UNIVERSAL MECHANISM 9

User`s manual

# Simulation of road vehicles

Simulation of road vehicle dynamics is considered

**2020**

# Contents

# 12. UM Module for simulation of road vehicles

## 12.1. 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.

## 12.2. Base system of coordinates

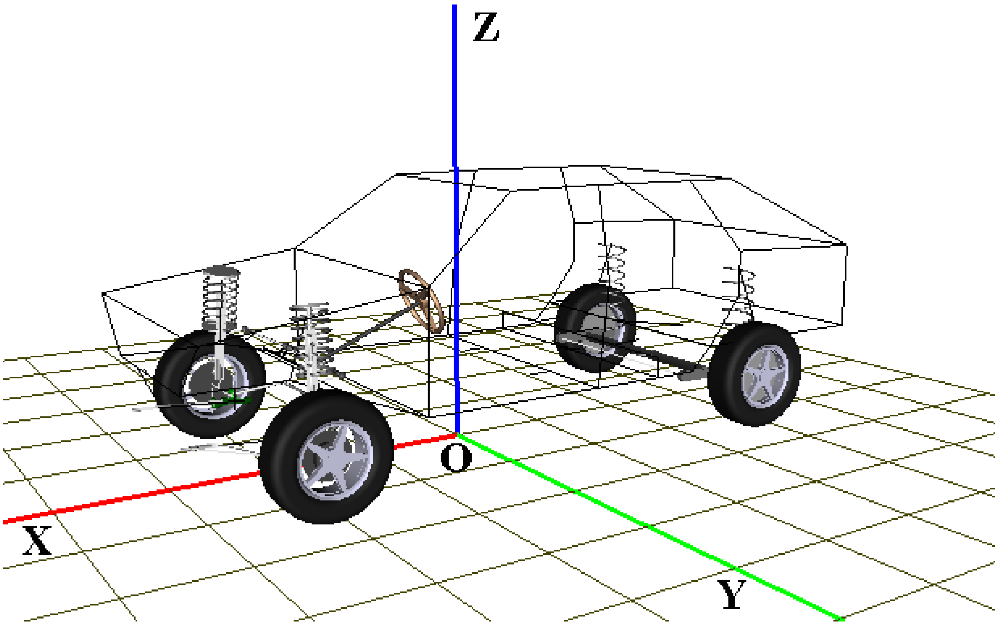



Figure 12.1. Base system of coordinates (SC0)

Inertial system of coordinates (SC0) in UM Automotive meets the following requirements (Figure 12.1):

- 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.

## 12.3. 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. A pair of vertical and horizontal profiles builds macro geometry of a track.

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.).

### 12.3.1. Track macro geometry

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 ($\gamma_i$, $s_i$), where $\gamma_i$ 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 | Create macrogeometry…** menu command. The wizard of macro geometry appears (Figure 12.2)

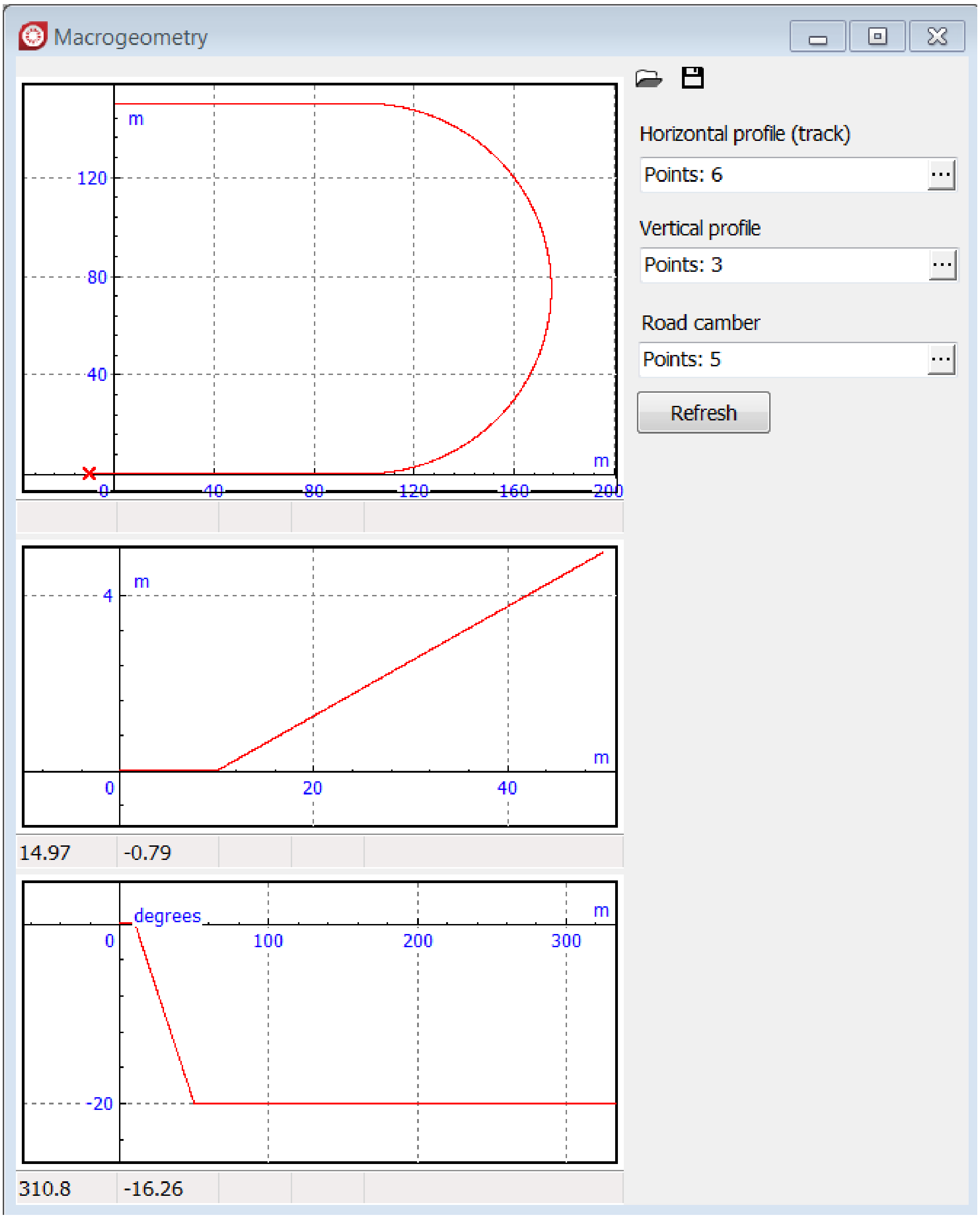



Figure 12.2. Wizard of macro geometry. Horizontal (upper plot), vertical (middle plot) profiles and road camber profile (lower plot)

Curves of profiles are created in the curve editor by clicking the ⋯ button (Figure 12.3). See Chapter 3, *Curve Editor* for more information.

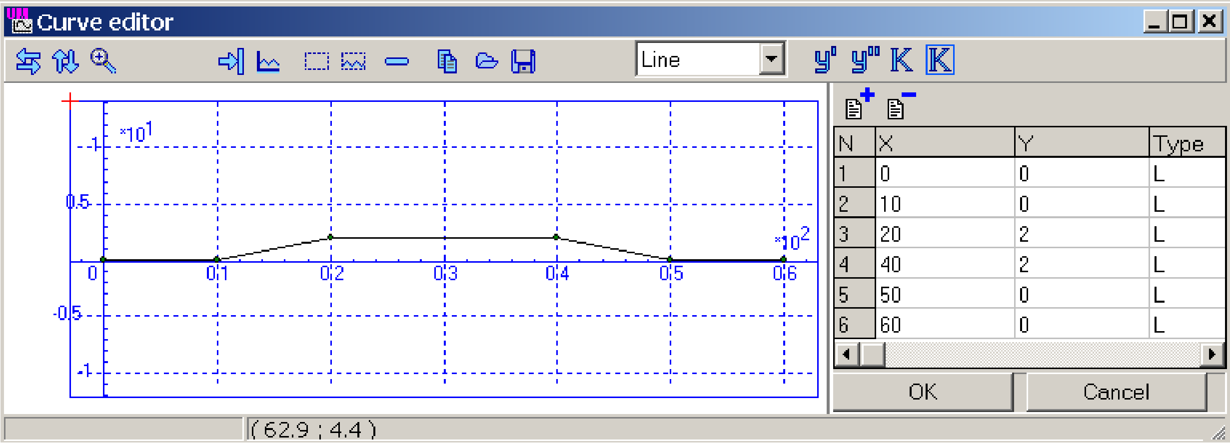



Figure 12.3. Curve editor

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.

**Remark** 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.

Text data corresponding to the macro geometry in Figure 12.2 is shown below.

```
 trackxy={
0     0     L
10    0     L
20    2     L
40    2     L
50    0     L
60    0     L };
trackz={
0     0     L
16.5  0.341 L
31.888739598798   0.5   C
60    0.5   L };
with end;
```

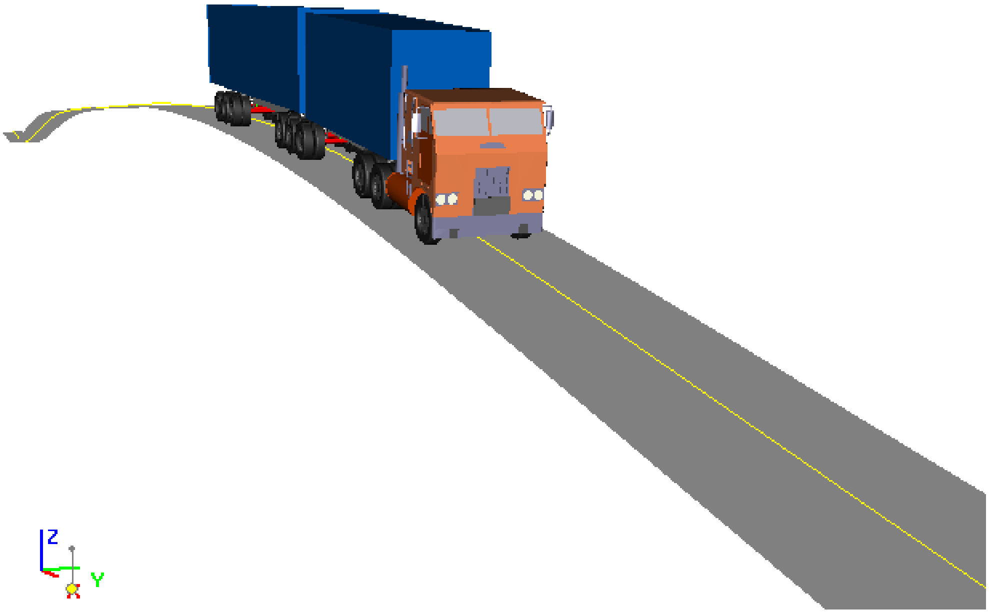



Figure 12.4. Vertical macro profile

**Note.** The PID-SOP driver model uses the derivative of the path following error (Sect. 12.4.3. *"Combination of PID controller and second order preview model"*, p. 12-31), which requires a differentiable function of the desired path. In this case a spline interpolation of the path curve is necessary.

## 12.3.2. 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
0.2  -0.0283658
0.3  -0.0294865
0.4  -0.0299168
0.5  -0.0298581
0.6  -0.0297213
```

---

[1] From ‘irregularities’

0.7  -0.029892
0.8  -0.0304888

### 12.3.2.1. 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. 12.3.2.2. *"Generation of irregularity files"*, p. 12-13.

#### 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 12.5). Note that the frequency in the above files is measured in rad/s, and **the Angular Frequency** key must be on.

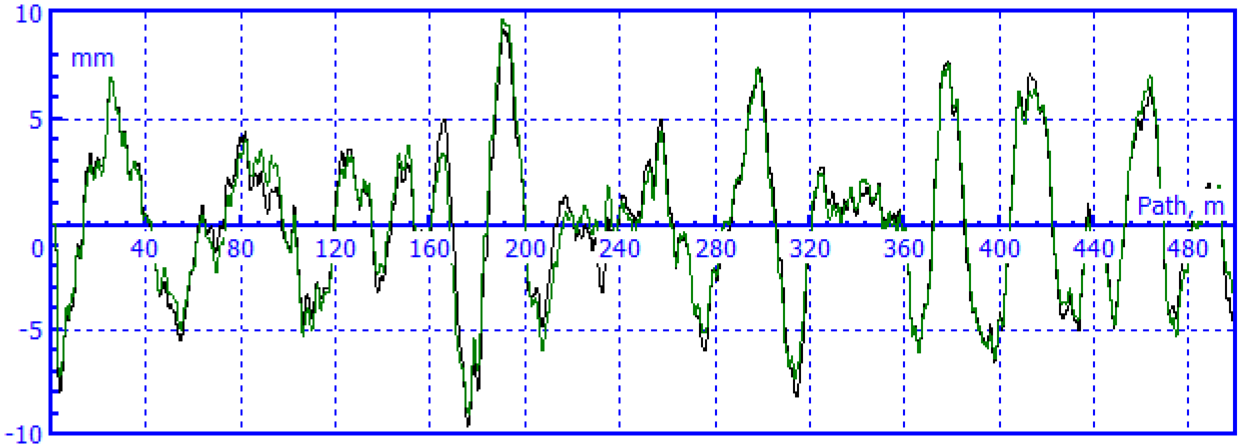
#### Irregularities

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

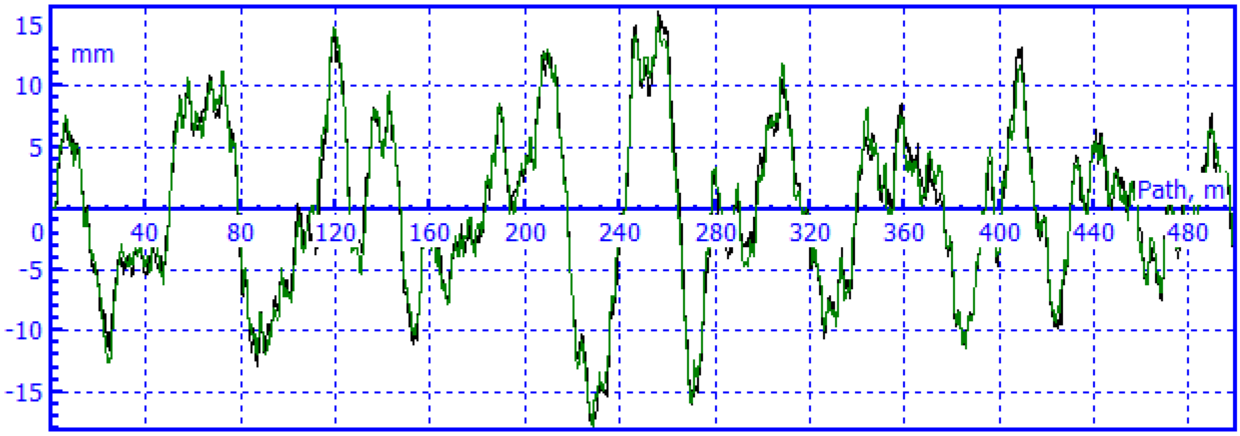
| **File *.irr** | **Comments*** |
|---|---|
| concrete_left, concrete_right | Concrete on rigid foundation |
| asphalt_fine_left, asphalt_fine_right | Asphalt, good state |
| asphalt_satisfactory_left, asphalt_satisfactory_right | Asphalt, satisfactory state |
| cobble_left, cobble_right | Cobblestone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

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

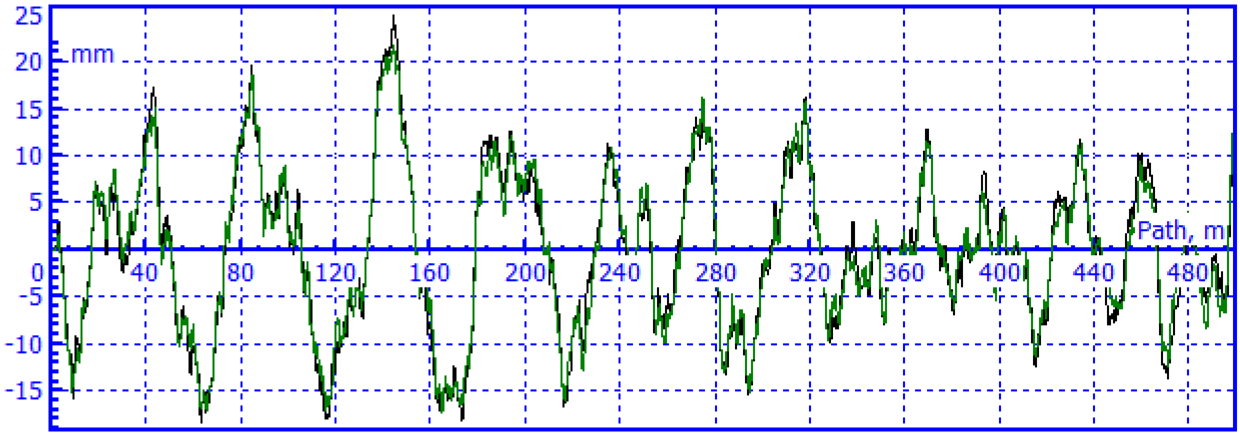
Plots of left and right irregularities from the library are shown below.



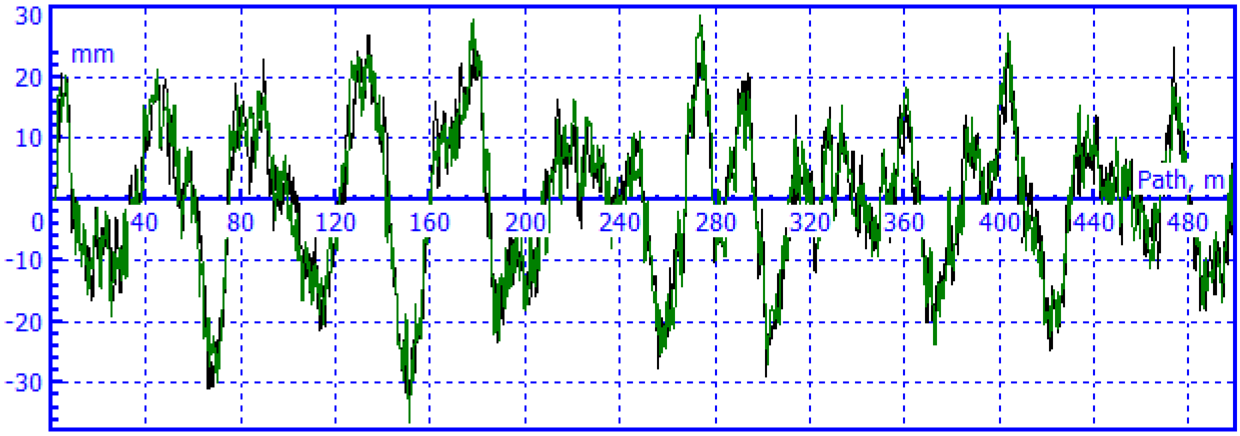

Irregularities “concrete”




Irregularities “asphalt_fine”




Irregularities “asphalt_satisfactory”




Irregularities “cobble”

#### 12.3.2.2. Generation of irregularity files

##### 12.3.2.2.1. Wizard of irregularities

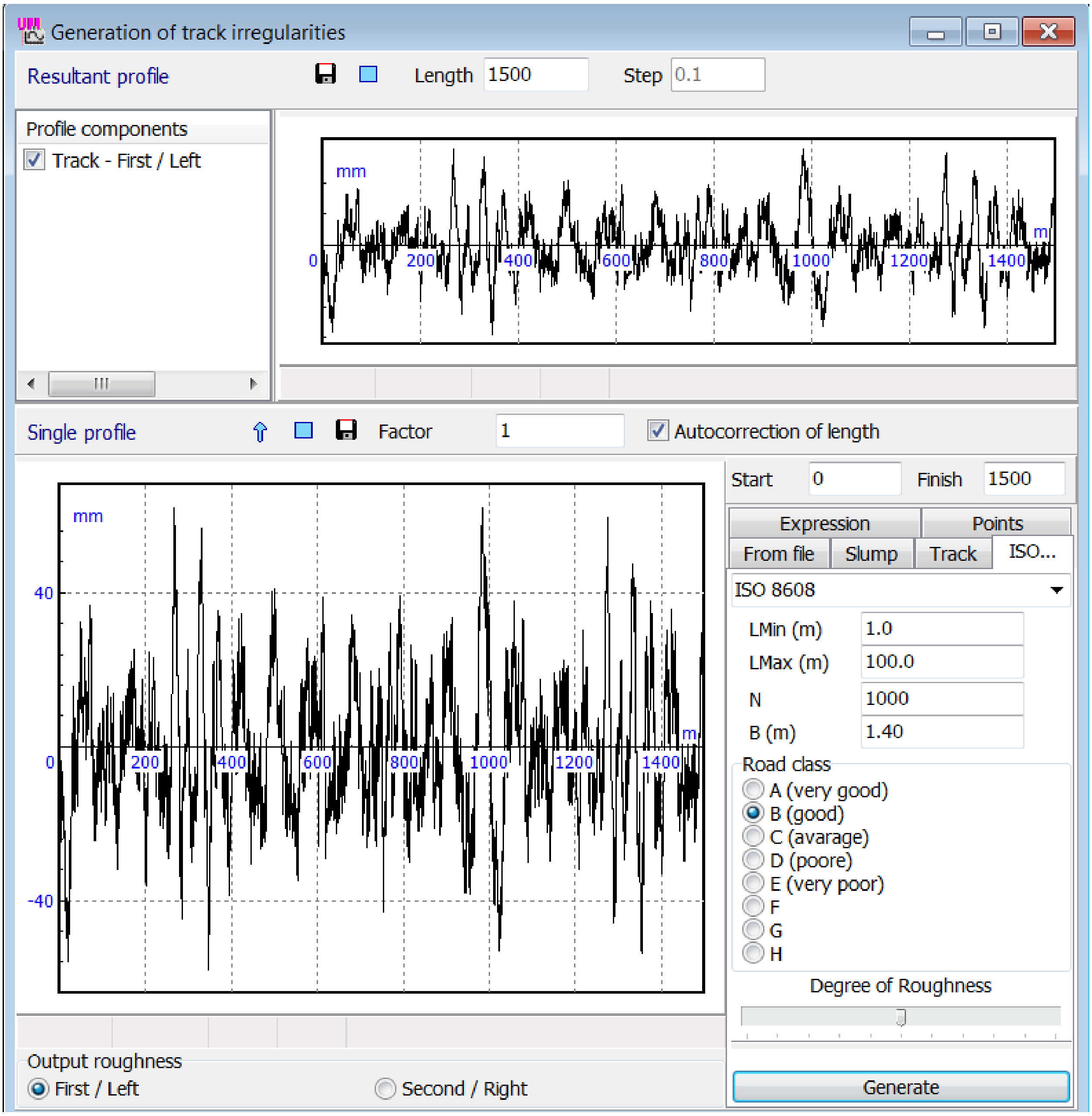



Figure 12.5. Wizard of irregularities

A new file of irregularities is created with a special tool, which is available in the **UM Simulation** program by clicking the **Tools | Create irregularities…** menu command (Figure 12.5).

*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 12.5). 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.

## 12.3.2.2.2. 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 $\Delta s$ 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, $\Delta n$ is the step size of frequency; $n_0$ is the minimal frequency, $\varphi_i$ 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 12.6. 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 $z_l = z_+ + z_-,\ z_r = z_+ - z_-$.

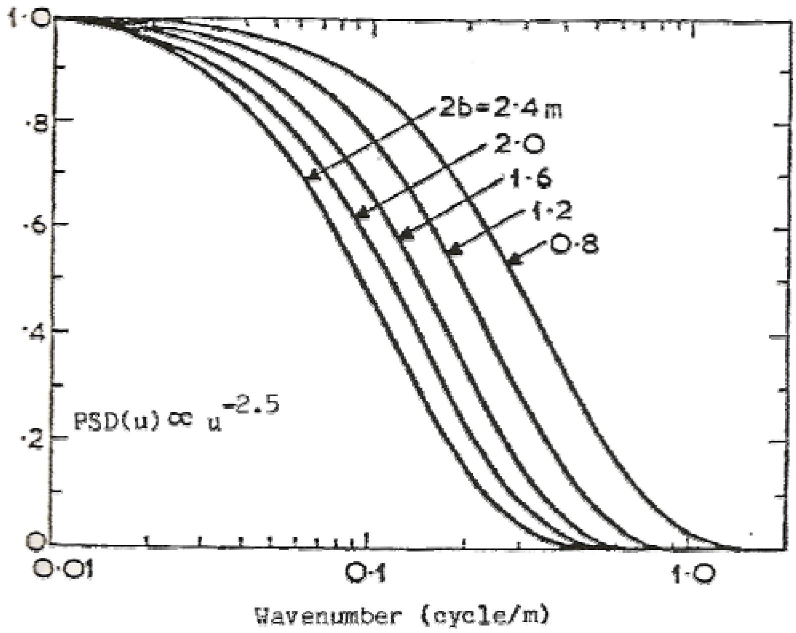



Figure 12.6. Coherence function for different values of track width [3]

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

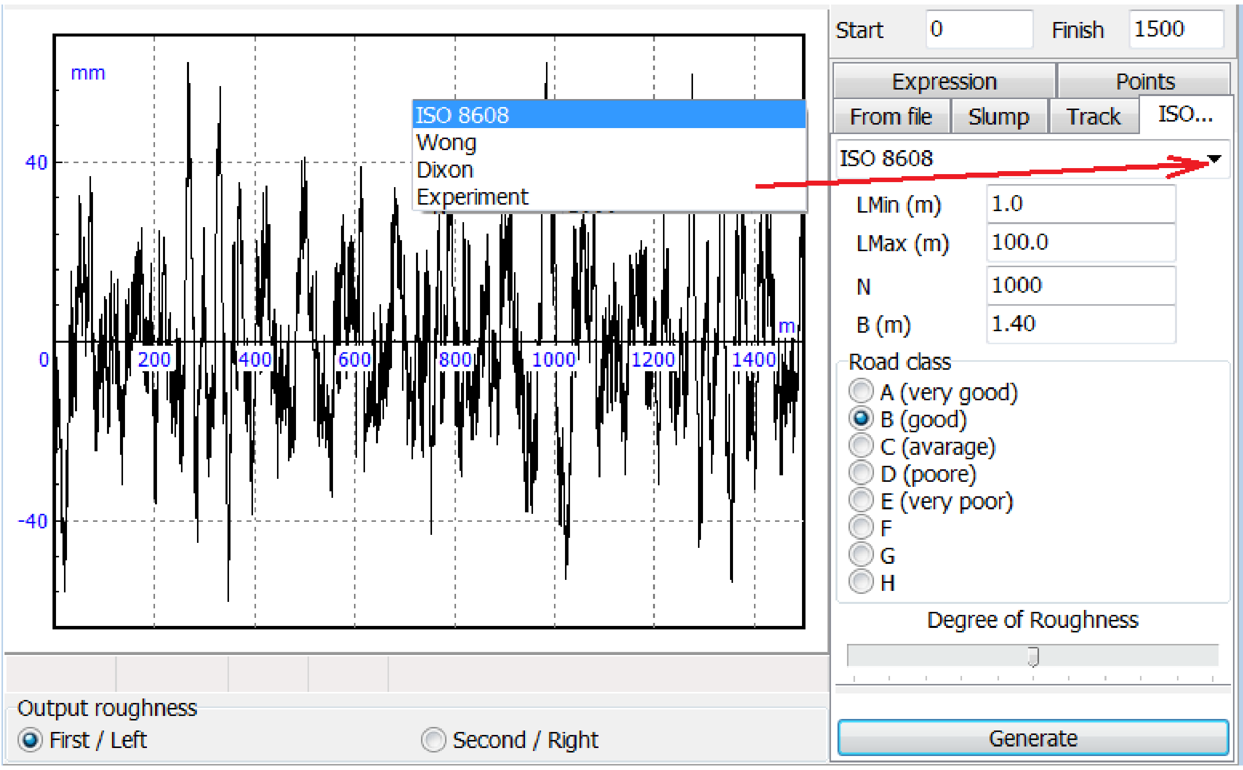



Figure 12.7. Generator of the left and right track roughness

To generate coherent irregularities for the left and right track, the following steps are necessary, Figure 12.7:

- 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. 12.3.2.2.1. *"Wizard of irregularities"*, p. 12-13.

Now consider different types of PSD functions implemented in UM.

#### 12.3.2.2.3.1. ISO 8608

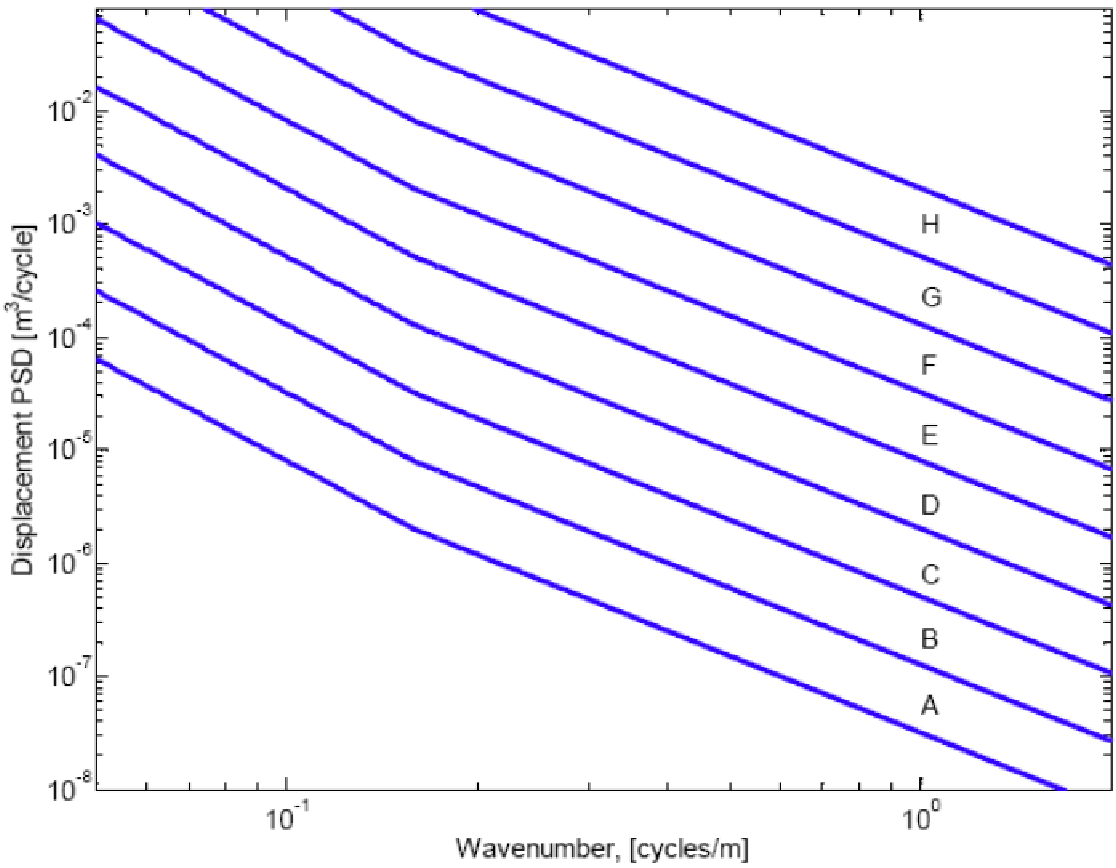



Figure 12.8. SPD function by ISO 8608

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 12.8:

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

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 12.1.

Table 12.1

**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 12.7.

Figure 12.9 shows roughness of class B, $S_0 = 20 \times 10^{-6}$, LMin = 3m, LMax=30m, number of harmonics N=3000.

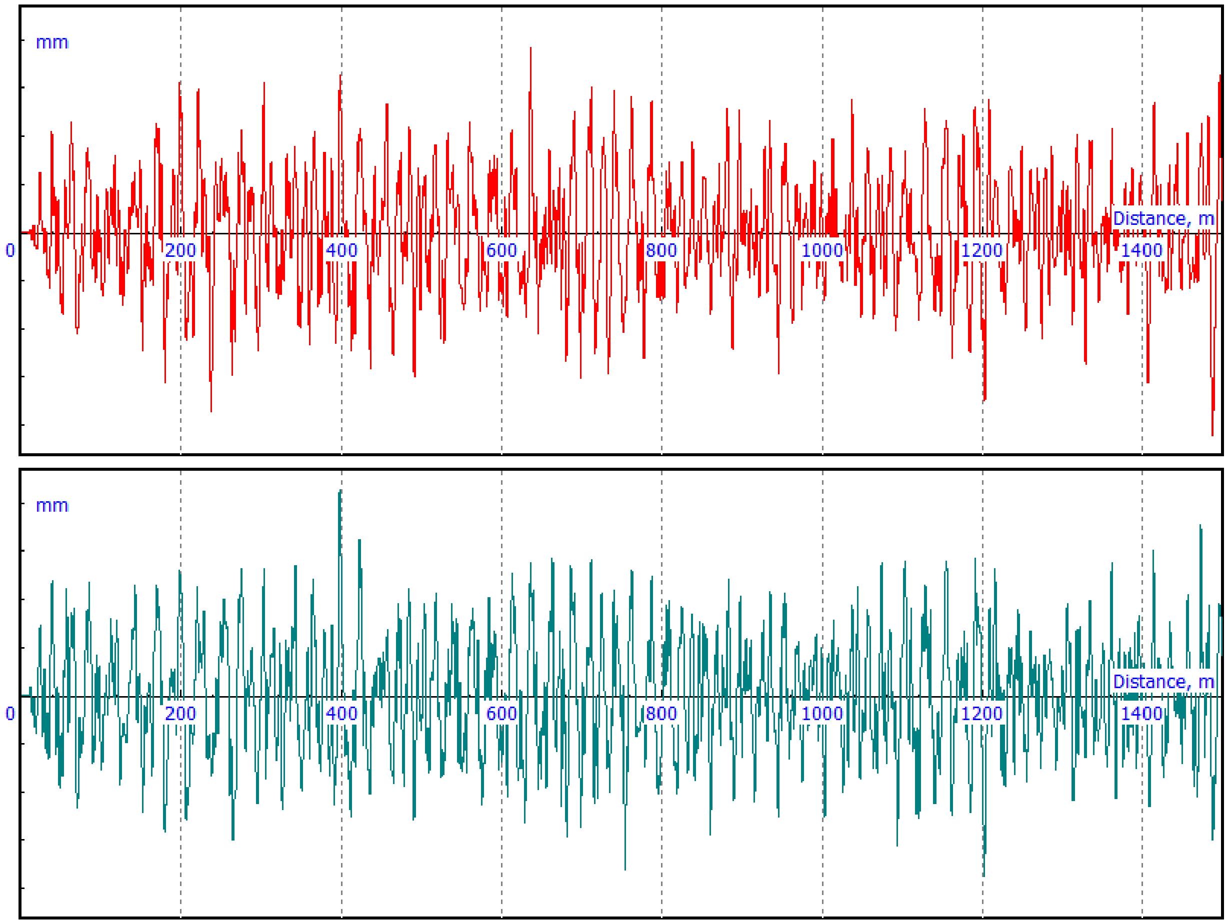



Figure 12.9. Example of road roughness for the left and right tracks

### 12.3.2.2.3.2. Wong

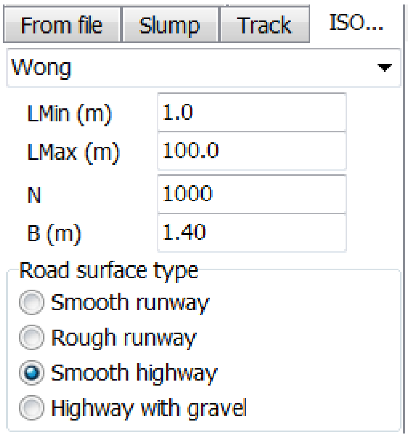



Figure 12.10. Road roughness parameters by parameters according to J.Y.Wong [2]

Table 12.2

**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 12.2, Figure 12.10.

### 12.3.2.2.3.3. Dixon

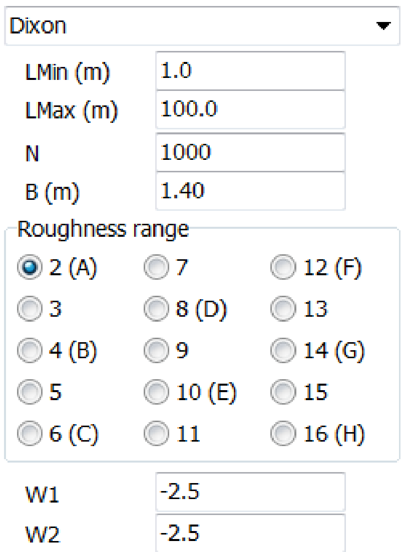



Figure 12.11. Road roughness classification by J. Dixon [4]

| 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 $w_1 = w_2 = -2.5$.

### 12.3.2.2.3.4. Experiment

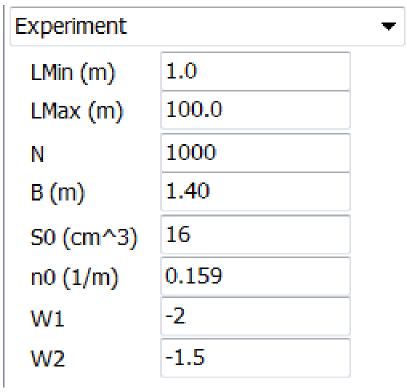



Figure 12.12. PSD parameters

In this case, the user can set arbitrary values of the PSD function, Figure 12.12

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

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

#### 12.3.2.2.3.5. Track

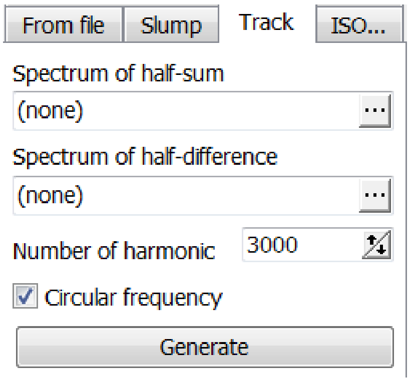



Figure 12.13. Pointwise description of PSD functions

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. 12.3.2.1. *"Library of irregularity files"*, p. 12-10). Please remember that the spectra from the library depend on the angular frequency, and the corresponding key must be checked (Figure 12.13).

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.

### 12.3.2.2.4. Other tools for description of road roughness

#### 12.3.2.2.4.1. 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.

#### 12.3.2.2.4.2. Slump

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

#### 12.3.2.2.4.3. 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.

#### 12.3.2.2.4.4. 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.

### 12.3.2.3. 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 12.14). Paths to selected files are stored in the configuration file **.car*.

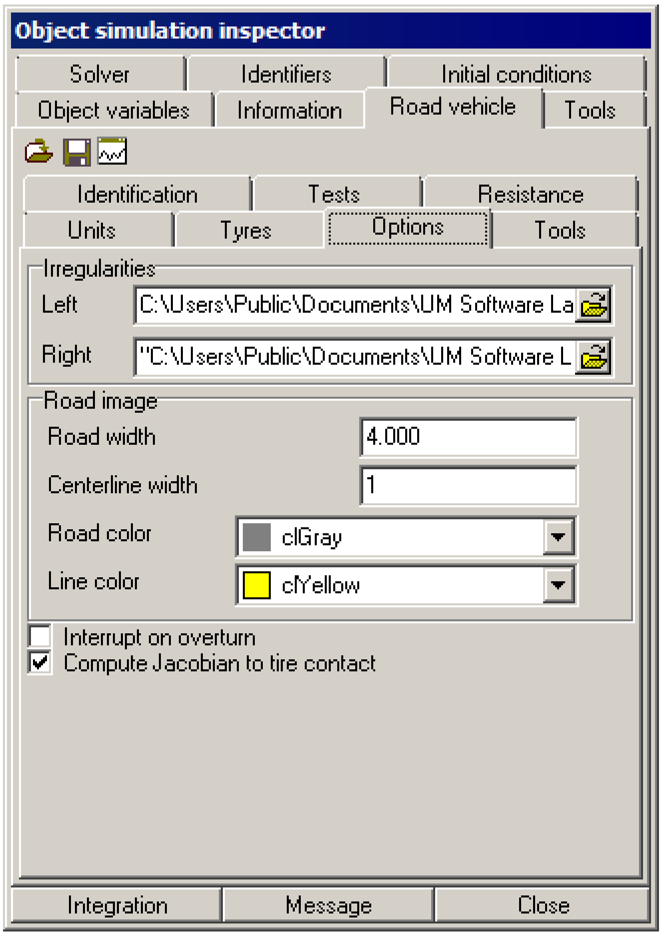



Figure 12.14. Setting current irregularities

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 12.15. Thus, the vehicle at start is always on an absolutely even horizontal plane.

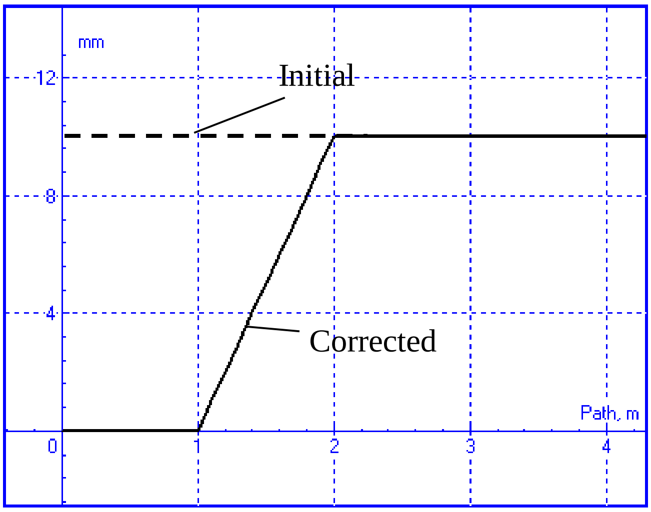



Figure 12.15. Correction of irregularities

# 12.4. Driver

## 12.4.1. 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 12.16). 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.

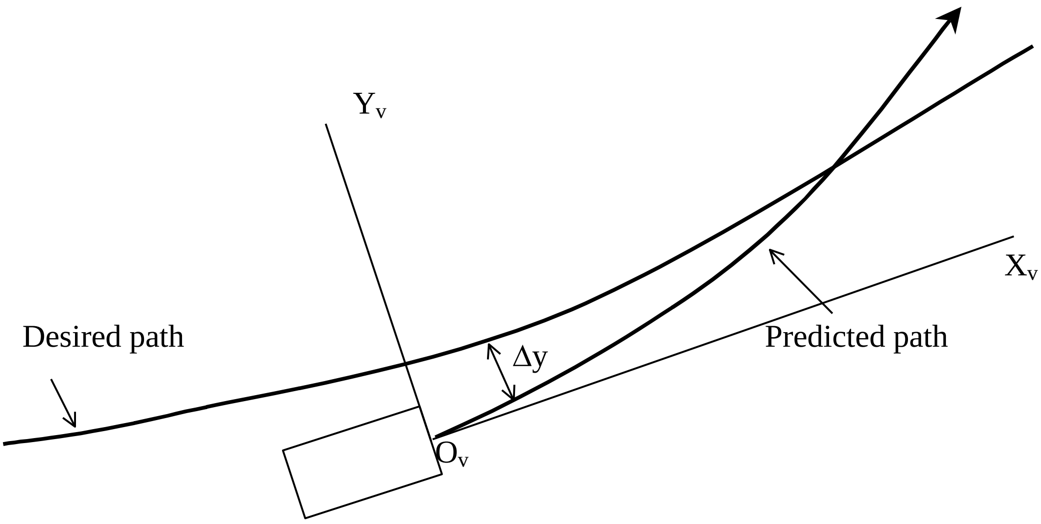



Figure 12.16. Desired and predicted paths

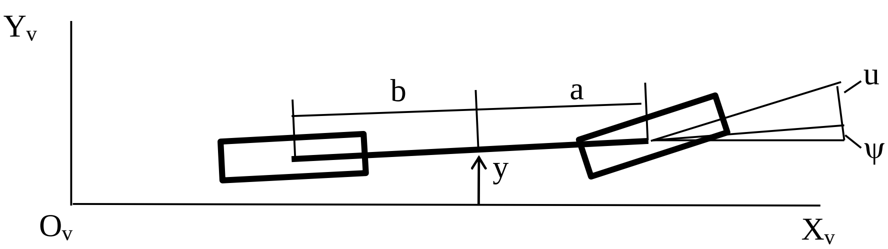



Figure 12.17. Two-wheel model of vehicle

If the steer angle u is given, the simplified model of the vehicle shown in Figure 12.17 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:

$$\dot{y} = v_x \psi + v_y, \tag{12.1}$$

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

Here $v_x, v_y$ are the projections of the vehicle velocity on the longitudinal and lateral axis of the vehicle ($v_x = const$), $\omega_z$ 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

$$y_v = y + a\psi. \tag{12.2}$$

Equations (12.1), (12.2) are linear with constant coefficients, and can be written in the matrix form as

$$\dot{x} = Ax + Bu, \tag{12.3}$$
$$y_v = C^T x,$$
$$x = \begin{pmatrix} y \\ \psi \\ v_y \\ \omega_z \end{pmatrix}, \quad A = \begin{pmatrix} 0 & v_x & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{C_f + C_r}{M v_x} & \frac{C_r b - C_f a}{M v_x} - v_x \\ 0 & 0 & \frac{C_r b - C_f a}{I_z v_x} & -\frac{C_f a^2 + C_r b^2}{I_z v_x} \end{pmatrix}$$
$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{C_f}{M} \\ \frac{C_f a}{M} \end{pmatrix}, C = \begin{pmatrix} 1 \\ a \\ 0 \\ 0 \end{pmatrix}$$

General solution of Eq. (12.3) with the assumption $u = const$ is

$$x(t) = e^{At}x_0 + \int_0^t e^{At} B d\tau u, \tag{12.4}$$
$$y_v(t) = F(t)x_0 + g(t)u.$$

Here $x_0$ is the matrix-column of initial conditions. The $1 \times 4$ 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. (12.3) 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 12.16). 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.

$$u = \frac{\sum_{i=1}^{N}(y_d(t_i) - F(t_i)x_0)g(t_i)}{\sum_{i=1}^{N} g^2(t_i)}, t_i = \frac{iT_p}{N} \tag{12.5}$$

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 $\delta$ 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 $\delta$ 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 12.3 contains a list of parameters characterizing the MacAdam's model of the driver.

Table 12.3

**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 12.18, Figure 12.19 with the following parameter values: v=5m/s, $T_p = 2$ s, $t_d = 0.15s$, $T_n = 0.1s$, $N_u = 3$.

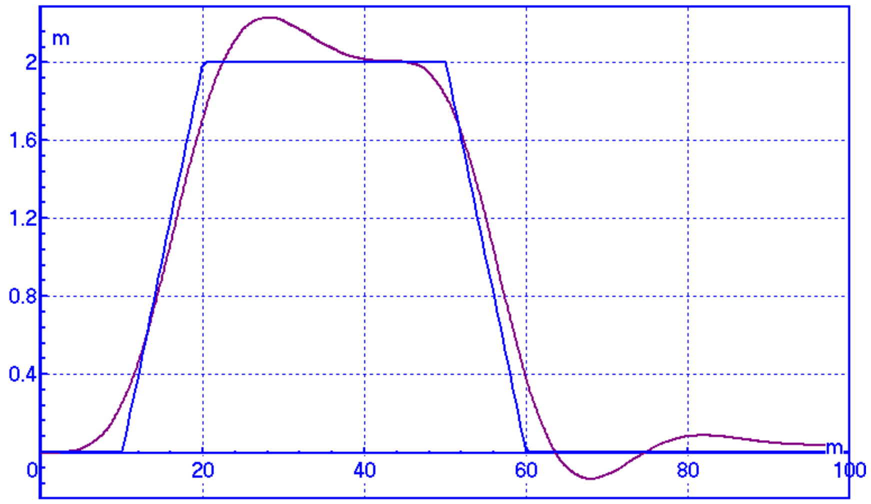



Figure 12.18. Desired and simulated path

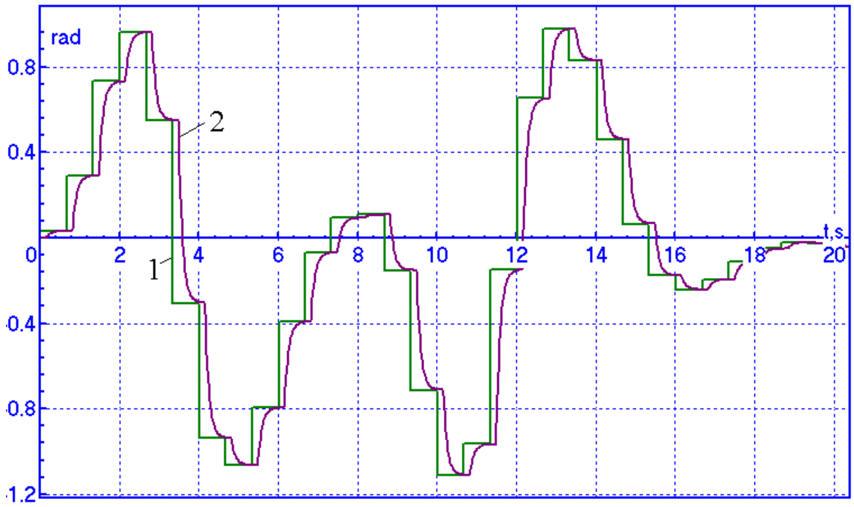



Figure 12.19. Steer wheel angle: control before (1) and after (2) the neuromuscular filter

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

## 12.4.2. Second order 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 12.20. 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 12.16. 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

$$y_p = y(t) + T_p\dot{y}(t) + \frac{T_p^2\ddot{y}(t)}{2} = T_p\dot{y}(t) + \frac{T_p^2\ddot{y}(t)}{2} = y(t + T_p) + O\left(T_p^3\dddot{y}(t)\right).$$

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.

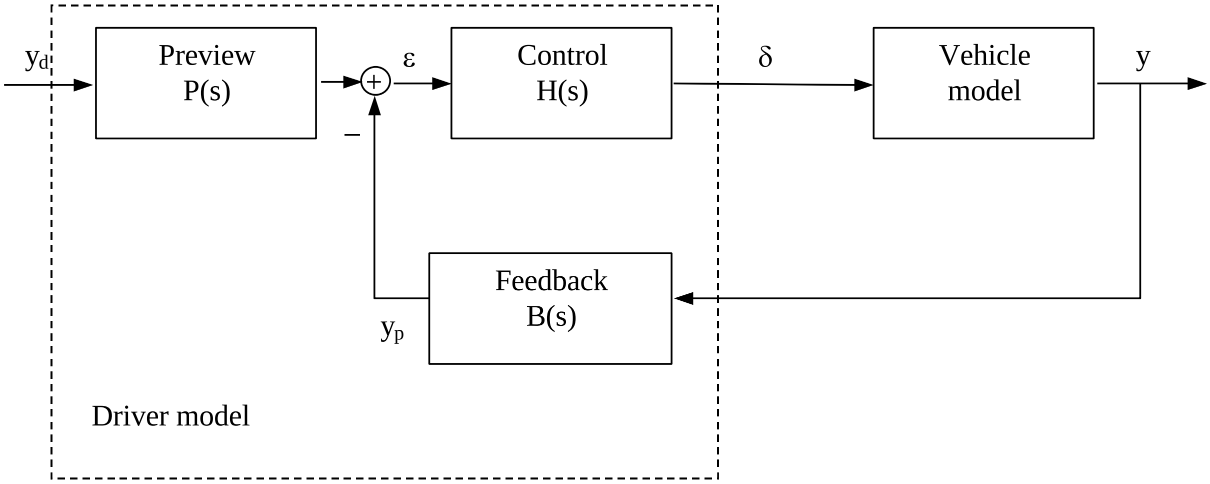



Figure 12.20. Block diagram of the control

The transfer functions are:

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

Control: $H(s) = \frac{K}{L_p} e^{-t_d s}, L_p = vT_p$

Feedback: $B(s) = 1 + T_p s + T_p^2 s^2/2$

Here $K$ is the gain.

Transformation in the time domain leads to the following equations:

$$\varepsilon(t) = y_d(t + T_p) - y_p,$$
$$y_p = T_p\dot{y}(t) + \frac{T_p^2\ddot{y}(t)}{2},$$
$$\delta(t) = \frac{K}{L_p}\varepsilon(t - t_d).$$

or

$$\delta(t) = \frac{K}{L_p}\left(y_d\left(t + T_p - t_d\right) - T_p\dot{y}(t - t_d) - T_p^2\ddot{y}(t - t_d)/2\right),$$
$$\alpha_s(t) = i_s\delta(t).$$

Table 12.4 contains a list of parameters characterizing the second order preview driver model.

Table 12.4

**Second order preview model parameters**

| Parameter | Comments | Recommended interval of values | Default value |
|---|---|---|---|
| $T_p$ | Preview time | 1-2c | 1s |
| $t_d$ | Reaction time delay | >0.15c | 0.15s |
| $K$ | Gain | 0.7-0.4 | 0.5 |

Simulation result for a maneuver of the car VAZ 2109 are shown in Figure 12.21, Figure 12.22 with the following parameter values: v=5m/s, $T_p = 1s, t_d = 0.15s, K = 0.5$.

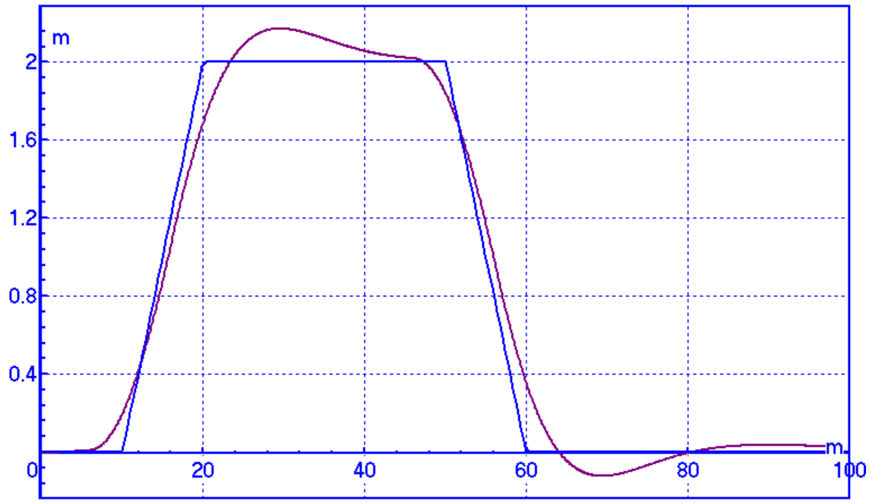



Figure 12.21. Desired and simulated path

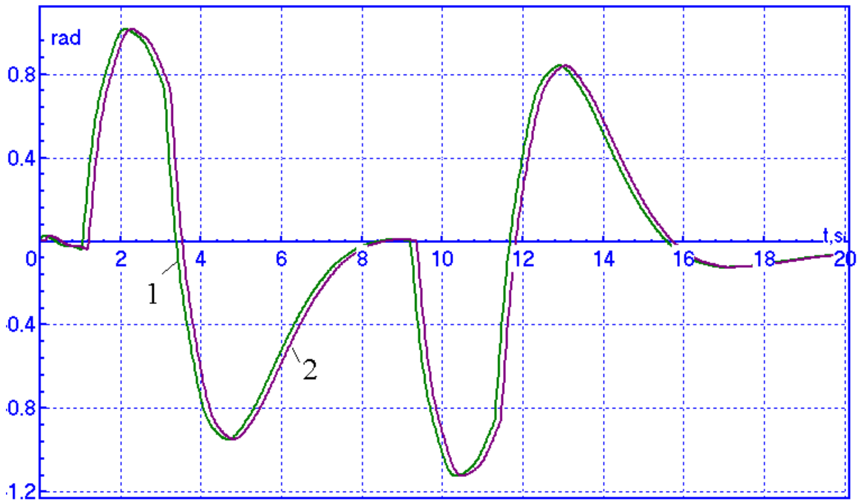



Figure 12.22. Steer wheel angle: control (1) and the driver output (2)

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

### 12.4.3. Combination of PID controller and second order preview model

Both the MacAdam and the second order 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.

$$\delta(t) = K_2 y_d(t) + K_d \dot{y}_d(t) + K_I \int_0^t \dot{y}_d(\tau) d\tau +$$
$$K\left(y_d\left(t + T_p - t_d\right) - T_p \dot{y}(t - t_d) - T_p^2 \ddot{y}(t - t_d)/2\right),$$

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.

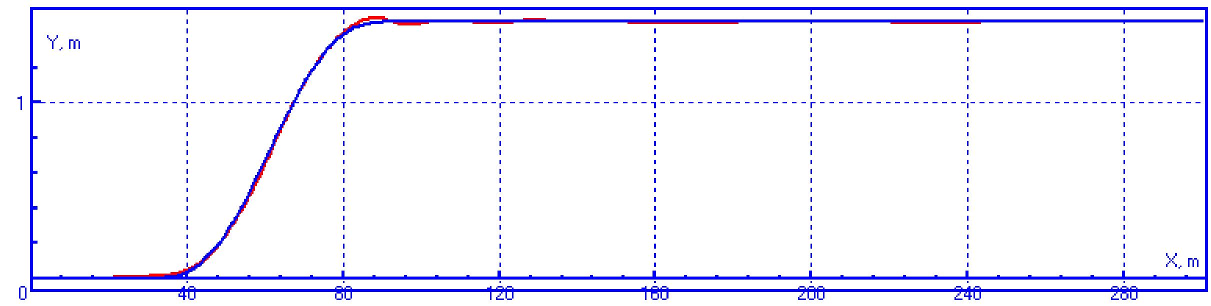



Figure 12.23. Lane change maneuver. Desired path and simulation result.

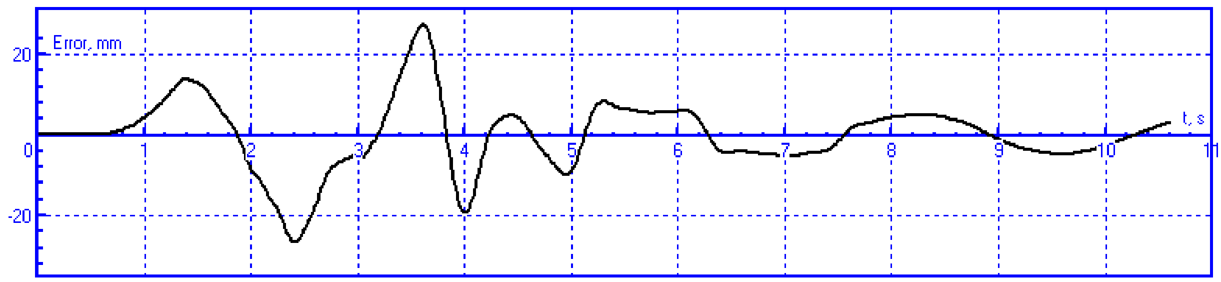



Figure 12.24. Lane change maneuver. Path following error.

Figure 12.23, Figure 12.24 show simulation results for a lane change maneuver obtained for a track/trailer model (Sect. 12.9.1.1. *"Units"*, p. 12-75). The following parameter values were used:

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

**Note.** The controller uses the derivative of the error $\dot{y}_d$, which requires a differentiable function of the desired path. In this case a spline interpolation of the path curve is necessary (Sect. 12.3.1. *"Track macro geometry"*, p. 12-6)

# 12.5. 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. 12.5.1;
- Pacejka Magic Formula, see Sect. 12.5.3;
- Tabular model, see Sect. 12.5.4;
- TMEasy tire model, see Sect. 12.5.5.

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).

## 12.5.1. Single point and multipoint normal contact models

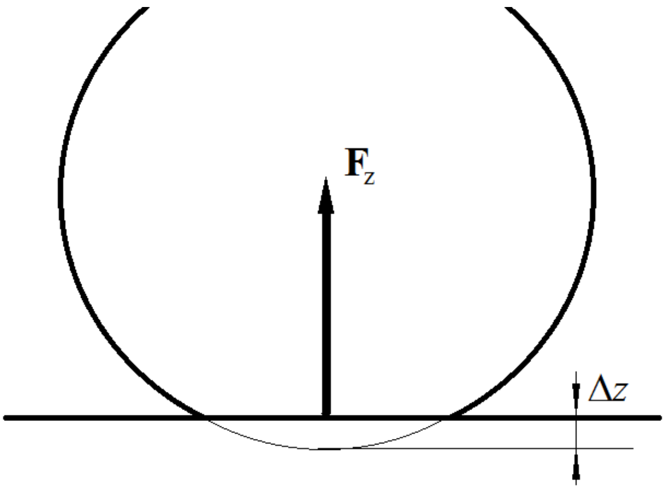



Figure 12.25. Single point contact

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 $\Delta z$, which can be computed as the maximal penetration of the rigid wheel circle with the road line like in Figure 12.25,

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

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

$$F_z = -k_z\Delta z - d_z\Delta\dot{z}. \quad (12.1)$$

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 12.26. In such cases the tire contact patch may consist of two or more separate sections.

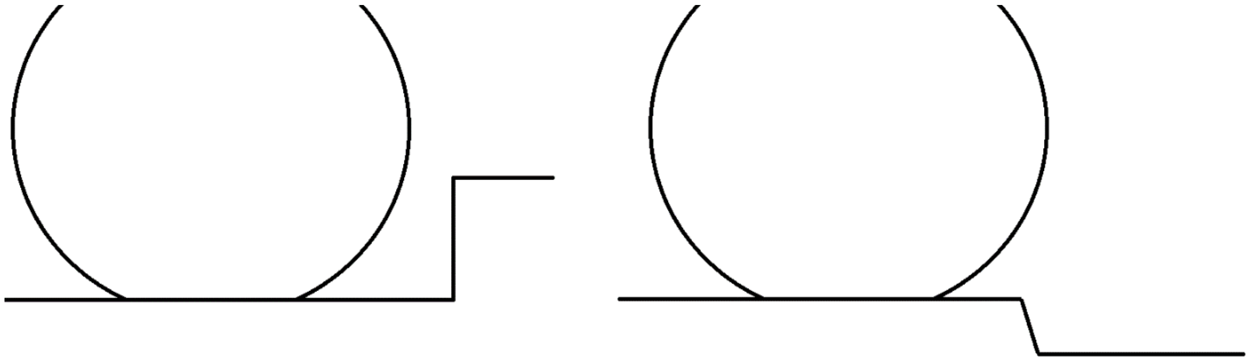


Figure 12.26. Special road deviations

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.

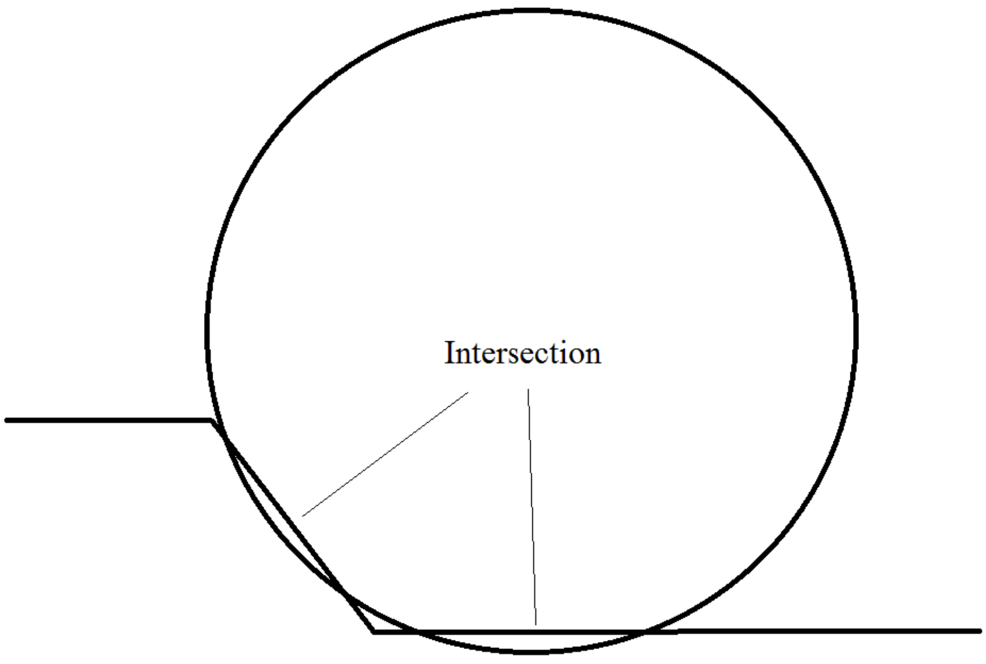



Figure 12.27. Two regions of intersection

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

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

The forces $F_{zi}$ are applied at the points of the maximal intersection depth and directed *to the center of the wheel*, Figure 12.28.

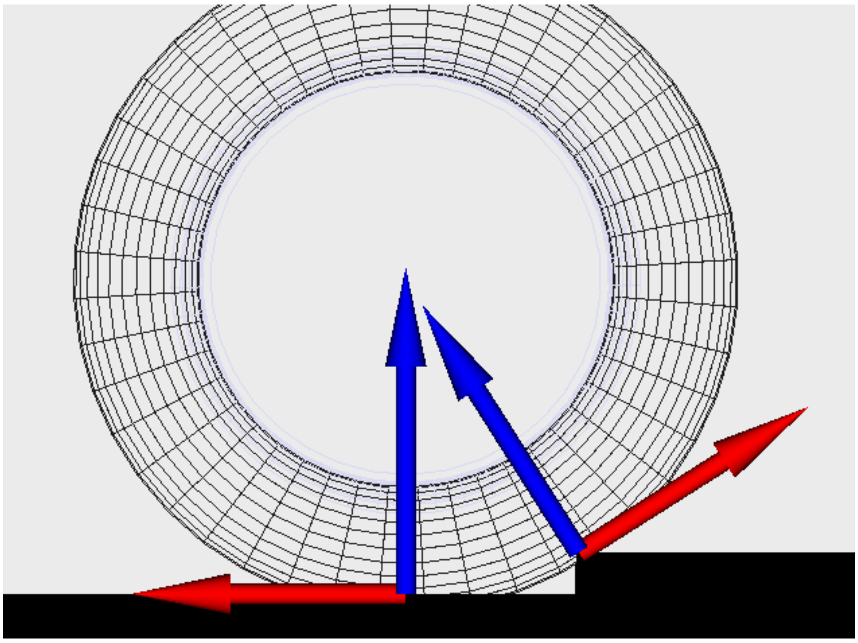


Figure 12.28. Wheel rolling up a step with the discrete point contact model

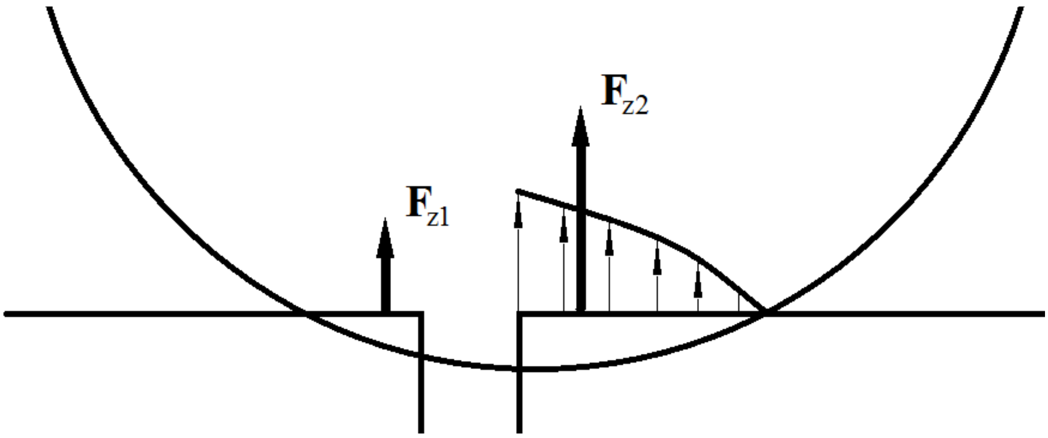



Figure 12.29. Distributed contact model

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 12.29

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

where $k_{zd}$ is the distributed contact stiffness constant, and **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,

$$F_{ezi} = k_{zd} \int_{x_{i1}}^{x_{i2}} q(x)dx = k_{zd} A_i, \tag{12.2}$$

Taking into account Eq. (12.2), we can compute the $k_{zd}$ constant equivalent to the tire stiffness $k_z$ from Eq. (12.1),

$$k_{zd} = k_z \Delta z_0 / A_0, \tag{12.3}$$

where $\Delta z_0$ 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 $\Delta z$. According to Eq. (12.2), for an ideal straight road section

$$F_{ez} = k_{zd} A(\Delta z) = \alpha r^2 - (r - \Delta z) r \sin\alpha \approx k_{zd}\sqrt{2r}\Delta z^{3/2} = \left(k_{zd}\sqrt{2r}\Delta z^{1/2}\right)\Delta z,$$
$$\sin\alpha = \sqrt{\frac{2\Delta z}{r} - \frac{\Delta z^2}{r^2}} \approx \alpha, \quad \alpha \approx \sqrt{\frac{2\Delta z}{r}}. \tag{12.4}$$

Here $\alpha$ 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 12.30. Backwards, the distributed flexible contact is appropriate for small potholes, and gives bad results for high steps.

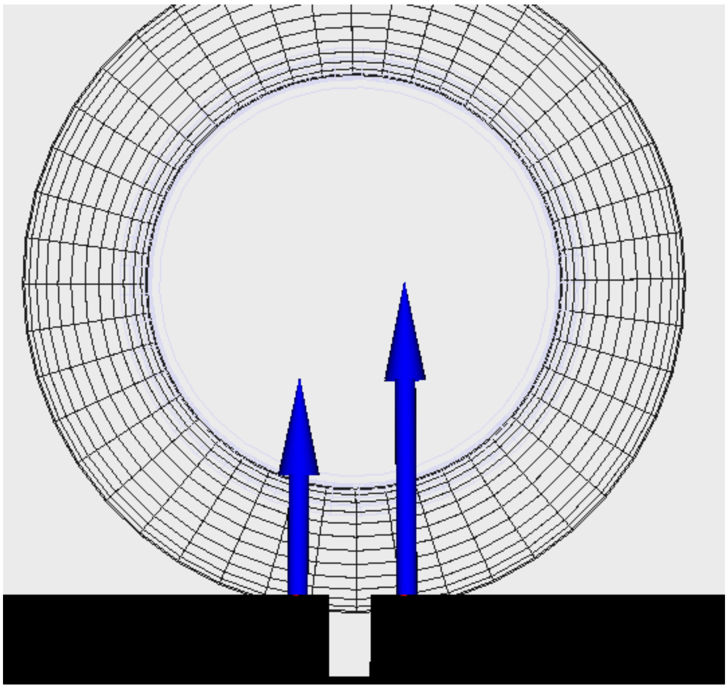


Figure 12.30. Wheel running over a small pothole with the flexible distributed contact model

| | |
|---|---|
| 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. 12.9.1.6. *Test section profile of road*. 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).

## 12.5.2. 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 |
| sx | - | Longitudinal slip | Computation by simulation |
| sy=tan α | - | Sideslip | Computation by simulation |
| Δr | - | Vertical tire deflection | Computation by simulation |
| VΔr | - | Rate of vertical tire deflection | Computation by simulation |
| r | R | Radius of unload wheel | Tire description file (*.tr) |
| kz | Kz | Tire vertical stiffness constant | Tire description file (*.tr) |
| kx | Kx | Tire longitudinal stiffness constant | Tire description file (*.tr) |
| kн | Ky | Tire lateral stiffness constant | Tire description file (*.tr) |
| βz | BetaZ | Damping ratio of critical | Tire description file (*.tr) |
| dz | - | 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 creep stiffness | Tire description file (*.tr) |
| cy | Cy | Cornering stiffness | Tire description file (*.tr) |
| rt | Rtorus | Toroidal radius of tire | Tire description file (*.tr) |

* Designation in the Wizard of tire models, Sect. 12.5.8. *"Tire model wizard"*, p. 12-53.

### 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.** $|s_y| \geq s'$

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

### Aligning moment ($M_z$)

**Case 1.** $|s_y| < s'$

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

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

$$M_z = 0$$

### 12.5.3. Pacejka Magic Formula

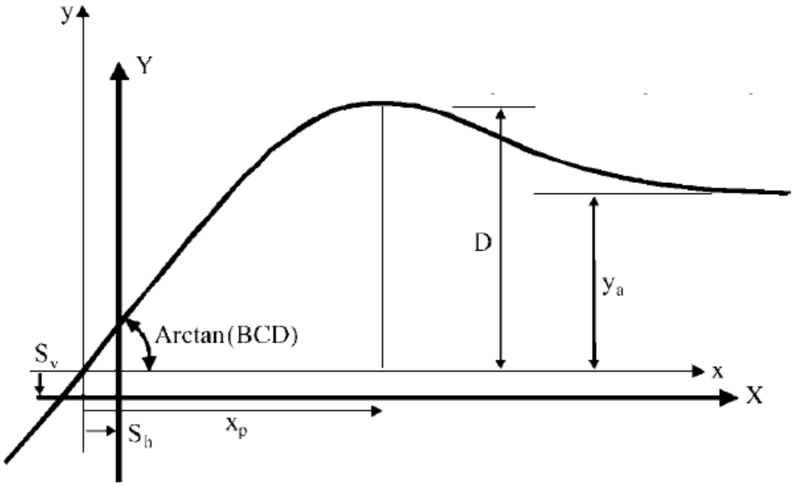



Figure 12.31. Magic formula

The Magic Formula (MF) is (Figure 12.31):

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

Here $Y(x)$ can be longitudinal ($F_x$), side ($F_y$) force or aligning moment ($M_z$), 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 $\gamma$.

**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 $F_y$.**

$$C = a_0,$$
$$D = F_z(a_1 F_z + a_2),$$
$$E = a_6 F_z + a_7,$$
$$B = \frac{1}{CD} a_3 sin\left(a_{15} arctan(F_z/a_4)\right)(1 - a_5|\gamma|),$$
$$S_h = a_8 \gamma + a_9 F_z + a_{10},$$
$$S_v = (a_{11} F_z + a_{12})\gamma F_z + a_{13} F_z + a_{14}.$$

**3. Aligning moment $M_z$.**

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

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 moment in dependence on the corresponding slip by $\gamma = 0$ for different values of the vertical load are shown in Figure 12.33, Figure 12.34. The MF with the above values of the parameters was used for computation of the forces.

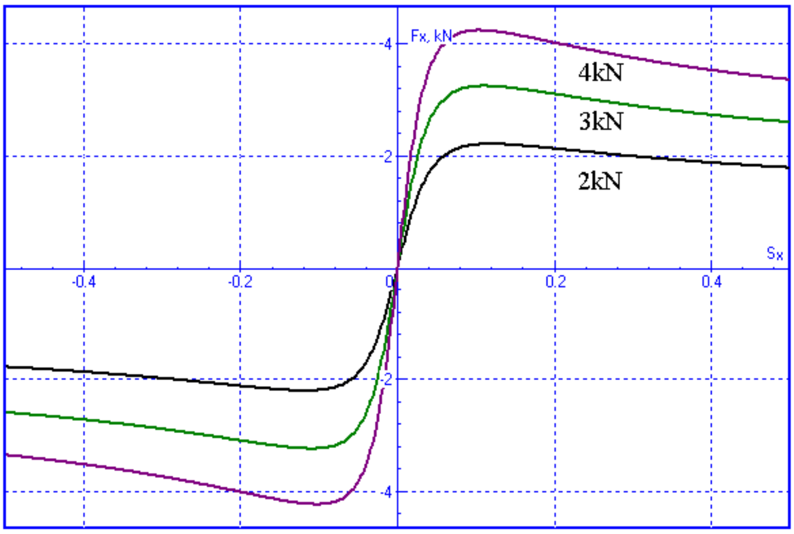



Figure 12.32. Longitudinal force

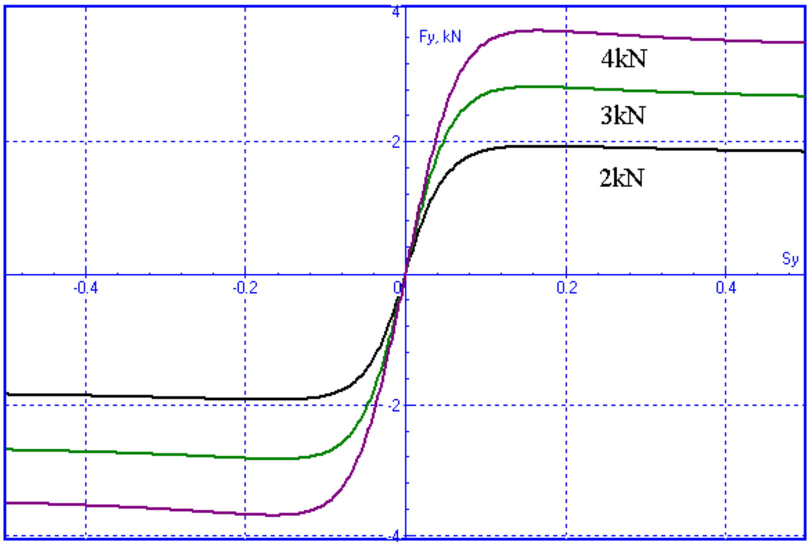



Figure 12.33. Side force

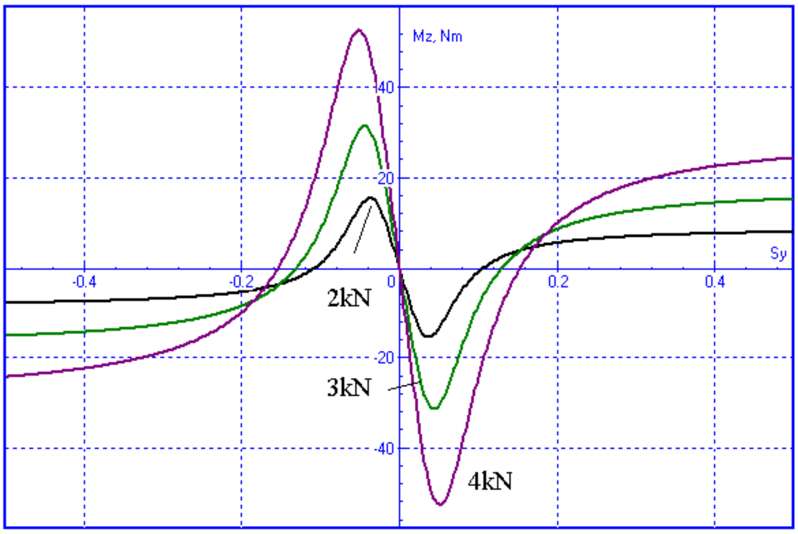



Figure 12.34. Aligning moment

## 12.5.4. Tabular tire model

Tabular model of a tire requires experimental data on the longitudinal, side forces and aligning moment, Figure 12.35, [7].

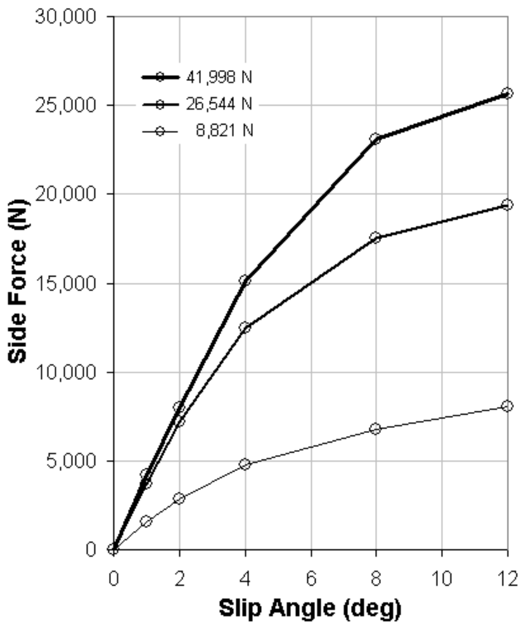



Figure 12.35. Tabular Side Force

Table 12.5

**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 $Y\left(x_j, F_{zj}\right), i = 1..N_Y, j = 1..N_{Fz}$ are the tabular data. A smoothed model of the force is obtained with two steps. First, a beta-spline approximation $\hat{Y}\left(x, F_{zj}\right)$ of the discrete function $\mathrm{Y}\left(x_j, F_{zj}\right)$ 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 12.36.

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 12.37, Figure 12.38.

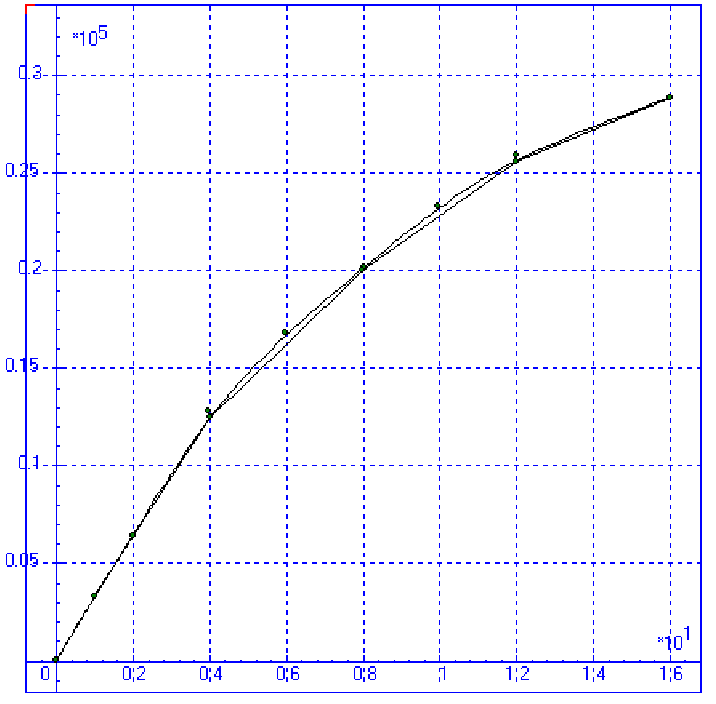



Figure 12.36. Polygon and smoothed curve

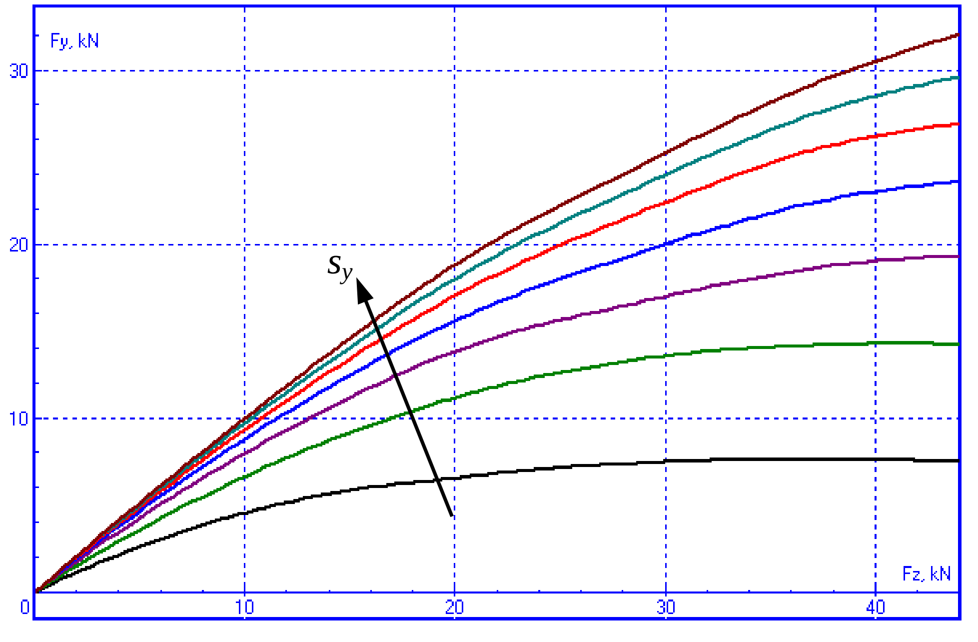



Figure 12.37. Smoothed model of the side force versus vertical load

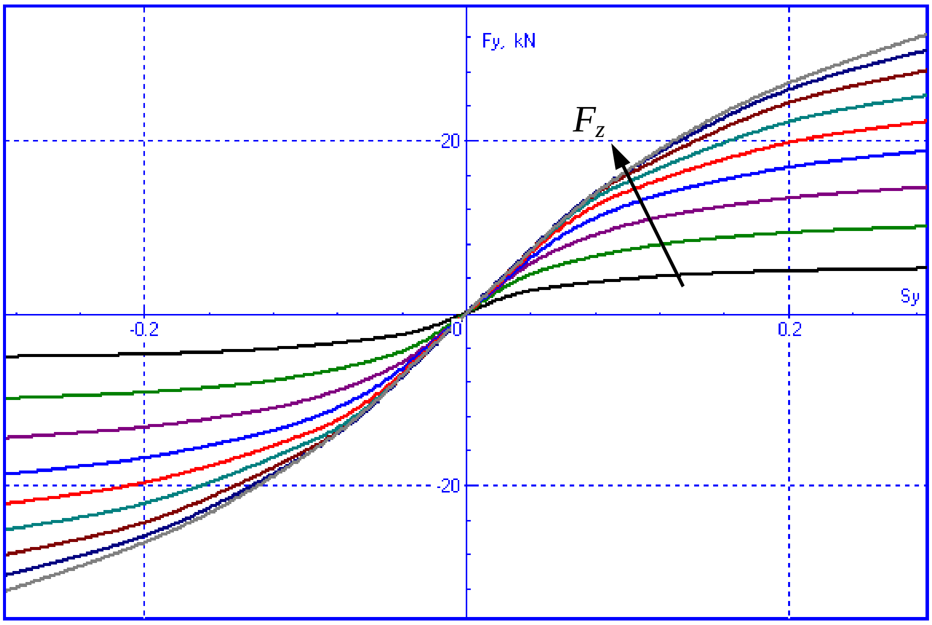



Figure 12.38. Smoothed model of the side force versus side slip

### 12.5.5. 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 12.39, 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 $s_x^S$

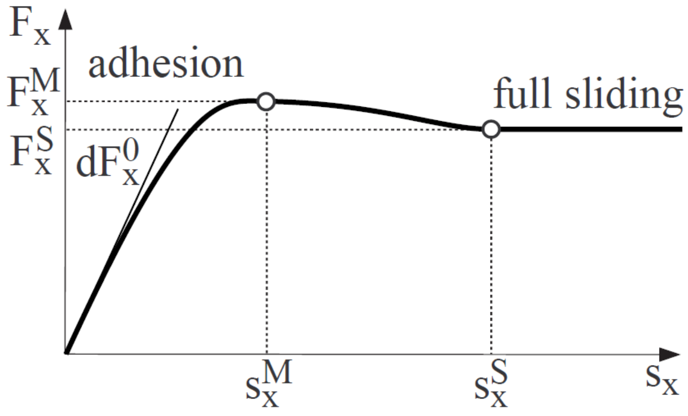



Figure 12.39. Typical longitudinal force characteristics

### 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 $s_y^S$

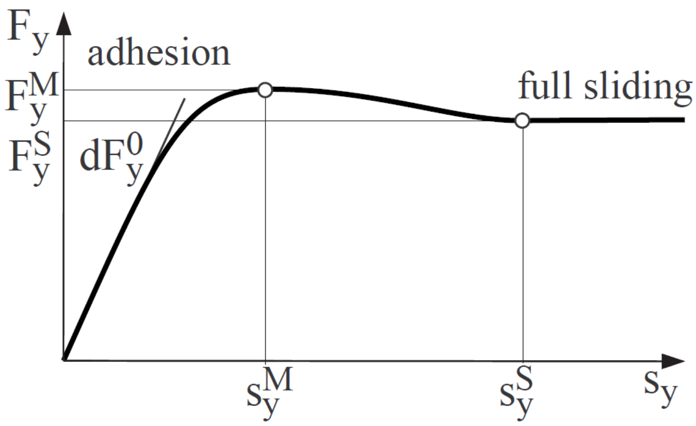



Figure 12.40. Typical lateral force characteristics

**Combined slip**

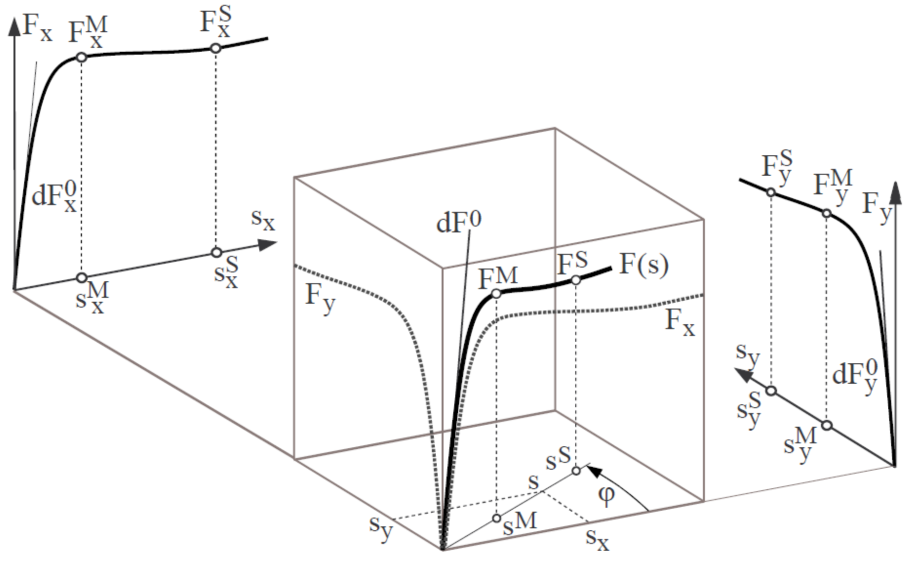



Figure 12.41. Generalized tire characteristics

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

$$s = \sqrt{\left(\frac{s_x}{\hat{s}_x}\right)^2 + \left(\frac{s_y}{\hat{s}_y}\right)^2} \tag{12.5}$$

Normation factors:

$$\hat{s}_x = \frac{s_x^M}{s_x^M + s_y^M} + \frac{F_x^M/dF_x^0}{F_x^M/dF_x^0 + F_y^M/dF_y^0}$$
$$\hat{s}_y = \frac{s_y^M}{s_x^M + s_y^M} + \frac{F_y^M/dF_y^0}{F_x^M/dF_x^0 + F_y^M/dF_y^0} \tag{12.6}$$

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

$$dF^0 = \sqrt{(dF_x^0 \cdot \hat{s}_x \cdot \cos\varphi)^2 + \left(dF_y^0 \cdot \hat{s}_y \cdot \sin\varphi\right)^2}$$
$$s^M = \sqrt{\left(\frac{s_x^M}{\hat{s}_x}\cos\varphi\right)^2 + \left(\frac{s_x^M}{\hat{s}_x}\sin\varphi\right)^2}$$
$$F^M = \sqrt{(F_x^M\cos\varphi)^2 + \left(F_y^M\sin\varphi\right)^2} \tag{12.7}$$
$$s^S = \sqrt{\left(\frac{s_x^S}{\hat{s}_x}\cos\varphi\right)^2 + \left(\frac{s_x^S}{\hat{s}_x}\sin\varphi\right)^2}$$
$$F^S = \sqrt{(F_x^S\cos\varphi)^2 + \left(F_y^S\sin\varphi\right)^2}$$

where

$$\cos\varphi = \frac{s_x/\hat{s}_x}{s} \qquad \sin\varphi = \frac{s_y/\hat{s}_y}{s} \tag{12.8}$$

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

$$F(s) = \begin{cases} s^M dF^0 \dfrac{\sigma}{1+\sigma\left(\sigma + dF^0 \dfrac{s^M}{F^M} - 2\right)}, & \sigma = \dfrac{s}{s^M} & 0 \le s \le s^M \\ F^M - (F^M - F^S)\sigma^2(3-2\sigma), & \sigma = \dfrac{s-s^M}{s^S - s^M} & s^M \le s \le s^S \\ F^S & & s < s^S \end{cases} \tag{12.9}$$

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

$$F_x = F\cos\varphi \qquad F_y = F\sin\varphi \tag{12.10}$$

## 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*:

$$M_z = -nF_y, \tag{12.11}$$

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

$$\frac{n}{L} = \left(\frac{n}{L}\right)_0 \begin{cases} 1 - |s_y|/s_y^0 & |s_y| \le s_y^0 \\ \dfrac{|s_y| - s_y^0}{s_y^0}\left(\dfrac{s_y^E - |s_y|}{s_y^E - s_y^0}\right)^2 & s_y^0 \le |s_y| \le s_y^E \\ 0 & |s_y| > s_y^E \end{cases} \tag{12.12}$$

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.

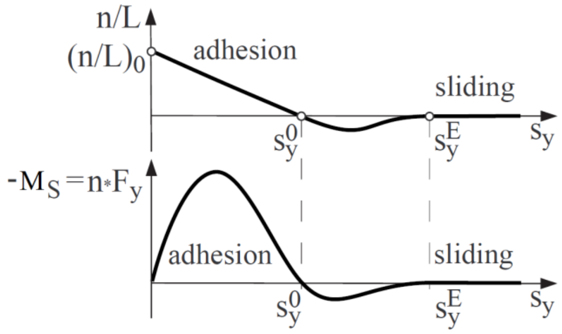



Figure 12.42. Typical plots of dynamic offset and self-aligning torque

## 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]

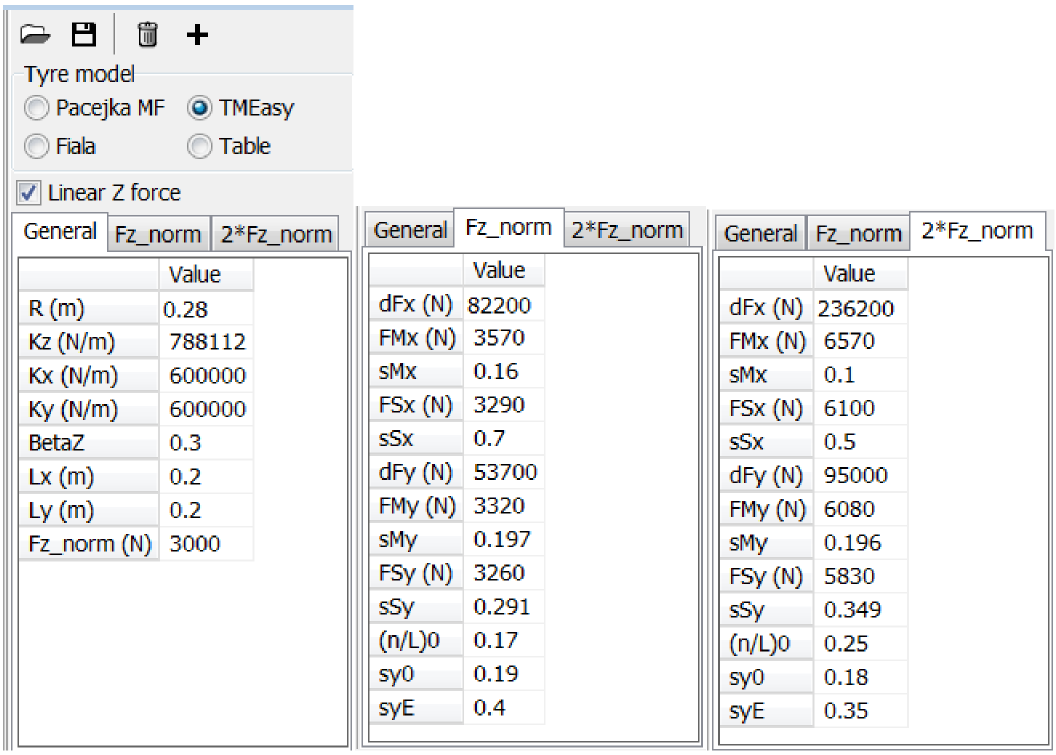



Figure 12.43. TMEasy in wizard of tire parameters

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;
```

## 12.5.6. 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 12.44. It is recommended to check the option for tests with braking processes.

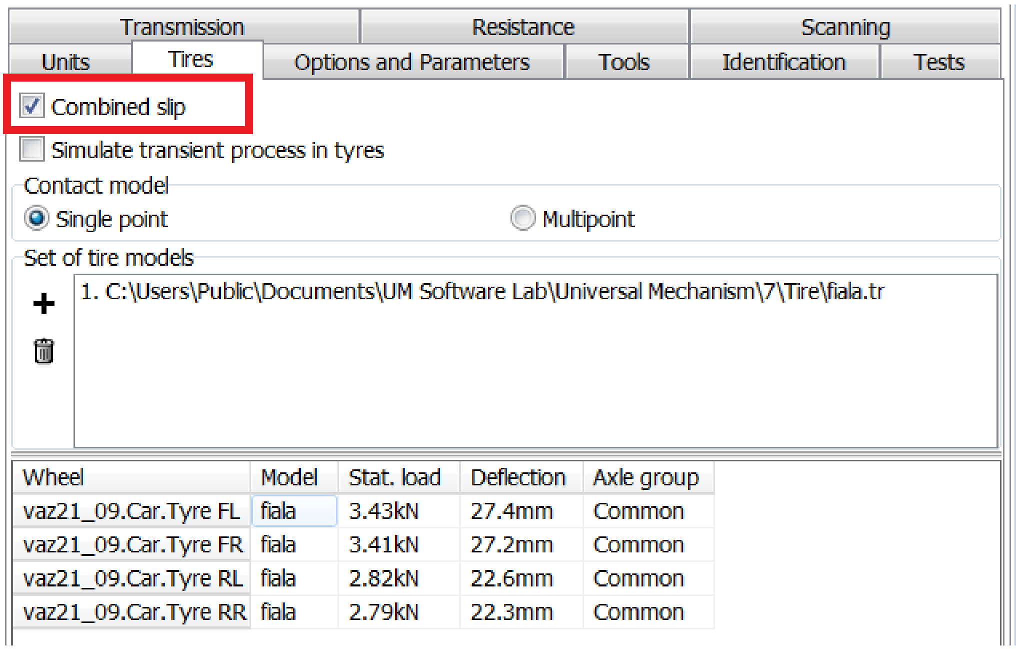



Figure 12.44. Combined slip option

## 12.5.7. 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 $v_x, v_y, \omega$ 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.

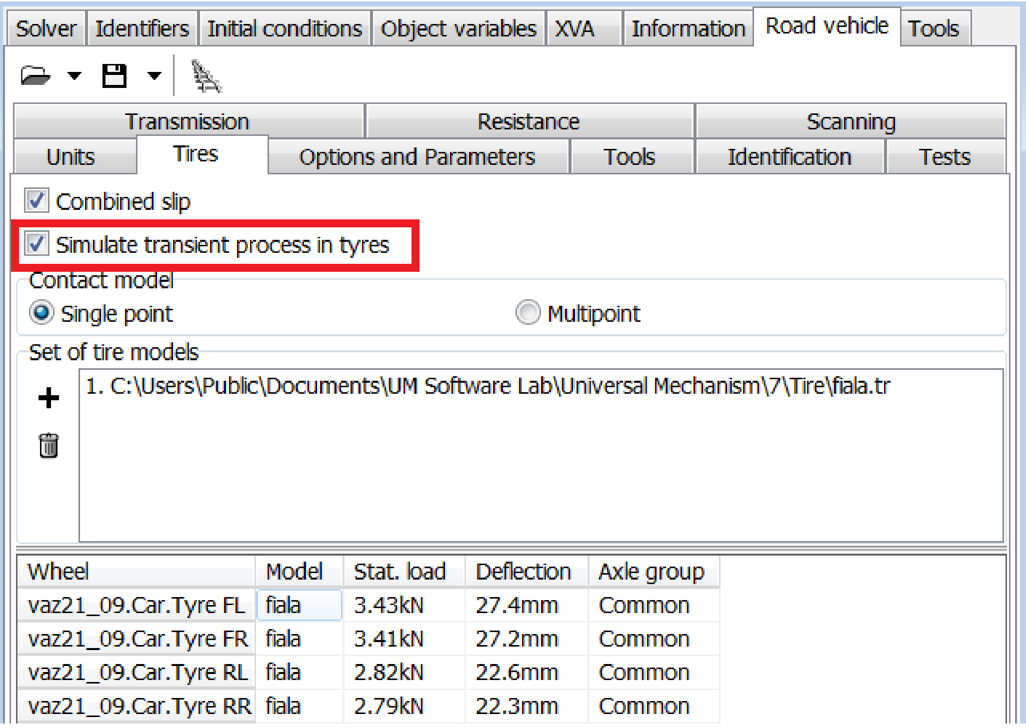



Figure 12.45. Option for transient processes in tire

Check the **Simulate transient process in tire** option in the inspector to activate the transient model, Figure 12.45.

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 12.46. The relaxation lengths are specified in meters in the tire wizard, Figure 12.47.

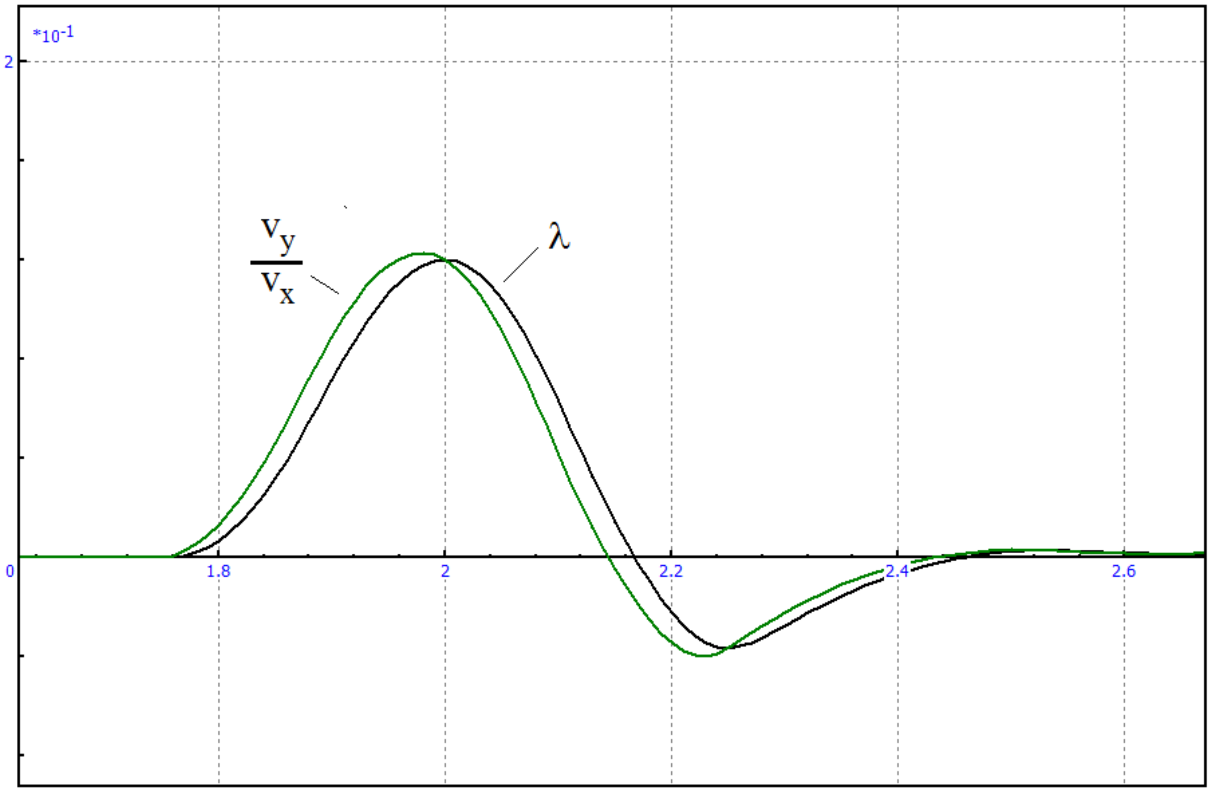



Figure 12.46. Comparison of λ and $v_y/v_x$ in a pulse steer test

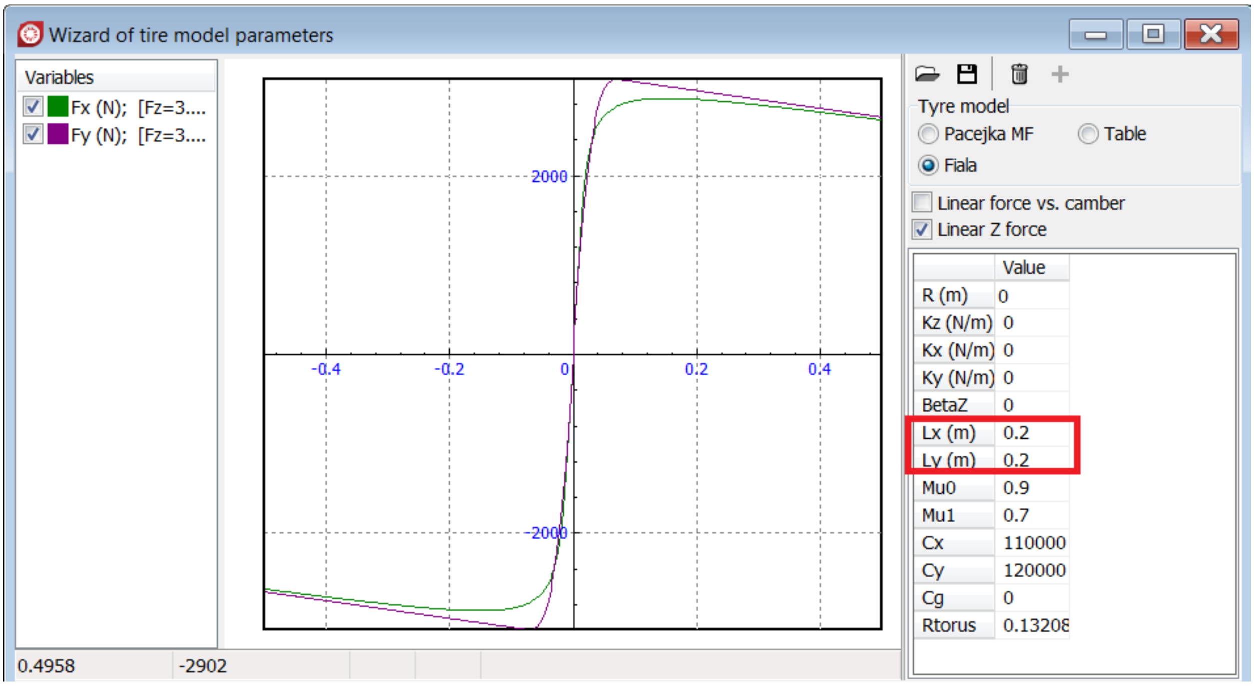



Figure 12.47. Setting tire relaxation lengths

## 12.5.8. Tire model wizard

Tire models are developed and analyzed with the help of **Wizard of tire models**, Figure 12.48. 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.

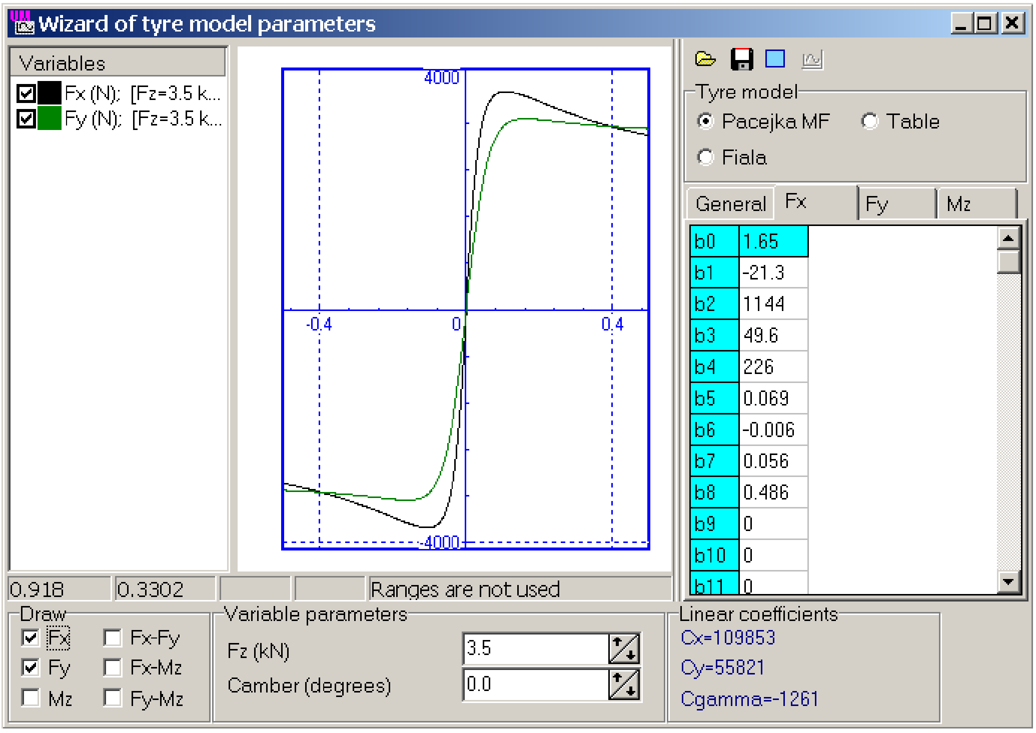



Figure 12.48. Wizard of tire models

### 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.

## 12.5.9. 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.

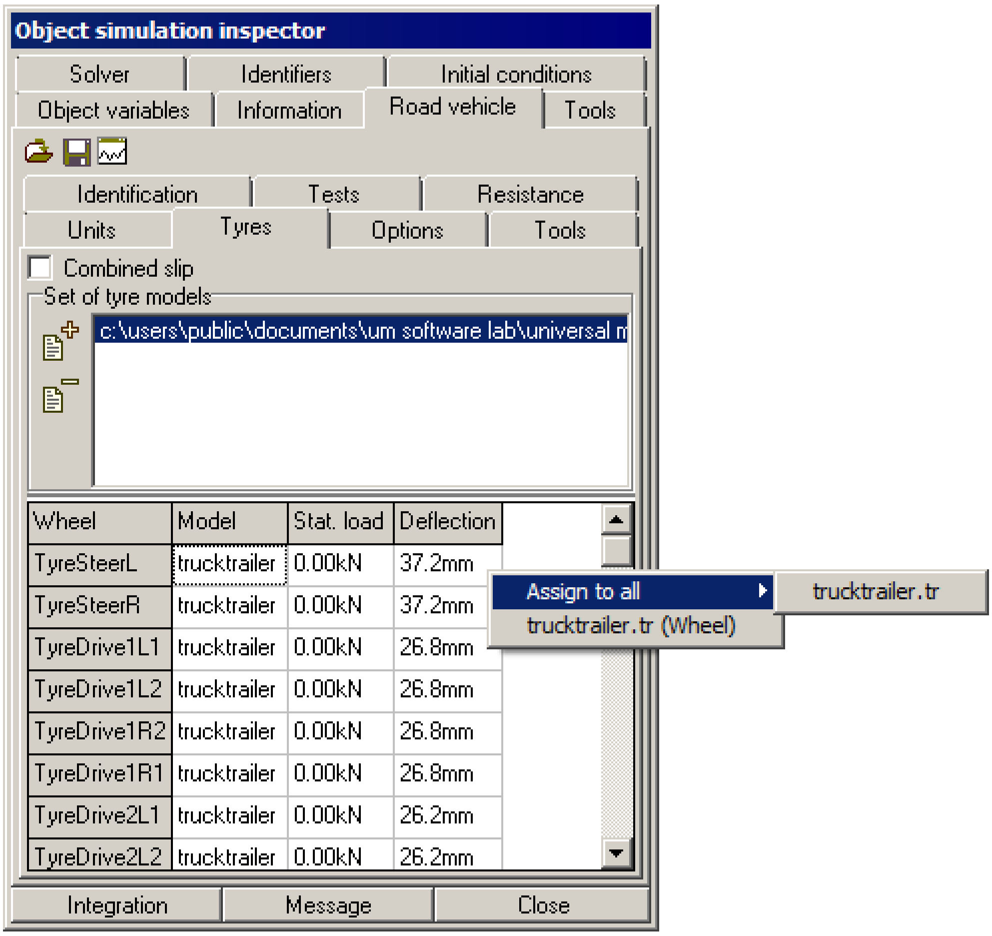



Figure 12.49. Assignment of tire models

### 12.5.10. Visualization of tire forces

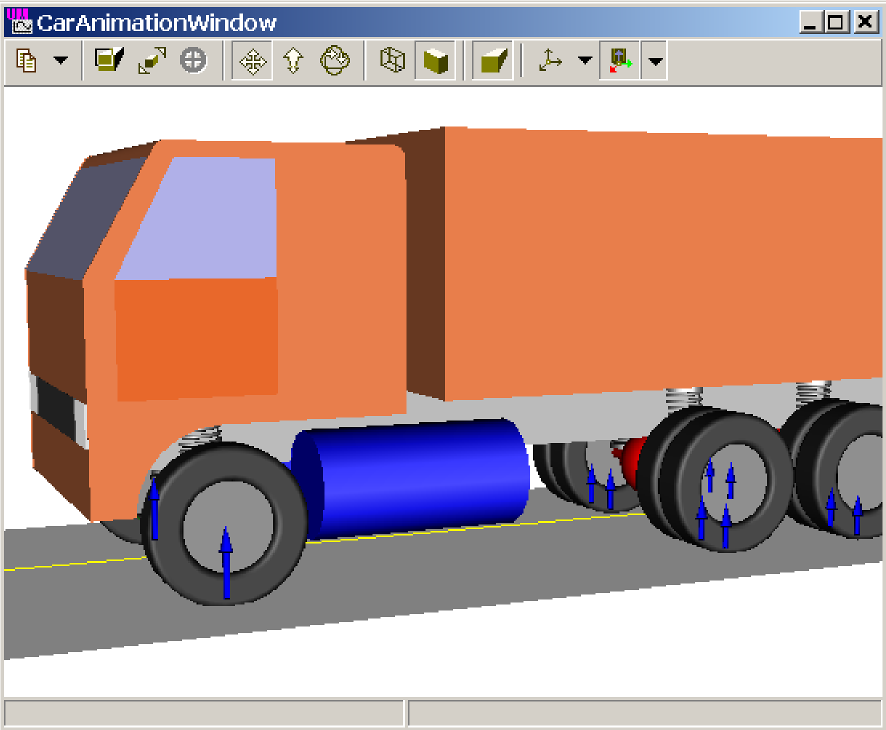
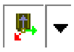



Figure 12.50. Tire forces

Vertical, side, longitudinal tire forces can be visualized during the simulation. Use the  button on the top of the animation window to hide/show the forces.

After the 12.9.2.4. *"Equilibrium test"*, p. 12-87 the scale factor for the forces is set automatically so that the static values of vertical forces for the steer axis correspond to a radius of the tires.

# 12.6. Resistance to vehicle motion

## 12.6.1. Aerodynamic forces

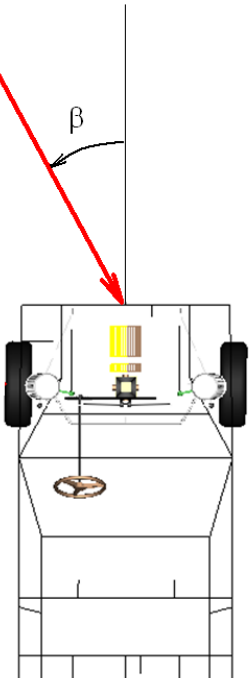



Figure 12.51. Positive angle of relative wind speed direction

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 12.51) and some other parameters. Aerodynamic coefficients and car area depend on the force or moment component.

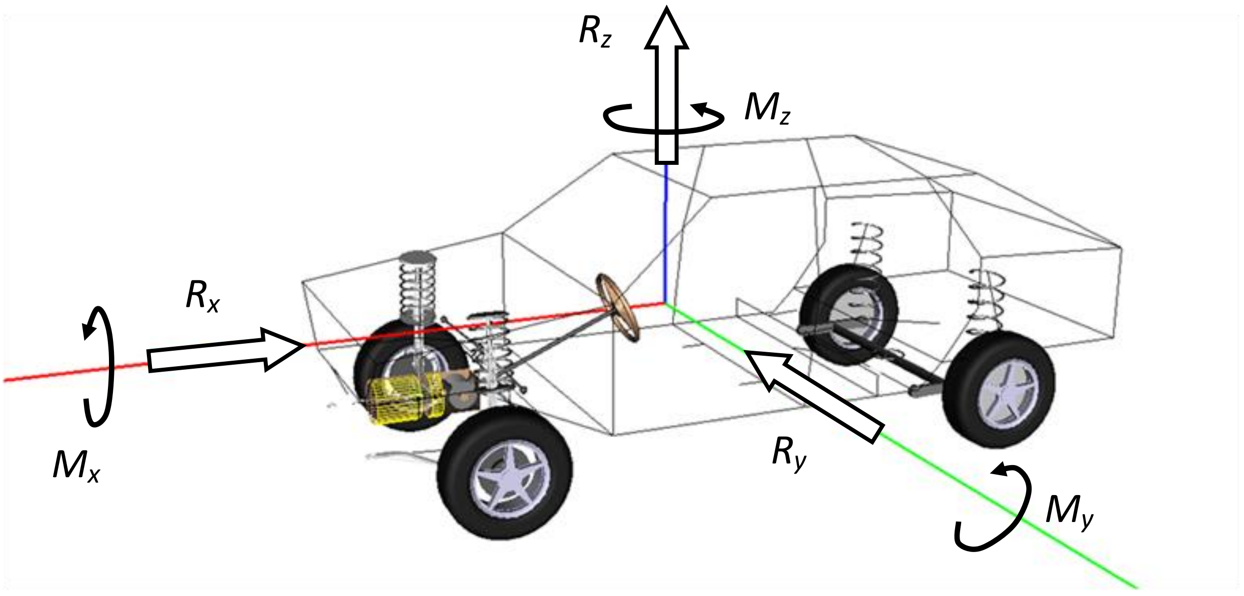



Figure 12.52. Positive directions of aerodynamic forces and moments

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 12.52.

- 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

$$R_z = C_zA_x\frac{\rho}{2}V_a^2$$

$C_z$ is the aerodynamic lift coefficient.

- Rolling moment

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

$L_y$ is the track width.

- Pitching moment

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

$L_y$ is the wheel base.

- Yawing moment

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

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 12.6.

Table 12.6

**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.

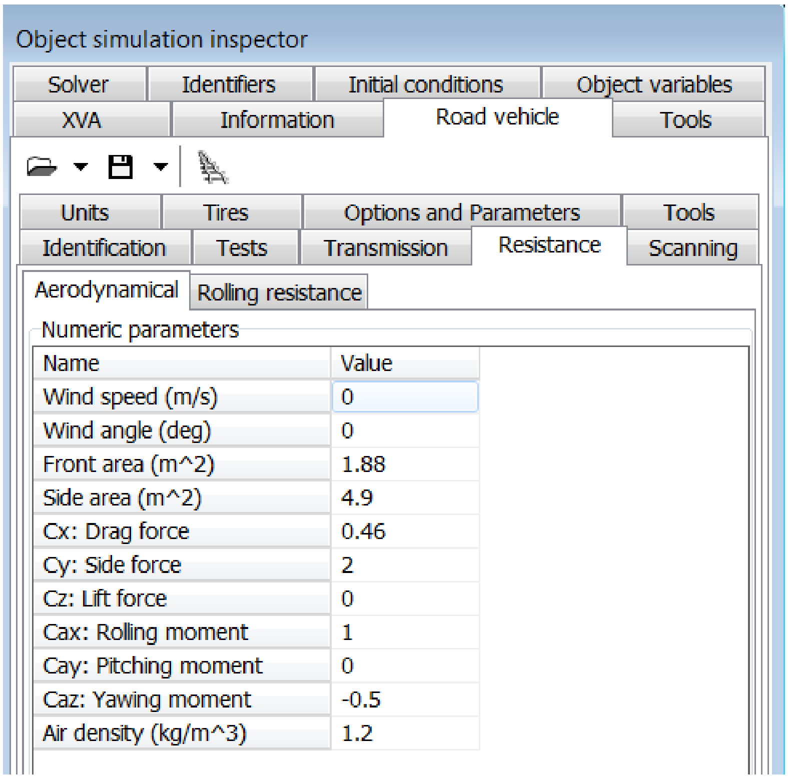



Figure 12.53. Parameters of aerodynamic forces

In **UM Simulation** program, the parameters of aerodynamic forces are set on the **Road vehicle | Resistance | Aerodynamical** tab of the simulation inspector, Figure 12.53. 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 12.51.

### 12.6.2. 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 [11]

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

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 12.7

Table 12.7

**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 |

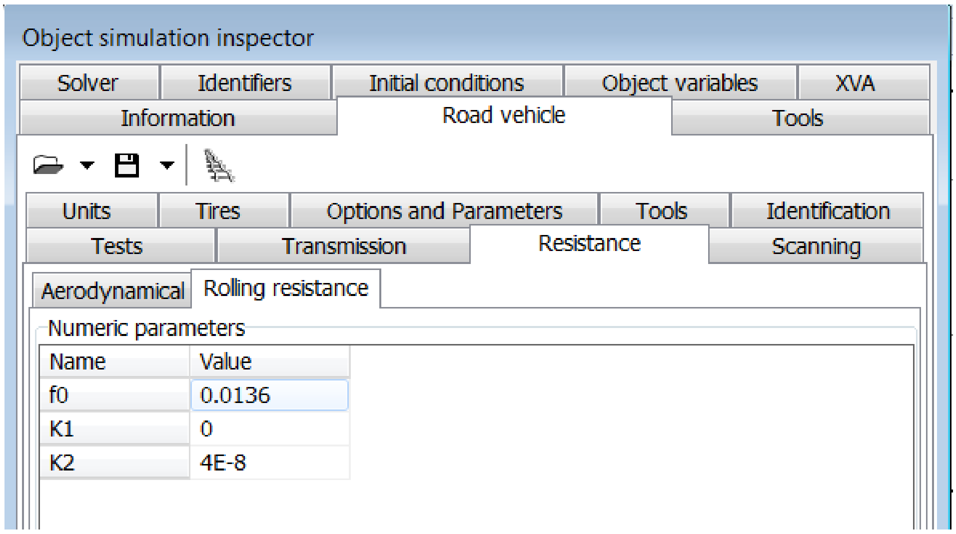



Figure 12.54. Parameters of rolling resistance

The rolling resistance parameters are set in the **Road vehicle | Resistance | Rolling resistance** tab of the simulation inspector, Figure 12.54.

# 12.7. Development of vehicle model

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

## 12.7.1. Model of a wheel

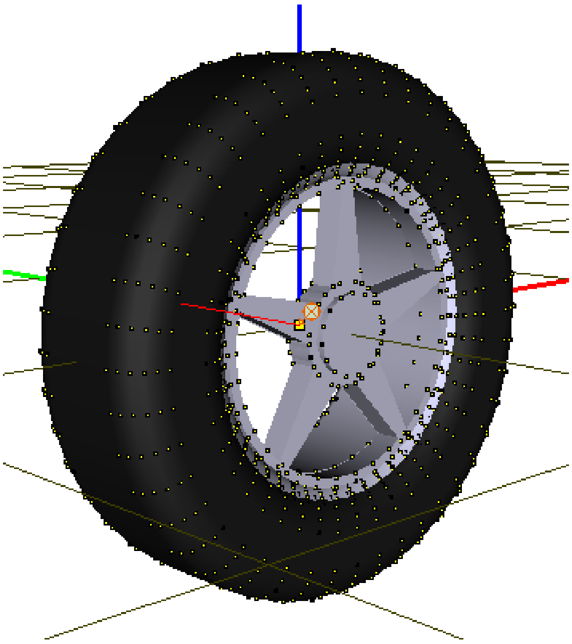


Figure 12.55. Model of a wheel as a body

A wheel in the UM model of a vehicle is a usual body with assigned image and inertia parameters, Figure 12.55. 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.

## 12.7.2. Visual wheel components


Figure 12.56. Wheel components

The **CarComponents.umc** library contains two visual components of wheels, Figure 12.56. Both of them add to the model a *right* wheel with a fully parameterized image (Figure 12.55), 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 12.55) 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 12.55) allows the user to create the rotational joint simultaneously.

Let us consider the process of visual adding the components.

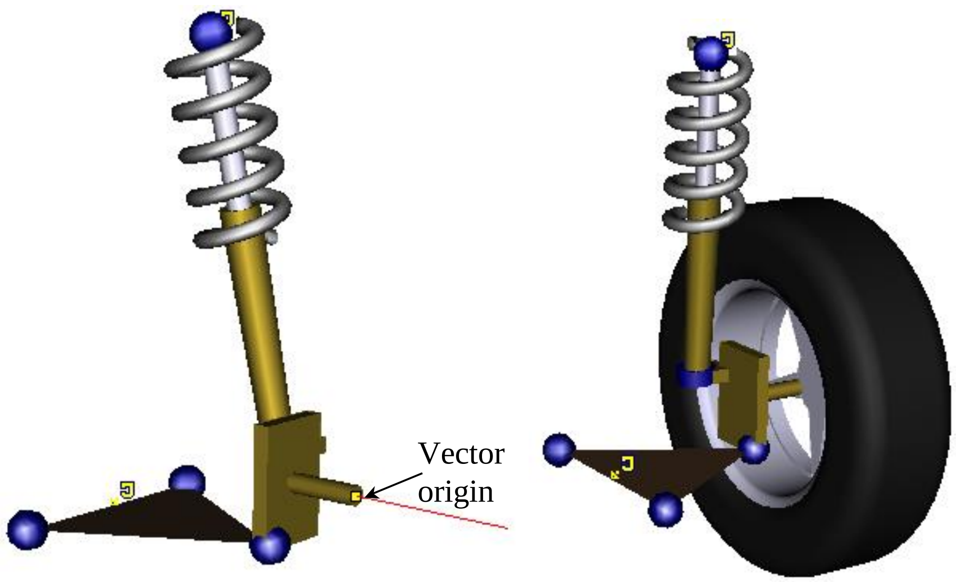
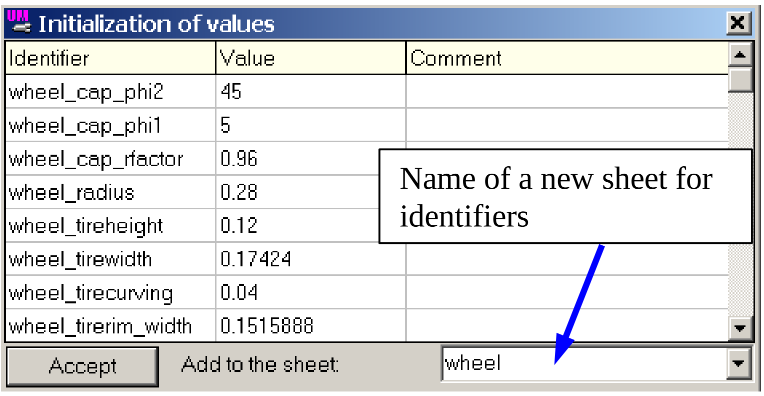



Figure 12.57. Visual adding of wheel and joint

*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.


Initialization of values

| Identifier | Value | Comment |
|---|---|---|
| wheel_cap_phi2 | 45 | |
| wheel_cap_phi1 | 5 | |
| wheel_cap_rfactor | 0.96 | |
| wheel_radius | 0.28 | |
| wheel_tireheight | 0.12 | |
| wheel_tirewidth | 0.17424 | |
| wheel_tirecurving | 0.04 | |
| wheel_tirerim_width | 0.1515888 | |

Accept Add to the sheet: wheel



Figure 12.58. 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 12.57).

2. Click the component button and then click on the origin of the vector.
3. Change identifiers corresponding to wheel inertia parameters and image.

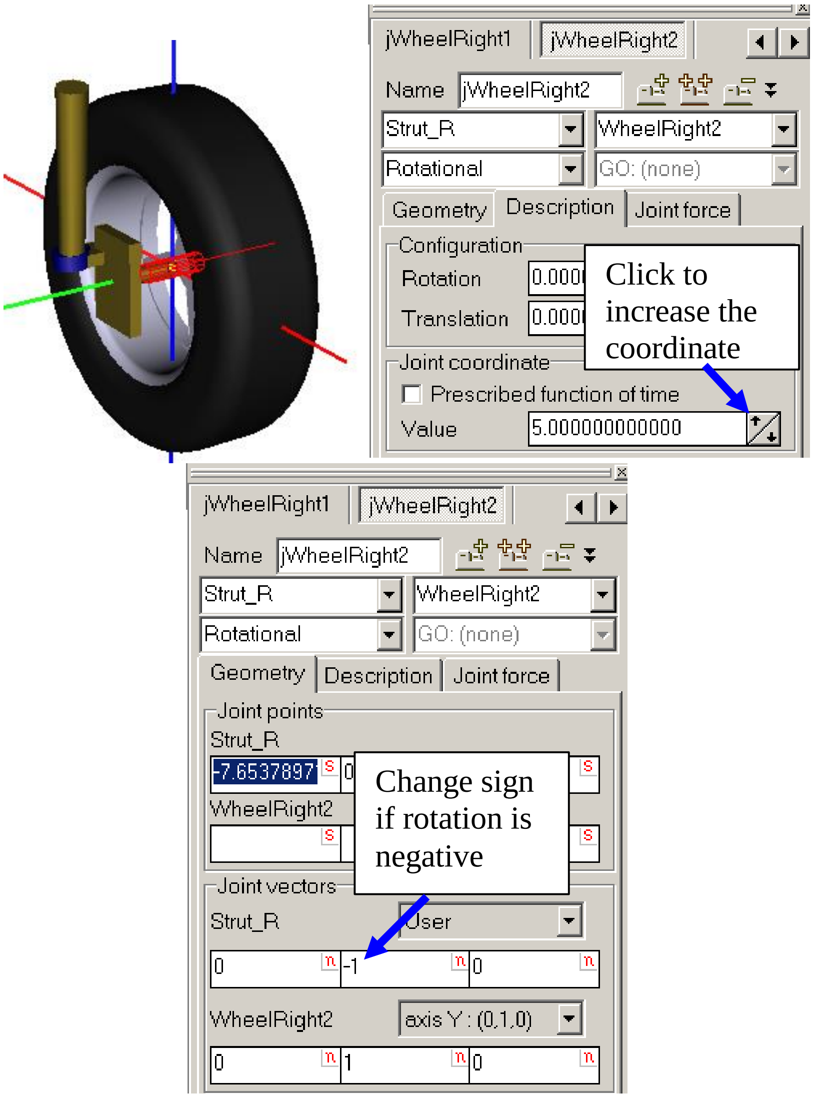



Figure 12.59. Verification of wheel rotational joint

**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 12.59.

## 12.7.3. 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.

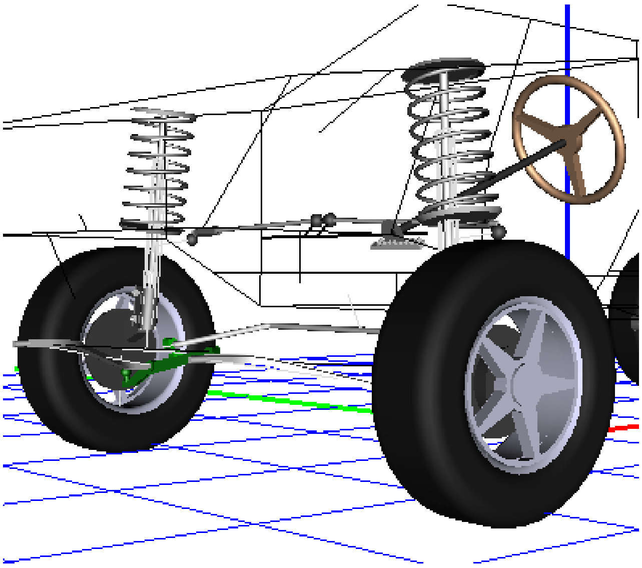


Figure 12.60. Suspension force elements

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.

## 12.7.4. Leaf springs

Figure 12.61. 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 12.62, 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 12.62. Stiffness matrix for generalized linear force element

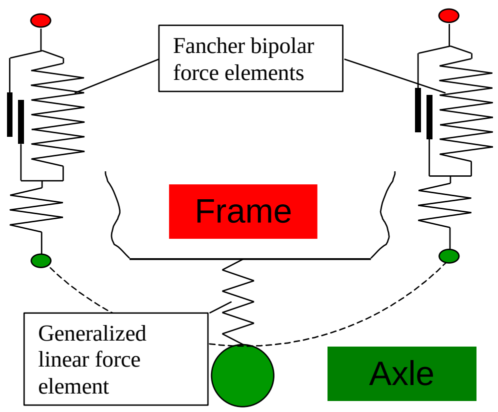



Figure 12.63. Model of a massless leaf spring

**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.

## 12.7.5. 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).

## 12.7.6. 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*.

### 12.7.7. Steer control

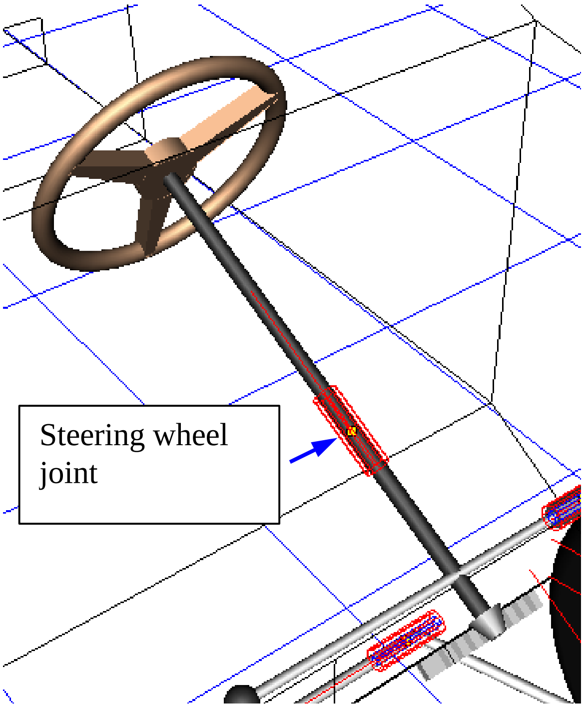



Figure 12.64. Steering wheel joint

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 12.64.

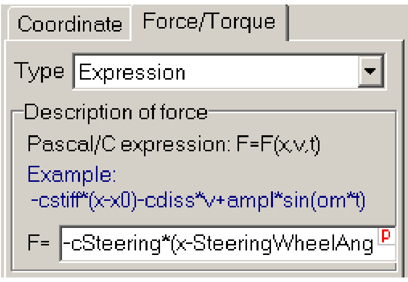



Figure 12.65. Steering control torque

The model of the steering control torque is described as a joint torque of the *Expression* type by the following equation, Figure 12.65:

*-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.

## 12.7.8. Longitudinal velocity control

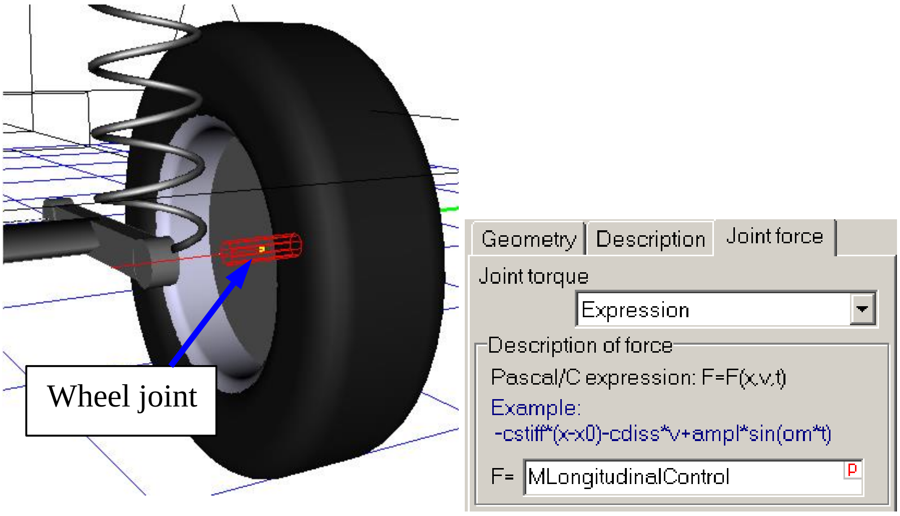



Figure 12.66. Joint torque for longitudinal velocity control

To make possible a control of the vehicle longitudinal velocity, the model of a vehicle needs a special traction joint torque. In the simplest case the torque is introduced in the driving-wheel joint, which is a rotational joint connecting the steering column with the vehicle, Figure 12.64. The model of the control torque is described as a joint torque of the *Expression* type by one and the same identifier for all of the driving wheels, Figure 12.65:

*MLongitudinalControl*

The user may introduce another name of identifier.

## 12.7.9. Locking vehicle movement

Some simulation results are obtained for a motionless vehicle, for example, computation of natural frequencies, evaluation of steering ratio, tests with harmonic loading, and so on. 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 12.67.

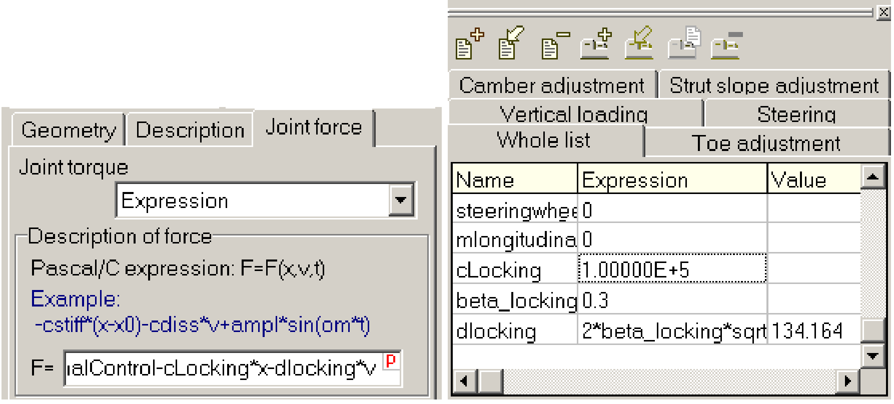



Figure 12.67. Locking joint torque (left); damping constant as an identifier-expression

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 12.67)

*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 = 1.0e5 N*m/rad, beta_locking=0.3*

## 12.8. 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 12.68. 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.

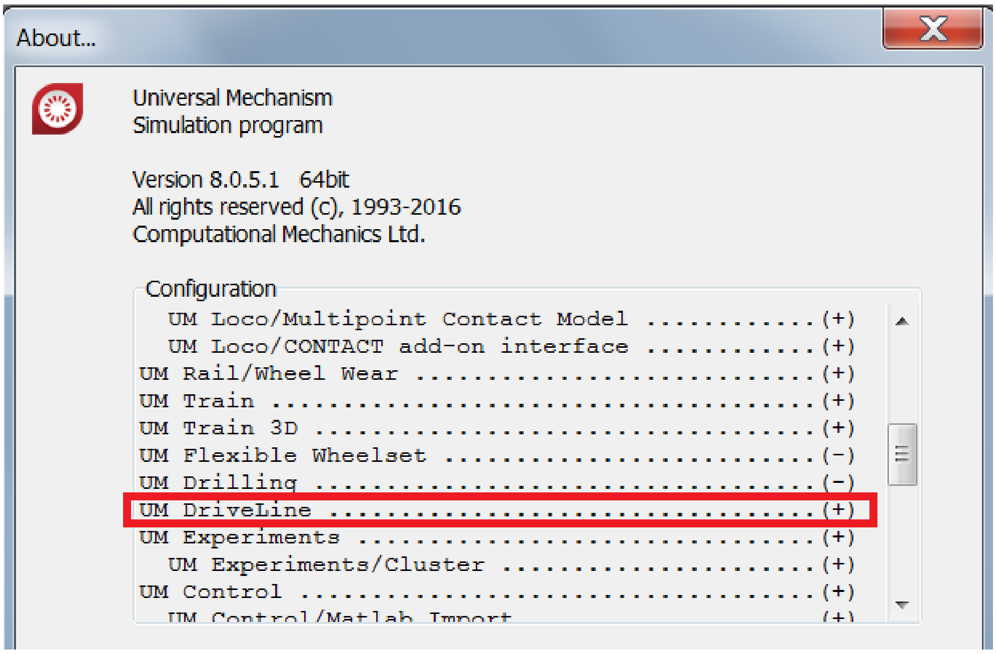



Figure 12.68. UM Driveline module in the current configuration

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 12.69.

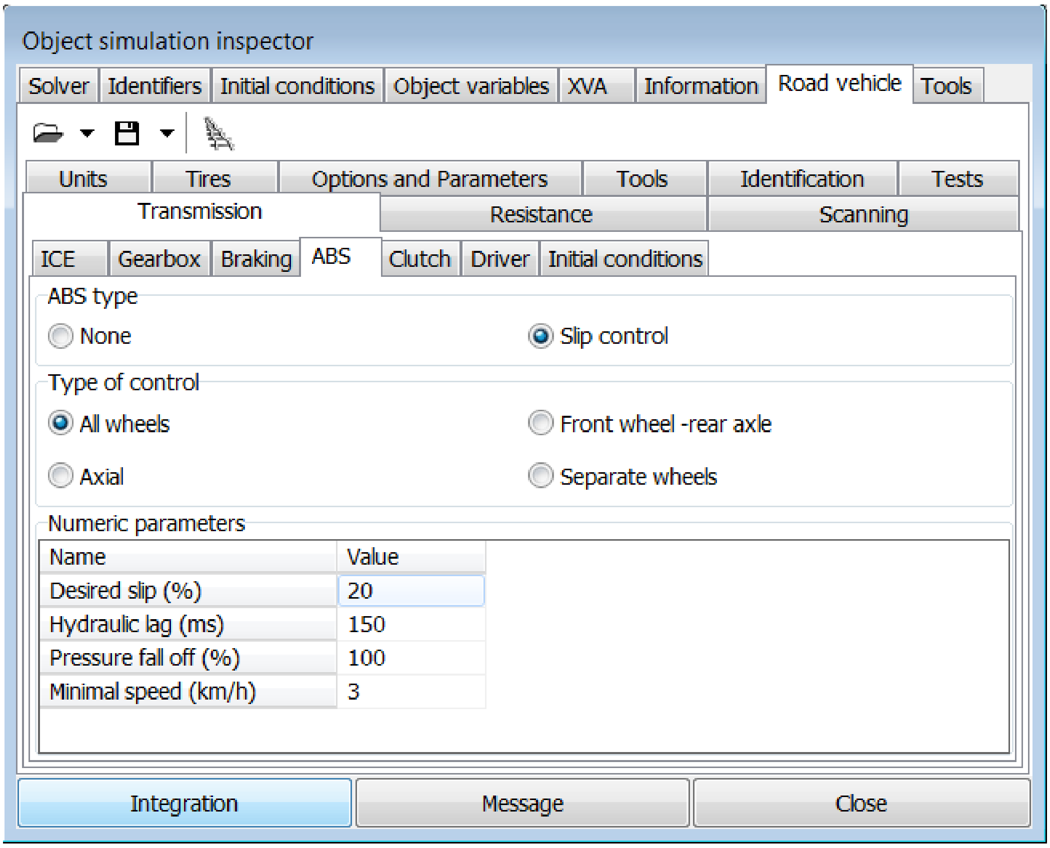



Figure 12.69. Tabs related to car transmission

## 12.8.1. Description of transmission elements in Input module

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

### 12.8.1.1. Internal combustion engine

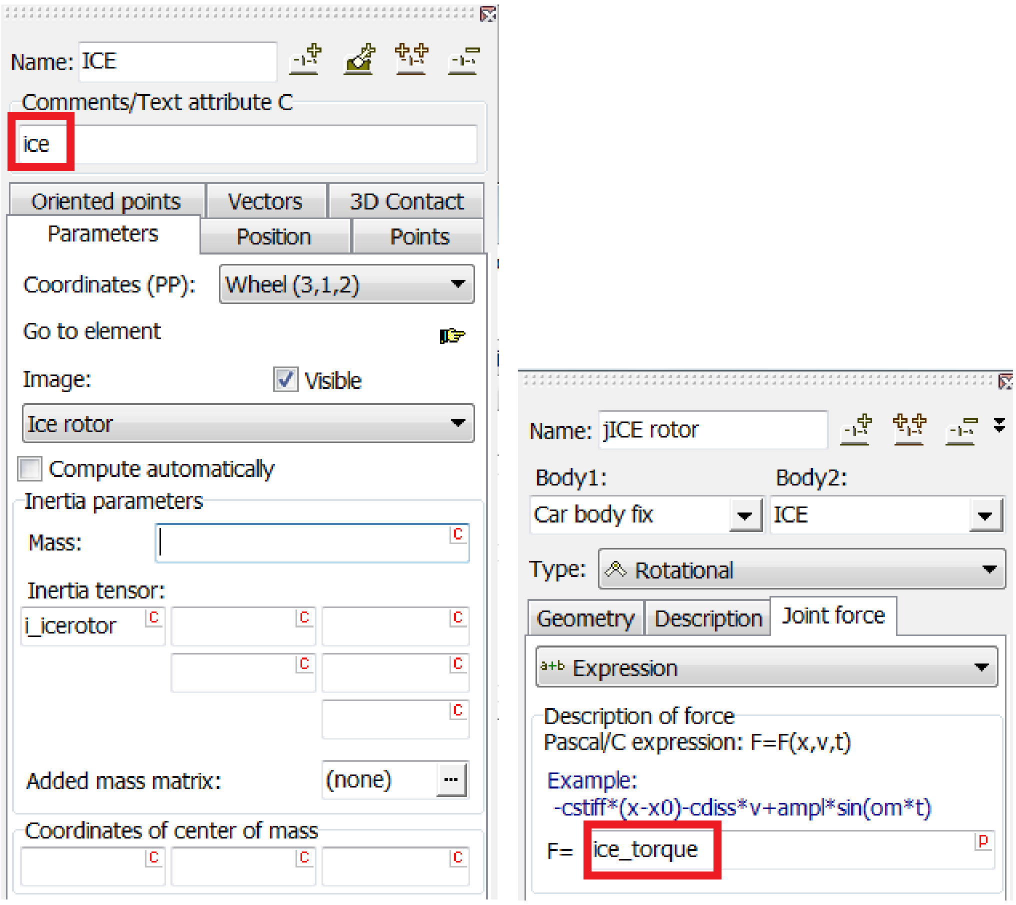



Figure 12.70. Example of an ICE shaft as a rigid body and a corresponding rotational joint

The following elements are necessary, Figure 12.70.

- 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 12.70, 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 12.70, right.

#### 12.8.1.2. Friction clutch

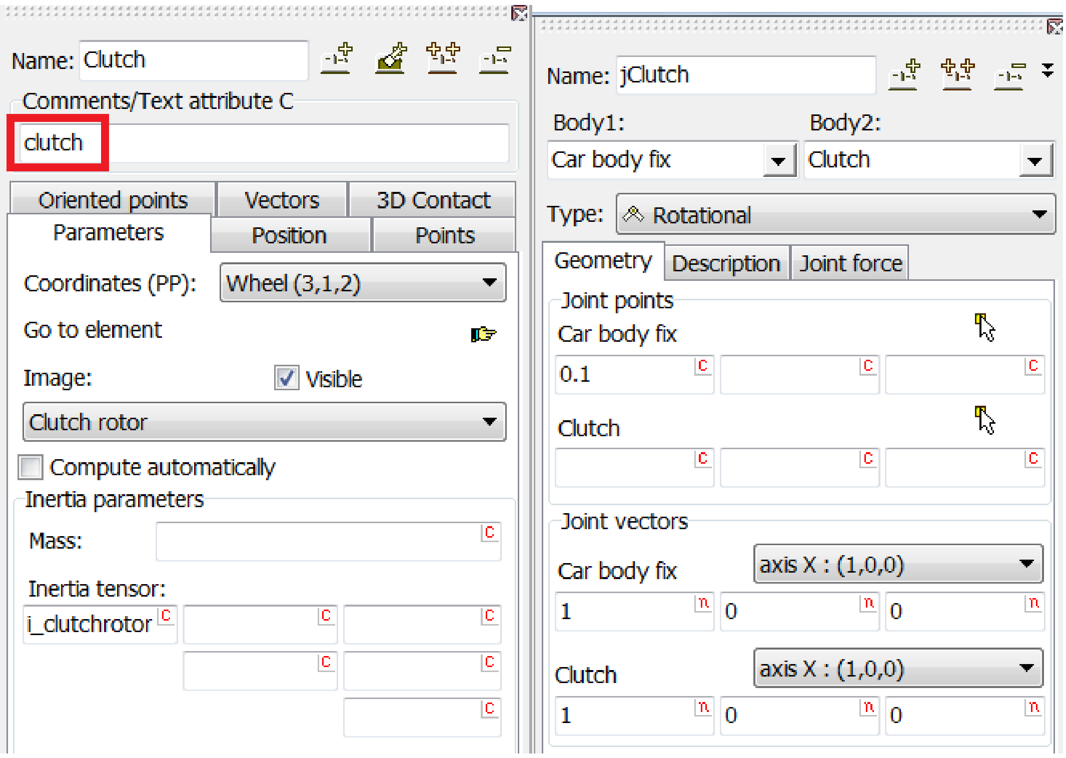



Figure 12.71. Example of a clutch plate as a rigid body and a corresponding rotational joint

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 12.71, .

- The body corresponding to the clutch plate must be marked by the text attribute C "clutch" Figure 12.71.
- 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.

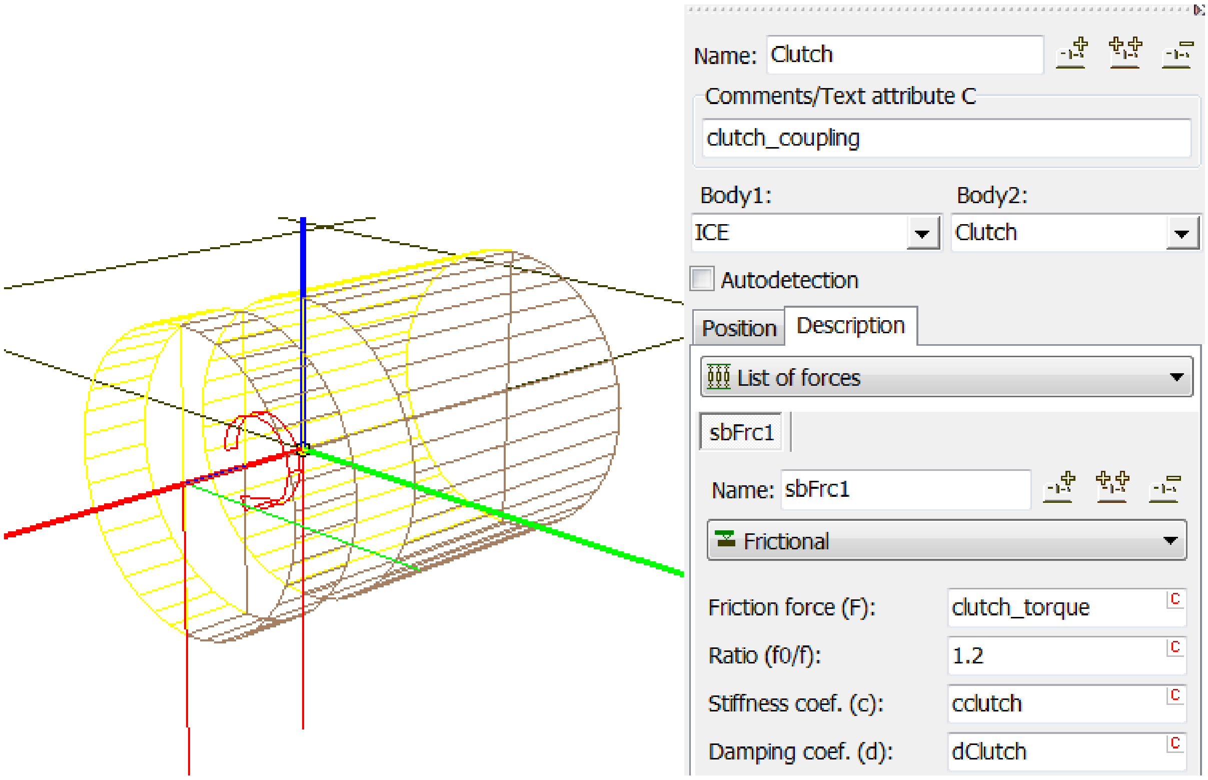



Figure 12.72. Scalar torque modeling friction between crankshaft and second clutch plate

- 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 12.72.

### 12.8.1.3. 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 12.73. 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.

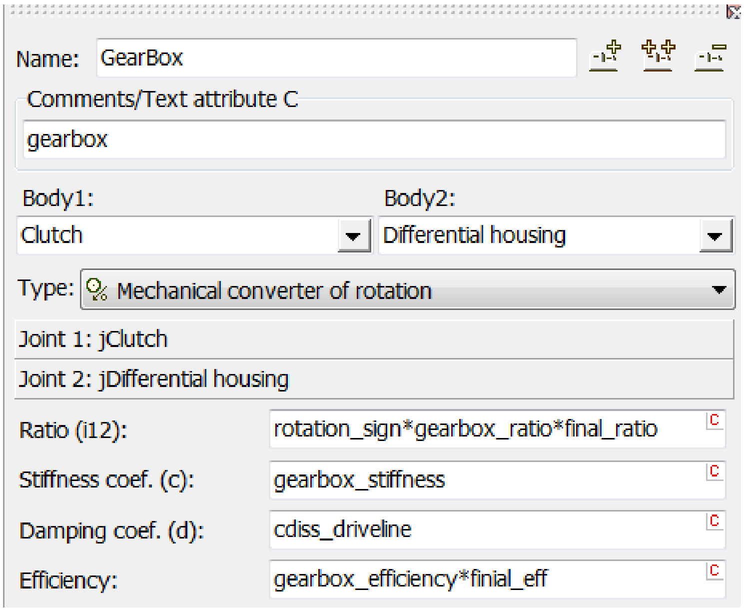



Figure 12.73. Example of modeling gearbox and final drive by single force element

Consider an example of modeling the gearbox and final drive by a single force element, Figure 12.73. 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.

# 12.9. Simulation of vehicle dynamics

## 12.9.1. Preparing for simulation

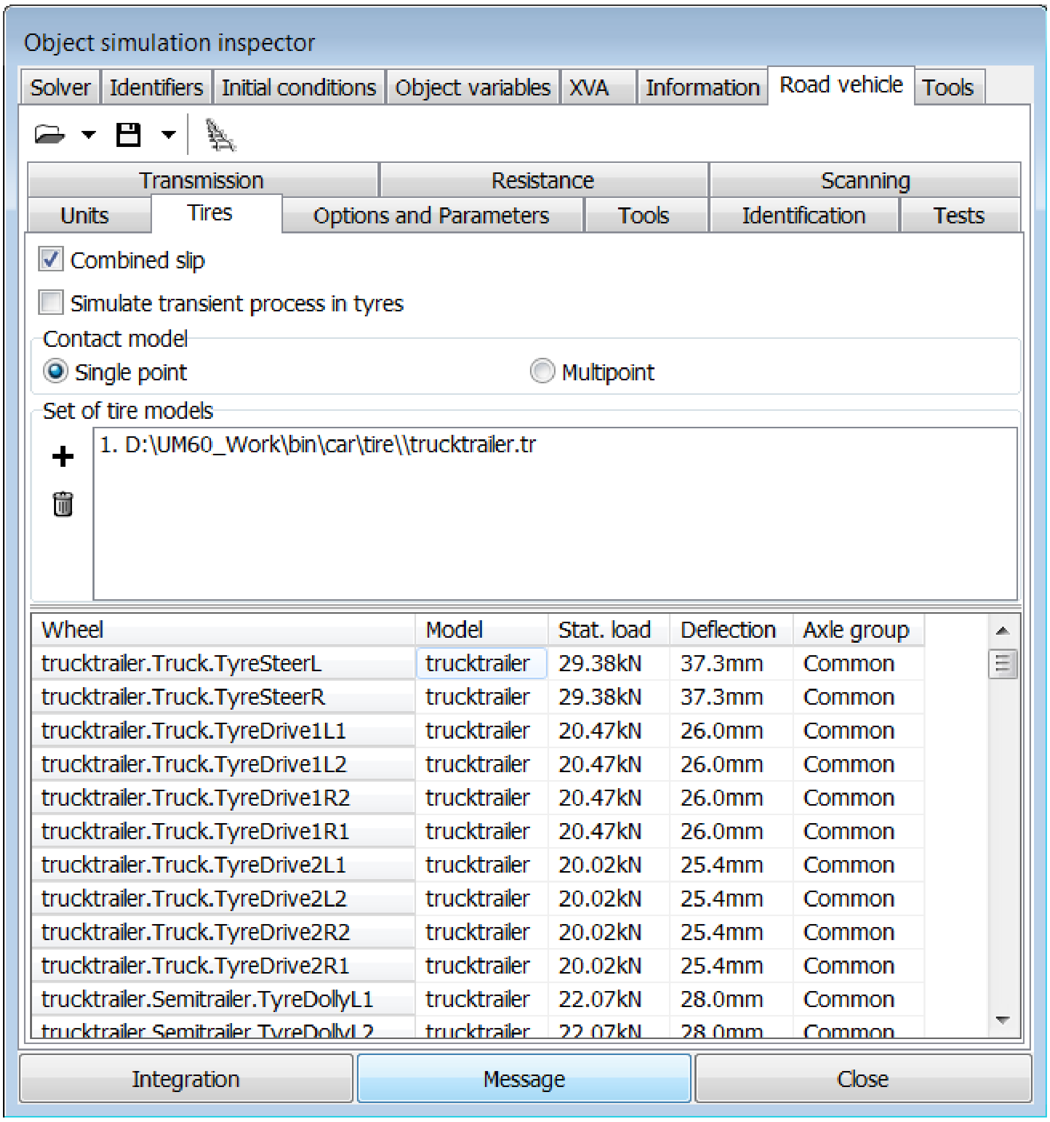



Figure 12.74. Object simulation inspector

The most part of the road vehicle specific data is entered and modified with the help of the **Road vehicle** tab in the **Object simulation inspector**, Figure 12.74. Use the **Analysis | Simulation…** menu command of the **UM Simulation** program to open the inspector. The entered data can be saved in 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** switch is on in the options of the **UM Simulation** program, Figure 12.75. Use the **Tools | Options…** menu command to call this window.

General information about **UM Simulation** program and its tools are concentrated in [Chapter 4](#).

The user should follow some definite steps to make a new created the model of a vehicle ready for simulation.

1. Generate and compile equations of motion in the **UM Input** program.
2. Run the **UM Simulation** program.
3. Assign tire models to the wheels, Sect. 12.5. *"Tire models"*, p. 12-33. If necessary, create new tire models.
4. Set current irregularities, Sect. 12.3.2.3. *"Assigning irregularities"*, p. 12-22.

Preparing the model requires identification of some substructures.

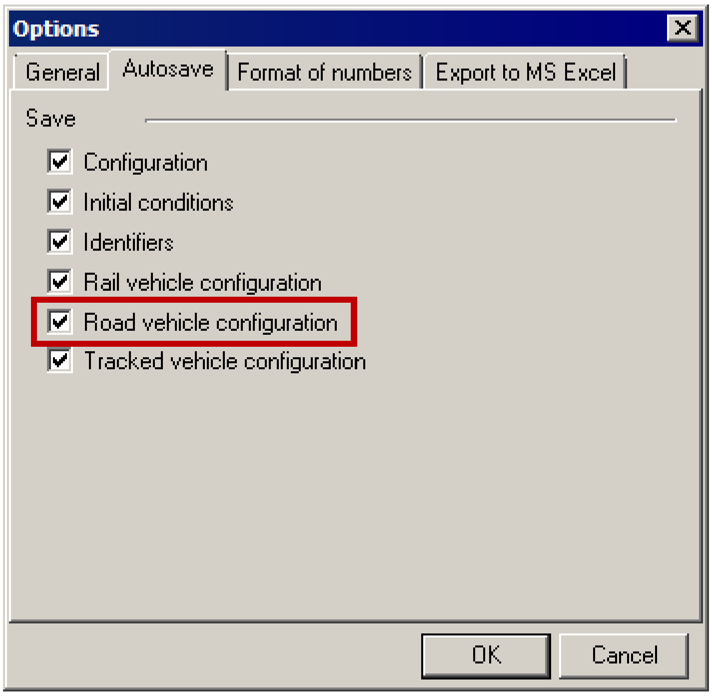



Figure 12.75. Options of **UM Simulation** program

### 12.9.1.1. Units

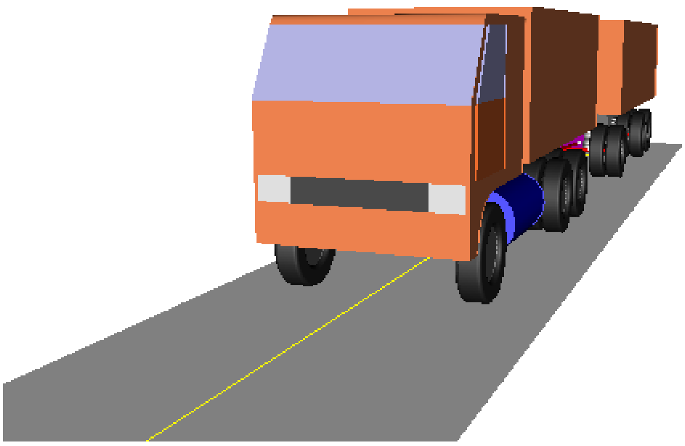


Figure 12.76. 2-unit vehicle: a truck with a trailer

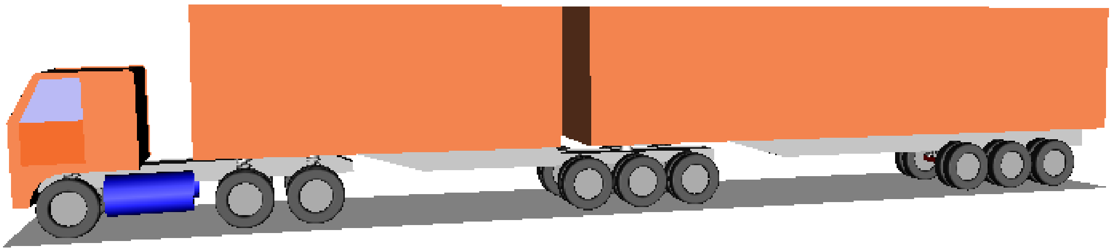


Figure 12.77. 3-unit vehicle: a truck with two semi- trailers

UM since version 5.0 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 12.76, Figure 12.77). 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 12.78.

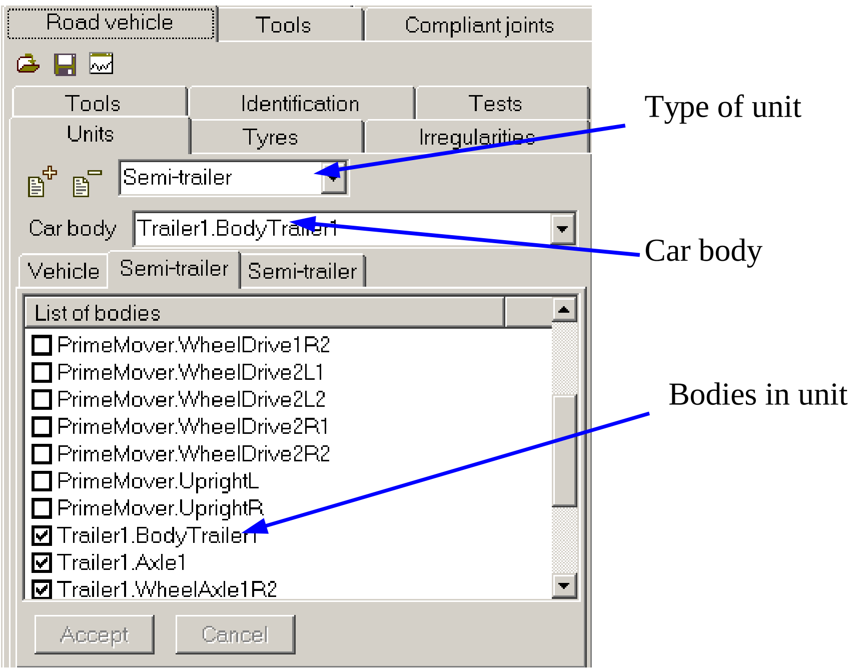



Figure 12.78. . Identifying units and car bodies

- 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.

### 12.9.1.2. Identification of steering

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

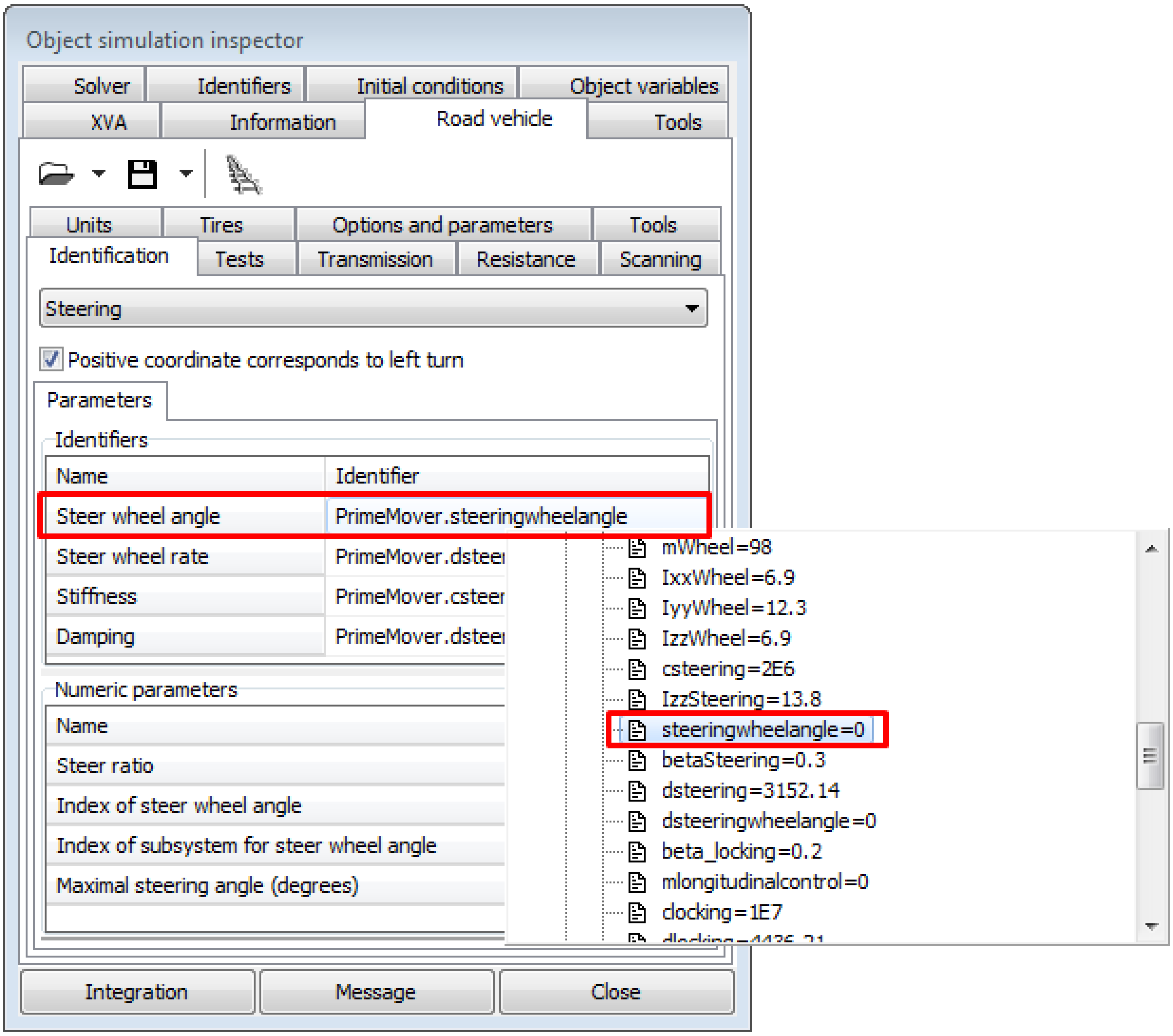



Figure 12.79. Identification of steering control

Identification of the steering control parameters of the model requires selecting four identifiers (see Sect. 12.7.7. *"Steer control"*, p. 12-65), Figure 12.79:

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

as well as two numeric values

- Steering ratio
- Index of the steer wheel angle

Double click by the left mouse button on the corresponding table row (Figure 12.79) to assign a model identifier to the steering control parameter. Use the direct input to set the numeric parameters.

To identify the index of the steering wheel angle use the **Initial conditions** tab of the **Object simulation inspector**. Find the wheel column joint by its name in the *Comment* column of the table, and the index is located in the first column (Figure 12.80).

**Remark.** The Steering wheel rotation test can be used for evaluation of the steering ratio.

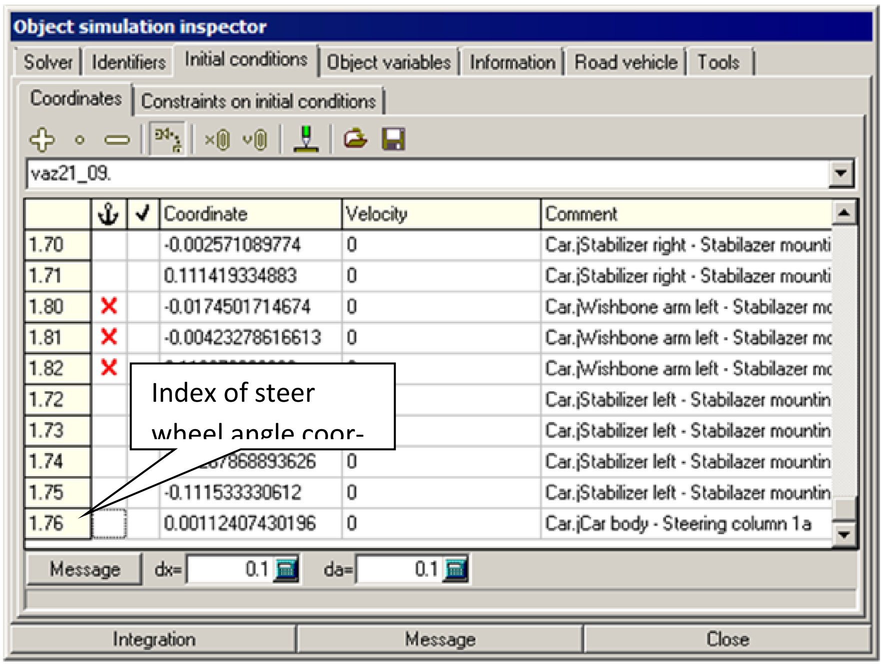



Figure 12.80. Identification of index of steering wheel angle coordinate

### 12.9.1.3. Identification of longitudinal velocity control

Use the **Road vehicle | Identification** tab of the **Object simulation inspector** to identify the *longitudinal velocity control* parameters. Select the *'Control V'* data type in the drop-down menu

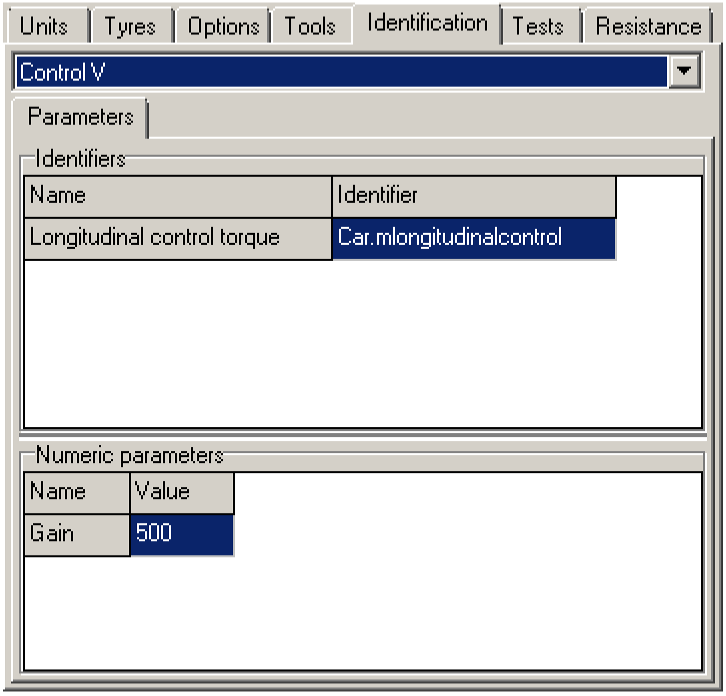



Figure 12.81. Identification of longitudinal velocity control parameters

Identification of the longitudinal velocity control parameters of the model requires selecting of one identifier (see Sect. 12.7.8. *"Longitudinal velocity control"*, p. 12-66), Figure 12.81:

- Longitudinal control torque

as well as one numeric values

- Control gain

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.

Use the direct input to set the gain value.

The control of the longitudinal velocity is realized to the proportional control law

$$M = -K(v - v_d),$$

where M is the torque (the value of the torque identifier), K is the gain, v is the current velocity of the vehicle, and $v_d$ is the desired velocity, which can be both constant and some function of time.

### 12.9.1.4. Identification of wheel rotation locking parameters

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

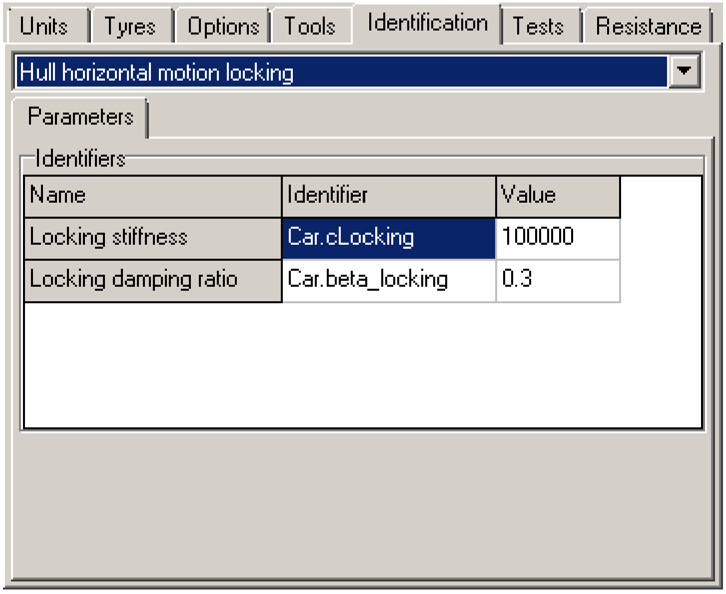



Figure 12.82. Identification of movement locking parameters

The following two identifiers should be assigned (Sect. 12.7.9. *"Locking vehicle movement"*, p. 12-66):

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

### 12.9.1.5. Open loop steering, longitudinal velocity and other functions

Use the **Road vehicle | Tools** tab of the **Object simulation inspector** to specify the desired *open loop steering, longitudinal velocity* and other functions functions.

Using this interface the user specifies a dependence on time or distance of the desired steering wheel angle and the longitudinal velocity, Figure 12.83.

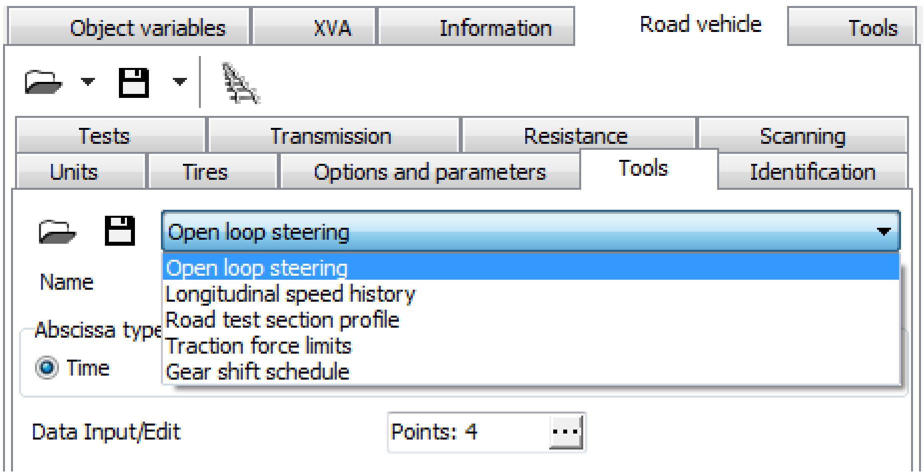



Figure 12.83. Interface for functions of time and distance

The function is a set of points with a possible spline smoothing. To set the function, the user calls the curve editor by clicking the ⋯ button.

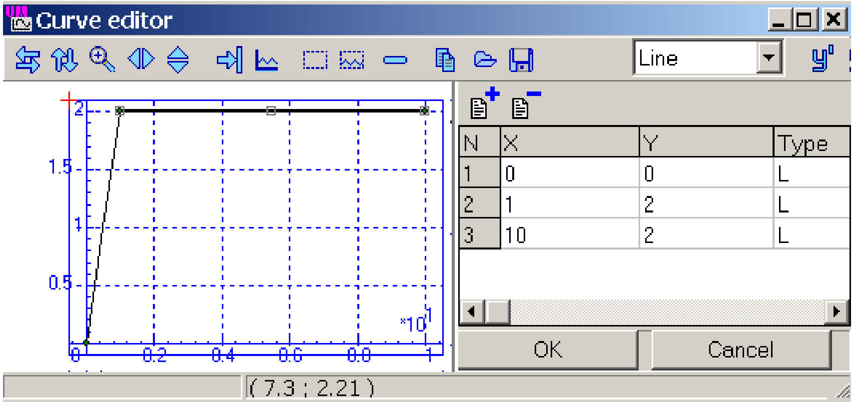



Figure 12.84. Setting functions with the curve editor

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

### 12.9.1.6. Test section profile of road

Figure 12.85. 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 12.28 or a speed bump in Figure 12.85 can be considered in UM as TSP only. **The tool is applied to the test with driver only.**

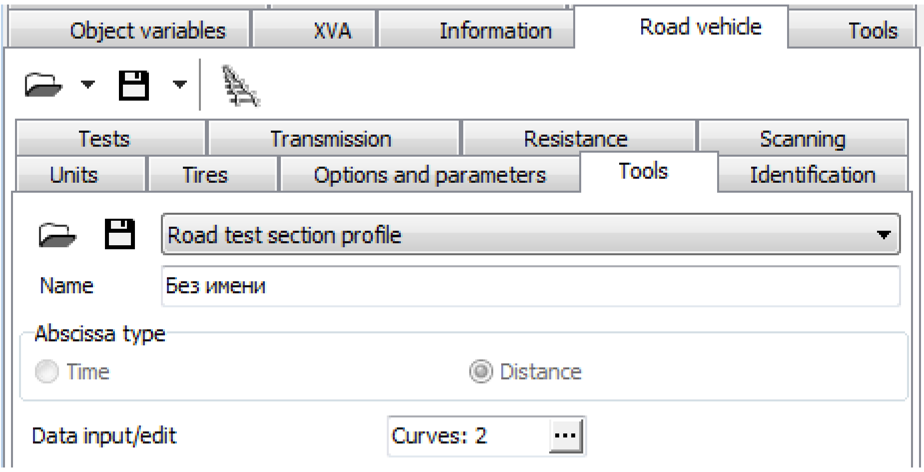



Figure 12.86. Tool for TSP description

TSP curves are described 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 12.87.

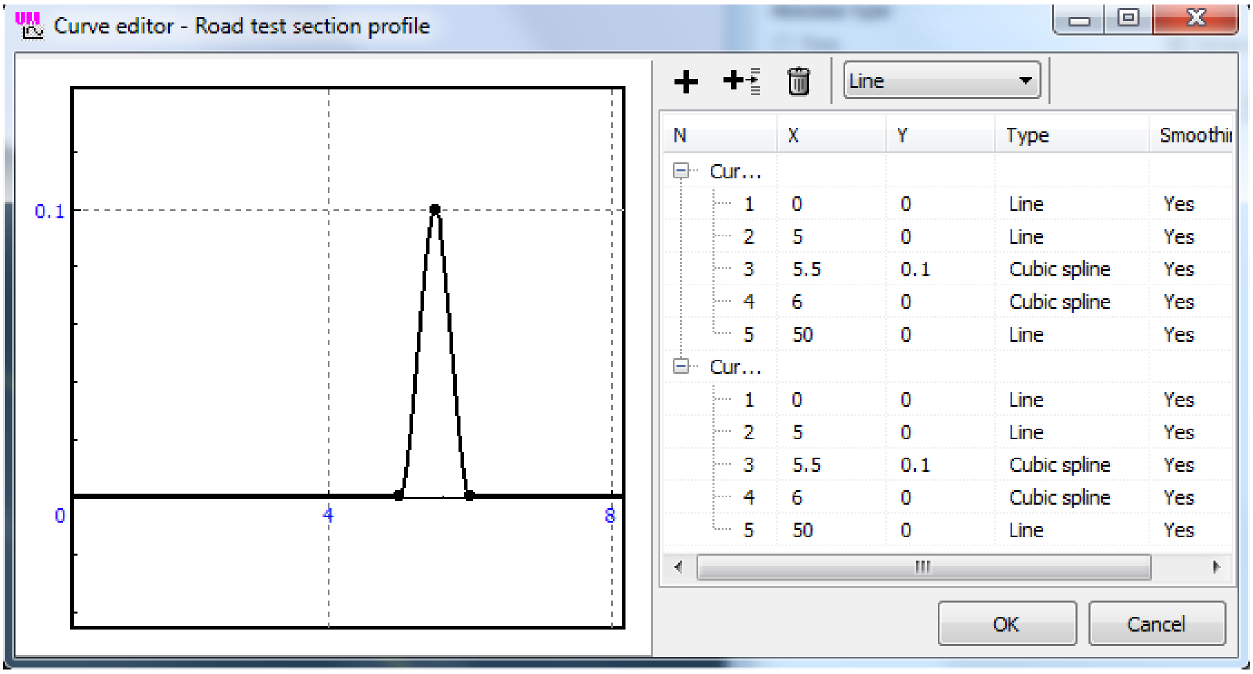



Figure 12.87. TSP curves for speed bump

If profile differs for the left and write tracks, the user should enter two curves like in Figure 12.87. 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.

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

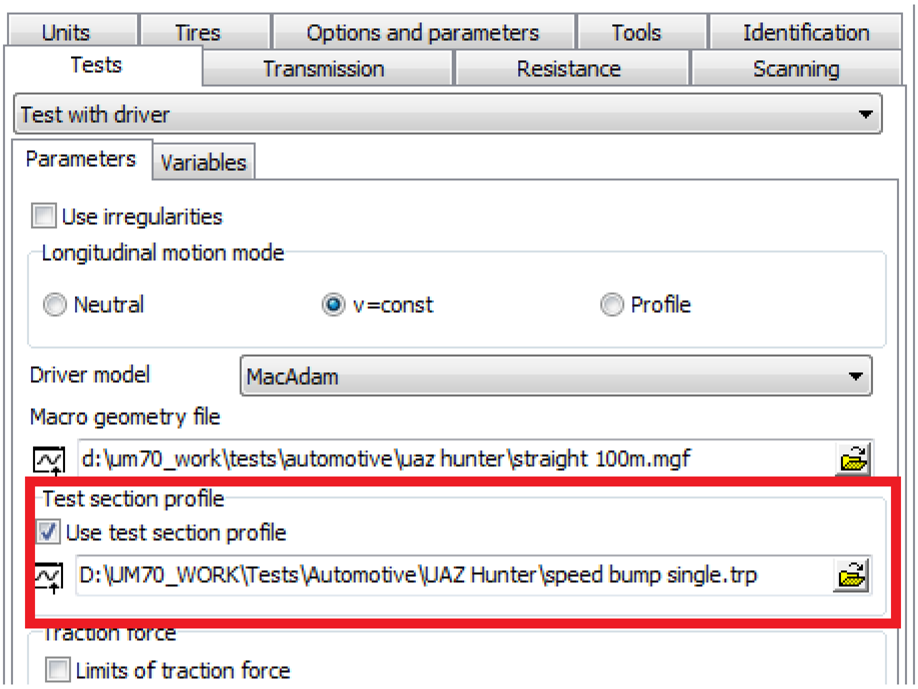



Figure 12.88. Choice of file with TSP

To run simulation with a TSP curve during the test with driver, select a *.trp file with the  button and check the **Use test section profile** option.

The **Use test section profile** option allows the user to compare promptly simulation results with and without TSP.

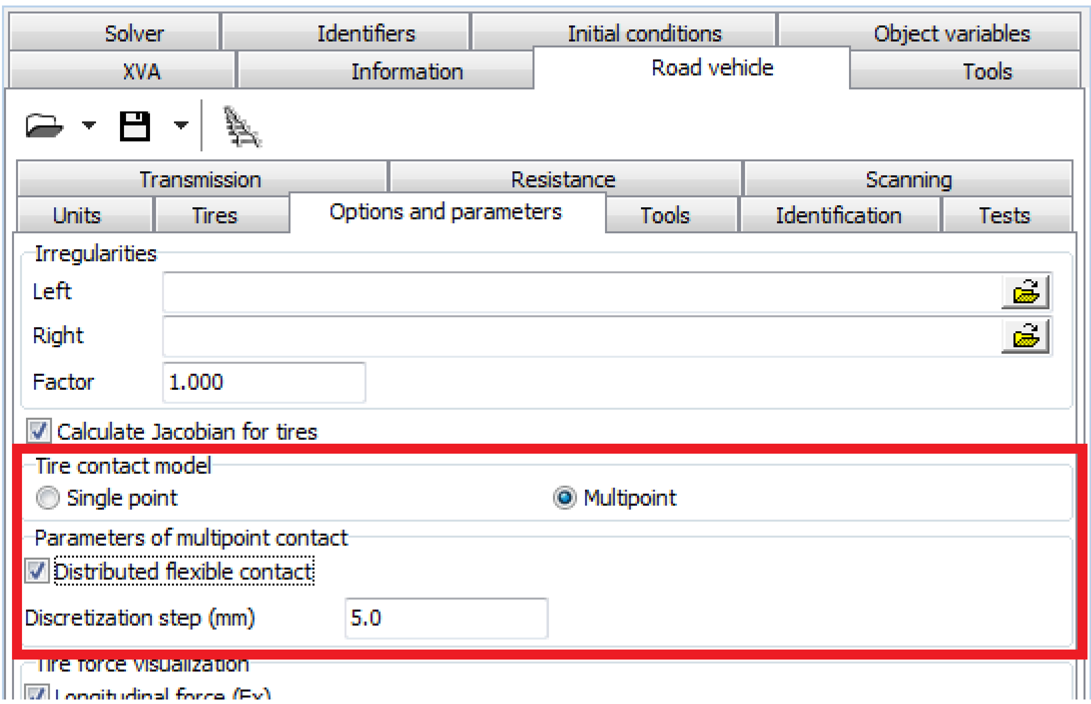



Figure 12.89. Parameters of tire contact model

Tune the tire contact model according to the selected TSP. Look at Sect. 12.5.1 *Single point and multipoint normal contact models* for detailed description of the tire contact models.

If the **Distributed contact model** option is unchecked, the *discrete point contact* is used, which usually applied for rolling up a step.

| | |
|---|---|
| Remark | An additional advantage of use the TSP consists in drowing 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. The user should set a small enough **Image step** to get an appropriate quality of the road deviation image in the animation window, Figure 12.90. |

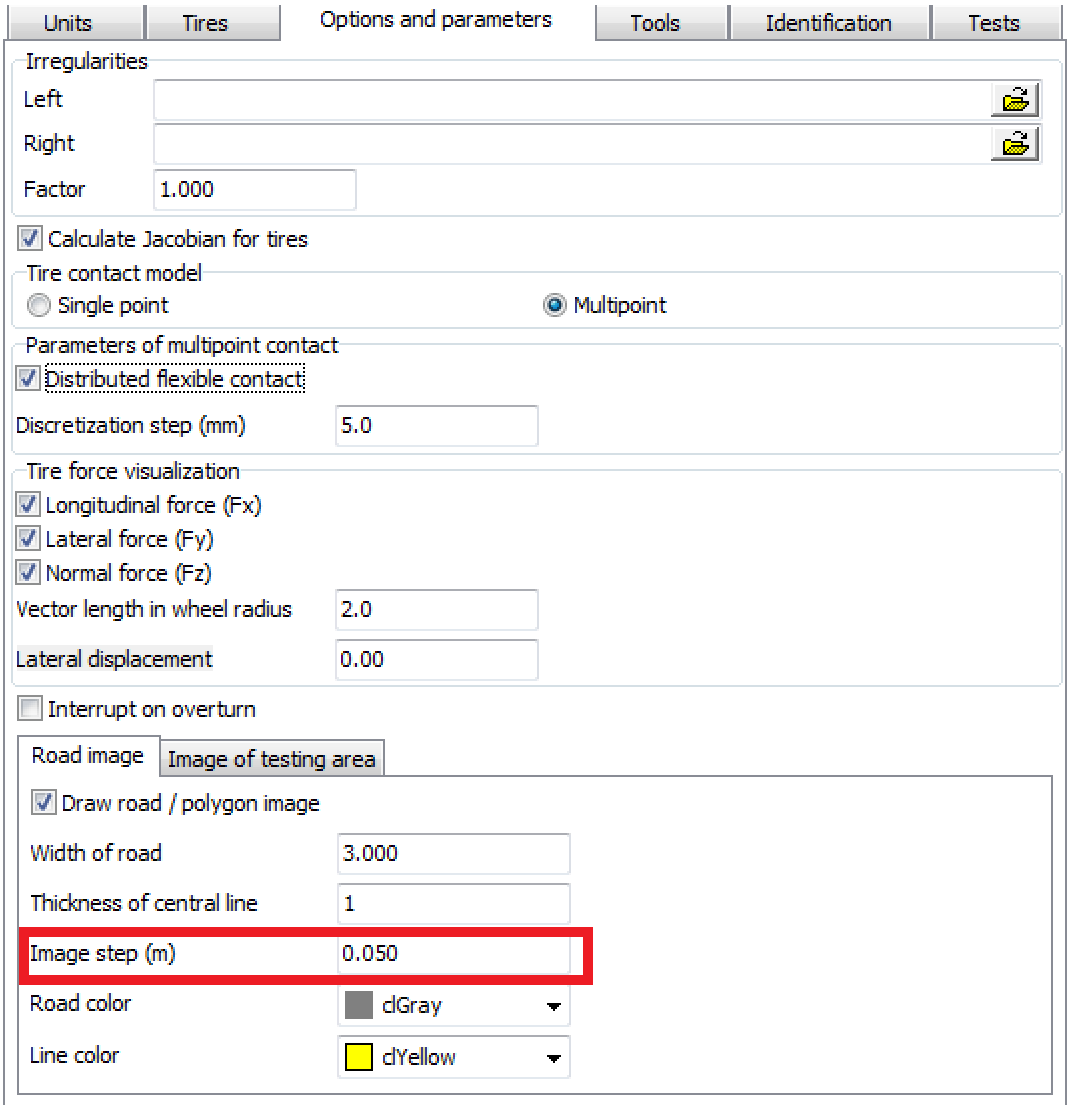



Figure 12.90. Image step parameter

## 12.9.2. Tests

### 12.9.2.1. General information

A set of tests realized in UM 5.0 is a basis for dynamic analysis of a vehicle. Currently the following test types are available.

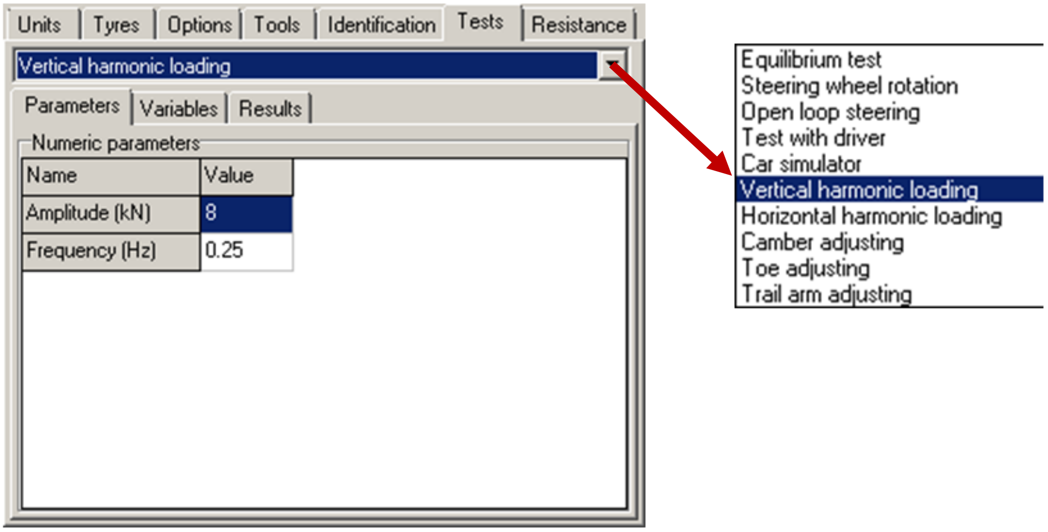



Figure 12.91. Choice of a test

**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.

**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) Vertical harmonic loading**

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

**f) Horizontal harmonic loading**

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

**g) Camber adjusting**

Quasistatic dependence of the camber angle on rotation of the adjusting bolts.

**h) Toe adjusting**

Quasistatic dependence of the toe angle on rotation of the adjusting bolts.

**i) Trail arm adjusting**

Quasistatic dependence of the pivot slope on rotation of the adjusting bolts.

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

Tests from the first group have the following features.

- Nonzero values of the movement locking parameters are requited (Sect. 12.7.9. *"Locking vehicle movement"*, p. 12-66, 12.9.1.4. *"Identification of wheel rotation locking parameters"*, p. 12-79);
- 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) and macro-geometry files; they may use longitudinal velocity functions (Sect. 12.9.1.5. *"Open loop steering, longitudinal velocity and other functions"*, p. 12-80) and irregularities. The tire models are used.

### 12.9.2.2. 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

### 12.9.2.3. Test variables

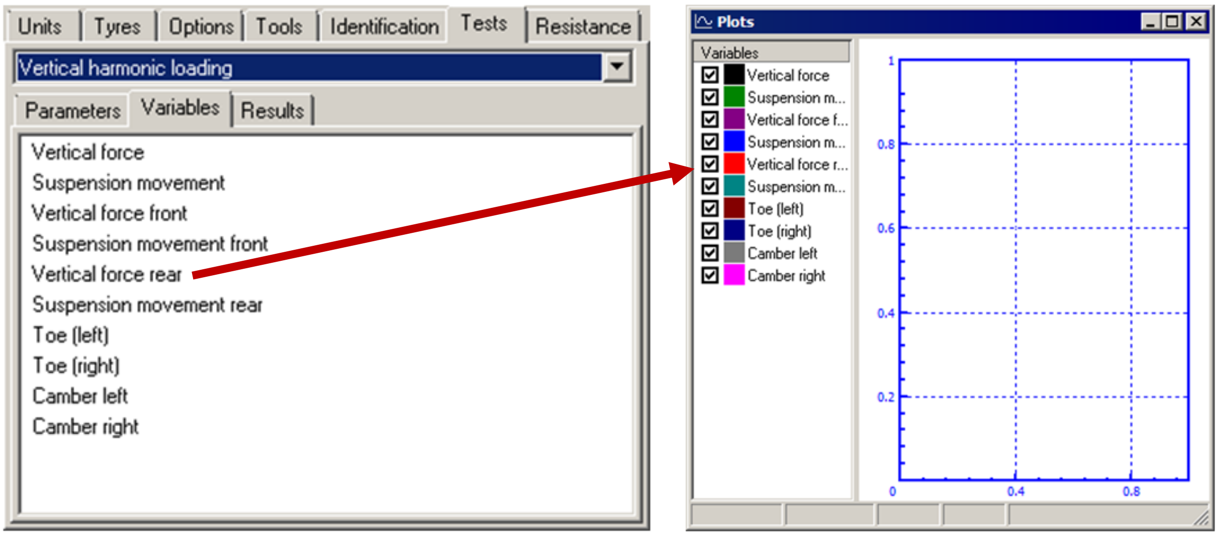



Figure 12.92. Dragging test variables in graphic window

Sets of standard variables are available for some of the tests. To get the plots of the variables during the simulation, the user should drag them by mouse into a graphic window, Figure 12.92 (see Chapter 4, Sect. *Simulation module notions and tools / Graphical window*).

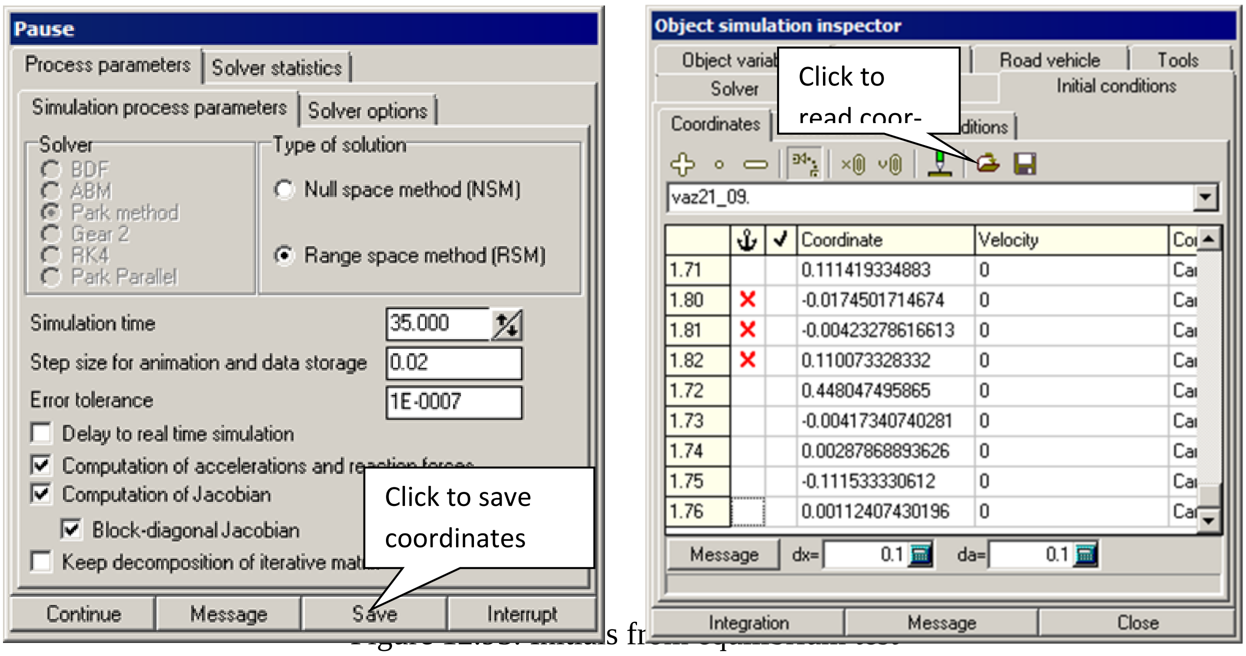
### 12.9.2.4. Equilibrium test

This is usually the first test with a new model of a vehicle. The test allows the user

- to get initial values of coordinates for usage with all other tests;
- to compute static load and deflections of wheels;
- to evaluate all static values of applied forces, e.g. in suspension.

The test requires Identification of wheel rotation locking parameters and strictly positive values of these parameters.

Click the **Integration** button on the **Object simulation inspector** to run the test.




To set the initial values of coordinates from the equilibrium test

- Run the test until the vehicle is near the equilibrium state; after end of the simulation time the **Pause** form appears; click the Save button to write the end values of coordinates and their time derivatives in a file; interrupt the test, Figure 12.93.
- Open the Initial conditions tab of the Object simulation inspector and read the coordinate values from the same file, Figure 12.93.
- Set zero values for velocities by clicking the v0 button.

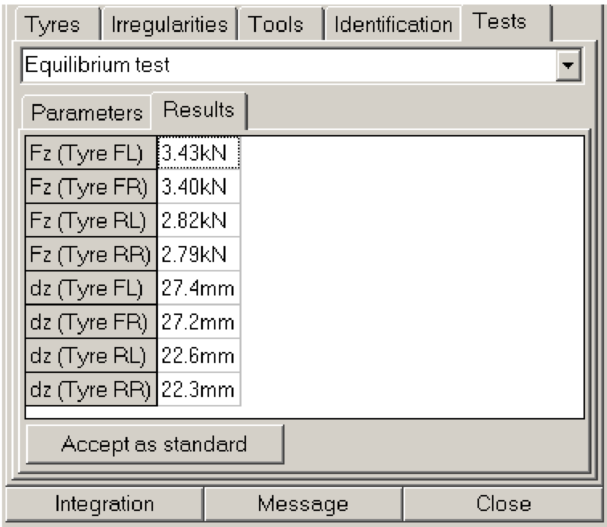



Figure 12.94. Results of equilibrium test

The results of the test include static loading (Fz) and deflections (dz) of wheels. These values are obtained at the moment of the test interruption after a long enough simulation time. Use the **Accept as standard** button to store the results for future usage:

- the static deflections are used for automatic computation of initial rolling rate of wheels before start of tests with longitudinal motion of vehicles,

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

- the static load is used in the MacAdam's model of driver.

### 12.9.2.5. 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 (four identifiers).

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 12.95.

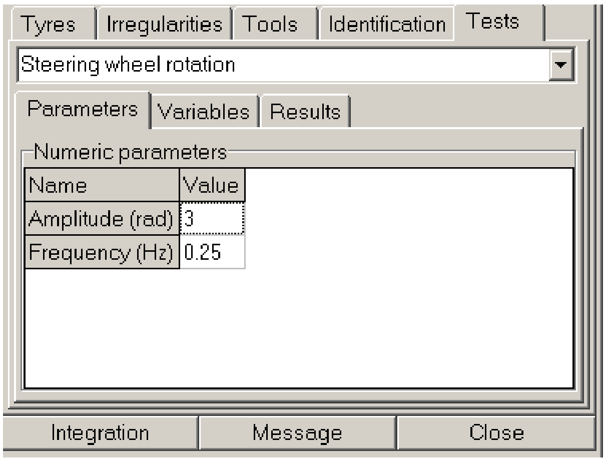



Figure 12.95. Parameters of steering wheel rotation test

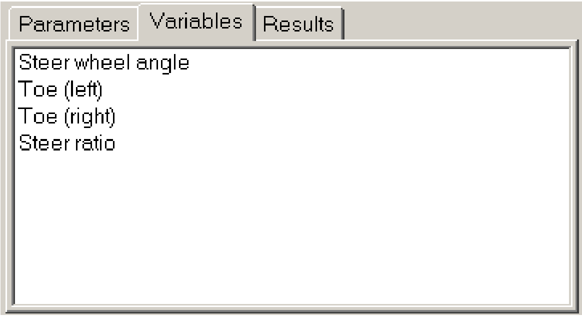



Figure 12.96. Variables of the test

Four standard variables are available with this test:

- Steering wheel angle $\alpha_w$
- Steer (toe) angles $\delta_l, \delta_r$
- Variable, which can be used for evaluation of the steering ratio

$$i_w^e = \begin{cases} \dfrac{3\alpha_w}{\delta_l + \delta_r}, |\delta_l + \delta_r| > 0.001 \\ 0, |\delta_l + \delta_r| \leq 0.001 \end{cases}$$

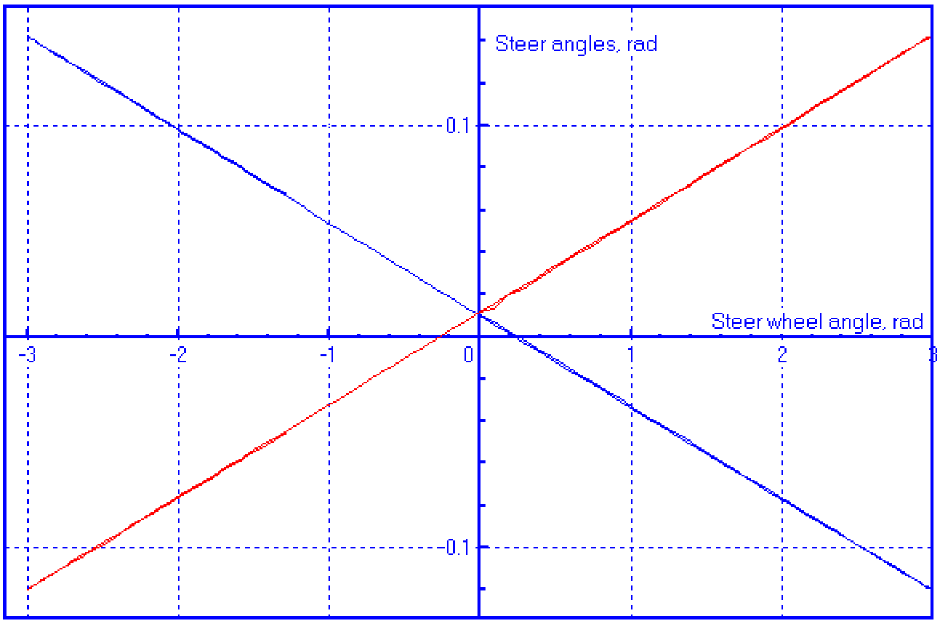



Figure 12.97. Steer angles versus steering wheel rotation angle

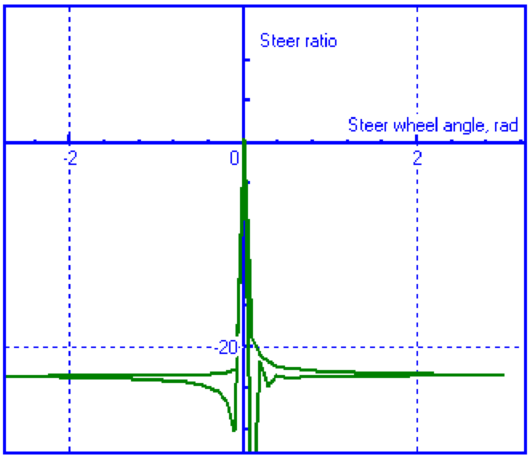



Figure 12.98. Variable $i_w^e$ versus steering wheel rotation angle

Figure 12.97, Figure 12.98 show examples of plotting the variables during the test.

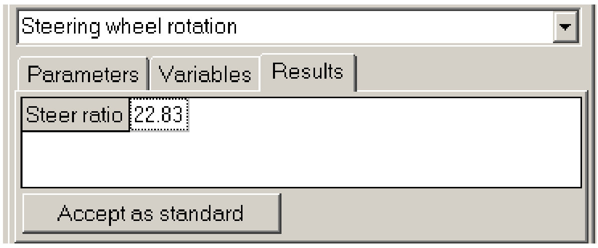



Figure 12.99. Result of steering wheel rotation test

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

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

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

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 accepts the computed steering ratio $i_w$ for other tests requiting identified steering system of the vehicle, Sect. 12.9.1.1. *"Units"*, p. 12-75.

#### 12.9.2.6. Open loop steering test

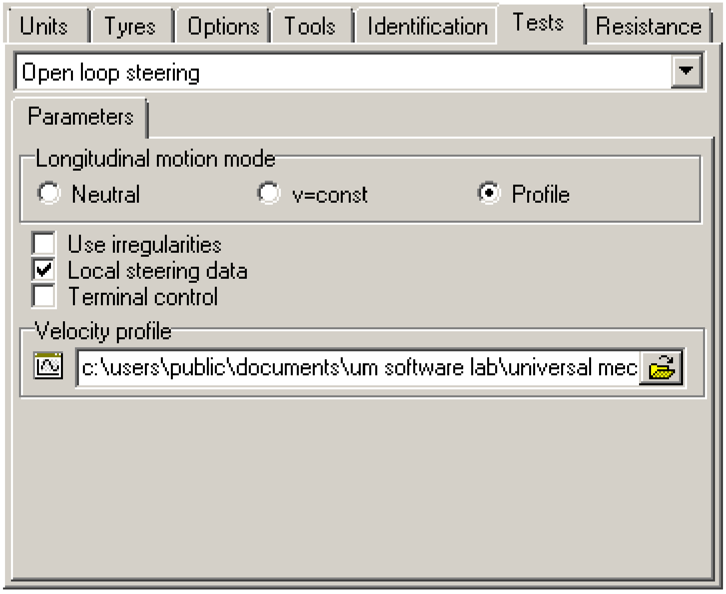



Figure 12.100. Open loop steering data

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. 12.5.9. *"Assignment of tire models to wheels"*, p. 12-54.
- 12.9.1.2. *"Identification of steering"*, p. 12-77 (four identifiers, and steer ratio).
- Steering angle function, Sect. 12.9.1.5. *"Open loop steering, longitudinal velocity and other functions"*, p. 12-80.
- Identification of the 12.7.8. *"Longitudinal velocity control"*, p. 12-66 if the **'v=const'** or the **Profile** item in the **Longitudinal motion mode** group is selected.
- Velocity function if the **Profile** item in the **Longitudinal motion mode** group is selected, Sect. 12.9.1.5. *"Open loop steering, longitudinal velocity and other functions"*, p. 12-80.
- Identification of irregularities if the Use irregularities box is checked, Sect. 12.3.2.3. *"Assigning irregularities"*, p. 12-22.

The following **longitudinal motion modes** are available in the test, Figure 12.100:

- **Neutral** – no driving moment, the motion by inertia.
- **v=const** – driving torques supports a constant value of the vehicle longitudinal velocity, which is defined by the standard identifier v0. Use the **Identifiers** tab of the **Objects simulation inspector** to set the desired speed of the vehicle. Note that the velocity unit is m/s, Figure 12.101.
- **Profile** – mode of motion with a variable velocity specified in a *. lvp file.

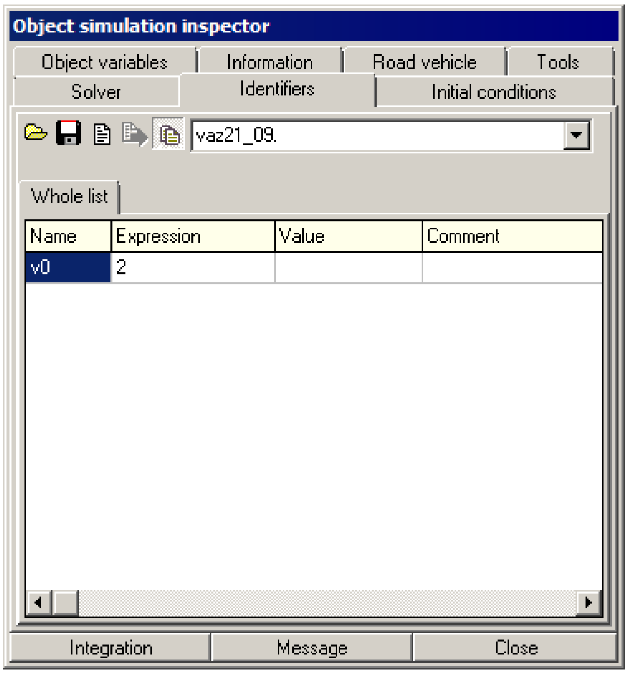



Figure 12.101. Identifier of velocity, m/s

The following check boxes specify some features of the test.

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

**Local steering data** – if off, a file with steering angle history should be assigned else the data currently presented in **the Road vehicle | Tools** tab are used, Sect. 12.9.1.5. *"Open loop steering, longitudinal velocity and other functions"*, p. 12-80.

**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.

#### 12.9.2.7. Closed loop steering test

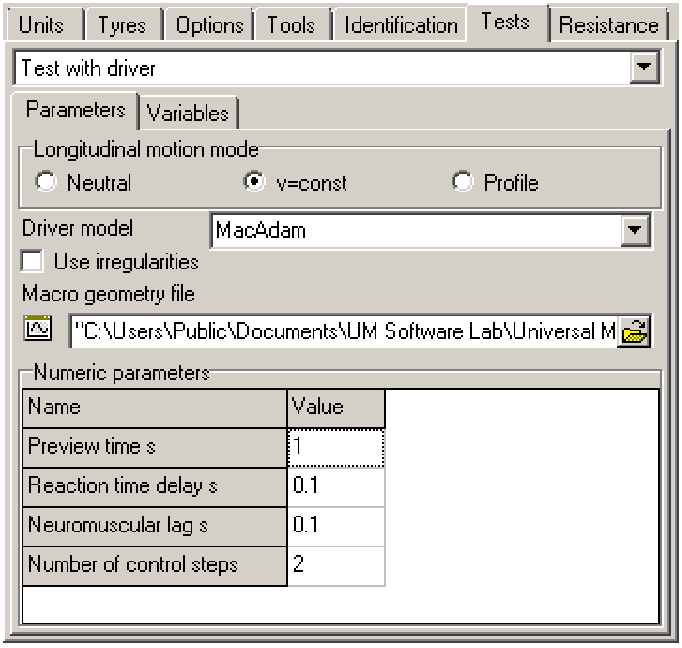



Figure 12.102. Closed loop steering test, MacAdam driver model

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. 12.4. *"Driver"*, p. 12-24.

The test requires

- Identification of the tire models, Sect. 12.5.9. *"Assignment of tire models to wheels"*, p. 12-54.
- 12.9.1.2. *"Identification of steering"*, p. 12-77 (four identifiers, and steer ratio).
- Identification of the 12.7.8. *"Longitudinal velocity control"*, p. 12-66 if the **'v=const'** or the **Profile** item in the **Longitudinal motion mode** group is selected.
- Velocity function if the **Profile** item in the **Longitudinal motion mode** group is selected, Sect. 12.9.1.5. *"Open loop steering, longitudinal velocity and other functions"*, p. 12-80.
- Identification of irregularities if the **Use irregularities box** is checked, Sect. 12.3.2.3. *"Assigning irregularities"*, p. 12-22.
- Macro-geometry describing a desired path, Sect. 12.3.1. *"Track macro geometry"*, p. 12-6.

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.

If the **Use irregularities** box is checked, irregularities are taken into account.

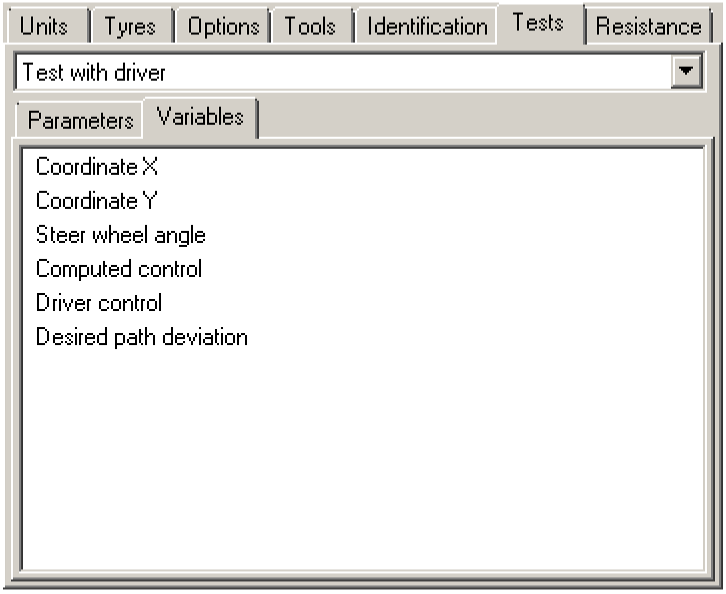



Figure 12.103. List of variables for closed loop steering test

The list of variables includes (Figure 12.103)

- 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).

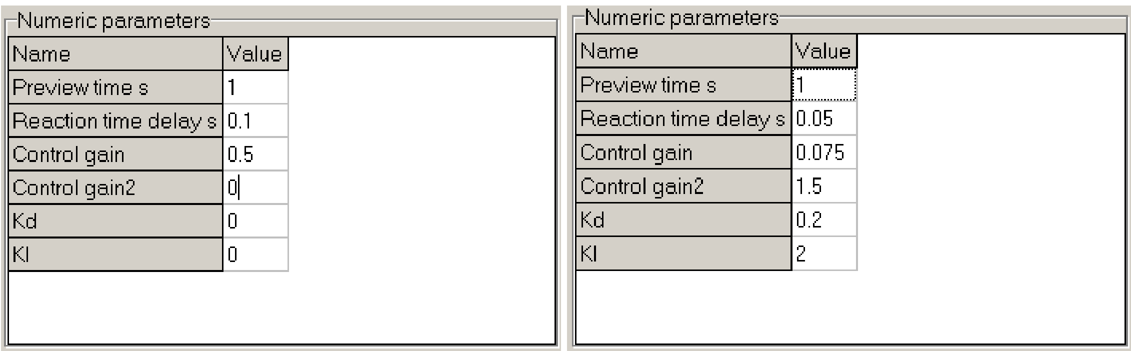



Figure 12.104. Parameters of SOP model (left) and PID-SOP model (right)

The second order preview model (SOP) and the PID controller + SOP (Sect. 12.4.2. *"Second order preview model"*, p. 12-28, 12.4.3. *"Combination of PID controller and second order preview model"*, p. 12-31) have the same list of parameters, Figure 12.104, and differs in their values.

- The SOP driver model has *always* zero values of parameters of PID controller, and the gain lies in the interval 0,5-1.
- The PID+SOP driver model requires nonzero values of PID constants: control gain 2 ($K_2$), $K_d, K_I$. The *Control gain* $K \in [0.05, 0.1]$.

**Note.** The PID-SOP model uses the derivative of the error, which requires a differentiable function of the desired path. In this case a spline interpolation of the path curve is necessary (Sect. 12.3.1. *"Track macro geometry"*, p. 12-6).

### 12.9.2.8. Vertical harmonic loading test

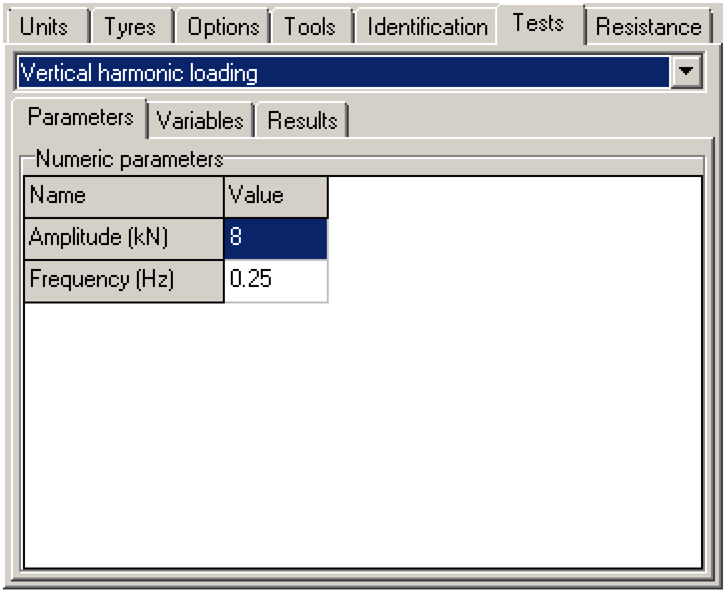



Figure 12.105. Vertical loading test parameters

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 12.105.

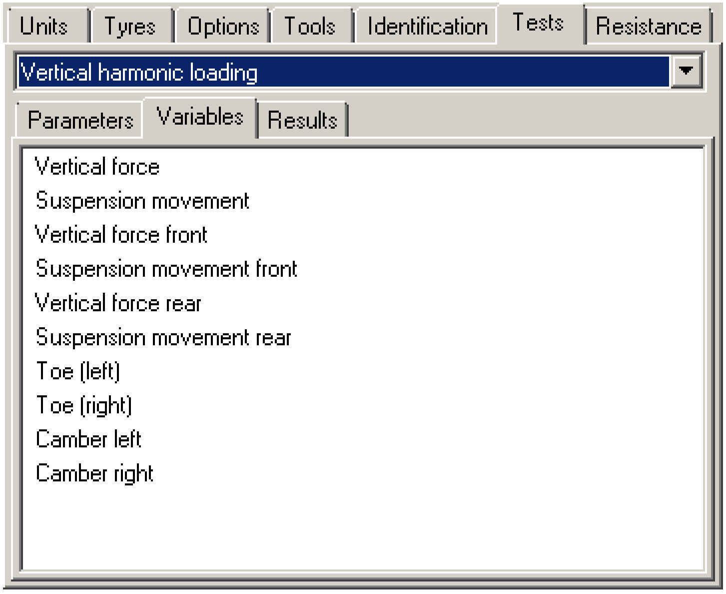



Figure 12.106. Vertical loading test variables

The list of test variables is shown in Figure 12.106. 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.

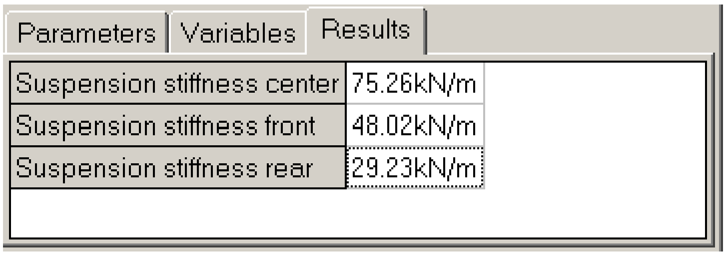



Figure 12.107. Vertical loading test results

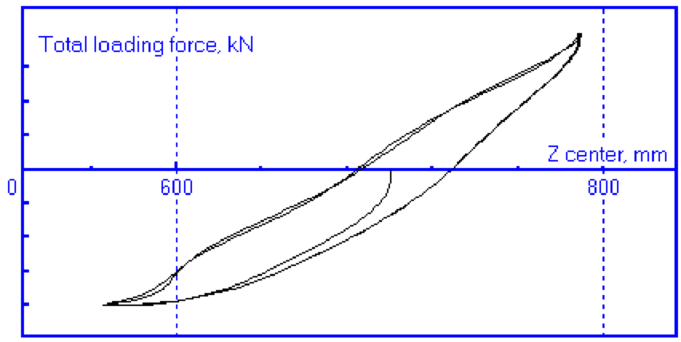



Figure 12.108. Simulation results: Load versus vertical position of center of mass

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.

### 12.9.2.9. Horizontal harmonic loading test

The second order preview model (SOP) and the PID controller + SOP (Sect. Second order preview model, Combination of PID controller and second order preview model) have the same list of parameters, Figure 12.98, and differs in their values.

- The SOP driver model has *always* zero values of parameters of PID controller, and the gain lies in the interval 0,5-1.
- The PID+SOP driver model requires nonzero values of PID constants: control gain 2 (K2), $K_d, K_I$. The *Control gain* $K \in [0.05, 0.1]$.

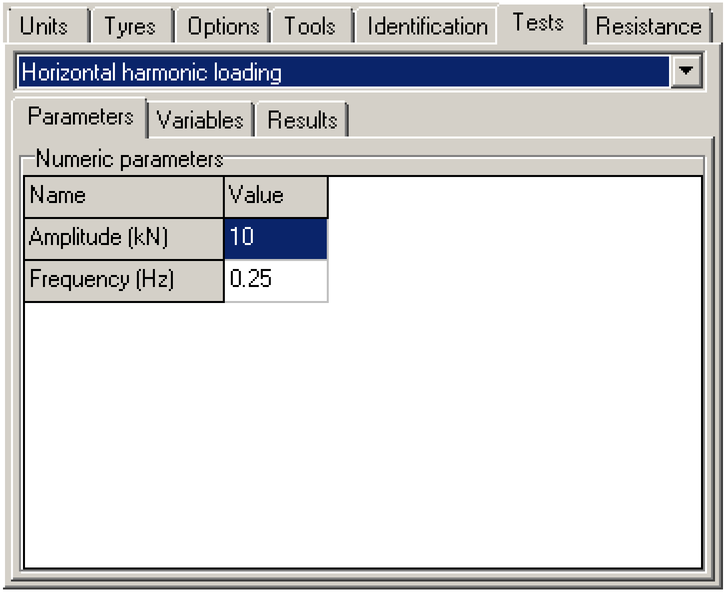



Figure 12.109. Horizontal loading test parameters

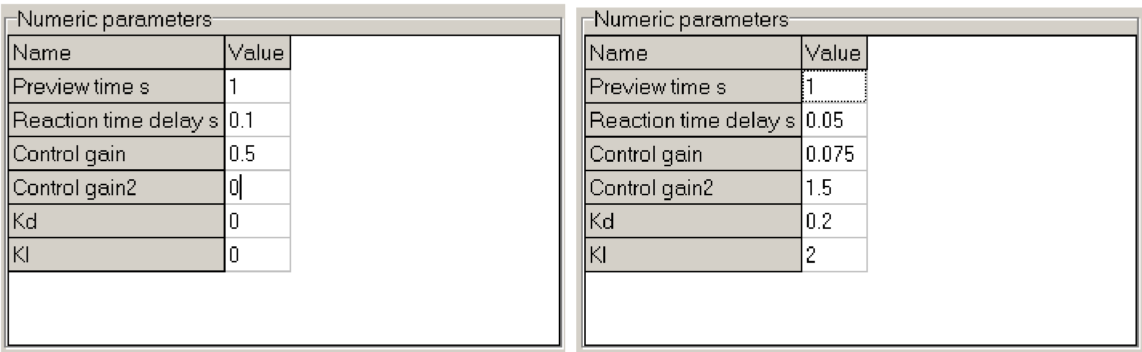



Figure 12.110. Parameters of SOP model (left) and PID-SOP model (right)

**Note.** The PID-SOP model uses the derivative of the error, which requires a differentiable function of the desired path. In this case a spline interpolation of the path curve is necessary (Sect. Track macro geometry).

#### 12.9.2.10. Toe, camber, trail arm adjusting

The tests require description in the model of the vehicle translational displacements caused by rotation of adjusting bolts. Currently the corresponding documentation is not available.

### 12.9.3. Road vehicle specific variables

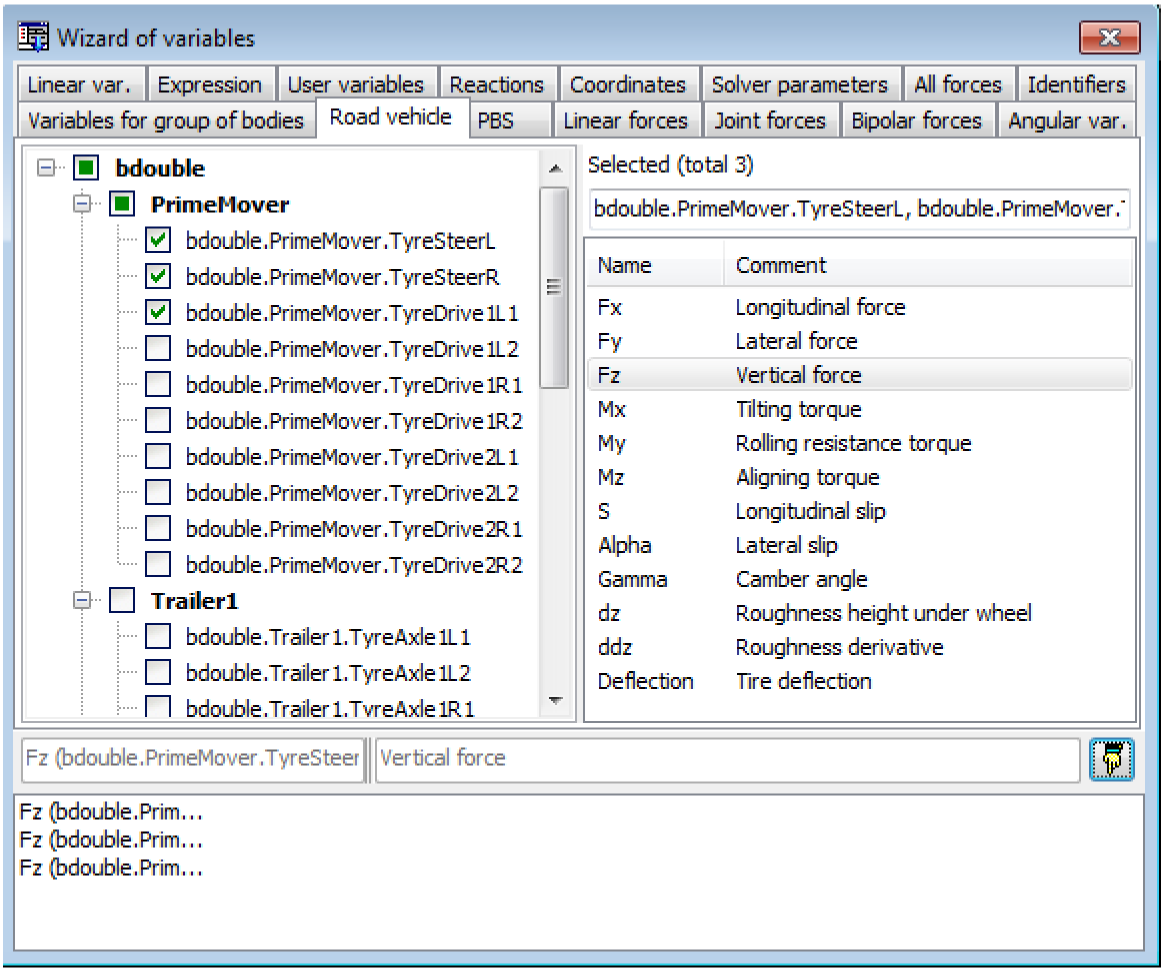



Figure 12.111. Variables related to tire/road interaction

Variables related to the tire/road interaction are available on the **Road Vehicle** tab of the **Wizard of variables**, Figure 12.111. 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.

# 12.10. Input satellite photo as background to animation

## 12.10.1. Creating picture and getting it's sizes

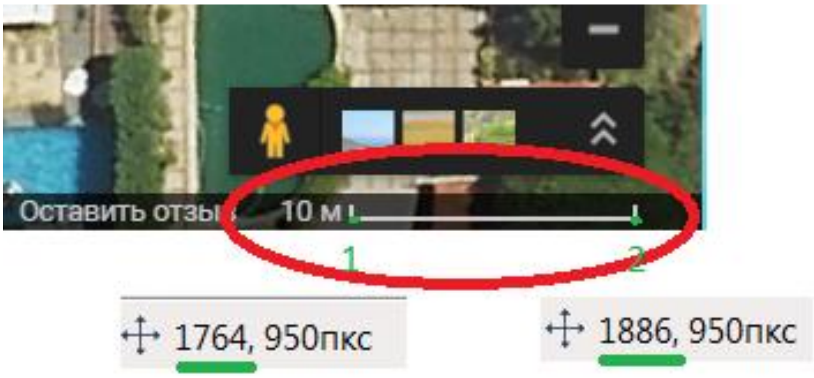
- Choose a place on Google maps


- Rotate view (Ctrl + Mouse move) so that initial vehicle direction will be vertical

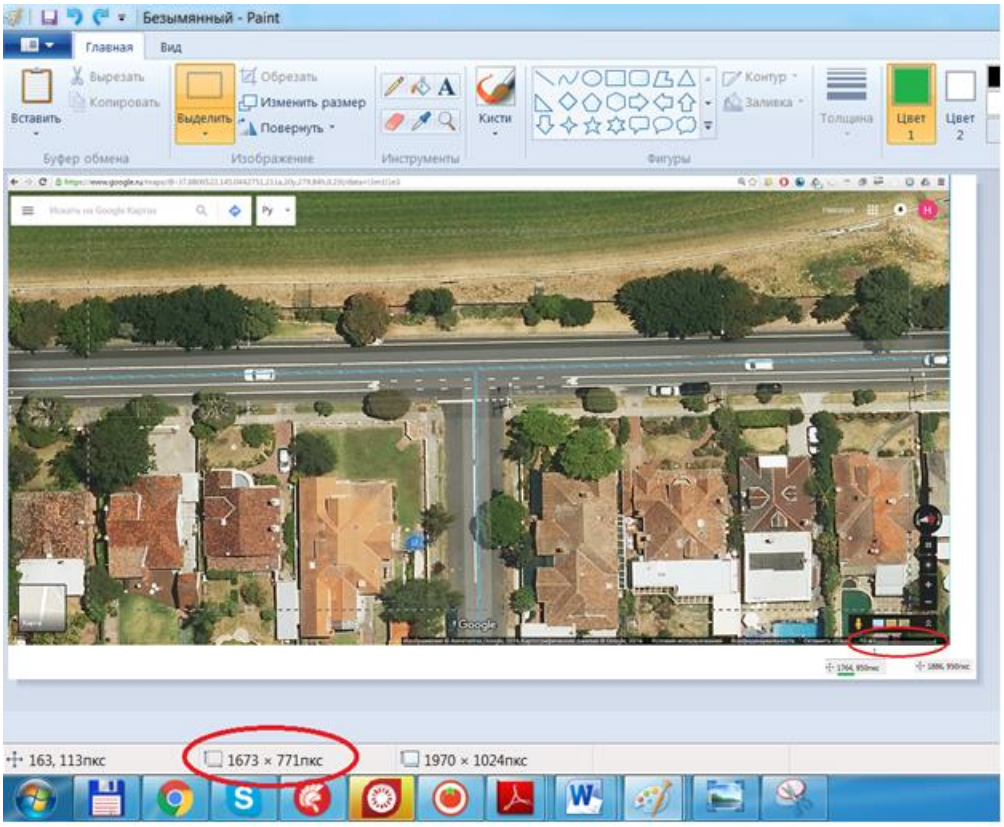

- Make a screenshot of view and paste it into Paint program
- In the right bottom angle of the picture you can see characteristic size of the view.
- Determine pixel length of this size
  - Get pixel coordinate of corners (they are shown in the left bottom corner of the Paint program
  - Get difference 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.



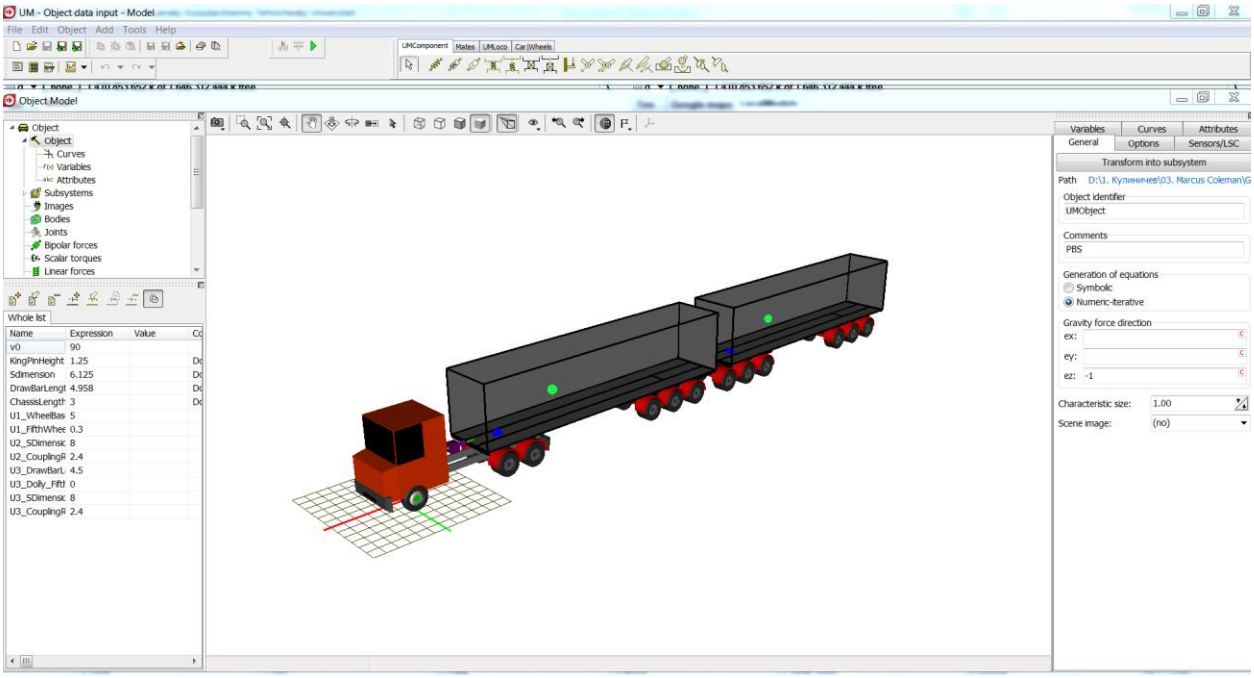

- Calculate real width and height of rectangle

realWidth = pixelWidth/pixelRatio = 1673 [pix] / 12.2 [pix/m] = 137.1 [m]

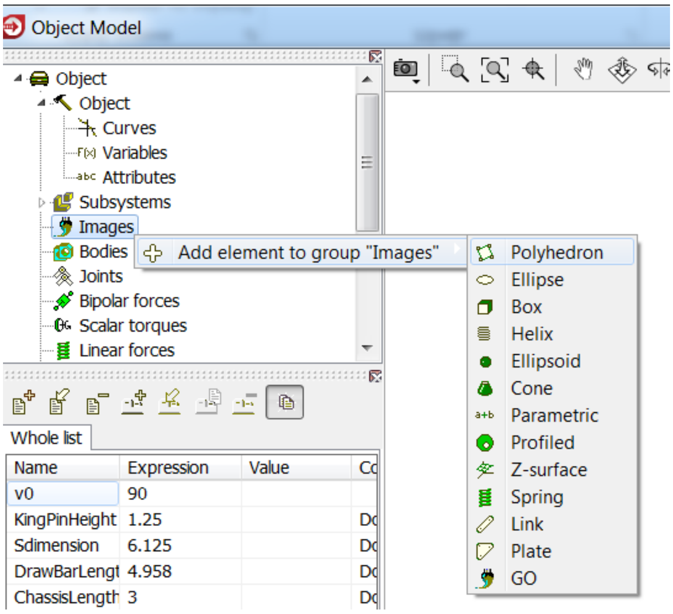
realHeight = pixelHeight/pixelRatio = 771 [pix] / 12.2 [pix/m] = 63.2 [m]

## 12.10.2. Add texture with picture in the UM model

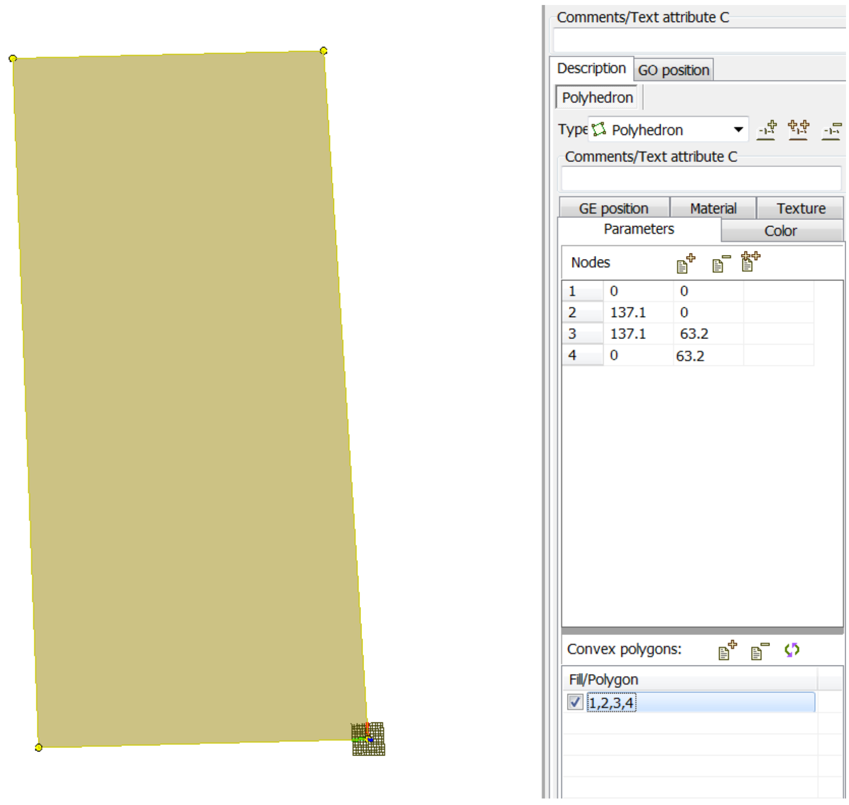
- Open UM model in the **UM Input** program



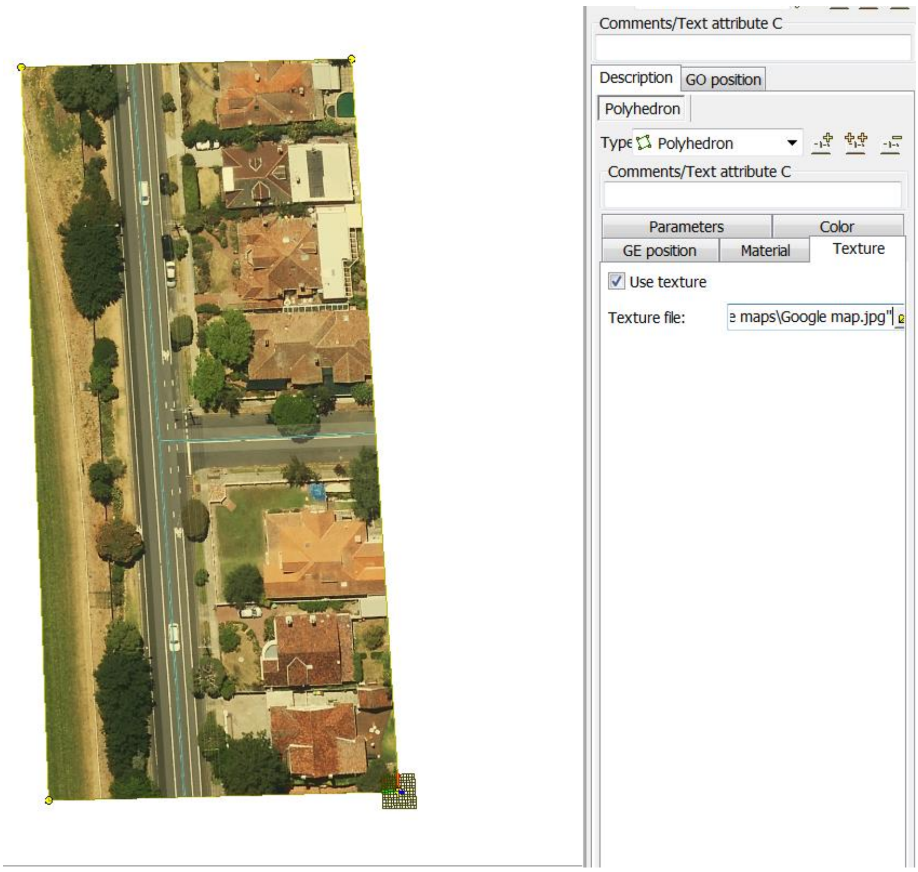

- Create new graphical object - Polyhedron and named it as "Map"



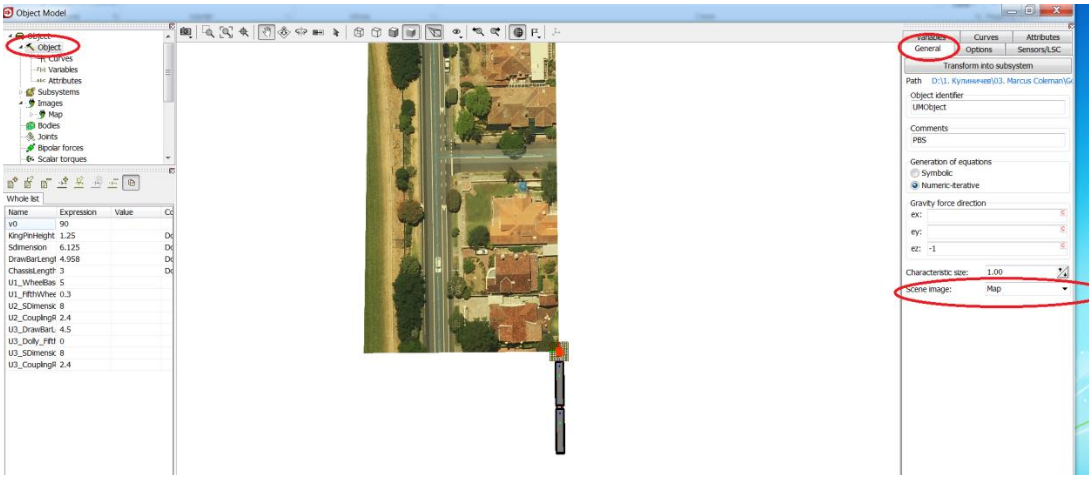

- Create nodes and polygon as shown at the figure



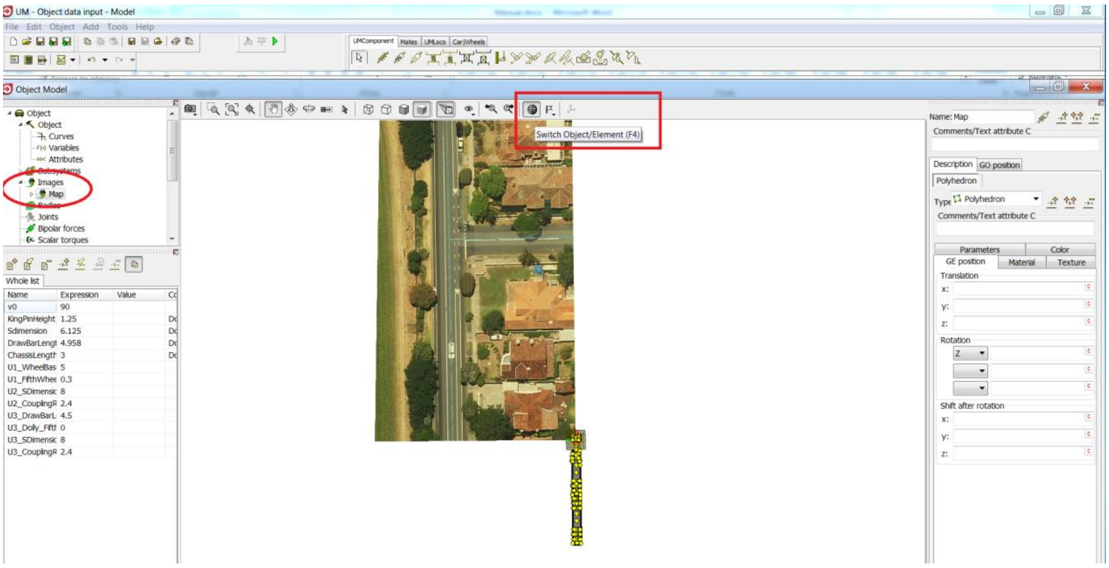

- Use a texture on this polyhedron




- Set graphical object "Map" as Scene Image

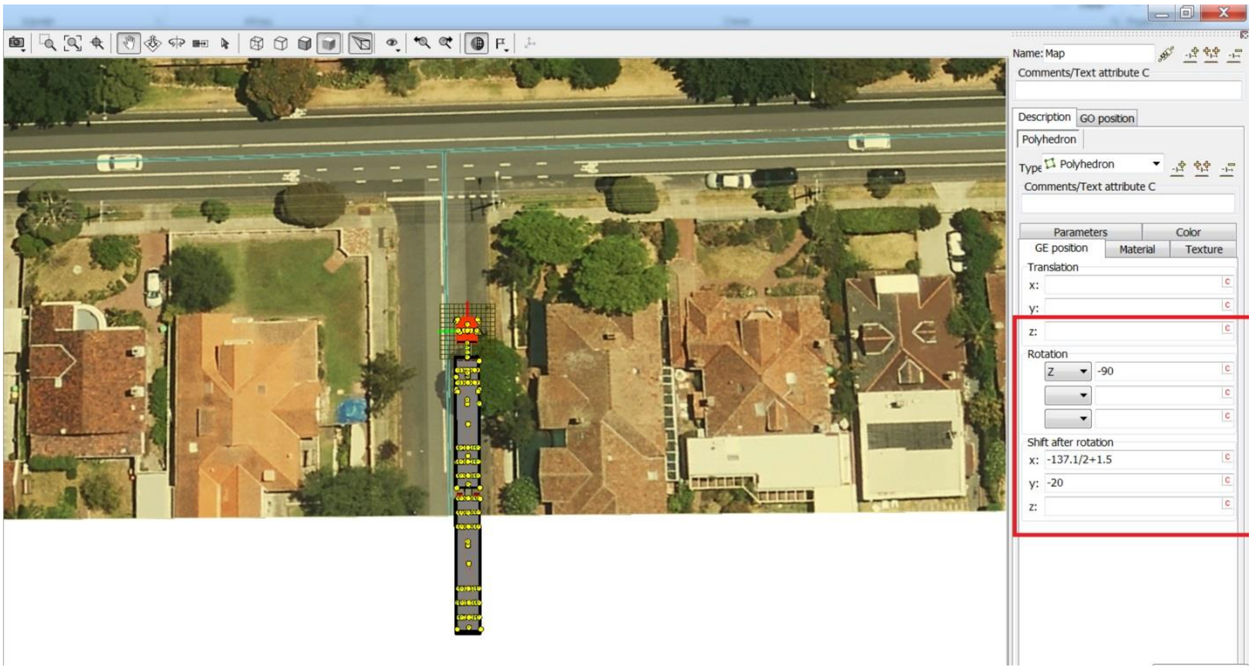


- Choose graphical object "Map" again and enable "Switch Object/Element (F4)" button




- 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.

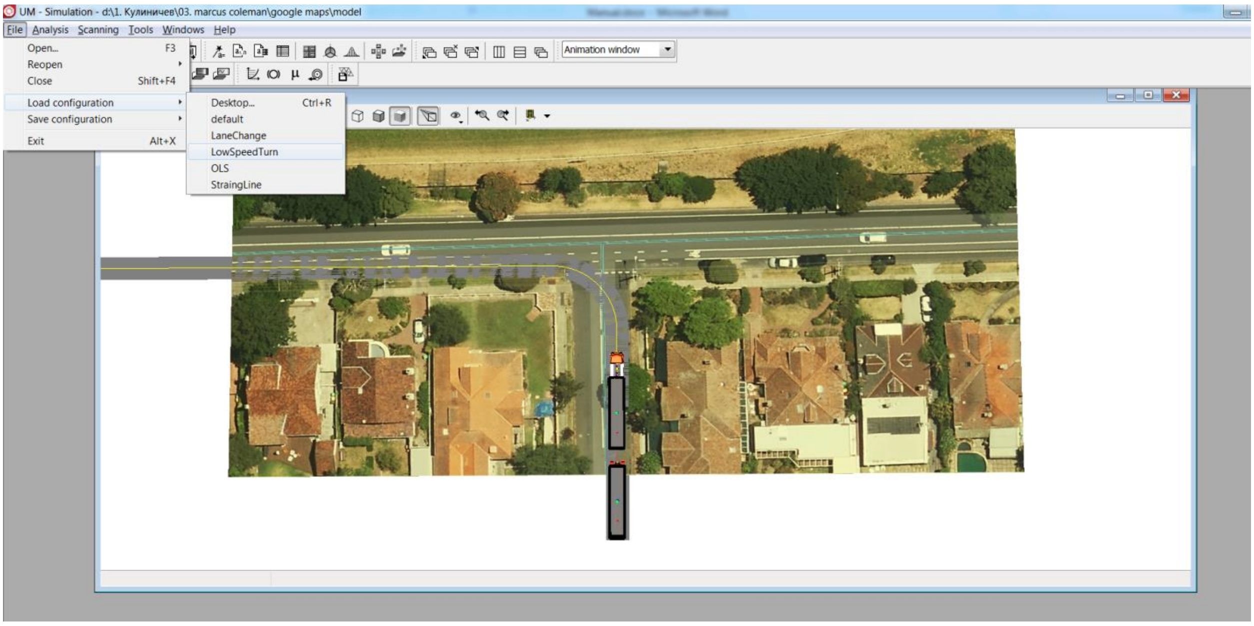



**Note:** As you can see, vehicle doesn't fit in this picture because vehicle is too long (it happens because I chose small region on Google Maps. You should choose larger region)

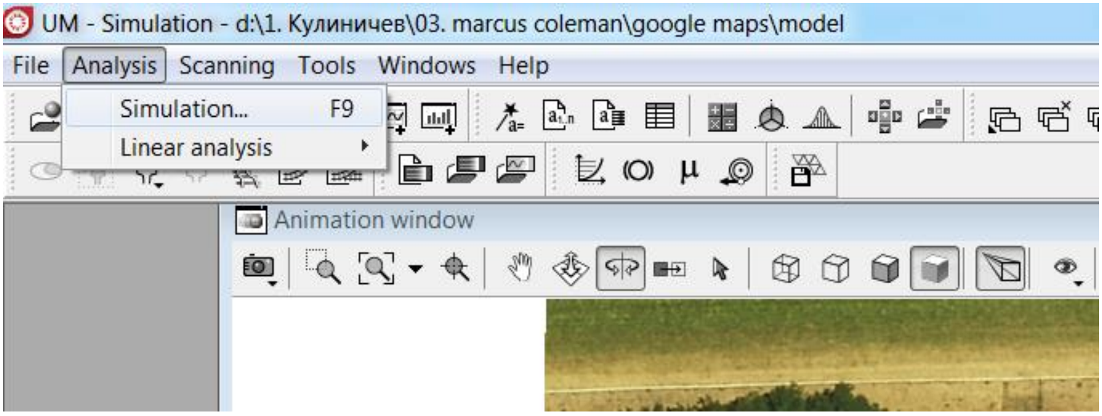
- Save and close model.

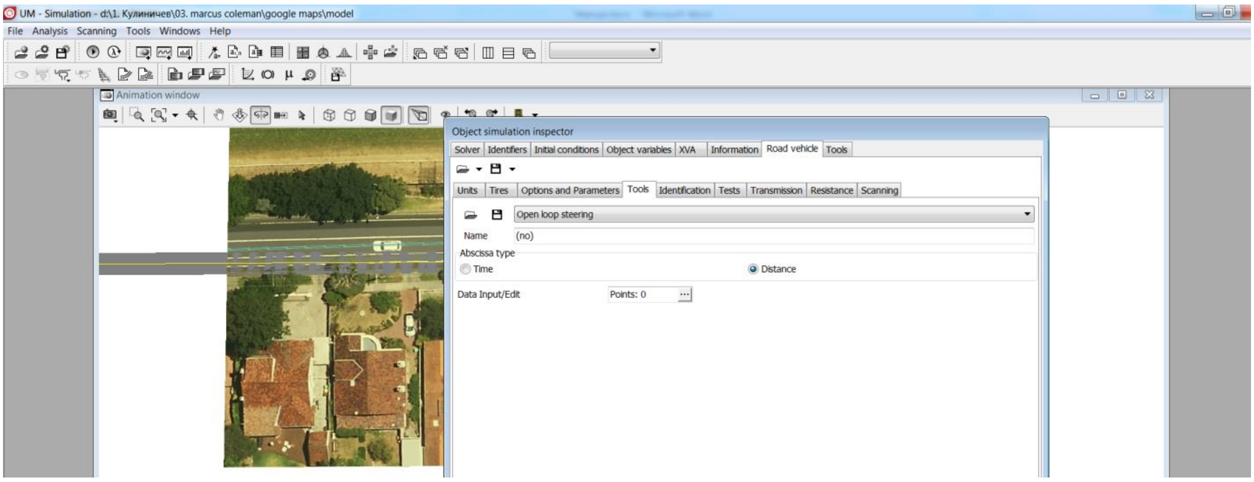
## 12.10.3. Run simulation

- Open model in the **UM Simulation** program and load necessary configuration (for example, LowSpeedTurn)

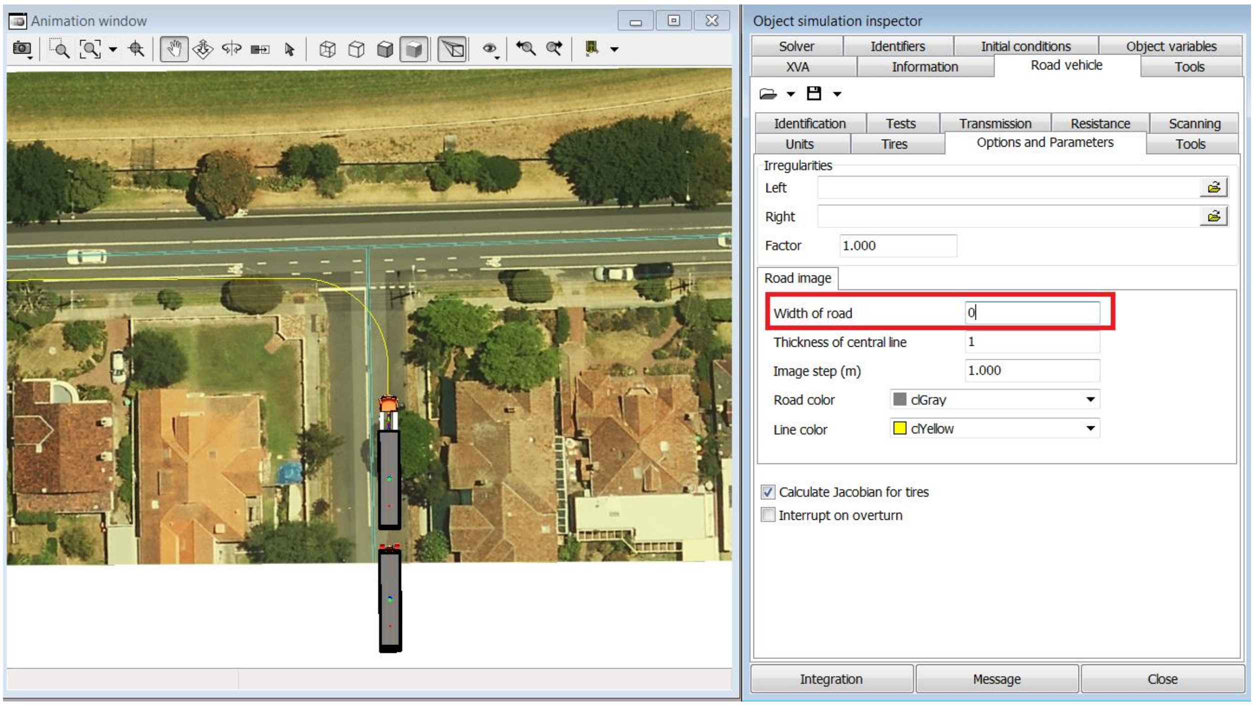



- Call "Object simulation inspector" (Analysis –> Simulation)

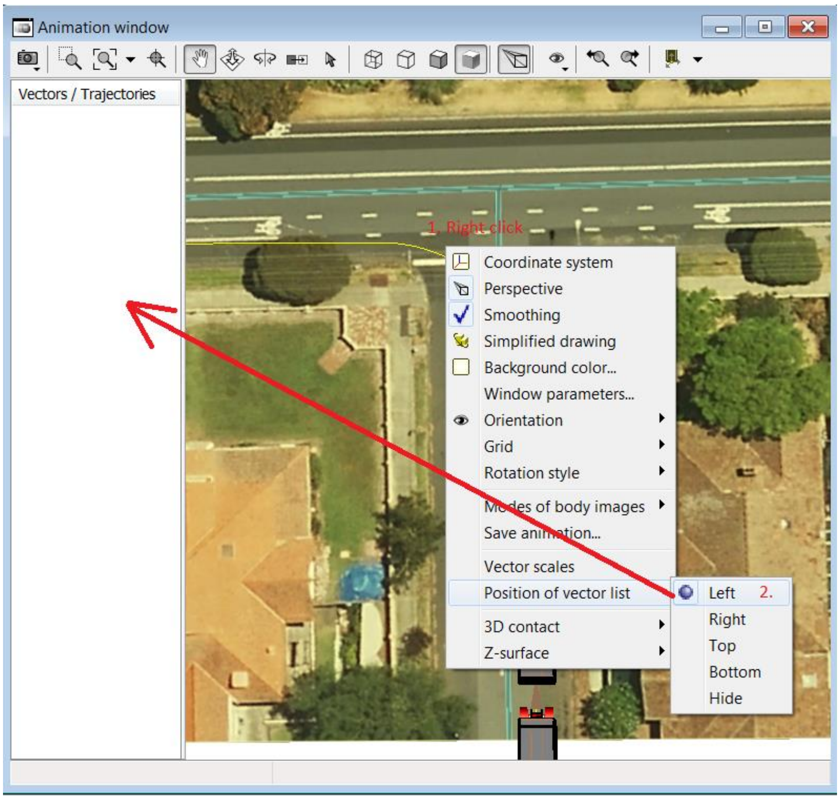


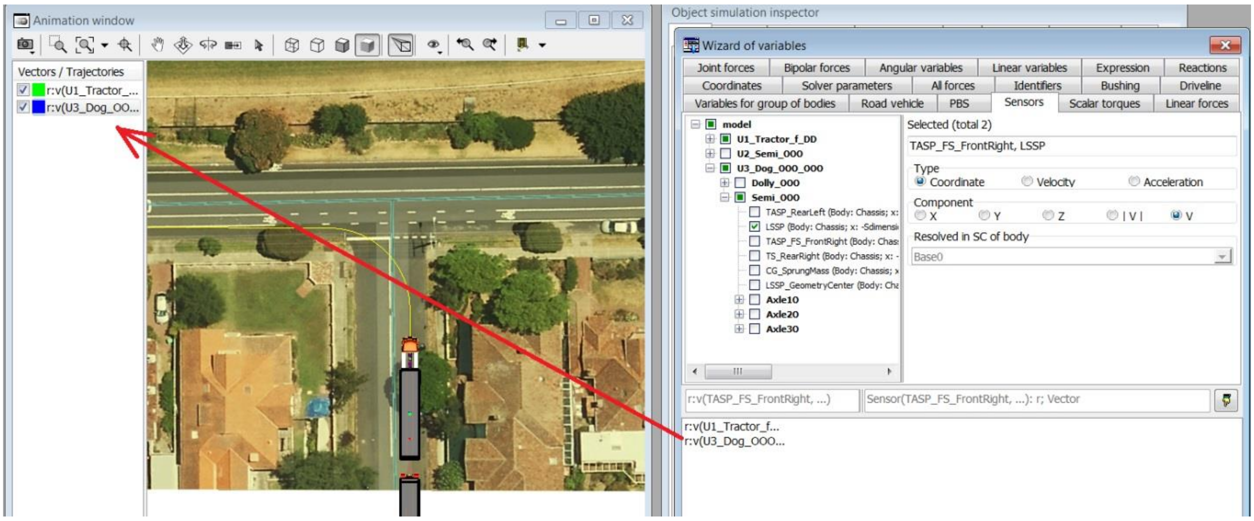




- Remove standard image of road by setting to zero "Width of road" parameter

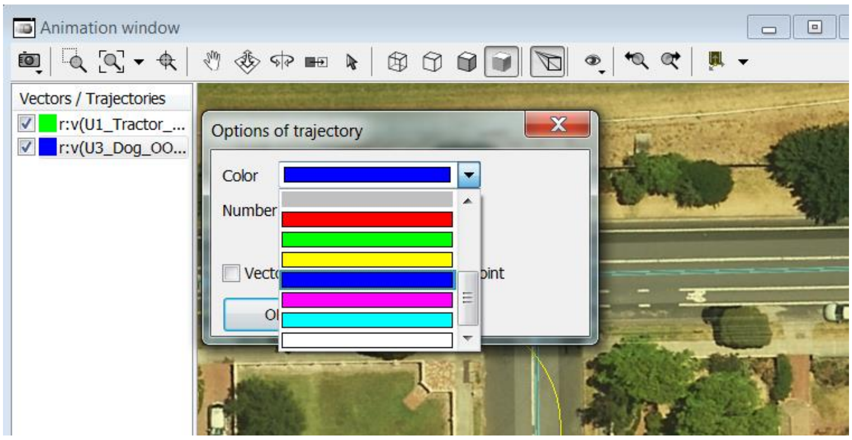



- Create and add trajectory of necessary points to animation window (for example, LSSP sensors):

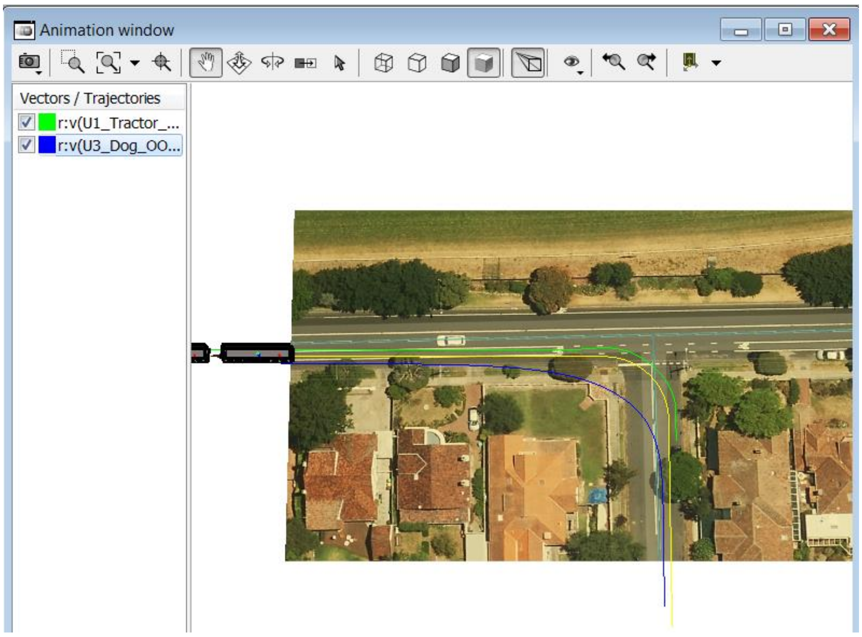




- Change the color of trajectories



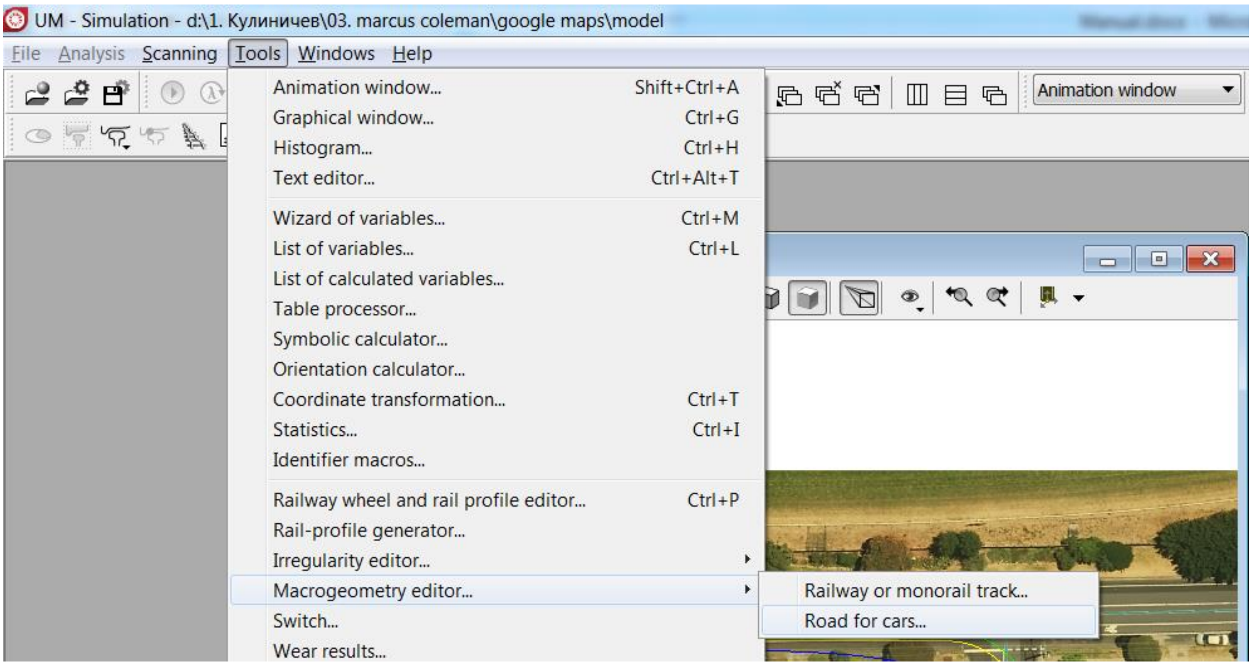

- Run integration



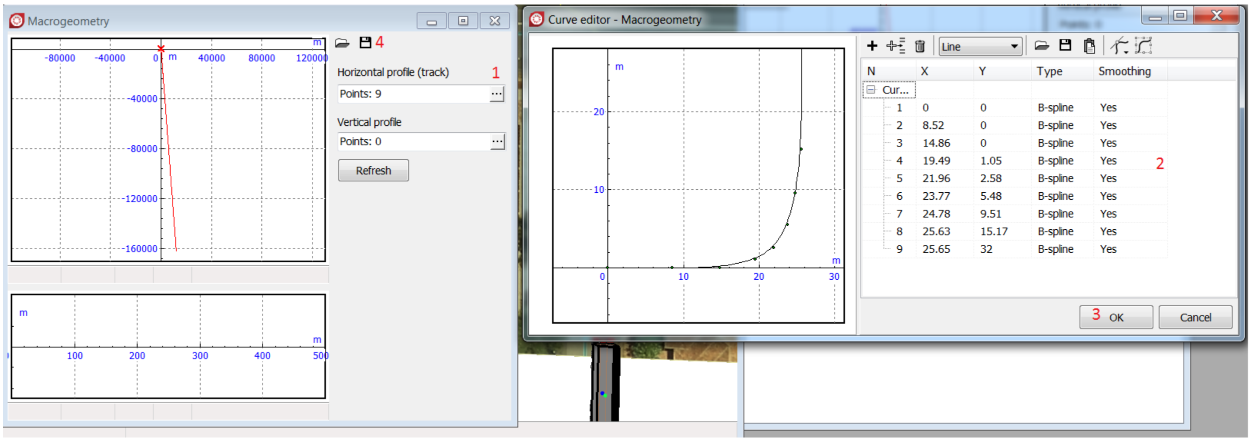
### 12.10.4. Editing macrogeometry

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

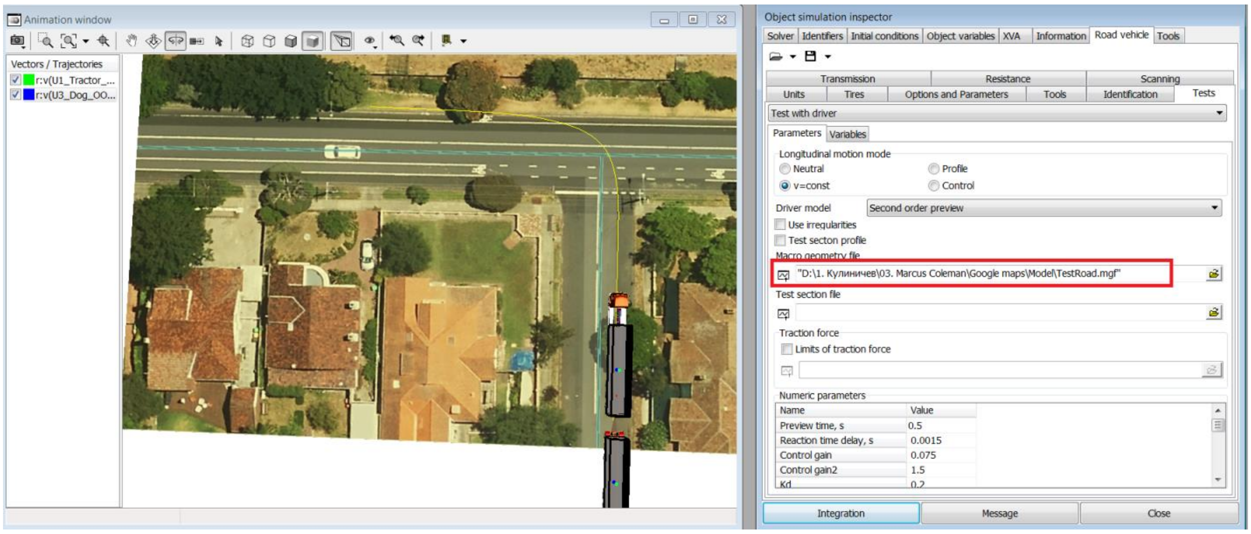
- Open macrogeometry editor



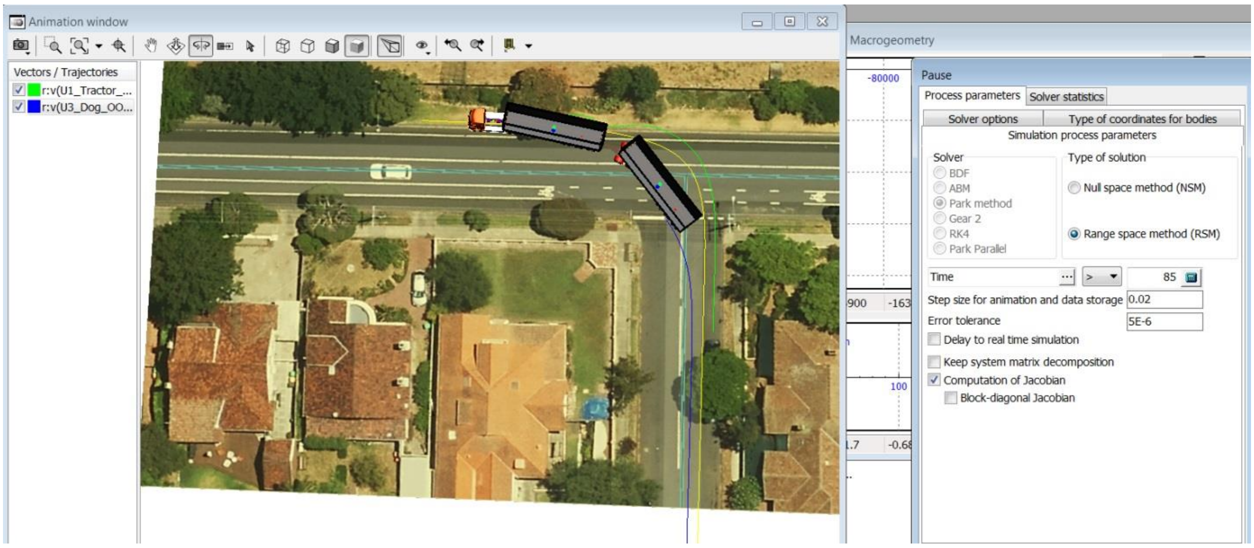

- Create necessary road and save it




- Load this macro in the UM



- Run simulation



# 12.11. Library of Car Suspensions

## 12.11.1. 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\Suspensions 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. 12.11.2.1, page 12-112;
- double wishbone suspension, see. Sect. 12.11.2.2, page 12-113;
- semi-trailing arm suspension, see Sect. 12.11.2.3, page 12-114;
- McPherson suspension, see Sect. 12.11.2.4, page 12-115;
- torsion suspension, see Sect. 12.11.2.5, page 12-116;
- multi-link suspension, see Sect. 12.11.2.6, page 12-117.

| | |
|---|---|
| **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. |

## 12.11.2. Brief description

### 12.11.2.1. 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.

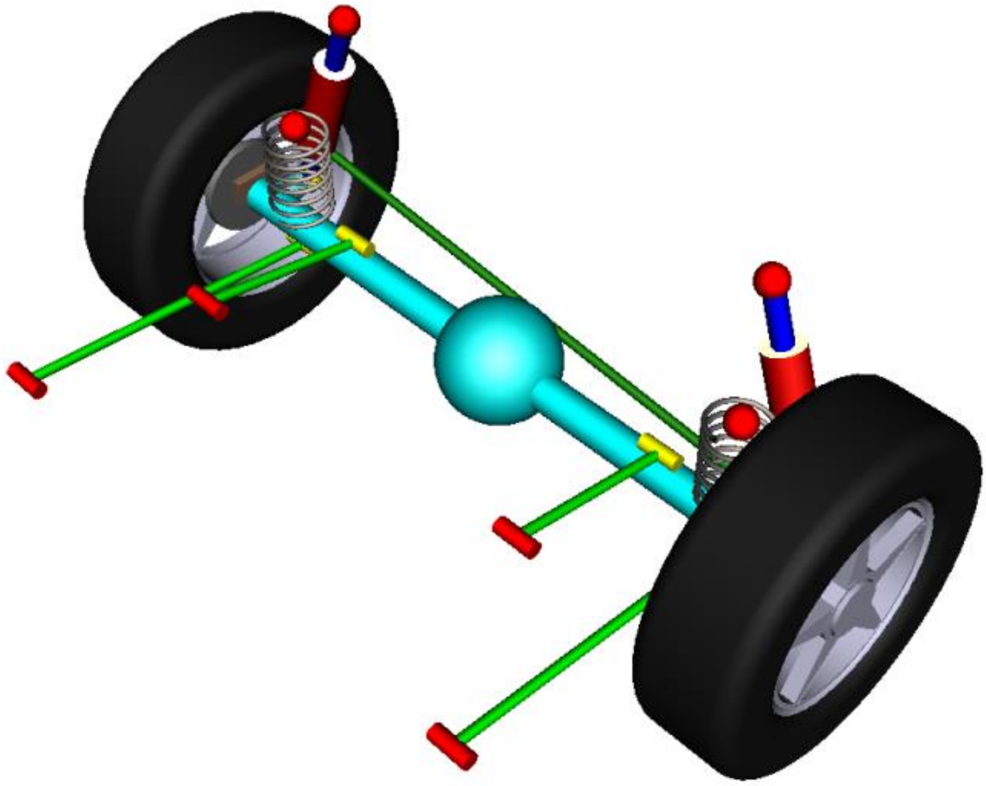


Figure 12.112. Axle suspension with guided arms

### 12.11.2.2. 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.

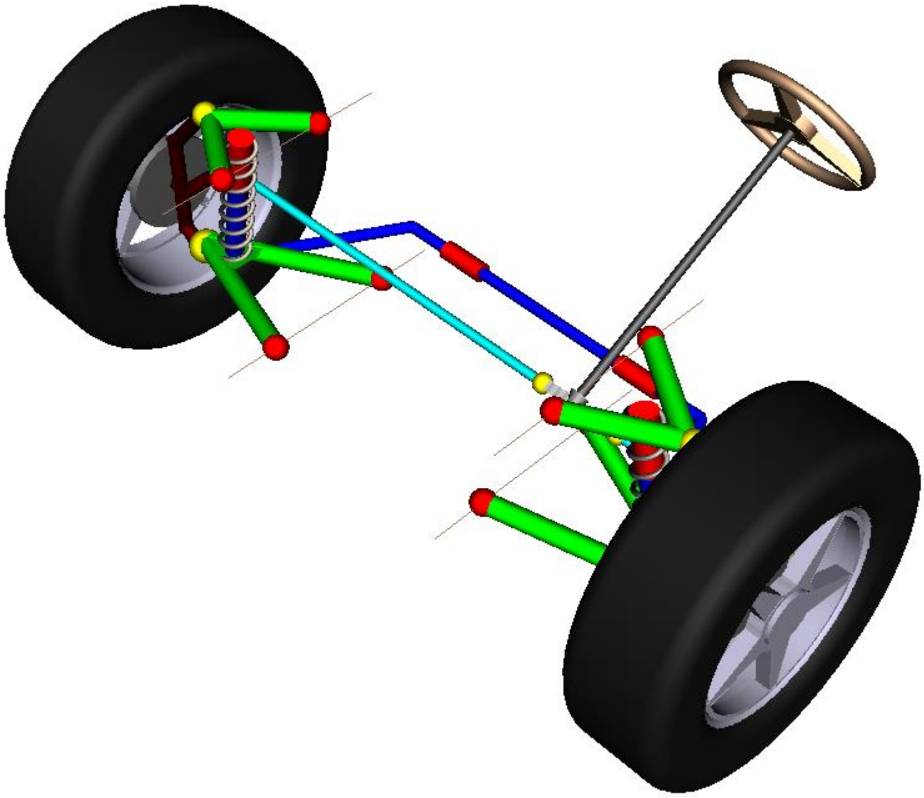


Figure 12.113. Double wishbone suspension

### 12.11.2.3. 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.

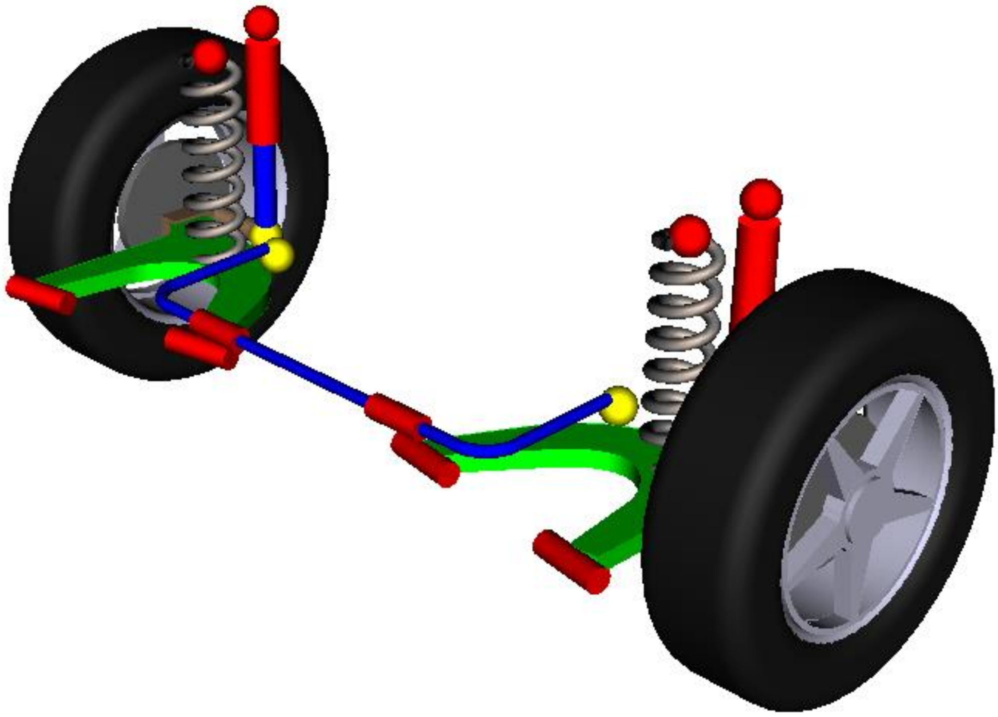


Figure 12.114. Semi-Trailing Arm Suspension

#### 12.11.2.4. 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.

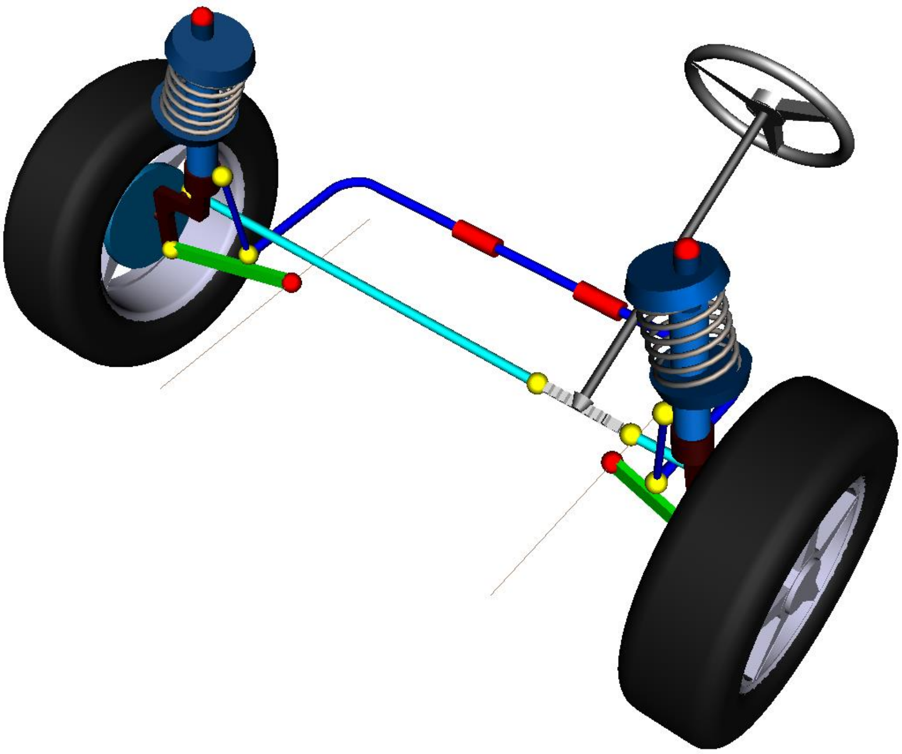


Figure 12.115. MacPherson Suspension

#### 12.11.2.5. 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.

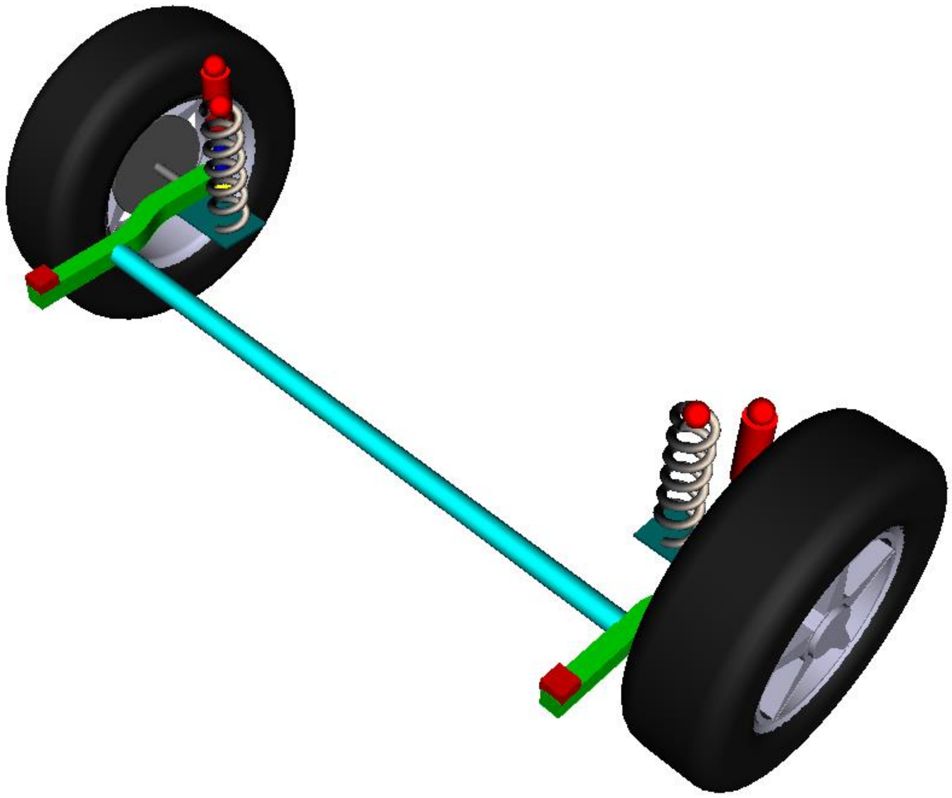


Figure 12.116. Torsion Suspension

### 12.11.2.6. 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 [12].

Model folder: {UM Data}\SAMPLES\Automotive\Suspensions\Multilink suspension.

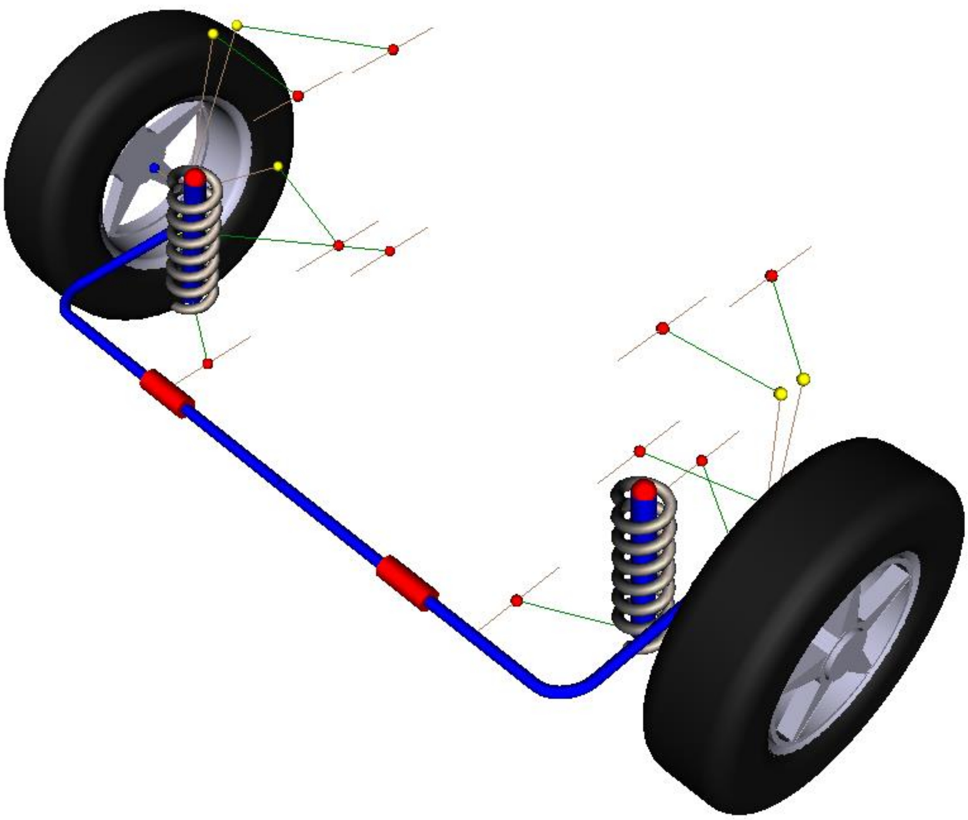


Figure 12.117. Multi-link Suspension

## 12.11.3. 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.

### 12.11.3.1. 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 12.118.

**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. |

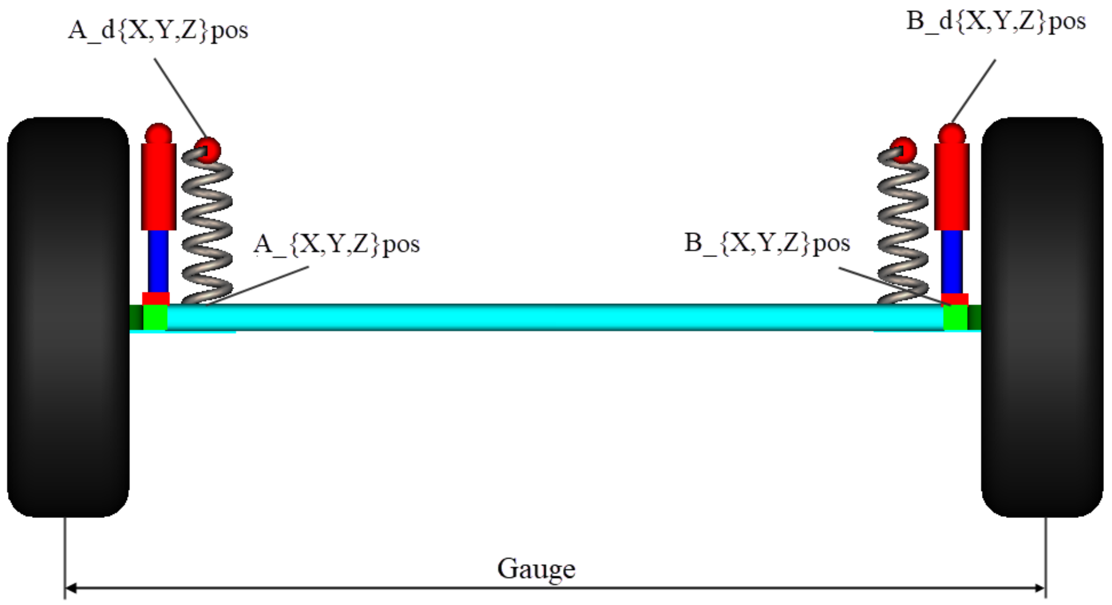



Figure 12.118. Parameterization of gauge and attachment points of force elements

#### 12.11.3.2. Parameterization of Wheels

Let us consider basic parameters that express wheel geometry and camber and toe angles, see. Figure 12.119 -.

**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.

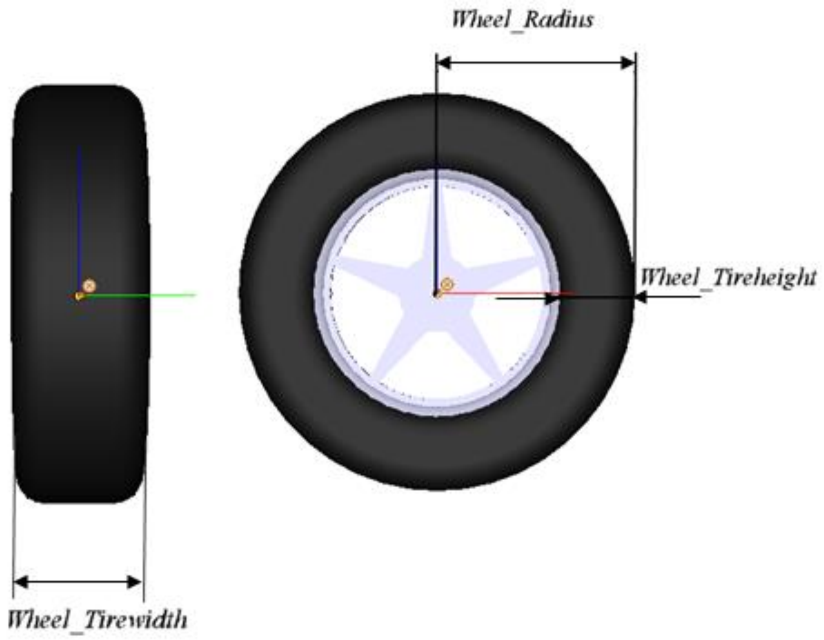



Figure 12.119. Basic geometrical parameters of the wheels

Positive toe, or toe in, is the front of the wheel pointing towards the centerline of the vehicle, see Figure 12.120. 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 12.121.

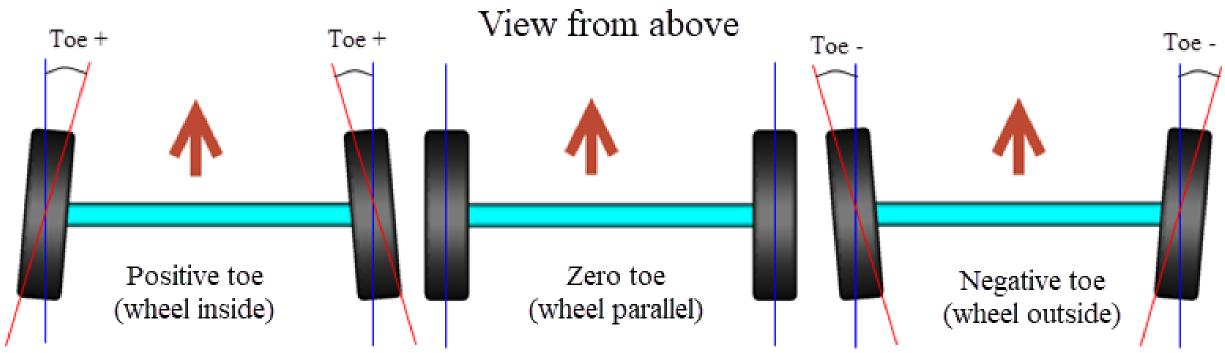



Figure 12.120. Toe angle

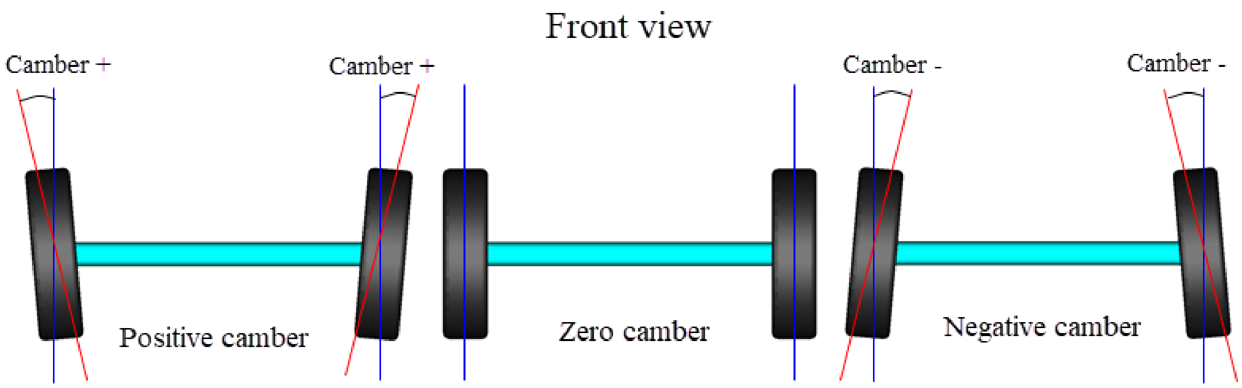



Figure 12.121. Camber angle

### 12.11.3.3. 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 12.122:

- **SteeringColumnLength** is the length of the steering column, m;
- **SteeringColumAngle** is the inclination angle in degrees described in Figure 12.123.

The inclination angle is parameterized and introduced in the **jSteering Column** generalized joint, see **RTy** elementary transformation, see Figure 12.123. By default 30º angle is used. Please note that the identifier **SteeringColumnAngleRad** showed in Figure 12.123 is expressed in radians and is dependent from the **SteeringColumnAngle**, given in degrees for easier parameterization.

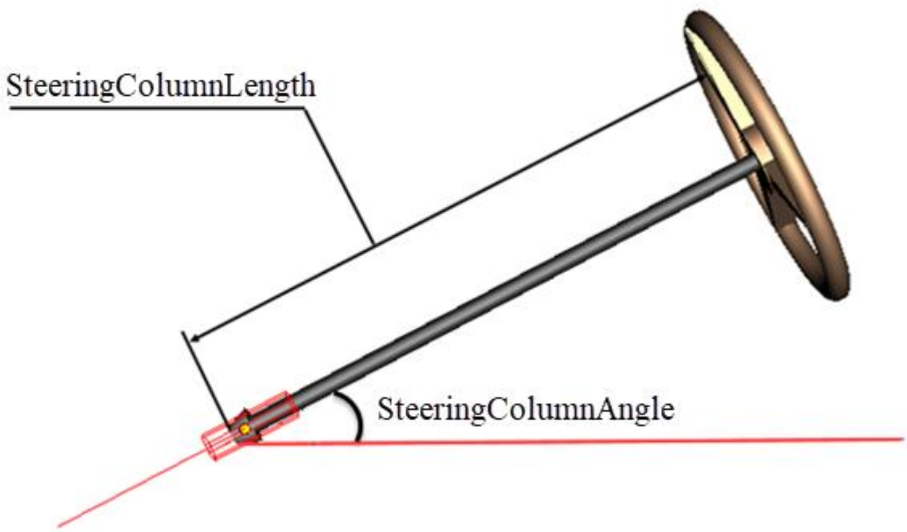



Figure 12.122. Geometric parameters of the steering column

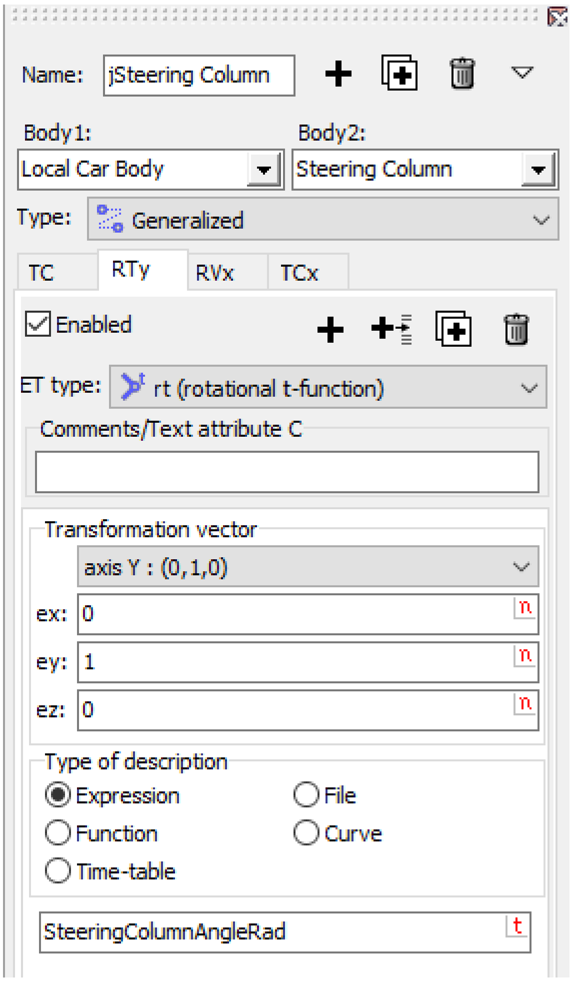



Figure 12.123. Setting the steering departure angle

#### 12.11.3.4. 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.

#### 12.11.3.5. 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$\cdot m^2$;

**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*.

## 12.11.4. Creating a Car Model Using Suspensions from Libraries

### 12.11.4.1. 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. 12.11.2.2), and the rear suspension is the axle suspension (12.11.2.1), see Figure 12.124. 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 12.124. Lada 4×4 and its model in **UM Input**

#### 12.11.4.1.1. 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 12.125):
   - **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 12.125.

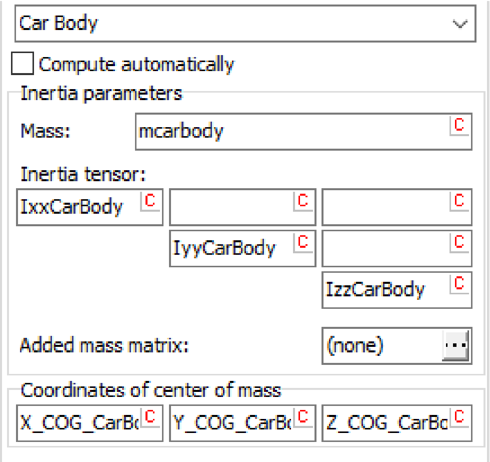



Figure 12.125. Inertial parameters of the body

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 12.126.

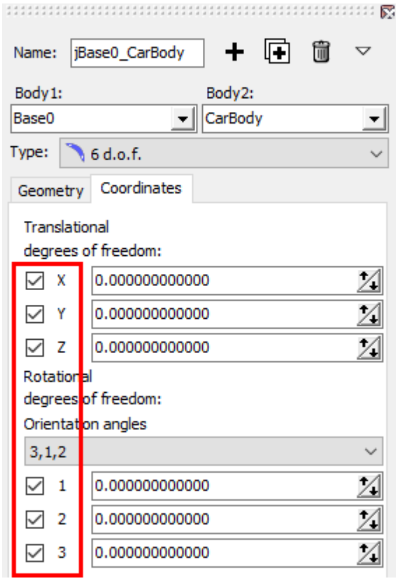



Figure 12.126. Creating a joint for the car body

### 12.11.4.1.2. 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. 12.11.2.2, page 12-113), 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 12.127, left. In fact, this step is optional. You can leave default value in the **Identifier** field.

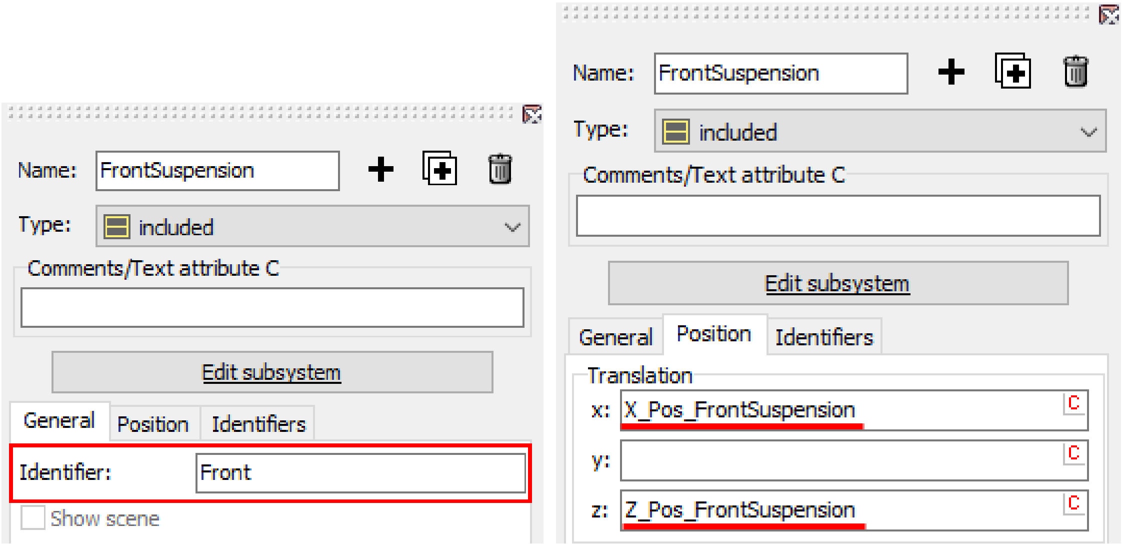



Figure 12.127. Identifier for the front suspension (optional)

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 12.127, 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 12.128.

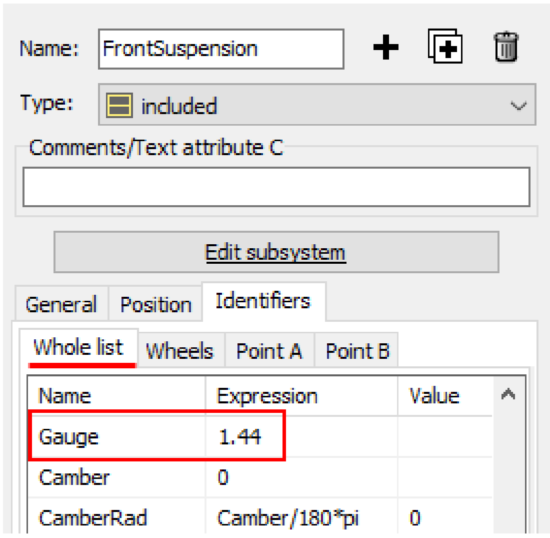



Figure 12.128. Gauge of the front suspension (1.44 m)

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 12.129.

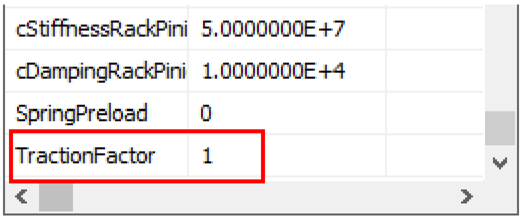



Figure 12.129. **TractionFactor** identifier for the front suspension

6. In the same way add the **Axle Suspension**, see Sect. 12.11.2.1, page 12-112. Set its name to **RearSuspension**. Set **Identifier** field on the **General** tab to **Rear** (Figure 12.130, 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 12.130, right.

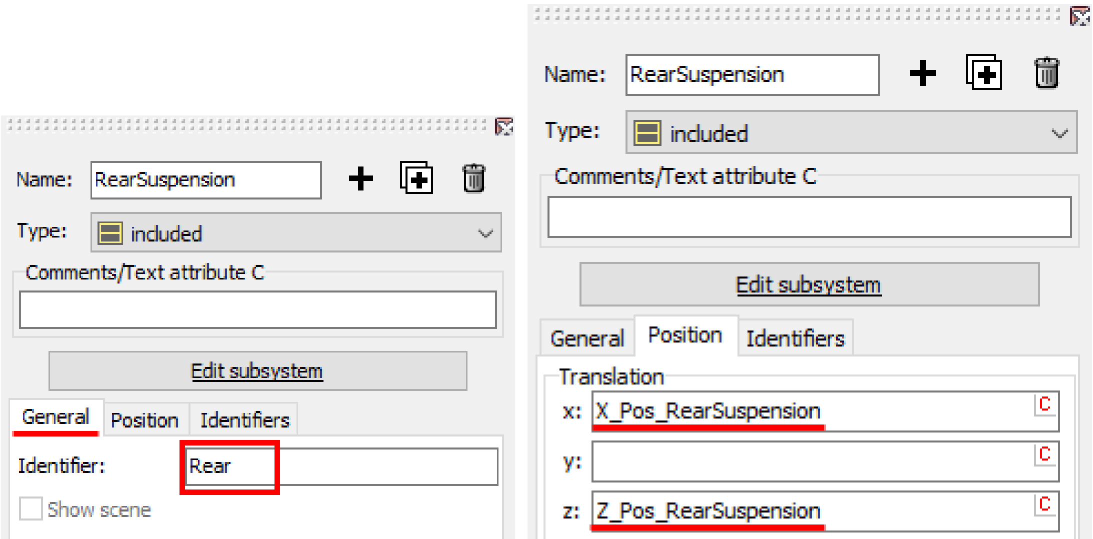



Figure 12.130. Identifier and position for the rear suspension

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 12.131. 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.

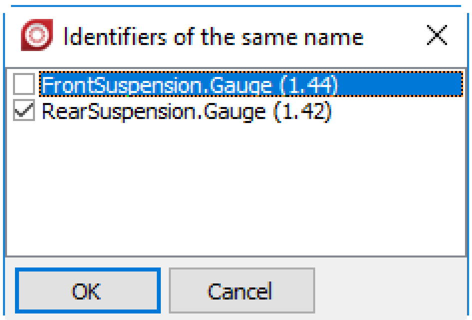



Figure 12.131. Identifiers of the same name for different subsystems

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**.

### 12.11.4.1.3. 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 12.132.

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 12.132. Via this joint the intermediate **Local Car Body** of the front suspension is rigidly connected to the car body.

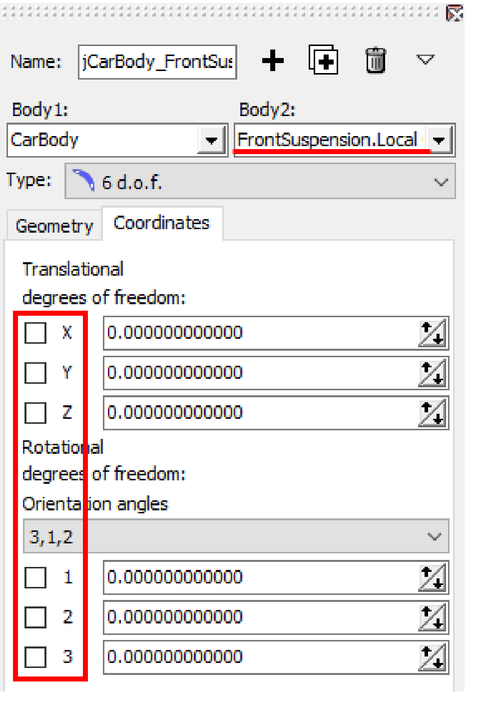



Figure 12.132. Creating a joint for the front suspension

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 12.133, 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 12.133, right.

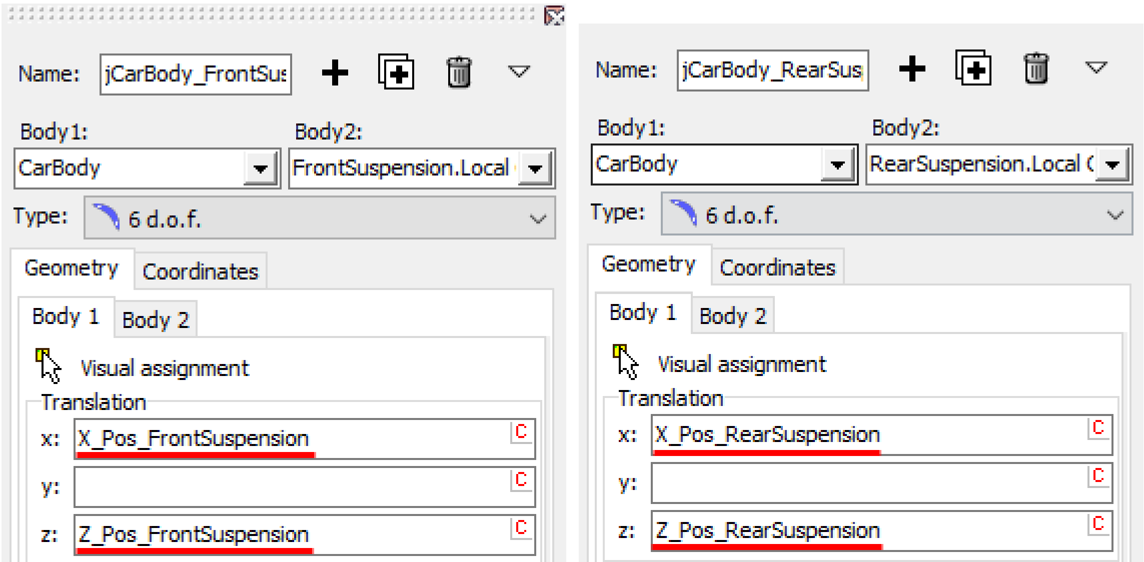



Figure 12.133. Joint points for the front (in the left) and rear (in the right) suspension

The model is ready. Finally your model should look like one depicted in Figure 12.134. Go to Summary node in the tree of elements and check that your model has no errors.

Figure 12.134. Newly created model of the car in UM Input

### 12.11.4.2. 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. 12.9.1. "*Preparing for simulation*", page 12-74.

#### 12.11.4.2.1. 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 12.135.

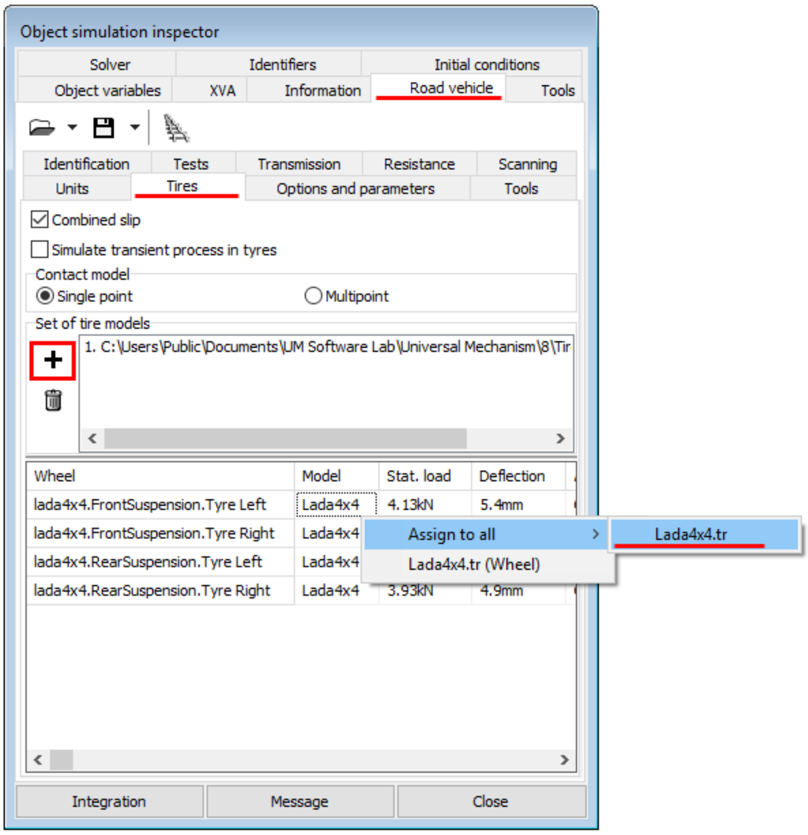



Figure 12.135. Assigning tire model for the vehicle

#### 12.11.4.2.2. 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 12.136-. 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 12.138. 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. 12.9.1.2. "*Identification of steering*", page 12-77.

**Note** The **Steer ratio** parameter is set automatically as a result of the steering wheel rotation test, see Sect. 12.11.4.2.5, "*Steering Wheel Rotation Test*", page 12-136. It should not be set manually on this step.

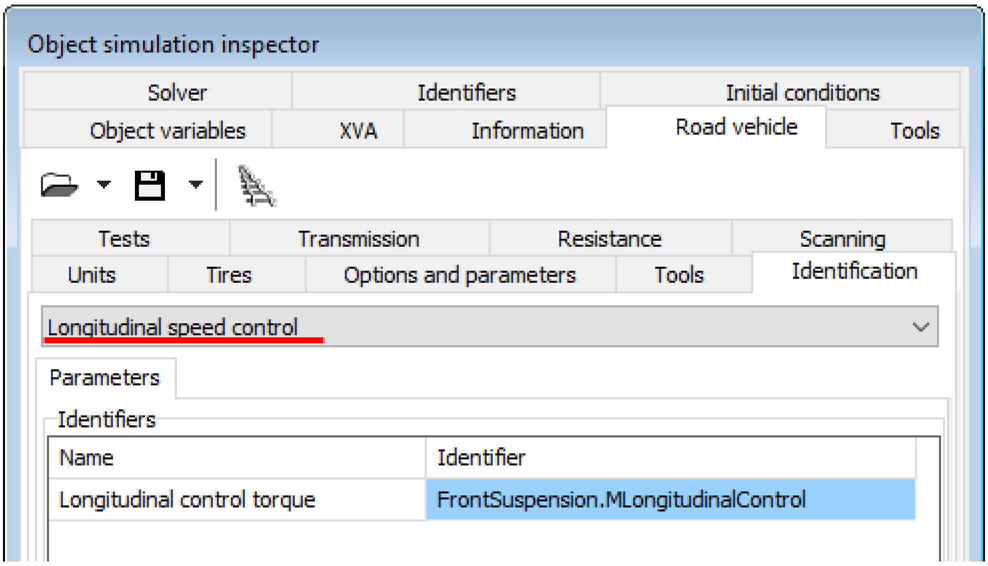



Figure 12.136. Identification of the longitudinal speed control

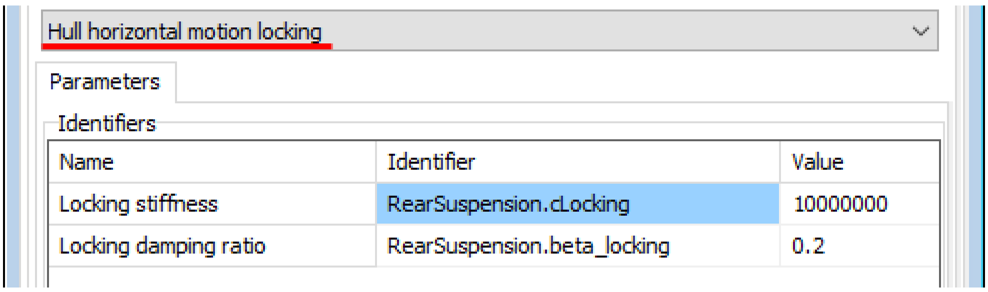



Figure 12.137. Identification of the horizontal motion locking

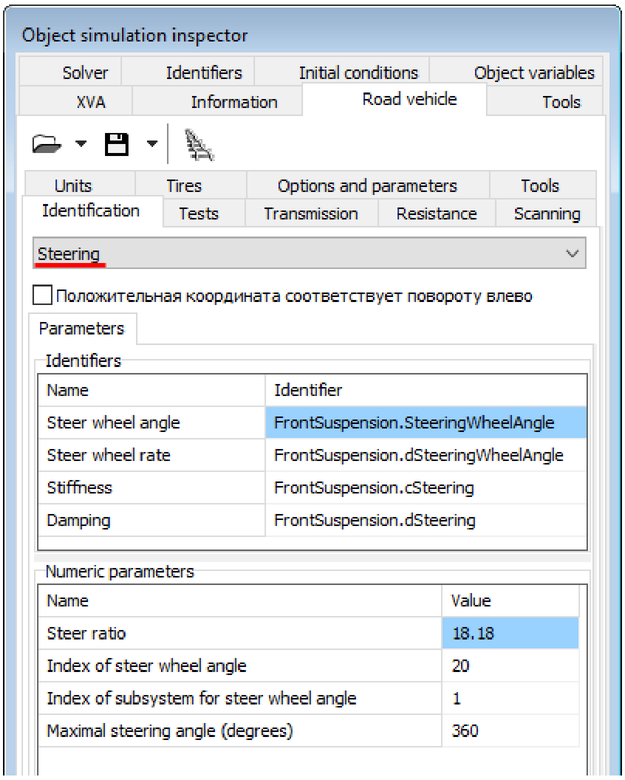



Figure 12.138. Identification of the steering control

#### 12.11.4.2.3. 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 12.139. Leave the rest settings by default.

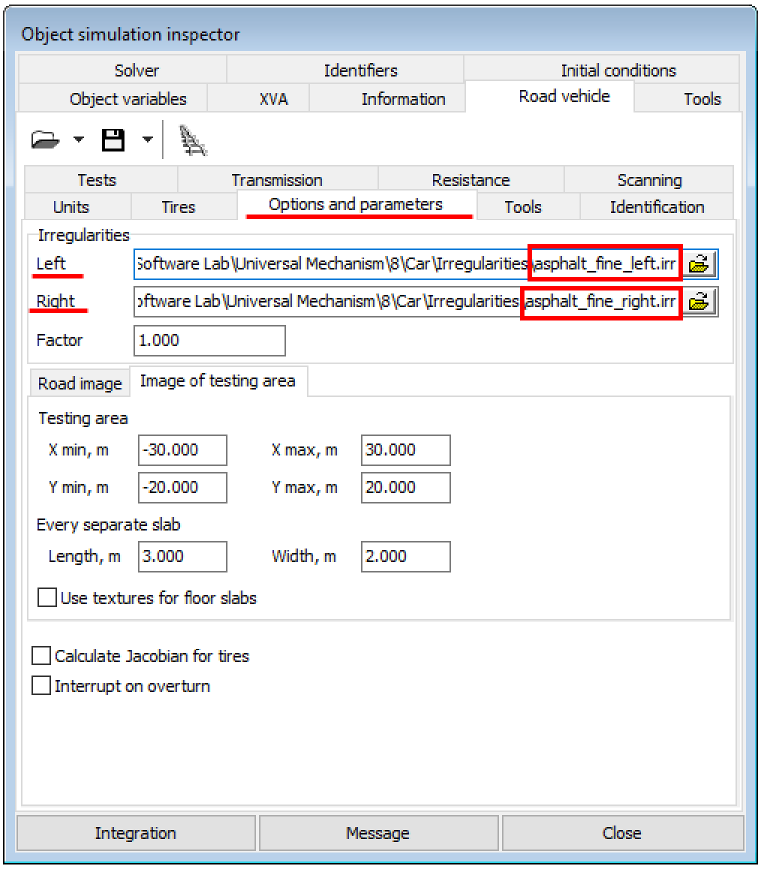



Figure 12.139. Irregularities setting

#### 12.11.4.2.4. 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 12.140. 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).

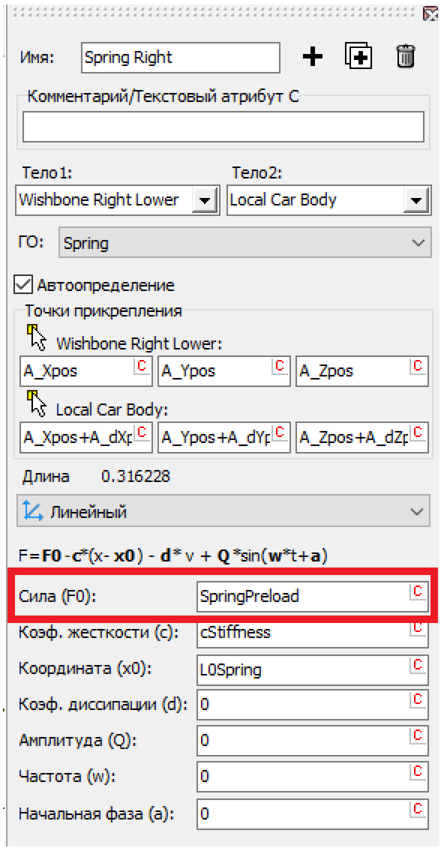



Figure 12.140. Spring preload

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 12.141. Create new variables and drag&drop them into the new graphical window.

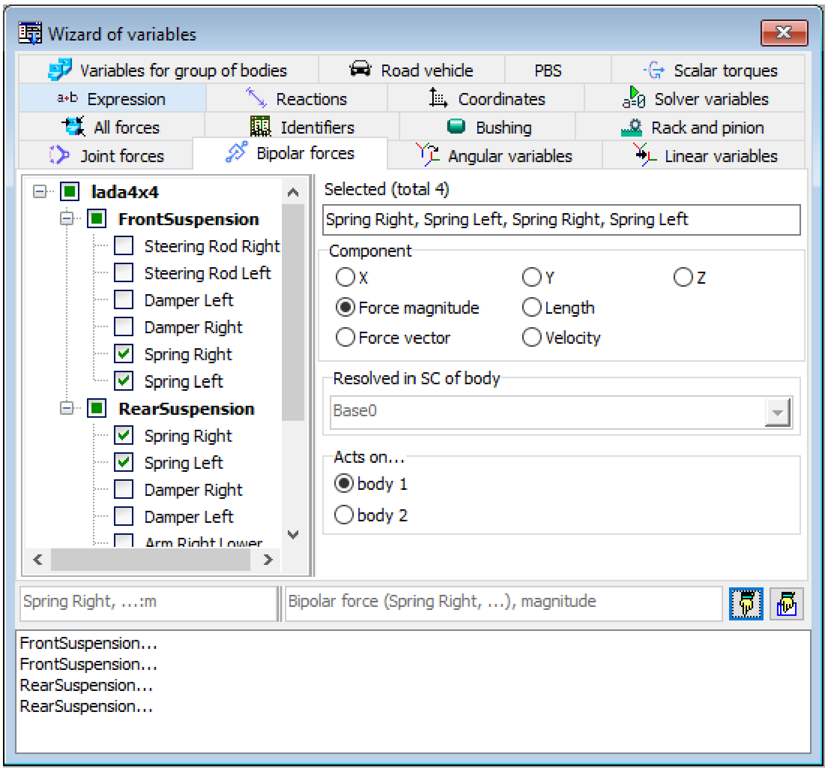



Figure 12.141. Force magnitude for springs

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 12.142. 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.

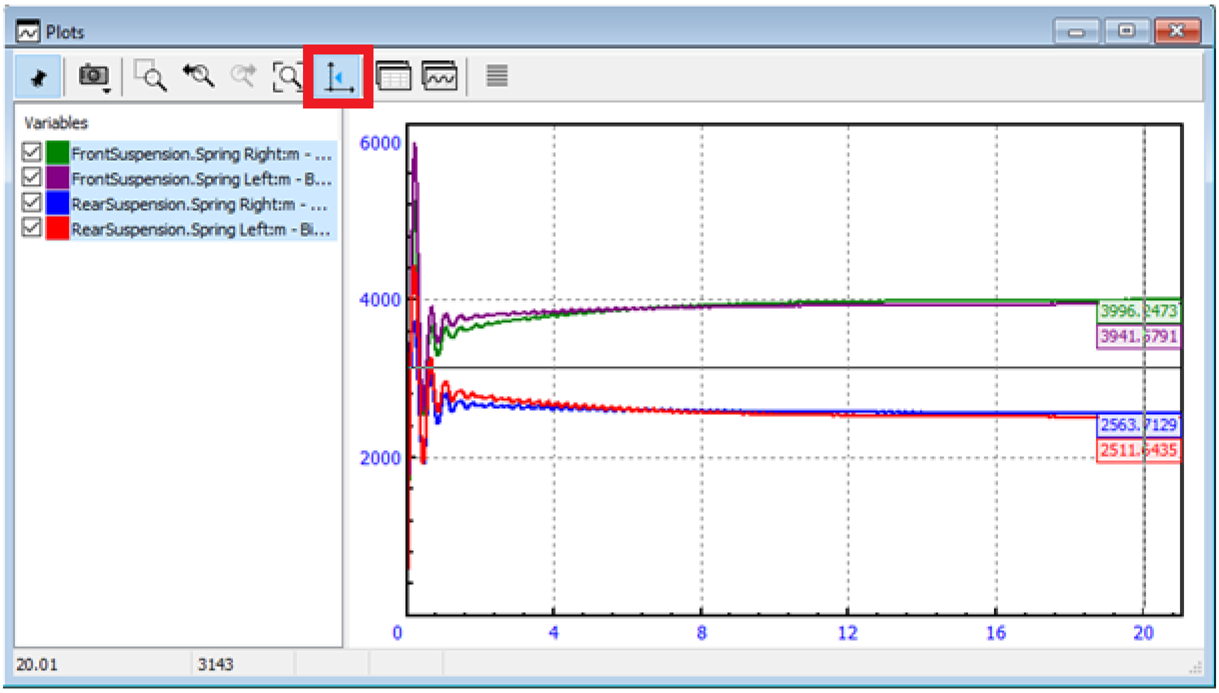



Figure 12.142. Spring forces at equilibrium position

6. In the **Object simulation inspector** select the **Identifiers** tab sheet. In the drop-down list select the **lada4x4.FrontSuspension** subsystem, see Figure 12.143. 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.

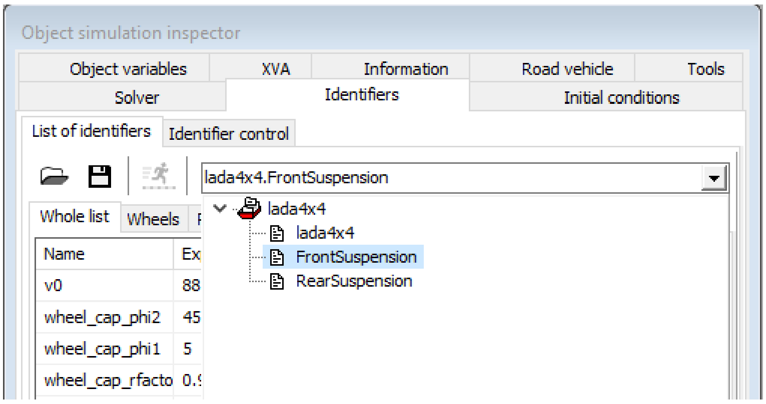



Figure 12.143. Select the subsystem

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 12.144.

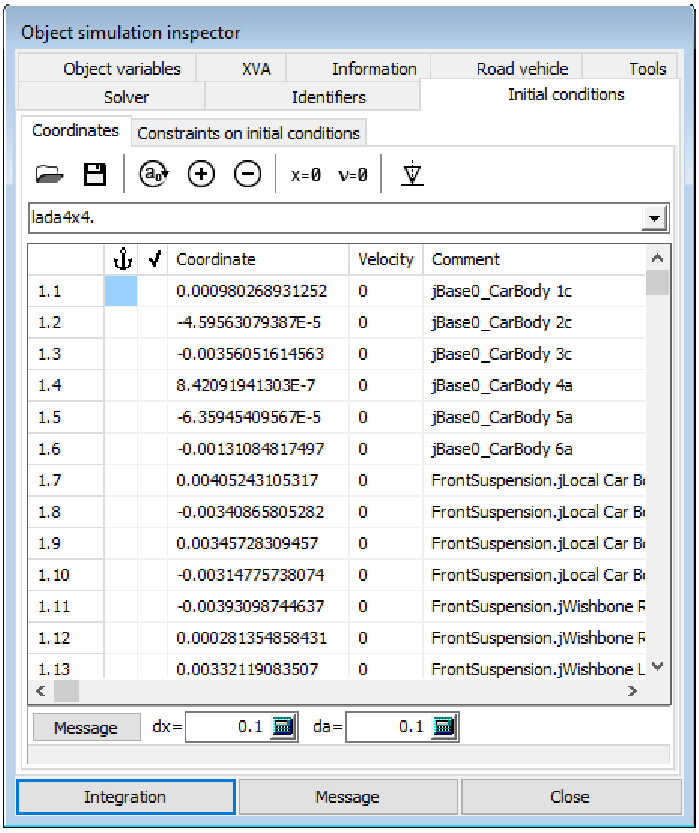



Figure 12.144. Equilibrium position

#### 12.11.4.2.5. 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. 12.9.2.5. "*Steering wheel rotation test*", page 12-89.

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 12.145, and set **Invisible** mode for the car body, see Figure 12.146.

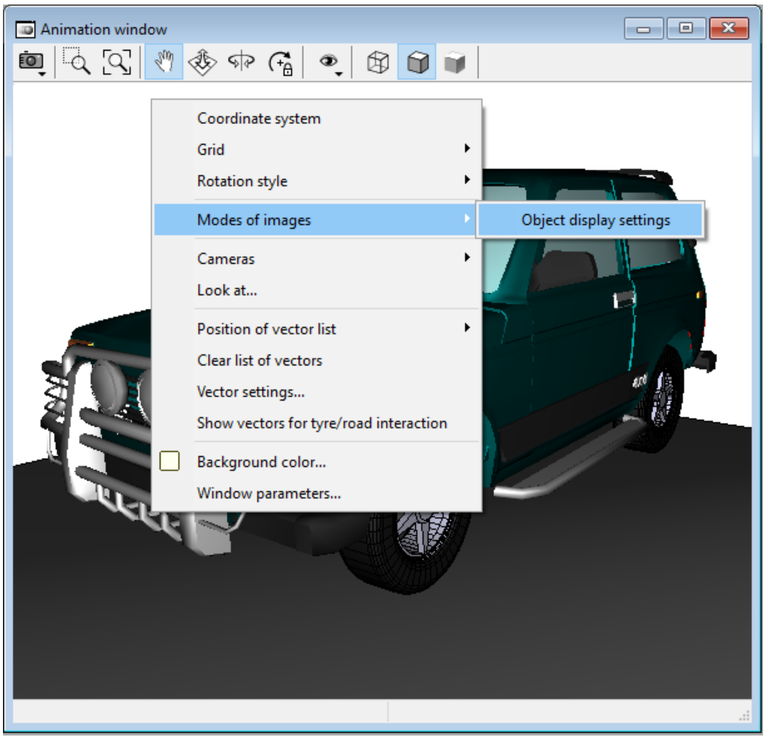



Figure 12.145. Modes of images | Object display settings

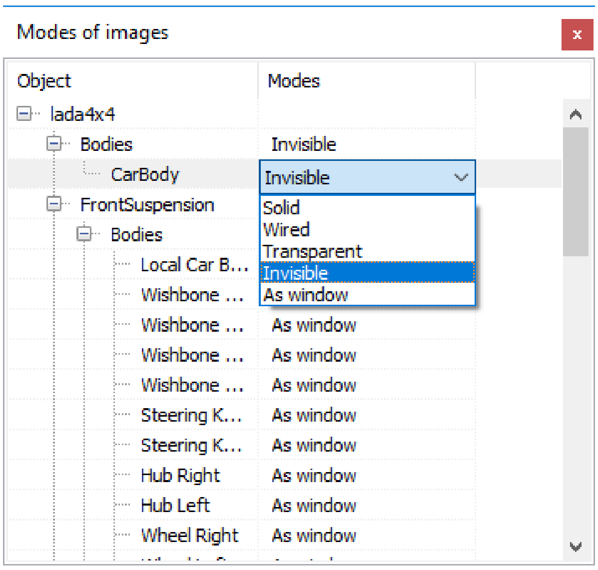



Figure 12.146. Invisible image mode for the car body

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 12.147.

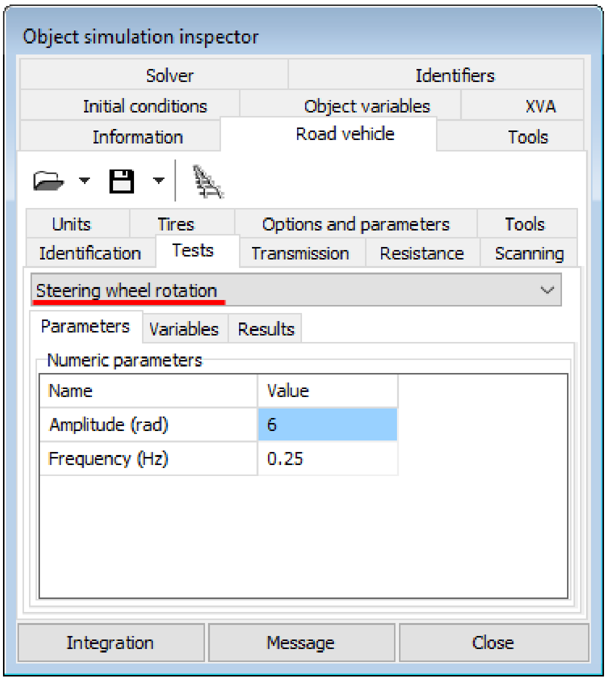



Figure 12.147. Numeric parameters for steering wheel rotation test

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 12.148.

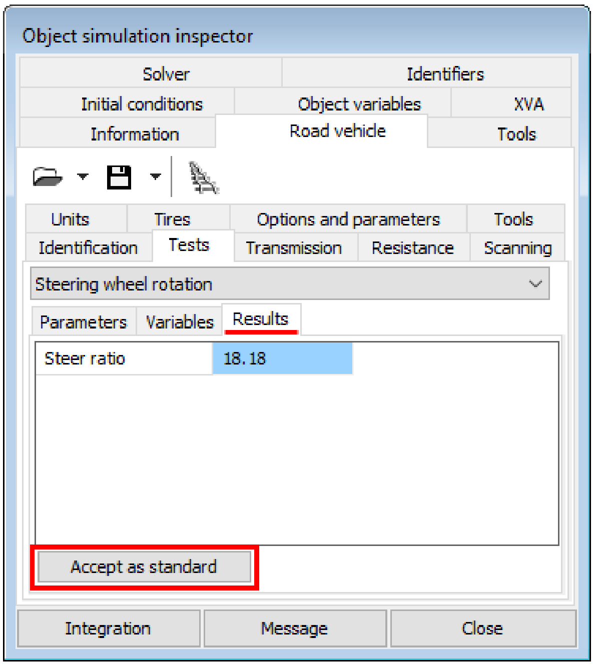



Figure 12.148. Obtained steer ratio

### 12.11.4.3. Tests with Driver

#### 12.11.4.3.1. 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 12.149. Click **OK** to close the **Options** window.

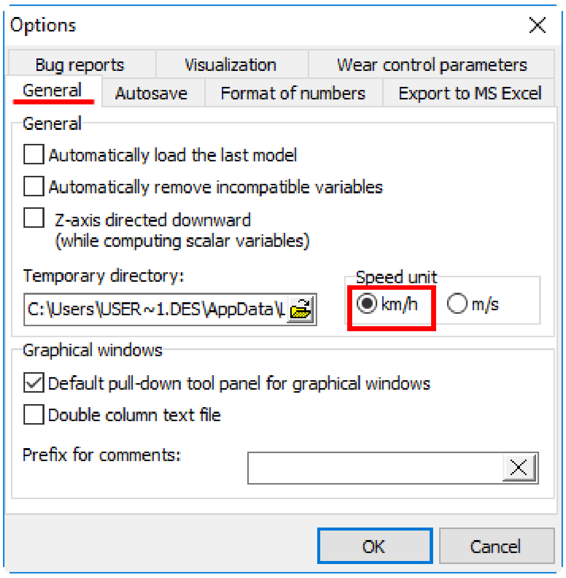



Figure 12.149. Speed unit

2. Then select the **Object simulation inspector** and click the **Identifiers | Whole list** tab sheet. Set **v0** to **5** km/h, see Figure 12.150.

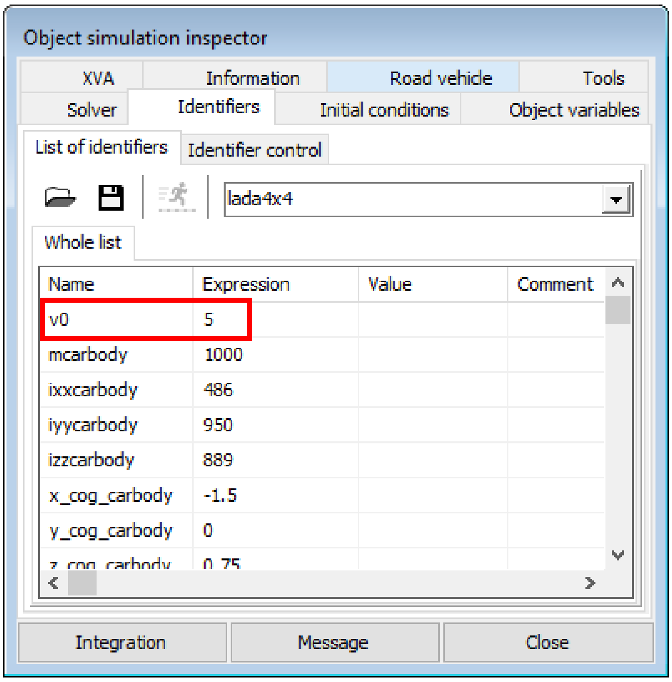



Figure 12.150. Initial speed of the vehicle is 5 km/h

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 12.151.

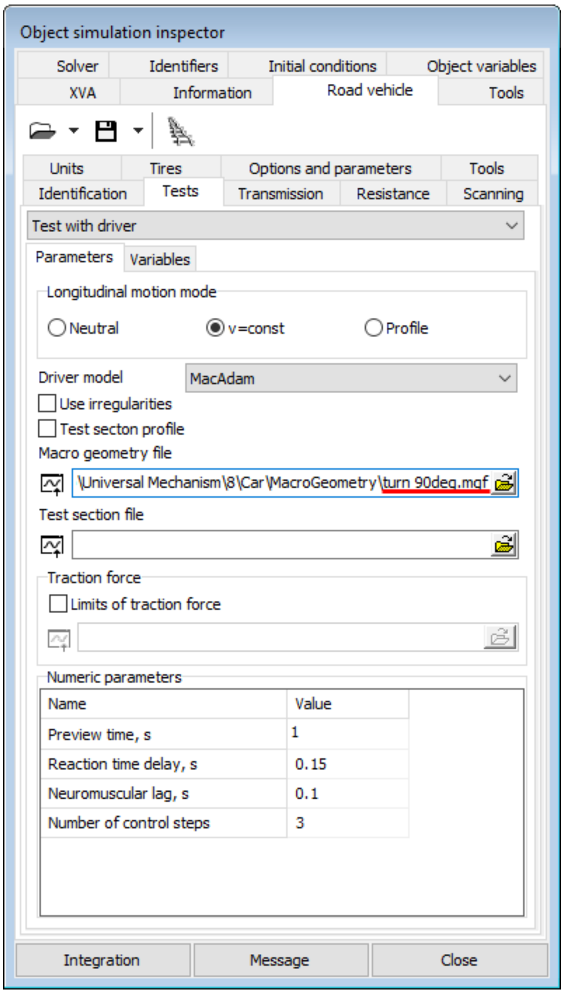



Figure 12.151. Settings for low-speed 90 degrees turn

5. Select the **Road vehicle | Tests | Variables** tab sheet. Create new graphical window and drag&drop there the **Desired path deviation** variable, see Figure 12.152.

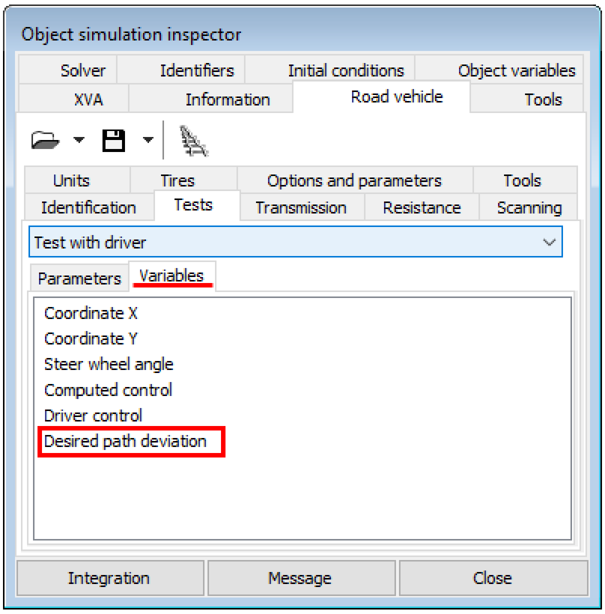



Figure 12.152. Desired path deviation

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.

### 12.11.4.3.2. Lane Change Manoeuvre

1. Select the **Identifiers | List of identifiers** tab sheet and set **v0** = **88** km/h, see. Figure 12.153.

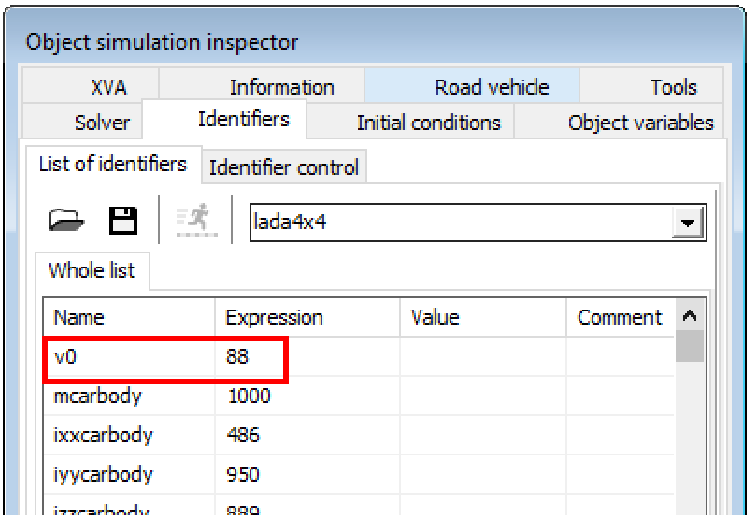



Figure 12.153. Initial speed of the vehicle is 88 km/h

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 12.154.

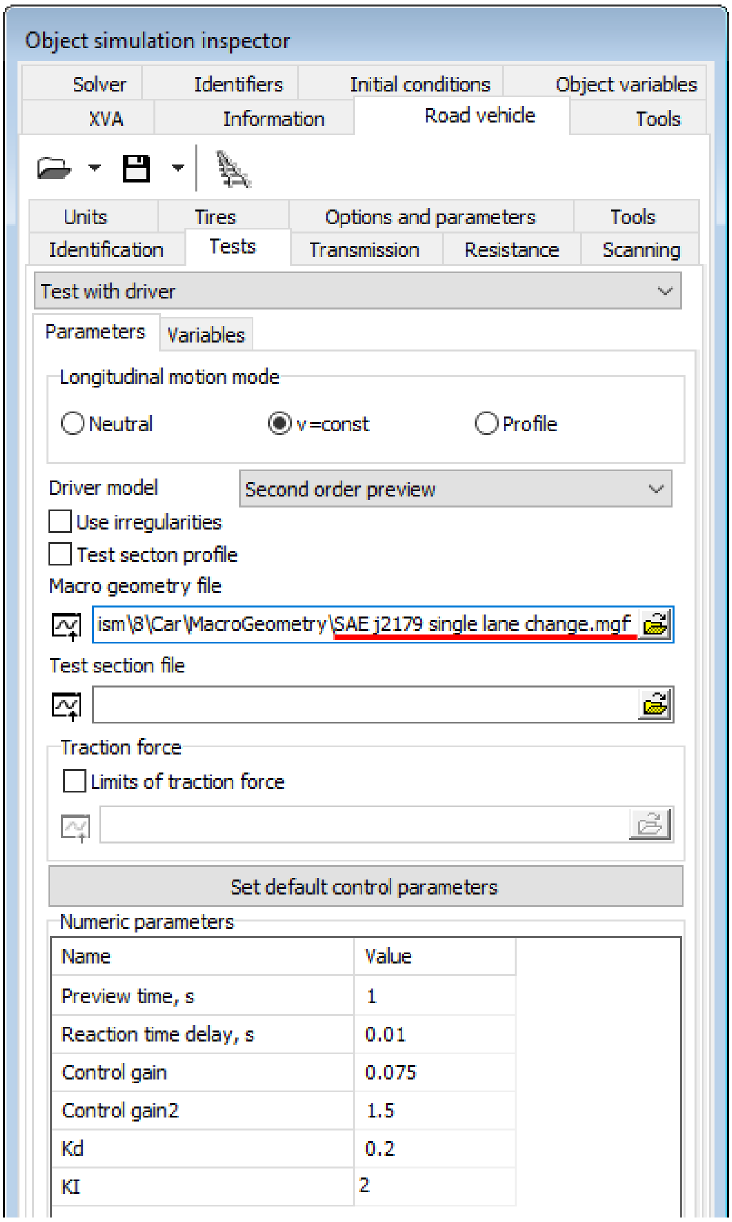



Figure 12.154. Settings for lane change manoeuvre

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. |

## 12.11.5. 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}\SAMPLES\Automotive.

### 12.11.5.1. 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.12.11.2.4. "*MacPherson Suspension*", page 12-115) is used as a front suspension, and the semi-trailing arm suspension (see Sect. 12.11.2.3. "*Semi-Trailing Arm Suspension*", page 12-114) 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 12.155. UM-model of MBW 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] Wong J.Y. Theory of Ground Vehicles. 4th Edition. Wiley. 2008.

[12] 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.

[13] Reimpel J. The Automotive Chassis: Suspension Design. Moscow, “Mashinostroenije”. 1989.

[14] Reimpel J. The Automotive Chassis: Steering. Moscow. “Mashinostroenije”. 1987.

[15] ISO 3888-1:1999 Passenger cars – Test track for a severe lane-change manoeuvre – Part 1: Double line-change.

[16] ISO 3888-2:2002 Passenger cars – Test track for a severe lane-change manoeuvre – Part 2: Obstacle avoidance.

[17] ISO 4138:2004 Passenger cars – Steady-state circular driving behaviour – Open-loop test methods.

[18] ISO 7401:2003 Road vehicles – Lateral transient response test methods – Open-loop test methods.

[19] ISO 7975:1996 Passenger cars – Braking in a turn – Open-loop test procedure.

[20] ISO/TR 8725:1988 Road vehicles – Transient open-loop response test method with one

period of sinusoidal input.

[21] ISO/TR 8726:1988 Road vehicles – Transient open-loop response test method with pseudo random steering input.

[22] ISO 9816:1993 Passenger cars – Power-off reactions of a vehicle in a turn – Open loop test method.

[23] ISO 12021-1:1996 Road vehicles – Sensitivity to lateral wind – Part 1: Open loop test method using wind generator input.

[24] ISO 14512:1999 Passenger cars – Straight ahead braking on surface with split coefficient of friction – Open loop test procedure.

[25] 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.