UNIVERSAL MECHANISM 10

User`s manual

Simulation of road vehicles

Simulation of road vehicle dynamics is considered

2024

Contents

1. UM Module for simulation of road vehicles

1.1. General information

Program package Universal Mechanism includes a specialized module **UM Automotive** for analysis of vehicle dynamics. The module includes additional tools integrated into the program kernel as well as libraries of typical suspension elements and transmissions, which are delivered separately. UM Automotive contains the following main components:

- tools for generation and visualization of track macro geometry;
- tools for generation and visualization of track micro profile (irregularities);
- library of files with road irregularities as well as power spectral density files;
- mathematical models of tire forces (tire/road contact forces);
- driver models;
- set of typical dynamic experiments.

UM Automotive allows the user to solve the following problems:

- estimation of vehicle vibrations due to irregularities;
- estimation of vehicle dynamic performances on various maneuvers;
- parametric optimization of vehicle elements according to various criteria;
- analysis of influence of transmission on stability and handling of vehicle;
- calculation of characteristics that determine the holding of the road by the vehicle in controlled and uncontrolled motion.

1.2. Base system of coordinates

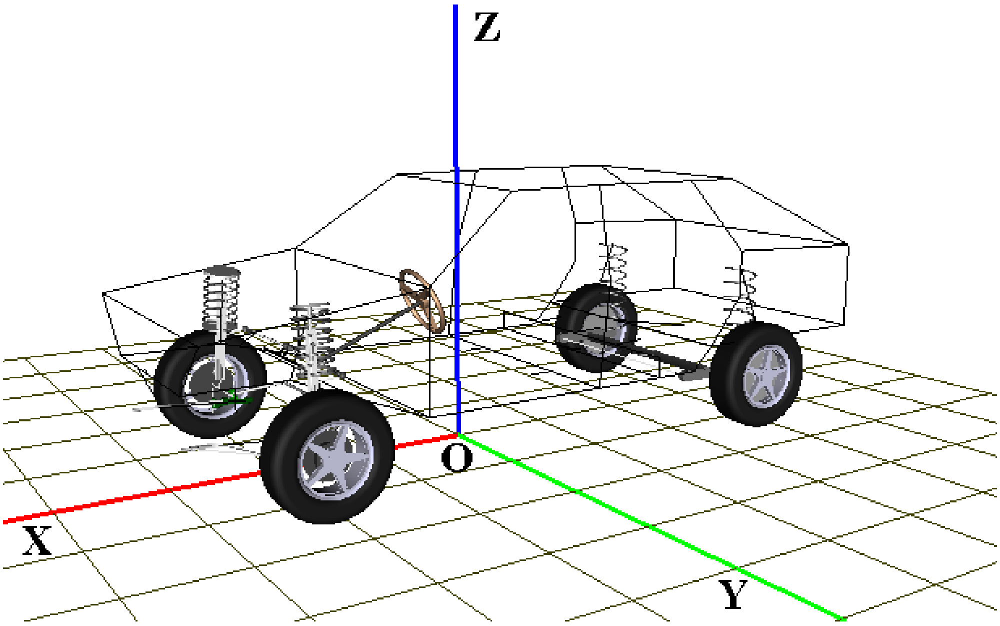


Figure 1.1. Base system of coordinates (SC0)

Inertial system of coordinates (SC0) in UM Automotive meets the following requirements (Figure 1.1):

- axis Z is vertical, axis X coincides with the vehicle longitudinal axis at its ideal position at the moment of motion start;
- origin of SC0 lies at the ideal road level.

1.3. Track macro and micro profiles

Track profile can be composed of three components: macro profile, micro profile and asperity, which exert different influence on the vehicle dynamics.

The *vertical macro profile* consists of smooth long vertical irregularities (wave length of 100 meters and more), it does not practically affect the vehicle vibrations but essentially influences the vehicle dynamics, regimes of engine and transmission. The *horizontal macro profile* contains description of a desired vehicle horizontal trajectory (path) for simulation of maneuvers. Two methods are available in UM for description of the macro profile^

- using two curves that define the horizontal and vertical profiles of the trajectory followed by the driver when the car is moving, and a curve that defines the slope of the road, Sect. 1.3.1 "*Defining a macro profile using curves*";
- using a triangulated 3D surface, Sect. 1.3.2 "*Vehicle movement on a triangulated surface (testing* area)".

The *micro profile* consists of vertical irregularities (wave length from 10 cm to 100 m), which excite vibrations of the vehicle suspension, but the profile does not contain long slopes, which change engine regimes.

The asperities (wave length less than 10 cm) are filtered by tires and do not excite vehicle vibrations. They affect the tire functioning (adhesion, wear, etc.).

1.3.1. Defining a macro profile using curves

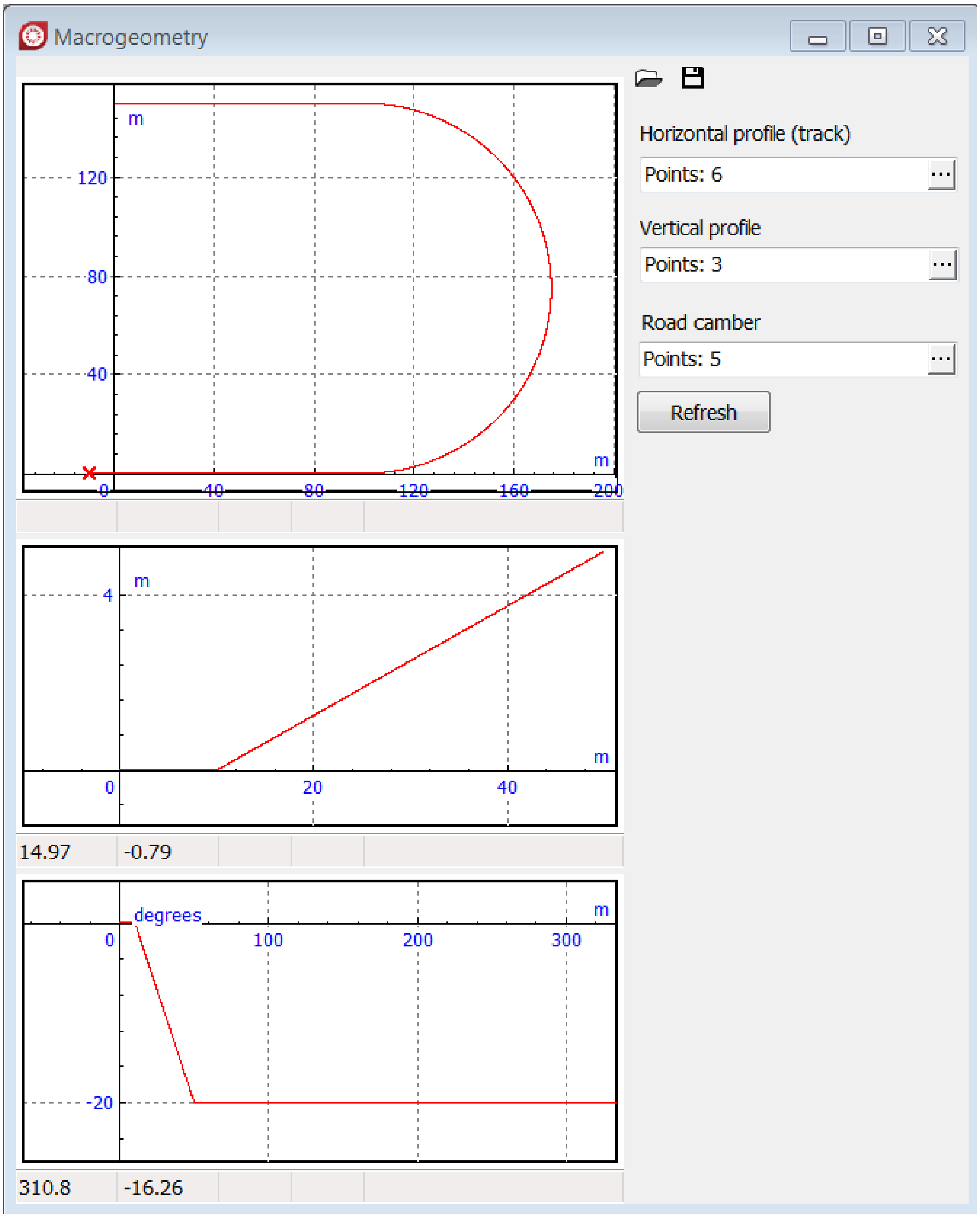


Figure 1.2. Wizard of macro geometry. Horizontal (upper plot), vertical (middle plot) profiles and road camber profile (lower plot)

Marco profiles are 2D curves consisting of a set of points connected by straight sections, circle arcs and splines. The horizontal macro profile is a set of (X_i, Y_i) coordinates on the path in SC0. The vertical profile is the set of points (Z_i, s_i), where Z_i is the vertical coordinate of the track in SC0, and s_i is the distance along the real trajectory of the vehicle (path coordinate). The profile of road camber is the set of points (γ_i, s_i), where γ_i is the camber angle of track (degrees).

Horizontal, vertical profiles are stored in *.*mgf* text files located by default in the **{UM Data}\car\macrogeometry** directory.

To generate a macro geometry file use the **Tools | Macrogeometry editor | Road for cars…** menu command. The wizard of macro geometry appears (Figure 1.2)

Curves of profiles are created in the curve editor by clicking the button (Figure 1.3). See Chapter 3, *Curve Editor* for more information.

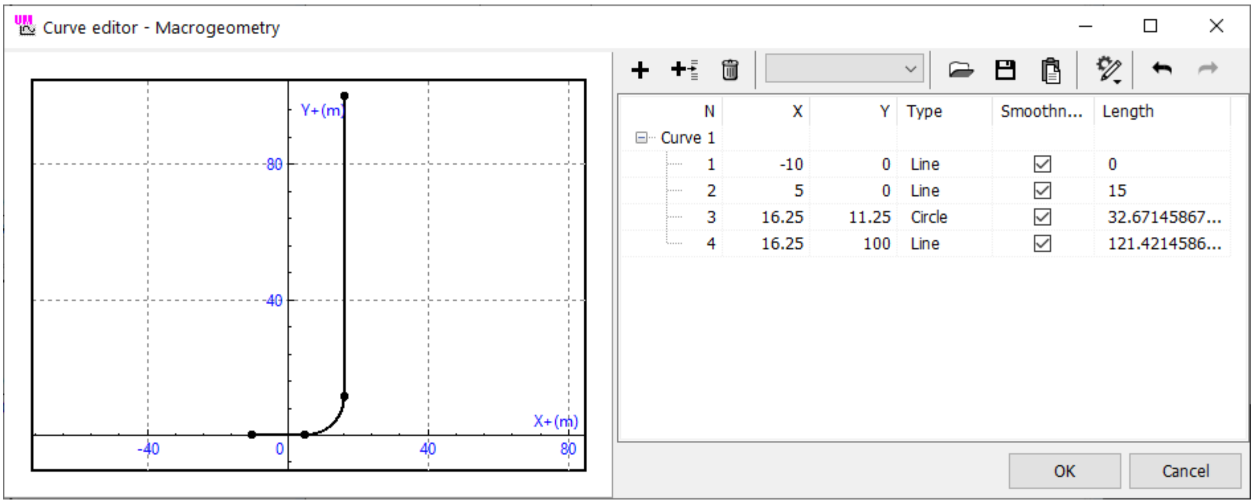


Figure 1.3. Curve editor

Use the **Refresh** button to synchronize the vertical and horizontal profiles. After clicking the button, a new horizontal profile is created with number of points equal to that for the vertical profile, and the path coordinate s_i is equal to distance along the vertical profile from initial point to point *i*.

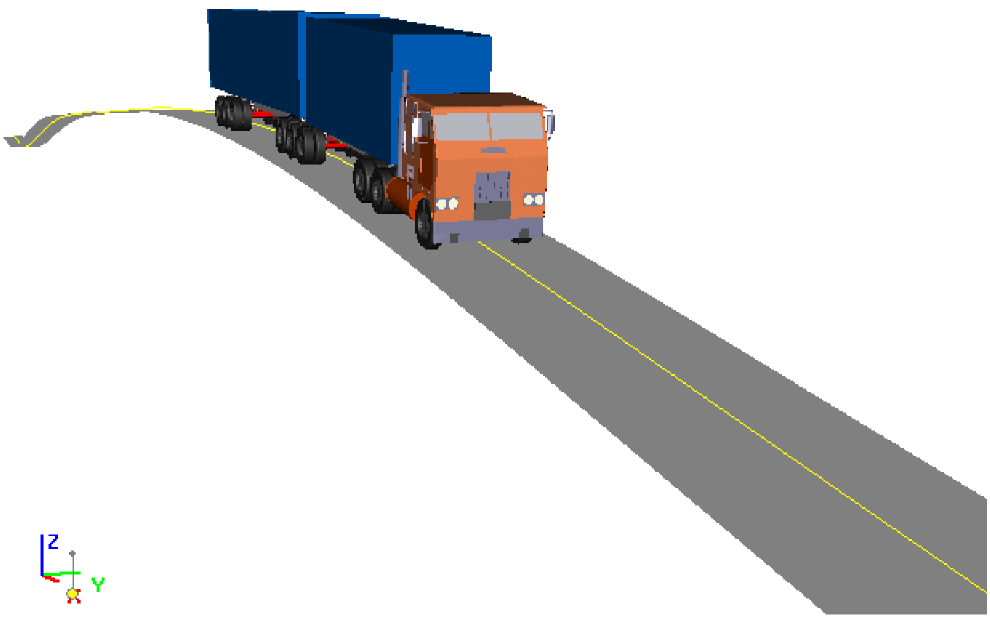


Figure 1.4. Vertical macro profile

Remark 1 It is recommended to locate the first point of the vertical profile at the origin (0, 0), and start the curve with the straight section along the X-axis.

Remark 2 The continuous driver model uses the derivative of the path following error (Sect. 1.4.3. *"Combination of PID controller and preview model"*, p. 1-40), which requires a differentiable function of the desired path. In this case a spline interpolation of the path curve is necessary.

1.3.2. Vehicle movement on a triangulated surface (testing area)

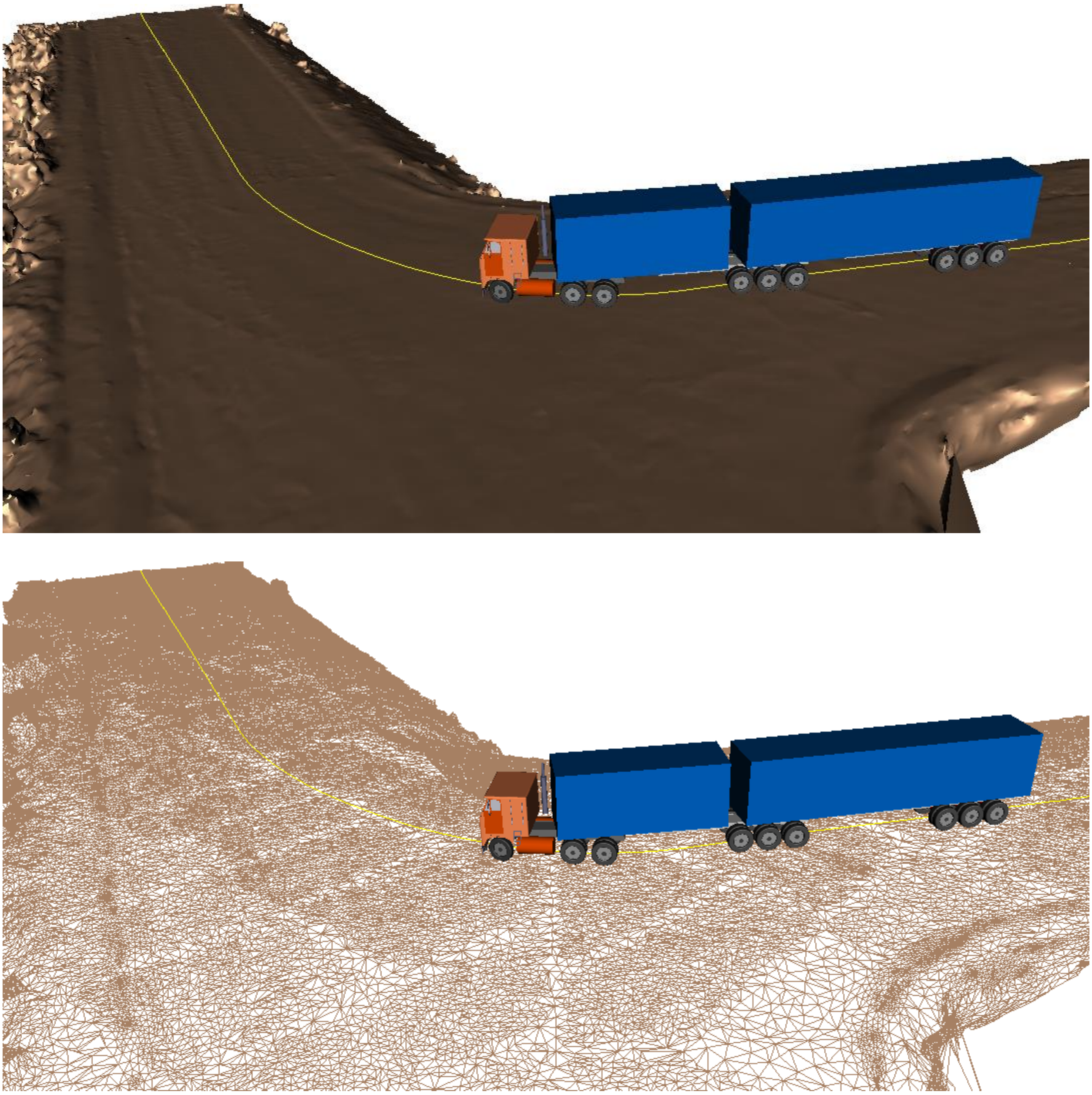


Figure 1.5. An example of a car moving along a triangulated road

Additional possibilities for creating a combination of a macro profile and a micro profile are provided by the triangulated surfaces (testing area), which appeared in UM 2023. The relief of the surface/road in this case is specified as a file containing a graphic image of the surface as a set of graphic elements of the ASC type (a set of triangles), Figure 1.5. The car follows a route -

a user-defined trajectory on the surface (yellow curve in Figure 1.5). When calculating the interaction of the wheel with the road, the actual geometry of the surface is taken into account. Terrain triangulation can be obtained using modern video recording tools.

Triangulated surface files are stored in the standard directory {UM Data}\Car\TestingArea\. The specific path depends on the installation of UM. Example:

c:\Users\Public\Documents\UM Software Lab\Universal Mechanism\2023\Car\TestingArea\

The standard configuration of UM 2023 contains three files for triangulating the roadbed and rough terrain:

- Road.img – example of a triangulated road surface with an intersection, Figure 1.5

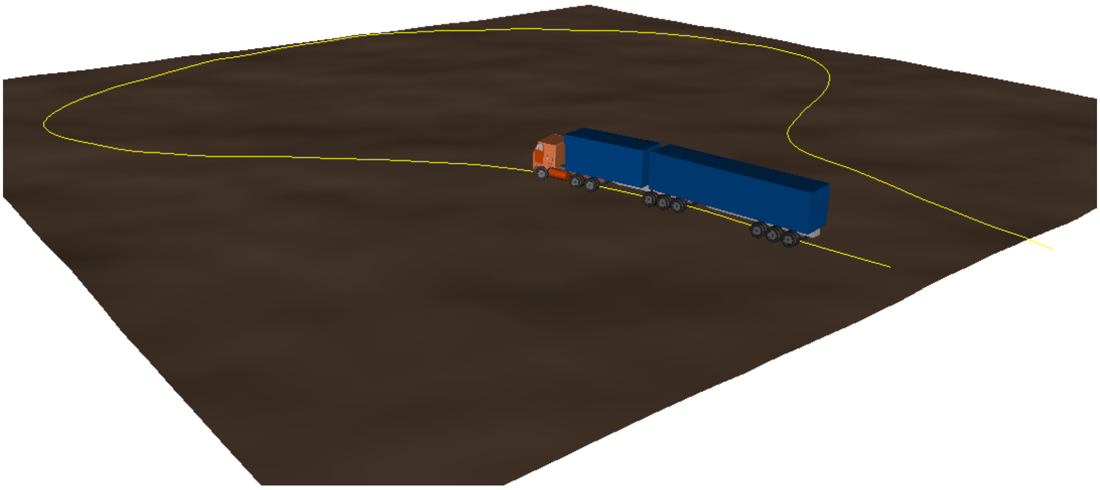


Figure 1.6. Square area of rough terrain

- Square testing ground.img – example of a square area of rough terrain;

Figure 1.7. Rough terrain with slope

- Surface with slope.img – rough terrain with a slope in the longitudinal and transverse directions.

1.3.2.1. Creating Triangulated Surface Files in UM Format

The surface must be previously created and saved in STL format. CAD programs or editors such as Blender 3D can be used to create them. To create a UM surface file

- run the UMInput.exe program;
- open the window for converting STL files using the menu command **Tools | Import from CAD | Files *.stl**.

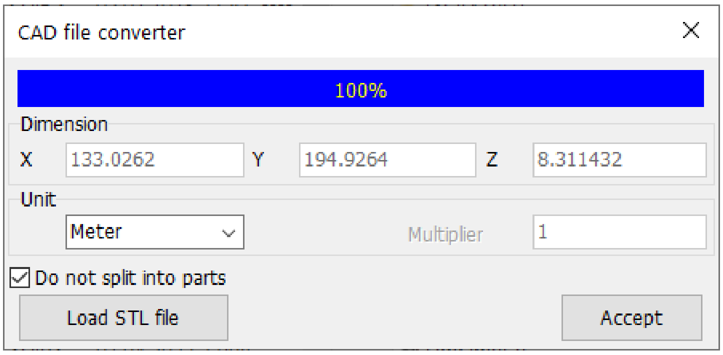


Figure 1.8. STL file converter

- open file with button **Load STL file (**Figure 1.8**)**; change the unit if necessary; enable the **Do not split into parts** option and perform the conversion using the **Accept** button.

Upon successful conversion, a UM model is created containing a graphical image of the surface, Figure 1.9.

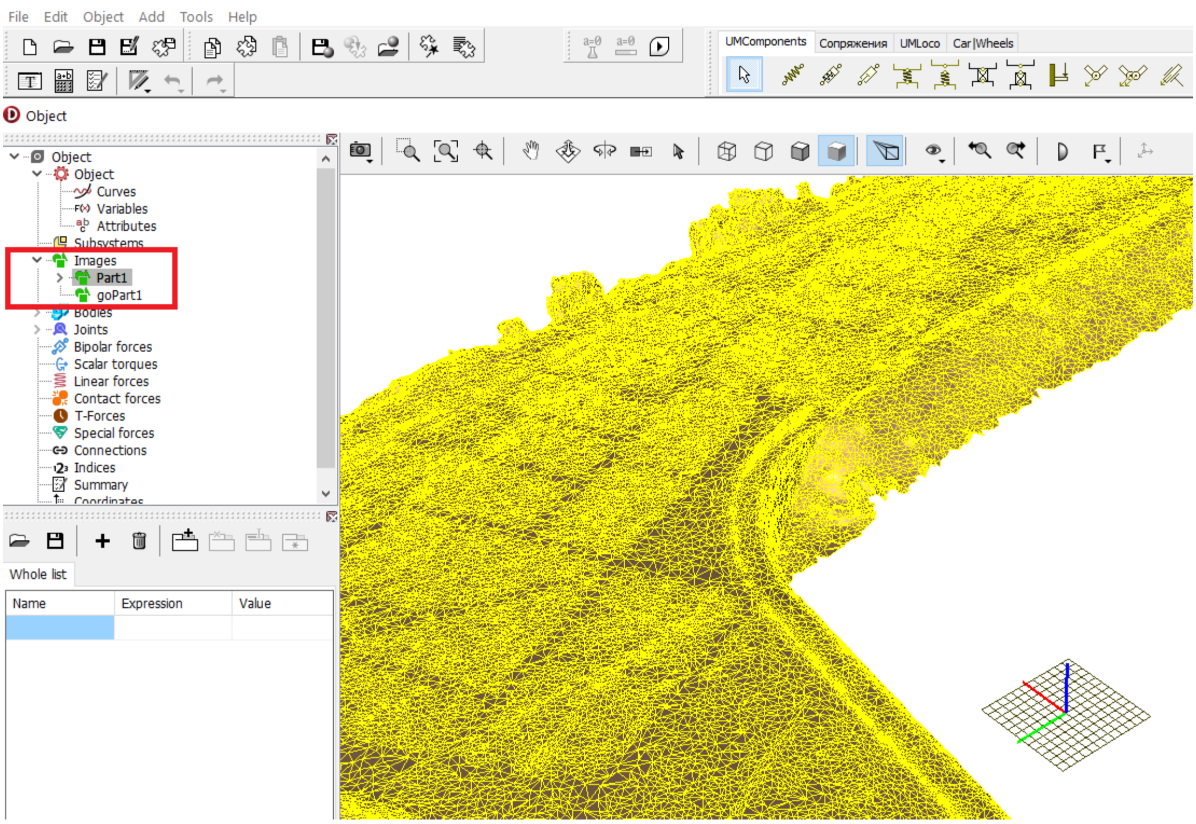


Figure 1.9. Graphical image of surface

To add the surface to the database, go to the Part1 graphical object (Figure 1.9) and save it in the standard folder with surface files of the automotive module using the button on the toolbar, Figure 1.10.

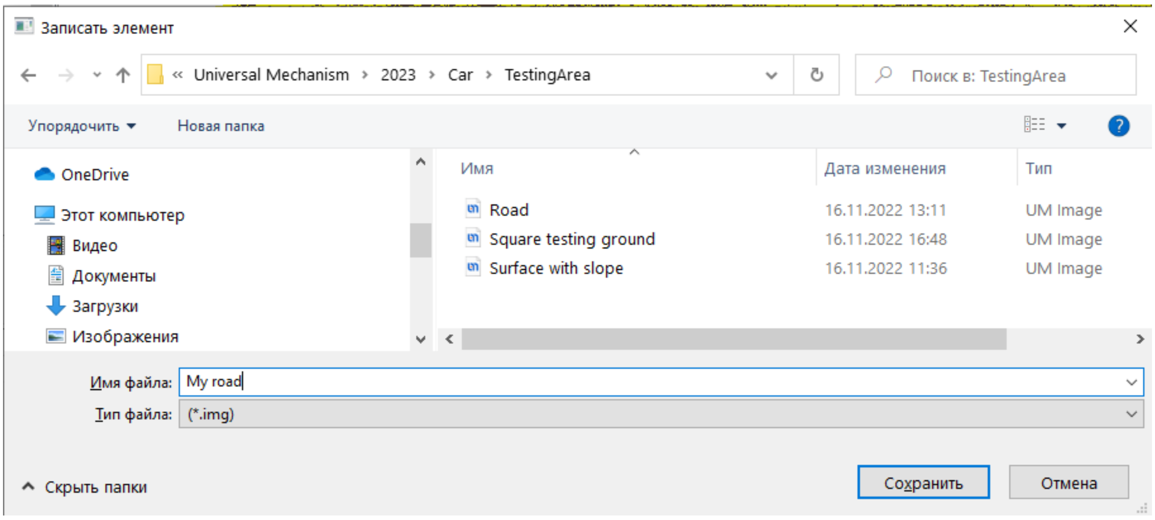


Figure 1.10. Saving the file under the name *My road* to the database of triangulated surfaces

1.3.2.2. Setting routes

To use a triangulated surface in tests with a driver, the user must specify a list of routes that the car will follow. To create and edit the list of routes

- start the simulation program UMSimul.exe;
- load any model of road vehicle;
- open the simulation inspector window and the tab **Road vehicle | Testing area**, Figure 1.11;

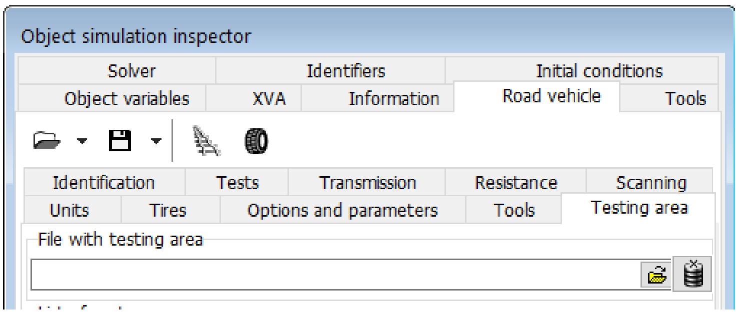


Figure 1.11. Tab for setting testing area and routes

Setting testing area

Use the button (Figure 1.11) to open a file *.img, which contains the triangulated surface image (testing area). If a list of routes has already been created for the selected file, it will be displayed in the table.

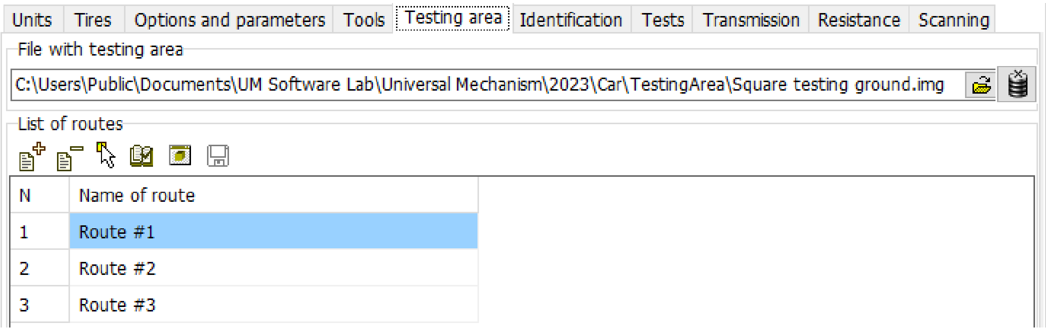


Figure 1.12. List of routes

Creating and editing routes

A route is a user-defined curve on a surface that a vehicle follows under the control of the built-in driver model

The following tools are used

Adding a new route to the list.

Deleting the selected route from the list.

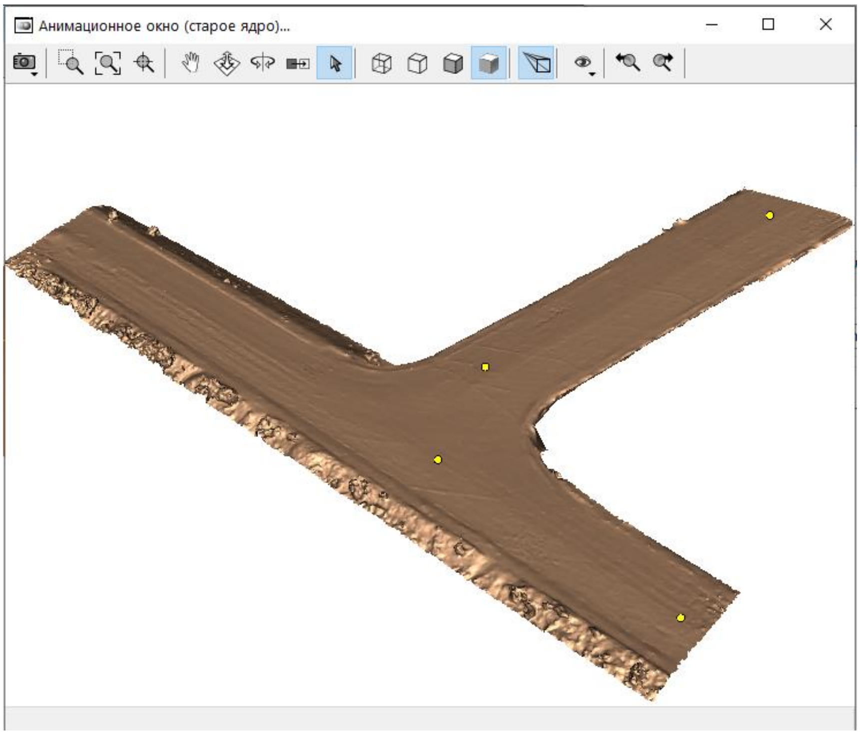


Figure 1.13. Setting key point for the route

Adding a new route and switching to the mode of setting key points of the route using the mouse.

When this button is clicked, an animation window appears with a surface image, in which the user sets markers (key points) on the route with the mouse (Figure 1.13). The user should be guided by the following rules for setting markers.

- The first and second points determine the initial position of the car, so that the origin of CK0 is located at the first point, and the X-axis (the longitudinal direction) is determined by the second point, Figure 1.14.

- The markers are placed in the positions corresponding to the change in the direction of movement, as well as at the points where it is supposed to start and end the change in the longitudinal speed of the vehicle, Figure 1.15.
- After setting the key points, close the animation window and confirm saving the input results.

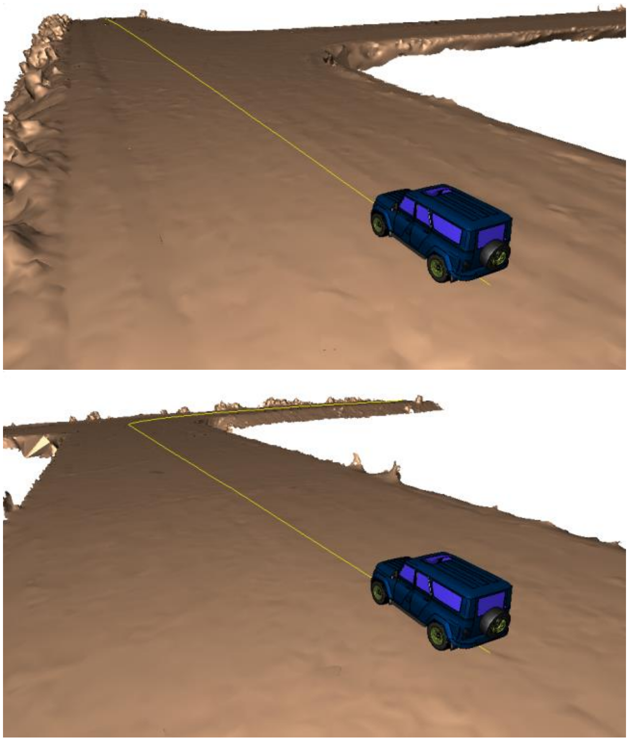


Figure 1.14. Examples of initial vehicle positions for various routes

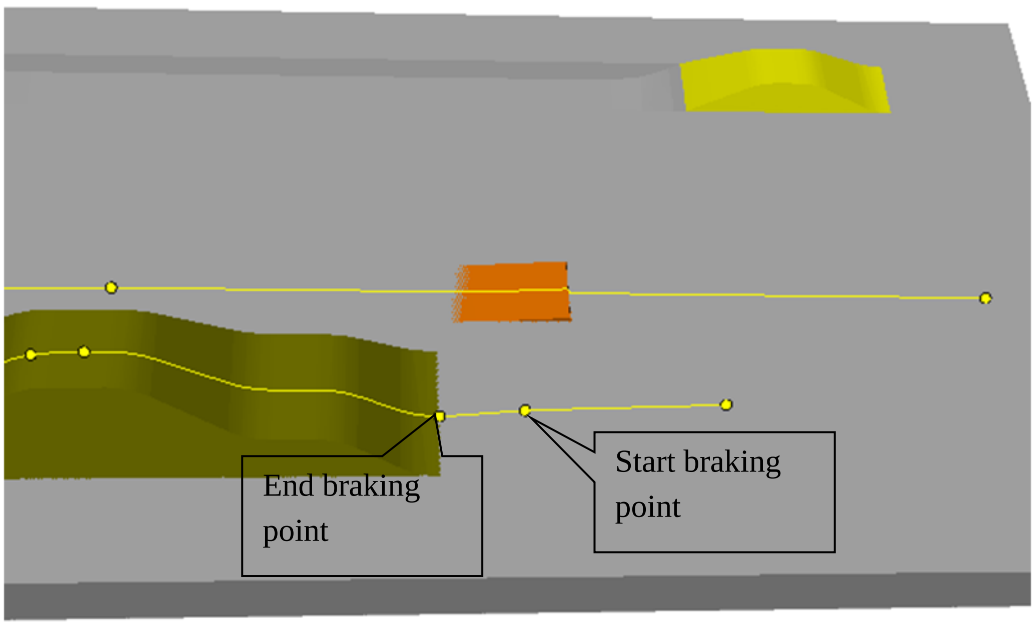


Figure 1.15. Markers corresponding to the start and end of braking

Editing the selected route in the curve editor.

Click the button to open the selected route in the curve editor, Figure 1.16.

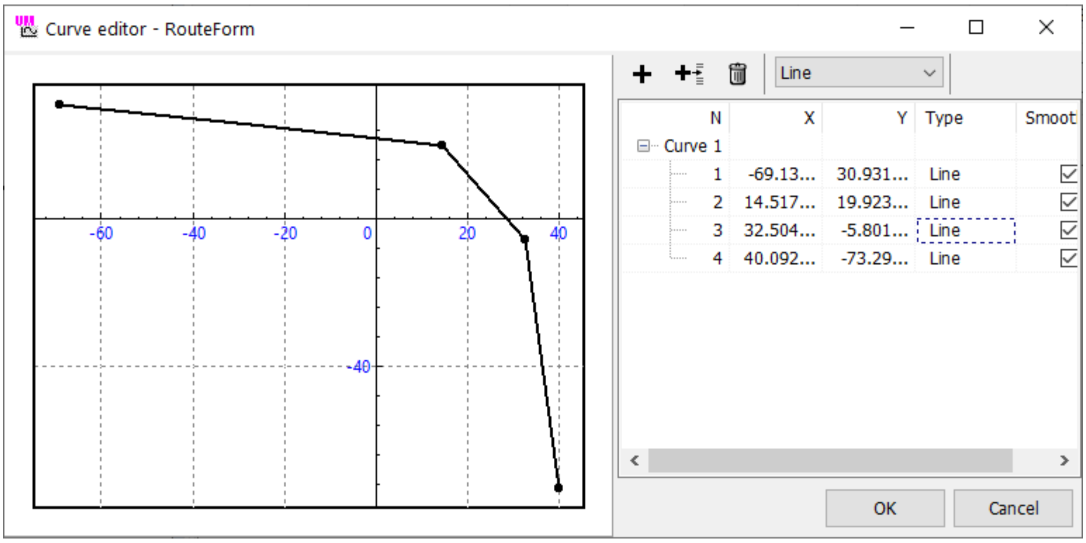


Figure 1.16. A route in the curve editor

In the cure editor window, the user can

- correct point coordinates;
- delete or add any number of points;
- perform route smoothing using circular arcs or splines, Figure 1.17.

Detailed information about this tool can be found in Chapter 3 of the User's Guide, Sect. “*Curve Editor*”.

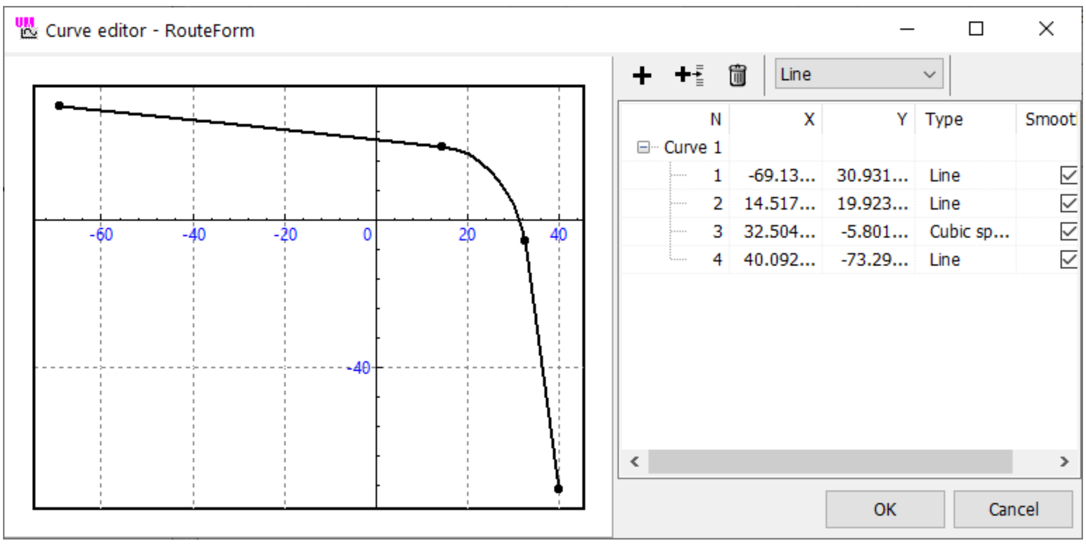


Figure 1.17. The route after editing

Viewing the current route in an animation window.

Highlight the required route in the list and click this button. An animation window will appear showing the testing area and the route with markers. If you move the mouse cursor to the marker, then the marker coordinates and the distance to it along the route *S* will appear in the status bar at the bottom of the window, Figure 1.18.

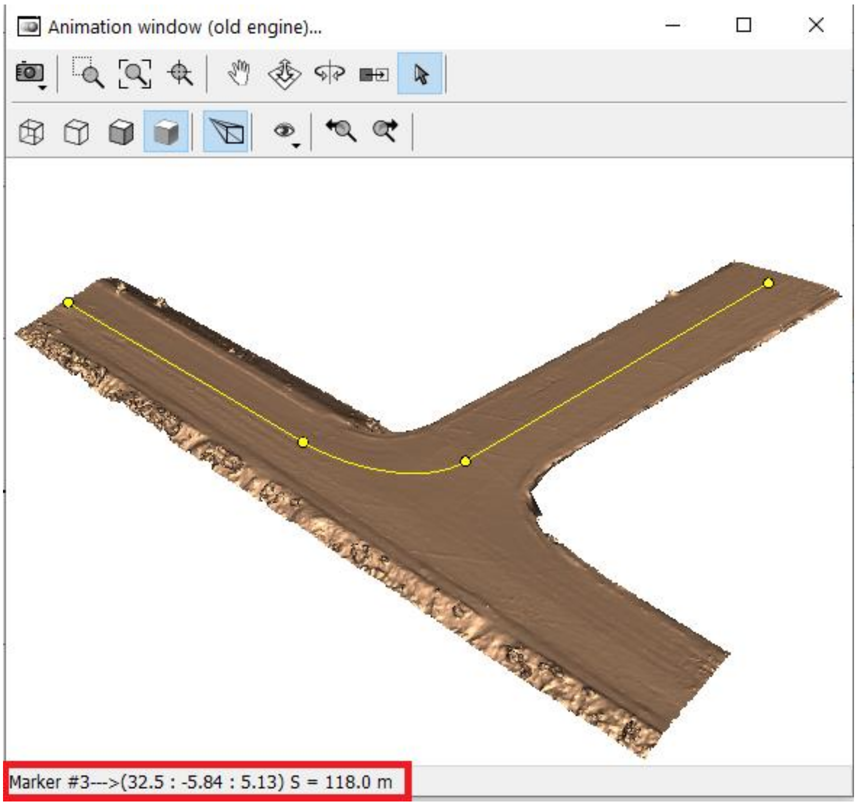


Figure 1.18. The route in animation window

Save the route list in a file.

The routes are saved to a text file with the *.rt extension with the same name as the graphic image of the surface and in the same directory. For example, for a surface

{Data UM} \Car\TestingArea\Road.img

The routes will be saved in the file

{Data UM}\ Car\TestingArea\Road.rt.

1.3.3. Micro profile (irregularities)

Micro profile or road roughness (irregularities) in UM is a function of the longitudinal distance s, which is the distance along the real trajectory of wheels at simulation. Irregularities are stored in **.irr*[1] text files for the left and right tracks separately. A file contains two columns separated by space(s). The first column contains the distance coordinate s, the second one is the height of irregularities. Both coordinates are in meters. When generated by the wizard of irregularities, the step size in the path coordinate is 0.1m. By simulation the irregularity function is smoothed with the B-spline. An example of the irregularity file is given below.

Note. Please, note that point is used as a decimal separator.

0 -0.0247274
0.1 -0.0266635

[1] From 'irregularities'

0.2 -0.0283658
0.3 -0.0294865
0.4 -0.0299168
0.5 -0.0298581
0.6 -0.0297213
0.7 -0.029892
0.8 -0.0304888

1.3.3.1. Library of irregularity files

UM in configuration with **UM Automotive** module includes a library of spectra and realizations of irregularities, which correspond to different roadway coverings and their states.

Spectra of half-sums and half-differences are obtained from [1] and correspond to the track width 1.8 m. Irregularity files in the library are generated with these spectra.

See Sect. 1.3.3.2. *"Generation of irregularity files"*, p. 1-21.

Irregularity spectra

Location:**{UM Data}\car\irregularities\spectrum**

File *.crv	**Comments***
concrete+, concrete-	Concrete on rigid foundation
asphalt_fine+, asphalt_fine-	Asphalt, good state
asphalt_satisfactory+, asphalt_satisfactory-	Asphalt, satisfactory state
cobble+, cobble-	Cobblestone road, satisfactory state

*Signs + and – correspond to half-sum and half-difference spectra

Use the **Track** tab of the irregularity generation wizard to get the file on half-sum and half-difference spectra (Figure 1.19). Note that the frequency in the above files is measured in rad/s, and **the Angular Frequency** key must be on.

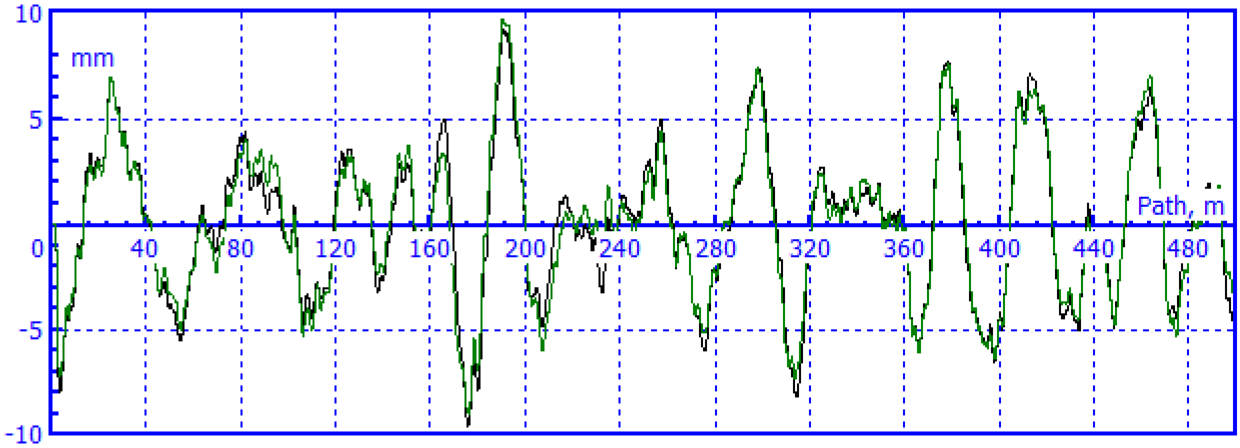
Irregularities

Location: **{UM Data}\car\irregularities**

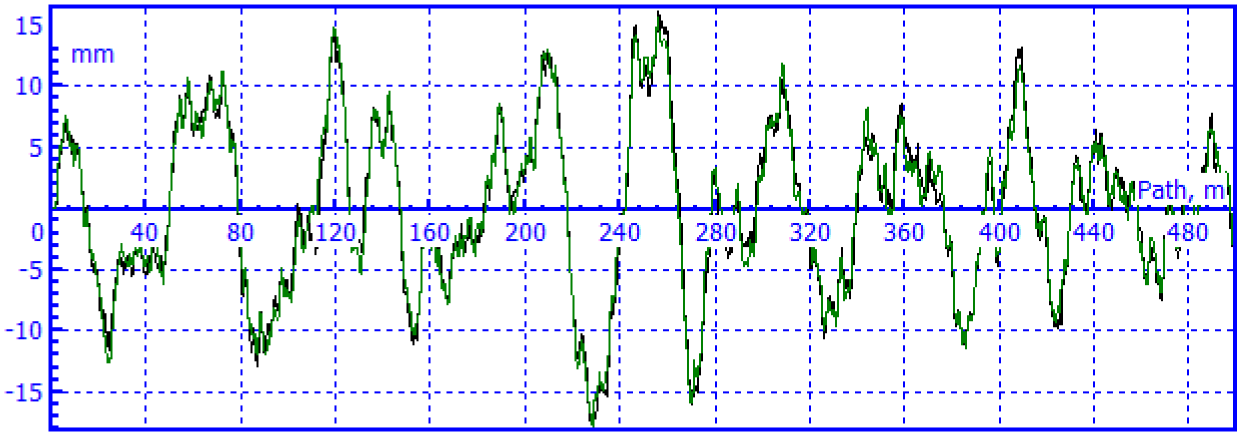
File *.irr	**Comments***
concrete_left, concrete_right	Concrete on rigid foundation
asphalt_fine_left, asphalt_fine_right	Asphalt, good state
asphalt_satisfactory_left, asphalt_satisfactory_right	Asphalt, satisfactory state
cobble_left, cobble_right	Cobblestone road, satisfactory state

* **left** and **right** correspond to the left and right track

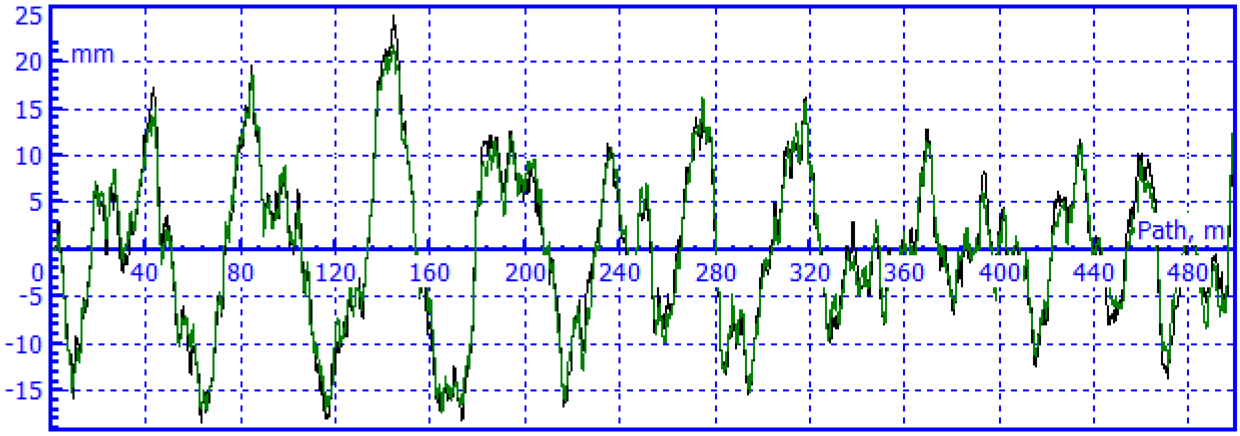
Plots of left and right irregularities from the library are shown below.



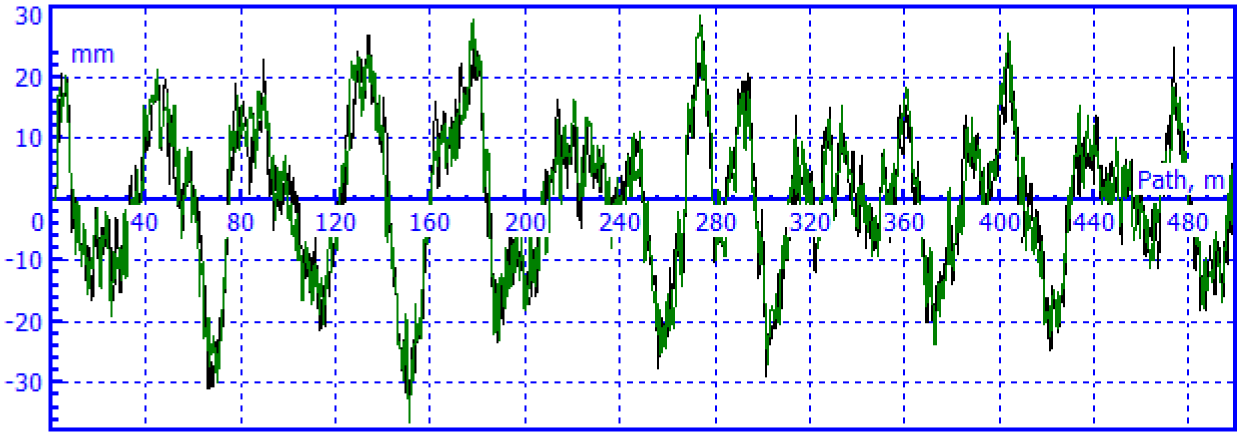
Irregularities “concrete”



Irregularities “asphalt_fine”



Irregularities “asphalt_satisfactory”



Irregularities “cobble”

1.3.3.2. Generation of irregularity files

1.3.3.2.1. Wizard of irregularities

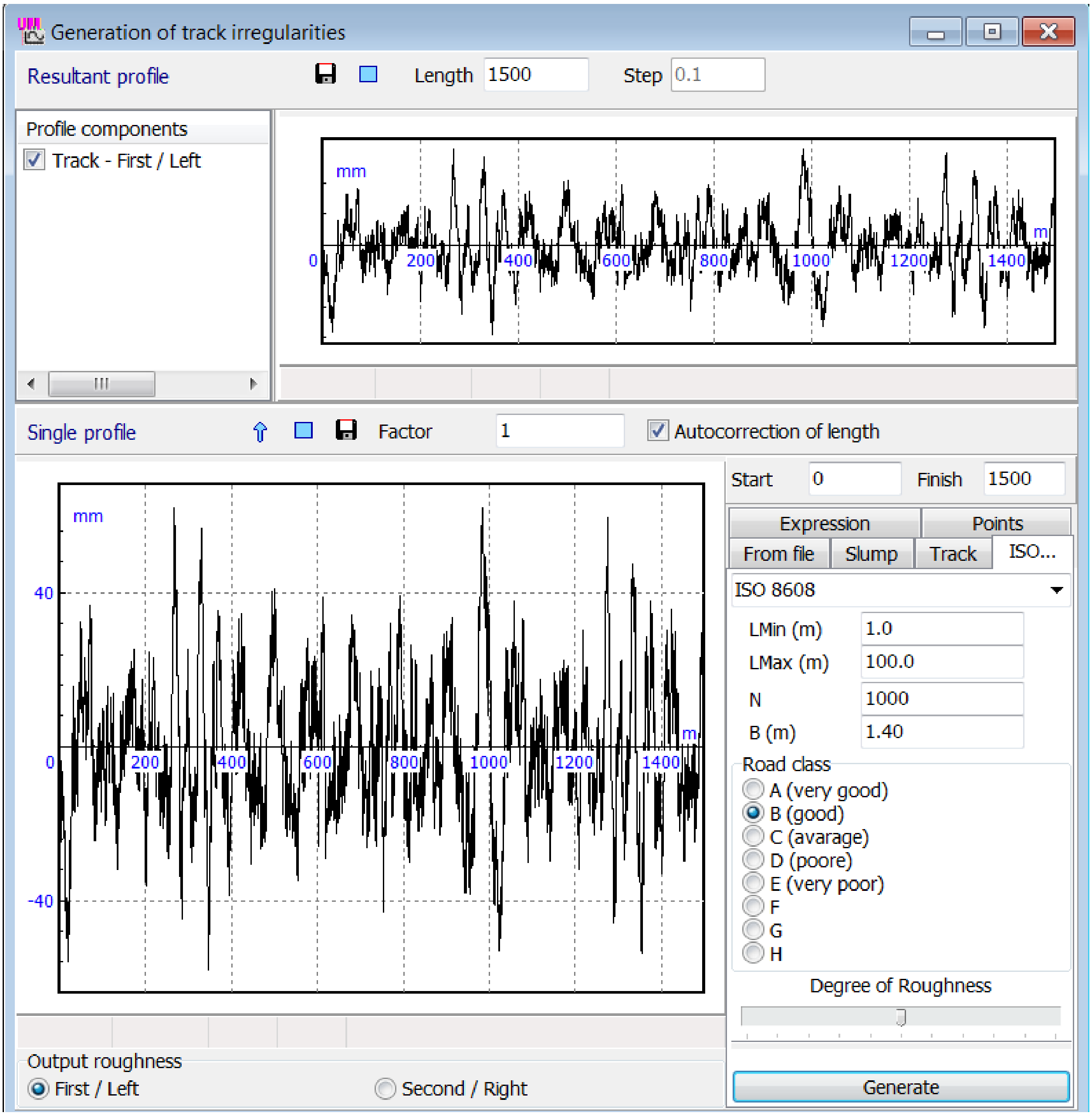


Figure 1.19. Wizard of irregularities

A new file of irregularities is created with a special tool, which is available in the **UM Simulation** program by clicking the **Tools | Irregularity editor | Road for car…** menu command (Figure 1.19).

Within this tool the longitudinal coordinate is measured in meters but the irregularities – in millimeters.

Workflow

The *resultant profile* of irregularities is plotted in the top part of the tool window as a sum of separate profiles, generated in the bottom part of the window. After a separate component of the profile is ready, use the button to add it to the resultant profile. Use the **Start, Finish, Factor** parameters while generation of the component. These parameters allow the user both add and stick profiles.

Use the in the window top to save the resultant profile to file.

Elements of control

Top part.

- Button is used for saving the profile in a file.
- Button clears the resultant profiles (removes all components).
- Parameter *Length* sets the length of the data along the track.

Bottom part

Tabs in the right bottom part are used for creation separate irregularities of different types. The corresponding plot is located in the left bottom part of the window (Figure 1.19). Buttons and parameters at the top have the following functions.

- Button adds the current separate irregularity to the resultant track profile.
- Button saves the current separate irregularity to file.
- Buttons clears the current separate irregularity.
- Parameter *Factor*: the current separate irregularity is added in the resultant one, it is multiplied by this factor. Consider an example. The user wants to convert some irregularity in text format data into UM format. Let the data be given in meters. The tool with the help of the *Points* tab can accept the irregularity. However the factor 1000 should be set before adding the data to the resultant profiles to convert it in millimeters.
- The *Autocorrection of length* check box: if it is on, the length of the resultant profile is automatically increased to match the adding separate irregularity.
- The *Start* parameter shows where the separate irregularity begins when added to the resultant profile. Note that the plot of the separate irregularity in the bottom graphic window always starts with zero.
- The *Finish* parameter sets the length of the current irregularity. More exactly, the length is the difference between the finish and the start parameter values.

1.3.3.2.2. Generation of irregularities by power spectral density function (PSD)

Irregularities can be generated by any power spectral density $S(n)$ with the help of the following formula:

$$z(s_k) = \sum_{i=0}^{N} \sqrt{2S(n_i)2\pi\Delta n} \cos(2\pi n_i s_k + \varphi_i), s_k = k\Delta s, n_i = n_0 + i\Delta n.$$

Here Δs is the step size, m; N is the number of harmonics; $S(n)$ is the PSD function, m3/(cycles/m); n is the spatial frequency, cycles/m, Δn is the step size of frequency; n_0 is the minimal frequency, φ_i is the stochastic phase uniform distributed in $[-\pi, \pi]$.

The following function is usually used for approximation of PSD [2]

$$S(n) = Cn^w$$

where *C, w* are some constants, i.e. in the logarithmic scale the PSD plots are straight lines which inclinations are defined by a negative constants *w*:

$$lgS = C + wlgn$$

A coherence function $\rho(n)$ is recommended to be used for generation of two-track irregularities. Estimation of the coherence function for different values of the track width *2b* is given in [3], Figure 1.20. It allows evaluation of PSD functions of a half-sum S_+ and half-difference S_- of the left and right irregularity heights by the given PSD function S as

$$S_+(n) = S(n)\big(1 + \rho(n)\big)/2,$$
$$S_-(n) = S(n)\big(1 - \rho(n)\big)/2.$$

PSD S_+, S_- functions are used for generation of the half-sum and half-difference profiles z_+, z_-, which result in profiles for the left and right tracks as $z_l = z_+ + z_-,\ z_r = z_+ - z_-$.

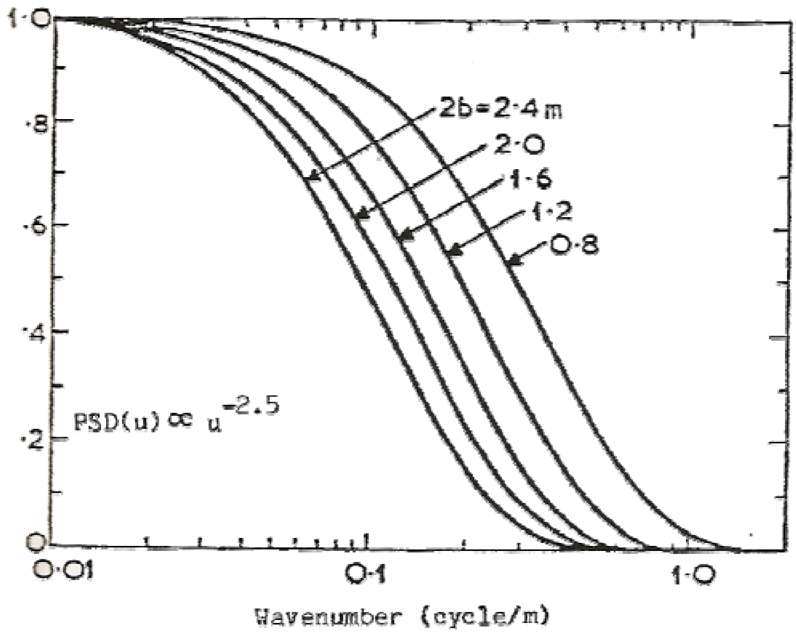


Figure 1.20. Coherence function for different values of track width [3]

1.3.3.2.3. Models of roughness generated by PSD: ISO 8608, Wong, Dixon, experiment, track

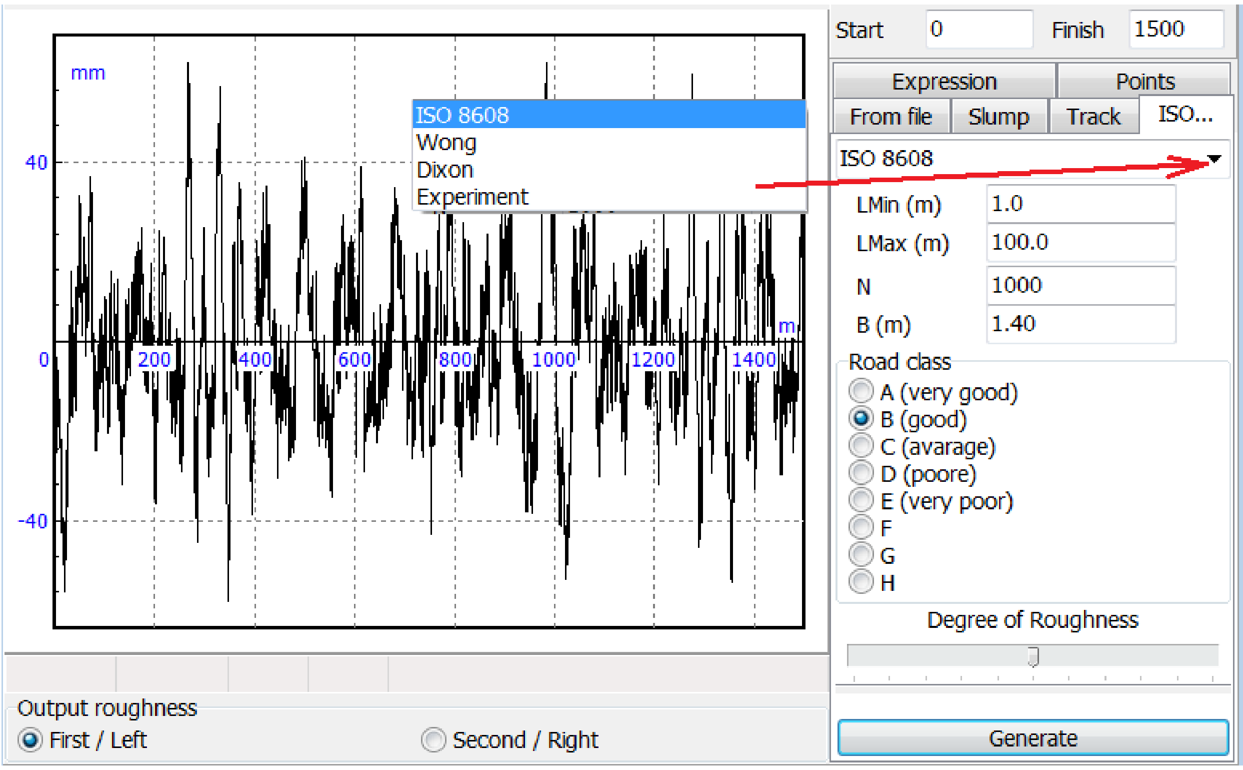


Figure 1.21. Generator of the left and right track roughness

To generate coherent irregularities for the left and right track, the following steps are necessary, Figure 1.21:

- Select the PSD type (ISO 8608, Wong, Dixon, Experiment);
- Set the minimal and maximal length of the roughness wave **LMin**, **LMax**;
- Set the number of harmonics N;
- Set the track width **B (m)**.
- Set other parameters depending on the roughness type, see below.
- Compute irregularities by the click on the **Generate** button.
- Select the **Output roughness** (Left/Right track).
- Save irregularities in two files as it is described in Sect. 1.3.3.2.1. *"Wizard of irregularities"*, p. 1-21.

Now consider different types of PSD functions implemented in UM.

1.3.3.2.3.1. ISO 8608

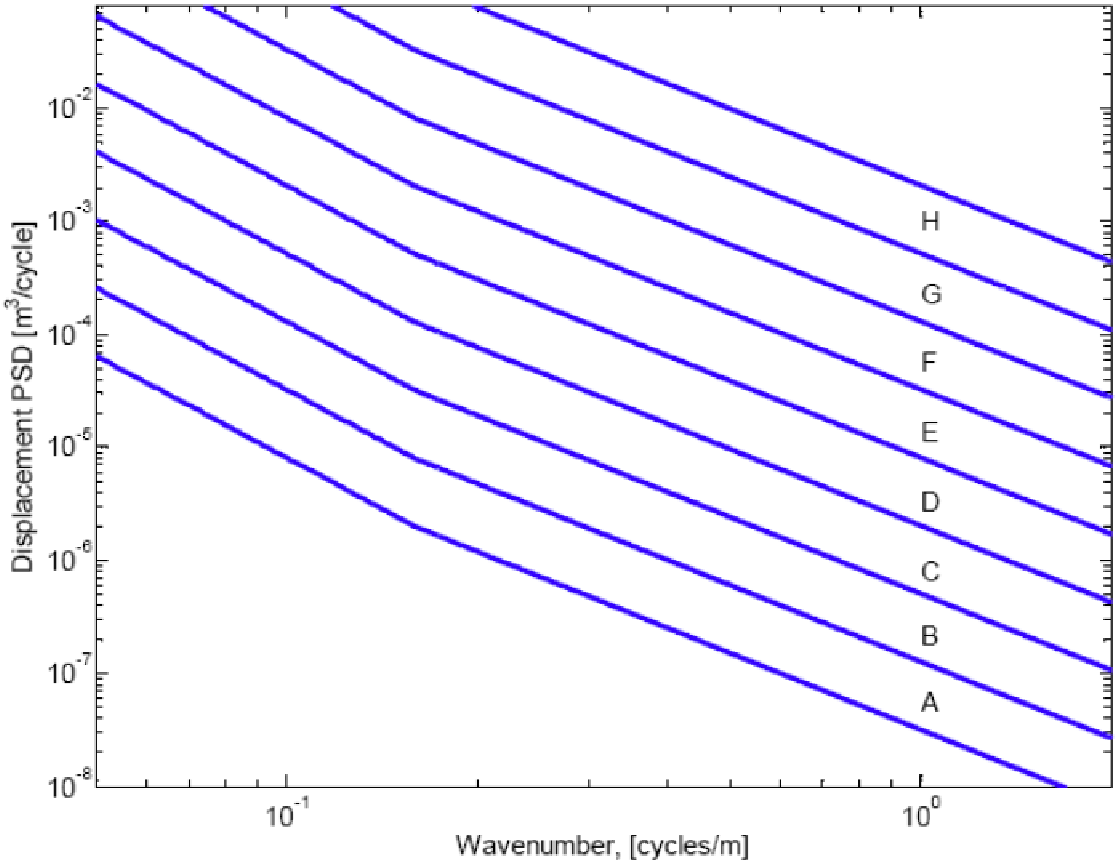


Figure 1.22. SPD function by ISO 8608

The standard ISO 8608 1995 (e) introduces the classification of the road roughness level (A-H) and a PSD function, which can be used for generation of track profiles. The PSD function is, Figure 1.22:

$$S(n) = \begin{cases} S_0(n/n_0)^{w_1}, n < n_0 \\ S_0(n/n_0)^{w_2}, n > n_0 \end{cases}$$

The following parameter values are recommended in ISO 8608:

$$n_0 = \frac{1}{2\pi}, w_1 = -2, w_2 = -1.5.$$

The S_0 parameter specifies the roughness level according to Table 1.1.

Table 1.1

Classification of road surface roughness by ISO 8608

Road class	Degree of roughness, S_0 ($\times 10^{-6} m^3/cycles$)
A (very good)	<8
B (good)	8-32
C (average)	32-128

D (poor)	128-512
E (very poor)	512-2048
F	2048-8192
G	8192-32768
H	>32768

To specify the roughness, the user should select the road class and the roughness degree within the selected class, Figure 1.21.

Figure 1.23 shows roughness of class B, $S_0 = 20 \times 10^{-6}$, LMin = 3m, LMax=30m, number of harmonics N=3000.

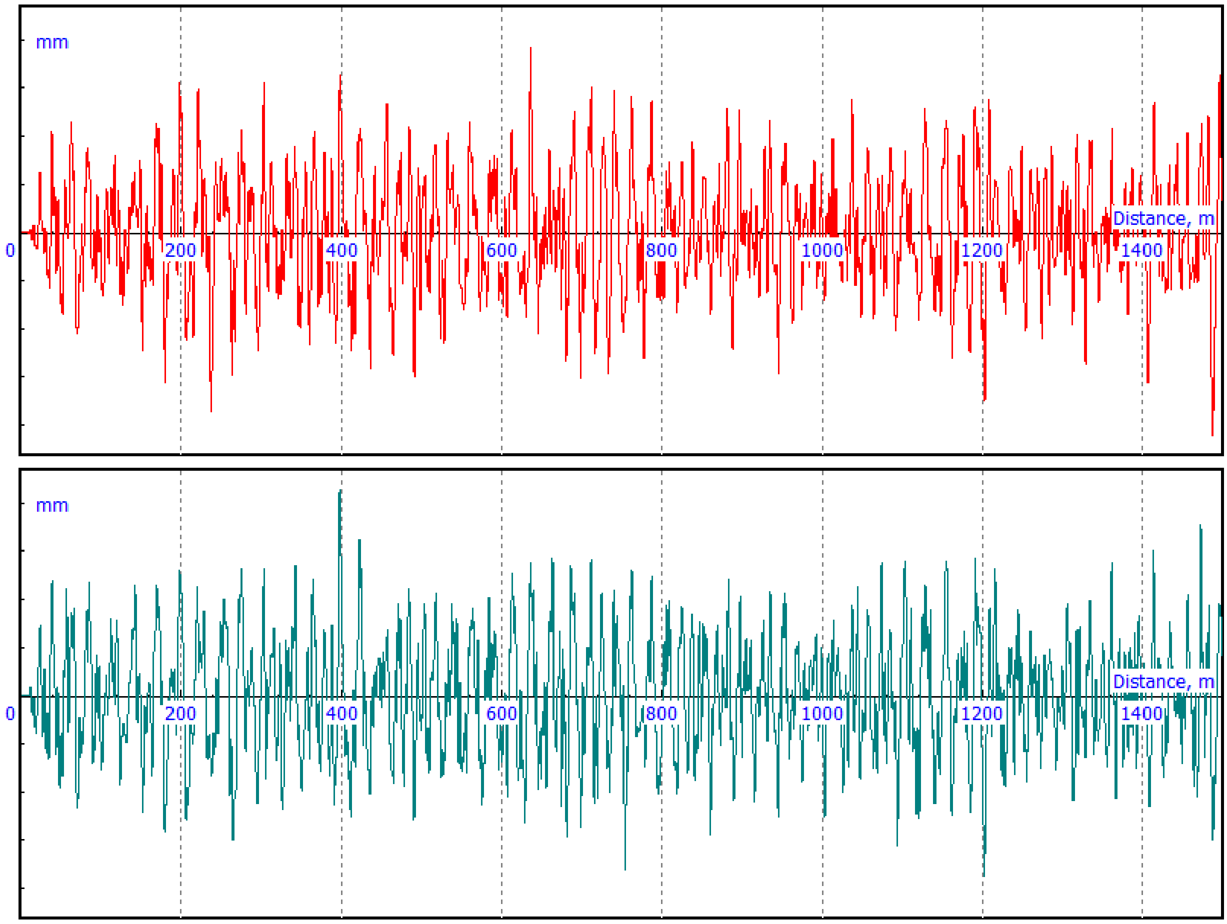


Figure 1.23. Example of road roughness for the left and right tracks

1.3.3.2.3.2. Wong

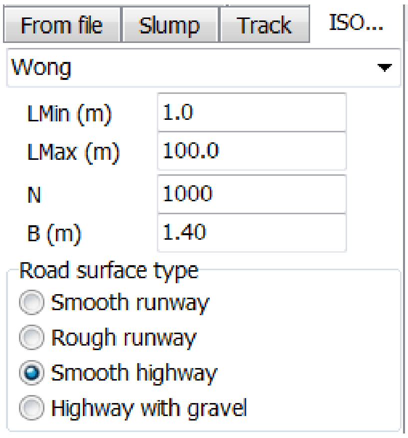


Figure 1.24. Road roughness parameters by parameters according to J.Y.Wong [2]

Table 1.2

PSD function parameters according to J.Y.Wong [2]

Road description	w	C
Smooth runway	-3.8	4.3×10-11
Rough runway	-2.1	8.1×10-6
Smooth highway	-2.1	4.8×10-7
Highway with gravel	-2.1	4.4×10-6

In the book of J.Y.Wong [2] some parameter values for the PSD function $S(n) = Cn^{-w}$ are given, see Table 1.2, Figure 1.24.

1.3.3.2.3.3. Dixon

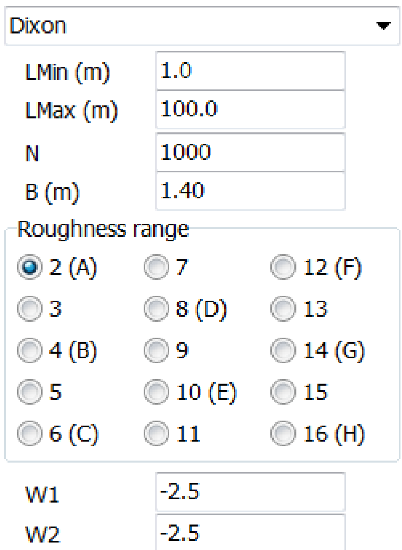


Figure 1.25. Road roughness classification by J. Dixon [4]

Rating	S mean (cm^3/c)	S range (cm^3/c)	ISO class	ISO description
2	4	<8	A	very good
3	8			
4	16	8–32	B	good
5	32			
6	64	32–128	C	average
7	128			
8	256	128–512	D	poor
9	512			
10	1024	512–2048	E	very poor
11	2048			
12	4096	2048–8192	F	—
14	16384	8192–32768	G	—
16	65536	>32768	H	—

An extended classification of road roughness is proposed in the book of J. Dixon [4], which includes ISO 8608 as a particular case. The road rating is specified from 2 to 16, where the roughness degree parameter S_0 increases twice when the rating increase by a unit, which corresponds to the growth of the roughness level by the factor $\sqrt{2}$. The same ISO 8608 PSD function is used

$$S(n) = \begin{cases} S_0(n/n_0)^{w_1}, n < n_0 \\ S_0(n/n_0)^{w_2}, n > n_0 \end{cases}$$

$n_0 = \frac{1}{2}\pi$. The parameters w_1, w_2 can be set be the user. The default values are $w_1 = w_2 = -2.5$.

1.3.3.2.3.4. Experiment

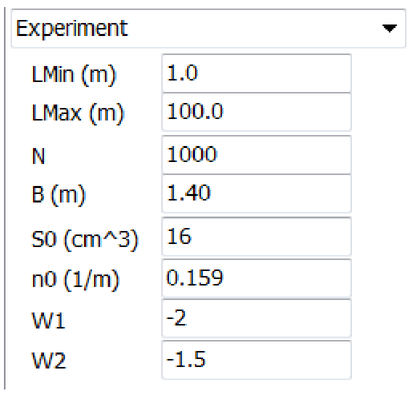


Figure 1.26. PSD parameters

In this case, the user can set arbitrary values of the PSD function, Figure 1.26

$$S(n) = \begin{cases} S_0(n/n_0)^{w_1}, n < n_0 \\ S_0(n/n_0)^{w_2}, n > n_0. \end{cases}$$

Thus, this is the more general case compared to the above descriptions, in particular, the user can set data obtained from field tests.

1.3.3.2.3.5. Track

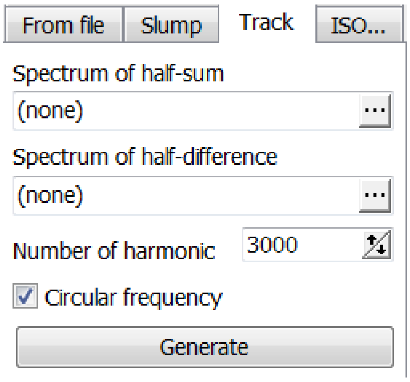


Figure 1.27. Pointwise description of PSD functions

Like above, this tool is used for generation of coherent track profiles.

The PSD functions of half-sum and half-difference spectra is set by points with the curve editor. The user can use files of spectrum library if necessary (see Sect. 1.3.3.1. *"Library of irregularity files"*, p. 1-18). Please remember that the spectra from the library depend on the angular frequency, and the corresponding key must be checked (Figure 1.27).

Two realizations are created by the half-sum and half-difference spectra, conditionally the left and the right ones. Use the **Output roughness** radio group to switch between them and to create two different files.

1.3.3.2.4. Other tools for description of road roughness

1.3.3.2.4.1. Analytic expression (the Formula tab)

Set an analytic expression *f(x)* in the *Function of irregularity* edit box and press the *Enter* button or click ... button. Standard functions can be used in the expression (Chapter 3, Sect. *Standard functions and constants*). Standard expressions can be assigned from the pull down list as well.

1.3.3.2.4.2. Slump

Create a special and often used irregularity. Set its position and length using the *Start* and *Finish* parameters.

1.3.3.2.4.3. From file

Here an already created file of irregularities *.irr can be read. To do this, use the button. A part of the irregularity, which length and position is determined by the *Start* and *Finish* parameter may be added to the resultant track profile.

1.3.3.2.4.4. Points

Here an irregularity is created as a set of points defined with the help of the curve editor (Chapter 3, Sect. *Object constructor/Curve editor*). To call the editor, click the ... button. In particular, here the user can convert an irregularity given in a text format into UM format. For this purpose the irregularity should be open in any text editor in a two-column format. The first column should contain abscissa values in meters, i.e. the longitudinal coordinate starting with zero value. The second column should contain the irregularities, e.g.

0 0
0.05 0.011
0.10 0.021
…..

To input this data from the clipboard

- Delete all previously added points
- Copy data into clipboard from any text editor in a standard manner;

Activate the curve editor by the mouse and paste the data from the clipboard (*Ctrl+V* or *Shift+Insert* hot keys).

Spline approximation can be applied to the data.

Use the *Factor* parameter if the irregularities are not measured in millimeters to convert data to the necessary unit (mm). For instance, if the ordinate is originally in meters, the factor must be 1000.

Note that points can be set with any step size on abscissa. But before saving the data into the **.irr* file they are interpolated with the step size 0.1m using B-spline smoothing. Thus, the result will be slightly different from the original due to features of the B-spline. This smoothing is physically similar to smoothing of small irregularities by the tire.

1.3.3.3. Assigning irregularities

Use the **Road vehicle | Options** tab of the **Object simulation inspector** to select the irregularity files for the left and right wheels by clicking the  buttons (Figure 1.28). Paths to selected files are stored in the configuration file **.car*.

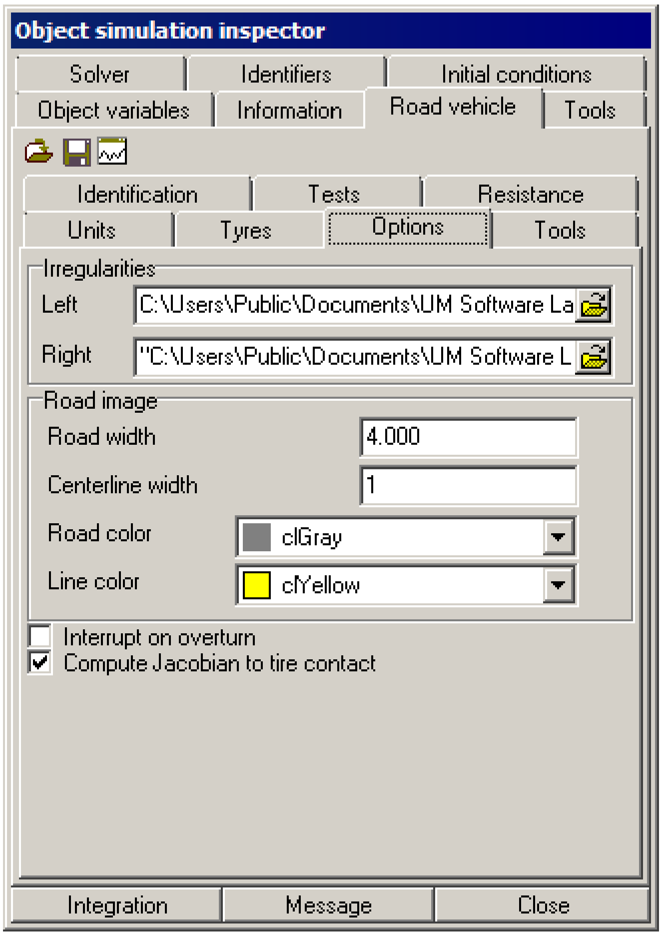


Figure 1.28. Setting current irregularities

Current irregularities are visualized by clicking the button.

Irregularity profiles are corrected at the first two-meter distance to provide a smooth run of a vehicle on the irregularities, Figure 1.29. Thus, the vehicle at start is always on an absolutely even horizontal plane.

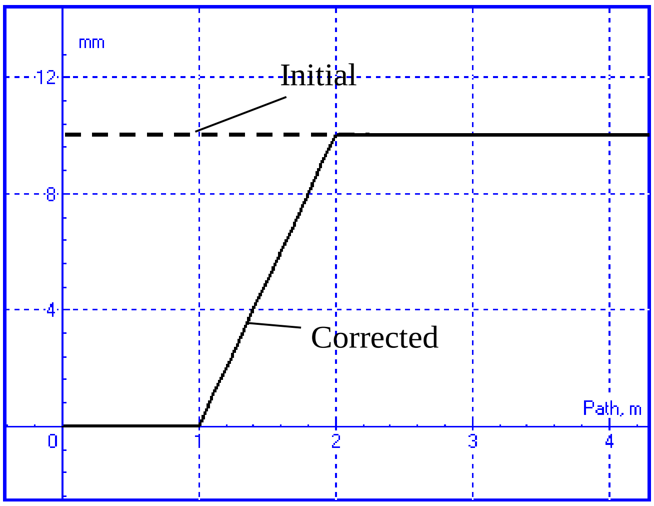


Figure 1.29. Correction of irregularities

1.3.4. Test section profile

Figure 1.30. Speed bump

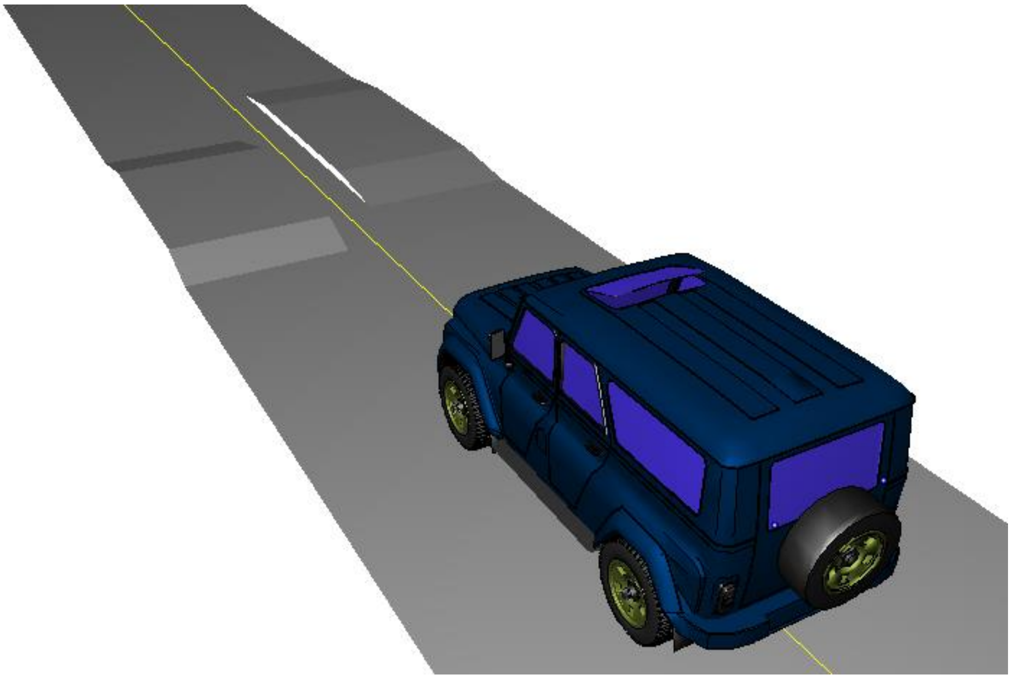


Figure 1.31. Asymmetric road irregularities for the left and right track

Test section profiles (TSP) are geometric deviations of road from the ideal state, which cannot be considered as smooth and small irregularities. For example, a speed bump in Figure 1.30 or steps in Figure 1.31 can be considered in UM as TSP only. **The tool is applied to the test with driver only** when specifying the macrogeometry of the vehicle movement using a flat curve, Sect. 1.3.1 “*Defining a macro profile using curves*”.

Creating the TSP files is considered in Sect. 1.9.1.3.2 “*Creating files with test section profiles*”.

Features of the tire model when using TSP are discussed in Sect. 1.5.1 “*Single point and multipoint normal contact models*”.

Assignment of TSP files in a test with driver see in Sect. 1.9.4.7.2.3 “*Simulation with test section profiles (TSP)*”.

1.4. Driver

1.4.1. MacAdam's model

The MacAdam's model is one of the efficient and frequently used models of a driver (path follower) in the case of a single-unit vehicle. A simplified linear model of a two-wheel vehicle with two degrees of freedom lies in the bases of this model. According to the driver model the steer angle is computed from the condition of minimal deviation of the predicted path from the desired one. Consider the mathematical side of the model in more details.

The control u (the desired steer angle) is a piecewise constant function. Consider the vehicle position at the time t_k when the next value of the control is evaluated, Figure 1.32). Without losing generality of solutions obtained below, this moment can be set to zero, $t_k = 0$. Let us introduce an inertial frame $O_v X_v Y_v$, connected with the current position of the vehicle. The origin of this system is located in the middle point of the centerline of the front axle; the abscissa axis coincides with the longitudinal axis of the vehicle.

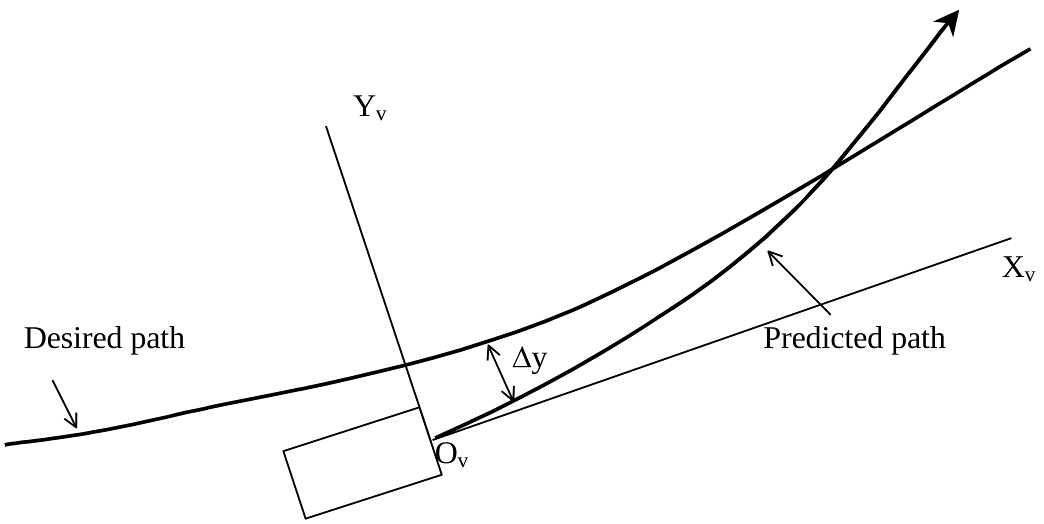


Figure 1.32. Desired and predicted paths

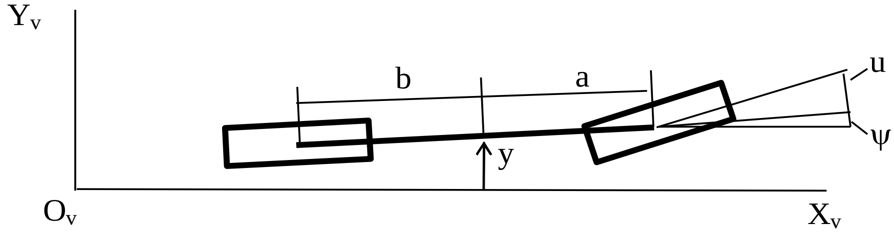


Figure 1.33. Two-wheel model of vehicle

If the steer angle u is given, the simplified model of the vehicle shown in Figure 1.33 has 2 degrees of freedom: the lateral coordinate of the vehicle center of mass y the yaw angle ψ. Linear equations of motion in these variables have the following form:

$$\dot{y} = v_x\psi + v_y, \tag{1.1}$$
$$\dot{\psi} = \omega_z,$$
$$M\dot{v}_y = -\frac{C_f + C_r}{v_x}\dot{y} + \left(\frac{C_r b - C_f a}{v_x} - Mv_x\right)\omega_z + C_f u,$$
$$I_z\dot{\omega}_z = \frac{C_r b - C_f a}{v_x}\dot{y} - \frac{C_f a^2 + C_r b^2}{v_x}\omega_z + C_f a u.$$

Here v_x, v_y are the projections of the vehicle velocity on the longitudinal and lateral axis of the vehicle ($v_x = const$), ω_z is the yaw rate, a, b are the distances from the mass center to the front and rear axles, M, I_z are the mass of the vehicle and its moment of inertia about the vertical central axis, C_f, C_r are the cornering stiffness constants for the front and rear tires.

The observed variable is the lateral coordinate of the middle point on the centerline of the from axle

$$y_v = y + a\psi. \tag{1.2}$$

Equations (1.1), (1.2) are linear with constant coefficients, and can be written in the matrix form as

$$\dot{x} = Ax + Bu, \tag{1.3}$$
$$y_v = C^T x,$$
$$x = \begin{pmatrix} y \\ \psi \\ v_y \\ \omega_z \end{pmatrix}, \quad A = \begin{pmatrix} 0 & v_x & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{C_f + C_r}{Mv_x} & \frac{C_r b - C_f a}{Mv_x} - v_x \\ 0 & 0 & \frac{C_r b - C_f a}{I_z v_x} & -\frac{C_f a^2 + C_r b^2}{I_z v_x} \end{pmatrix}$$
$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{C_f}{M} \\ \frac{C_f a}{M} \end{pmatrix}, C = \begin{pmatrix} 1 \\ a \\ 0 \\ 0 \end{pmatrix}$$

General solution of Eq. (1.3) with the assumption $u = const$ is

$$x(t) = e^{At}x_0 + \int_0^t e^{At} B d\tau u, \tag{1.4}$$
$$y_v(t) = F(t)x_0 + g(t)u.$$

Here x_0 is the matrix-column of initial conditions. The 1×4 matrix $F(t)$ and the scalar function $g(t)$ are obtained from the relations

$$F(t) = Ce^{At}, g(t) = \int_0^t F(\tau) B d\tau.$$

The state transition matrix e^{At} can be computed by numeric integration if differential equations with the identity matrix as initial conditions, i.e. ith column of this matrix is the solution of Eq. (1.3) with the initial conditions $x_{i0} = 1, x_{j0} = 0, i \neq j$. The more effective method of computation the e^{At} matrix is based on solving the eigenvalues/eigenvector problem for the matrix A.

Let $y_d(t)$ be the desired path (Figure 1.32). Determine the control u minimizing the deviation of the predicted path from the desired one $\Delta y(t) = y_d(t) - y_v(t)$ on the preview time interval T_p. The following expression is the minimized functional

$$J(u) = \int_0^T \left(\Delta y(\tau)\right)^2 d\tau = \int_0^T (y_d(\tau) - F(\tau)x_0 - g(\tau)uu)^2 d\tau.$$

The desired control is computed from the equation

$$\frac{dJ}{du} = 2\int_0^{T_p} (y_d(\tau) - F(\tau)x_0 - g(\tau)u)g(\tau)d\tau =$$

$$= 2\int_0^{T_p} (y_d(\tau) - F(\tau)x_0)g(\tau)d\tau - 2u\int_0^{T_p} g^2(\tau)d\tau = 0$$

or

$$u = \frac{\int_0^{T_p}(y_d(\tau) - F(\tau)x_0)g(\tau)d\tau}{\int_0^{T_p} g^2(\tau)d\tau}.$$

The obtained solution can be simplified if the integrals are replaced by finite sums. For this purpose we divide the preview time T_p into N equal subintervals.

$$u = \frac{\sum_{i=1}^{N}(y_d(t_i) - F(t_i)x_0)g(t_i)}{\sum_{i=1}^{N} g^2(t_i)}, t_i = \frac{iT_p}{N} \quad (1.5)$$

Currently UM uses $N = 10$.

The driver reaction is taken into account as the neuromuscular filter, which in the operator form looks like

$$\delta(s) = D(s)u,$$

$$D(s) = \frac{e^{-t_d s}}{1 + T_n s}$$

Here δ is the steer angle, t_d is the driver reaction delay, and T_n is the neuromuscular lag. After the transition this expression in the time domain we obtain the differential equation

$$T_n\dot{\delta} + \delta = u(t - t_d).$$

Taken into account that the control $u(t)$ is a piecewise constant function, the equation is solved analytically. Let t_k be the moment in which the control u is computed. Then

$$\delta(t) = (\delta_k - u)e^{-t/T_n} + u, \qquad t \in [t_k + t_d, t_{k+1} + t_d]$$

$$\delta_k = \delta(t_k + t_d)$$

The steer wheel angle is obtained after multiplying the angle δ by the steer ratio i_s

$$\alpha_s = i_s\delta.$$

Thus, the control is computed taking into account the desired path on the preview distance $L_p = vT_p$, where v is the longitudinal velocity of the vehicle, and T_p is preview time. But the control value change can be done with a period less then T_p. Let us introduce the notion of a number N_u of control steps on the preview time interval so that $t_{k+1} = t_k + T_p/N_u$. For instance, if $T_p = 1$ s and $N_u = 2$, the new control u is computed with the period 0.5s.

Table 1.3 contains a list of parameters characterizing the MacAdam's model of the driver.

Table 1.3

MacAdam' model parameters

Parameter	Comments	Recommended interval of values	Default value
T_p	Preview time	1-2s	1s
t_d	Reaction time delay	>0.15s	0.15s
T_n	Neuromuscular lag	0.1-0.2s	0.15s
N_u	Number of control steps	1-4	2

Simulation result for a maneuver of the car VAZ 2109 are shown in Figure 1.34, Figure 1.35 with the following parameter values: v=5m/s, $T_p = 2$ s, $t_d = 0.15s$, $T_n = 0.1s$, $N_u = 3$.

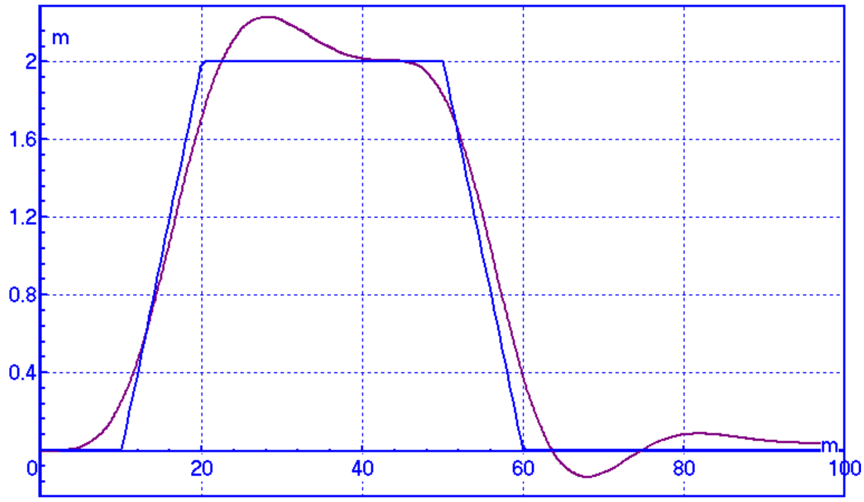


Figure 1.34. Desired and simulated path

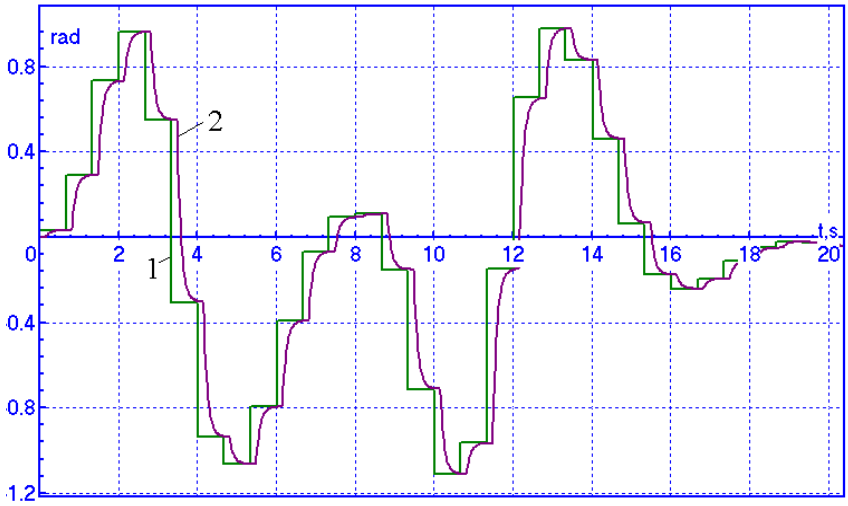


Figure 1.35. Steer wheel angle: control before (1) and after (2) the neuromuscular filter

Note. Currently the MacAdam driver model cannot be used in case of a multiunit vehicle.

1.4.2. Continuous preview model

Unlike the MacAdam's model the control in this case is continuous, i.e. the control is computed on each step of the simulation. Let L_p be the preview distance, which depend on the vehicle speed v and the preview time T_p as $L_p = vT_p$. The driver reaction delay t_d is taken into account as well.

The block diagram of the control is shown in Figure 1.36. The preview block generates the lateral coordinate $y_d(t + T_p)$ on the desired path at the distance L_p in the vehicle coordinate system, Figure 1.32. The driver predicts the lateral displacement of the vehicle y_p after the preview time T_p using the current values of the lateral velocity and acceleration of the vehicle as

$$y_p = y(t) + T_p\dot{y}(t) = y(t + T_p) + O(T_p^2\ddot{y}(t))$$

The control is proportional to the error, which is the deviation of the predicted and desired lateral coordinates taking into account the driver reaction delay.

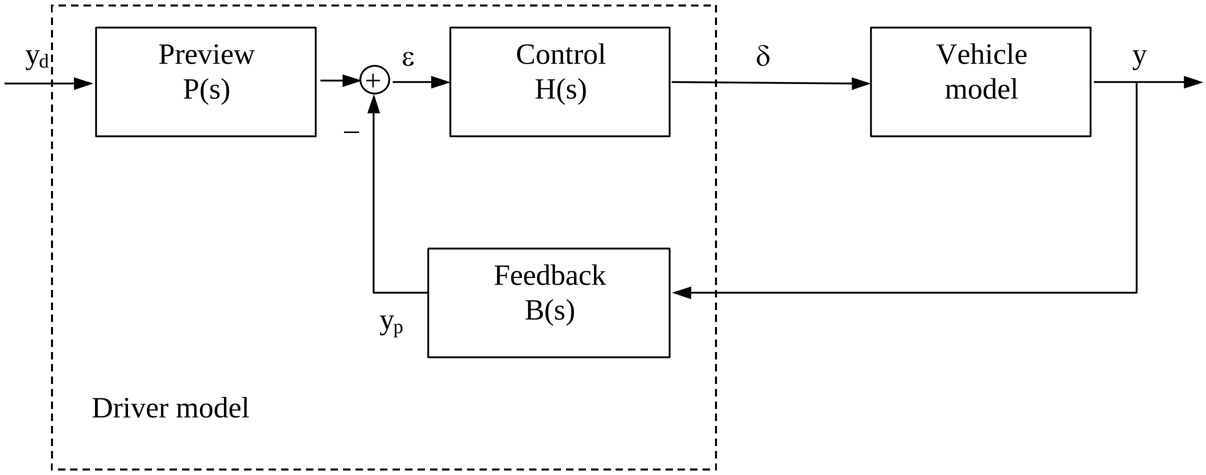


Figure 1.36. Block diagram of the control

The transfer functions are:

Preview: $P(s) = e^{T_p s}$

Control: $H(s) = Ke^{-t_d s}, L_p = vT_p$

Feedback: $B(s) = 1 + T_p s$

Here K is the gain.

Transformation in the time domain leads to the following equations:

$$\varepsilon(t) = y_d(t + T_p) - y_p,$$
$$y_p = T_p \dot{y}(t),$$
$$\delta(t) = K\varepsilon(t - t_d).$$

or

$$\delta(t) = K\left(y_d(t + T_p - t_d) - T_p \dot{y}(t - t_d)\right),$$
$$\alpha_s(t) = i_s \delta(t).$$

Simulation result for a maneuver of the car VAZ 2109 are shown in Figure 1.37, Figure 1.38 with the following parameter values: v=5m/s, $T_p = 1s, t_d = 0.15s, K = 0.1$.

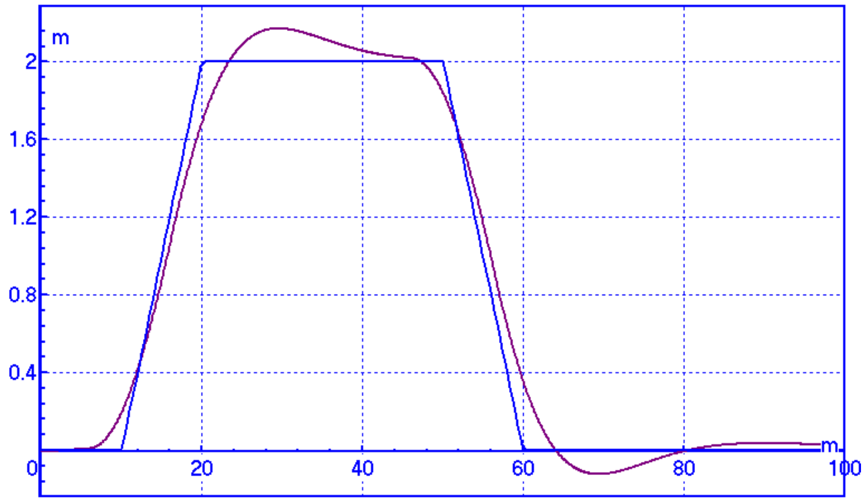


Figure 1.37. Desired and simulated path

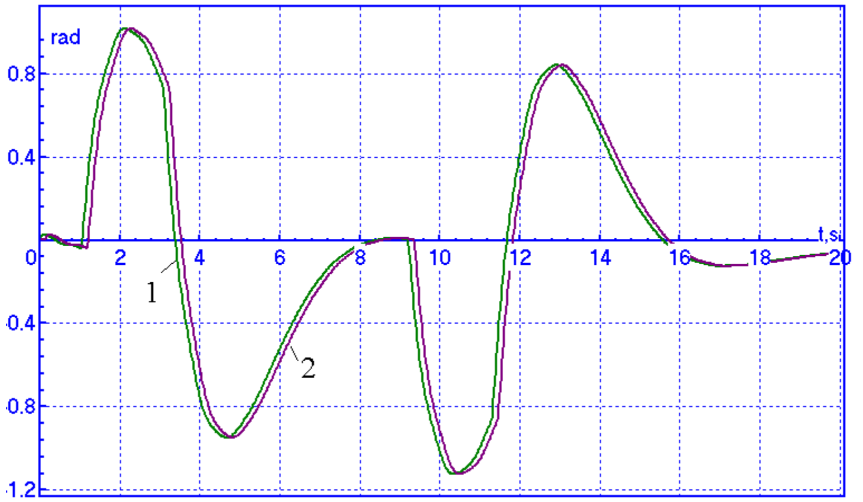


Figure 1.38. Steer wheel angle: control (1) and the driver output (2)

Note. In case of a multiunit vehicle the control is applied to the Unit 1.

1.4.3. Combination of PID controller and preview model

Both the MacAdam and the preview diver models are used in cases when a nearly real behavior of the driver is necessary. They cannot guarantee a strictly path following. At the same time some standard and frequently used closed loop maneuvers require a very exact following the path to make possible the comparison of simulation results obtained with different software. In UM such type of the driver model is realized as a combination of a PID controller and the second order preview model.

$$\delta(t) = K_2\Delta y + K_d\Delta\dot{y} + K_I\int_0^t \Delta y(\tau)d\tau + K\left(y_d\left(t+T_p-t_d\right) - T_p\dot{y}(t-t_d)\right) \quad (1.1)$$

where three first terms correspond to the PID controller with three new control parameters K_2, K_d, K_I. Note that the gain K does not depend on the preview distance, and its value is not equal to the gain in the second order preview model.

As is known from the theory of motion control, the integral term of the control is included to compensate for stationary errors in the control. This type of error occurs, for example, in the presence of a constantly acting lateral force such as a side air drag force, or if there are deviations in symmetry in the model.

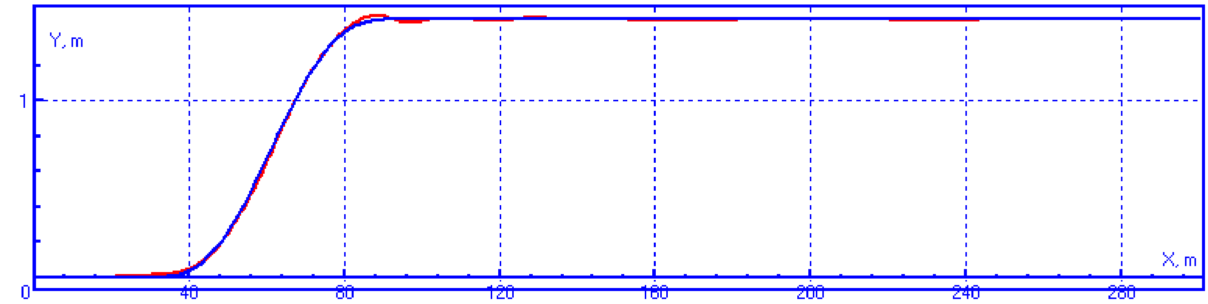


Figure 1.39. Lane change maneuver. Desired path and simulation result.

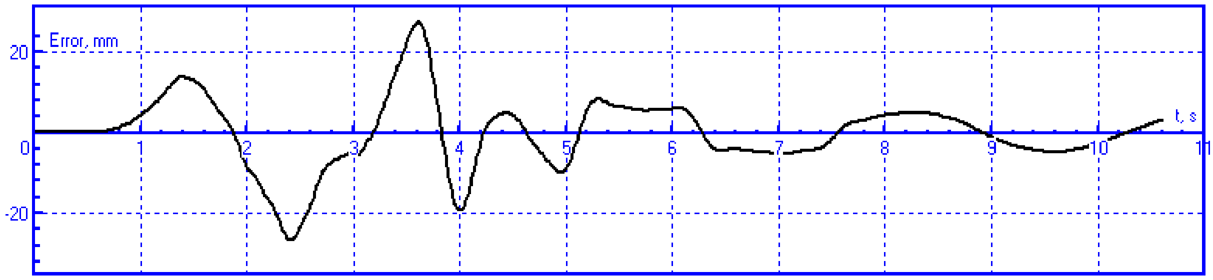


Figure 1.40. Lane change maneuver. Path following error.

Figure 1.39, Figure 1.40 show simulation results for a lane change maneuver obtained for a track/trailer model (Sect. 0. "

Units", p. 1-94). The following parameter values were used:

v=88 km/h, $K_2 = 1.5, K_d = 0.2, K_I = 2, K = 0.075, T_p = 1s, t_d = 0.05s$

Table 1.4 contains the list of parameters for the continuous driver model.

Table 1.4

Parameters of continuous control

Parameter	**Comment**	**Default value**	
		Passenger car	**Truck**
T_p	Preview time	1 s	1s
t_d	Reaction delay	0.02 s	0.05s
K	Gain factor	0.075	0.075
K_2	Gain factor 2	0.2	1.5
K_d	Factor for differential part of control	0.05	0.2
K_I	Factor for integral part of control	0.3	2

Note. The controller uses the derivative of the error $\Delta\dot{y}$ which requires a differentiable function of the desired path. In this case a spline interpolation of the path curve is necessary, Sect. 1.3.1 "*Defining a macro profile using curves*".

1.4.4. Selection of parameters for continuous control

The default parameter values given in Table 1.4 often give satisfactory control results. However, the behavior of the model depends on many parameters: tire properties, suspension parameters, speed, type of a trajectory, etc. Since the problem of vehicle control is nonlinear and does not have a rigorous analytical solution, in some cases, in order to achieve a given control quality, it is necessary to change the values of the parameters. To do this, we recommend two approaches: linear analysis and multivariate calculations. Both approaches require the parameterization of control coefficients, that is, the assignment of identifiers specified in the input program to the coefficients, see Sect. 1.9.1.2.4 "*Parameterization of driver model*".

Investigation of the driver model by means of linear analysis

Using the root locus, one can obtain important information about the influence of the parameters of the continuous driver model on the behavior of the vehicle when driving in a straight line. In particular, to determine the area of loss the control stability at high speeds, see Sect. 1.10.3.3 «*Root locus: dependence of eigenvalues on driver control* parameters».

Investigation of the driver model by means of multivariate calculation

To determine the values of the control parameters that provide the desired quality when the vehicle passes curved trajectories, it is recommended to use multivariate calculations. This tool allows you to quickly obtain and process simulation results when changing the parameters specified by identifiers.

Figure 1.41shows the simulation results for a VAZ 2109 car when passing the lane change test at a speed of 80 km/h for the values of the control parameter $K_2 \in [0,0.4]$ with a step of 0.05. The corresponding multivariate calculation is included in the standard delivery of the program and is available in the directory.

{Data UM}\SAMPLES\Automotive\MVC\SAE lane change

In order to open the corresponding multivariate calculation, use the menu command **Scanning | Load project**... and select the project.

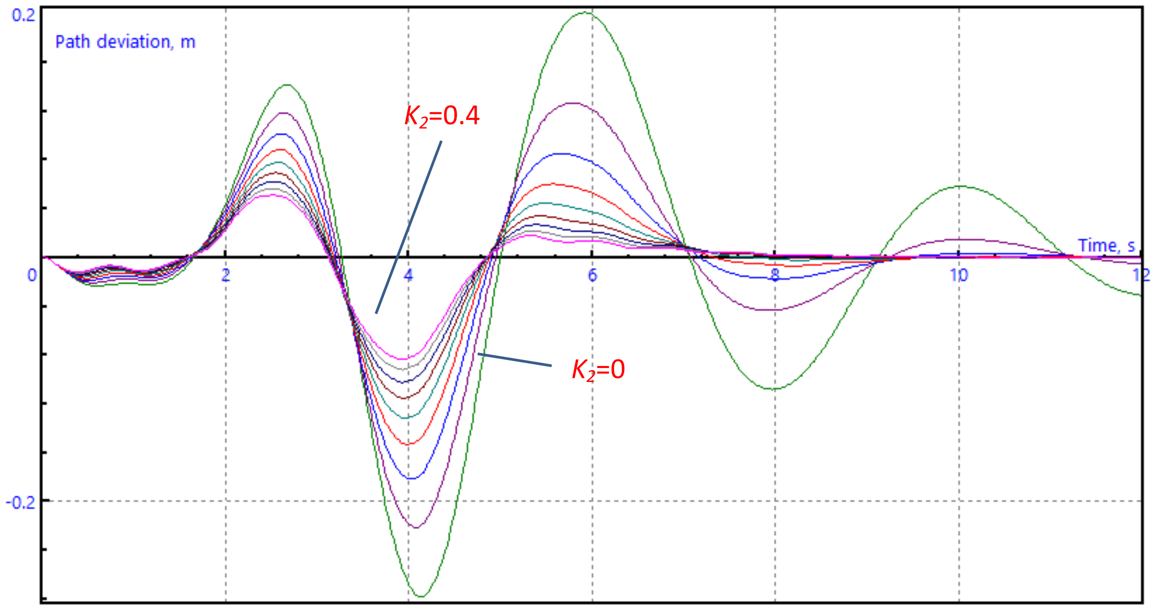


Figure 1.41. Comparison of the results of the lane change test at different values of the control parameter K_2

1.5. Tire models

Models of tire/road interaction forces allow computation of the forces in dependence of some kinematical variables: longitudinal slip, sideslip, camber. Three tire models are implemented in UM:

- FIALA model, see Sect. 1.5.1;
- Pacejka Magic Formula, see Sect. 0;
- Tabular model, see Sect. 1.5.4;
- TMEasy tire model, see Sect. 1.5.5.

Parameters describing the models are stored in **.tr* files. The default directory for these files is **{UM Data}\car\tire**. The user may use the built-in **Wizard of tire models** for changing model parameters.

Tire models described here are used both in **UM Automotive** and **UM Monorail train** modules, see Chapter 26: Simulation of Monorail Train Dynamics (file 26_um_monorail_train.pdf).

1.5.1. Single point and multipoint normal contact models

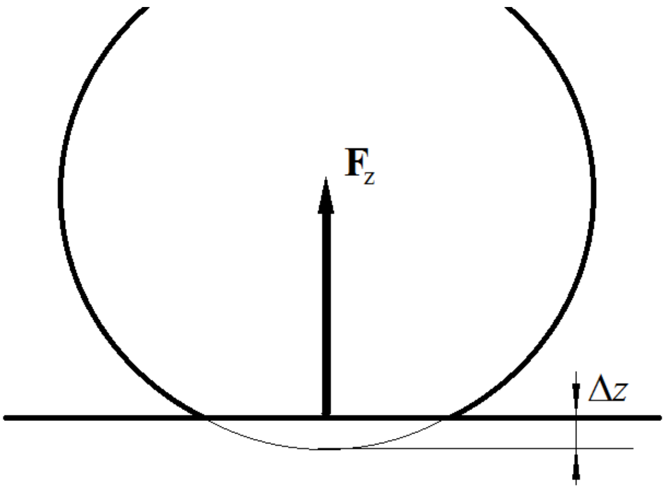


Figure 1.42. Single point contact

The **single point** model is the common method for description of the normal force F_z in the contact between the road and tire. The force depends on the tire deflection Δz, which can be computed as the maximal penetration of the rigid wheel circle with the road line like in Figure 1.42,

$$F_z = F_z(\Delta z, \Delta\dot{z}).$$

Usually a linear dependence of the force on Δz and its time derivative $\Delta\dot{z}$ is used.

$$F_z = -k_z\Delta z - d_z\Delta\dot{z}. \qquad (1.1)$$

The force is applied to the point of the maximal deflection perpendicular to the local road line.

The **multipoint** tire contact model is applied when the road has special deviations like obstacles, potholes or something like that, Figure 1.43. In such cases the tire contact patch may consist of two or more separate sections.

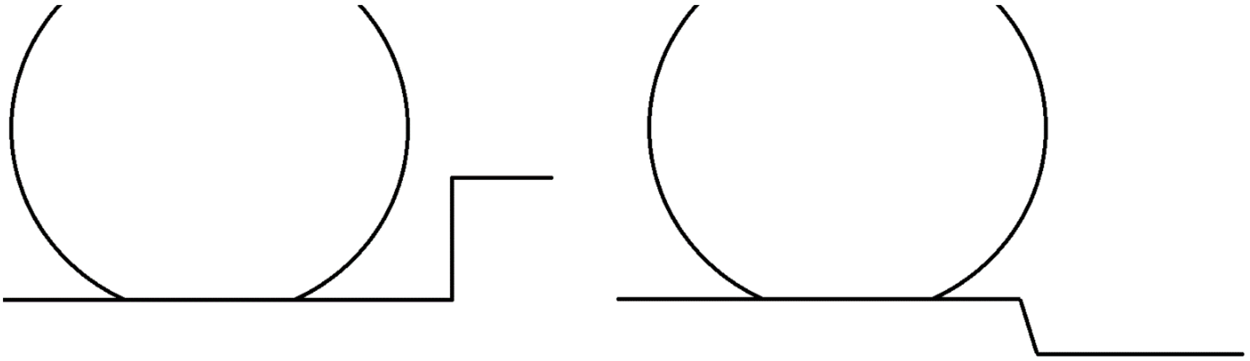


Figure 1.43. Special road deviations

Two different methods are implemented for the multipoint contact:

- discrete point contact
- flexible distributed contact.

In both cases, the regions of intersection between the tire circle and the road line are computed.

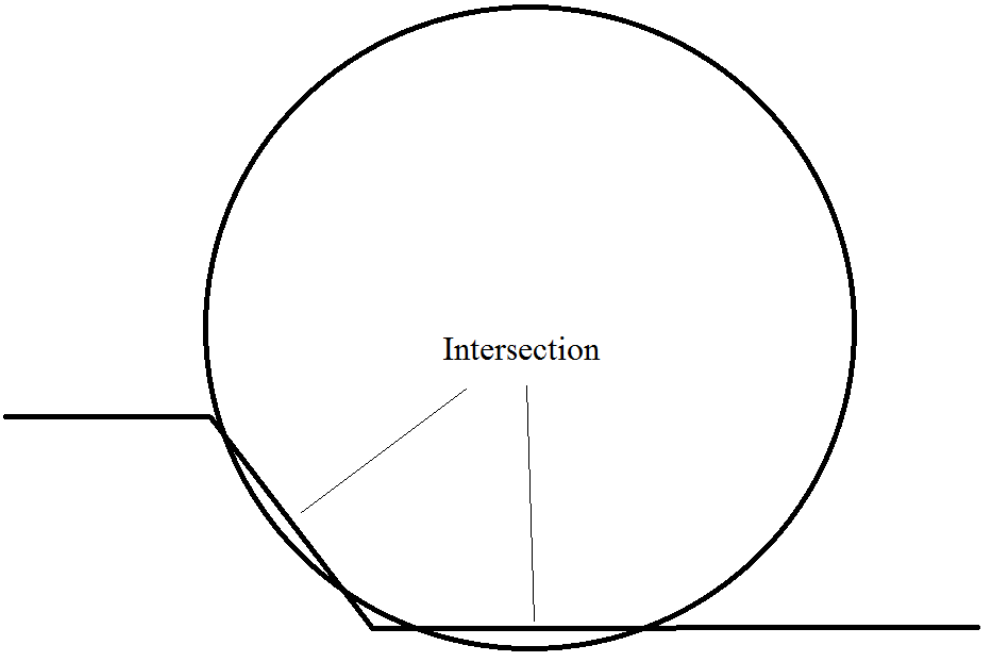


Figure 1.44. Two regions of intersection

If the **discrete point contact** is used, the normal forces at each of the region depend on the maximal penetration depth Δz_i ,

$$F_{zi} = -k_z \Delta z_i - d_z \Delta \dot{z}_i.$$

The forces F_{zi} are applied at the points of the maximal intersection depth and directed *to the center of the wheel*, Figure 1.45.

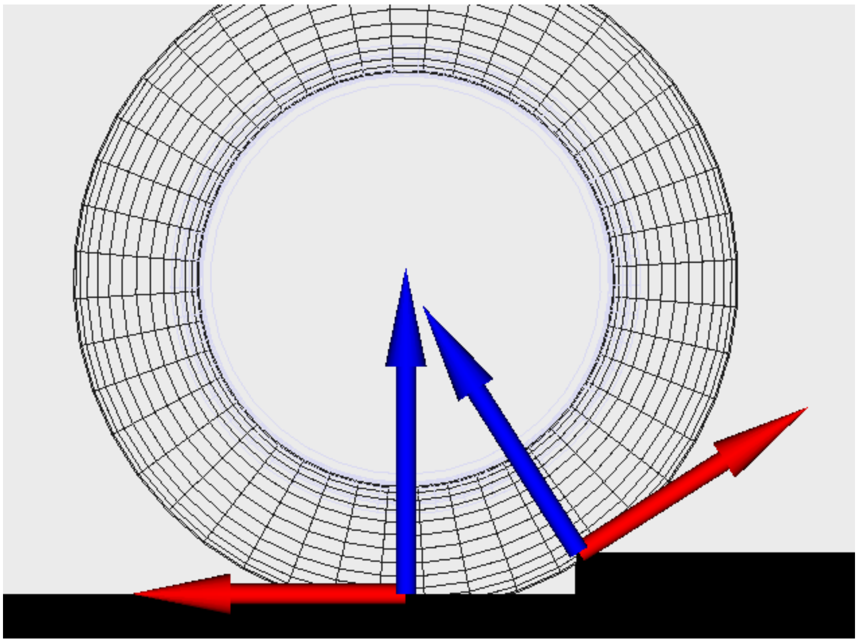


Figure 1.45. Wheel rolling up a step with the discrete point contact model

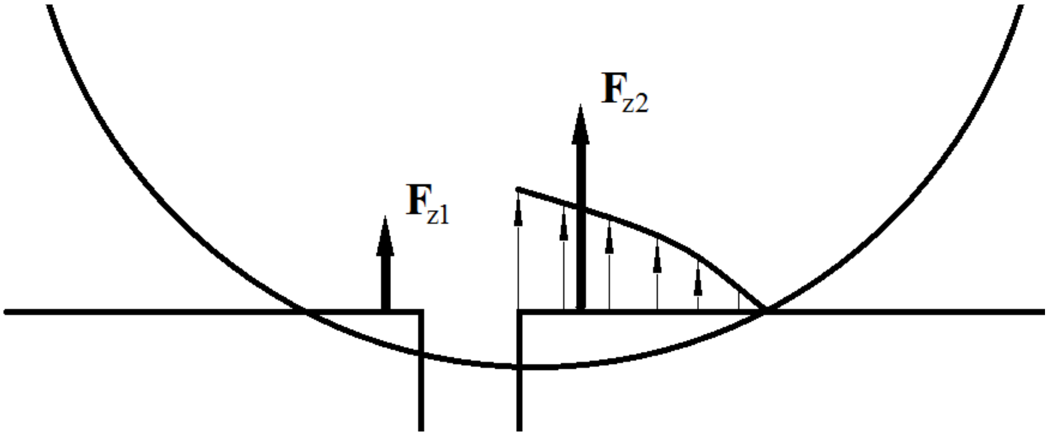


Figure 1.46. Distributed contact model

In the case of the **flexible distributed contact**, the normal force for a separate contact region is computed as a resultant force of a distributed load. The distributed load is proportional to the penetration depth function $q(x)$ along the region, Figure 1.46

$$\mathbf{F}_{zi} = k_{zd} \int_{x_{i1}}^{x_{i2}} q(x)\mathbf{n}(x)dx,$$

where k_{zd} is the distributed contact stiffness constant, and $\mathbf{n}$ is the normal to the road curve. If the road curve is a straight line, the elastic component of the force is proportional to the intersection area,

$$F_{ezi} = k_{zd} \int_{x_{i1}}^{x_{i2}} q(x)dx = k_{zd} A_i, \quad (1.2)$$

Taking into account Eq. (1.2), we can compute the k_{zd} constant equivalent to the tire stiffness k_z from Eq. (1.1),

$$k_{zd} = k_z \Delta z_0 / A_0, \quad (1.3)$$

where Δz_0 is the static tire deflection, and A_0 is the area of intersection on the tire circle with the road line at the static position. This means, all the models will give the same normal force and deflection at static position of the vehicle.

The flexible distributed contact presents a nonlinear dependence of the contact force on the tire deflection Δz. According to Eq. (1.2), for an ideal straight road section

$$F_{ez} = k_{zd} A(\Delta z) = \alpha r^2 - (r - \Delta z) r \sin\alpha \approx k_{zd} \sqrt{2r} \Delta z^{3/2} = \left(k_{zd} \sqrt{2r} \Delta z^{1/2}\right) \Delta z,$$
$$\sin\alpha = \sqrt{\frac{2\Delta z}{r} - \frac{\Delta z^2}{r^2}} \approx \alpha, \quad \alpha \approx \sqrt{\frac{2\Delta z}{r}}. \quad (1.4)$$

Here α is a half of the central wheel angle for the contact patch, and r is the undeformed tire radius.

So, the stiffness is proportional to the square root of the deflection like in the case of the Herz contact.

Choice of the contact model depends on type of the special road deviations. The discrete point contact gives good results for rolling up a step and bad results for run over a small pothole like in Figure 1.47. Backwards, the distributed flexible contact is appropriate for small potholes, and gives bad results for high steps.

The flexible distributed contact is recommended for simulation of motion of a vehicle for a triangulated surface (testing area), 1.9.4.7.3 “*Feature of driver test in case of 3D testing area (triangulated surface)*”.

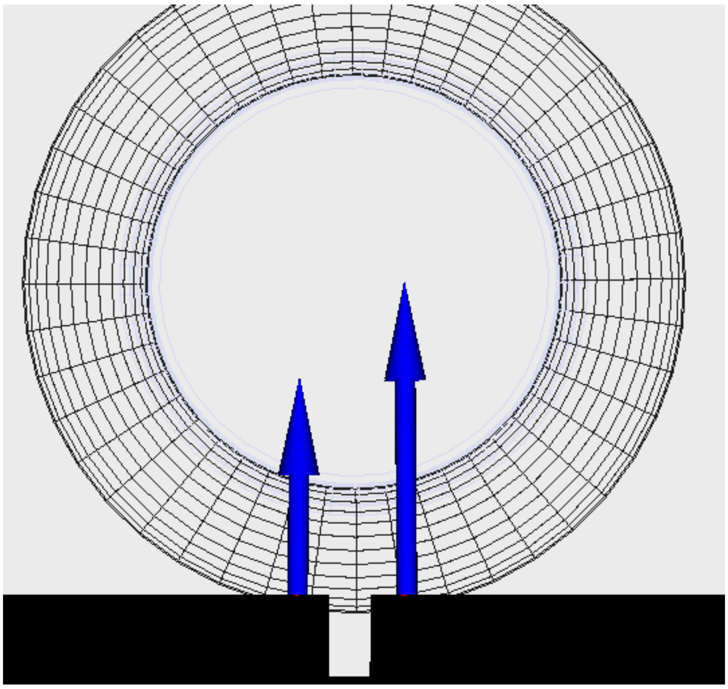


Figure 1.47. Wheel running over a small pothole with the flexible distributed contact model

Remark 1	The multipoint contact uses a stepwise discretization of road curve under the wheel over the interval of two wheel radii. The step value can be varied by the user.
Remark 2	In **UM Automative**, the special road deviations is equivalent to the notion 'Road test section profile', see Sect. 1.9.1.3.2. *Creating files with test section* profiles. In **UM Monarail train** module we use the 'Special track deviations' term, see Chapter 26, Sect. *Special track deviations* (file 26_um_monorail_train.pdf).

1.5.2. FIALA tire model

Assumptions and admissions

- Rectangular contact patch
- Normal contact pressure is constant within the patch
- Tire is modeled by a beam on elastic foundation
- Contact forces do not depend on camber

Contact parameters and variables

Parameter	Parameter of contact model*	Description	Source
α	-	Slip angle	Computation by simulation
γ	-	Camber	Computation by simulation
s_x	-	Longitudinal slip	Computation by simulation
s_y=tan α	-	Sideslip	Computation by simulation
Δr	-	Vertical tire deflection	Computation by simulation
$V_{\Delta r}$r	-	Rate of vertical tire deflection	Computation by simulation
r	R	Radius of unload wheel	Tire description file (*.tr)
k_z	Kz	Tire vertical stiffness constant	Tire description file (*.tr)
k_x	Kx	Tire longitudinal stiffness constant	Tire description file (*.tr)
k_y	Ky	Tire lateral stiffness constant	Tire description file (*.tr)
β_z	BetaZ	Damping ratio of critical	Tire description file (*.tr)
d_z	-	Vertical damping constant. Computed as $d_z = 2\beta_z\sqrt{mk_z}$, where m is the wheel mass, kg.	Precomputation of tire contact forces
µ0	Mu0	Static coefficient of friction	Tire description file (*.tr)
µ1	Mu1	Dynamic coefficient of friction	Tire description file (*.tr)
Cx	Cx	Longitudinal stiffness	Tire description file (*.tr)
Cy	Cy	Cornering stiffness	Tire description file (*.tr)
c_γ	Cgamma	Coefficient at the camber angle when calculating the aligning	Tire description file (*.tr)

		moment M_x	
rt	Rtorus	Toroidal radius of tire	Tire description file (*.tr)

* Designation in the Wizard of tire models, Sect. 1.5.8. *"Tire model wizard"*, p. 1-64.

Vertical force (F_z)

(1) Linear viscous-elastic force

$$F_z = -k_z \Delta r - d_z V_{\Delta r}.$$

(2) If the wheel detaches the supporting surface ($\Delta r > 0$) or the computed value is negative $F_z < 0$, the vertical force is zero, $F_z = 0$.

Longitudinal force (F_x)

$$s = \sqrt{s_x^2 + s_y^2}$$

$$\mu = \mu_0 + (\mu_1 - \mu_0)s$$

$$s^* = \frac{\mu F_z}{2c_x}$$

Case 1. $|s_x| < s$

$$F_x = s_x c_x$$

Case 2. $|s_x| \geq s^*$

$$F_x = sign(s_x)\left[\mu F_z - \frac{(\mu F_z)^2}{4|s_x|c_x}\right]$$

Side force (F_y)

$$s' = \frac{3\mu F_z}{c_y}$$

Case 1. $|s_y| < s'$

$$h = 1 - \frac{c_y|s_y|}{3\mu F}$$

$$F_y = \mu F_z(1 - h^3)sign(s_y)$$

Case 2. $|s_y| \geq s'$

$$F_y = \mu F_z sign(s_y)$$

Self-aligning torque (M_z)

Case 1. $|s_y| < s'$

$$M_z = -2\mu F_z r_t (1 - h)h^3 sign(s_y)$$

Case 2. $|s_y| \geq s'$

$$M_z = 0$$

Overturning moment (M_x)

$$M_x = -c_\gamma \gamma$$

1.5.3. Pacejka Magic Formula

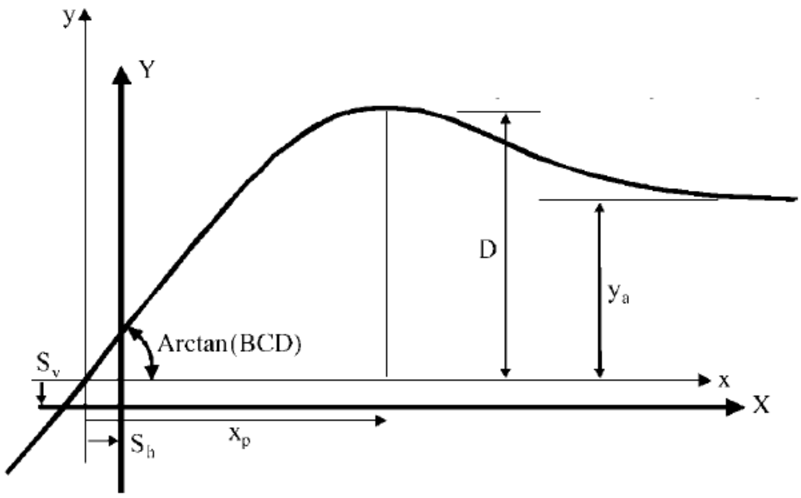


Figure 1.48. Magic formula

The Magic Formula (MF) is (Figure 1.48):

$$Y(x) = D sin\left[C arctan\left\{Bx - E\left(Bx - arctan(Bx)\right)\right\}\right] + S_v,$$
$$x = X + S_h$$

Here $Y(x)$ can be longitudinal (F_x), side (F_y) force or aligning torque (M_z), and X is the longitudinal creep (F_x) or the sideslip (F_y, M_z).

According to [5] [6], the MF coefficients are functions of the vertical load F_z and the camber angle γ.

1. Longitudinal force F_x.

$$C = b_0,$$
$$D = F_z(b_1 F_z + b_2),$$
$$B = \frac{1}{CD}(b_3 F_z^2 + b_4 F_z) e^{-b_5 F_z},$$
$$E = b_6 F_z^2 + b_7 F_z + b_8,$$
$$S_h = b_9 F_z + b_{10},$$
$$S_v = b_{11} F_z + b_{12}.$$

2. Side force F_y.

$$C = a_0,$$

$$D = F_z(a_1 F_z + a_2),$$
$$E = a_6 F_z + a_7,$$
$$B = \frac{1}{CD} a_3 sin\big(a_{15} arctan(F_z/a_4)\big)(1 - a_5|\gamma|),$$
$$S_h = a_8\gamma + a_9 F_z + a_{10},$$
$$S_v = (a_{11}F_z + a_{12})\gamma F_z + a_{13}F_z + a_{14}.$$

3. Aligning torque M_z.

$$C = c_0,$$
$$D = (c_1 F_z + c_2)F_z,$$
$$E = (c_7 F_z^2 + c_8 F_z + c_9)(1 - c_{10}|\gamma|),$$
$$B = \frac{1}{CD}(c_3 F_z^2 + c_4 F_z)(1 - c_6|\gamma|)e^{-c_5 F_z},$$
$$S_h = c_{11}\gamma + c_{12}F_z + c_{13},$$
$$S_v = (c_{14}F_z^2 + c_{15}F_z)\gamma + c_{16}F_z + c_{17}.$$

Use of these formulas requires fitting the coefficients $a_0..a_{15}, b_0..b_{10}, c_0..c_{17}$ with the help of test data. The default values of the coefficients in UM are obtained from [5]:

$$a_0 = 1.3, a_1 = -22.1, a_2 = 1011, a_3 = 1078, a_4 = 4.902, a_5 = 0.022, a_6 = -0.354, a_7 = 0.707, a_8 = 0.029, a_9 = 0, a_{10} = 0, a_{11} = 14.8, a_{12} = 0, a_{13} = 0, a_{14} = 0, a_{15} = 1.82$$
$$b_0 = 1.65, b_1 = -21.3, b_2 = 1144, b_3 = 49.6, b_4 = 226, b_5 = 0.069, b_6 = -0.006, b_7 = 0.056, b_8 = 0.486, b_9 = 0, b_{10} = 0$$
$$c_0 = 2.4, c_1 = -2.72, c_2 = -2.28, c_3 = -1.86, c_4 = -2.73, c_5 = 0.11, c_6 = 0.03, c_7 = -0.07, c_8 = 0.643, c_9 = -4.04, c_{10} = 0.03, c_{11} = 0.015, c_{12} = 0, c_{13} = 0, c_{14} = -0.066, c_{15} = 0.945, c_{16} = 0, c_{17} = 0$$

Plots of the longitudinal and side forces as well as the aligning torque in dependence on the corresponding slip by $\gamma = 0$ for different values of the vertical load are shown in Figure 1.50, Figure 1.51. The MF with the above values of the parameters was used for computation of the forces.

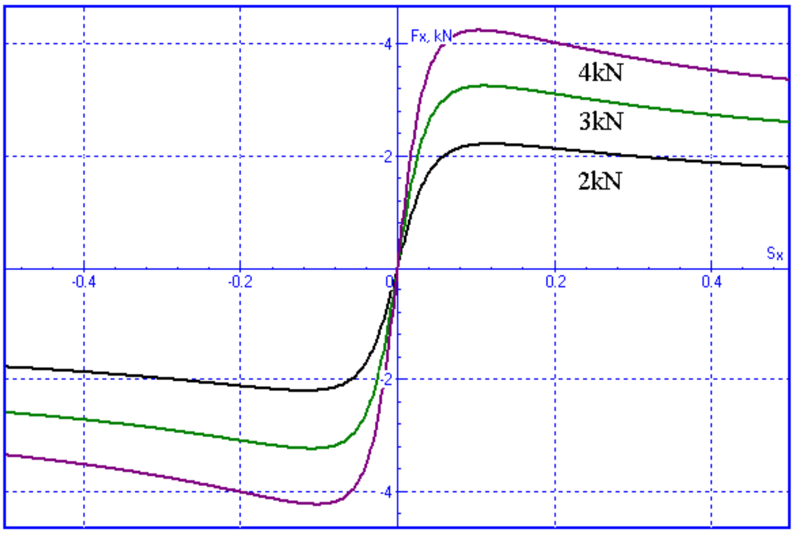


Figure 1.49. Longitudinal force

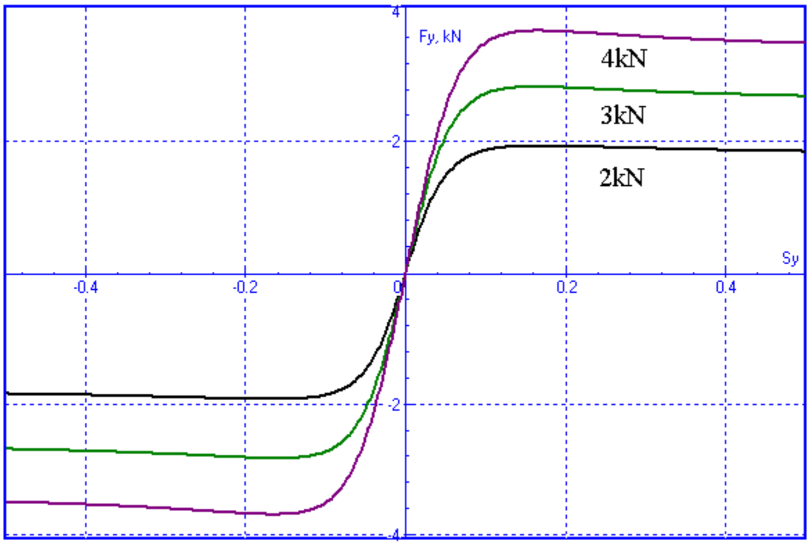


Figure 1.50. Side force

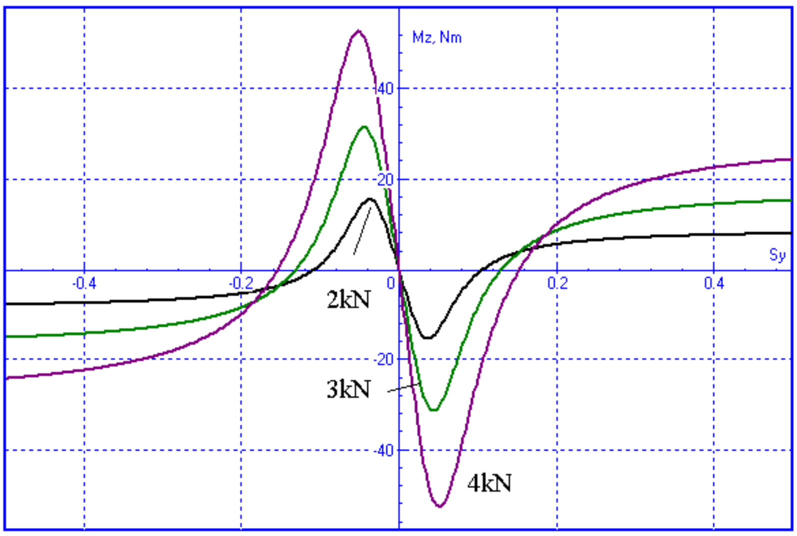


Figure 1.51. Aligning torque

1.5.4. Tabular tire model

Tabular model of a tire requires experimental data on the longitudinal, side forces and aligning torque, Figure 1.52, [7].

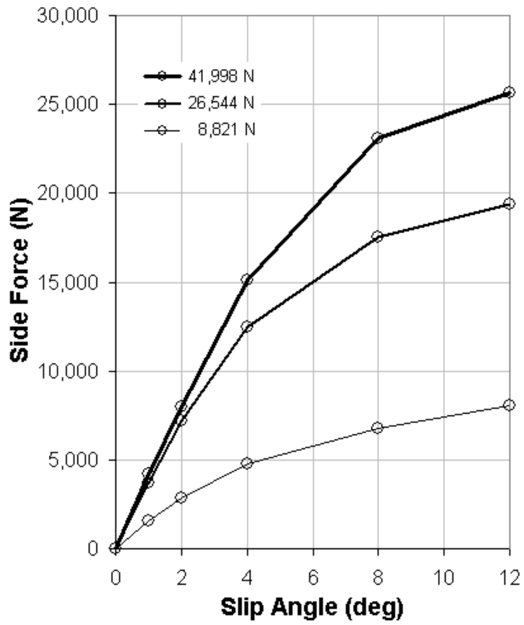


Figure 1.52. Tabular Side Force

Table 1.5

Tire Side Force Characteristics

Slip Angle (deg)	Vertical Force (kN)		
	8,821	26,544	41,998
	Lateral Force(kN)		
1.00	1,587.8	3,716.2	4,199.3
2.00	2,822.8	7,166.9	7,979.6
4.00	4,763.5	12,475.6	15,119.2
8.00	6,792.4	17,519.0	23,098.8
12.00	8,027.4	19,377.1	25,618.7

The tabular model is implemented in UM with the following assumptions:

- forces do not depend on camber;
- force plots are antisymmetric functions of slips.

Let $Y(x_j, F_{zj}), i = 1..N_Y, j = 1..N_{Fz}$ are the tabular data. A smoothed model of the force is obtained with two steps. First, a beta-spline approximation $\hat{Y}(x, F_{zj})$ of the discrete function $Y(x_j, F_{zj})$ is developed for each value of the vertical force F_{zj}. This operation can be done with the help of the curve editor (see Chapter 3. Sect. *Curve Editor*). If necessary, additional points should be added to the curve to improve the approximation accuracy, Figure 1.53.

Finally, the second order Lagrange interpolation polynomials are used to compute a smoothed value of the force for definite values of the slip x and the load F_z

$$Y(x, F_z) = P\left(F_z, \hat{Y}(x, F_{z1}), \dots, \hat{Y}\left(x, F_{zN_{Fz}}\right)\right).$$

An example of smoothed tabular model of a side force is shown in Figure 1.54, Figure 1.55.

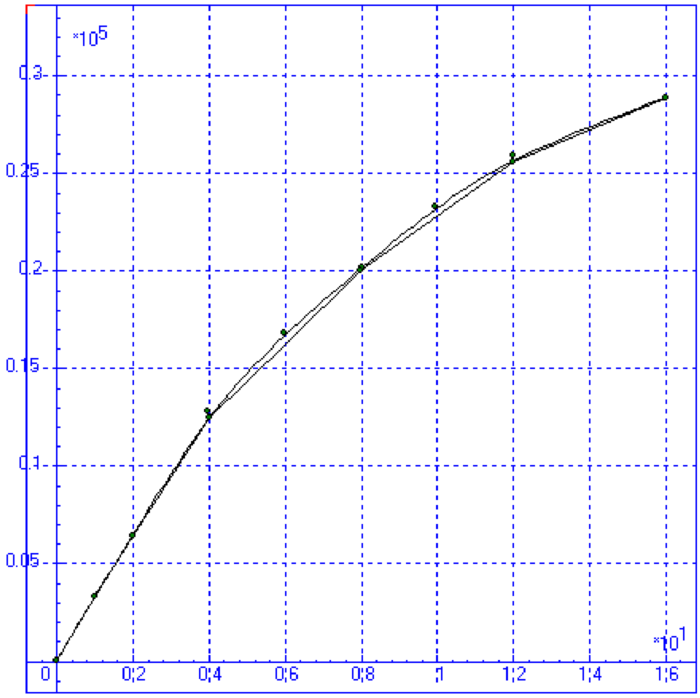


Figure 1.53. Polygon and smoothed curve

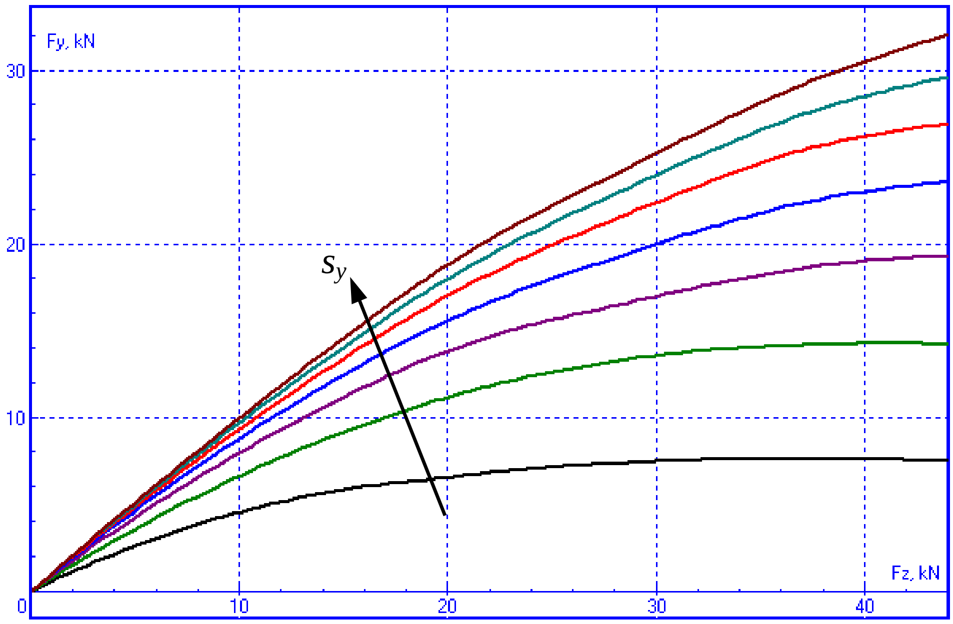


Figure 1.54. Smoothed model of the side force versus vertical load

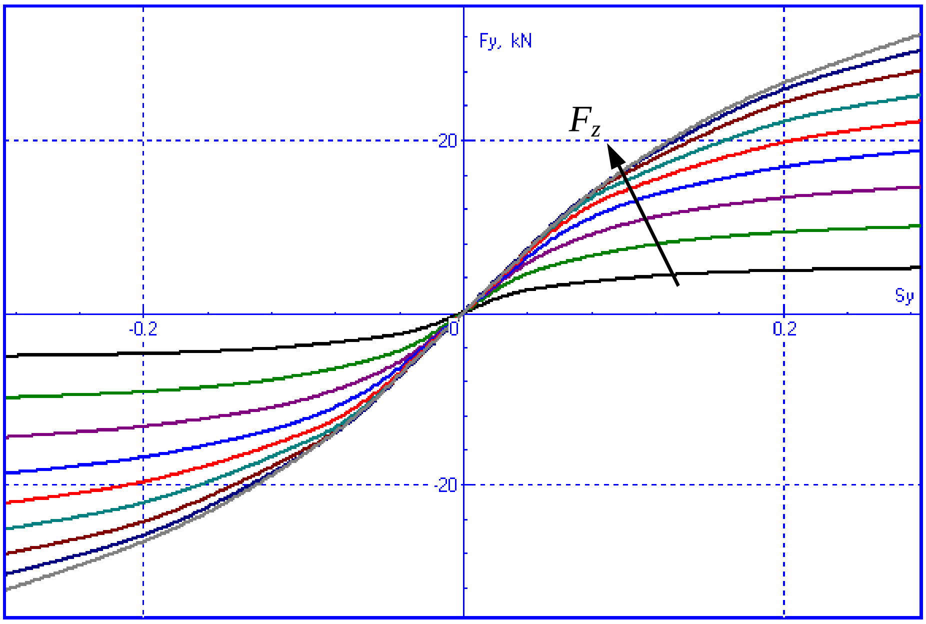


Figure 1.55. Smoothed model of the side force versus side slip

1.5.5. TMEasy tire model

The aim of TMEasy is to give useful tire forces from little information with model parameters that have physical meaning ([8], page 67, [9]).

Assumptions and admissions

- Contact forces do not depend on camber
- TMEasy simulates the tire behavior in combined slip in combined slip situations by generalizing the tire characteristics through a normalization process
- Self-aligning torque M_z is a function of lateral force F_y

Longitudinal force (F_x)

As shown in Figure 1.56, a typical longitudinal force F_x as a function of longitudinal slip s_x can be described by the following parameters:

- Initial inclination (longitudinal stiffness) dF_x^0
- Maximum longitudinal force F_x^M
- Longitudinal slip at maximum force s_x^M
- Sliding force F_x^S
- Longitudinal slip at sliding force s_x^S

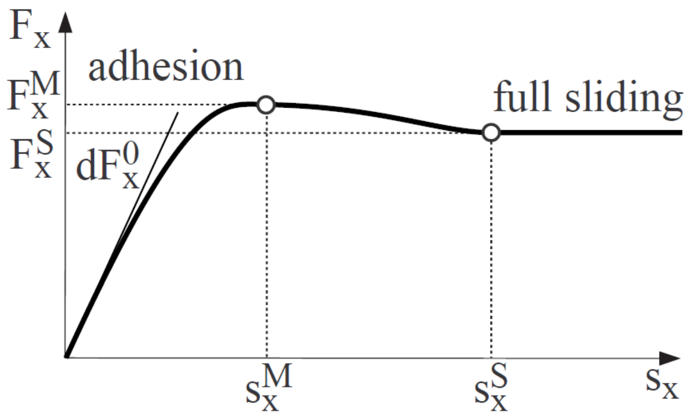


Figure 1.56. Typical longitudinal force characteristics

Lateral force (F_y)

The parameters describing the lateral force are:

- Initial inclination (cornering stiffness) dF_y^0
- Maximum lateral force F_y^M
- Lateral slip at maximum force s_y^M
- Sliding force F_y^S
- Lateral slip at sliding force s_y^S

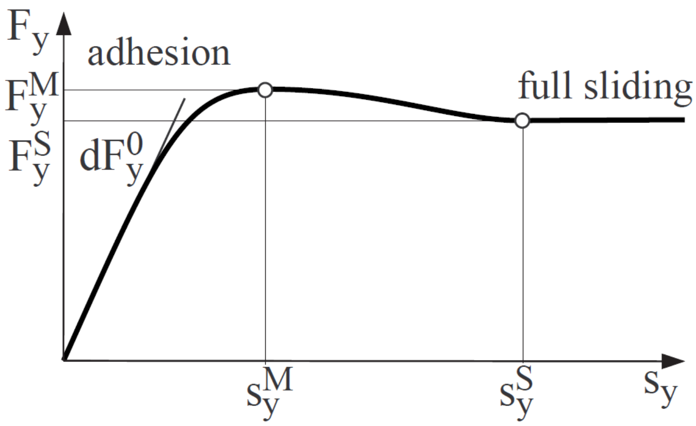


Figure 1.57. Typical lateral force characteristics

Combined slip

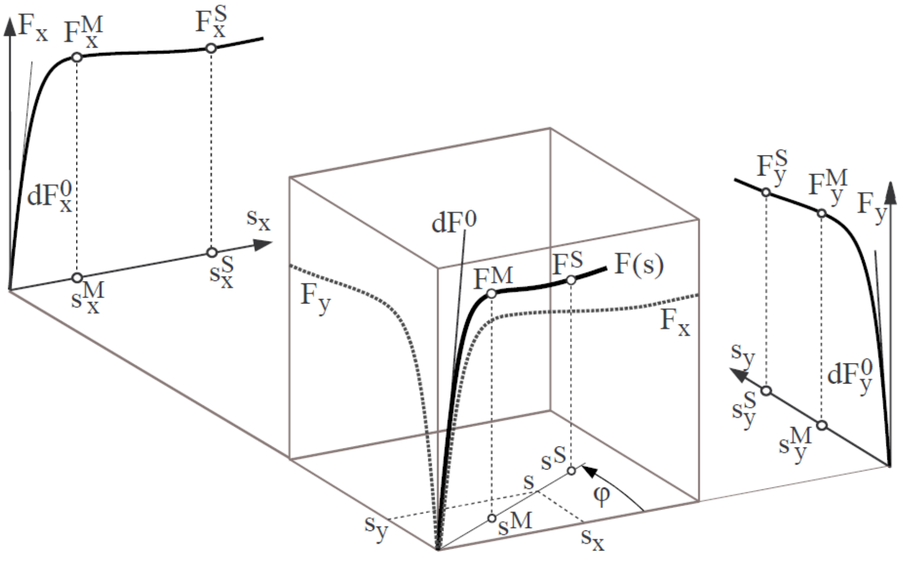


Figure 1.58. Generalized tire characteristics

The longitudinal slip s_x and lateral slip s_y can be generalized by vector addition as:

$$s = \sqrt{\left(\frac{s_x}{\hat{s}_x}\right)^2 + \left(\frac{s_y}{\hat{s}_y}\right)^2} \qquad (1.5)$$

Normation factors:

$$\hat{s}_x = \frac{s_x^M}{s_x^M + s_y^M} + \frac{F_x^M/dF_x^0}{F_x^M/dF_x^0 + F_y^M/dF_y^0}$$
$$\hat{s}_y = \frac{s_y^M}{s_x^M + s_y^M} + \frac{F_y^M/dF_y^0}{F_x^M/dF_x^0 + F_y^M/dF_y^0} \qquad (1.6)$$

The generalized tire parameters are then calculated with the corresponding values of the longitudinal and lateral tire parameters and normalization factors:

$$dF^0 = \sqrt{(dF_x^0 \cdot \hat{s}_x \cdot \cos\varphi)^2 + \left(dF_y^0 \cdot \hat{s}_y \cdot \sin\varphi\right)^2}$$
$$s^M = \sqrt{\left(\frac{s_x^M}{\hat{s}_x}\cos\varphi\right)^2 + \left(\frac{s_x^M}{\hat{s}_x}\sin\varphi\right)^2}$$
$$F^M = \sqrt{(F_x^M\cos\varphi)^2 + \left(F_y^M\sin\varphi\right)^2} \qquad (1.7)$$
$$s^S = \sqrt{\left(\frac{s_x^S}{\hat{s}_x}\cos\varphi\right)^2 + \left(\frac{s_x^S}{\hat{s}_x}\sin\varphi\right)^2}$$
$$F^S = \sqrt{(F_x^S\cos\varphi)^2 + \left(F_y^S\sin\varphi\right)^2}$$

where

$$\cos\varphi = \frac{s_x/\hat{s}_x}{s} \qquad \sin\varphi = \frac{s_y/\hat{s}_y}{s} \tag{1.8}$$

The function F = F(s) can be described in intervals by a broken rational function, a cubic polynomial and a constant F^G

$$F(s) = \begin{cases} s^M dF^0 \dfrac{\sigma}{1+\sigma\left(\sigma + dF^0 \dfrac{s^M}{F^M} - 2\right)}, & \sigma = \dfrac{s}{s^M} \qquad 0 \le s \le s^M \\ F^M - (F^M - F^S)\sigma^2(3-2\sigma), & \sigma = \dfrac{s - s^M}{s^S - s^M} \qquad s^M \le s \le s^S \\ F^S & s < s^S \end{cases} \tag{1.9}$$

By projecting the generalized force in longitudinal and lateral directions, the corresponding forces can be obtained:

$$F_x = F\cos\varphi \qquad F_y = F\sin\varphi \tag{1.10}$$

Self-aligning torque (M_z)

The self-aligning torque M_z is then obtained by multiplying the resultant lateral force F_y by the dynamic tire offset or pneumatic trail *n*:

$$M_z = -nF_y, \tag{1.11}$$

The dynamic offset is approximated as function of the lateral slip by a line and a cubic polynomial:

$$\frac{n}{L} = \left(\frac{n}{L}\right)_0 \begin{cases} 1 - |s_y|/s_y^0 & |s_y| \le s_y^0 \\ \dfrac{|s_y| - s_y^0}{s_y^0}\left(\dfrac{s_y^E - |s_y|}{s_y^E - s_y^0}\right)^2 & s_y^0 \le |s_y| \le s_y^E \\ 0 & |s_y| > s_y^E \end{cases} \tag{1.12}$$

Where:

$\frac{n}{L}$ is a dynamic tire offset which is normalized by the length of the contact area L

$\left(\frac{n}{L}\right)_0$ is an initial value of normalized dynamic tire offset

L is the length of the contact area

s_y^0 , s_y^E are additional model parameters – slip values, where the normalized dynamic tire offset passes the s_y – axis and reaches zero again.

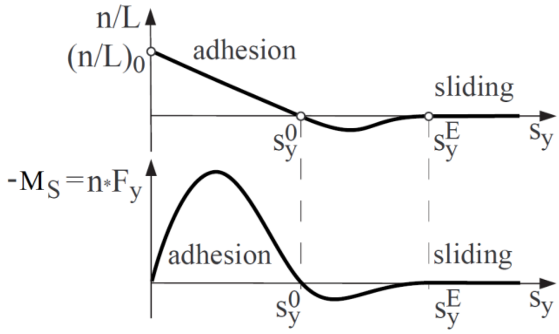


Figure 1.59. Typical plots of dynamic offset and self-aligning torque

List of tire parameters

The TMEasy model depends on 13 parameters for single value of vertical load F_z :

$dF_x^0, F_x^M, s_x^M, F_x^S, s_x^S, dF_y^0, F_y^M, s_y^M, F_y^S, s_y^S, \left(\frac{n}{L}\right)_0, s_y^0, s_y^E$

The full model description includes numerical values of these parameters for the *Nominal normal load* $F_{z,\text{norm}}$ as well for two times the normal load $2F_{z,\text{norm}}$.

TMEasy example [9]

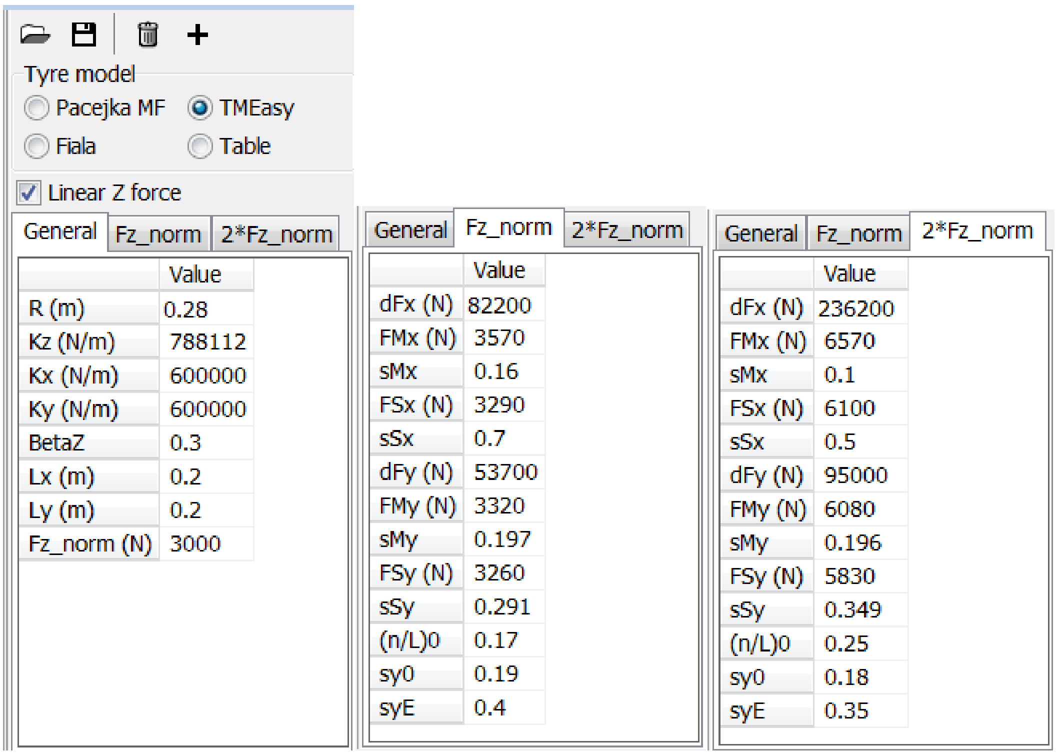


Figure 1.60. TMEasy in wizard of tire parameters

File with tire parameters:

```
tmeasy.tr

runloaded=0.28;
cstiffz=788112;
cstiffx=600000;
cstiffy=600000;
```

```
dampingratioz=0.3;
relaxationx=0.2;
relaxationy=0.2;
linearzforce=true;

with tire_tmeasy;
  fx1_Fz=3000;
  fx1_dF=82200;
  fx1_FM=3570;
  fx1_sM=0.160;
  fx1_FS=3290;
  fx1_sS=0.700;

  fx2_Fz=6000;
  fx2_dF=236200;
  fx2_FM=6570;
  fx2_sM=0.100;
  fx2_FS=6100;
  fx2_sS=0.500;

  fy1_Fz=3000;
  fy1_dF=53700;
  fy1_FM=3320;
  fy1_sM=0.197;
  fy1_FS=3260;
  fy1_sS=0.291;

  fy2_Fz=6000;
  fy2_dF=95000;
  fy2_FM=6080;
  fy2_sM=0.196;
  fy2_FS=5830;
  fy2_sS=0.349;

  mz1_nL=0.17;
  mz1_s0=0.190;
  mz1_sE=0.400;

  mz2_nL=0.25;
  mz2_s0=0.180;
  mz2_sE=0.350;
```

1.5.6. Combined slip

The combined slip option in computation of tire forces is used if both longitudinal slip and sideslip are not small.

Let α be the slip angle and s_x be the lateral slip. The longitudinal and side forces are computed according to the formulas [10]

$$\sigma_x = \frac{s_x}{1+s_x}, \sigma_y = \frac{tan\alpha}{1+s_x},$$

$$\sigma = \sqrt{\sigma_x^2 + \sigma_y^2},$$

$$F_x = \frac{\sigma_x}{\sigma} F_{x0}(\sigma), F_y = \frac{\sigma_y}{\sigma} F_{y0}(\sigma).$$

Here F_{x0}, F_{y0} are the dependences of the longitudinal and side forces on slips described above.

The combined slip option is available on the Road vehicle | Tires tab of the simulation inspector, Figure 1.61. It is recommended to check the option for tests with braking processes.

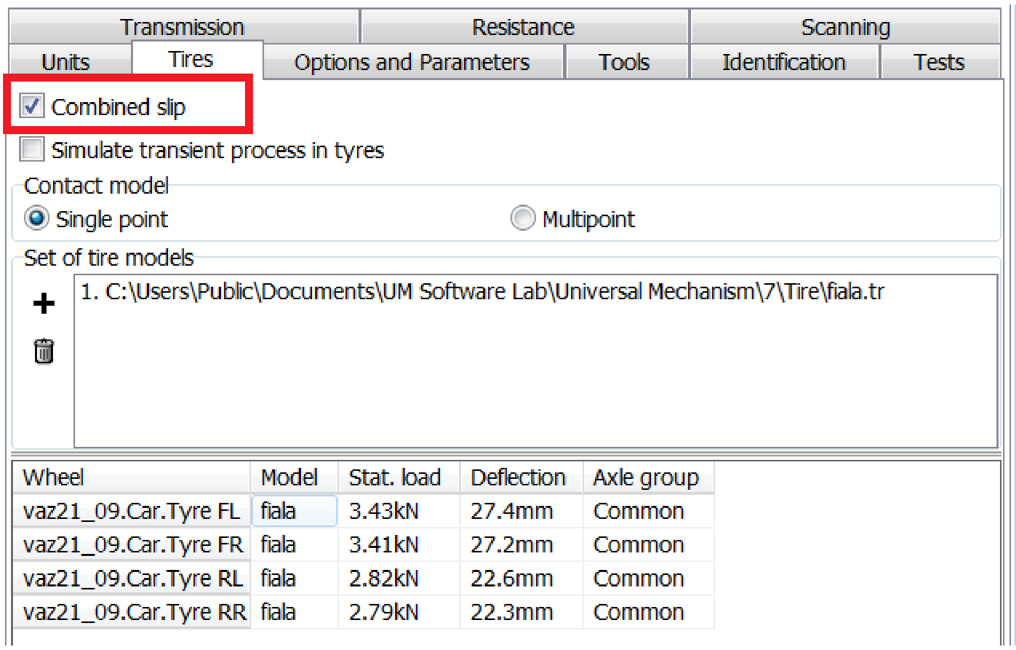


Figure 1.61. Combined slip option

1.5.7. Transient processes in tire

A simplified model of transient processes in tire were proposed in [20]. The transient process affects the sideslip $\lambda = tg\alpha$ and lateral slip s_x. The following first order differential equations specify the slip values

$$\frac{L_y}{v_x}\frac{d\lambda}{dt} + \lambda = \frac{v_y}{v_x},$$

$$\frac{L_x}{v_x}\frac{ds_x}{dt} + s_x = \frac{\omega R - v_x}{v_x},$$

where v_x, v_y, ω are the longitudinal and lateral velocities as well as the angular velocity of the wheel, R is the wheel rolling radius, L_x, L_y are the so called tire relaxation length in the longitudinal and lateral directions.

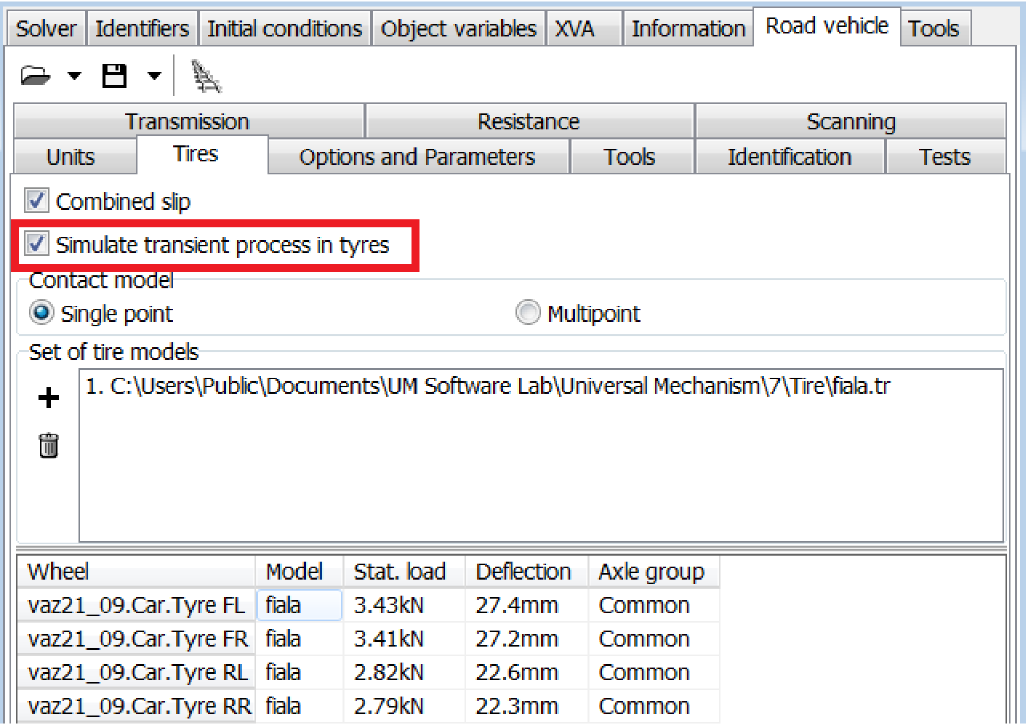


Figure 1.62. Option for transient processes in tire

Check the **Simulate transient process in tire** option in the inspector to activate the transient model, Figure 1.62.

The main effect of the transient process consists in a delay of λ and s_x values in comparison with the $\frac{v_y}{v_x}$ and $\frac{\omega R - v_x}{v_x}$ values. The delay depends on lengths of relaxations, i.e. on the time constants $\frac{L_y}{v_x}$ and $\frac{L_x}{v_x}$, Figure 1.63. The relaxation lengths are specified in meters in the tire wizard, Figure 1.64.

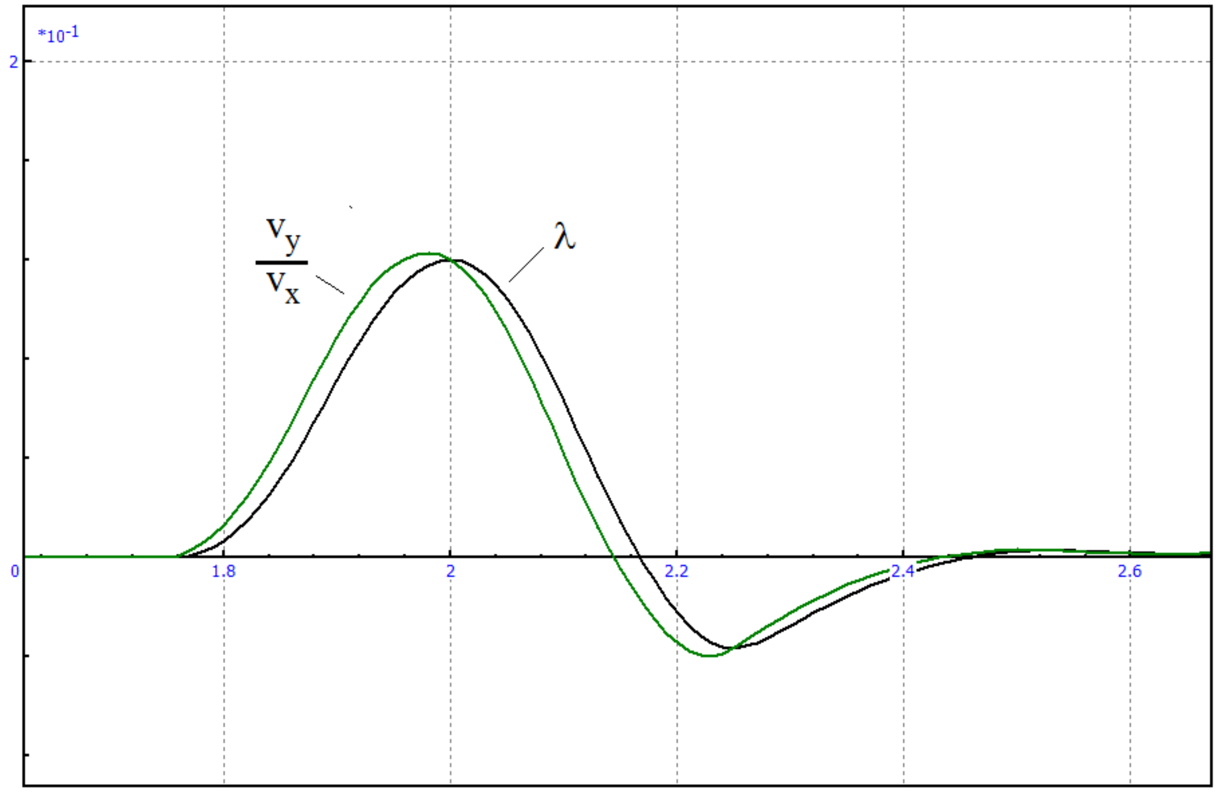


Figure 1.63. Comparison of λ and v_y/v_x in a pulse steer test

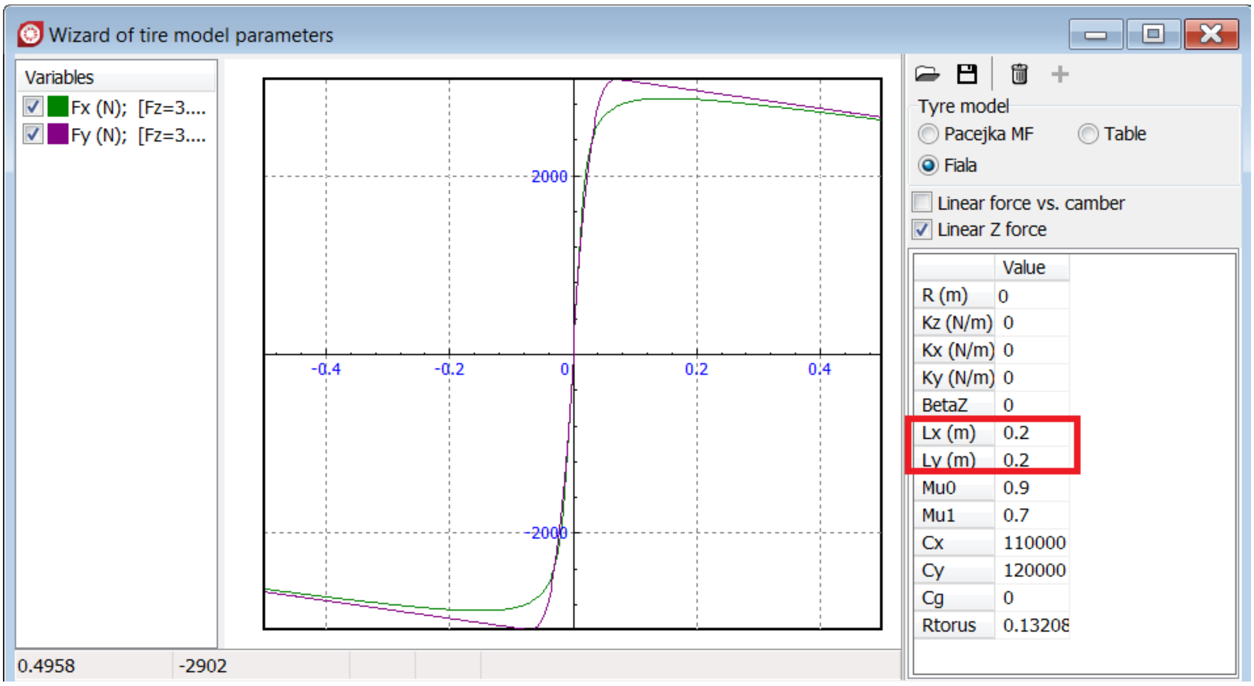


Figure 1.64. Setting tire relaxation lengths

1.5.8. Tire model wizard

Tire models are developed and analyzed with the help of **Wizard of tire models**, Figure 1.65. Use the **Tools | Wizard of tire models…** menu command to open the window. This tool is used to set parameters of a tire and to save them in a *.tr file, which can be later used in simulations.

The wizard allows visualizing the models as well.

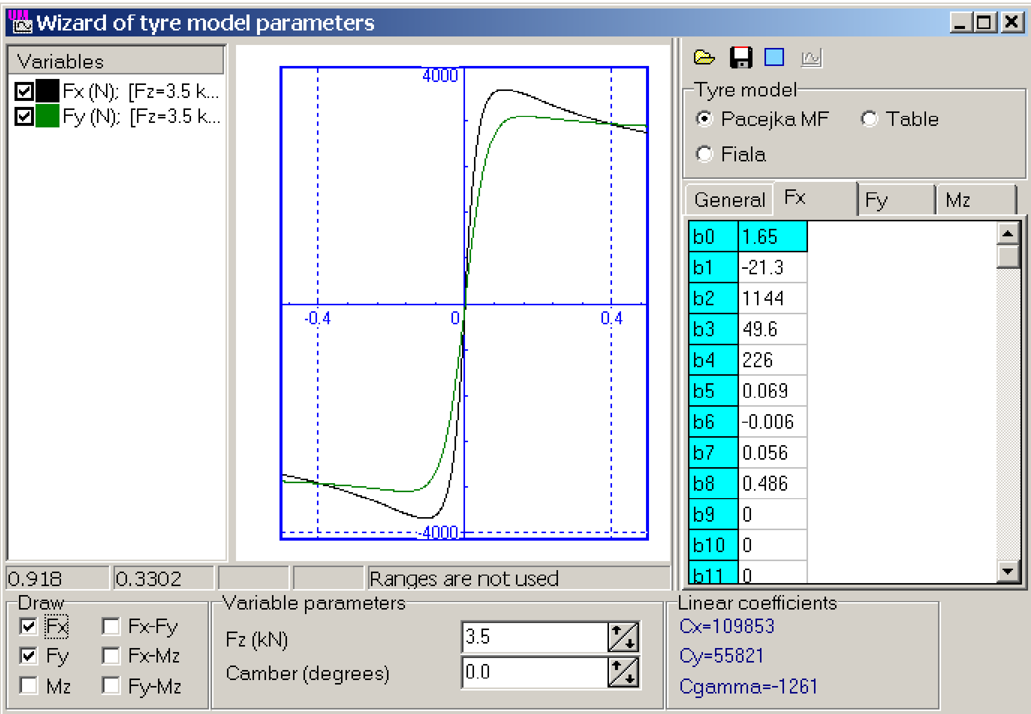


Figure 1.65. Wizard of tire models

Sequence of model development

- Select a desired model in the **Tire model** group or open an existing file by the button.
- Set tire model parameters in the right part of the window.
- Verify correctness of data by plotting the dependences of forces on slips.
- Save parameters to a file **.tr* by the button.

Drawing plots

To draw plots:

- Select desired plots in the **Draw** group.
- Set values of the vertical load and the camber in the **Variable parameters** group. Note that both **Fiala** and **Table** models do not depend on the camber.
- Click the button to draw plots.

Note. Tabular tire model files are currently created using an external text editor. The wizard is used to visualize the data only.

1.5.9. Assignment of tire models to wheels

Use the **Road vehicle | Tires** tab of the **Object simulation inspector** to assign a tire model to wheels.

- Use the to add/delete a file with the tire model to/from the list.
- Call the popup menu to assign a model from the list to the selected wheel or to all of the wheels.

These settings are saved in the vehicle configuration file **.car* by clicking the button.

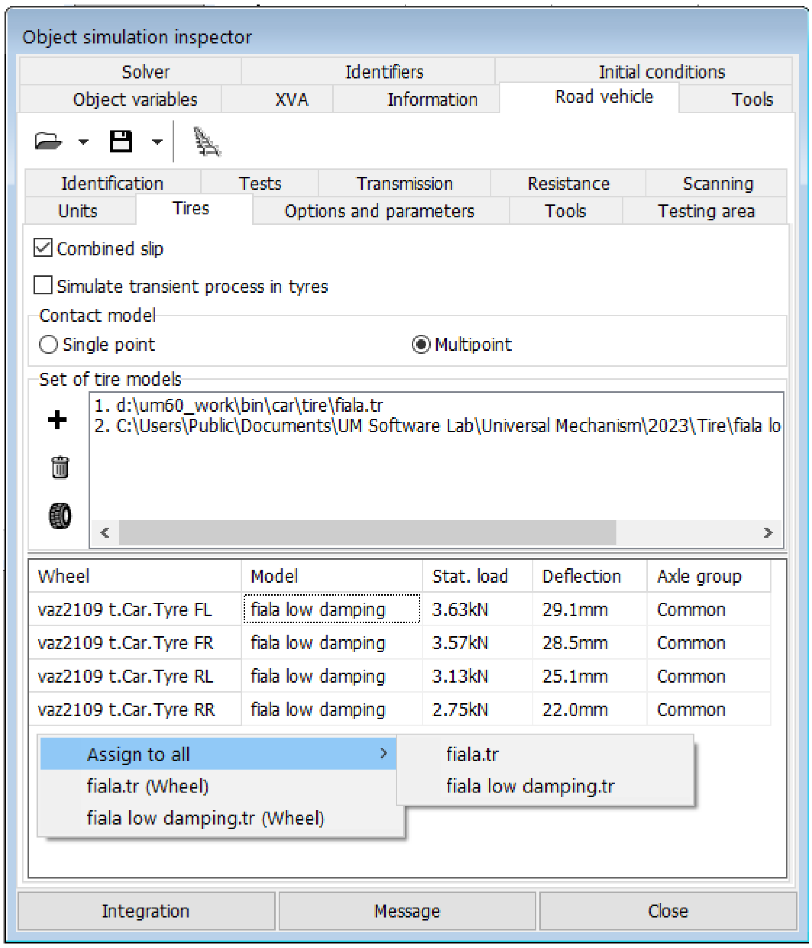
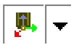


Figure 1.66. Assignment of tire models

1.5.10. Visualization of tire forces

Vertical, lateral and longitudinal contact forces between the wheel and the road can be displayed in the animation window. To hide/show contact forces, use (Figure 1.67)

• for the new animation window: command **Show vectors of tire/road interaction**

• for the old animation window: the button  on the window top.

Force vector animation parameters are described in 0 “

Animation of tire-road forces”.

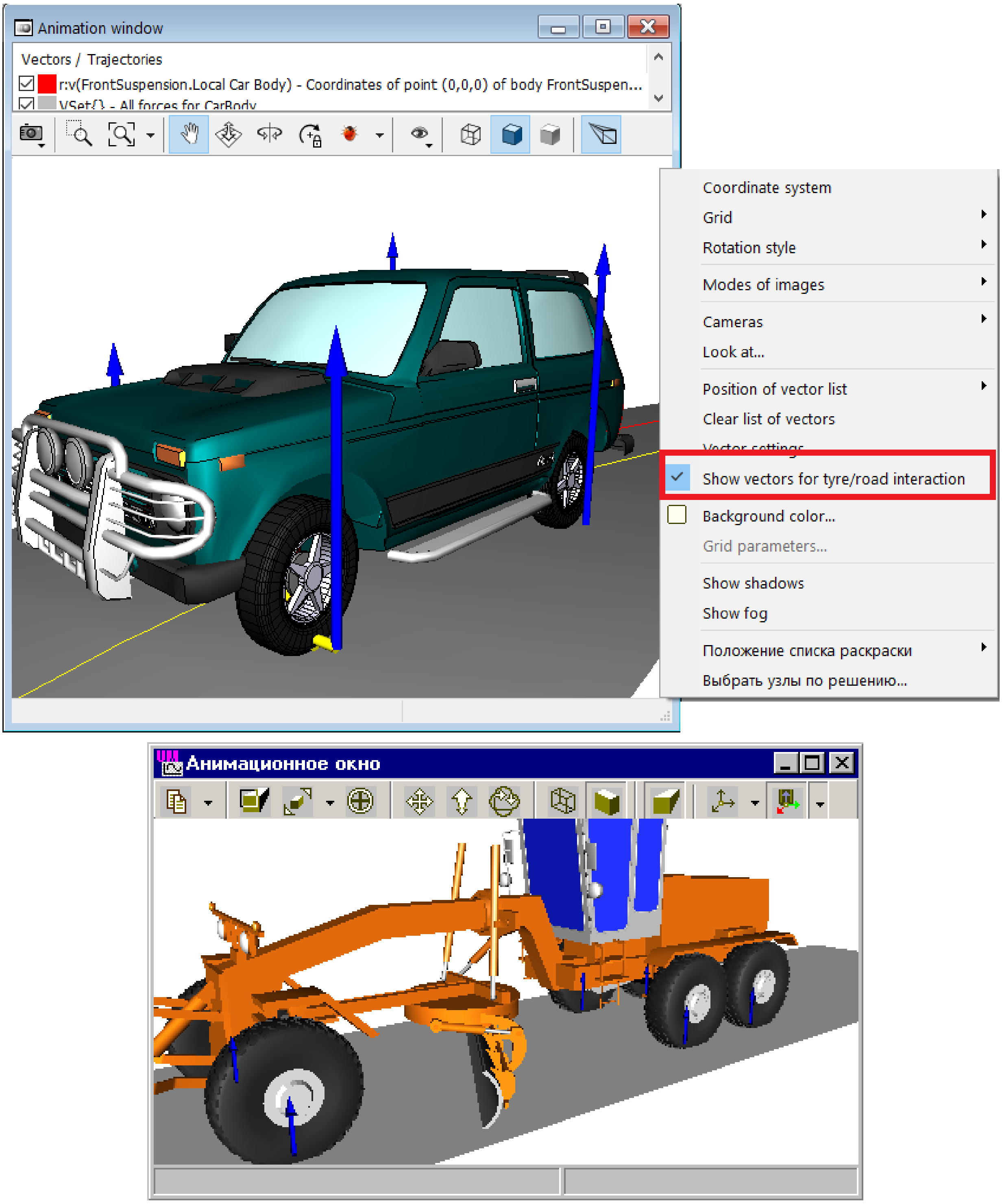


Figure 1.67. Tire forces

1.5.11. Generation of approximate FIALA tire model according to rated parameters

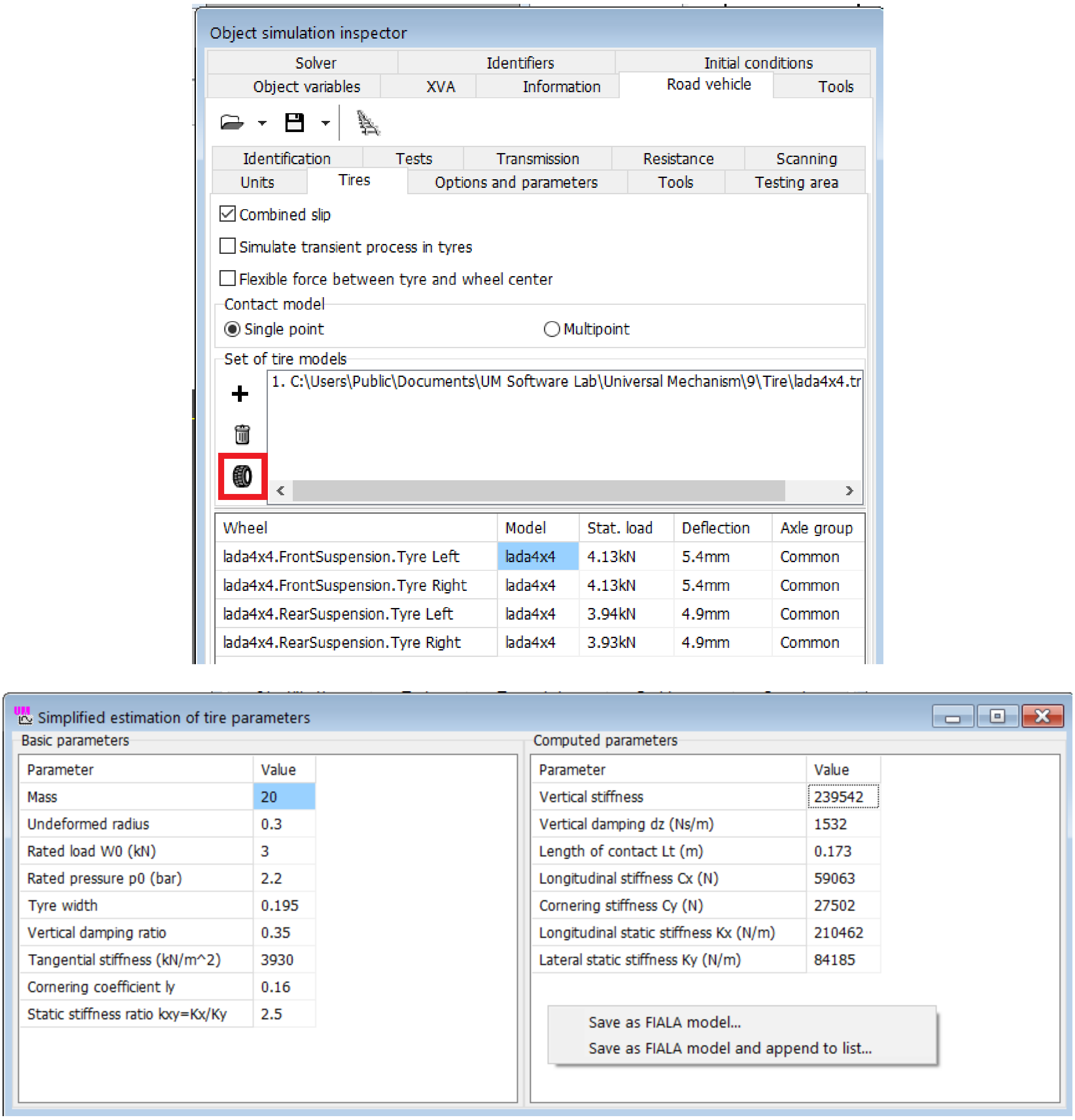


Figure 1.68. Calculation of estimated parameters of tires and saving in a file

Often the values of a tire parameters included in the models described above are unknown. In this case, approximate estimates of these parameters may be useful, allowing a simple model to be built, such as Fiala, Sect. 1.5.2 “*FIALA tire model*”.

The following tire parameters are used for the calculation:

r_w is the tire radius in undeformed state;

w is the tire section width;

W_0 is the rated tire load;

$\beta_z \in [0.3{,}0.75]$ is the vertical damping ratio for the tire; the parameter is used for evaluation of the tire damping constant d_{z0} in the model of the vertical force;

k_t is the tangential tire stiffness, kPa, the default value is 3930 kPa;

$\lambda_y \in [0.8{,}0.18]$ is the cornering coefficient for evaluation of the tire side stiffness c_{y0};

$k_{xy} = K_x/K_y$ is the ratio of the static stiffness constants of the tire (longitudinal to lateral)

To open the window for calculation of tire parameters, use the button on the **Road vehicle | Tires** tab of the simulation inspector, Figure 1.68. The tire parameters are entered in the left part of the window, the calculated values are displayed in the right part. To save the parameters to a file as a Fiala *.tr model, the pop-up menu commands are used that appear on a right-click. In addition to creating a file, a model can be added to the model list for assignment to wheels of the current vehicle.

1.5.11.1. Approximate vertical stiffness and damping

According to [11], the vertical tire spring constant with less than 20% error tolerance can be computed as

$$K_{z0} = \pi p_0 \sqrt{2Rw}.$$

The damping constant is evaluated according to the damping ratio parameter

$$d_{z0} = 2\beta_z \sqrt{K_{z0} m_w}$$

Here m_w is mass of wheel.

1.5.11.2. Approximate cornering stiffness

According to J.Wong [2], the cornering coefficient $\lambda_y \in [0.8{,}0.18]$ is equal to the ratio of the lateral force to the tire load at 1 degree of sleep angle, so the cornering stiffness is

$$C_{y0} = \lambda_y W_0 \frac{180}{\pi}.$$

A table of cornering coefficient values for different tires is presented in [2]. A typical value for bias tires is 0.12, for radial tires 0.16.

1.5.11.3. Approximate longitudinal stiffness

Expression for evaluation of the longitudinal tire stiffness C_{s0} is proposed in the report [12]

$$C_{s0} = \frac{2}{L_t^2} \boldsymbol{\kappa} W_0$$

where $\boldsymbol{\kappa} \approx 18$, L_t is length of tire contact patch,

$$L_t \approx 2R \sqrt{\frac{2W_0}{K_{z0} R}}$$

1.5.11.4. Longitudinal and lateral static stiffness

These parameters are necessary for modeling a standing car. In [2], the coefficient of longitudinal static stiffness is calculated through the value of the cornering stiffness

$$K_y = \frac{C_{y0}}{0.8r_w + L_t/2}.$$

Typically, the longitudinal static stiffness of a tire is several times greater than the lateral stiffness, so that

$$K_x = k_{xy}K_y.$$

The recommended value is $k_{xy} = 2.5$.

1.6. Resistance to vehicle motion

1.6.1. Aerodynamic forces

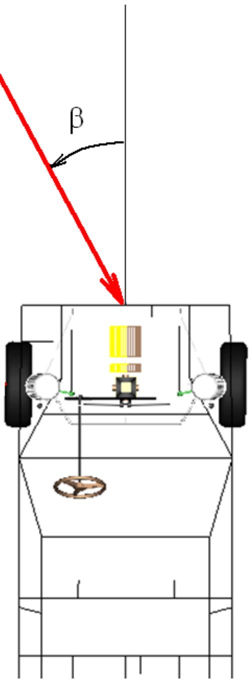


Figure 1.69. Positive angle of relative wind speed direction

Aerodynamic forces depend on the air velocity relative to the vehicle V_a, on the air density ρ, on aerodynamic coefficients, on the car area, and the angle of wind relative to the car β (Figure 1.69) and some other parameters. Aerodynamic coefficients and car area depend on the force or moment component.

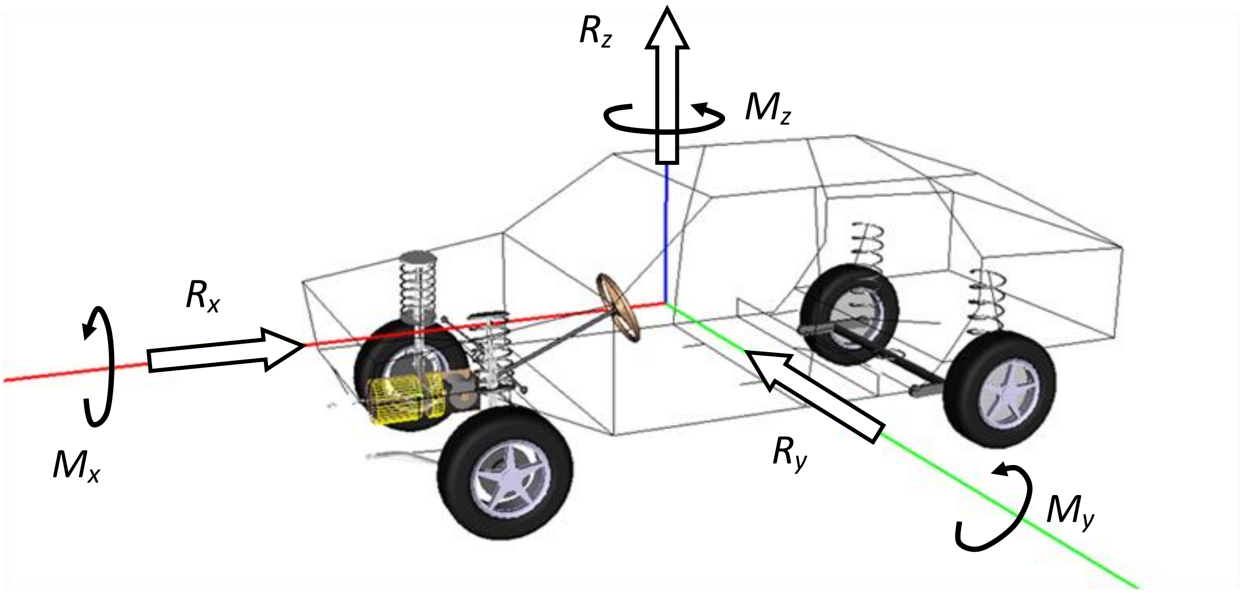


Figure 1.70. Positive directions of aerodynamic forces and moments

Consider formulas, which are used in UM for computation of force and moment components relative to the coordinate system connected with the car body. Positive directions of the components for β >0 are shown in Figure 1.70.

- Drag force

$$R_x = C_x(\beta)A_x\frac{\rho}{2}V_a^2,$$

C_x (C_d) is the drag coefficient, A_x is the frontal area, i.e. the area of the vehicle projection on a plane, which is perpendicular to the vehicle longitudinal axis.

- Side force

$$R_y = C_y(\beta)A_y\frac{\rho}{2}V_a^2,$$

C_y is the coefficient of side force, A_y is the side area, i.e. the area of the vehicle projection on a plane, which is perpendicular to the vehicle lateral axis.

- Lift force

$$R_z = C_z A_x\frac{\rho}{2}V_a^2$$

C_z is the aerodynamic lift coefficient.

- Rolling moment

$$M_x = C_{ax}(\beta)L_y A_y\frac{\rho}{2}V_a^2$$

L_y is the track width.

- Pitching moment

$$M_y = C_{ay}(\beta)L_x A_x\frac{\rho}{2}V_a^2$$

L_y is the wheel base.

- Yawing moment

$$M_z = C_{az}(\beta)L_x A_y\frac{\rho}{2}V_a^2$$

The following simplified dependencies of the aerodynamic coefficient on the angle β are used:

$$C_x(\beta) = C_x(0)cos\beta, C_y(\beta) = C_y(0)sin\beta, C_{ax}(\beta) = C_{ax}(0)sin\beta,$$
$$C_{ay}(\beta) = C_{ay}(0)sin\beta, C_{az}(\beta) = C_{az}(0)sin\beta$$

Some typical values of coefficients are written in Table 1.6.

Table 1.6

Typical values of aerodynamic coefficients

Coefficient	Passenger car	Van	Truck
$C_x(0)$	0.3 ÷0.4	0.5 ÷0.6	0.6 ÷1.2
$C_y(0)$	1.8 ÷2.8	3.0 ÷4.0	4.0 ÷7.0

$C_{az}(0)$	0.3 ÷0.8	0.04 ÷1.1	0.1 ÷1.0
$C_{ax}(0)$	0.8 ÷1.2	2.0 ÷3.6	0.9 ÷1.1

According to Wong [2], the coefficient of the lift force C_z is 0.2÷0.5, and the coefficient of the pitching moment $C_{ay}(0)$ is 0.05÷0.21.

The drag coefficient C_x and the frontal area A_x for many cars can be found in internet, see http://rc.opelgt.org/indexcw.php.

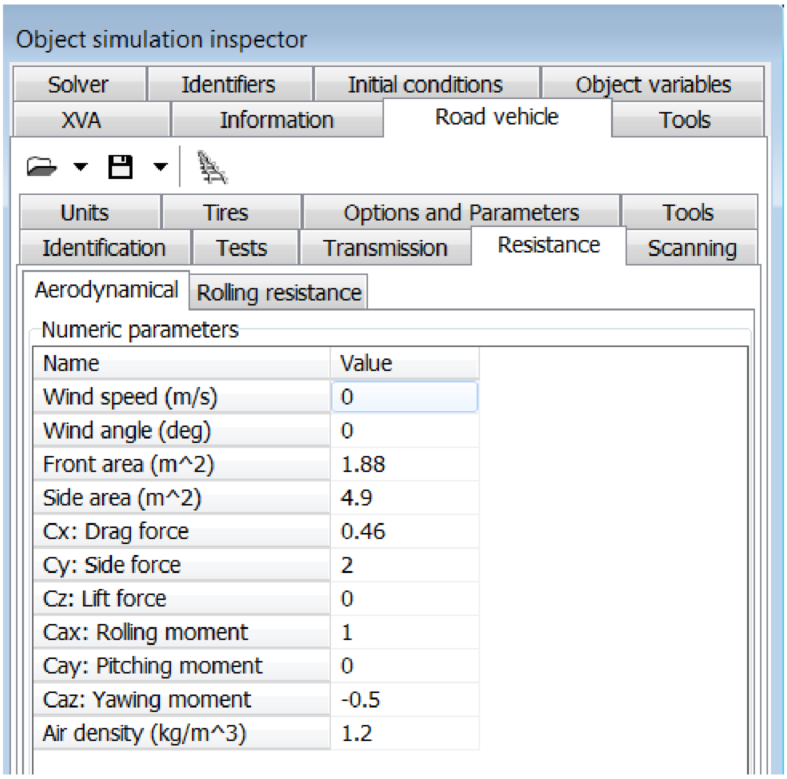


Figure 1.71. Parameters of aerodynamic forces

In **UM Simulation** program, the parameters of aerodynamic forces are set on the **Road vehicle | Resistance | Aerodynamical** tab of the simulation inspector, Figure 1.71. The wind speed is specified relative to Base0. The wind angle is computed relative to the X axis, its positive direction is determined similar to Figure 1.69.

1.6.2. Tire rolling resistance

The rolling resistance is considered as a torque $T_{rf} = F_{rf}R$ applied to the wheel and directed opposite to the wheel roll, R is the rolling radius of the tire. According to Wong [2], the resistance force is

$$F_{rf} = fN$$

where f is the coefficient of friction, and N is the tire normal force. The coefficient of friction depends on the vehicle speed as [7]

$$f = f_0 + k_1 v + k_2 v^2$$

Here v is the speed in km/h, and f_0, k_1, k_2 are empirical constants, which values are set by the SetRollingFriction method. Typical values of the coefficients can be found in [2], see Table 1.7

Table 1.7

Parameters of rolling friction

Tire	f_0	k_1	k_2
radial-ply passenger car tire	0.0136	0	0.4e-7
bias-ply passenger car tire	0.0169	0	0.19e-6
radial-ply truck tire	0.006	0	0.23e-6
bias-ply truck tire	0.007	0	0.45e-6

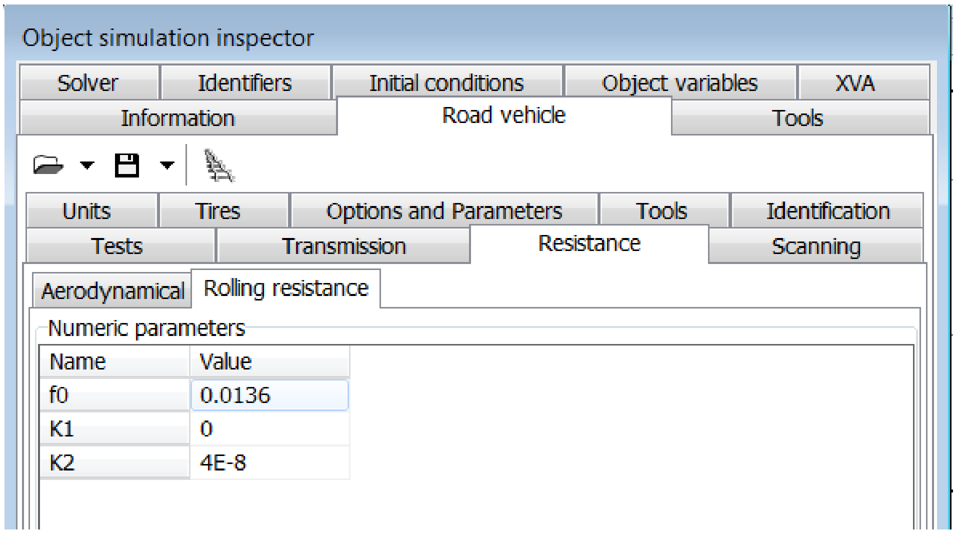


Figure 1.72. Parameters of rolling resistance

The rolling resistance parameters are set in the **Road vehicle | Resistance | Rolling resistance** tab of the simulation inspector, Figure 1.72.

1.7. Development of vehicle model

In this section we consider approaches to modeling of main elements of vehicle: wheels, springs, shock absorbers, leaf springs etc.

1.7.1. Model of a wheel

Starting with version UM 7, the program provides two wheel models, with rigid and elastic mounting of the tire to the disk center. Below we will call the first option "rigid wheel", and the second - "elastic wheel"

1.7.1.1. Rigid wheel

A rigid wheel in the UM model of a vehicle is a usual body with assigned image and inertia parameters, Figure 1.73. The following special features distinguish the wheel from other bodies in the model.

- Center of mass is located at the origin of the wheel-fixed system of coordinates (SC).
- Wheel rotation axis coincides with the Y-axis of the wheel-fixed SC.
- A special force element of **Tire** type should be created for each of the wheel. Body, which corresponds to the tire, must be assigned as the *second body* in description of the force element. As a rule, the *first body* in the force element is Base0.
- The wheel should be connected to the vehicle by a rotational joint; **increment of joint coordinates must correspond to the motion of the vehicle ahead**.

You can use the visual component ‘Wheel’ to add wheel to vehicle models.

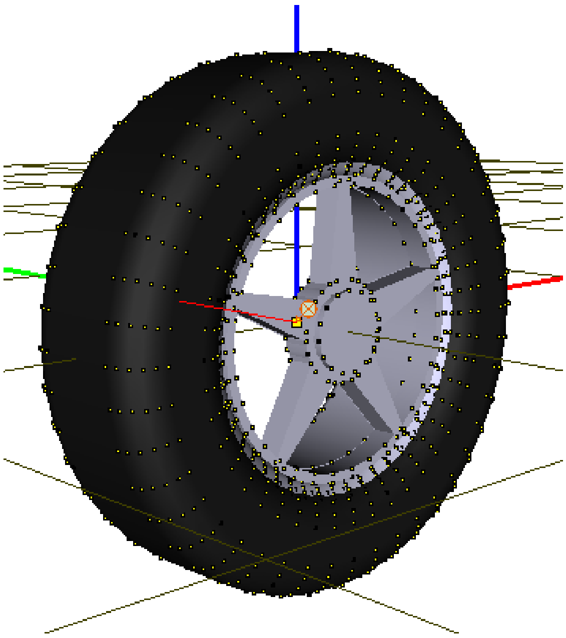


Figure 1.73. Model of a wheel as a rigid body

1.7.1.2. Elastic wheel

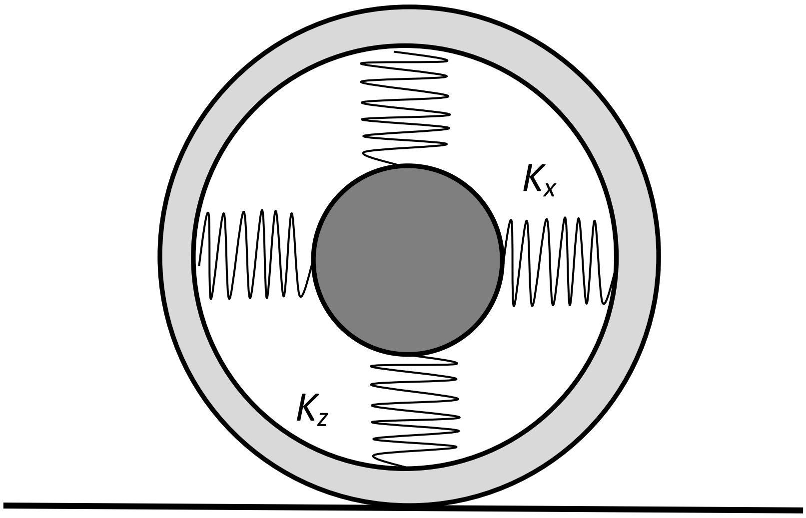


Figure 1.74. Elastic wheel

An elastic wheel is formed by two rigid bodies corresponding to a tire and a rim connected to each other by an elastic element, Figure 1.74. The tire must have at least three degrees of freedom relative to the rim: shifts in the vertical and longitudinal directions and rotation relative to the wheel axis. It is also possible to introduce a tire shift relative to the center along the wheel axis.

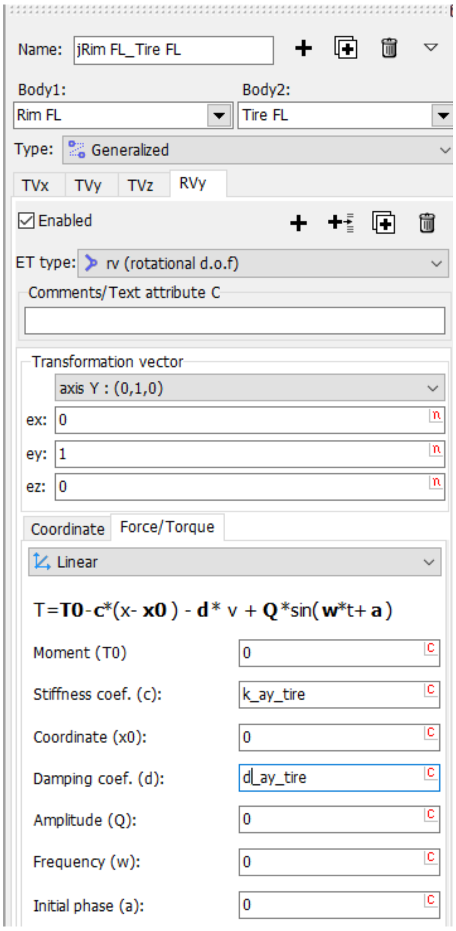


Figure 1.75. Joint for elastic wheel

To introduce the degrees of freedom of the tire, it is recommended to use a generalized type hinge, fig. 12.79. The elastic stiffness constants K_x, K_y, K_z are specified in the tire model, Sect. 1.5.2 “*FIALA tire model*”, and the corresponding forces are automatically calculated in the simulation program. An angular elastic connection can be described as a joint force for the corresponding degree of freedom, Figure 1.75.

A special power force of the "Tire" type should be assigned to the corresponding body. The tire image is assigned to the rim body.

Before simulation of the car, you must confirm the use of the elastic wheel model on the **Road Vehicle | Tire** tab of the simulation inspector, Figure 1.76.

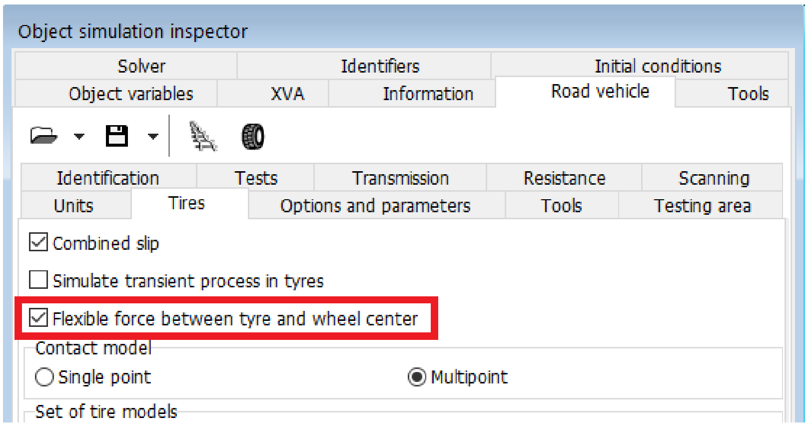


Figure 1.76. Confirmation of the tire elastic model

The vertical stiffness in contact between the tire and the road is assumed to be ten times greater than the verical tire stiffness K_z.

1.7.2. Visual wheel components

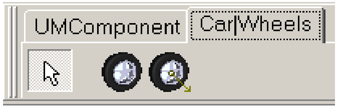


Figure 1.77. Wheel components

The **CarComponents.umc** library contains two visual components of wheels, Figure 1.77. Both of them add to the model a *right* wheel with a fully parameterized image (Figure 1.73), inertia parameters, a special tire force element, and a joint. The difference consists in the joint model. The first component (*'Right wheel'*, left in Figure 1.73) adds a hidden joint with 6 degrees of freedom, and user must create an additional rotational joint to connect the wheel and the vehicle. In contrary, the second component (*'Right wheel + Joint'*, right in Figure 1.73) allows the user to create the rotational joint simultaneously.

Let us consider the process of visual adding the components.

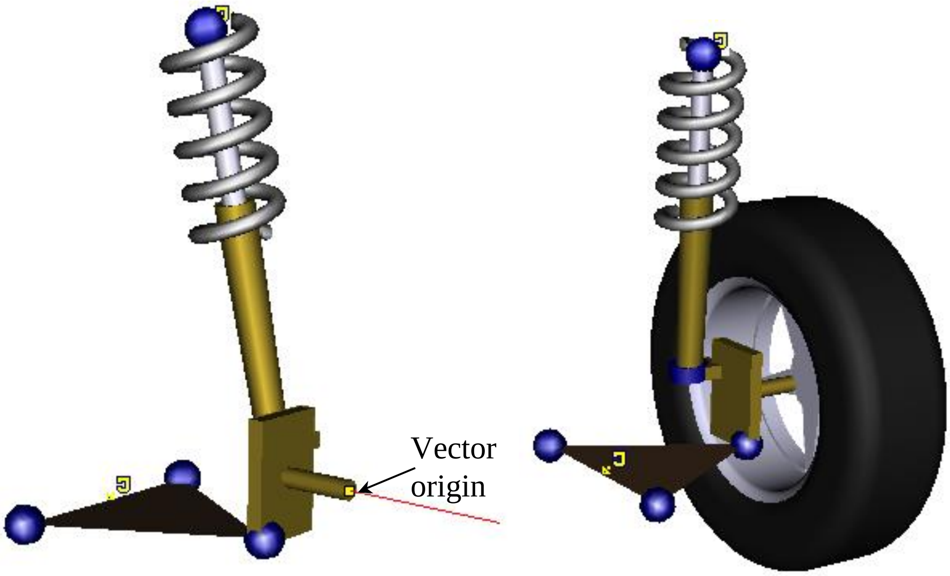


Figure 1.78. Visual adding of wheel and joint

Right wheel. Click the component button and then click on the desired grid point to add the wheel at the selected grid position. Change wheel image and inertia identifiers, if necessary, assign a separate sheet for these identifiers.

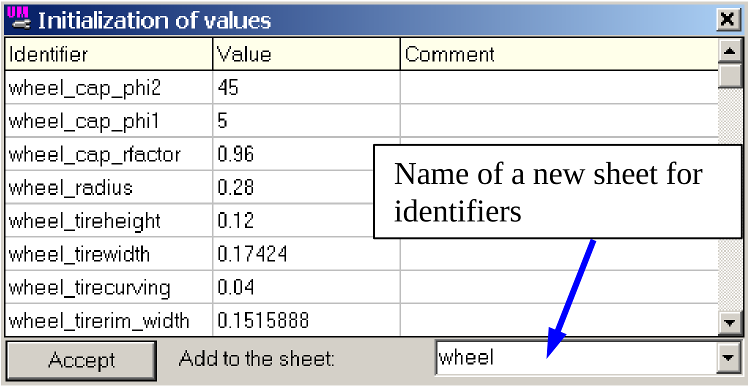


Figure 1.79. Standard wheel image and inertia identifiers

Right wheel + Joint

1. A vector corresponding to the joint point and joint vector must be created for the body, which the wheel is connected to (e.g. the strut). Moreover, this body must be in the object tree, i.e. it must be visible in the full object mode of the animation window (Figure 1.78).
2. Click the component button and then click on the origin of the vector.
3. Change identifiers corresponding to wheel inertia parameters and image.

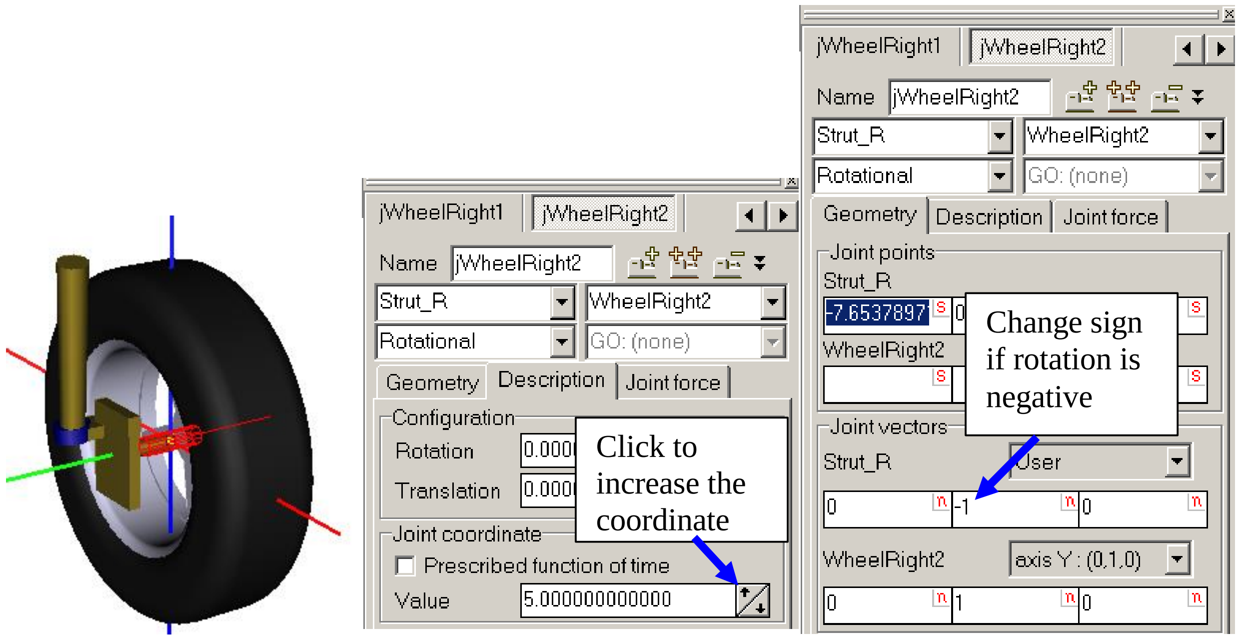


Figure 1.80. Verification of wheel rotational joint

Remark. If increase of the joint coordinate corresponds to the negative rotation of the wheel, one of the joint vectors should be changed to the opposite one directly in the description of the joint after it creation, Figure 1.80.

1.7.3. Suspension springs and shock absorbers

Linear suspension springs can be modeled by the *generalized linear force elements* (Chapter 2) if a stiffness matrix describes their stiffness properties.

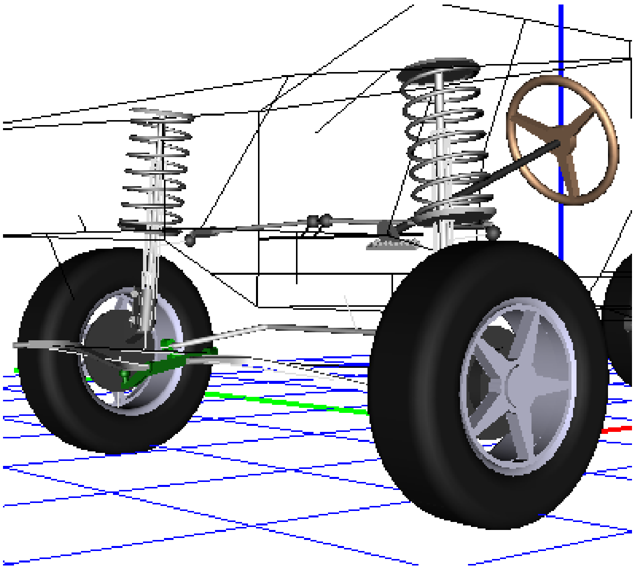


Figure 1.81. Suspension force elements

Both linear and nonlinear bipolar springs and shock absorbers can be modeled by *bipolar force elements* (Chapter 2).

Sometimes two bodies connected with a translational joint present the shock absorber. For example, in the case of the MacPherson strut these bodies are the tube and the rod. The joint force describes properties of the shock absorber as a force element.

1.7.4. Leaf springs

Figure 1.82. Leaf springs

A massless leaf spring model is the combination of a generalized linear force element and a one (central) or two (at the spring ends) bipolar elements *'Fancher leaf spring'*. The stiffness matrix of the linear force element has at least five non-zero diagonal elements, see Figure 1.83, representing the lateral, longitudinal, pitch, roll and yaw stiffness properties of the spring. The *Fancher model* is proved to be efficient in modeling the vertical hysteresis characteristic of the leaf spring.

$$\begin{pmatrix} C_x & 0 & 0 & 0 & 0 & 0 \\ 0 & C_y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{ax} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{ay} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{az} \end{pmatrix}$$

Figure 1.83. Stiffness matrix for generalized linear force element

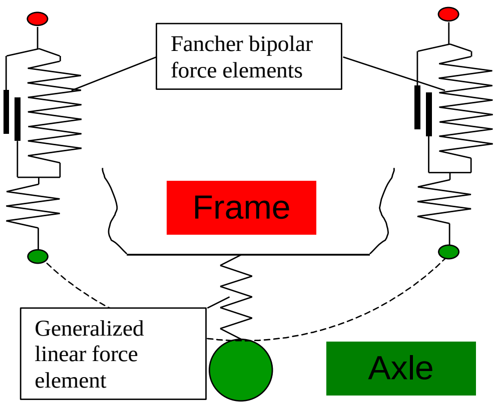


Figure 1.84. Model of a massless leaf spring

Remark. The user should remember that bipolar force elements degenerate by zero length. It is recommended that the lengths of the *Fancher elements* in the model of the leaf spring must be at least two times greater than the maximal dynamic shortening the element.

1.7.5. Air springs

The air springs are modelled with the help special force of **Airspring** type (Chapter 2, Sect. *Special forces/ Air springs*) or **Pneumatic subsystem** (Chapter 31).

1.7.6. Bushings

UM supports both linear and nonlinear bushings. The mathematical model of bushings is described in Chapter 2, Sect. *Special forces/Bushings*. Input of the element parameters see in Chapter 3, Sect. *Data Input / Input of force elements / Special forces / Bushings*.

Use the joints of generalized type to describe both nonlinear bushings and bushings with friction. The joint should include all six d.o.f., the stiffness and damping for locked degrees of freedom can be described as joint forces. The mathematical model of joints is described in Chapter 2, Sect. *Joints/Generalized joint*. Input of the joint parameters see in Chapter 3, Sect. *Data Input / Input of joints/ Input of joint of generalized type*.

1.7.7. Steer control

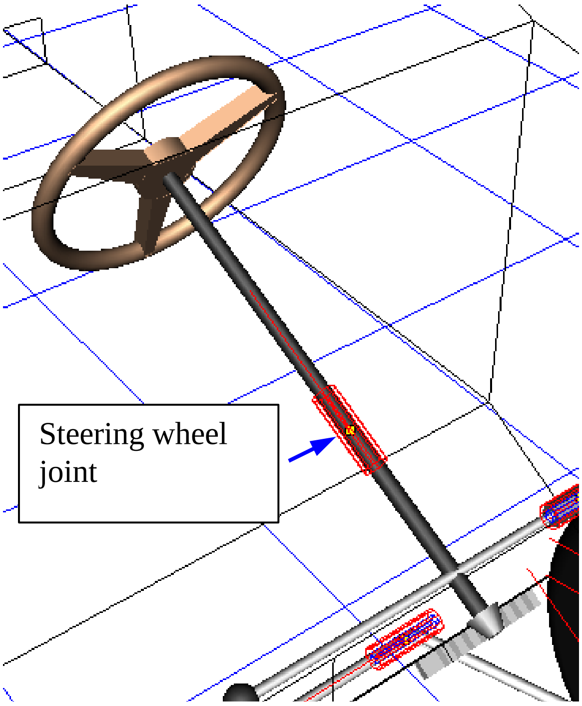


Figure 1.85. Steering wheel joint

To make possible an open and closed loop steer control, the model of a vehicle needs a special joint torque. The torque is introduced in the steering wheel joint, which is a rotational joint connecting the steering column with the car body, Figure 1.85.

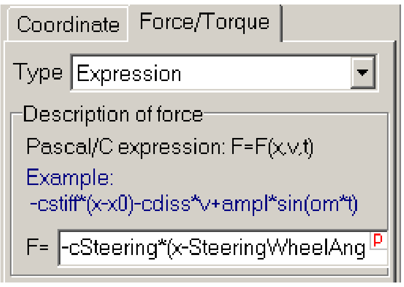


Figure 1.86. Steering control torque

The model of the steering control torque is described as a joint torque of the *Expression* type by the following equation, Figure 1.86:

-cSteering(x-SteeringWheelAngle)-dSteering*(v-dSteeringWheelAngle)*

Here *cSteering* and *dSteering* are the stiffness and damping constants of the steering control, *SteeringWheelAngle* and *dSteeringWheelAngle* are the desired values of the steer wheel angle and its rate obtained from the control strategy during the simulation process. The user may introduce they own identifiers for these four parameters.

Note. Identifiers for the stiffness, steer wheel angle and its rate cannot be expressions, i.e. they cannot be expressed through other identifiers.

1.7.8. Force element for simplified control of the speed of the longitudinal movement

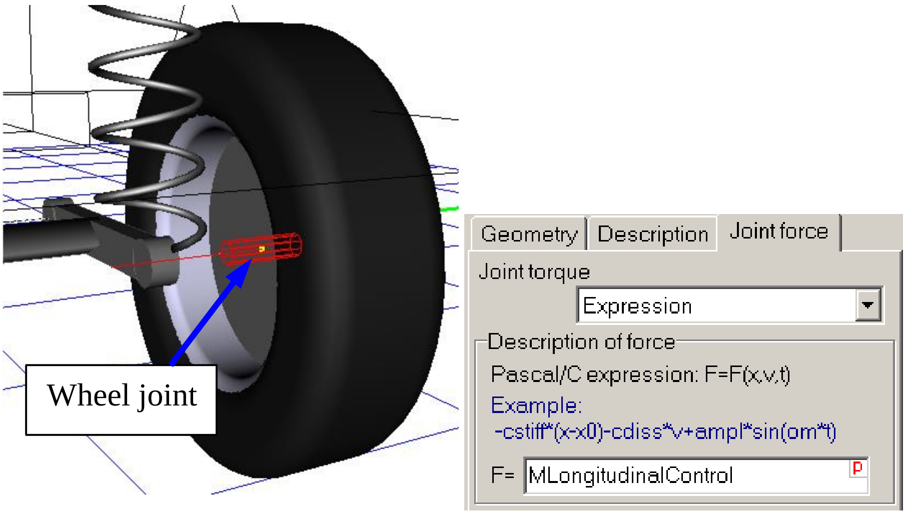


Figure 1.87. Joint torque for longitudinal velocity control

In a simplified car model, the description of the transmission is usually absents. In this case, in order to be able to control the speed of the longitudinal movement of the car, it is necessary to add a special traction torque to the model. In the simplest case the torque is introduced in the driving-wheel joint, which is a rotational joint connecting the driving wheel with the vehicle, Figure 1.87. The model of the control torque is described as a joint torque of the *Expression* type by the same identifier for all of the driving wheels, Figure 1.87:

MLongitudinalControl

This is the default name. The user may introduce another name of the identifier.

The identifier must be assigned to the simplified longitudinal control identification in simulation program, see Sect. 1.9.1.2.1 "*Identification of parameters for simplified longitudinal speed control*".

1.7.9. Locking wheel rotation

Some simulation results are obtained for a motionless vehicle, for example, vehicle equilibrium test. For this purpose we recommend to introduce a locking joint torques for some wheels. Often the rear wheels are chosen for locking. The following linear elastic-dissipative model of the torques could be used

*MlongitudinalControl-cLocking*x-dLocking*v,*

with *cLocking* and *dLocking* the a stiffness and damping constant. In this example the torque locking the wheel rotation is parallel to the traction joint torque from the previous section, Figure 1.88.

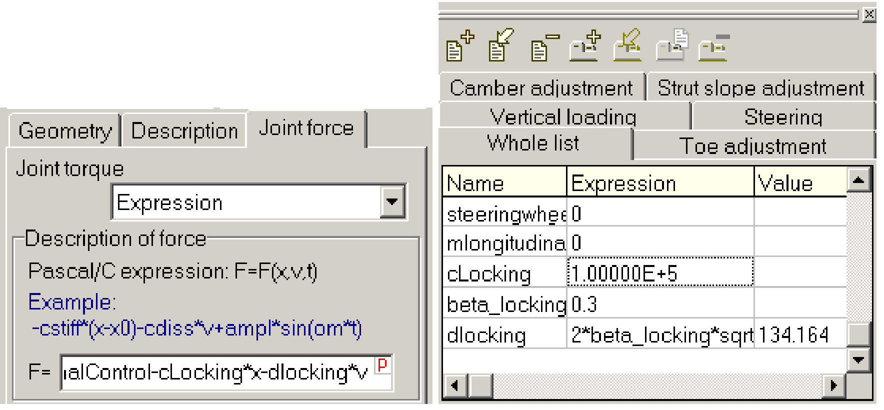


Figure 1.88. Locking joint torque (left); damping constant as an identifier-expression

To get a reasonable damping constant for a definite stiffness value, a new identifier for the damping ratio of critical should be introduced. Let it be *beta_locking*. Then the *dLocking* should be computed according to the expression (Figure 1.88)

*dLocking=2*beta_locking*sqrt(clocking*IWheelY),*

where *IwheelY* is the moment of inertia of the wheel relative to the wheel axis.

The recommended values for the independent identifiers are

$$cLocking = 1.0 \times 10^5 \ Nm/rad, \ beta\_locking = 0.3$$

1.8. Transmission

The **UM Driveline** module is required. Use the **Help | About** menu command to verify whether this module is available in the current un configuration, Figure 1.89. Usually a car transmission is modeled by a set of rigid bodies with one rotational d.o.f. connected by special force elements. A detailed description of these force elements can be found in the user's manual, Chapter 22.

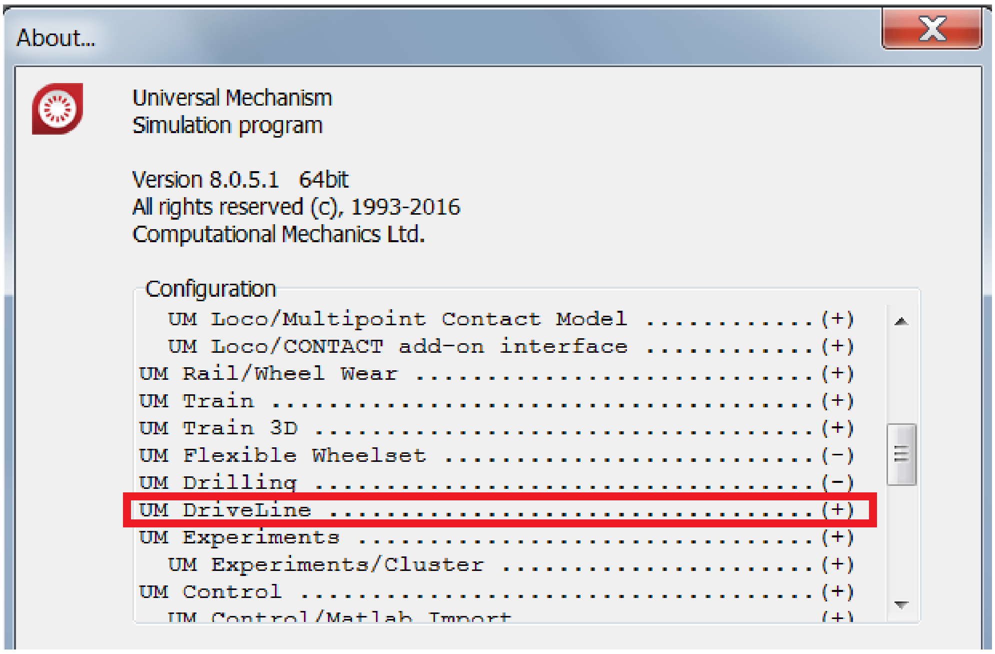


Figure 1.89. UM Driveline module in the current configuration

As a rule, a transmission model in UM includes the following elements:

- internal combustion engine (ICE);
- clutch (mechanical gearbox);
- torque converter (automatic gearbox);
- gearbox;
- differential;
- braking system;
- ABS.

Most of these items must be described in the Input module as force elements by developing the car model (cutch, torque converter, gearbox, differential and so on). Parameters of ICE, ABS and braking system are specified in the Simulation module on tabs of the simulation inspector, Figure 1.90.

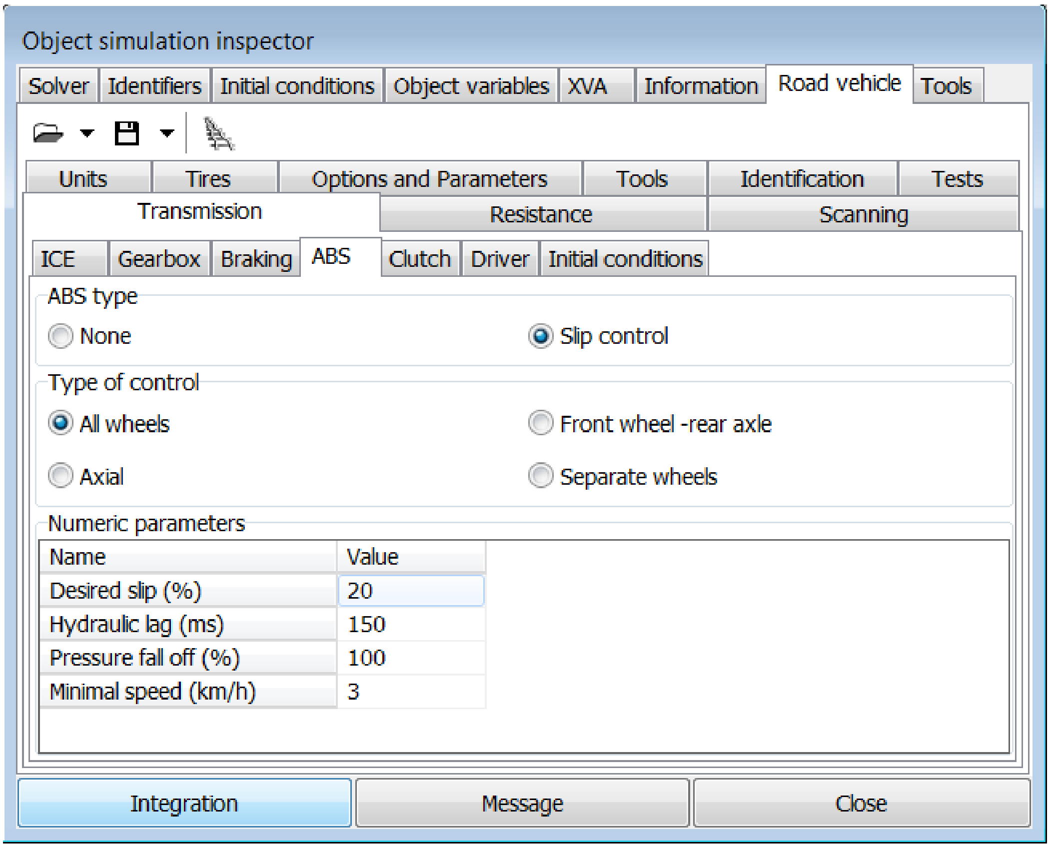


Figure 1.90. Tabs related to car transmission

1.8.1. Description of transmission elements in Input module

Here we consider how elements of transmission are modeled in Input module.

1.8.1.1. Internal combustion engine

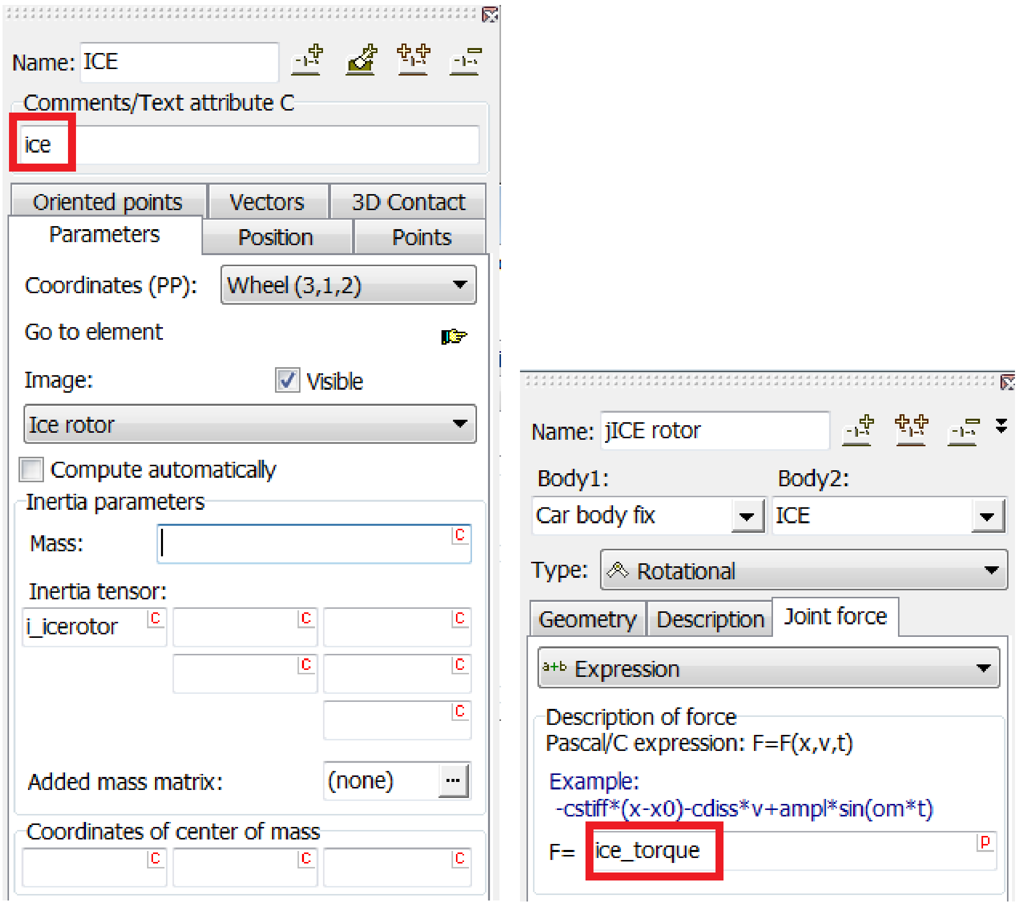


Figure 1.91. Example of an ICE shaft as a rigid body and a corresponding rotational joint

The following elements are necessary, Figure 1.91.

- A body modeling the crankshaft. Moment of inertia relative to the rotational axis must take into account all moving parts of the ICE. The body is marked by the text attribute C "ice", Figure 1.91, left.
- A rotational joint assigned to the shaft describes a joint force of the *Expression* type, which parameterizes the engine torque acting on the shaft. It is recommended to use the standard identifier **ice_torque** for parameterization of the torque, Figure 1.91, right.

1.8.1.2. Friction clutch

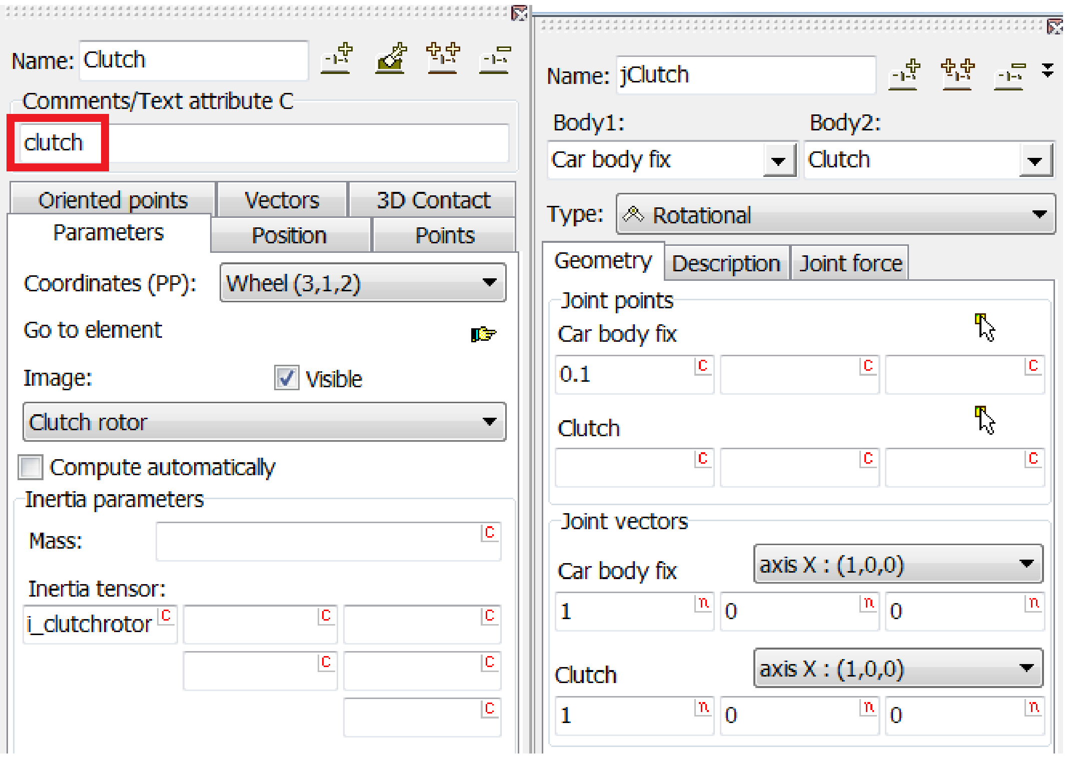


Figure 1.92. Example of a clutch plate as a rigid body and a corresponding rotational joint

A simplified model of a friction clutch includes one body (the second clutch plate) with assigned rotational joint, and one frictional force element between the crankshaft and the second clutch plate. The following elements should be created, Figure 1.92, .

- The body corresponding to the clutch plate must be marked by the text attribute C "clutch" Figure 1.92.
- A rotational joint assigned to the plate introduces the plate rotation relative to the car body. It is not recommended to define the plate rotation relative to the crankshaft.

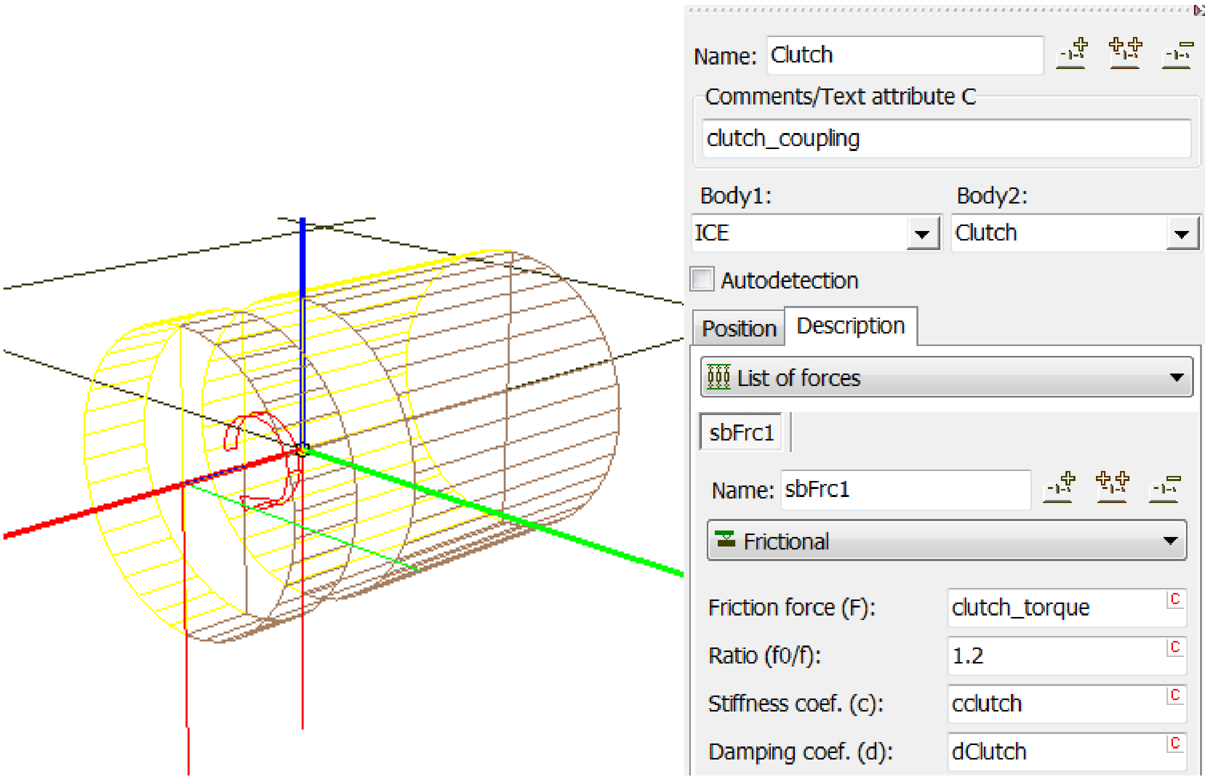


Figure 1.93. Scalar torque modeling friction between crankshaft and second clutch plate

- A scalar torque models the friction between the crankshaft and the clutch plate. Please take care of the local coordinate systems of the force elements: Z axes of the local systems must be oriented along the rotation axis of the interacting bodies, see Chapter 2 of the user's manual, Sect. *Scalar torque*). Select the **Frictional** type of the torque and set the friction torque value by an identifier; it is recommended to use the standard identifier *clutch_torque*, Figure 1.93.

1.8.1.3. Gearbox. Final drive

It is recommended to use the 'Mechanical rotation converter' force element for simplified modeling both the gearbox and the final drive of the transmission, see Chapter 22 of the user's manual, Sect. *Mechanical rotation converter*. In case of a mechanical transmission, the gearbox and the final drive can be described by one force element. In this case the force elements transfer the rotation directly from the second clutch plate to the differential housing, Figure 1.94. In the case of a separate modeling the gearbox and final drive by two force elements, an intermediate body for output shaft of the gearbox as well as the corresponding rotational joint must be added.

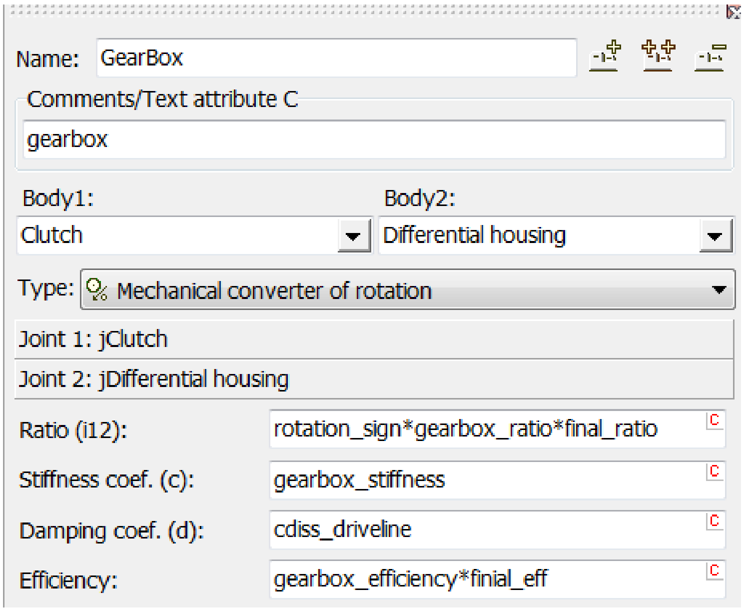


Figure 1.94. Example of modeling gearbox and final drive by single force element

Consider an example of modeling the gearbox and final drive by a single force element, Figure 1.94. A special force **Mechanical converter of rotation** is used. The element parameters are as follows:

- Ratio (i12):

*rotation_sign*gearbox_ratio*final_ratio*

Here three identifiers are introduced:

rotation_sign is an auxiliary identifier, which value is +1 or -1 to get the rotation of wheels in the correct direction;

gearbox_ratio is the identifier for the obligatory parameterization of the gear ratio; here we used the recommended name of the identifier;

final_ratio is the ratio of the final drive; the user can use the numeric value instead the identifier if the ratio is not planned to be varied;

- Stiffness and damping constants of the converter;
- Efficiency factor specifies the energy losses in transmission; in this example the efficiency is the product of the corresponding values for the gearbox *gearbox_efficiency* (the recommended identifier name) and the final drive.

1.9. Simulation of vehicle dynamics

This section is entirely concerned with the features of modeling the dynamics of road vehicles in the UM Simulation program. General information about working in the UM Simulation software is described in Chapter 4 of the User's Guide.

1.9.1. Preparing for simulation

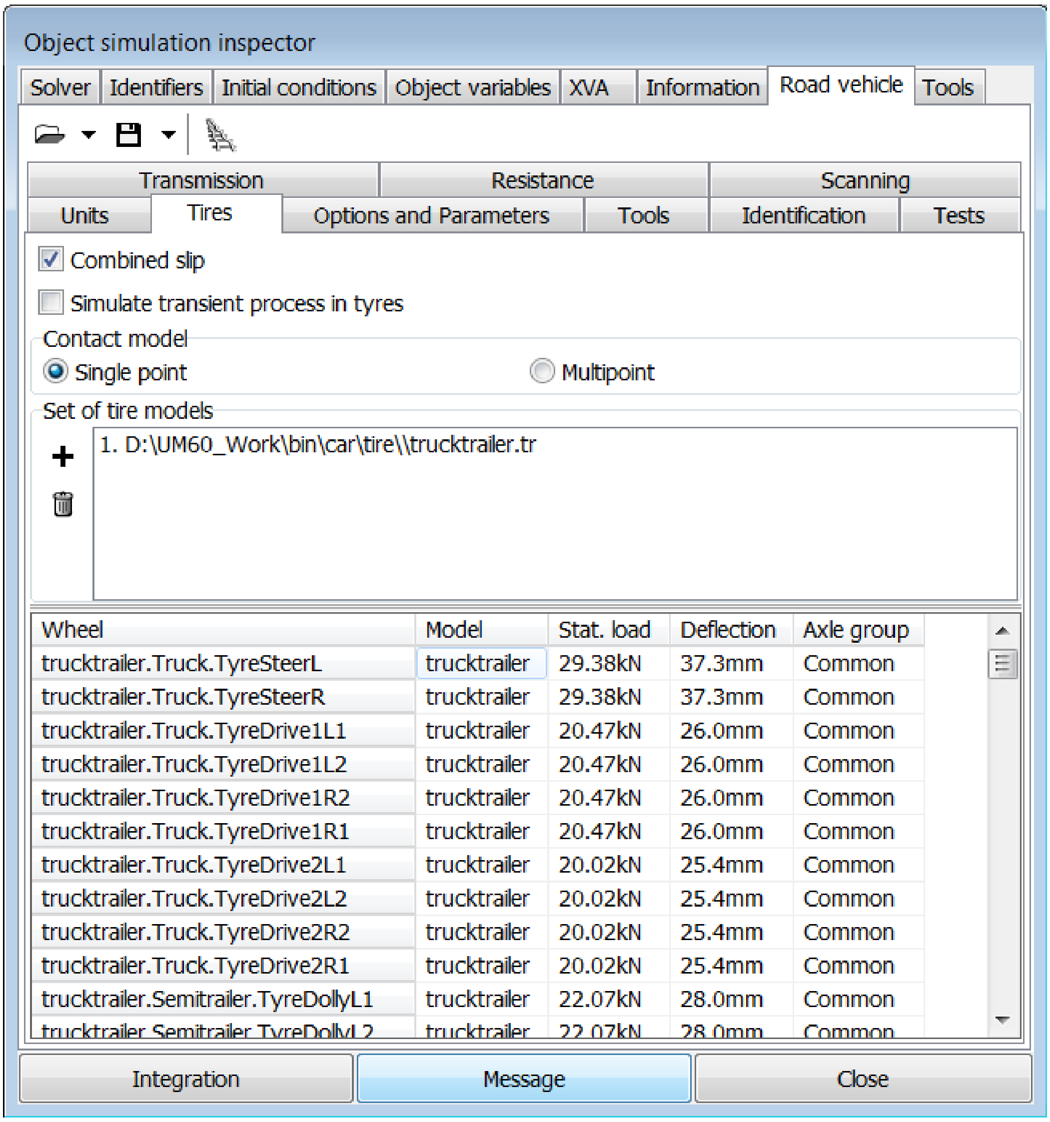


Figure 1.95. Object simulation inspector

The most part of the road vehicle specific data is located on the **Road vehicle** tab in the **Object simulation inspector**, Figure 1.95. Use the **Analysis | Simulation…** menu command of the **UM Simulation** program to open the inspector. The road vehicle specific data is saved in the vehicle configuration files **.car*. Use the buttons on the tab to read/write data.

The vehicle configuration data is saved automatically in the *last.car* file if the **Road vehicle configuration** option is checked on in the options window, Figure 1.96. Use the **Tools | Options…** menu command to call this window.

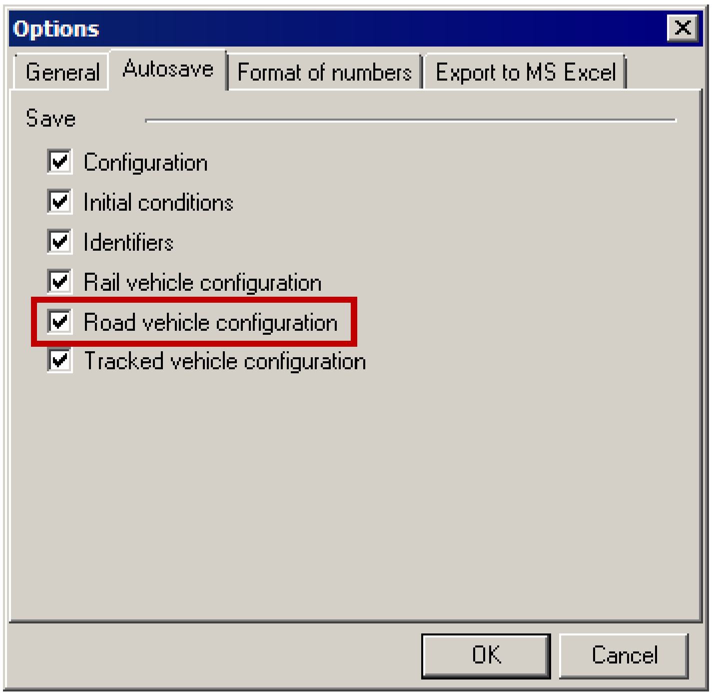


Figure 1.96. Options of **UM Simulation** program

1.9.1.1. Units

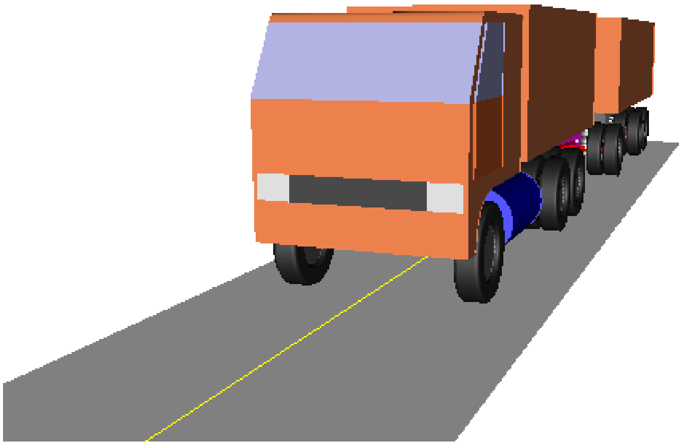


Figure 1.97. 2-unit vehicle: a truck with a trailer

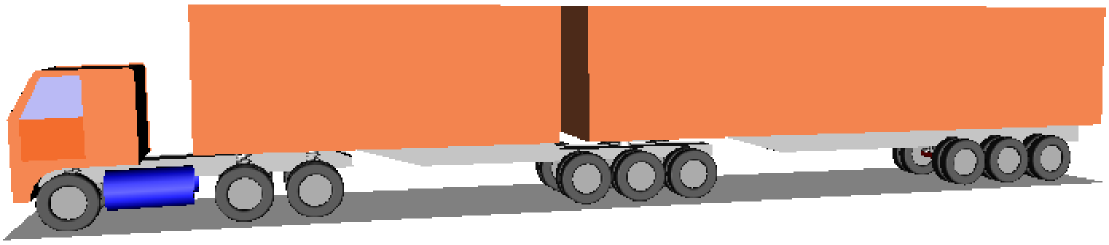


Figure 1.98. 3-unit vehicle: a truck with two semi- trailers

Since version 5.0 the UM software allows the user to create vehicles containing any number of units. Unit one should be a car or a truck with a steering system. Other units can be trailers or semi-trailers (Figure 1.97, Figure 1.98). Distribution of bodies on units should be identified.

Simulation of vehicle dynamics requires identifying the car bodies for each of the units even if the model contains only one unit.

Use the **Road Vehicle | Units** tab of the **Object Simulation Inspector** in the Simulation module to make the necessary identification, Figure 1.99.

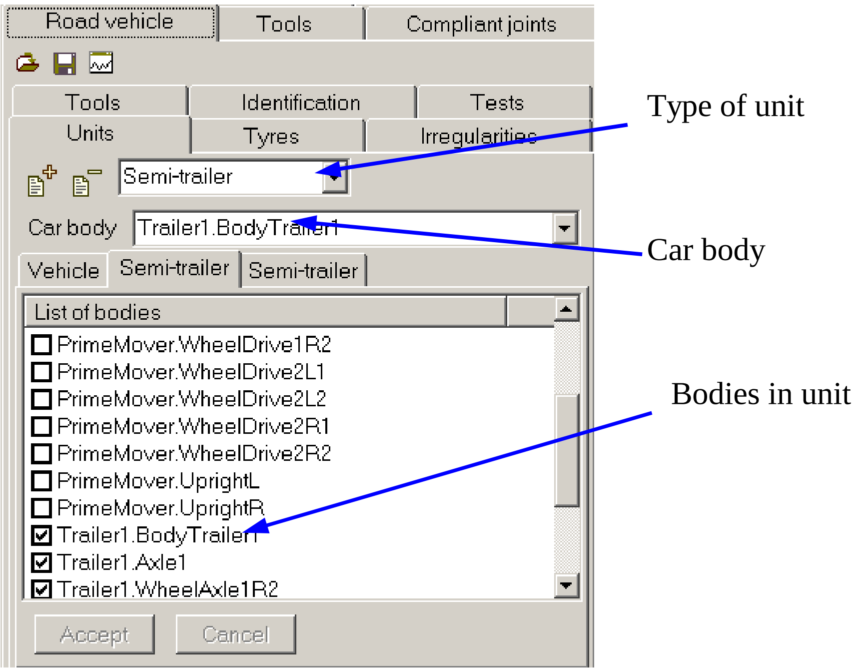


Figure 1.99. . Identifying units and car bodies

- Use the buttons to add/remove a unit (except of Unit 1).
- Select type of the unit (trailer or semi-trailer).
- Check in the list all bodies included in the unit.
- Click the **Accept** key.
- Select a car body.

Note 1. The car body is selected automatically as a body included in the unit with the biggest mass. Change the assignment if necessary.

Note 2. To check the bodies for a unit the user can either use a mouse or he may select first items in the list and then click the Enter key.

1.9.1.2. Identification of vehicle subsystems

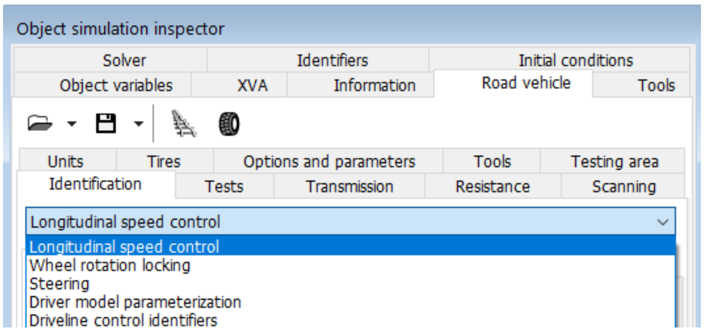


Figure 1.100. List of subsystems for identification

Before starting the simulation of the vehicle motion, it is necessary to identify its subsystems, the list of which is shown in Figure 1.100. Identification means setting the numerical values of the parameters corresponding to a given vehicle model, as well as assigning identifiers that parameterize the models of force elements.

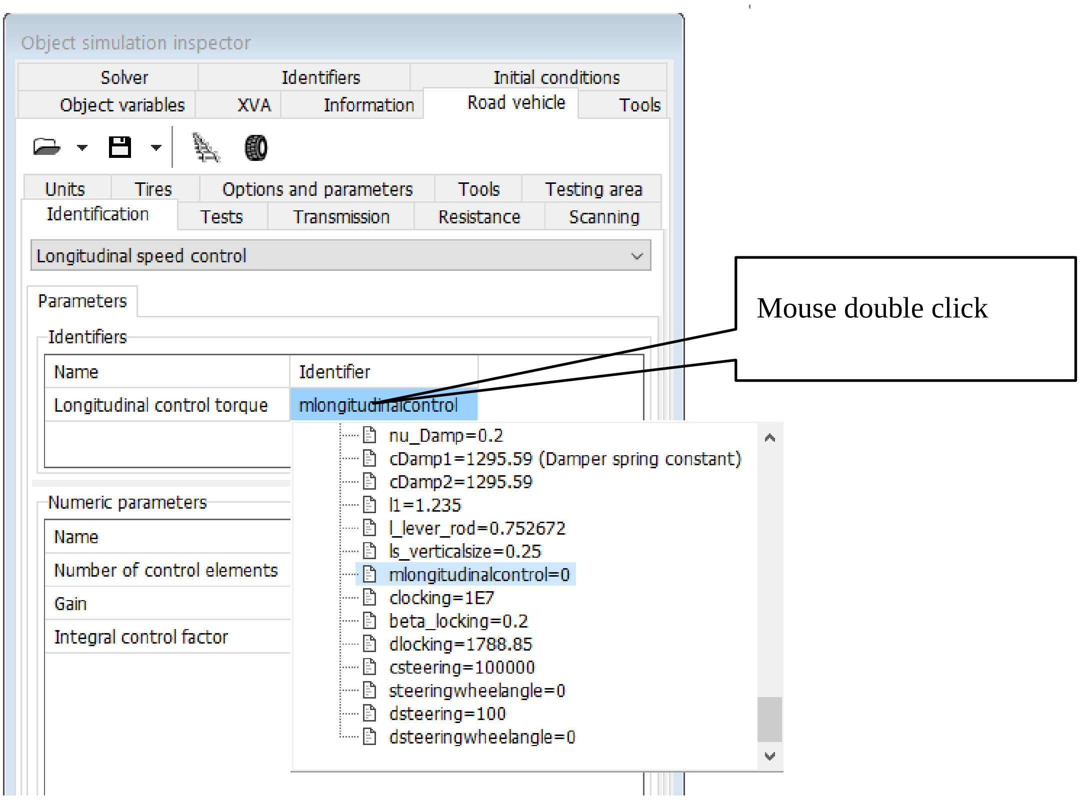


Figure 1.101. Assignment of identifiers

To assign an identifier, double-click on the table row and select an identifier from the list that appears, Figure 1.101.

[Identification of parameters for simplified longitudinal speed control](#)
[Identification of wheel rotation locking parameters](#)
[Identification of steering](#)
[Parameterization of driver model](#)
[Identification of transmission control](#)

1.9.1.2.1. Identification of parameters for simplified longitudinal speed control

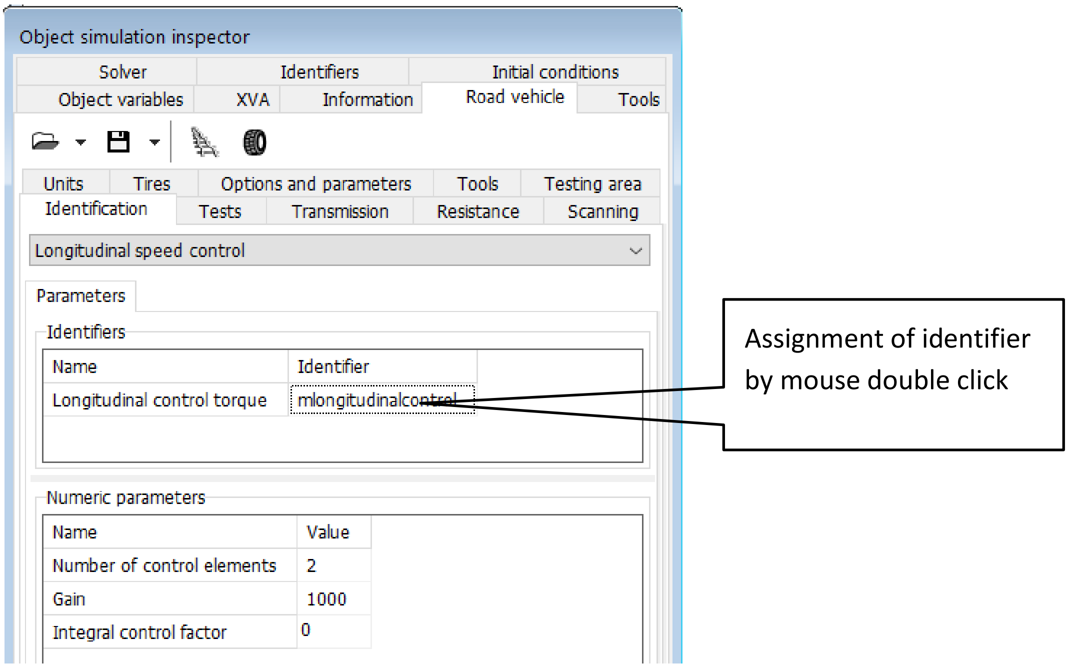


Figure 1.102. Identification of simplified speed control

These parameters are used in a simplified way to control the longitudinal speed of the vehicle, see Sect. 1.9.3.1 “*Speed modes for simplified longitudinal control*”. The identification of the control system in this case implies the choice of one parameter and three numerical constants, Sect. 1.7.8 “*Force element for simplified control of the speed of the longitudinal movement*”.

Identifier:

- Longitudinal control torque – you must specify an identifier that parameterizes the control moment. The default name is *mlongitudinacontrol*. The numerical value of the identifier is calculated by the program in the process of simulation of the vehicle dynamics.

Remark. Setting the name of the moment identifier as the default value allows the user to automate the identification of parameters.

Double click by the left mouse button on the corresponding table row to assign a model identifier. Identifier for the control torque can be selected from the head of model or from any of subsystems. If several subsystems include identifiers with the same name, their numeric values will be set by the program equal to the value of selected identifier.

Numeric parameters:

- Number of control elements *N*: number of force elements in the model, which description includes the control torque identifier (in the figure it is *mlongitudinalcontrol*). For example, this is the number of traction wheels.
- Control gain *K* - coefficient at the proportional term of the speed PI controller;
- Integral control factor K_I is then coefficient at the integral term of the speed PI controller

The value of numerical parameters are set directly from the keyboard.

1.9.1.2.2. Identification of wheel rotation locking parameters

Use the **Road vehicle | Identification** tab of the **Object simulation inspector** to identify the *wheel rotation locking* parameters. The parameters are used in tests when the wheels must be locked, for example, in equilibrium calculation or test. Select the **Wheel rotation locking** data type in the drop-down menu

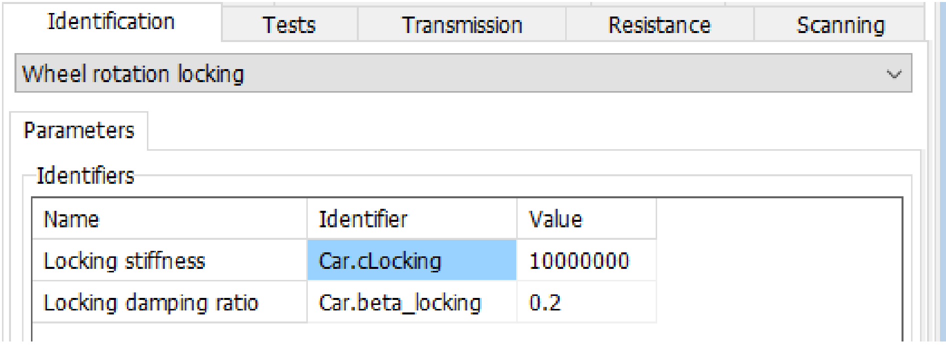


Figure 1.103. Identification of wheel rotation locking parameters

The following two identifiers should be assigned (Sect. 1.7.9. “*Locking wheel rotation*”):

- Locking stiffness constant (Nm/rad)
- Locking damping ratio.

1.9.1.2.3. Identification of steering

Use the **Road vehicle | Identification** tab of the **Object simulation inspector** to identify the *steering control* parameters.

Identification of the steering control parameters of the model requires selecting four identifiers (see Sect. 1.7.7. *"Steer control"*, p. 1-83), Figure 1.104:

- Steering wheel angle
- Steering wheel rate
- Steering stiffness
- Steering damping

Double click by the left mouse button on the corresponding table row (Figure 1.104) to assign a model identifier to the steering control parameter. Use the direct input to set the numeric parameters.

Four numeric values

- Steering ratio
- Index of the steer wheel angle and the index of corresponding subsystem
- The maximal value of the steering wheel rotation angle in degrees

Steering ratio is the ratio of the angle of rotation in the joint, in which the control is introduced, to the angle of rotation of the wheels, Sect. 1.7.7 “*Steer control*”. If a simplified car model does not contain steering wheel and the wheels turn directly in the hinge between the wheel and the hub or axle, the gear ratio is 1, see e.g. models **bdouble**, **trucktrailer**, **GAZ-66** and many other models from the directory {DATA UM}\Samples\Automotive. In the case when the car model includes the full description of the steering system, use the *Steering wheel rotation test* to automatically determine the ratio, see Sect. 1.9.4.5. The full steering system can be found e.g. in the models **vaz2109, vaz2109 T, BMW3_E36, Lada4x4**.

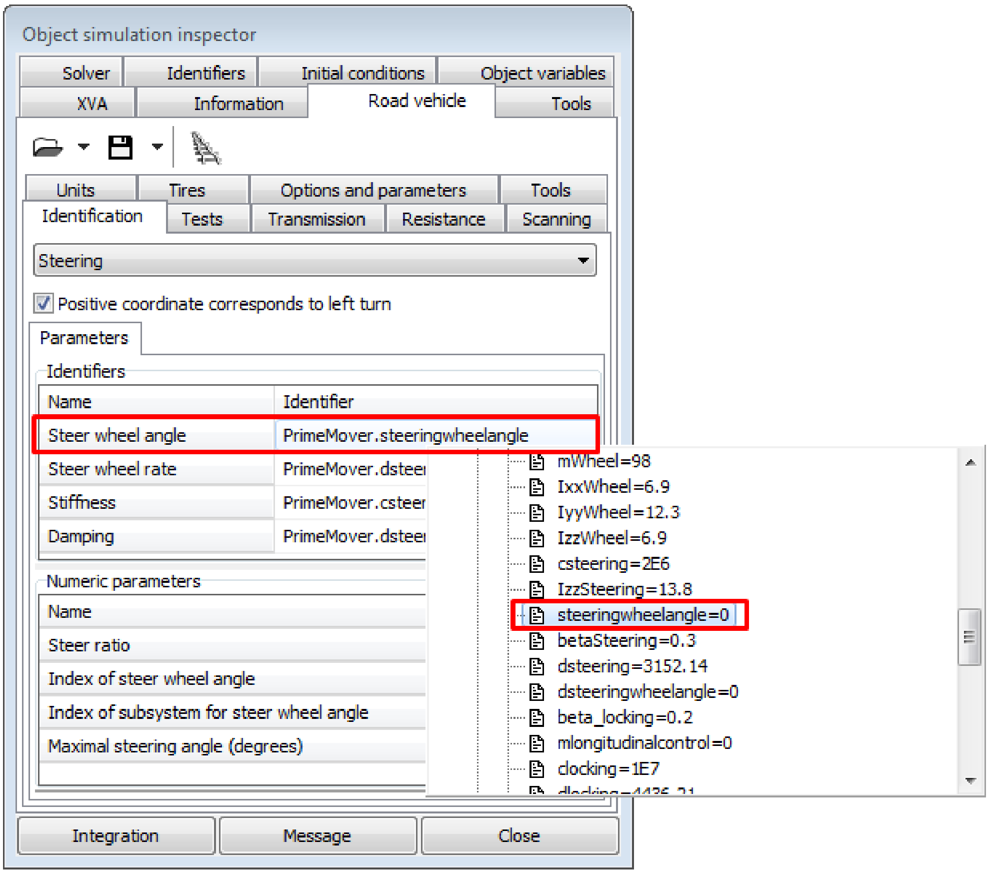


Figure 1.104. Identification of steering control

Index of steer wheel angle and *Index of subsystem for steer wheel angle* are the index of the coordinate in the joint responsible for steering control and the index of corresponding subsystem. For example, it is the coordinate in the rotational joint connecting the steering column and the car body (Figure 1.105) or in the joint between the steer wheel hub and the axle/strut when the steering system is not directly included in the model. The first number in the coordinate identification number (1.76 in Figure 1.105) corresponds to the subsystem index. If the model does not contain external subsystems, this number is always 1.

This data allows the program to access the current value of the steering wheel rotation angle. It is used in tests 1.9.4.7 "*Closed loop steering test: test with driver*", 1.9.4.8 "*Car simulator*" (position of trackbar **Steering wheel** when the steering control is off)

The maximal steering angle introduces a limitation on the control calculated by the program. If the calculated rotation angle is greater than this number, this value is assigned.

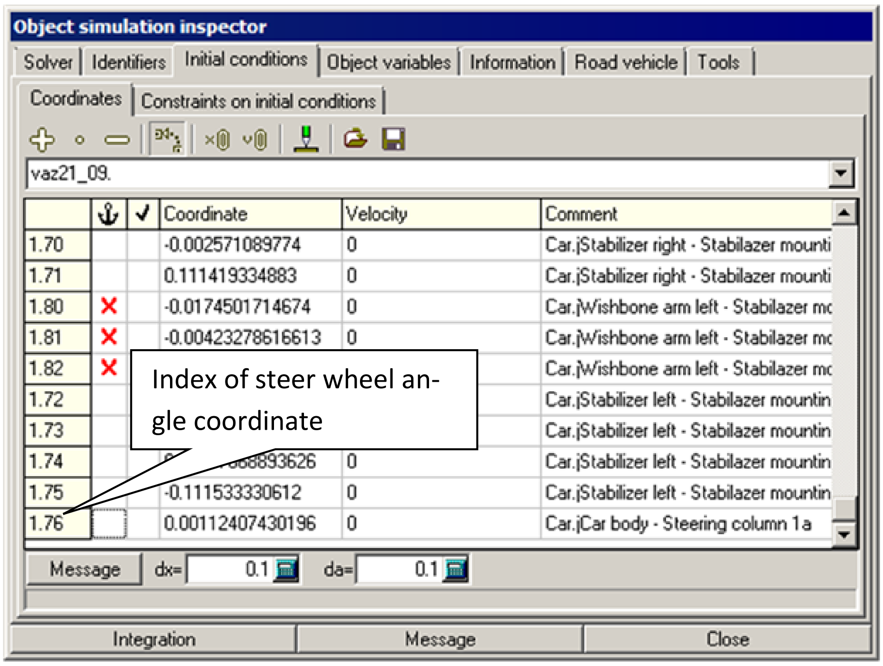


Figure 1.105. Identification of index of steering wheel angle coordinate

1.9.1.2.4. Parameterization of driver model

The assignment of identifiers to the parameters of the driver model is used in linear analysis to evaluate the impact of each of these parameters on control, see Sect. 1.10.2 "*Linear analysis of influence of driver model parameters*", as well as in multivariate calculations when analyzing the influence of driver model parameters on vehicle movement in curved sections of the road, Sect. 1.4.4 "*Selection of parameters for continuous control*".

The control parameters determine the continuous steering control (1.1)

$$T_p, t_d, K, K_2, K_d, K_I$$

Identifiers parameterizing the control should be added to the identifier list of the vehicle model in the UM Input program, Figure 1.106 left. The identifiers are assigned to the parameters on the **Road vehicle | Identification** tab of the simulation inspector in UM Simulation program, Figure 1.106 right.

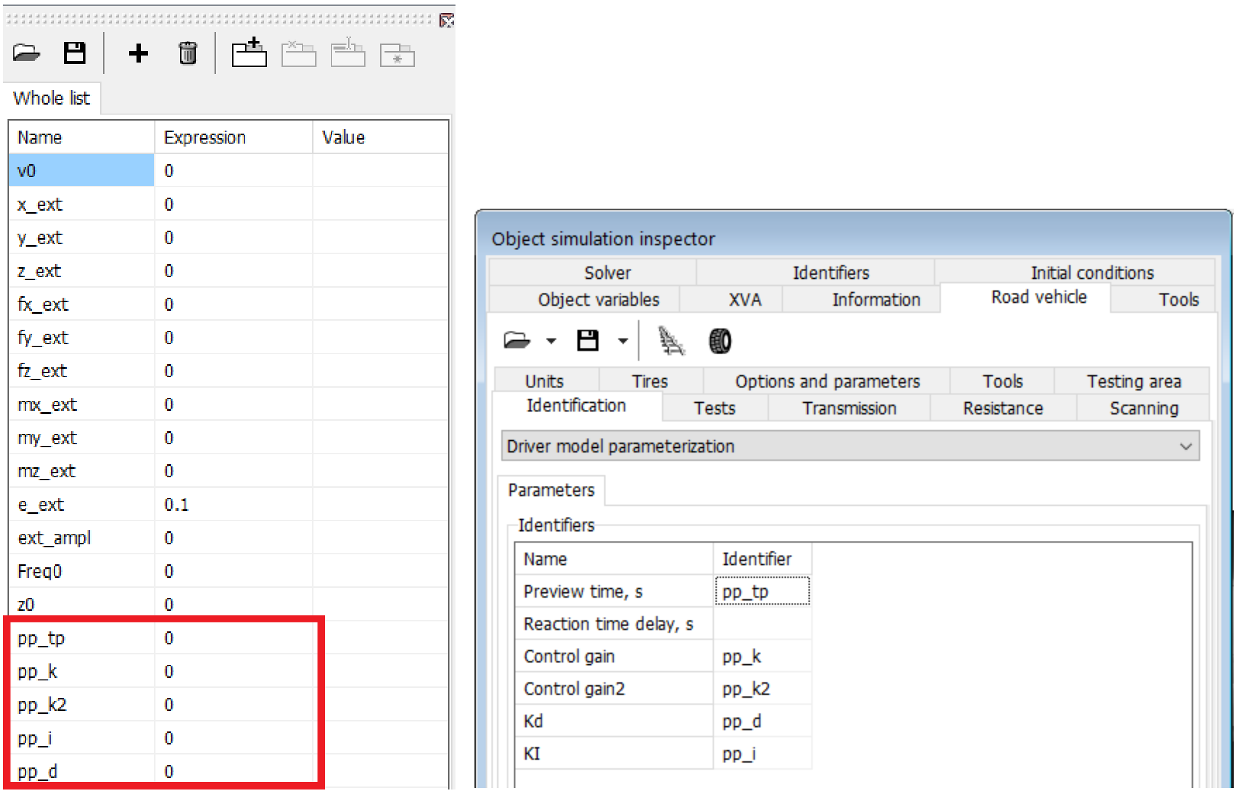


Figure 1.106. Assigned identifiers for driver model

1.9.1.2.5. Identification of transmission control

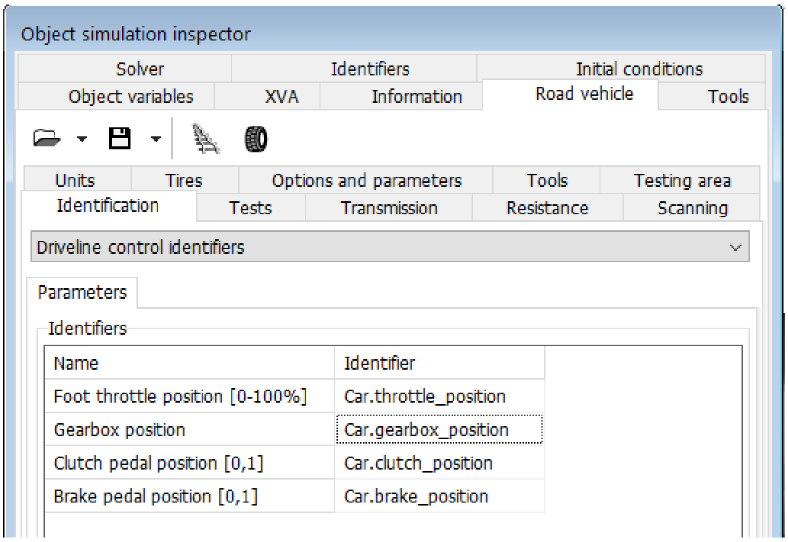


Figure 1.107. Assigned identifiers for transmission control

On this tab, identifiers are assigned that parametrize the transmission control. This assignment makes sense only if the vehicle model includes a transmission (engine, gearbox, clutch, etc.), see Sect. 1.8 “*Transmission*”.

A description of the transmission modeling in UM is contained in Chapter 22 of the user’s manual, file 22_UM_Driveline.pdf. The use of identifiers for control parameterization can be found in this file in section “*Identifier control*”.

1.9.1.3. Tools tab for creating data files

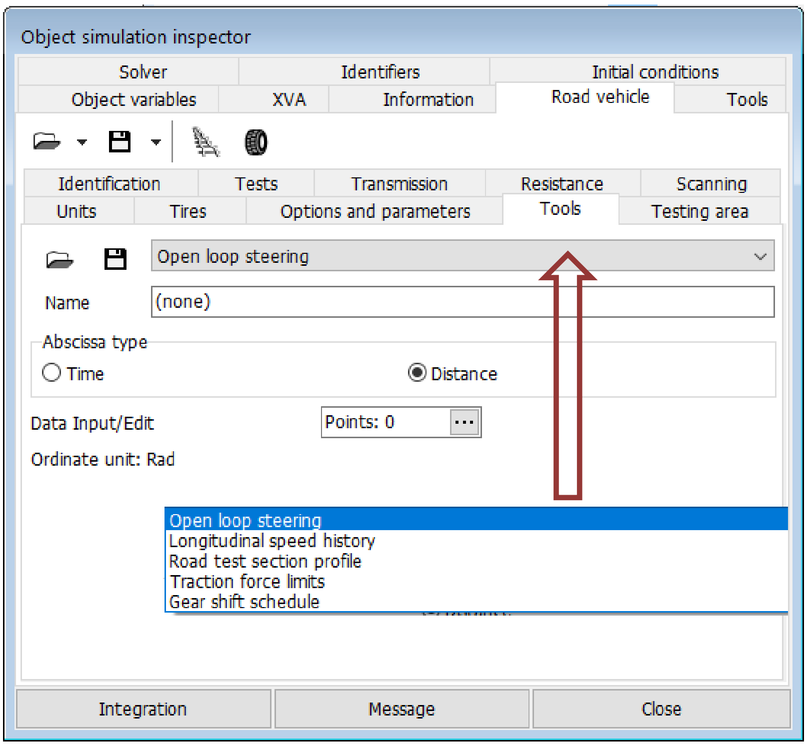


Figure 1.108. List of data types

When performing vehicle dynamics simulation tests, various types of data are required that contain the necessary information: steering angle versus time for an open loop vehicle control test, a plot of vehicle speed versus time or distance travelled, and so on. This type of data is created on the **Road vehicle | Tools** tab of the simulation inspector, Figure 1.108

Some of the data can be entered depending on the time or distance travelled, for example, the steering wheel rotation or the history of the longitudinal speed. In this case, the user should select the **Abscissa type**. Other data may depend on only one variable, for example, the test section profile depends on the longitudinal coordinate.

The button ⋯ serves for calling the curve editor, which is used for entering data.

The entered data is saved to a file using the button 💾. A previously created file can be opened for modification or viewing by clicking the button 📂.

Setting graphs for steering wheel angle and vehicle speed
Creating files with test section profiles
File with traction force limitation for simplified speed control
File with gear shift schedule

1.9.1.3.1. Setting graphs for steering wheel angle and vehicle speed

Open the **Road vehicle | Tools** tab in the simulation inspector, Figure 1.108.

Select one of the data types:

- Open loop steering;
- Longitudinal speed history.

Steering angle and speed profile can be functions of both time and distance, see, Figure 1.108.

The functions are described in the Curve editor by clicking on the ... button. An example of the steering wheel rotation history is shown in Figure 1.109.

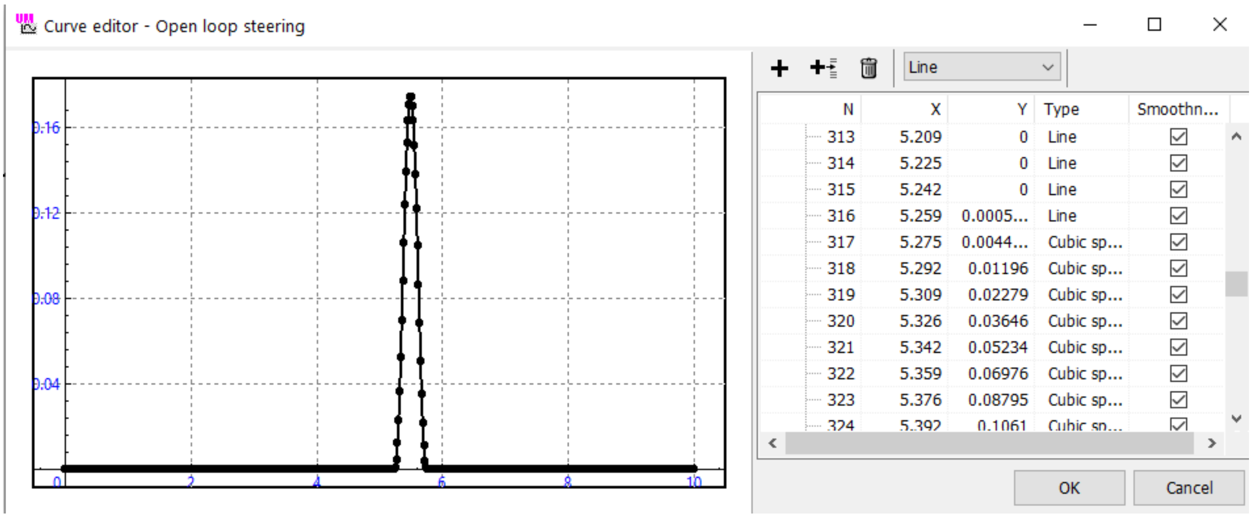


Figure 1.109. Pulse steer time history

The steering angle is entered in radians. The speed of longitudinal movement is set in meters per second.

Use the buttons to read/save data from/to file.

1.9.1.3.2. Creating files with test section profiles

Figure 1.110. Speed bump

Test section profiles (TSP) are geometric deviations of road from ideal state, which cannot be considered as smooth and small irregularities. For example, a step in Figure 1.45 or a speed bump in Figure 1.110 can be modelled in in UM as TSP only. **The tool is applied to the test with driver only**, see Sect. 1.9.4.7.2.3 “*Simulation with test section profiles (TSP)*”**.**

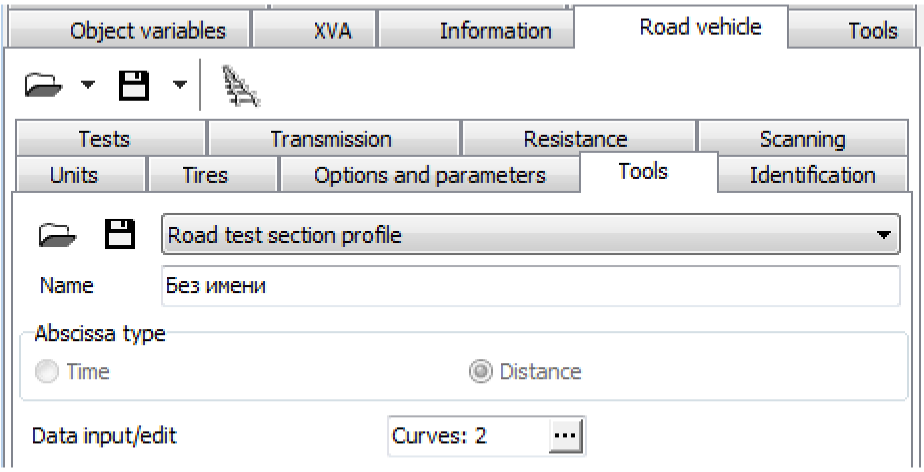


Figure 1.111. Tool for TSP description

TSP curves are created with a tool, located on the **Road vehicle | Tools** tab of the Object simulation inspector. Select the **Road test section profile** item of the pull-down menu and click on the … button to open the curve editor for description of the TSP, Figure 1.113.

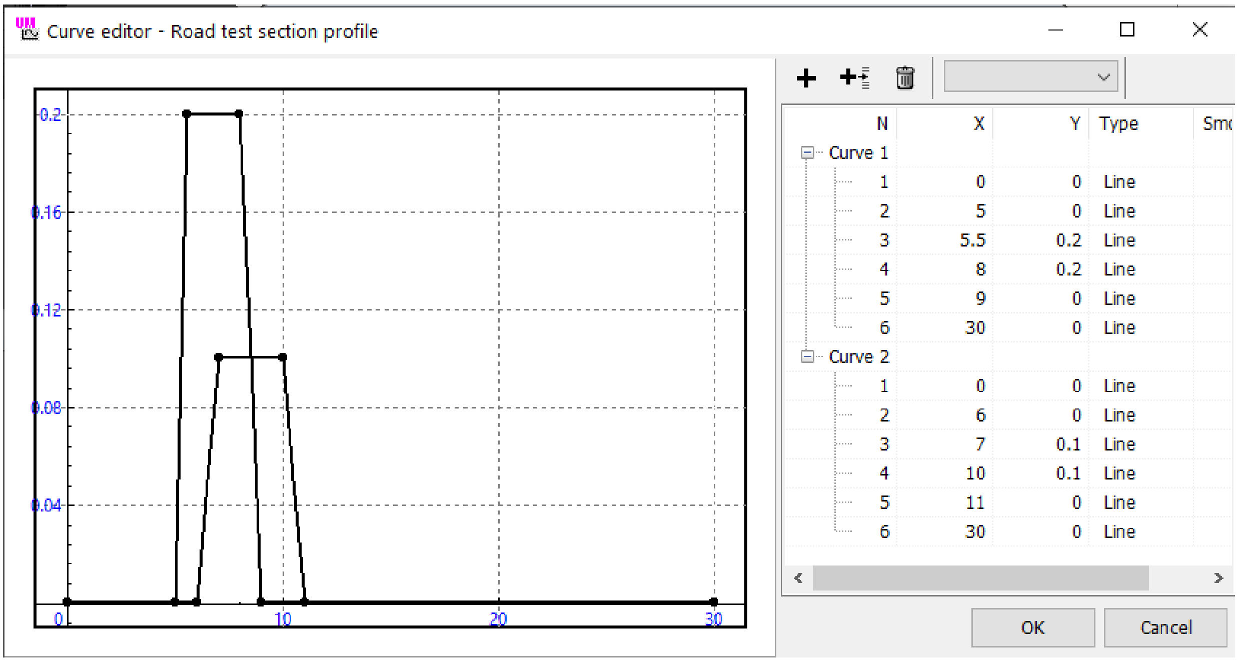


Figure 1.112. Different obstacles to the left and right tracks

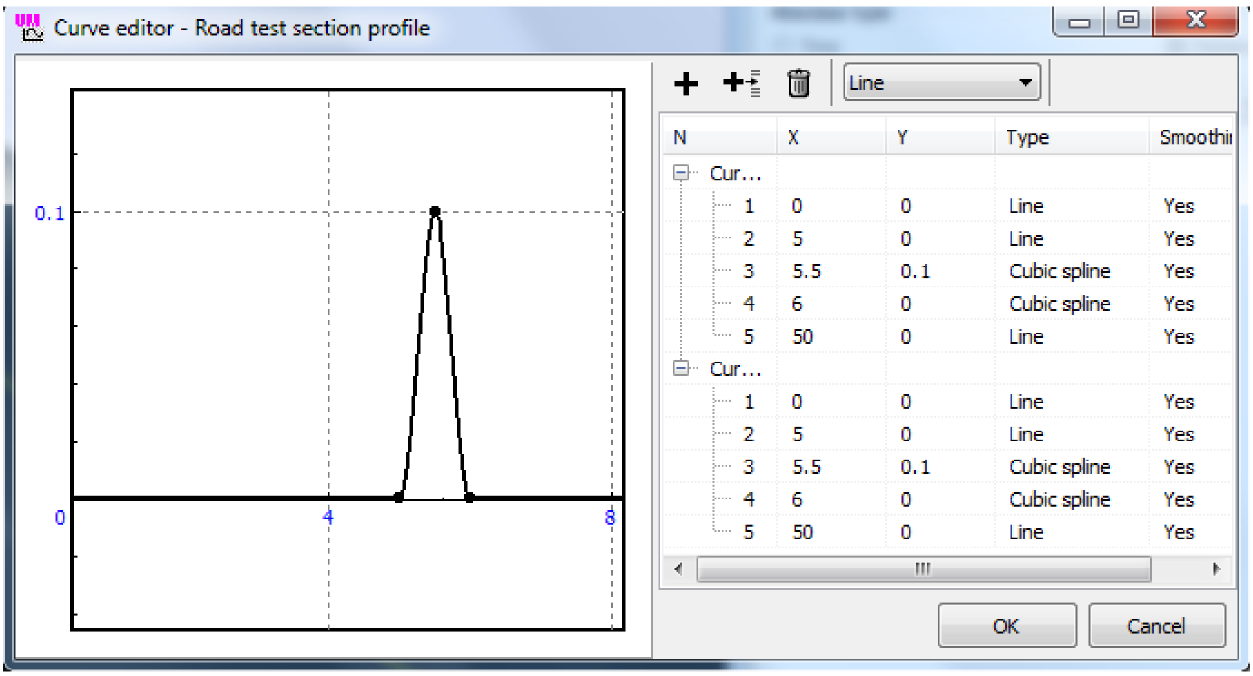


Figure 1.113. TSP curves for speed bump

If profile differs for the left and right tracks, the user should enter two curves like in Figure 1.112. The first curve corresponds to the left track. If only one curve is defined, the profile is considered as identical for the left and right tracks. That is, in the case shown in Figure 1.113, one curve can be deleted.

Use the button on the **Tools** tab to save the curves as a *.trp file.

1.9.1.3.3. File with traction force limitation for simplified speed control

The traction force limitation is the dependence of the limiting total torque transmitted to the wheels on the vehicle speed. The limit is used if a simplified speed control method is implemented without transmission simulation, Sect. 1.9.3.1 "*Speed modes for simplified longitudinal control*". The limit curve is saved in a *.tfl file. The torque unit is Nm, the speed unit is m/s.

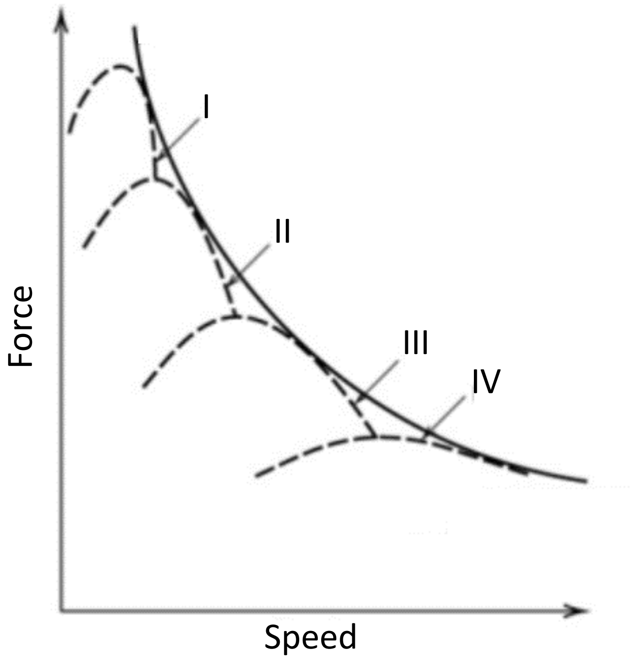


Figure 1.114. Force-speed diagram

To obtain the dependence of the limit on the value of the total torque on the traction wheels M depending on the vehicle speed v, the user can apply a typical force-speed diagram for a given vehicle. For a simplified version, it is proposed to use the torque limiting curve for rated power P_N

$$M < \frac{P_N r}{v}$$

where r is the wheel rolling radius. At low speeds, the slip limitation must be taken into account

$$M < \mu F_z r$$

Here F_z is the total vertical force for the driving wheels, μ is the coefficient of friction.

Thus, the limiting total torque on the traction wheels is determined by the formula

$$M_{max} = \min\left\{\frac{P_N r}{v}, \mu F_z r\right\}.$$

Consider an example: a 50kW motor, a wheel radius of 0.3m, a total load on the traction wheels of 5000N, and a tire-road friction coefficient of 0.8. The corresponding torque limiting curve is shown in Figure 1.115.

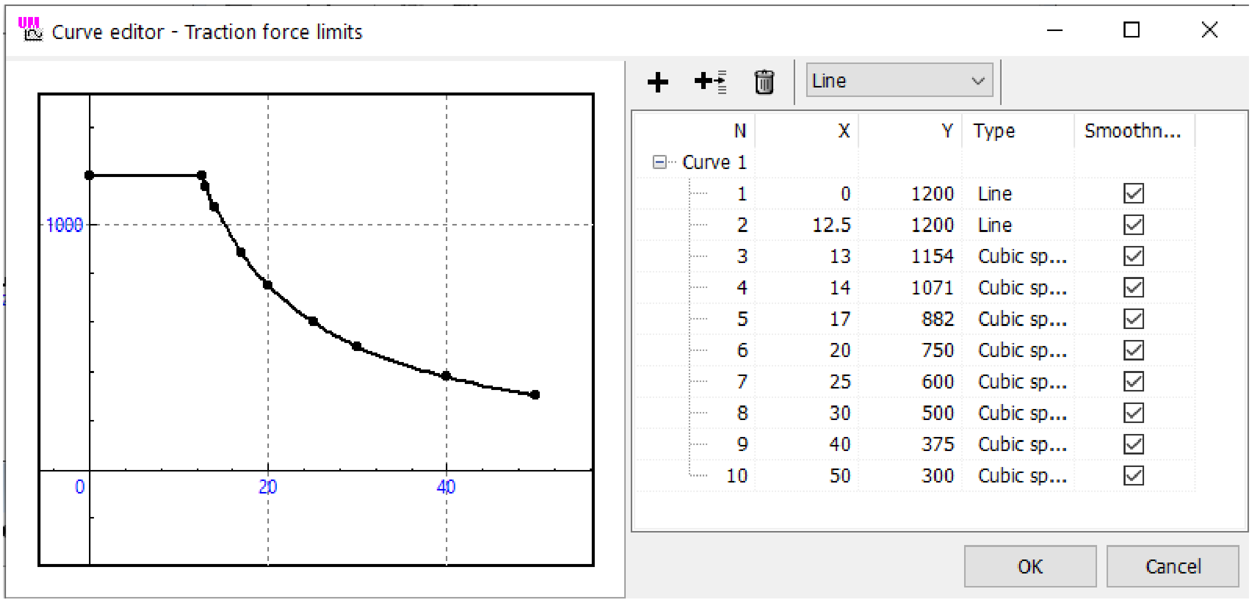


Figure 1.115. Example of driving torque limitation

1.9.1.3.4. File with gear shift schedule

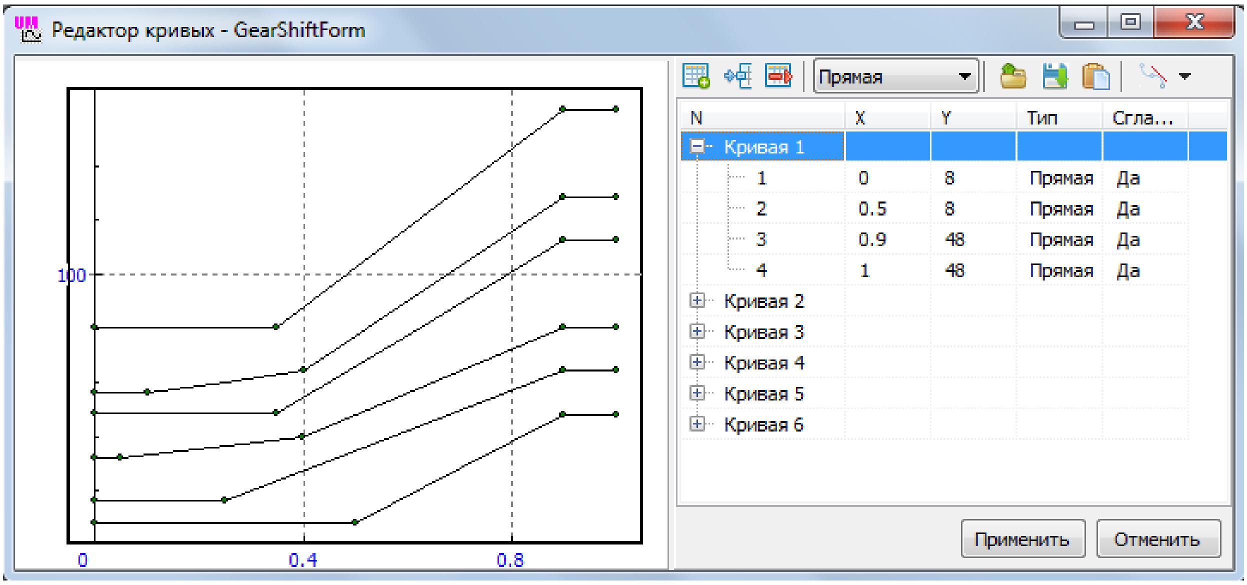


Figure 1.116. Gear shift curves

Gear shift schedule files *.gss are used for gear shifting in an automatic transmission, as well as in the longitudinal speed controller. The file contains gear upshift and downshift curves for vehicle or engine rotor speed versus throttle position, Figure 1.116.

More information can be found in Chapter 22, file 22_UM_Driveline.pdf, Sect. "*File with gear shift schedule*".

1.9.1.4. General Settings for Vehicle Dynamics Simulation

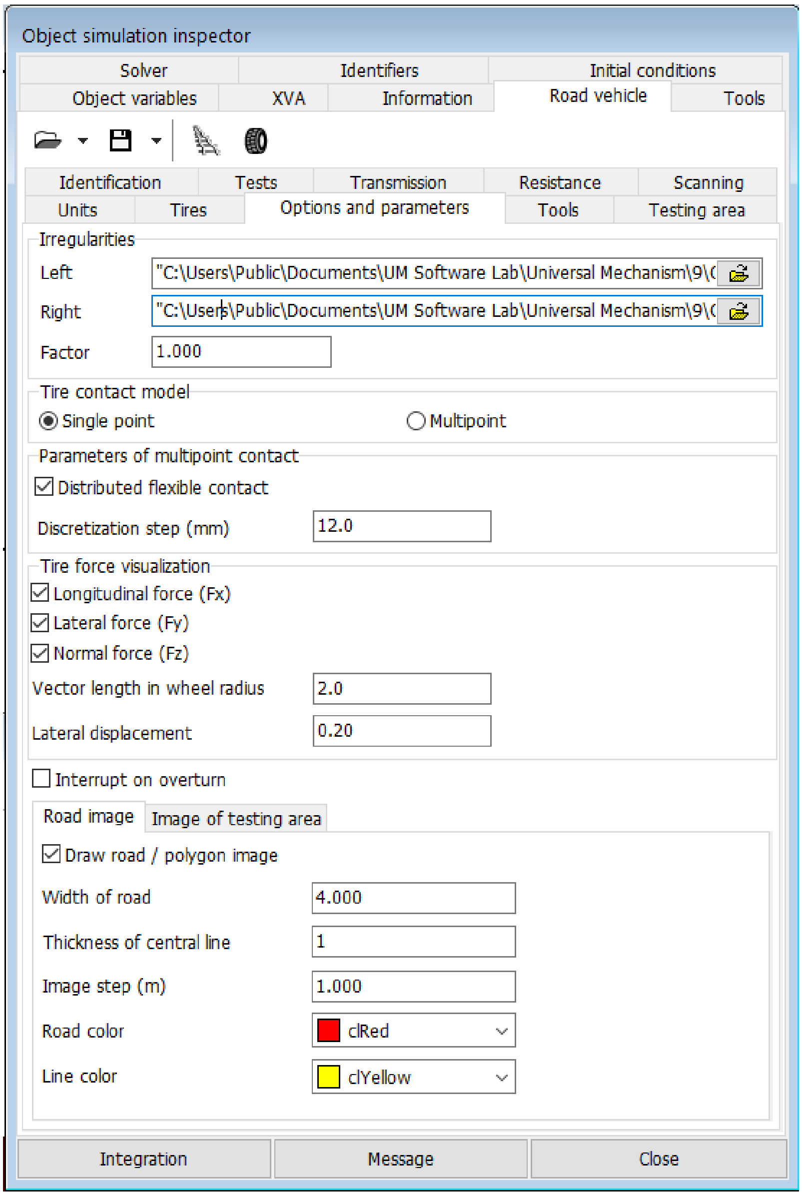


Figure 1.117. General option for road vehicle simulation

Below we consider options for simulation of vehicle on the **Road vehicle | Options and parameters,** Figure 1.117.

Setting road roughness files for left and right tracks
Options for tire/road contact

Animation of tire-road forces
Road image
Drawing the testing area for the open loop steering and simulator test
Vehicle overturn

1.9.1.4.1. Setting road roughness files for left and right tracks

To select a file, click on the button . Creation of irregularity files is described in Sect. 1.3.3 "*Micro profile (irregularities)*".

1.9.1.4.2. Options for tire/road contact

The user can select either a single point or multipoint contact model. In the case of a multipoint contact, the type of contact must be specified: discrete or distributed (the option **Distributed flexible contact**), as well as the desired value of the contact discretization step. It should be remembered that reducing the step together with the calculation refinement leads to a slowdown in the simulation process when moving along a triangulated surface.

A description of the models of tire contact with the road can be found in Sect. 1.5.1 "*Single point and multipoint normal contact models*".

1.9.1.4.3. Animation of tire-road forces

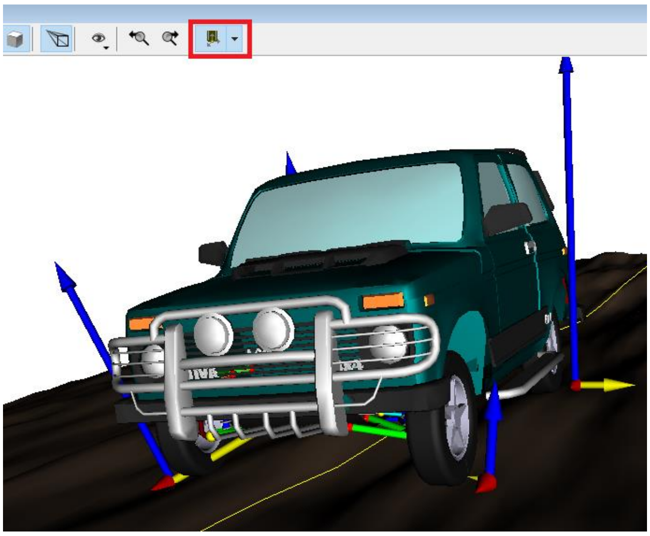


Figure 1.118. Animation of contact forces in animation window of old type

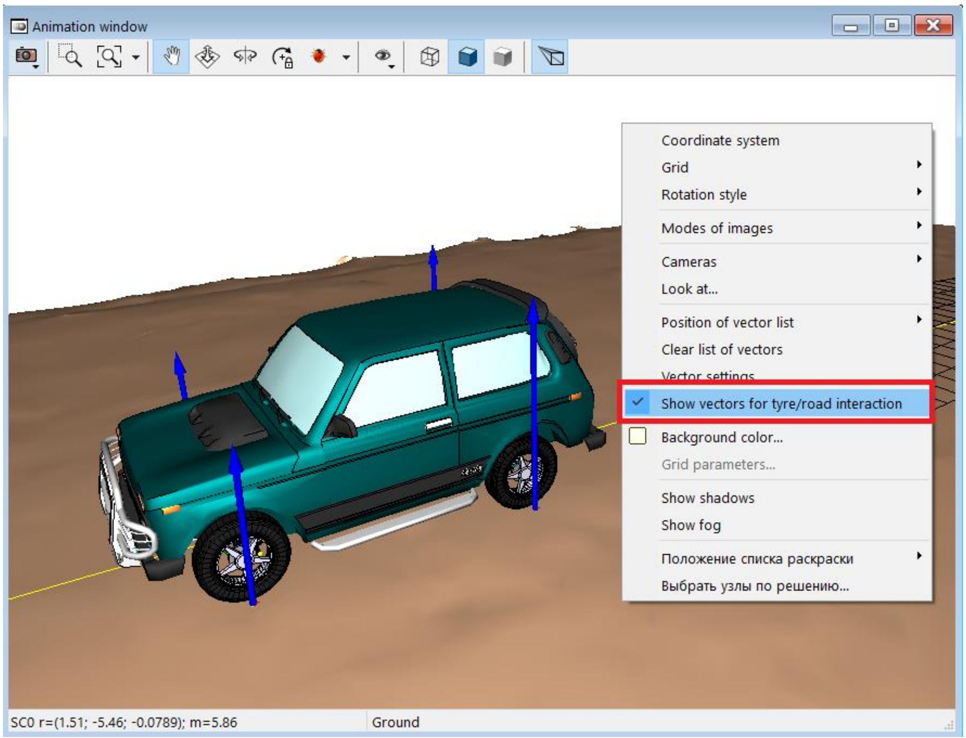
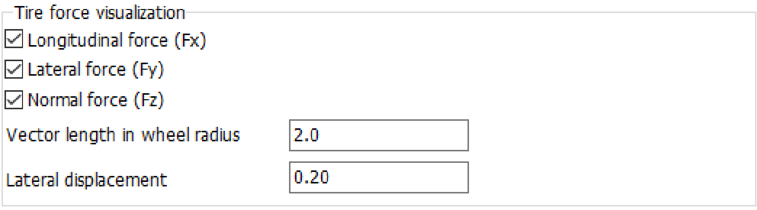


Figure 1.119. Animation of contact forces in animation window of new type

The force vectors of the interaction of tires with the road can be displayed in the animation window during the simulation. To enable the image of forces, one should

- in the animation window of the *old type*: click on the toolbar button , Figure 1.118;
- in the animation window of the *new type*: use the popup menu command **Show vectors for tire/road interaction**, Figure 1.119.



The corresponding section of the general options is used to customize the animation of the force vectors, Figure 1.117**:**

- the user can choose which forces are drawn, e.g. only lateral forces;

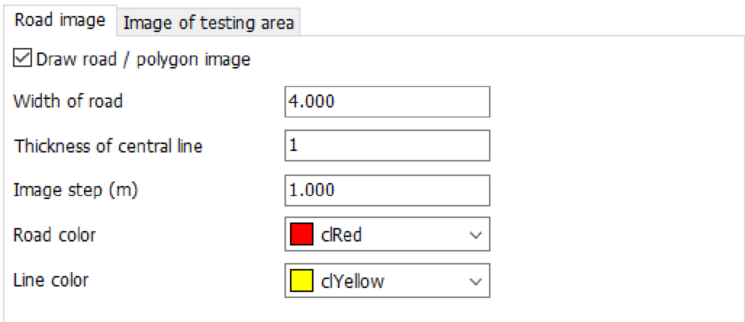
- the user can set the scale of the vectors relative to the tire radius by specifying the parameter *d* (**Vector length in tire radius**) - the length of the force vector *F* is calculated by the formula

$$\frac{Frd}{F_0}$$

where *r* is the radius of the first tire without load, F_0 is the static load for the first tire (if the equilibrium test is successfully passed) or the average load for one wheel;

- shift the point, to which the force images are applied, outward from the car by specifying a non-zero value in meters of the parameter in the box **Lateral displacement.**

1.9.1.4.4. Road image



The user can change the style of the road image in the animation window

- Hide the road image (in the animation window of new type);
- Set the road image width in meters
- Set the road image step in longitudinal direction in meters.

The road is drawn with quadrilaterals, the width of which in the longitudinal direction (step) can vary. These quads are visible in the wire mode of the animation window, Figure 1.120. When using test section profiles, it is recommended to reduce the step value for more accurate rendering of the profile, see Sect. 1.3.4 "*Test section profile*", 1.9.4.7.2.3 "*Simulation with test section profiles*".

- Change the color of the road and the trajectory curve that the vehicle follows in the driver test.

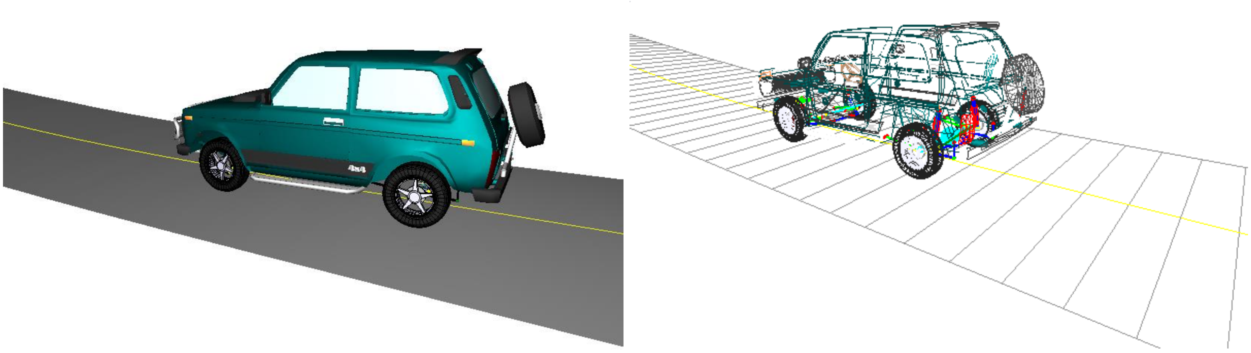
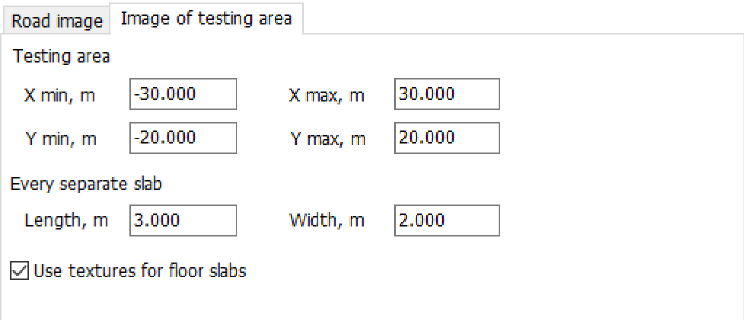


Figure 1.120. Road image in shaded and wire window mode

1.9.1.4.5. Drawing the testing area for the open loop steering and simulator tests



In the open loop steering test as well as in the simulator test, the horizontal surface on which the car moves is depicted in a new animation window using a generated periodic surface, Figure 1.181. The size of the surface image can be changed by the user.

Figure 1.121. Testing area image for the loop steering and simulator tests in a new window

1.9.1.4.6. Vehicle overturn

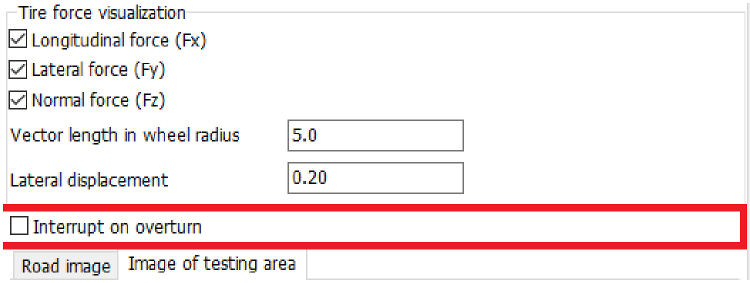


Figure 1.122. Option to stop simulation when the vehicle is turned over; the option disabled

There are two modes for simulation of a vehicle during an overturn. The modes are switched with the help of the **Interrupt on overturn** checkbox, Figure 1.122.

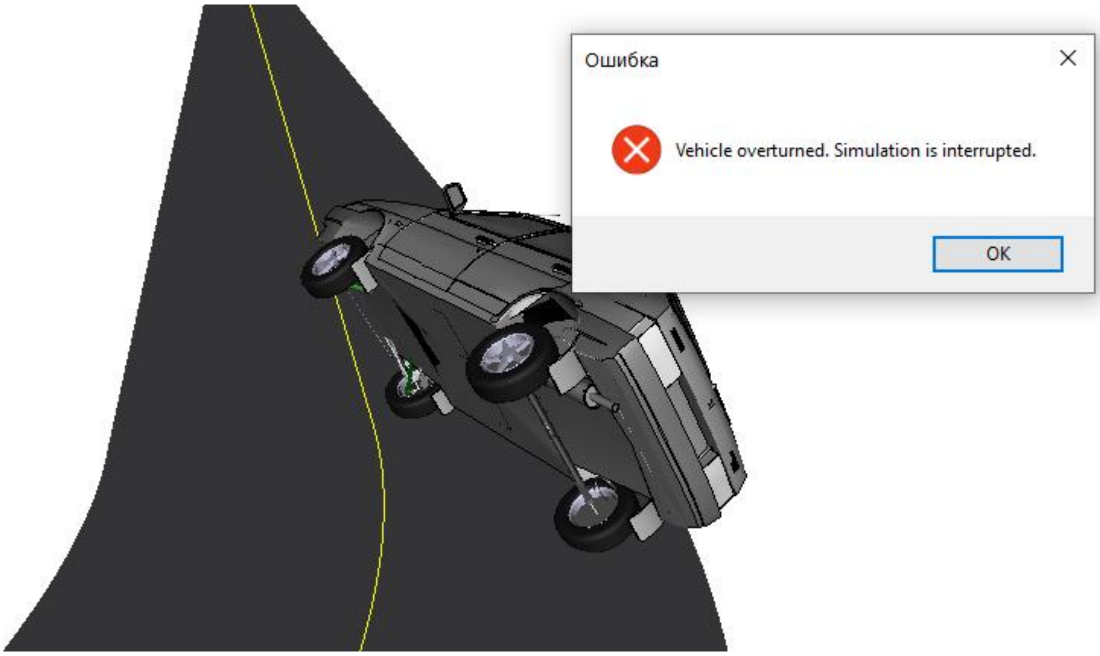


Figure 1.123. Simulation stop when vehicle overturned

- Simulation interrupt mode with error message

The simulation is interrupted when the initially horizontal plane associated with the body reaches a position close to vertical, or the symmetry plane of one of the wheels becomes close to horizontal, Figure 1.123.

- Simulation mode of the overturning process

If the simulation interruption mode during a turning over is disabled, then it is possible to simulate the entire process of the vehicle overturning, up to a complete stop when sliding, Figure 1.124. For this, contact interactions of the body with the terrain should be added to the model in case of a possible overturn. In the simplest case, this is a contact force element of the **points-plane** type, if the car moves along a horizontal plane, Figure 1.125. If the terrain is not flat, then a **points-Z-surface** contact can be used.

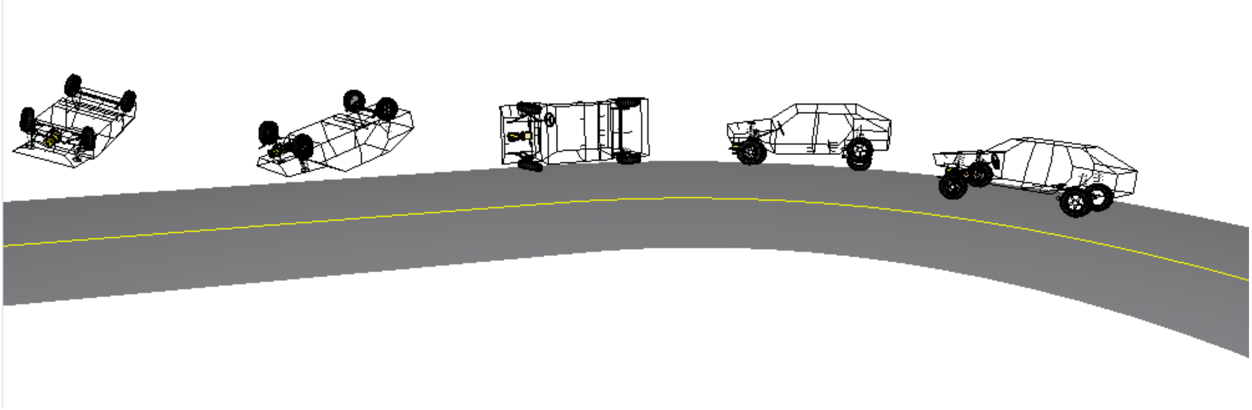


Figure 1.124. Simulation of vehicle overturning with contact forces

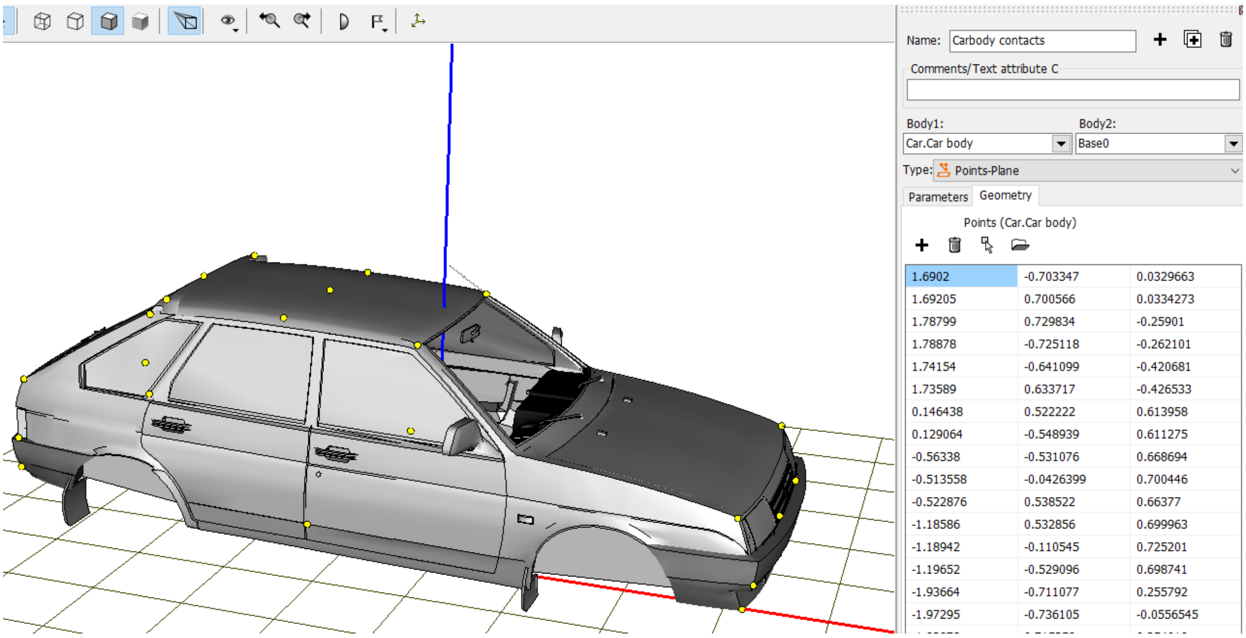


Figure 1.125. Contact force element for interaction of the car body with the horizontal plane, model ‘vaz2109 T’

1.9.2. Portability of a model from one computer to another

When executing tests, the program uses a number of files, without which simulation is impossible. If the model is transferred to another computer, then the auxiliary files created by the user will be found by the program if they are located directly in the car model directory. Here is a list of file types:

- Macro geometry (macro profile) *.mgf
- Irregularities (microprofile) *.irr
- Test section profile *.trp
- Open loop steering *.ols
- Gearshift logic *.gss
- Triangulated surface and route definition files *.img, *.rt

1.9.3. Speed modes and speed control

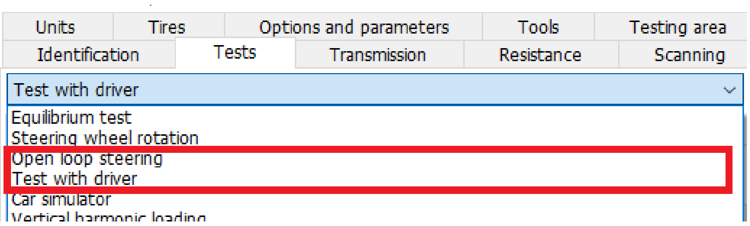
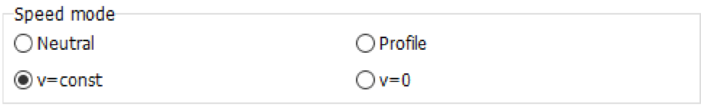
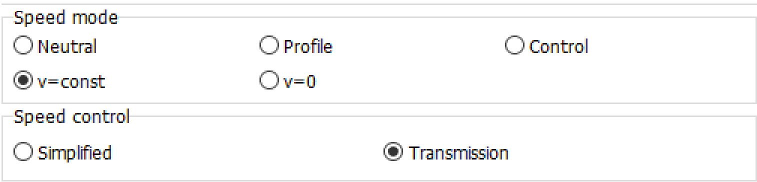


Figure 1.126. Main dynamic tests with longitudinal motion of vehicle



Model of vehicle without transmission



Model of vehicle with transmission
Figure 1.127. Speed modes

The two main dynamic tests with a vehicle model use several modes of longitudinal motion (speed modes). The main tests are (Figure 1.126)

Open loop steering test

Closed loop steering test: test with driver

The implementation and the number of modes of longitudinal movement depends on whether the vehicle model includes a transmission or not. For a model without a transmission, only a simplified speed control method is used, the parameters of which are entered in Sect. 1.9.1.2.1 “*Identification of parameters for simplified longitudinal speed control*”. In this case, the control torques are directly applied to the driving wheels of the vehicle. For models with transmission, along with a simplified method, a transmission control is used, including gear shifting, changing the position of the accelerator and clutch pedals, braking, etc.

The *v*0 identifier (Figure 1.128) is used to set the initial speed value in the **Neutral, v=const, Control** modes, Figure 1.127. In the **Profile** mode, the speed is set by a graph depending on the time or distance traveled, Sect. 1.9.1.3.1 “*Setting graphs for steering wheel angle and vehicle speed*”, the value of the identifier *v*0 is ignored in this case. The speed can be specified either in m/s or km/h. The speed unit for the current model is set by the user using the **Options** window (the **Tools | Options...** menu command) or the toolbar and is saved for the current model in the *.icf configuration file, Figure 1.129.

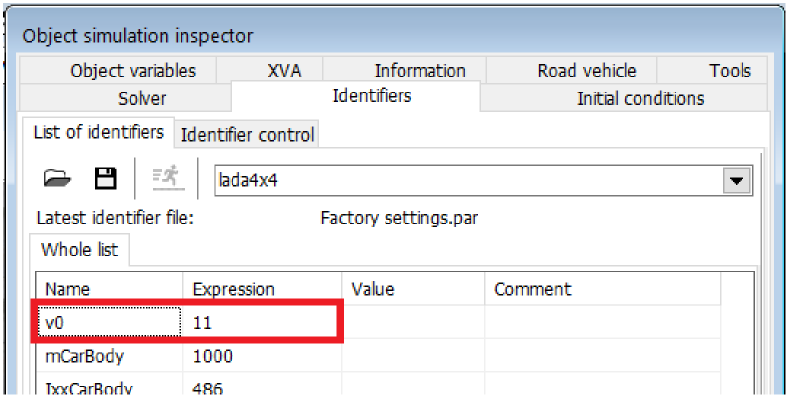


Figure 1.128. Setting initial speed by identifier v0

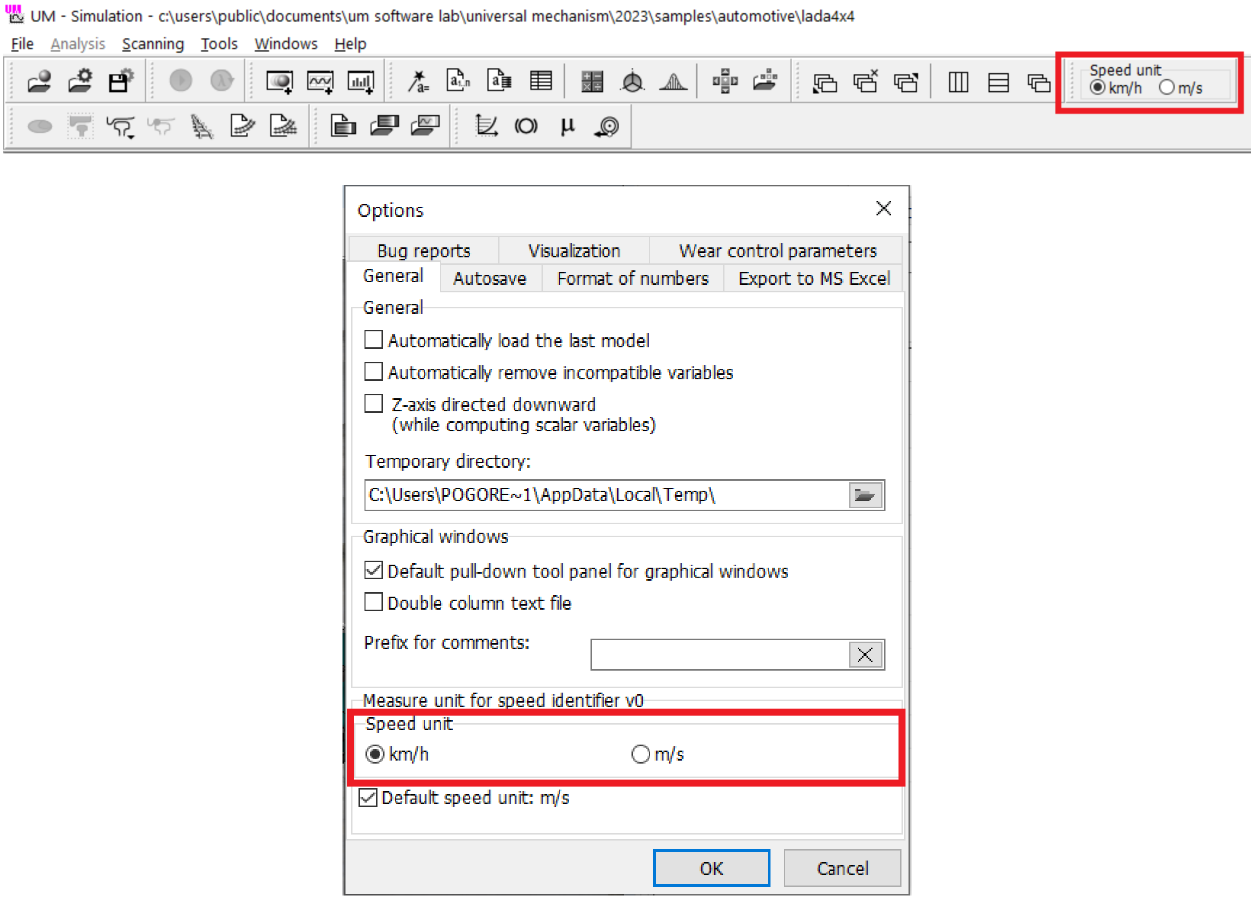
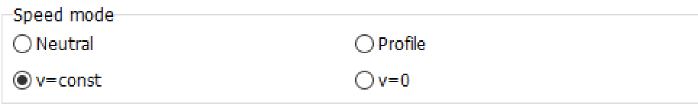


Figure 1.129. Setting speed unit

The zero speed mode **v=0** is similar to the equilibrium test and is mainly used to calculate the initial position of the car when driving along a triangulated surface (testing area), 1.9.4.7.3.2 “*Initial conditions for motion on testing area*”. In contrast to the equilibrium test, in this mode, the car body locking is automatically activated. When calculating the interaction of tires with a surface, the friction forces are assumed to be equal to zero, that is, the surface is considered to be absolutely smooth.

1.9.3.1. Speed modes for simplified longitudinal control



Four modes of longitudinal movement are available, Figure 1.127.

Neutral

The control is disabled, the motion occurs by inertia or under the action of longitudinal forces specified in the vehicle model.

v=const, Profile

A torque is applied to each drive wheel, depending on the difference between the desired and current speed. For this purpose, the identifier of the torque specified by the user is automatically used, Sect. 1.7.8 “*Force element for simplified control of the speed of the longitudinal movement*”, 1.9.1.2.1 “*Identification of parameters for simplified longitudinal speed control*”.

The proportional-integral speed controller is used

$$M = -K(v - v_d) - K_I \int_0^t (v - v_d)dt,$$

where M is the control torque (this values is assigned to the torque identifier),

v is the current vehicle speed, m/s; this is the speed of the vehicle car body,

v_d is the desired speed, which is either constant (**v=const**) or a given function of the time or distance (**Profile**), m/s.

K, K_I are the controller parameters specified by the user, Sect. 1.9.1.2.1 “*Identification of parameters for simplified longitudinal speed control*”.

The value of the moment can be limited due to the restrictions imposed by the engine power, Sect. 1.9.1.3.3 “*File with traction force limitation for simplified speed control*”.

v=0

The stationary vehicle mode described above in Sect. 1.9.3 “*Speed modes and speed control*”.

1.9.3.2. Speed modes when controlling the transmission

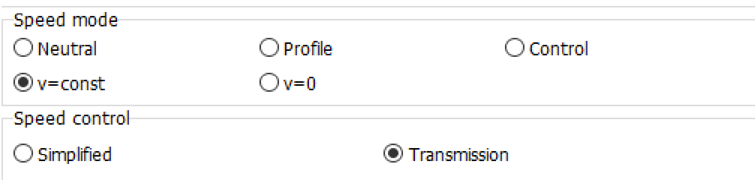


Figure 1.130. Speed modes under the transmission control

The section is actual for vehicle models with transmission, Sect. 1.8 “*Transmission*”. In this case, it is possible to use both the simplified longitudinal speed control described in the previous section, and the direct transmission control, including

- gear shift
- changing the position of the accelerometer pedal
- clutch control in the case of a manual transmission

- brake control

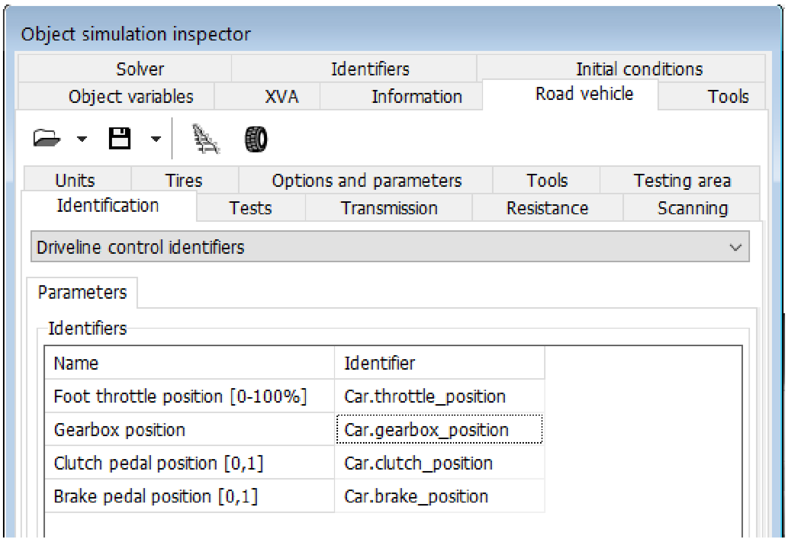


Figure 1.131. Assigned identifiers for transmission control

The transmission is controlled using four identifiers corresponding to these points, Sect. 1.9.1.2.5 "*Identification of transmission control*".

1.9.3.2.1. Speed mode "Neutral"

In this mode, the transmission control is carried out by directly setting values to identifiers by assigning constant or variable values. The user can use both the **Identifier control** tool and user-created control panels to set variable values. At the begin of integration, the program automatically starts the engine. If the gear position is not set to the neutral one, i.e. the corresponding identifier is different from zero, then the engine speed is automatically calculated in accordance with the longitudinal speed specified by the identifier *v*0 and the gear ratios of the gearbox and final drive.

Table 1.8

Some examples of the transmission control

Identifier	Options			
	1	2	3	4
Position of accelerator pedal (throttle_position)	0	0	>0	>0
Gear (gearbox_position)	0	0	>0	-1
Position of clutch pedal (clutch_position)	0	0	0	0
Position of brake pedal (brake_position)	0	>0	0	0

Several options for setting the values of control identifiers are given in Table 1.8. It is assumed in the examples that when the given values of the identifiers are set, the modeling process is started and these values are not changed by the user.

Option 1. The car moves by inertia with a small deceleration due to air resistance and rolling friction of the wheels.

Option 2. Vehicle is braking.

Option 3. The vehicle speed tends to a constant value, which depends on the position of the accelerator pedal and the gear position. The initial speed identifier value *v*0 must be positive.

Option 4. Reverse motion. The speed of movement tends to a constant value, which depends on the position of the accelerator pedal. The initial speed identifier value *v*0 must be negative.

To shift gears in this mode, the user can use the **Identifier control** tool. Consider an example. At the beginning of the simulation, the car speed is zero (*v*0=0), the accelerator pedal is pressed by 20%, the gear is in neutral position, the clutch pedal is released, Figure 1.132. When the simulation begins, the engine starts. At time 1s, the clutch is disengaged and the first gear is engaged. During the next 1.5s, the clutch is released.

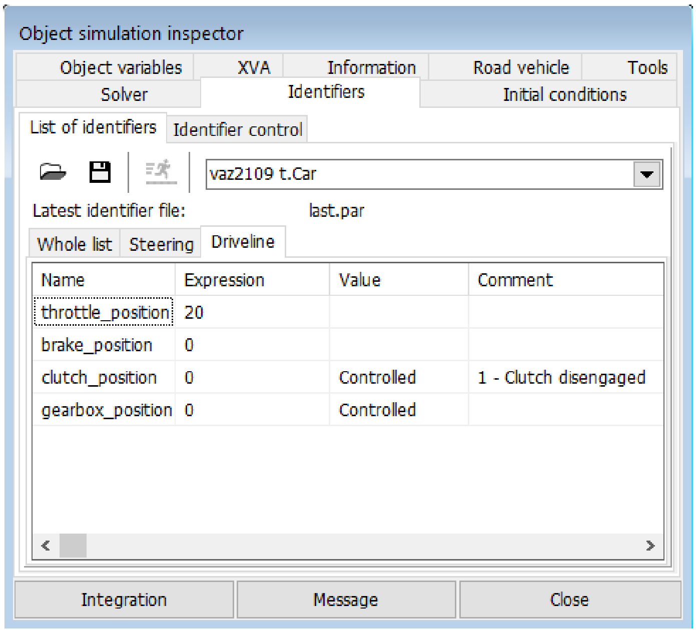


Figure 1.132. Accelerator pedal position 20%

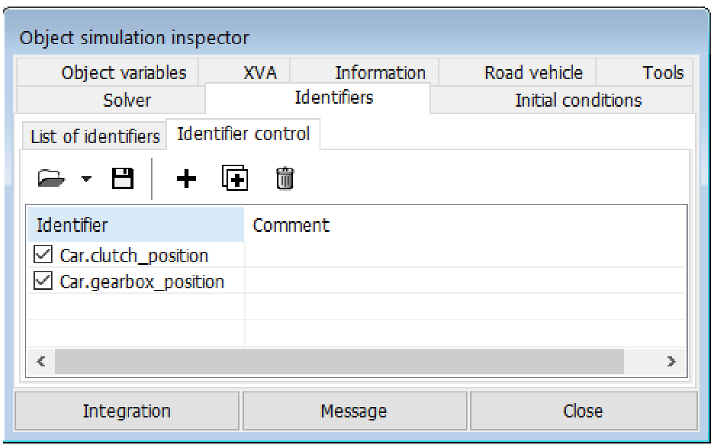


Figure 1.133. Control for gear and clutch identifiers

Consider an example when then control is carried out by two identifiers, Figure 1.133. Time history for the identifiers realizing the described scenario are shown in Figure 1.134.

This example is implemented for the model {Data UM}\SAMPLES\Automotive\Vaz2109 T. To view the example, open the model full configuration **Identifier control for transmission**.

The time dependences of the engine speed and the longitudinal speed of the car during the simulation are shown in Figure 1.135.

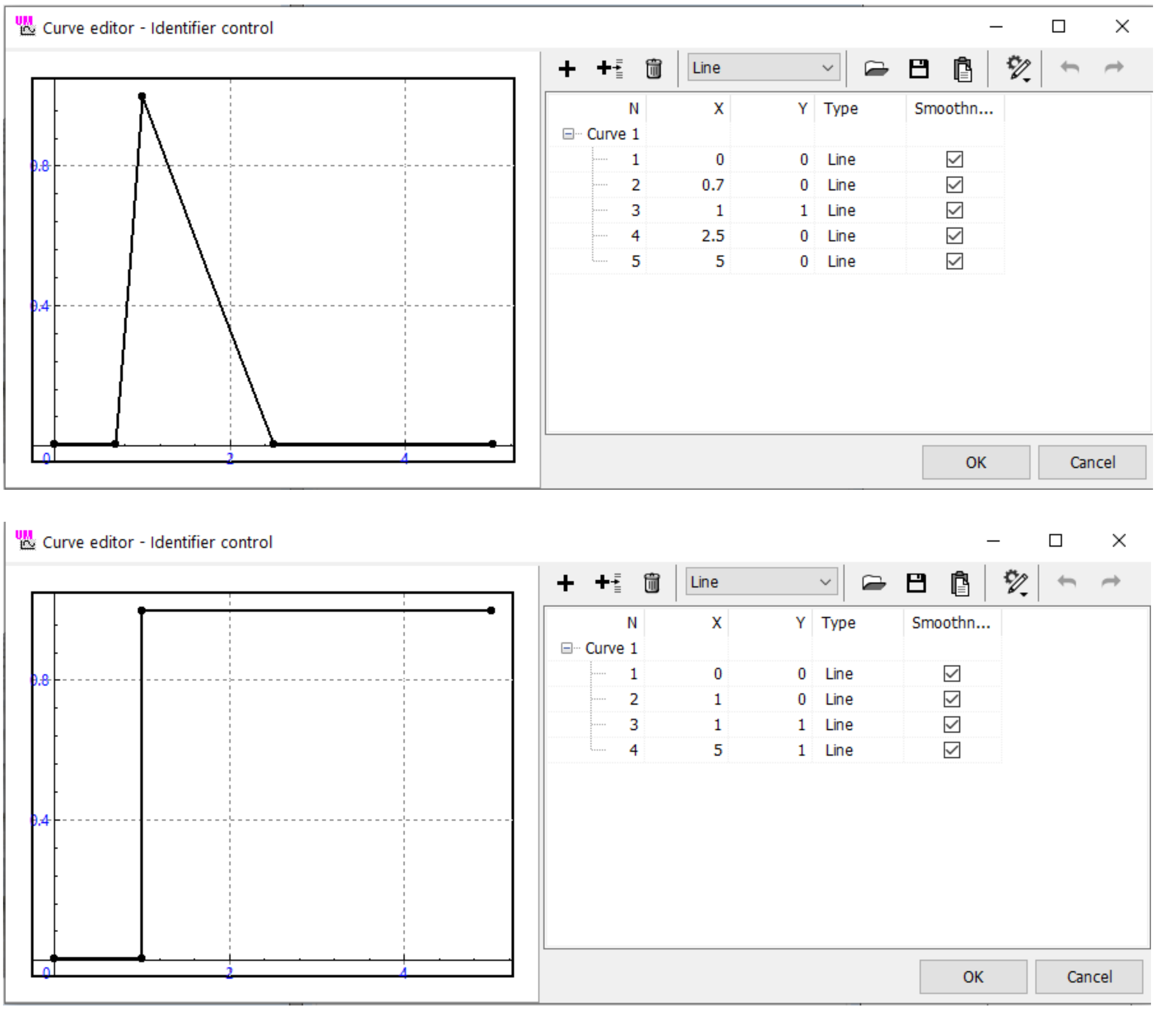


Figure 1.134. Time histories for the clutch and gear identifiers

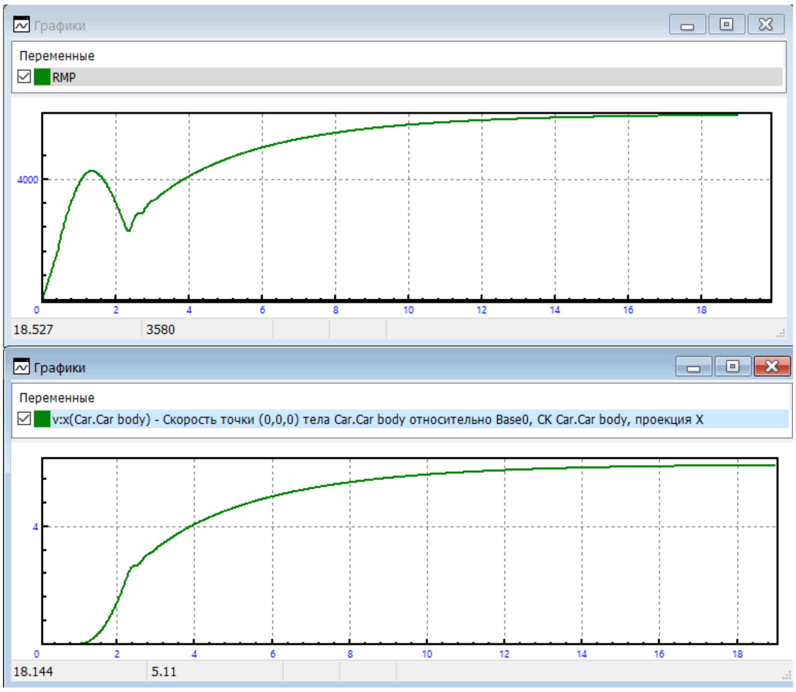


Figure 1.135. Engine speed and car speed vs time

1.9.3.2.2. Speed modes “v=const” and “Profile”

A full automatic control of the transmission is provided in these speed modes, when the speed is constant or set by a graph depending on time or distance, see Sect. 1.9.3.3 “*Transmission controller*”.

1.9.3.2.3. Speed mode “Control”

This speed mode is a combination of transmission control described in Sect. 1.9.3.2.1 *Speed mode “Neutral”*, 1.9.3.3 “*Transmission controller*”:

- the accelerator pedal and brakes are controlled directly via the identifiers, as described in Sect. 1.9.3.2.1 *Speed mode “Neutral”*;

- gear shifting is controlled of the transmission controller, Sect. 1.9.3.3 “*Transmission controller*”.

1.9.3.3. Transmission controller

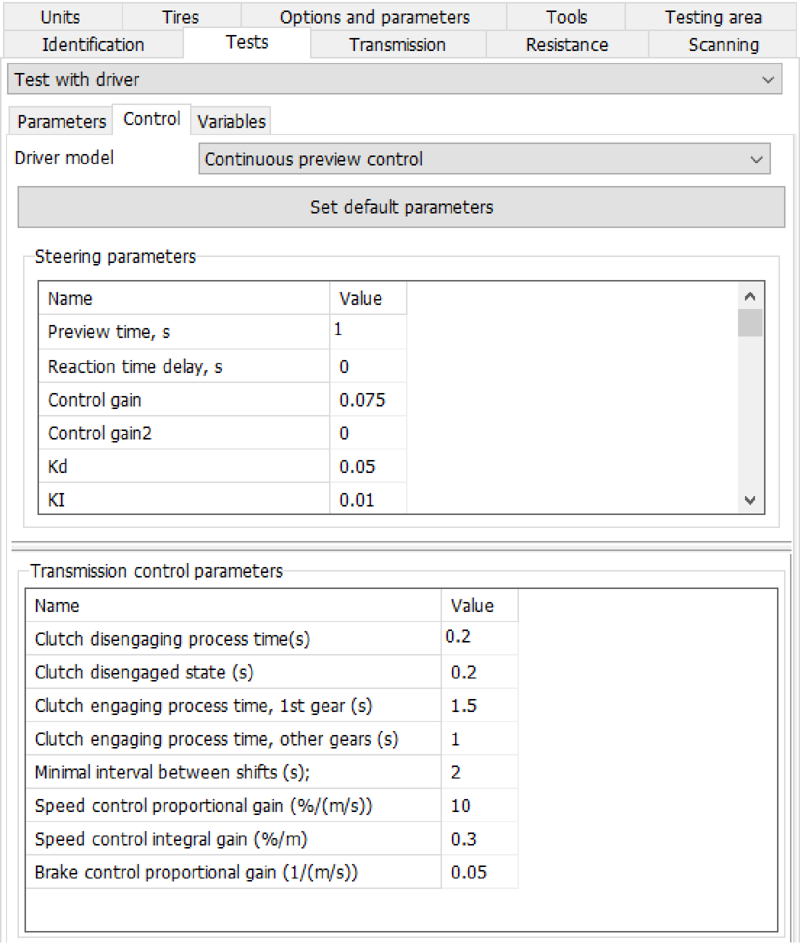


Figure 1.136. Parameters of transmission controller

The transmission controller is a model of automatic control of the longitudinal motion of the vehicle when the model has a transmission, Sect. 1.8 “*Transmission*”. See Chapter 22 for general information about modeling transmissions in UM.

1.9.3.3.1. Parameters for automatic control of transmission

The controller parameters are available on the tab **Control,** Figure 1.136. The list of parameters contains three groups.

- Parameters for clutch pedal operation

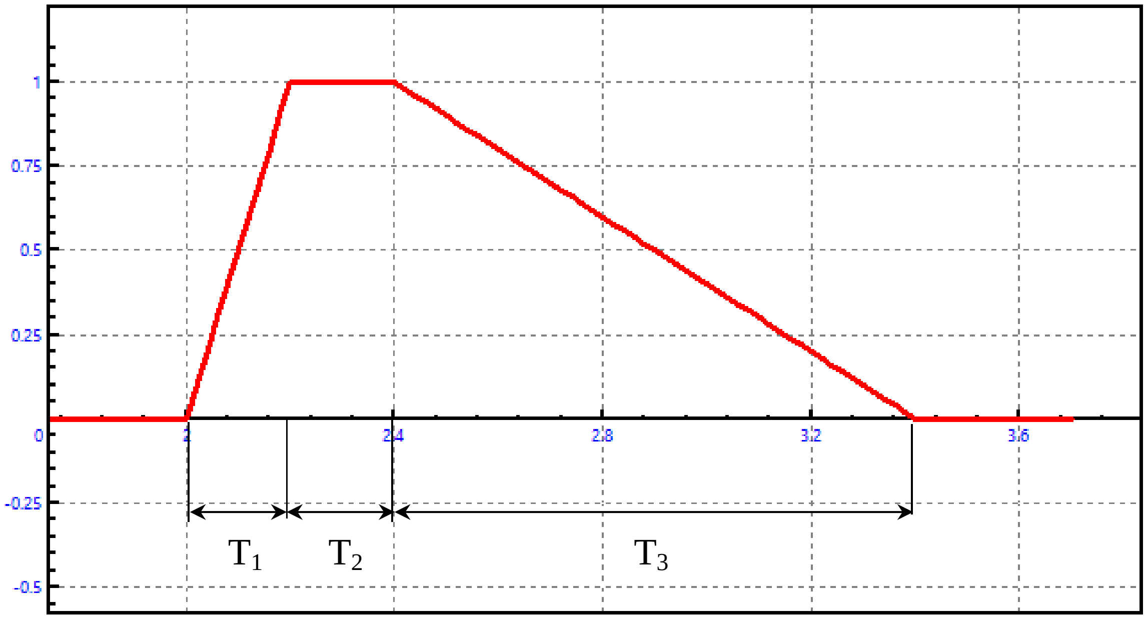
A typical graph of the clutch pedal position identifier during gear shifting is shown in Figure 1.137. It is important to remember that the value 0 means the clutch is engaged, and 1 means the clutch is disengaged. The first four controller parameters in the table correspond to the duration of three time intervals T_1, T_2, T_3:

T_1 – Clutch disengaging process time(s),

T_2 – Clutch disengaged state (s),

T_3 – Clutch engaging process time, 1st gear (s),

T_3 – Clutch engaging process time, other gears (s).



- Minimal time interval between gear shifts

The controller cannot shift the gear if the time interval since the last shift is less then this value.

- Parameters for control of accelerator pedal

The proportional-integral control of the accelerator pedal is applied. The controlled variable is the difference between the desired and current speed of vehicle:

$K_{acc,p}$ – Speed control proportional gain

$K_{acc,i}$ – Speed control integral gain

- Braking control

The proportional control of the brake pedal is used.

K_b – Brake control proportional gain.

This control of the accelerator and brake pedals is used in the cruise control modes described in Sect. 1.9.3.2.2 “*Speed modes “v=const” and “Profile*”

1.9.3.3.2. Gear shift

For gear shifting, a file containing upshift and downshift curves depending on vehicle speed or engine speed is used, Sect. 1.9.1.3.4 “*File with gear shift schedule*”. Detailed information can be found in Chapter 22, file 22_UM_Driveline.pdf, Sect. “*File with gear shift schedule*”. If a manual transmission is implemented in the vehicle model, then the controller disengages the clutch before switching the gear, and after switching the gear it releases the clutch, Figure 1.137. Gear shifting controller is used in speed modes **v=const, Profile, Control**.

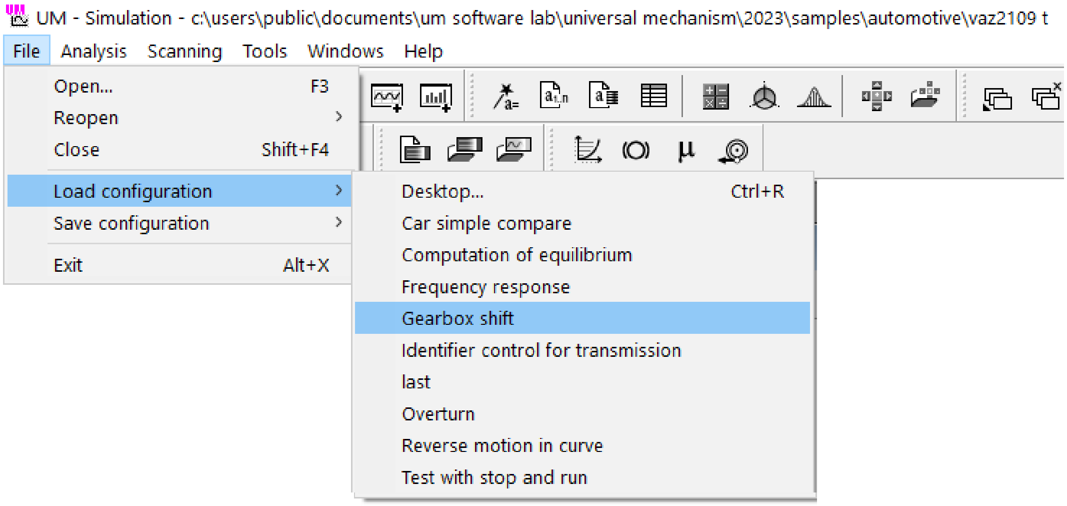


Figure 1.138. Loading a full configuration

Example.

Open the model {Data UM}\SAMPLES\Automotive\Vaz2109 T in UM Simulation program and load the full model configuration **Gearbox shift**. Simulation of vehicle motion in a straight line with variable speed is performed with a constant position of the accelerator pedal 100%, Figure 1.139.

Dependences of the vehicle speed on time, graphs of clutch pedal position and gear shifting during simulation are shown in Figure 1.140. During the test, the vehicle reaches a speed close to the maximum speed for a given engine type.

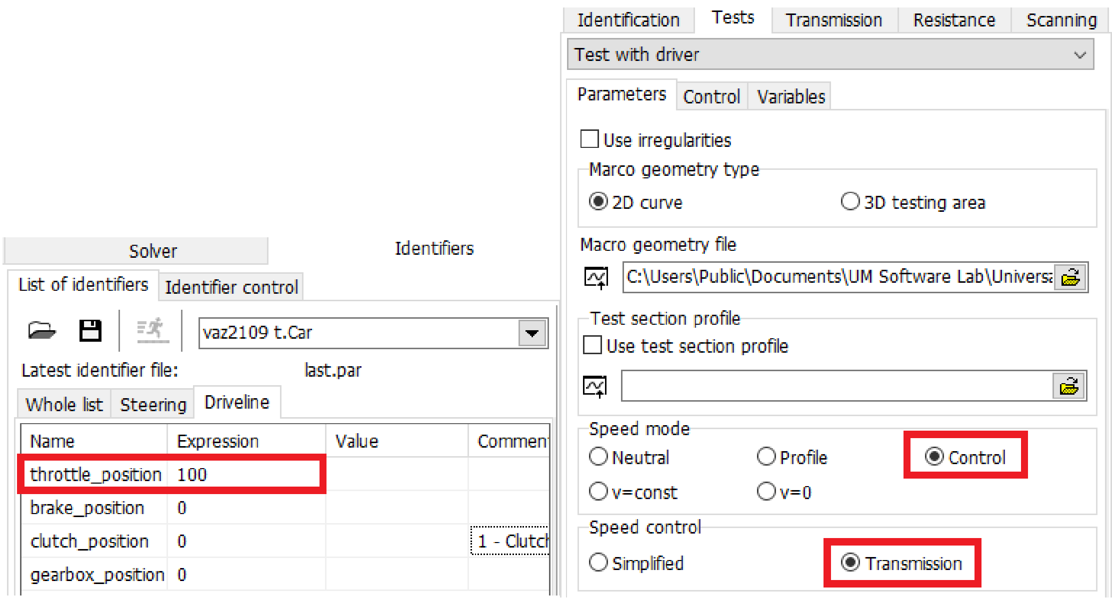


Figure 1.139. Options of gear shift test

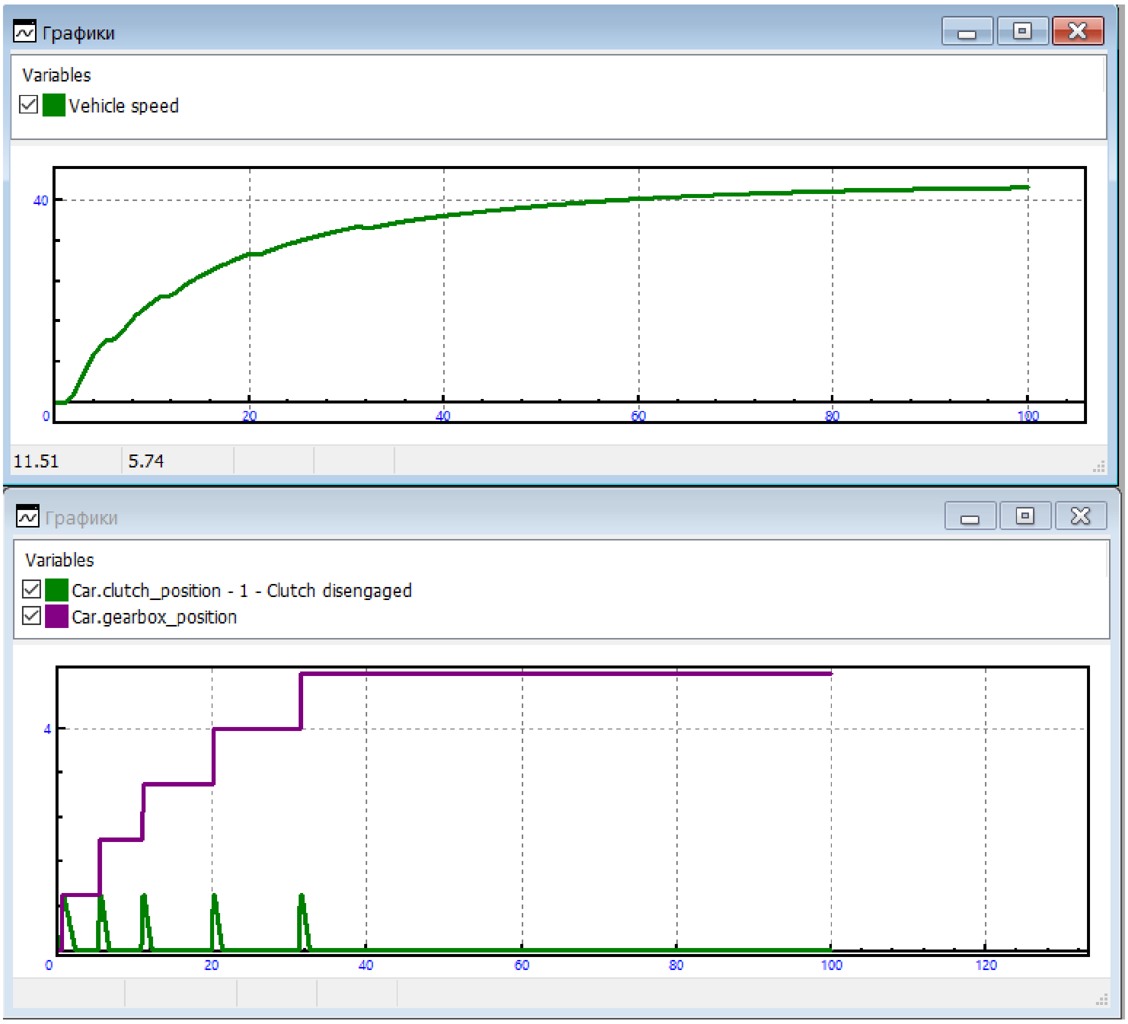


Figure 1.140. Longitudinal vehicle speed, clutch control and gear shifting

1.9.3.3.3. Accelerator and brake control

This type of control is used only in the **v=const**, **Profile** speed modes, that is, in cases where the dependence of speed on time or position is known. In these modes, it is possible to estimate the predicted acceleration of vehicle a_p. Thus, it is possible to determine the total driving resistance force, consisting of the aerodynamic drag force F_a, tire rolling resistance F_r, the gravitational component when driving on an inclined section of the road F_g and the inertial force $F_i = -ma_p$. The total resistance is

$$R = F_a + F_r + F_g + F_i.$$

First, consider the traction mode, that is, the case when it is necessary to press the accelerator pedal by a certain position to realize the required torque on the engine shaft to maintain a given speed. Neglecting the losses in the transmission, it is possible to estimate the predicted value of the torque on the motor shaft M_{ep} from the power balance

$$M_{ep}\omega_e = Rv.$$

Here ω_e, v are the current values of the angular velocity of the motor shaft (rad/s) and the longitudinal velocity of the vehicle (m/s), approximately coupled by the relation

$$\omega_e \approx \frac{v i_g 3.6}{r_w i_m \, 30/\pi}.$$

The multipliers 3.6 and $30/\pi$ convert the transmission ratio i_m to m/s / rad/s from the original km/h / rpm, see Chapter 22, section “Identifier for transmission ratio”.

The final expression for the predicted engine torque at the current vehicle speed and the gear position is

$$M_{ep} = \frac{R r_w i_m \, 30/\pi}{i_g 3.6} \text{ Nm}.$$

Based on the calculated value of the torque M_{ep} at the current engine speed, the predicted position of the accelerator pedal $\eta_{acc,p} \in [0.100\%]$ is calculated. For this, the engine torque map of the internal combustion engine is used (see Chapter 22, section “Engine torque map”). The resulting pedal position is the sum of the predicted value and two terms of proportional-integral speed control

$$\eta_{acc} = \eta_{acc,p} - K_{acc,p}(v - v_d) - K_{acc,i} \int_0^t (v - v_d) dt,$$

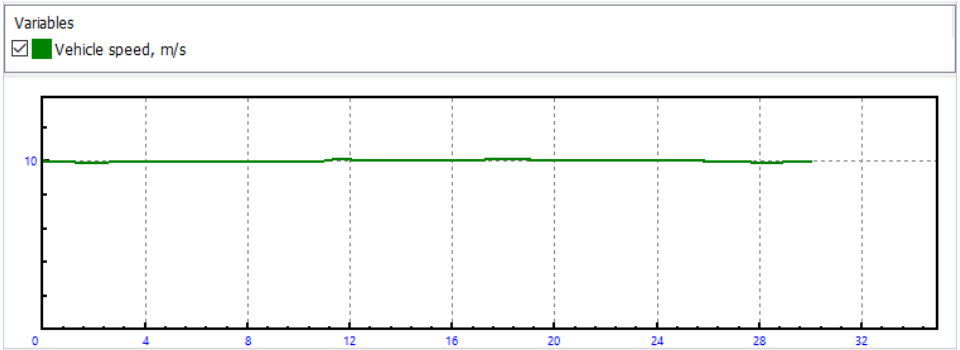
Here v, v_d are the current and desired speeds, $K_{acc,p}$, $K_{acc,i}$ are the coefficient of the PI controller, Sect. 1.9.3.3.1. “*Parameters for automatic control* of transmission”.

If the obtained value of η_{acc} is negative, it is set to zero and a transition to the braking mode is possible. Braking is activated if the sum of the predicted torque M_{ep} and the idle torque M_{ei} is negative, i.e. engine braking does not provide the desired speed reduction. The brake pedal position $\eta_b \in [0,1]$ is calculated as the sum of η_{bp}, which provides the brake force predictive value, and the proportional brake control with gain K_b, Sect. 1.9.3.3.1. “*Parameters for automatic control* of transmission”.

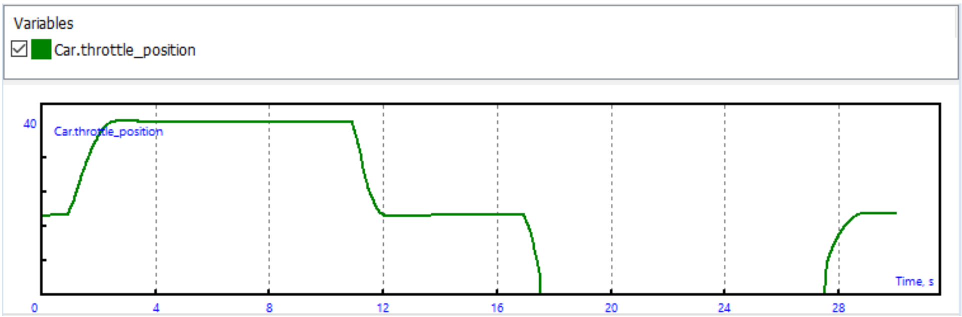
$$\eta_b = \eta_{bp} - K_b(v - v_d).$$

Example 1. Motion at a constant speed, ascending and descending

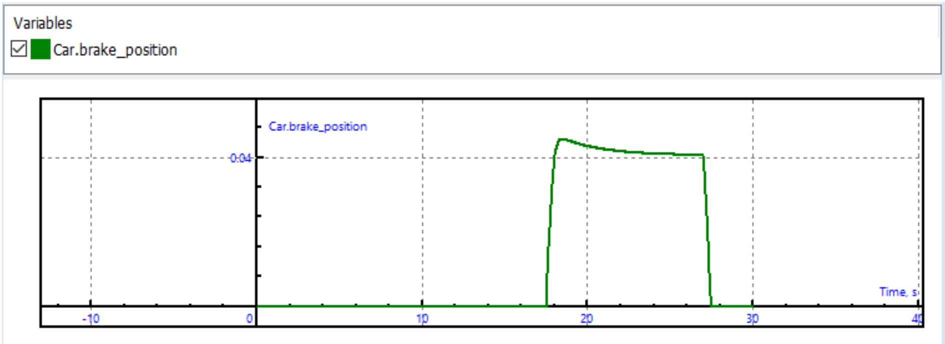
Open the model {Data UM}\SAMPLES\Automotive\Vaz2109 T in UM Simulation program. Load the full model configuration **v=10**. The case study is the motion with a constant speed of 10 m/s. The macro-geometry file contains straight-line path on a road with ascending and descending sections of 10% with a 100m length of sections. The simulation results are shown in Figure 1.141. The described control system allows keeping almost constant speed.



Speed



Accelerator pedal position



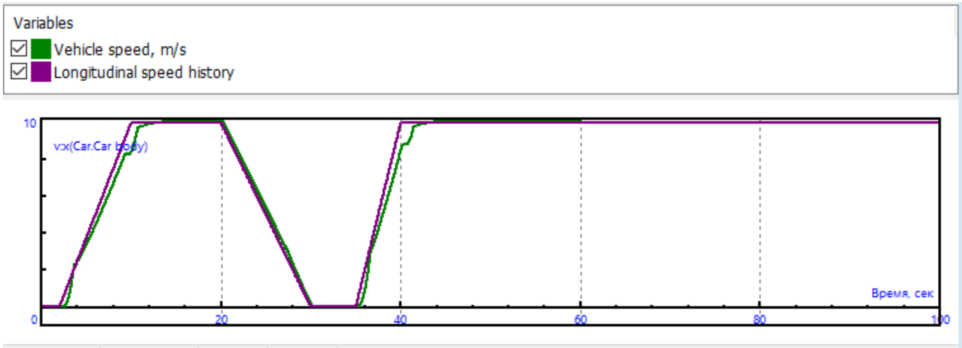
Brake pedal position

Figure 1.141. Simulation results for motion with controlled speed on a road with ascending and descending sections

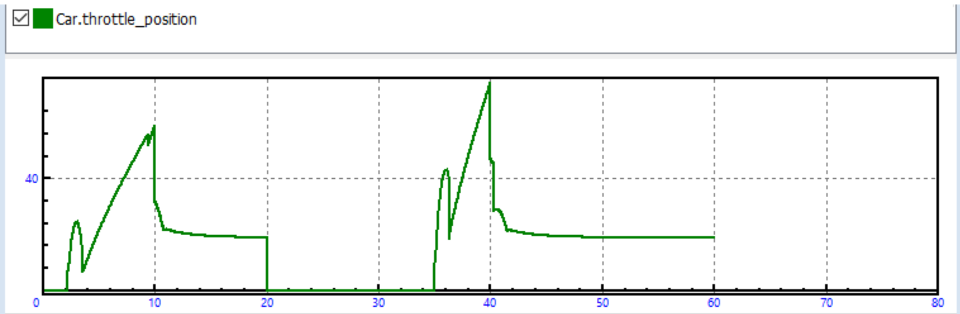
Example 2. Motion forward at variable speed

Open the model {Data UM}\SAMPLES\Automotive\Vaz2109 T in the UM Simulation program and load the full model configuration **Test with stop and run**. Simulation of motion on a horizontal section of the road with a variable speed is performed. The car accelerates to 10 m / s, slows down to a complete stop, and then accelerates again. The simulation results are shown in Figure 1.142.

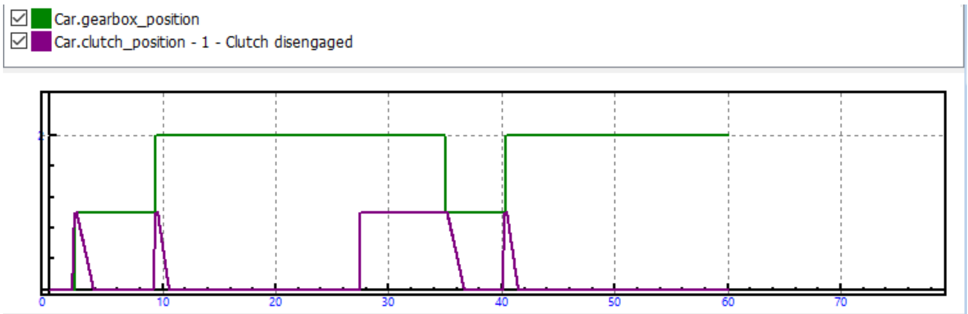
Before and during a stop, the control system disengages the clutch, leaving the second gear.



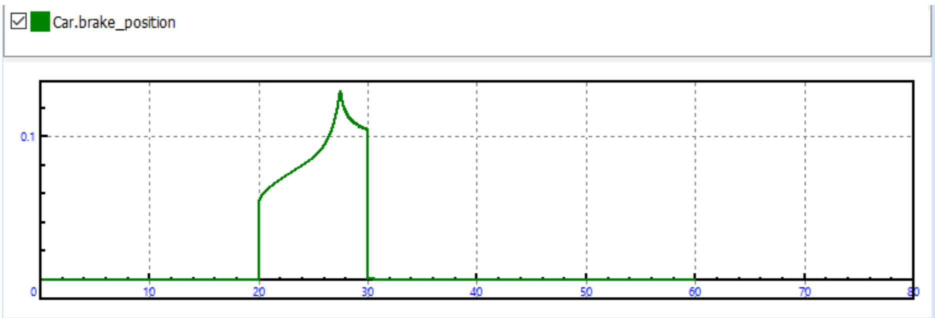
Simulated and desired speeds



Accelerator pedal position



Gearbox and clutch pedal positions

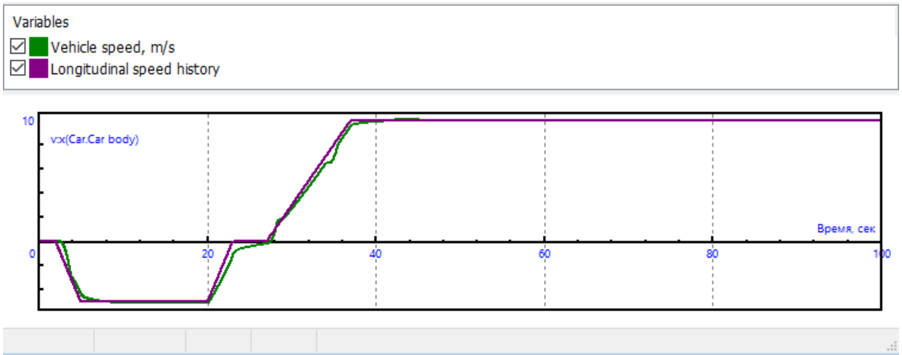


Brake position

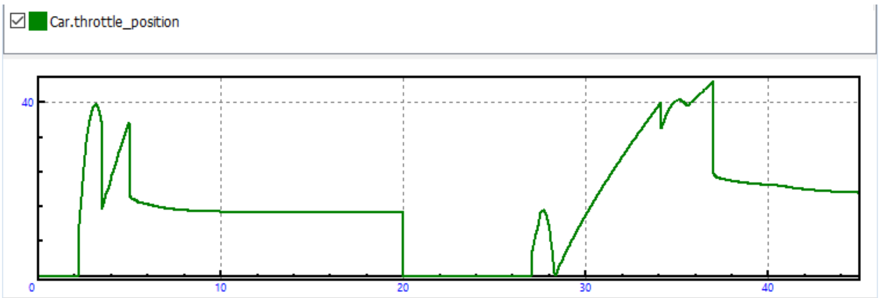
Figure 1.142. Results of simulation of motion with variable speed and intermediate stop

Example 3. Reverse and forward motion in curve

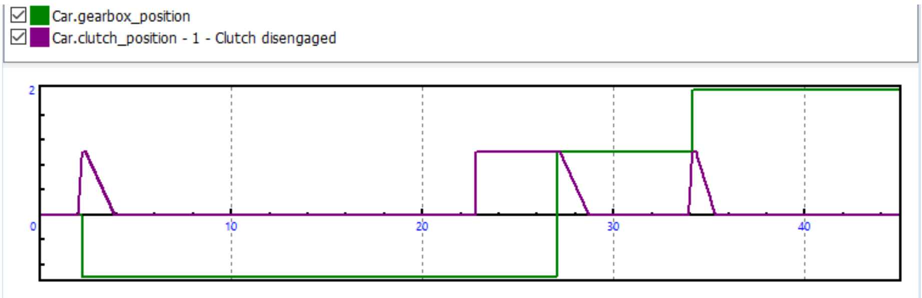
Open the model {Data UM}\SAMPLES\Automotive\Vaz2109 T in the UM Simulation program and load the full model configuration **Reverse motion in curve**. The simulation of motion in a curve along a horizontal section of the road with a variable speed is performed. The car moves backward first, then forward. The simulation results are shown in Figure 1.143.



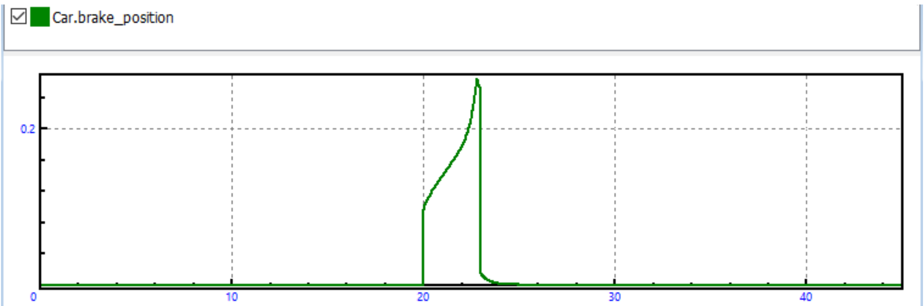
Simulated and desired speeds



Accelerator pedal position



Gearbox and clutch pedal positions



Brake position

Figure 1.143. Simulation results for reverse and forward motion in curve

1.9.4. Tests as the main tools for simulation of vehicle dynamics

1.9.4.1. General information

A set of tests realized in UM is a basis for dynamic analysis of a vehicle. Currently the following test types are available, Figure 1.144:

- Equilibrium test, Sect. 1.9.4.4
- Steering wheel rotation, Sect. 1.9.4.5
- Open loop steering, Sect. 1.9.4.6
- Test with driver, Sect. 1.9.4.7
- Car simulator, Sect. 1.9.4.8
- Vertical harmonic loading, Sect. 1.9.4.9
- Horizontal harmonic loading, Sect. 1.9.4.10

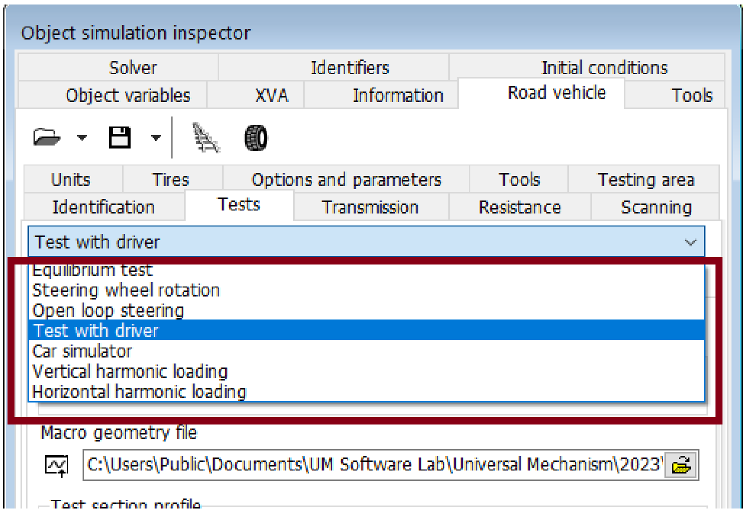


Figure 1.144. Choice of a test

a) Equilibrium test

Usually this is the first test to bring the new model into the equilibrium state and to store the corresponding initial values of coordinates. The test is also important for evaluations of static forces.

In version UM10, a tool for performing various types of calculations with a unmoving car has been added to the equilibrium test, for example, computation of amplitude-frequency characteristics using the gliding frequency method.

b) Steering wheel rotation

Test for evaluation of steering ratio and dependence of the steer angle on the steering wheel angle.

c) Open loop steering

Simulation of maneuvers with an open loop control.

d) Lateral driver test

Simulation of maneuvers with a closed loop control using a driver model.

e) Car simulator

The test can be used both for models with a description of the car's transmission, and without it. The test allows the user to control interactively the motion of a model using a special window containing control elements: steering, accelerator pedal, clutch, gear, brake.

f) Vertical harmonic loading

Quasistatic loading with a harmonic vertical force applied to the car body center of mass.

g) Horizontal harmonic loading

Quasistatic loading with a harmonic lateral force applied to the car body center of mass.

The test can be divided into two groups: tests with locked rotation of wheels (a, b, f, g) and test with vehicle longitudinal motion and steering control (c, d, e).

Tests from the first group have the following features.

- Nonzero values of the movement locking parameters are requited (Sect. 1.7.9. “*Locking wheel rotation*”, 1.9.1.2.2. *"Identification of wheel rotation locking parameters"*);
- Simplified models of tire as plane-circle contact elements are used (Chapter 2, *Force elements/ Contact forces*); the vertical stiffness and damping constants are equal to those for the tire model;
- Irregularities and macro-geometry are ignored.

Tests from the second group

- require zero values of movement locking parameters (made automatically);
- require macro-geometry files;
- optionally may use longitudinal velocity functions (Sect. 1.9.1.3.1 “*Setting graphs for steering wheel angle and vehicle speed*”);
- optionally use road roughness files (irregularities);
- tire models are used.

1.9.4.2. Initialization of test parameters

Most of the tests should be initialized before their usage. In general some of the following parameters and data could be necessary for a test.

- Identifiers
- Numeric values
- Open loop steering functions
- Longitudinal velocity function
- Macro-geometry
- Irregularities
- Driver model and its parameters

1.9.4.3. Test variables

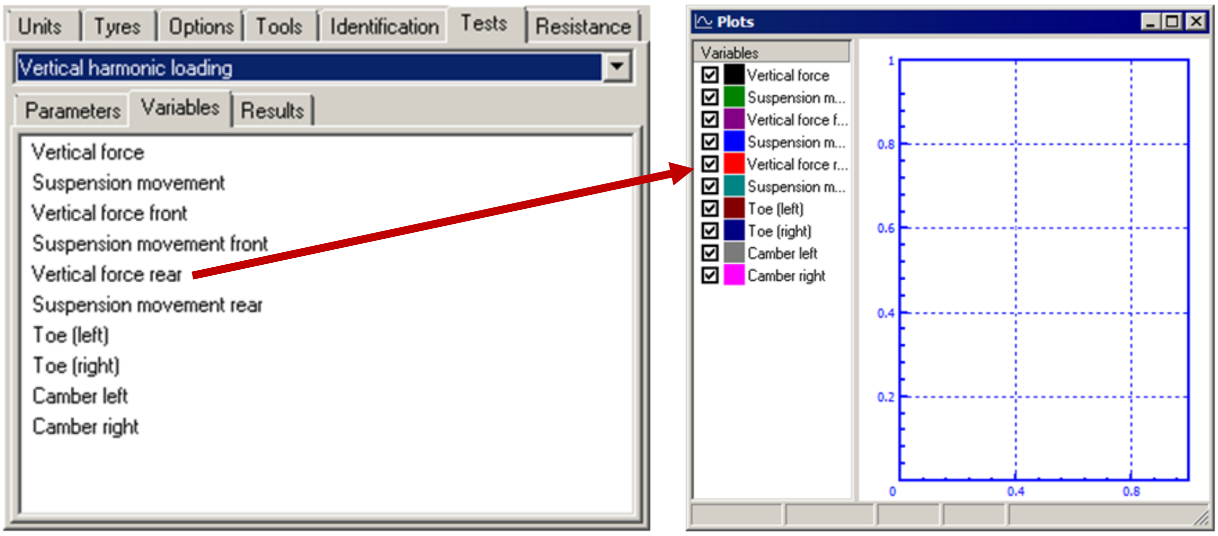


Figure 1.145. Dragging test variables in graphic window

Along with the typical way of creating variables in the Wizard of Variables (see Chapter 4 of the User's Manual for more details, section. “Wizard of variables”), a number of the most typical variables for each test are placed on the tab **Road vehicle | Tests | Variables**. To build graphs of variables, simply drag the desired variables with the mouse into the graphical window, Figure 1.145.

1.9.4.4. Equilibrium test

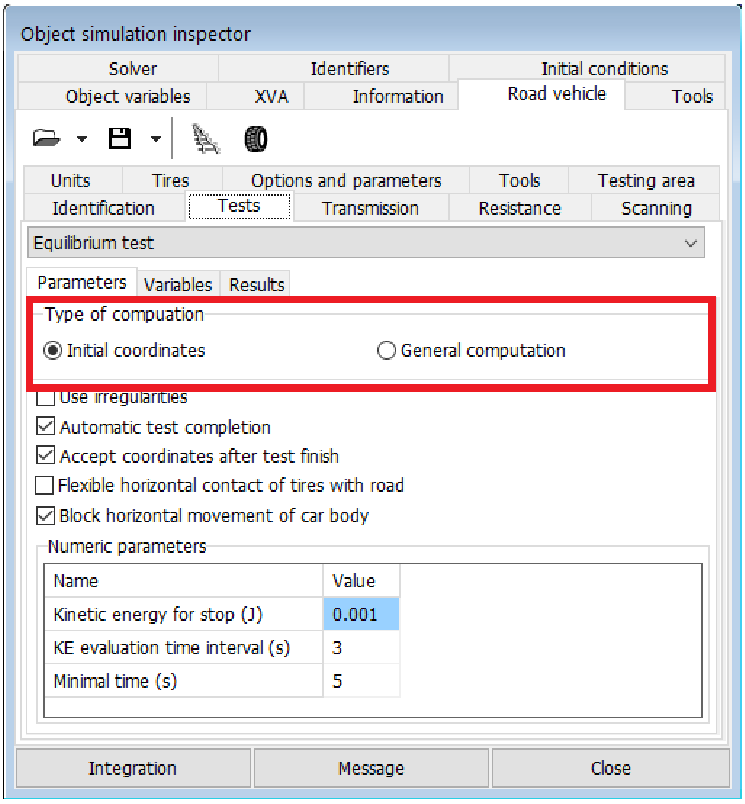


Figure 1.146. Modes of equilibrium test. Computation of initial coordinates is active

This test is performed with a parked vehicle, the rotation of the wheels is automatically blocked, Sect. 1.7.9 “*Locking wheel rotation*”, 1.9.1.2.2 “*Identification of wheel rotation locking parameters*”. To prevent horizontal displacement of the vehicle, two options are used, Figure 1.146:

- **Block horizontal movement of car body**
  When this function is enabled, the center of mass of the car body of each vehicle and trailer is blocked by a linear elastic-dissipative force element. Shift in two horizontal directions (stiffness constant 1×10^7 N/m) and rotation around the vertical axis (stiffness constant 1×10^6 Nm) are locked. The damping constant depends on the mass of the car body and provides the 30% damping ratio for the corresponding degrees of freedom. For example, the damping constant for the horizontal degrees of freedom is $0.6\sqrt{Km}$, where m is the mass of the body, $K=1\times10^7$ is the stiffness constant.
- **Flexible horizontal contact of tires with road**
  When the option is on, the contact points of the wheels are elastically connected to the supporting surface. For locking, the static stiffness coefficients in the longitudinal k_x and transverse k_y directions are used, which are specified in the tire model, see Sect. 1.5.2 “*FIALA tire model*”.

These two methods of blocking the horizontal movement are mutually exclusive. It is recommended to set only one of them, depending on the selected test mode, see the description of the modes below.

Test with a standing car has two modes Figure 1.146:

- **Mode for computation of initial coordinates**
- **General type simulation with unmoved vehicle**

1.9.4.4.1. Mode for computation of initial coordinates

This mode is usually the first and necessary calculation for each vehicle model. Here are the goals of the calculation:

- determine and save the initial values of the coordinates corresponding to the equilibrium position;
- calculate and save the static load and deflection for the wheels;
- determine the values of applied and reaction forces in the equilibrium position.

The test is performed for all vehicle loading options, as well as when changing suspension parameters and tire models that affect the calculated values in equilibrium, see Sect. 1.9.4.4.3. "*Example for computation of initial conditions*".

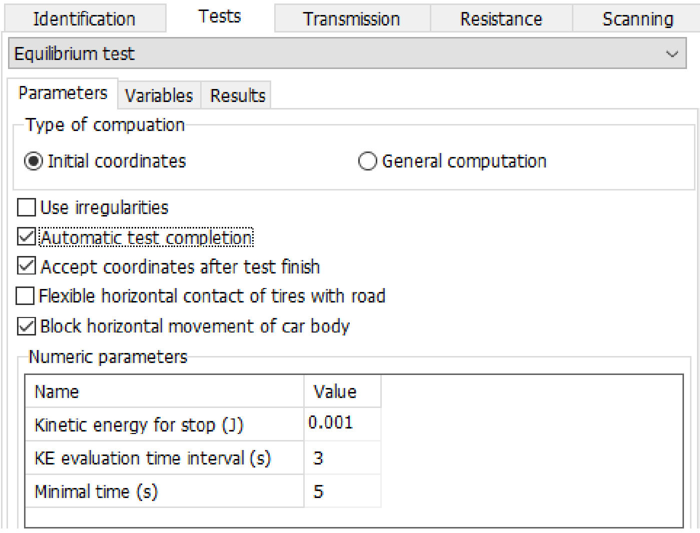


Figure 1.147. Parameters for computation of initial coordinates

The test is considered as successfully completed if the kinetic energy of the vehicle (KE) during the assessment interval is less than the specified threshold value, Figure 1.147. By default, the threshold value of the KE is 0.001 J, and the evaluation interval is 3 s.

The test is performed for all vehicle loading options, as well as when changing suspension parameters that affect the calculated values in equilibrium

Follow these steps to run the test.

1. Open the equilibrium test and set its parameters and options.

Use irregularities

Usually this option is disabled, since it is desirable to assign the initial position of the car on a flat horizontal surface with zero vertical coordinate. The option is enabled when the initial position of vehicle is shifted forward, so that the irregularities must be taken into account. In this case, the equilibrium will be computed taking into account the height of the irregularity under each of the wheels. The initial conditions found in this way may not be suitable for other tests, for example, with other irregularities or with a different initial position of the car.

Automatic test completion

If this option is enabled, the test will finished automatically at the moment when the program has determined that the test has been completed successfully. That is, when the kinetic energy (KE) is less than the threshold value at the specified time interval. The user can set the minimum test execution time, the default value is 5s, and the simulation will continue until this interval expires.

Accept coordinates after test finish

When this option is enabled, the program sets the calculated values of the model coordinates as the current ones in case of successful completion of the test. The accepted coordinate values are available on the **Initial Conditions** tab of the simulation inspector. It is recommended to save these values to a file.

It is recommended to disable the option **Flexible horizontal contact of tires with road** and enable the option **Block horizontal movement of car body**.

2. Click the Integration button to start the simulation process.

It is convenient to observe the process of calculating the equilibrium position by drawing a graph of kinetic energy. To do this, open a new graphical window before starting the calculation process and drag the corresponding variable into it with the mouse, Figure 1.148. An example of a graph is shown in Figure 1.149.

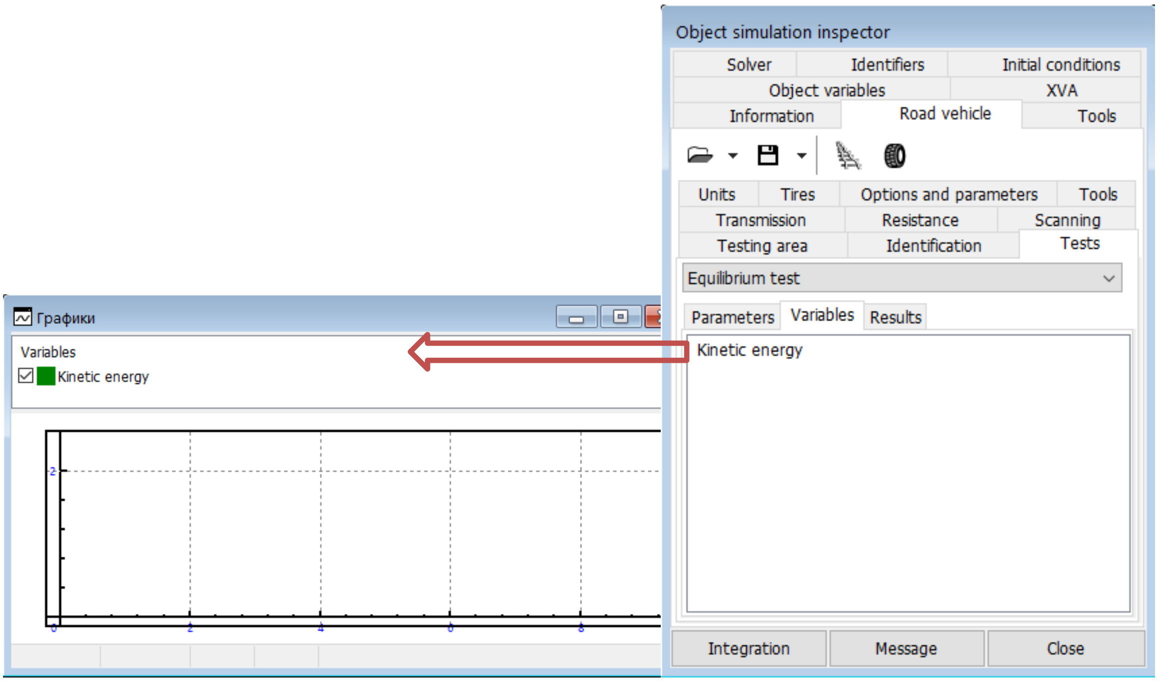


Figure 1.148. Test variable: kinetic energy of vehicle

3. When the automatic test completion mode is enabled, the simulation duration is controlled by the program, as described above. If the test is successful, the message "Simulation over" appears. If the automatic completion is disabled, the user can brake simulation any moment

to store the appropriate coordinates. The stop the simulation, either the **Esc** key or the button  in the process window are used, Figure 1.150.

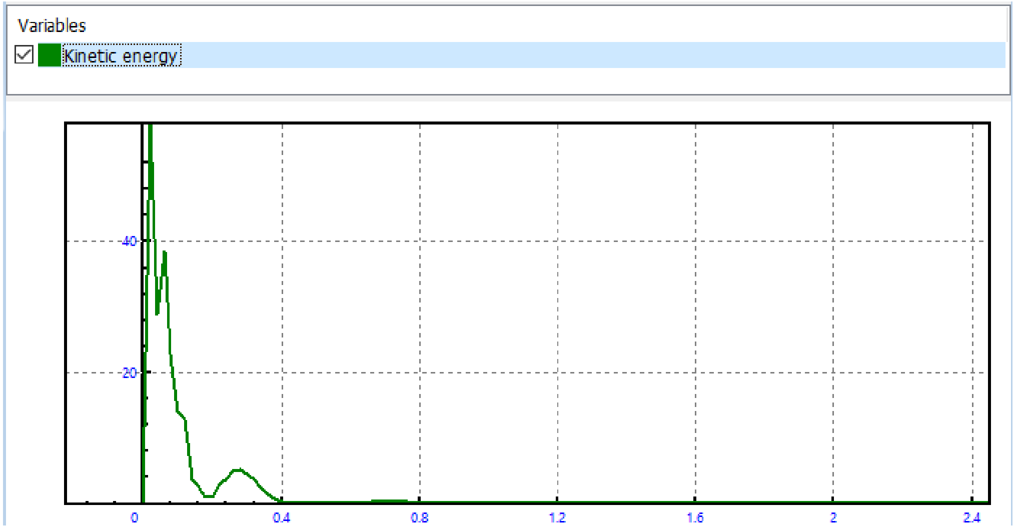


Figure 1.149. Example of kinetic energy plot while computation equilibrium position

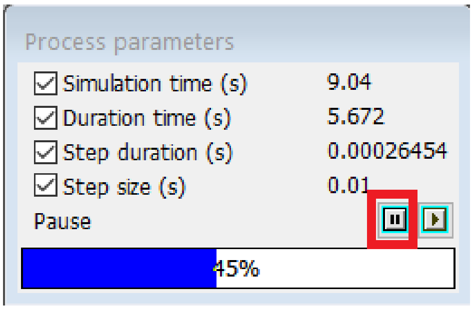


Figure 1.150. Pause button

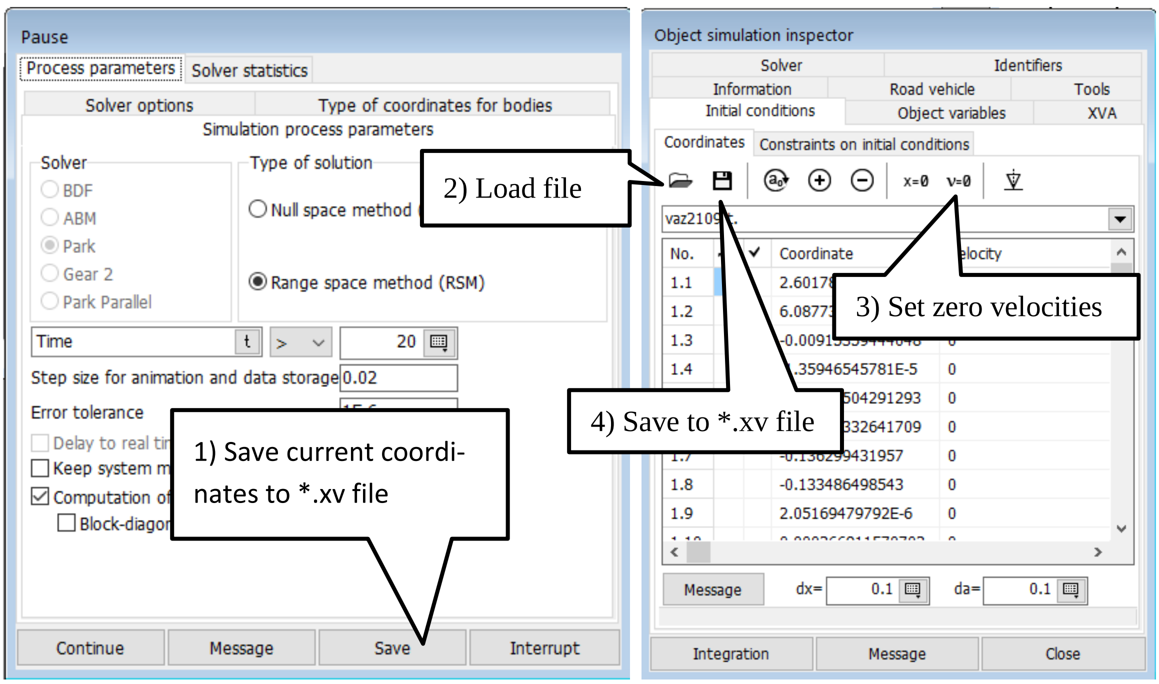


Figure 1.151. Equilibrium position as initial conditions

After the end of the numerical simulation, the **Pause** window appears. In this mode, the current coordinate values can be written to a *.xv file for further use, Figure 1.151, Step 1. This file can be read on the **Initial Conditions** tab, Fig. 12.153, Step 2. It is recommended to set zero values of velocities, Fig. 12.153, Step 3, and save the coordinate in the file again, step 4.

However, if the **Accept coordinates after test finish** is enabled, then the initial conditions will be assigned automatically upon successful completion of the test.

4. Click on the **Interrupt** button in the **Pause** window to finish the test, Fig. 12.153, left.

In case of successful completion of the test, you will be prompted to *accept the results as a standard*, Figure 1.152. We are talking about the static deflection of each tire and the loads on the tires in the equilibrium position. This data is required to perform dynamic tests with the vehicle model. The values calculated and accepted as a standard are displayed on the **Road vehicle | Tires** tab, Figure 1.153. These results are stored in the car model configuration file *.car.

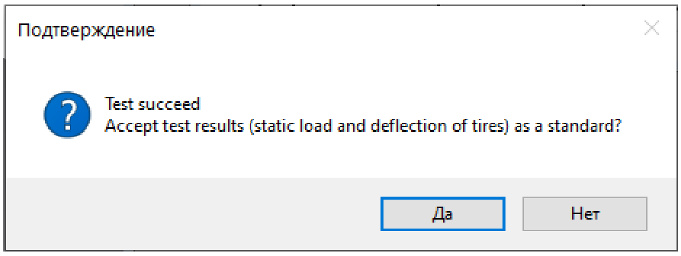


Figure 1.152. Confirmation of successful test completion

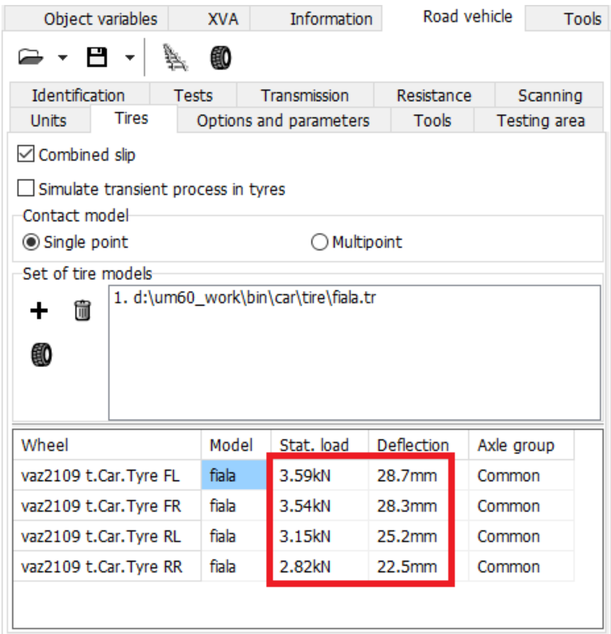


Figure 1.153. Accepted results of equilibrium test for tires

5. Open the **Initial Conditions** tab in the simulation inspector. If necessary, read the initial conditions file created in step 3, set velocities to zero, Figure 1.151, Steps 2,3. In any case, it is recommended to create a file with calculated equilibrium values of coordinates, Figure 1.151, Step 4.
6. Open the **Vehicle | Tests | Results** tab in the simulation inspector. The results of the equilibrium computation include the values of the static forces and deflections of the tires, Fig-

ure 1.154. If the calculation results are not accepted on request in Figure 1.152, the user can get the same action by clicking on the **Accept as standard** button.

These data will be used by the UM software to determine the initial speed of rotation of the wheels corresponding to the initial vehicle speed, and the static values of the wheel load takes into account in the MacAdam's driver model.

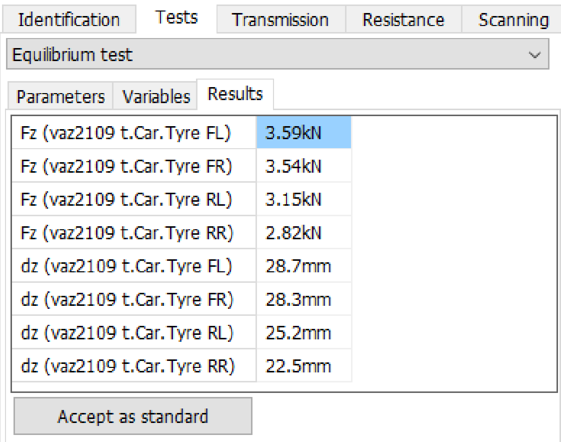


Figure 1.154. Computed static values for tires

The initial angular velocity of the wheels is calculated from the given vehicle speed v_0 and the radius of the wheel in the equilibrium position $r_w - dz$

$$\omega_0 = \frac{v_0}{r_w - dz}.$$

1.9.4.4.2. General type simulation with unmoved vehicle

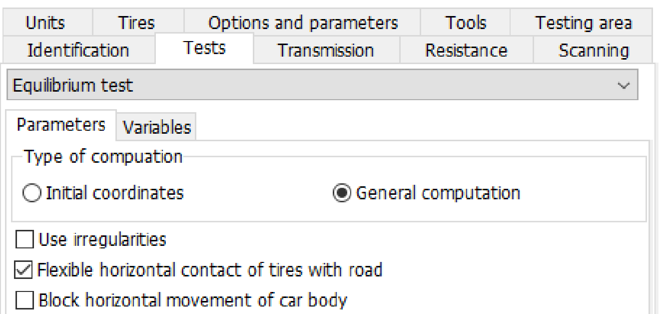


Figure 1.155. The mode of studying the properties of an unmoved vehicle

In this mode, the user can analyze the dynamic properties of an unmoved vehicle. Here is the list of some problems to be solved.

- Response to a force excitation, for example, to a harmonic or step excitation.
- Response to a kinematic excitation, such as a rotation of steering wheel, or toe and camber adjustments.

Before executing the test, it is required to bring the model to an equilibrium state, as described in the previous section.

This type of simulation requires enabling the option **Flexible horizontal contact of tires with road** and disabling the option **Block horizontal movement of car body**.

To perform such study, introduction of additional elements into the car model is usually required: force elements for excitation modeling, kinematic elements for simulation of kinematic excitation etc.

1.9.4.4.3. Example for computation of initial conditions

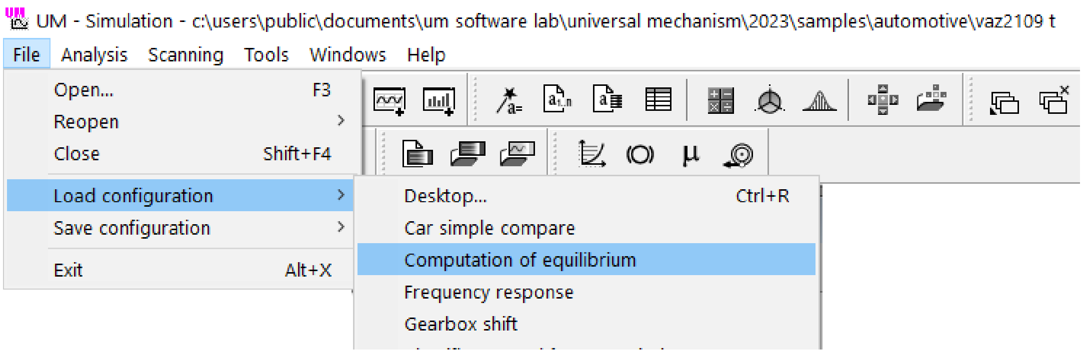
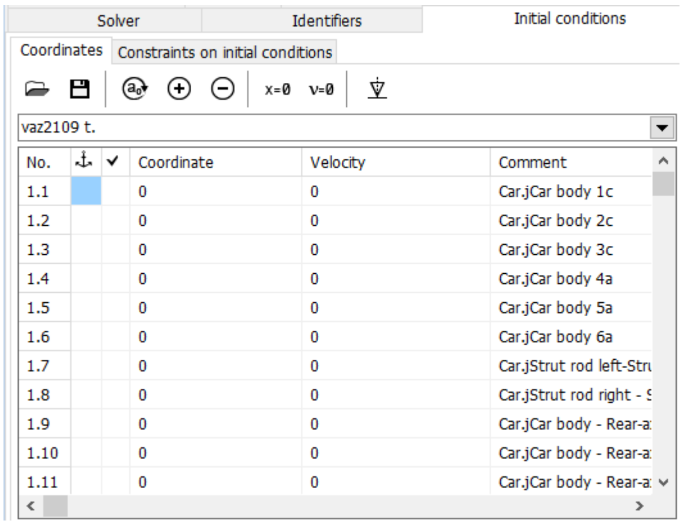
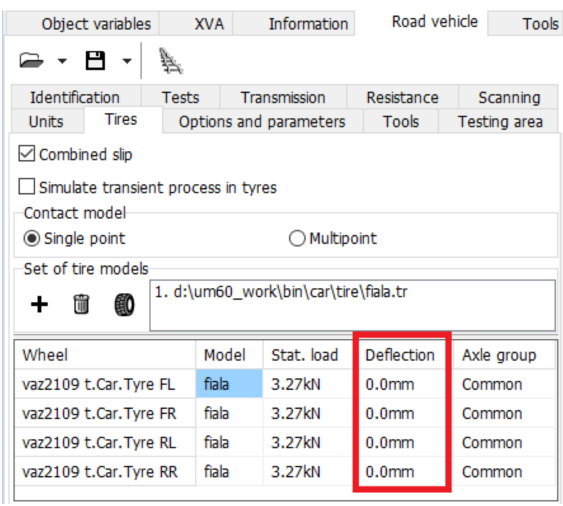
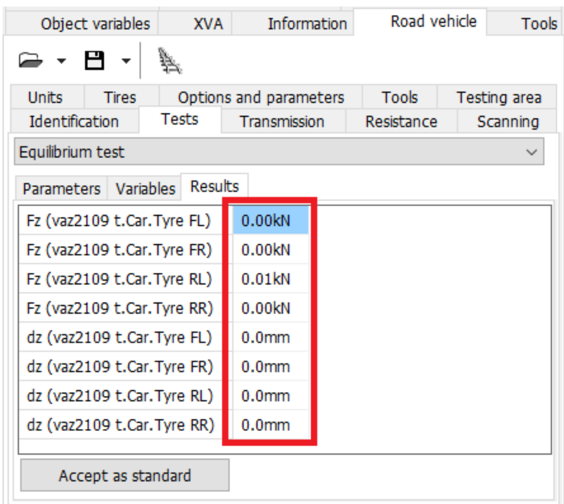


Figure 1.156. Loading full configuration



a)





b) c)

Figure 1.157. Values before test run: coordinates (a), static load and deflection of tires (b,c)

Open the model {Data UM}\SAMPLES\Automotive\Vaz2109 T. Load the full configuration **Computation of equilibrium**, Figure 1.156. Make sure that the model coordinates and tire deflections take zero values in this configuration, Figure 1.157. The test parameters correspond to the standard values in Figure 1.147.

Run simulation by the **Integration** button on the inspector.

After the message “Simulation over”, the program goes into pause mode. The decrease in the kinetic energy of the car is shown in Figure 1.158.

Exit the simulation by clicking the **Interrupt** button. Confirm the acceptance of the results as a standard, Figure 1.152.

Test completed successfully. Some results are shown in Figure 1.159/

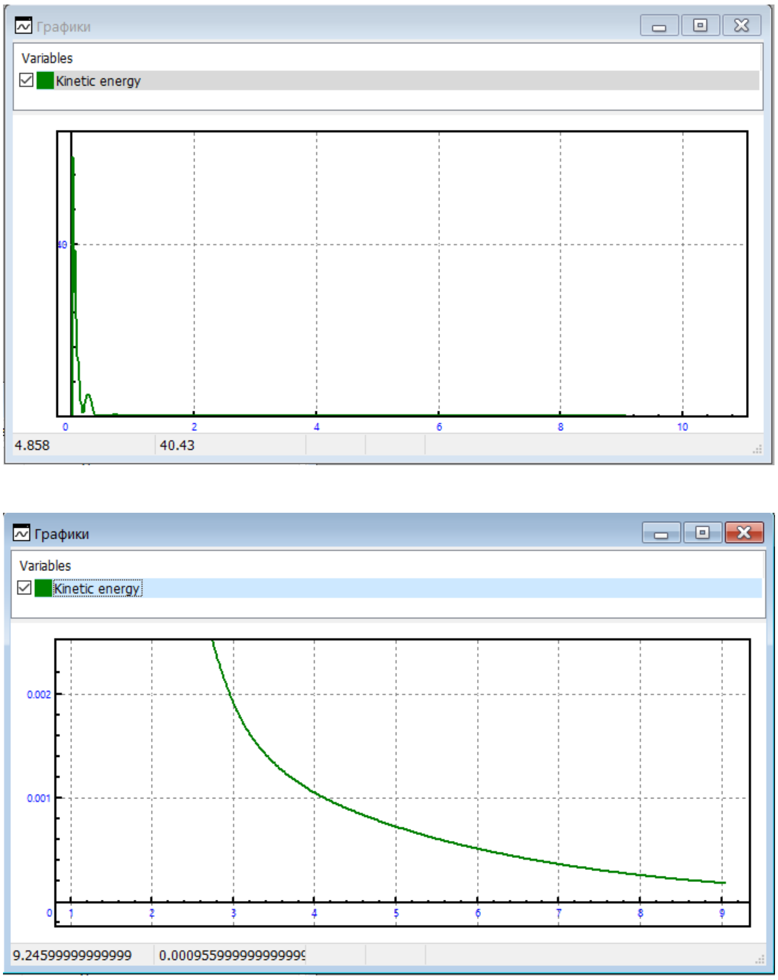
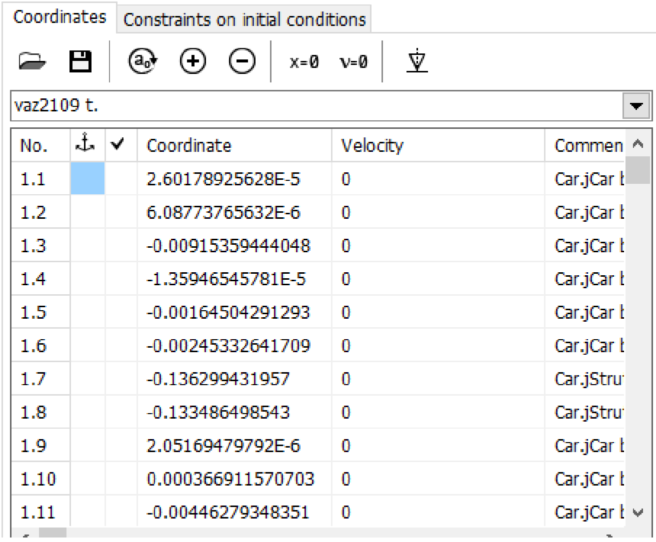
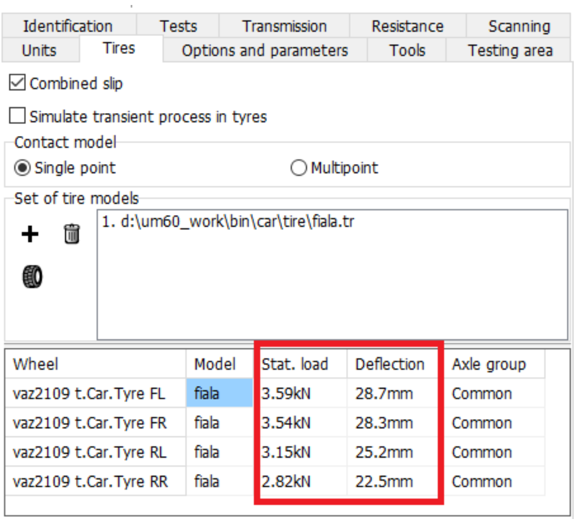
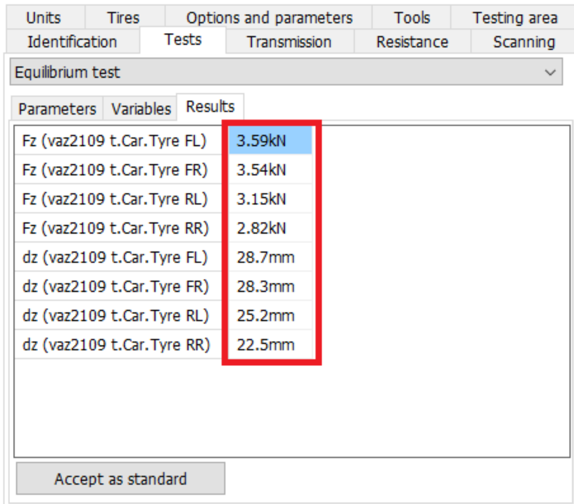


Figure 1.158. Kinetic energy vs time



a)





b) c)

Figure 1.159. Test results: coordinates (a), static loads and deflections (b, c)

1.9.4.4.4. Frequency response of a passenger car

Consider an example of computation of a frequency response for an unmoved passenger car. Using the equilibrium test, oscillations are calculated taking into account the non-linearity of forces, that is, the oscillations are non-linear. To compute the response, the gliding frequency method is used.

Additional element in the vehicle model

First, consider which elements are created in the model to perform this test. Load the model {Data UM}\SAMPLES\Automotive\Vaz2109 T in UM Input program.

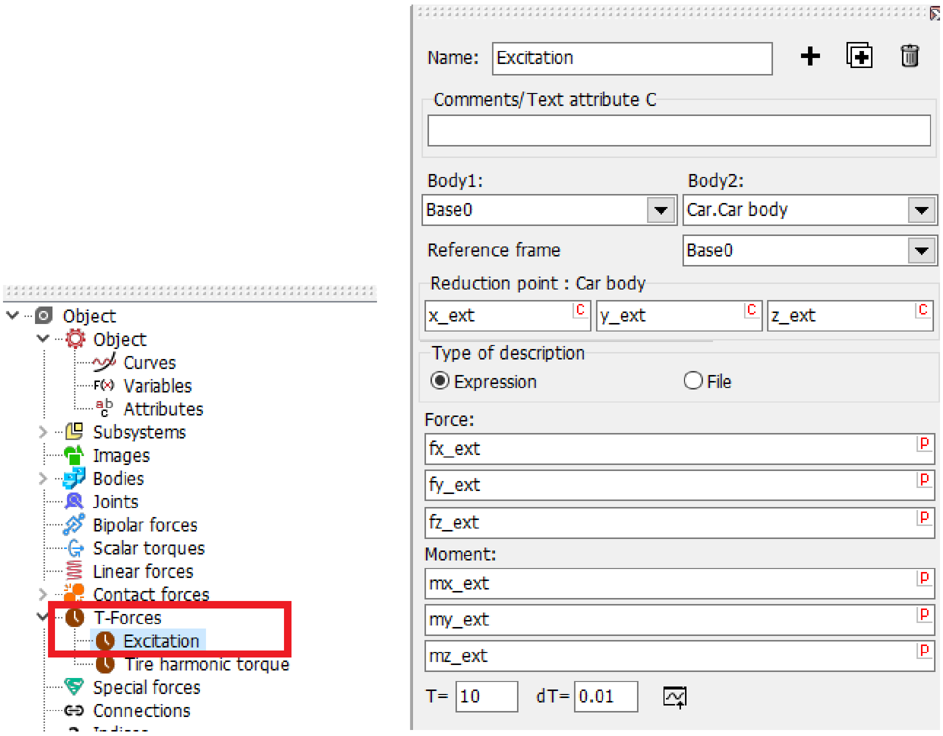


Figure 1.160. Excitation as a T-force

A T-force *Excitation* has been introduced, which will be used to model the excitation force and moment. The force is applied to the car body. Coordinates of the force attachment point, force and moment component are parameterized by identifiers. Several auxiliary identifiers are added as well

ext_ampl – excitation amplitude;

e_ext – rate of the excitation frequency growth Hz/s;

Freq0 – start value of excitation frequency, Hz.

Note that in this car model, the origin of the body-fixed system of coordinates is located in the center of mass of the car body. Thus, when the coordinates of point of force application are zeroes, the excitation force is applied to the center of mass of the car body.

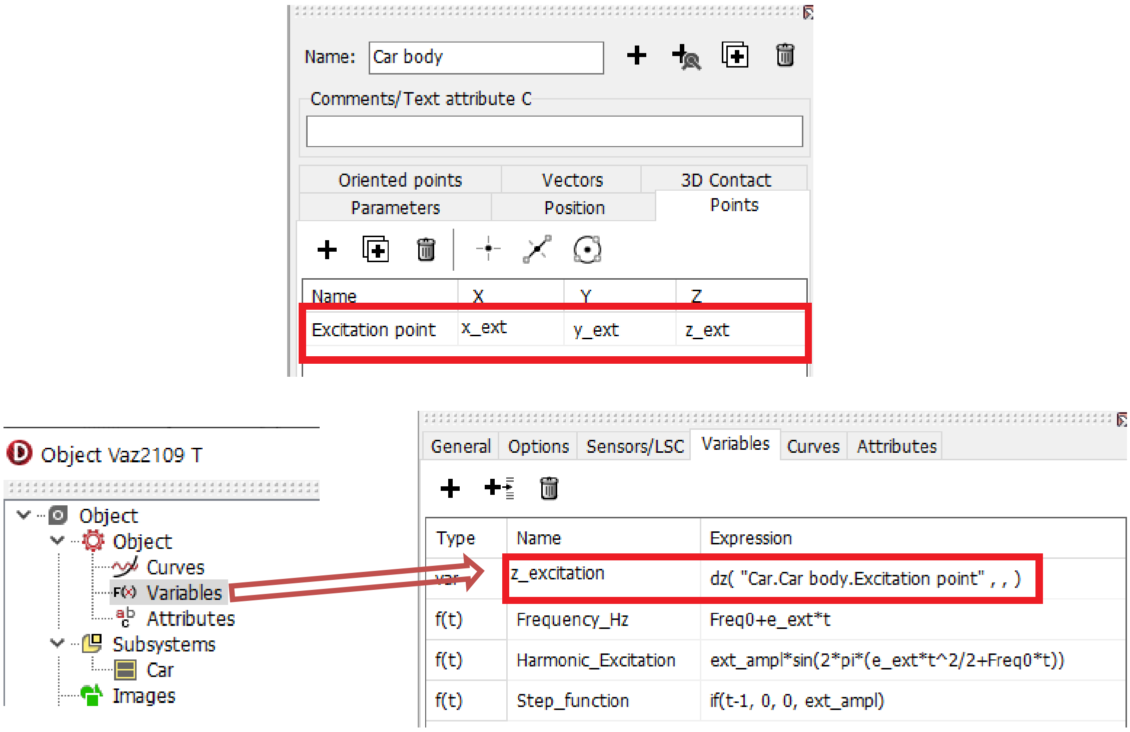


Figure 1.161. Parameterized connection point for car body and variable, which is the vertical coordinate of this point

We will draw a plot of the vertical displacement of the point to which the excitation force is applied. To do this, a parameterized *Excitation point* was added to the connection points of the car body (Figure 1.161, top) and a kinematic variable *z_excitation* was created equal to the vertical coordinate of this point relative to CK0 (Figure 1.161, bottom).

Building frequency response in UM Simulation

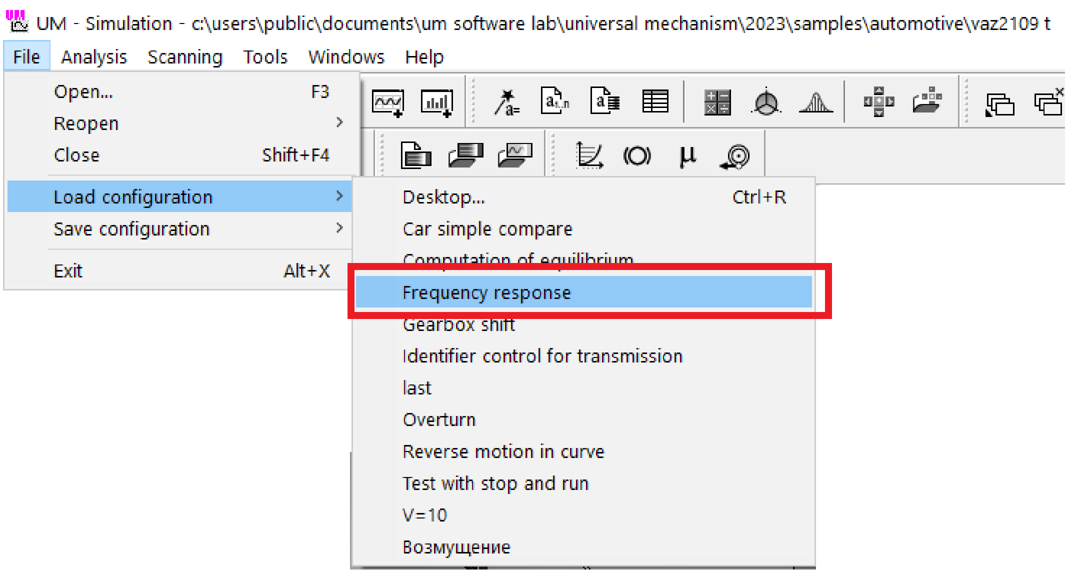


Figure 1.162. Loading the full configuration “Frequency response”

Open the model {Data UM}\SAMPLES\Automotive\Vaz2109 T in UM Simulation program. Load the model full configuration **Frequency response**, Figure 1.162. The model has already been brought into equilibrium. In this configuration, the general calculation mode is set in the equilibrium test, Figure 1.155.

1) Assignment of a time function to the force identifier

In the loaded configuration, a harmonic excitation with a gliding frequency is assigned to the vertical force component specified by the identifier *fz_ext*

fx_ext=ext_ampl*sin(2*pi*(Freq0+ e_ext*t/2)*t).

The **Identifier control** tool is used for this assignment. A variable **Excitation (t)** was created in the Wizard of Variables. The variable is the function of time, corresponding to the right side of this expression. Let us show how this variable can be opened in the wizard for viewing and possible modification, Figure 1.163.

- Open the tab **Identifiers | Identifier control** in the simulation inspector.
- Double click by the *left mouse button* on the line with the identifier **fz_ext** to open the window for identifier control description.
- Click by the *right mouse button* on the box with the name of variable **Excitation (t)** to call the popup menu.
- Select the command **Open ordinate in the Wizard of variables**.

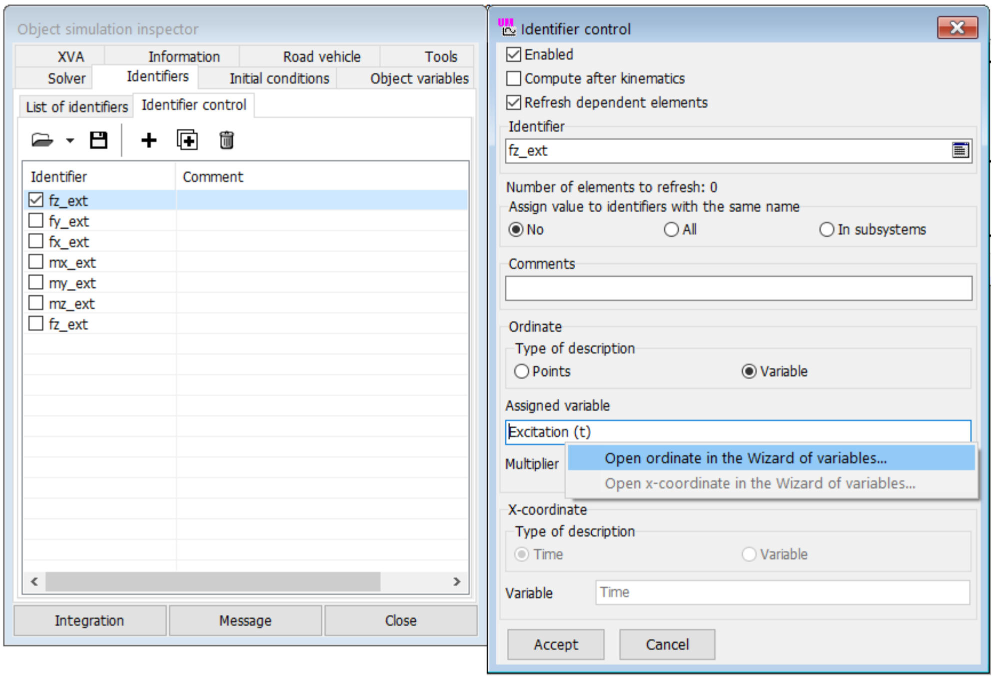
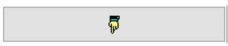


Figure 1.163. Steps to open a variable in the wizard

Let us explain how this variable was created, Figure 1.164. First, the variable identifiers *e_ext*, *ext_ampl*, *Freq0* are placed in the container on the **Identifiers** tab of the wizard, (by the way, now, at the stage of viewing the variable, and not creating it, these variables can be sent to the container using the button  on the left side of the **Expression** tab, Figure 1.164). Then the desired function is programmed line by line. The created variable should be dragged with the mouse into the corresponding box of the **Identifier control** window, Figure 1.163.

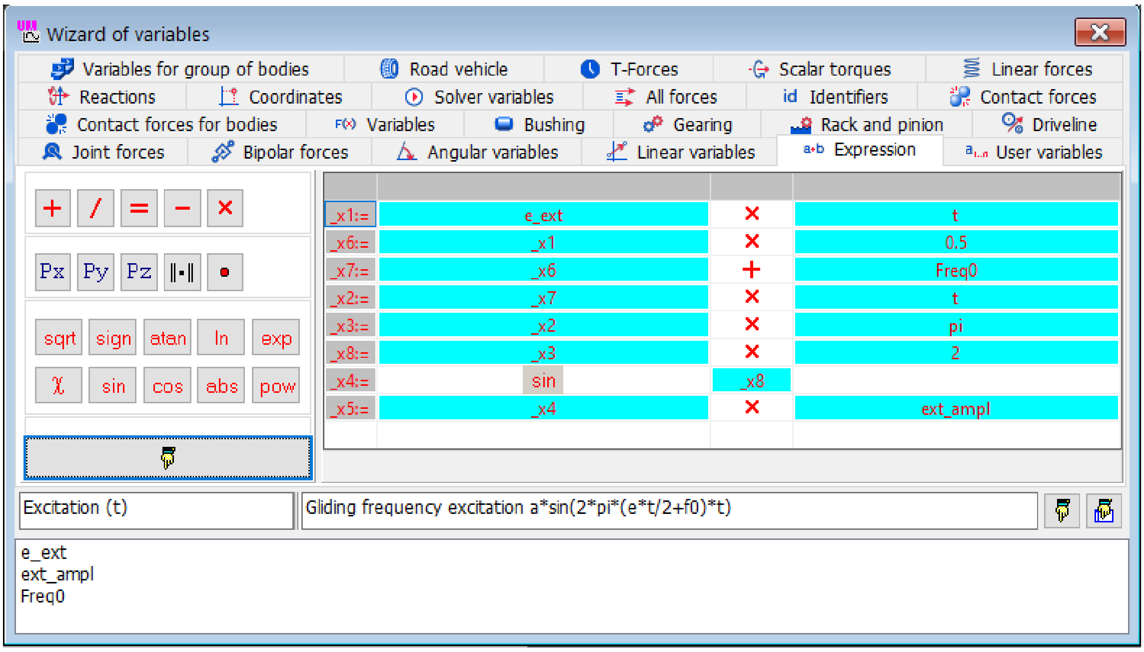


Figure 1.164. Variable Excitation (t) in the wizard of variables

2) Assignment of coordinates of the force application point

Coordinates (1, 0.5, 0) are sect to the force application point in the body-fixed SC, Figure 1.165, Figure 1.166.

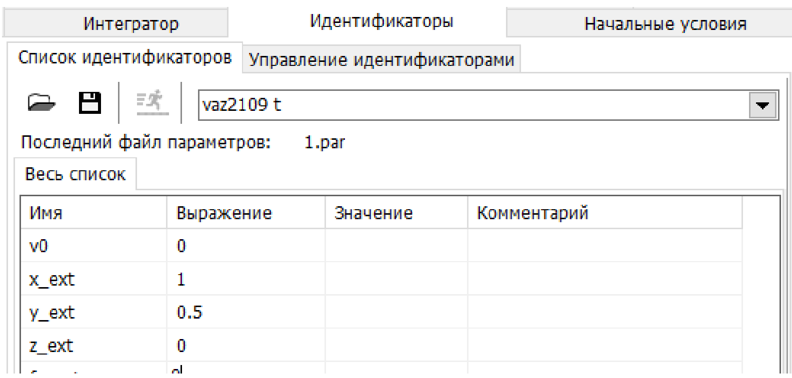


Figure 1.165. Identifier values

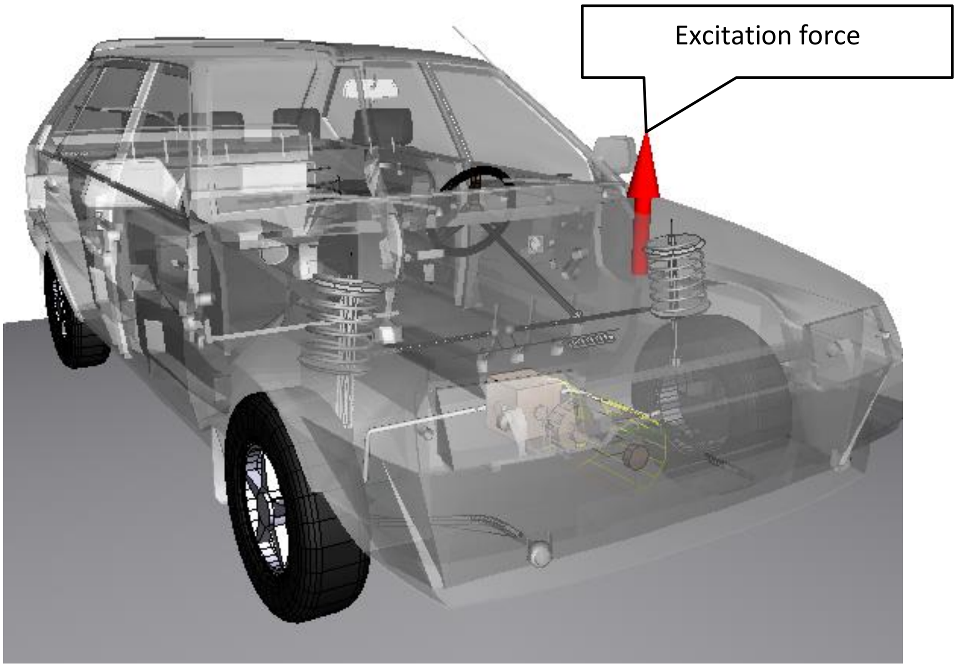


Figure 1.166. Excited oscillations of a car

3) Creating variables for plotting

The following variables are presented on the frequency response plots in the example:

dz_excitation is the vertical displacement of the application point of the excitation force; the displacement in the difference between the Z coordinate of this point (the variable *z_excitation* created in the UM Input program, see Figure 1.161) and the value of this co-ordinate in equilibrium position specified by the identifier *z*0, Figure 1.167, top;

om:x(Car.Car body) is the angular velocity of the car body about X axis;

om:y(Car.Car body) - is the angular velocity of the car body about Y axis.

Plots of variables are drawn in dependence on the excitation frequency in Hz given by the variable **Frequency (Hz)**, e_ext*t+Freq0, Figure 1.167, bottom.

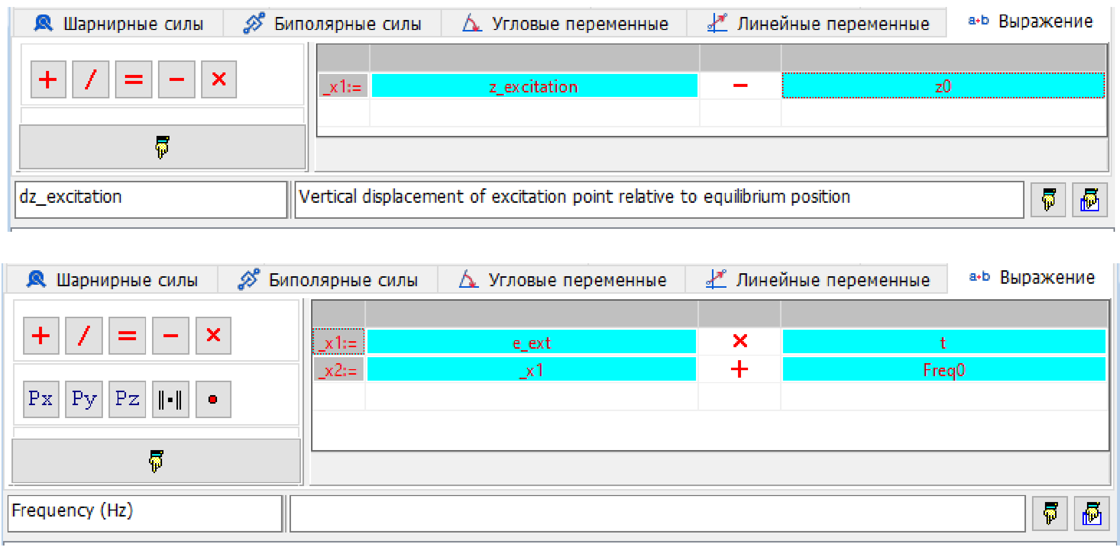


Figure 1.167. Vertical displacement of the force application point (top) and dependence of the excitation frequency in Hz on time

4) Computation of the stationary value of vertical coordinates of force application point

For the correct calculation of the *dz_excitation* variable, it is required to assign to the identifier z0 the value of the coordinate of the force application point in the equilibrium position. To do this, perform an auxiliary calculation with *ext_ampl*=0, z0=0. The value of the variable obtained by integration on the chart dz_excitation = 0.682 is assigned to the identifier z0.

5) Assignment of identifier values for excitation

e_ext	0.02		
ext_ampl	500		
Freq0	0		
z0	0.682		Static Z coordinate of excitation point in SC0

In our example, the frequnce rate identifier *e_ext* is 0.02 Hz/s, and the excitation force amplitude *ext_ampl* =500 N.

6) Computation of frequency response

Run the integration process with the given values of identifiers. The dependencies of the variables on the excitation frequency are shown in Figure 1.168. Please note that the variable *Frequency (Hz)* is set in plots as abscissa (x-axis), and the simulation time was 300 s.

The amplitude characteristics correspond to the envelopes of the obtained curves.

In the pause mode, copy the calculated variables as static for the following compare with the linear analysis.

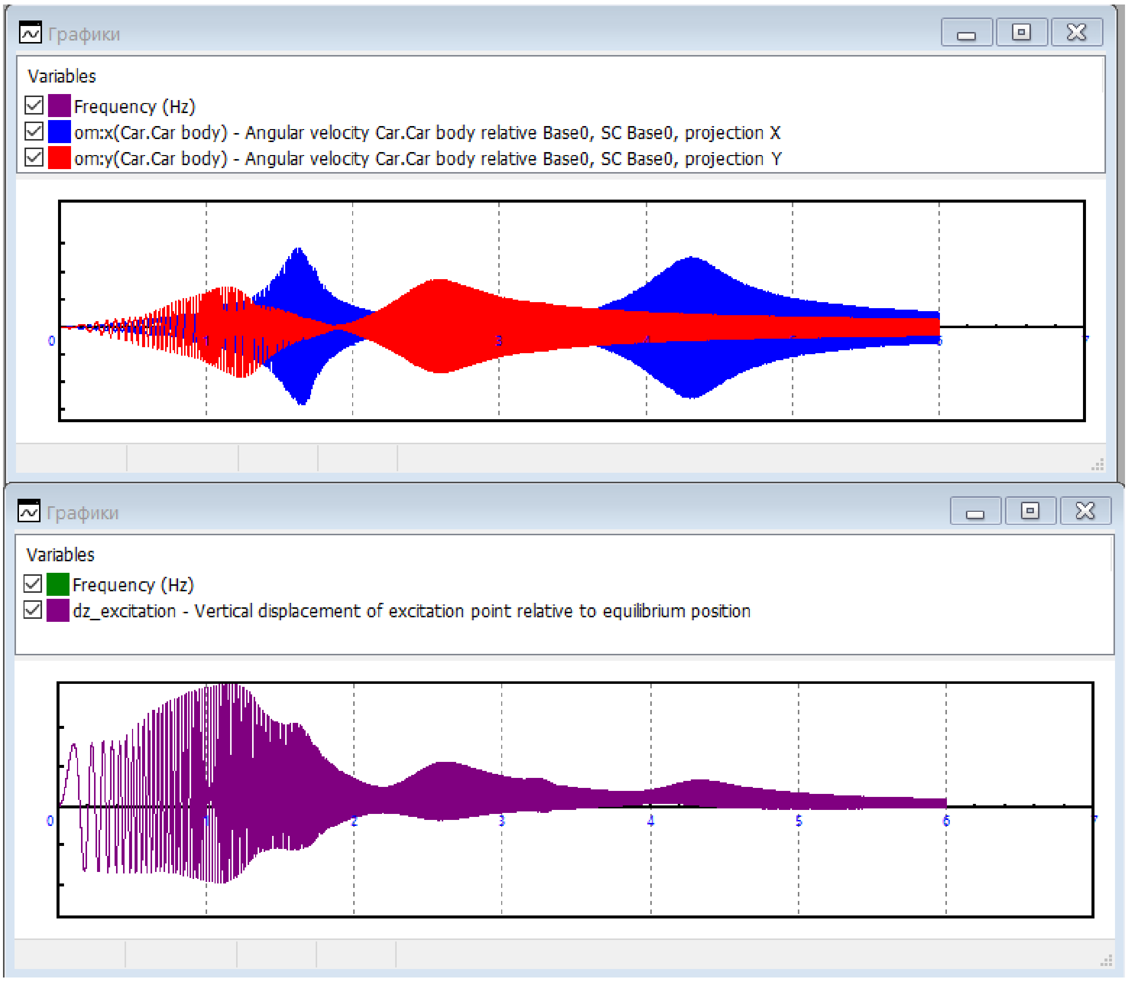


Figure 1.168. Frequency response results

7) Comparison with linear analysis

Close the **Simulation inspector** and open the **Static and linear analysis window** using the main menu command **Analysis | Static and linear analysis**. The analysis settings must contain the enabled locking of wheel rotation and wheel contact with the road, Figure 1.169, left.

Calculate the natural frequencies of the model on the **Frequencies/Eigenvalues** tab, Figure 1.169, right. Some of the lower frequencies is present on the frequency response plots in Figure 1.168.

Open the **Linear vibrations** tab. The vibrational analysis settings correspond exactly to the above numerical simulation, Figure 1.170. Perform the calculation by clicking on the button and drag the computed variables to the corresponding graphical windows with simulation results. Comparison of the frequency response obtained by nonlinear (integration of nonlinear equations) and linear analysis is shown in Figure 1.171. The results of the linear vibration analysis are drawn by markers. The significant difference is mainly due to the strongly nonlinear damping characteristic of shock absorbers, Figure 1.172.

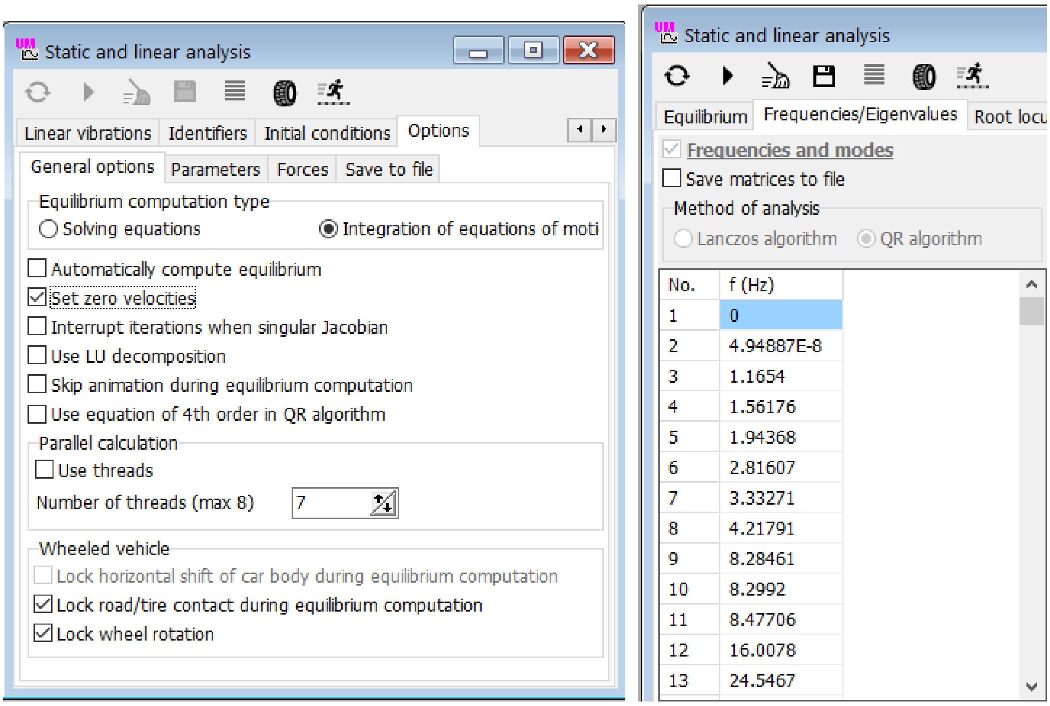


Figure 1.169. Settings for linear analysis and natural frequencies of the model vaz2109 T

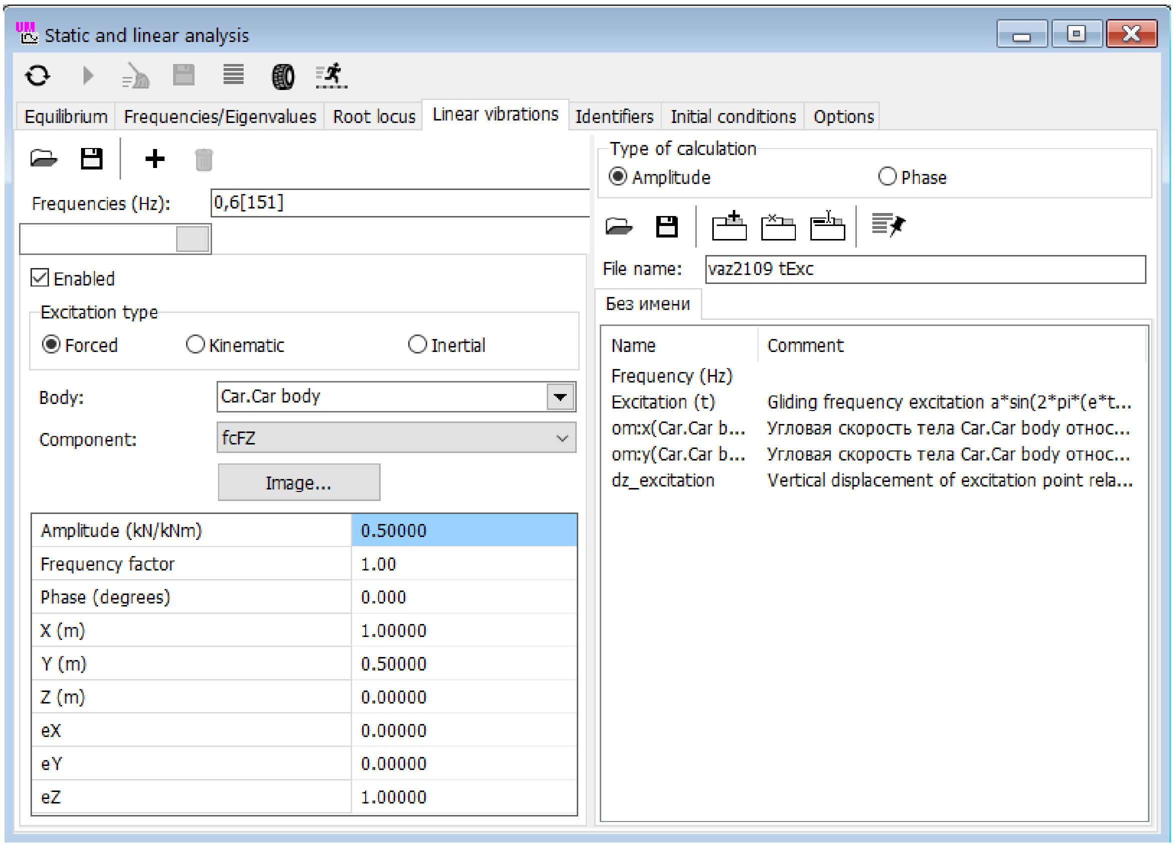
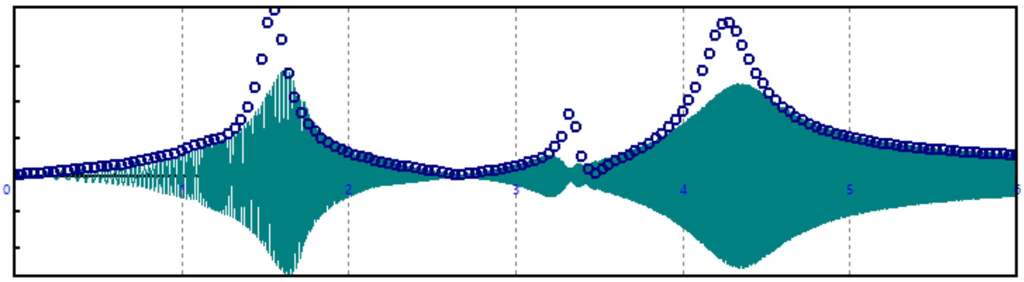
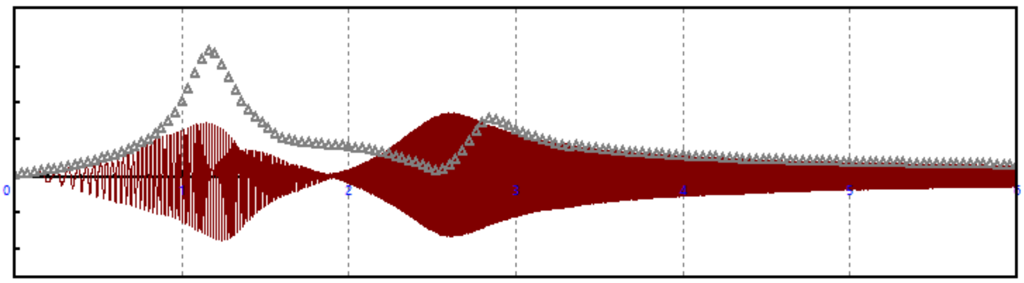


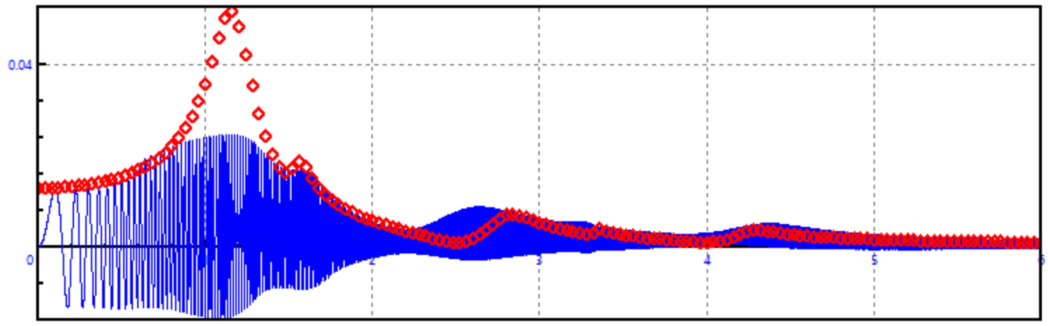
Figure 1.170. Settings for linear vibration analysis



Variable om:x(Car.Car body)



Variable om:y(Car.Car body)



Variable dz_excitation

Figure 1.171. Comparison nonlinear frequency response with linear one

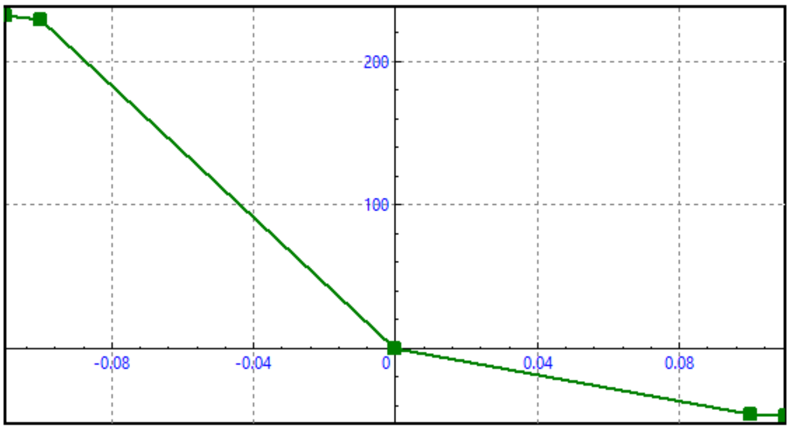


Figure 1.172. Damping characteristics of shock absorbers in the model Vaz2109 T

8) Additional and alternative features

- **Harmonic oscillation with a constant frequency**

If the growth rate of the frequency *e_ext* equals zero, one gets a usual harmonic excitation with a frequency specified by the *Freq*0 identifier. For example, at *Freq*0=2 we have a harmonic excitation with a frequency of 2Hz. Thus, the user can get the same results as in the **Vertical harmonic loading** test by setting the desired point of application of the force. Similarly, the user can form the conditions of the **Horizontal harmonic loading** test by assigning the excitation to the identifier of the lateral force *fy_ext*.

- **Assignment of excitation to another force or moment components**

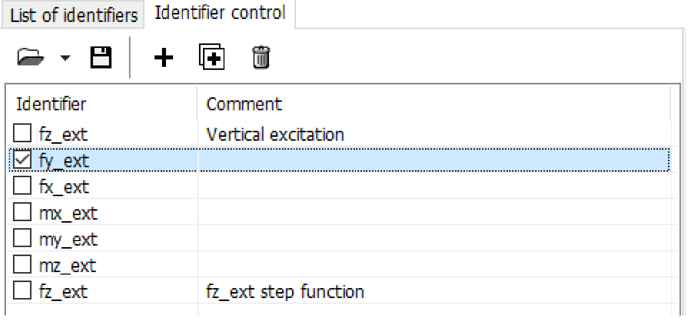


Figure 1.173. Control of identifier fy_ext is activated

If we assign the same time function to other identifiers corresponding to the components of the excitation force and moment, then we can simulate the excitation from the desired projection, disabling the other. For example, Figure 1.173 shows the case when excitations are created for all six components, but only the lateral force is enabled.

- **Other types of excitation**

Along with the harmonic oscillations, other types of excitations can be realized, for example, a step function response, Figure 1.174, Figure 1.175.

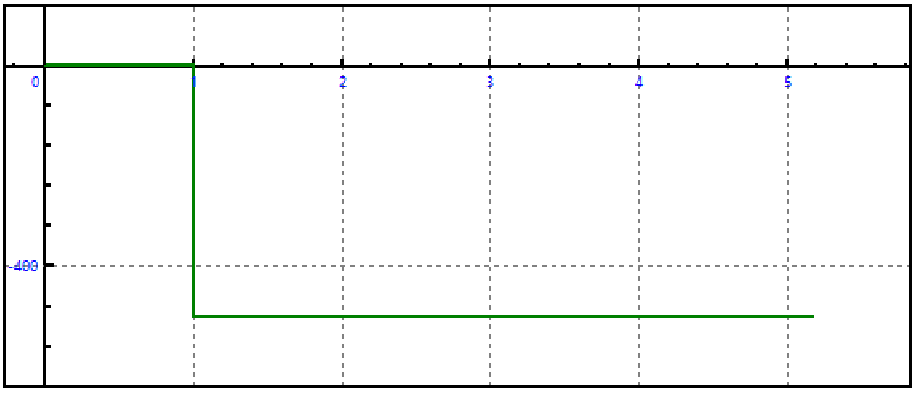


Figure 1.174. Step function

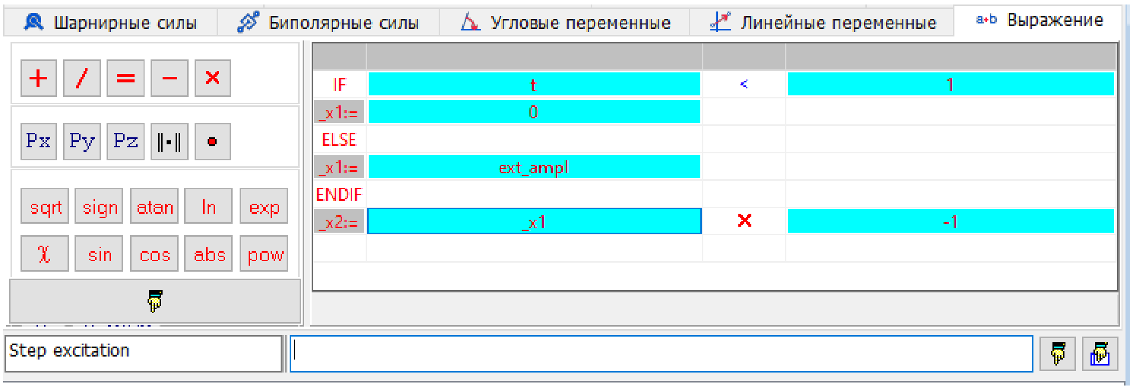


Figure 1.175. Variable implementing the step function

- **Alternative methods for setting excitation functions**

Above in this section, we only considered the method of setting time functions using the wizard of variables (Figure 1.164, Figure 1.167, Figure 1.175). There are two simple ways to describe these functions in the UIM Input program.

The **first way** is to directly set the force components as functions of time, Figure 1.176.

The **second way** is to create variables as time functions in the UM Input program, Figure 1.177. To access these variables in the UM Simulation, use the wizard of variables as it is shown in Figure 1.178. The variable is assigned to an identifier in the standard way on the **Identifier control** tab.

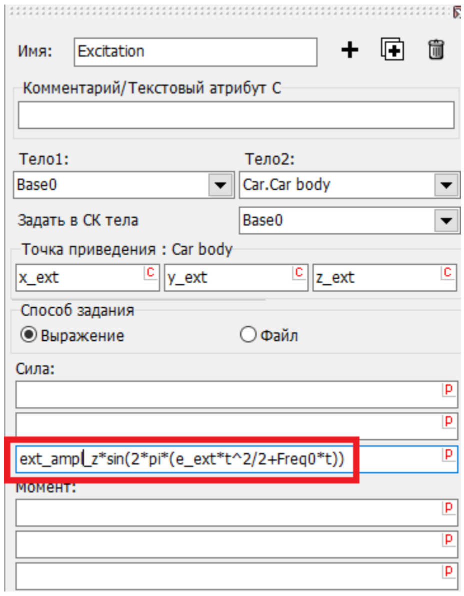


Figure 1.176. Direct description of time functions in force declaration in UM Input

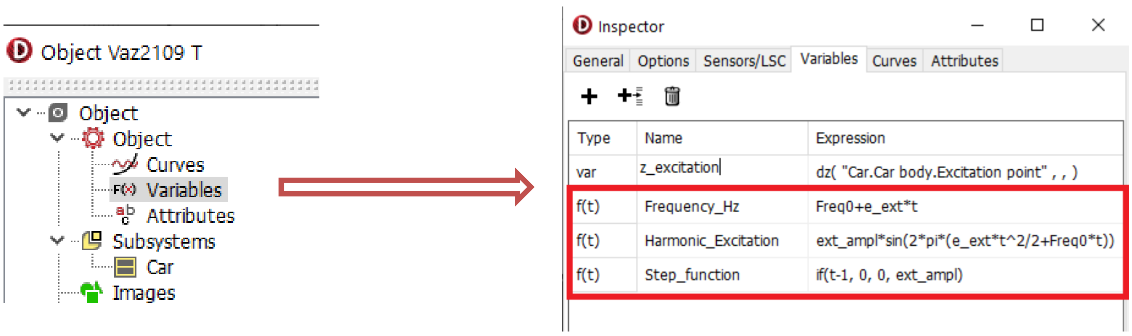


Figure 1.177. Description of variables as time function in UM Input

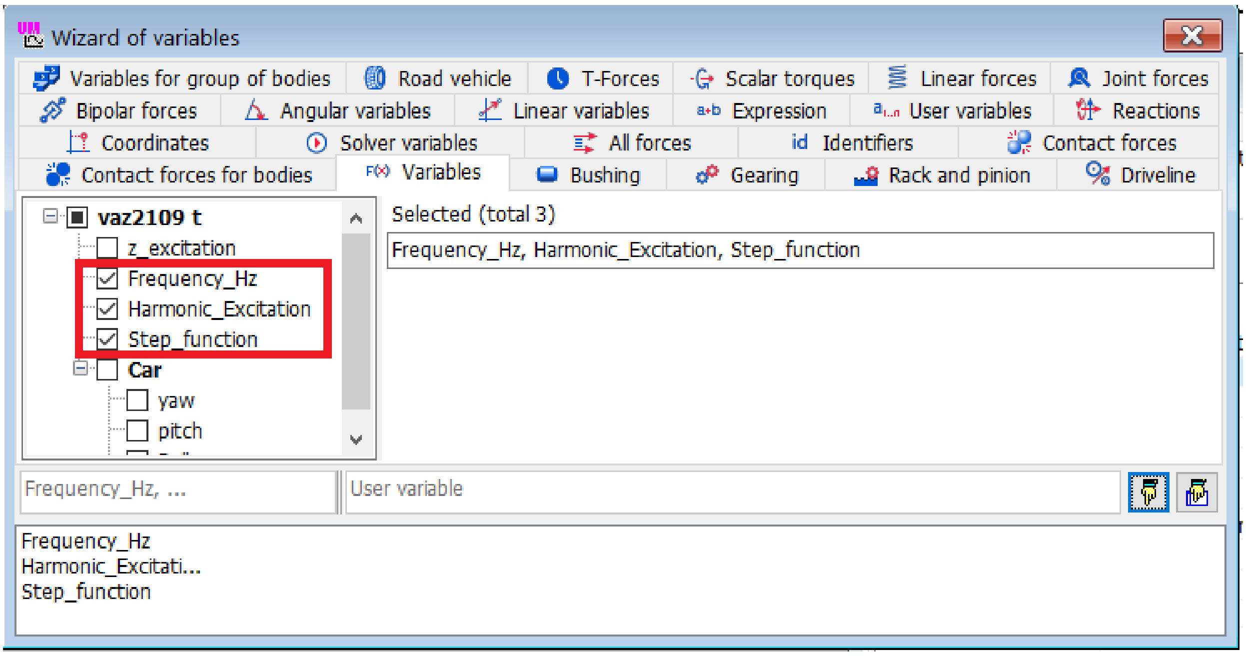


Figure 1.178. Access to variables created in UM Input program

1.9.4.5. Steering wheel rotation test

The test computes dependence of steer angles on steer wheel rotation; in particular, it allows the user to estimate the steering ratio.

The test requires

- Identification of wheel rotation locking parameters and strictly positive values of these parameters.
- Identification of steering.

Test starts from the equilibrium position of the vehicle and consists in rotation of the steering wheel according to the formula

$$\alpha_w = a_w \sin 2\pi f_w t,$$

where a_w, f_w are the amplitude (rad) and the frequency (Hz) of rotation of the wheel. These parameters should be set by the user, Figure 1.179.

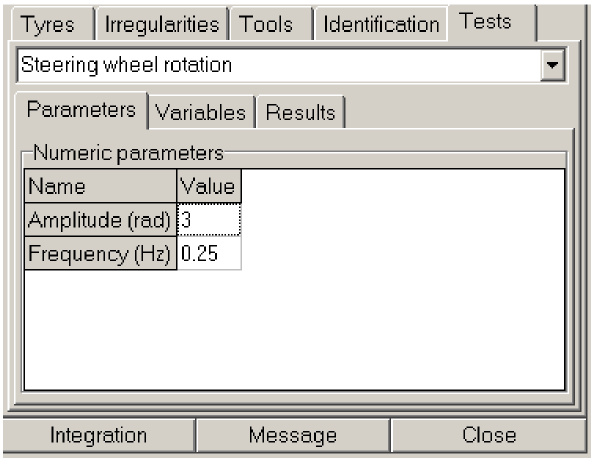


Figure 1.179. Parameters of steering wheel rotation test

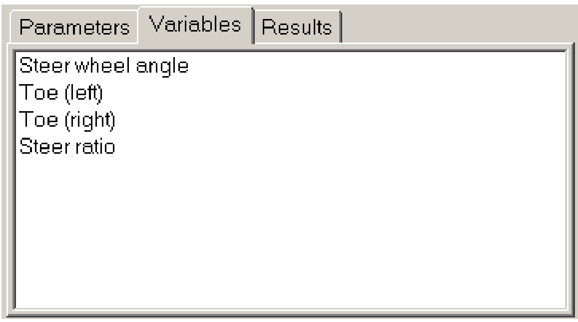


Figure 1.180. Variables of the test

Four standard variables are available with this test:

- Steering wheel angle α_w
- Steer (toe) angles δ_l, δ_r
- Variable, which can be used for evaluation of the steering ratio

$$i_w^e = \begin{cases} \dfrac{2\alpha_w}{\delta_l - \delta_r}, |\delta_l - \delta_r| > 0.001 \\ 0, |\delta_l - \delta_r| \le 0.001 \end{cases}$$

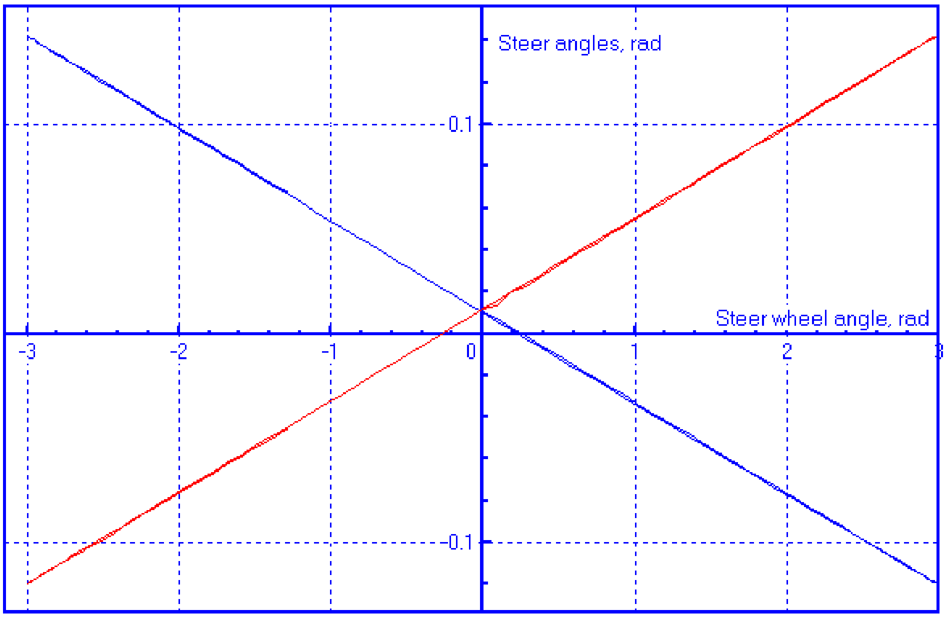


Figure 1.181. Steer angles versus steering wheel rotation angle

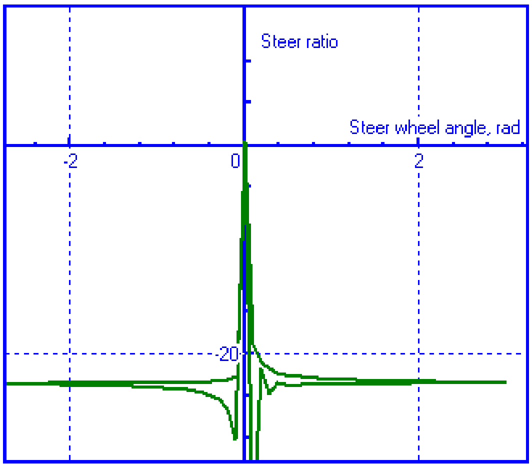


Figure 1.182. Variable i_w^e versus steering wheel rotation angle

Figure 1.181, Figure 1.182 show examples of plotting the variables during the test.

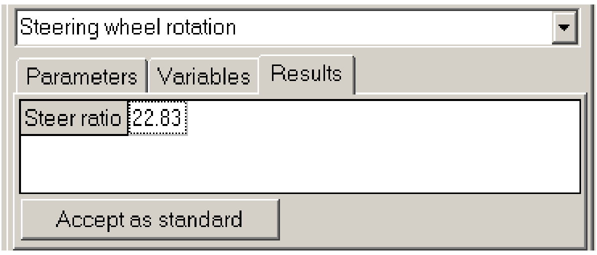


Figure 1.183. Result of steering wheel rotation test

After the end of the test the steering ratio is computed as

$$i_w = \frac{2a_w}{\delta_{l,max} - \delta_{l,min}}$$

where $\delta_{l,max}$, $\delta_{l,min}$ are the maximal and the minimal values of the left steer angle, and a_w is the amplitude of the steering wheel rotation. Click the **Accept as standard** button to accept the computed steering ratio i_w for other tests, see 1.9.1.2.3 “*Identification of steering*”.

1.9.4.6. Open loop steering test

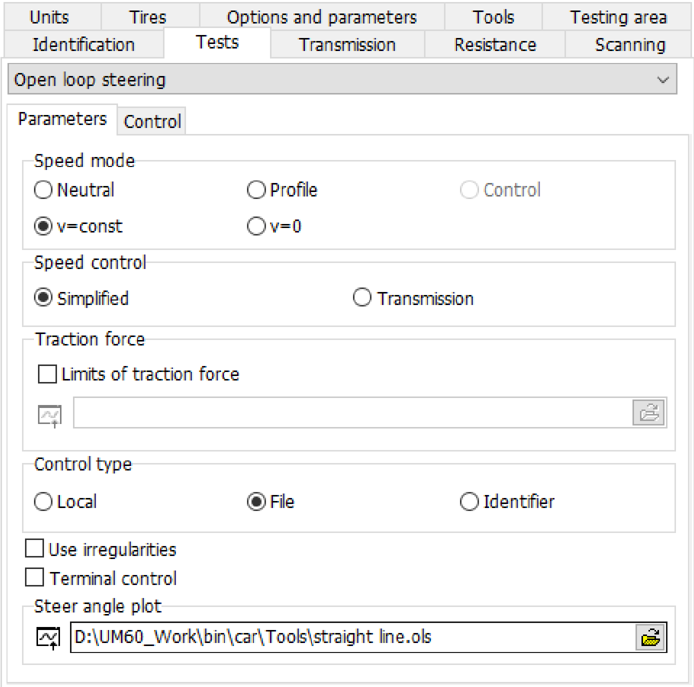


Figure 1.184. Open loop steering data

The test is used for simulation of maneuvers with an open loop steering, i.e. the time/distance history for the steering wheel angle should be used. The test requires

- Identification of the tire models, Sect. 1.5.9. *"Assignment of tire models to wheels"*, p. 1-65.
- Identification of steering, see Sect. 1.9.1.2.3 “*Identification of steering*” (four identifiers, and steer ratio).
- Steering angle function, Sect. 1.9.1.3.1 “*Setting graphs for steering wheel angle and vehicle speed*”.
- Identification of the simplified speed controller, see Sect. 1.9.1.2.1 “*Identification of parameters for simplified longitudinal speed control*”;
- Identification of transmission controller, Sect. 1.9.1.2.5 “*Identification of transmission control*”;
- Selection of **Speed mode**, Sect. 1.9.3 “*Speed modes and speed control*”;
- Identification of irregularities if the **Use irregularities** box is checked, Sect. 1.3.3.3. *"Assigning irregularities"*, p. 1-30.

The following check boxes specify some features of the test.

Use irregularities – if on, irregularities are taken into account.

Terminal control – if on, the steering wheel gets free when the end of the steering angle data is reached. For example, if the last point in the data corresponds to t=2s. Then, the steering wheel gets free since this time moment.

Control type

- **Local**: the steering angle plot is entered directly in the **Tools** tab, Sect. 1.9.1.3.1 "*Setting graphs for steering wheel angle and vehicle speed*".
- **File**: the steering wheel angle is specified by a file **.ols*, Figure 1.184. Use the button in the **Steering angle plot** group to assign a file. Creation of *.ols files is described in Sect. 1.9.1.3.1 "*Setting graphs for steering wheel angle and vehicle speed*".

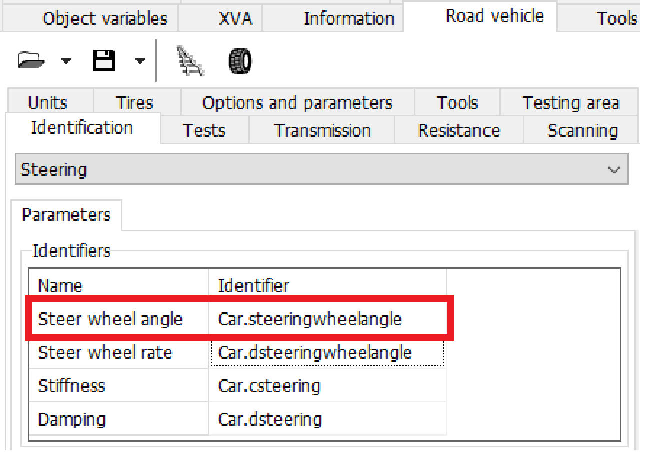


Figure 1.185. Assigned identifier for the steering angle

- **Identifier**: The steering angle is set via the angle identifier, Figure 1.185, see Sect. 1.9.1.2.3 "*Identification of steering*". The angle unit is radian. To change the value of the identifier during simulation, you can use several techniques described below.

  1. The **identifier control** tool; in this case in identifier can be a function of time or other variables, Figure 1.186. See Chapter 4. Sect. *"Identifier control"* for the tool description.

  2. The **Control panel** tool allows the user to create windows for interactive change of identifiers, See Chapter 4. Sect. *"Control panel"* for the tool description.

  3. Use interfaces to external software such as Matlab/Simulink, see Chapter 5, Sect. *"Creating and using external libraries"*.

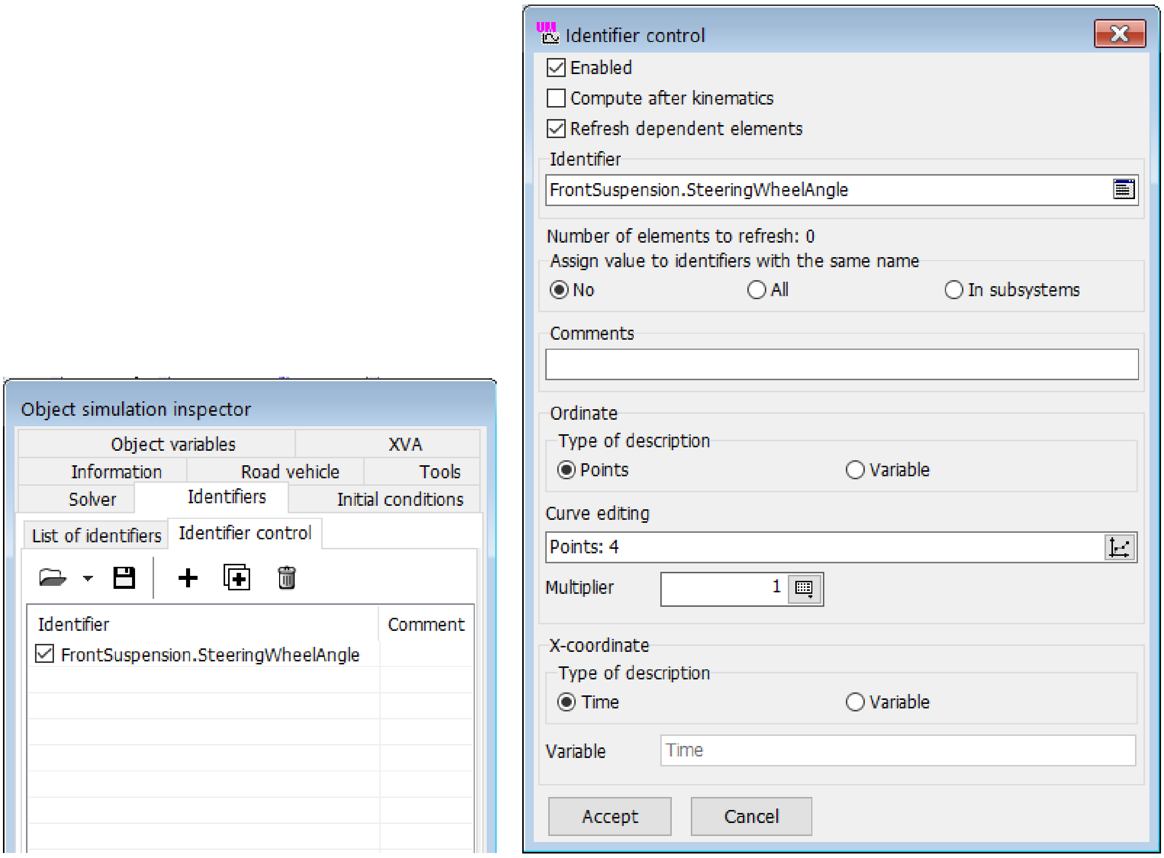


Figure 1.186. Use of the identifier control tool for setting steering wheel angle

1.9.4.7. Closed loop steering test: test with driver

1.9.4.7.1. General test parameters

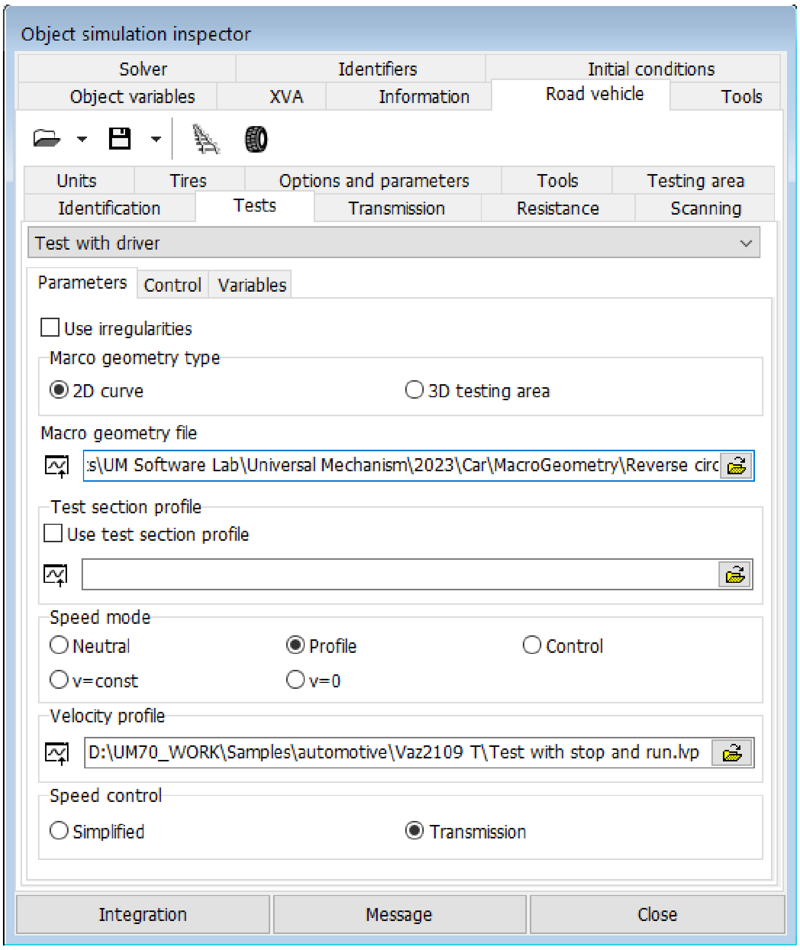


Figure 1.187. Closed loop steering test, model with transmission

The test is used for simulation of maneuvers with the closed loop steering, i.e. one of the driver models is used to follow the path, Sect. 1.4. *"Driver"*, p. 1-34.

The test requires

- Identification of the tire models, Sect. 1.5.9. *"Assignment of tire models to wheels"*, p. 1-65.
- Identification of steering, see Sect. 1.9.1.2.3 (four identifiers, and steer ratio).
- Identification of simplified speed control, see Sect. 1.9.1.2.1 "*Identification of parameters for simplified longitudinal speed control*";
- Identification of transmission control (for models with transmission), Sect. 1.9.1.2.5 "*Identification of transmission control*";
- Description of **Speed mode**, Sect. 1.9.3 "*Speed modes and speed control*";
- Identification of irregularities if the **Use irregularities box** is checked, Sect. 1.3.3.3. *"Assigning irregularities"*, p. 1-30.
- Creating a file of test section profile is necessary, Sect. 1.9.1.3.2 "*Creating files with test section profiles*", 1.9.4.7.2.3 "*Simulation with test section profiles*";
- Selection of **Macro geometry type**: a flat curve (**2D Curve**) or a triangulated surface (3D testing area).

The test stops

- if the simulation time is over; the user can continue the test after increasing the simulation time value in the pause mode;
- if the end of the desired path in the macro geometry path is reached; in this case the test cannot be continued.

UM supports two driver models that allow the car to follow a given path: the MacAdam model and the continuous preview model, Sect. 1.4 "*Driver*". The driver model parameters are shown in Figure 1.188.

The continuous preview model allows controlling the vehicle motion forward and backward (Reverse), while the values of the control parameters are different, see Figure 1.188, right. The first group of parameters determines the control when the car moves forward, and the second, marked as the 'Reverse', corresponds to the backward motion.

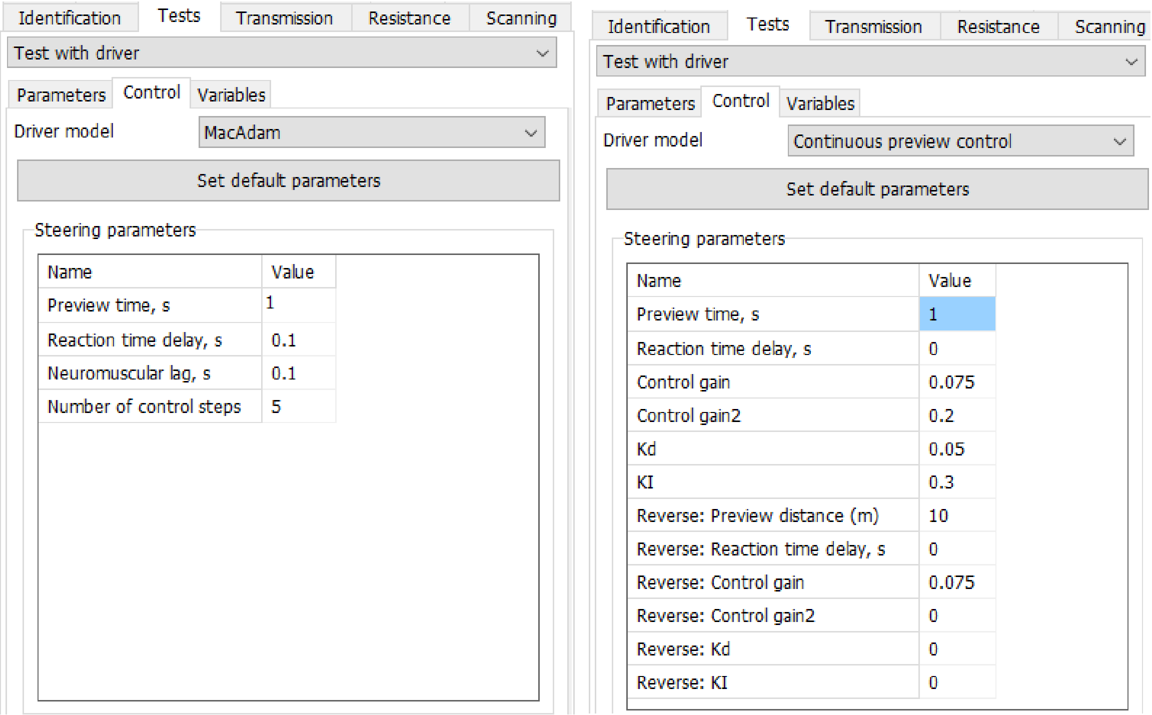


Figure 1.188. Parameters of driver model

Let us consider the features of the controlled reverse motion of vehicle.

- Controlled reverse motion can only be used for vehicles without a trailer or semitrailer. *For vehicles with a trailer, the control algorithm does not provide stable trajectory following*.
- The reverse motion corresponds to the motion at a negative speed. In particular, the speed identifier *v*0 can take negative values.
- The macro geometry curve must be designed with the backward movement in mind. It should be remembered that modeling is possible only within the given trajectory; the simulation stops if the distance from the vehicle to the boundary of the trajectory is less than the preview distance, both when moving forward and backward.

Remark 1. The MacAdam driver model has been developed for a vehicle without a trailer or semi-trailer. For simulation of vehicles with trailers or semi-trailers, it is recommended to use a continuous preview model, and it is often necessary to adjust the values of the control parameters in order to achieve a stable movement.

Remark 2. The continuous preview control model uses the derivative of the error function, which requires a differentiable function of the desired path. In this case a spline interpolation of the path curve is necessary (Sect. 1.3.1 "Defining a macro profile using curves").

The list of test variables includes (Figure 1.189)

- Coordinate X – Cartesian coordinate X of the vehicle;
- Coordinate Y – Cartesian coordinate Y of the vehicle;
- Steer wheel angle – the real value of the angle; normally it is close to the driver control variable;
- Computed control – the computed value of the steering wheel angle before the driver neuromuscular filter;
- Driver control – the computed value of the steering wheel angle after the driver neuromuscular filter;
- Desired path deviation – error in path following (deviation of the real path from the desired one).
- Resistance force is the force of the total resistance to vehicle motion including aerodynamic drag and tire rolling friction, converted to the force.

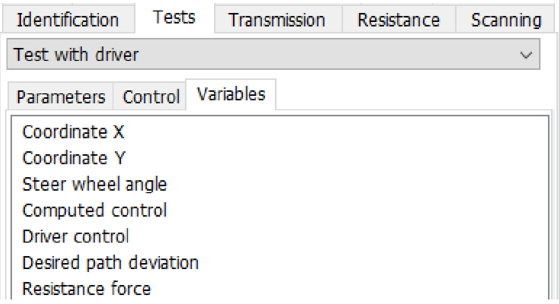


Figure 1.189. List of variables for closed loop steering test

1.9.4.7.2. Features of driver test in case of 2D curve macro geometry

The test with driver is the main dynamic test for a vehicle model, which allows the user to study the behavior of the vehicle when passing curved sections of the road taking into account the vertical profile of the terrain. When performing the test, the user obtains information about the loading of the model elements, can assess agreement with the standards, can perform a durability analysis. The simulation is used to evaluate the satisfaction of the vehicle with standard tests such as lane change, tight curve driving, rollover resistance, standard obstacles (speed bump, curb, road damage), etc.

1.9.4.7.2.1. Selection of macro geometry file

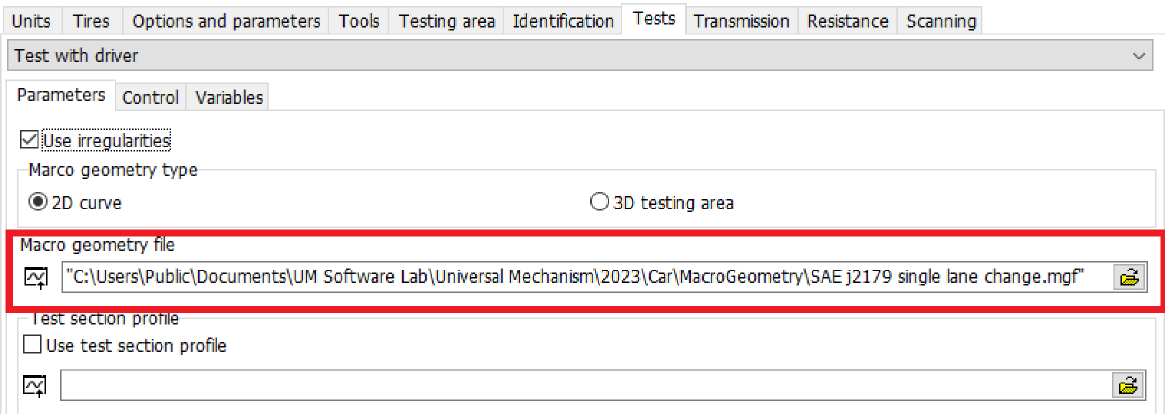


Figure 1.190. Selected macro geometry file

The user can use the standard UM files or create their own file.

Before run the test, a macro geometry file must be loaded by clicking on the button, Figure 1.190. The user can use the standard UM files or create their own file, Sect. 1.3.1 "*Defining a macro profile using curves*".

1.9.4.7.2.2. Use of irregularity files

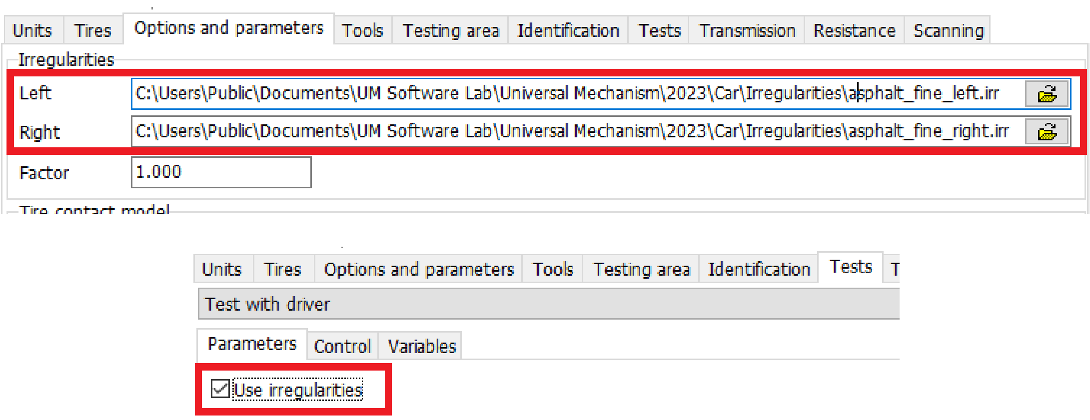


Figure 1.191. Assignment of irregularities

Unlike a triangulated surface, which includes roughness in the surface geometry, in this case the micro profile is defined using the files of irregularities, Figure 1.191, Sect. 1.3.3.3 "*Assigning irregularities*". The files are assigned on the **Options and parameters** tab.

1.9.4.7.2.3. Simulation with test section profiles (TSP)

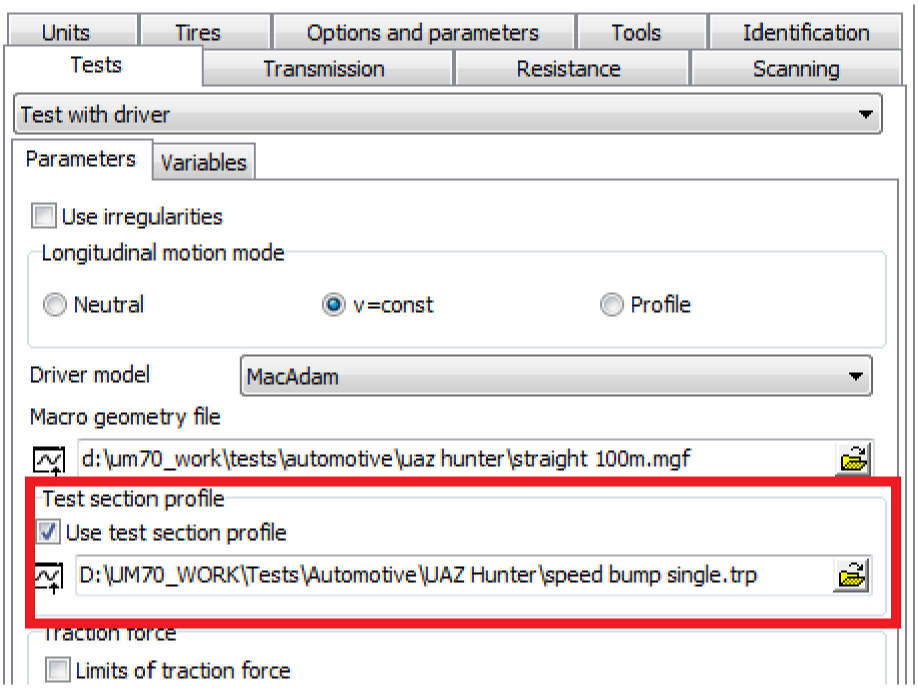


Figure 1.192. Choice of file with TSP

To run simulation with a TSP curve during the test with driver, select an existing *.trp file with the  button and check the **Use test section profile** option. Switching on/off the **Use test section profile** option allows the user to compare promptly simulation results with and without TSP.

Creating TPS files is described in Sect. 1.9.1.3.2 "*Creating files with test section profiles*", the general information about TSP can be found in Sect. 1.3.4 "*Test section profile*".

Tune the tire contact model according to the selected TSP, Figure 1.193. Look at Sect. 1.5.1 *Single point and multipoint normal contact models* for detailed description of the tire contact models.

It is recommended to use the **multipoint** contact model. If the **Distributed flexible contact** option is disabled, then discrete point contact of the tire with the road is used, which is usually used when the tire hits a step or curbstone.

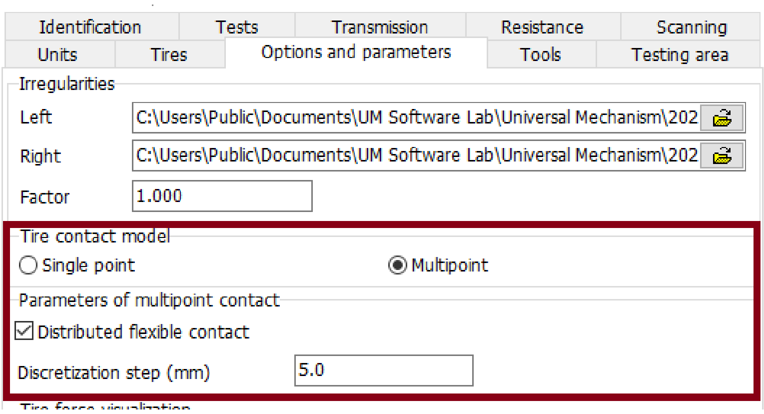


Figure 1.193. Tire/road contact model

Remark An additional advantage of use the TSP consists in drawing the corresponding deviations in animation window. Usual irregularities are not drawn, and if the user wants to see a short vertical irregularity during the animation, he should describe the vertical irregularity as TSP. It is recommended to set a small enough **Image step** to get an appropriate quality of the road deviation image in the animation window, Figure 1.194, Sect. 1.9.1.4.4 “*Road image*”.

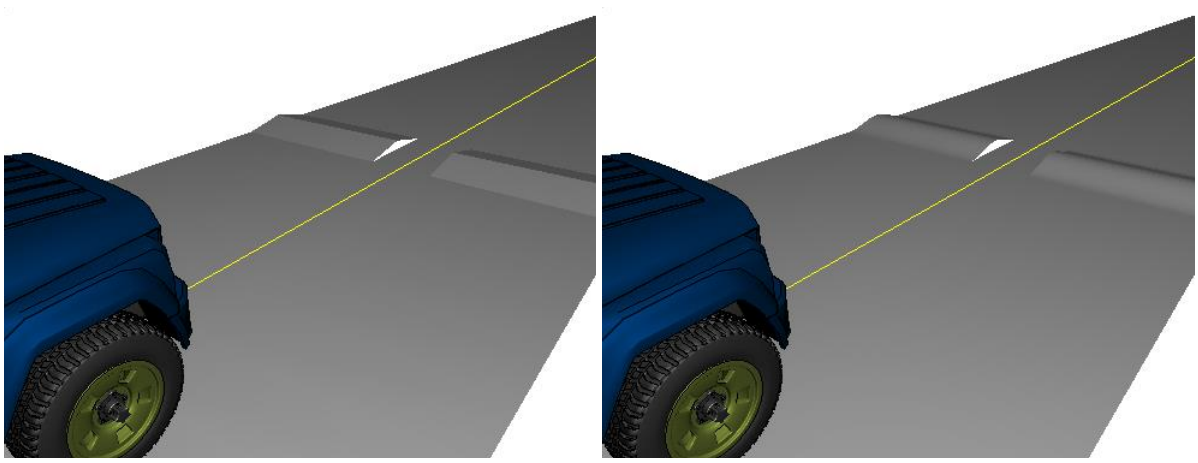


Figure 1.194. "Bump" with an image step of 0.5 m and 0.05 m

1.9.4.7.3. Feature of driver test in case of 3D testing area (triangulated surface)

1.9.4.7.3.1. Assignment of testing area and route

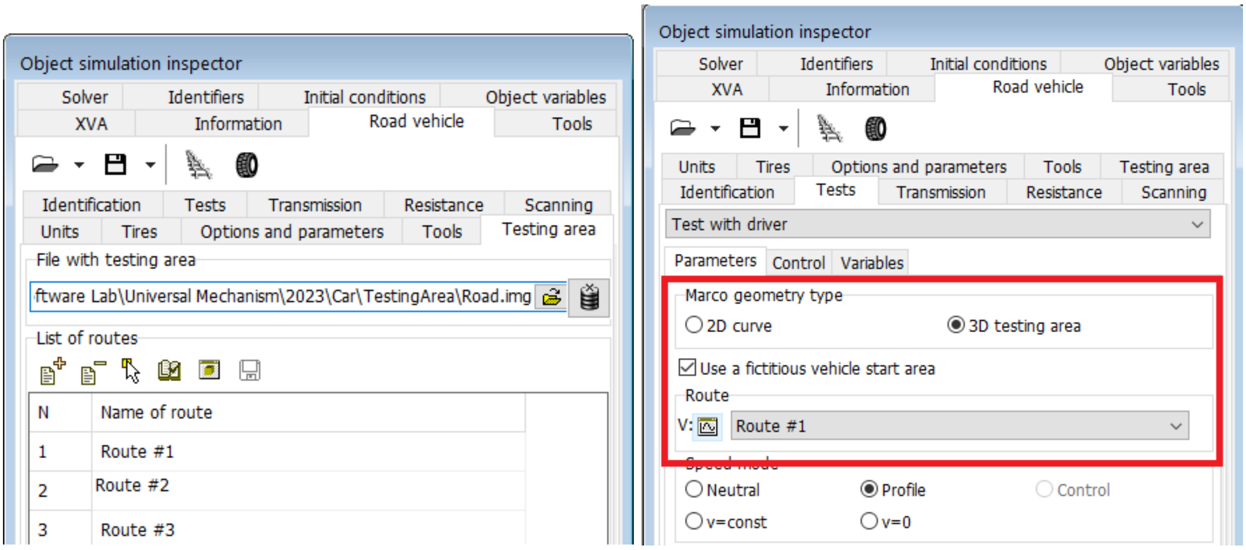


Figure 1.195. Setting parameters for motion on triangulated surface

When moving along a triangulated surface, the user should first select a surface file on the **Testing area** tab and, if necessary, create or edit routes (see section 12.3.2.2. "Setting routes"). With the selected file and the presence of at least one route, the assignment of a **3D testing area** becomes available, Figure 1.195, right.

1.9.4.7.3.2. Initial conditions for motion on testing area

When moving along a triangulated surface, it is important to set correctly the initial conditions, that is, the coordinate values for which there are no intensive transient processes at the start of the simulation.

1. Use a fictitious start area to enter the surface.

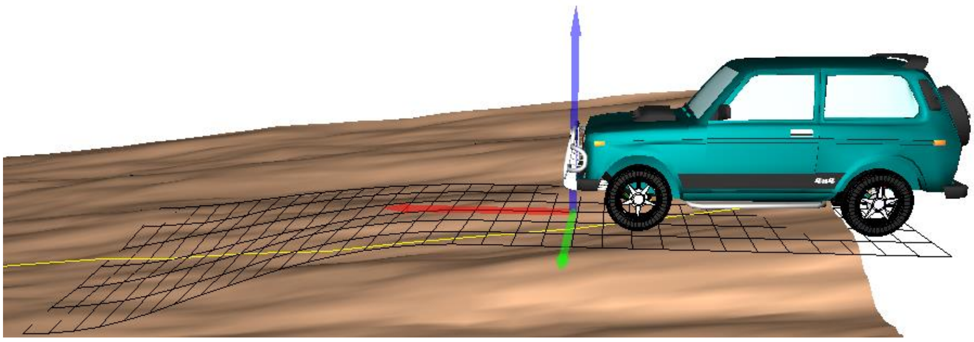


Figure 1.196. Fictitious start area

The fictitious area consists of two sections, Figure 1.196:

- a rectangular horizontal section having a zero vertical coordinate; the front boundary of the first section is located at a distance of 1 m in the direction of movement from the origin of the SC0 in Figure 1.196;

- a transition section that allows the car to smoothly enter the roughness of the surface; the section length is 5m.

Use of the fictitious area has several advantages.

- the standard initial conditions obtained as a result of the equilibrium test can be applied;
- initial conditions do not depend on the chosen route;
- in the initial position, a part of the wheels can be outside the area of the triangulated surface, like the rear wheels of a car in Figure 1.196 or most of the truck wheels in Figure 1.197.

Despite these advantages, this method is recommended to be used for surfaces with a small height difference in the area at the initial position of the car, like in Figure 1.197. Otherwise, it is better to use an alternative approach described below.

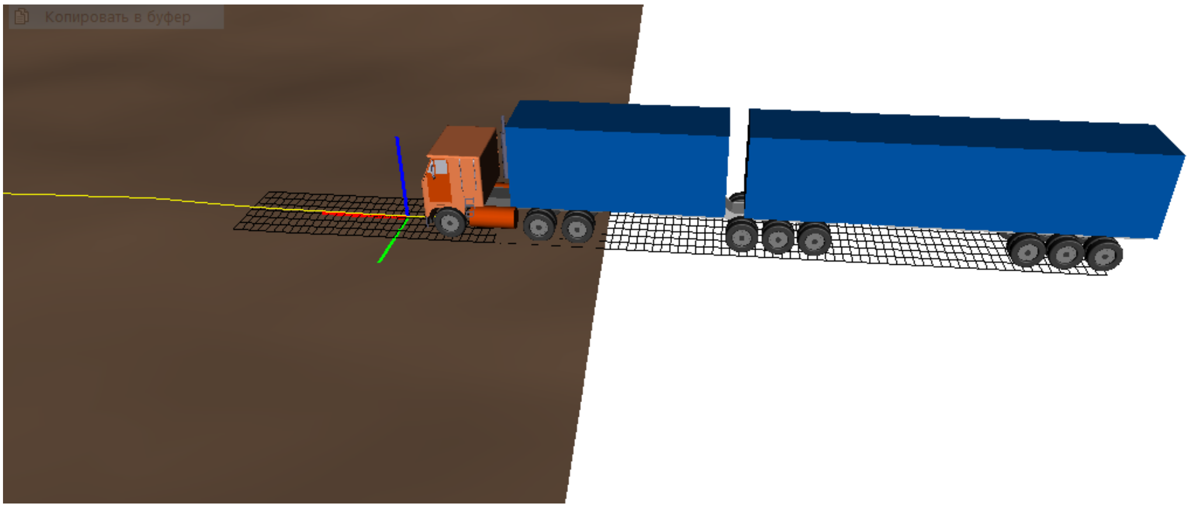


Figure 1.197. Fictitious area for a truck

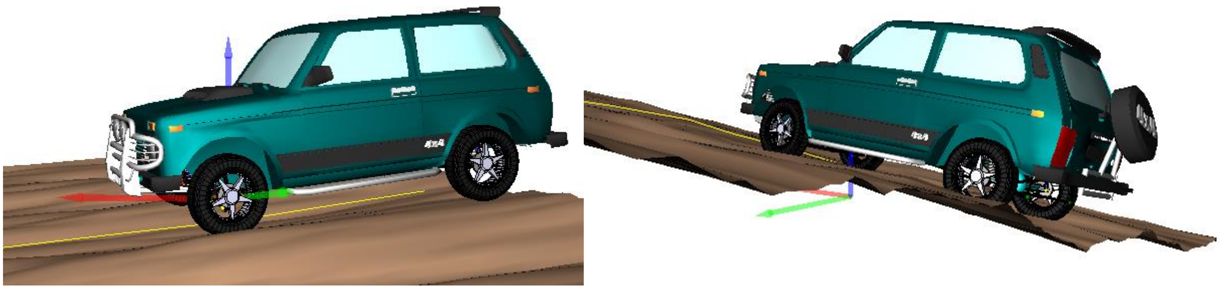


Figure 1.198. Initial car positions for different routes on the same surface

2. Calculation of initial conditions corresponding to the surface profile and route.

The calculation of the initial conditions should be performed for each route, saved in files and loaded when changing the route before starting the simulation, Figure 1.198.

To calculate the initial conditions, the following sequence of steps is recommended:

- assign the desired route from the list, Figure 1.199, right;

- set zero speed mode **v=0**;
- set the multipoint flexible contact of the tires with the road, Sect. 1.9.4.7.3.3;
- on the **Initial conditions** tab, place the car in such a way that all wheels are above the surface and do not penetrate into it, Figure 1.199, left; you can place the wheels above the surface by editing the route e.g. by increasing or decreasing the distance of the first point from the border in the curve editor, Figure 1.16;
- start the simulation process and switch to the pause mode after the car comes to the equilibrium position; in the pause mode window, save the coordinates to a *.xv file using the **Save** button, Figure 1.200; it is recommended to use the name of the route in the name of the file with coordinates;
- before starting to move along the selected route, set the initial conditions on the **Initial conditions** tab using the created file; for this purpose click on the button ; After the first file upload, it is recommended to set the speeds to zero by the button v=0 and to rewrite the file with button .

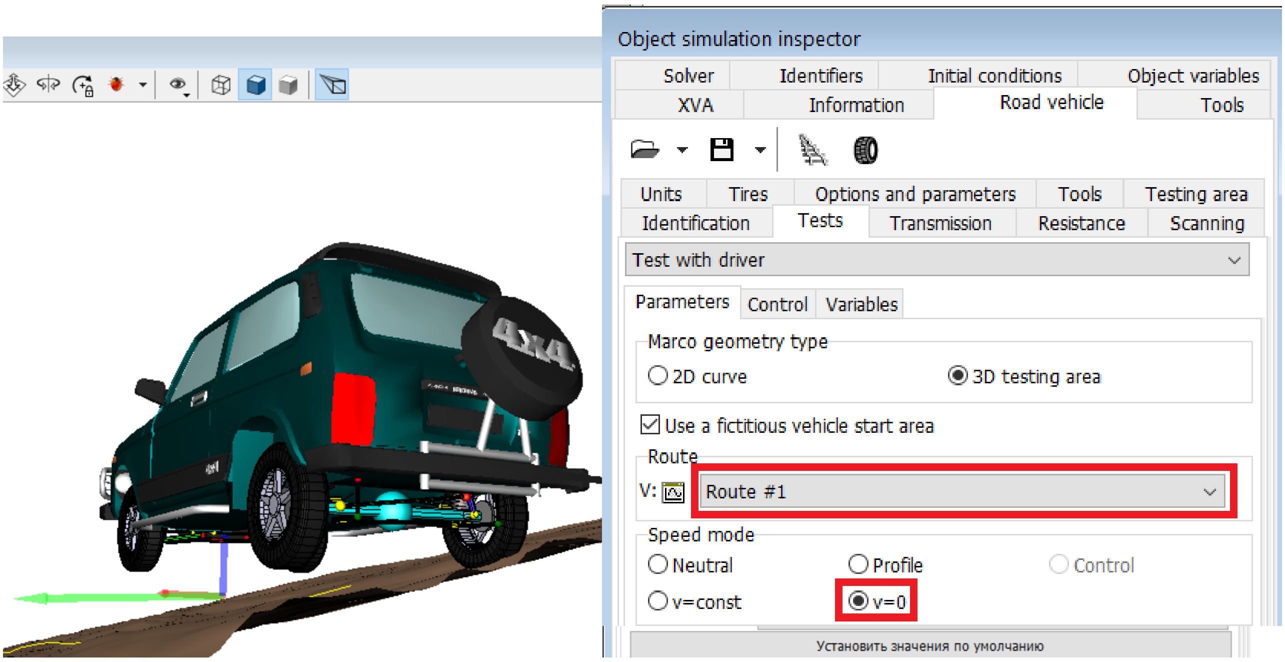


Figure 1.199. Position of a car before computation of initial conditions

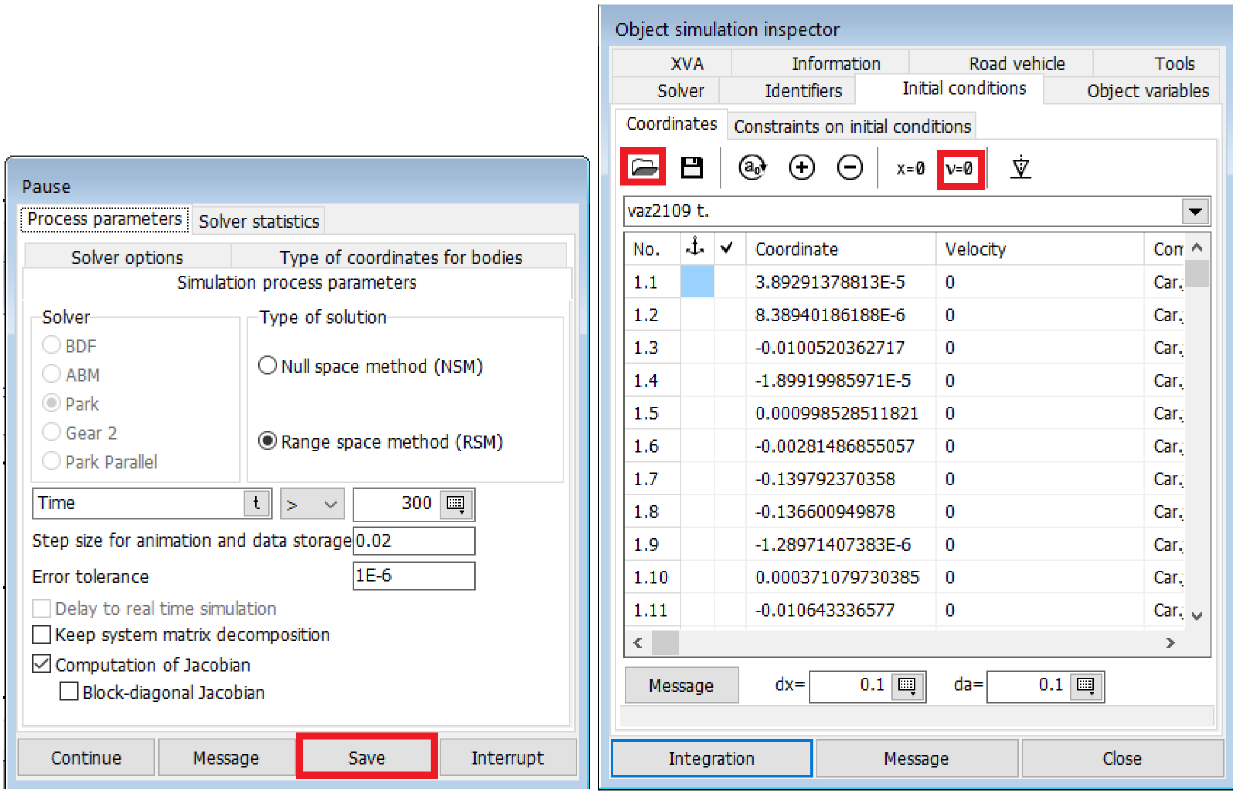


Figure 1.200. Saving coordinates in the pause mode and assigning coordinates by loading file

1.9.4.7.3.3. Model of tire/road interaction

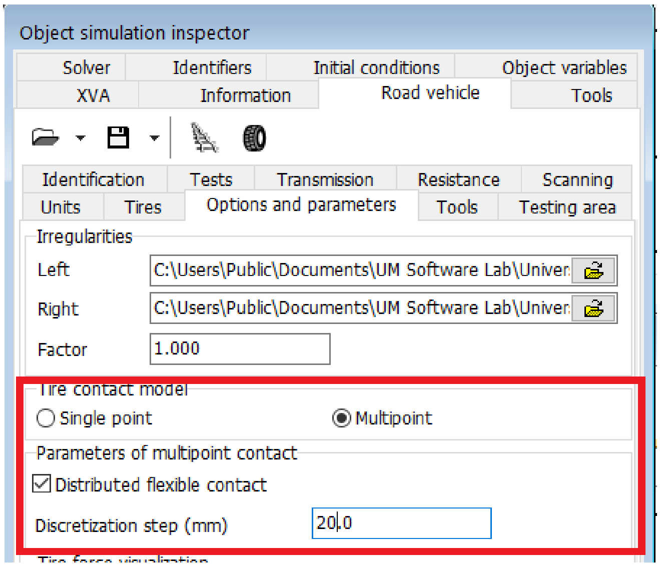


Figure 1.201. Typical setting for tire contact model when modeling motion on triangulated surface

When simulating the motion along the testing area, it is recommended to use a multipoint tire contact model with distributed flexible contact. The distributed flexible contact makes it possible to smooth out jumps in the direction of the normal force caused by a change in the normal to the triangulated surface when crossing the mesh edges. The same contact model should also be used when calculating the initial conditions corresponding to the surface profile, Sect. 1.9.4.7.3.2 “*Initial conditions for motion on testing area*”.

1.9.4.8. Car simulator

1.9.4.8.1. General information

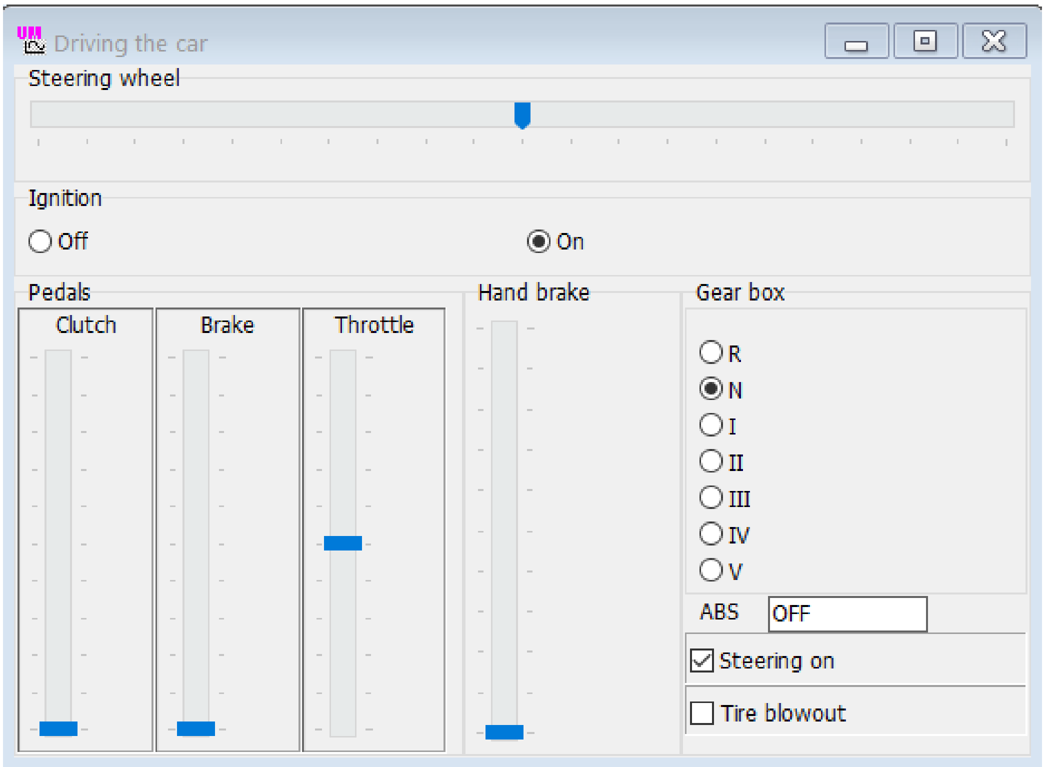


Figure 1.202. Simulator control panel for model with transmission

This test allows the user to control the car during the simulation of motion using the control panel, Figure 1.202. The test can be used with or without a transmission model. In the first case, both steering and transmission control are available. In the second case, only steering is used.

The trackbar at the top of the panel is applied for steering wheel rotation, Figure 1.202. To move the trackbar slider, use both the mouse and the ← → keys on the keyboard.

Steering can be disabled during the simulation using the **Steering on** option. In this case, the steering wheel becomes free and the motion of the car without steering control is modeled. The control returns with the help of the same option. In the uncontrolled motion mode, the position of the **Steering wheel** trackbar is set by the program in accordance with the current value of the coordinate corresponding to the steering angle, section 1.9.1.2.3 “*Identification of steering*”.

The **Tire blowout** option allows simulating the process of driving a car after a front left tire blowout, Figure 1.203. The simulation of the blowout consists in reducing the rolling radius of the tire and in increasing its rigidity four times.

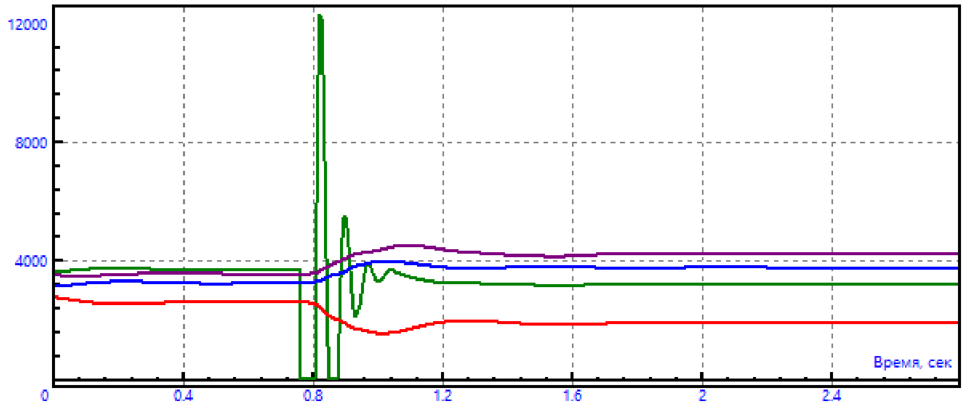


Figure 1.203. Normal tire force when simulation a tire blowout

1.9.4.8.2. Simulator for vehicle with transmission model

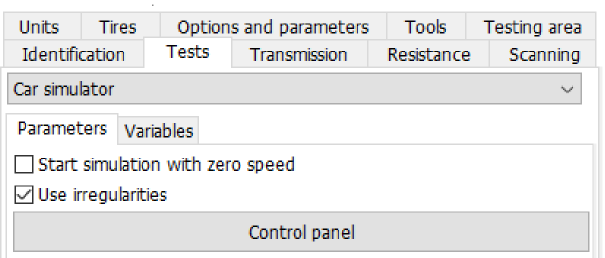


Figure 1.204. Simulator options for model with transmission

When using the **Start simulation with zero speed** key, the simulation starts from the equilibrium position (the main simulator mode), otherwise the initial speed is set by the identifier *v*0.

By clicking the **Control Panel** button, the window in Figure 1.202 appears before starting the simulation, otherwise it appears automatically at start.

The lower positions of the control trackbars correspond to the following states:

Clutch - the clutch pedal is released, that is, the clutch is engaged;

Brake - the brake pedal is released, there is no braking;

Throttle - the accelerator pedal is released, the fuel supply with the engine on corresponds to idling;

Hand brake - no braking.

An example of the sequence of actions when executing a test (when the simulation process is running)

1. Turn on the engine.

2. Disengage the clutch by placing the trackbar in the top position and the accelerator bar in the desired position.

3. Set a gear position in the **Gearbox** group.

4. Release the clutch.

5. Next, control the movement using the panel elements

1.9.4.8.3. Simulator for vehicle without transmission model

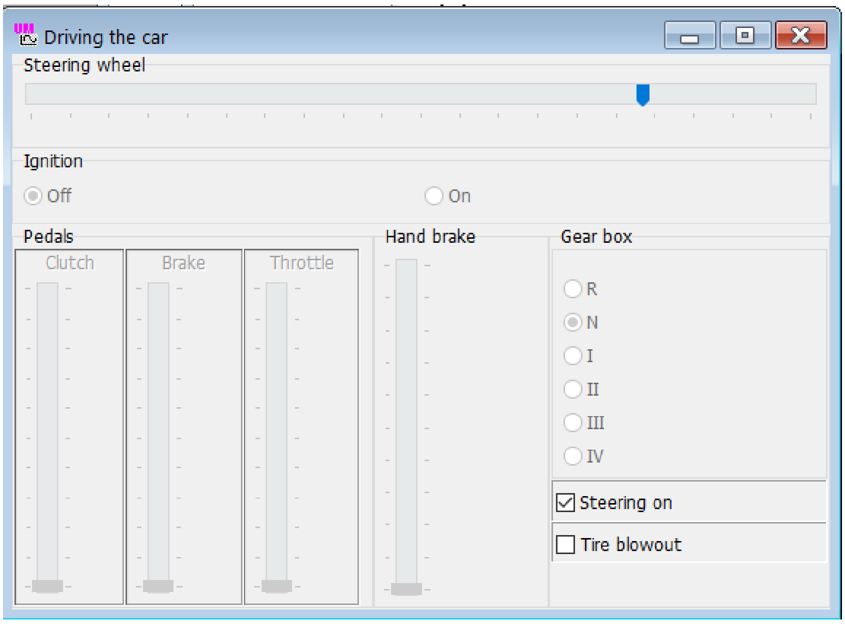


Figure 1.205. Control panel for model with transmission

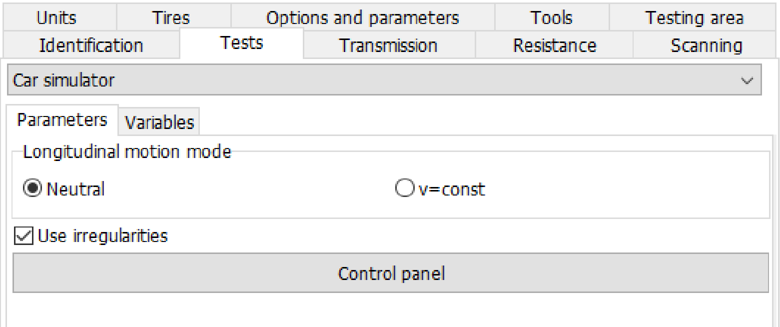


Figure 1.206. Simulator options for model without transmission

In the absence of a transmission model, only the steering of the car is available, Figure 1.205.

The car motion in the simulator mode is possible either in neutral mode or at a constant speed, Figure 1.206. The speed is set on the identifiers tab with the value *v*0.

1.9.4.9. Vertical harmonic loading test

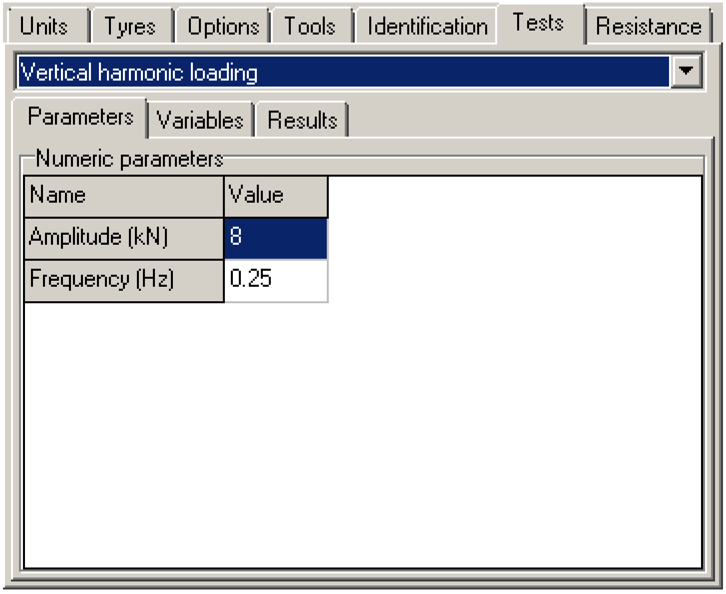


Figure 1.207. Vertical loading test parameters

The test computes quasistatic deflection of suspension caused by a slow harmonic vertical force applied in the chassis center of mass.

The test requires

- Identification of wheel rotation locking parameters and strictly positive values of these parameters.
- (four identifiers).

Test starts from the equilibrium position of the vehicle. The force is computed as

$$P_Z = P_0 \sin 2\pi f_p t,$$

where P_0, f_p are the amplitude (kN) and the frequency (Hz) of the force law. These parameters should be set by the user, Figure 1.207.

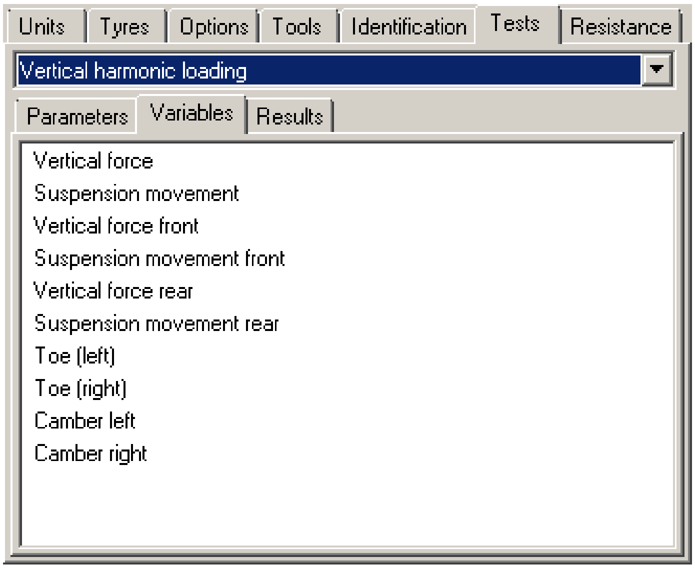
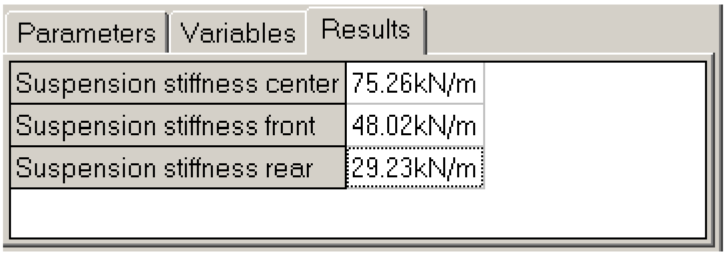


Figure 1.208. Vertical loading test variables

The list of test variables is shown in Figure 1.208. The front part of the vertical force is computed as a sum of vertical forces acting on the front wheels. Analogously the rear part of the loading is evaluated.



Suspension stiffness center	75.26kN/m
Suspension stiffness front	48.02kN/m
Suspension stiffness rear	29.23kN/m

Figure 1.209. Vertical loading test results

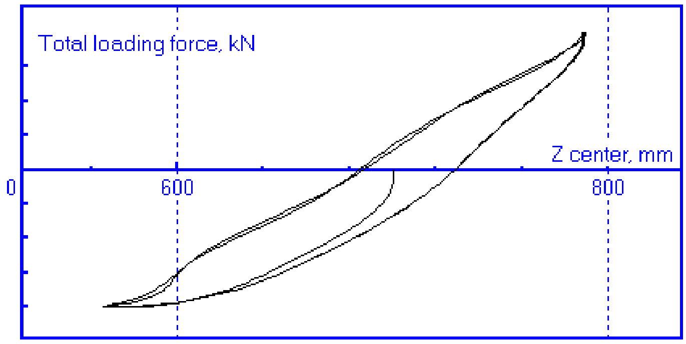


Figure 1.210. Simulation results: Load versus vertical position of center of mass

The list of results contains values of three stiffness constants: the total stiffness of the suspension, and stiffness of the front and the rear suspensions. The stiffness constants are evaluated from the linear regression analysis.

1.9.4.10. Horizontal harmonic loading test

Parameters, variables and results of the horizontal harmonic loading test are fully consistent with the vertical loading, Sect. 1.9.4.9 “*Vertical harmonic loading test*”.

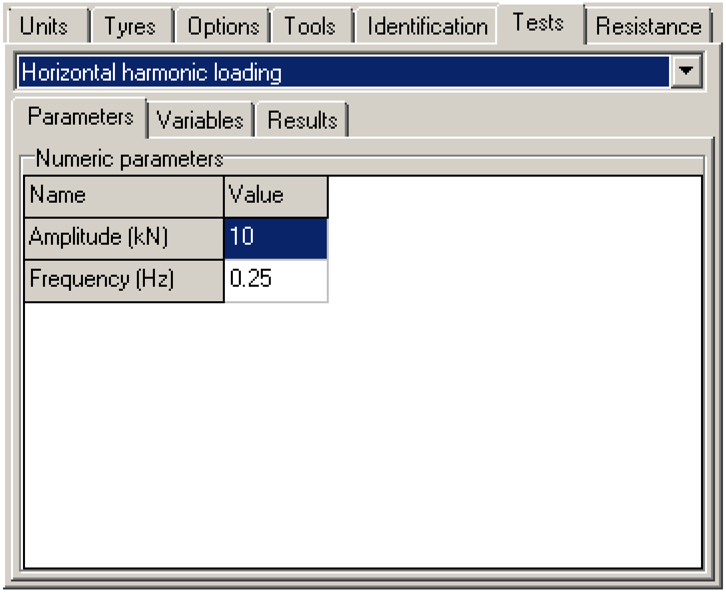


Figure 1.211. Horizontal loading test parameters

1.9.5. Road vehicle specific variables

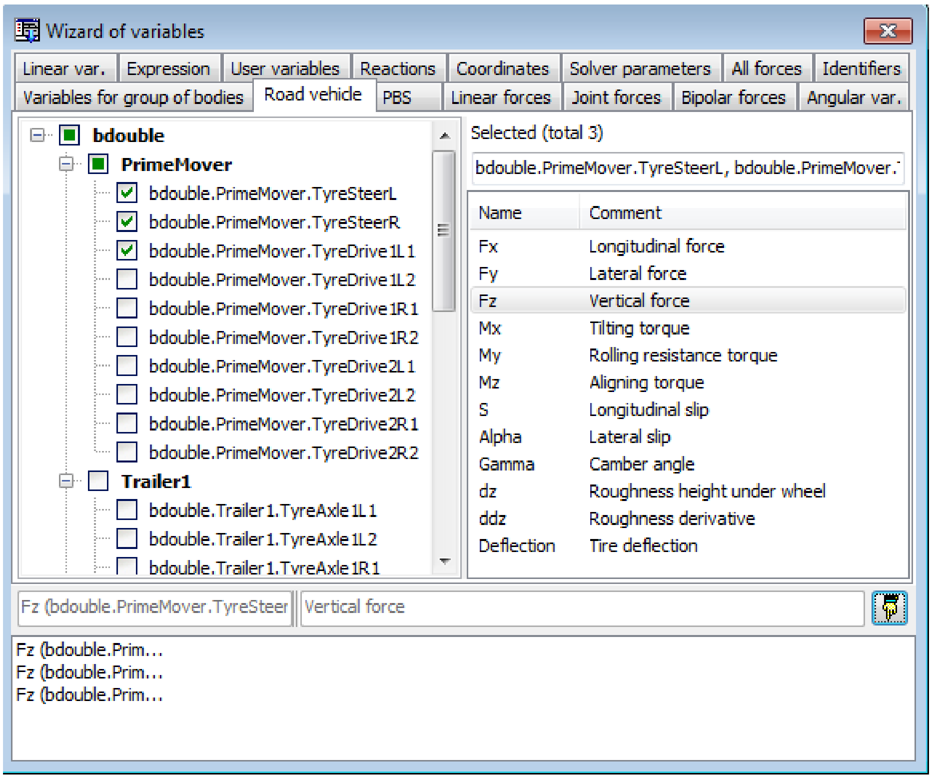


Figure 1.212. Variables related to tire/road interaction

Variables related to the tire/road interaction are available on the **Road Vehicle** tab of the **Wizard of variables**, Figure 1.212. Use the **Tools | Wizard of variables…** menu command to open this window. Use other tabs of the wizard to create kinematic and dynamic variables different from the tire variables.

To get information about creating variables and their usage see Chapter 4.

1.10. Features of linear analysis of road vehicles

The use of linear analysis of road vehicles, in particular, the calculation of eigenvalues and root locus, has a number of features that we consider in this section. General methods of static and linear analysis are discussed in Chapter 4, in the section "*Static and linear analysis*". See also Sect. "*SLA for wheeled vehicles*".

1.10.1. Modes of vehicle linear analysis

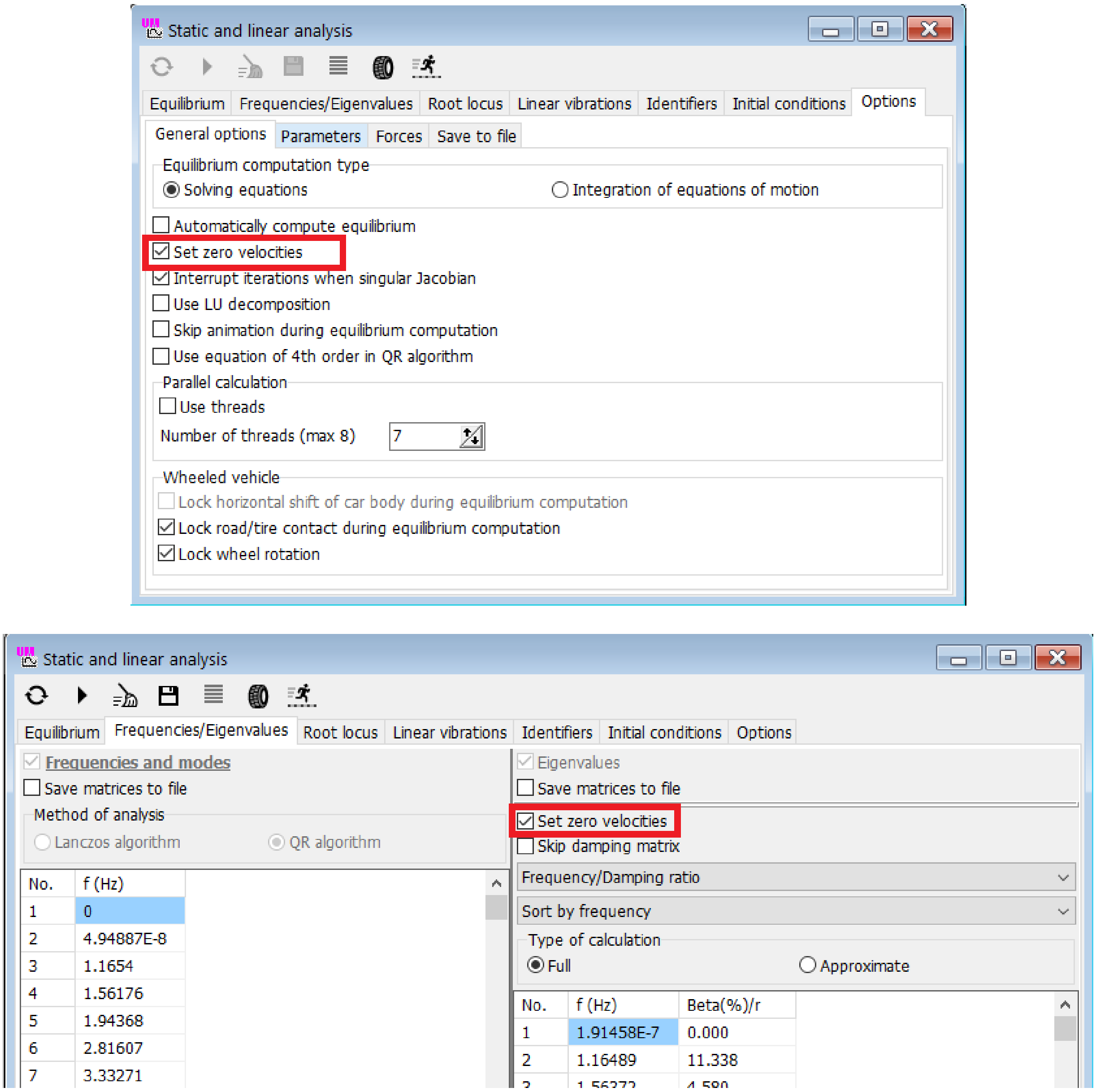


Figure 1.213. Setting mode of standing vehicle

For the linear analysis of a vehicle, two modes are used that differ in the model of the interaction of tires with the road: **the mode of a standing vehicle** and the longitudinal motion mode. In standing vehicle mode, enable the checkbox **Set zero velocities** either on the **Options** tab or on the tab for calculation of frequencies and roots, Figure 1.213.

In the **standing vehicle mode**, the vehicle speed is equal to zero. Each tire at the point of contact is connected to the road by linear springs with static stiffness coefficients Kx, Ky in the longitudinal and transverse directions. These coefficients are specified in the tire model, Figure 1.214. This mode is the main one when calculating the natural frequencies and damping ratios of the model.

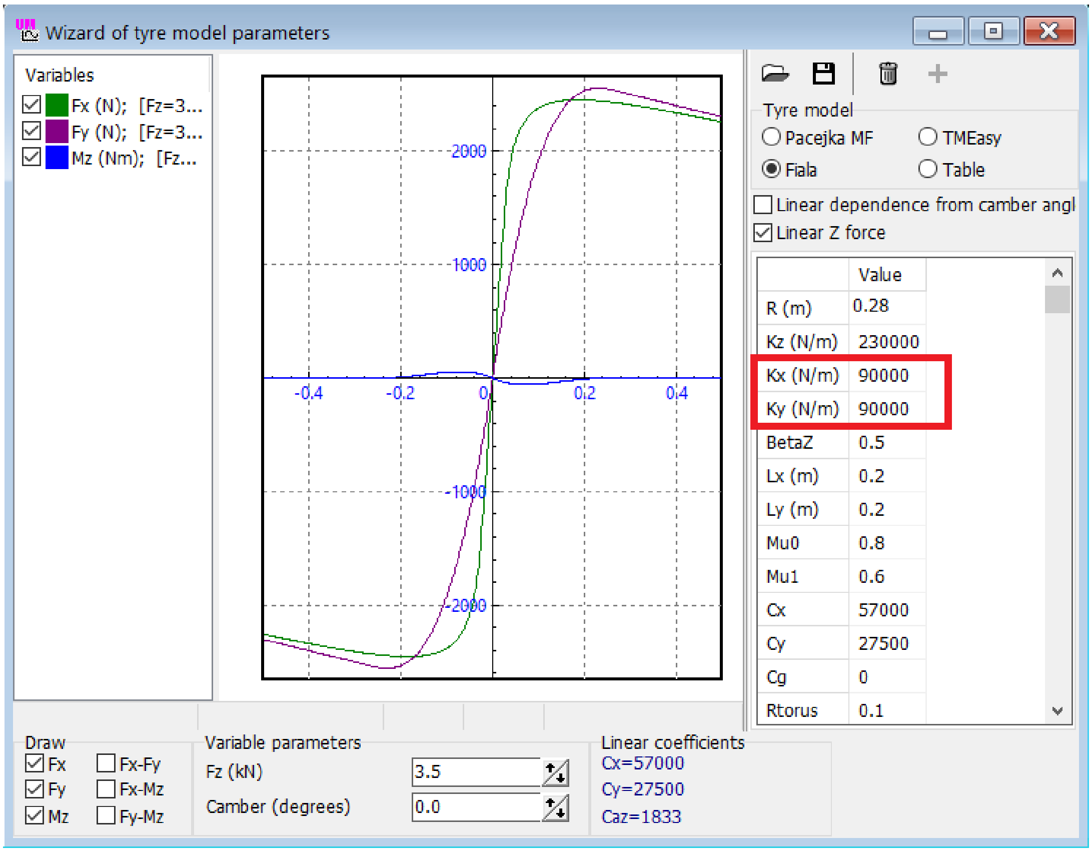


Figure 1.214. Static stiffness coefficients in tire model

The vehicle speed differs from zero in the **driving mode of the car** and the wheels rotate. Forces of interaction between the tire and the road are taken into account and linearized, see Sect. 1.5 “*Tire models*”. The checkbox **Use/Set zero velocities** is disabled**.** In this mode, the user can study the influence of the parameters of the driver model.

1.10.2. Linear analysis of influence of driver model parameters

Linear analysis in the **driving mode** of the car (Sect. 1.10.1 “*Modes of vehicle linear analysis*”) allows studying the influence of the driver model parameters (Sect. 1.4.3 “*Combination of PID controller and preview model*”) on the eigenvalues of the linearized model. The most effective technique is the root locus drawing depending on these parameters

To use this feature of the program, it is necessary to parameterize the control coefficients (1.1)

$$T_p, K, K_2, K_d, K_I, \qquad (1.2)$$

see Sect. 1.9.1.2.4 “*Parameterization of driver model*”.

The linear analysis in UM does not take into account the phenomenon of delay, so the delay value in the driver's reaction is not included in this list.

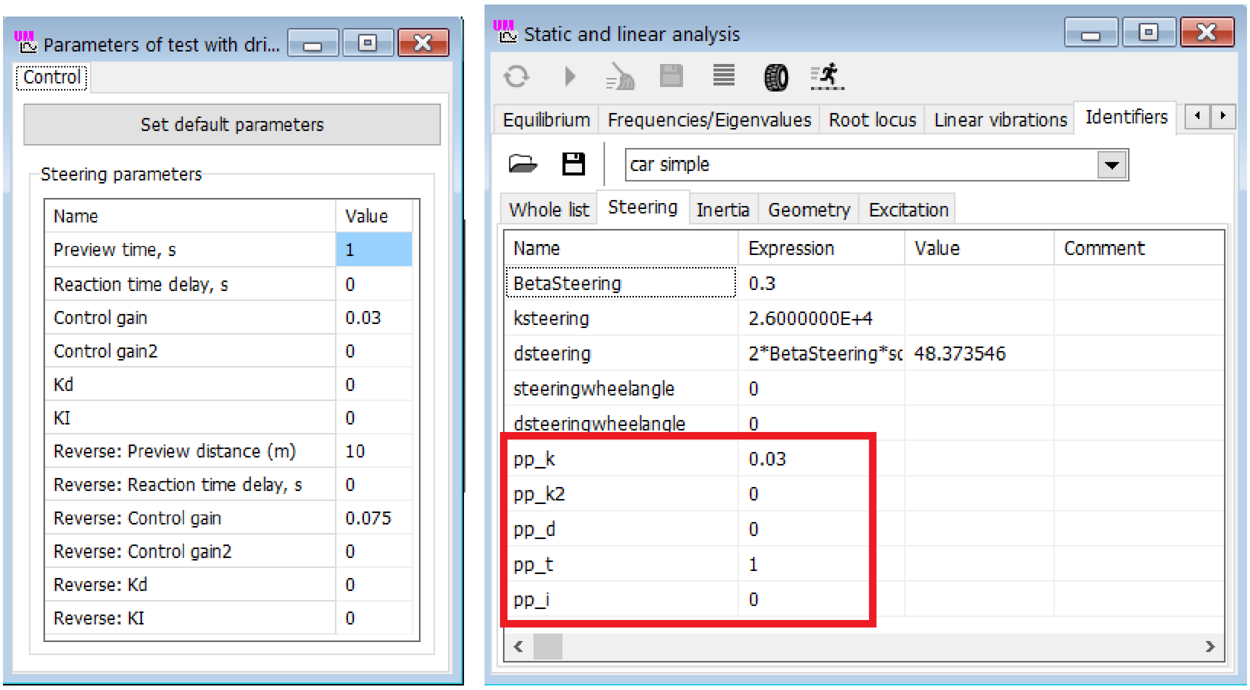


Figure 1.215. Windows with the current parameters of driver model and list of identifiers

To change the current values of control parameters in the linear analysis mode, use the button  in the upper part of the window in Figure 1.215, which opens a window with the table of parameters (Figure 1.215, left). The control parameters can be changed through the values of the corresponding identifiers that parameterize the control coefficients (Figure 1.215, right). Changing the parameter values in the two described ways is automatically synchronized, that is, changing the values in the table in Figure 1.215 left automatically leads to a change in identifiers and vice versa.

When the **Set zero velocities** option is disabled (Figure 1.213), the parameters of the driver model affect the eigenvalues through the forces of interaction between the tire and the road

$$F_x = c_x s_x, \qquad (1.3)$$
$$F_y = c_y s_y.$$

The parameters of the driver model affect the lateral force F_y through the side slip s_y. We will study this effect in the next section on a simplified car model.

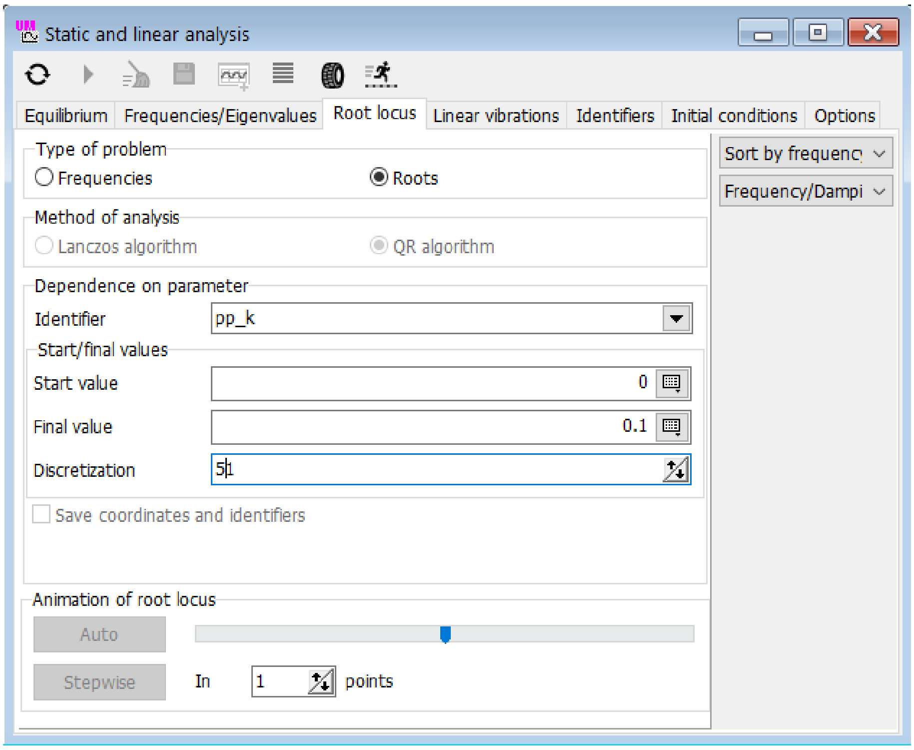
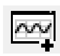


Figure 1.216. Setting parameters for computation of root locus

The most effective way to study the influence of the parameters of the driver model is to draw root locus, that is, the dependences of eigenvalues of the linearized equations (in other words, the roots of the characteristic equation) on the value of one of the parameters. Dependencies are drown on the complex plane.

The following steps are recommended (Figure 1.216):

- Open the **Root locus** tab;
- Set **Roots** computation type;
- Select an identifier from the list (the identifier pp_k parameterizing the gain factor K is selected in Figure 1.216);
- Set numerical bounds for the identifier as well as the discretization number, i.e. the number of computations for uniform division of the interval;
- Run computation by the button ;
- Draw the root locus by clicking the button , Figure 1.217.

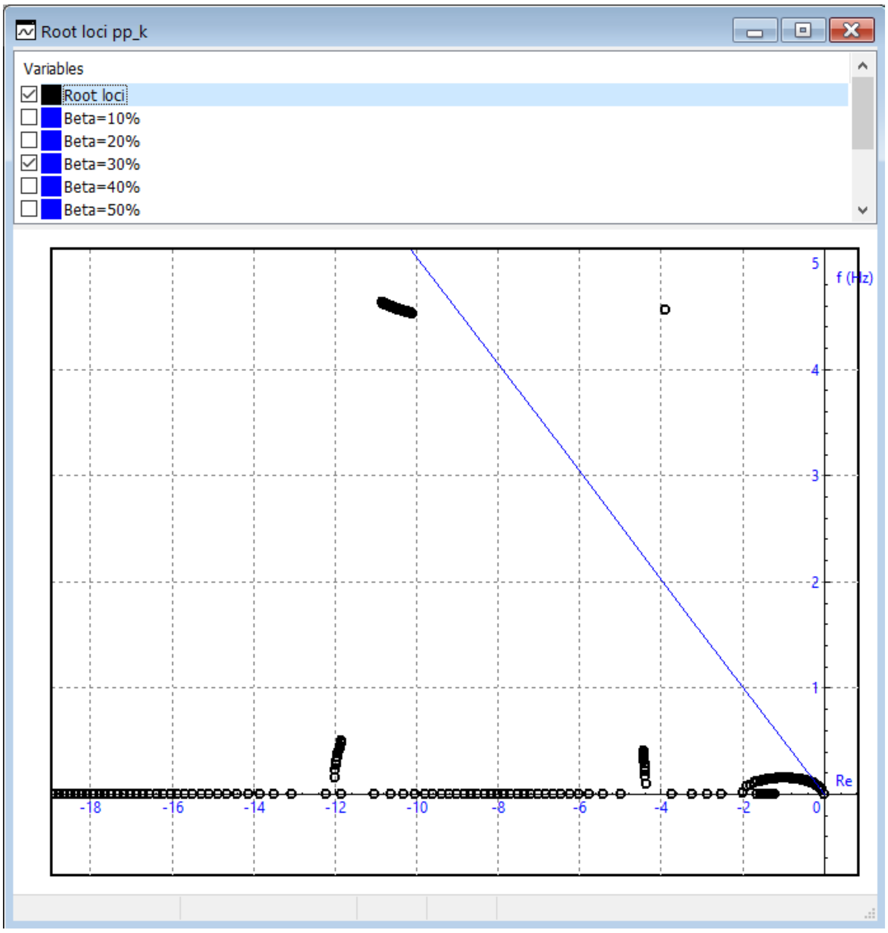


Figure 1.217. Example of root locus

1.10.3. Example of analysis of linearized equations of a simplified car model

We will consider in this section the linearized equations of a simplified car model and some results on the study of the influence of control parameters.

1.10.3.1. Simplified model of a car

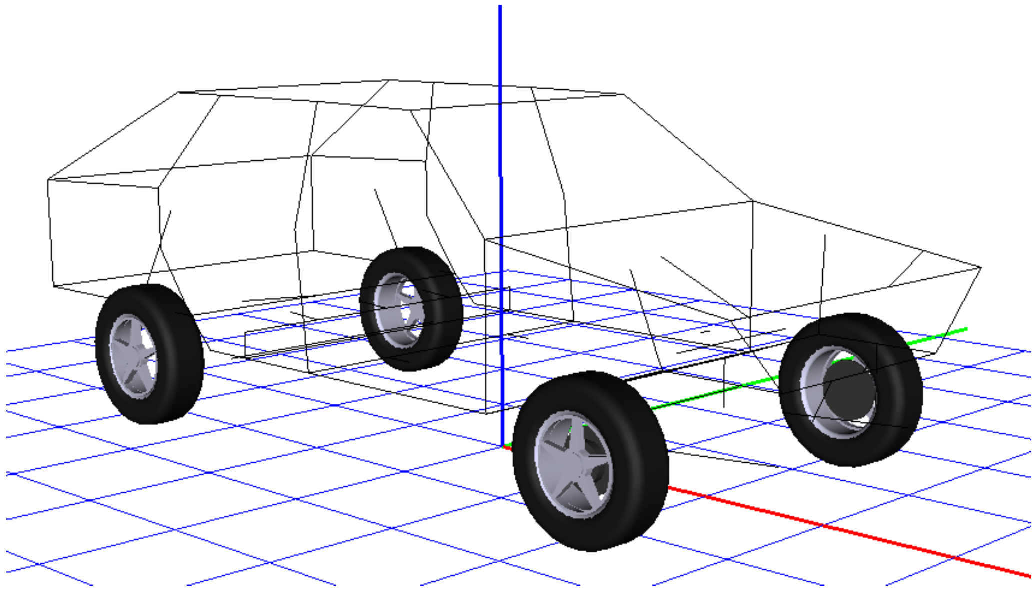


Figure 1.218. Simplified car model

The simplified car model that we describe here is located in the directory {Data UM}\SAMPLES\Automotive\Car_simple. The model has only 12 degrees of freedom, which makes it possible to obtain an explicit form of the matrices of linearized equations. In particular, the model lacks suspension, steering wheel and other elements. There is a car body, four wheels and two steering knuckles, allowing the control system to turn the steer wheels and follow the trajectories of motion with the help of the control model.

Let us consider the main elements of the model. A list of bodies, the parameterization of their inertia parameters and the corresponding numerical values are given in Table 1.9.

Table 1.9

Bodies and inertia parameters

Name	Mass, kg	Moments of inertia, $kg \cdot m^2$		
Car body	mcarbody 1000	icarx 400	icary 1200	icarz 1200
Steering knuckle left Steering knuckle right	m_knuckle 10	iknucklex 0.15	iknuckley 0.15	iknucklez 0.1
Wheel FL Wheel FR Wheel RL Wheel RR	mwheel 20	iwheelx 0.25	iwheely 0.5	iwheelx 0.25

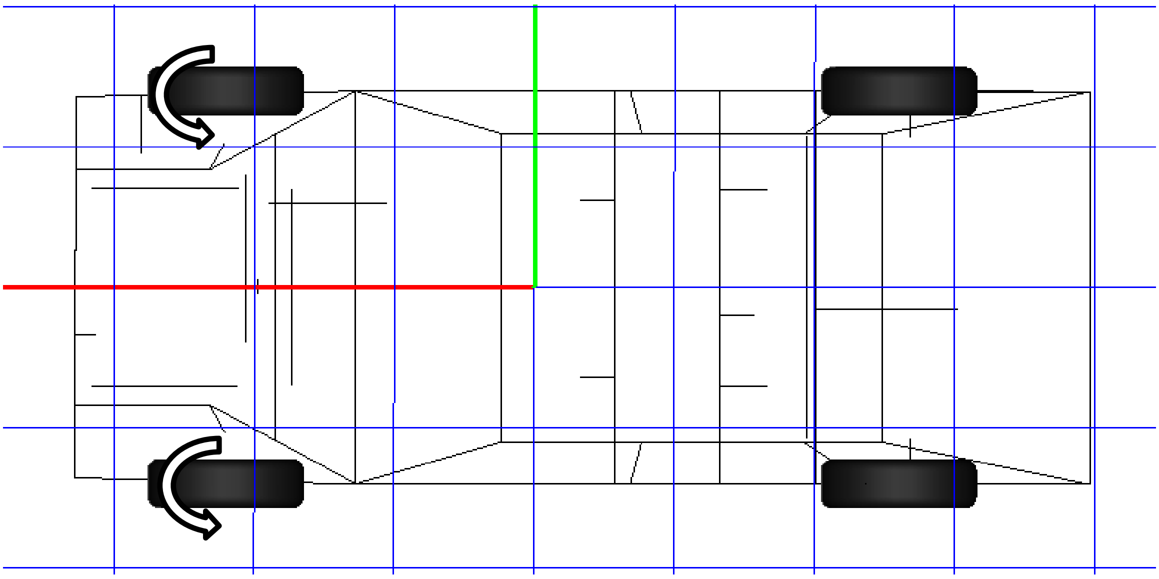


Figure 1.219. Angular degrees of freedom of steering knuckles

Joints introduce degrees of freedom for the bodies.

jCar body is a joint with 6 d.o.f. for the car body. The identifier zc=0.7m sets the vertical position of the center of gravity.

jCarbody - knuckle left, jCarbody - knuckle right; these rotational joints introduce the corresponding angular degrees of freedom of the steering knuckles relative to the car body. In the simplified setting, the rotation of the knuckles is set about the vertical axis, Figure 1.219.

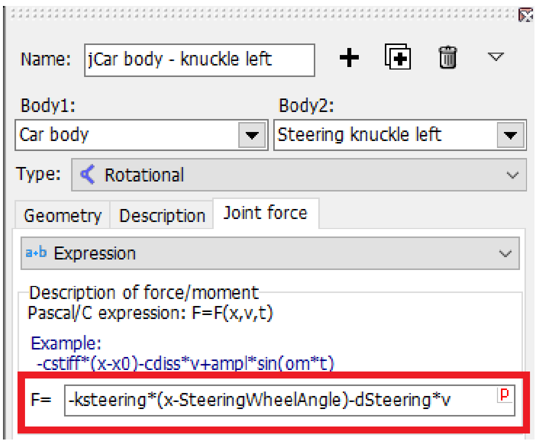


Figure 1.220. Joint torque implementing the controlled rotation of the knuckle

Linear elastic-dissipative torques in the joints **jCarbody - knuckle left, jCarbody - knuckle right** are used for steer control of the car, Figure 1.220. The torque mathematical model is

$$M_s = -k_s(\delta - \delta_s(t)) - d_s\dot{\delta}, \quad (1.4)$$

where k_s, d_s are the stiffness and damping constants, δ is the angle of rotation of the knuckle, and, $\delta_s(t)$ is the steering angle determined by the driver model. The torque implements the steer control.

The stiffness k_s and damping d_s constants in the joint torque models are parameterized by the identifiers *ksteering*, *dsteering*, the steering angle is set by the identifier *SteeringWheelAngle*. Thus, the recommended identifiers are used, Sect. 1.9.1.2.3 "*Identification of steering*".

Five identifiers parameterize the driver model, Figure 1.215, right. Assigning the identifiers to the corresponding parameters is done on the **Road vehicle | Identification** tab, Sect. 1.9.1.2.3 "*Identification of steering*".

1.10.3.2. Equations of motion for simplified car model

The parameters of the simplified car model, for which the calculations were made, as well as the corresponding identifiers and numerical values are presented in Table 1.10.

Table 1.10

Parameters of car model

Parameter	Identifier (expresson)	Value	Comments
M_c	mcarbody	1100	Car body mass (in a simplified example it includes the mass of wheels and steering knuckles), kg
I_{cx},I_{cy},I_{cz}	iCarx, iCary, iCarz	400, 1200, 1200	Moments of inertia of the car body relative to the longitudinal, lateral and vertical axis, kg·m^2
M_k	mknuckle	0	Mass of steering knuckle, kg
I_{kx},I_{ky},I_{kz}	iknucklex, iknuckley, iknucklexz	0.15, 0.15, 0.1	Moments of inertia of steering knuckle, kg·m^2
M_w	mwheel	0	Mass of wheel, kg
I_{wx},I_{wy}	iwheelx, iwheely	0.25, 0.5	Moments of inertia of a wheel relative to the longitudinal and lateral axes, kg·m^2
M_Σ	$M_c + 2M_k + 4M_w$	1100	Total mass of car, kg
I_x,I_y, I_z	$I_x = I_{cx} + 2I_{kx} + 4I_{wx}$ $I_y = I_{cy} + 2I_{ky} + 4I_{wy}$ $I_z = I_{cz} + 2I_{kz} + 4I_{wz}$	401.3 1202.3 1201.2	Total moment of inertia for car, kg·m^2
a, b	a, b	1.2	Longitudinal positions of the front a and rear b axles relative to the center of gravity of car body, m
w	w	0.7	Wheel semibase: a half of distance between centers of front or rear wheels as well as be-

			tween the knuckle rotation axes, m
z_{c0},	zc	0.7	Vertical coordinate of center of gravity of car body at zero tire deflection, m
Δz	$M_\Sigma g/4K_z$	0.0117	Static tire deflections for a=b, m
z_c	$z_{c0}-\Delta z$	0.69568	Vertical coordinate of center of gravity of car in equilibrium, m
r_0		0.28	Undeformed wheel radius, m
r	$r_0-\Delta z$		Wheel rolling radius
k_z	Kz	230 000	Vertical stiffness of tire, N/m
β_z	BetaZ	0.5	Vertical damping ratio for tires
d_z	$2\beta_z\sqrt{M_w k_z}$	2145	Vertical tire damping, Ns/m
c_x	Cx	57000	Longitudinal tire stiffness, N
c_y	Cy	27500	Cornering tire stiffness, N/rad
c_a	$\left.\frac{\partial M_z}{\partial s_y}\right\|_{s_y=0}$	1833	Tangent of the angle of inclination of the curve of the aligning moment, Nm
k_s d_s	ksteering dsteering	26000 48.4	Stiffness and damping coefficients in the steering mechanism (1.4), Nm/pad, Nms/rad
T_p, K, K_2, K_d, K_I	pp_t, pp_k, pp_k2, pp_d, pp_i		Drive model parameters (1.2)
v_0	v0		Vehicle speed
ω_0	v_0/r		Angular velocity of wheels

List of 13 coordinates includes

- Tree Cartesian coordinates of the car body center of gravity x, y, z;
- Tree orientation angles for the car body in the sequence 3,2,1 γ, β, α (i.e. the first turn is about *z* axis, the second one about *y*, and the third about *z*);
- Two angles of rotation of the steering knuckles about the vertical axis δ_l, δ_r (subscripts *l* for the left and *r* for the right sides);
- Four angles of rotation of the wheels $\varphi_{fl}, \varphi_{fr}, \varphi_{rl}, \varphi_{rr}$; here and below in the force designations, the first subscript denotes *f* - front, *r* - rear, the second subscript *l* - left, *r* – right;
- Additional variable I_{dy} corresponds to the integral term in the driver control model (1.1) $I_{dy} = \int_0^t y_d(\tau)d\tau$, which is computed as a solution of the differential equation $\dot{I}_{dy} = y_d(\tau)$.

Consider the matrices of equations of motion of a car, the model of which is described in Sect. 12.10.3.1. "*Simplified car model*". The option **Set zero velocities** is disabled, that is, the speed is different from zero, and the linearized forces of the interaction of tires with the road (1.3) are taken into account.

First, we present the linearized equations of motion of the simplified car model. In these equations, we neglect the terms corresponding to air drag

$$M_\Sigma \ddot{x} - m_{26}\ddot{\beta} = F_{xfl} + F_{xfr} + F_{xrl} + F_{xrr},$$
$$M_\Sigma \ddot{y} + m_{24}\ddot{\gamma} + m_{26}\ddot{\alpha} = F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr},$$
$$M_\Sigma \ddot{z} - m_{24}\ddot{\beta} = -M_\Sigma g + F_{zfl}+F_{zfr}+F_{zrl}+F_{zrr},$$
$$I_z\ddot{\gamma} + m_{24}\ddot{y} + (I_{kz} + I_{wx})\left(\ddot{\delta}_l + \ddot{\delta}_r\right) =$$
$$= w\left(-F_{xfl} + F_{xfr} - F_{xrl} + F_{xrr}\right) + a\left(F_{yfl} + F_{yfr}\right) - b\left(F_{yrl} + F_{yrr}\right) + M_{afl} + M_{afr} + M_{arl} + M_{arr} - 4I_{wx}\omega_0\dot{\alpha},$$
$$I_y\ddot{\beta} - m_{26}\ddot{x} + I_{wy}\ddot{\varphi}_{fl} + I_{wy}\ddot{\varphi}_{fr} + I_{wy}\ddot{\varphi}_{rl} + I_{wy}\ddot{\varphi}_{rr} =$$
$$= M_\Sigma g(z_c - r)\beta - a\left(F_{zfl} + F_{zfr}\right) + b(F_{zrl} + F_{zrr}) - z_c\left(F_{xfl} + F_{xfr}+F_{xrl} + F_{xrr}\right)$$
$$I_x\ddot{\alpha} = M_\Sigma g z_c\alpha + w\left(F_{zfl} - F_{zfr}+F_{zrl} - F_{zrr}\right) + z_c\left(F_{yfl} + F_{yfr}+F_{yrl} + F_{yrr}\right) + I_{wy}\omega_0\left(4\dot{\gamma} + \dot{\delta}_l + \dot{\delta}_r\right),$$
$$I_{kz}\ddot{\delta}_l + (I_{kz} + I_{wx})\ddot{\gamma} = -k_s\left(\delta_l - \delta_s(t)\right) - d_s\dot{\delta}_l + M_{afl} - I_{wy}\omega_0\dot{\alpha},$$
$$I_{kz}\ddot{\delta}_r + (I_{kz} + I_{wx})\ddot{\gamma} = -k_s\left(\delta_r - \delta_s(t)\right) - d_s\dot{\delta}_r + M_{afr} - I_{wy}\omega_0\dot{\alpha},$$
$$I_{wy}\ddot{\varphi}_{fl} + I_{wy}\ddot{\beta} = -rF_{xfl} - M^r_{fl},$$
$$I_{wy}\ddot{\varphi}_{fr} + I_{wy}\ddot{\beta} = -rF_{xfr} - M^r_{fr},$$
$$I_{wy}\ddot{\varphi}_{rl} + I_{wy}\ddot{\beta} = -rF_{xrl} - M^r_{rl},$$
$$I_{wy}\ddot{\varphi}_{rr} + I_{wy}\ddot{\beta} = -rF_{xrr} - M^r_{rr},$$
$$\dot{I}_{dy} = -y - a\gamma - (z_c - r)\alpha.$$

Here the inertia parameters are introduced

$$m_{26} = (z_c - r_w)(2M_k + 4M_w), \qquad m_{24} = 2aM_k,$$
$$I_z = I_{cz} + 4I_{wz} + 2I_{kz} + 2(a^2 + w^2)(M_w + M_k) + 2(b^2 + w^2)M_w,$$
$$I_y = I_{cy} + 4I_{wy} + 2I_{ky} + 2(a^2 + (z_c - r_w)^2)(M_w + M_k) + 2(b^2 + (z_c - r_w)^2)M_w,$$
$$I_x = I_{cx} + 4I_{wx} + 2I_{kx} + (w^2 + (z_c - r_w)^2)(4M_w + 2M_k).$$

The steering angle for the front wheels $\delta_s(t)$ corresponding to the driver control is

$$\delta_s(t) = -(K + K_2)(y + a\gamma + (z_c - r)\alpha) - \left(KT_p + K_d\right)\dot{y}+K_I I_{dy}.$$

In the linear formulation, the longitudinal and side tire forces have the following form:

$$F_{xfl} = F_{x0} - c_x\left((a\beta - z)/r + \left(\dot{x} - z_c\dot{\beta} - r\dot{\varphi}_{fl} - w\dot{\gamma}\right)/v_0\right),$$
$$F_{xfr} = F_{x0} - c_x\left((a\beta - z)/r + \left(\dot{x} - z_c\dot{\beta} - r\dot{\varphi}_{fr} + w\dot{\gamma}\right)/v_0\right),$$
$$F_{xrl} = F_{x0} - c_x\left((-a\beta - z)/r + \left(\dot{x} - z_c\dot{\beta} - r\dot{\varphi}_{rl} - w\dot{\gamma}\right)/v_0\right),$$
$$F_{xrr} = F_{x0} - c_x\left((-a\beta - z)/r + \left(\dot{x} - z_c\dot{\beta} - r\dot{\varphi}_{rr} + w\dot{\gamma}\right)/v_0\right).$$
$$F_{yfl} = -c_y(-\gamma - \delta_l + (\dot{y} + z_c\dot{\alpha} + a\dot{\gamma})/v_0),$$
$$F_{yfr} = -c_y(-\gamma - \delta_r + (\dot{y} + z_c\dot{\alpha} + a\dot{\gamma})/v_0),$$
$$F_{yrl} = -c_y(-\gamma + (\dot{y} + z_c\dot{\alpha} - b\dot{\gamma})/v_0),$$
$$F_{yrr} = -c_y(-\gamma + (\dot{y} + z_c\dot{\alpha} - b\dot{\gamma})/v_0),$$

The stationary value of the longitudinal force F_{x0} depends on the air drag force and the rolling friction moments of the wheels $M^r_{fl},\ldots$.

Linearized aligning moments for tire are

$$M_{afl} = c_a(-\gamma - \delta_l + (\dot{y} + z_c\dot{\alpha} + a\dot{\gamma})/v_0),$$
$$M_{afr} = c_a(-\gamma - \delta_r + (\dot{y} + z_c\dot{\alpha} + a\dot{\gamma})/v_0),$$
$$M_{arl} = c_a(-\gamma + (\dot{y} + z_c\dot{\alpha} - b\dot{\gamma})/v_0),$$
$$M_{arr} = c_a(-\gamma + (\dot{y} + z_c\dot{\alpha} - b\dot{\gamma})/v_0).$$

The vertical forces applied to wheels:

$$F_{zfl} = M_\Sigma gb/2(a+b) - k_z(z + w\alpha - a\beta) - d_z(\dot{z} + w\dot{\alpha} - a\dot{\beta}),$$
$$F_{zfr} = M_\Sigma gb/2(a+b) - k_z(z - w\alpha - a\beta) - d_z(\dot{z} - w\dot{\alpha} - a\dot{\beta}),$$
$$F_{zrl} = M_\Sigma ga/2(a+b) - k_z(z + w\alpha + b\beta) - d_z(\dot{z} + w\dot{\alpha} + b\dot{\beta}),$$
$$F_{zrr} = M_\Sigma ga/2(a+b) - k_z(z - w\alpha + a\beta) - d_z(\dot{z} - w\dot{\alpha} + a\dot{\beta}),$$

Let us write the equations of motion in the matrix form.

$$M\ddot{q} + C\dot{q} + Kq = Q_0$$

Matrices of inertia M, coefficients at velocities C and positional forces K are shown in Figure 1.222 - Figure 1.224 both in analytical and numerical forms for the values of the model parameters given in Table 1.10 for speed v_0=10m/s. The control coefficient values for numerical matrices are

$$T_p = 1, K = 0.075, K_2 = 0, K_d = 0, K_I = 0.1$$

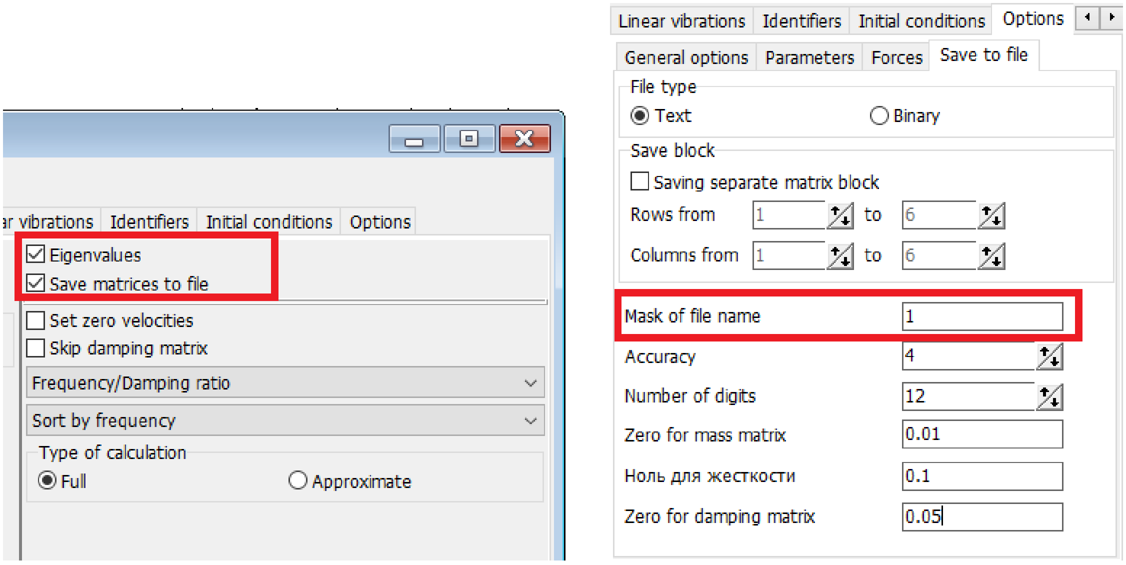


Figure 1.221. Writing matrices of linearized equations in files while computation of eigenvalues

The numerical values of the matrices were obtained by the simulation program in the linear analysis mode. To do this, the user must enable the mode of saving matrices before start computation of eigenvalues and specify the template for the name of the matrices in the linear analysis options, Figure 1.221.

$$M=\begin{pmatrix}
M_\Sigma & 0 & 0 & 0 & -m_{26} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & M_\Sigma & 0 & m_{24} & 0 & m_{26} & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & M_\Sigma & 0 & -m_{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & m_{24} & 0 & I_z & 0 & 0 & I_{wz}+I_{kz} & I_{wz}+I_{kz} & 0 & 0 & 0 & 0 & 0\\
-m_{26} & 0 & -m_{24} & 0 & I_y & 0 & 0 & 0 & I_{wy} & I_{wy} & I_{wy} & I_{wy} & 0\\
0 & m_{26} & 0 & 0 & 0 & I_x & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & I_{wz}+I_{kz} & 0 & 0 & I_{kz} & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & I_{wz}+I_{kz} & 0 & 0 & 0 & I_{kz} & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & I_{wy} & 0 & 0 & 0 & I_{wy} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & I_{wy} & 0 & 0 & 0 & 0 & I_{wy} & 0 & 0 & 0\\
0 & 0 & 0 & 0 & I_{wy} & 0 & 0 & 0 & 0 & 0 & I_{wy} & 0 & 0\\
0 & 0 & 0 & 0 & I_{wy} & 0 & 0 & 0 & 0 & 0 & 0 & I_{wy} & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1100	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1100	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1100	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1201	0	0	0.35	0.35	0	0	0	0	0
5	0	0	0	0	1202	0	0	0	0.5	0.5	0.5	0.5	0
6	0	0	0	0	0	401.3	0	0	0	0	0	0	0
7	0	0	0	0.35	0	0	0.35	0	0	0	0	0	0
8	0	0	0	0.35	0	0	0	0.35	0	0	0	0	0
9	0	0	0	0	0.5	0	0	0	0.5	0	0	0	0
10	0	0	0	0	0.5	0	0	0	0	0.5	0	0	0
11	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0
12	0	0	0	0	0.5	0	0	0	0	0	0	0.5	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 1.222. Mass matrix in analytical and numerical forms

$$C=\begin{pmatrix}
4C_x & 0 & 0 & 0 & -4C_xz_c & 0 & 0 & 0 & -rC_x & -rC_x & -rC_x & -rC_x & 0\\
0 & 4C_y & 0 & 2C_y(a-b) & 0 & 4C_yz_c & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 4d_z & 0 & 2(b-a)d_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & c_{42} & 0 & c_{44} & 0 & c_{46} & 0 & 0 & rwC_x & -rwC_x & rwC_x & -rwC_x & 0\\
-4C_xz_c & 0 & 2(b-a)d_z & 0 & c_{55} & 0 & 0 & 0 & I_{wy} & rz_cC_x & rz_cC_x & rz_cC_x & 0\\
0 & 4C_yz_c & 0 & c_{64} & 0 & 4C_zw^2+4C_yz_c^2 & -I_{wy}\omega_0 & -I_{wy}\omega_0 & 0 & 0 & 0 & 0 & 0\\
0 & c_{72} & 0 & -aC_a & 0 & -z_cC_a+I_{wy}\omega_0 & d_s & 0 & 0 & 0 & 0 & 0 & -K_Ik_s\\
0 & c_{72} & 0 & -aC_a & 0 & -z_cC_a+I_{wy}\omega_0 & 0 & d_s & 0 & 0 & 0 & 0 & -K_Ik_s\\
-rC_x & 0 & 0 & rwC_x & rz_cC_x & 0 & 0 & 0 & r^2C_x & 0 & 0 & 0 & 0\\
-rC_x & 0 & 0 & -rwC_x & rz_cC_x & 0 & 0 & 0 & 0 & r^2C_x & 0 & 0 & 0\\
-rC_x & 0 & 0 & rwC_x & rz_cC_x & 0 & 0 & 0 & 0 & 0 & r^2C_x & 0 & 0\\
-rC_x & 0 & 0 & -rwC_x & rz_cC_x & 0 & 0 & 0 & 0 & 0 & 0 & r^2C_x & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

$$C_x=c_x/v_0, C_y=c_y/v_0, C_a=c_a/v_0$$

$$c_{44}=2C_y(a^2+b^2)+4C_x-2(a-b)C_a,\qquad c_{42}=2C_y(a-b)-4C_a,\qquad c_{55}=2d_z(a^2+b^2)+4C_xz_c^2$$

$$c_{46}=2C_yz_c(a-b)-4C_az_c+4I_{wy}\omega_0,\qquad c_{64}=2C_yz_c(a-b)-4I_{wy}\omega_0,\qquad c_{72}=-C_a+(KT_p+K_d)k_s$$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2.28E4	0	0	-0.01117	-1.569E4	0	0	0	-1529	-1529	-1529	-1529	0
2	0	1.1E4	0	0	0	7571	0	0	0	0	0	0	0
3	0	0	8579	0	0	0	0	0	0	0	0	0	0
4	0	-733.3	0	2.701E4	0	-430.2	0	0	1070	-1070	1070	-1070	0
5	-1.569E4	0	0	0	2.315E4	0	0	0	1052	1052	1052	1052	0
6	0	7571	0	-74.54	0	9415	-18.64	-18.64	0	0	0	0	0
7	0	1767	0	-220	0	-107.5	48.37	0	0	0	0	0	-2600
8	0	1767	0	-220	0	-107.5	0	48.37	0	0	0	0	-2600
9	-1529	0	0	1070	1052	0	0	0	410.2	0	0	0	0
10	-1529	0	0	-1070	1052	0	0	0	0	410.2	0	0	0
11	-1529	0	0	1070	1052	0	0	0	0	0	410.2	0	0
12	-1529	0	0	-1070	1052	0	0	0	0	0	0	410.2	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 1.223. Matrix of coefficients at velocities in analytical and numerical forms

$$K = \begin{pmatrix}
0 & 0 & -4c_x/r & 0 & 2(a-b)c_x/r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -4C_y & 0 & 0 & -c_y & -c_y & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 4k_z & 0 & 2(b-a)k_z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2(b-a)c_y+4c_a & 0 & 4w^2c_x/r & -ac_y+c_a & -ac_y+c_a & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2(b-a)k_z+4c_xz_c/r & 0 & 2(a^2+b^2)k_z-M_\Sigma g(z_c-r) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -4z_cc_y & 0 & 4w^2k_z-M_\Sigma gz_c & -z_cc_y & -z_cc_y & 0 & 0 & 0 & 0 & 0 \\
0 & (K+K_2)k_s & 0 & c_a+a(K+K_2)k_s & 0 & (z_c-r)(K+K_2)k_s & k_s+c_a & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & (K+K_2)k_s & 0 & c_a+a(K+K_2)k_s & 0 & (z_c-r)(K+K_2)k_s & 0 & k_s+c_a & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & c_x & 0 & -ac_x & wc_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & c_x & 0 & -ac_x & -wc_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & c_x & 0 & bc_x & wc_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & c_x & 0 & bc_x & -wc_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & a & 0 & z_c-r & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	-8.498E5	0	0	0	0	0	0	0	0	0	0
2	0	0	0	-1.1E5	0	0	-2.749E4	-2.749E4	0	0	0	0	0
3	0	0	9.2E5	0	1.932	0	0	0	0	0	0	0	0
4	0	0	0	7332	0	4.164E5	-3.116E4	-3.116E4	0	0	0	0	0
5	0	0	5.849E5	0.7845	1.32E6	0	0	0	0	0	0	0	0
6	0	0	0	-7.571E4	0	4.434E5	-1.892E4	-1.892E4	0	0	0	0	0
7	0	1950	0	4173	0	819	2.783E4	0	0	0	0	0	0
8	0	1950	0	4173	0	819	0	2.783E4	0	0	0	0	0
9	0	0	5.699E4	0	-6.838E4	3.989E4	0	0	0	0	0	0	0
10	0	0	5.699E4	0	-6.838E4	-3.989E4	0	0	0	0	0	0	0
11	0	0	5.699E4	0	6.839E4	3.989E4	0	0	0	0	0	0	0
12	0	0	5.699E4	0	6.839E4	-3.989E4	0	0	0	0	0	0	0
13	0	1	0	1.2	0	0.42	0	0	0	0	0	0	0

Figure 1.224. Matrix of positional forces in analytical and numerical forms

1.10.3.3. Root locus: dependence of eigenvalues on driver control parameters

In this section, we consider how the root locus can be used to analyze the influence of the driver model parameters on the stability and oscillations of the car when driving along a straight section of the road.

The motion equations derived in the previous section show that the controlled motion of the car in a straight road depends on three parameters

$$K + K_2, \quad KT_p + K_d, \quad K_I$$

In fact, this means that we can limit ourselves to studying the influence of only the parameters $K, \ K_d, \ K_I$.

1.10.3.3.1. Study of influence of parameter *K*

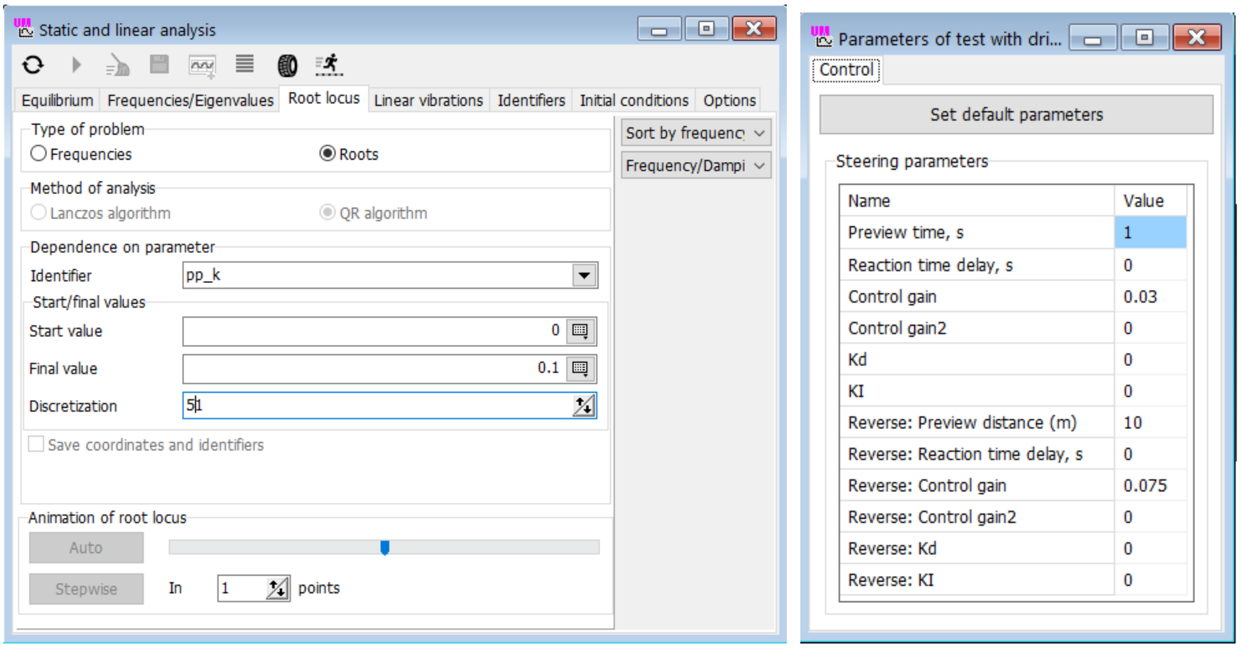


Figure 1.225. Settings for computation of eigenvalue dependence on gain factor *K*

To compute a root locus for the control parameter *K*, open the **Root locus** tab, set the calculation type **Roots**, select the identifier *pp_k* corresponding to the control parameter *K*, and set the interval of change and discretization for this identifier, Figure 1.225 left.

Set the values of other control parameters in the window that appears by clicking the button on the toolbar of the SLA window, Figure 1.225, right. The calculation is started by clicking on the button . After the calculation is completed, root locus, that is, the dependence of the roots of the characteristic equation of linearized equations on the parameter, can be drawn by clicking on the button .

An example of calculation at a speed of 10 m/s is shown in Figure 1.226. The real part of the roots corresponds to the abscissa axis, the imaginary part divided by 2π, that is, the frequency in Hertz, corresponds to the ordinate axis.

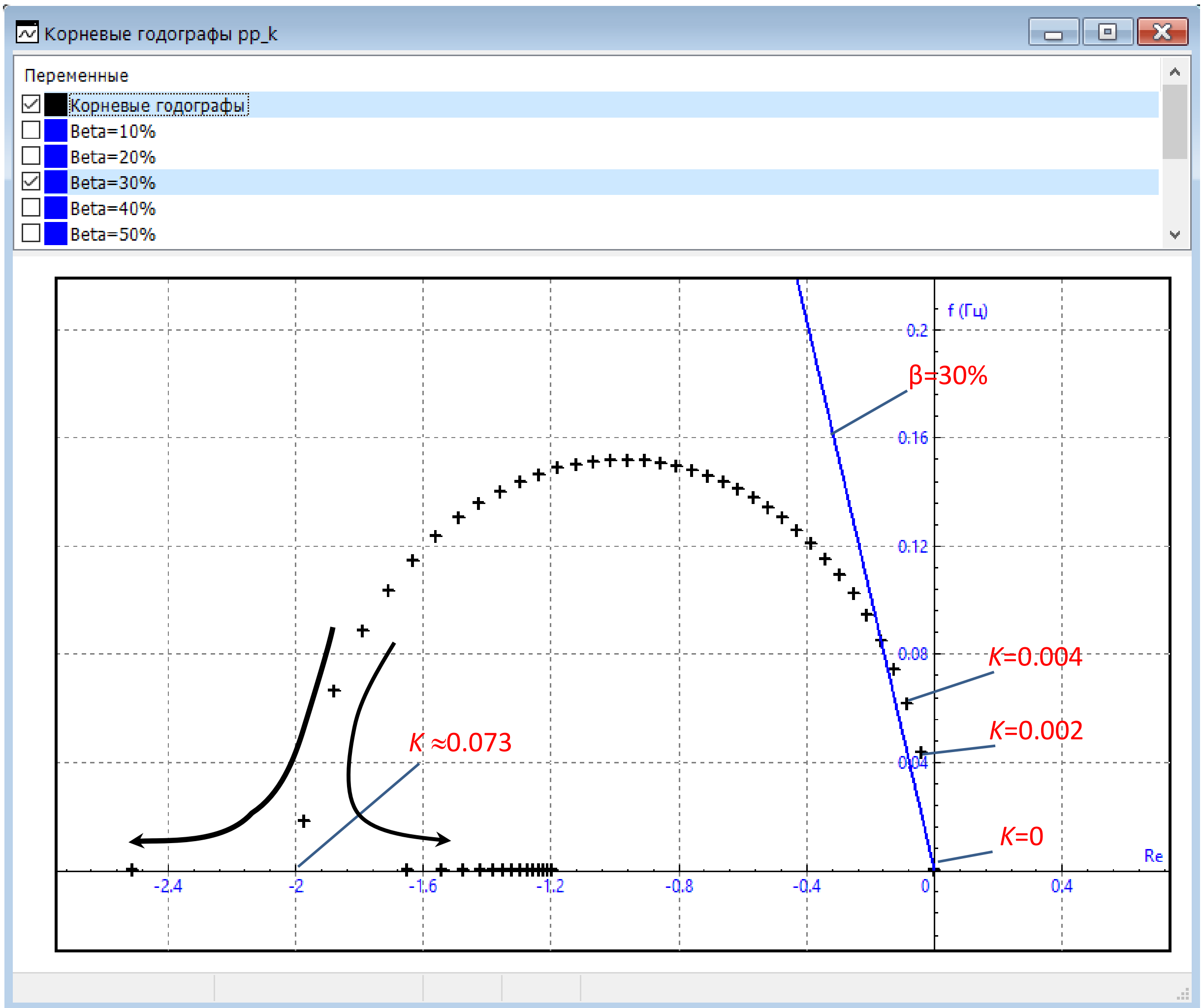


Figure 1.226. Fragment of root locus for parameter *K*

The figure shows a fragment of the root locus near the origin. A curve shows a pair of eigenvalues that correspond to the wobbling of the car due to the influence of control. When the *K*=0, that is, in the absence of control, the corresponding eigenvalues are zero, that is, the motion is unstable. A pair of complex conjugate roots with a negative real part appears for small non-zero values of *K*, that is, the motion is asymptotically stable and has the character of damped oscillations.

With the further increase in the control coefficient, the real part of the root moves in the negative direction, and the frequency first increases and then decreases. The maximum value of the oscillation frequency approximately corresponds to 1 rad/s. At the value of *K*≈0.73, the complex roots merge (the value is approximately equal to -2) and a pair of real negative roots appears, one of which shifts to the negative region, and the other tends to a limiting value of approximately -1 as *K* increases. The motion is asymptotically stable, and is not oscillatory (overdamped) in nature.

Let us compare this result of the linear analysis with the simulation of the controlled motion of the car in a straight line. Let us consider the process of stabilization of the initial deviation of the car from the ideal position: the model is given an initial rotation around the vertical axis by 0.01 radians. The speed is 10m/s. As a simulation result, consider the time dependence of the deviation of the central point between the front wheels from the given rectilinear trajectory. Figure 1.227 shows plots of this variable for three control options. The values of the control coefficient and the roots corresponding to it on the root locus in Figure 1.226 are

K = 0.008: frequency 0.0852 Hz (time period 11.7c), damping ratio 30%;

K = 0.02: frequency 0.126 Hz (time period 7.9c), damping ratio 47.7%;

K = 0.074: overdamped, the root with the minimal absolute value of is -1.65.

It can be seen from the figure that the period and decrement for damped oscillations at *K* = 0.008, 0.02 correspond to the roots. At *K* = 0.074, we have an overdamped process.

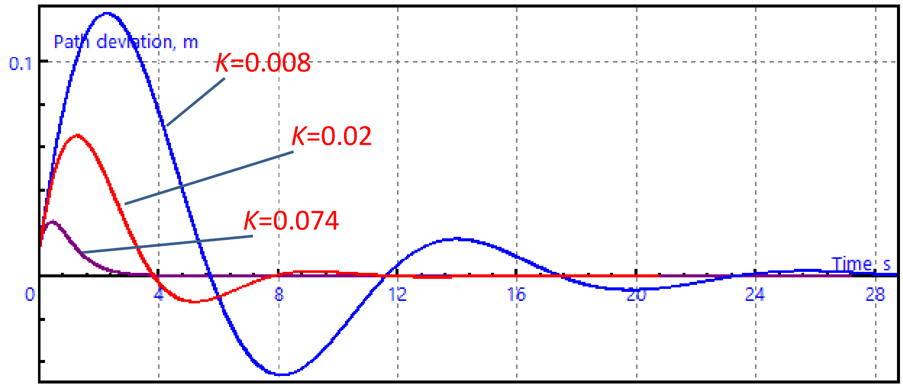


Figure 1.227. Car wobbling for different values of gain *K*

To repeat these simulation results, load the model {Data UM}\SAMPLES\Automotive\Car simple into the UM Simulation program. Read the full configuration using the main menu command **File | Load configuration | Control compare**. When reading the configuration, the value K = 0.008 is set.

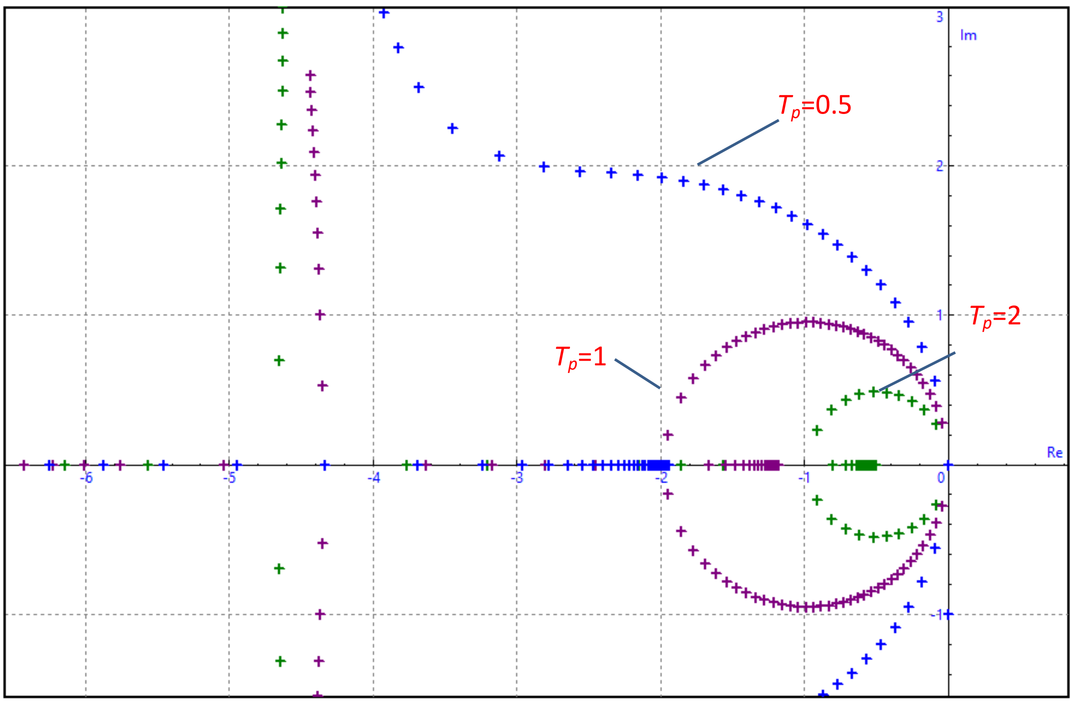


Figure 1.228. Fragment of root loci for different values of preview time T_p

Root locus for a car at speed of 10 m/s for various values of the prediction time T_p are shown in Figure 1.228. Please note that in Figure 1.226 and Figure 1.228, the ordinates have different dimensions: in the first case, this is the frequency in Hz, and in the second, the imaginary part of the root in rad/s, that is, the value is 2π times larger.

There is an analogy of the obtained results with the root locus of a linear oscillator with damping proportional to stiffness, see Sect. 1.10.3.3.5 "*Root locus of a linear oscillator with damping proportional to the stiffness coefficient*". The results are close for prediction times of the order of 1 s and more, and for smaller values of T_p, differences are observed at values of the control coefficient K>0.1, which are not used in practice.

Let us consider the influence of speed on the stability of the control system.

Figure 1.229 shows the root locus K∈[0,0.1] at various vehicle speeds. From this figure, an important conclusion follows. **A loss of vehicle stability is possible with an increase in speed**, which is characterized by the appearance of roots with a positive real part. For this car model, the instability appears at speeds greater than a certain critical speed value (about 30 m/s) with an increase in the control coefficient. At a speed of 30 m/s, the instability take place for K>0.054, and at a speed of 40 m/s, for K>0.019.

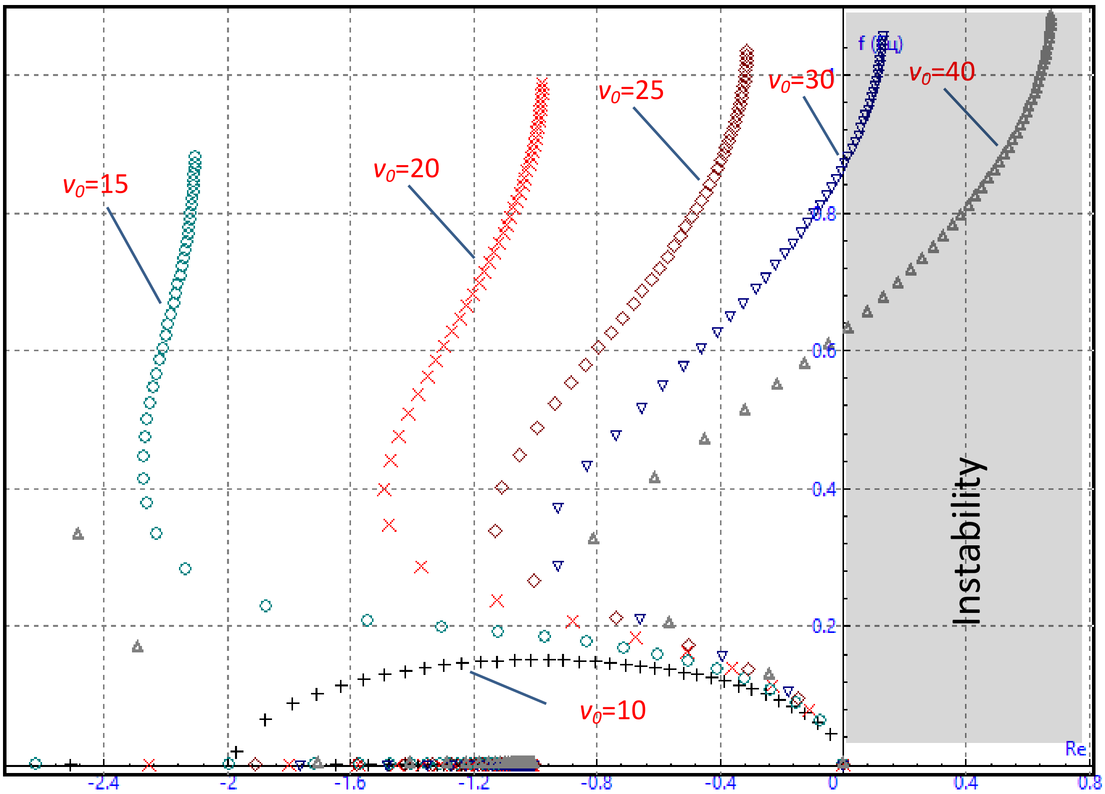


Figure 1.229. Root loci for different speeds

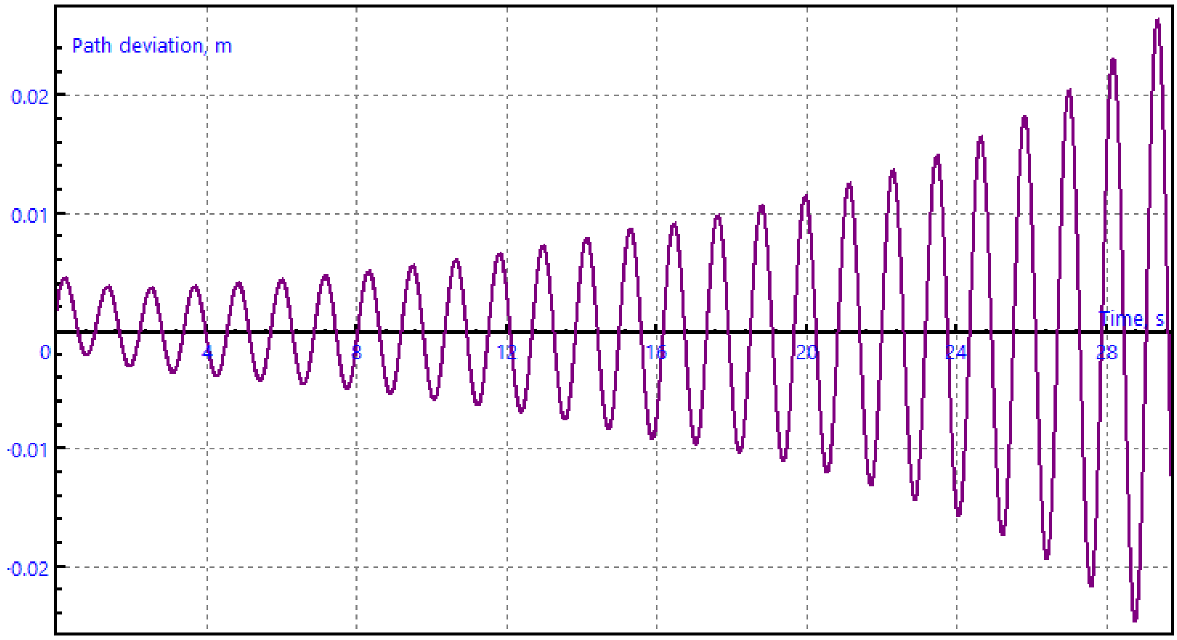


Figure 1.230. Unstable motion of car for speed 30 m/s, $K = 0.058$

Thus, at high vehicle speeds, the value of the control coefficient K should be reduced. For this purpose a preliminary study using the root locus is required.

Note that in a small neighborhood of the instability boundary, the accuracy of the numerical integration method can affect the oscillations; therefore, in order to obtain results similar to the graph in Figure 1.230, it is recommended to increase the accuracy of the numerical method. It is also recommended to set the value of the minimum number of iterations greater than one in the settings of the numerical methods.

1.10.3.3.2. Influence of parameter K_d

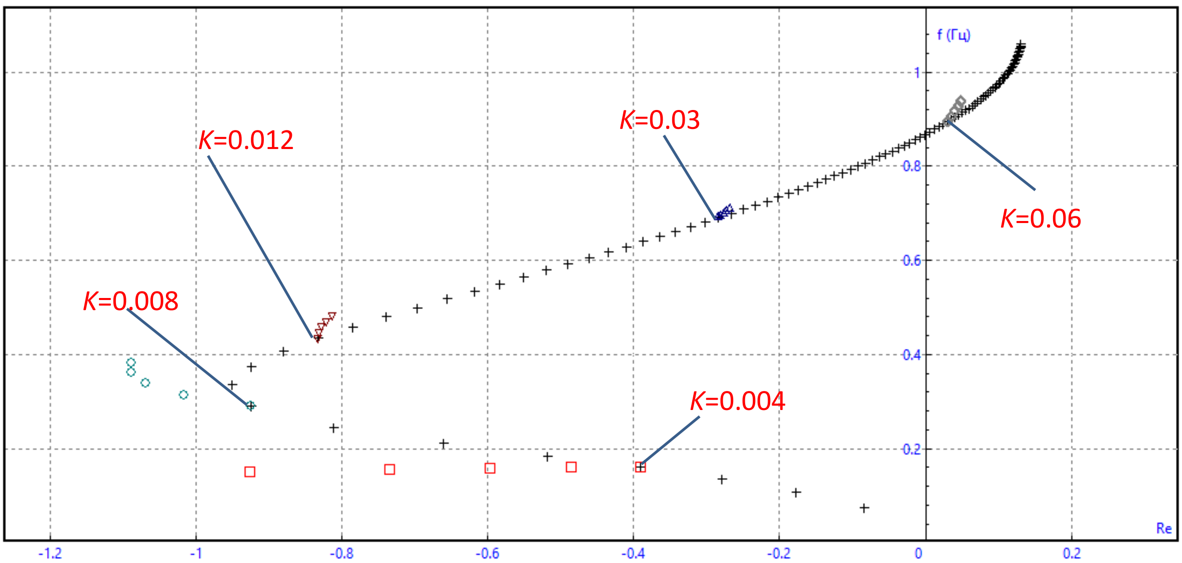


Figure 1.231. Influence of parameter K_d on root locus

To assess the influence of the control parameter K_d, consider the root locus for K at a speed of 30 m/s (see Figure 1.229) and append it the dependences of the roots on K_d for some K, Figure 1.231. For all values of K, except for K=0.06, the interval of change $K_d \in [0, 0.001]$ is set, for the last value $K_d \in [0, 0.01]$.

The analysis of this plot shows that for small values of K, the parameter K_d increases the damping and shifts the roots to the left. As K increases, the effect weakens. In particular, the parameter K_d **cannot stabilize the control**, that is, it cannot shift the roots from the region of instability to the region of stability.

1.10.3.3.3. Influence of parameter K_I

The integral control term is usually used in control systems to remove stationary errors in controlled process in the presence of constant disturbances. For example, if a lateral force (for example, resistance from a crosswind) constantly acts during the motion of a car, then in the absence of an integral term, the controlled movement tends to a non-zero value of the lateral coordinate, that is, there exists a stationary error. Adding an integral term ensures that the coordinate tends to zero.

Figure 1.232 shows the results of simulation of a car motion under the action of a constant lateral force of 300N applied to the center of mass in the absence and presence of an integral term. At the initial moment of time, the car is turned around the vertical axis by 0.1 rad. To repeat these simulation results, you should load the {Data UM}\SAMPLES\Automotive\Car simple model into the UM Simulation program and read the full configuration using the main menu command **File | Load configuration | Integrated control**.

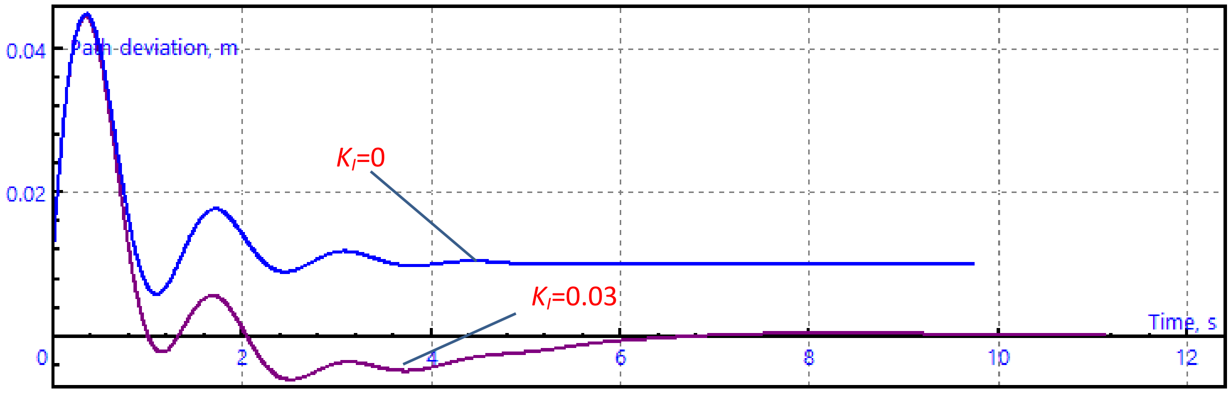


Figure 1.232. Influence of integral term on stationary error in presence of lateral force

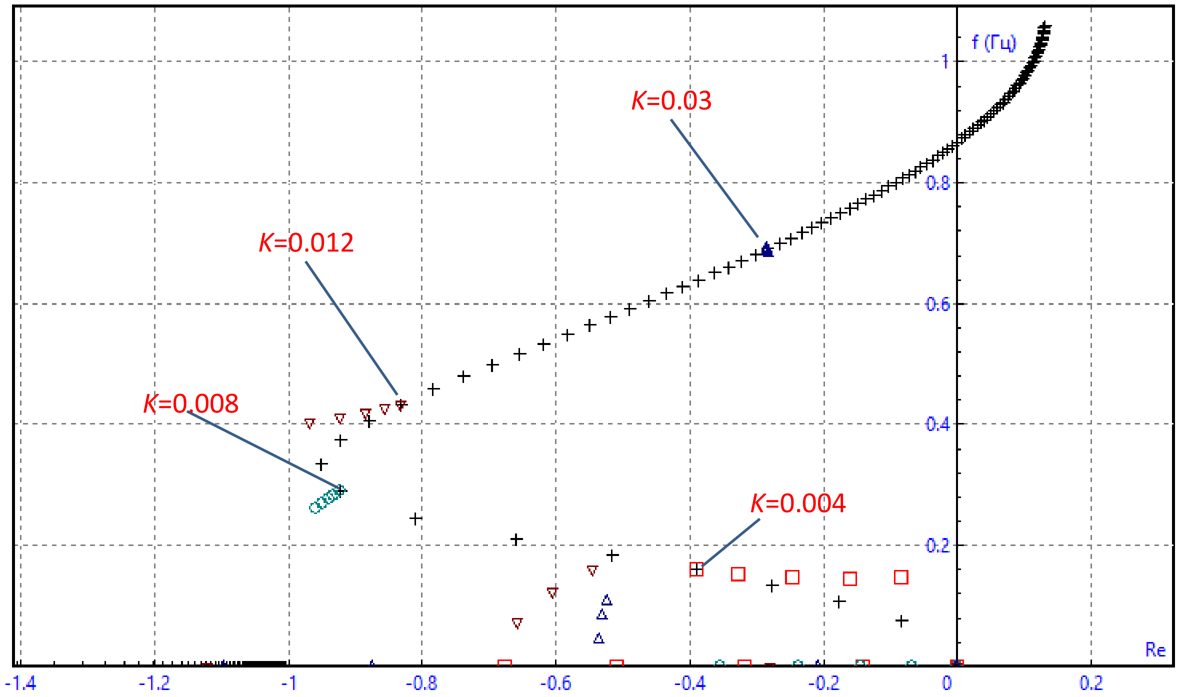


Figure 1.233. Influence of parameter K_I on root locus

To assess the influence of the control parameter K_I, consider the root locus for K at a speed of 30 m/s (see Figure 1.229) and append it the dependences of the roots on K_I for some K, Figure 1.233. For values K=0.004, 0.008 the interval of change $K_I \in [0,0.002]$ is set, for K=0.012, 0.03 the values $K_I \in [0,0.02]$.

Analysis of these plots shows that for small values of K, the parameter K_I reduces the damping. As K increases, the effect becomes insignificant. It can also be concluded that the parameter K_I **cannot stabilize the unstable control**.

1.10.3.3.4. Root locus for advanced model of a passenger car

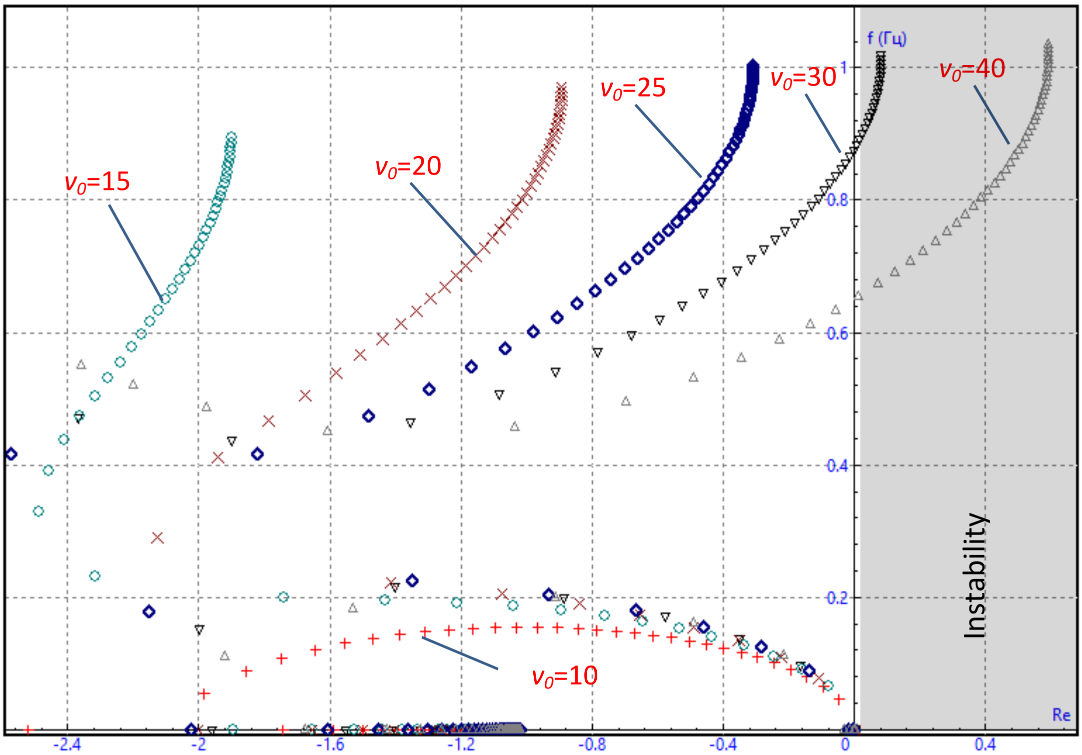


Figure 1.234. Root locus for gain factor $K \in [0, 0.1]$ at different speeds of the model Vaz2109 T

The previous study was done on a simplified 12 DOF passenger car model, so it is important to compare the results to a fully functional model. Let us consider an advanced model of a VAZ 2109 passenger car ({Data UM}\SAMPLES\Automotive\Vaz2109 T) with 36 degrees of freedom. Figure 1.234 shows the root locus for the parameter K at different values of vehicle speeds. Comparison with a similar figure for a simplified model (Figure 1.229) shows a good qualitative and quantitative agreement of the results, including instabilities. Thus, **a simplified car model can be used to assess the influence of control parameters on rectilinear motion**.

1.10.3.3.5. Root locus of a linear oscillator with damping proportional to the stiffness coefficient

This section is auxiliary and helps, using the simplest example, to analyze the root locus, which is qualitatively close to that obtained in the study of the influence of the control parameter *K*, Figure 1.226, Figure 1.228.

A linear oscillator is a mass point *m*, moving in a straight line, attached to a linear spring with a stiffness constant *k* in the presence of linear damping. Consider the case when the damping coefficient *Tk* is proportional to the stiffness, the proportionality factor *T* has the dimension of time. Free motion of the oscillator in this case is described by the equation

$$m\ddot{x} + Tk\dot{x} + kx = 0.$$

The characteristic equation

$$m\lambda^2 + Tk\lambda + k = 0$$

Has roots

$$\lambda_{1,2} = \frac{-Tk \pm \sqrt{T^2k^2 - 4mk}}{2m}$$

If $k < 4m/T^2$, we have a pair of complex conjugates roots

$$\lambda_{1,2} = A(k) \pm iB(k), \qquad A(k) = \frac{-Tk}{2m}, B(k) = \frac{\sqrt{4mk - T^2k^2}}{2m}$$

For the boundary value $k^* = 4m/T^2$, the multiple real root is

$$\lambda_1 = \lambda_2 = -\frac{2}{T}$$

For $k > 4m/T^2$ we obtain two real roots

$$\lambda_1 = \frac{-Tk + \sqrt{T^2k^2 - 4mk}}{2m} = \frac{-4mk}{2m\left(Tk + \sqrt{T^2k^2 - 4mk}\right)} \xrightarrow[k\to\infty]{} -\frac{1}{T},$$

$$\lambda_2 = \frac{-Tk - \sqrt{T^2k^2 - 4mk}}{2m} \xrightarrow[k\to\infty]{} -\infty.$$

The dependence of the roots of the characteristic equation of the oscillator on the parameter *k* in the complex plane (that is, the root locus), is shown in Figure 1.235. The root locus is drawn for the value $T = 1$. Note that a pair of complex conjugate roots lies on a circle of radius $1/T$ centered at the point $(-1/T, 0)$, since the real and imaginary parts of the root satisfy the relation

$$(A(k) + 1/T)^2 + \left(B(k)\right)^2 = 1/T^2.$$

After merging the complex roots, one root moves to the left along the real axis, and the second one moves to the right to the limit value coinciding with the center of the circle.

Comparing the root locus for the simplified car model in Figure 1.226 at $T_p = 1$ and the oscillator, we can conclude on their obvious qualitative and quantitative coincidence. The analogue of the mass of the oscillator is the mass of the car, the coefficient of rigidity *k* is the stiffness $2c_yK$, depending on the control coefficient *K* and the tire side stiffness c_y, and the analogue of the proportionality coefficient *T* is the preview time T_p.

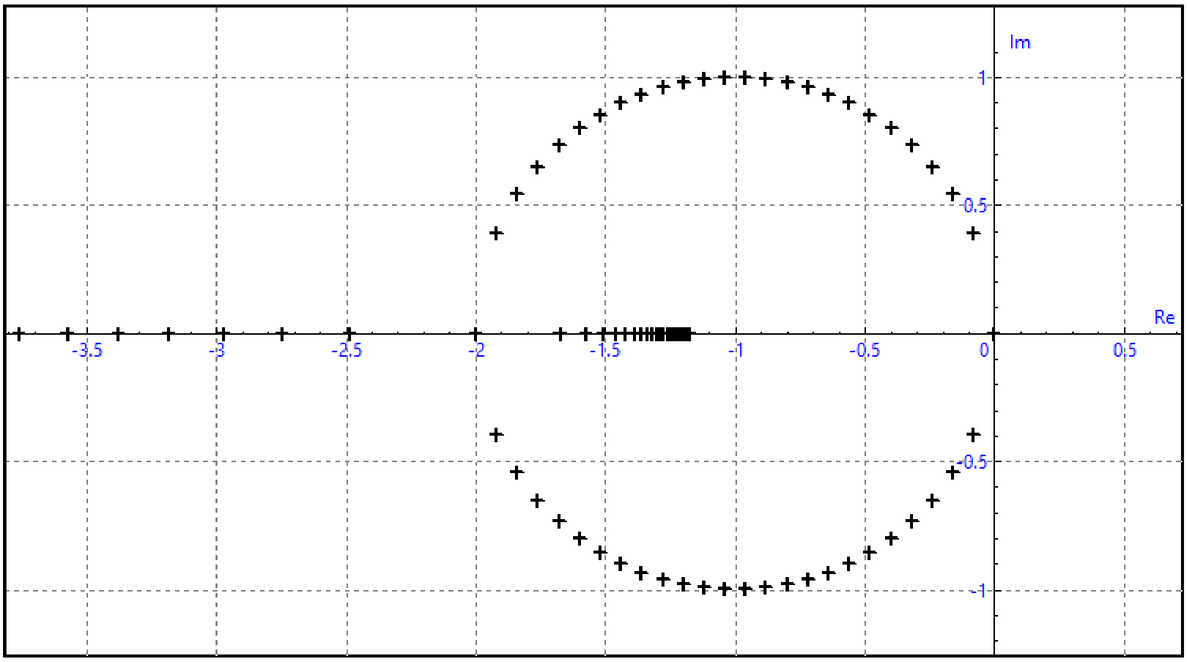


Figure 1.235. Root locus for linear oscillator

1.11. Input satellite photo as background to animation

1.11.1. Creating picture and getting it's sizes

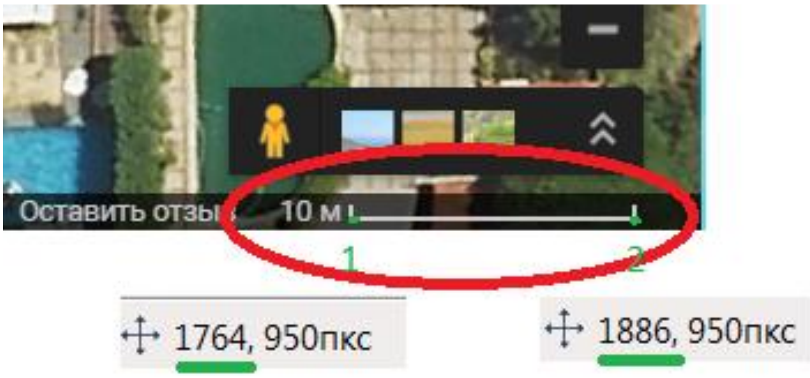
- Choose a place on Google maps

- Rotate view (Ctrl + Mouse move) so that initial vehicle direction will be vertical

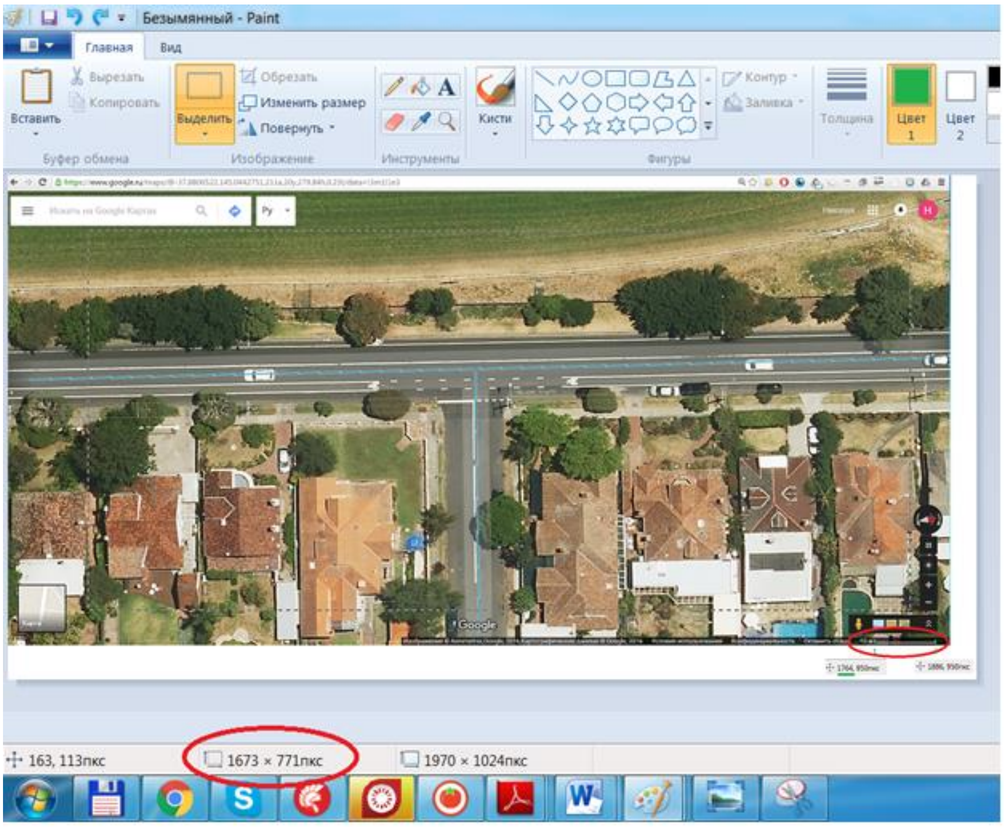
- Make a screenshot of view and paste it into Paint program
- In the right bottom angle of the picture you can see characteristic size of the view.
- Determine pixel length of this size
  - Get pixel coordinate of corners (they are shown in the left bottom corner of the Paint program
  - Get difference between X values (pixelLength = X2 - X1 = 1886 - 1764 = 122 [pix])

- Calculate pixel ratio:

  pixelRatio = pixelLength/realLength = 122 [pix] / 10 [m] = 12.2 [pix/m]



- Select necessary part of map with rectangle. In the left bottom corner you can see pixel width and height of rectangle (pixelWidth = 1673 [pix]; pixelHeight = 771 [pix]). Copy and save selected rectangle as separate .jpg file.



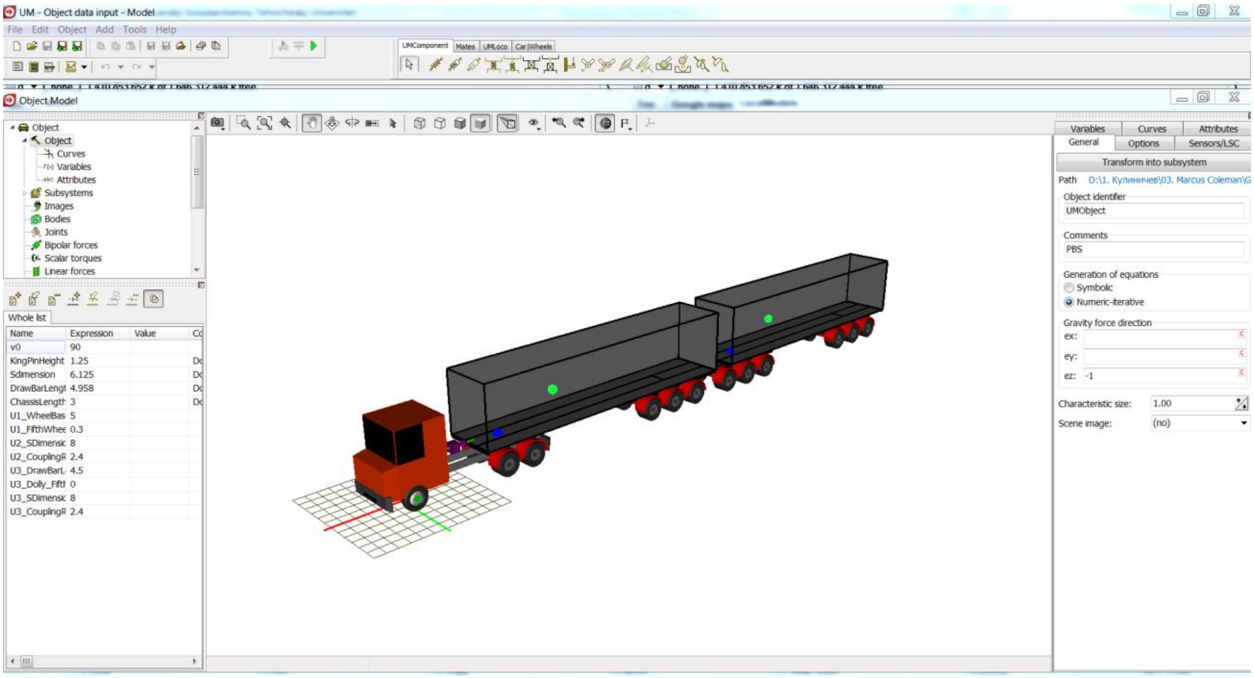
- Calculate real width and height of rectangle

realWidth = pixelWidth/pixelRatio = 1673 [pix] / 12.2 [pix/m] = 137.1 [m]

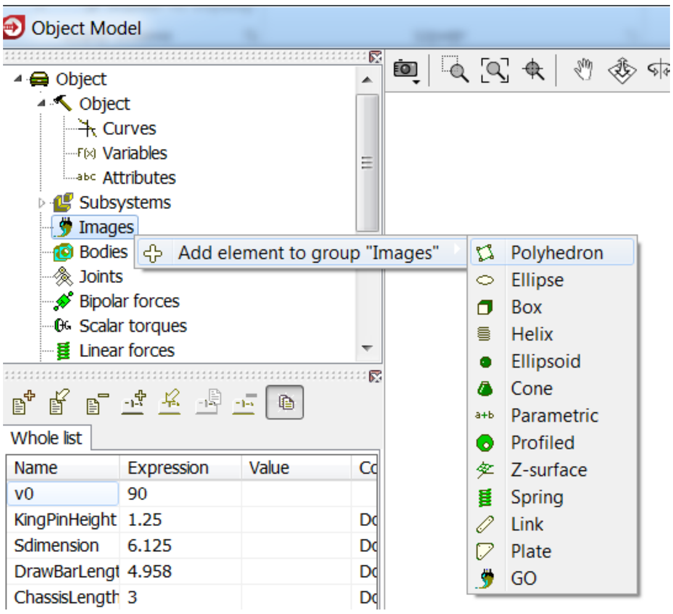
realHeight = pixelHeight/pixelRatio = 771 [pix] / 12.2 [pix/m] = 63.2 [m]

1.11.2. Add texture with picture in the UM model

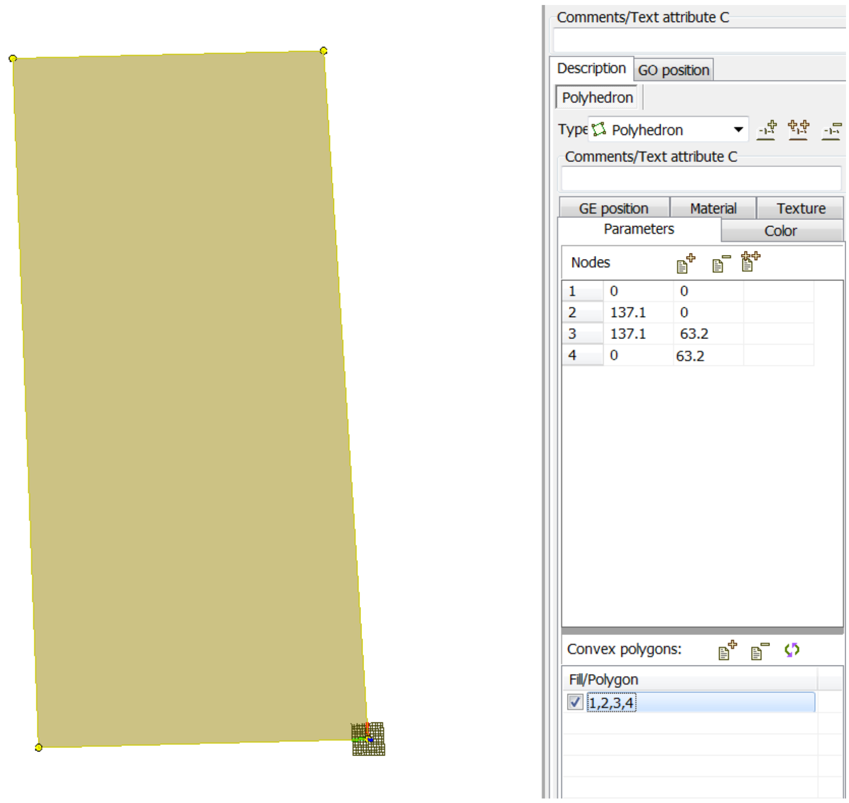
- Open UM model in the **UM Input** program



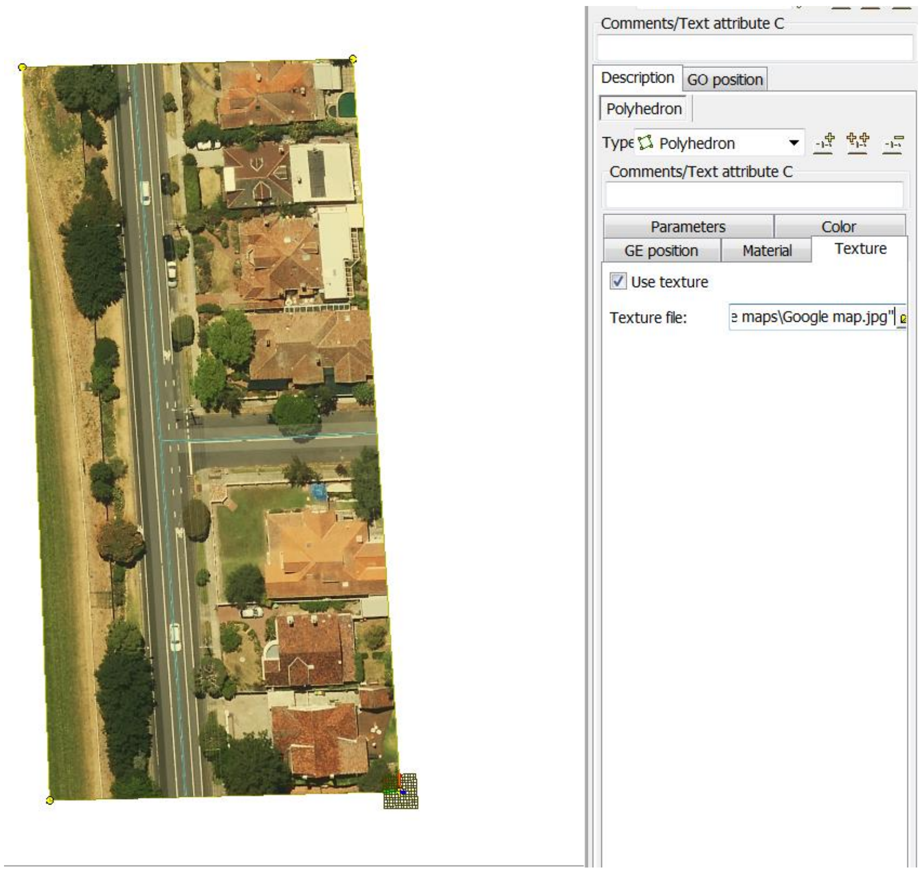
- Create new graphical object - Polyhedron and named it as "Map"



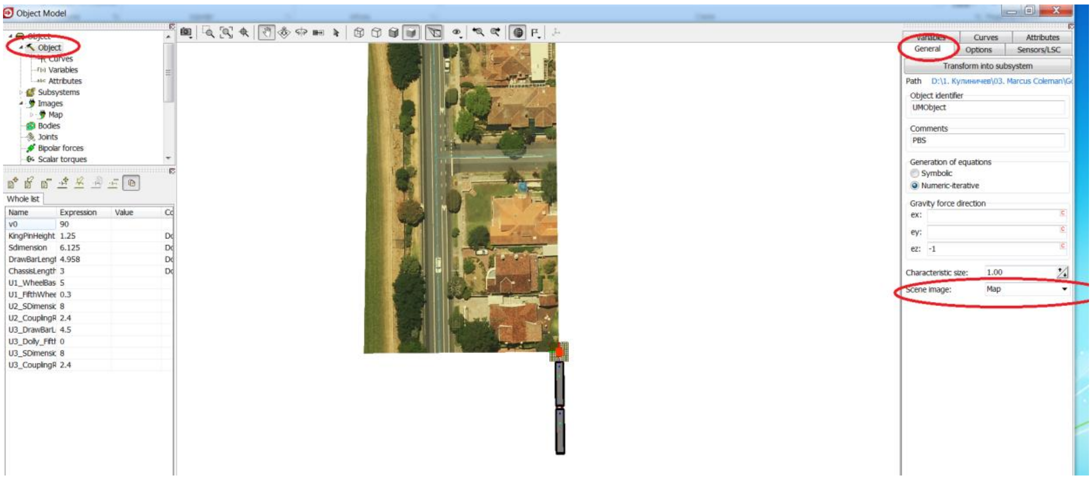
- Create nodes and polygon as shown at the figure



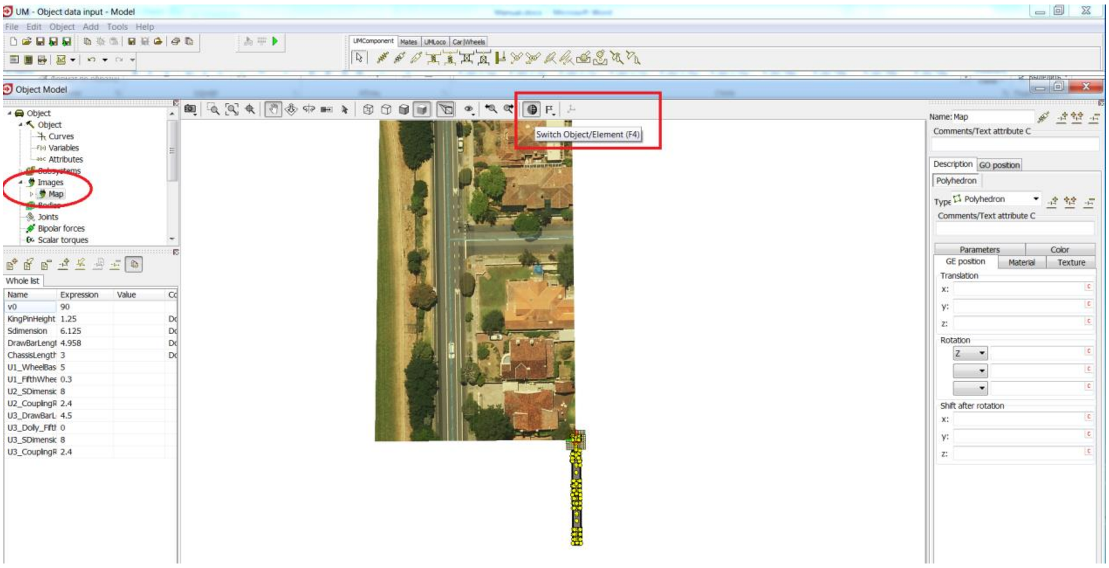
- Use a texture on this polyhedron



- Set graphical object "Map" as Scene Image

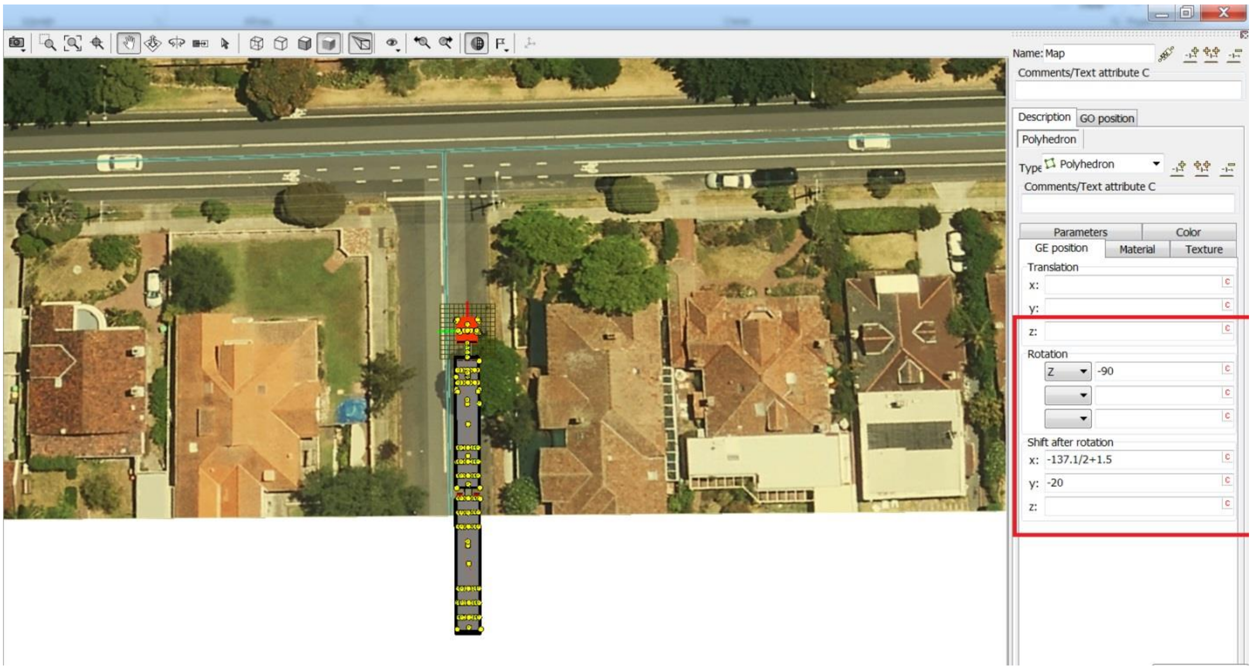


- Choose graphical object "Map" again and enable "Switch Object/Element (F4)" button



- Rotate graphical object "Map" so it's vertical direction and Axle X of the model will be collinear

- Move graphical object "Map" relative vehicle so vehicle will be on it's initial position.

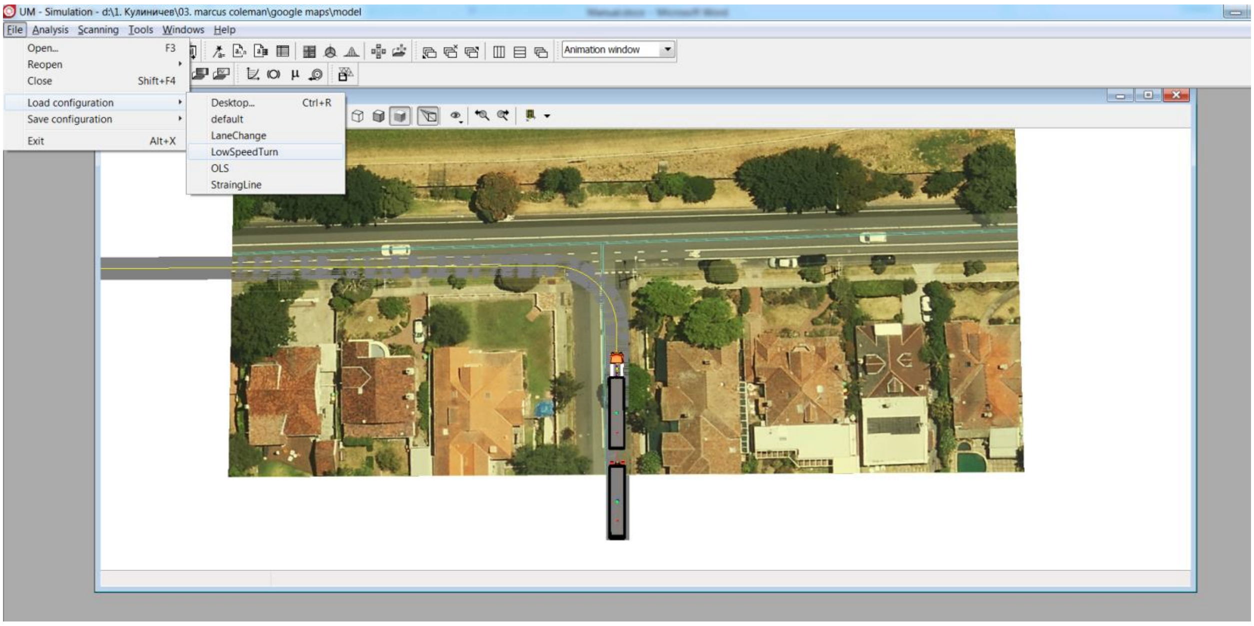


Note: As you can see, vehicle doesn't fit in this picture because vehicle is too long (it happens because I chose small region on Google Maps. You should choose larger region)

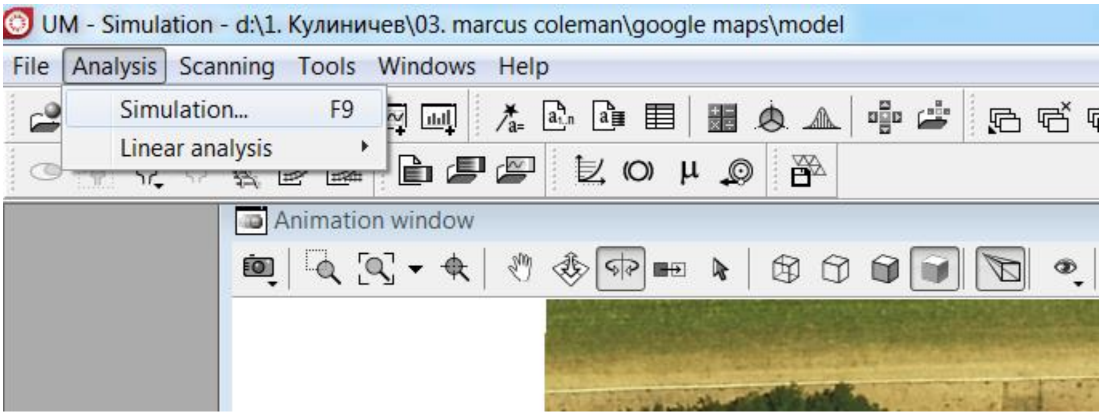
- Save and close model.

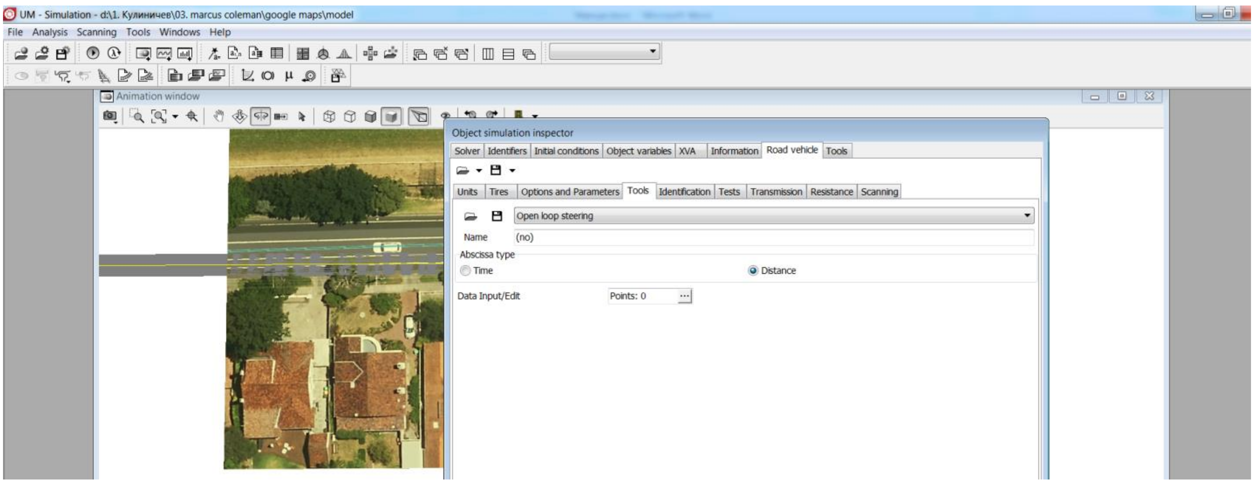
1.11.3. Run simulation

- Open model in the **UM Simulation** program and load necessary configuration (for example, LowSpeedTurn)

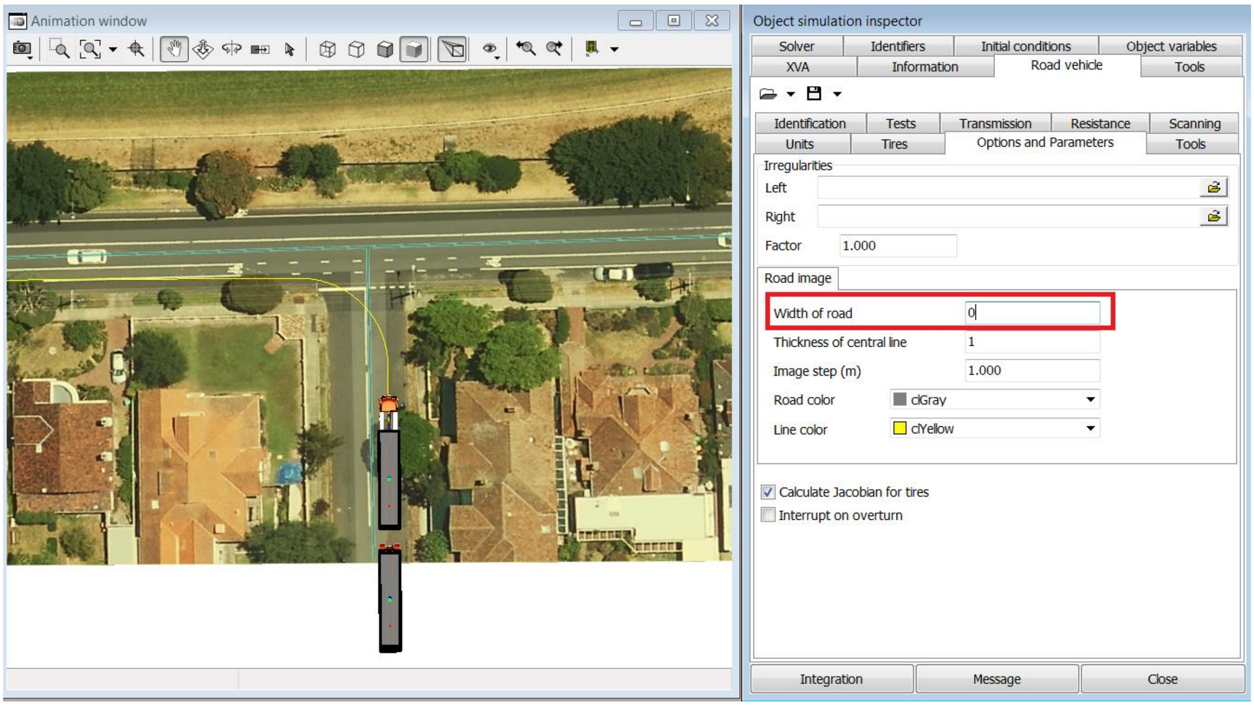


- Call "Object simulation inspector" (Analysis –> Simulation)

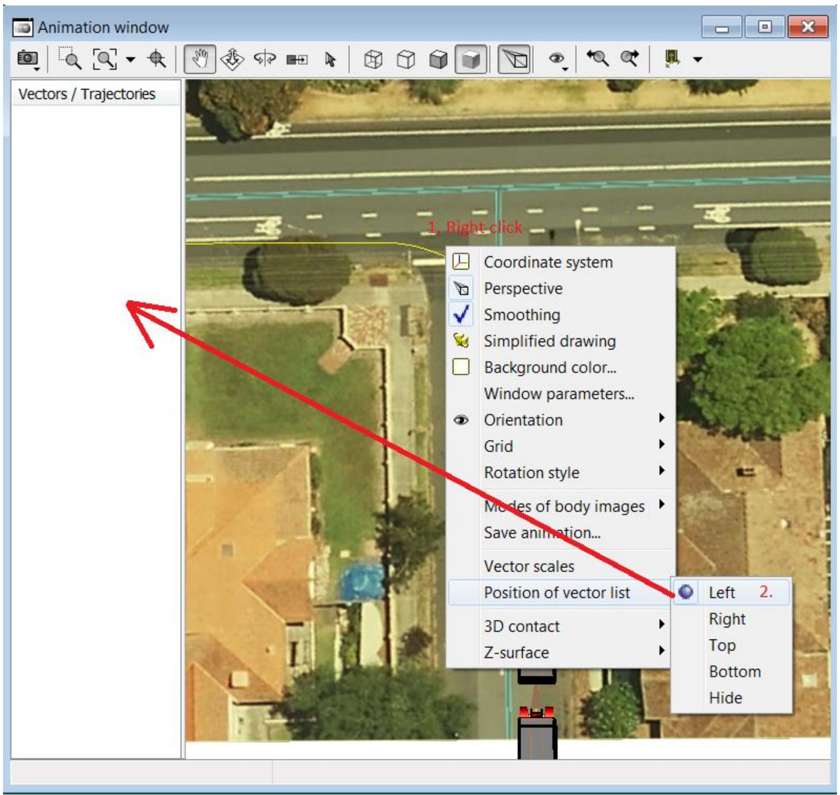


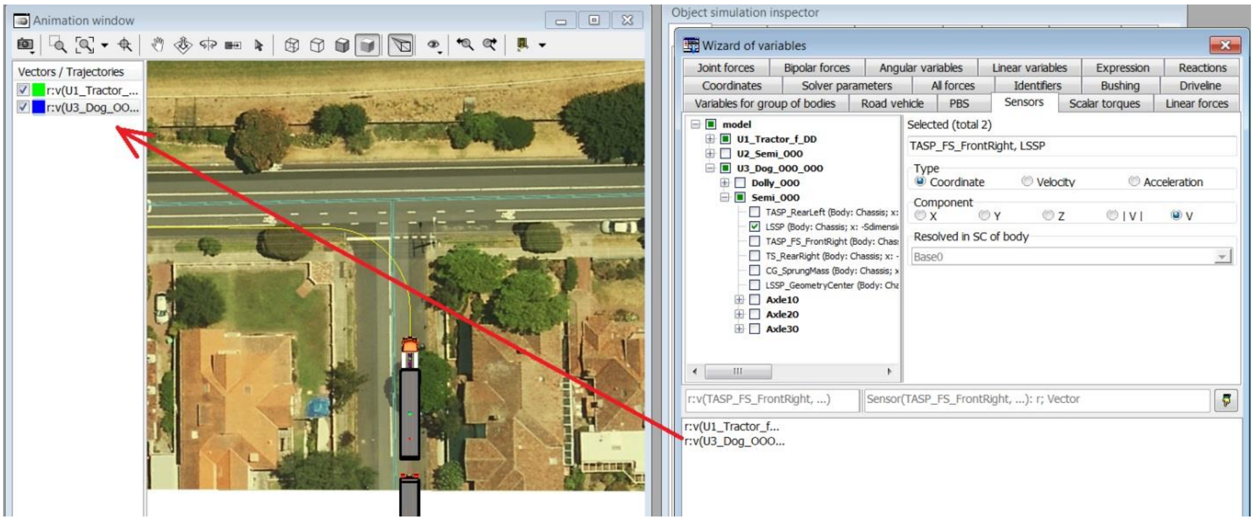


- Remove standard image of road by setting to zero "Width of road" parameter

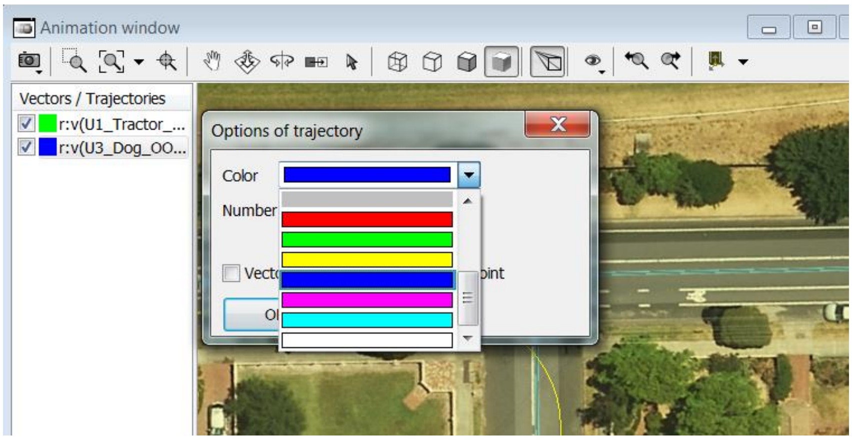


- Create and add trajectory of necessary points to animation window (for example, LSSP sensors):

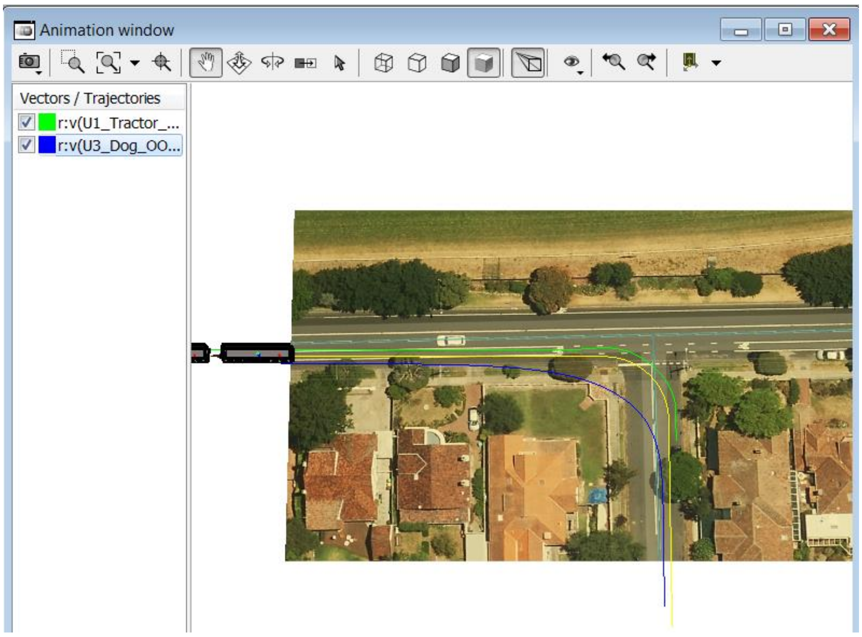




- Change the color of trajectories



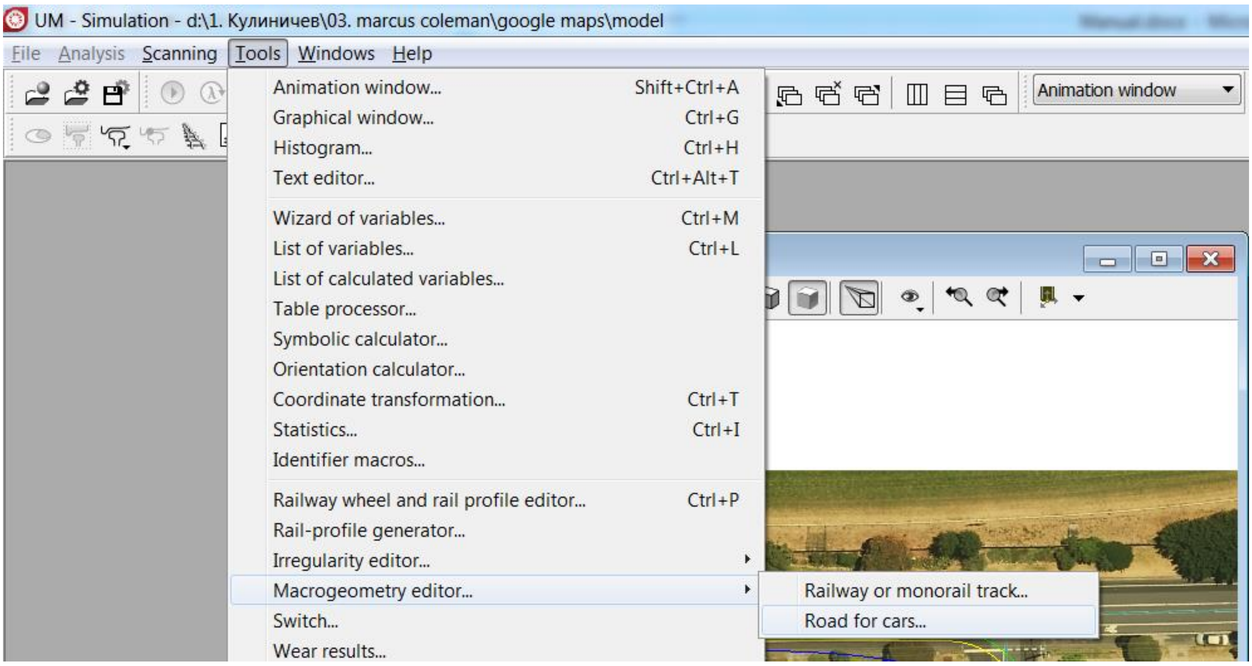
- Run integration



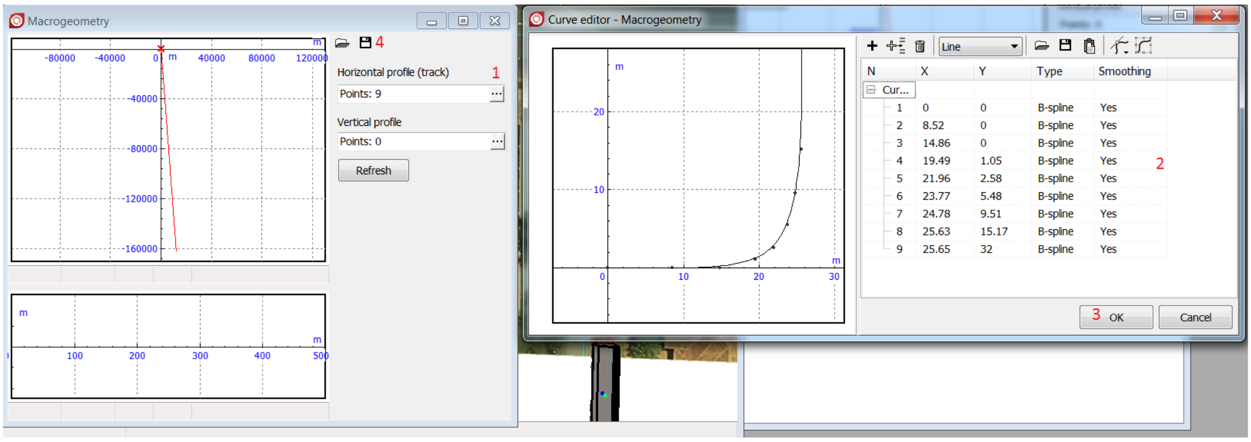
1.11.4. Editing macrogeometry

As you can see on figure above, standard macrogeometry doesn't fit in the our road picture, so we must create another macrogeometry. You can create road via macrogeometry editor

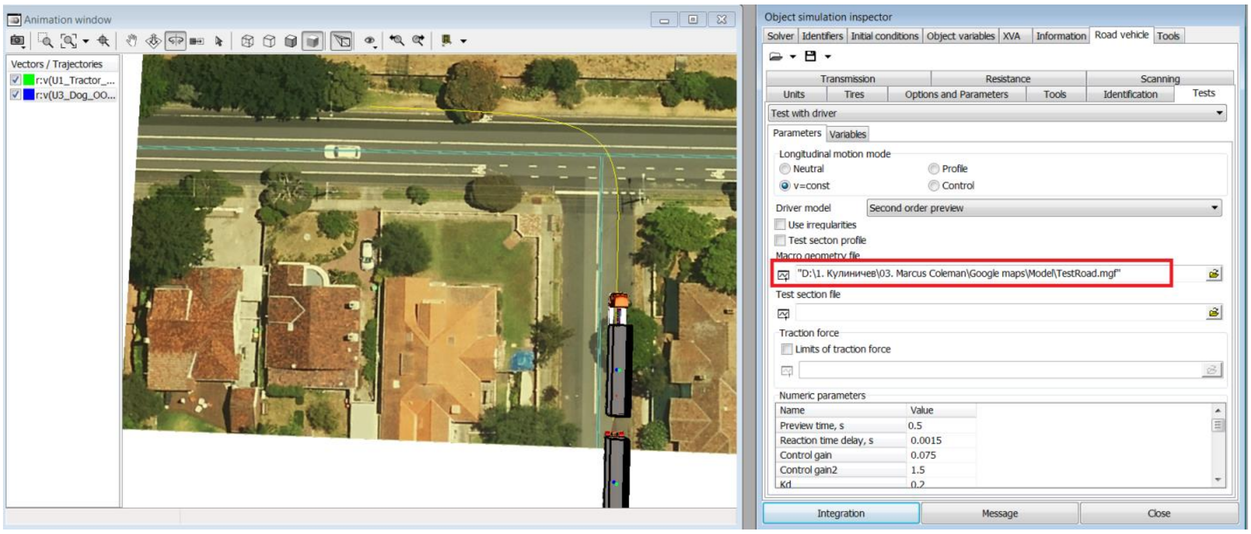
- Open macrogeometry editor



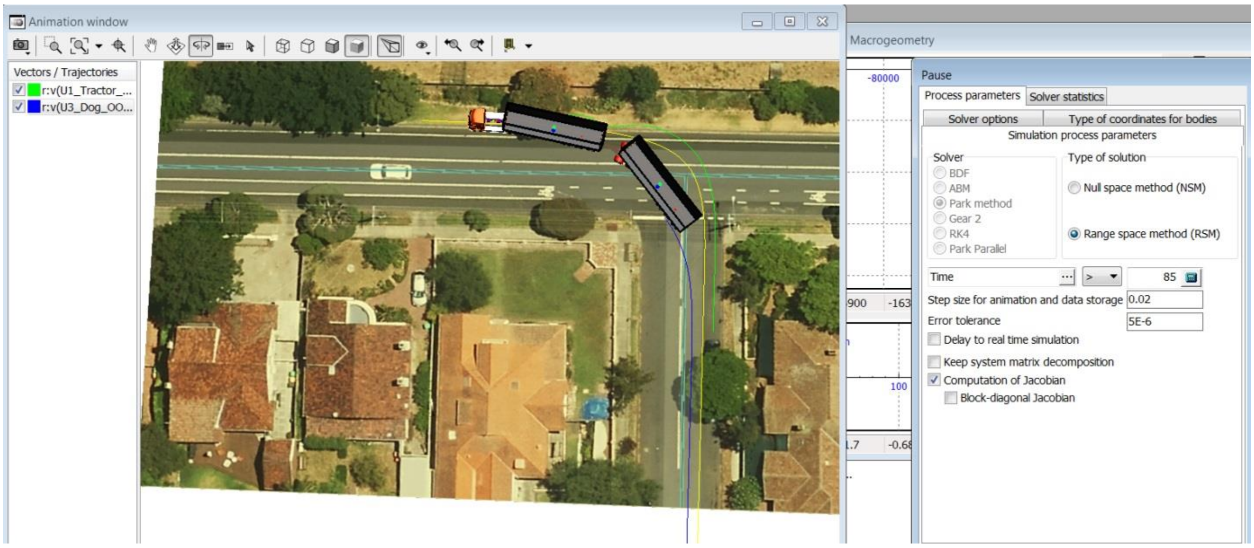
- Create necessary road and save it



- Load this macro in the UM



- Run simulation



1.12. Library of Car Suspensions

1.12.1. Introduction

This user manual describes the models of some typical car suspensions, distributed as parts of the "Universal mechanism" software (UM). Suspension models are combined into a library, which is situated in the catalog {UM Data}\SAMPLES\Automotive\Suspension after the installation of the "Universal Mechanism".

Library of car suspensions contains the most common types of suspensions for cars and trucks. The following suspension models are available in the current version:

- axle suspension, see Sect. 1.12.2.1, page 1-213;
- double wishbone suspension, see. Sect. 1.12.2.2, page 1-214;
- semi-trailing arm suspension, see Sect. 1.12.2.3, page 1-215;
- McPherson suspension, see Sect. 1.12.2.4, page 1-216;
- torsion suspension, see Sect. 1.12.2.5, page 1-217;
- multi-link suspension, see Sect. 1.12.2.6, page 1-218.

Note Please note that this set of suspensions and bodies is a prototype of the real objects and does not reflect all their geometric and dynamic properties. The models are intended for illustrative and educational purposes only.

1.12.2. Brief description

1.12.2.1. Axle Suspension

This is a rear axle suspension, which includes a rigid beam that connects the wheels, four guide arms and one transverse which is called Panhard rod. The levers are attached on one side to the beam and on the other side to the body of the car. Springs and dampers are used as elastic and damping elements. Currently, this type of suspension is widely used on off-road vehicles VAZ 2121, VAZ 2123, Dodge Ram. The scheme and the suspension arrangement can be found by clicking on the link: https://en.wikipedia.org/wiki/Beam_axle.

Model folder: {UM Data}\SAMPLES\Automotive\Suspensions\Axle Suspension.

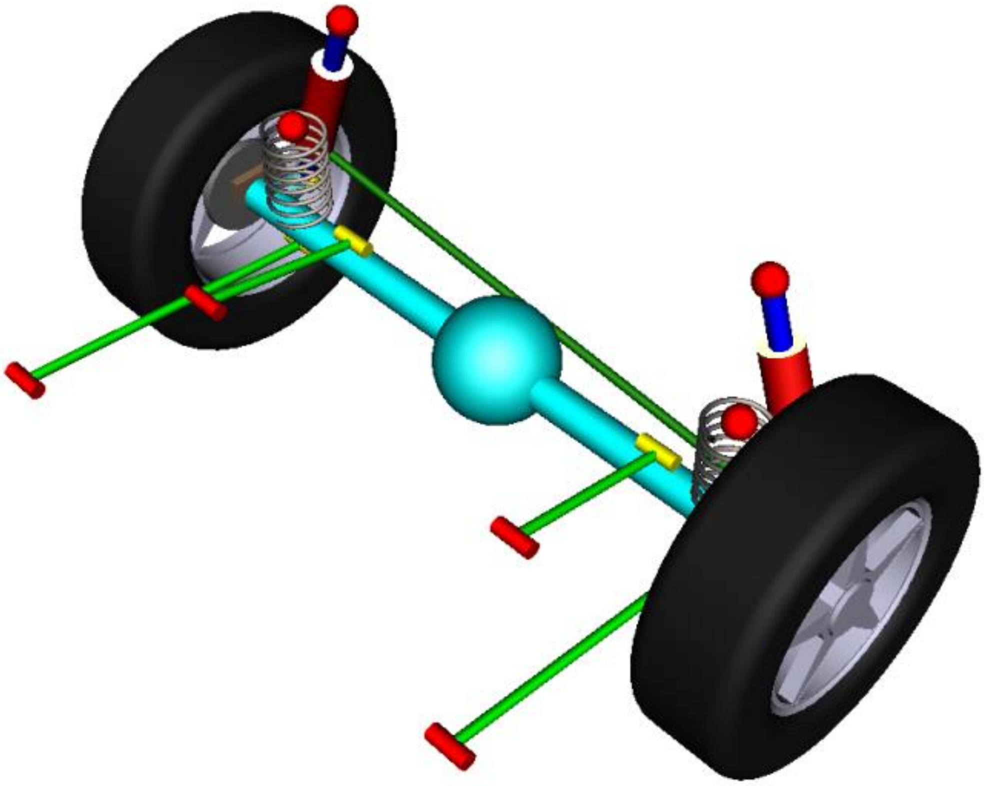


Figure 1.236. Axle suspension with guided arms

1.12.2.2. Double Wishbone Suspension

This type of suspension is one of the most widespread options of the front independent suspension. On each side there are two U-shaped transverse wishbones, the inner ends of which are attached to the car body, and the outer ones to the steering knuckle. The model also has a stabilizer bar. The steering is modeled by the steering rack and the associated steering rods. Springs with shock absorbers are made coaxially.

The suspension is used on many sport cars, for example on Ferrari, TVR, Lotus, and also on Mercedes-Benz, BMW, Honda, Alfa Romeo. A more detailed description of this suspension type can be found via the following link: https://en.wikipedia.org/wiki/Double_wishbone_suspension.

Model folder: {UM Data}\SAMPLES\Automotive\Suspensions\Double Wishbone Suspension.

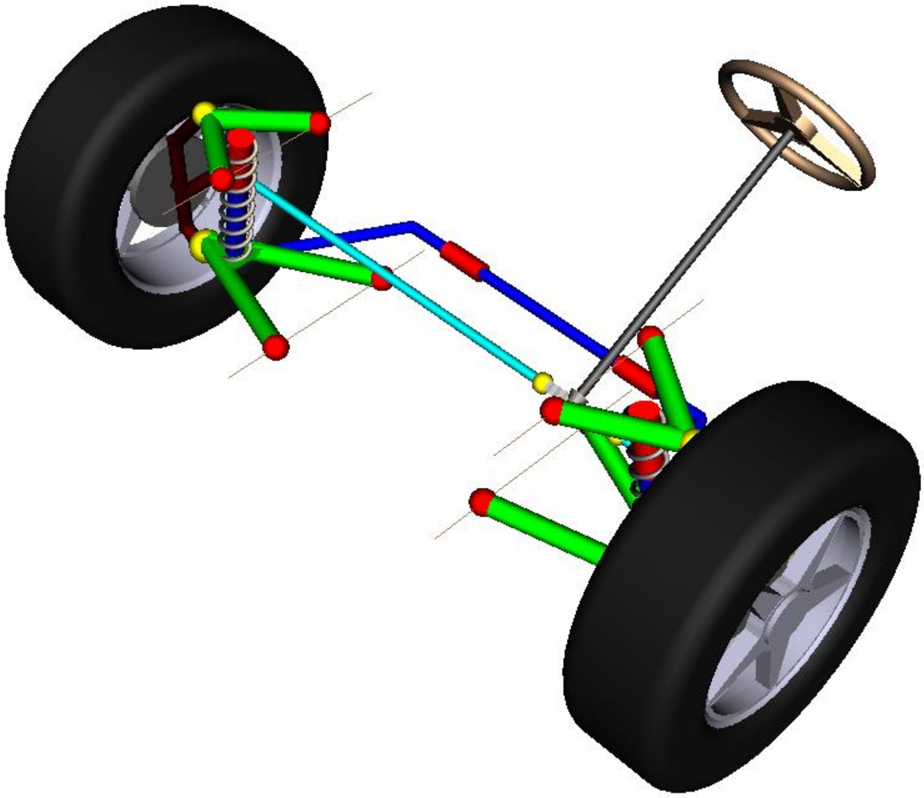


Figure 1.237. Double wishbone suspension

1.12.2.3. Semi-Trailing Arm Suspension

A semi-trailing arm suspension is a supple independent rear suspension system for automobiles where each wheel hub is located only by a large, roughly triangular arm that pivots at two points. Viewed from the top, the line formed by the two pivots is somewhere between parallel and perpendicular to the car's longitudinal axis; it is generally parallel to the ground. Trailing-arm and multilink suspension designs are much more commonly used for the rear wheels of a vehicle where they can allow for a flatter floor and more cargo room.

This suspension design can be found in early BMW cars 3 series, Opel, Fiat. For more detailed information see: https://en.wikipedia.org/wiki/Trailing-arm_suspension.

Model folder: {UM Data}\SAMPLES\Automotive\Suspensions\Semi-trailing Arm Suspension.

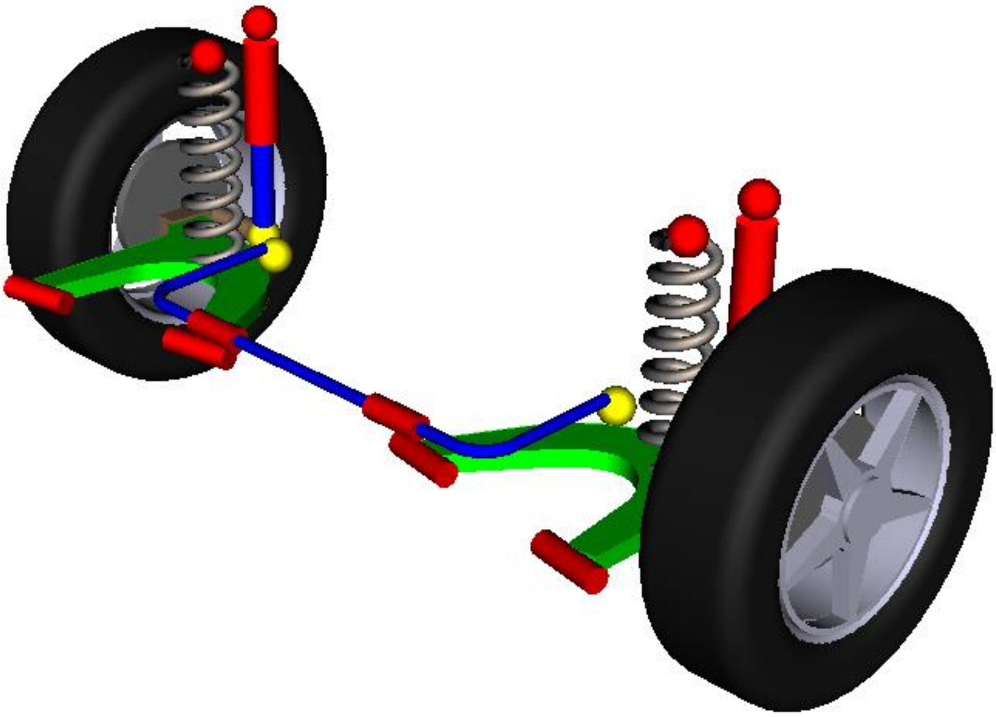


Figure 1.238. Semi-Trailing Arm Suspension

1.12.2.4. MacPherson Suspension

The MacPherson strut is a type of automotive suspension system that uses the top of a telescopic damper as the upper steering pivot. It is widely used in the front suspension of modern vehicles and is named for American automotive engineer Earle S. MacPherson, who invented and developed the design.

A MacPherson strut uses a wishbone, or a substantial compression link stabilized by a secondary link, which provides a mounting point for the hub carrier or axle of the wheel. This lower arm system provides both lateral and longitudinal location of the wheel. The upper part of the hub carrier is rigidly fixed to the bottom of the outer part of the strut proper; this slides up and down the inner part of it, which extends upwards directly to a mounting in the body shell of the vehicle, see https://en.wikipedia.org/wiki/MacPherson_strut.

Nowadays it is one of the most popular front suspensions for cars from mass segment and can be found in many cars including Hyundai Creta, Mitsubishi Lancer, Audi 80, Chevrolet Aveo, Ford Focus, Skoda Octavia, Toyota Camry etc.

Model folder: {UM Data}\SAMPLES\Automotive\Suspensions\MacPherson.

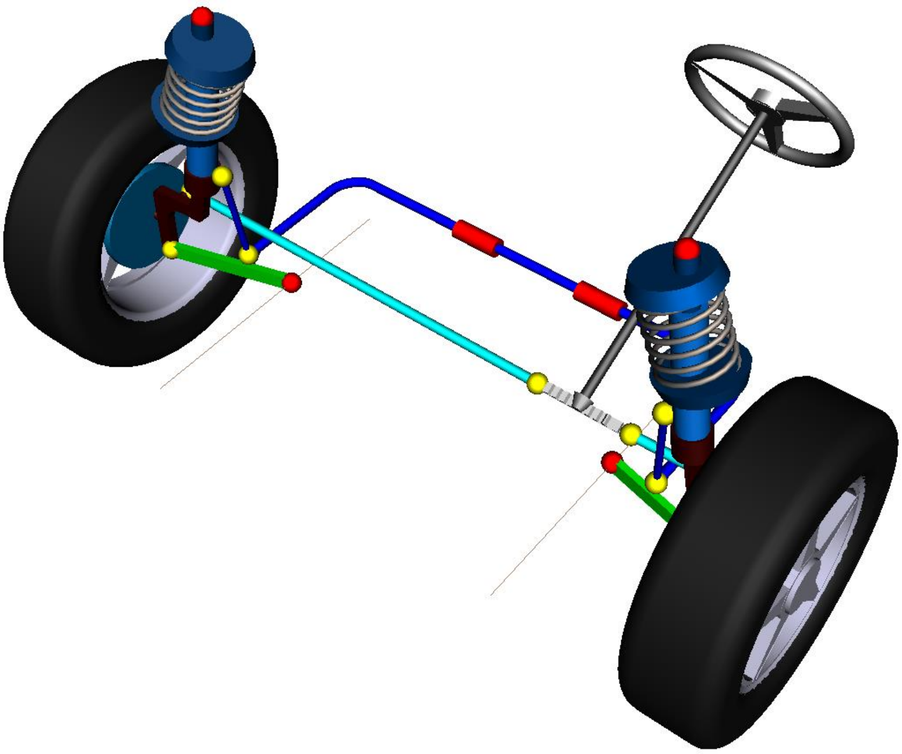


Figure 1.239. MacPherson Suspension

1.12.2.5. Torsion Suspension

A torsion bar suspension, also known as a torsion spring suspension, is a vehicle suspension that uses a torsion bar as its main weight-bearing spring. One end of a long metal bar is attached firmly to the vehicle chassis; the opposite end terminates in a lever, the torsion key, mounted perpendicular to the bar, that is attached to a suspension arm, a spindle, or the axle. Vertical motion of the wheel causes the bar to twist around its axis and is resisted by the bar's torsion resistance. The effective spring rate of the bar is determined by its length, cross section, shape, material, and manufacturing process, see https://en.wikipedia.org/wiki/Torsion_bar_suspension for more details.

This type of suspension is used in some of cars by Renault and Honda, in Opel Mokka and Toyota Corolla.

Model folder: {UM Data}\SAMPLES\Automotive\Suspensions\Torsion Suspension.

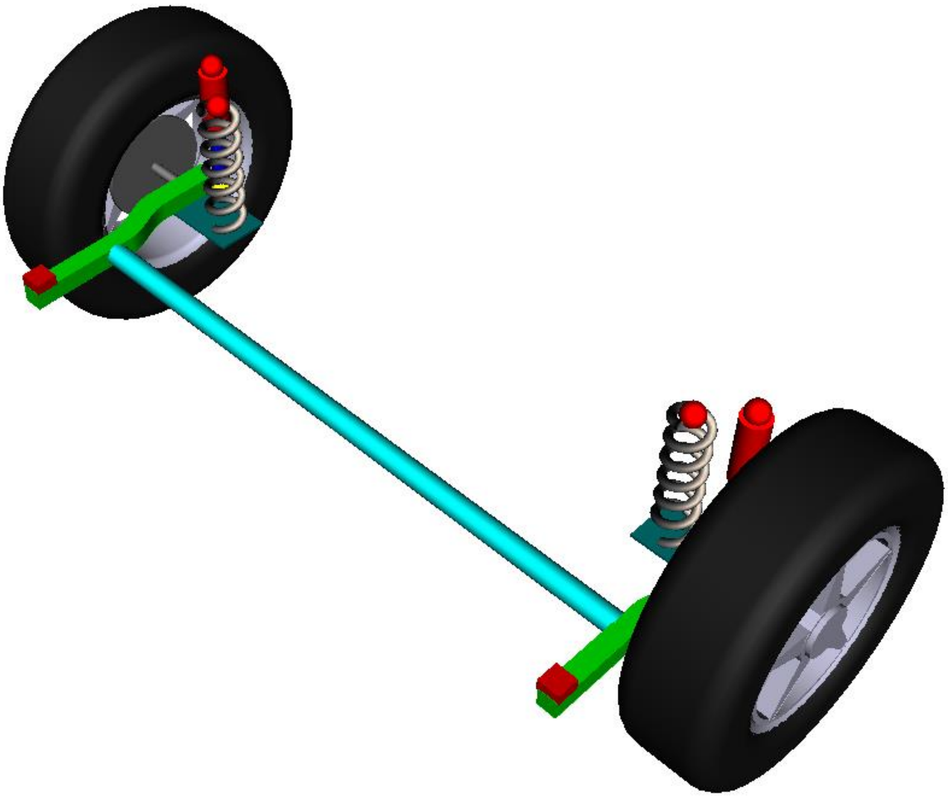


Figure 1.240. Torsion Suspension

1.12.2.6. Five-Link Suspension

At present the multi-link suspension is one of the most popular rear independent suspensions. A wider definition considers any independent suspensions having three control links or more multi-link suspensions. These arms do not have to be of equal length, and may be angled away from their "obvious" direction. It was first introduced in the late 1960s on the Mercedes-Benz C111 and later on their W201 and W124 series.

Typically each arm has a spherical (ball) joint or rubber bushing at each end. Consequently, they react to loads along their own length, in tension and compression, but not in bending. Some multi-links do use a trailing arm, control arm or wishbone, which has two bushings at one end. Please find more details via the following link: https://en.wikipedia.org/wiki/Multi-link_suspension.

Nominal geometry of the suspension and lengths of rods are taken from [13].

Model folder: {UM Data}\SAMPLES\Automotive\Suspensions\Multilink suspension.

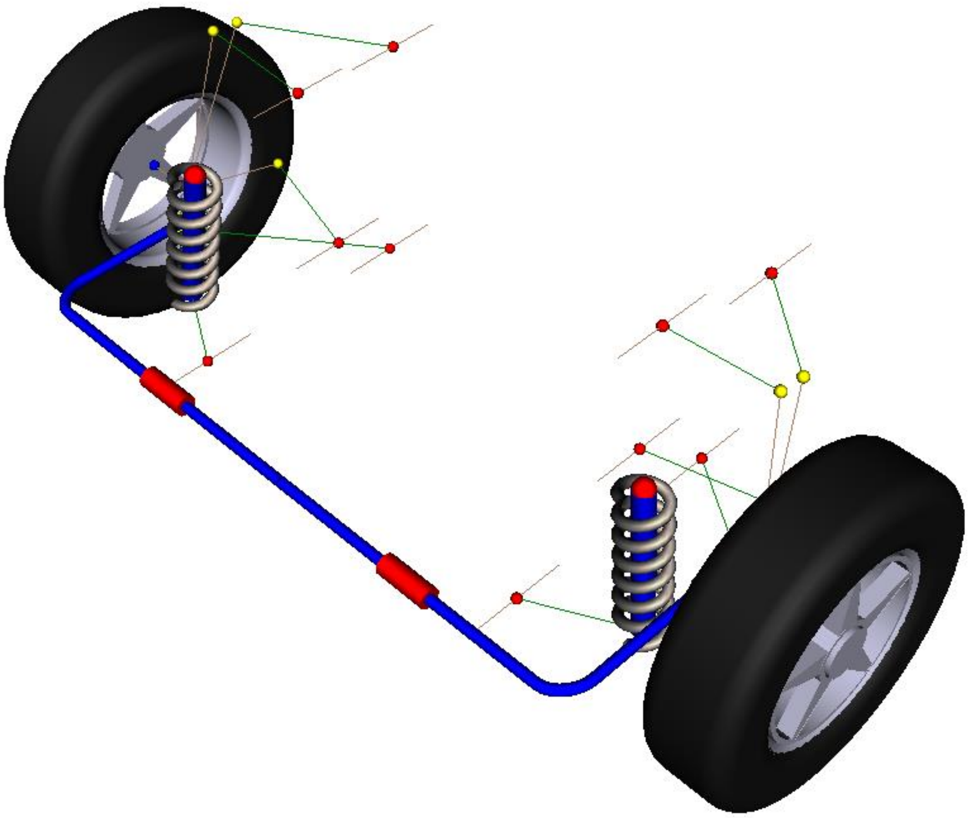


Figure 1.241. Multi-link Suspension

1.12.3. Parameterization and Structure of Models

Models of suspensions from the library are parameterized uniformly and designed so as to provide the easiest way to create car models based on these suspensions. Key suspension properties are parameterized for easily tuning for every specific car model.

Each suspension model includes the **Local Car Body** that is considered as an intermediate substitution for a car body. Subsequently while including the suspension model into the car model this **Local Car Body** rigidly connected to the car body. Such technique provides the simplest way to create a model of a vehicle based on the suspension from the library.

For a better illustrativeness similar elements of suspension models from the library have the same color. Steering rods are blue, arms and wishbones are green, stabilizers are dark blue, dampers are red and blue, springs are grey.

1.12.3.1. Geometrical parameters

The distance in between centers of wheels in meters is introduced by the **Gauge** identifiers.

Coordinates of attachment points for force elements are introduced with the help of named points **A**, **B**, **C** etc and parameterized in the following way, see Figure 1.242.

A_{X, Y, Z}pos is the project of the **A** point on X, Y, Z axis, where **A** point is the attachment point of the spring to the beam, m.

A_{dX, dY, dZ}pos is the distance between projection of the attachment point of the spring to the car body and to the beam, m.

B_{X, Y, Z}pos is the project of the B point where the damper is attached to the beam, m.

B_{dX, dY, dZ}pos is the distance between projection of the attachment point of the damper to the car body and to the beam, m.

Note	Please pay attention to changing values of identifiers of the same name. Creating the model of a car with the help of suspension from the library you may add suspensions with same identifiers for A, B etc. points. Make sure that you change identifier(s) for the suspension you need only.

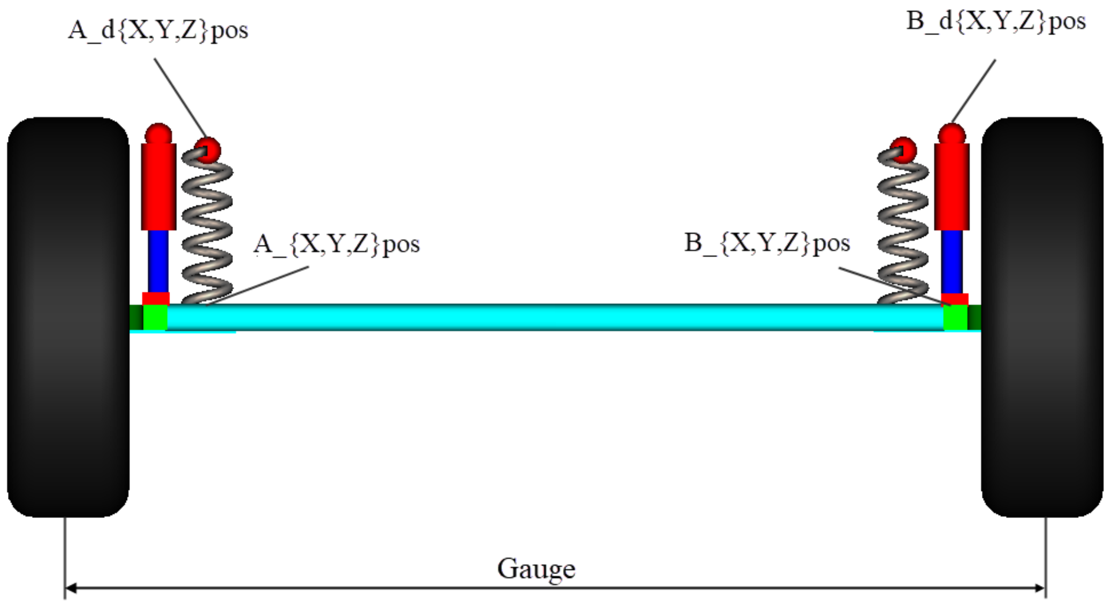


Figure 1.242. Parameterization of gauge and attachment points of force elements

1.12.3.2. Parameterization of Wheels

Let us consider basic parameters that express wheel geometry and camber and toe angles, see. Figure 1.243 -.

Wheel_TireHeight is the height of the tire as a part of the whole wheel radius, m.

Wheel_Radius is the radius of the unloaded wheel, m.

Wheel_TireWidth is the width of the tire, m.

Camber and **Toe** are the camber and toe angle correspondingly in degrees.

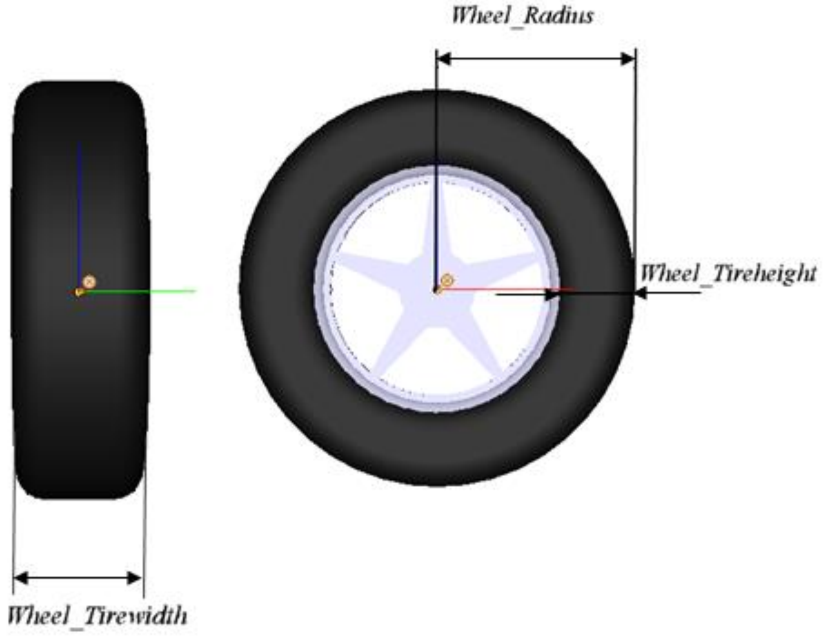


Figure 1.243. Basic geometrical parameters of the wheels

Positive toe, or toe in, is the front of the wheel pointing towards the centerline of the vehicle, see Figure 1.244. If the top of the wheel is farther out than the bottom (that is, away from the axle), it is called positive camber; if the bottom of the wheel is farther out than the top, it is called negative camber, see Figure 1.245.

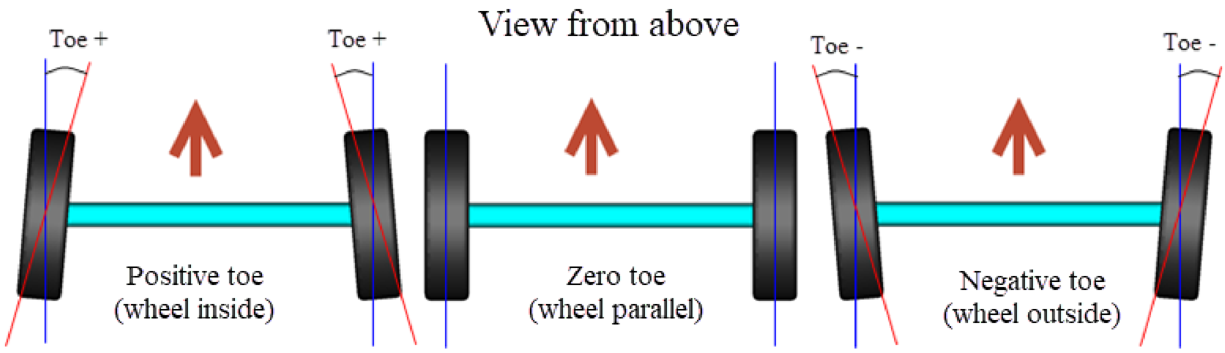


Figure 1.244. Toe angle

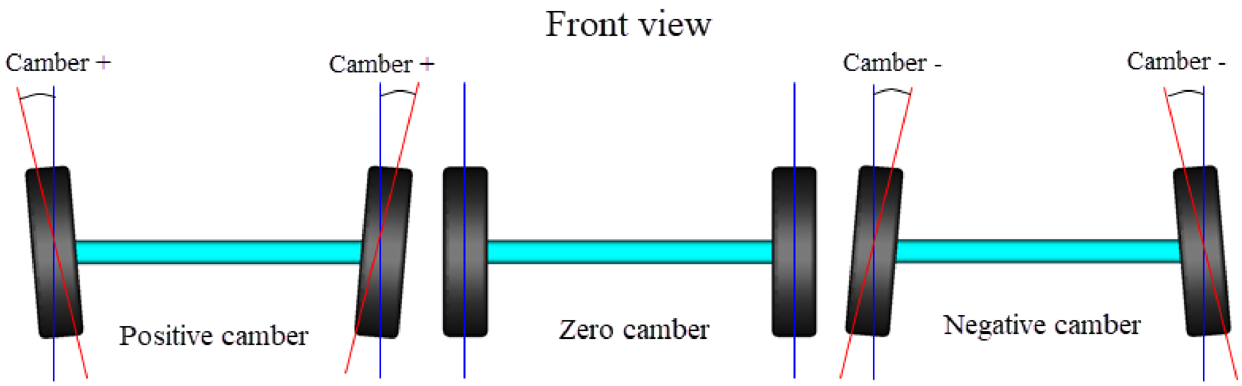


Figure 1.245. Camber angle

1.12.3.3. Steering Control

In the models of the front suspensions, a rack-and-pinion steering mechanism is introduced. To give a user a possibility to tune the model the length and the inclination angle of the steering column were introduced, see Figure 1.246:

- **SteeringColumnLength** is the length of the steering column, m;
- **SteeringColumAngle** is the inclination angle in degrees described in Figure 1.247.

The inclination angle is parameterized and introduced in the **jSteering Column** generalized joint, see **RTy** elementary transformation, see Figure 1.247. By default 30º angle is used. Please note that the identifier **SteeringColumnAngleRad** showed in Figure 1.247 is expressed in radians and is dependent from the **SteeringColumnAngle**, given in degrees for easier parameterization.

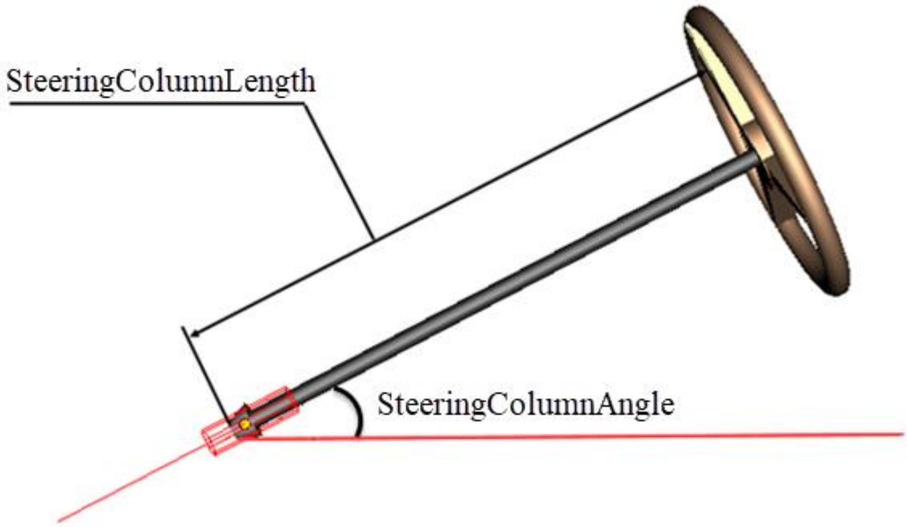


Figure 1.246. Geometric parameters of the steering column

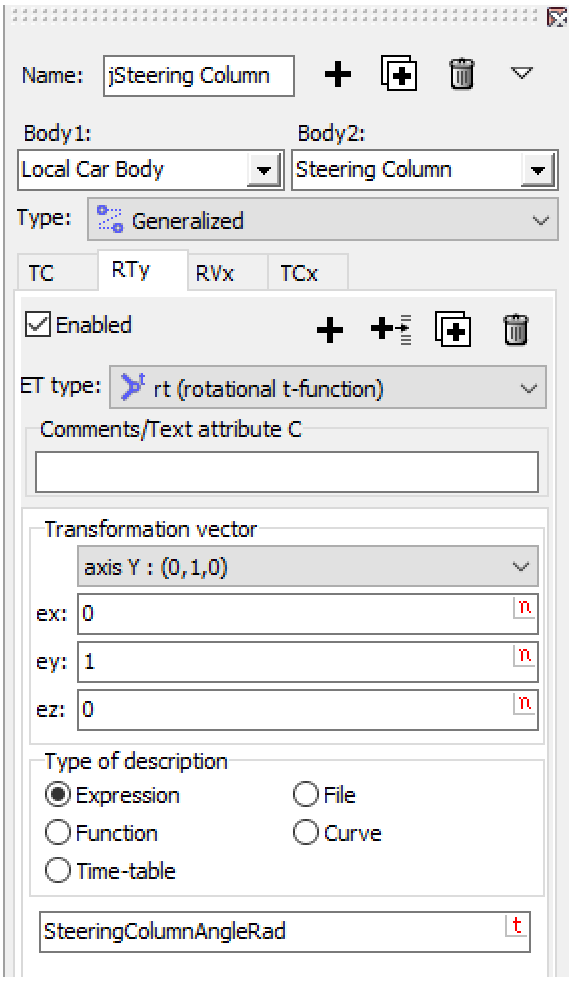


Figure 1.247. Setting the steering departure angle

1.12.3.4. Modeling of Powered Wheels

To model the driving wheels in the suspensions from the library, a driving torque is introduced, which is described by the expression **MLongitudinalControl*TractionFactor**. When you set **TractionFactor=1** the traction torque is transmitted to the wheels and the suspension axis becomes the powered one. If **TractionFactor=0** transmission of the moment is not carried out. Thus, by controlling the **TractionFactor** identifier the same suspension model can be powered and non-powered.

1.12.3.5. Inertial parameters

In the library of the suspensions the following notations for the inertial parameters of the bodies are used:

m[Body] is the Body mass, kg;

Ixx[Body], **Iyy[Body]**, **Izz[Body]**, **Ixy[Body]**, **Ixz[Body]**, **Iyz[Body]** are the moments of inertia of the Body, kg· m^2;

X_COG_[Body], **Y_COG_[Body]**, **Z_COG_[Body]** are X, Y and Z is the position of the center of gravity of the "Body", m.

Note **COG** is the acronym for *Center of Gravity*.

1.12.4. Creating a Car Model Using Suspensions from Libraries

1.12.4.1. Creating Car Model

Let us consider the creation of a four-wheel drive *Lada 4×4*. The first suspension is the double wishbone suspension (Sect. 1.12.2.2), and the rear suspension is the axle suspension (1.12.2.1), see Figure 1.248. The gauge of the front suspension is 1440 mm and the rear is 1420 mm. We will use the following "factory" settings. The camber angle is 0.5 º, toe-in is 3 mm or 0.125 º. In the model we will use the recommended tires which size is "175/80 R16".

Figure 1.248. Lada 4×4 and its model in **UM Input**

1.12.4.1.1. Creating Car Body

1. Run **UM Input** program and create a new model. Save it as **Lada 4x4**.
2. Load an image of the body. To do this, click the button **Read element from file** and go to folder {**UM Data}**\SAMPLES\Automotive\Car bodies. From the list of available files select **Lada 4x4.img**. The car body appears in the animation window
3. Create a new body **CarBody** and select just loaded **CarBody** as the graphical image.
4. Assign the inertia parameters of the body as follows (Figure 1.249):
   - **mCarBody=1000**,
   - **IxxCarBody=486**,
   - **IxyCarBody=355**,
   - **IxzCarBody=-158**,
   - **IyyCarBody=950**,
   - **IyzCarBody=-72**,
   - **IzzCarBody=889**.
5. In field Coordinates of center of mass set the following values **X_COG_CarBody=-1.94**, **Y_COG_CarBody=0**, **Z_COG_CarBody=0.75**, see Figure 1.249.

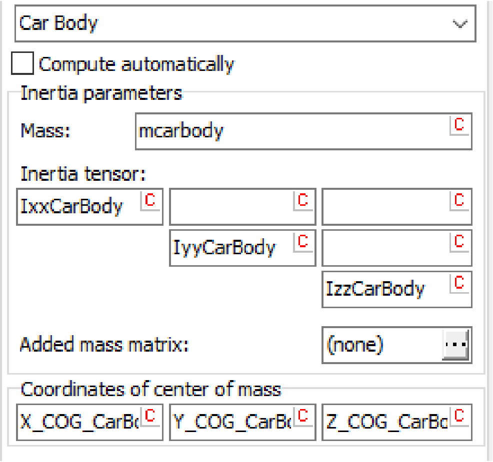


Figure 1.249. Inertial parameters of the body

6. Create a joint **6 d.o.f.** for **CarBody**. As the first body select **Base0** and turn on all d.o.f in this joint. Set the joint name to jBase0_CarBody, see Figure 1.250.

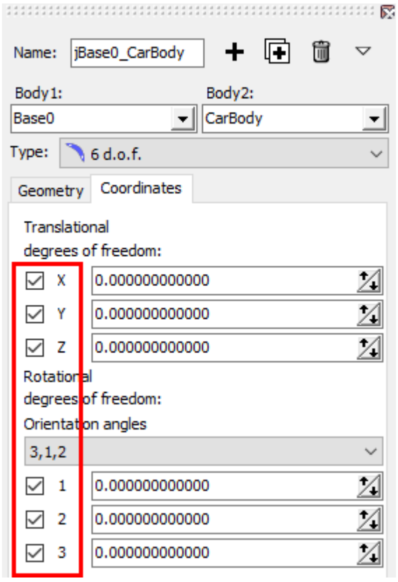


Figure 1.250. Creating a joint for the car body

1.12.4.1.2. Adding a Suspension Model from Library

1. Now we will add the front suspension. To do this, select **Subsystems** in the tree of elements and click **Add new element**. In field **Name** type **FrontSuspension**, and in field **Type** choose **included**. After that, a window to select the model to be added as a subsystem will appear. Go to the folder where the suspensions are located and select **Double Wishbone Suspension** (Sect. 1.12.2.2, page 1-214), then click **OK**.

Note	Models from the suspension library are located in the **{UM Data}\SAMPLES\Automotive\Suspensions** folder.

2. Now let us set the suspension position. Select the **General** tab and in the **Identifier** field input **Front**, see Figure 1.251, left. In fact, this step is optional. You can leave default value in the **Identifier** field.

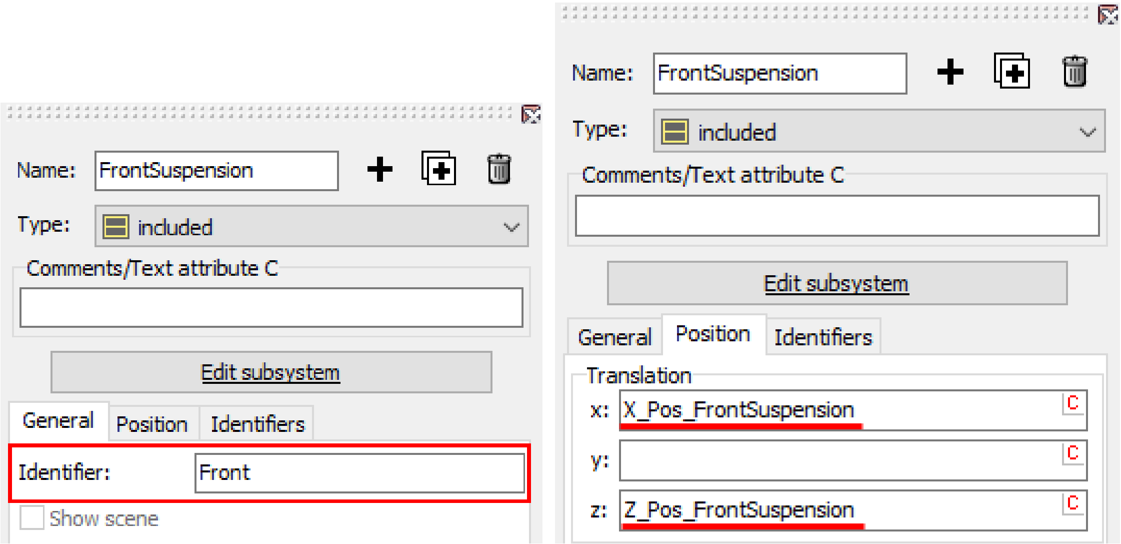


Figure 1.251. Identifier for the front suspension (optional)

3. The next step is also optional. It will help you to move the suspension to the right position right in the beginning of your creation of the model. Select the **Position** tab. In the fields **Translation | x** and **Translation | z** input **X_Pos_FrontSuspenstion=-0.721** (m) and **Z_Pos_FrontSuspenstion**=0.343 (m) correspondingly, see Figure 1.251, right.

4. Now we will set the actual gauge of the front suspension. Click the **Identifiers | Whole list** tab sheet. Find the **Gauge** identifier and set it to **1.44** (m), see Figure 1.252.

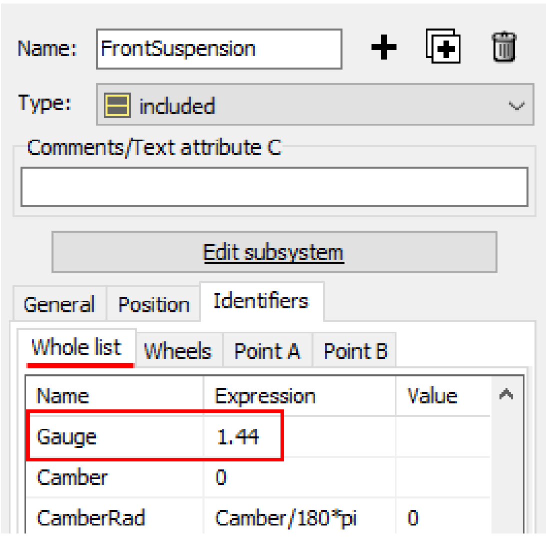


Figure 1.252. Gauge of the front suspension (1.44 m)

5. Since *Lada 4×4* is the four-wheel drive vehicle, so the driving torque should be applied on both front and rear wheels. Set **TractionFactor** identifier to **1**, see Figure 1.253.

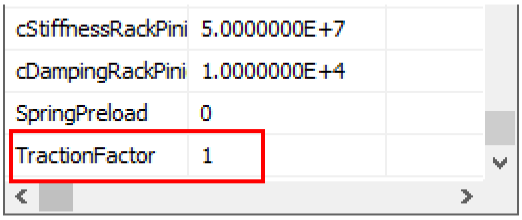


Figure 1.253. **TractionFactor** identifier for the front suspension

6. In the same way add the **Axle Suspension**, see Sect. 1.12.2.1, page 1-213. Set its name to **RearSuspension**. Set **Identifier** field on the **General** tab to **Rear** (Figure 1.254, left). Then click the **Position** tab and in the fields **Translation | x** and **Translation | z** input **X_Pos_RearSuspenstion=-3.16**, **Z_Pos_RearSuspenstion=0.343**, see Figure 1.254, right.

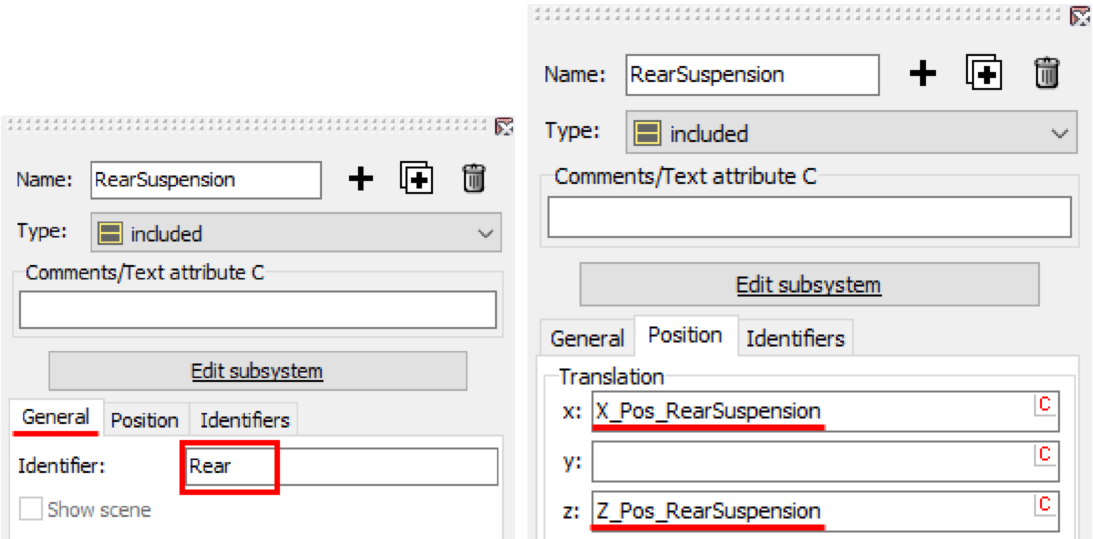


Figure 1.254. Identifier and position for the rear suspension

7. Set the **Gauge** identifier for the rear suspension to **1.42** (m), and **TractionFactor** set to **1**. Please note that you should set the different values for **Gauge** identifier for the front and the rear suspensions. The window **Identifiers of the same name** will appear on changing the **Gauge** identifier, see Figure 1.255. Since the gauge for the front and rear suspension is different (1,44 m for the front suspension and 1,42 m for the rear one), so in the **Identifiers of the same name** window you should turn off the check box at the **FrontSuspension.Gauge** identifier in order not to change it. So as you will change the **RearSuspension.Gauge** identifier only.

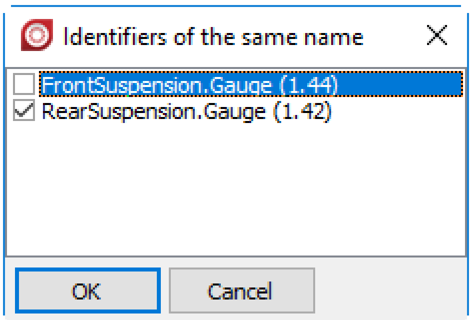


Figure 1.255. Identifiers of the same name for different subsystems

8. Let us configure the graphical objects for wheels so as they to satisfy the mentioned above tire type "*175/80 R16*", where *175* is the nominal width of tire in millimeters; *80* is the ratio of height to width in percent; *16* is the rim diameter in inches. Set the following values for the identifiers listed below for both front and rear suspensions:

- **wheel_tirewidth = 0.175** (m),
- **wheel_tireheight = 0.14** (m),
- **wheel_radius = 0.3432** (m).

Besides that do not forget to specify camber and toe angles for the front suspension as follows:

- **Camber = 0.5**;
- **Toe = 0.125**.

1.12.4.1.3. Connecting Suspension with the Car Body

1. Create a new joint. In the field **Body1** select **CarBody**. In the field **Body2** select the **FrontSuspension.Local Car Body**, see Figure 1.256.

2. Set joint name to **jCarBody_FrontSuspension**.

3. Set joint type to **6 d.o.f.** and turn off all check boxes for degrees of freedom, see Figure 1.256. Via this joint the intermediate **Local Car Body** of the front suspension is rigidly connected to the car body.

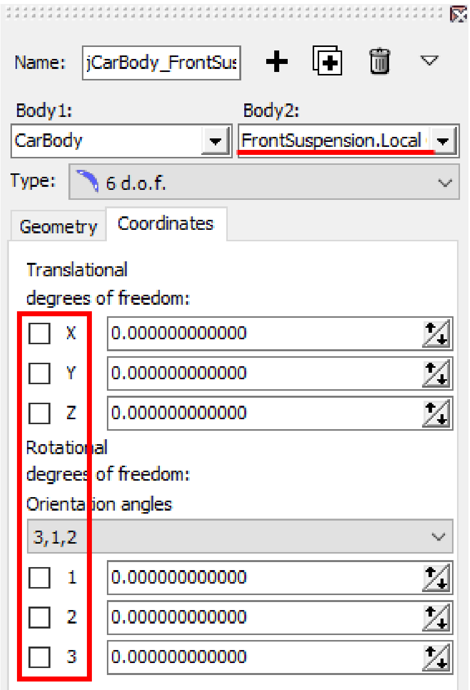


Figure 1.256. Creating a joint for the front suspension

4. Set joint name to **jCarBody_FrontSuspension**. Select the **Geometry | Body 1** tab sheet. In the fields **Translation | x** and **Translation | z** type **X_Pos_FrontSuspension** and **Z_Pos_FrontSuspension** correspondingly, see Figure 1.257, left.

5. Create the joint for the rear suspension in the same way. Select the **Geometry | Body 1** tab sheet. In the fields **Translation | x** and **Translation | z** type **X_Pos_RearSuspension** and **Z_Pos_RearSuspension** correspondingly, see Figure 1.257, right.

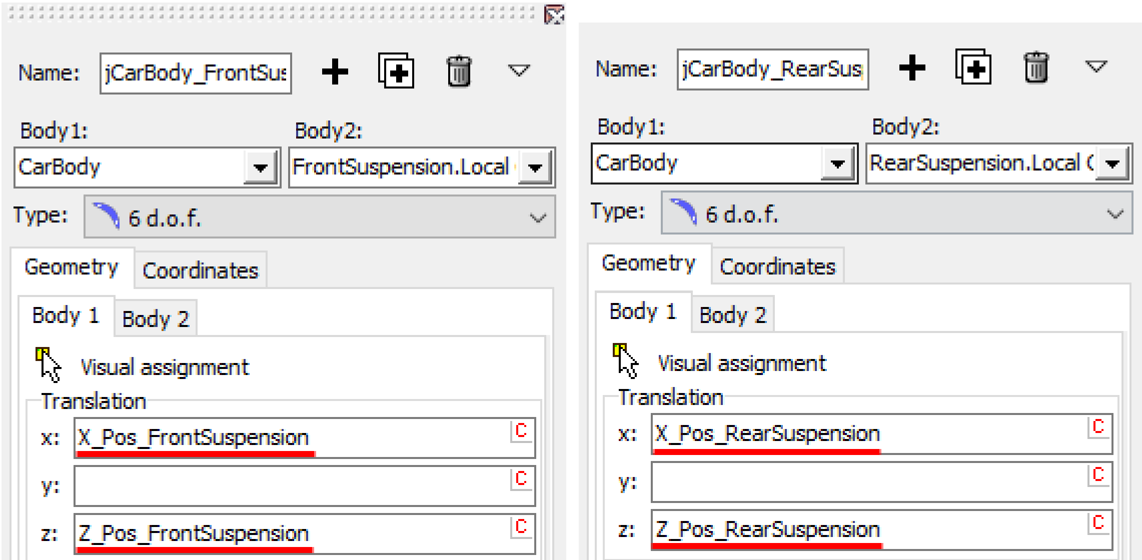


Figure 1.257. Joint points for the front (in the left) and rear (in the right) suspension

The model is ready. Finally your model should look like one depicted in Figure 1.258. Go to Summary node in the tree of elements and check that your model has no errors.

Figure 1.258. Newly created model of the car in UM Input

1.12.4.2. Preparing for Simulation

Now you have to prepare you model for simulations: specify tire models, irregularities, pass through the model identification etc. Detailed description of these steps is given in Sect. 1.9.1. "*Preparing for simulation*", page 1-92.

1.12.4.2.1. Tire Models

Run **UM Simulation** program.

Firstly we will assign the tire model for wheels of the vehicle. For that select the **Analysis | Simulation** menu item and then click the **Road vehicle | Tires** tab sheet in the **Object simulation inspector**. With the help of the **Add type file(s) to the list** button as **Lada4x4.tr** file. And then set this model for all wheels like it is shown in Figure 1.259.

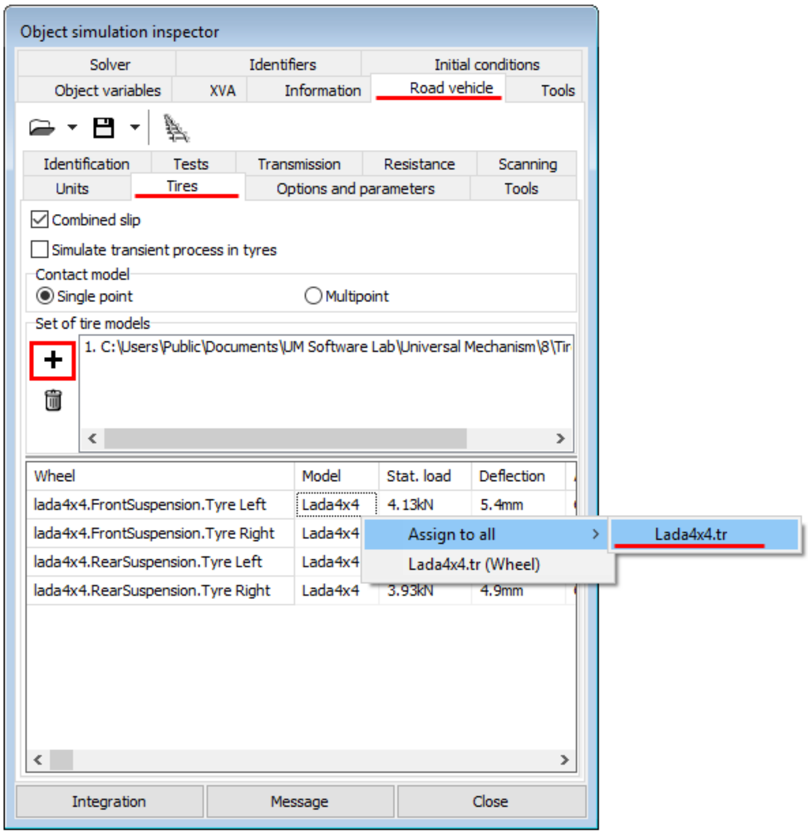


Figure 1.259. Assigning tire model for the vehicle

1.12.4.2.2. Identification of the Model

After that we have to go through the procedure of model identification. Select the **Object simulation inspector** window and then click the **Road vehicle | Identification** tab sheet. In the drop-down list select **Longitudinal speed control**, **Hull horizontal motion locking** and **Steering** and make sure that all parameters are set how they are shown in Figure 1.260-. If some parameters are not set properly by default, set them manually.

Whilst **Steering** identification in the **Index of subsystem for steer wheel angle** field set **1**, and in the **Index of steer wheel angle** field set **20**, see Figure 1.262. The **Index of subsystem for steer wheel angle** can be found as an index of a degree of freedom in the correspondent **Local Car Body_Steering Column** joint that can found in the **Initial conditions** tab. You can find more detailed description of the identification of the steering control in the Sect. 1.9.1.2.3 "*Identification of steering*".

Note The **Steer ratio** parameter is set automatically as a result of the steering wheel rotation test, see Sect. 1.12.4.2.5, "*Steering Wheel Rotation Test*", page 1-237. It should not be set manually on this step.

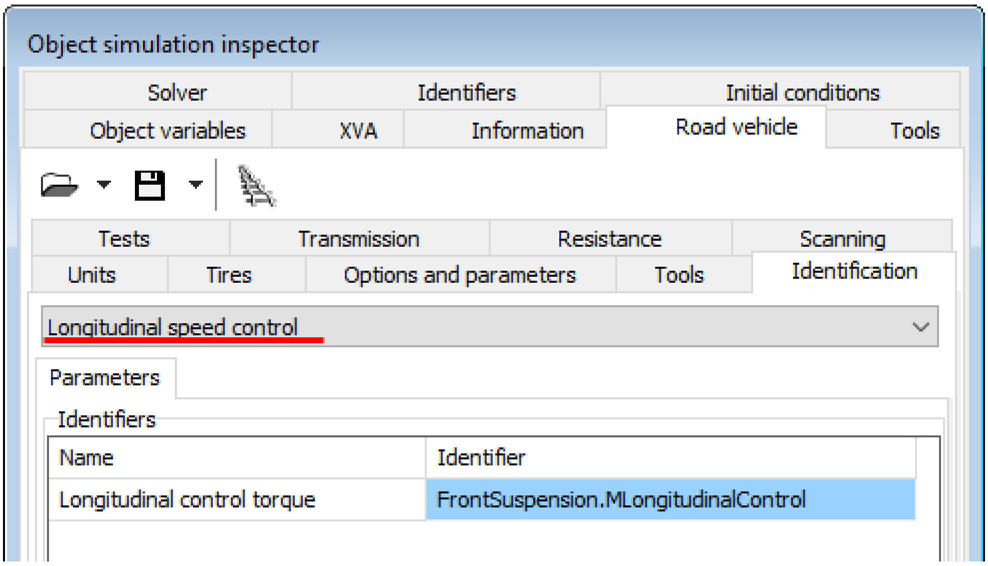


Figure 1.260. Identification of the longitudinal speed control

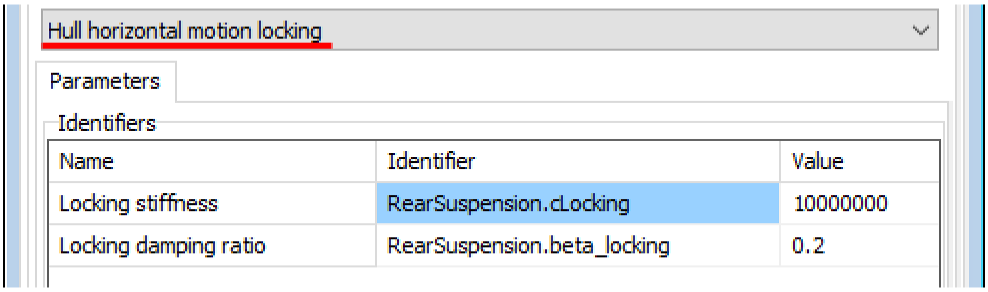


Figure 1.261. Identification of the horizontal motion locking

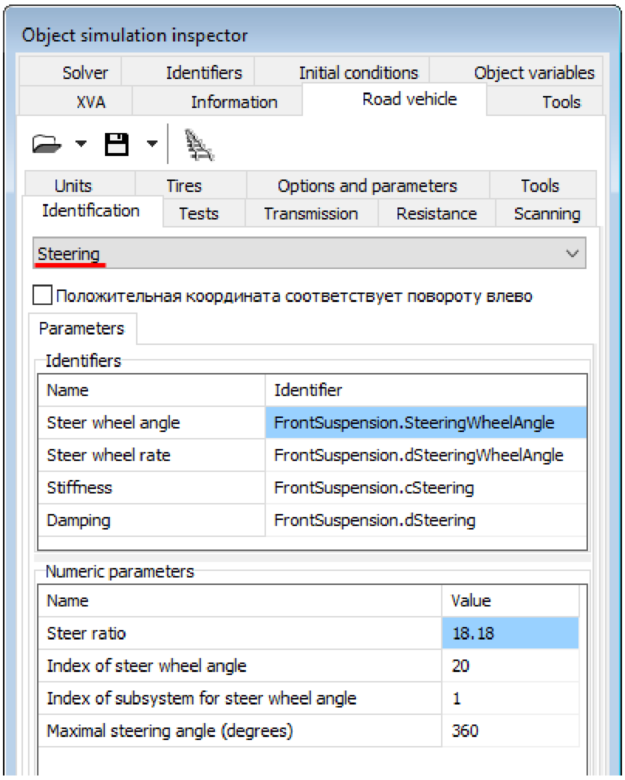


Figure 1.262. Identification of the steering control

1.12.4.2.3. Irregularities

Select the **Road vehicle | Options and parameters** tab and in the fields **Left** and **Right** load irregularity files **asphalt_fine_left.irr** and **asphalt_fine_right.irr** correspondingly, see Figure 1.263. Leave the rest settings by default.

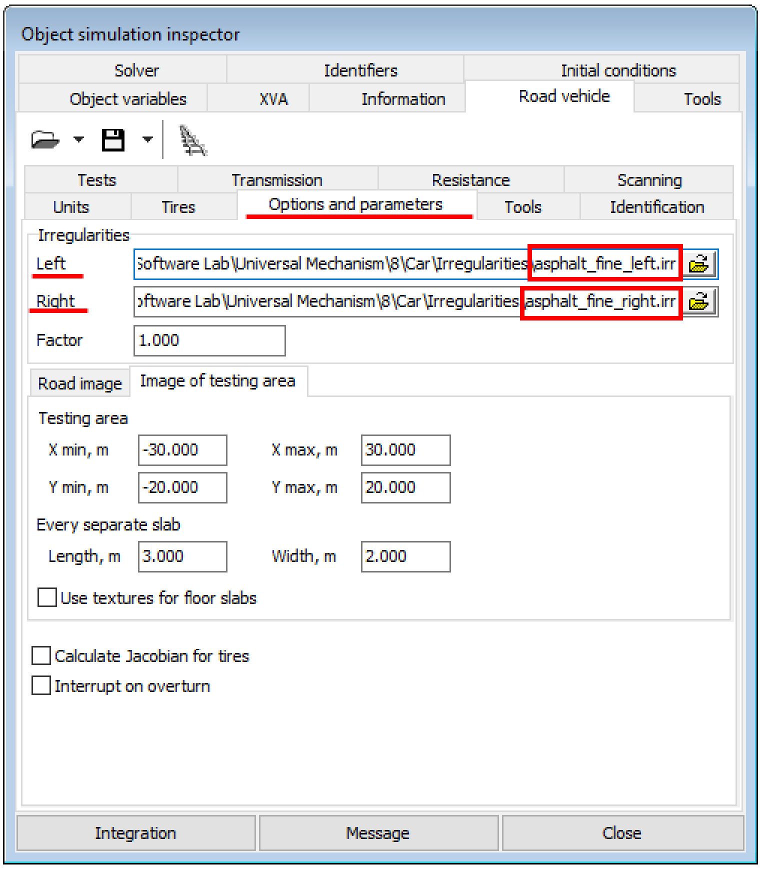


Figure 1.263. Irregularities setting

1.12.4.2.4. Determination of preload for springs of suspensions

To come to the simulation of vehicle dynamics it is necessary to specify preload force for springs of both front and rear suspensions. Spring preload is expressed with the help of **SpringPreload** identifier, see Figure 1.264. Initially on the suspension level preload force is not specified and should be when the complete model of a vehicle is prepared. Let us determine the preload force so as the configuration of the suspension under the weight of the car body would be close to initial configuration (at zero coordinates).

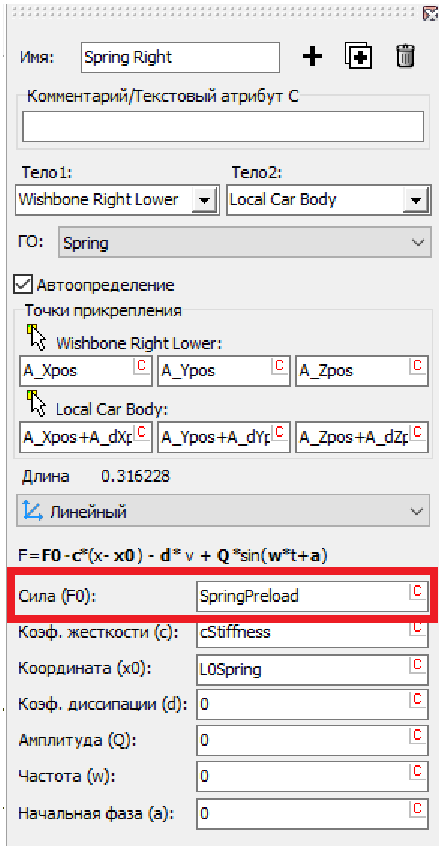


Figure 1.264. Spring preload

We will find the equilibrium position of the model with zero preload forces and obtain magnitude of forces in springs. Then we will specify these obtained forces at equilibrium position as preload forces.

1. Open **Object simulation inspector** and select the **Road vehicle | Tests** tab sheet. Select the **Equilibrium test** in the drop-down list.
2. Then in the **Parameters** tab set **Minimal time (s)** to **10 (s)**.
3. Open **Wizard of variables** and select the **Bipolar forces** tab sheet. Then select the **Spring Right** and **Spring Left** forces for the front and the rear suspension, in the **Component** group select **Force magnitude** as it is shown in Figure 1.265. Create new variables and drag&drop them into the new graphical window.

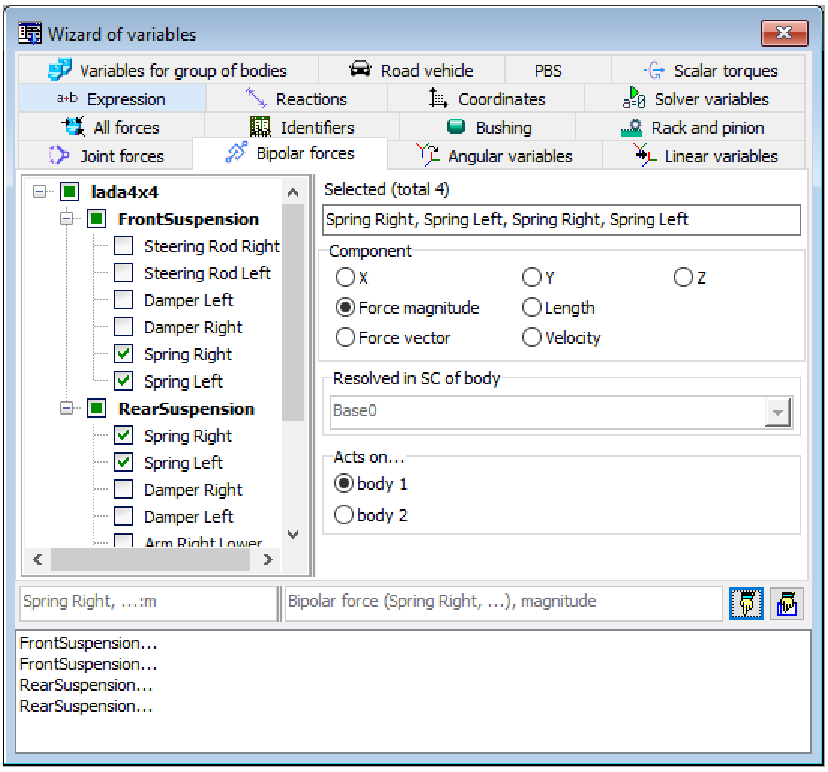


Figure 1.265. Force magnitude for springs

4. Select **Object simulation inspector** and click the **Integration** button. When simulation finishes select the graphical window, turn on the "**Show ordinate value**" and pick the plot values close to the end of simulation time when the equilibrium position is reached, see Figure 1.266. Round and average the obtained values as follows – 3970 N for the front suspension and 2535 N for the rear one.

5. In the **Pause** window click the **Interrupt** button.

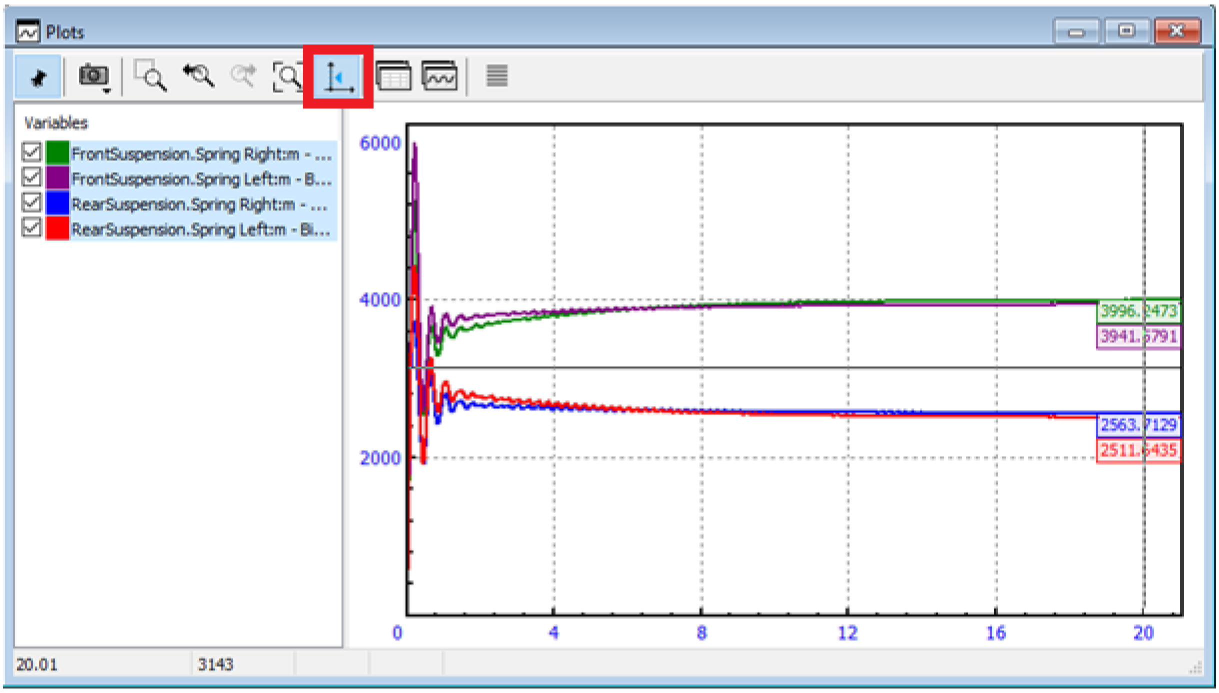


Figure 1.266. Spring forces at equilibrium position

6. In the **Object simulation inspector** select the **Identifiers** tab sheet. In the drop-down list select the **lada4x4.FrontSuspension** subsystem, see Figure 1.267. Set the **SpringPreload** identifier to **3970** N for the front suspension.

7. Then select the **lada4x4.RearSuspension** and set **SpringPreload = 2535** N for the rear suspension.

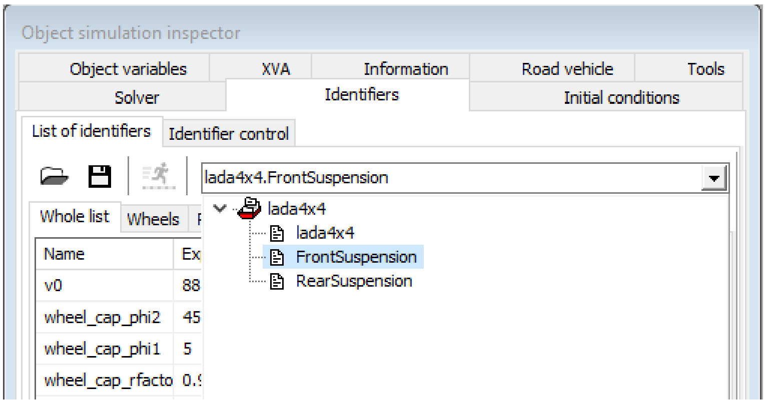


Figure 1.267. Select the subsystem

8. Come back to **Object simulation inspector | Road vehicle | Tests** tab sheet. Turn on the **Accept coordinates after test finish** flag and run simulation.

9. When the tests finishes select the **Object simulation inspector | Initial conditions** tab sheet. Now you can see initial conditions that correspond to equilibrium position of the vehicle, see Figure 1.268.

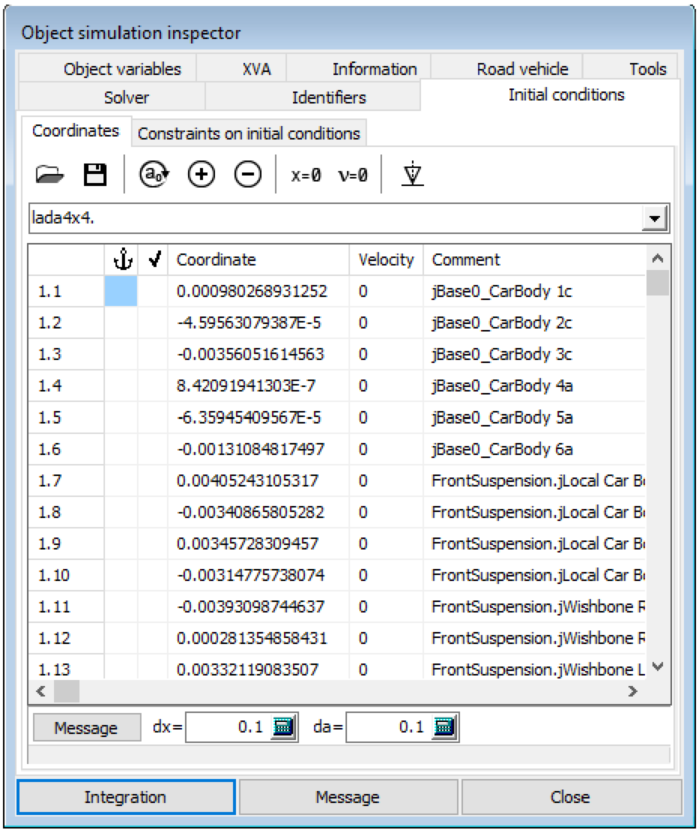


Figure 1.268. Equilibrium position

1.12.4.2.5. Steering Wheel Rotation Test

Steering wheel rotation test helps you to check your model and obtain steering ratio that is used for driver models, see Sect. 1.9.4.5. "*Steering wheel rotation test*", page 1-156.

To watch steering wheel rotation you should make the car body transparent or invisible. Select an animation window and in the context menu select the **Modes of images | Object display settings menu item**, see Figure 1.269, and set **Invisible** mode for the car body, see Figure 1.270.

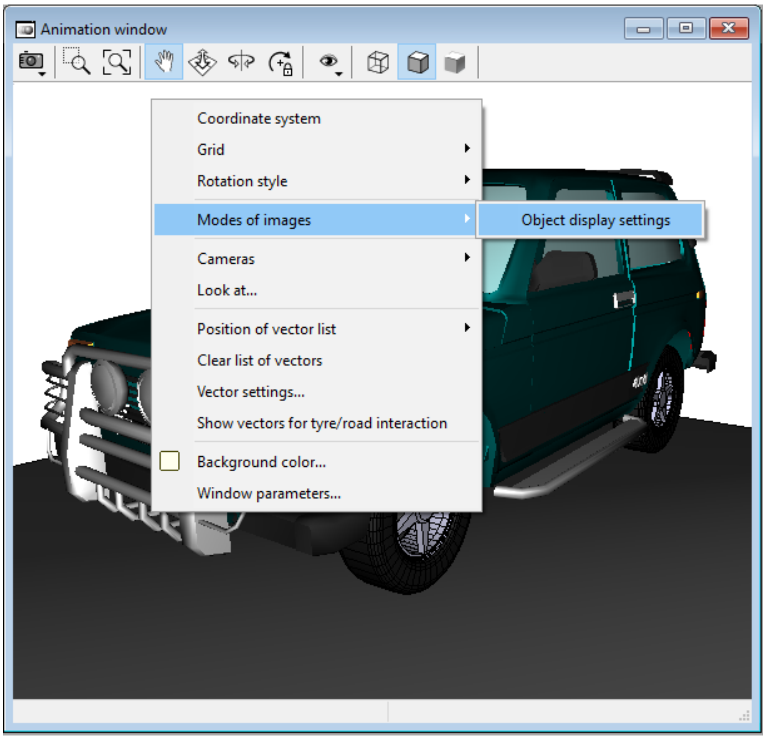


Figure 1.269. Modes of images | Object display settings

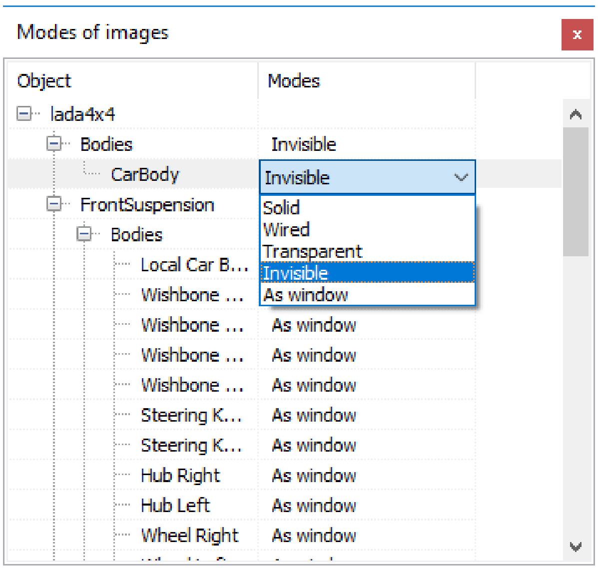


Figure 1.270. Invisible image mode for the car body

Open the **Object simulation inspector** and select the **Road vehicle | Tests** tab sheet. Select the **Steering wheel rotation** test in the drop-down list. Set **Amplitude** and **Frequency** as it is shown in Figure 1.271.

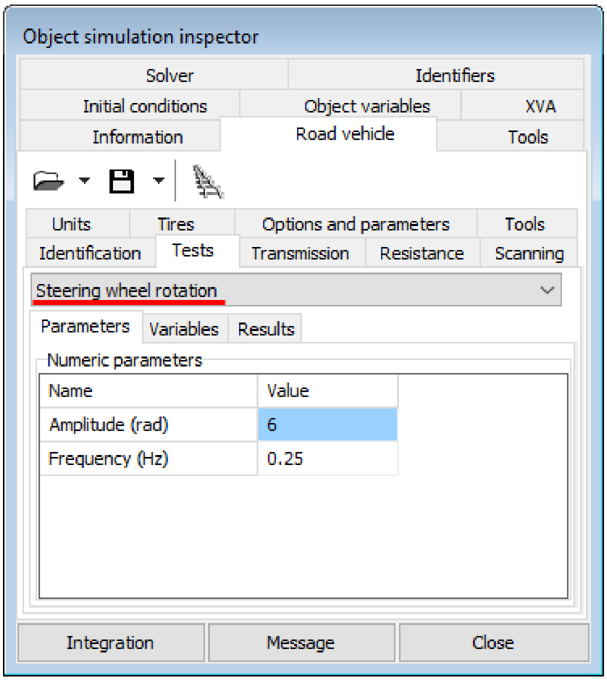


Figure 1.271. Numeric parameters for steering wheel rotation test

Click the **Integration** button. When simulation finishes click the Interrupt button.

In the **Object simulation inspector** select the **Results** tab and click **Accept as standard** to use the obtained steer ratio in the future tests with driver, see Figure 1.272.

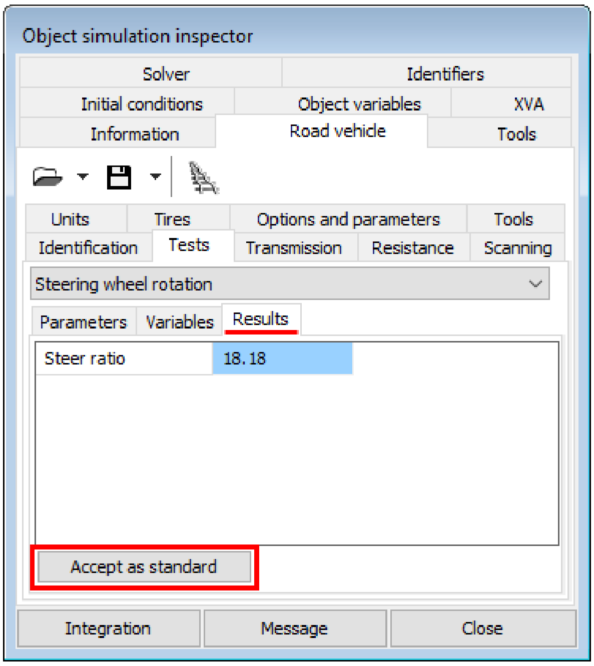


Figure 1.272. Obtained steer ratio

1.12.4.3. Tests with Driver

1.12.4.3.1. Low-Speed 90 º Turn

1. Prior coming to the rest part of this manual select the **Tools | Options** menu item, click the **General** tab sheet and in the **Speed unit** field select **km/h**, see Figure 1.273. Click **OK** to close the **Options** window.

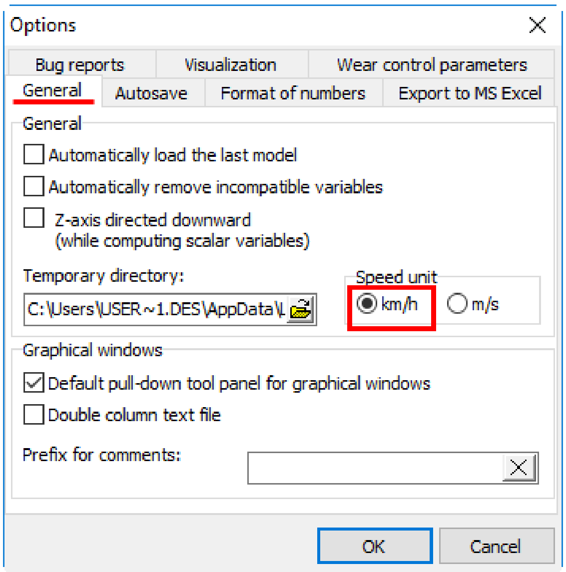


Figure 1.273. Speed unit

2. Then select the **Object simulation inspector** and click the **Identifiers | Whole list** tab sheet. Set **v0** to **5** km/h, see Figure 1.274.

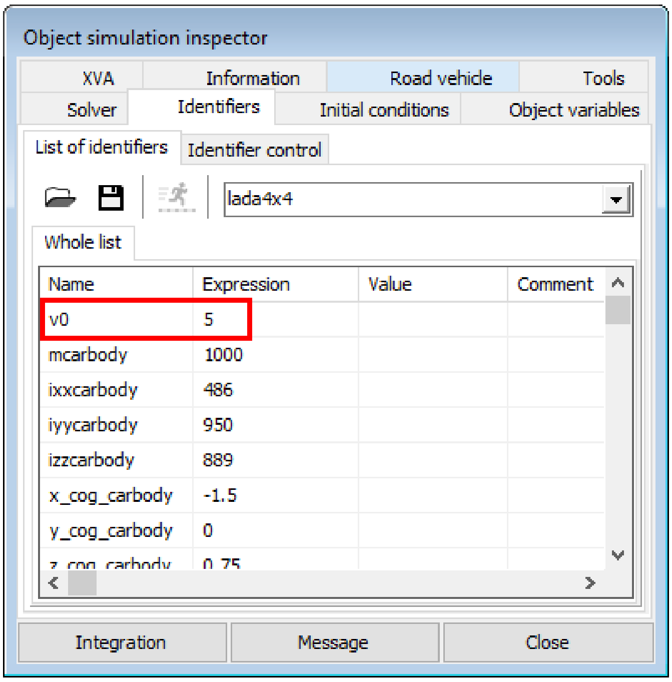


Figure 1.274. Initial speed of the vehicle is 5 km/h

3. Then select the **Solver** tab sheet and set simulation time to 50 seconds.

4. Then click **Road vehicle | Tests | Parameters** tab sheet and from the drop-down list select **Test with driver**. Then specify the turn **90deg.mgf** file as a **Macro geometry file**. In the **Driver model** field select the **MacAdam** model. The rest parameters set as they are shown in Figure 1.275.

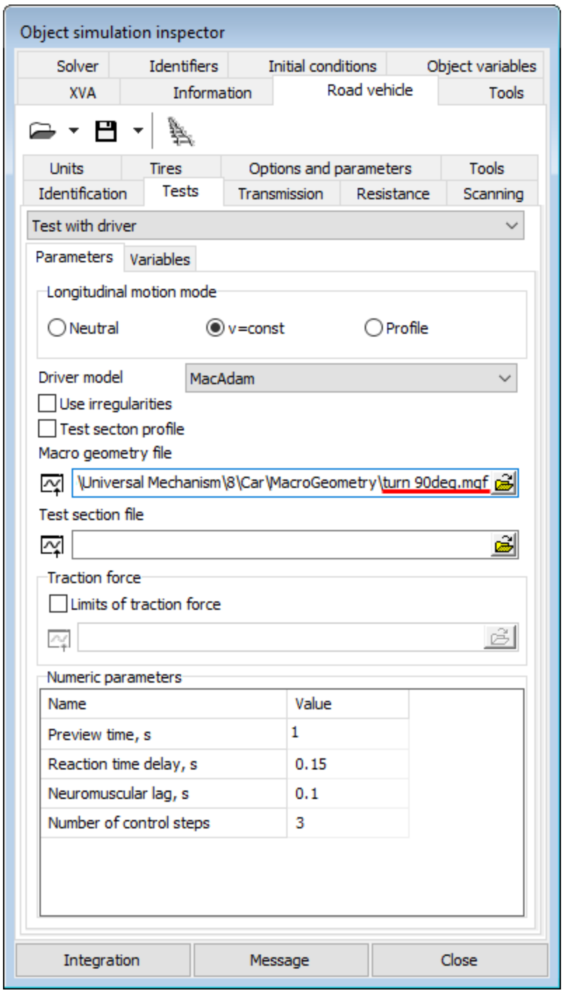


Figure 1.275. Settings for low-speed 90 degrees turn

5. Select the **Road vehicle | Tests | Variables** tab sheet. Create new graphical window and drag&drop there the **Desired path deviation** variable, see Figure 1.276.

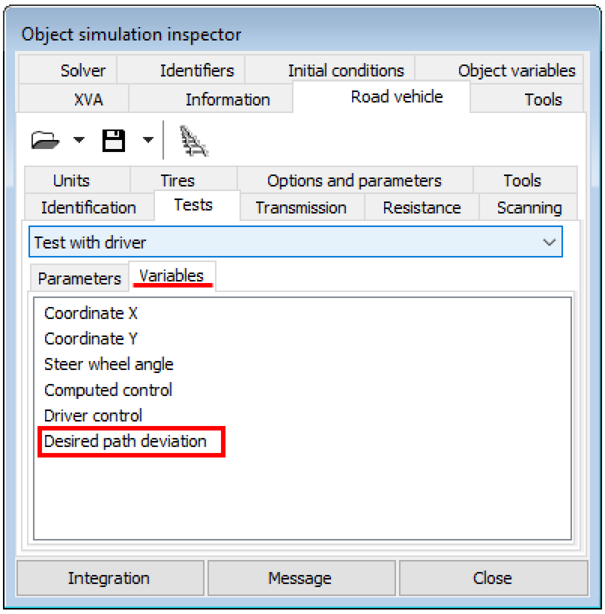


Figure 1.276. Desired path deviation

6. Before simulation open an animation windows, if none is opened, and adjust the viewpoint.
7. In the **Object simulation inspector** click Integration. When simulation finishes check how close or far the actual vehicle path from the desired one.
8. Click **Interrupt** to close the **Pause** inspector.

1.12.4.3.2. Lane Change Manoeuvre

1. Select the **Identifiers | List of identifiers** tab sheet and set **v0** = **88** km/h, see. Figure 1.277.

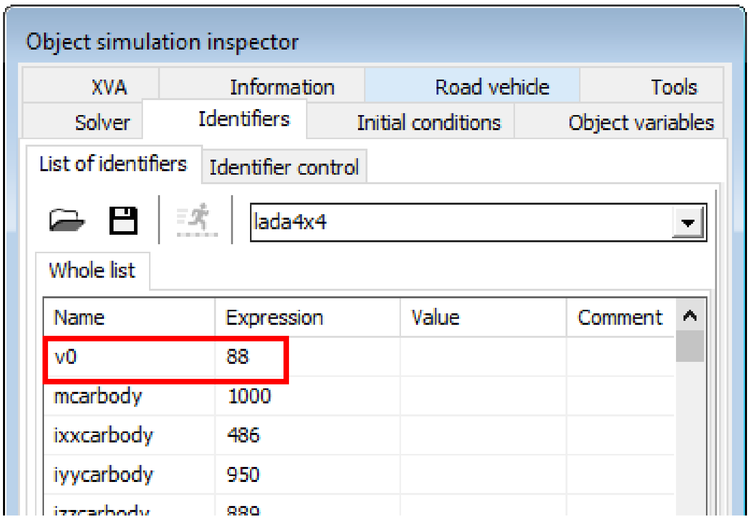


Figure 1.277. Initial speed of the vehicle is 88 km/h

2. Select the **Road vehicle | Tests** tab sheet. In the **Macro geometry file** select "**SAE j2179 single lane change.mgf**". Set **Driver model** to **Second order preview**. Set the rest parameters as it is shown in Figure 1.278.

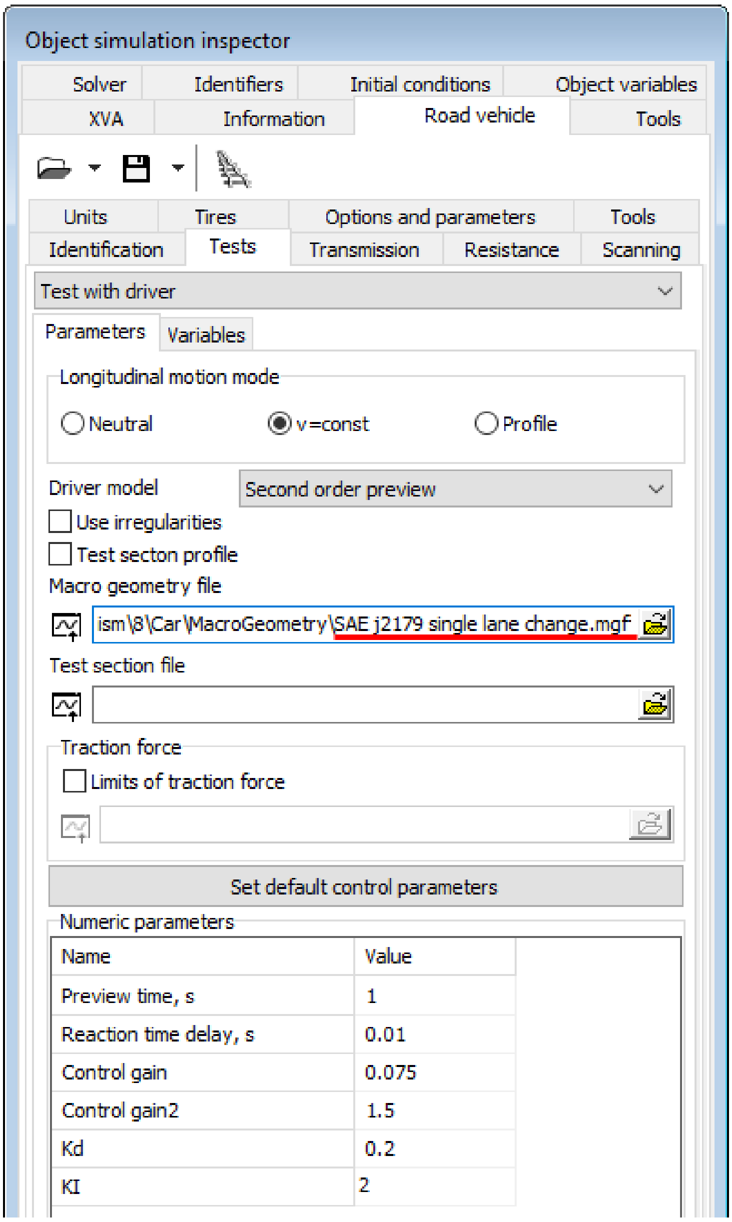


Figure 1.278. Settings for lane change manoeuvre

3. Click **Integration**. Watch the simulation process in the animation window.

Note	Pre-configured tests already prepared in the model's folder. Use **File | Load configuration** menu item to load them.

1.12.5. Available Car Models and Configurations

Universal Mechanism includes models of Lada 4x4, Audi Q7, GAZ-66, Opel Astra, Red American, BMW 3 series with pre-configured settings for *low-speed turn* and *SAE lane change* in the folder {UM Data}\SAMPELS\Automotive.

1.12.5.1. BMW 3 Series

UM library includes classic rear-wheel drive car of BMW 3 series with E36 car body. You can find more detailed information about that car via the following link: https://en.wikipedia.org/wiki/BMW_3_Series_(E36). The McPherson suspension (see Sect.1.12.2.4. "*MacPherson Suspension*", page 1-216) is used as a front suspension, and the semi-trailing arm suspension (see Sect. 1.12.2.3. "*Semi-Trailing Arm Suspension*", page 1-215) is used as a rear suspension. The gauge of the front wheels is 1418 mm and 1423 mm stands for rear ones. The follows "*factory settings*" were used for the camber and toe angles: camber is 1.167 º, toe is 0.3 º. Tires *195/65 R15* were used as default settings. The correspondent UM file for tire model is located in the following folder:{UM Data}\Tire.

Model folder: {UM Data}\SAMPLES\Automotive\BMW3_E36.

Figure 1.279. UM-model of BMW 3 series (E36)

References

[1] Khachaturov A.A., Dynamics of system roadway – tyre – vehicle – driver. Moscow, “Mashinostroenije”. 1976.

[2] Wong J.Y., Theory of Ground Vehicles. 4th Edition. Wiley. 2008.

[3] Robson J.D., (1979) Road Surface Description and Vehicle Response, International Journal of Vehicle Design,. 1(1), 25–35.

[4] Dixon J., Suspension Analysis and Computational Geometry. John Wiley and Sons, 14.12.2009. 417 P.

[5] Bakker E., Pacejka H.B. and Lidner L., A New Tyre Model with Application in Vehicle Dynamics Studies. Proc. 4th Int. Conf. Automotive Technologies, Monte Carlo, 1989, SAE paper 890087, 1989.

[6] Pacejka H.B. and Bakker E., The Magic Formula Tyre Model. Proc. 1st International Tyre Colloquium, Delft, 1991. Vehicle System Dynamics 21 (Suppl.) (1991), pp. 1–18.

[7] Ervin R.D. and Guy Y., Vehicle Weights and Dimensions Study: Volume 1 – The Influence of Weights and Dimensions on the Stability and Control of Heavy Trucks in Can-ada Part 1. Roads and Transportation Association of Canada: Ottawa, Canada, 1986.

[8] Georg Rill, Road Vehicle Dynamics: Fundamentals and Modeling. CRC Press, 2012.

[9] Mohhamadi Foad, Tire Characteristics Sensitivity Study. Chalmers University Of Technology. Gothenburg, Sweden 2012.

[10] Hans B. Pacejka. Tyre and vehicle dynamics. Second edition. Elsevier, 2006.

[11] Nybakken G.H., Clark S.K., Vertical and lateral stiffness characteristics of aircraft tires. NASA contractor report NAS CR-1488, University of Michigan, 1969.

[12] Szostak H.T., Allen W.R., Rosenthal T.J., Analytical Modeling of Driver Response in Crash Avoidance Maneuvering Volume II: An Interactive Model for Driver/Vehicle Simulation, U.S Department of Transportation Report NHTSA DOT HS-807-271, April 1988.

[13] Prof. Dr.-Ing. habil M. Hiller & Dipl.-Ing. S. Frik (1993) Road Vehicle Benchmark 2 Five-Link Suspension, Vehicle System Dynamics, 22:S1, 254-262, DOI: 10.1080/00423119308969497.

[14] Reimpel J. The Automotive Chassis: Suspension Design. Moscow, “Mashinostroenije”. 1989.

[15] Reimpel J. The Automotive Chassis: Steering. Moscow. “Mashinostroenije”. 1987.

[16] ISO 3888-1:1999 Passenger cars – Test track for a severe lane-change manoeuvre – Part 1: Double line-change.

[17] ISO 3888-2:2002 Passenger cars – Test track for a severe lane-change manoeuvre – Part 2: Obstacle avoidance.

[18] ISO 4138:2004 Passenger cars – Steady-state circular driving behaviour – Open-loop test methods.

[19] ISO 7401:2003 Road vehicles – Lateral transient response test methods – Open-loop test methods.

[20] ISO 7975:1996 Passenger cars – Braking in a turn – Open-loop test procedure.

[21] ISO/TR 8725:1988 Road vehicles – Transient open-loop response test method with one period of sinusoidal input.

[22] ISO/TR 8726:1988 Road vehicles – Transient open-loop response test method with pseudo random steering input.

[23] ISO 9816:1993 Passenger cars – Power-off reactions of a vehicle in a turn – Open loop test method.

[24] ISO 12021-1:1996 Road vehicles – Sensitivity to lateral wind – Part 1: Open loop test method using wind generator input.

[25] ISO 14512:1999 Passenger cars – Straight ahead braking on surface with split coefficient of friction – Open loop test procedure.

[26] Wade Allen R., Theodore J. Rosenthal et al., A LOW COST PC BASED DRIVING SIMULATOR FOR PROTOTYPING AND HARDWARE-IN-THE-LOOP APPLICATIONS. SAE Paper No. 98-0222. 1997.

[27] Раймпель Й., Шасси автомобиля: Конструкции подвесок/Пер. с нем. В. П. Агапова. – М.: Машиностроение, 1989. – 328 с.

[28] Раймпель Й., Шасси автомобиля. Рулевое управление. М.: Машиностроение, 1987.

[29] Хачатуров А.А., Динамика системы дорога – шина – автомобиль – водитель. М., «Машиностроение». 1976.

[30] ISO 3888-1:1999 Passenger cars - Test track for a severe lane-change manoeuvre – Part 1: Double line-change.

[31] ISO 14512:1999 Passenger cars - Straight ahead braking on surface with split coefficient of friction – Open loop test procedure.