`

1.2 Input Variables (Sources)

To define the external input variables, Sources&Sinks-routines of **cosin/io** are used, cf. *cosin/io* User's Guide, section 1.5). With these input variables, position and velocity of car-body or hub- carrier, resp. are defined, thus controlling directly or indirectly the movement of the rim. In the sources file (identifier `ISRC`), *FTsim* looks for the data block `$sources`, and interprets this block to define the following sources:

Name in input file	Unit	Meaning
`rolling_speed`	m/s	vehicle speed (`rolling_speed = 0` allowed)
`tire_deflection`	mm	controls height of hub-carrier or car-body, resp. Tire deflection exactly is prescribed by this input variable only if (a) road is even with height zero (b) simulation mode is **FTire** without *QC*
`wheel_load`	kN	approximately prescribes static wheel load. Overrides `tire_deflection`. On the basis of `wheel_load` , FTsim estimates a related value for tire deflection. At high rolling speeds, wheel load will be underestimated to a certain amount (if `wheel_load_regulated = 0`). In this case, the user should choose a somewhat smaller value and check the resulting actual wheel load in the plot-file. Similarly, for large camber an- gles, wheel load might be overestimated and should be corrected by choosing a greater value. Alterna- tively, by choosing `wheel_load_regulated = 1`, wheel load can be regulated to nearly exactly and automatically take the prescribed value
`wheel_slip`	%	prescribes longitudinal slip. Only in effect if `free_spinning = 0` and `wheel_revs` is not de- fined. Positive for driving, negative for braking
`wheel_revs`	rad/s	wheel rotational speed. Overrides `wheel_slip`. Only in effect if `free_spinning = 0`

Name in input file	Unit	Meaning
`side_slip_angle`	deg	slip angle. Positive if steady-state side-force is positive, ie. lateral sliding velocity is negative. Only applicable if `rolling_speed > 0`. Otherwise, use `toe_in_angle` to apply side-force
`drive_torque`	Nm	prescribes driving torque. Only applicable if `free_spinning = 1`. May be positive or negative
`brake_torque`	Nm	prescribes braking torque. Only applicable if `free_spinning = 1`. Should be positive. Correct sign is opposed to that of wheel rotational speed and automatically observed in **FTire**
`toe_in_angle`	deg	toe-in-angle to prescribe slip angle on a test drum, when `rolling_speed = 0`. Positive toe-in results in positive side-force. Rim toe-in angle exactly is prescribed by this input variable only if simulation mode is **FTire** without *QC*
`camber_angle`	deg	camber angle. Positive if positive rotation along x-axis is applied to hub-carrier (that is: `camber_angle` normally is positive for left wheel and negative for right wheel, in design position). Rim camber angle exactly is prescribed by this input variable only if si- mulation mode is **FTire** without *QC*
`free_spinning`	-	0: prescribe wheel rotation velocity 1: prescribe driving and braking moments and integrate wheel rotation in **FTire**
`wheel_load_regulated`	-	0: tire deflection estimated from radial characteristic 1: tire deflection chosen by wheel-load regulator **Remark:** wheel load can alternatively be regulated by using the **cosin/ar** PID controller function, described in the **cosin/io** documentation

1.3 Road Profiles and Obstacles

To specify road excitation, **FTire/sim** uses the package *cosin/road*. This package is described in a separate manual.

2 Quarter-Car (QC)

A simple 'Quarter-Car' (QC) model is used in **FTire/sim** as a test 'vehicle model' to aid in evaluating the performance of **FTire**.

2.1 Modeling Approach

QC comprises a model of the hub-carrier, with 3 degrees of freedom: longitudial, lateral, and verti- cal displacement with respect to the car-body. The car-body itself, by default, has no degree of freedom. Instead, its position and velocity is controlled through certain source variables, that is, through external inputs to the simulation model (cf.. section 3.4 and cosin/io User's Guide). Alter- natively, the vertical displacement of the car-body can be given a degree of freedom by specifying a positive 'sprung mass'.

The hub-carrier is coupled to the car-body by

- the condition 'no rotation relative to car-body',
- a nonlinear suspension spring that acts along a general direction vector, with a certain transmission ratio relative to wheel travel,
- a nonlinear shock-absorber that acts in the same direction as the suspension spring, but with a different transmission ratio. The shock-absorber carries a nonlinear elastic bearing, inner fric- tion, and a constant gas force,
- a linear longitudinal spring/damper combination, as well as a linear lateral spring/damper combination. Both work along a general direction vector,
- up to 5 friction elements. Every element consists of a linear spring/damper element which is itself in line with a dry friction element. The dry friction element is defined through two values, one for static friction force, and one for sliding friction force. Every friction element works along an individual direction vector. Transmission ratio for the friction element is defined to be 1.0.

2.2 QC Parameters

Name in input file	Unit	Meaning
sprung_mass	kg	portion of car-body mass that is supported by the wheel. If sprung_mass is not specified or set to a value less or equal to zero, no degree of freedom is assigned to the car-body
unsprung_mass	kg	mass of hub-carrier, including suspension parts, rim, brake, etc., but excluding tire belt mass (cf. **FTire** basic parameters)
transmission_ratio_suspension_spring		spring deflection / wheel travel
transmission_ratio_shock_absorber		shock-absorber deflection / wheel travel
shock_absorber_gas_force	N	constant gas force of shock-absorber, tends to push hub-carrier down

Name in input file	Unit	Meaning
`suspension_direction_vector`	-	vector with three components in hub-carrier-fixed frame, indicating the direction of the suspension spring and shock-absorber force **Remarks**: 1. direction vector needn't be normalized; this is done automatically by *QC* 2. point of attack is not relevant, because moments do not affect hub-carrier (no ro- tation!)
`long_stiffn_hubcarrier`	N/mm	linear stiffness between car-body and hub-carrier, essentially in longitudinal direction
`long_damping_hubcarrier`	Ns/mm	linear damping between car-body and hub-carrier, essentially in longitudinal direction
`long_stiffn_direction_vector`	-	vector with three components in hub-carrier-fixed frame, indicating the exact direction of the longitudi- nal stiffness and damping
`lat_stiffn_hubcarrier`	N/mm	linear stiffness between car-body and hub-carrier, essentially in lateral direction
`lat_damping_hubcarrier`	Ns/mm	linear damping between car-body and hub-carrier, essentially in lateral direction
`lat_stiffn_direction_vector`	-	vector with three components in hub-carrier-fixed frame, indicating the exact direction of the lateral stiffness and damping
`number_of_friction_elements`	-	number of friction elements (between 0 and 5)
`friction_element_i (i=1,...5)`	div	data of friction elements. `friction_element_i` consists of 7 components, that may all reside in one or, equally well, in several consecutive lines of the input-file (cf. **cosin/io** User's Guide): 1: stiffness [N/mm] 2: damping [Ns/m] 3: static friction force [N] 4: sliding friction force [N] 5-7: direction vector [-]
`$suspension_spring`	mm, N	spline data block for suspension spring characteristic. Contains data pairs in ([mm], [N]) for spring deflection and spring force. Pre-load is contained in the charac- teristic

Name in input file	Unit	Meaning
`$shock_absorber`	m/s, N	spline data block for shock-absorber characteristic. Contains data pairs in ([m/s], [N]) for shock-absorber deflection velocity and shock-absorber force. Coulomb friction is contained in the characteristic
`$shock_absorber_bearing`	mm, N	spline data block for shock-absorber bearing charac- teristic. Contains data pairs in ([mm], [N]) for bearing deflection and bearing force. Pre-load is contained in the characteristic