



User`s manual



# Simulation of Rail Vehicle Dynamics

Tools and methods for simulation of rail vehicle dynamics with Universal Mechanism software are considered

## Contents

<b>1. GENERAL INFORMATION .....</b>	<b>1-6</b>
<b>1.1. CREATING MODELS OF RAIL VEHICLES .....</b>	<b>1-8</b>
1.1.1. Type of object – Rail Vehicle .....	1-8
1.1.2. Base coordinate system .....	1-8
1.1.3. Wheelset and single wheel .....	1-9
1.1.3.1. Adding a wheelset or a wheel .....	1-9
1.1.3.2. Wheelset with six degrees of freedom .....	1-10
1.1.3.3. Wheelset with independent rotation of wheels .....	1-11
1.1.3.4. Single wheel .....	1-12
1.1.3.5. Wheelset geometry .....	1-13
1.1.3.6. Editing wheelset parameters .....	1-15
1.1.3.7. Setting torsion stiffness of axle .....	1-17
1.1.3.8. Adding other wheelsets .....	1-17
1.1.3.9. Visual adding wheelset .....	1-18
1.1.4. Modeling axle-boxes .....	1-20
1.1.4.1. Axle-boxes as massless graphic images .....	1-20
1.1.4.2. Axle-box as a rigid body .....	1-22
1.1.4.3. Visual adding an axle-box .....	1-23
1.1.5. Modeling linear springs of primary and secondary suspension .....	1-24
1.1.6. Modeling dampers, traction rods .....	1-24
1.1.7. Some features of development of locomotive models .....	1-26
1.1.7.1. Model of traction .....	1-26
1.1.7.1.1. Bodies and joints .....	1-26
1.1.7.1.2. Traction torque .....	1-27
1.1.7.1.3. Gearing .....	1-30
1.1.7.1.4. Computation of initial angular velocities by fixation file .....	1-31
1.1.7.1.5. Computation of initial angular velocities by constraints on initial values .....	1-33
1.1.7.2. Example of development and analysis of locomotive model .....	1-37
1.1.7.2.1. Development of model .....	1-37
1.1.7.2.1.1. Development of wheelset-motor assembling .....	1-37
1.1.7.2.1.2. Development of bogie model .....	1-42
1.1.7.2.1.3. Finalization of locomotive model .....	1-44
1.1.7.2.2. Simulation of locomotive dynamics .....	1-45
1.1.7.2.2.1. Creation of fixation file .....	1-45
1.1.7.2.2.2. Creation of constraints on initial conditions .....	1-47
1.1.7.2.2.3. Test for force start values .....	1-50
1.1.7.2.3. Analysis of some locomotive performances .....	1-50
1.1.7.2.3.1. Computation of dynamic factors .....	1-51
1.1.7.2.3.2. Critical speed of locomotive .....	1-53
1.1.7.2.3.3. Comparison of wheel wear factors by locomotive curving .....	1-54
1.1.7.3. Support of braking mode .....	1-58
1.1.7.3.1. General information .....	1-58
1.1.7.3.2. Use of visual components .....	1-59
1.1.8. Railway test rig model development and simulation features .....	1-63
<b>1.2. TRACK .....</b>	<b>1-66</b>
1.2.1. Track geometry .....	1-66
1.2.1.1. Geometry of rails in an ideal track .....	1-66
1.2.1.2. Geometry of curve .....	1-67
1.2.1.3. Switch geometry .....	1-71
1.2.1.4. Track system of coordinates .....	1-72
1.2.1.5. Track irregularities .....	1-73
1.2.2. Track models .....	1-74
<b>1.3. WHEEL-RAIL CONTACT .....</b>	<b>1-76</b>
1.3.1. Computation of the contact between the wheel and massless rail .....	1-76
1.3.1.1. Algorithms for wheel-rail contact geometry .....	1-76
1.3.1.1.1. Algorithm for computation of nearest points between two profiles .....	1-76

1.3.1.1.2. Computing tables of contact points .....	1-78
1.3.1.1.3. Simplified contact geometry. Equivalent conicity and contact angle parameter .....	1-82
1.3.1.1.3.1. Equivalent conicity .....	1-82
1.3.1.1.3.2. Contact angle parameter.....	1-83
1.3.1.1.3.3. Simplified geometry of contact .....	1-84
1.3.1.2. Contact forces .....	1-85
1.3.1.2.1. Method for computation of rail deflections and contact force .....	1-85
1.3.1.2.2. Algorithms for computing creep forces .....	1-87
1.3.1.2.2.1. Mueller's model .....	1-88
1.3.1.2.2.2. Minov's model .....	1-89
1.3.1.2.2.3. FASTSIM.....	1-90
1.3.1.2.2.4. FASTSIM_A.....	1-91
1.3.2. Computation of contact between the wheel and the inertial rail .....	1-91
1.3.3. Coefficient of friction in wheel-rail contact.....	1-95

## **1.4. SIMULATION OF RAILWAY VEHICLES ..... 1-97**

1.4.1. Tools for preparing simulation process.....	1-97
1.4.1.1. Creation of wheel and rail profiles.....	1-97
1.4.1.1.1. Input as a set of points with successive spline approximation.....	1-98
1.4.1.1.2. Input of profiles as a set of line segments and circle arcs.....	1-99
1.4.1.1.3. Tool for analysis of pairs of profiles.....	1-101
1.4.1.1.4. Automatic generation of new rail profiles .....	1-104
1.4.1.1.5. Import wheel profiles from CAD .....	1-107
1.4.1.1.6. Loading MiniProf wheel and rail profiles .....	1-108
1.4.1.1.7. Conversion of MiniProf wheel and rail profiles .....	1-109
1.4.1.2. Creation of track irregularities .....	1-113
1.4.1.2.1. Creation of irregularity files .....	1-113
1.4.1.2.1.1. Irregularities as analytic expression (the Expression tab) .....	1-115
1.4.1.2.1.2. Irregularities as slump .....	1-115
1.4.1.2.1.3. From file.....	1-115
1.4.1.2.1.4. Irregularities as points .....	1-115
1.4.1.2.1.5. Generation irregularities by spectrum .....	1-116
1.4.1.2.1.5.1. Spectrum: Points .....	1-117
1.4.1.2.1.5.2. Spectrum (FRA).....	1-118
1.4.1.2.1.5.3. Spectrum (ERRI) .....	1-121
1.4.1.2.1.5.4. Spectrum (Expression).....	1-122
1.4.1.2.2. Programming irregularities in the Control file .....	1-122
1.4.1.3. Creation of macrogeometry files .....	1-125
1.4.1.3.1. Track macrogeometry in horizontal plane .....	1-125
1.4.1.3.1.1. Tangent section .....	1-126
1.4.1.3.1.2. Curve section.....	1-126
1.4.1.3.1.3. Switch section .....	1-127
1.4.1.3.1.4. Point curve .....	1-130
1.4.1.3.2. Track macrogeometry in vertical plane .....	1-132
1.4.1.3.2.1. Constant slope section.....	1-133
1.4.1.3.2.2. Point section.....	1-133
1.4.1.3.3. Import of measured track geometry.....	1-134
1.4.1.4. Creation of files with switch parameters.....	1-135
1.4.1.5. Estimation of track quality.....	1-135
1.4.2. Setting parameters of rail vehicle simulation .....	1-136
1.4.2.1. Configuration files .....	1-137
1.4.2.1.1. Configuration file of rail vehicle .....	1-137
1.4.2.1.2. Group of configuration files of rail vehicle .....	1-138
1.4.2.2. Modes of longitudinal motion of vehicle .....	1-139
1.4.2.2.1. Neutral.....	1-139
1.4.2.2.2. $v=\text{const}$ .....	1-139
1.4.2.2.3. Profile .....	1-140
1.4.2.2.4. $v=0$ .....	1-141
1.4.2.2.5. Motion of vehicles with different speed .....	1-142
1.4.2.2.6. Speed unit .....	1-143
1.4.2.3. Assignment of rail and wheel profiles .....	1-145
1.4.2.3.1. Assignment of rail profiles .....	1-145
1.4.2.3.2. Assignment of wheel profiles .....	1-146

1.4.2.3.3. Rail profile evolution along the track .....	1-149
1.4.2.4. Track parameters .....	1-150
1.4.2.4.1. Track model choosing and parameters setting .....	1-150
1.4.2.4.1.1. Track stiffness and damping .....	1-151
1.4.2.4.1.1.1. Massless rail.....	1-151
1.4.2.4.1.1.2. Inertial rail .....	1-152
1.4.2.4.1.1.3. Flexible track .....	1-153
1.4.2.4.2. Track macrogeometry.....	1-153
1.4.2.4.2.1. Track type: tangent.....	1-153
1.4.2.4.2.2. Track type: curve .....	1-153
1.4.2.4.2.3. Track type: S-curve .....	1-154
1.4.2.4.2.4. Features of setting cant values in curve and in S-curve .....	1-154
1.4.2.4.2.5. How to model the motion in a left curve? .....	1-155
1.4.2.4.2.6. Track type: switch .....	1-156
1.4.2.4.2.7. Track type: from file. Set of macrogeometry files .....	1-157
1.4.2.4.3. Assignment of track irregularities .....	1-158
1.4.2.4.3.1. Irregularities from file. Group of irregularities .....	1-158
1.4.2.4.3.1.1. List of irregularity groups .....	1-159
1.4.2.4.3.2. Deterministic irregularities.....	1-160
1.4.2.4.3.3. Setting irregularities by identifiers .....	1-162
1.4.2.4.4. Track image in the animation window .....	1-166
1.4.2.5. Parameters for computation of rail-wheel contact forces.....	1-167
1.4.2.5.1. Parameters of massless rail contact .....	1-167
1.4.2.5.2. Parameters of inertial rail contact.....	1-169
1.4.2.5.3. Parameters of flexible track.....	1-170
1.4.2.5.4. Setting coefficients of friction in rail/wheel contact.....	1-170
1.4.2.6. Setting in-train forces .....	1-173
1.4.2.6.1. Assigning forces .....	1-173
1.4.2.6.2. Creating files with force description.....	1-174
1.4.2.6.3. Plotting and visualization of forces .....	1-174
1.4.2.6.4. Example of in-train forces .....	1-175
1.4.2.7. Additional parameters.....	1-178
1.4.2.7.1. Deviation of wheel form from ideal circle .....	1-178
1.4.2.7.1.1. Flat .....	1-178
1.4.2.7.1.2. Ellipse .....	1-180
1.4.2.7.1.3. Harmonic deviation.....	1-181
1.4.2.7.1.4. Points.....	1-183
1.4.2.7.2. Wheel radii difference .....	1-184
1.4.2.7.3. Rail/wheel contact options.....	1-185
1.4.3. Tools for visualization and analysis of railway vehicle dynamics .....	1-186
1.4.3.1. Some features of creation of variables .....	1-186
1.4.3.1.1. Rail/wheel contact variables.....	1-186
1.4.3.1.2. Variables related to the wheelset .....	1-190
1.4.3.1.3. Variables related to the railway vehicle.....	1-192
1.4.3.1.4. Wheel climb derailment criteria .....	1-193
1.4.3.1.4.1. Nadal and Weinstock criteria .....	1-193
1.4.3.1.4.2. Safety factor in Russian railways. Refined and combined factors .....	1-194
1.4.3.1.4.3. Simulation of motion of a freight car over dangerous combination of irregularities .....	1-196
1.4.3.1.5. Lateral and frame forces .....	1-197
1.4.3.1.6. Example: creating variable $\Sigma Y$ .....	1-198
1.4.3.1.7. Example: creating variable $\Sigma H$ .....	1-200
1.4.3.1.8. Example: draw plots for lateral motion of wheelset in the rail gangway.....	1-202
1.4.3.1.9. Kinematic characteristics relative to track system of coordinates .....	1-205
1.4.3.1.10. Use of Distance variable.....	1-206
1.4.3.1.10.1. Use of Distance variable in graphic windows .....	1-206
1.4.3.1.10.2. Use of <i>Distance</i> variable in list of calculated variables.....	1-207
1.4.3.2. Animation window .....	1-208
1.4.3.3. Contact animation window .....	1-210
1.4.3.4. Contact patch animation window.....	1-212
1.4.3.5. Table processor.....	1-213
1.4.3.6. Ride comfort functionals .....	1-214
1.4.3.7. Methods for quantifying passenger ride comfort index .....	1-217
1.4.3.7.1. Continuous ride comfort .....	1-217

1.4.3.7.2. Ride comfort on curve transitions .....	1-219
1.4.3.7.3. Ride comfort on discrete event .....	1-224
1.4.4. Solvers .....	1-228
1.4.4.1. Solver: Park method .....	1-229
1.4.4.2. Solver: Park Parallel .....	1-230
1.4.4.2.1. Conditions for use of Park Parallel solver .....	1-230
1.4.4.2.2. Solver parameters .....	1-231
1.4.4.3. Comparison of solver performances in simulation of freight car with three-piece bogies.....	1-232
<b>1.5. SIMULATION OF GEAR (COG) RAIL TRAINS .....</b>	<b>1-235</b>
1.5.1. General information .....	1-235
1.5.2. Gear wheels for propulsion system and speed control .....	1-236
1.5.3. Force element for cog rail train propulsion .....	1-239
1.5.4. Cog rail image.....	1-240
1.5.5. Simulation of vehicle on a track with cog (gear) rail .....	1-242
<b>1.6. SCANNING PROJECTS WITH MODELS OF RAIL VEHICLES .....</b>	<b>1-244</b>
1.6.1. List of internal identifiers parameterizing operation conditions of rail vehicles .....	1-244
1.6.2. Example: factorial experiment .....	1-247
<b>1.7. METHODS FOR EVALUATION OF CRITICAL SPEED .....</b>	<b>1-249</b>
1.7.1. Method of slow decrease of speed .....	1-250
1.7.2. Evaluation of critical speed by multivariant calculations .....	1-254
<b>REFERENCES .....</b>	<b>1-257</b>

# 1. General information

**UM Loco** is an additional UM module that is aimed at simulation of dynamics of rail vehicles (RV): locomotives, cars, trains.

The **UM Loco** module includes a number of additional tools in **UM Input** and **UM Simulation** programs. The module is available in the UM configuration if the sign + is set in the corresponding line of the **About** window, the **Help | About...** menu command, Figure 1.1.

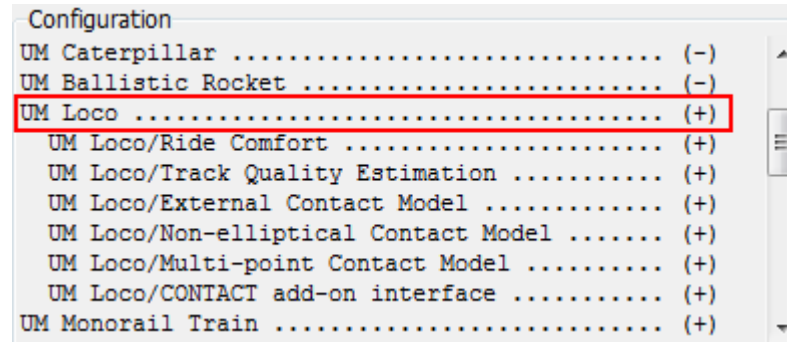


Figure 1.1. UM Loco module is available

In addition to the standard UM configuration it includes the following items:

- Wheelset as a standard subsystem;
- Algorithms of forces computation in wheel/rail contact including the FASTSIM algorithm, the model by W. Kik and J. Piotrowski, the interface to the Kalker's CONTACT model.
- Specialized graphical interface for animation of contact forces;
- Interface for creating rail and wheel profiles and track irregularities;
- Interface for setting curve and switch parameters;
- Standard list of variables, which characterize wheel/rail interaction (creepages, total, normal and creep forces in wheel/rail contacts, angle of attack, wear factors, etc.; more than 30 variables for each wheel of the vehicles);
- Database of profiles and track irregularities;
- Models of various vehicles.

**UM Loco** allows the user:

- to calculate the critical speed of a vehicle;
- to analyze 3D dynamics of a vehicle or a train in time domain in tangent track or in curves with/without irregularities;
- to analyze vehicle dynamics in dependence on wheel and rail profiles;
- to include 3D vehicles in a train model;
- to create scanning projects for scanning the vehicle/train dynamics depending on any parameters;
- to compute natural frequencies and modes, eigenvalues and eigenforms as well as root locus of linearized equations of motion;
- to create hybrid rigid-elastic models of vehicles.

Simulation of vehicle dynamics is performed in time domain by means of numeric integration of automatically generated differential or differential-algebraic equations of motion.

**UM Loco** allows the user to create *fully parameterized models* of vehicles. Geometrical, inertia, force parameters may be specified using identifiers and symbolic expressions. The parameterization of model is the base for its optimization.

More general information about use of UM for simulation can be found in the PowerPoint presentation at <http://www.universalmechanism.com/download/docs/eng/umloco.ppsx>.

Below in the current chapter we consider some features of description of a RAIL VEHICLE in UM as well as some useful notions and special utilities.

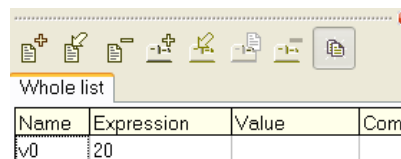
## 1.1. Creating models of rail vehicles

User creates UM models of rail vehicles (RV) similar to any other multibody systems, see [Chapter 3](#). The vehicle is considered as a system of rigid or flexible bodies connected by means of joints and force elements. Usually a model of a rail vehicle contains the following rigid bodies: vehicle body, bogie frames, wheelsets, axle-boxes (often can be removed from the model) etc. In the case of a locomotive, a motor model consists often of two bodies: a motor case and a rotor.

See the [gs\\_UM\\_Loco.pdf](#) file with detailed information about process of a rail vehicle model development.

### 1.1.1. Type of object – Rail Vehicle

To get an UM model of a vehicle the user should create a new UM object by clicking the **File** | **New object** menu item, *and at least one wheelset must be added to the object*. After that the standard identifier **v0** for the initial speed of the vehicle is added to the list of identifiers [Figure 1.2](#); more precisely, this identifier specifies the initial value of speed in the neutral and constant speed modes of longitudinal motion, [Sect. 1.4.2.2. "Modes of longitudinal motion of vehicle"](#), p. 1-139.



Name	Expression	Value	Com
v0	20		

Figure 1.2. Identifier of speed

### 1.1.2. Base coordinate system


Base coordinate system (SC0, Base0) is the system of coordinate in which the object is described and simulated. For a rail vehicle, the SC0 satisfies the following requirements:

- Z-axis is directed vertically upwards;
- X-axis is horizontal along direction of motion at the vehicle initial position;
- SC0 origin is usually located either at the level of the rail head or at the wheelset axes level at their ideal initial position (obsolete).

We consider an element of the vehicle model as a *left* one (left wheel, left spring etc.) if it has a positive Y-coordinate, i.e. if the element is on the left to the motion direction.

### 1.1.3. Wheelset and single wheel

#### 1.1.3.1. Adding a wheelset or a wheel

Wheelset (WS) in UM is a *standard subsystem*. To add a wheelset or a single wheel, select the **Subsystem** item in the list of elements, add a subsystem by the  button and select the **Wheelset** type of the subsystem, Figure 1.3.

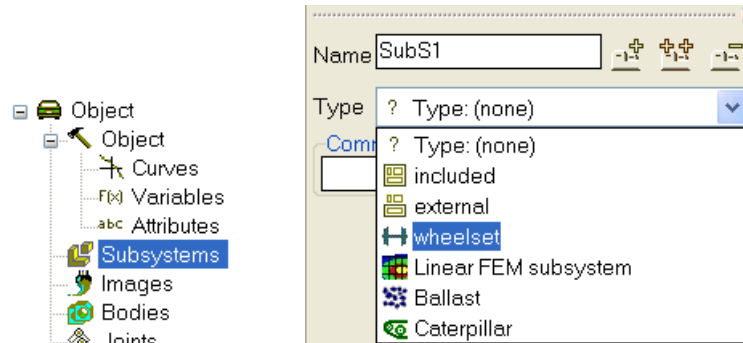


Figure 1.3. Adding a wheelset

Two types of wheelsets are available in UM. These types differ in number of degrees of freedom as well as a separate wheel with an individual suspension, Figure 1.4:

- **Standard:** a WS with six d.o.f. and a rigid axle;
- **Independent rotation:** is used for modeling wheelsets with independently rotating wheels as well as for modeling torsion flexibility of the standard WS;
- **Single wheel (left);**
- **Single wheel (right).**

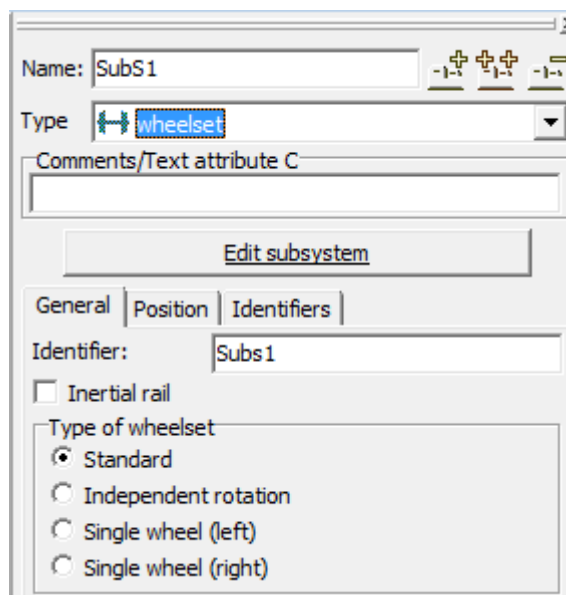


Figure 1.4. Type of wheelset

Templates of the subsystems are located in the directory **{UM data}\rw\wset:**  
 Standard.wst

Independent.wst

SingleRight.wst

SingleLeft.wst

Starting from UM 9.0 there is a possibility to use wheelset subsystems including inertial rails. Rails are considered as rigid bodies and named as *leftrail* and *rightrail*. Rails as rigid bodies might have three d.o.f. – two translational d.o.f. relative to lateral (Y) and vertical (Z) axes and one rotational d.o.f. relative to longitudinal X axis. All three degrees of freedom are optional. Every mentioned above degree of freedom may be turned off. Rest three degrees of freedom have no effect on simulation results since plane model of inertial rail is considered. Rails are connected with *Base0* body through *Bushing* force elements that simulate road bed. To use wheelset subsystems with inertial rails turn on flag **Inertial rail**.

Wheelset subsystems can be modified by the user. **It is strongly recommended not to change the number of coordinates as well as their sequence.**

To edit a template

- create its backup,
- create a new UM object,
- read the template by the **Edit | Read from file...** menu command or by the 📄 button on the tool panel,
- apply modifications,
- save the template by the **File | Save as component...** menu command or by the 📁 button.

### 1.1.3.2. Wheelset with six degrees of freedom

The model of **wheelset with 6 d.o.f.** includes two bodies, two joints and a wheelset image (Figure 1.5). The first body presents the *base* of the wheelset with 5 degrees of freedom. This body lacks for the rotation around the wheelset axle. The standard name of the base is *Wset*. The second body with the standard name *WSetRotat* is a gyrostat. It has the only degree of freedom relative to the base – rotation about the axle. It is known that equations of motion of the base plus gyrostat completely coincide with equations of motion of one body with six degrees of freedom.

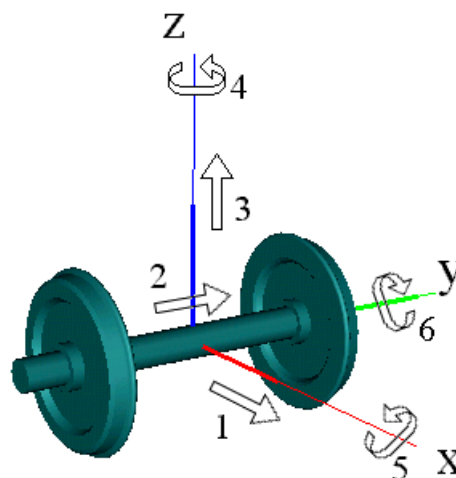


Figure 1.5. Wheelset degrees of freedom

One of the goals of introduction of two bodies for one wheelset is as follows. In many cases it allows avoiding the introduction of axle-boxes as separate bodies and reduces considerably CPU expenses while simulation of the vehicle dynamics. In fact, it is impossible to attach linear force elements that are usually used for the modeling the primary suspension (springs, dampers, guides etc.) directly to rotating wheelset with 6 d.o.f., axle-boxes are necessary. In contrary, it is possible to attach these elements to the wheelset base (it does not rotate about the axle!) and the axle-boxes can be omitted in the model.

The coordinates of the wheelset are numbered in the following sequence (Figure 1.5):

- 1 is transition X;
- 2 is transition Y;
- 3 is transition Z;
- 4 is rotation Z;
- 5 is rotation X;
- 6 is rotation Y.

Starting from UM 9.0 built-in **Wheelset** subsystem includes two rigid bodies that simulate inertia properties of rails. Default names of bodies are *leftrail* and *rightrail*. Both bodies have three d.o.f. – two translational relative to Y and Z axes and one rotational relative to X axes. Inertia rails are supported by **Multipoint** contact model only. Other contact models ignore inertia properties of the rails.

The subsystem contains identifiers, which parameterize inertia and geometrical parameters of the WS, Table 1.1.

Table 1.1

Identifier	Comments
<i>mwset</i>	Mass of WS
<i>ixwset</i>	Moment of inertia relative to the longitudinal central axis
<i>iywset</i>	Moment of inertia relative to the lateral central axis
<i>axlelength</i>	Standard identifier for the axle image length. It is not recommended to remove or rename this identifier
<b>Parameters for inertial rails</b>	
<i>mleftrail, mrighttrail</i>	Masses of left and right rail correspondingly
<i>cyleftrail, cyrighttrail</i>	Lateral stiffness
<i>dissyleftrail, dissyrighttrail</i>	Lateral damping
<i>czleftrail, czrighttrail</i>	Vertical stiffness
<i>disszleftrail, disszrighttrail</i>	Vertical damping
<i>canglleftrail, cangrighttrail</i>	Torsion stiffness
<i>dissangleftrail, dissangrighttrail</i>	Torsion damping

### 1.1.3.3. Wheelset with independent rotation of wheels

This type of the WS is used for modeling of a WS with independent rotation of the wheels as well as for modeling an ordinary WS with the torsion flexibility of the axle, Figure 1.6.

In this case, the WS has seven degrees of freedom. In addition to six d.o.f., a rotation of the right wheel relative to the left one is introduced. The model of the WS consists of three bodies and three joints.

Two identifiers  $iywset\_l$ ,  $iywset\_r$  parameterize the moment of inertia relative to the Y axis of the left and right wheels.

To introduce the torsion flexibility of the axle, the user should add a joint torque for the third joint  $jWsetRightWheel$ . The torque must contain both elastic and dissipative parts.

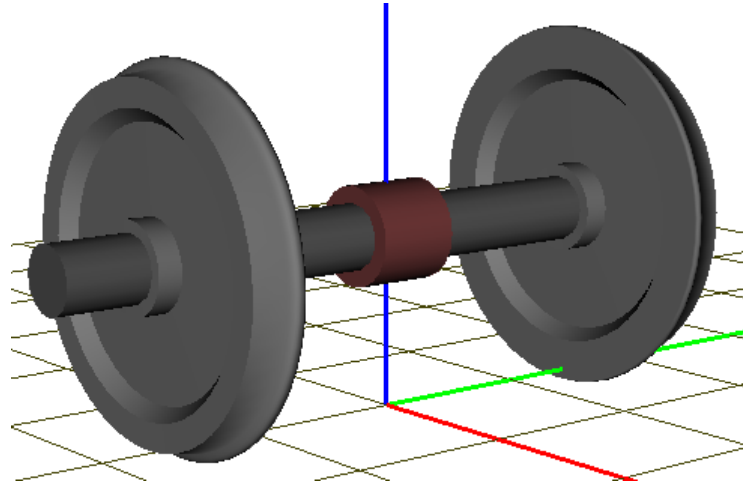


Figure 1.6. Wheelset with independent wheels

#### 1.1.3.4. Single wheel

This type of WS can be used for modeling wheels with independent suspensions, Figure 1.7. Subsystem description is similar to the WS with six d.o.f. An additional identifier  $ycg$  specifies the lateral position of the center of gravity.

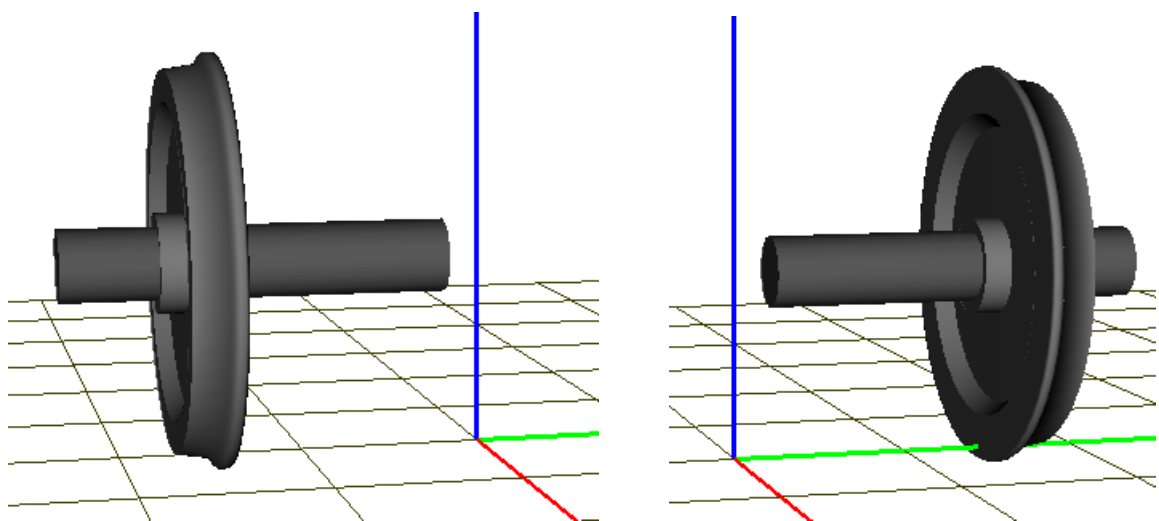


Figure 1.7. Single left and right wheels

Two single wheels can be used for modeling torsion and bending of the WS axle. In this case, both the left and the right single wheels are to be added with the same longitudinal positions. After that a bushing force element is added describing flexibility of the axle. Values of stiffness and damping parameters of the bushing are evaluated similar to that in Sect. 1.1.3.7. "*Setting torsion stiffness of axle*", p. 1-17.

### 1.1.3.5. Wheelset geometry

Geometrical properties of a wheelset are fully set with the following data (Figure 1.8, Figure 1.9):

- wheelset semibase ( $L/2$ );
- running circle radius  $r$  ;
- difference of running circles radii  $dr_l, dr_r$ , Sect. 1.4.2.7.2. "*Wheel radii difference*", p. 1-184;
- deviation of the wheel form from the circle (flat, ellipse and so on, Sect. 1.4.2.7.1. "*Deviation of wheel form from ideal circle*", p. 1-178);
- tread (wheel) profiles for the left and right wheels, which should be given in a special coordinate system of profile.

The first two parameters are specified in the **UM Input** program as the parameters of the standard subsystem (wheelset).

Differences of running circles radii  $dr_l, dr_r$  are specified both for the left and right wheels. They are the difference between the real values  $r_l, r_r$  of the wheel radii, and the nominal one  $r$ , Sect. 1.4.2.7.2. "*Wheel radii difference*", p. 1-184,

$$dr_l = r_l - r, \quad dr_r = r_r - r.$$

Thus, one WS can have wheels with different radii. Besides, different WS can have different radii of wheels.

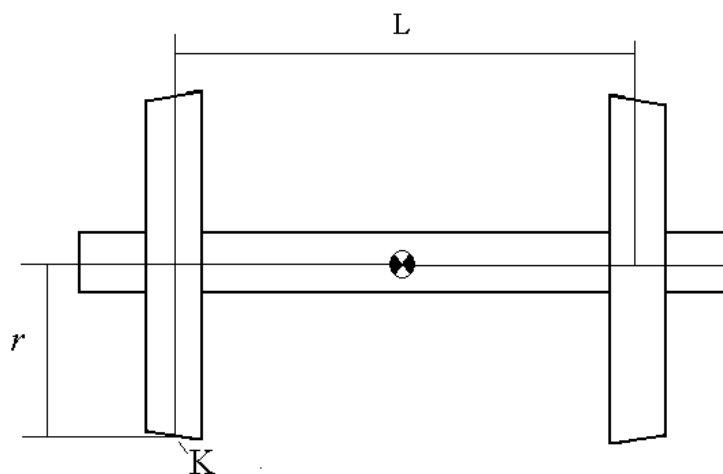


Figure 1.8. Geometrical parameters of wheelset

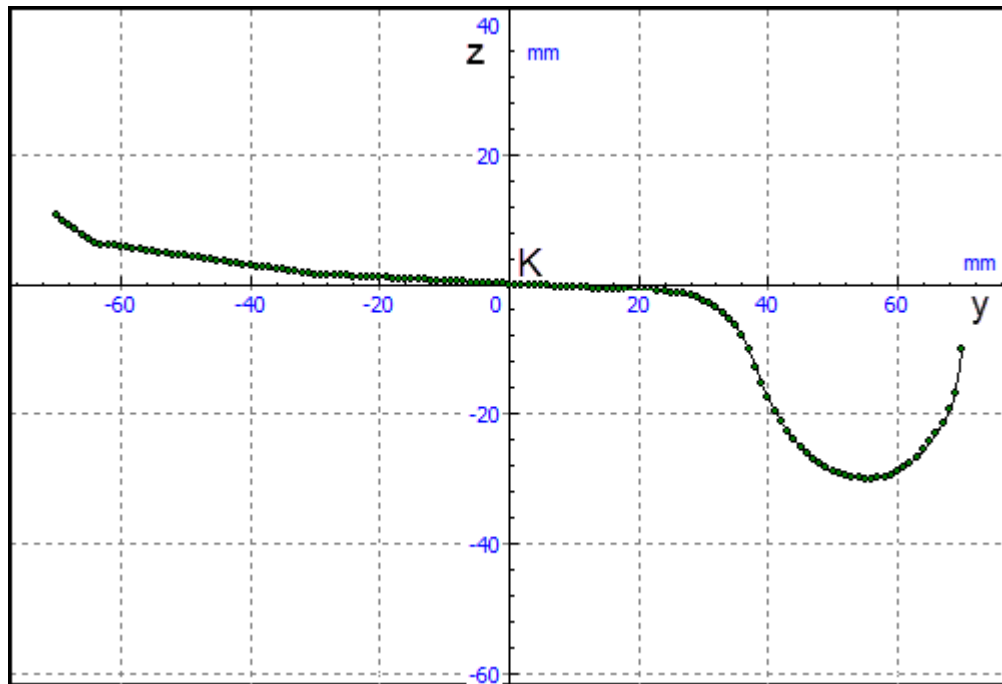


Figure 1.9. System of coordinates of a wheel profile

Profiles are chosen from the UM database or created by the user with a special tool in the **UM Simulation** program (Sect. 1.4.1.1. *"Creation of wheel and rail profiles"*, p. 1-97).

Let us introduce the notion of system of coordinate of a wheel (SCW) profile Figure 1.9. The SCW origin is located at point K on the running circle (Figure 1.8) and corresponds to the middle point of the profile along abscissa. The abscissa axis (y) is parallel to the wheelset axle and is directed towards the flange. Z-axis is vertical. Coordinates of the profile points are set in mm.

### 1.1.3.6. Editing wheelset parameters

The following parameters are available in the inspector window (Figure 1.10):

- **Name** of the subsystem, e.g. Wheelset1;
- **Identifier** of the subsystem must consist of Latin letters and digits, e.g. Wset1. The identifier is used for access to the identifiers of the wheelset while programming in the UM environment. *The identifier should be unique within the object.*
- **Type of WS** (see. Sect. 1.1.3.1. "Adding a wheelset or a wheel", p. 1-9).

Further parameters of the wheelset:

- **Radius** of the running circle (Sect. 1.1.3.5. "Wheelset geometry", p. 1-13);
- **Semibase** of the wheelset (Sect. 1.1.3.5. "Wheelset geometry", p. 1-13).

Other parameters in the inspector window allow the user to modify the wheelset image:

- **Axle length** increases or decreases the length of the axle image

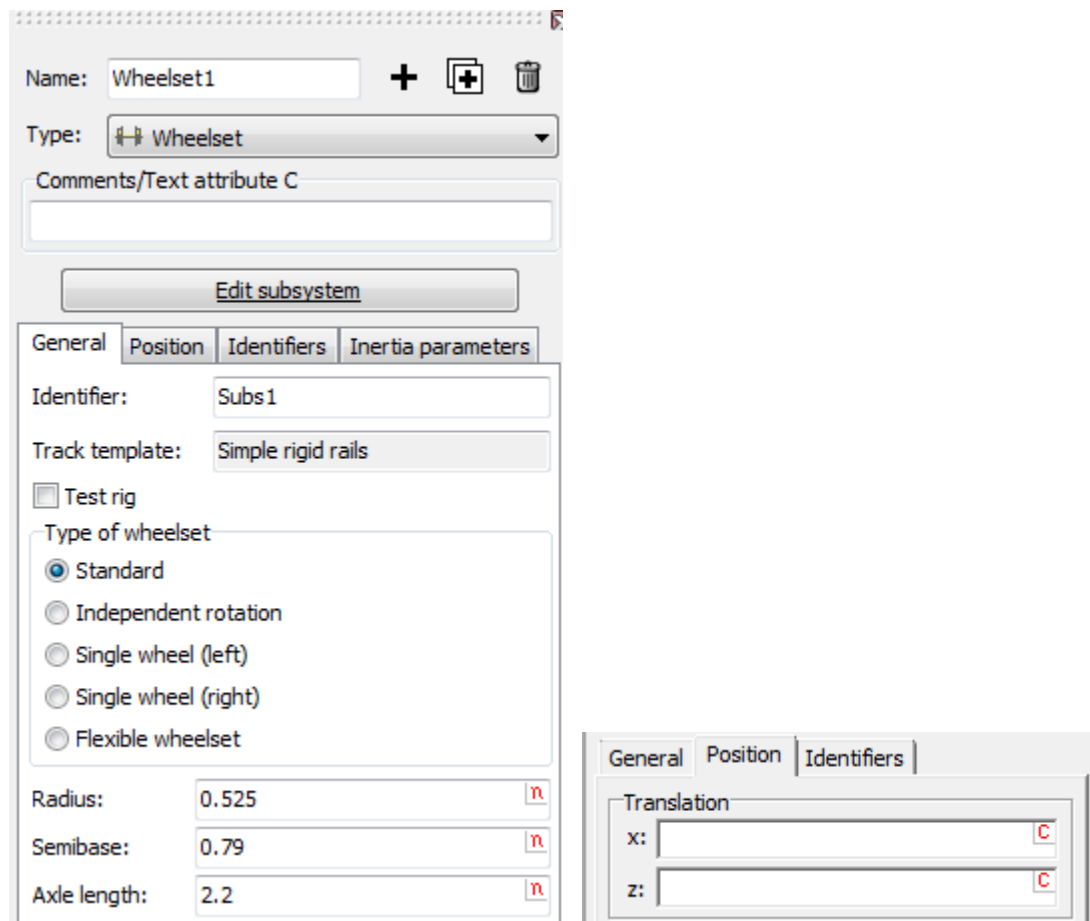


Figure 1.10. Wheelset parameters

Changing geometric parameters of the wheelset results in automatic modification of its image (Figure 1.11).

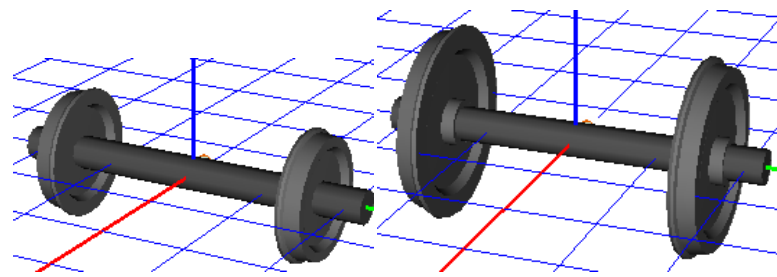


Figure 1.11. Images of wheelset

Use the **Position** tab (Figure 1.10) to specify the longitudinal and vertical position of the wheelset. If the object is a bogie, the lateral position is set relative to the SC of the bogie. If the object is a vehicle, the lateral position is set relative to the corresponding SC. Position of the wheelset can be parameterized, i.e. specified by a symbolic expression ([Chapter 3](#), Sect. Constant symbolic expression). Set the z coordinate of the wheelset equal to the wheel radius to locate the origin of SC0 at the rail head level (Sect. 1.1.2. "Base coordinate system", p. 1-8). Zero value of this coordinate corresponds location of the SC0 origin on the wheelset center level.

Use the **Edit subsystem** button to change inertia parameters or to modify the image (e.g. to add images of axle-boxes). This button opens the subsystem as a multibody system in a separate window where the necessary modifications of the wheelset such as changing parameters or re-naming elements can be done in a usual manner (see [Chapter 3](#), [Chapter 9](#)).

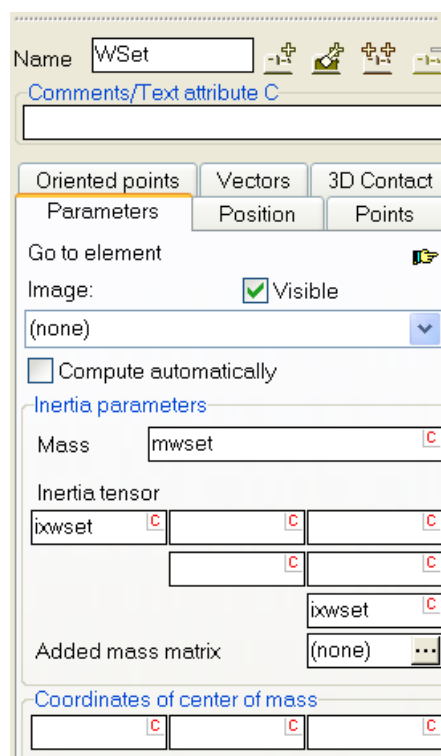


Figure 1.12. Inertia parameters of WS base

### 1.1.3.7. Setting torsion stiffness of axle

To set the axle torsion stiffness and damping in the case of a WS with seven d.o.f. (Sect. 1.1.3.3), a joint torque of the **Linear** type must be specified with proper values of stiffness and damping constants, Figure 1.13.

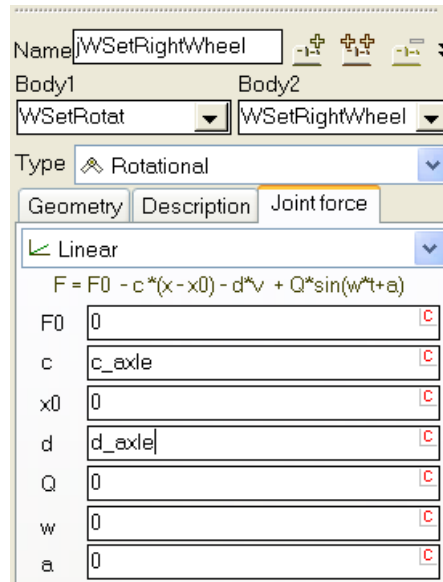


Figure 1.13. Setting torsion stiffness and damping

To evaluate the stiffness  $c$  and damping  $d$  constants in case of equal moments of inertia of the left and right wheels  $I_y$ , it is recommended to use the value of the first torsion frequency  $f$  of WS as well as the damping ratio  $\beta$ :


$$c = \frac{I_y(2\pi f)^2}{2}, d = 2\beta \sqrt{\frac{I_y c}{2}}$$

For instance, the recommended values for  $f=40$  Hz,  $\beta=0.02$ ,  $I_z=150$  kgm<sup>2</sup> are

$$c = 4.74 \times 10^6 \text{ Nm}, d = 754 \text{ N} \cdot \text{m} \cdot \text{rad/s}.$$

### 1.1.3.8. Adding other wheelsets

To add all other wheelsets:

- use the operation of copying the described wheelset by clicking the  button;
- change the name and identifier of the new wheelset;
- change the longitudinal position and other parameters (if necessary).

### 1.1.3.9. Visual adding wheelset

The **UM Loco** tab is used for visual adding wheelsets, Figure 1.14. The standard UM Loco configuration includes three components:

- standard WS with six d.o.f.;
- WS with independent rotation of the wheels;
- WS with torsion flexible axle; torsion stiffness and damping constants.

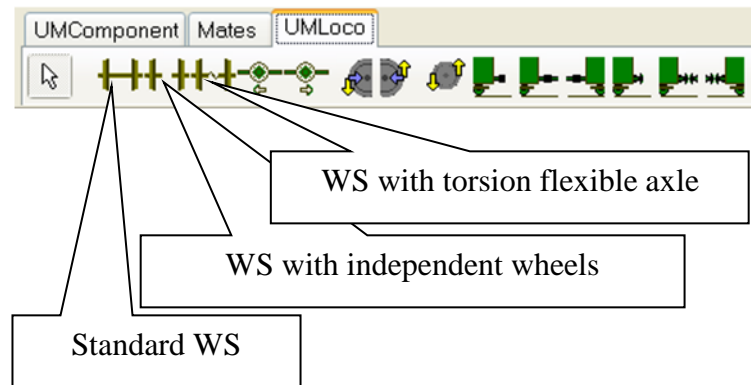


Figure 1.14. Visual components of UM Loco

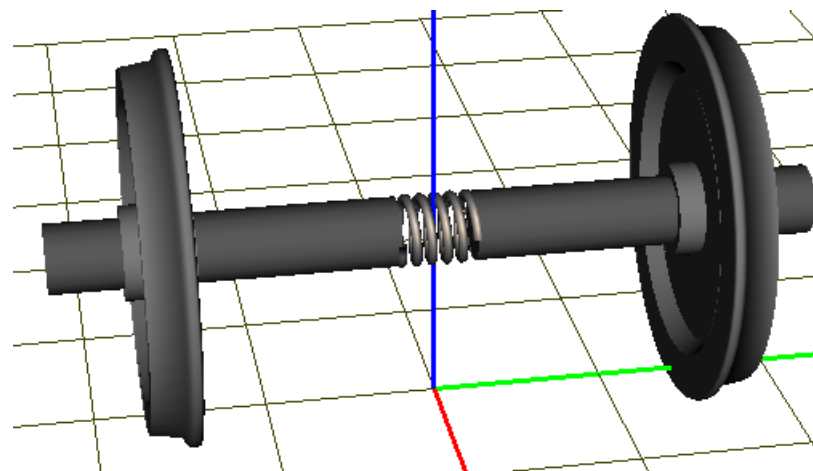


Figure 1.15. Wheelset with torsion flexible axle

Origin of system of coordinates is located on the rail head level. After changing the wheel radius, the vertical position of the WS must be modified on the **Position** tab.

Image of WS with torsion flexible axle is shown in Figure 1.15. Values of the torsion frequency and damping ratio can be modified in the **Identifiers** tab of the subsystem, Figure 1.16.

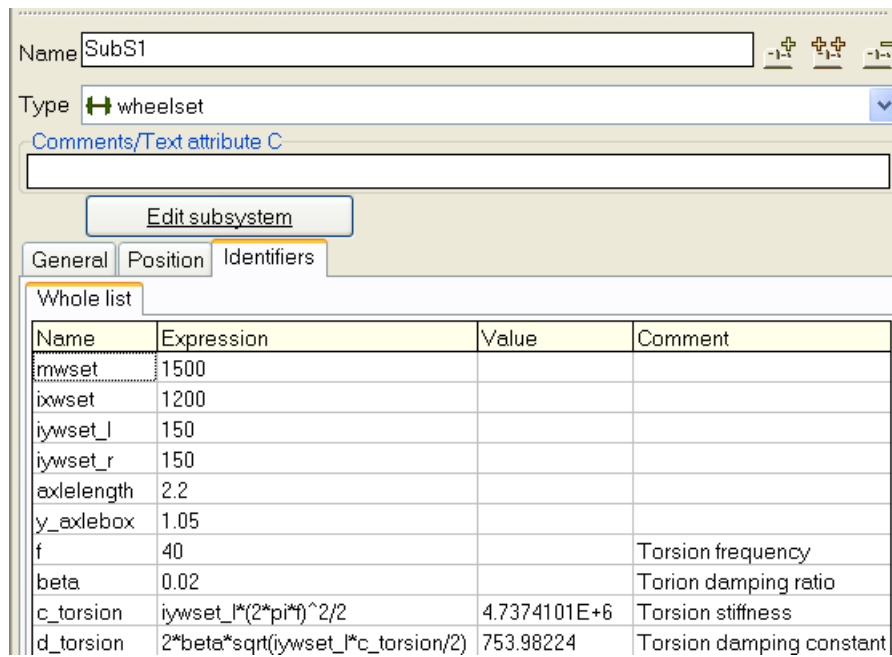


Figure 1.16. Identifiers parameterizing the WS with torsion flexible axle

**Remark.** The user can develop his/her own components and place them on the tab of visual components, see [Chapter 3](#).

### 1.1.4. Modeling axle-boxes

Sometimes axle-boxes are not included in vehicle models as separate bodies (Sect. 1.1.3.2. "Wheelset with six degrees of freedom", p. 1-10), and the bogie frame is connected directly with the base of the wheelset (non-rotating part of the wheelset). In this case graphic images of axle-boxes can be introduced to make the vehicle model more realistic. The images of the axle-boxes are assigned to the base of the wheelset.

In other cases introduction of axle-boxes as separate rigid bodies is necessary to make the model correct. Some of these cases are listed below.


- Non-symmetric attachment and different stiffness of a pair of springs connected with the axle-box.
- Non-symmetric attachment of traction rods for traction wheelset;
- Lateral gaps in axle-box assembly for some locomotives with 3 axle bogies.

#### 1.1.4.1. Axle-boxes as massless graphic images

Let us consider how we can add images of axle-boxes without adding them as rigid bodies. A simplified standard image for an axle-box can be obtained from the file

`\bin\rw\images\axlebox.img`

Make the following actions with a wheelset already added to the object.

- Run modification of the subsystem by clicking the **Edit subsystem** button (Figure 1.16).
- Read image of the **left** axle-box from the `\bin\rw\images\axlebox.img` file by the  button on the tool panel. Rename the image, e.g. *Axlebox L*.
- Copy the image of the left axle-box, rotate it on 180 degrees about the vertical axis and rename the copy as *Axlebox R*.

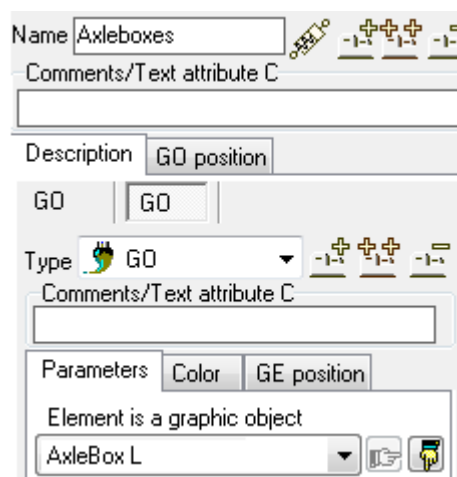


Figure 1.17. Graphic element of "GO" type

- Create a new graphic object with two graphic elements of the **GO** type, assign images of the axle boxes to the elements, Figure 1.17. Set lateral position of the axle-boxes and rename the image. The result is shown in Figure 1.18.

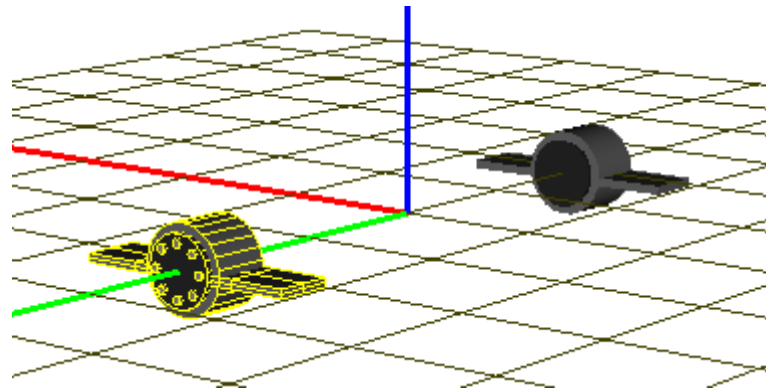


Figure 1.18. Image of fictitious axle-boxes

- Finally, select the *Wset* body in the list of bodies (the base of the wheelset, Sect. 1.1.3.2. "Wheelset with six degrees of freedom", p. 1-10), and assign to it the created image, Figure 1.19.

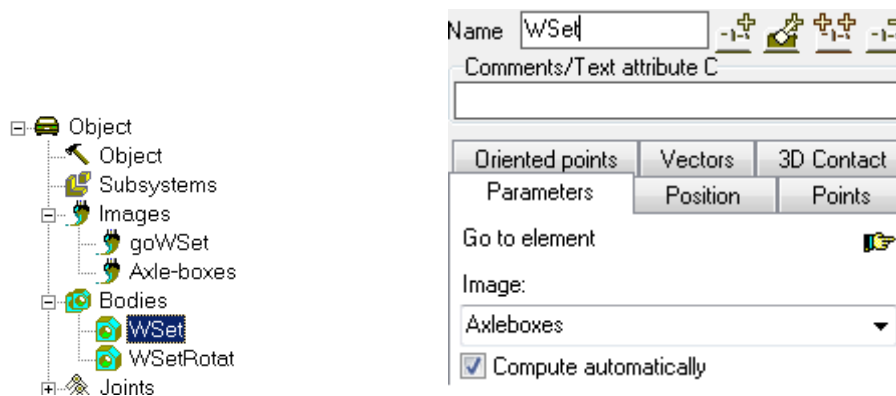


Figure 1.19. Selection of the Wset body in the list and assignment of an image

- Close editing the wheelset subsystem by clicking the **Accept** button in the dialog window or use the **X** button and confirm the acceptance of the modifications, Figure 1.20. As a result, the wheelset image will look like Figure 1.21.

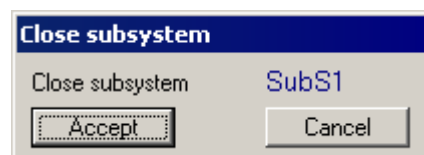


Figure 1.20. Close subsystem window

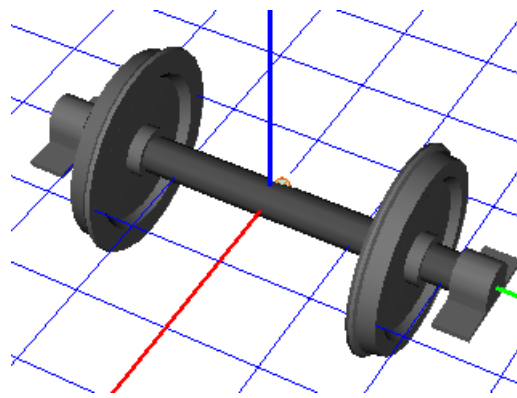



Figure 1.21. Wheelset with fictitious axle-boxes

Of course, the image of the axle-boxes cannot be assigned to the rotating part of the wheelset, because this assignment will lead to the corresponding rotation of the image during simulations.

### 1.1.4.2. Axle-box as a rigid body

If an axle-box should be considered as a separate body, the user should use the following instructions.

1. Do not open the wheelset for editing. Modify the object, which contains the wheelset as an included subsystem.
2. Create a new graphic object for the axle box image. In the simplest case, the standard image `bin\graph\axlebox.img` could be used (read it by the  button in the tool panel).
3. Add a new body for the axle-box, rename it, assign the image and set inertia parameters.

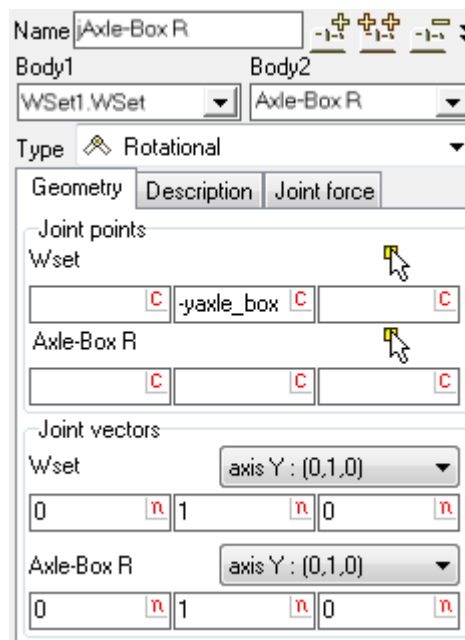



Figure 1.22. Adjustment of a joint to the right axle-box

4. Adjust a joint to the axle-box by the  button (Figure 1.22). If the axle-box has one rotational degree of freedom relative to the axle, the rotational joint should be assigned. After that

- change the first body (base of the wheelset instead of Base0);
- set axis of rotation (0, 1, 0) about Y-axis for each of the bodies;
- set the axle-box position on the axle (expression – *yaxle\_box* in Figure 1.22 for the right axle-box).

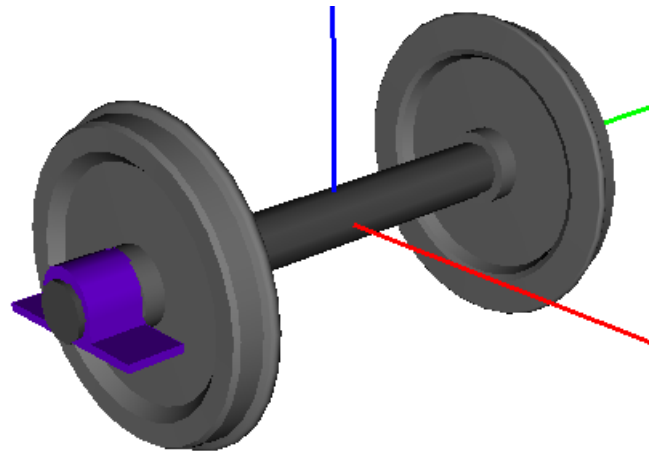



Figure 1.23. Wheelset with the right axle-box

### 1.1.4.3. Visual adding an axle-box

Axle boxes can be added by visual components.

- Open the **UMLoco** tab of the visual component lists.
- Click on the axle-box button, then click somewhere on the grid for a temporary location of the axle-box.
- Assign a revolute joint using either method above or the visual component  of the joint.

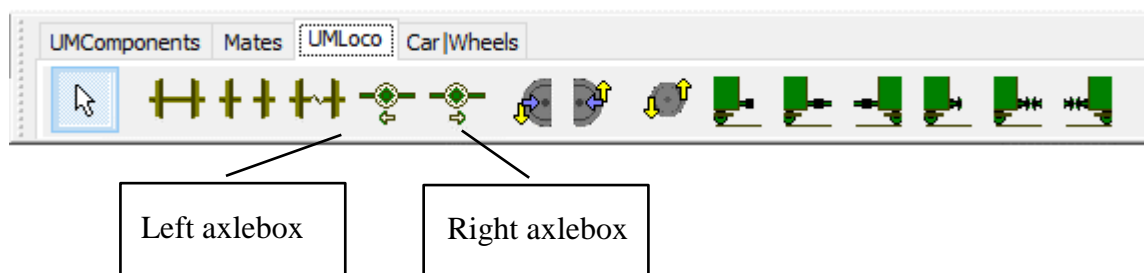


Figure 1.24. Visual components of UMLoco module

### 1.1.5. Modeling linear springs of primary and secondary suspension

The following types of force elements are used for modeling both linear and nonlinear springs.

- *Special force element Spring*

The element is used for modeling linear springs.

Mathematic model of the element is described in [Chapter 2](#), Sect. *Special forces/Spring, Generalized linear force element*.

Examples of description and/or usage:

- [Chapter 7](#), Sect. Models of Springs
- Model: [\UM Data\ \SAMPLES\Rail\\_Vehicles\ac4](#)

- *Generalized linear force element*

The element allows modeling both the same springs as the previous one and more complicated or simplified cases, e.g. when the shear stiffness depends on direction of deflection. It can be used also for modeling bilinear springs.

Mathematic model of the element is described in [Chapter 2](#), Sect. *Generalized linear force element*.

Examples of description and/or usage:

- [Chapter 7](#), Sect. *Models of Springs*;
- Model: [\UM Data\ \SAMPLES\Rail\\_vehicles\Manchester\\_benchmarks\Vehicle1](#);
- Model: [\UM Data\ \SAMPLES\Rail\\_vehicles\wedgetest](#).

- Bipolar force element

The element can be used for modeling both linear and nonlinear springs, which produce force directed along the element attachment points

Mathematic model of the element is described in [Chapter 2](#), Sect. *Bipolar force element*.

Examples of description and/or usage:

- Model: [\UM Data\ \SAMPLES\Rail\\_vehicles\Manchester\\_benchmarks\Vehicle1](#);
- Model: [\UM Data\ \SAMPLES\Rail\\_vehicles\Manchester\\_benchmarks\Vehicle2](#).

General recommendations:

1. models should be created near the equilibrium position at zero values of object coordinates; stationary values of vertical loads should be set for springs;
2. set the lower body as the first attachment body of a vertical spring and the upper as the second one; in this case the stationary vertical force is positive.

### 1.1.6. Modeling dampers, traction rods

As a rule, bipolar force elements are used for modeling dampers and traction rods.

Mathematic model of the element is described in [Chapter 2](#), Sect. *Bipolar force element*.

Examples of description and/or usage:

- Model: [\UM Data\ \SAMPLES\Rail\\_vehicles\Manchester\\_benchmarks\Vehicle1](#);
- Model: [\UM Data\ \SAMPLES\Rail\\_vehicles\Manchester\\_benchmarks\Vehicle2](#);

- Model: [{UM Data}\SAMPLES\Rail\\_vehicles\ac4.](#)

### 1.1.7. Some features of development of locomotive models

Features of locomotive models are traction systems.

#### 1.1.7.1. Model of traction

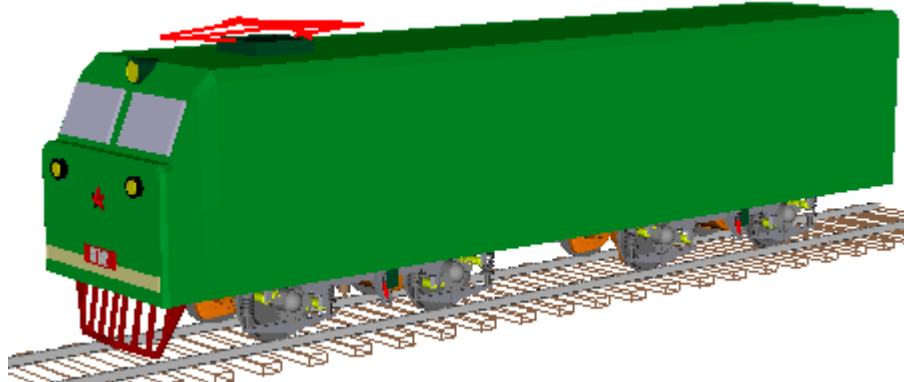


Figure 1.25. Model of one unit of VL80 electric locomotive

Simplified modeling of traction motors and wheel driving systems is necessary for realization of traction mode. Specific features of the model depend on a type of the drive system. Here we consider the axle-hung traction motors taking the VL80 locomotive as an example, Figure 1.25.

##### 1.1.7.1.1. Bodies and joints

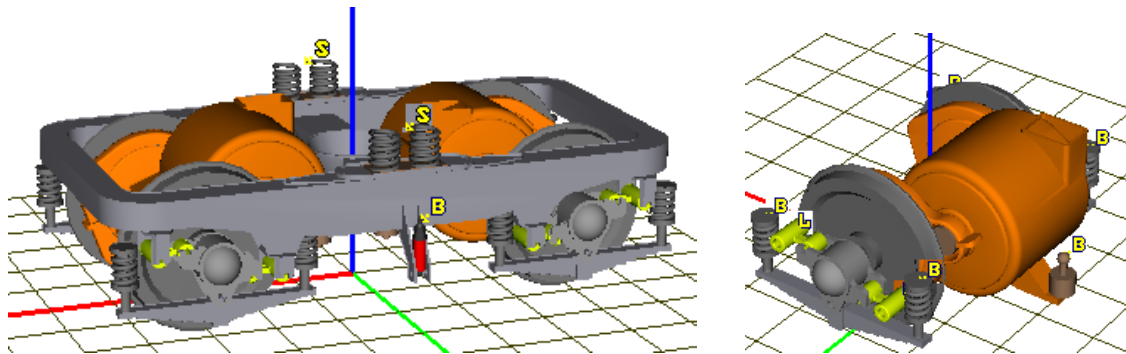


Figure 1.26. Subsystems “bogie” and “wheel-motor assembly”

Model of one unit of the locomotive includes two subsystems “bogie”, which contain two subsystems “wheel-motor assembly” each. Traction system is realized within the “wheel-motor assembly” subsystem.

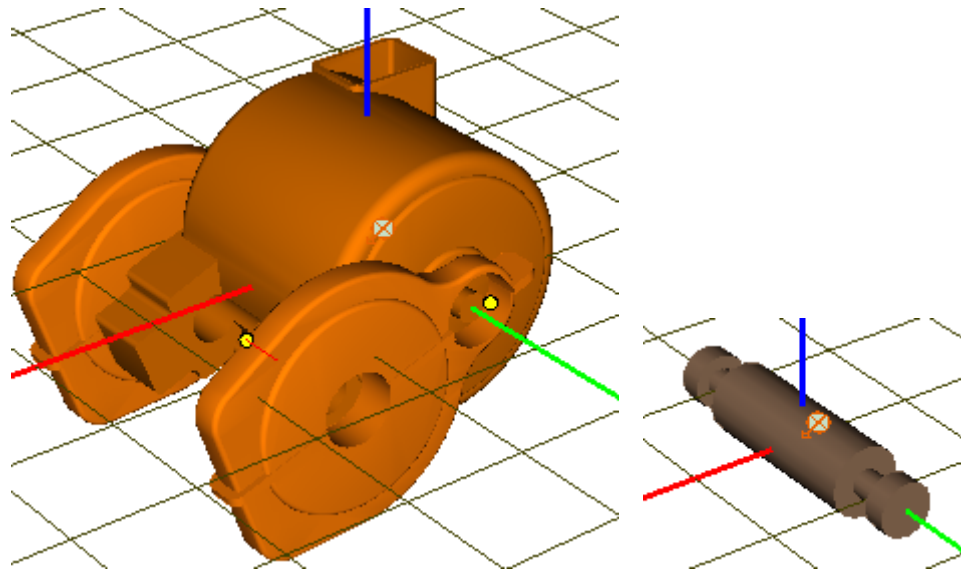


Figure 1.27. Bodies: motor and reducer casings; drive shaft

Model of the wheel-motor assembly contains a wheelset, two axle-boxes, motor and reducer casings as one rigid body, rotor with two pinions (drive shaft, Figure 1.27) as well as primary suspension force elements. All the suspension force elements are described as external ones (the *External* second body) and have activated the *autodetection* attribute.

Motor casing is connected to the wheelset base by a revolute joint. Another revolute joint connects the rotor with the motor casing. Joint axes are parallel to the lateral axis Y.

#### 1.1.7.1.2. Traction torque

Driving torque is introduced as a joint torques in the revolute joint connecting the drive shaft with the motor casing, Figure 1.28. It is recommended to use the **List of forces** type of the joint torque. The list contains two elements. The first element of the **List of characteristics** type stands for modeling the torque value depending on the rotor angular velocity for different positions of the throttle. The second element of the **Expression** type includes a parameterized expression, and can be used for direct setting the traction torque if the locomotive model is included in a 3D model of a train.

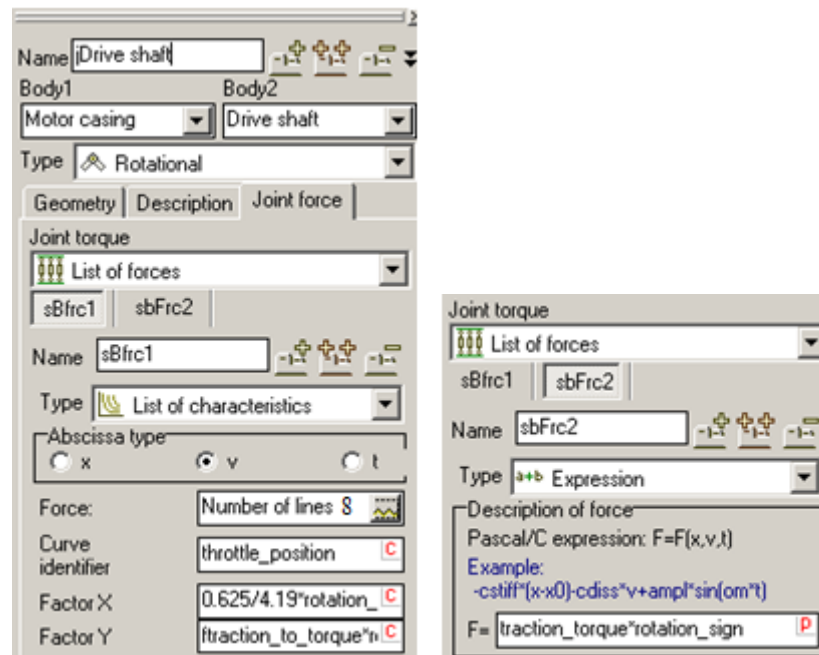


Figure 1.28. Setting traction torque

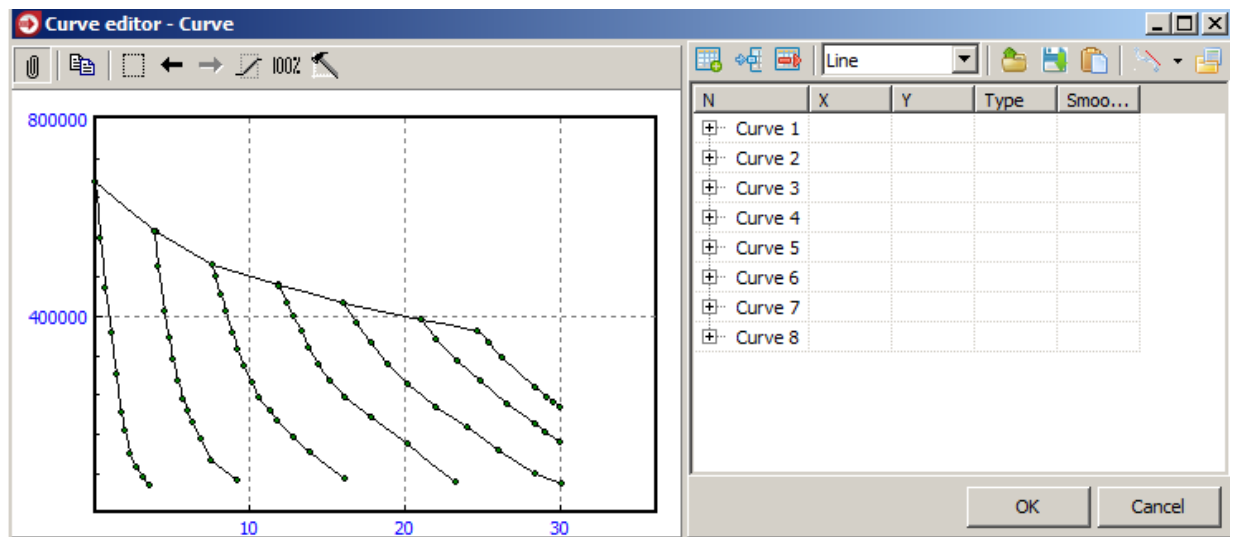


Figure 1.29. Total traction force versus unit speed

In our example the traction torque is defined by the set of force-speed curves where the force is the total traction force of the vehicle, Figure 1.31. The curves are described in the sequence corresponding to the throttle position number (1...N).

The following parameters should be set for additional description of the element (Figure 1.28, left).

- **Curve identifier:** the identifiers which numerical value 0...N sets the throttle position and selects one of the curve; zero value of the identifier means zero torque. **It is strongly recommended** to use the *throttle\_position* identifier, which automates the recognition of the identifier when the locomotive is included into a train 3D model.
- **Factor X:** the factor should differ from unity if abscissa in Figure 1.28 is not angular velocity of the rotor (e.g. the unit speed). It should be also used when the drive shaft rotates in the

negative direction, i.e. the corresponding joint velocity is negative. In the example in Figure 1.28 the expression  $0.625/4.19*rotation\_sign$  is used for the factor, which, first, converts the rotor angular velocity to the vehicle speed (0.625 is the wheel radius, 4.19 is the gear ratio), second, makes the velocity value positive by the *rotation\_sign* identifier if the rotor rotates in negative direction. Thus, the *rotation\_sign* should be +1 if the drive shaft rotates positively, and -1 in the opposite case (see Remark 2 in Sect. 1.1.7.1.4. "Computation of initial angular velocities by fixation file", p. 1-31).

- **Factor Y:** the factor should differ from unity if the ordinate in Fig. is not a torque applied to the drive shaft (e.g. the total traction force of the unit). It should be also used when the drive shaft rotates in the negative direction, i.e. the corresponding joint velocity is negative. In the example in Figure 1.28 the expression  $ftraction\_to\_torque*rotation\_sign$  is used. The *ftraction\_to\_torque* multiplier converts the total traction force into the torque on the rotor for a single motor. In our case this identifier defined by the following expression:  $ftraction\_to\_torque = 1/4*0.625/4.19$  (4 is the number of wheelsets, 0.625 is the wheel radius, 4.19 is the gear ratio). The *rotation\_sign* multiplier makes the sign of the torque equal to that of the joint velocity, which corresponds to the traction mode.

Now consider the direct setting the traction torque, which is necessary if the locomotive is included in the train 3D model, see. Figure 1.28. The torque is set by the expression  $traction\_torque*rotation\_sign$  where the *rotation\_sign* factor takes into account the rotor rotation direction and the *traction\_torque* identifier is equal to the torque value. The last identifier should be expressed in terms of the total traction force as

$$traction\_torque = traction\_force*ftraction\_to\_torque.$$

**It is strongly recommended** to use the *traction\_force* identifier for the total traction force to automate its detection if the locomotive is included in a train 3D model.

Finally, it is recommended to introduce the identifier *n\_throttle\_positions* with numeric value equal to the number of the throttle positions. The identifier should be added to the identifier list of the locomotive model (not in the subsystems).

1.1.7.1.3. Gearing

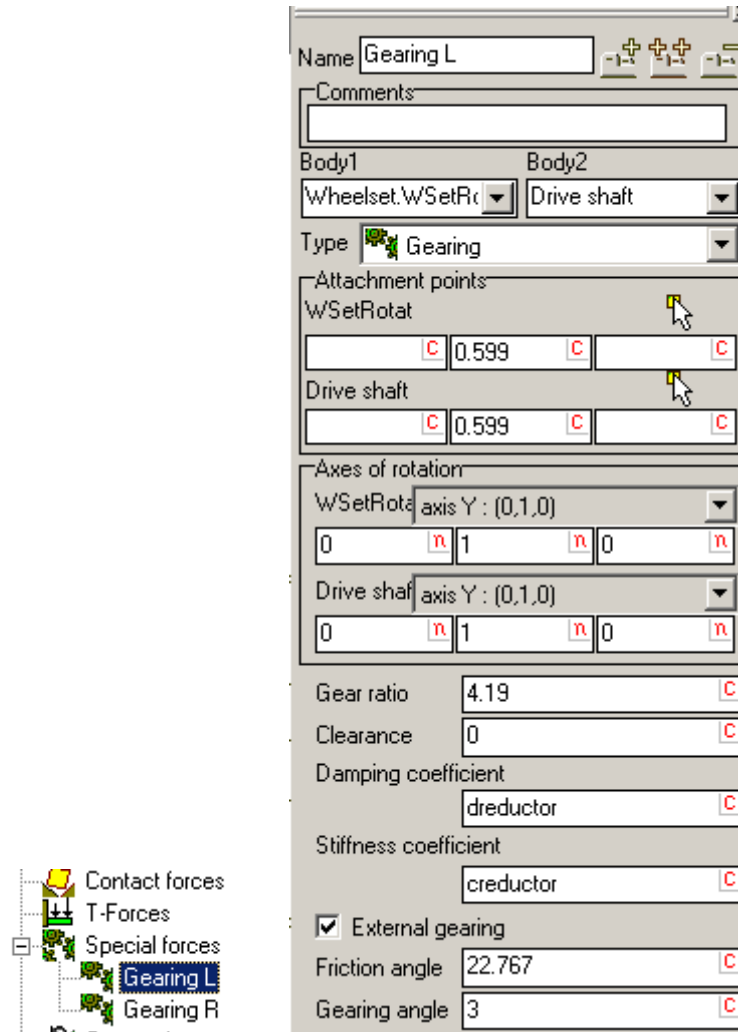


Figure 1.30. Gearing force element description

Model of a gearing is the necessary part of the drive system. A special force of the **Gearing** type is used for modeling the gearing. The force element connects the rotational part of the wheelset (the WSetRotat body) as the first body and the drive shaft as the second body. The following parameters specify the force element, Figure 1.30:

- **Attachment points** are coordinates of bull gear and pinion centers in system of coordinates of the wheelset and rotor respectively;
- **Axes of rotation** are unit vectors (0, 1, 0);
- **Gear ratio** is the bull gear radius divided by the pinion radius;
- **Clearance** is the value of possible teeth clearance due to wear (m);
- **Damping and stiffness coefficients** characterize the drive system compliance reduced to the gear teeth;
- The **External gearing** key must be checked.

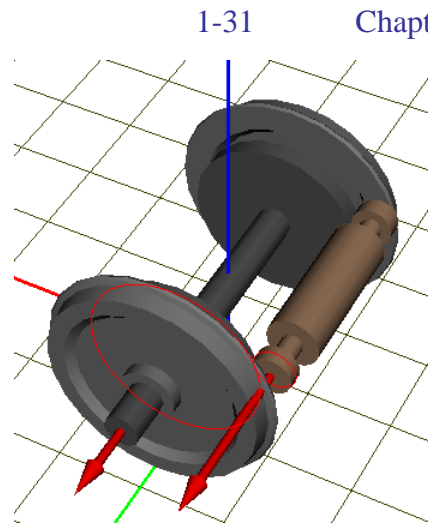



Figure 1.31. Visualization of the gearing force element

Switch the animation window mode to the visualization of a single element by the  button to control the correctness of geometrical parameters of the element. Red vectors in the window correspond to the gear axes, and red circles shows the central gear circles, Figure 1.31.

**Remark.** Use of gearing force elements requires creation of fixation file or setting constraints for initial velocities (see below).

#### 1.1.7.1.4. Computation of initial angular velocities by fixation file

Use of gearing requires setting the correct initial angular velocities of some bodies. Neglecting this requirement leads to large forces in the drive system, to lifting wheels from rails at the beginning simulations. One of the main tools for automatic computation of initial angular velocities (more precisely, joint velocities) is the creation by the user of the so-called **fixation file**, that informs the program which joint velocities must be kept without change by computation of initial values.

To illustrate the fixation of a coordinate, consider the drive system of one of the wheelset of locomotive VL80 with the axle-hung suspension of motor. Due to the gearing, the following equation takes place:

$$i\omega_w = -(\omega_r - \omega_m),$$

where  $\omega_m$  is the angular velocity of the wheelset rotation,  $\omega_m$  is the angular velocity of the motor casing relative to the wheelset base,  $\omega_r$  is the angular velocity of the rotor relative to the motor casing,  $i$  is the gear ratio. It is clear that this equation does not have a unique solution. The program cannot automatically choose which of three angular velocities should be computed from this expression and may choose any of them. Often this leads to a false solution: either the motor casing has an angular velocity equal to the wheelset rotation velocity multiplied by the gear ratio or all angular velocities are zeroes. The fixation file allows the program to solve the equation correctly. Namely, if change of angular velocities of wheelset and casing  $\omega_w\omega_r$  is forbidden (or the corresponding coordinates are *fixed*), the program automatically selects the only velocity  $\omega_r$

to be changed and compute it according to the current values of angular velocities of wheelset and casing. Of course, the casing initial angular velocity must be zero.

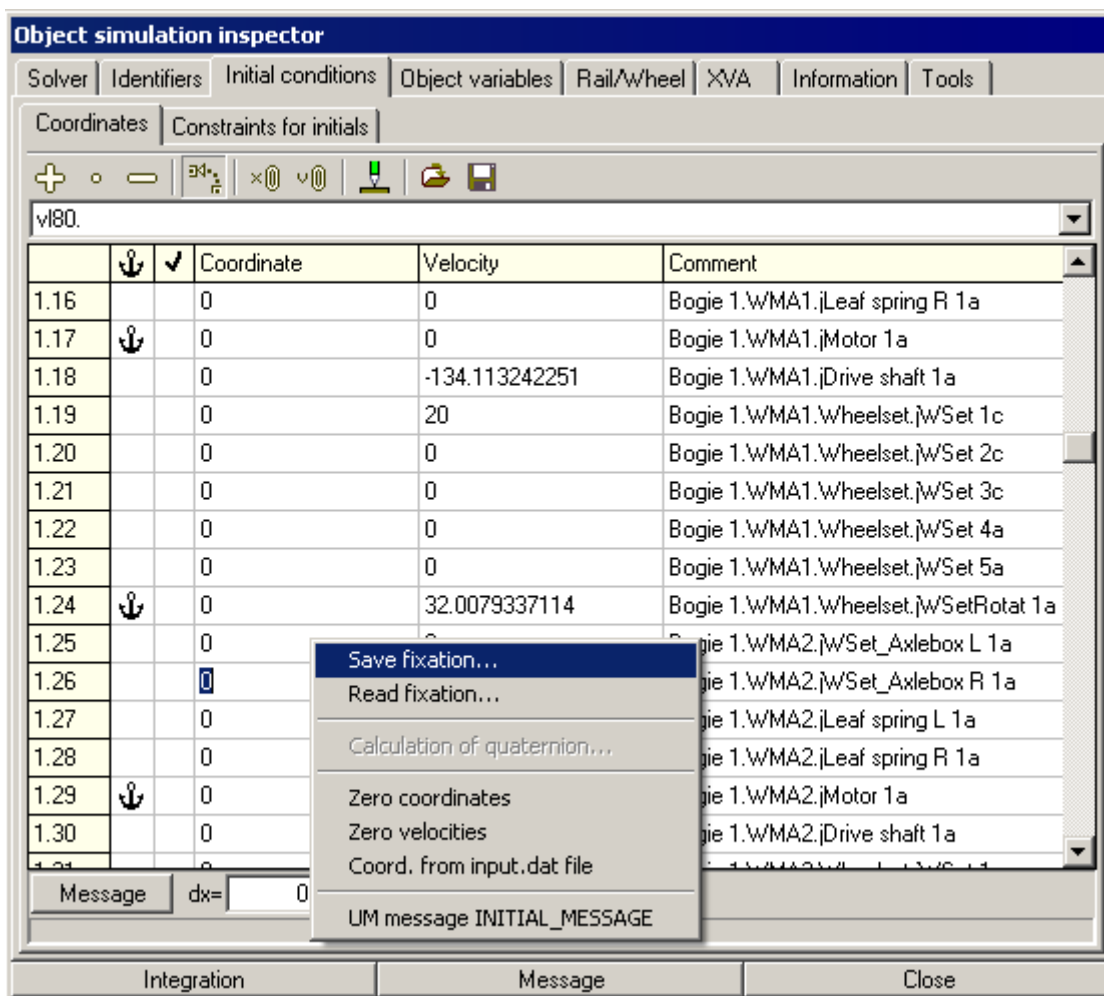



Figure 1.32. Fixation of coordinates and creation of fixation file

Fixation of a coordinate means that the coordinate and its time derivative (velocity) cannot be changed during computation of initial values of coordinates and velocities. The fixation file contains the list of fixed coordinates. Thus, in the case of locomotive VL80 it is necessary to fix coordinates of wheelsets corresponding to rotation of wheelsets about its lateral axis (joint coordinates in joints **jWSetRotat**) as well as coordinates in revolute joints, setting rotation of motor casings about wheelset axes (joints **jMotor** in the figure). The fixation file is created in the **UM Simulation** program on the **Initial conditions | Coordinates** tab of the object simulation inspector, Figure 1.32.

- Click the upper of two **Message** buttons. After this action, the program automatically computes angular velocities of wheelsets as well as longitudinal velocities of some of bodies (e.g. car body). This stage of creation of fixation file is not necessary, but it allows the user to see the current values of wheelset angular velocities which must be fixed.
- Fix the necessary coordinates by clicking on cells of the table column marked by the image on the top.

- Save fixation to the file with name of the model in the directory of the model (the default file name and path).
- If the first item of this list is executed, click the  button to compute the initial velocities and to verify the correctness of the fixation (angular velocities of rotors must be computed).

**Remark 1.** For some types of traction motor suspensions (e.g. in case of quill drive) the fixation file is not sufficient for computation of all the necessary initial velocities. In such cases additional constraints on initial velocities must be used, see the next section of the manual.

**Remark 2.** The sign of the rotor angular velocity (i.e. the sign of the corresponding joint velocity) obtained from Figure 1.32 should be used for specifying the *rotation\_sign* identifier value, see Sect. *Traction torque*.

#### 1.1.7.1.5. Computation of initial angular velocities by constraints on initial values

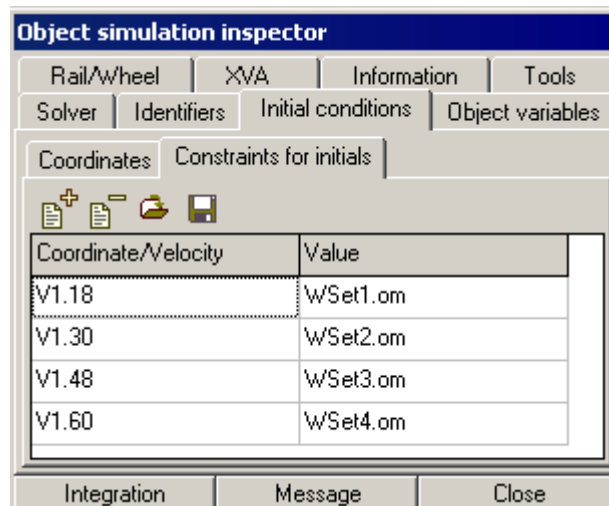


Figure 1.33. Tools for setting constraints for initial values

Constraints on initial values are an alternative of the fixation file, and give the user more opportunities to control initial conditions.



A constraint is an equation of one of the following two types

$$x_i = X_i,$$

$$v_i = V_i,$$

where  $x_i$  is the joint coordinate,  $v_i = \dot{x}_i$  is the first time derivative of the joint coordinate (joint velocity). The right hand sides are arbitrary expressions created with the wizard of variables (see [Chapter 4](#), Sect. *Wizard of variables*).

The following steps create a new constraint.

- Add a row to the table of constraints by the  button.
- Open the wizard of variables with the help of the **Tools | Wizard of variables...** menu command or by the  button on the tool panel.

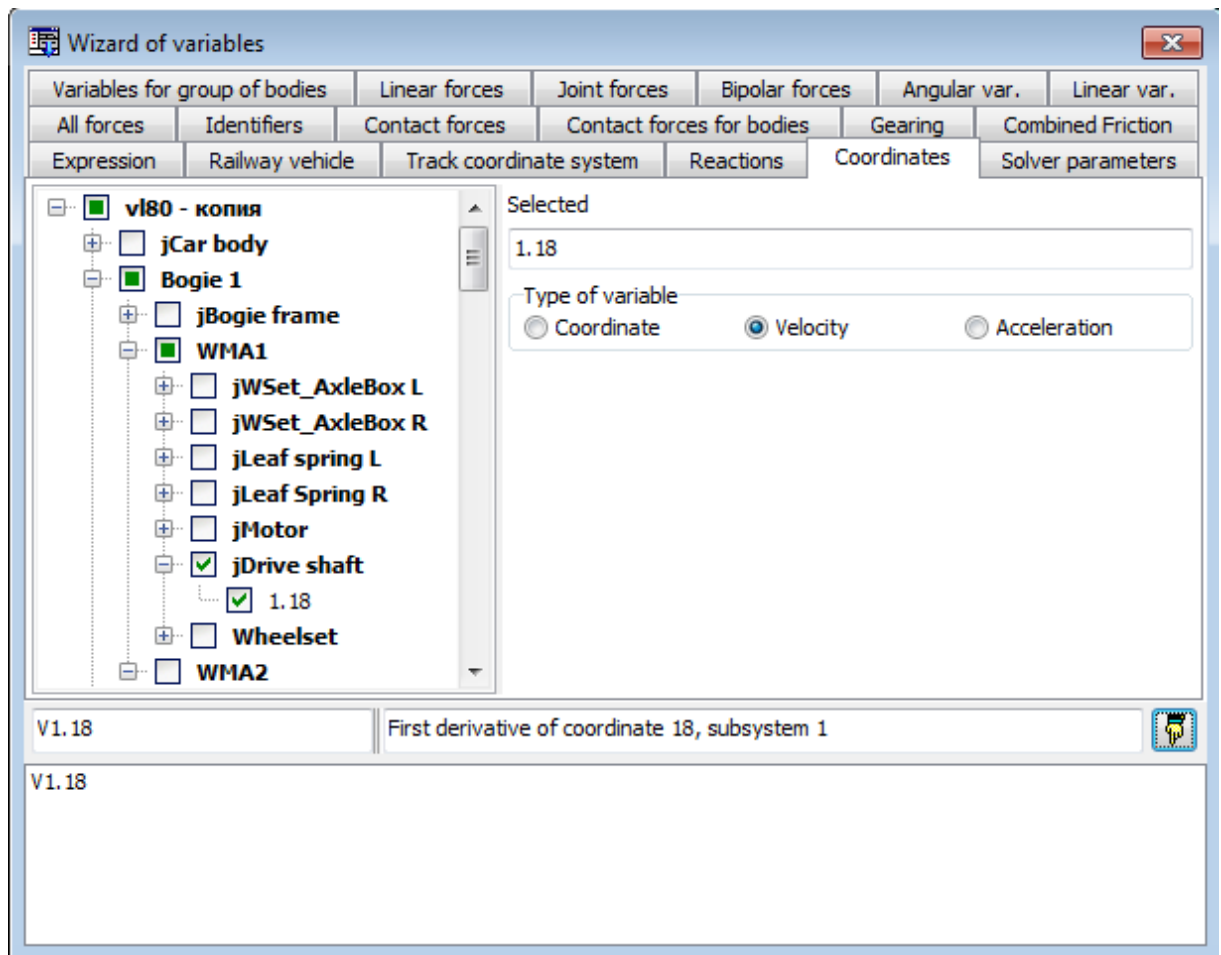



Figure 1.34. Creating variable: velocity of drive shaft rotation relative to the motor casing

- Create a variable corresponding to a joint velocity for setting the initial value: open the **Coordinates** tab, select the necessary joint coordinate in the list located in the left part of the wizard. Set the variable type **Velocity** and send in to the container by the  button, Figure 1.35.
- Drag the variable by the mouse and drop it into the left cell of the constraint data row, Figure 1.34.

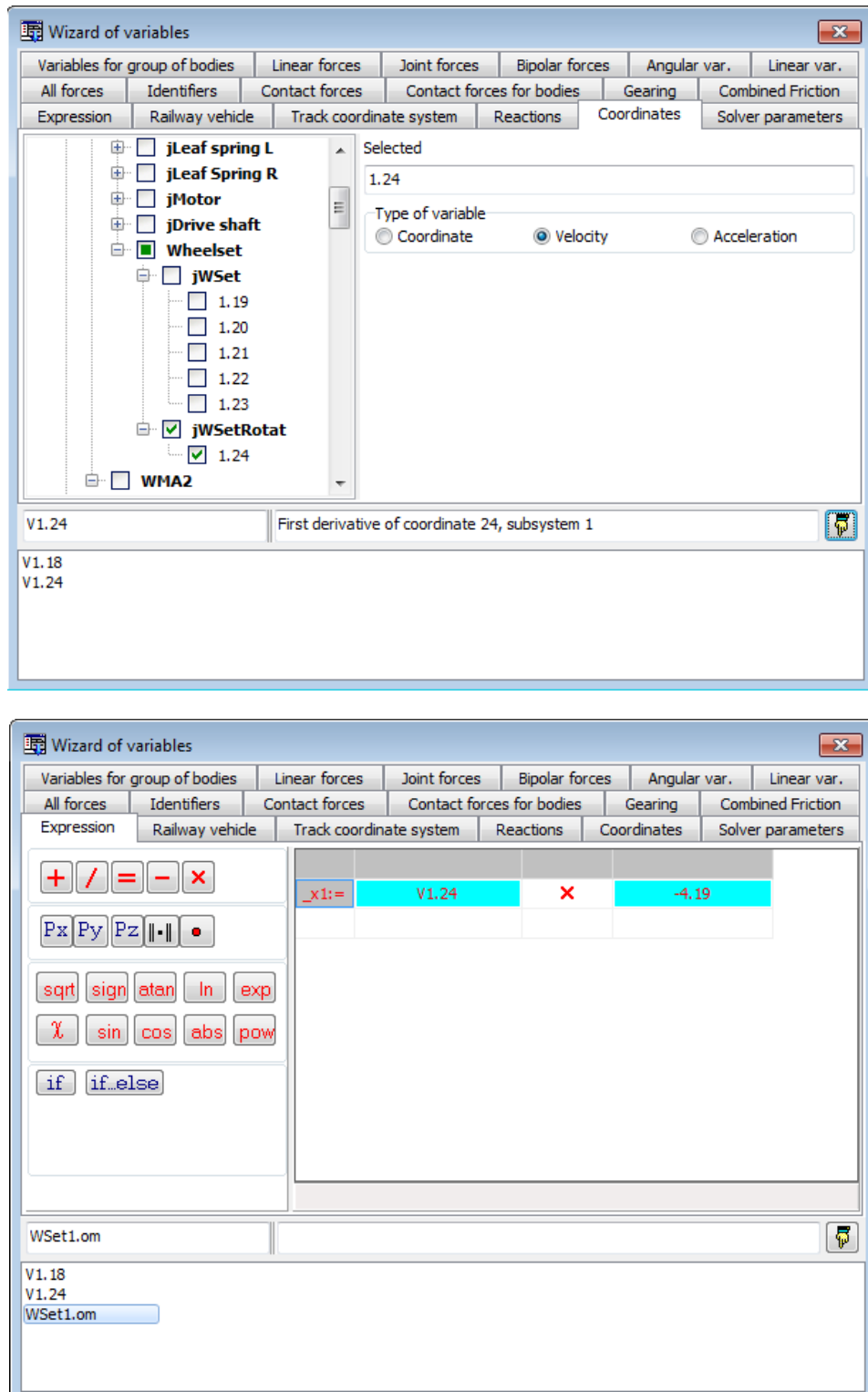





Figure 1.35. Creation of variables: angular velocity of wheelset and computed value of drive shaft angular velocity

- Create a variable corresponding to the desired value of the velocity and put it in the right cell of the constraint row. Let us consider the following example. Let the rotor angular velocity is equal to the wheelset angular velocity multiplied by the gear ratio 4.19. It is necessary to take into account that the wheelset and the rotor rotate in opposite directions, i.e. the joint velocities have different signs.

- Create a variable corresponding to the angular velocity of the wheelset (more precisely, the derivative of the angle of rotation); this is the joint velocity in the joint **jWSetRotat**.
  - Open the **Expression** tab of the wizard. Add a multiplication operation by the  button, Figure 1.35.
  - Set the wheelset angular velocity by the mouse as the first operand; enter the number '-4.19' (negative!) by the clipboard as the second operand.
  - Set the variable name instead of the default value *Expression*.
  - Send the variable into container by the  button and drag it into the constraint cell, Figure 1.33.
- When all the constraints are specified, save them in a file in the object directory. If constraint file has the name of the object, the constraints are automatically loaded at each load of the object in the simulation module. Constraints are used for automatic computation of initial conditions right before the start of the simulation process.
  - To verify the correctness of the constraints, open the **Initial conditions | Coordinates** tab of the inspector, run the upper command **Message**, and compute the initial values by the  button. Compare the computed values of velocities with the desired ones.

### 1.1.7.2. Example of development and analysis of locomotive model

Consider basic principles of development of a locomotive with two COCO bogies, Figure 1.36. Each of the bogie models contains three Wheelset-Motor assemblies (WMA). In a WMA, the motor is rigidly fixed relative to the frame, and the traction reducer is connected to the wheelset by a rotational joint.

The ready model of the locomotive is located in the directory [{UM Data}\SAMPLES\Rail\\_Vehicles\Co-Co.](#)

#### 1.1.7.2.1. Development of model

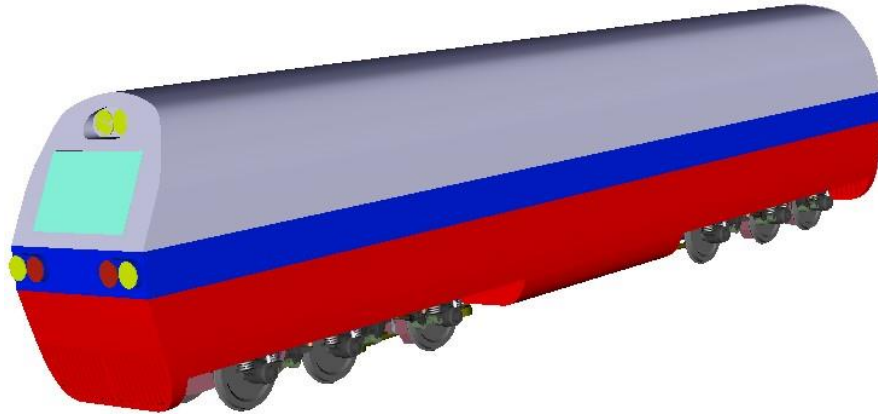


Figure 1.36. Locomotive model

##### 1.1.7.2.1.1. Development of wheelset-motor assembling

Let us consider some details of the WMA model. Run **UM Input** and open the model or click on the hyperlink [{UM Data}\SAMPLES\Rail\\_Vehicles\Co-Co.](#) Open the subsystem *Bogie\_1*, and then the subsystem *Wheelset\_motor\_assembling\_1*, Figure 1.37.

The WMA model contains several rigid bodies:

- two axle-boxes 3 (*Axle-box L* and *Axle-box R*),
- *Reduction gearbox 7*,
- *Traction motor 5*,
- *Rotor*,
- *Gear wheel*,
- *Cardan shaft*.

The wheelset is the standard subsystem, Sect 1.1.3. "*Wheelset and single wheel*", p. 1-9.

Consider the structure of the model in more details, Figure 1.38. The traction motor casing is rigidly fixed to the bogie frame. The rotor is connected to the motor casing by the rotational joint *jTraction motor\_Rotor*. The cardan shaft has two degrees of freedom relative to the rotor; see the joint *jRotor\_Cardan\_shaft* in which four degrees of freedom are disabled. The rubber annulus coupling is modeled by a bushing force element *Rubber-cord coupler*. The shaft transmits the traction torque by the coupling to the *Gear wheel* of the reducer. The *Reduction gearbox* is con-

nected to the wheelset by the rotational joint  $jWSet\_Reduction\ gearbox$  and suspended to the frame by the linear bipolar force element  $Rod$ .

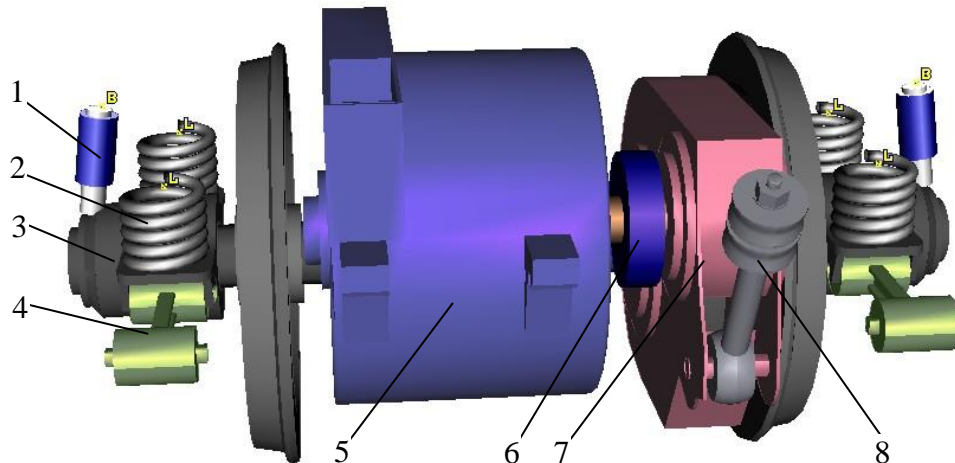


Figure 1.37. Model of wheelset-motor assembly: damper (1), primary suspension (2), axle-box (3), traction rod (4), traction motor (5), elastic coupling (6), reduction gearbox (7), reducer suspension rod (8)

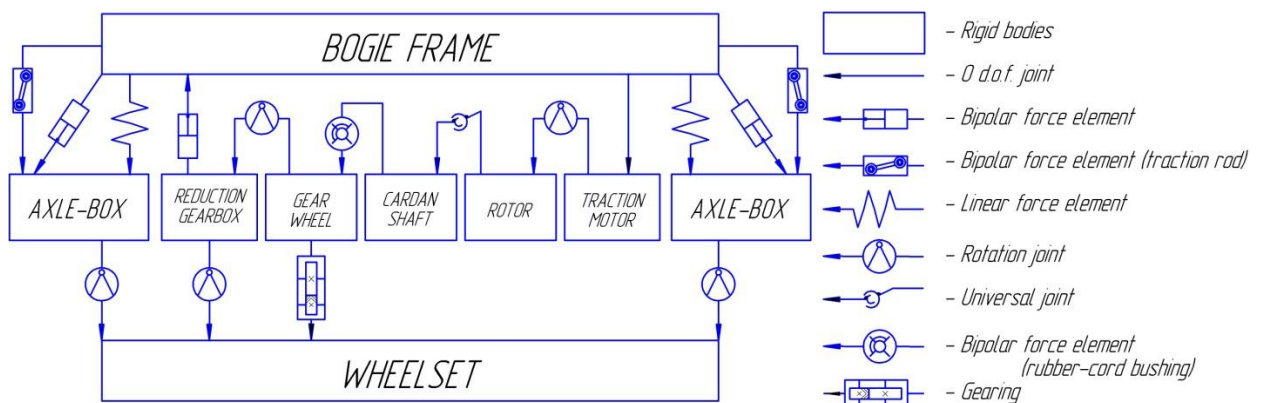


Figure 1.38. Structure of WMA

**Remark.** Simple images are created directly in UM. Advanced images can be imported from CAD programs. In the case of our locomotive models some images are created in UM (Rotor, Gear wheel, Cardan, Spring, Damper), other images are imported from CAD.

The joint  $jBase\_Traction\ motor$  with 6 degrees of freedom sets the position of the motor casing relative to SC0. This joint will be automatically removed when the motor will be fixed to the frame in the upper subsystem (the bogie model).

**Remark.** It is recommended to use the following names of joints for the better identification: «j» + “Name of first body” + “\_” + “Name of second body”.

A joint of the generalized type introduces the rotational degree of freedom of the rotor relative to the motor casing. A rotational joint could be used as well. The traction torque is described by the joint torque of the points (numeric) type.

The cardan joint is created with the “6 d.o.f” joint *jRotor\_Cardan\_shaft*. The joint is specified by the joint points; four degrees of freedom of six are locked; rotations about the X and Z axes are allowed.

The rotational joint *jReduction\_gearbox\_Gear\_wheel* describes rotation of the gear relative to the reducer casing, which rotates about the wheelset axle (*jWSet\_Reduction\_gearbox*).

Finally, the joints *jAxle-box L\_WSet* and *jAxle-box R\_WSet* connect axle-boxes to the wheelset axle. These joints differ for the front and rear WMA on the one hand, and for the middle WMA on the other hand. The joint are rotational in the first case (Figure 1.39), and the joints for the middle WMA has 15 mm gap relative to the axle in the lateral direction (Figure 1.40).

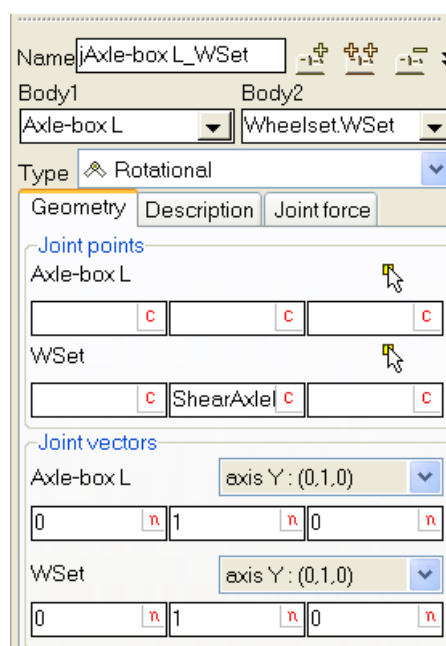


Figure 1.39. Joint *jAxle-box L\_WSet* for the front and read WMA

Open the *Wheelset\_motor\_assembling\_2* subsystem to see the description of the joint *jAxle-box L\_WSet* with two degrees of freedom. The joint type is *generalized* and it has three elementary transformations (ET, see [Chapter 2](#), Sect. *Generalized joint*):

- *tc* (translation constant) sets the lateral position of the box on the axle *ShiftAxleBox* = 1,085 m, Figure 1.40a;
- *tv* (translational degree of freedom) specifies the lateral motion of the box;
- *rv* (rotational degree of freedom) introduces the rotation of the box relative to the axle.

Let us consider the ET of the *tv* type in more details.

The tab **Force/Torque** contains the description of forces acting on the box by its motion relative to the axle in the lateral direction. The type of torque is the *List of forces* with two elements:

- the *Bump stop* element describes the gap, Figure 1.40b;
- the *AxleBox\_Friction* element introduces the friction force, Figure 1.40c.

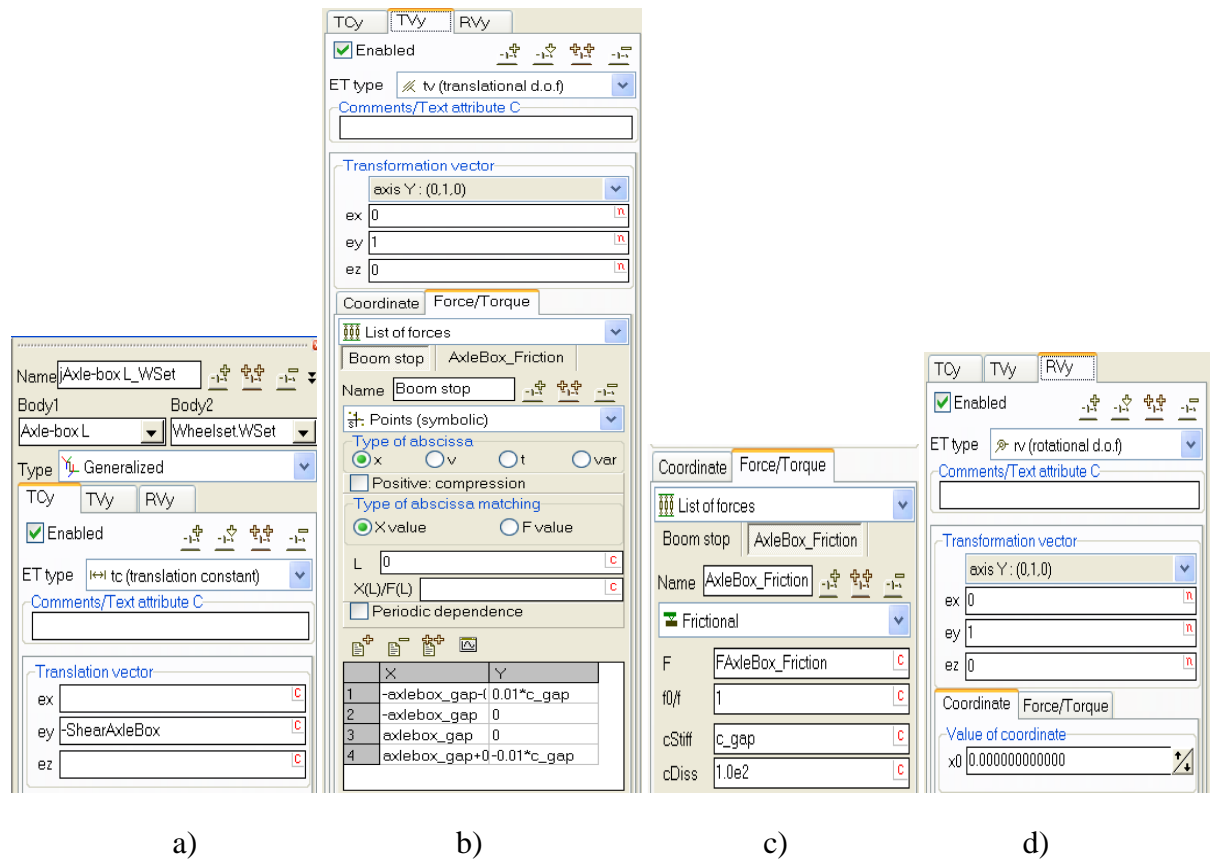


Figure 1.40. Description of joint jAxle-box L\_WSet for the middle WMA

Bipolar force elements are used for modeling the reducer support and the inclined primary damper. It is important that the second body in the force elements is *External* (the second body will be assigned in the bogie subsystem), and the *Autodetection* mode is checked, see [Chapter 3](#), Sect. “*Interconnection of subsystem. Use of external elements*” for more details.

A bipolar force of the *Linear* type describes the reducer suspension rod. An image is assigned to the rod for the visualization.

Characteristics of the hydraulic dampers *Damper L* and *Damper R* (Figure 1.42) are described by the expression

$$-(d\_Axlebox\_1part*v^2*heavi(v\_damper-abs(v))+heavi(-v\_damper+abs(v))*(d\_Axlebox\_1part* * v\_damper ^2+d\_Axlebox\_2part*(abs(v)- v\_damper)))*sign(v).$$

See the list of identifiers for the values of the identifiers in the expression.

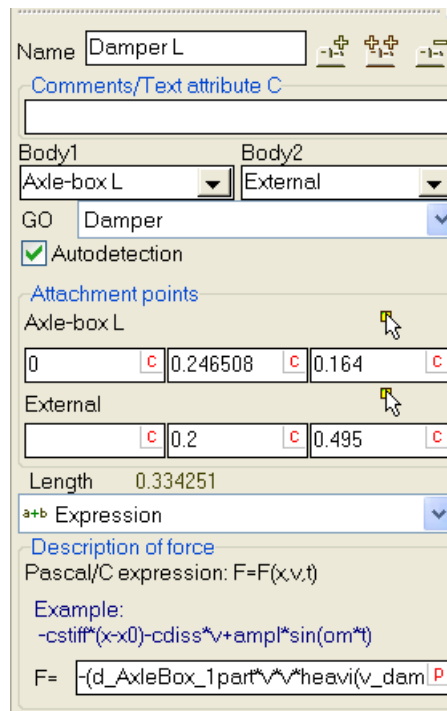


Figure 1.41. Primary damper

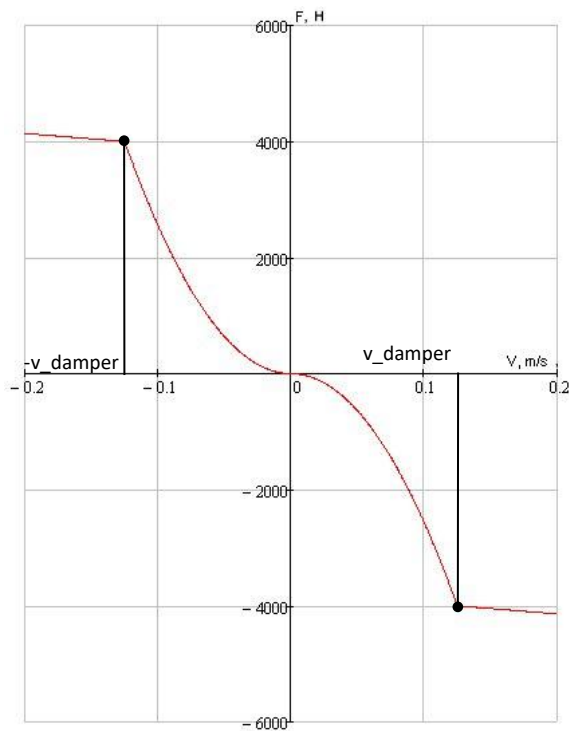


Figure 1.42. Characteristics of primary damper

Springs and traction rods are often modeled by *Linear force* elements.

The *Rubber-cord coupler* is modeled by a *Special force* element of the *Bushing* type. Characteristics of the coupler are nonlinear. If the deflection is more than 0.06m, the stiffness of the bushing increases.

**Remark.** It is recommended to parameterize all elements of force description for the possible modification during the simulation.

### 1.1.7.2.1.2. Development of bogie model

Consider some features in description of the bogie model. Open the *Bogie\_1* subsystem. Development of the bogie model starts with creation of images. Image of a frame is imported from a CAD program, and all other images are developed in UM.

The subsystem includes two bodies *Frame* and *Traction rod*.

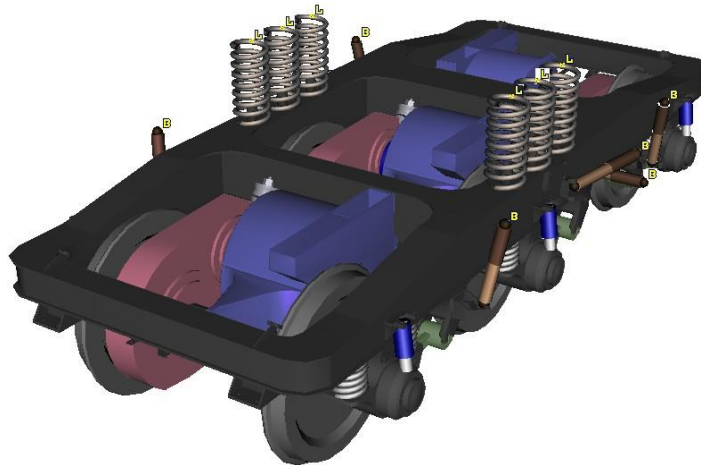



Figure 1.43. Bogie model

The joints *jBase\_Frame*, *jBase\_Traction* introduce six degrees of freedom of the frame and traction rod relative to SC0.

The model contains three included subsystems WMA, Sect. 1.1.7.2.1.1. "Development of wheelset-motor assembling", p. 1-37. Please note that the front and the rear WMA are identical, whereas the middle differs in description of axle-boxes degrees of freedom. The button  is recommended for copying the subsystem. Longitudinal positions are parameterized by the identifiers *x\_Wheel\_1*, *x\_Wheel\_3*, *r\_Wheel*. The rear WMA is rotated on 180 degrees about the vertical axis.

The *jFrame\_Wheelset\_motor\_assembling\_1*, *jFrame\_Wheelset\_motor\_assembling\_2*, *jFrame\_Wheelset\_motor\_assembling\_3* joints fix motor casing to the frame. After introduction of these joints, the fictitious joints *jBase\_Traction motor* are ignored for all of the three WMA. For fixation joints we have used the *generalized* type of joints. The position of the casings are set by the elementary transformations of the *tc* type. An additional rotation about the *Z* axis is assigned to the rear motor casing with the transformation of the *rc* type.

Bipolar forces *DamperZ 1R*, *DamperZ 2R*, *DamperZ 1L*, *DamperZ 2L* describe the vertical dampers; lateral dampers are *Damper Y L* and *Damper Y R*, and yaw dampers are *Damper X L* and *Damper X R*. In all three cases we have used the simplified linear model of the dampers. The second body for all forces is selected *External*, and the **Autodetection** key is checked.

The linear force elements describe six secondary suspension springs for a bogie. The stationary force in springs is vertical; it is equal to the product of the spring constant on the static spring

deflection. The stiffness matrix specifies the spring characteristics. Second body is the **External** one and **Autocomputation** for the second body key is checked.

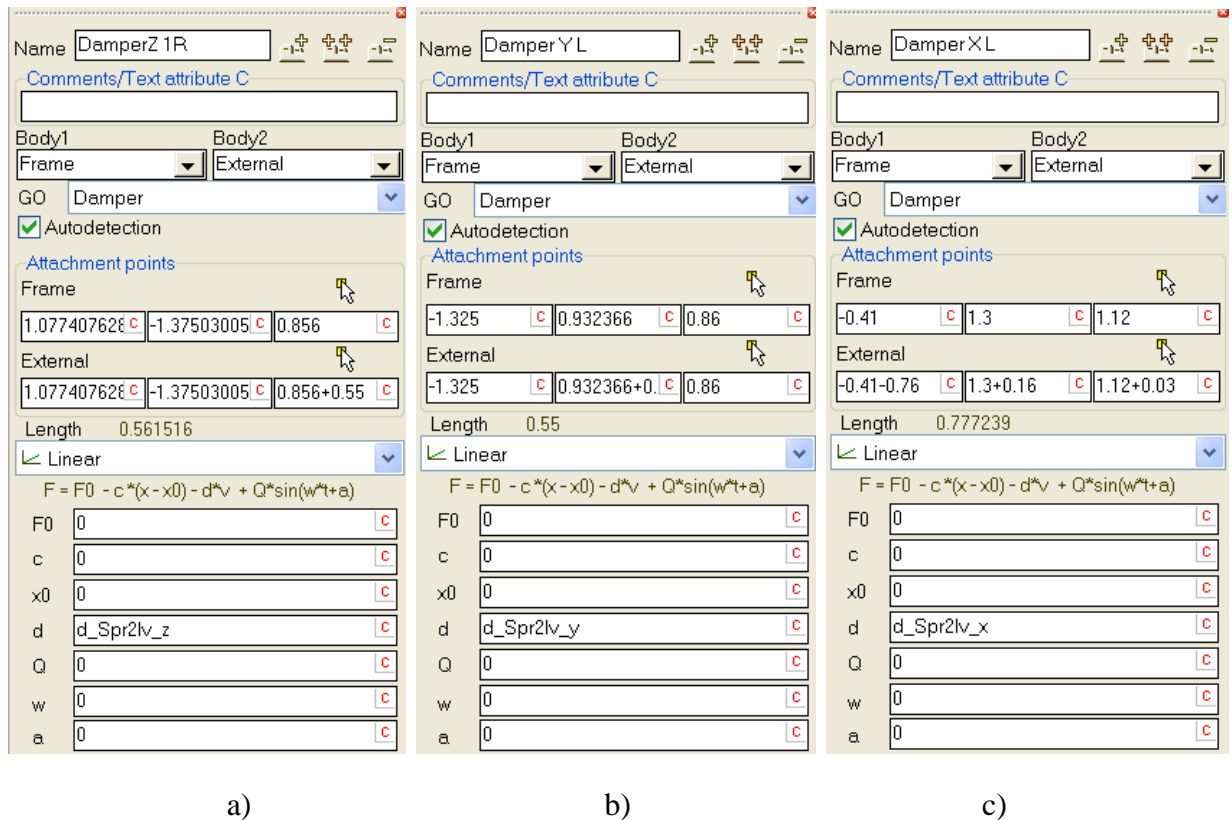


Figure 1.44. Secondary dampers: vertical (a), lateral (b) and yaw (c)

An inclined traction rod is used in the bogie model for transfer of the traction efforts. The traction rod is connected with the bogie frame and car body by special force elements of the *bushing* type *Joint\_traction\_1* and *Joint\_traction\_2*. The second bushing is the **external** one. The **Autodetection** key is checked for both of the bushings.

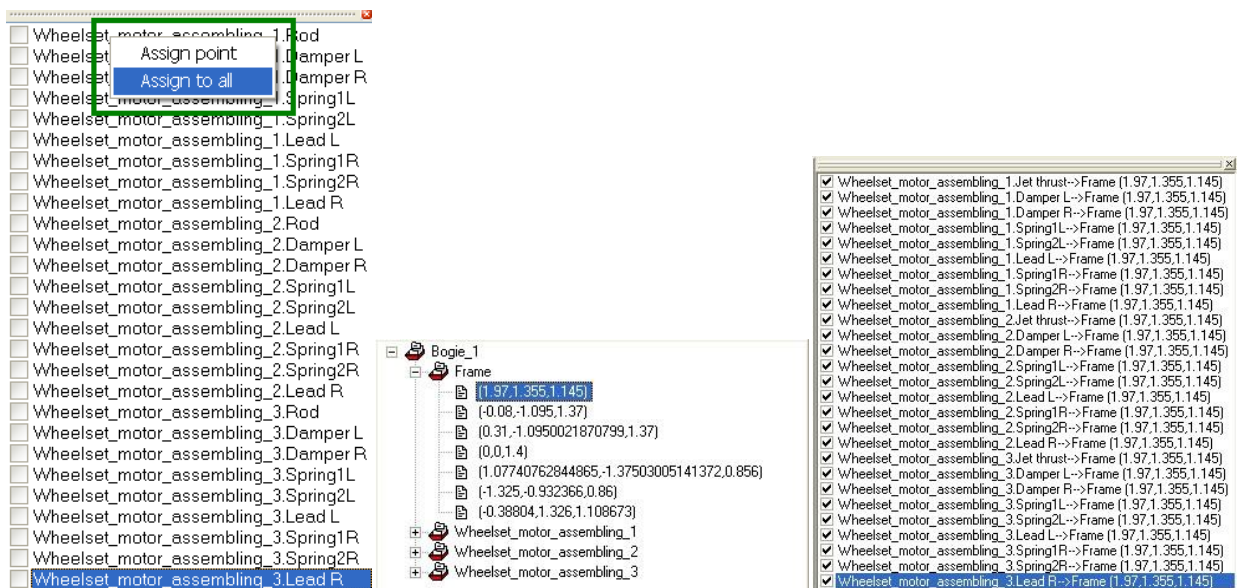


Figure 1.45. Assignment of connection points with the “Assign to all” menu command

To finalize the bogie model, we must to assign connections to all of the external elements introduced in the WMA subsystems. Open the **Connections** tab of the inspector by the item in the element tree. In our model all external elements are described in the autodetection mode. All external elements are connected to the bogie frame, and the assignment of this body as the second one for the force elements can be done by the “**Assign to all**” command of the pop-up menu. It is necessary to have at least one connection point in description of the frame body. Select one of the external elements in the list by the mouse and call the pop-up menu, Figure 1.45. Select the “Assign to all” command and choose one of the connection point for the frame body. After this operation the frame body will be assigned to all external elements in the WMA subsystems, and coordinates of connection points to the frame body will be evaluated automatically.

See [Chapter 3](#), Sect. “*Interconnection of subsystem. Use of external elements*” for additional information.

### 1.1.7.2.1.3. Finalization of locomotive model

Let us consider the full locomotive model. Firstly, the new model must be created or the bogie model must be converted into a subsystem.

After that, the car body image must be created. We have done it in a CAD program and imported to UM.

**Remark.** Each graphic element in UM has one color only. It is recommended to develop in CAD a car body image consisting of several parts to assign them different colors. In our case we have used six parts, which are joint in a single GO as a collection of references to the part images Figure 1.46.

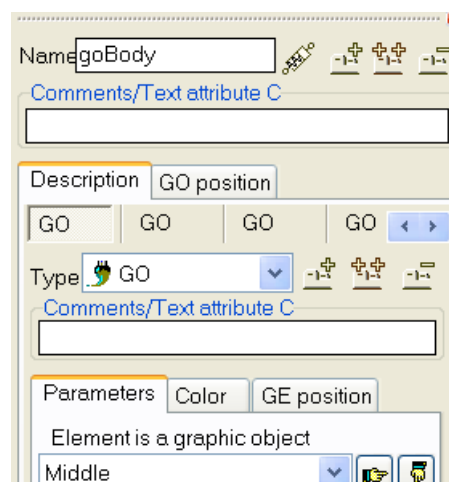


Figure 1.46. Image as a collection of part images

The model includes one body corresponding to the car body, and one joint with six degrees of freedom *jBody* introducing coordinated of the car body relative to the *SC0*.

If the bogie is not converted into a model subsystem, it must be added to the locomotive model as an included subsystem *Bogie\_1*. The second bogie *Bogie\_2* is created as a copy of the first one. The first bogie longitudinal position is set by the expression  $vehiclebase/2$ . The second

bogie position has the opposite sine, and in addition the subsystem must be rotated about the Z-axis on 180 degrees.

Connection of external elements with the car body is done in the same manner as in the case of the bogie model, Sect. 1.1.7.2.1.2. "*Development of bogie model*", p. 1-42. The **Summary** tab in inspector informs us about possible errors in the model.

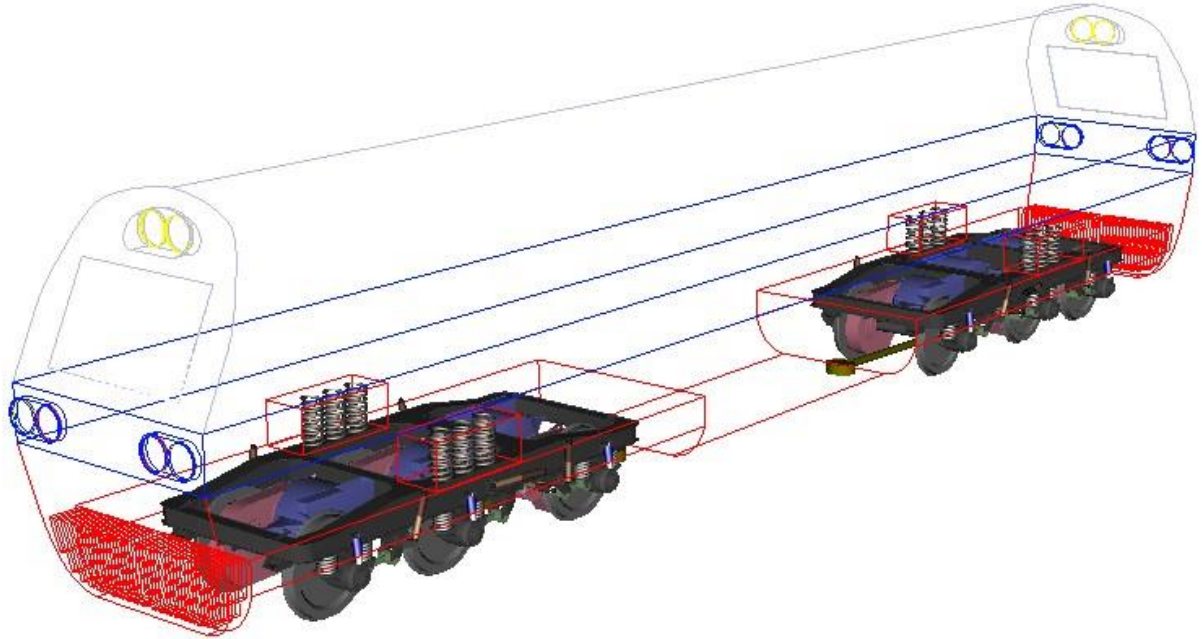


Figure 1.47. Model of locomotive

### 1.1.7.2.2. Simulation of locomotive dynamics

Open the locomotive model in the simulation program.

#### 1.1.7.2.2.1. Creation of fixation file

Notion of the fixation file is introduced in Sect. 1.1.7.1.4. "*Computation of initial angular velocities by fixation file*", p. 1-31. For correct automatic computation of initial values of angular velocities in the traction system of locomotive we must fix several coordinates. The kinematic scheme of the WMA is shown in Figure 1.48.

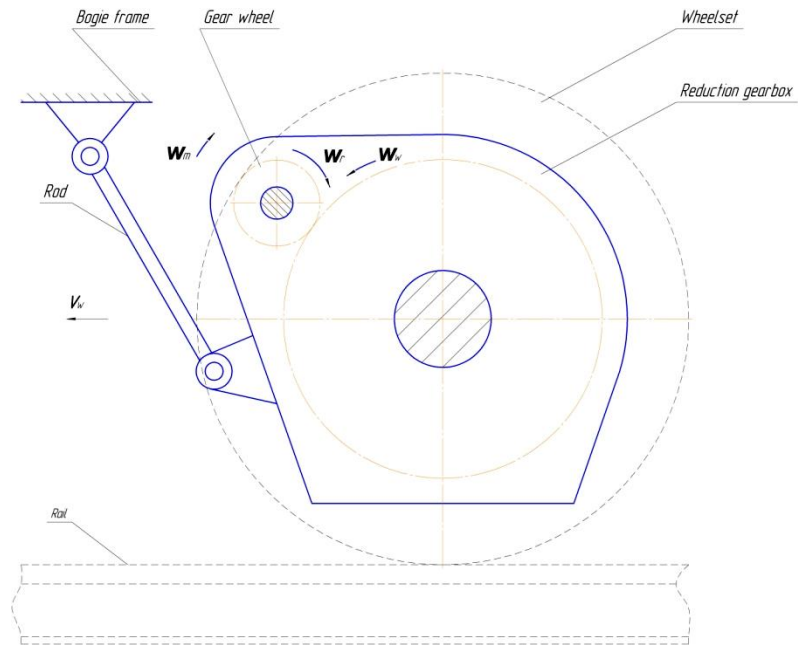



Figure 1.48. To computation of angular velocities

According to the scheme, we must **fix** angular coordinate of the reducer casing as well as the wheelset coordinate for rotation about the wheelset axle in each WMA.

The fixation file is created with the help of the **Initial conditions** | **Coordinates** tab of the inspector, Figure 1.49. Use the left mouse button to fix the necessary 12 coordinates by the mark  in the first column of the coordinate table. The file with the model name is created corresponding by the pop-up menu command.

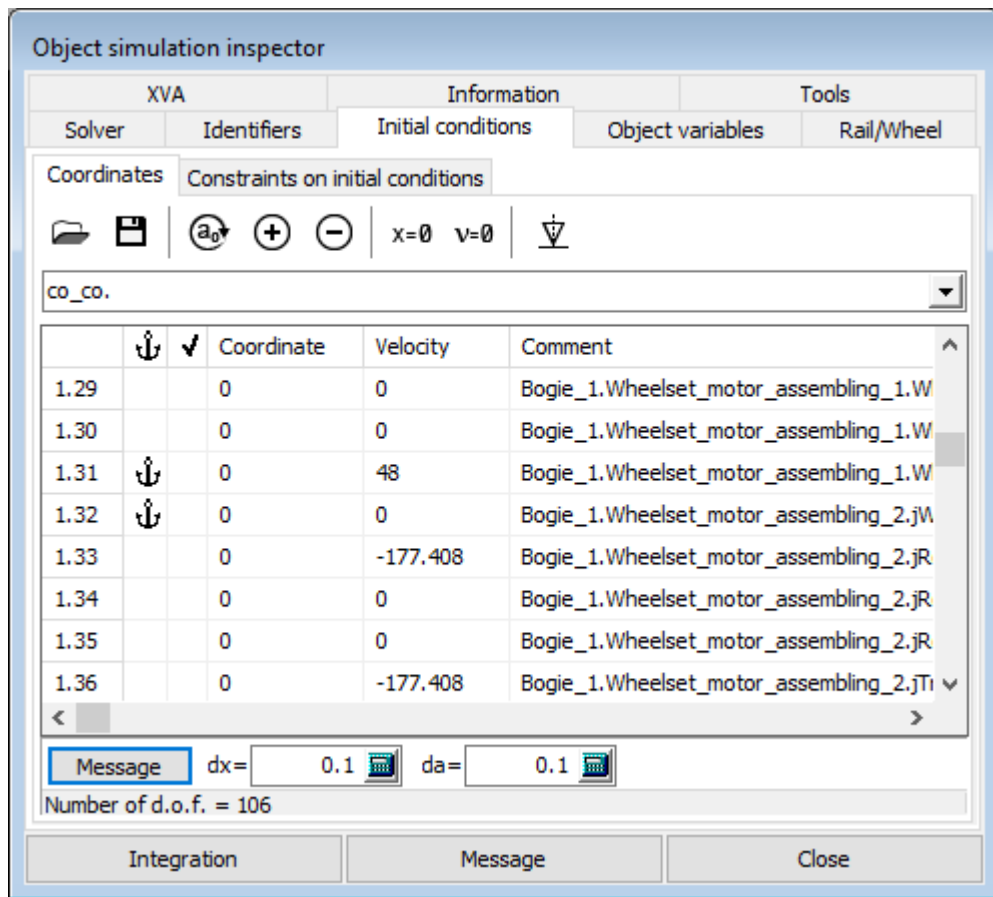


Figure 1.49. Fixed coordinates

### 1.1.7.2.2.2. Creation of constraints on initial conditions

Fixation file allows computing correctly the gear angular velocities, but not the rotor rates. In addition to the fixation file, we must create six constraints on initial conditions to make the rotor angular velocity equal to that of the gear wheel, see Sect. 1.1.7.1.5. *"Computation of initial angular velocities by constraints on initial values"*, p. 1-33.

The variables corresponding to the angular velocities of rotors and gear wheels are created in the Coordinates tab of the Wizard of coordinates. After that, the variables corresponding to the rotor rates are dragged into the first column of the Constraints on initial conditions tab of inspector, Figure 1.50, and angular velocities of the gears are dragged into the second column, Figure 1.51. The list of constraints must be saved in the file with the name of the model.

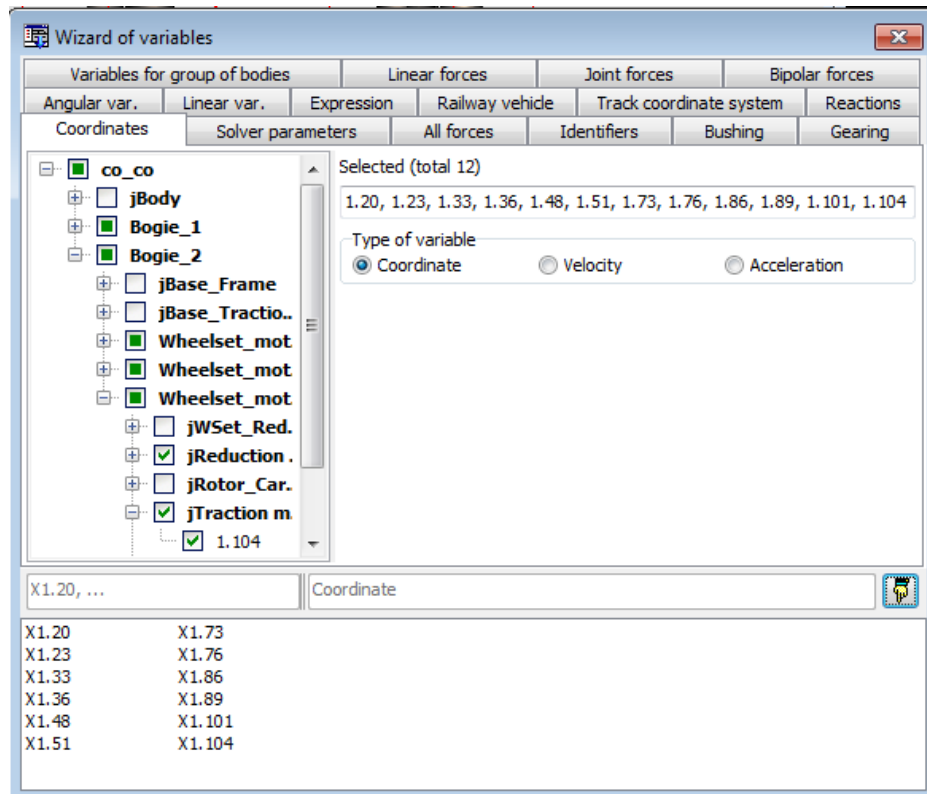


Figure 1.50. Development of variables – velocities

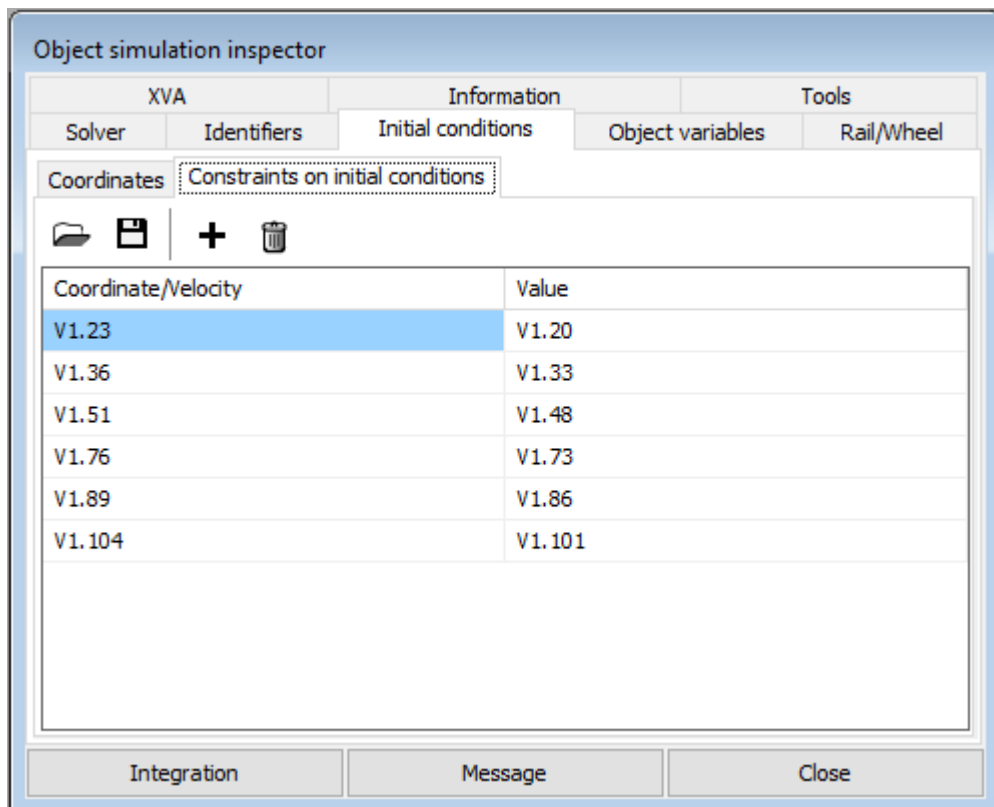



Figure 1.51. List of constraints

To verify the correctness of the fixation file and the constraints, open the **Initial conditions** | **Coordinates** tab, click on the upper Message button to assign initial velocities to wheelsets,

frames and other bodies, and click  button to compute other initial velocities the according to fixation and constraints. Verify the results. For instance, rates of the rotors and wheels must be equal, Figure 1.52.

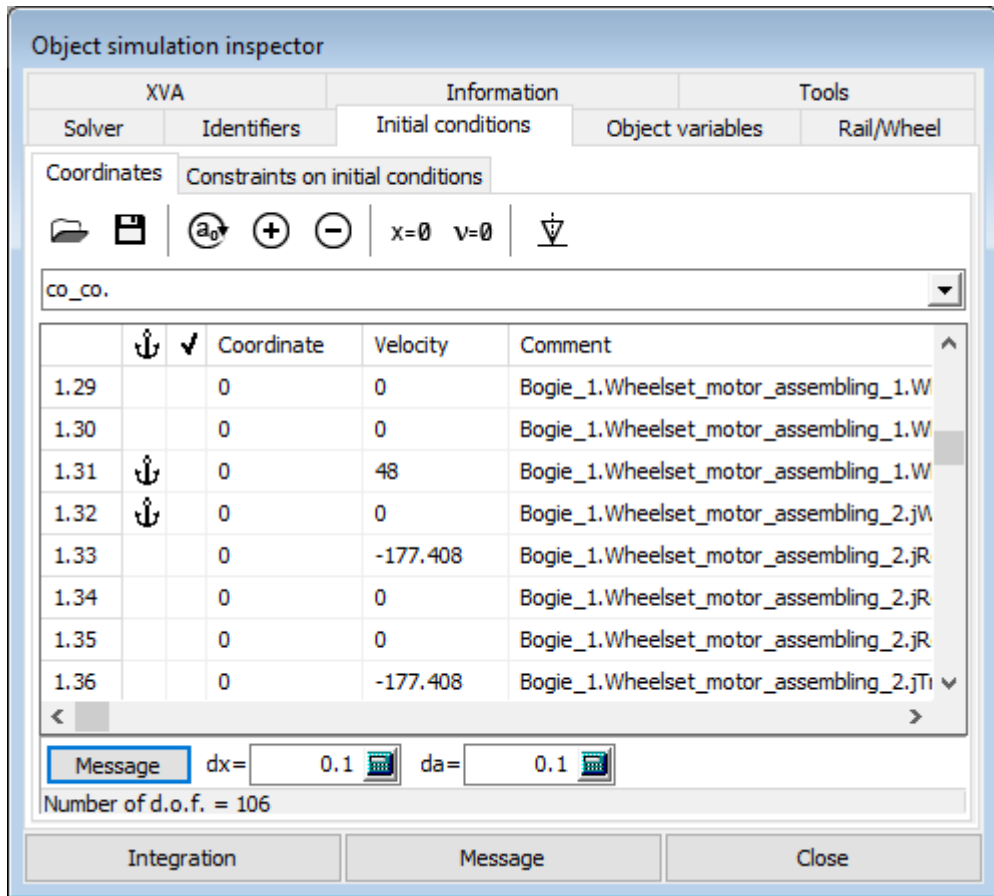


Figure 1.52. Result of computations of initial conditions

**1.1.7.2.2.3. Test for force start values**

A very useful tool is available for verification of correctness of force description in the model. The tool computes the force values at start of the simulation. Usually this test is run for just developed models to find possible errors in description of force elements.

To run the test, open the **Tools | Test** tab of the inspector and click the **Compute** button. The program computes the forces and displays the force component values on SC0, Figure 1.53. The user compares the values with the expected ones.

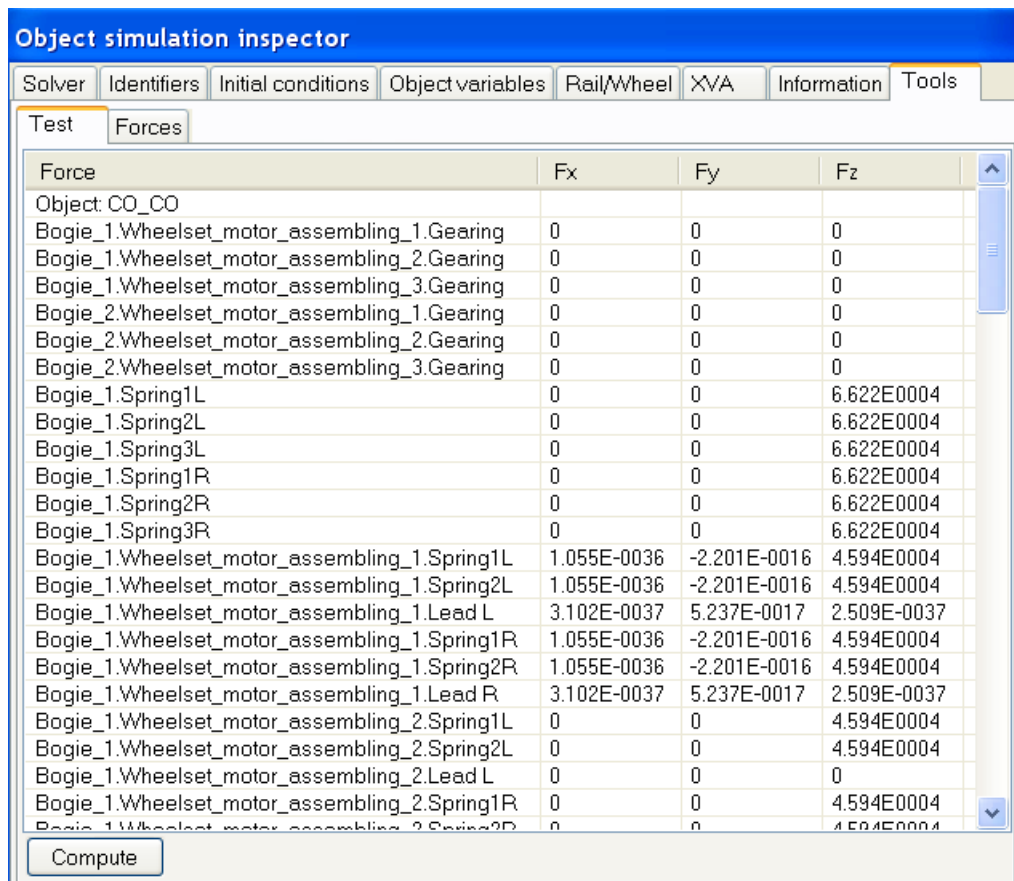


Figure 1.53. Test for forces

**1.1.7.2.3. Analysis of some locomotive performances**

The developed model allows analyzing dynamic performances of the locomotive in tangent sections and curves, with constant or variables speed and so on. Evaluated dynamic performances are usually the same as in the field tests: guiding and frame forces, derailment criteria, lateral and vertical accelerations, dynamic factors, wear factors in rail\wheel contacts, etc.

Consider the following list of simple tests.

- Computation of dynamic factors for the primary and secondary suspensions by different irregularities.
- Evaluation of the critical speed.
- Evaluation of wear factors by run in curves.

### 1.1.7.2.3.1. Computation of dynamic factors

Let us compute dynamic factors of the primary and secondary suspensions for different level of track irregularities. To vary the level of irregularities, we change value of parameter *Factor* which in fact is multipliers to assigned irregularities, Figure 1.54.

Let the locomotive initial speed be 30 m/s, and the neutral speed mode is selected; track be a tangent section with assigned irregularities “ERRI High”, Figure 1.54. Simulation time is 10s.

Variables corresponding to dynamic factors are created with the wizard of variables, Sect. 1.4.3.1. "Some features of creation of variables", p. 1-186, see [Chapter 4](#) for more details. The variables are dragged into graphical windows.

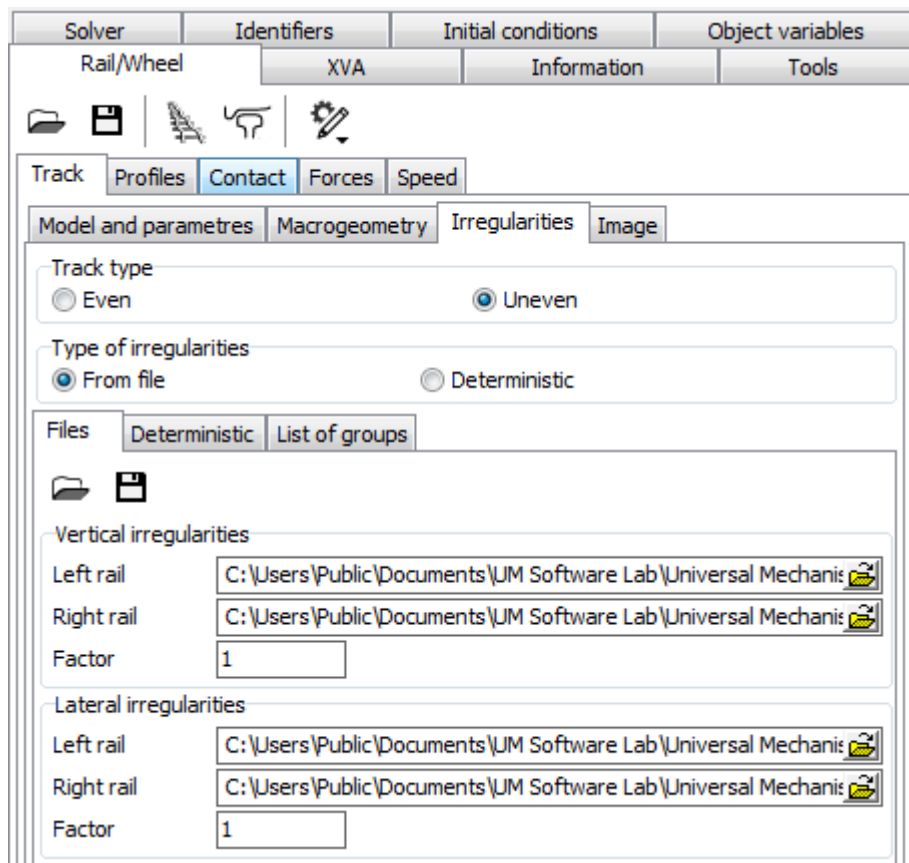


Figure 1.54. Track irregularities

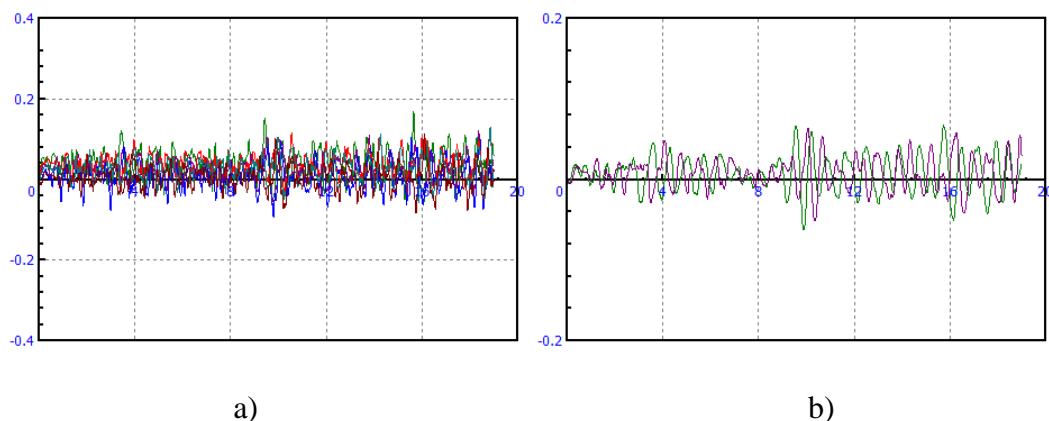


Figure 1.55. Dynamic factors for the primary (a) and secondary (b) suspensions; irregularity factors are 1

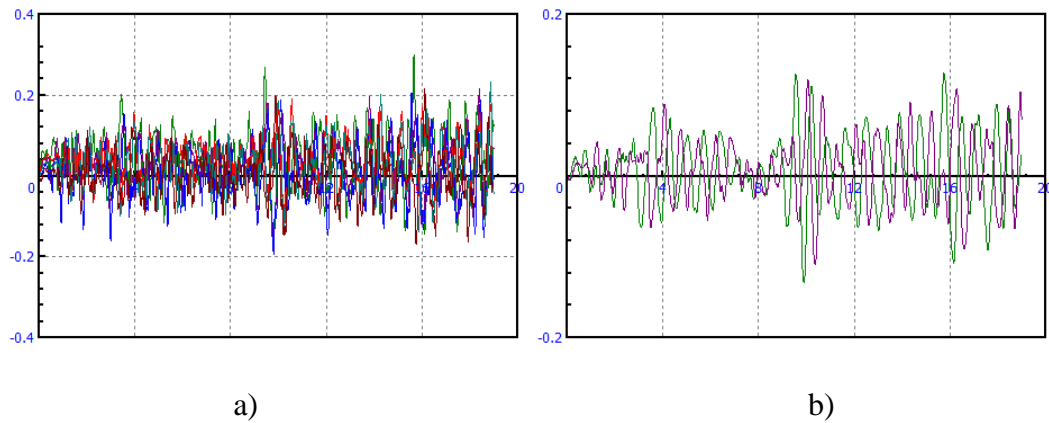


Figure 1.56. Dynamic factors for the primary (a) and secondary (b) suspensions; irregularity factors are 2

The processing of the plots is usually done by the table processor, available by the **Tools | Table processor** menu command. Some results for the **\_4Max\_Mean** functional (Sect. 1.4.3.5, "Table processor", p. 1-213) are shown in Figure 1.57.

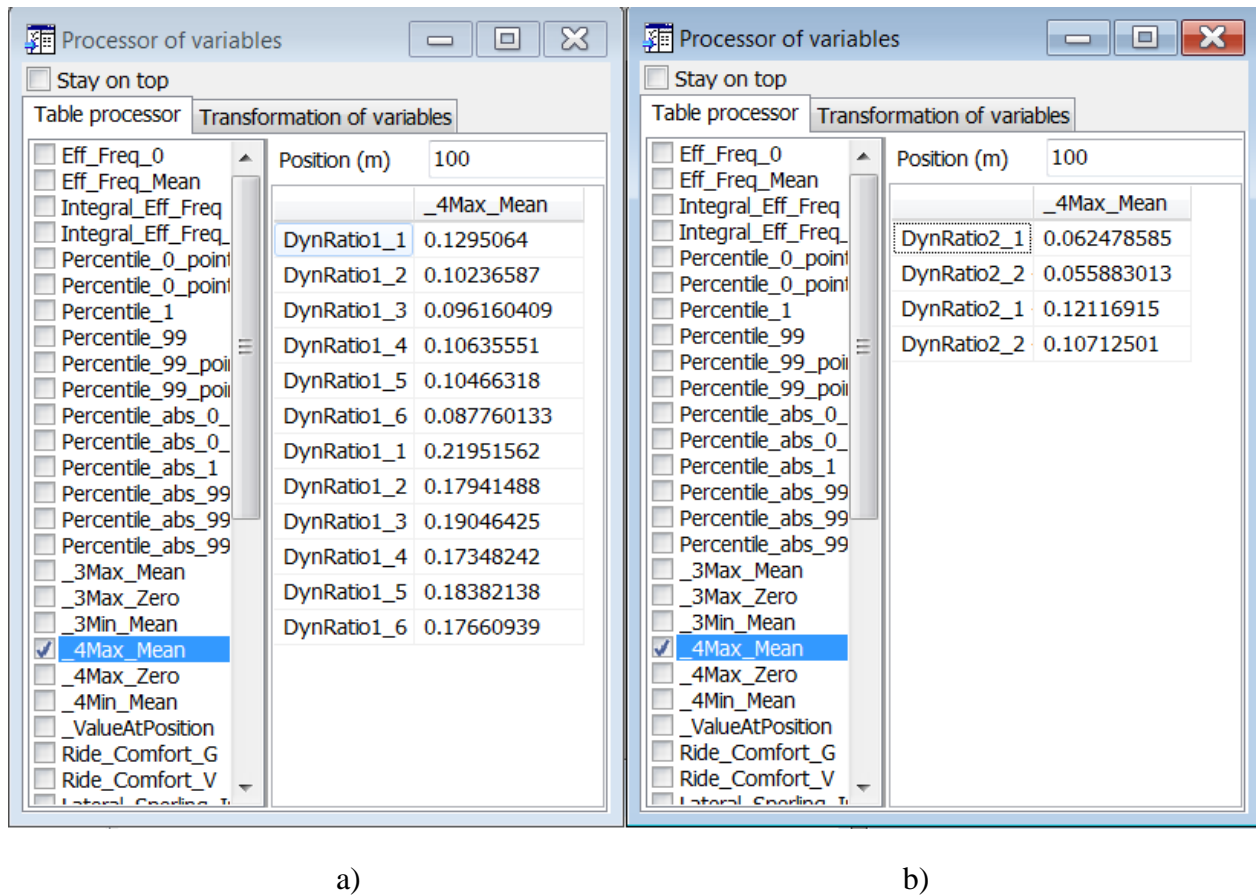


Figure 1.57. Maximal dynamic factors for the first and the second tests

**1.1.7.2.3.2. Critical speed of locomotive**

Evaluation of the critical speed is an important problem. The only way to do it on the design stage is the simulation.

**Note.** Methods for evaluation of critical speed are detailed considered in Sect. 1.7. *"Methods for evaluation of critical speed"*, p. 1-249.

In our case we evaluate the critical speed in the following steps.

- With the deterministic type of irregularities, we assign a single lateral irregularity; in the vertical direction rails are ideal, Figure 1.58.
- We run a series of simulations with increasing speed.
- Basing on the lateral displacements of wheelset, we estimate the critical speed.

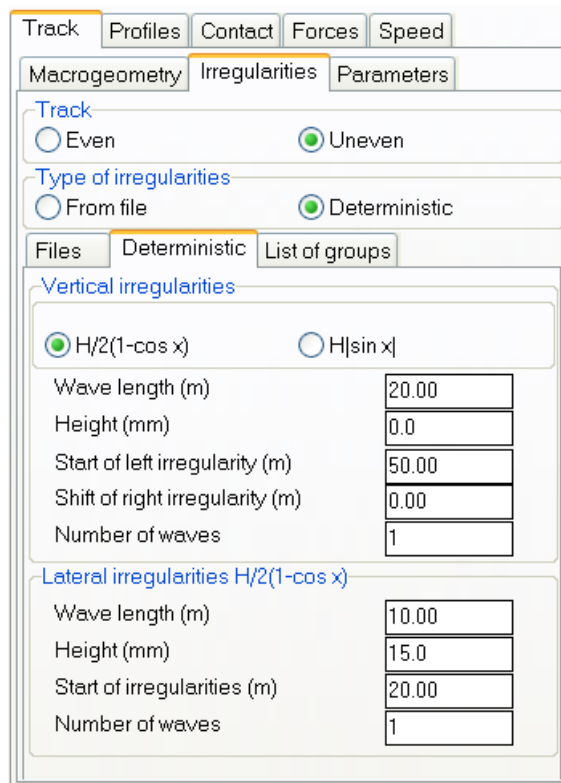


Figure 1.58. Deterministic irregularities

Critical speed of the locomotive model is 70 m/s. Compare lateral motion of wheelsets for 60 m/s and 80 m/s in Figure 1.59.

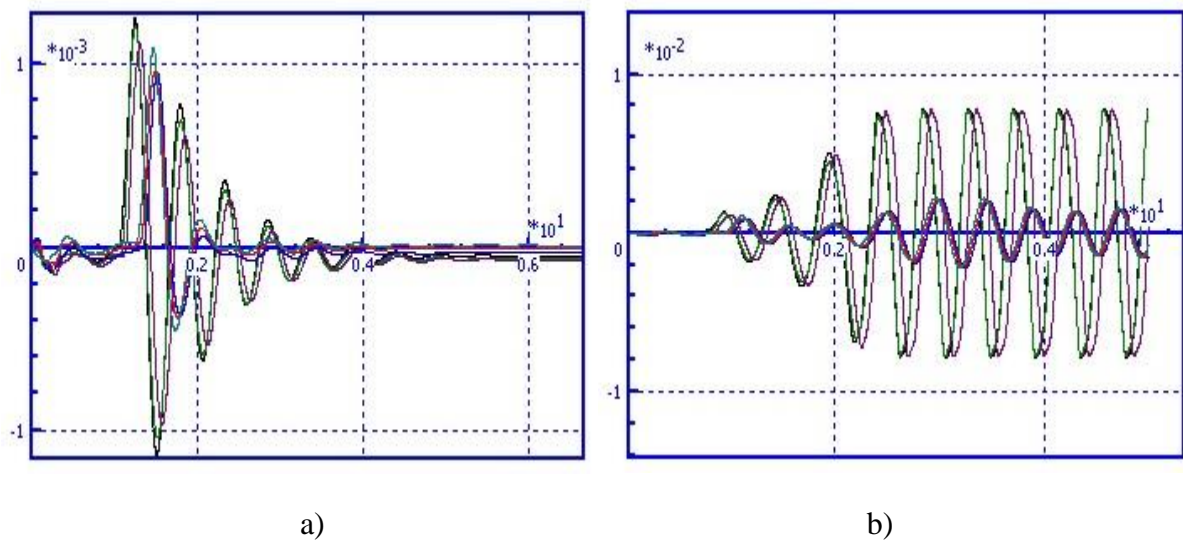


Figure 1.59. Lateral motion of wheelsets for speed 60 m/s (a) and 80 m/s (b)

### 1.1.7.2.3.3. Comparison of wheel wear factors by locomotive curving

In this test, we compare the wheel wear factor by motion of the locomotive in a curve  $R = 300$  m with different speeds, with and without rail lubrication.

The wheel wear factor estimation will be based on the friction work in wheel/rail contact. The right curves are considered in UM by default. That's why the wear is usually defined on the left (climbing) wheels. Let us consider the total friction work in the wheel/rail contact for all left wheels of the locomotive. You can create such variable using **Wizard of variables**. To do it select *AWear1* (work of friction forces on tape circle) on the **Rail/Wheel** tab in the **Variables for wheel** group (see Sect. 1.4.3.1.1. "Rail/wheel contact variables", p. 1-186) and choose all left wheels, Figure 1.61. Make the same steps for the *AWear2* (work of friction forces on flange) variable. Then use the **Expression** tab to sum works for all left wheels, Figure 1.62. Move the obtained variable *AWearSum* to the graph window.

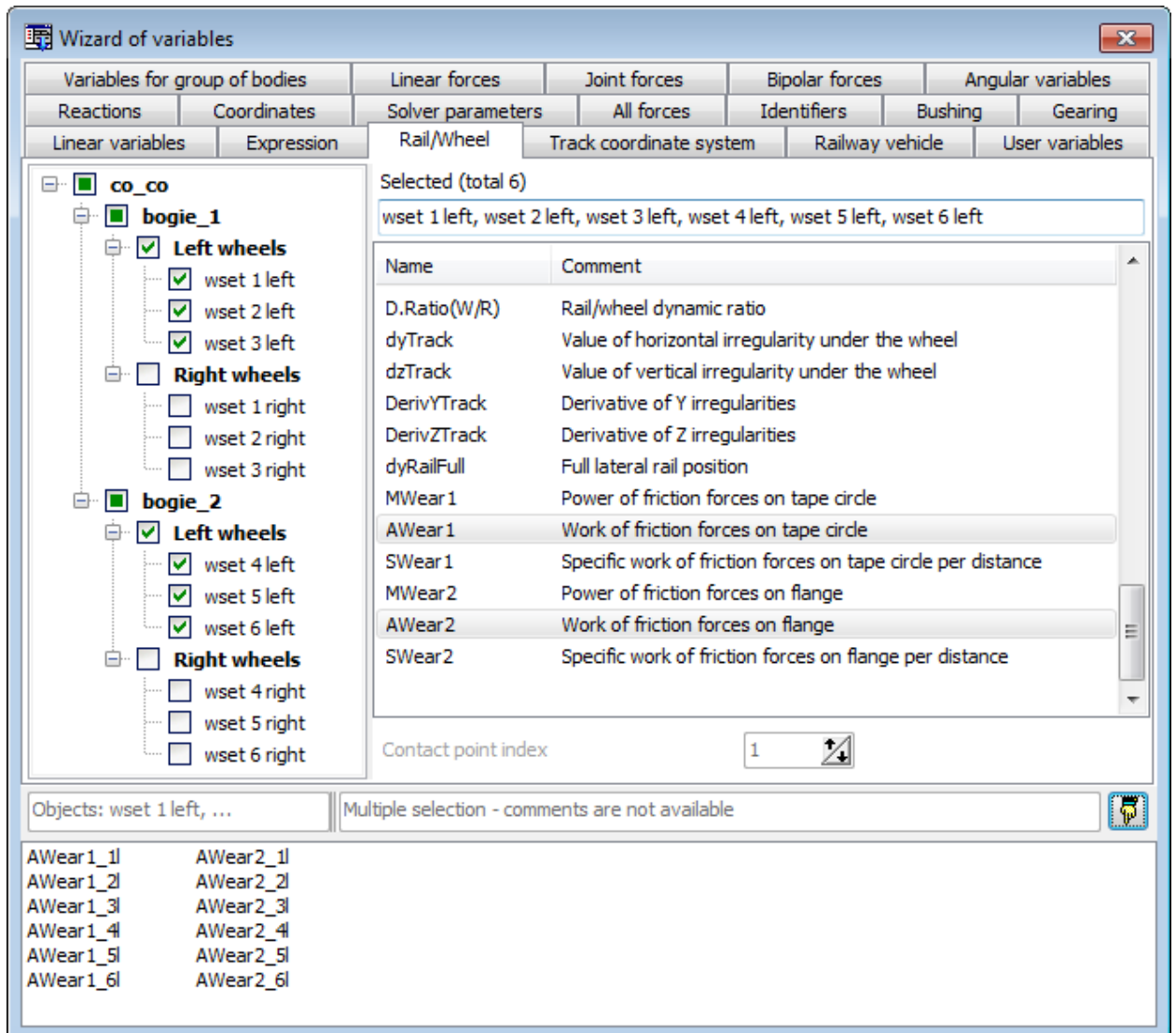


Figure 1.60. Creating "Work of friction forces" variable

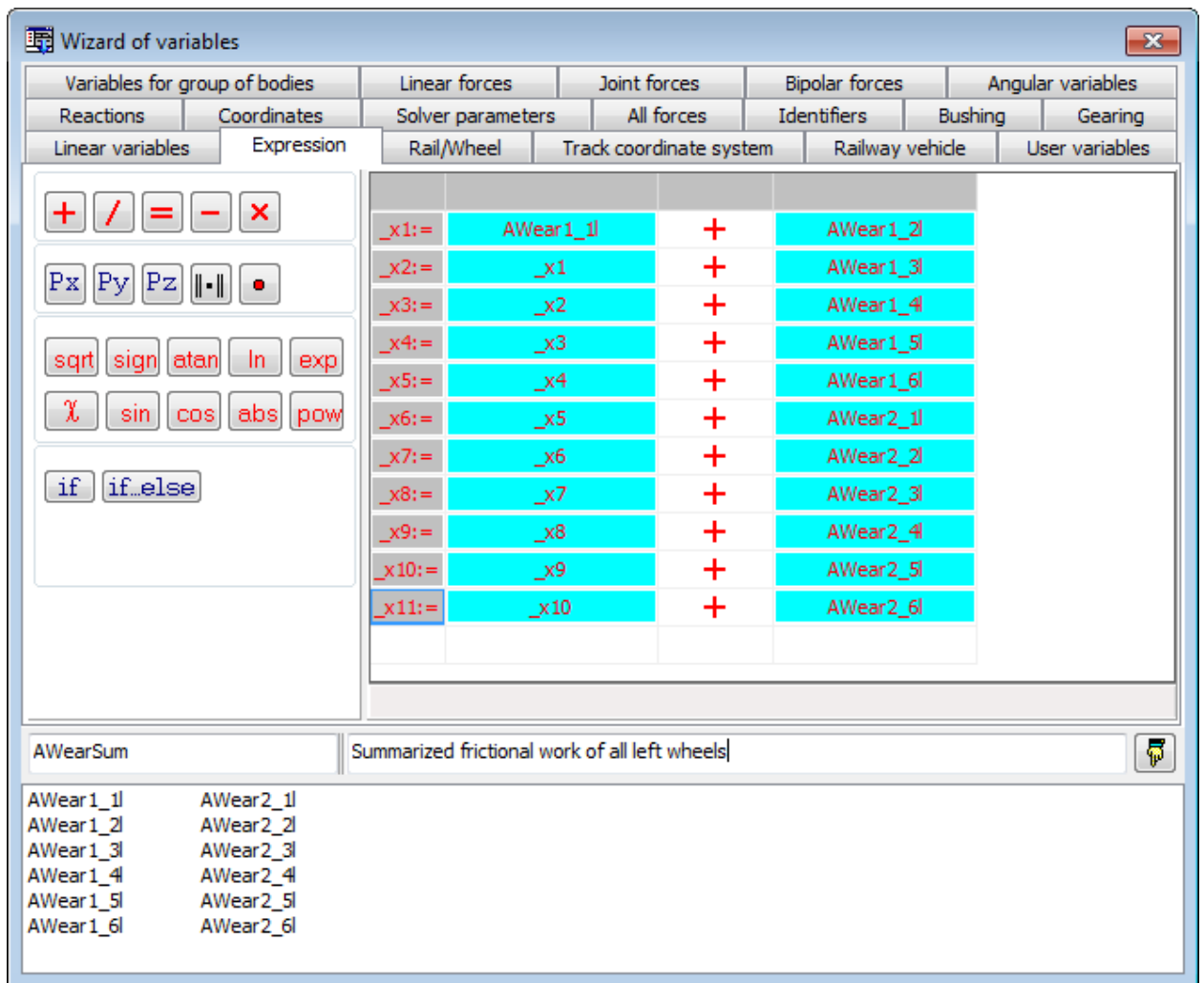


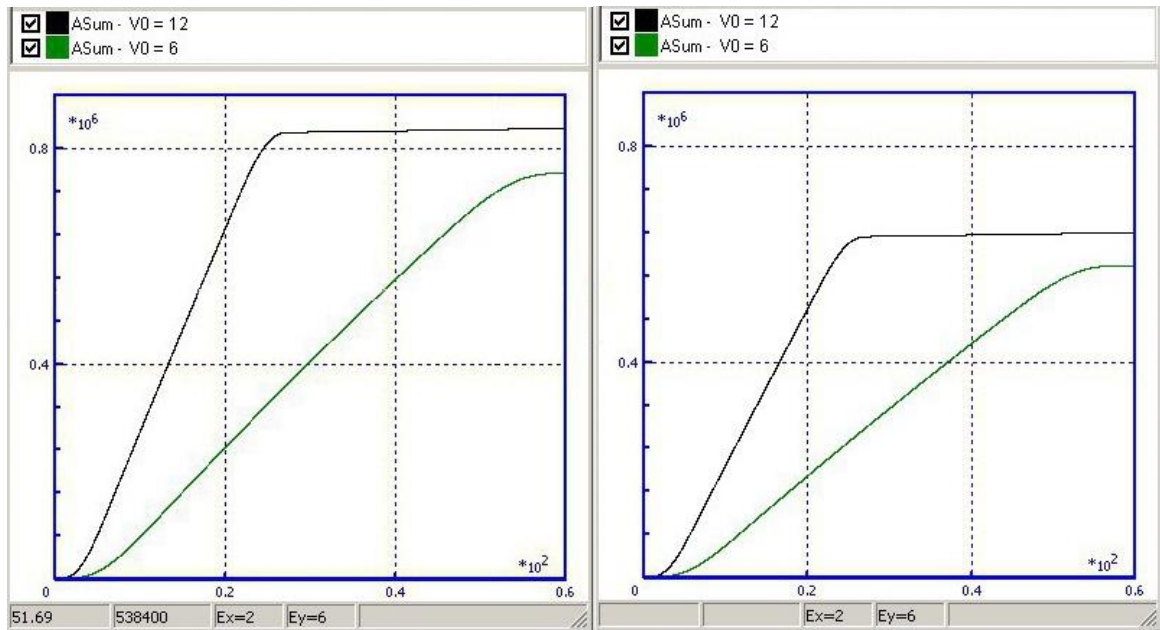
Figure 1.61. Summarized frictional work of all left wheels

We have compared the work of creep forces for the following cases:

- speed 6 m/s, without lubrication;
- speed 12 m/s, without lubrication;
- speed 6 m/s, with lubrication;
- speed 12 m/s, with lubrication.

Let the coefficient of friction without lubrication be 0.25, and with lubrication on the rail side face be 0.15. The values of the coefficients of friction are set in the **Rail/Wheel | Contact | Friction** tab of the inspector.

Such the tests are usually made with the scanning tool if the **UM Experiments** module is available. After run of the experiments, we get the results in Figure 1.62.



a)

b)

Figure 1.62. Summarized frictional work for speed 6 and 12 m/s:  
 a) without lubrication, b) with lubrication

### 1.1.7.3. Support of braking mode

#### 1.1.7.3.1. General information

Simulation of braking process is discussed in the [Chapter 15](#) of UM user's manual, which is devoted to the longitudinal train dynamics. Here we consider adding to a vehicle model (both car and locomotive) force elements only, which allows converting the braking force computed in the train module to the moments and forces acting between bodies of the rail vehicle. Using UM the user could realize a detailed model of mechanical part of the braking system, but here we consider a simplified force model, which is quite precise in many cases.

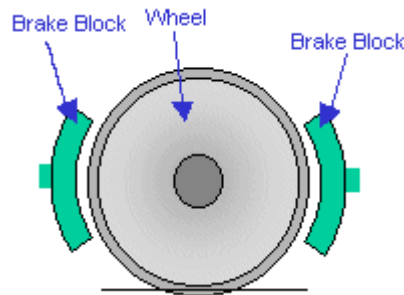


Figure 1.63. Bilateral brake blocks

Let the total braking force for the vehicle is set by an identifier. It is recommended to use for this purpose the standard identifier *braking\_force* to automate the process of its recognition in the train 3D models. Consider bilateral brake blocks, which are pressed symmetrically against the wheel tread, Figure 1.63. It is assumed also that all the wheelsets of the vehicle produce equal braking forces. In this case the brake force for each of the wheelsets can be realized by torque acting from the bogie frame on the wheelset opposite to its rotation

$$\pm \text{braking\_force}/N_w * R_w$$

where  $N_w$  is the number of wheelsets, and  $R_w$  is the wheel radius. The + sign is set if the wheelset forward motion corresponds to the negative angular velocity (it is possible if a subsystem including the wheelset is rotated on 180 degrees about the vertical axis), see the sign of angular velocity of the wheelset in Figure 1.32.

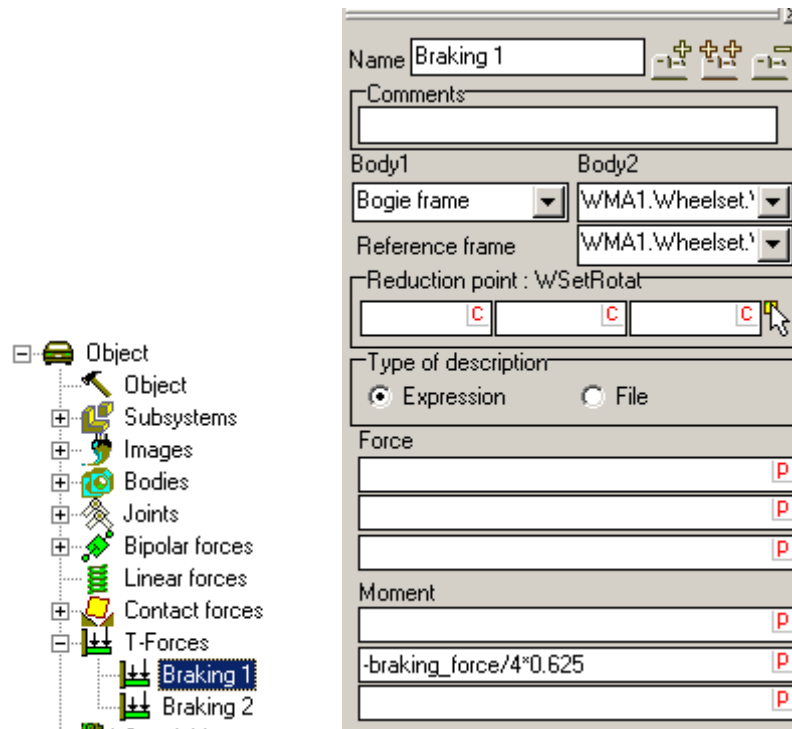


Figure 1.64. Setting braking torque for locomotive VL80

Such a moment should be added for each of the wheelsets using the *T-force* element, Figure 1.64. The first body in this element is the bogie frame, the second one is the rotational part of the wheelset (*WSetRotat*), and the reference body must be the base of the wheelset. It is an error if the reference body is Base0.

The same model can be used in case of brake disks. If unilateral brake blocks are used, the force element must contain in addition a vertical force equal to the braking force; the user should take care of the signs of torque and force. The signs both depend on direction of rotation of the wheelset, and on position of the brake block relative to the wheelset.

**1.1.7.3.2. Use of visual components**

Visual components make faster the process of development of force elements that describe braking forces, Sect. 1.1.7.3.1.

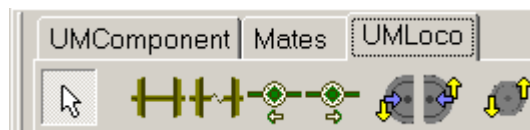


Figure 1.65. Visual components of UM Loco

The following steps are necessary if the visual components are used.

- Click a component button, Figure 1.65.
- Click by the mouse on the frame or other body attached to the braking mechanism; for instance, it is the bolster in case of freight three-piece bogie (the first body).
- Click on the wheelset (the second body).
- Click on the connection point in the center of the wheelset (reference body for the force).

- Set desired values of identifiers in the **Initialization of values** window: number of wheelsets and wheel radius, Figure 1.66. This initialization is not necessary if identifier is already introduced.

**Three components are used for modeling brakes.**



Bilateral brake blocks or disc brakes. A brake torque is added to the frame and wheelset. List of identifiers is shown in Figure 1.68.

Identifier	Value	Comment
braking_force	0	Total braking force for the vehicle
n_wheelsets	4	Number of wheelsets
radius_wheel	0.475	Wheelset radius

Accept    Add to the sheet:    Whole list

Figure 1.66. List of identifiers parameterizing the bilateral brake



Unilateral brake blocks, rear position of the block. The force summarizes action from two blocks of the wheelset.

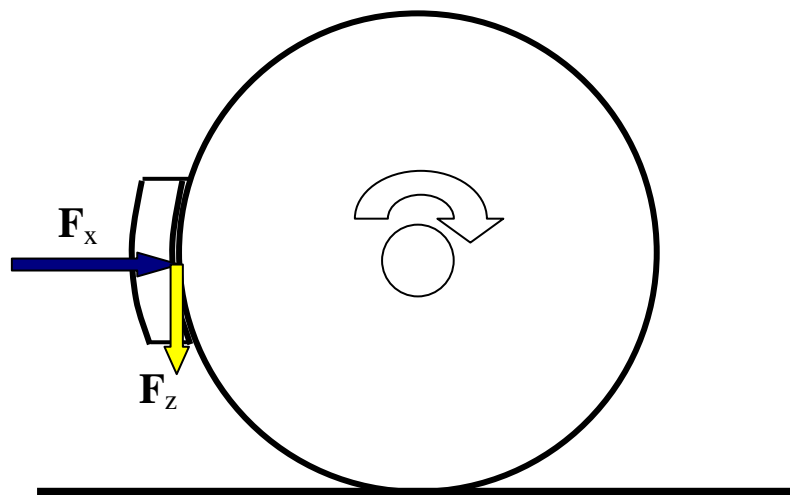


Figure 1.67. Loading scheme for rear unilateral brake block

Loading scheme is shown in Figure 1.67. Normal force  $F_x$  and friction force  $F_z$  satisfy the equation  $F_z = fF_x$ , where  $f$  is the coefficient of friction. List of identifier includes coefficient of friction, Figure 1.68.

k_friction_brake_block	0.3	Coefficient of friction for a brake block
------------------------	-----	---

Figure 1.68. Additional identifier for an unilateral brake

Forces  $F_x, F_z$  are applied to the wheel tread, but T-force must be applied to a fixed point of a body. That is why the transformation of the force to the center of wheelset is done, and a brake torque is added

$$M_y = F_z r_w,$$

where  $r_w$  is the wheel radius.



Unilateral brake blocks, front position of the block. The force summarizes action from two blocks that are installed on the wheelset.

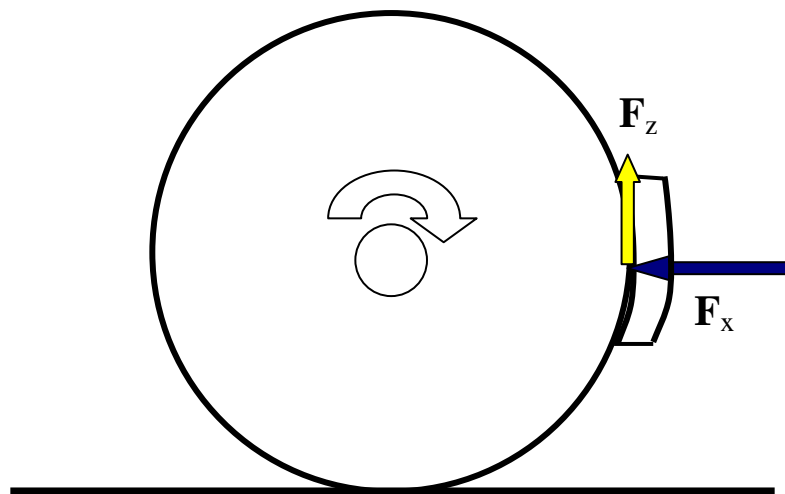


Figure 1.69. Loading scheme for rear unilateral brake block

Description of the element is similar to the previous one.

Correction of signs of force and torque

In the case of a unilateral brake blocks, signs of the force  $F_x$  and torque  $M_y$  depend on the sign of wheelset angular velocity. If the velocity is negative, signs of the force and torque must be changed to opposite, Figure 1.70. For instance, if a wheelset is rotated on 180 about the vertical axis together with the bogie, its angular velocity will be negative. The sign of the wheelset can be obtained from Figure 1.32.

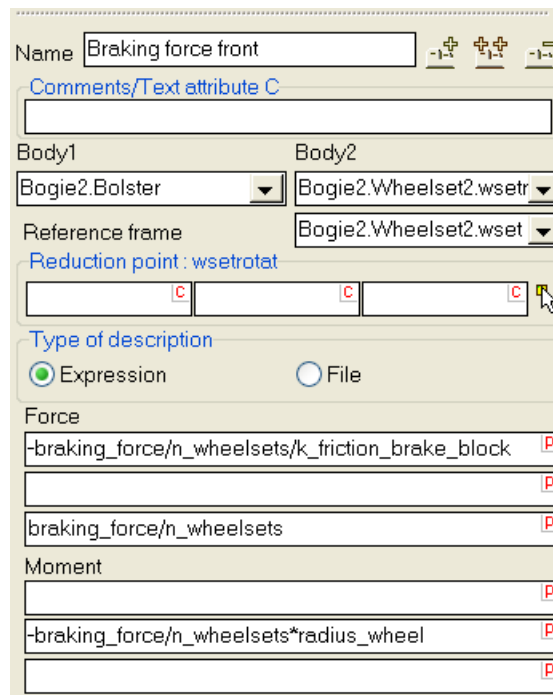


Figure 1.70. Standard description of force element that models braking forces

### 1.1.8. Railway test rig model development and simulation features

Roller rigs (Figure 1.71) are used for a wide range of studies which concern the stability of the rail vehicle, passengers comfort, braking systems, wear prediction. The Roller rigs are easy to use and require significantly less investment than the full-scale field tests.

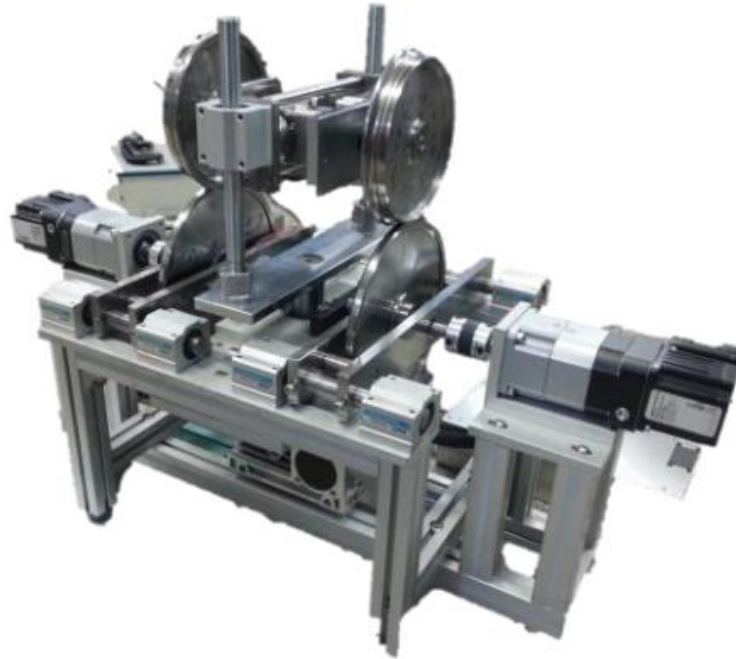


Figure 1.71. Roller rig (Hongik University, the Republic of Korea), reprinted from [1]

To create a roller rig model take the following steps in the **UM Input** program:

1. Add bodies, which model the rollers, set their inertia parameters, assign the corresponding graphical objects and joints. One roller must be assigned to each of the wheels. The following requirements take place:
  - All rollers must be identical;
  - Each of the rollers has one rotational DOF relative to the Y axis below the wheelset. It can be done by a *rotational joint* ([Chapter 2](#), Sect. "*Translational and rotational joint*"), which description includes:
    - Connecting bodies: Base0 as the first body and the roller as the second one;
    - Axis of rotation (0,1,0) for each of the bodies;
    - Roller position as the joint point for the Base0 body, see Figure 1.72.

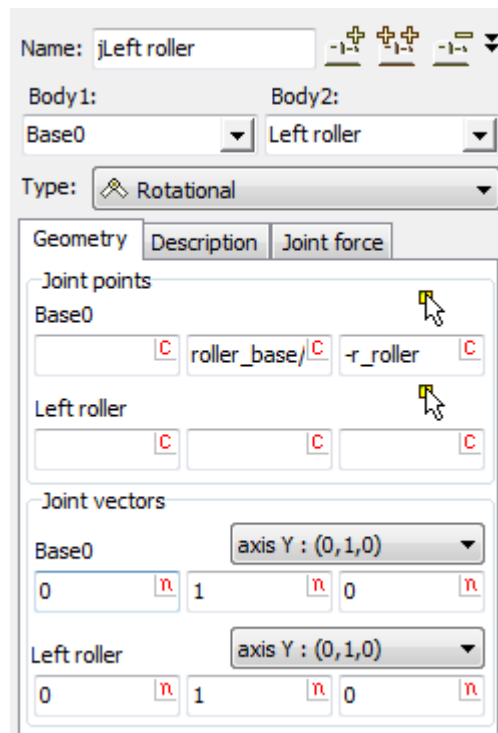


Figure 1.72. Rotational joint which describes roller DOF relative to the base

- Go to data inspector of the "**Wheelset**" subsystem window and activate the key "**Test rig**". The standard identifier **r\_roller** corresponding to the roller radius is automatically added to the list of identifiers after this operation. Then the roller bodies must be assigned to the left and right wheels like in Figure 1.73.

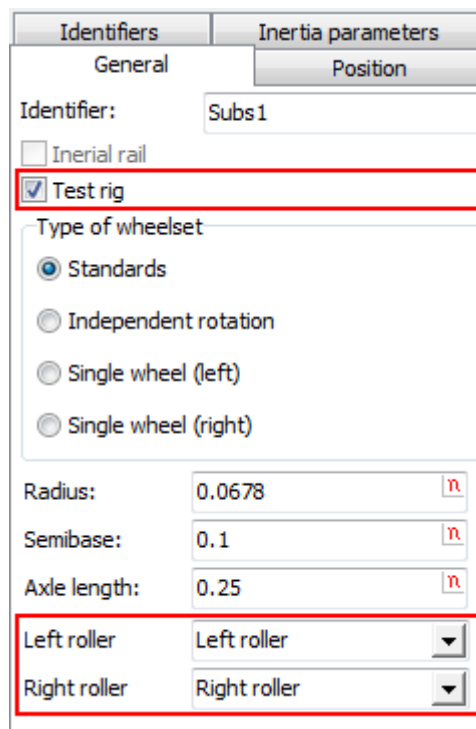


Figure 1.73. Assigning roller-bodies to the wheelset

Let us consider briefly some features of specifying initial values for roller rotation speed by simulation. If initial rotation velocity differs from zero, it must be computed depending on the value of identifier **v0**. This identifier corresponds to 'imaginary' longitudinal speed of the test rig. The initial rotation speed of a wheelset is equal to,  $v_0/r_w$  where  $r_w$  is the wheel radius. Respectively, the roller angular velocity  $\omega_r$  is computed as:

$$\omega_r = -\frac{v_0}{r_r},$$

where  $r_r$  is the roller radius. Initial angular velocity can be assigned to the roller by the 'initial constrain' tool, see [Chapter 4](#), Sect. "*Constraints on initial conditions*".

If the test rig starts with zero speed, the identifier **v0** value should be zero.

Roller profiles are assigned on the **Rail/Wheel | Profiles | Rails** tab of the simulation inspector, see Sect. 1.4.2.3.1 "*Assignment of rail profiles*", p. 1-145.

## 1.2. Track

### 1.2.1. Track geometry

Track geometry includes the following components:

- geometry of rails in an ideal tangent track (track gauge, inclination of rail, rail profiles);
- macro geometry of curves;
- track irregularities.

#### 1.2.1.1. Geometry of rails in an ideal track

The geometry of rails in an ideal tangent track includes:

- profiles of left and right rails,
- distance between centers of heads of rails,
- rail inclination.

A rail profile in UM should be set in a special system of coordinates (SCR), Figure 1.74. SCR origin is located at the profile top on its symmetry axis (i.e. the profile curve passes through the origin). The abscissa axis (y) is perpendicular to the rail profile axis and directed inside the track. The ordinate axis (z) is directed upwards. Unit for profile data is millimeter.

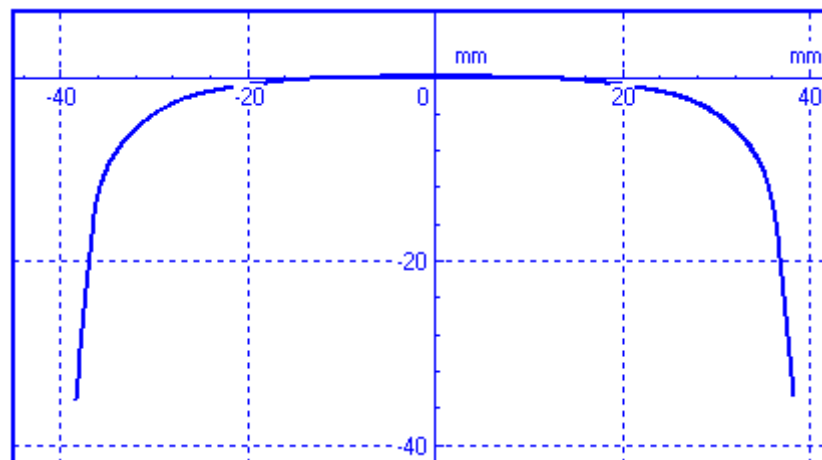


Figure 1.74. System of coordinates of rail

Lateral position of rails in an ideal straight section is set by the SCR-SCW distance  $\Delta y$  relative to the wheelset base (Figure 1.75)

$$\Delta y = \frac{L_r - L}{2},$$

where  $L_r$  is the distance between the rail head centers,  $L$  is wheel base (distance between running circles, Sect. 1.1.3.5. "Wheelset geometry", p. 1-13). In other words, the SCR-SCW distance is the lateral distance between the origins of two profile frames (wheel and rail) at ideal symmetric position of the wheelset.

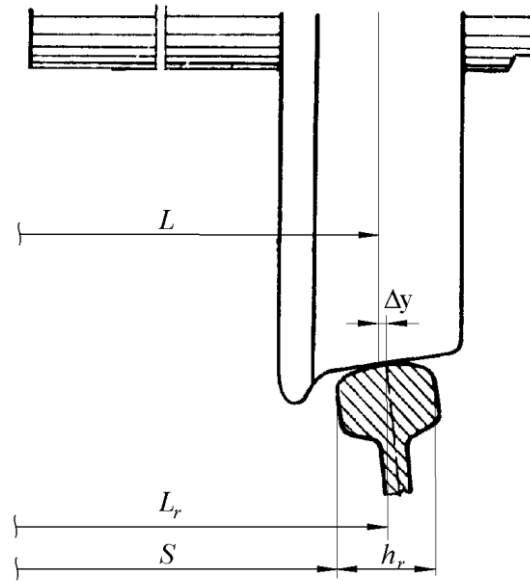


Figure 1.75. SCR-SCW distance

Further expression for the SCR-SCW:

$$\Delta y = \frac{S + h_r - L}{2}$$

where  $S$  is the gauge,  $h_r$  is the width of the rail head. Default value is  $\Delta y = 0.003$  m.

Rail inclination  $\alpha_{r0}$  is the angle between the rail profile axis of symmetry and the vertical direction in an ideal straight track. Unit for the inclination is radian. The angle is positive for inclination inside the track. The default value is  $\alpha_{r0} = 0,05$  rad.

The rail profile can be changed with the distance along the track (Sect. 0). To do this, enter a set of profiles as well as positions of each along the track. Let profiles R1 and R2 have positions  $S1$  and  $S2$  ( $S2 > S1$ ). Then the program computes an intermediate profile  $R(S)$  for every necessary position  $S \in [S1, S2]$  as a linear interpolation of profiles R1 and R2 such as  $R(S1) = R1$ ,  $R(S2) = R2$ .

### 1.2.1.2. Geometry of curve

The following types of curves are available in UM (Figure 1.76):

- right curve
- S-curve (a right curve followed by a left one)
- left curve (can be obtained as a S-curve with a very short right curve section)

The following designations are used for the S-curve in Figure 1.76:

- $L0$  is the length of a straight section before the curve;
- $P11$  is the length of the first transition for the right curve;
- $S1$  is the length of steady curve;
- $R1$  is the radius of steady curve;
- $H1$  is the cant for the outer rail;

- $P12$  is the length of second transition for the right curve;
- $dy$  is the additional gauge widening in curve;
- $L$  is the length of a straight section between the right and the left curves (for S-curves only).  
Other parameters for the second part of the S-curve have quite the same meaning.

**Note.** To get a left curve, small values should be set for  $P11$ ,  $S1$ ,  $P12$ , zero value for  $H1$  and a large value for  $R1$ , e.g.,  $P11=0.01$ ,  $S1=0.01$ ,  $P12=0.01$ ,  $R1=100000$ .

Transient sections are formed by a cubic parabola or by clothoid, Sect. 1.4.2.4.1. "Track model choosing and parameters setting", p. 1-150. The curvature in transitions changes according approximately (for s cubic parabola) and exactly (for clothoid) linear law.

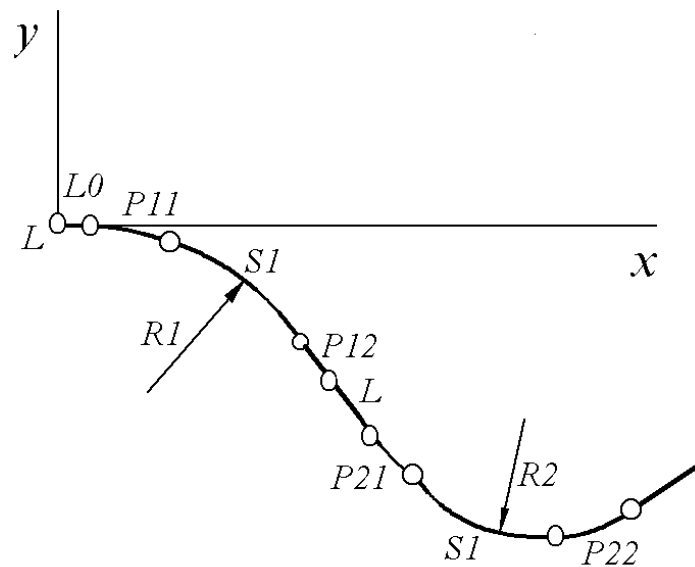


Figure 1.76. On macro geometry of curve

Cant increases/decreases at transient by linear function except the **Curve** type of macro-geometry, where the cant can be specified arbitrary by a set of points. It is possible to smooth the vertical junctions at ends of the transition by an arc.

*Additional widening of track in a curve ( $dy$ ) is proposed automatically:  $dy = 10$  mm for  $R \in [300,350]$ ,  $dy = 15$  mm for  $R < 300$  m. The widening is realized as a symmetric lateral shift of both the rails on a half of the widening. The widening in transient sections is the linear function of the position. The user may set his/her own widening, see Sect. 1.4.2.4.1. "Track model choosing and parameters setting", p. 1-150.*

- Let us introduce a number of designations for curve parameters computed by UM:
- $X(s)$ ,  $Y(s)$  are equations for a curve,  $s$  is the length along the curve;
  - $\psi_b(s)$  is an angle between the X-axis and the tangent to the curve;
  - $\rho(s)$  is a curvature radius;
  - $S_0$  is a traveled distance.

Consider an example of an S-curve for the following parameters (Figure 1.77-Figure 1.81):

$L_0=10\text{m}$ ,  $p_{11}=70\text{m}$ ,  $s_1=150\text{m}$ ,  $R_1=300\text{m}$ ,  $p_{12}=60\text{m}$ ,  $h_1=0.09\text{m}$ ,

$L=10\text{m}$ ,  $p_{21}=50\text{m}$ ,  $s_2=140\text{m}$ ,  $R_2=330\text{m}$ ,  $p_{22}=70\text{m}$ ,  $h_2=0.1\text{m}$ .

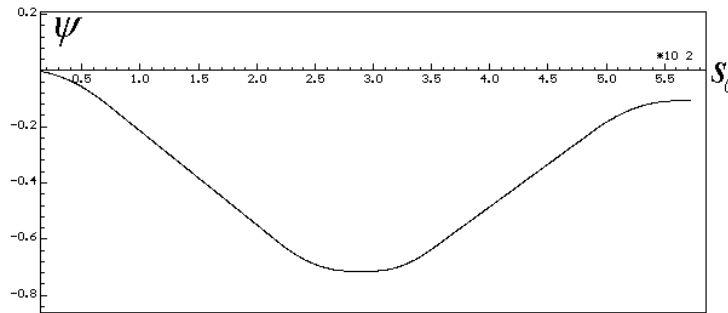


Figure 1.77. Angle between rail and X –axis

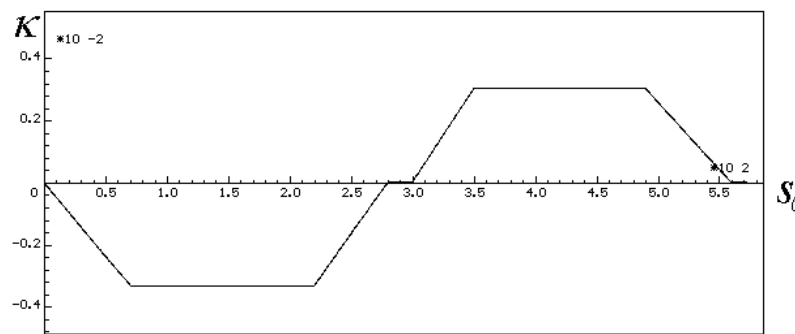


Figure 1.78. Curvature

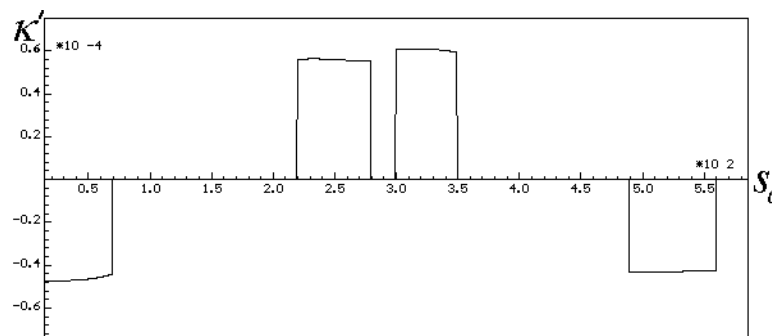


Figure 1.79. Derivative of the curvature

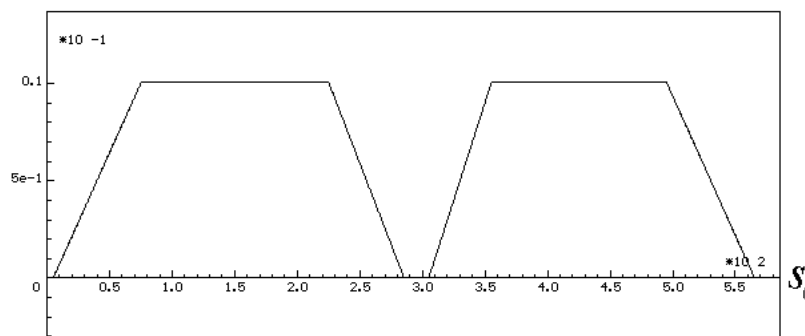


Figure 1.80. Gauge widening

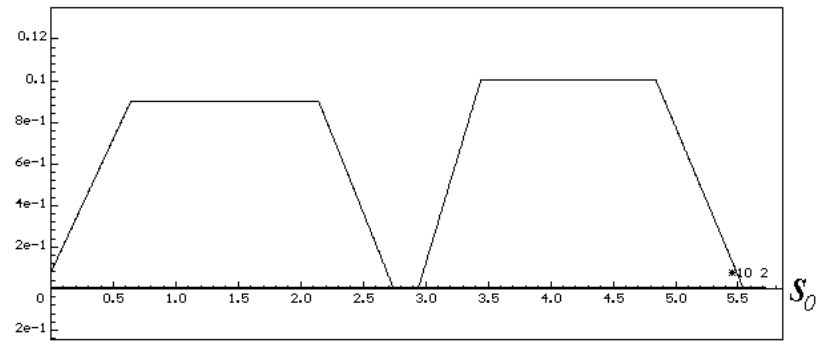


Figure 1.81. Cant for outer rail

1.2.1.3. Switch geometry

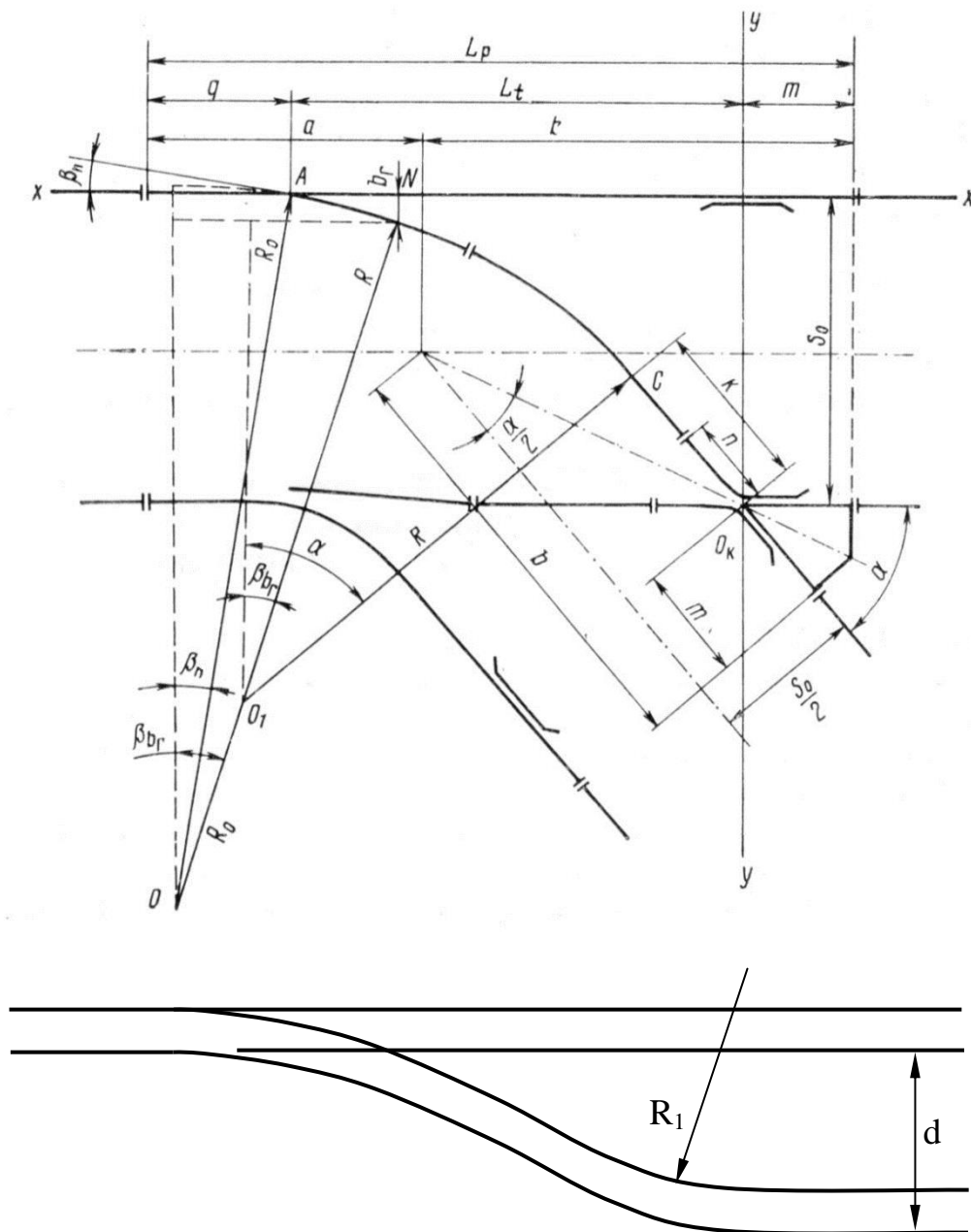


Figure 1.82. Geometric parameters of a switch

Motion in left and right switches is implemented in UM. The basic geometric parameters of a switch are shown in Figure 1.82. The following parameters are used for description of the switch.

- $q$  is a stock rail overhang;
- $\beta_n$  is an initial angle;
- $\alpha$  is a switch angle;
- $R_0$  is a radius of point;
- $R$  is a radius of switch;
- $b_r$  is a switch deviation for  $R_0$ ;

$m$  is a frog tail length;

$d$  is a track spacing.

$R_1$  is a radius behind frog.

The parameters define fully the switch geometry and some additional parameters in Figure 1.82 like

$L_p$  is the full switch length;

$L_t$  is the theoretical length;

$k$  is tangent section before frog.

UM includes the standard R65 1/11 and R65 1/9 exUSSR switches. The user can create files with any switch parameters.

### 1.2.1.4. Track system of coordinates

A track system of coordinates is introduced for each body  $i$  of the vehicle (TSC). Its origin  $W_i$  coincides with the projection of a body-fixed point on the ideal central axis of the track. This body-fixed point is either the origin of the body-fixed system of coordinates, or a body-fixed sensor. The sensor is any point which kinematics is analyzed. The abscissa axis  $x_i$  is the tangent to the ideal track centerline. The ordinate axis  $y_i$  lies in the track plane on the left to the motion direction.

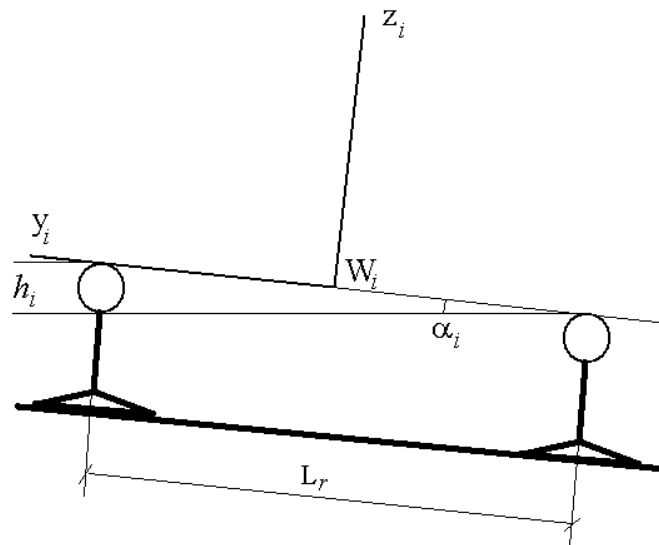


Figure 1.83. TSC axes in presence of a cant

A cant of the outer rail  $h_i$  in Figure 1.84 gives the rotation of the track plane on the angle

$$\alpha_i = \arcsin \frac{h_i}{L_r}.$$

TSC is used for calculation of some dynamic performances of the vehicle in curves, Sect. 1.4.3.1.9. "Kinematic characteristics relative to track system of coordinates", p. 1-205.

### 1.2.1.5. Track irregularities

Vertical and horizontal irregularities of rails are stored in files \*.way and assigned to rails. Step size of the irregularities is 0.1 m (Sect. 1.4.1.2. "Creation of track irregularities", p. 1-113).

It is possible to program user's irregularities in the *Control File* (Sect. 1.4.1.2.2. "Programming irregularities in the Control file", p. 1-122).

**Note.** To avoid a force jump while going up the begin of an irregularity, the irregularity is *set to zero on the first ten meters*, and then irregularities are multiplied by a special factor that changes from 0 to 1 on the next 20 m. As an example consider an irregularity of a constant height of 1 mm. In fact the irregularity in Figure 1.84 will be applied. This is valid for file irregularity only, and not for the irregularities that are programmed in the *Control File*.

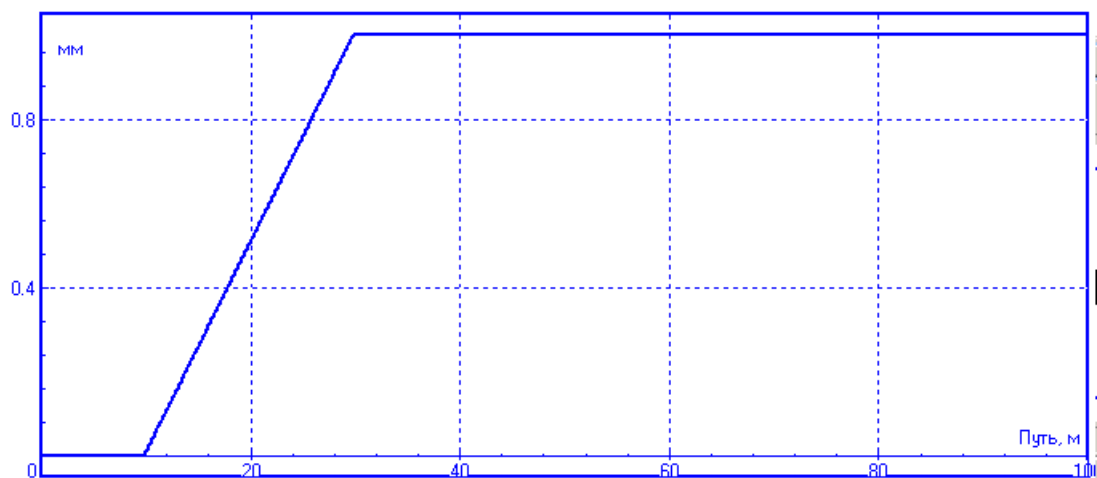


Figure 1.84. Smoothing an irregularity

## 1.2.2. Track models

**Universal Mechanism** supports three track models that consider track with different level of details:

- **Massless rail;**
- **Inertial rail;**
- **Flexible track.**

**Massless rail** track model treats rail as a massless force element. For such a rail model generalized coordinates are not introduced. Rail deflections are calculated as a result of solution of equilibrium equations (Sect. 1.3.1.2.1. "*Method for computation of rail deflections and contact force*", p. 1-85). This model is recommended to use for analysis/optimization of running gears of railway vehicles since intrinsic rail dynamics weakly effects on simulation results of rail vehicles. **Massless rail** model is used as the default track model.

**Inertial rail** track model considers rails as rigid bodies under each wheel, see Figure 1.85. Every rigid body that simulates inertial rails has three degrees of freedom: two longitudinal d.o.f. relative to lateral (Y) and vertical (Z) axes and one rotational d.o.f. relative to longitudinal (X) axis. Equations of motion for inertial rails are given in track coordinate system (Sect. 1.2.1.4. "*Track system of coordinates*", 1-72). Underrail base is modeled as a *Special force* of *Bushing* type. **Inertial rail** model is recommended to use for simulation of complex scenario of wheel-to-rail contact: railway track evolution in the switches and turnouts, flange-back and conformal contacts, simulation of vehicle derailment cases, prediction of wheel and rail wear, etc.

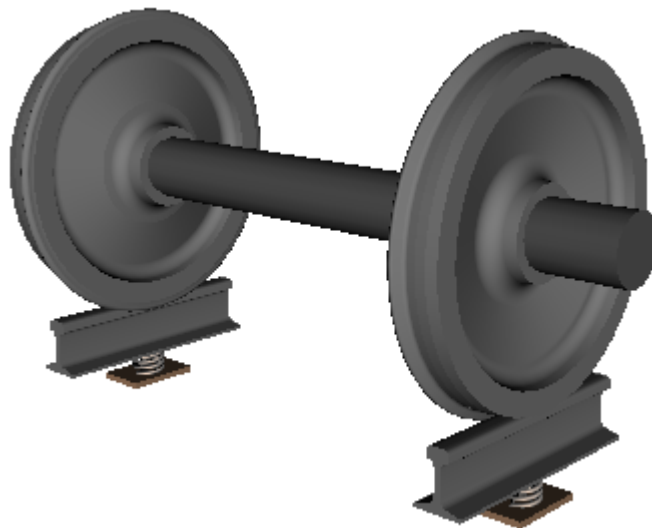


Figure 1.85. **Inertial rail** track model

The presence of rail bodies in a railway vehicle model is regulated by the flag **Inertial rail** in **Wheelset** subsystem settings in **UM Input** program (Sect. 1.1.3.1. "*Adding a wheelset or a wheel*", 1-9).

**Flexible track** model is a detailed 3D track model that includes flexible rails, fasteners, sleepers and sleeper foundation. **UM Flexible railway track** module is required for **Flexible track** model. See [Chapter 27](#) how to work with **UM Flexible track** module. **Flexible track**

model is recommended for problems that are focused on dynamics of the railway track and railway track foundation.

Listed above track models **Massless rail**, **Inertial rail** and **Flexible track** treat sequentially more and more complex models and approaches to simulation railway track. In fact, more complex models provide more accurate results but require more CPU efforts. The following rough estimations of relative CPU efforts while using different track models might be given. **Inertial rail** is about 2-3 times slower and **Flexible track** model is about 50-80 times slower than **Massless rail** model.

Please note that the frequency range for **Massless rail** model is 0-20 Hz. **Inertial rail** provides reliable simulation in the frequency range up to 100 Hz, and **Flexible track** – up to 1000 Hz.

A railway vehicle model can simultaneously include all the track models listed above. You can choose a track model on the tab sheet **Rail/Wheel | Track | Model and parameters**, Figure 1.86.

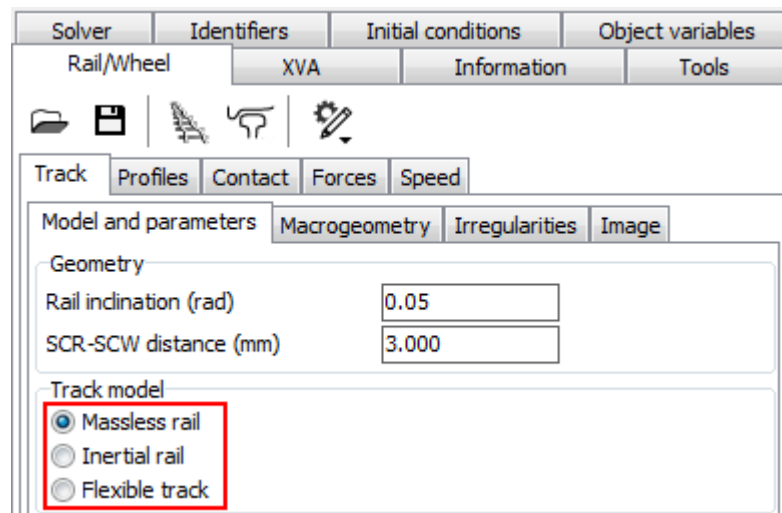


Figure 1.86. Track model choosing

## 1.3. Wheel-rail contact

Three main parts of contact computation can be pointed out:

- contact geometry, i.e. computing locations of contact points for a given position of a wheel-set relative to rails;
- computation of kinematical characteristics at contacts (creepages and spins);
- computation of normal forces and creep forces at contacts according to geometrical and kinematical parameters.

As it is shown below, these problems are not solved independently; a general iterative procedure is necessary.

A large part of the results, presented in this section, is obtained in cooperation with V.S. Kossov. In exposition of this section we will mainly follow the results and algorithms presented in the thesis by V.S. Kossov, making some refinements, modifications and generalizations.

The main method of calculating the geometry of the wheel and rail contact interaction, considered in this section, is based on the account of the exact geometry of the curves (profiles) which are the wheel and rail cross-sections. The simplified method, based on the notion of efficient conicity and effective contact angle parameter value, is discussed in Sect. 1.3.1.1.3. "*Simplified contact geometry. Equivalent conicity and contact angle parameter*", 1-82.

To improve the reliability and speed of calculating the contact point position two basic ideas were realized. Firstly, if the rail profile is constant along the track, the calculation of the coordinates of contact points for the set of profiles is performed once before the first start of the modeling process. Contact points coordinates are arranged in the arrays depending on the wheel profile displacement relative to the rail profile. It is assumed that the wheel has two degrees of freedom relative to the rail: rotation around the longitudinal axis and transverse displacement. In the process of coordinate values movement modeling (at the current position of the wheel profile relative to the rail) are calculated via the mentioned precalculated arrays using interpolation. This lets significantly reduce the number of arithmetic operations and raise the speed of the process.

Secondly, computation of contact points on the profiles is based on a procedure, which computes the nearest point between two curves. These algorithms proved to be very fast and reliable, and do not depend on smoothness of the curves.

### 1.3.1. Computation of the contact between the wheel and massless rail

#### 1.3.1.1. Algorithms for wheel-rail contact geometry

##### 1.3.1.1.1. Algorithm for computation of nearest points between two profiles

Let us consider an arbitrary position of the wheel profile relative to the system of coordinates of the rail profile  $Y_r Z_r$  (SCR, Sect. 1.2.1.1. "*Geometry of rails in an ideal track*", p. 1-66), Figure 1.87. Introduce a new SC (SCR0)  $Y_0 Z_0$ , which origin coincides with the origin of SCR. The  $Z_0$ -axis is perpendicular to the track plane as if the inclination angle  $\alpha_{r0}$  is zero. The SCR  $Y_r Z_r$  is

inclined on  $\alpha_{r0}$  relative to SCR0. Position of SCW (system of coordinates of the wheel profile, Sect. 1.1.3.5. "Wheelset geometry", p. 1-13)  $Y_w Z_w$  relative to SCR0 is defined by coordinates of the origin  $\Delta y, \Delta z$ , and the angle  $\Delta\alpha$ . Coordinates in Figure 1.87 are positive. All angles are considered to be small.

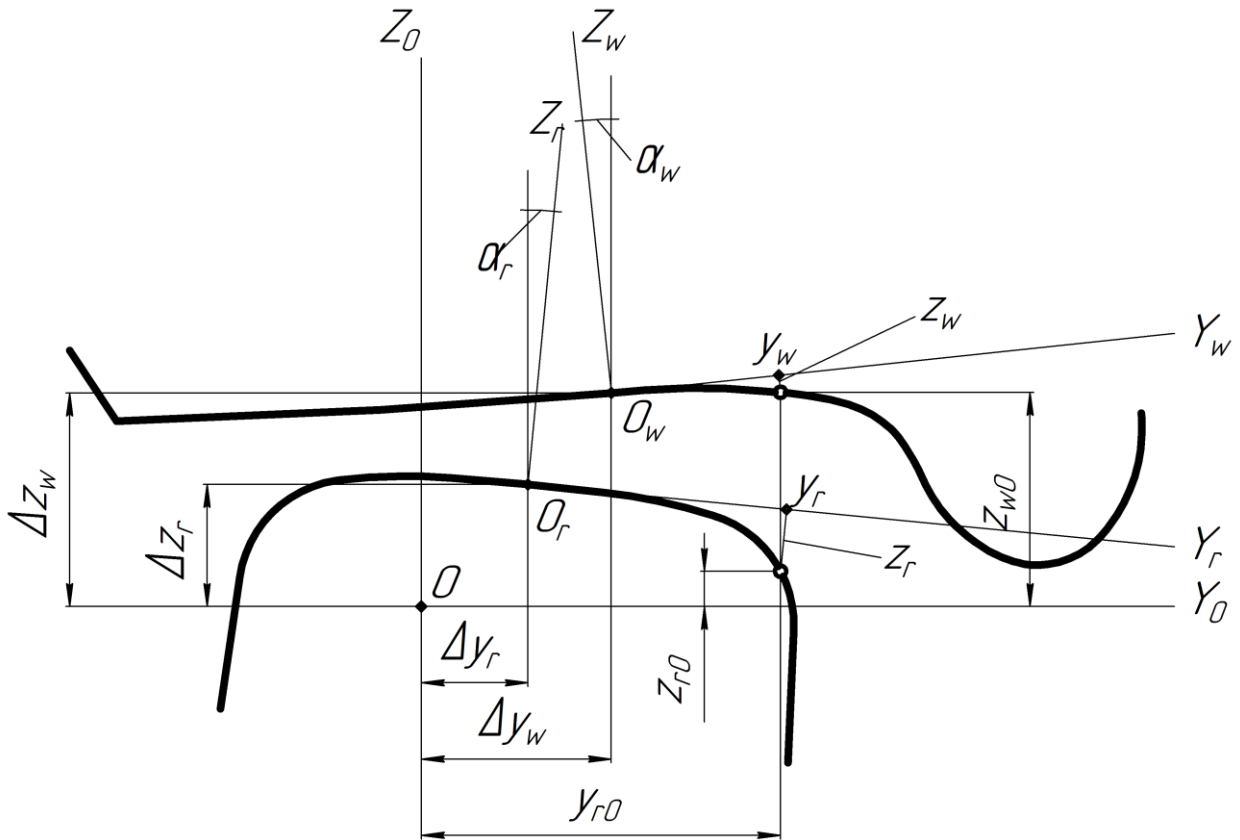


Figure 1.87. Relative position of profiles

It is necessary to find a pair of points on the profiles, which have the same Y-coordinate in SCR0 and the minimal difference in Z-coordinate in SCR0. Obviously, the solution does not depend on  $\Delta z$ .

Let us start with the algorithm for computing the value

$$\delta z = z_{w0} - z_{r0},$$

for two points having the same Y-coordinates in SCR0. The coordinate  $y_r$  of a point on the rail profile in SCR ( $Y_r Z_r$ ) has a given value. We should compute the coordinates  $y_w, z_w$  of the corresponding point on the wheel profile in SCW as well as  $\delta z$ .

Firstly, here are the coordinates of the point on the rail profile in SCR0

$$y_{r0} = y_r + z_r(y_r)\alpha_r,$$

$$z_{r0} = z_r(y_r) - y_r\alpha_r.$$

where  $z_r$  is the coordinate of the point in SCR.

Then a point on the wheel profile with coordinates  $y_w$  and  $z_w$  in SCW should be found, which has the following abscissa in SCR0:  $y_{w0} = y_r - \Delta y$ , Figure 1.87. Since the angle  $\Delta\alpha$  is small, the following nonlinear relation takes place:

$$y_{w0} = y_w - z_w(y_w)\Delta\alpha,$$

or

$$y_w = y_{w0} + z_w(y_w)\Delta\alpha.$$

To solve this nonlinear equation relative to  $y_w$ , direct iterations

$$\begin{aligned} y_w^0 &= y_{r0} - \Delta y, \\ y_w^{i+1} &= y_{w0} + z_w(y_w^i)\Delta\alpha, \quad i = 0,1,2 \dots \end{aligned}$$

can be applied.

As it is known, direct iterations converge if the condition

$$\left| \frac{dz_w}{dy_w} \Delta\alpha \right| < 1.$$

takes place. Since  $\Delta\alpha \ll 1$ , this condition is always valid for real profiles.

Finally, the coordinate  $z_{w0}$  is computed from the formula

$$z_{w0} = y_w\Delta\alpha + z_w(y_w) + \Delta z.$$

and the value of minimized function  $\delta z(y_r)$  is evaluated.

In this way the algorithm allows us to compute a pair of points and the distance between them. The main advantage of the algorithms is its simplicity, reliability and independence on smoothness of profiles.

Computing the minimal value  $\delta z_{\min}$  is executed according to the following algorithm. Consider a sequence of values of coordinate  $y_r$  with a contact step size  $h = 1$  mm and select the points with the minimal value  $\delta z$ . After that the process is repeated near the found point with smaller step size (0.1 mm).

### 1.3.1.1.2. Computing tables of contact points

As it is already mentioned, computation of tables of contact geometry information is executed before start of the simulation, if the rail profile does not change along the track. This information is obtained in dependence on the wheel profile position relative to the rail profile. If we consider a rail as a massless element, there may be two possible contact types: one-point and two-point contact, (Figure 1.88).

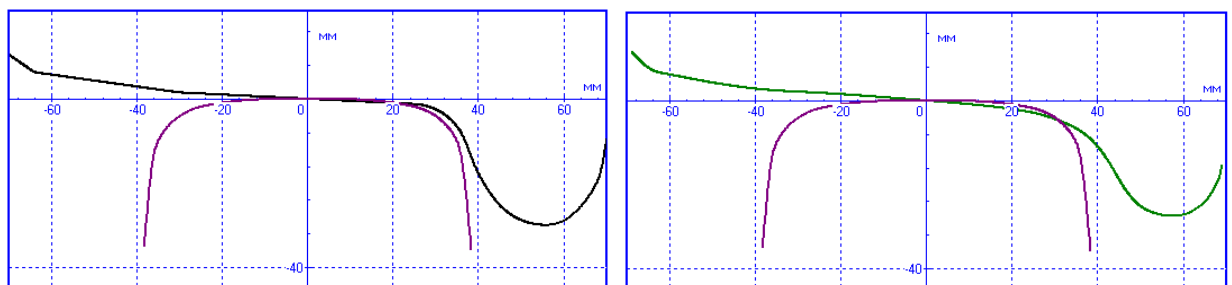


Figure 1.88. Pairs of profiles allowing two-point contact (left) and one-point contact only (right)

In a one-point contact mode the contact point position depends on the lateral shift  $\Delta y$  and the angle  $\Delta\alpha$ , Figure 1.87. Coordinates of the contact point correspond to the profile point with the minimal distance in the vertical direction (Sect. 1.3.1.1.1. "Algorithm for computation of nearest points between two profiles", 1-76). Thus, the table contains coordinates of the contact point in SCR and SCW for a discrete set of variables  $\Delta y_i, \Delta\alpha_j, i = 1,2 \dots N_y, j = 1,2 \dots N_\alpha$  with a proper step size of discretization.

Computation of one-point contact is realized according to the algorithm in Sect. 1.3.1.1.1. "Algorithm for computation of nearest points between two profiles", 1-76. Here we consider some features of computing a two-point contact.



Figure 1.89. Types of contact for a pair of profiles allowing two-point contact: one-point contact, two-point contact and creeping up

Considering that in case of one-point contact the coordinates computing is implemented using the algorithm, given in the previous paragraph, let us discuss the two-point contact computation aspects. Fix the value  $\Delta\alpha$  and give the maximal value of the lateral displacement (Figure 1.89, left). In this position of the profile we have obviously a one-point contact. Now decrease  $\Delta y$  without changing  $\Delta\alpha$ , i.e. shift the wheel profile to the left. There exists some critical value of the lateral coordinate  $\Delta y^*$  for which a two-point contact occurs, if the profiles allow it (Figure 1.89, center). Further decreasing  $\Delta y$  does not change coordinates of contact points, the rail profile will move to the left together with the wheel profile. This mode of the two-point contact can disappear in two different ways. Firstly, the wheel will move to the right and the flange contact disappears (Figure 1.89, left). Secondly, the contact on the running surface disappears (the corresponding normal force becomes zero) and the wheel goes to the creeping up state (Figure 1.89, right). This mental experiment is the base of an algorithm for computing the two-point contact. Let us consider the algorithm in details.

Let us choose the uniform discretization of the real change area of  $\Delta\alpha$  and  $\Delta y$  parameters. Now for each fixed value  $\Delta\alpha_j, j = 1,2 \dots N_\alpha$  compute coordinates of contact point on the running surface according to Sect. 1.3.1.1.1. "Algorithm for computation of nearest points between two profiles", p. 1-76 successively decreasing the lateral shift  $\Delta y_i, i = N_y, N_y - 1 \dots 1$ . If the profiles allow the two-point contact, the position of the contact point for some  $\Delta y_i$  will change by a large enough jump. This fact means that the flange contact occurs inside the latest change of  $\Delta y$ . Denote this value of  $\Delta y$  as  $\Delta y^*$ . Two conditions formalize the notion of a 'large enough jump' and give a criterion of passing through the two-point contact:

$$y_{r,i} - y_{r,i-1} > \eta_y, \quad n_{r,y}/n_{r,z} > \eta_n,$$

where  $y_{r,i}, y_{r,i-1}$  are the successive coordinates of the contact point on the rail (in SCR),  $n_{r,y}, n_{r,z}$  are the projections on the normal to the rail profile at the contact point at the latest step,  $\eta_y, \eta_n$  are some empirical criterion numbers, e.g.  $\eta_y = 20 \text{ mm}, \eta_n = 0.5$ . Thus, the two-point contact is considered to be found, if the contact coordinate on the rail profile changes by a large

jump, and the new coordinate of the contact point lies on the side of the rail. Value of parameters  $\eta_y, \eta_n$  can be modified by the user, Sect. 1.4.2.7.3. "Rail/wheel contact options", p. 1-185.

After confirmation of the fact, that the state of the two-point contact is crossed, the interval  $y_{r,i-1}, y_{r,i}$  is discretized on small subintervals (about 0.1 mm), and the critical value  $\Delta y_j^*$  is defined more exactly.

To model the wheel climb process, the computation is continued for  $\Delta y > \Delta y_j^*$ .

The computations are repeated for other values of the angle  $\Delta \alpha_j$ .

The contact tables contain coordinates of contact points for various values  $\Delta \alpha_j, \Delta y_i$  as well as the critical values of the lateral shift  $\Delta y_j^*$  for each value of the angle  $\Delta \alpha_j$ .

Computation of two-point contact geometry assumes evaluation of a wheel "overswing"  $x_c$ . i.e. the longitudinal coordinate of the flange contact for nonzero value of the angle of attack  $\gamma$ . UM uses an approximate analytic expression for obtaining the value of this parameter. To get it consider a simplified geometrical model of the flange-rail contact, Figure 1.90. In this model the side surface of the rail near the contact is replaced by a plane with the same normal, and the flange is replaced by a circle with a radius equal to the wheel radius at the contact.

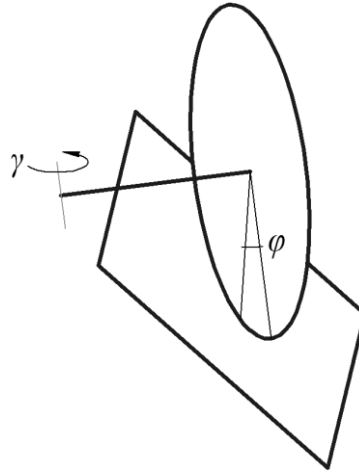


Figure 1.90. Simplified model of the flange contact for computing an overswing

Let  $\gamma$  be the angle of attack,  $\varphi$  is the angle defining the flange contact with the overswing. The  $\varphi$  value can be found from the condition that the tangent to the flange circle at contact is perpendicular to the plane normal  $\mathbf{n} = (0 \quad -n_{ry} \quad n_{rz})^T$ . The unit tangent vector in SC of the wheelset base is set by the following expression

$$\boldsymbol{\tau} = (r_w \cos \varphi \quad 0 \quad r_w \sin \varphi)^T$$

where  $r_w$  is the wheel radius at the flange contact.

Introducing the direct cosine matrix corresponding to the angle of attack,

$$\mathbf{A}_{01} = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

yields the following orthogonal condition

$$\mathbf{n}^T \mathbf{A}_{01} \boldsymbol{\tau} = 0$$

or, taking into account that angles  $\gamma$  and  $\varphi$  are small,

$$-n_{ry}\gamma + n_{rz}\varphi = 0$$

For small angles the overswing can be found as  $x_c = r_w\varphi$ , which results in the final formula

$$x_c = r_w n_{ry} / n_{rz}.$$

### 1.3.1.1.3. Simplified contact geometry. Equivalent conicity and contact angle parameter

Sometimes it is important to analyze the behavior of a vehicle depending on the wheel/rail profile wear level without exact profiles curves, and taking into account on the important notions of *equivalent conicity* and *contact angle parameter*.

Consider definitions of these parameters.

#### 1.3.1.1.3.1. Equivalent conicity

Consider a pair of rail/wheel profiles in a contact. When the wheel profile shifts in the lateral direction on distance  $y$ , the contact point change its position of the profile curves. Let  $\Delta r_l(y)$ ,  $\Delta r_r(y)$  be changes in radii of the left and right wheels. The value  $\Delta r = \Delta r_l - \Delta r_r$ , is noted as the *Rolling Radius Difference (RRD)*, and its plot depending on the lateral shift is used for evaluation of the equivalent conicity of a pair of profiles.

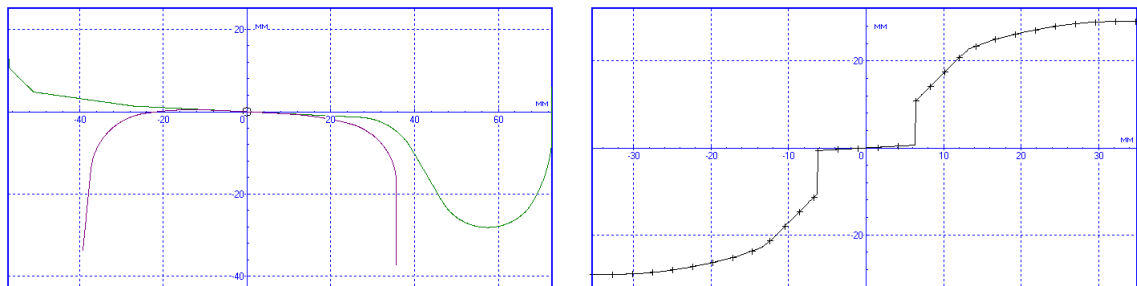


Figure 1.91. A pair of new profiles: R65 and Russian wagon wheel (left); the corresponding RRD (right)

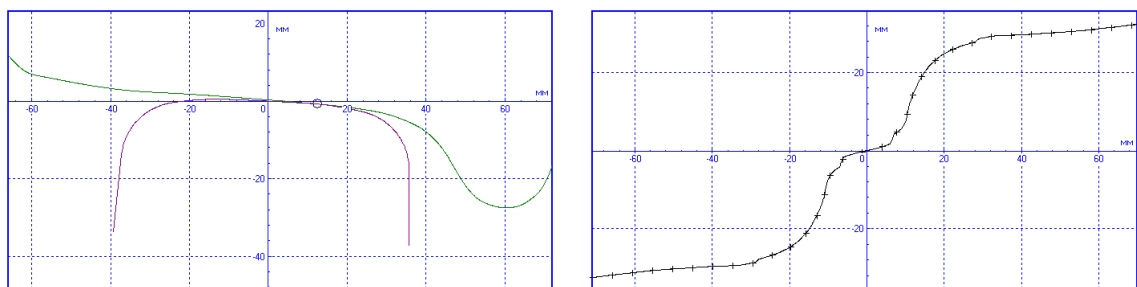


Figure 1.92. A Pair of profiles: R65 and DMetI (left), the corresponding RRD (right)

For the new R65 and wagon wheel profile the RRD is a straight line in the region of one point contact. The tangent of the line inclination angle is equal to the double wheel profile conicity  $\lambda = 1/20$ , Figure 1.91. Really, the wheel profile function is

$$r_l = r_0 + \lambda y, \quad r_r = r_0 - \lambda y,$$

and  $\Delta r = 2\lambda y$

RRD for worn or curved profiles are non-linear even for small shifts of the wheelset, Figure 1.92. In this case the notion of equivalent conicity  $\lambda$  is introduced as a mean value for a definite lateral shift of the wheelset  $\Delta y$ , according to the formula

$$\min_{\lambda} \int_0^{\Delta y} f(y)(\Delta r(y) - 2\lambda y)^2 dy, \tag{1.1}$$

where  $f_y$  is the distribution function for the lateral shift  $y$ . Thus, the equivalent conicity is computed to fit the nonlinear RRD by a linear function on the  $\Delta y$  interval. Often  $\Delta y$  is the shift of the wheelset corresponding to the start of the flange contact. The normal distribution

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}},$$

is used in Eq. (1.1) in the current UM version. The default value of the standard deviation is  $\sigma = 2,5 \text{ mm}$ .

Differentiating Eq. (1.1) with respect to  $\lambda$  yields the formula for the equivalent conicity as

$$\lambda = \frac{\int_0^{\Delta y} y f(y) \Delta r(y) dy}{2 \int_0^{\Delta y} f(y) y^2 dy}.$$

### 1.3.1.1.3.2. Contact angle parameter

Contact angle  $\beta$  is the angle between the normal to the profiles at the contact point and the perpendicular to the track plane, which is vertical for tangent sections, Figure 1.93.

The designation  $\beta_l(y), \beta_r(y)$  are introduced for the contact angles of the left and right wheels depending on the wheelset shift. If the left and right pairs of profiles are equal, the value  $\beta_l(0) = \beta_r(0) = \beta_0$  corresponds to the contact angle for symmetric position of the wheelset.

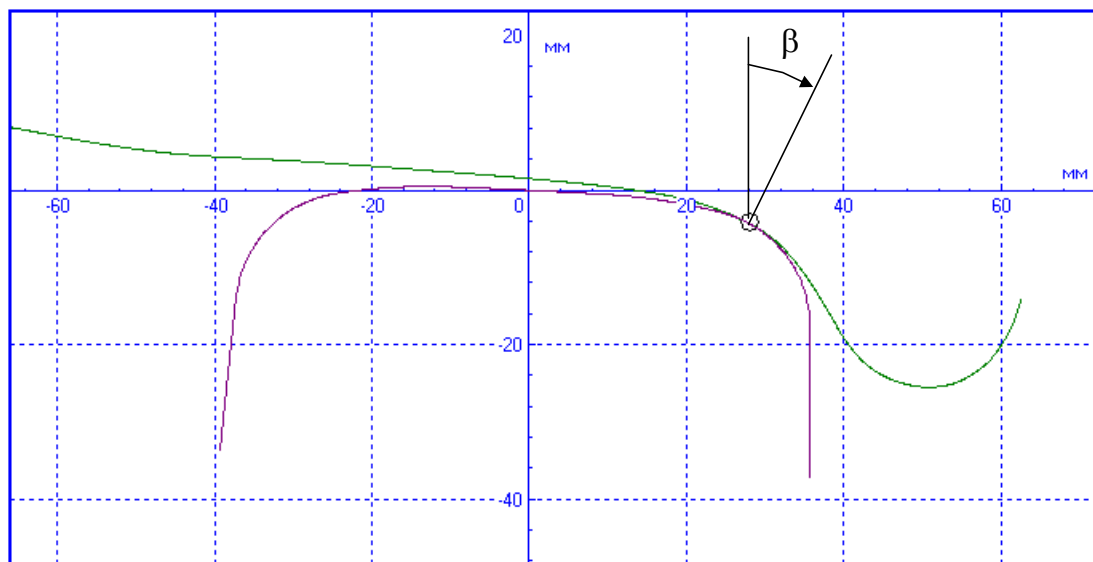


Figure 1.93. Contact angle

The dependence

$$E(y) = \frac{\beta_l(y) - \beta_r(y) S}{2} \frac{S}{2}$$

on the lateral shift is used for definition of the *contact angle parameter*. Here  $S$  is the distance between the wheelset rolling radii, Figure 1.8.

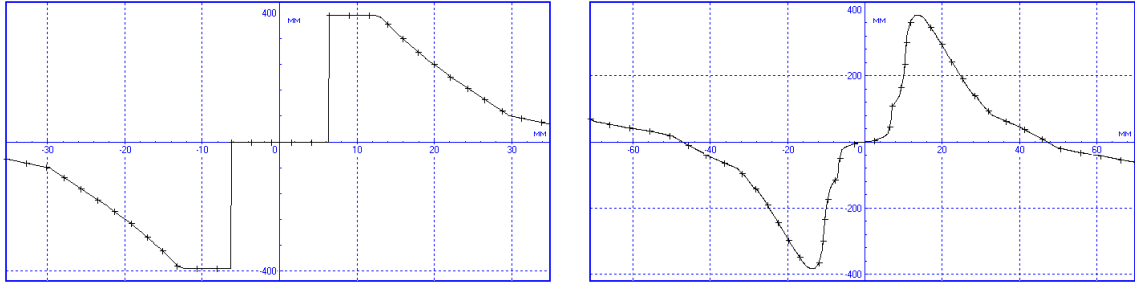


Figure 1.94. Function  $E(y)$  for pairs of profiles in Figure 1.88, Figure 1.89

The *contact angle parameter*  $\varepsilon$  is used for linear approximation of the function  $E(y)$  on the interval  $\Delta y$  of the lateral shift of the wheelset.

$$E(y) = \frac{\beta_l(y) - \beta_r(y) S}{2} \frac{S}{2} \approx \varepsilon y.$$

The formula is similar to the equivalent conicity is used for evaluation of  $\varepsilon$

$$\varepsilon = \frac{\int_0^{\Delta y} y f(y) E(y) dy}{2 \int_0^{\Delta y} f(y) y^2 dy}.$$

### 1.3.1.1.3.3. Simplified geometry of contact

The following simplified formulae are used for evaluation of wheel radii at contacts and for contact angles depending on the lateral shift of the wheelset in terms of the equivalent conicity and contact angle parameter:

$$\begin{aligned} r_l &= r_0 + \lambda y_l, & r_r &= r_0 - \lambda y_r, \\ \beta_l &= \beta_0 + \frac{2\varepsilon y_l}{S/2}, & \beta_r &= \beta_0 - \frac{2\varepsilon y_r}{S/2}. \end{aligned}$$

Here  $y_l, y_r$  are lateral shifts of the left and right wheels relative to the corresponding rails, which are in general different due to the elastic deflections of rails.

If the lateral shift of a wheel towards the rail is greater than some given value  $y^*$  then a double contact takes place. The contact point position on the side of the rail is specified by the coordinates of in the SCR  $(y_r^*, z_r^*)$  and by flange angle  $\beta^*$ . The longitudinal contact overswing is computed according to the results of the previous section.

### 1.3.1.2. Contact forces

#### 1.3.1.2.1. Method for computation of rail deflections and contact force

A rail model as a massless element is based on the following assumptions:

- deflections of a rail for different wheelsets are independent and can be computed separately;
- deflections of the left and right rails are independent;
- rail deflections include independent lateral  $\Delta y_r$ , vertical  $\Delta z_r$  deflections (Figure 1.95), which are parallel to the corresponding SC of the track;
- the rail as a linear force element both in the lateral and vertical directions; the lateral dissipation is taken into account for two-point contact mode only.

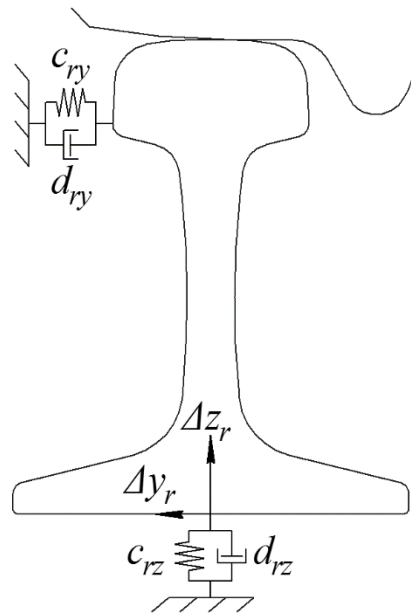


Figure 1.95. Rail as massless force element

Let  $c_y, c_z$  be the lateral and the vertical stiffness of the rail,  $d_y, d_z$  be the corresponding damping constants. Forces acting on the rail due to the deflections are the following:

$$\begin{aligned} R_y &= -c_{ry}\Delta y_r - d_{ry}\Delta \dot{y}_r, \\ R_z &= -c_{rz}\Delta z_r - d_{rz}\Delta \dot{z}_r, \end{aligned} \quad (1.2)$$

where  $c_{ry}, c_{rz}, d_{ry}, d_{rz}$  are stiffness and damping coefficients of rail in lateral and vertical directions correspondingly.

Because the rail has no mass, these forces must be balanced by contact forces acting on the rail from the wheel. The contact forces acting on the wheel for one- and two-point contacts are shown in Figure 1.96. Longitudinal forces are not shown in this figure.

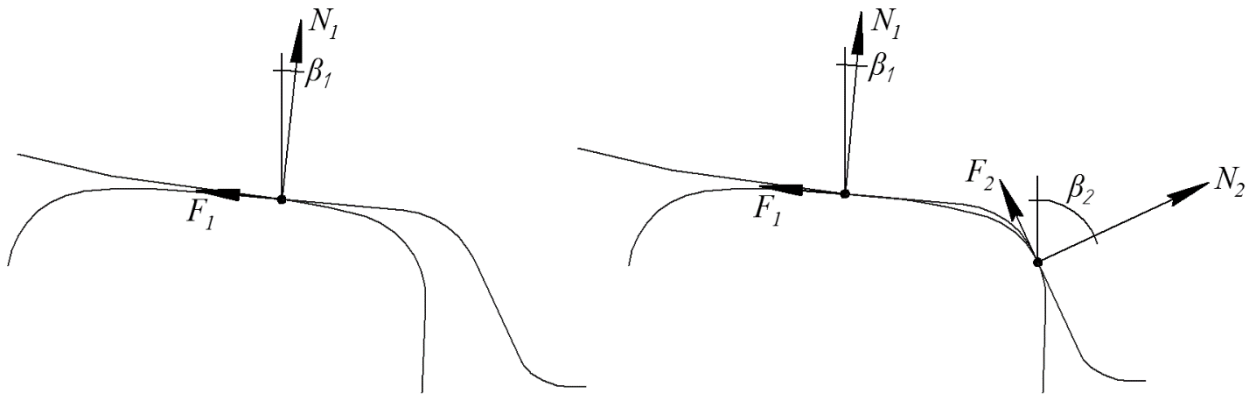


Figure 1.96. Forces acting on wheel at one- and two-point contacts

Equilibrium equations for one-point contact written in SC of the track are

$$\begin{aligned} R_y - F_1 \cos \beta_1 + N_1 \sin \beta_1 &= 0, \\ R_z - N_1 \cos \beta_1 - F_1 \sin \beta_1 &= 0. \end{aligned} \quad (1.3)$$

Analogous equations are valid for a two-point contact

$$\begin{aligned} R_y - F_1 \cos \beta_1 + N_1 \sin \beta_1 - F_2 \cos \beta_2 + N_2 \sin \beta_2 &= 0, \\ R_z - N_1 \cos \beta_1 - F_1 \sin \beta_1 - N_2 \cos \beta_2 - F_2 \sin \beta_2 &= 0. \end{aligned} \quad (1.4)$$

Here  $\beta_1, \beta_2$  are the angles between the normal to the rail at contact and the axis perpendicular to the track.

Eq. (1.3) for a one-point contact and (1.4) for a two-point contact are complicated systems of nonlinear algebraic equations relative to unknown deflections of the rail and normal reactions  $N_1, N_2$ . Consider the main ideas for their solving without going into details.

- Position of the wheel as well as the rail shift due to irregularities, gauge widening and cant are known by the computing the contact problem; the only unknown are deflections  $\Delta y_r, \Delta z_r$  and their time derivatives. Note that the vertical deflection  $\Delta z_r$  is not an independent variable because when the lateral  $\Delta y_r$  deflection is known, the value of  $\Delta z_r$  can be obtained from the geometry of the contact. This value is equal to the value  $\delta z$ , Sect. 1.3.1.1.1. "Algorithm for computation of nearest points between two profiles", p. 1-76.
- Calculation of the rail deflections and contact forces is an iterative process. Iterations include two cycles: internal and external. The internal iterations are used for solving equations (1.3), (1.4) for known values of tangential forces (lateral creep forces). External iterations calculate creep forces. Values of creep forces on the previous integration step are used as initial approximations. Thus, the contact computation looks like this: internal cycle of iterations computes lateral deflection of the rail and normal force/forces in contact/contacts, the lateral creep forces are taken from the previous step. After that the new values of creep forces are computed. When the new values differ from the previous ones more than an error tolerance, the external iterations start, and equations (1.3), (1.4) are solved for corrected values of creep forces.

Consider some features of realization of internal iterations.

1. Lateral deflection  $\Delta y_r$  is the sum of two components:  $\Delta y_r = \Delta y_{r1} + \Delta y_{r2}$ .

- The first component corresponds to the rail deflection by forces at the first contact point, the second one differs from zero by the two-point contact and results from the forces at the flange contact  $N_2, F_2$ .
  - The previous value of the  $\Delta y_{r1}$  variable is used as the initial value for iterative solving the equilibrium equations. Iterations make the value more accurate.
2. Each of the iterations includes:
- Evaluation of the wheel profile position relative to the rail  $\Delta y, \Delta \alpha$  according to the known data (position of the wheel, irregularities, gauge widening etc. These parameters are used for determination of the contact type (one- or two-point) as well as for interpolation of coordinates of the contact point (or points) with the help of the preliminary computed tables (Sect. 1.3.1.1.2. "Computing tables of contact points", p. 1-78), computing the vertical rail deflection  $\Delta z_r$ , the angle  $\beta_1$  (and  $\beta_2$  for the two-point contact). The second part of the rail deflection  $\Delta y_{r2}$  is computed from the geometrical conditions.
  - Forces  $R_y, R_z$  are computed according to Eq. (1.2).
  - The normal force  $N_1$  for a one-point contact is obtained from Eq. (1.3); in the case of a two-point contact the normal reactions  $N_1$  and  $N_2$  are computed from Eq. (1.4).
3. The new value of the deflection  $\Delta y_{r1}$  is computed and stored for the next iterations.

**1.3.1.2.2. Algorithms for computing creep forces**

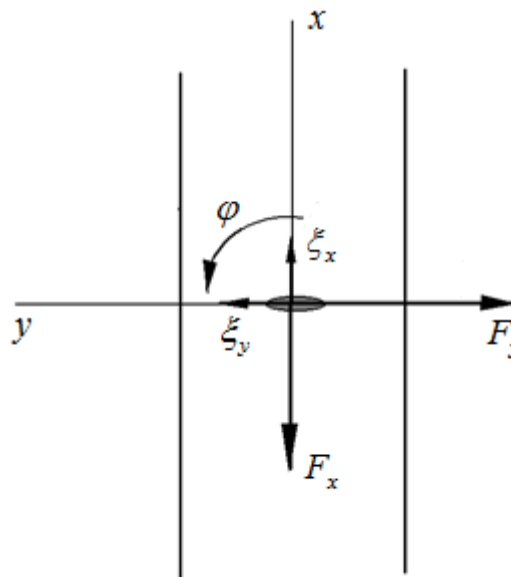


Figure 1.97. Creepages, spin and creep forces

Modern models of tangential forces in a wheel-rail contact are based on nonlinear dependencies of the general form:

$$F_x = F_x(N, \xi_x, \xi_y, \varphi, f, p), \quad F_y = F_x(N, \xi_x, \xi_y, \varphi, f, p).$$

Here the following notations are used:

$F_x, F_y$  are the longitudinal and lateral creep forces lying in the tangential plane of the rail;

$N$  is the normal force in the contact;

$\xi_x, \xi_y$  are the longitudinal and lateral creepages;

$\varphi$  is the spin;

$f$  is a friction coefficient in contact point;

$p$  is a set of geometrical parameters characterizing rail and wheel profiles, e.g. curvatures of contact surfaces in the case of the FASTSIM algorithm.

As it is known, the creepages and the spin satisfy the following relations:

$$\xi_x = v_x/v_0, \quad \xi_y = v_y/v_0, \quad \varphi = \omega_n/v_0,$$

where  $v_x, v_y$  are the corresponding components of sliding velocity at the contact point on the wheel relative to the rail;  $v_0$  is the longitudinal velocity of the wheelset;  $\omega_n$  is the projection of the wheel angular velocity on the normal to the rail at the contact point.

**Note.** For slow-speed movements up to the zero value initial velocity when  $v_0 < v_0^* = 0.1$  m/s the following formulae to compute creep and spin are applied:

$$\xi_x = v_x/v_0^*, \quad \xi_y = v_y/v_0^*, \quad \varphi = \omega_n/v_0^*$$

Parameter  $v_0^*$  is the **critical speed for creep computation**, which value can be changes by the user, see Sect. 1.4.2.5. "*Parameters for computation of rail-wheel contact forces*", p. 1-167.

For the wheel contact model with the massless rail the following algorithms for creep forces computing are implemented: *Mueller's method*, *Minov's method*, *FASTSIM*, *FASTSIM\_A*.

### 1.3.1.2.2.1. Mueller's model

*Mueller's method* is the simplest one for computation of the creep forces according to the following analytic expressions:

$$F_{xy} = \frac{1000k_c}{\sqrt[m]{1 + \left(\frac{k_c \xi}{fP}\right)^m}},$$

$$\xi = \sqrt{\xi_x^2 + \xi_y^2}, \quad P = 0,001N, \quad k_c = (235 - (2,4 - 0.01P)P)P$$

$$F_x = -\xi_x F_{xy}, \quad F_y = -\xi_y F_{xy}.$$

Thus, the model is very simple. In particular, forces do not depend on the spin. If the two-point contact is presented, forces at the flange contact are computed as simple friction forces in the sliding mode.

Some advantage of the algorithm consists in its simplicity. The disadvantage is its lower accuracy especially for a one-point flange contact where the spin is not small.

**Note.** Mueller's method is also used to compute creep forces at a simplified profile setting using the equivalent conicity and contact angle parameter (see Sect. 1.3.1.1.3.3. "*Simplified geometry of contact*", p. 1-84).

**1.3.1.2.2.2. Minov's model**

The method is used in simulation of locomotive in traction modes, and based on experimental dependence of adhesion force on creep, Figure 1.98. K.D. Minov supposed an analytic approximation for this curve consisting of three sections:

1.  $0 \leq |\xi| \leq 0,0014$ : linear section of elastic sliding

$$k = 359,61178|\xi|$$

2.  $0,0014 \leq |\xi| \leq 0,025$ : nonlinear section of elastic sliding

$$k = \frac{350|\xi| - 0,155}{0,195 + 336|\xi|}$$

3.  $|\xi| > 0,025$ : sliding

$$k = \frac{1}{1 - \chi v_0(0,025 - |\xi|)}$$

where  $k$  is the ratio of the force to its maximal value;  $v_0$  is the vehicle speed, m/s;  $\chi$  is the stiffness of the third section of the curve, s/m. The stiffness  $\chi$  depends of the speed  $v_0$  as in the Table 1.2.

Table 1.2

**The dependence of  $\chi$  stiffness on longitudinal movement of the  $v_0$  wheelset**

$v_0$ , km/h	0-5	5-20	20-40	40-120
$\chi$	0,9	0,6	0,5	0,35

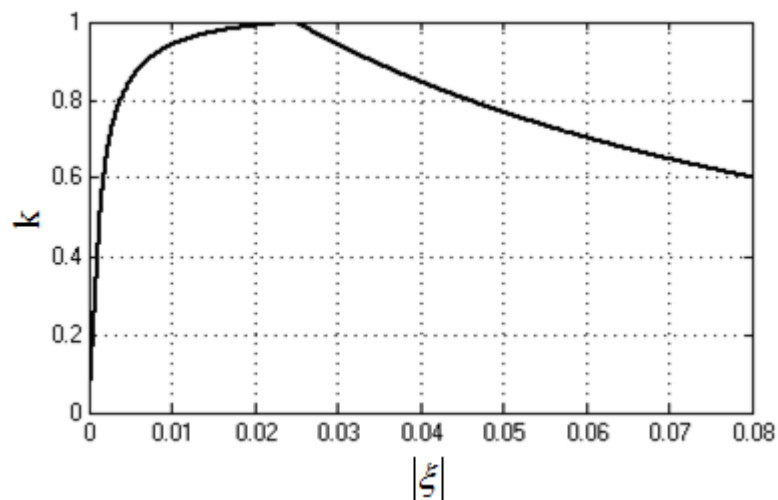


Figure 1.98. Sticking coefficient versus creep in the *Minov's* model

Creep forces are computed according to the formulas

$$F_{xy} = f \cdot N \cdot k(|\xi|),$$

$$\xi = \sqrt{\xi_x^2 + \xi_y^2},$$

$$F_x = -F_{xy} \frac{\xi_x}{\xi}, \quad F_y = -F_{xy} \frac{\xi_y}{\xi}.$$

In case of a two-point contact, the flange friction forces are computed as for the pure sliding.

*Minov's* method allows to calculate creep forces for locomotives in traction mode (braking) and to simulate electro-mechanical processes in coupling failure. The advantage of the algorithm is in its simplicity and high-speed computing.

### 1.3.1.2.2.3. FASTSIM

*FASTSIM* algorithm, based on Kalker's creep forces linear theory [2], has become a standard of creep forces computing in rail-wheel contact in the multibody dynamics software for rail vehicle dynamics simulation.

The following parameters and variables are the initial data for *FASTSIM* algorithm.

- **Rail and wheel material properties**, which are supposed to be equal (Young's modulus and Poisson's ratio) are set by the user;
- **Current geometric characteristics of the contact point**: principal curvatures of the contact surfaces are computed by the program;
- **The normal force  $N$**  in the contact is computed by the program.
- **Semi-axis of the elliptic contact patch** are calculated by the program using Herz's theory in the dynamic modeling process;
- **Longitudinal and lateral  $\xi_x, \xi_y$  creepages and  $\varphi$  spin** are calculated by the program in the dynamic modeling process;
- **Friction coefficient in the contact point** is calculated by the program in the dynamics modeling process using parameters defined by user.

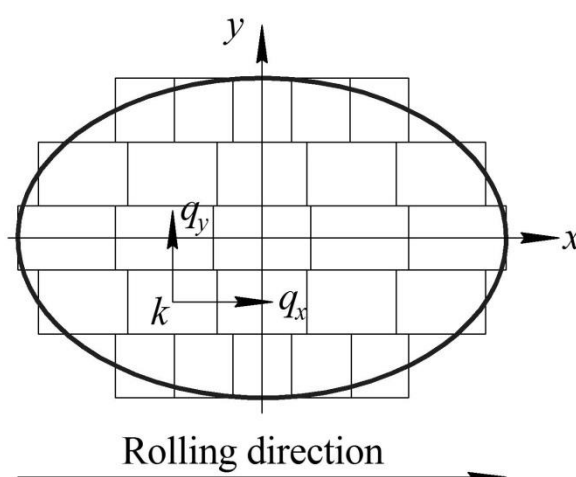


Figure 1.99. Discretization of the contact patch

According to this data *FASTSIM* solves a system of differential equations (in the adhesion area of the contact patch) or a system of differential-algebraic equations (in the sliding area of the contact patch) relative to tangential stresses. For this purpose the contact ellipse is divided into a

number of narrow slices of the same width. In turn, each slice is divided into  $n$  elements of equal length within one slice (Figure 1.99). Number of slices  $m$  and elements  $n$  is set by the user. The default values are 10.

To compute the creep forces and to obtain adhesion and sliding areas of the patch, *FASTSIM* solves the above equations for each of the slice successively. In fact, the discretization on elements gives the constant step size for numeric solving the differential equations by the explicit Euler method. Thus, CPU expenses are of order  $m \times n$  operations, and depend on the discretization level.

#### 1.3.1.2.2.4. FASTSIM\_A

*FASTSIM\_A* is a semi-analytic modification of the classical *FASTSIM* algorithm. For a slice it was found an exact solution of the *FASTSIM* governing differential equations in the adhesion area of the contact patch and an approximate analytic solution for differential-algebraic equations in the sliding area. The solution was implemented in UM as *FASTSIM\_A* (*FASTSIM* – Analytic) procedure. The number of operation for computing creep forces is proportional to the number of slices  $m$ . The procedure for  $m = 10$  is at least two times faster than *FASTSIM* and gives quite similar numeric results for computing creep forces in case of moderate spin values ( $\varphi < 0,5$ ) [3].

### 1.3.2. Computation of contact between the wheel and the inertial rail

When a rail is considered as an inertial element (rigid body or flexible beam) then in a wheel-rail contact the rigid interpenetration of wheel and rail profiles is allowed. At the same time, the amount of contact points is formally unlimited. The function of the distance between the undeformed profiles defines the form and size of the contact patches.

Modification of the algorithm, described in Sect. 1.3.1.1.1. "*Algorithm for computation of nearest points between two profiles*", p. 1-76, is used to define the points of intersection of profiles. It is obvious that the point of intersection of profiles is determined if the value  $\delta z$  has changed its sign.

Coordinates of contact points in the profiles coordinate system are defined using the following system of equations:

$$\begin{cases} \mathbf{n}_w \times \mathbf{n}_r = 0, \\ \mathbf{n}_r \times \mathbf{d} = 0, \end{cases}$$

where  $\mathbf{n}_w$  and  $\mathbf{n}_r$  are the normals to the wheel and rail profiles in CSR0,  $\mathbf{d} = \mathbf{r}_{p_r^c} - \mathbf{r}_{p_w^c}$ , see Figure 1.100.

This system of vector equations defines the two points corresponding to the maximum interpenetration of the wheel and rail profiles. The first equation of the system defines the condition of collinearity of the normals to the profiles in the points. The second equation defines the condition of collinearity of the one of the normals to the radius vector which connects the points. System of vector equations forms the system of two non-linear algebraic equations relative to  $y_r$  and  $y_w$ . A good initial approximation when solving the obtained system using Newton-Raphson algorithm is the two points that fit the minimum condition of the function  $\delta z$  in the considered area

of the profiles intersection. If smoothness of the profiles does not provide the convergence of Newton-Raphson's method, then the brute-force method for finding contact points is used.

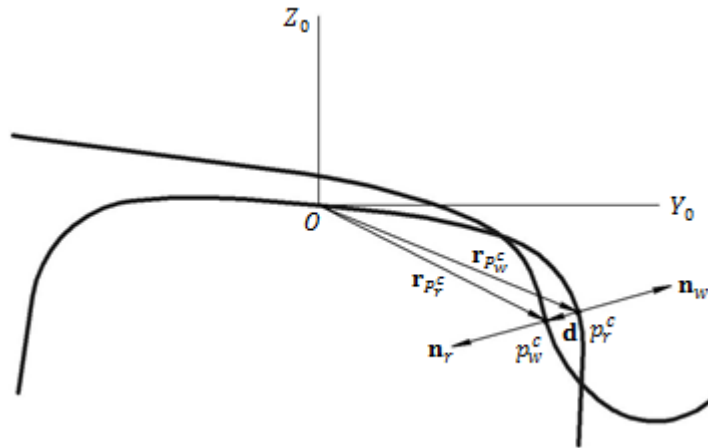


Figure 1.100. Determination of contact points

It takes too much computational efforts (up to 60% of CPU time) to define the points of intersection of profiles and to search contact points. If the rail profile is constant along the track, the calculation of the coordinates of the potential contact points is performed once before the start of the modeling process. The coordinates of the potential contact points depending on the discrete shifting of the wheel profile relative to the rail profile are written in tables. At the same time it is considered that the wheel profile relative to the rail profile has two degrees of freedom: rotation around the longitudinal axis and lateral displacement. The potential contact points are the points in which the value  $\delta z$  has the minimum, see Figure 1.101. In the simulation process for the current profiles position the potential contact points are computed with the help of the precalculated tables with interpolation usage.

After finding the contact points for each potential contact area, the local coordinate system of the contact patch (SCC) is introduced. SCC origin is located in contact point on the rail profile, Z-axis is directed along the normal to the wheel and rail profiles, Y-axis is directed along the tangent. The function of the distance between the undeformed profiles is defined in SCC, Figure 1.102.

Creeps and spin are computed according to the following formulas:

$$\xi_x = \frac{(\mathbf{v}_w - \mathbf{v}_r)\boldsymbol{\tau}_1}{v_0}, \quad \xi_y = \frac{(\mathbf{v}_w - \mathbf{v}_r)\boldsymbol{\tau}_2}{v_0}, \quad \varphi = \frac{(\boldsymbol{\omega}_w - \boldsymbol{\omega}_r)\mathbf{n}}{v_0},$$

where  $\mathbf{v}_w, \mathbf{v}_r$  are the forward velocities of the contact point;  $\boldsymbol{\omega}_w, \boldsymbol{\omega}_r$  are the angular velocities of a wheelset and rail respectively.  $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \mathbf{n}$  are the tangents and normal to the contact point in CSR0;  $v_0$  is the longitudinal velocity of the wheelset.

Contact forces are computed using the models by W. Kik and J. Piotrowski [4] or CONTACT program [5]. In the model by W. Kik and J. Piotrowski FASTSIM algorithm, modified for non-elliptical contact patches, is used to compute creep forces.

Use the following formula to calculate the total normal reaction:

$$N = N_e + N_d,$$

where  $N_e$  is an elastic force (which is computed using the corresponding contact forces model),  $N_d$  is a damping force.

$$N_d = 2\zeta\sqrt{c_c m_w}|\dot{\delta}|\frac{\text{sign}(\dot{\delta}) - 1}{2}$$

where  $\zeta$  is a damping ratio,  $c_c$  is a contact stiffness,  $m_w$  is a mass of the wheel,  $\dot{\delta} = (\mathbf{v}_w - \mathbf{v}_r)\mathbf{n}$  is a penetration velocity.

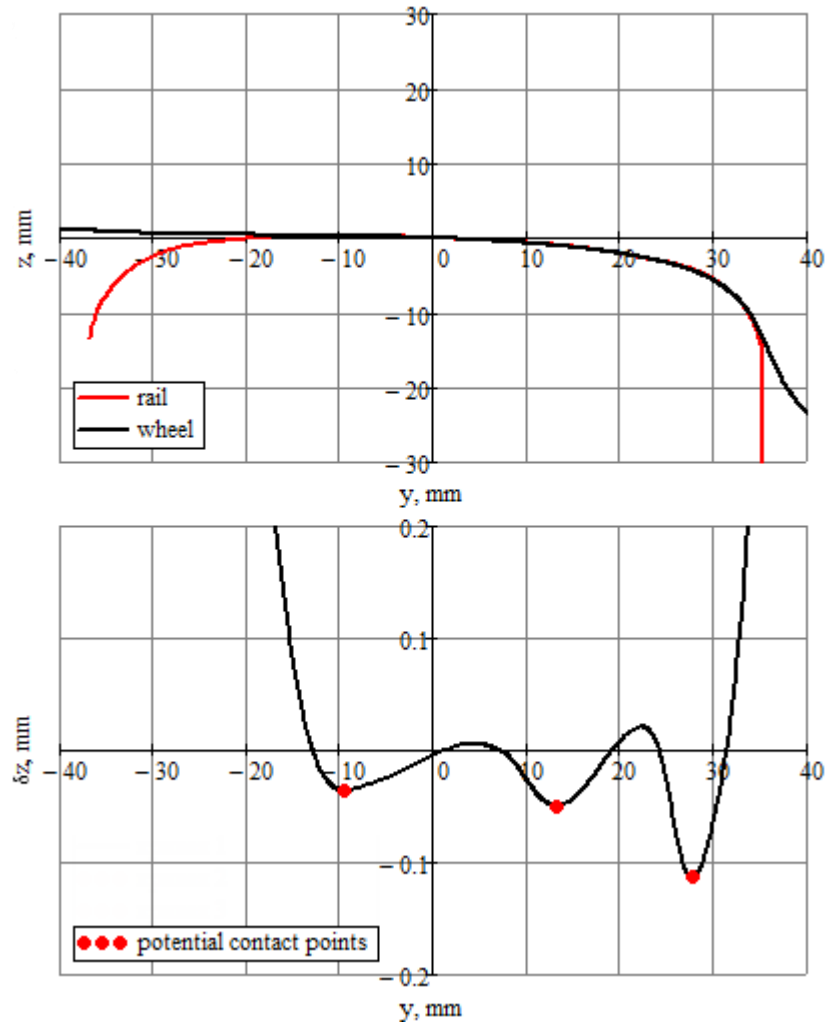


Figure 1.101. Relative position of profiles and  $\delta z$  function

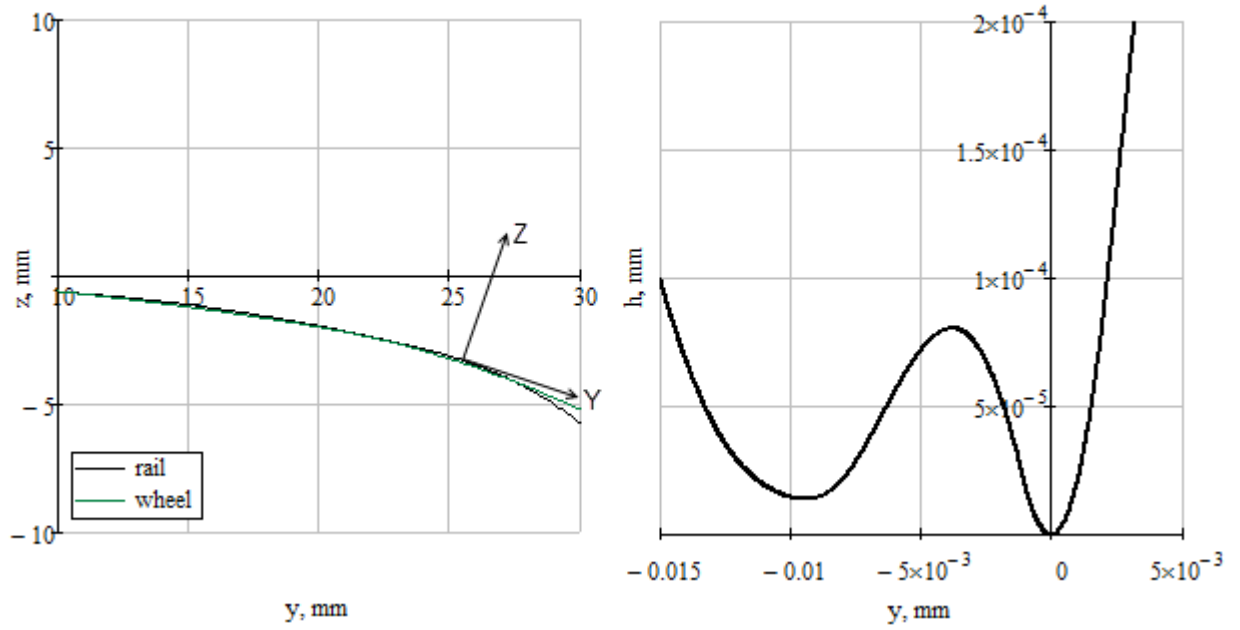


Figure 1.102. Local coordinate system of the contact patch (on the left) and the function of the distance between the undeformed profiles (on the right)

### 1.3.3. Coefficient of friction in wheel-rail contact

UM uses both variable and constant coefficients of friction in contacts of the wheel and rail. Two main coefficients of friction are introduced for each of two rails. The first one is the coefficient of friction on the rail running surface  $f_r$ , the second one is on the inward rail side  $f_s$ . These coefficients are defined either by numeric constants or by functions of the longitudinal coordinates along the track  $f_r(x), f_s(x)$  or on time  $f_r(t), f_s(t)$ .

If the coefficients for a given longitudinal position have different values, i.e.  $f_r \neq f_s$ , the coefficient of friction is considered as a variable one on the rail profile in the lateral direction.

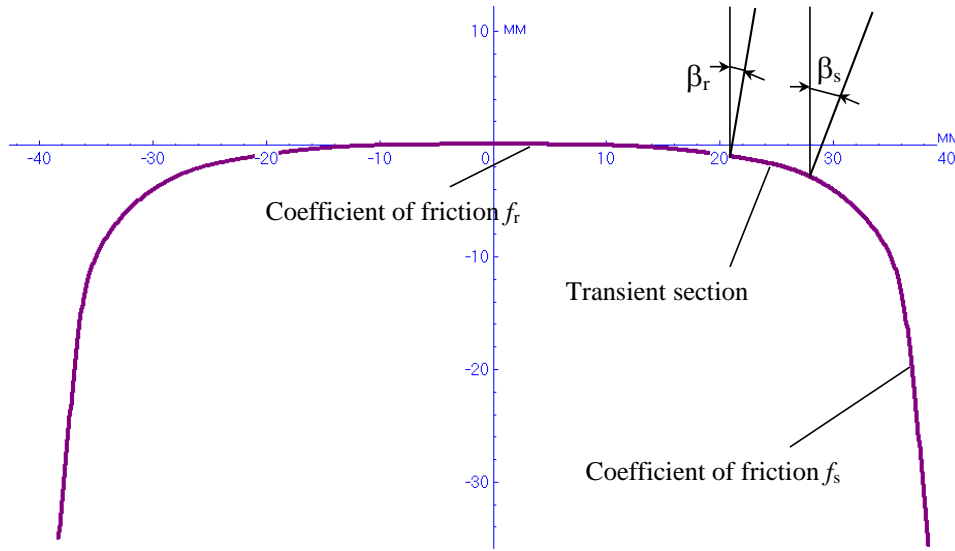


Figure 1.103. Changing the coefficient of friction along the rail profile

As a result, the profile is divided into three parts, Figure 1.103. The first one is the running surface with a constant coefficient  $f_r$ . The second part is the inward side of the rail with the coefficient  $f_s$ . Finally, a transient section between the previous two parts, where the coefficient changes continuously from  $f_r$  to  $f_s$ . The transition is linear in the angle  $\beta$ . To divide the profile on these sections, the angles  $\beta_r, \beta_s$  should be set.

Dependences of the coefficient of friction on the longitudinal coordinate allow the user to model e.g. an oil stain on a rail.

Different values of the coefficient on the running surface of the rail and on its side are used mainly for modeling lubrications in curves.

Dependence of the coefficient of friction on sliding velocity is implemented in UM 6.0 according to the model [6]

$$f = f_0((1 - A)e^{-Bv_1} + A),$$

Here  $f_0$  is the coefficient value for zero sliding,  $A = f_\infty/f_0$  is the ratio of coefficient for infinite and zero sliding velocities,  $B$  is the factor of exponential decrease of the coefficient of friction.

Decrease of the coefficient of friction is shown in Figure 1.104 for  $A = 0.4, B = 0.6$  s/m.

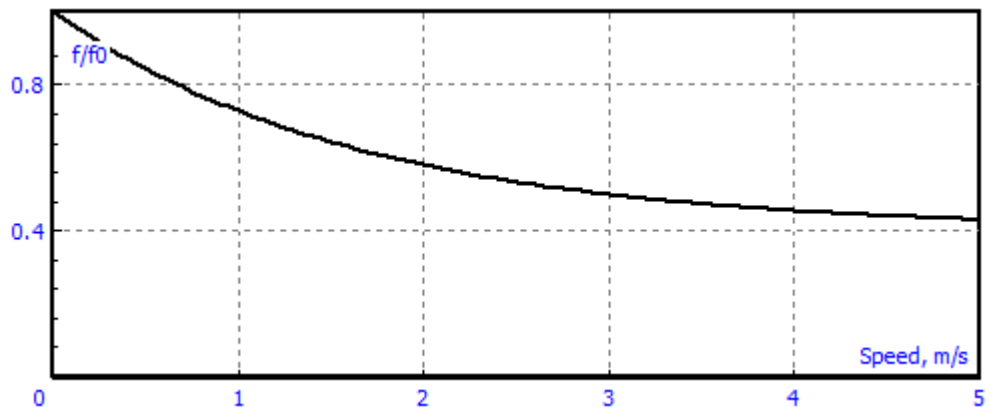


Figure 1.104. Decrease of coefficient of friction with the growth of sliding velocity

## 1.4. Simulation of railway vehicles

### 1.4.1. Tools for preparing simulation process

#### 1.4.1.1. Creation of wheel and rail profiles

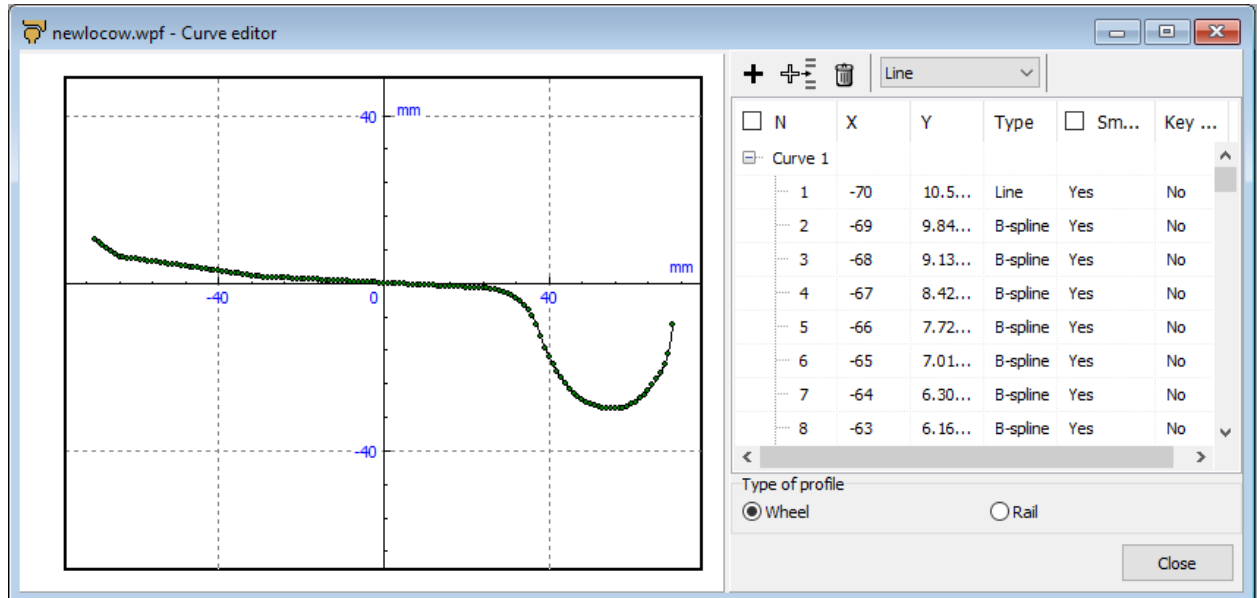


Figure 1.105. Editor of curves for creation of profiles

Wheel and rail profiles are located in the  $\{\text{UM Data}\}\backslash\text{rw}\backslash\text{prf}$  directory in form of separate files with extension *\*.wpf* (wheels) and *\*.rpf* (rails). The profiles are described in a special systems of coordinates (Sect. 1.1.3.5. "Wheelset geometry", p. 1-13, Sect. 1.2.1.1. "Geometry of rails in an ideal track", p. 1-66) with the help of a special tool, which is available in the **UM Simulation** program by clicking the **Tools | Railway wheel and rail profile editor**. Creation of new profiles and modification of old ones are made with the help of the **Curve editor** (Figure 1.105). Detailed description of the editor can be found in [Chapter 3](#), Sect. *Object constructor/Curve editor*. Creation of new profiles is possible in two modes:

- input as a set of points with successive spline approximation;
- input of profiles as a set of line segments and circle arcs.

The second type of the profile description is used mainly for new standard profiles.

**1.4.1.1.1. Input as a set of points with successive spline approximation**

Use either list of points in the left part of the editor window or clipboard for input of a profile as a set of points. The points must be ordered from the left to the right (according to increase their abscissa). The unit for data is millimeter.

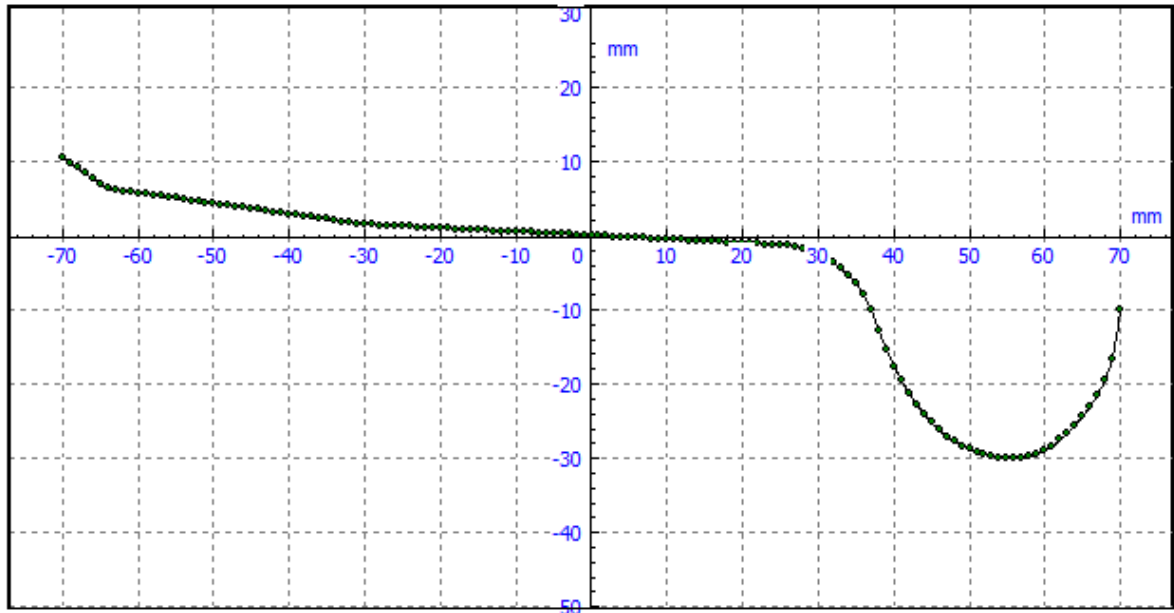


Figure 1.106. Input as a set of points


For input from the clipboard points should be written as a text in two columns. The first column contains abscissa values, the second one – the ordinate values:

```
-68.9    11.7
-66.4    8.88
-63.9    6.98
-61.4    6.48
-58.9    5.99
.....
```

To set points from the clipboard to do the follows:

- clear the editor;
- copy the new data to the clipboard from any text editor in a standard manner;
- activate the curve editor by the mouse and paste data from the clipboard (*Ctrl+V* or *Shift+Insert*).

When all the points are set into the editor, select the single profile by the mouse and set the **B-spline** interpolation type (Figure 1.108).

Save the profile with the help of the  button on the toolbar of the editor.

**Remark 1.** It is not recommended to use step size for abscissa less than 1 mm. Otherwise the first derivative plot might look as a saw, and the curvature plot might have large overfall, see Figure 1.107. Such cases may lead to deterioration of the contact

point computation process and to violation of continuity the contact point coordinate.

**Remark 2.** Smoothed profile curvature is used by computation of creep forces with the help of the FASTSIM algorithm.

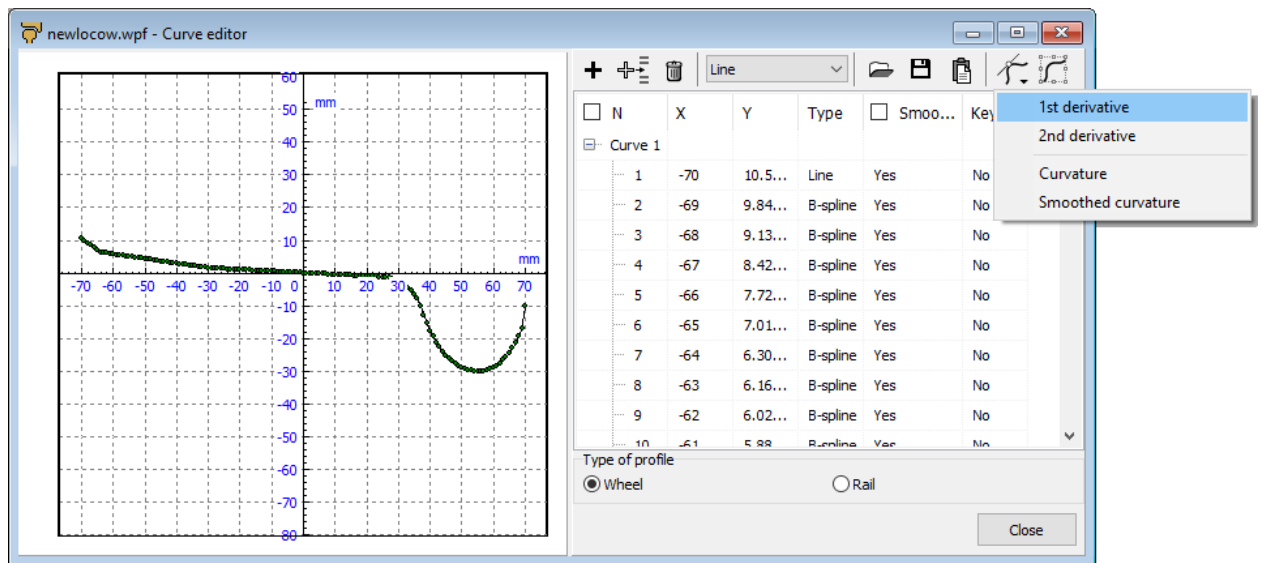


Figure 1.107. How to create profile derivatives and curvature

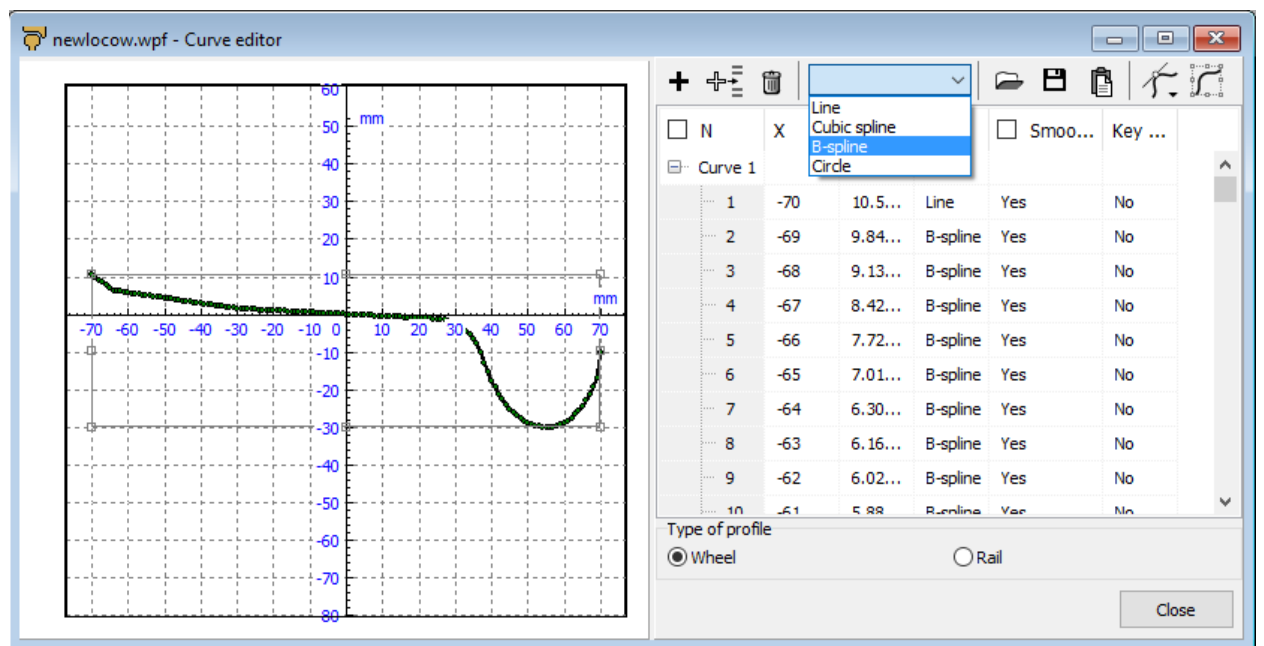


Figure 1.108. Selection of curve and setting the interpolation

#### 1.4.1.1.2. Input of profiles as a set of line segments and circle arcs

To create a profile as a set of line segments and circle arcs:

1. set coordinates of end points of line segments and arcs as a broken line from the left to the right (increasing abscissa);
2. select by the mouse a section of a sequence of sections, which should be replaced by circle arcs and set the **Circle** item as the type of sections (Figure 1.109);

3. save the profile.

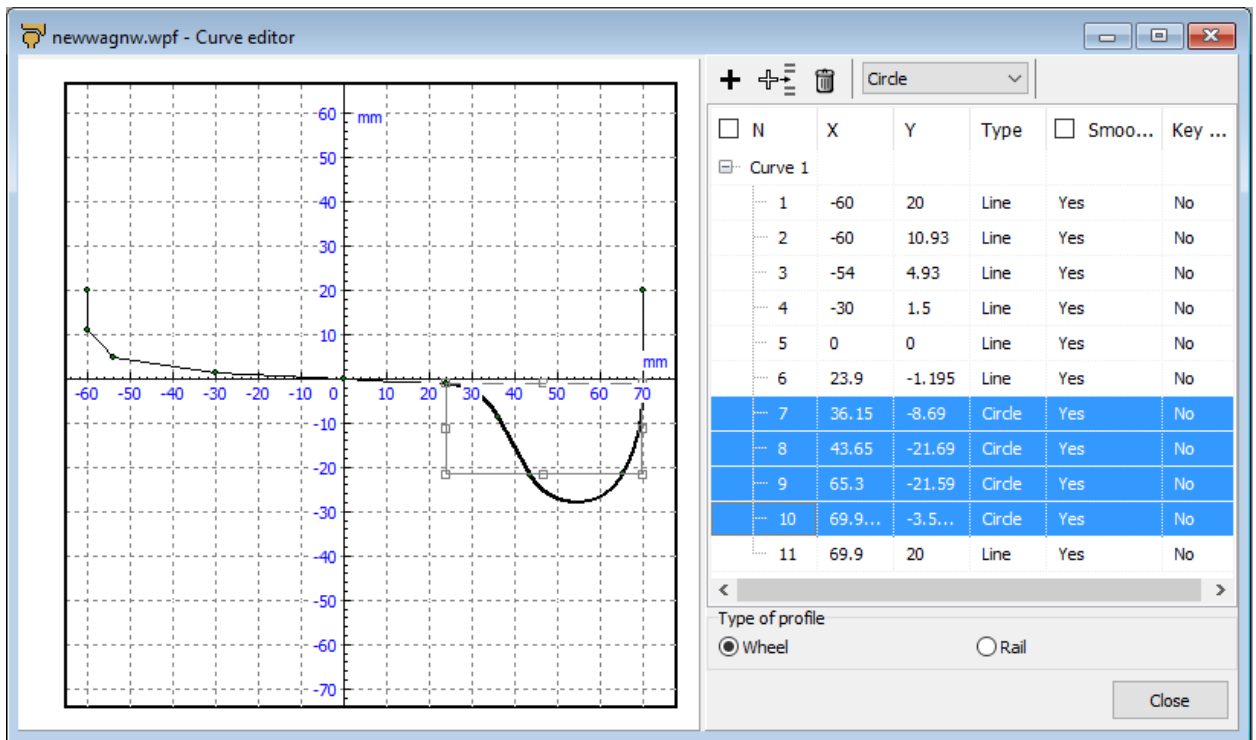
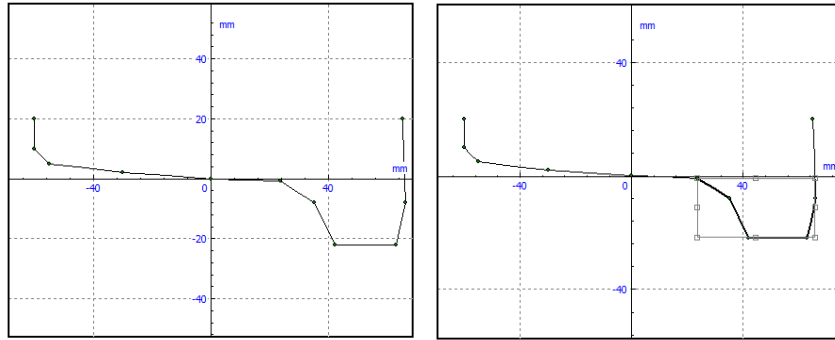


Figure 1.109. Creation of profile as a set of line sections and circle arcs

1.4.1.1.3. Tool for analysis of pairs of profiles

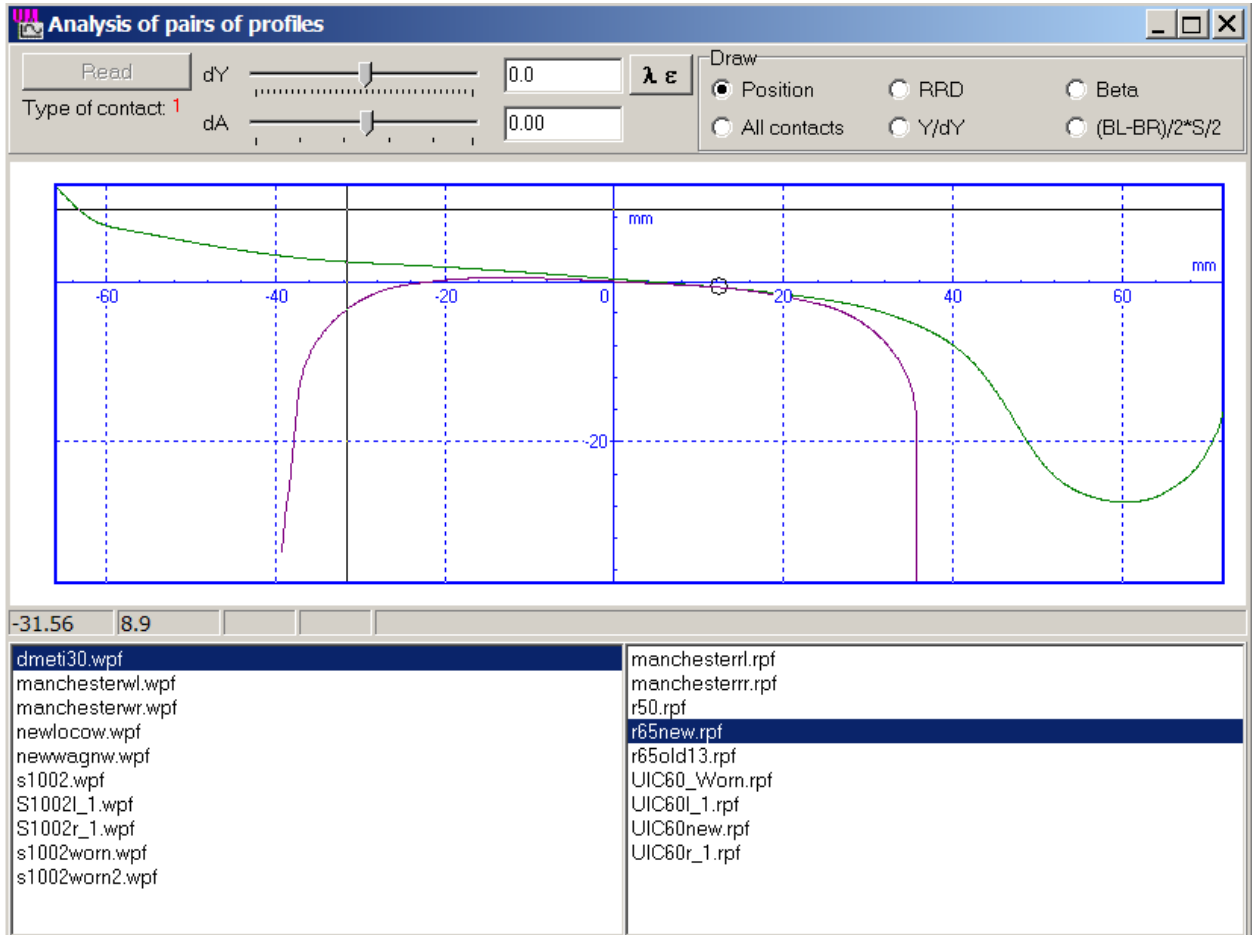


Figure 1.110. Profile analysis window

The tool for analysis of profiles is available in the simulation program for the current values of the track gauge and rail inclination (Sect. 1.2.1.1. "Geometry of rails in an ideal track", p. 1-66). Use the **Tools | Analysis of pairs of profiles** menu command.

**Container with profiles**

The container contains lists of wheel (left) and rail (right) profiles located in the standard directory **{UM Data}\rw\prf**.



To add profiles from other directories to the list, click the right mouse button on the corresponding part of the container, select one of two menu commands ('without sorting' adds profiles to the end of the list), and open the file with the help of the standard dialog window

**Selection of a pair of profiles**

To choose profiles for analysis, select them in the lists by the mouse Figure 1.110, and click the **Read** button on the top of the window.

### Analysis of results

The following results are available with the tool.

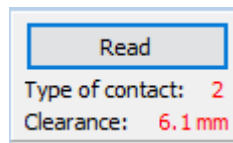


Figure 1.111. Type of contact and flange clearance on a side

- **Type of contact** (one or two-point contact), Figure 1.111.
- **Clearance** is the lateral shift of the wheel relative to the rail, which lead to the double contact (for profiles allowing a two-point contact), Figure 1.111.

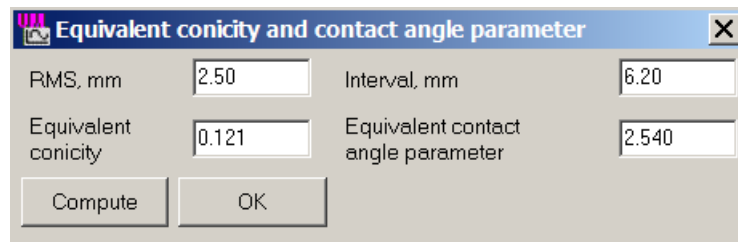


Figure 1.112. Equivalent conicity and contact angle parameter

- $\lambda, \varepsilon$  – Equivalent conicity and contact angle parameter, Sect. 1.3.2. "*Computation of contact between the wheel and the inertial rail*", p. 1-91. The parameters are computed for given values of the standard deviation of the lateral shift (RMS, mm) and averaging interval, Figure 1.112.

The following graphic information is available depending on the option selected in the **Draw** group.

- **Position** shows contact points for different values of the wheel shift relative to the rail  $dY$  as well as small wheel rotation angle relative to the longitudinal axis  $dA$ , Figure 1.109. To change values of  $dY$  (mm) and  $dA$  (degrees), use either track bars or direct input in the text boxes. After direct input in a text box, use the **Enter** key to redraw the contact. The angle  $dA$  changes in the interval from -1.4 to 1.4 degrees.

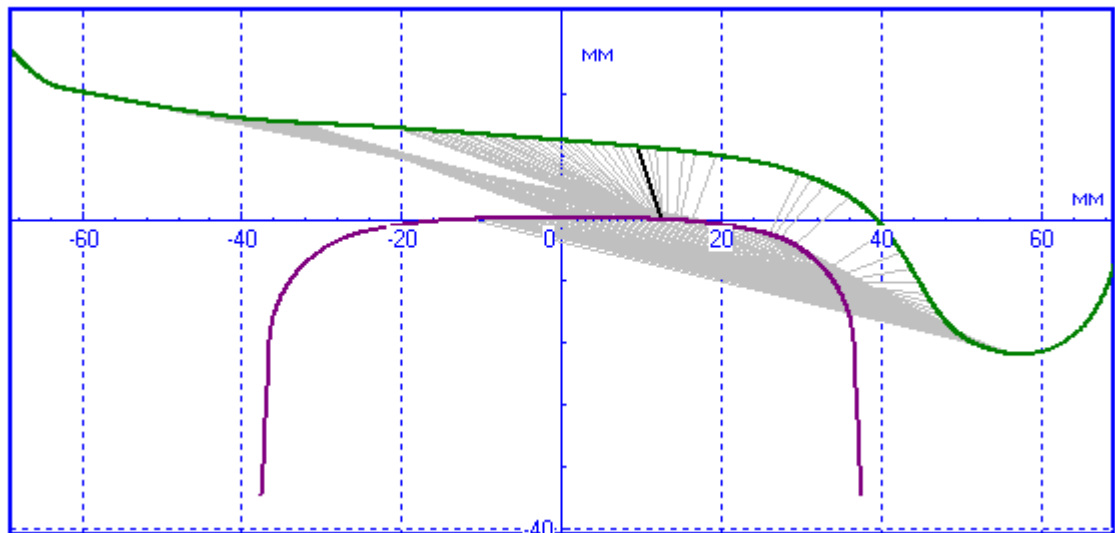



Figure 1.113. All contacts of a pair of profiles

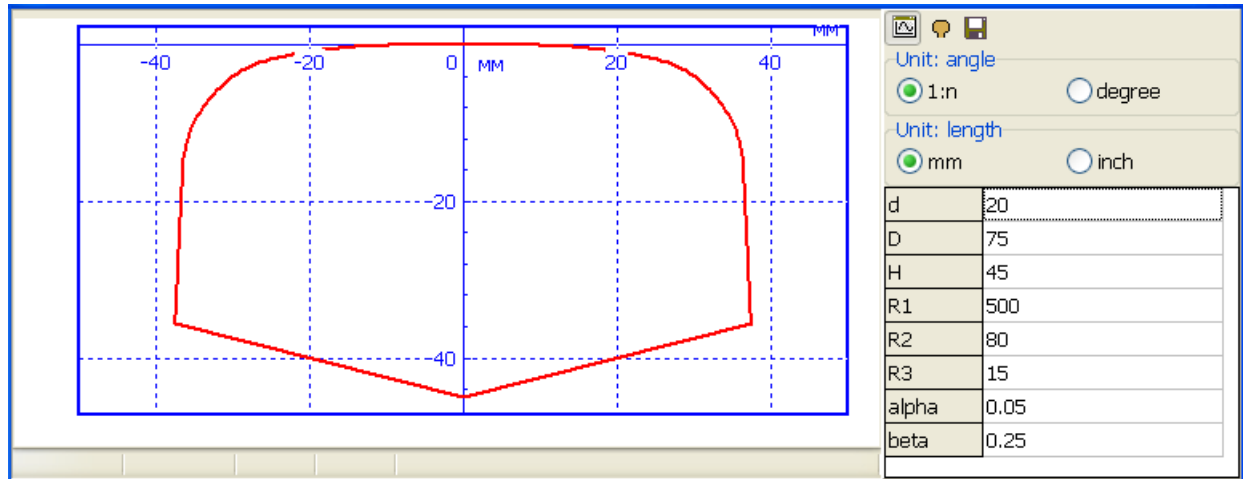
- **All contacts:** contacts on rail and wheel profiles for different values of lateral shift are connected by segments. The thick segment corresponds to the current value of the shift.
- **RRD** option draws the rolling radius difference curve, Sect. 1.3.2. *"Computation of contact between the wheel and the inertial rail"*, p. 1-91.
- **dY/Y** is Y coordinates of contact point on rail and wheel profiles in SC of the corresponding profile depending on the lateral wheel shift.
- **Beta** is the contact angle versus lateral wheel shift, Figure 1.93.
- **(BL-BR)/2\*L/2** plot of the function

$$E(y) = \frac{\beta_l - \beta_r S}{2} \frac{S}{2}$$

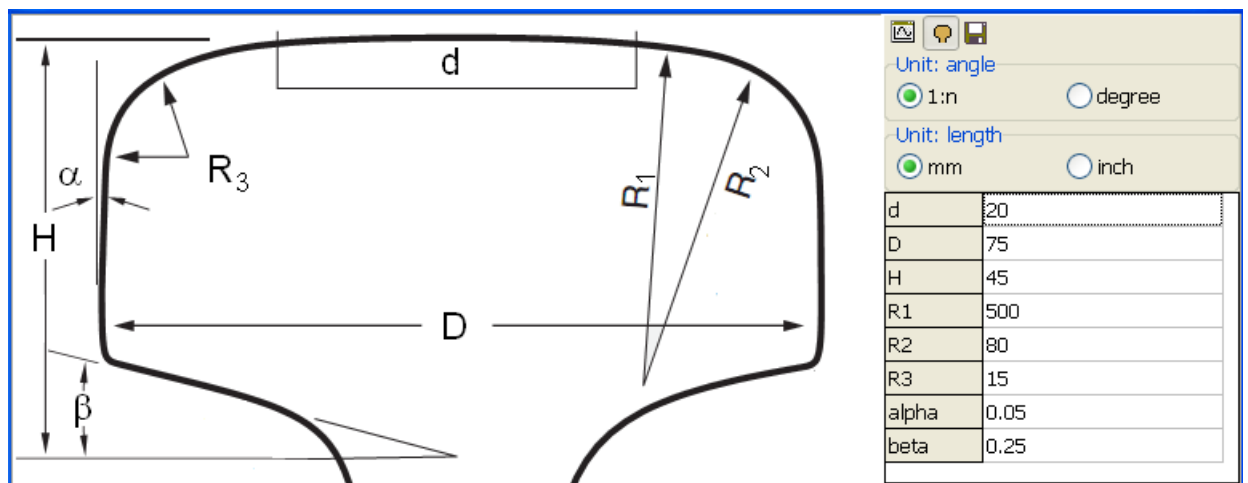
used in computation of contact angle parameter, Sect. 1.3.1.1.3. *"Simplified contact geometry. Equivalent conicity and contact angle parameter"*, p. 1-82.

### 1.4.1.1.4. Automatic generation of new rail profiles

New (not worn) rail profiles can be easily created with a special tool. To open the tool, use the **Tools | Rail-profile generator...** menu command or the  button on tool panel followed by the same command, Figure 1.114. The default numerical data in the table of the generator correspond to the new Russian rail R65.






a)



b)

Figure 1.114. Generator in modes of profile drawing (a) and profile scheme (b)

To create a profile, the following steps are necessary.

- Set values of geometric parameters, which sense is clear from Figure 1.114b. If the type 1:n is chosen for the angle unit, a conicity should be set instead of the angle, i.e., the tangent of the corresponding angle. For instance, the value 0.25 corresponds to the conicity 1:4 in Figure 1.116, and the value 0.05 corresponds to the conicity 1:20.
- Use the  button in Figure 1.114a to see the generated profile, and the  button for saving. The  button sets back the profile scheme in Figure 1.114b.

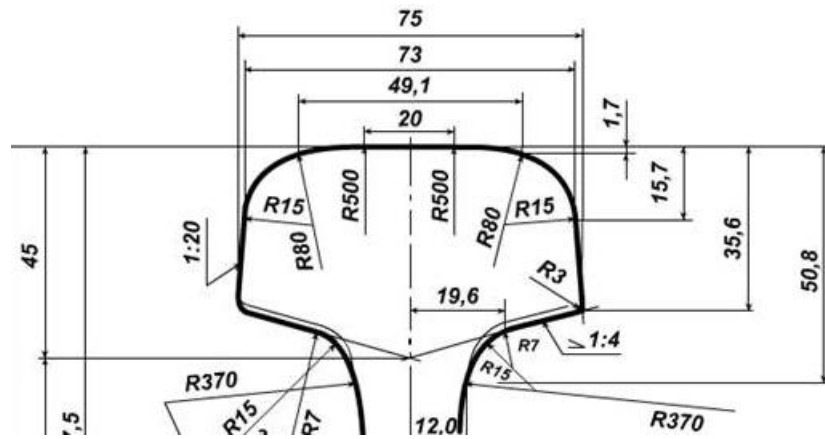


Figure 1.115. Rail R65



Figure 1.116. Rail R65 in the contact animation window

**Remark.** Generator creates both \*.rpf file with the rail profile data, and the \*.img file for contact animation window, Figure 1.116.

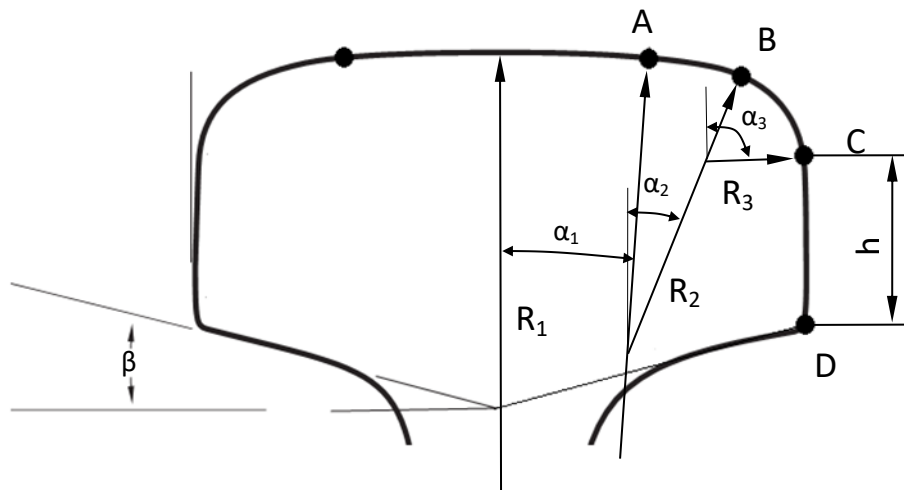


Figure 1.117. Scheme for computation of profile

### Algorithm for computation of profile

Consider mathematical relations, which are used for computation of the rail profile by geometrical data in Figure 1.113. Computational scheme of the profile is shown in Figure 1.116.

Angles  $\alpha_1$  and  $\alpha_3$  are computed from the formulas

$$\alpha_1 = \arcsin \frac{d}{2R_1}, \alpha_3 = \frac{\pi}{2} - \alpha.$$

Projections on abscissa and ordinate axes yield two main equations

$$(R_1 - R_2) \sin \alpha_1 + (R_2 - R_3) \sin \alpha_2 + R_3 \cos \alpha + h \operatorname{tg} \alpha = \frac{D}{2},$$

$$R_1 - [(R_1 - R_2) \cos \alpha_1 + (R_2 - R_3) \cos \alpha_2 + R_3 \sin \alpha] + h = H - \frac{D}{2} \operatorname{tg} \beta$$

relative to the unknown parameters  $\alpha_2$  and  $h$ . Shift of known terms into the right hand side of equations leads to

$$\begin{aligned} (R_2 - R_3) \sin \alpha_2 + h \operatorname{tg} \alpha &= \alpha_1, \\ -(R_2 - R_3) \cos \alpha_2 + h &= \alpha_2, \end{aligned}$$

Where

$$\begin{aligned} \alpha_1 &= \frac{D}{2} - (R_1 - R_2) \sin \alpha_1 - R_3 \cos \alpha, \\ \alpha_2 &= H - \frac{D}{2} \operatorname{tg} \beta - R_1 + (R_1 - R_2) \cos \alpha_1 + R_3 \sin \alpha. \end{aligned}$$

Eliminating the parameter  $h$  from the first equation gives the following expression for evaluation of  $\alpha_2$ :

$$\sin(\alpha_2 + \alpha) = \frac{\alpha_1 \cos \alpha - \alpha_2 \sin \alpha}{R_2 - R_3}.$$

Defined parameters allow estimating coordinates of points A, B, C, D, which are sufficient for the creation of profile.

### 1.4.1.1.5. Import wheel profiles from CAD

If image import from CAD of STEP, IGES format is available in the current UM configuration, the user can create a wheel profile in a CAD program and import it into UM Format.

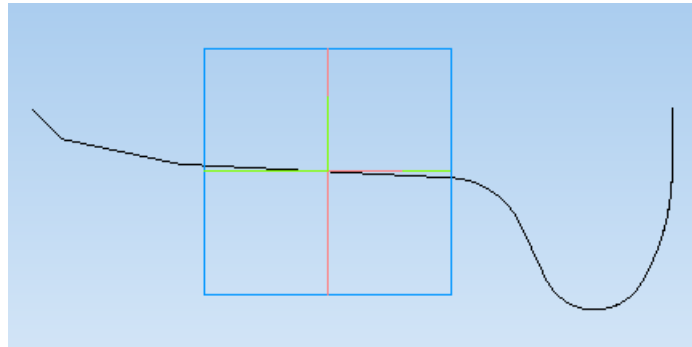


Figure 1.118. Wheel profile in CAD KOMPAS

The following steps are necessary.

1. A sketch of the profile is created in a CAD program; the length unit must be millimeter.
2. Based on the sketch, a thin part is created; extrusion width in both directions must be about 0.001 mm, Figure 1.118.
3. If **KOMPAS**, **SolidWorks** or **Inventor** is used, the profile is imported directly from the CAD program. Otherwise, the part must be saved in STEP or IGES format.
4. In **UM Input** program, the profile is imported by the **Tools | Import wheel profile from CAD** menu command, Figure 1.119.
5. Path and file name for the profile in UM format is selected in the standard window.

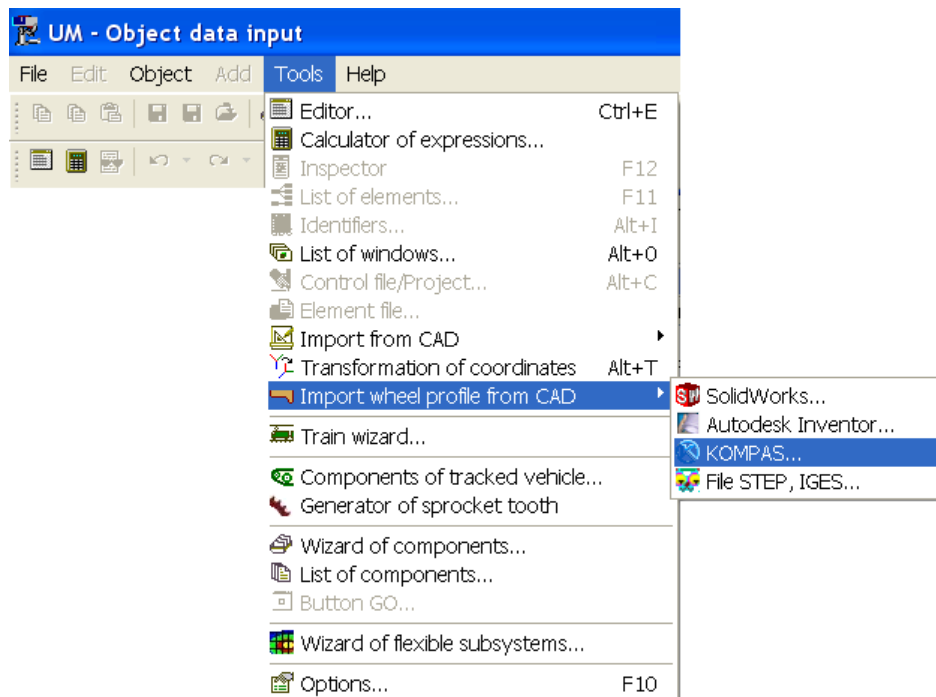


Figure 1.119. Wheel profile import in UM Input

### 1.4.1.1.6. Loading MiniProf wheel and rail profiles

UM supports direct loading the MiniProf \*.whl, \*.mpt, \*.ban files. MiniProf is a rather popular tool for wheel and rail profile measurement. You can find more information about MiniProf at <http://www.greenwood.dk/mini-prof.php>.

To load MiniProf files open a tool for development of wheel and rail profiles by clicking the **Tools | Create wheel/rail profile** menu command. Then click **Load from file** button and select one of the supported file formats in the dialog window, see Figure 1.120.

Along with direct reading the MiniProf files one can import such files manually using built-in UM tools. How to do that is shown in the Sect. 1.4.1.1.7. "Conversion of MiniProf wheel and rail profiles", p. 1-109.

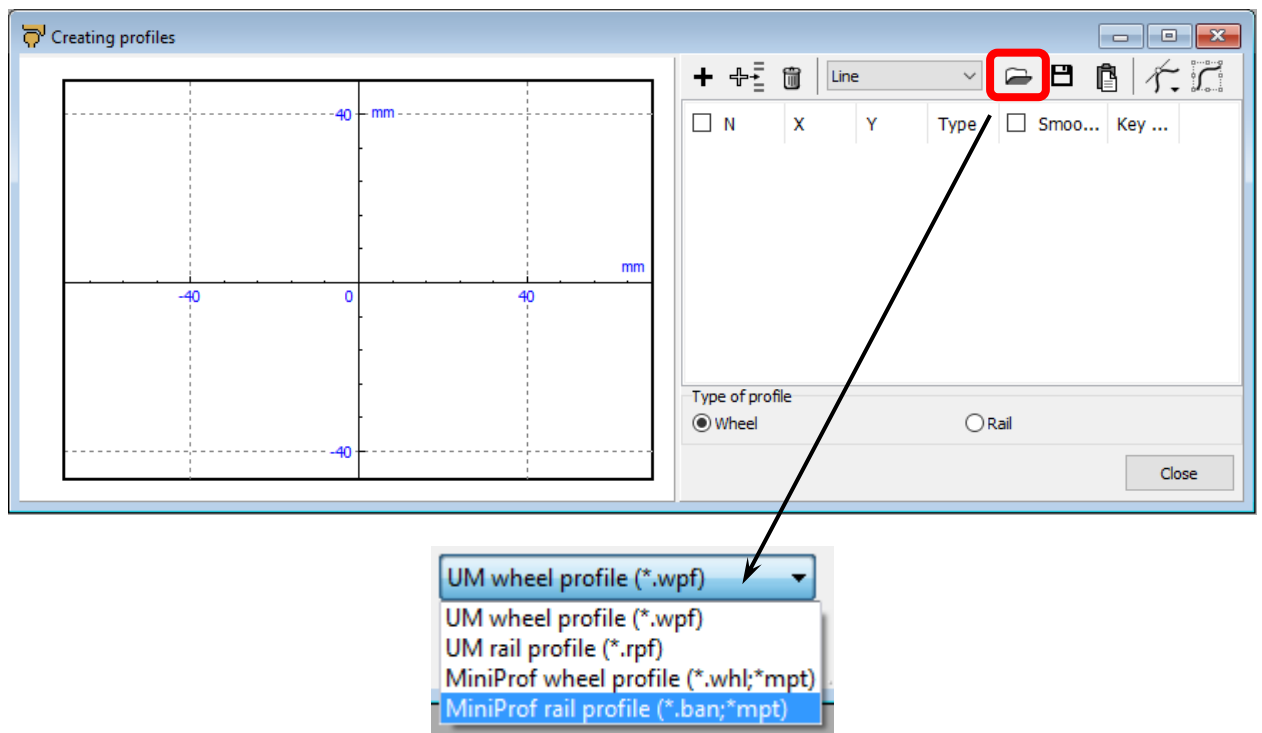


Figure 1.120. How to load MiniProf files into profile editor

**1.4.1.1.7. Conversion of MiniProf wheel and rail profiles**

Let us consider a wheel profile file \*.whl. Similar process is recommended for conversion of the rail profiles from MiniProf to UM format.

1. Open the \*.whl file in the text editor and delete information lines from the top of the file.  
The file must contain numeric data only  
0.5264 14.3366 1.4173 -0.107470274760285 1  
0.5279 14.4724 1.4173 -0.107470274760285 31  
0.5720 14.7482 1.3878 -0.101892819434936 92  
...
2. Open **MS Excel** and read the file. If necessary, shift columns to set X and Y data in the neighboring columns (the first column must be X, the second one Y).
3. Run **UM Simulation**. Open a tool for development of wheel and rail profiles by clicking the **Tools | Railway wheel and rail profile editor** menu command.
4. Copy the XY profile data from **MS Excel** to clipboard, make the profile creation tool in UM active and paste the data (Ctrl+V), see Figure 1.121.
5. Open **Transformation** window by clicking the marked tool button, see Figure 1.121.  
**Transformation** window appears, see Figure 1.122.

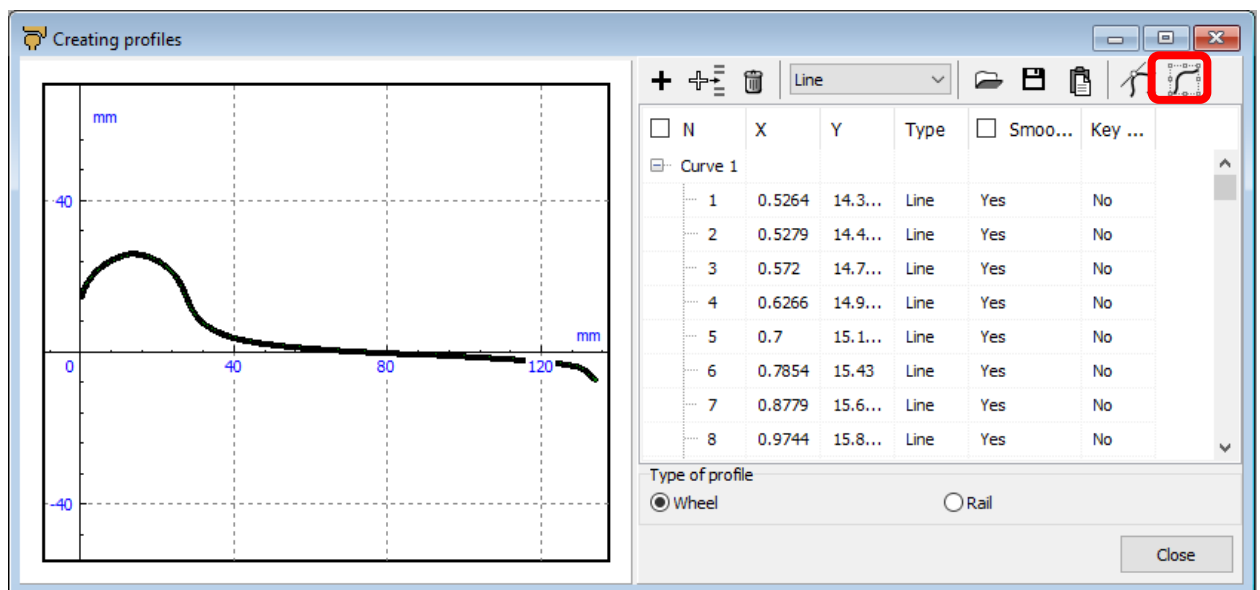


Figure 1.121. MiniProf file right after pasting from clipboard

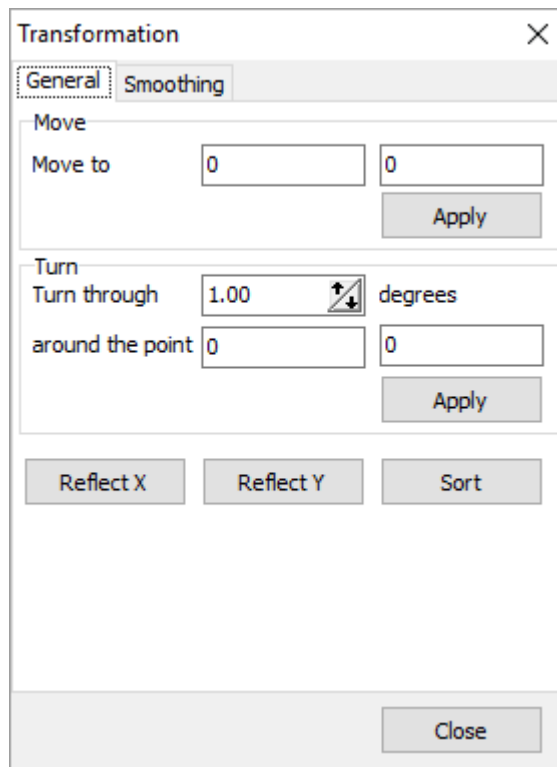


Figure 1.122. Profile transformation tool

6. Click the **Reflect X** button to reflex the profile relative horizontal axis, see results in Figure 1.123.

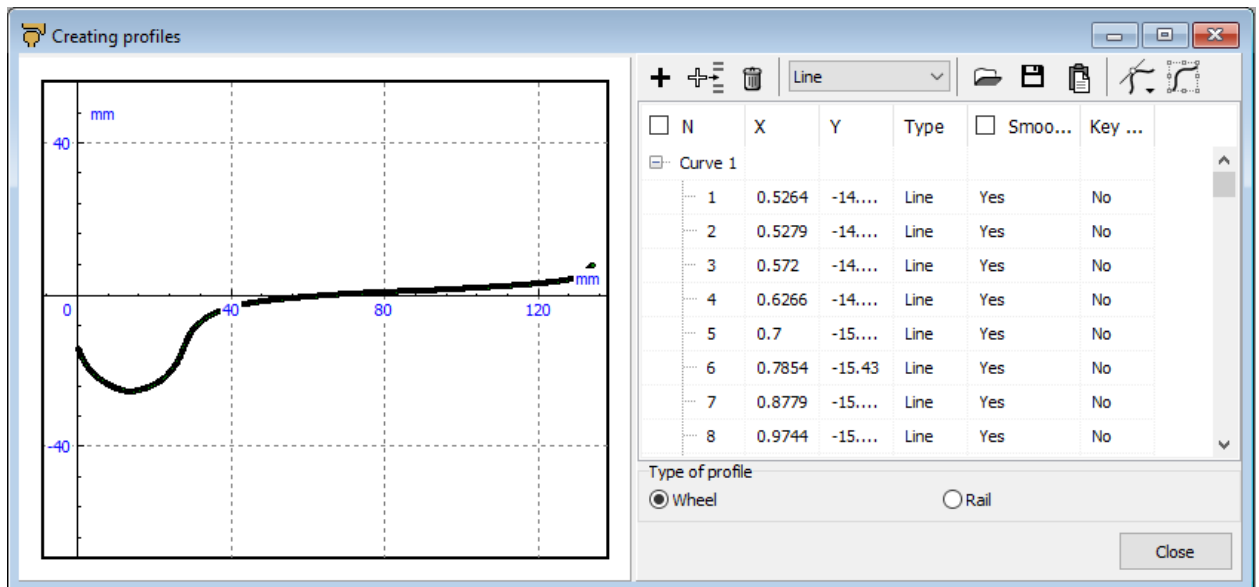


Figure 1.123. Profile after clicking **Reflect X** button

7. Click the **Reflect Y** button to reflect the profile relative to vertical axis, see results in Figure 1.124.

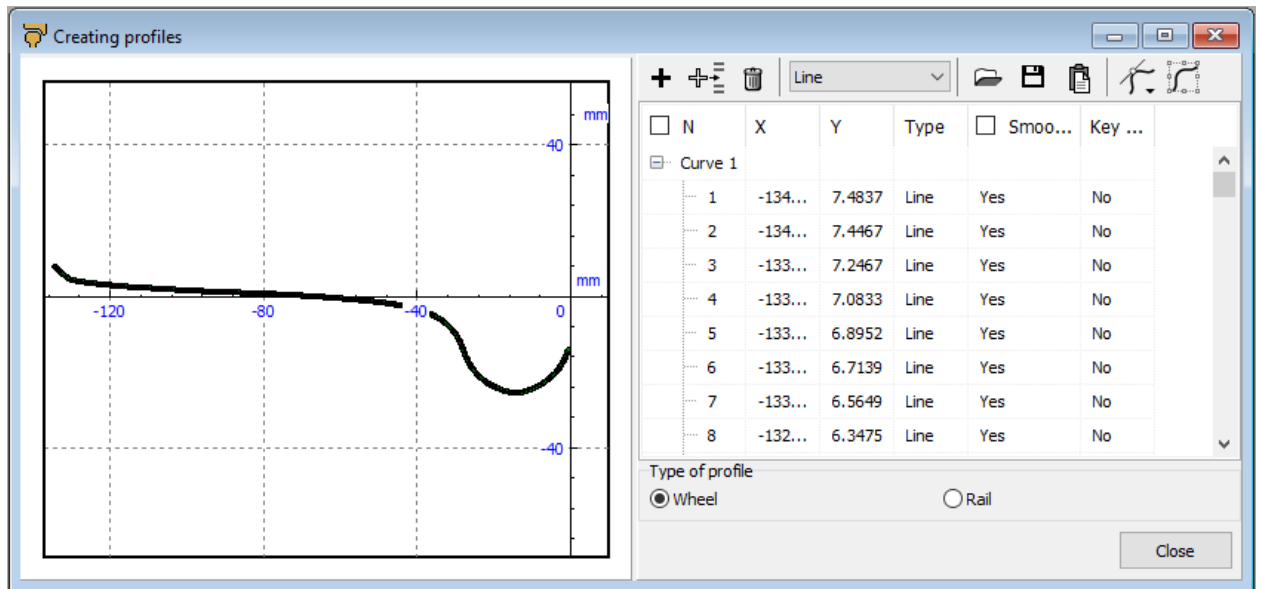


Figure 1.124. Profile after clicking **Reflect Y** button

- Set **Move to** vector, in the considered example it is (67.5, 0), and click the **Apply** button, see Figure 1.125. Operation result is shown in Figure 1.126.

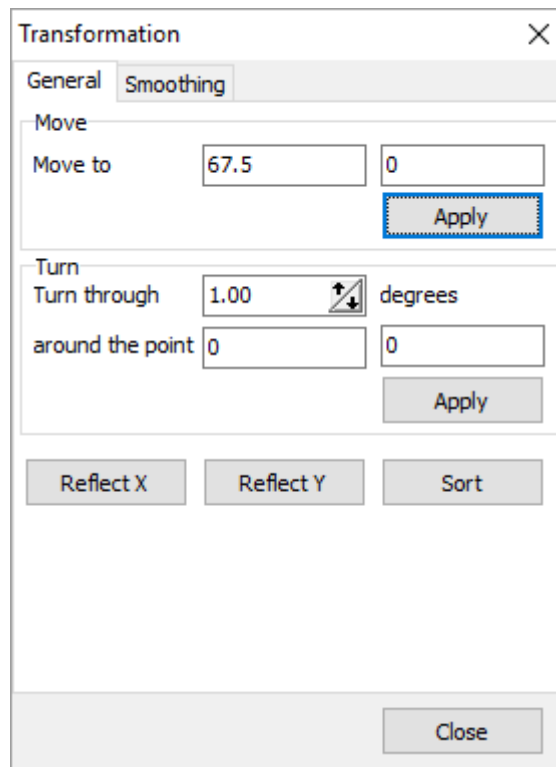


Figure 1.125. How to move the profile

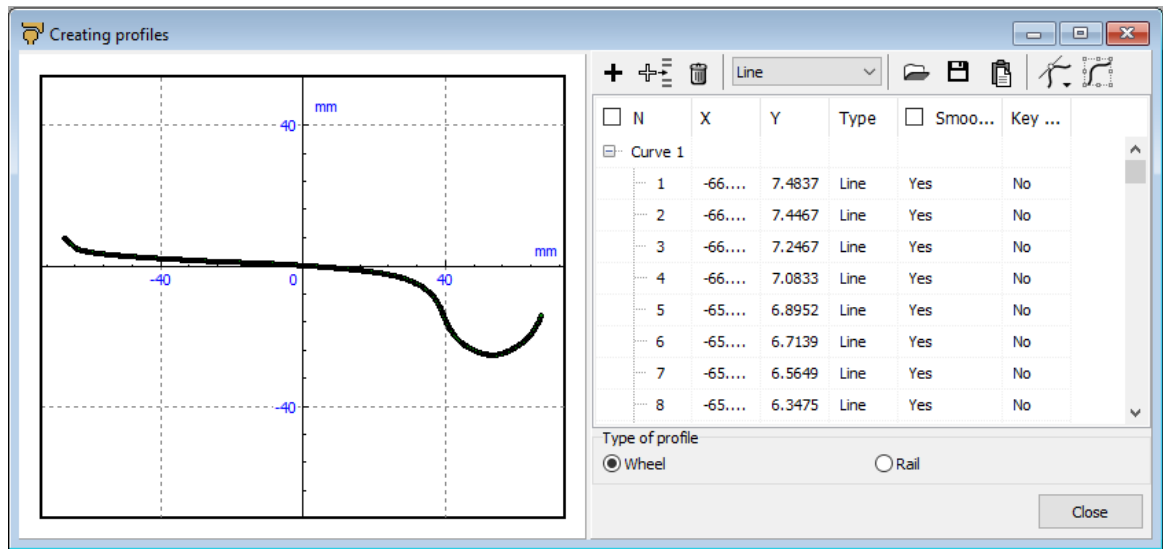


Figure 1.126. Final profile

- The profile conversion is over. Close the transformation window and save the wheel profile.

### 1.4.1.2. Creation of track irregularities

Three variants of track irregularities description are available in UM:

- track irregularity files;
- deterministic functions;
- programming in the *Control file* of the model.

Irregularity files and deterministic functions are the most often used approaches. At the same time, programming of the irregularities can be more effective in some cases; as soon as it enables parameterization and modification of irregularity data (such as irregularities height, length, etc.) during simulation. All the variants enable study of the dependencies of dynamic properties of a vehicle on the irregularity parameters by a scanning project, Sect. 1.6.1. "*List of internal identifiers parameterizing operation conditions of rail vehicles*", p. 1-244.

#### 1.4.1.2.1. Creation of irregularity files

Irregularities are stored in \*.way files located by default in the {UM data}\rw directory. A file contains a sequence of irregularity values in meters with the constant step size (0.1 m) along the track; *single* accuracy format (4-byte floating point numbers) is used.

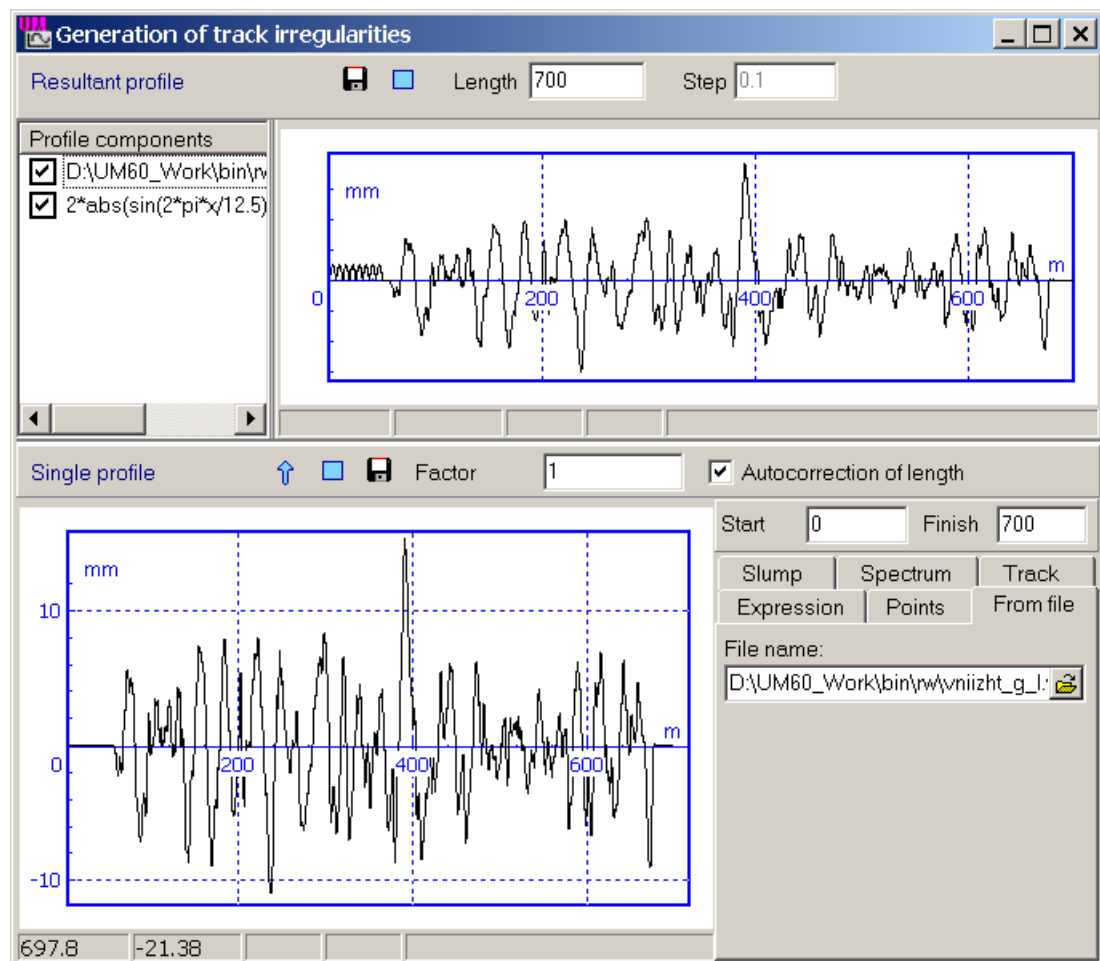


Figure 1.127. Tool for creation of track irregularities

A new file of irregularities can be generated by the special tool, which is available in the **UM Simulation** program by clicking the **Tools | Irregularity editor | Railway track** menu command.

**Note.** Within this tool the longitudinal coordinate is measured in meters but the irregularities should be given in millimeters.

Let us consider the structure of the tool and the meaning of its parts. The resultant track irregularity can be defined as a combination of single irregularities of various types; resultant irregularity profile is plotted in the top part of the tool window, see Figure 1.128. The list of single irregularities is located in the left top part of the window. Deleting or switching off an element of the list removes the corresponding component from the resultant profile.

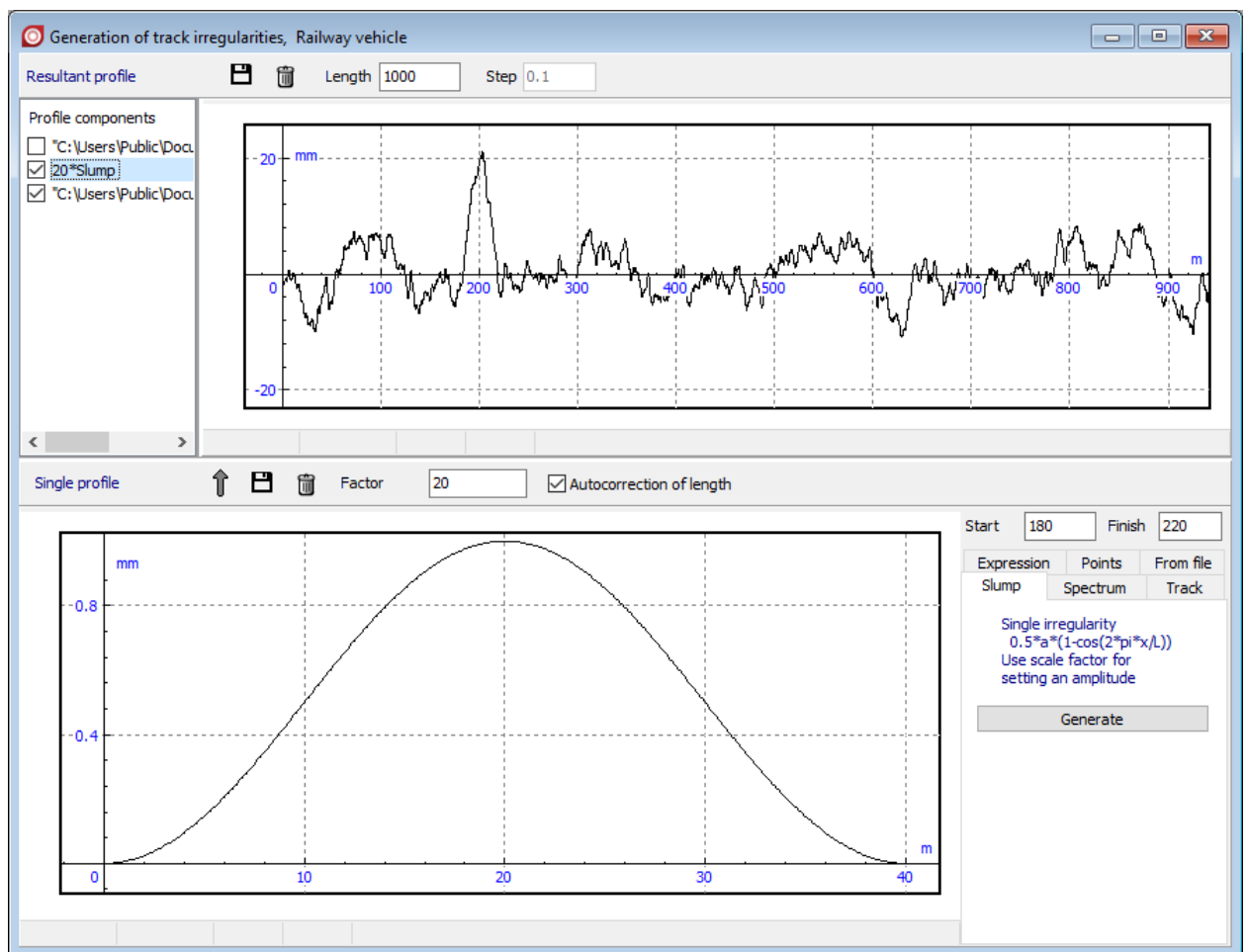





Figure 1.128. Railway track irregularities

Buttons and parameters in the **Resultant profile** section are described below.

- Parameter **Length** defines the length of the record in meters.
- Parameter **Step** defines the interval between next points of the record. By default step is 0.1 m.

Tab sheets in the right bottom part are used for creation separate irregularities as components of resultant ones. The corresponding plot is located in the left bottom part of the window.


Buttons and parameters in the **Single profile** section are described below.

- Button  adds the current single irregularity to the resultant track profile.
- Button  saves the current single irregularity to file.
- Buttons  clears the current single irregularity.
- Parameter **Factor** is the multiplier which is used for current single irregularity scaling during adding to the resultant profile. In an example, the user wants to convert irregularity profile defined in text format in UM data format. The source data points can be loaded by the tool placed on *Points* tab. By the way if the source file contains values of irregularities in meters, the 1000 scaling **Factor** value should be defined for correct conversion of the loaded data to millimeter units during adding to the resultant profile.
- If the **Autocorrection of length** check box is checked, the length of the resultant profile is automatically increased to match the adding separate irregularity. Otherwise the resultant profile length is defined by the **Length** control on the top panel.
- One can use **Start** parameter to define the start point of the resultant profile the first point of the adding single irregularity will be placed to. Note that the plot of the separate irregularity in the bottom graphic window always starts with zero abscissa value.
- The **Finish** parameter sets the end point of the adding irregularity record on the resultant profile.

In other words, the part of single record between **Start** and **Finish** points is added to the resultant profile.

Consider types of irregularities.


#### 1.4.1.2.1.1. Irregularities as analytic expression (the Expression 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 drop down list as well.


#### 1.4.1.2.1.2. Irregularities as slump

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

#### 1.4.1.2.1.3. From file

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

#### 1.4.1.2.1.4. Irregularities as points

Here an irregularity is created as a set of points 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 in millimeters, e.g.

```
0 0
0.05 11
0.10 21
0.15 17
.....
```

To input this data with the help of the clipboard do the follows:

- 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 interpolation can be applied to the data.

#### 1.4.1.2.1.5. Generation irregularities by spectrum

Track irregularities can be generated in accordance to PSD (spectral power density) function. The Rice-Pearson algorithm is used to generate the irregularity values by the equation:

$$x[n\Delta s] = \sum_{m=0}^M \sqrt{2S_c(m\Delta\omega) \Delta\omega} \cos[m\Delta\omega n\Delta s + \varphi(m\Delta\omega)],$$

where

$\Delta s$  is the irregularity step size, m;

$M$  is the total number of harmonics in the sum;

$S_c(\omega)$  is the PSD function,  $m^2/(\text{rad}/m)$ ;

$\Delta\omega$  is the frequency increment,  $\text{rad}/m$ ;

$\varphi(m\Delta\omega)$  is the phase uniformly distributed on interval  $[-\pi, \pi]$ .

The PSD can be function of frequency measured both in  $\text{rad}/m$  (circular frequency) and in  $\text{oscillation}/m$ . Use the **Circular frequency** check box to specify the PSD function abscissa units.

Selection of spectrum

FRA       Points  
 ERRI       Expression

Spectrum

(none) ...


Circular frequency

Number of harmonics

3000 ↕

Compute

1.4.1.2.1.5.1. Spectrum: Points

Consider generation of irregularities by the PSD function shown in Figure 1.129. Open the curve editor by the  button and specify four points on the PSD curve. Generated irregularities are given in Figure 1.130.

Note that this example illustrates the sequence of steps of the process, and *cannot be used* as a realization of irregularities by simulation of rail vehicles.

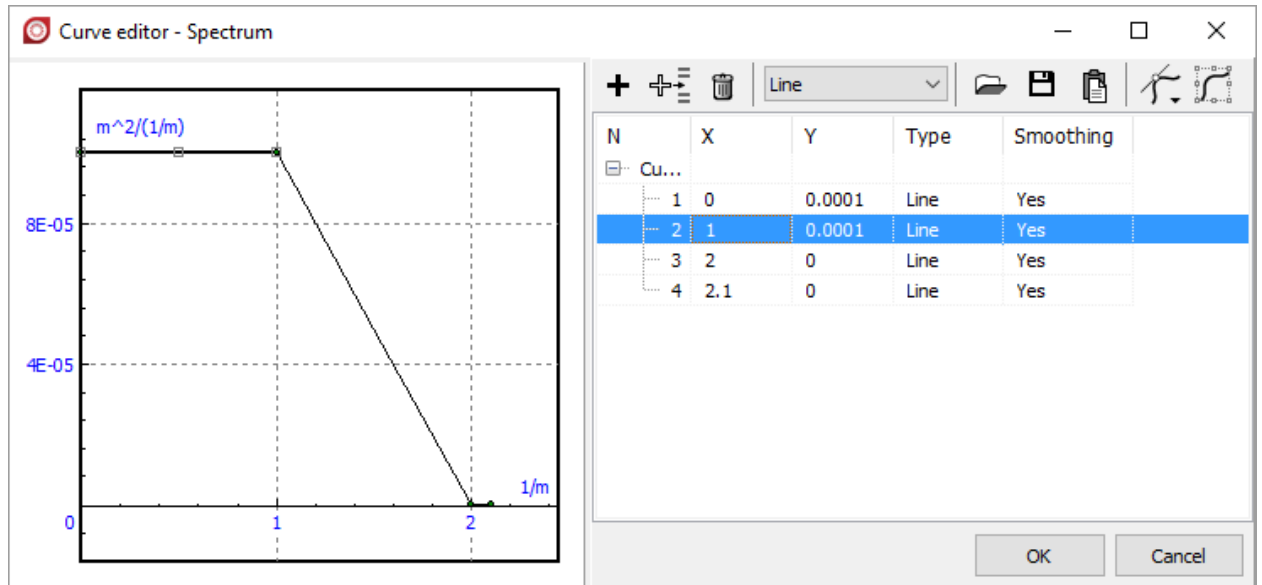


Figure 1.129. PSD function as a sequence of points

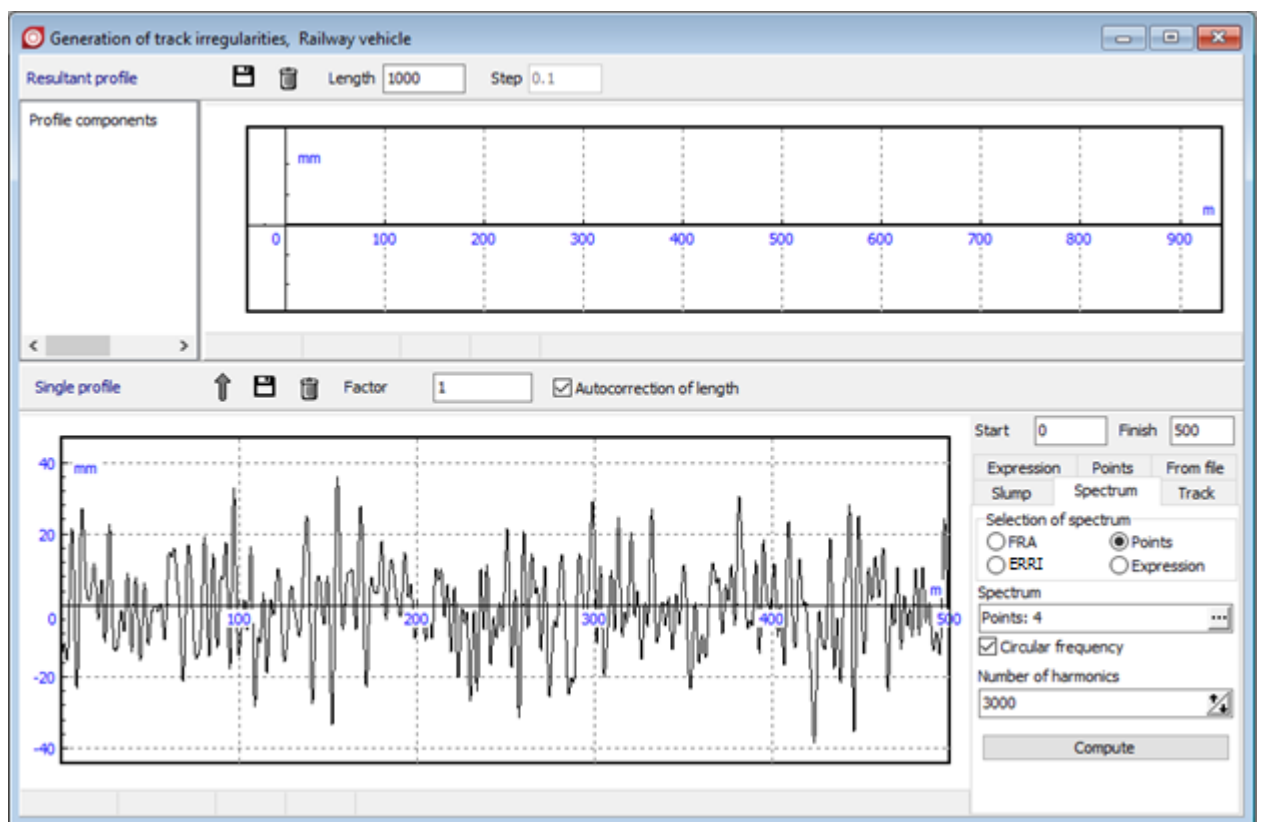


Figure 1.130. Irregularities generated by spectrum

**1.4.1.2.1.5.2. Spectrum (FRA)**

Federal Railroad Administration (FRA) of the United States Department of Transportation uses 9 (nine) classes for the description of railroad quality (first class is the worst one, ninth class is the best one). First five classes (1 ... 5) correspond to normal-speed railroads, and the last ones (6...9) are used for high-speed railroads, with speed over 90 mile/h (145 km/h) for passenger trains and over 80 mile/h (130 km/h) for freight ones.

PSD functions of the first six classes (1 ... 6) PSD functions are stated below [7]:

- PSD of half-sum of horizontal track irregularities:

$$\Phi(\Omega) = \frac{A_a \cdot \Omega_c^2}{\Omega^2 \cdot (\Omega^2 + \Omega_c^2)}, \quad \Omega > 0$$

$\Omega$  – circular frequency, rad/m

- PSD of half-sum of vertical track irregularities:

$$\Phi(\Omega) = \frac{A_v \cdot \Omega_c^2}{\Omega^2 \cdot (\Omega^2 + \Omega_c^2)}, \quad \Omega > 0$$

- PSD of half-difference of horizontal track irregularities:

$$\Phi(\Omega) = \frac{4 \cdot A_v \cdot \Omega_c^2}{(\Omega^2 + \Omega_c^2) \cdot (\Omega^2 + \Omega_s^2)}, \quad \Omega > 0$$

- PSD of half-difference of vertical track irregularities:

$$\Phi(\Omega) = \frac{4 \cdot A_v \cdot \Omega_c^2}{(\Omega^2 + \Omega_c^2) \cdot (\Omega^2 + \Omega_s^2)}, \quad \Omega > 0$$

Class	Parameter			
	$A_v$ , cm2 rad/m	$A_a$ , cm2 rad/m	$\Omega_s^2$ , rad/m	$\Omega_c^2$ , rad/m
1 (worst)	1,2107	3,3634	0,6046	0,8245
2	1,0181	1,2107	0,9308	0,8245
3	0,6816	0,4128	0,8520	0,8245
4	0,5376	0,3027	1,1312	0,8245
5	0,2095	0,0762	0,8209	0,8245
6 (best)	0,0339	0,0339	0,4380	0,8245

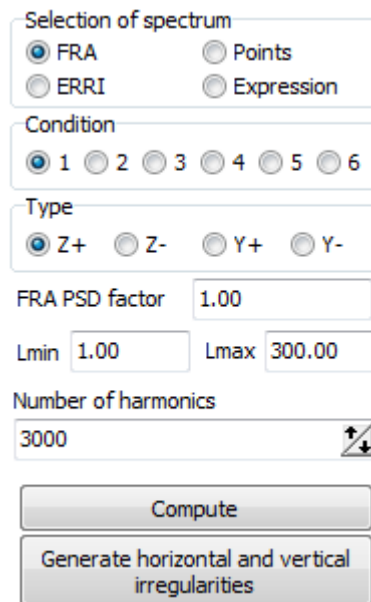




Figure 1.131. Standard FRA spectrum generation

**Create track irregularities** tool enables generation of track irregularities corresponded to first six FRA track quality classes. The **Lmin** and **Lmax** parameters specify the minimal and the maximal length of irregularity wave in the realization (m).

Vertical and horizontal irregularities of the left and right rails can be generated by FRA spectrum in *manual* and *automatic* modes. The following sequence can be used for manual generation of vertical irregularities of the left and right rails:

- a) select the **Z+** option in the **Type** group, which denotes a PSD for a half sum of the vertical irregularities of the left and right rails;
- b) generate realization by the **Compute** button and add it to the resultant irregularity by the  button;
- c) select the **Z-** option in the **Type** group, which denotes a PSD for a half difference of the vertical irregularities of the left and right rails;
- d) generate realization by the **Compute** button and add it to the resultant irregularity by the  button keeping **Factor** = 1, Figure 1.127; save the result to file for the left rail;

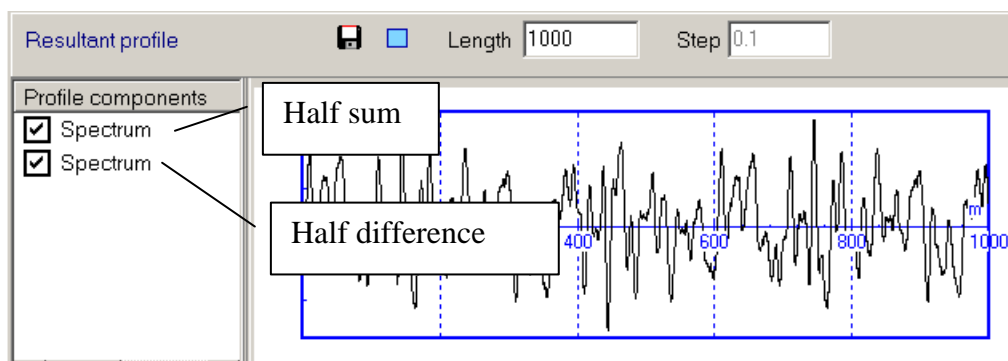


Figure 1.132. Resultant profile with a half difference realization

- e) remove or switch off the second part of the resultant function corresponding to the difference in the irregularities of the left and right rails;

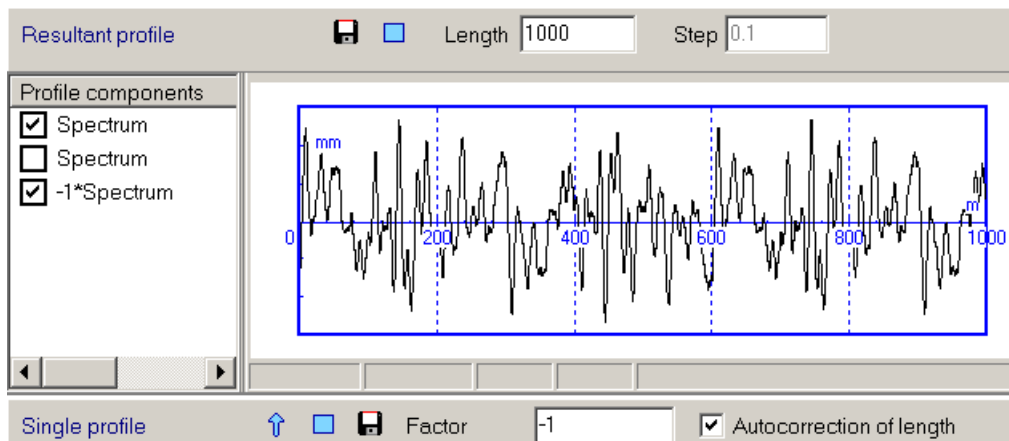



Figure 1.133. Resultant profile without a half difference realization

- f) subtract the half difference realization from the resultant irregularity by the  button setting **Factor** = **-1**; save the result to file for the right rail.

As an alternative, one can use **Generate horizontal and vertical track irregularities** button to generate horizontal and vertical irregularities of left and right rails by one click. Four track irregularity realizations are generated automatically and added to the resultant profile in the following sequence:

- Vertical irregularities of the left rail (...\_Z\_Left)
- Vertical irregularities of the right rail (...\_Z\_Right)
- Horizontal irregularities of the left rail (...\_Y\_Left)
- Horizontal irregularities of the right rail (...\_Y\_Right)

The sequence above corresponds to the sequence of track irregularities files in irregularity group files **\*.tig**. The following captions are used for the files by default: FRA\_< Class\_I >\_Z\_Left, where I is a number from the interval [1...6].

One can save the set of irregularity realizations (four items of the resultant profile list) as a group with the **Save irregularities group \*.tig** popup menu command. New **\*.tig** file and corresponded directory with irregularities files **\*.way** will be created on a hard drive.

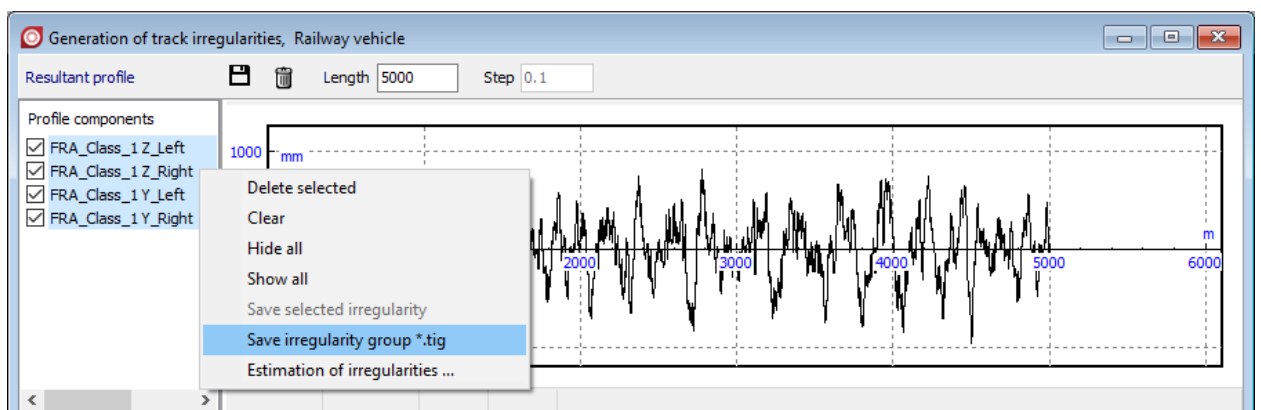


Figure 1.134. Saving irregularities as group

**1.4.1.2.1.5.3. Spectrum (ERRI)**

Create track irregularities tool enables generation of track irregularities corresponded to tracks of good and bad quality according to ERRI B176 which are widely used by European researchers. PSD functions corresponded to tracks of different quality are stated below [8]:

- PSD of horizontal track irregularities:

$$\Phi(\Omega) = \frac{a_h \cdot \Omega_c^2}{(\Omega^2 + \Omega_R^2) \cdot (\Omega^2 + \Omega_c^2)}, \quad \Omega > 0$$

- PSD of half-sum of vertical track irregularities:

$$\Phi(\Omega) = \frac{a_v \cdot \Omega_c^2}{(\Omega^2 + \Omega_R^2) \cdot (\Omega^2 + \Omega_c^2)}, \quad \Omega > 0$$

- PSD of half-difference of vertical track irregularities:

$$\Phi(\Omega) = \frac{1}{b_A^2} \cdot \frac{\Omega^2}{(\Omega^2 + \Omega_s^2)} \cdot \frac{a_v \cdot \Omega_c^2}{(\Omega^2 + \Omega_R^2) \cdot (\Omega^2 + \Omega_c^2)}, \quad \Omega > 0$$

Track quality	Parameter		
	$a_v$ , cm2 rad/m	$a_h$ , cm2 rad/m	
Bad	1,08e-6	0,6125e-6	
Good	0,4032e-6	0,2119e-6	
Parameter			
$\Omega_c$ , rad/m	$\Omega_R$ , rad/m	$\Omega_s$ , rad/m	$b_A$
0,8246	0,0206	0,4380	0,75

Note that the description of vertical irregularities is similar to that used in FRA standards, but horizontal irregularities for the left and right rails are equal.

Selection of spectrum

FRA       Points  
 ERRI       Expression

Level

High       Low

Type


Z+       Z-       Y

Lmin  Lmax

Number of harmonics

Figure 1.135. Generation of irregularities by ERRI spectrum

#### 1.4.1.2.1.5.4. Spectrum (Expression)

Set an analytic expression  $f(w)$  in the **Function** edit box and press the **Enter** button or **Compute** button. Use the **Circular frequency** check box to specify the frequency type used in the expression. **Wmax** and **Wmin** labels show bounds of the frequency interval evaluated from **Lmin** and **Lmax** values. Click  button to plot or edit the spectrum function. Standard functions can be used in the expression ([Chapter 3](#), Sect. *Standard functions and constants*). Standard expressions can be assigned from the drop down list (**Example spectra**) as well.

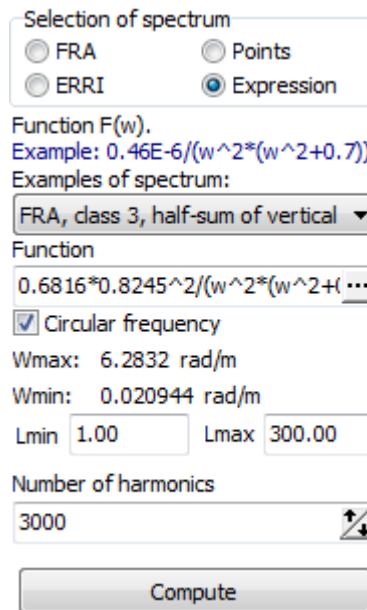


Figure 1.136. Setting PSD function as an expression

Let's consider the sequence of irregularities generation according to the following data:

- $F(w) = 1E-4 * \sin(3.14 * w)$
- **Circular frequency** is off;
- **Start** = 0 m; **Finish** = 2000 m.

#### 1.4.1.2.2. Programming irregularities in the Control file

Control file is one of the main tools for the user programming in UM environment ([Chapter 5](#)). The control file for railway vehicles contains the following function:

```
function TrackProfile(position: real_; var LeftZ, RightZ, LeftY, RightY, der-
LeftZ, derRightZ, derLeftY, derRightY: real_): integer;
begin
  Result := 0;
end;
```

The user may compute any vertical or horizontal irregularity within this function.

**Input:**

*position* is longitudinal coordinate for which the irregularities are computed.

**Output:**

*result* returns function value. If the return value is equal to 1 the programmed values will be taken into account, otherwise they are ignored.

$LeftZ$ ,  $RightZ$  are vertical irregularities of the left and right rails in meters;  
 $LeftY$ ,  $RightY$  are horizontal irregularities of the left and right rails in meters;  
 $derLeftZ$ ,  $derRightZ$  are derivatives of vertical irregularities of the left and right rails w.r.t. longitudinal coordinate;  
 $derLeftY$ ,  $derRightY$  are derivatives of horizontal irregularities of the left and right rails w.r.t. longitudinal coordinate.

Irregularities computed in the function are added with those assigned to rails from files. If it is necessary to take into account the programmed irregularities only, the *NoIrregularities.way* file should be assigned to the corresponding rails.

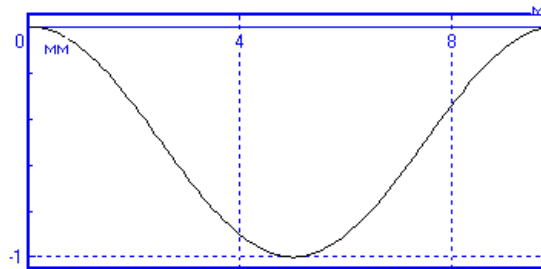


Figure 1.137. Slump

Let us consider a separate irregularity of the *Slump* type, Figure 1.137. The irregularity of the *left rail* and its derivative are defined by the formulae

$$h = -\frac{H}{2} \left( 1 - \cos \frac{\pi(x - x_0)}{L} \right);$$

$$h' = -\frac{\pi H}{2L} \sin \frac{\pi(x - x_0)}{L},$$

$$x \in [x_0, x_0 + L].$$

Here  $H, L, x_0$  are the depth and length of the slump as well as the start position along the track. Let the same irregularity for the *right rail* be moved along the track in the longitudinal direction on  $dx$  relatively to the left rail. It is supposed that all of these four parameters are included in the list of identifiers of the corresponding UM model ([Chapter 3](#), Sect. *Basic elements of constructor*).

```
function TrackProfile( position : real_; var LeftZ, RightZ, LeftY, RightY,
derLeftZ, derRightZ, derLeftY, derRightY : real_ ) : integer;
begin
  Result := 1;
  LeftY:=0;
  RightY:=0;
  derLeftY:=0;
  derRightY:=0;

  // Irregularities for the left rail
  if (position>_pzAll[1].x0) and (position<_pzAll[1].x0 +
_pzAll[1].L) then begin
    LeftZ:=-0.5*_pzAll[1].H*(1-cos(pi*(position-_pzAll[1].x0)/
_pzAll[1].L));
    derLeftZ:=-0.5*pi*_pzAll[1].H*sin(pi*(position-_pzAll[1].x0)/
_pzAll[1].L))/_pzAll[1].L;
```

```
end else begin
  LeftZ:=0;
  derLeftZ:=0;
end;

//Irregularities for the right rail
if (position>_pzAll[1].x0+dx) and (position<_pzAll[1].x0 +
_pzAll[1].L+_pzAll[1].dx) then begin
  RightZ:=-0.5*_pzAll[1].H*(1-cos(pi*(position-_pzAll[1].x0-
_pzAll[1].dx)/_pzAll[1].L));
  derRightZ:=-0.5*pi*_pzAll[1].H*sin(pi*(position-_pzAll[1].x0 -
_pzAll[1].dx)/_pzAll[1].L))/_pzAll[1].L;
end else begin
  RightZ:=0;
  derRightZ:=0;
end;
end;
```

### 1.4.1.3. Creation of macrogeometry files

The macrogeometry files \*.mcg are used in the following cases:

- Track macrogeometry differs from tangent section, curve, S-curve or switch;
- Vertical track macrogeometry is taken into account.

A macrogeometry file can include any number of curves, tangents, switches, variable friction conditions along the track as well as an arbitrary vertical profile.

Use the **Tool | Macrogeometry editor | Railway or monorail track** menu command to start the window where the track is described.

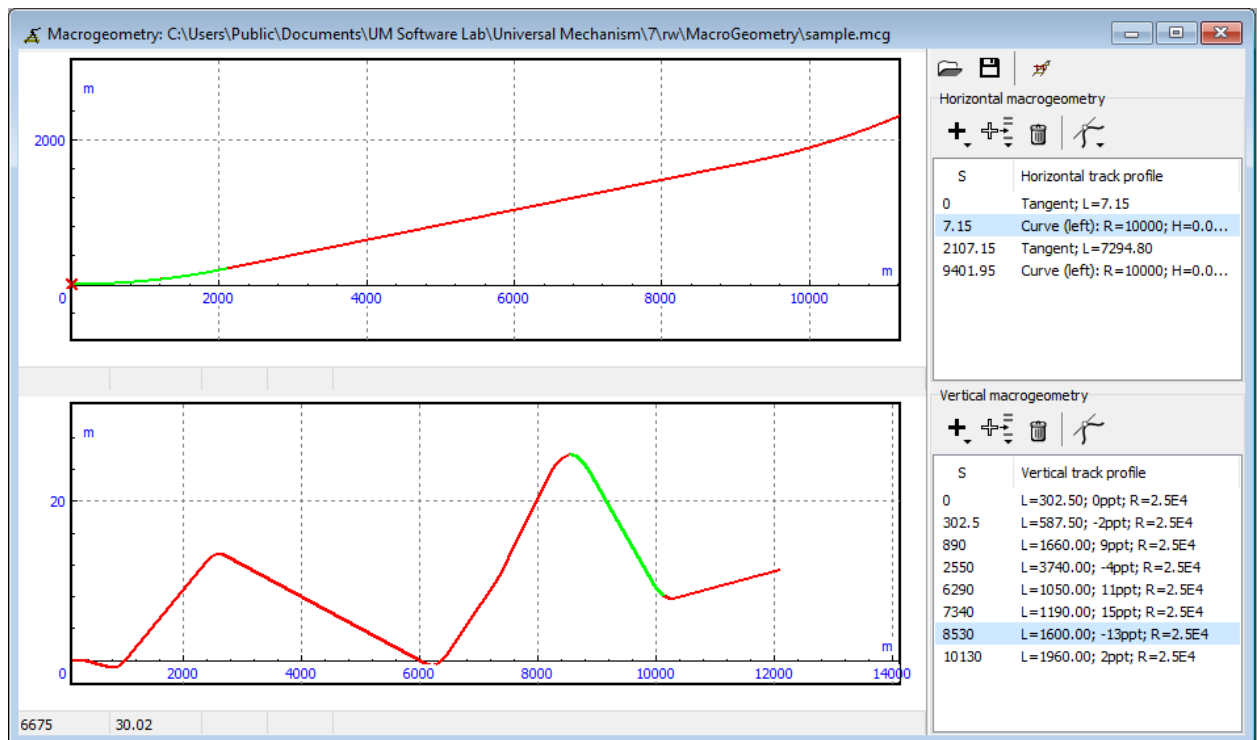


Figure 1.138. Macrogeometry window

#### 1.4.1.3.1. Track macrogeometry in horizontal plane

The upper part of the window in Figure 1.138 is used for description of the track geometry in the horizontal plane.

- To *add* a section, click on the **+** button and select the section type in the menu.

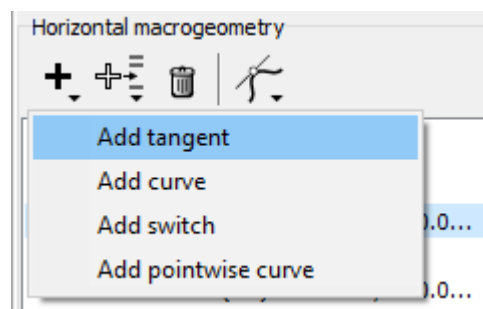
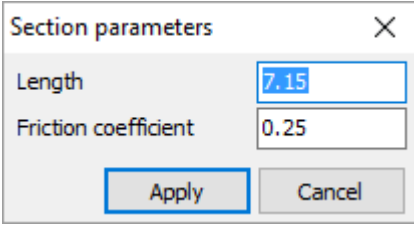


Figure 1.139. Section menu

- To *edit* the section parameters double click on the corresponding line of the section list or select the line and press **Enter**.

#### 1.4.1.3.1.1. Tangent section



Section parameters	
Length	7.15
Friction coefficient	0.25
Apply	
Cancel	

Figure 1.140. Parameters of a tangent section

Tangent section window contains values of section length and coefficient of friction.

#### 1.4.1.3.1.2. Curve section

Curve parameter window includes (Sect. Geometry of curve)

- type of curve (left or right);
- geometric parameters of the curve: lengths of transient sections (P1, P2), length of steady curve section (S), radius (R), cant of outer rail (H) as well as additional gauge widening in curve (dY);
- coefficients of friction on running surfaces of outer and inner rails, on the inward side of the outer rail (flange) as well as angles  $\beta_r, \beta_s$ , which specify the transient from the coefficients of friction on the running surface  $f_r$  and of the rail inward side  $f_s$  if  $f_r \neq f_s$ . Sect. 1.3.3. "Coefficient of friction in wheel-rail contact", p. 1-95.

Curve parameters

Type of curve  
 Left  Right

P1 50

S 2000

P2 50

R 10000

H 0.09

dY 0.01

L 2100

Friction coefficients

Outer rail 0.25

Inner rail 0.25

Flange 0.25

Transient section  
for the side friction

from 30 to 40 degrees

Apply Cancel

Figure 1.141. Window with curve parameters

**Remark.** Transient sections of curves are set by clothoid, i.e. by a curve with a uniform increase of curvature.

#### 1.4.1.3.1.3. Switch section

Switch parameter window includes values of (Figure 1.135, Sect. 1.2.1.3. "Switch geometry", p. 1-71).

- $q$  is stock rail overhang;
- gauge;
- $\beta_n$  is initial angle;
- $\alpha$  is switch angle;
- $R_0$  is radius of point;
- $R$  is radius of switch;
- $b_r$  is switch deviation for  $R_0$ ;
- $m$  is frog tail length;
- $d$  is track spacing.
- $R_1$  is radius behind the frog.

The parameters define fully the switch geometry and some additional parameters:

- $L_p$  is full switch length;
- $L_t$  is theoretical length;
- $k$  is tangent section before the frog.

Parameter	Value
Rail overhang (mm)	2769.0
Gauge (mm)	1520
Initial angle (mrad)	7.9488
Switch angle (mrad)	90.6602
Radius of point R0 (m)	300.000
Radius of switch R (m)	300.000
Switch deviat. for R0 (mm)	75.0

Full switch length (m) 34.862  
Theoretical length (m) 28.048  
Computed tangent section (mm) 3285.0  
Min dist. between tracks (m) 3.118

Movement  
 Facing  Trailing  
 Left switch  
Coefficient of friction 0.250

OK Cancel

Figure 1.142. Switch parameters

**Additional parameters**

- direction of motion (facing, trailing)
- type of switch (left or right);
- coefficient of friction.

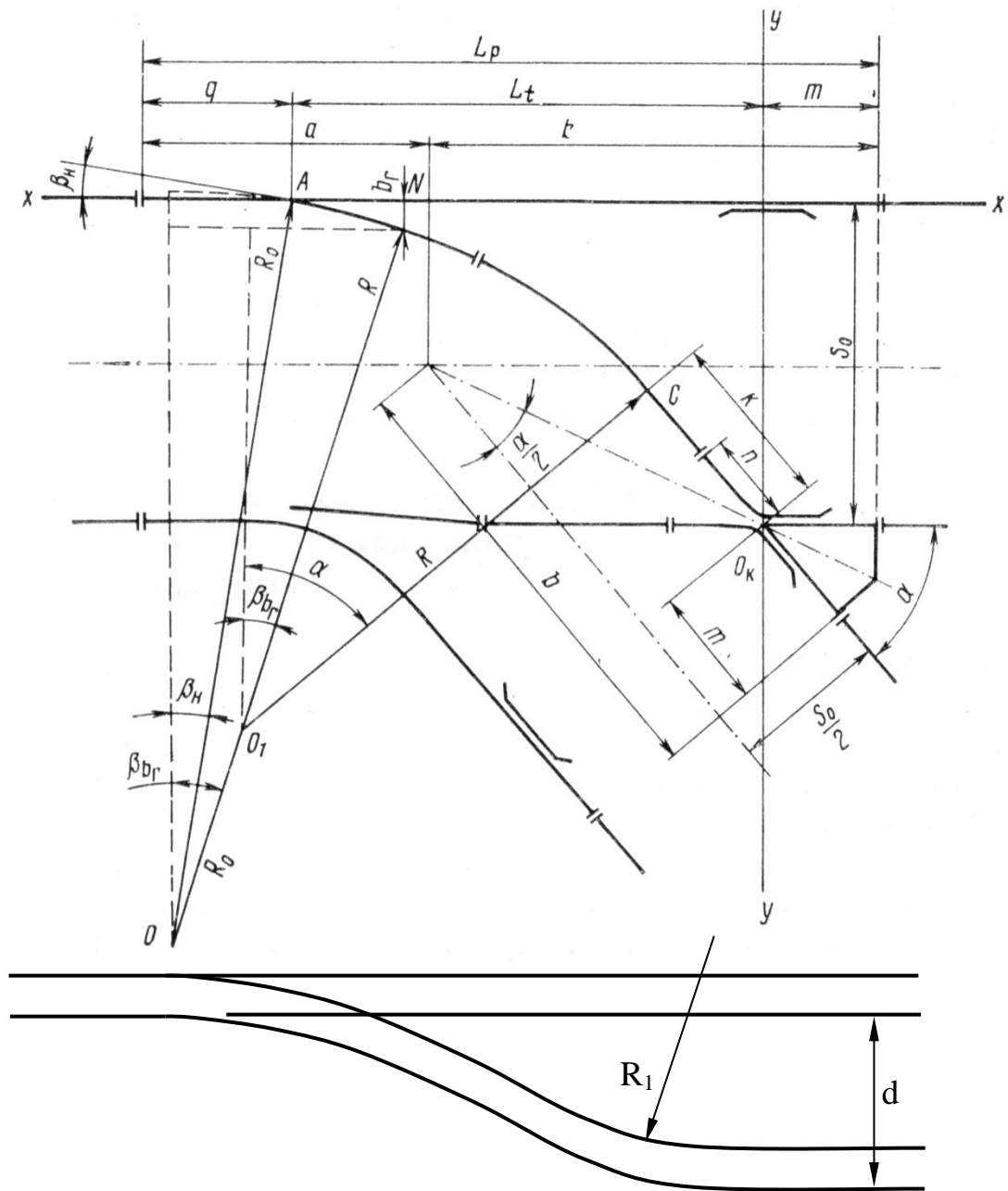



Figure 1.143. Switch geometry parameters

**1.4.1.3.1.4. Point curve**

Point curve can be used for the description of track geometry (shape of curves, cant, widening, etc.) with arbitrary functions. Curve editor is used for description of functions. Click  button for editing of the curve parameter, Figure 1.144.

- **Curve XY**

Curve editor is used for setting of track axis position in XY plane. Section length is evaluated automatically as a length of XY curve.

**First point must have zero coordinates. Tangent at the first point must be equal to the positive direction of X axis.**

Number of points and curve length are not limited. For example, it can be measured track data or analytically computed coordinates of nonstandard transient curve section.

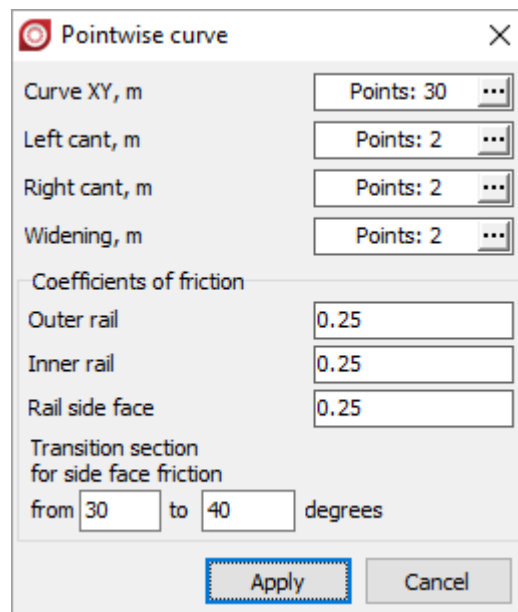


Figure 1.144. Point curve parameter window

The point list can be the following:

X	Y	X	Y
0	0	26	0.4394
2	0.0002	28	0.5488
4	0.0016	30	0.675
6	0.0054	32	0.8192
8	0.0128	34	0.9826
10	0.025	36	1.1664
12	0.0432	38	1.3718
14	0.0686	40	1.6
16	0.1024	42	1.8522
18	0.1458	44	2.1296
20	0.2	46	2.4334
22	0.2662	48	2.7648

24	0.3456	50	3.125
26	0.4394		

The list is prepared in MS Excel as a two-column table. Then it was copied to clipboard and insert to curve editor (all other points must be preliminary deleted from the curve editor list!). The curve shape is shown in Figure 1.145. B-Spline is recommended for curve shape approximation. Automatically evaluated curve length is shown in Figure 1.146.

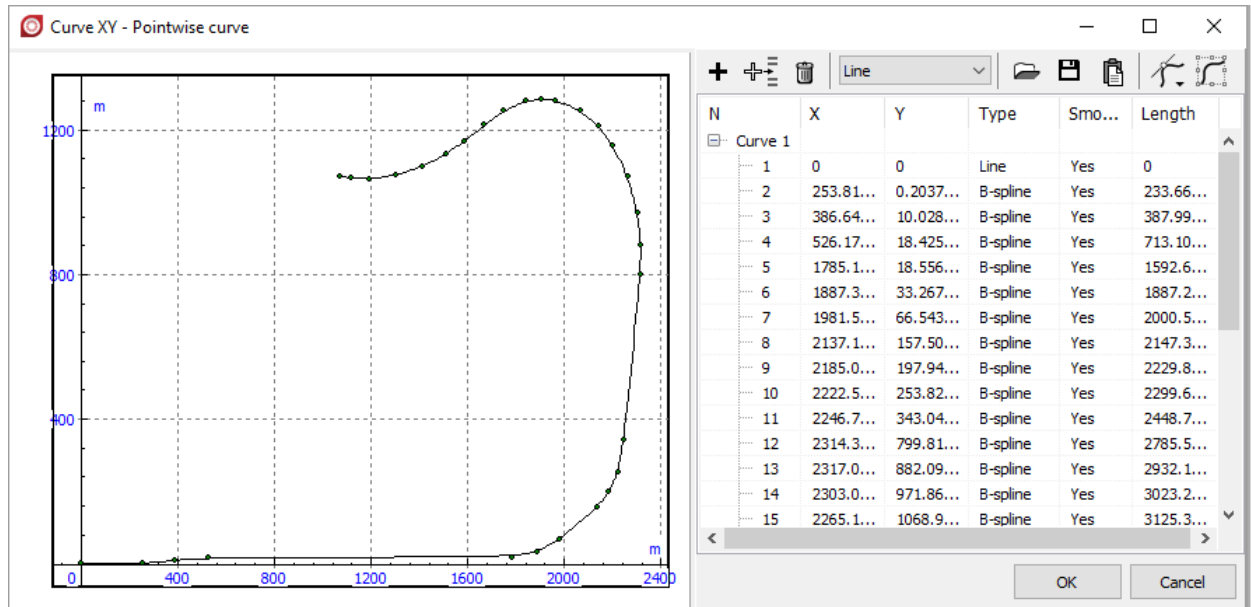


Figure 1.145. XY curve editor

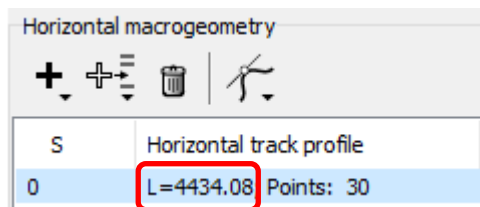


Figure 1.146. Point curve length

**Remark.** Approximation with splines results in zero values of second derivatives at the first and the last points. It means that sometimes second derivative and curvature functions can have breaks at the points of connection of point curve with plane sections of other types. For example, if transient section described with point curve is sibling to a circular curve which is with standard curve with zero transient length there will be a break in curvature function, see Figure 1.147. In this situation first derivative is continuous. If points of point curve are set with a small step (1-2 meters), break in curvature doesn't affect the simulation results. To exclude the problem of curvature breaking the user can describe the whole curve (all sections) with point curve element.

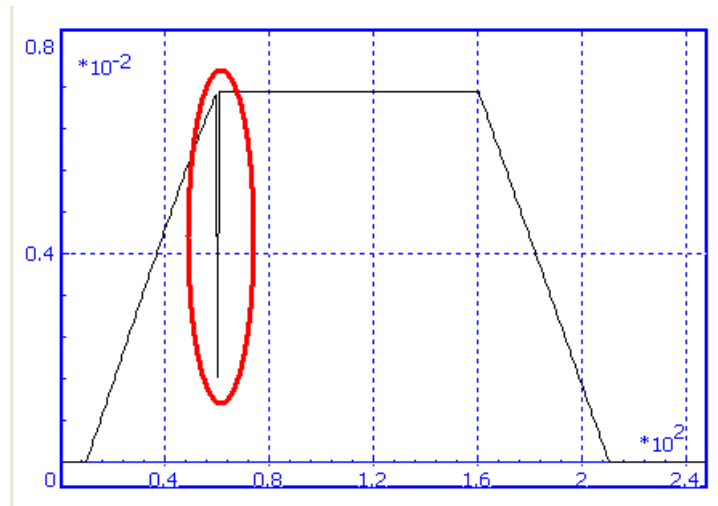


Figure 1.147. Break in curvature

- **Cant functions of left and right rails**

Cant functions for both rails are set with point curves from XY curve length. Left and right cants can be not zero simultaneously and get positive and negative values. Curve length values for each of XY points are shown at the last column of the curve editor table, see Figure 1.145.

- **Widening**

Widening is set as a function from XY curve length.

- **Friction coefficients** are set the same as in Sect Curve section.

**Remark.** The user should control the continuity of cant and widening functions for the element, as well as for the areas of connection with sibling elements of horizontal track profile.

### 1.4.1.3.2. Track macrogeometry in vertical plane

The lower part of the window in Figure 1.138 is used for description of the track geometry in the vertical plane.

- To *add* a section, click on the  button and select the section type in the menu.

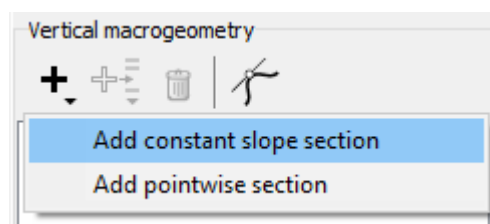


Figure 1.148. Section menu

- To *edit* the section parameters double click on the corresponding line of the section list or select the line and press Enter.

### 1.4.1.3.2.1. Constant slope section

The following parameters can be set in Gradient window (Figure 1.149):

- length of section (m);
- gradient in ppt (parts per thousand or meters per kilometer);
- radius of circle on smoothing the gradient change.

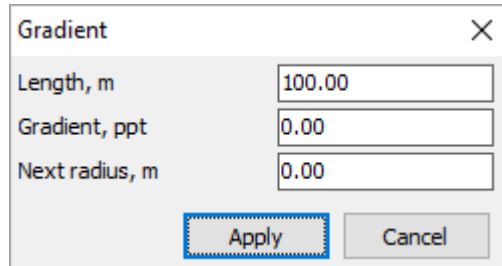


Figure 1.149. Constant slope section parameters

### 1.4.1.3.2.2. Point section

Point curve can be used for the description of vertical track axis coordinate with arbitrary functions from XY curve length. Curve editor is used for description of functions, see Figure 1.150. Curve editor opens when section editing starts.

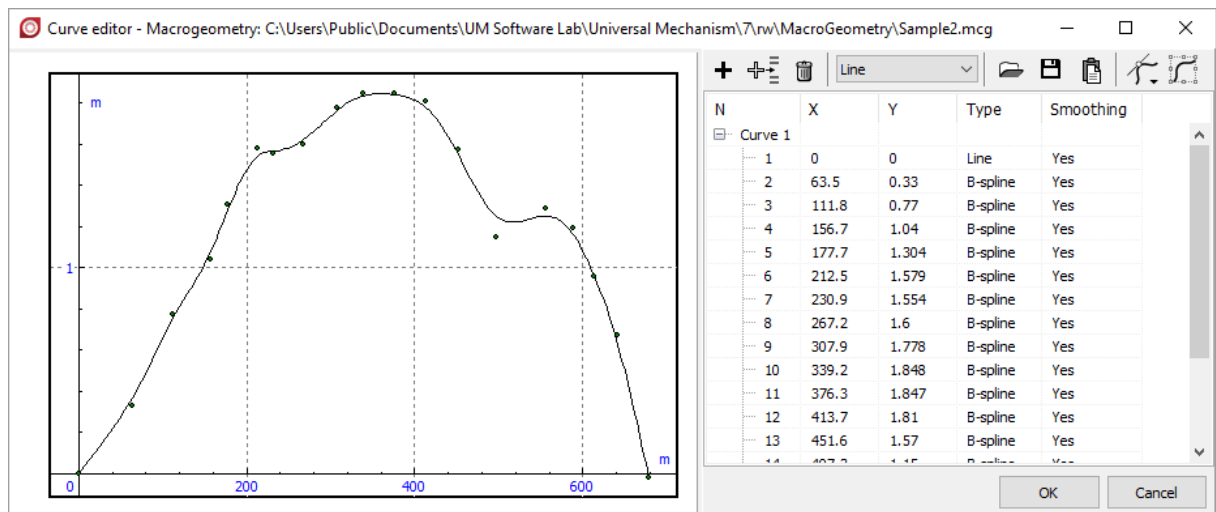


Figure 1.150. Parameters of point section

X parameter corresponds to curve length value; Y parameter corresponds to Z coordinate of track axis. B-splines are recommended for approximation.

**Remark.** Use zero smoothing radius value for constant slope section if point section is next to it. Vertical profile will have a break if not.

**1.4.1.3.3. Import of measured track geometry**

Track measurement is applied for getting of real geometrical characteristics of the railroad path relative to some basic point. Modern methods of measurements based on the usage of high-precision optical tachometers, GPS receivers and inclinometer provide list of data for each point of measurement like coordinates, track width, cant, etc.

Text files with measured data can be imported to macrogeometry model. The following format of text files is used by default:

Track description							
N0	X	Y	Z	Mark	TW	DZ	Side
10000	5953037.71	322385.255	472.55	2 st	1530	100	Left
10001	5953045.67	322375.568	472.534	2s "Y2"	1530	100	Right

Mentioned designations are as follows:

*N0* is a number of measured point (not used);

*X, Y, Z* are coordinates;


*Mark* is additional data (not used);

*TW* is track width;

*DZ* is curve cant;

*Side* is curve side (used for cant setting).

The files can be prepared in Excel program and saved as text files with tab delimiters.

Press  button at the top right corner of **Macrogeometry** form to import data. **Track measured data loading** form will appear (Figure 1.144). Select the file, set standard track width and curve approximation type.

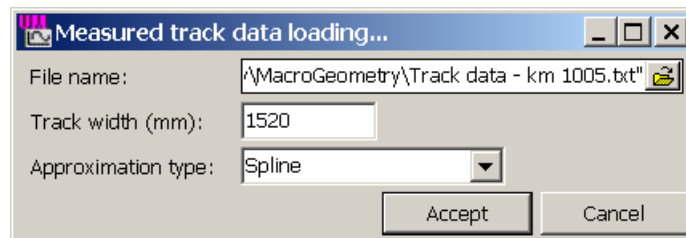


Figure 1.151. Track measured data loading form

New point curve section will be added to the horizontal plane section list and vertical profile section list (see Sect. 1.4.1.3.1.4. "Point curve", p. 1-130, Sect. 1.4.1.3.2.2. "Point section", p. 1-133). Start point of the approximated curve will be placed at the last point of plane and profile correspondently. Section length is evaluated from XY curve length. The user can change curves or approximation type.

**Remark.** Remember that track widening, cant and Z coordinate of track are set relative to the track distance, which is evaluated from XY curve shape. Changing of the XY curve can result in data distortion. Therefore we recommend not to change XY curve after import and repeat import if it is necessary to change XY curve approximation type.

#### 1.4.1.4. Creation of files with switch parameters

To create or edit a file with switch parameters use the **Tools | Switch** menu command.

Switch parameters are saved in \*.swt file. The upper text box contains the name of the switch (optional parameter).

See Sect. 1.2.1.3. "*Switch geometry*", p. 1-71 for more details.

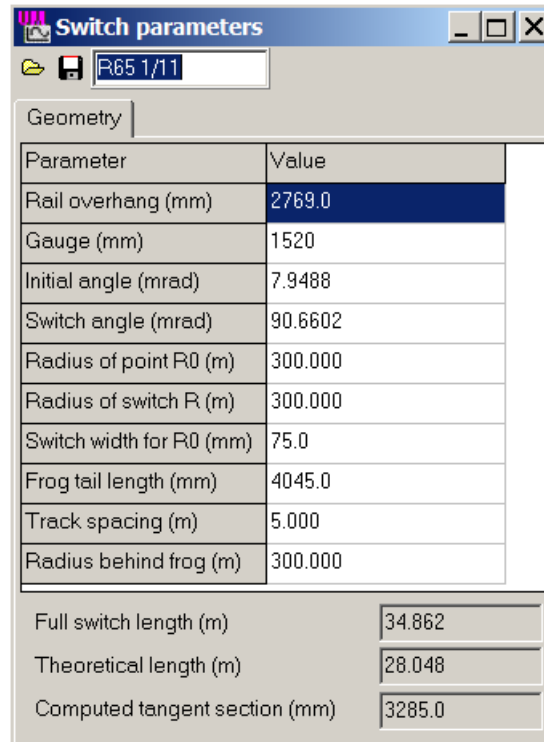


Figure 1.152. Switch window parameters

#### 1.4.1.5. Estimation of track quality

This tool corresponds to the Russian regulations and available for Russian version of UM only.

### 1.4.2. Setting parameters of rail vehicle simulation

Parameters for simulation of rail vehicle dynamics are available in the simulation inspector. To call the inspector use one of the following methods:

- **Analysis | Simulation** menu command;
- **F9** key.

A considerable part of the parameters can be set in a standard manner, [Chapter 4](#), Sect. *Preparing for integration*. Here we consider some features of the parameters setting for a rail vehicle.

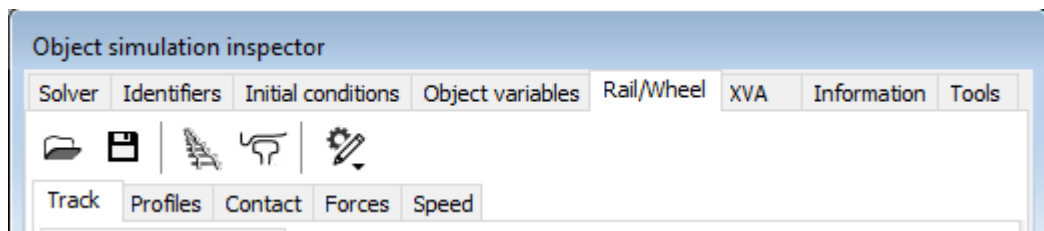


Figure 1.153. Tab for rail vehicle simulation parameters

Parameters and options related to rail vehicle are located on the Rail/Wheel tab, Figure 1.153. The tab contains the following buttons:

- is used for reading a rail vehicle configuration file \*.rwc;
- is used for saving the current options and parameters in the configuration file \*.rwc;
- is used to view the assigned rail irregularities in a graphic window;
- is used to view of selected rail and wheel profiles in a graphic window;
- is the tab navigator; click the button to get the menu for direct access of tabs related to rail vehicle, Figure 1.154.

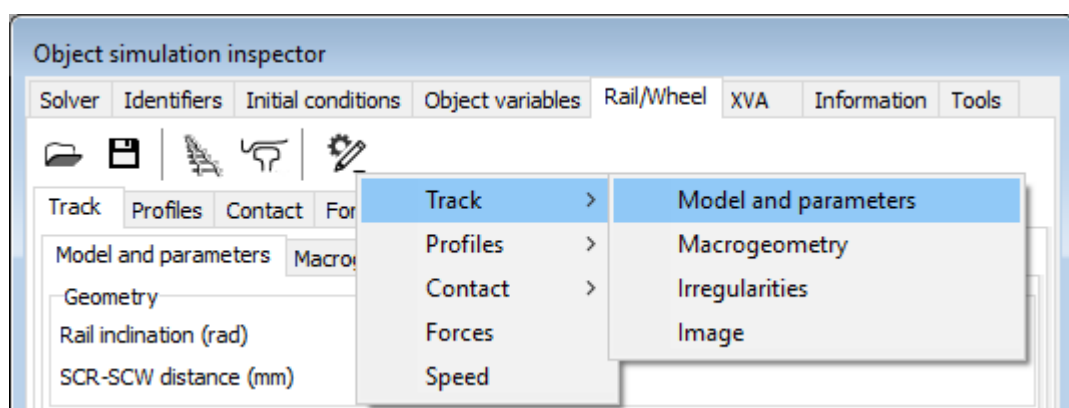


Figure 1.154. Menu of navigator

A number of tabs are available on the **Rail/Wheel** tab.

**Track:** assignment of macrogeometry, irregularities, stiffness, gauge, rail inclination and so on, Sect. 1.4.2.4. "Track parameters", p. 1-150,

**Profiles:** assignment of wheel and rail profiles, wheel forms, wheel radius differences, Sect. 1.4.2.3. "Assignment of rail and wheel profiles", p. 1-145, Sect. 1.4.2.7. "Additional parameters", p. 1-178,

**Contact:** selection of contact model and its parameters, coefficient of friction in rail/wheel contact, Sect. 1.4.2.5. "Parameters for computation of rail-wheel contact forces", p. 1-167,

**Forces:** setting forces in automatic couplers from files,

**Speed:** setting mode of longitudinal motion of vehicle (neutral, constant speed, zero speed, speed as a function of time or distance), Sect. 1.4.2.2. "Modes of longitudinal motion of vehicle", p. 1-139.

### 1.4.2.1. Configuration files

#### 1.4.2.1.1. Configuration file of rail vehicle

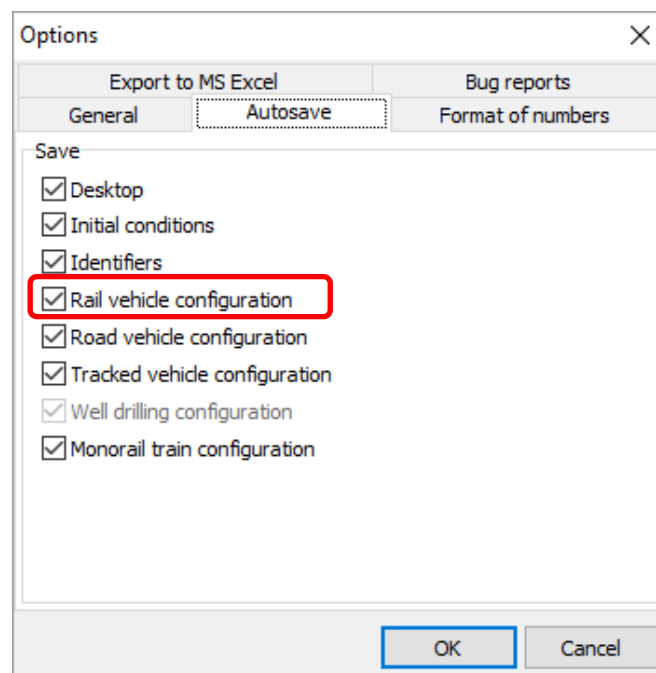




Figure 1.155. Key for automatic saving of rail vehicle configuration file

All data entered in the **Rail/Wheel** tab are stored in a text file \*.rwc by the  button. The previously created files can be read by the  button. All current options and parameters can be save in the file *last.rwc* on close of the **UM Simulation** program or on change of the model. The key **Rail vehicle configuration** in the **Options** window is user to activate or deactivate this function, Figure 1.155. The **Tools | Options** menu command calls the **Options** window.

**Remark 1.** If *last.rwc* file is presented in the model directory, it is read independently on value of automatic saving key in Figure 1.155, and the corresponding rail vehicle parameters are set. The same is valid for the file of identifiers **last.par**. **That is why changes in identifier values made in the UM Input program are ignored if the last.par is available in the object directory.** To accept the numeric value of identifiers from the input.dat file, the last.par file must be **deleted** before loading the object.

**Remark 2.** Rail vehicle configuration file can be read from the directory of another model.

#### 1.4.2.1.2. Group of configuration files of rail vehicle

The current configuration of a rail vehicle is stored together with all other options by the **File | Save configuration | All options** menu command, Figure 1.156. As a result, a group of configuration files is created, which contain the full information about the current state of the model. Files in the group have equal names but different extensions. Later the user can read this group by the **File | Load configuration | [Name of group]** command, Figure 1.157.

For instance, we have completely prepared settings for analysis of stability of a rail vehicle numeric method is selected, simulation time and accuracy is specified, a number of graphic and animation windows are prepared, numeric value of identifiers, wheel and rail profiles, rail inclination, and coefficients of friction and so on are assigned. After that we have made the analysis of stability of the vehicle and decided to store the successive options and parameters for later usage. We save all options by the **File | Save configuration | All options** command with the group name **Stability**. The following files are created:

Stability.icf (program desktop and parameters of numeric method), lists of disabled and non-stiff forces;

Stability.par (numeric values of identifiers);

Stability.rwc (rail vehicle configuration file);

Stability.xv (file of initial coordinate values);

Stability.sim (Matlab/Simulink interface options);

All this files will be read by the **File | Load configuration | Stability** command.

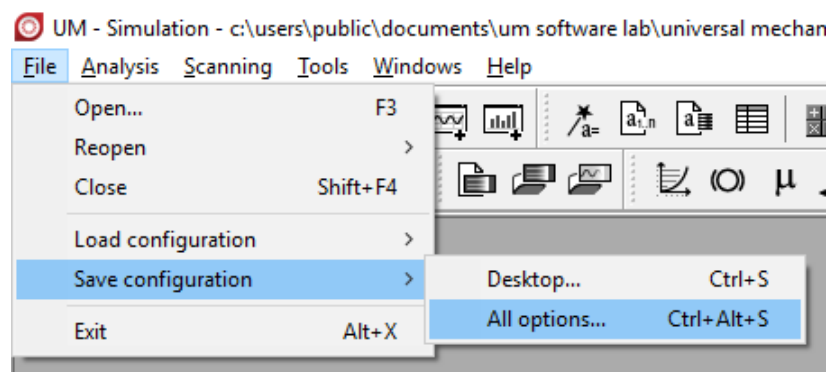


Figure 1.156. Creating a group of configuration files

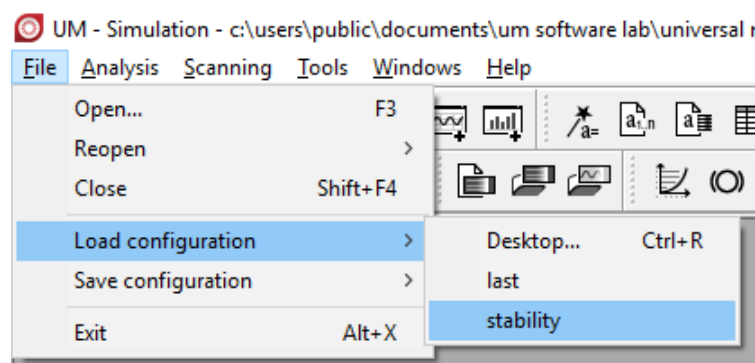


Figure 1.157. Reading a group of configuration files

### 1.4.2.2. Modes of longitudinal motion of vehicle

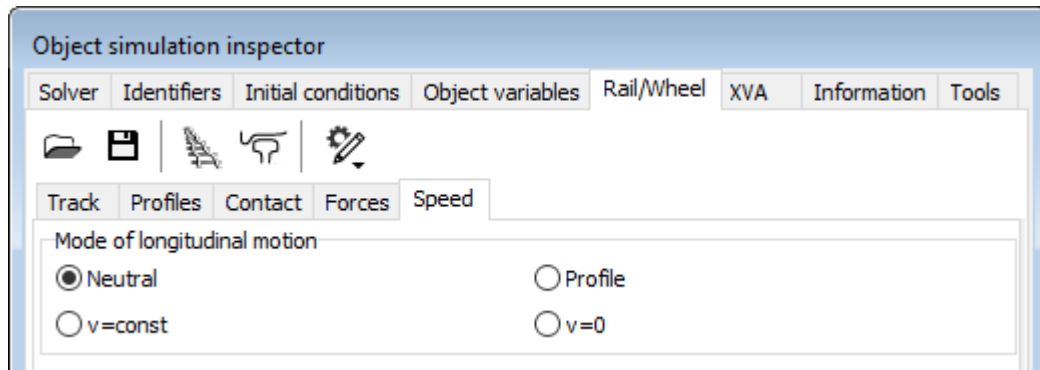


Figure 1.158. Longitudinal motion modes

Modes of longitudinal motion of vehicle are set on the **Speed** tab of the inspector.

#### 1.4.2.2.1. Neutral

In this mode the initial speed value is set by the **v0** identifier, Figure 1.159. The speed decreases due to resistance wheel-to-rail forces.

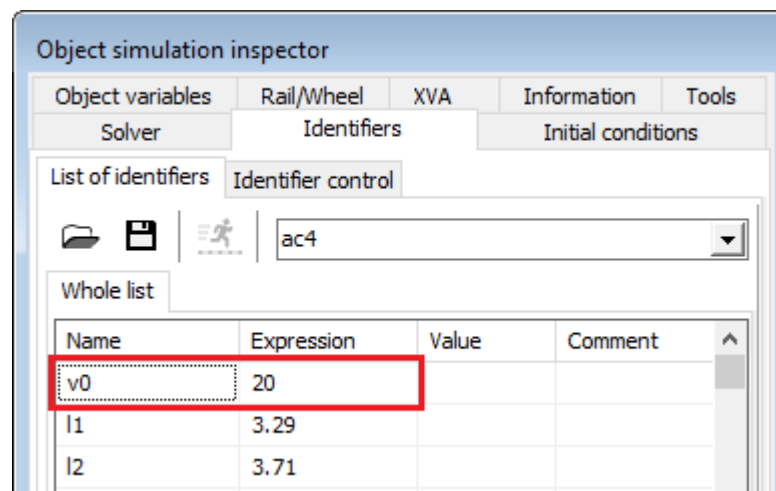


Figure 1.159. Identifier of speed

#### 1.4.2.2.2. v=const

It is a constant speed mode. The nearly constant value of the vehicle speed is supported automatically by the longitudinal force

$$F = -k(v-v_0),$$




where  $v_0$  is the desired speed,  $v$  is the current speed, and  $k$  is the amplifier. The force is applied to a body selected by the user (usually to the car body) to a point which coordinates should be set in the body-fixed system of coordinates, Figure 1.160. Usually the point lies on the coupling level; the longitudinal coordinate of the point is not important if the body is rigid.

In this mode the desired speed value is set by the **v0** identifier, Figure 1.159.

Mode of longitudinal motion		
<input type="radio"/> Neutral	<input type="radio"/> Profile	
<input checked="" type="radio"/> v=const	<input type="radio"/> v=0	
Speed control parameters		
Body	ac4.Car body.	
Point	0.000	0.000 1.2
Amplifier	1000000	

Figure 1.160. Constant speed mode

### 1.4.2.2.3. Profile

The vehicle speed is controlled according to a dependence on a time or distance, the **Abscissa type** group. The control force is similar to that in the previous mode (**v=const**). A curve editor is available by the  button for setting the speed profile, Figure 1.161, Figure 1.162. The   buttons are used for reading previously created profiles and for saving the current curve.

**Remark.** If initial speed is zero, dependence of the speed on time must be used (not on distance).




Mode of longitudinal motion		
<input type="radio"/> Neutral	<input checked="" type="radio"/> Profile	
<input type="radio"/> v=const	<input type="radio"/> v=0	
Speed control parameters		
Body	ac4.Car body.	
Point	0.000	0.000 1.2
Amplifier	1000000	
Speed profile		
 		
Create/Edit speed profile		
Abscissa type		
<input type="radio"/> Time	<input checked="" type="radio"/> Distance	
Data Input/Edit	Points: 0 	

Figure 1.161. Speed profile mode

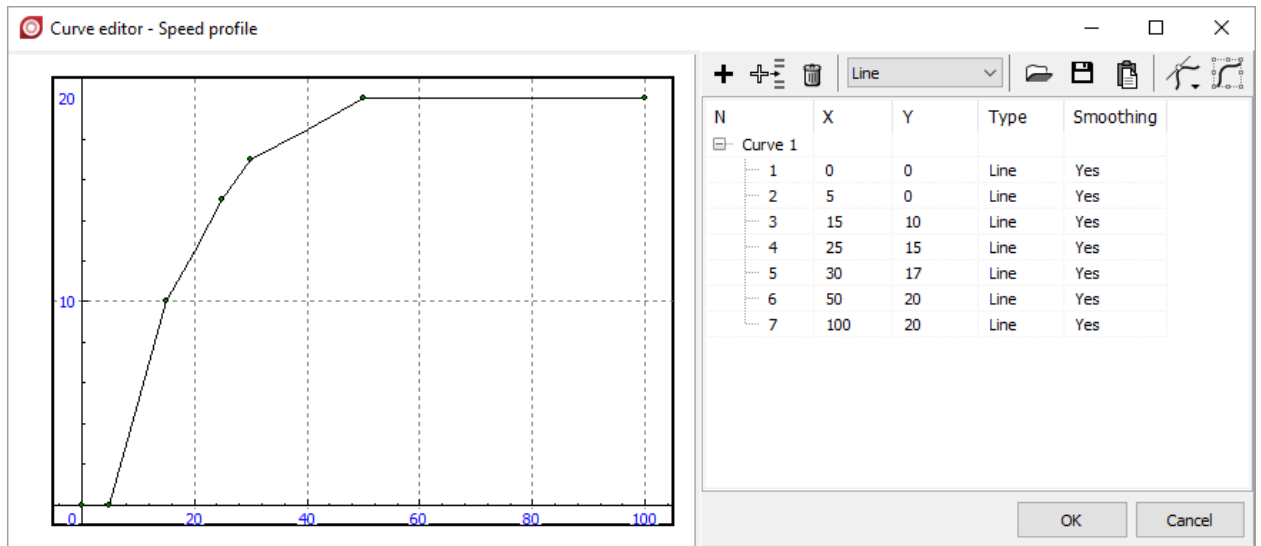


Figure 1.162. Example: vehicle speed vs. time

1.4.2.2.4. v=0

Zero velocity mode. This mode is used to bring the vehicle to the equilibrium or load-induced oscillations simulation, defined depending on time. In the second case blocking of several degrees of freedom of wheelsets could be useful: longitudinal and lateral motions (X, Y) as well as rotations about the lateral and vertical axes (aX, aZ), Figure 1.163. To block a degree of freedom turn on the corresponding check box.

The figure shows a dialog box with the following options:

- Mode of longitudinal motion:
  - Neutral
  - Profile
  - v=const
  - v=0
- Block wheelset shift:
  - X
  - Y
  - aY
  - aZ
- Finish test automatically

Figure 1.163. Parameters of zero velocity mode

To bring the vehicle to equilibrium position turn on **Finish test automatically** (Figure 1.163), then start the integration process and wait until the test finishes. As a rule, this is the first step when the work with a new railway vehicle model begins. The plot of vehicle summarized kinetic energy can be useful during the equilibrium test. The variable of kinetic energy is available in tab **Variables for group of bodies** of **Wizard of variables**. Equilibrium test is finished when kinetic energy value becomes small enough (Figure 1.164).

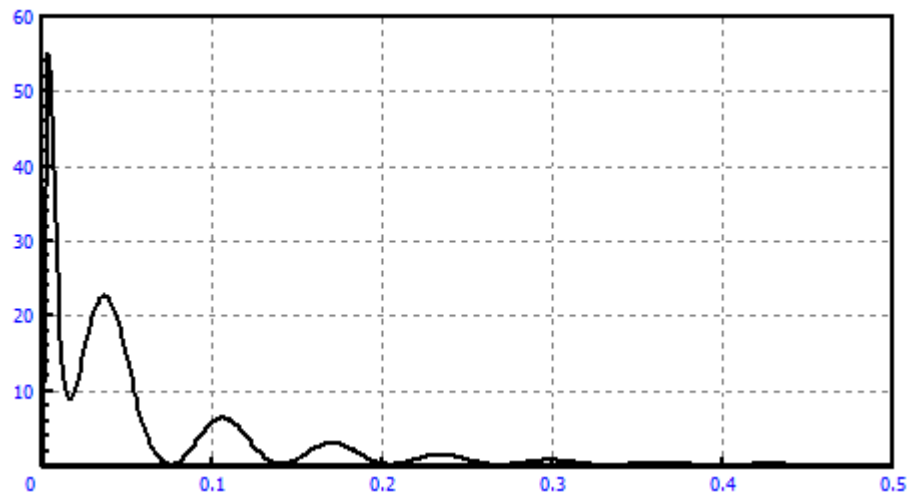


Figure 1.164. Changing of kinetic energy during the equilibrium test implementation

#### 1.4.2.2.5. Motion of vehicles with different speed

It is possible to model of several vehicles or groups of vehicles moving with different initial speed. The following conditions are important.

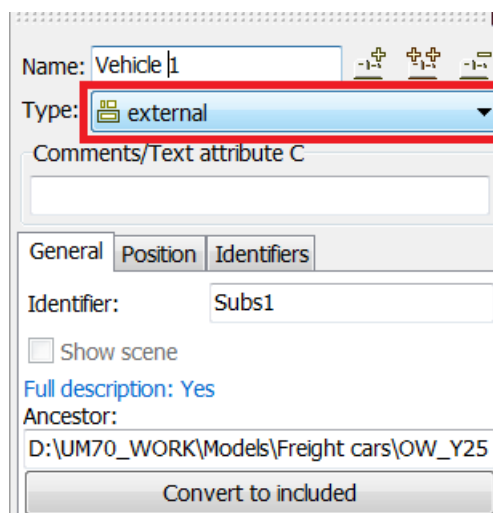


Figure 1.165. Vehicle as external subsystem

- Each group of vehicles must be included in the model as an **external** subsystem, Figure 1.165. For instance, the user wish simulate two vehicles moving with different speed, so the vehicles must be included in the model as two external subsystems, see [Chapter 2](#), Section *Subsystems*, [Chapter 3](#), Section *Subsystems*.

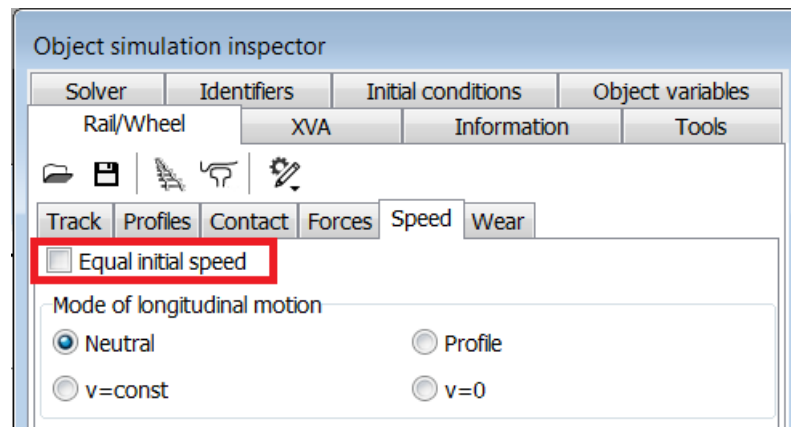


Figure 1.166. Option for different speed of vehicles

- Neutral mode of longitudinal motion must be selected.
- **Equal initial speed** option must be unchecked, Figure 1.166.

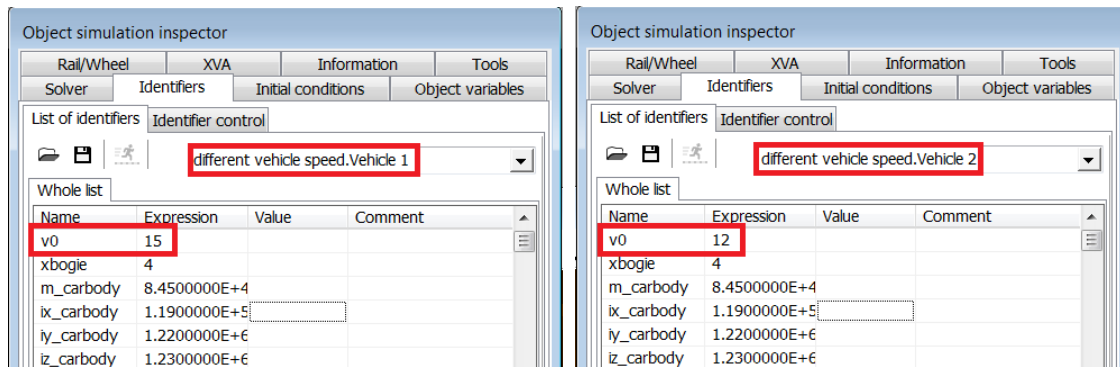


Figure 1.167. Assignment of speeds

- Initial speed for each of the external subsystem is specified by the corresponding identifier  $v0$  in the corresponding subsystem, Figure 1.167.

#### 1.4.2.2.6. Speed unit

Speed specification both by the identifier  $v0$  and by a function can be done either in km/h or m/s. To specify the unit, open the **Option** window by the **Tools | Options** menu command and set the desired unit, Figure 1.168. The selected option is stored in the registry of the local computer. **This option can be used carefully, because errors possible due to transferring the model to another computer or by reading old configuration files.**

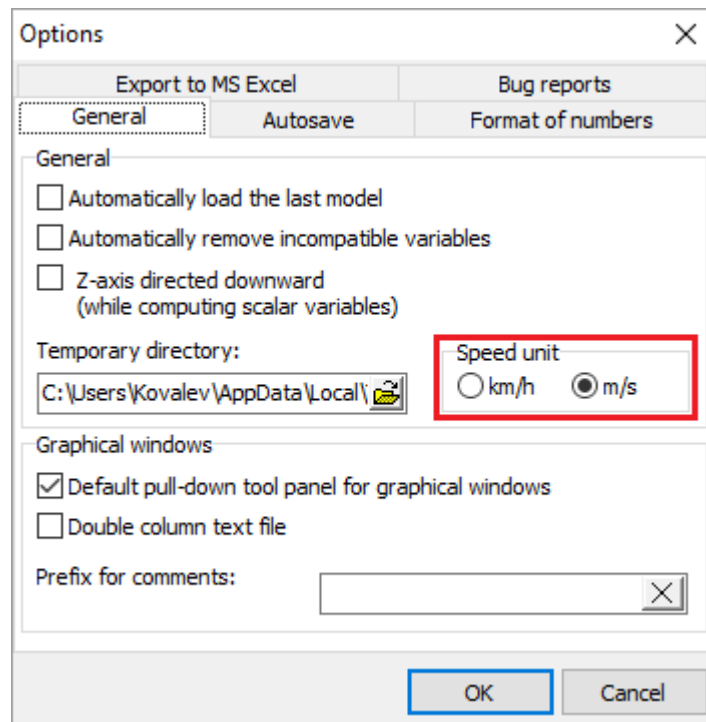




Figure 1.168. Setting speed unit

### 1.4.2.3. Assignment of rail and wheel profiles

The **Rail/Wheel | Profiles** tab of the inspector is used for assignment profiles to the rail and wheels. References to files the assigned profiles are stored in a vehicle configuration file \*.rwc.

#### 1.4.2.3.1. Assignment of rail profiles

Use the  buttons to assign profiles for the left and right rails (Figure 1.169). The set of rails is used both for the fast assignment of profiles, and for their internal parameterization in scanning projects, Sect. 1.6.1. "List of internal identifiers parameterizing operation conditions of rail vehicles", p. 1-244. The  button helps the user to view the assigned profiles.

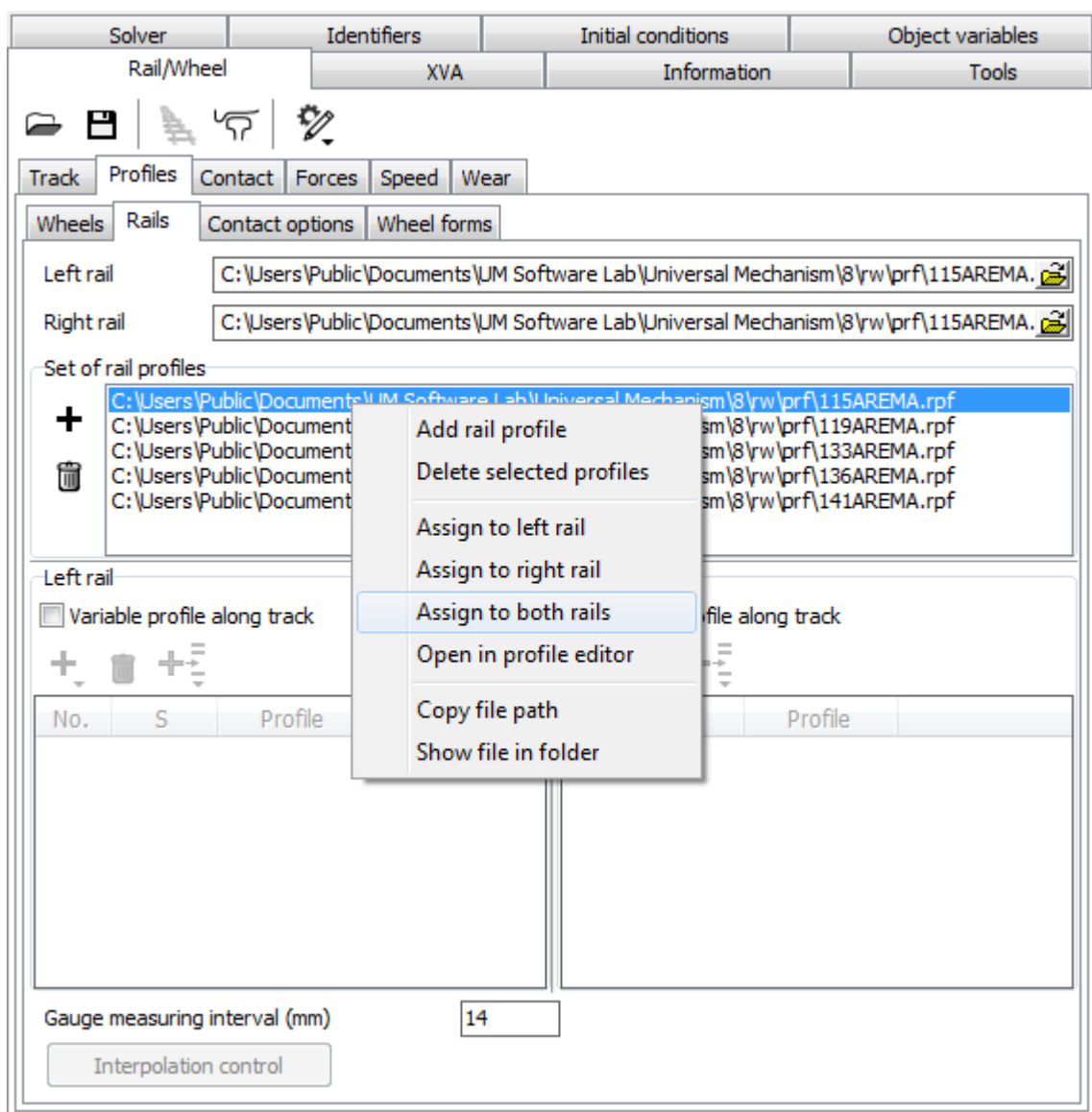


Figure 1.169. Assignment of rail profiles. Popup menu of the rail profile list.

The following rail profiles are delivered with UM Loco:

115AREMA.rpf	NA profile, 115 lb/yd
119AREMA.rpf	NA profile, 119 lb/yd
132AREMA.rpf	NA profile 132 lb/yd
133AREMA.rpf	NA profile, 133 lb/yd
136AREMA.rpf	NA profile, 136 lb/yd
141AREMA.rpf	NA profile, 141 lb/yd
BS113A_crv.rpf	European profile BS113
Chinese R60.rpf	Chinese profile R60
Chinese R75.rpf	Chinese profile R75
r50.rpf	Russian R50
r65new.rpf	Russian R65
r65old13.rpf	Russian R65 with 13 wear
UIC60new.rpf	European UIC60
UIC60l_1.rpf	UIC60 (is used in Manchester tests)
UIC60r_1.rpf	UIC60 (is used in Manchester tests)
UIC60_crv.rpf	UIC60
UIC60_Worn.rpf	UIC60

**Gauge measuring interval** is a distance below a plane that rests across the top of the two rails which defines gauge measuring points taking the rail inclination into account (Figure 1.170). It is used to create *Gauge* variable, see. Sect. 1.4.3.1.2. "Variables related to the wheelset", p. 1-190.

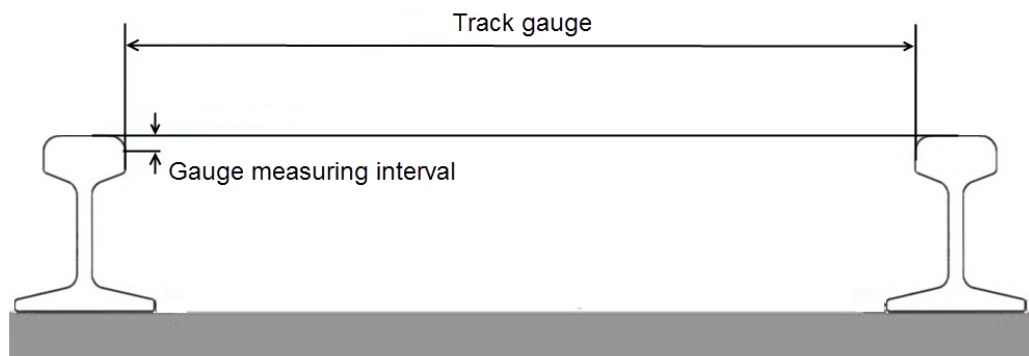



Figure 1.170. Gauge measurement

### 1.4.2.3.2. Assignment of wheel profiles

Assignment of wheel profiles has two stages. At the first stage a list of wheel profiles is created (Figure 1.171). The list should include at least one profile. The **+** button in the left panel is used to add a profile to the list. The button  shows all profiles in the list.

To assign profiles from the list to the wheels

- open the **Rail/Wheel | Profiles | Wheels | Profiles** tab, Figure 1.171;
- call the popup menu by clicking the right mouse button and select a profile to assign it to *all* the wheels;

- if different profiles should be assigned to different wheels, double click on the necessary wheel by the left mouse button and change the profile until the desired profile name appears.

**Remark.** List of profiles is used for internal parameterization of the profiles in scanning project, Sect. 1.6.1. *"List of internal identifiers parameterizing operation conditions of rail vehicles"*, p. 1-244.

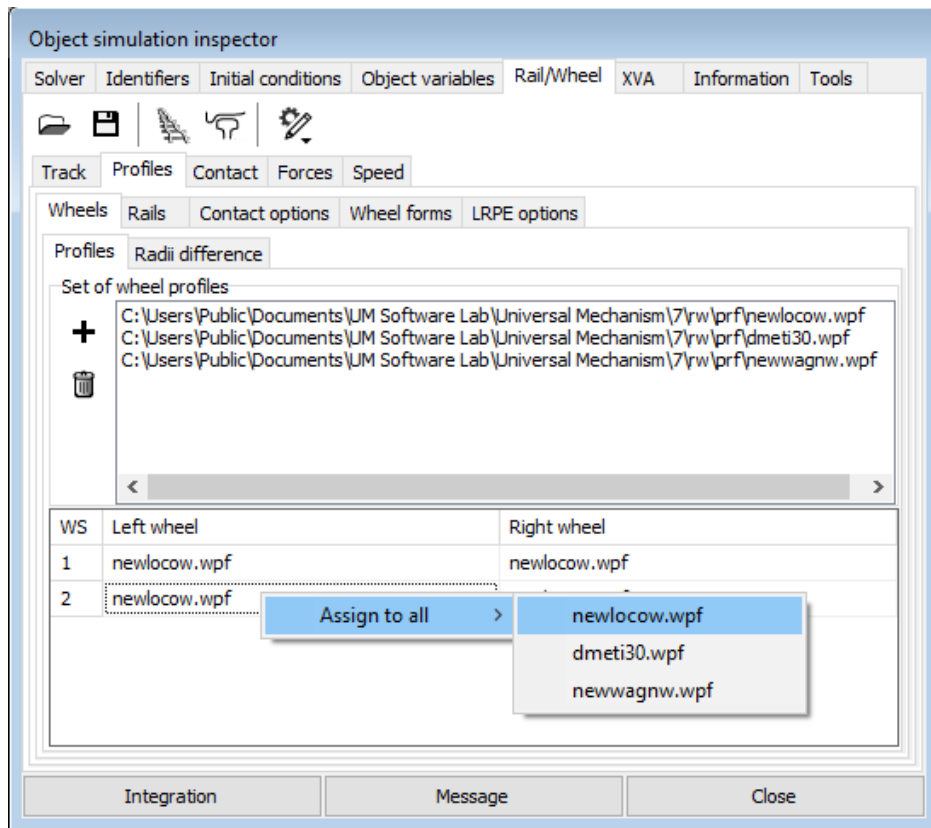


Figure 1.171. Assignment wheel profiles

UM database of wheel profiles includes the following files.

AAR.wpf	NA profile AAR
Chinese LM.wpf	Chinese profile LM
Chinese LMA.wpf	Chinese profile LMA
dmeti30.wpf	Russian profile DMetI
newlocow.wpf	Russian new locomotive profile
newwagnw.wpf	Russian new car profile
s1002.wpf	European S1002
S1002l_1.wpf	European S1002 (Manchester benchmarks)
S1002r_1.wpf	European S1002 (Manchester benchmarks)
s1002worn.wpf	S1002, worn
s1002worn2.wpf	S1002, worn

### 1.4.2.3.3. Rail profile evolution along the track

To describe a changing the profile of the left or/and right rail along the track (Sect. 1.2.1.1. "Geometry of rails in an ideal track", p. 1-66), the following steps should be done.

- Create the list of profiles with the **Rail/Wheel | Profiles | Rails** tab. Use the **+** **🗑** buttons or the popup menu to add and delete profiles.
- Use the list **Rail profiles in control points** to set a sequence of profiles  $R_i$  the track as well as their positions along the track  $S_i$  (Sect. 1.2.1.1. "Geometry of rails in an ideal track", p. 1-66),  $i=1,2,\dots$ . Add profiles to the sequence with the help of the popup menu. After that set their positions in the **Track** column, see Figure 1.172.

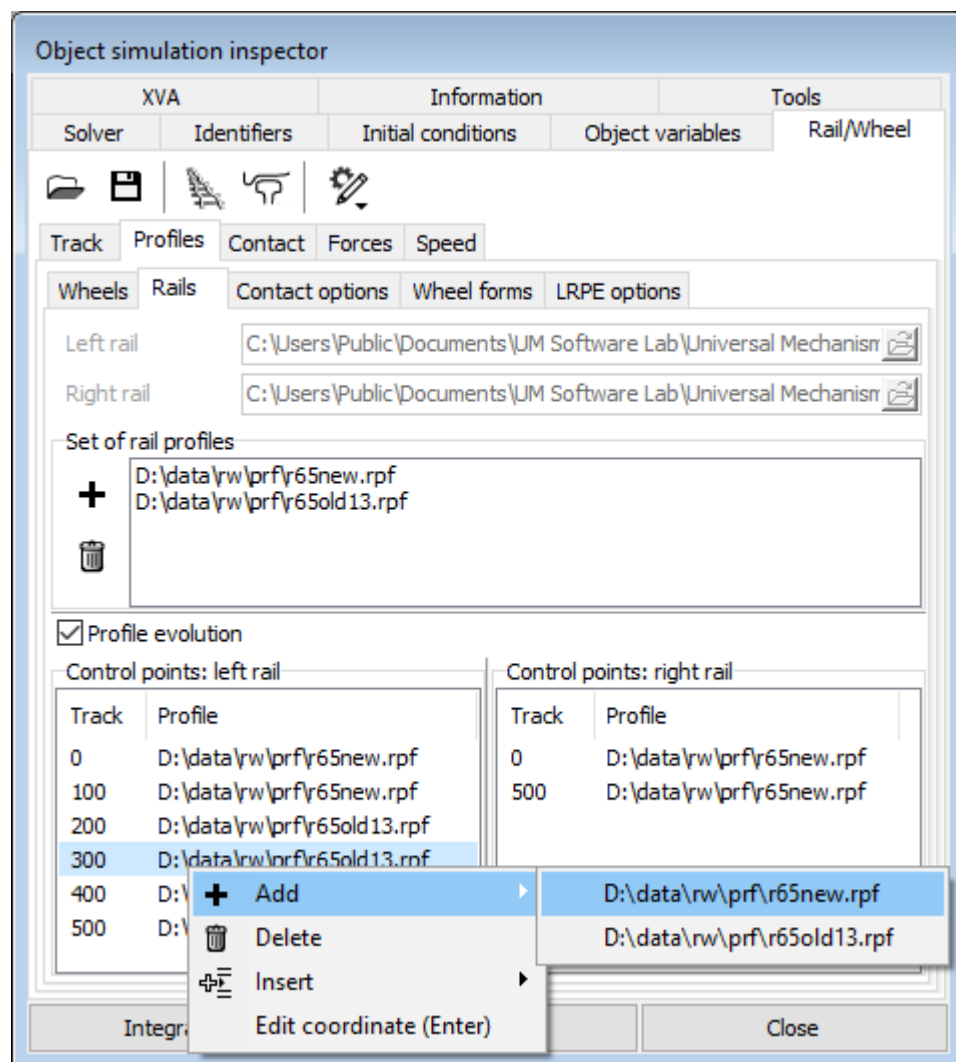


Figure 1.172. Creation of sequence of profiles along the track

**Example.** Figure 1.172 shows a creation of sequence of profiles for the outer rail of a right curve with the following parameters (Sect. 1.2.1.2. "Geometry of curve", p. 1-67).

$$L0 = 10 \text{ m}, P11 = 50 \text{ m}, S1 = 200 \text{ m}, P12 = 50 \text{ m}.$$

The new profile *r65new* (Figure 1.173) is used at straight sections. At the first 50 m transition (section from 10 m to 60 m) the new profile is transformed to the worn one *r65old*. In the

steady curve (from 60 m to 260 m) the profile does not change. At the second 50 m transition (from 260 m to 310 m) the profile is transformed from the worn to the new one.

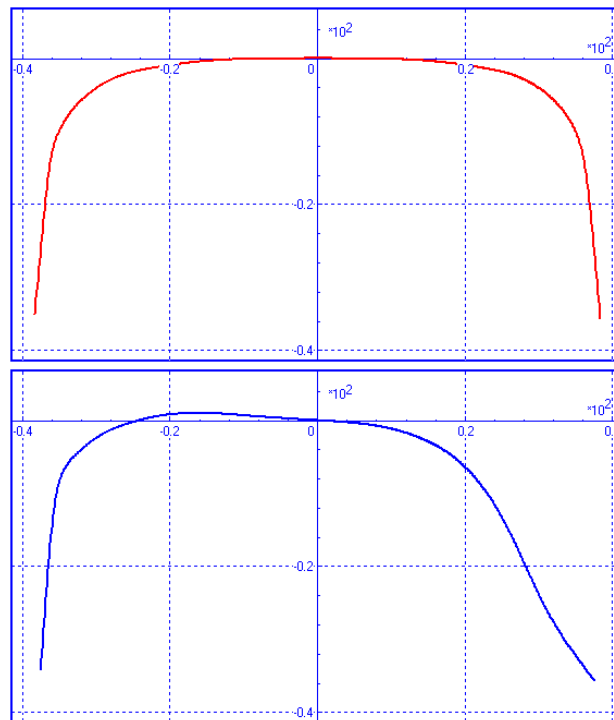


Figure 1.173. Profiles *r65new* (left) and *r65old*

#### 1.4.2.4. Track parameters

The track parameters are defined on the tab sheet **Rail/Wheel | Track** in **Object simulation inspector**. The tab contains **Model and parameters**, **Macrogeometry**, **Irregularities** and **Image** tabs. On the tab **Model and parameters** rail inclination and track gauge are set, track model and its parameters are chosen. On the **Macrogeometry** tab track macrogeometry is set. On the **Irregularities** tab the irregularities of rail threads are set. You can set the track image in animation window on the **Image** tab.

##### 1.4.2.4.1. Track model choosing and parameters setting

Rail inclination, track gauge, track model choosing and its parameters are set on the **Model and parameters** tab, Figure 1.174.

**Rail inclination**  $\alpha_{r0}$  is set in radians, Sect. 1.2.1.1. "*Geometry of rails in an ideal track*", p. 1-66.

Exact value of the gauge is specified by the distance between the systems of coordinates of rail and wheel  $\Delta y$  (SCR-SCW distance), Sect. 1.2.1.1. "*Geometry of rails in an ideal track*", p. 1-66.

The track models are described in Sect. 1.2.2. "*Track models*", p. 1-74.

Solver	Identifiers	Initial conditions	Object variables
Rail/Wheel	XVA	Information	Tools
Track	Profiles	Contact	Forces
Model and parameters <span style="float: right;">Macrogeometry</span> <span style="float: right;">Irregularities</span> <span style="float: right;">Image</span>			
Geometry			
Rail inclination (rad)	<input type="text" value="0.05"/>		
SCR-SCW distance (mm)	<input type="text" value="3.000"/>		
Track model			
<input checked="" type="radio"/> Massless rail <input type="radio"/> Inertial rail <input type="radio"/> Flexible track			
Parameters			
Type of stiffness parameter definition			
<input checked="" type="radio"/> Number <span style="margin-left: 100px;"><input type="radio"/> Curve</span>			
Stiffness per one rail			
Vertical stiffness (N/m)	<input type="text" value="44 000 000"/>		
Lateral stiffness (N/m)	<input type="text" value="18 000 000"/>		
Torsional stiffness (Nm/rad)	<input type="text" value="6.6E25"/>		
<input type="checkbox"/> Torsional stiffness is taken into account			
Damping per one rail			
Vertical damping (Ns/m)	<input type="text" value="400 000"/>		
Lateral damping (Ns/m)	<input type="text" value="100 000"/>		

Figure 1.174. Track parameters

#### 1.4.2.4.1.1. Track stiffness and damping

##### 1.4.2.4.1.1.1. Massless rail

Vertical and lateral stiffness of a rail can be specified in two different forms.

- Constant stiffness along the track.

In this case the stiffness constants are equal for the left and right rails. Their numeric values are set in the boxes **Vertical stiffness**, **Lateral stiffness**, **Torsional stiffness**, Figure 1.174. The torsional stiffness is taking into account if the corresponding option is active, Figure 1.174.

- Variable stiffness along the track, Figure 1.175.

The variable rail stiffness is described by sets of points in dependence on the longitudinal coordinate separately for the left and right rails. Use the button to call the curve editor for entering the variable stiffness. In the case of variable vertical and lateral stiffness, the constant torsional stiffness can be applied.

Parameters	
Type of stiffness parameter definition	
<input type="radio"/> Number	<input checked="" type="radio"/> Curve
Stiffness along the track (N/m)	
Vertical, left rail	(none) ...
Vertical, right rail	(none) ...
Horizontal, left rail	(none) ...
Horizontal, right rail	(none) ...
Damping per one rail	
Vertical damping (Ns/m)	400 000
Lateral damping (Ns/m)	100 000

Figure 1.175. Variable stiffness

**Damping constants** (Ns/m) are equal for the left and right rails, Figure 1.174.

To get track stiffness and damping, which is variable in the longitudinal direction, as well as set different values of these parameters for the left and right rail, the user should use the programming in the *Control file* (Chapter 5, Sect. 5.1). The following procedure in the *Control file* is used for programming the track parameters.

```
function TrackStiffness( position : real_; var cLeftZ, cRightZ, cLeftY,
cRightY, dLeftZ, dRightZ, dLeftY, dRightY : real_ ) : integer;
begin
  Result := 0;
end;
```

**Input:**

*position* is longitudinal coordinate for which the irregularities are computed.

**Output:**

*Result* returns function value. If the return value is 1, the programmed values will be taken into account, otherwise they are ignored.

*cLeftZ*, *cRightZ* are vertical stiffnesses of the left and right rails, N/m;

*cLeftY*, *cRightY* are lateral stiffnesses of the left and right rails, N/m;

*dLeftZ*, *dRightZ* are vertical damping coefficients of the left and right rails, Ns/m;

*dLeftY*, *dRightY* are lateral damping coefficients of the left and right rails, Ns/m.

**Remark 1.** If the function returns 1, parameters entered in Figure 1.174 are ignored except the cases in the Remark 2.

**Remark 2.** Zero values of stiffness coefficients *cLeftZ*, *cRightZ*, *cLeftY*, *cRightY* are ignored and replaced by values entered in Figure 1.174.

#### 1.4.2.4.1.1.2. Inertial rail

Stiffness and damping properties of railway track for **Inertial rail** model are defined by the identifiers (Sect. 1.1.3.2. *Wheelset with six degrees of freedom*, p. 1-10). To change the identifier use **Identifiers | List of Identifiers** tab in the Object simulation inspector is used. To define var-

ables along the track stiffness and dissipation identifiers control is used (**Object Simulation Inspector** | **Identifiers** | **Identifier control**).

#### 1.4.2.4.1.1.3. Flexible track

You can define or change the track stiffness and damping of **Flexible track** model in **Wizard of flexible railway track**, [Chapter 27](#), Sect. 27.5.5.

#### 1.4.2.4.2. Track macrogeometry

To define track macrogeometry use the **Rail/Wheel** | **Track** | **Macrogeometry** tab in the object simulation inspector, Figure 1.176.

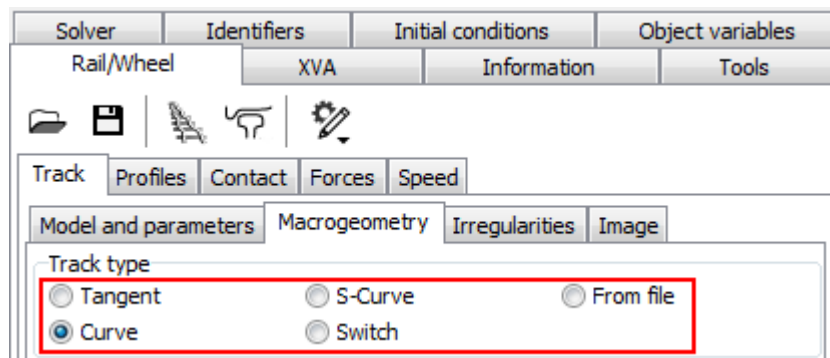


Figure 1.176. Choice of track macrogeometry

##### 1.4.2.4.2.1. Track type: tangent

Motion in a tangent section of unlimited length.

##### 1.4.2.4.2.2. Track type: curve

Motion in a right curve (Figure 1.177) including a tangent section before the curve (**L1**), transient sections (**P11**, **P12**), steady curve of radius **R1** and length **S1**, a *positive cant for the outer rail*, and an additional widening in the curve **dY1**. All parameters of the curve including the additional widening **dY1** should be set in meters, see Sect. 1.2.1.2. "Geometry of curve", p. 1-67 for more details.

The **L** is equal to the full length of the curve including the tangent section **L1**. The **V'** is equal to the vehicle speed (m/s) in case of zero value of the uncompensated acceleration.

**Smoothing** parameter is used for smoothing the vertical junctions at ends of the transition by arc of circle. The parameter here sets the meters length of the smoothed section.

Figure 1.177. Curve parameters

### 1.4.2.4.2.3. Track type: S-curve

S-curve is a combination of a right curve followed by the left curve.



**Remark.** Transient sections of a curve and S-curve are cubic parabolas. It is assumed that the ratio of length of the transient section to the curve radius is small, and does not exceed 0.5. In the case of long transients, geometry type **From file** must be used, where the clothoid is implemented for transient sections.

Figure 1.178. Parameters of S-curve

### 1.4.2.4.2.4. Features of setting cant values in curve and in S-curve

In case of motion in a right curve and in S-curve (Figure 1.177, Figure 1.178) the cant value can be assigned for the *outer* rail by two different ways.

The first method is used if the cant value is positive and constant in steady part of the curve, and increases/decreases uniformly in the transients. In this case the numeric cant value should be set directly in the boxes **H1**, **H2**.

The second method is used if the conditions of the first one do not take place, e.g. when the cant value is not constant in the steady curve. The variable cant value along the track both for the first (right) and for the second (left) curves is set by the  button in the right hand side of the **H1** box, Figure 1.177, Figure 1.178. The curve editor appears, and the cant value (m) versus curve distance can be specified by a sequence of the points, Figure 1.179. After that the box looks like . To set a constant cant back, the user should delete all points in the list in Figure 1.179.

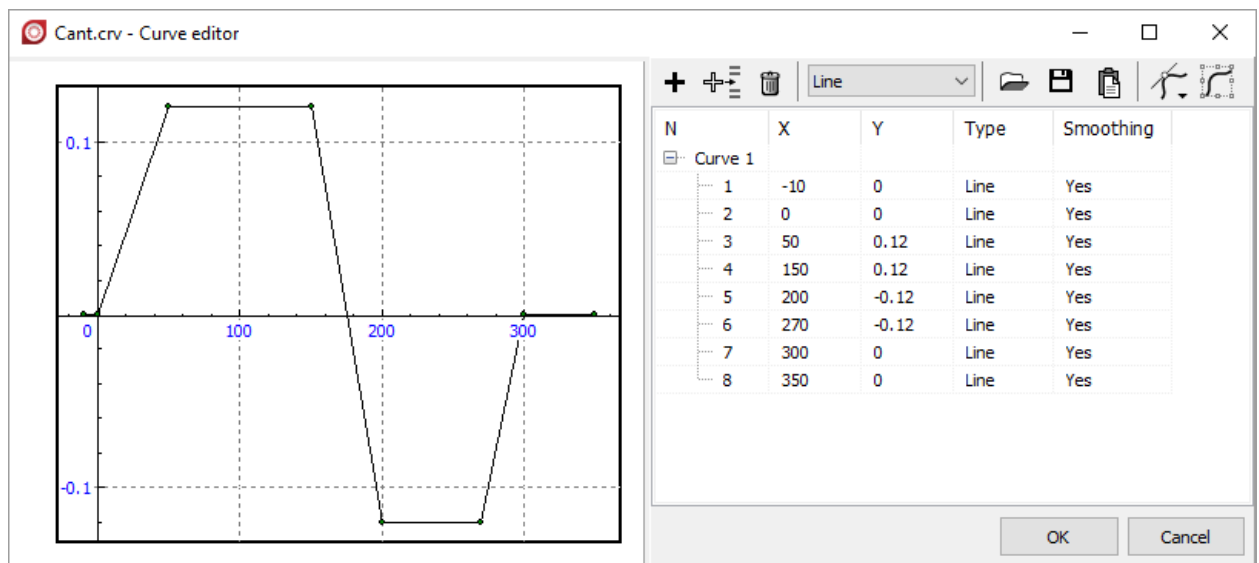


Figure 1.179. Cant versus track distance

The user should take into account the following important remarks concerning usage of the variable cant value.

1. The cant function should be always *zero at tangent sections* of the track (before the first right curve, between the curves, after the second curve). Violation of this requirement leads to discontinuities in the vertical rail position.
2. The cant function starts since *begin of the first transient section*, and does not include the tangent section before the curve. For example, the point (0, 0) in Figure 1.179 corresponds to the start of the curve (not to the start of the track!). The cant function in S-curve takes into account the length of the tangent section between the curves, the **L1** section is ignored only!
3. In case of S-curve, the cant function is valid for both of the curves, not for the first one. In particular it means that the cant value entered in the **H2** edit box is ignored.

**1.4.2.4.2.5. How to model the motion in a left curve?**


S-curve can be easily used for simulation of motion in a left curve if lengths corresponding to the first (right) curve are small, cant is zero, and the radius is large.

**Remark.** In this mode, the smoothing length is ignored.

Track type	
<input type="radio"/> Tangent	<input checked="" type="radio"/> S-Curve
<input type="radio"/> Curve	<input type="radio"/> Switch
<input type="radio"/> From file	
First section	Second section
L1 10	L2 50
P11 0.00001	P21 50
S1 0.00001	S2 200
R1 1000000	R2 600
H1 0	H2 0.09
P12 0.00001	P22 50
dY1 0	dY2 0
L 10.00003	L 350
V' 0	V' 18.310611
Smoothing 8.00	

Figure 1.180. Trick: left curve as S-curve

#### 1.4.2.4.2.6. Track type: switch

Use the  button to open a \*.swt file with the switch parameters, Figure 1.181 (see Sect. 1.4.1.3. "Creation of macrogeometry files", p. 1-125). Other parameters allow the user to simulate both facing and trailing movements in the left and right switches. The **Tangent section** parameter sets the length of a tangent section before the switch.


Track type	
<input type="radio"/> Tangent	<input type="radio"/> S-Curve
<input type="radio"/> Curve	<input checked="" type="radio"/> Switch
<input type="radio"/> From file	
Movement	
<input checked="" type="radio"/> Facing	<input type="radio"/> Trailing
<input type="checkbox"/> Left switch	
Switch	R65 1/11 
Tangent section	10.00

Figure 1.181. Assignment of switch

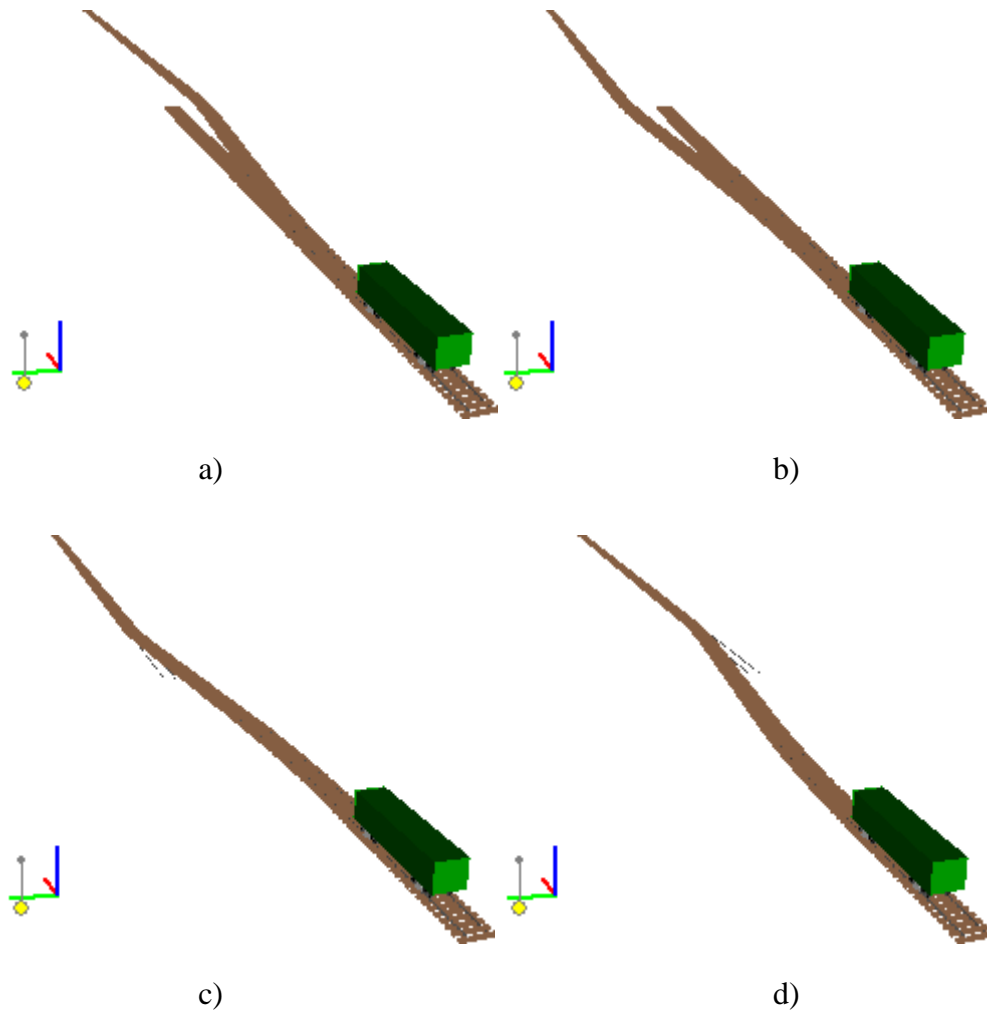




Figure 1.182. Switches: a, b are facing and trailing movements in the right switch; c, d are facing and trailing movements in the left switch

#### 1.4.2.4.2.7. Track type: from file. Set of macrogeometry files

A preliminary created file with track macrogeometry is assigned by the  button, Figure 1.183, see Sect. 1.4.1.3. "Creation of macrogeometry files", p. 1-125.

View and modification of the file data are available by the  button.

**Set of macrogeometry files** is used for a quick assignment of different files, as well for internal parameterization of files used in scanning projects, Sect. 1.6.1. "List of internal identifiers parameterizing operation conditions of rail vehicles", p. 1-244. Double click on the element of the set to assign it as the current one.

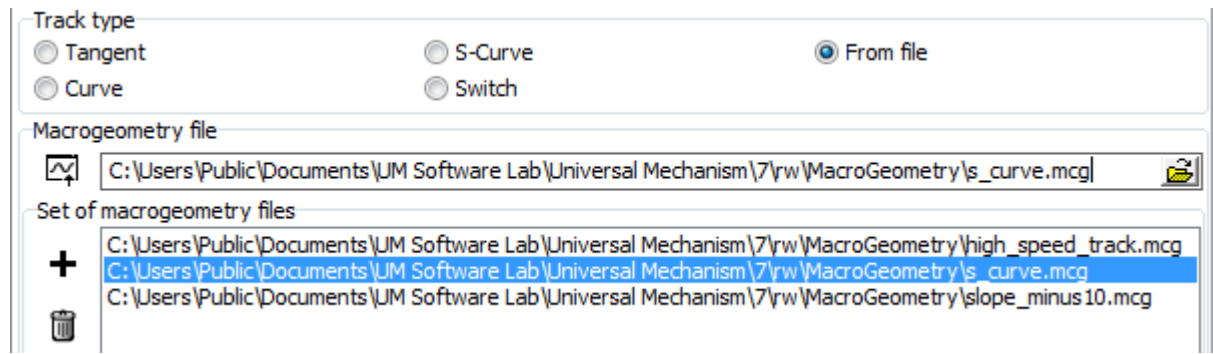



Figure 1.183. Setting track geometry from file

### 1.4.2.4.3. Assignment of track irregularities

The **Rail/Wheel | Track | Irregularities** tab is used for assignment of rail irregularities. The irregularities can be assigned in two forms: from files or deterministic, the **Type of irregularities** group, Figure 1.184. If the **Even** track state is activated, simulation runs in an ideal track without irregularities.

#### 1.4.2.4.3.1. Irregularities from file. Group of irregularities

Database of UM includes a number of files with irregularities located in the {Path to UM}\bin\rw directory. See Sect. 1.4.1.2.1. "Creation of irregularity files", p. 1-113 for creation of files with irregularities. The **NoIrregularities.way** file is used if a rail has no irregularities or the user programs the irregularities in the control file.

To set both the vertical (Z) and the horizontal (Y) irregularities to the left and right rails use the corresponding  button (Figure 1.184).

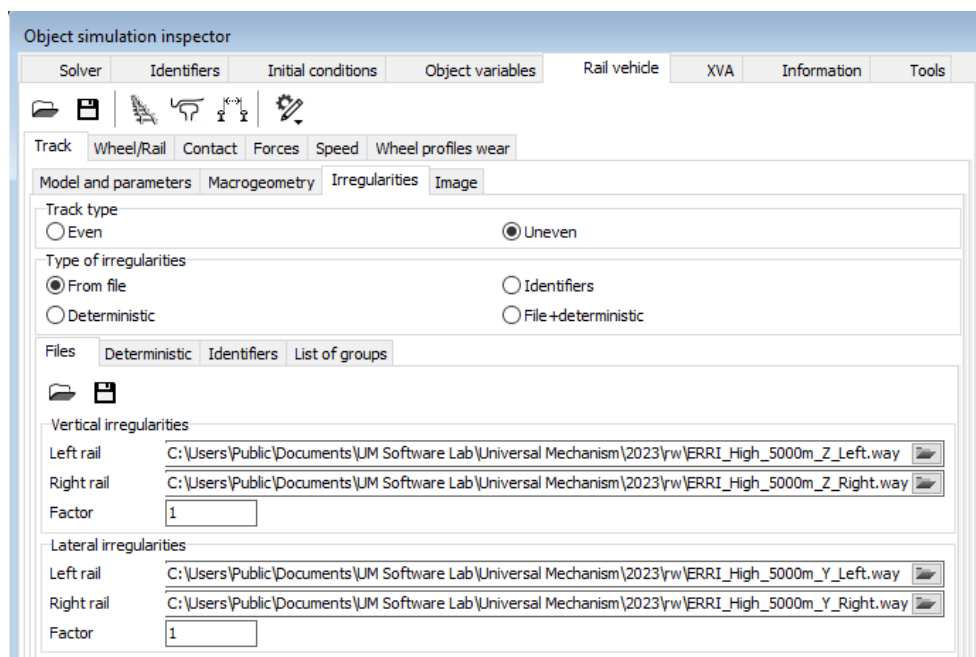




Figure 1.184. Assignment irregularities

The **Group of irregularities** section contains buttons, which are used for reading and writing groups of four files. For example, the user creates four files with irregularities assigns them to the track as above. After that the user saves names of files in a group file \*.tig by the  button, and can assign later the whole group by the  button without opening all of them separately.

**Example.** The file ERRI\_Low\_5000m.tig contains list of four files with irregularities corresponding to good condition according to ERRI, length of realization is 5000 m:

```
ERRI_Low_5000m_Z_Left.way  
ERRI_Low_5000m_Z_Right.way  
ERRI_Low_5000m_Y_Left.way  
ERRI_Low_5000m_Y_Right.way
```

The **Factor** parameter is used for increasing or decreasing the both assigned and computed in the control file irregularities. For example if the Y factor is 0.7, the decrease the horizontal irregularities to 30%.

#### 1.4.2.4.3.1.1. List of irregularity groups

List of irregularity groups is used for a quick assignment of irregularities by double clicking as well as for internal parameterization of the irregularities used in scanning projects, Sect. 1.6.1. "*List of internal identifiers parameterizing operation conditions of rail vehicles*", p. 1-244.

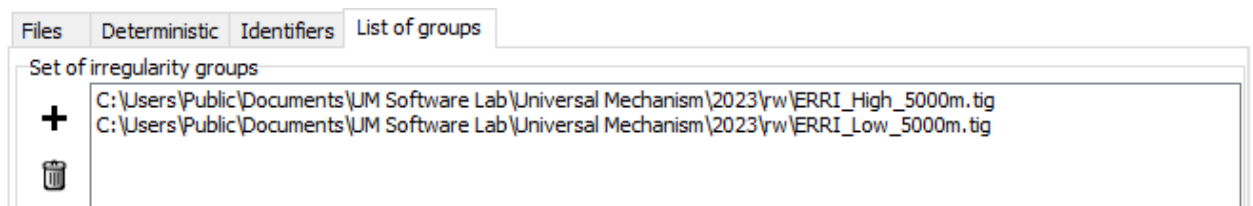


Figure 1.185. List of irregularity groups

### 1.4.2.4.3.2. Deterministic irregularities

Two types of vertical deterministic irregularities are available, Figure 1.186.

The screenshot shows a software window with three tabs: 'Files', 'Deterministic', and 'List of groups'. The 'Deterministic' tab is active. It contains two main sections:

- Vertical irregularities:**
  - Type of irregularities:  H/2(1-cos x) and  H|sin x|
  - Wave length (m): 20.00
  - Height (mm): 15.0
  - Start of left irregularity (m): 50.00
  - Shift of right irregularity (m): 0.00
  - Number of waves: 1
- Lateral irregularities H/2(1-cos x):**
  - Wave length (m): 20.00
  - Height (mm): 15.0
  - Beginning of irregularities (m): 50.00
  - Number of waves: 1

Figure 1.186. Parameters of deterministic irregularities

#### 1. Hump or deep

This type of irregularities is computed according to the formula

$$\Delta z = \frac{H}{2} \left( 1 - \cos \frac{2\pi x}{L} \right),$$

where  $H$  is the height of the wave (negative for a deep);  $L$  is the wave length,  $x$  is the longitudinal coordinate.

Along with the length and height of the wave, the following parameters should be set:

- start position of the left irregularity  $x_0$ ;
- shift of the right irregularity relative to the left one  $\Delta x$ ;
- number of waves  $N$ .

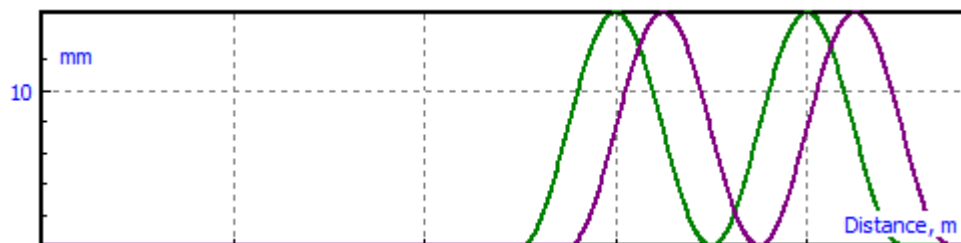


Figure 1.187. Example of deterministic vertical irregularities of "Hump" type for the following parameter values:  $L = 20 \text{ m}$ ,  $H = 15 \text{ mm}$ ,  $x_0 = 50 \text{ m}$ ,  $\Delta x = 5 \text{ m}$ ,  $N = 2$

#### 2. Absolute value of sinus

This type of irregularities is computed according to the formula

$$\Delta z = H \left| \sin \frac{\pi x}{L} \right|$$

The same parameters as they were for the "Hump" type irregularity are used.

Figure 1.188 shows irregularities with the same values of parameters as in Figure 1.187.

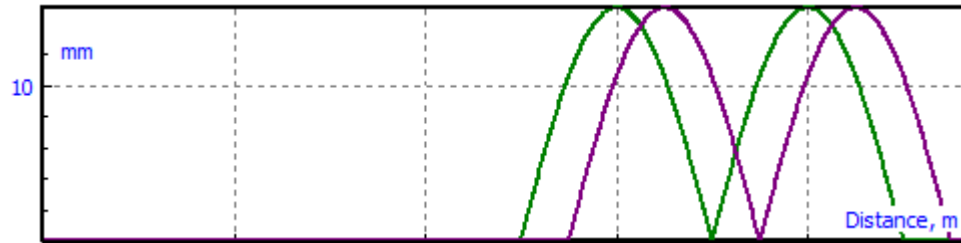


Figure 1.188. Example of irregularities 'absolute value of sinus'

Lateral deterministic irregularities are described by the same formula as the 'hump'. Phase between the left and right lateral irregularity is not supported.

### 1.4.2.4.3.3. Setting irregularities by identifiers

Horizontal and vertical irregularities can be set for each of the wheel independently with an 'Identifier control' tool and with the Matlab/Simulink interface. This method has some additional features compared to the irregularity files:

- Arbitrary and variable step size for irregularity functions
  - Rail vertical and horizontal motion as function of time.
- Remark 1. Irregularities are set in meters
  - Remark 2. When pointwise description of irregularity functions is used, the derivative of the curve is recommended to be smooth, which can be achieved by a spline interpolation.

Consider a workflow for setting irregularities by identifiers.

#### b) Adding standard identifiers for irregularities to the model

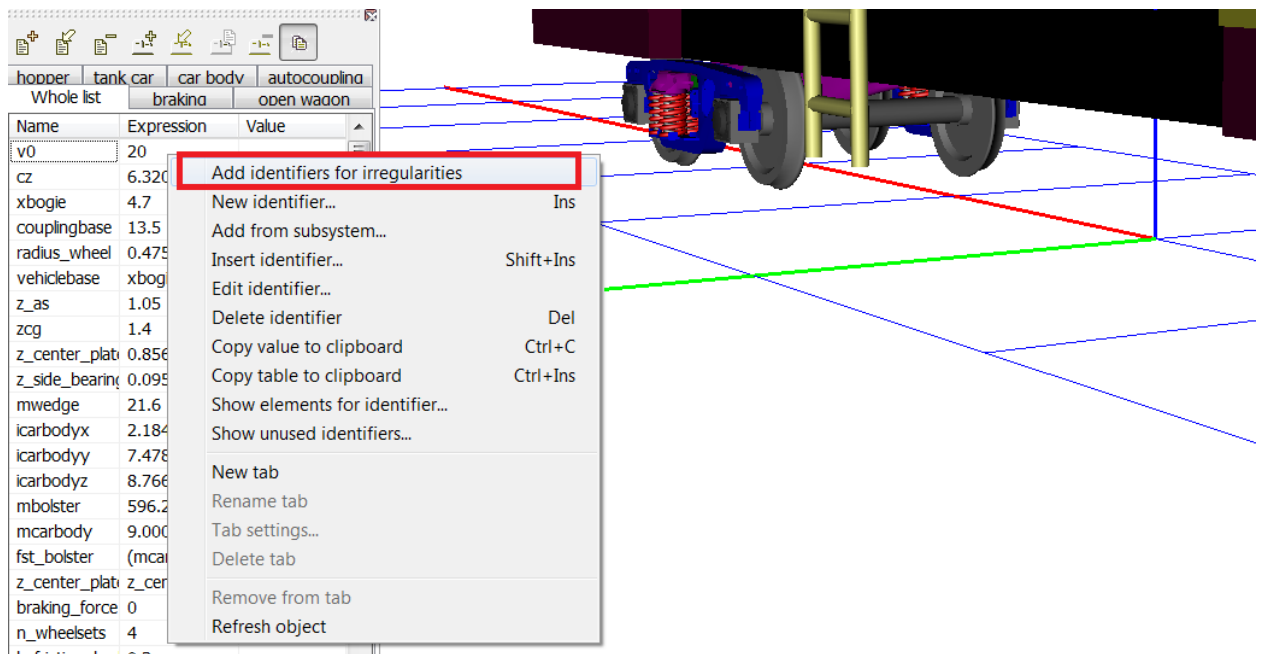


Figure 1.189. Adding identifiers for irregularities

Open a model in the Input program, click by the right mouse button on the list of identifiers and select the **Add identifiers for irregularities** command, Figure 1.189.

The number for adding identifiers is four times more than the number of wheelsets in the model, Figure 1.190. Syntax of the standard identifiers is as follows:

irr[type of irregularity]\_[side]\_[index of wheelset].

Name	Expression	Value
nrear_dg	1	
i_plateac	10	
i_plate	10	
irrZ_left_1	0	
irrY_left_1	0	
irrZ_right_1	0	
irrY_right_1	0	
irrZ_left_2	0	
irrY_left_2	0	
irrZ_right_2	0	
irrY_right_2	0	
irrZ_left_3	0	
irrY_left_3	0	
irrZ_right_3	0	
irrY_right_3	0	
irrZ_left_4	0	
irrY_left_4	0	
irrZ_right_4	0	
irrY_right_4	0	

Figure 1.190. Standard irregularity identifiers for a model with four wheelsets

**c) Selection of irregularity type in simulation program**

Track Profiles Contact Forces Speed Wear

Model and parameters Macrogeometry Irregularities Image

Track type  
 Even  **Uneven**

Type of irregularities  
 From file  Deterministic  **Identifiers**

Files Deterministic Identifiers List of groups

Name	Identifier	Value
Irregularity Z wset 1, left wheel	irrZ_left_1	0
Irregularity Y wset 1, left wheel	irrY_left_1	0
Irregularity Z wset 1, right wheel	irrZ_right_1	0
Irregularity Y wset 1, right wheel	irrY_right_1	0
Irregularity Z wset 2, left wheel	irrZ_left_2	0
Irregularity Y wset 2, left wheel	irrY_left_2	0
Irregularity Z wset 2, right wheel	irrZ_right_2	0
Irregularity Y wset 2, right wheel	irrY_right_2	0
Irregularity Z wset 3, left wheel	irrZ_left_3	0
Irregularity Y wset 3, left wheel	irrY_left_3	0

Figure 1.191. Setting type of irregularity

Select **Identifiers** type of irregularity description in the simulation inspector, Figure 1.191. If a user specifies identifiers, which names differ from the standard ones, he can assign them on this tab by double click of the mouse on the corresponding row and column of the table.

**d) Setting irregularity by points as a function of wheelset position**

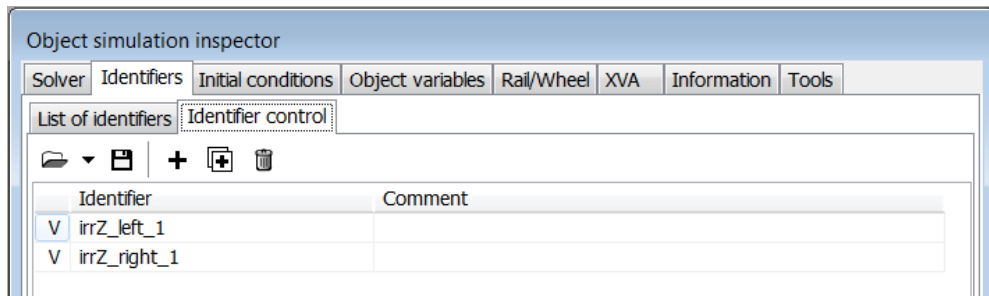


Figure 1.192. Identifier control tab

Consider first setting an irregularity as a function of wheelset position. Add a new control element on the **Identifiers|Identifier control** tab, Figure 1.192:

- Click on the **+** button to add a new identifier control;

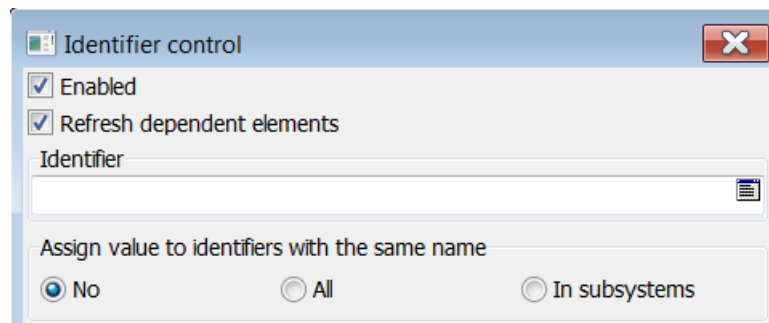



Figure 1.193. Window for control description

- In the appeared window, click on the  button to select an identifier.

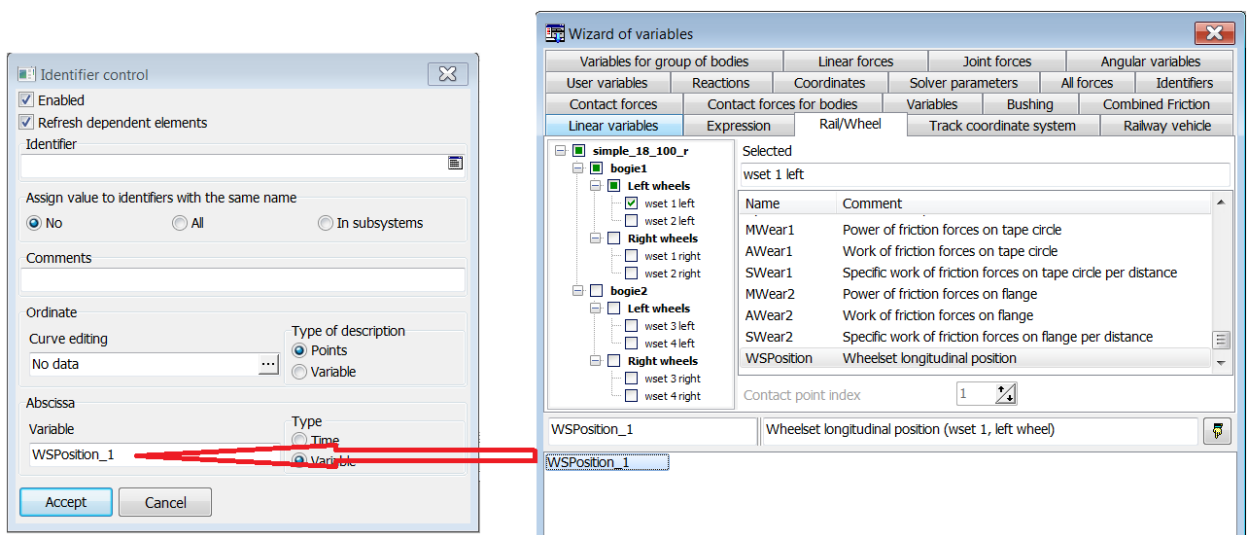


Figure 1.194. Setting wheelset position as an abscissa

- Set **Variable** type of abscissa; create a variable corresponding to the longitudinal position of the wheelset in track with the Wizard of variables and drag the variable to the corresponding box of the identifier control description, Figure 1.194.

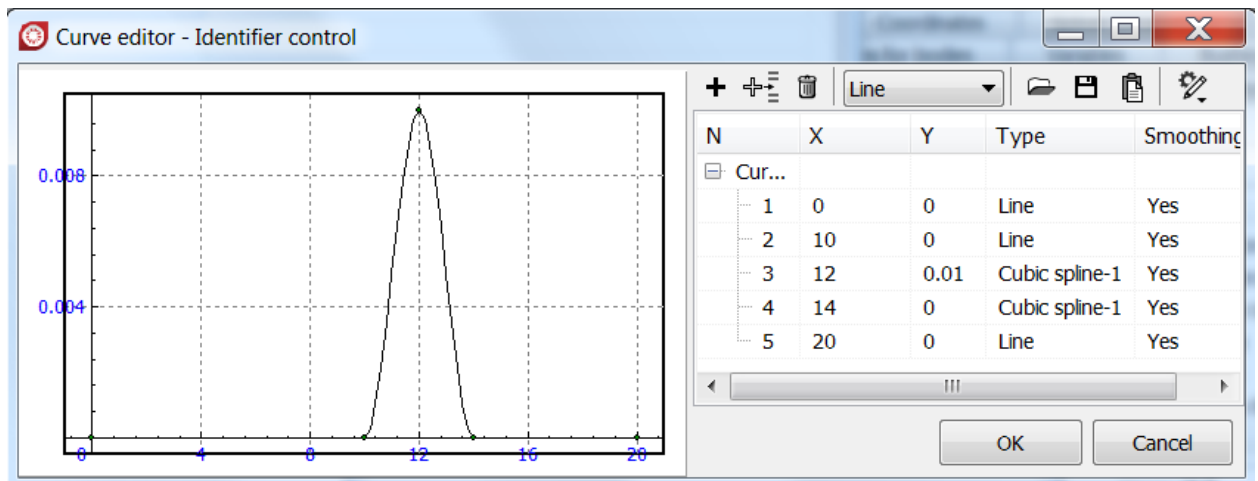



Figure 1.195. Example of irregularity function in the curve editor

- Use the  button to describe the irregularity function in the curve editor, Figure 1.195. The function can be described point by point or read from file.

**e) Position of rail under wheel as a function of time**

One of the purposes for setting a rail displacement under the wheel as a function of time consists in modeling of a test rig, when a kinematic excitation is assigned to wheelsets. In this case, the zero speed mode is recommended, Section 1.4.2.2.4  $v=0$ . Another purpose of this irregularity type consists in setting the rail displacement obtained from other sources such as field tests or from FEA programs.

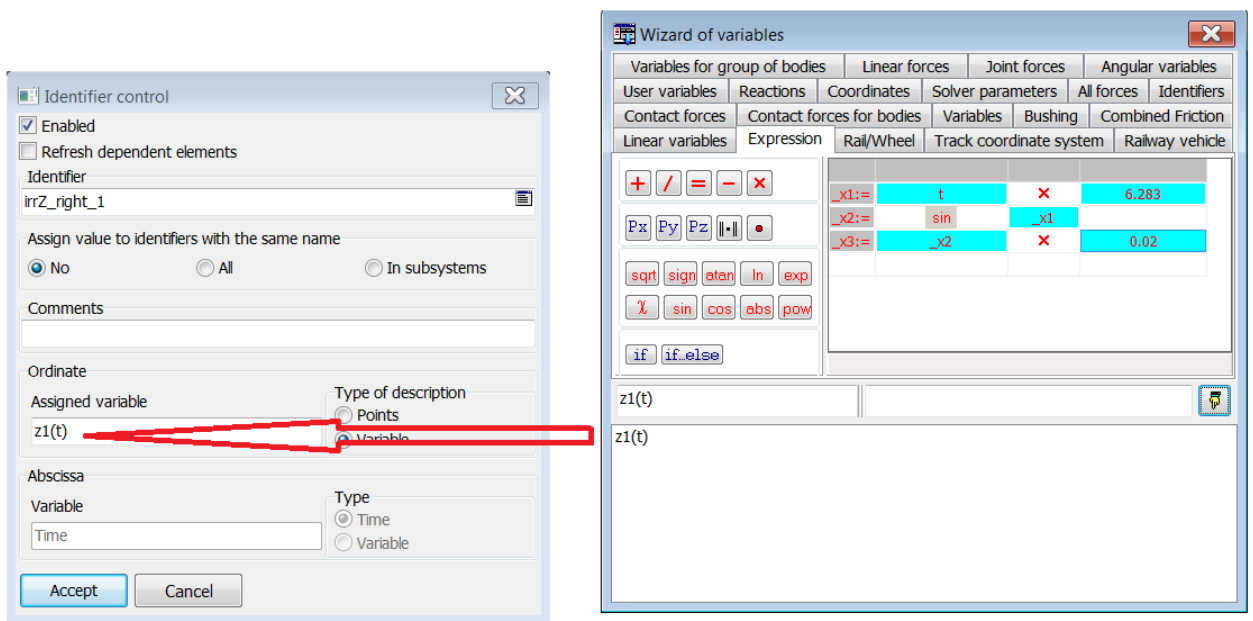


Figure 1.196. Harmonic rail oscillation

Rail displacement as a time function can be specified either pointwise with the curve editor or by a variable created with the Wizard of variables. Figure 1.196 shows assignment of the right rail oscillations according to the expression

$$z_1(t) = 0.02 \sin 2\pi t$$

#### 1.4.2.4.4. Track image in the animation window

Track image in the animation window is set on **Rail/Wheel | Track | Image** tab, Figure 1.197. If the **Show rails** check box is active, rails and sleepers are drawn in animation window. If **Show Irregularities** check box is active, rails are drawn taking into account irregularities. The rail is drawn along a polyline with a constant step size specified in **Rail-image step** box. Graphical image parameters of the sleepers are separated in **Sizes of sleepers** group.

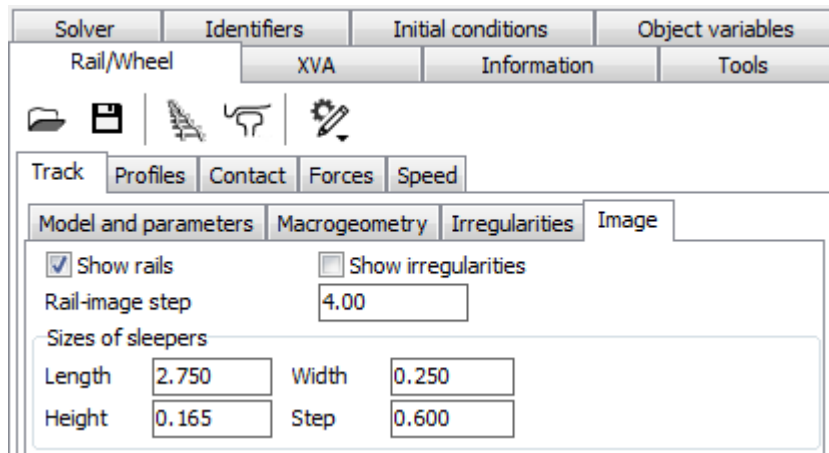


Figure 1.197. Track image in the animation window setting

### 1.4.2.5. Parameters for computation of rail-wheel contact forces

Contact and creep forces models can be chosen on the **Rail/wheel | Contact | Contact forces** tab. Different contact and creep forces models correspond to the different track models.

Parameter **Critical speed for creep** is used for every model of contact. The parameter allows the user to change the value of the critical speed  $v_0^*$ , Sect. 1.3.1.2.2. "*Algorithms for computing creep forces*", p. 1-87, Figure 1.199.

#### 1.4.2.5.1. Parameters of massless rail contact

A method for creep forces computation in wheel-rail contact when rail is a massless element is described in Sect. 1.3.1.2.1. "*Method for computation of rail deflections and contact force*", p. 1-85. The following creep forces models are available: *Mueller's*, *Minov's*, *FASTSIM*, *FASTSIM\_A*, see Figure 1.198.

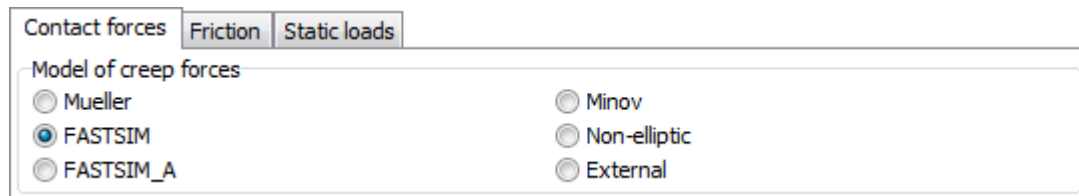


Figure 1.198. Creep forces models

- Mueller's model

The only parameter is **m**, the default value is 3, see Sect. 1.3.1.2.2.1. "*Mueller's model*", p. 1-88.

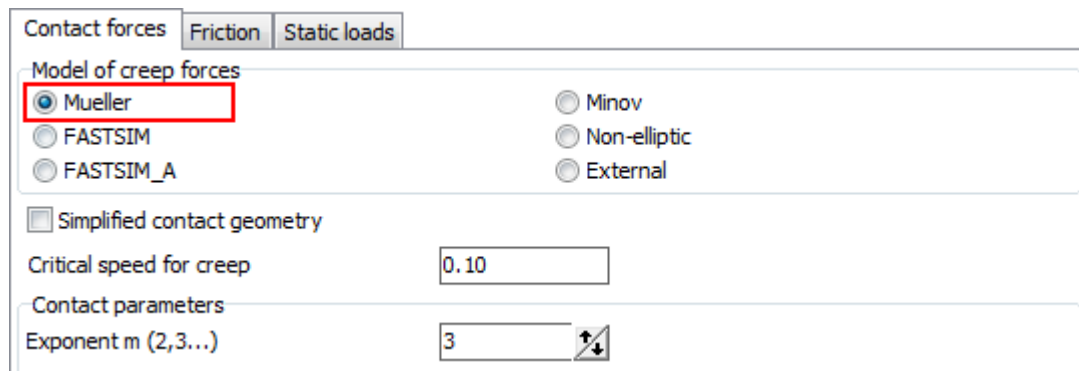


Figure 1.199. Choice of Mueller's model

- Minov's model

Computation of creep forces according to empirical analytic expressions. The model is used for simulation of locomotives in traction mode. See Sect. 1.3.1.2.2.2. "*Minov's model*", p. 1-89.

- FASTSIM, FASTSIM\_A

See Sect. 1.3.1.2.2.3. "*FASTSIM*", p. 1-90 and Sect. 1.3.1.2.2.4. "*FASTSIM\_A*", p. 1-91 for description of the models. The model parameters are: **Young's modulus** and **Poisson's ratio**, **Number of strips** and **Number of elements**, Figure 1.200.

Simplified contact geometry  
 Critical speed for creep: 0.10  
 Contact parameters:  
 Young's modulus: 210 000 000 000  
 Poisson's ratio: 0.27  
 Number of strips: 10  
 Number of elements: 20

Figure 1.200. FASTSIM and FASTSIM\_A model parameters

The simplified model of the pair of profiles is used when the flag **Simplified contact geometry** is activated, Figure 1.201, see Sect. 1.3.1.1.3. "Simplified contact geometry. Equivalent conicity and contact angle parameter", p. 1-82.

Simplified contact geometry  
 Critical speed for creep: 0.10  
 Profile parameters:  
 Equivalent conicity: 0.05  
 Equivalent contact angle parameter: 0.00  
 Nominal contact angle (degrees): 2.9  
 Maximal displacement (mm): 6.50  
 Flange contact angle (degrees): 70.00  
 Contact2 (Y,Z), mm: 35.0 -12.0

Figure 1.201. Parameters of the simplified model of wheel-rail pair of profiles

There are following parameters of the simplified contact model:

- **Equivalent conicity**  $\lambda$  (non-dimensional value);
- **Equivalent contact angle parameter**  $\varepsilon$  (non-dimensional value);
- **Nominal contact angle**  $\beta_0$  (degrees);
- **Maximal displacement** is a maximal lateral wheel-rail deflection before the  $y^*$  (mm) two-point contact starts;
- **Flange contact angle** is a flange contact angle with the rail lateral area in the  $\beta^*$  (degrees) two-point contact;

- **Contact2** are the coordinates of flange contact for the two-point contact in the  $y_r^*$  and  $z_r^*$  (mm) rail coordinate system.

**Note.** In the simplified contact model of wheel-rail pair of profiles Mueller's method is used for computation of creep forces.

### 1.4.2.5.2. Parameters of inertial rail contact

A method for computation of the wheel-inertial rail contact is described in Sect. 1.3.2. "Computation of contact between the wheel and the inertial rail", p. 1-91. For **Inertial rail** track model the contact forces models by W. Kik and J. Piotrowski and Kalker's CONTACT model are applicable, see Figure 1.202. **UM Loco/Multi-point Contact Model** tool is required for the model by W. Kik and J. Piotrowski. **UM Loco/CONTACT add-on interface** is a required tool for using CONTACT program. Note that the only interface for CONTACT is included in UM. Licenses for the CONTACT add-on for UM are provided and supported by Vtech CMCC, <https://www.cmcc.nl/software/>. You may request for trial license by e-mailing to [info@cmcc.nl](mailto:info@cmcc.nl). In your e-mail request please specify your name, position and a company/university. After you get the license copy the license file in UM set-up folder C:\Program Files\UM Software Lab\Universal Mechanism\10\bin (by default) and restart the UM.

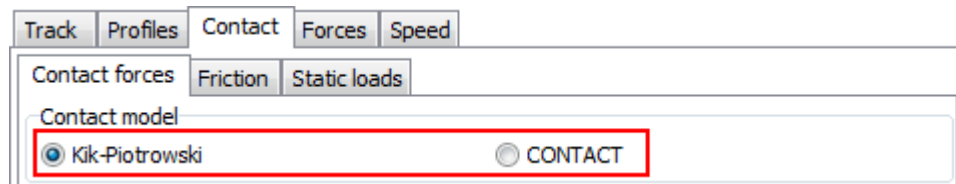


Figure 1.202. Choice of contact forces model for **Inertial rail** track model

The **Interpenetration factor** and **Damping ratio** are added in the model by W. Kik and J. Piotrowski, apart from the already considered parameters, see Figure 1.203.

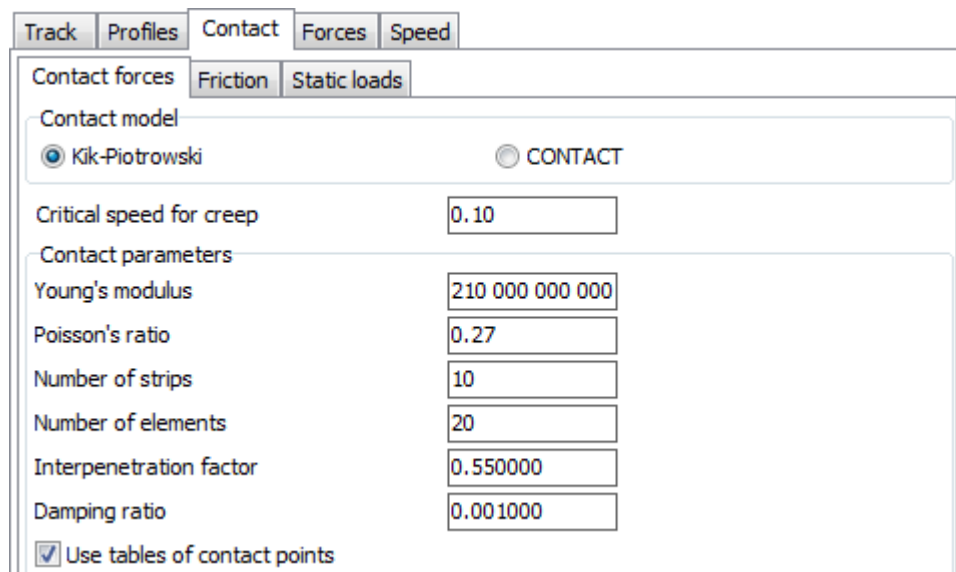


Figure 1.203. Parameters of contact forces model by W. Kik and J. Piotrowski

**Interpenetration factor** ( $k_\delta$ ) is a coefficient between the area of rigid interpenetration of profiles and the contact patch [4]. For example, if  $k_\delta = 1$ , the whole area of the rigid interpenetration will be accepted as the contact patch; if  $k_\delta = 0,5$ , then the area of the rigid interpenetration, obtained by profiles shifting on the penetration value decreased by a factor of 2, will be accepted as the contact patch. The recommended value of the interpenetration factor is 0,55.

**Damping ratio** ( $\zeta$ ) is included in the formula of normal reaction in the contact patch computing, Sect. 1.3.2. "Computation of contact between the wheel and the inertial rail", p. 1-91.

Tables of contact points when computing a contact are used when the flag **Use tables of contact points** is turned on.

**Note.** The usage of the contact forces model by W. Kik and J. Piotrowski or CONTACT model leads to the stiff equations of motion. For these models it is recommended to turn on the **Jacobian for wheel/rail forces** flag, see Sect. 1.4.4.1. "Solver: Park method", p. 1-229.

### 1.4.2.5.3. Parameters of flexible track

**Flexible track** model is supported only by the contact forces model by W. Kik and J. Piotrowski.

### 1.4.2.5.4. Setting coefficients of friction in rail/wheel contact

The **Rail/Wheel | Contact | Friction** tab is used for setting the coefficient of friction, Figure 1.204.

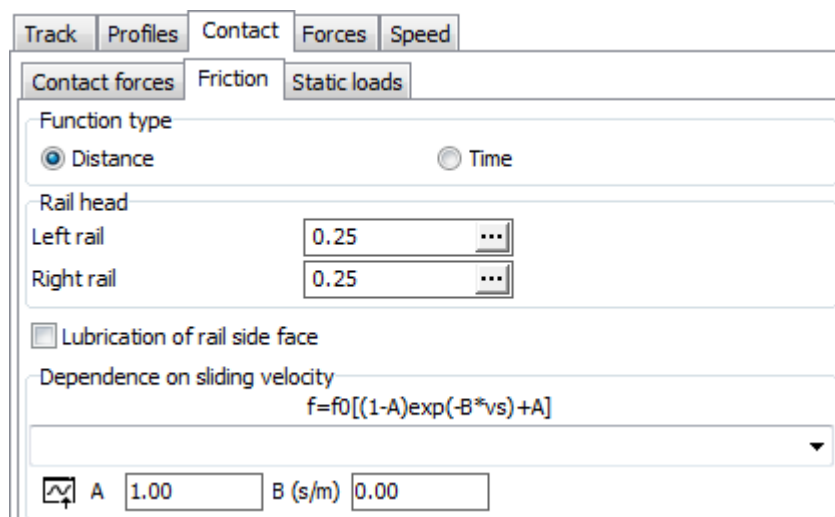


Figure 1.204. Setting of friction coefficients in the rail-wheel contact

See Sect. 1.3.3. "Coefficient of friction in wheel-rail contact", p. 1-95 for the detailed information about the coefficient of friction.

To set the constant friction coefficients on the running surface  $f_r$  of the left and right rails, the corresponding fields of the **Rail head** group are used, Figure 1.204. If a coefficient of friction is variable, choose the function type first (either **Distance** or **Time**) and then click the **...** button.

The curve editor to set the curve dependence of the friction coefficient on the chosen parameter will appear.

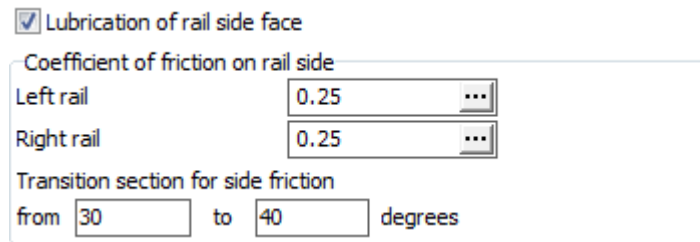


Figure 1.205. Setting of friction coefficients on rail side

**Rail side face lubrication**

If lubrication of the rail side face is modeled, activate the **Lubrication of rail side face** key, (Figure 1.205), and set the value of  $f_s$  friction coefficient on rail side. The angles  $\beta_r$  and  $\beta_s$ , which define the transition section should be set in the **from** and **to** edit boxes.

**Dependence of the friction coefficient on sliding velocity**

Parameters A, B in Figure 1.206 are included in dependence of the friction coefficient on sliding velocity, Sect. 1.3.3. "Coefficient of friction in wheel-rail contact", p. 1-95:

$$f = f_0((1 - A)e^{-Bv_1} + A).$$

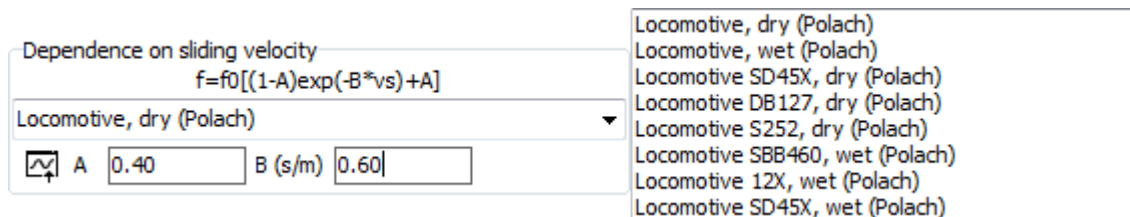


Figure 1.206. Dependence of the friction coefficient on sliding velocity. Dependence models.

If  $A = 1$ ,  $B = 0$ , the coefficient of friction is constant. Parameter  $A$  is the ratio of coefficient of friction for infinite sliding velocity to that for zero sliding. For example,  $A = 0.4$  corresponds to the case when the coefficient of friction decreases in 2.5 times for large sliding. Parameter  $B$  determines the rate of decrease of the coefficient, Figure 1.207.

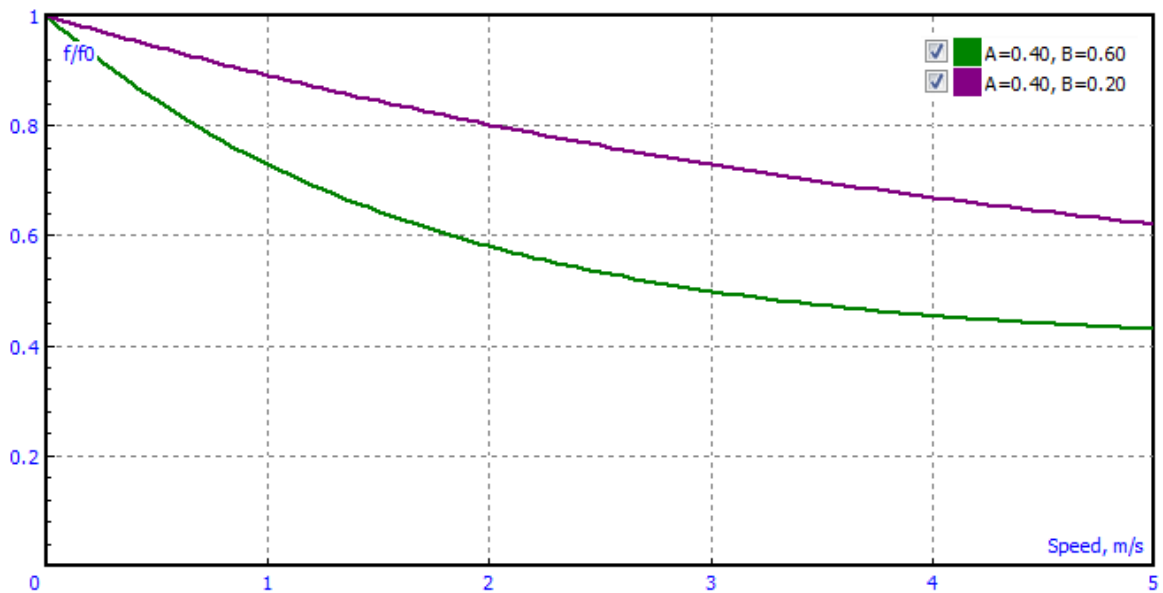


Figure 1.207. Examples of decrease of coefficient of friction with growth of sliding velocity

Drop down menu in Figure 1.206 allows the user to choose the values of parameters A, B from the database [6], Table 1.3.

Table 1.3

**Parameters of the dependence of friction coefficient on sliding velocity**


Locomotive	State of rail	A	B (s/m)
Typical	Dry	0.4	0.6
Typical	Wet	0.4	0.2
SD45X	Dry	0.44	0.6
DB127	Dry	0.38	0.7
S252	Dry	0.36	0.55
SBB460	Wet	0.5	0.16
12X	Wet	0.4	0.4
SD45X	Wet	0.38	0.18

### 1.4.2.6. Setting in-train forces

Use the **Rail/Wheel | Forces** tab to specify in-train forces acting on the vehicles (Figure 1.208).

#### 1.4.2.6.1. Assigning forces

Forces can be obtained from experiments or from **UM Train** module (the module for analysis of longitudinal train dynamics). Both front and rear forces can be applied to a vehicle or to a group of vehicles. Each of the forces is specified by the following sets of data.

- Body, which the force is applied to, as a rule is a car body.
- Force application point. Coordinates X and Y set *in the body-fixed SC*, whereas the Z coordinate is measured *from the rail head level*.
- Files with force component descriptions. The force is described by three components in the body-fixed SC. To set a component, a preliminary created file should be selected by the  buttons, Figure 1.208. File format description can be found in the next section.
- Use the **On** key to switch on/off the forces.
- If the **Function of distance** key is on, the forces are functions of the vehicle travel along the track otherwise they are functions of time.
- The **Change sign** key is used if it is necessary to change the sign of the front force to the opposite one. For instance, if the force is obtained from simulation of longitudinal train dynamics with **UM Train**, this key must be activated because the in-train forces in the coupling in **UM Train** are considered to be applied to the front vehicle in a coupling.

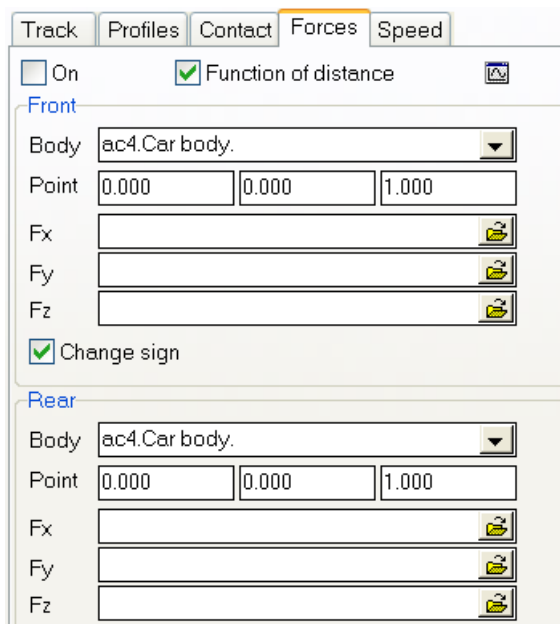


Figure 1.208. Window for specifying in-train forces

If files do not assign to some components of forces, they are zeroes.

The  button is used for plotting the assigned functions.

Two methods are used to delete the file assignment.

- Locate the text cursor in the box with the name of file and press Delete;

- Use the popup menu of the text box.

#### 1.4.2.6.2. Creating files with force description

Forces in Sect. 1.4.2.6.1. "*Assigning forces*", p. 1-173 must be preliminarily stored in files, which contain forces versus time or distance histories in projections on the car body system of coordinates. Units are N, s, m.

A linear interpolation is used to get a continuous dependence.

Two different formats are used for description of the force.

- **Format \*.frc, obsolete**

A binary force file contains successive force values with step size 0.1m (0.1s) starting from zero distance (time). Single binary format is used for force values (4 byte floating point numbers).

It is recommended to use the window for development of track irregularities, which uses the format of files \*.way (Sect. 1.4.1.2.1. "*Creation of irregularity files*", p. 1-113). After creating the file its extension should be renamed to *frc* instead of *way*. It is recommended to store files in the model directory or in the **../rw** directory.

- **Format \*.txt**

In this case the text file contains two columns. The first column is the time or distance, the second one is the force. Example:

0	0
50	0
70	30000
150	30000
170	0
180	0

#### 1.4.2.6.3. Plotting and visualization of forces

The variables for creation of forces plots in the railway vehicle simulation process with the aim of controlling the task correctness are available on the tab **Railway vehicle** of **Wizard of variables** (See 1.4.3.1.3. "*Variables related to the railway vehicle* ", p. 1-192).

To visualize the force vectors in animation window, move the mouse cursor to the car body, click the right mouse button, and select the **Show forces for [Name of Body]** popup menu command Figure 1.209. If the forces are invisible during the simulation, either the force application coordinates are not correct, or scale factor for forces is too large. In the last case use the **Scale of vectors** command of the popup menu to decrease the scale factor, Figure 1.209.

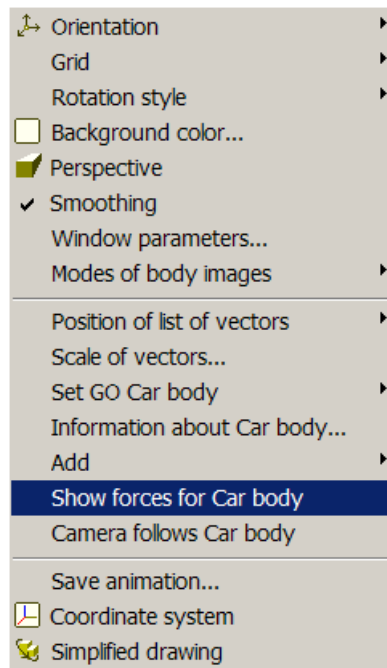


Figure 1.209. Visualization of forces for Car Body in animation window

#### 1.4.2.6.4. Example of in-train forces

Consider lateral forces applied to the front and rear couplers, Figure 1.210. The forces are zero on the first 50 m of the vehicle travel, and then they grow uniformly to 3000 N on the interval from 50 m to 70 m. The forces are constant till 150 m, and decreases uniformly on the 20 m interval. The front force is positive, the rear one is negative.

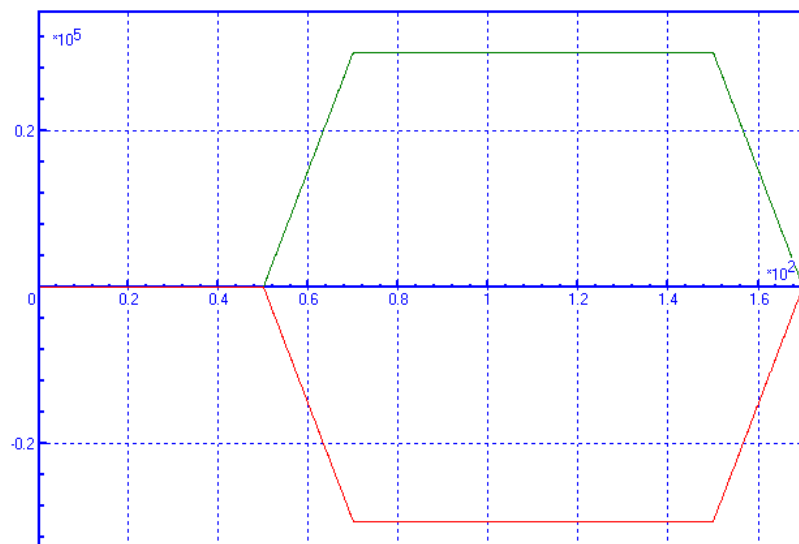



Figure 1.210. Lateral forces versus distance

Open the window for development of files with irregularities by the **Tools | Create irregularities| Railway track** menu command, open the **Points** tab and start the curve editor by the  button.

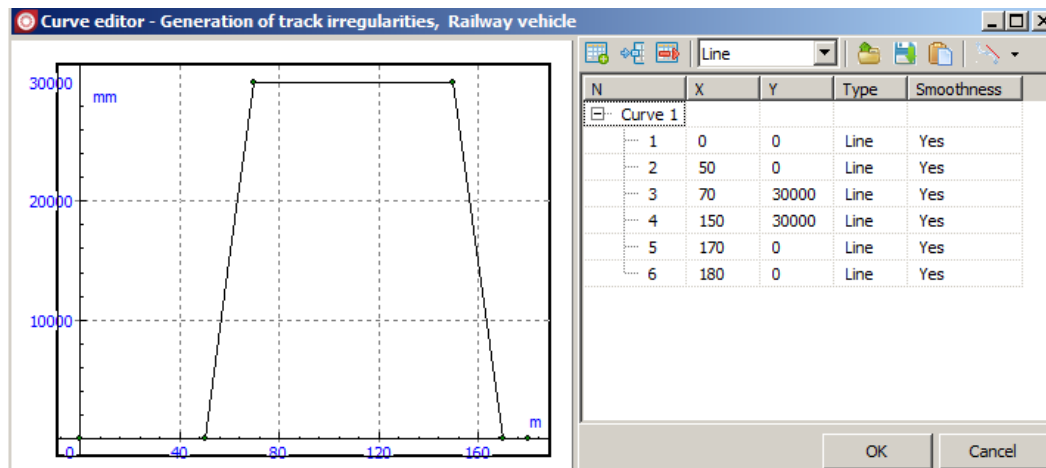


Figure 1.211. Force description by points

Add 6 points by the button, and set their coordinates as in Figure 1.211. Optimize the polygon view by then button.

Close the editor by the **OK** button and save the result in a file by the lower button . Rename the file extension manually to *frc* instead of *way*.

Let us consider an alternative method for adding points, which is highly useful if the force description includes many points. The polygon should be presented by two columns in any text editor. The first column corresponds to the abscissa (distance or time), the second one contains the force values.

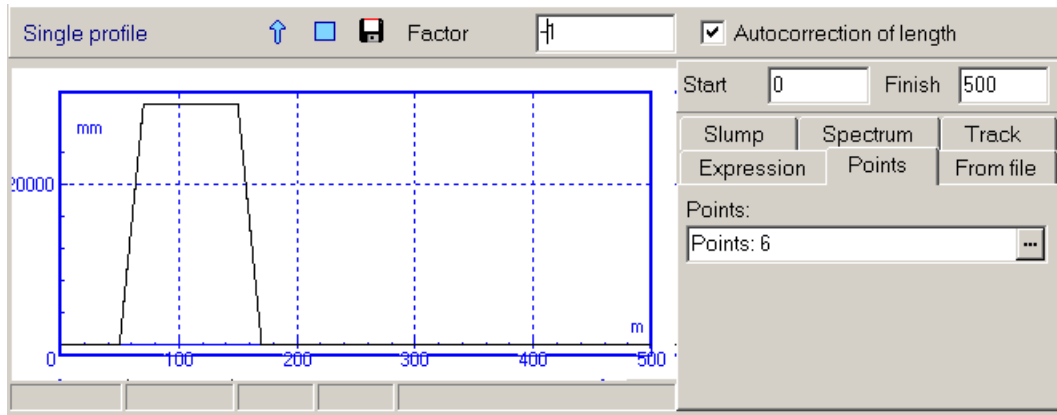
0	0
50	0
70	30000
150	30000
170	0
180	0

Copy data into the clipboard by Ctrl+C and paste into the curve editor Figure 1.211 by Ctrl+V. In this manner large data can be converted into the necessary format in the case when forces are obtained from experiment or from simulation of longitudinal train dynamics with **UM Train**. In the last case the corresponding variable should be saved in a text file from a graphic window.

To create a file with the force having the reverse sign, set -1 (minus unit) in the **Factor** box, send the plot to the upper part of the window by the button, save it by button, and rename the extension.


The created files can be immediately assigned to the force components, Figure 1.208, and used in simulation process.

**Remark.** It is recommended to use an alternative format of force description in \*.txt files, Sect. 1.4.2.6.2. "Creating files with force description", p. 1-174.



### 1.4.2.7. Additional parameters

#### 1.4.2.7.1. Deviation of wheel form from ideal circle

**Rail/Wheel | Profiles | Wheel forms** tab is used for specifying deviations of wheel forms from an ideal circle. A direct access to this tab is available with the  button by the **Form of wheels** command, Figure 1.212. The list of wheelset is located in the left part of the tab. Icon image for a wheelset depends of the form type. For instance, in Figure 1.212 it has a rail vehicle with two wheelsets. A flat is assigned to the first WS, the second one has the form of an ideal circle.

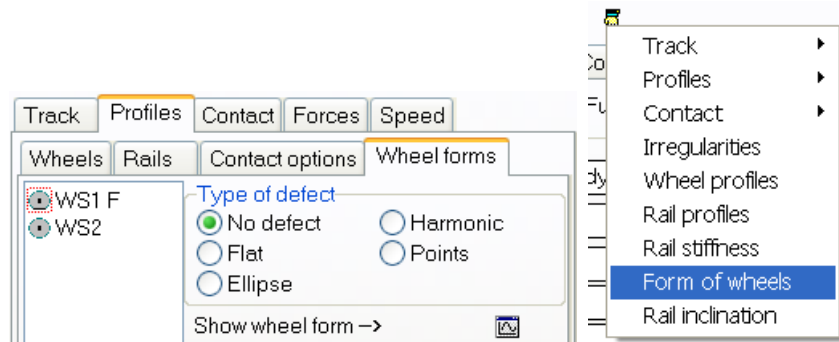


Figure 1.212. Setting wheel form

Click a WS icon in the list to get access to the form of its wheels.

Form of the wheels influences dynamically the normal force in contact as wheel as the longitudinal position of the contact.

##### 1.4.2.7.1.1. Flat

A flat is a defect of the wheel shape in terms appears of the flat spot on the running surface, Figure 1.213.

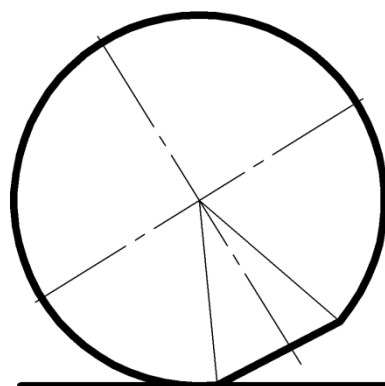


Figure 1.213. Wheel flat

There are two types of wheel flat geometries:

1. The newly formed wheel flat with sharp edges as occurring right after formation.
2. The rounded wheel flat, which rapidly develops from the newly formed flat as a result of wheel tread wear and plastic deformation.

The shape of the flat is specified in the **Type** group, Figure 1.214.

Type of defect

No defect
  Flat
  Ellipse
  Harmonic
  Pointwise

Show wheel form -->

Phase (degrees)

Flat parameters

Type

Newly formed
  Rounded

d (mm)

L0 (mm)

L1 (mm)

Lateral position (mm)

Flat cross-section

Figure 1.214. Parameters of wheel flat

The following parameters specify the flat geometry.

**d** is the depth of the flat. The length of the newly formed flat is calculated as the chord length:  $L_0 = \sqrt{8Rd - d^2}$ , Figure 1.215.

**L1** is the length of the rounded flat, Figure 1.215. The shape of the rounded flat in the longitudinal direction is approximated by the function [9].

$$z_{rf}(x) = \frac{d}{2} \left[ 1 + \cos \left( 2 \frac{\pi x}{L_1} \right) \right].$$

The minimum value of **L1** that guarantees the convex form of the wheel is  $\pi L_0/2$ .

**Lateral position** is the lateral distance between the origins of the SC of flat cross-section and the SCW.

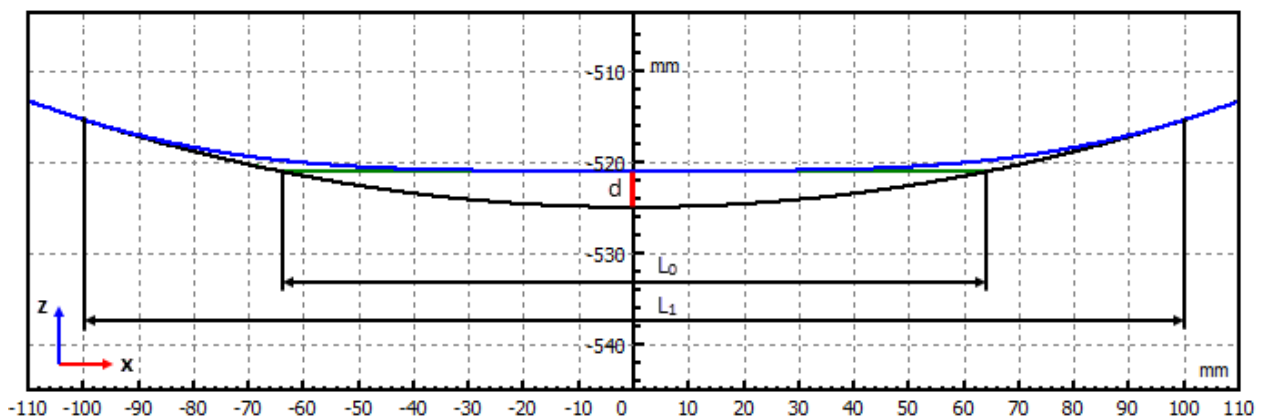


Figure 1.215. Geometry of newly formed and rounded wheel flats in the longitudinal direction

**Flat cross-section** is the section of the flat by yOz plane. The flat cross-section corresponds to the shape of the rail head (rail inclination take into account) on which it was formed, Figure 1.216.

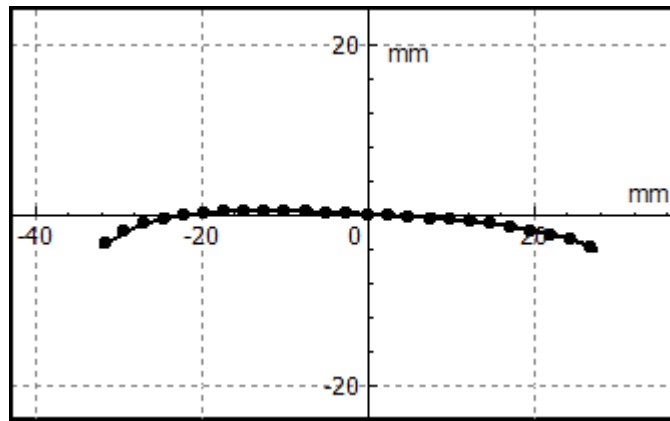


Figure 1.216. Flat cross-section

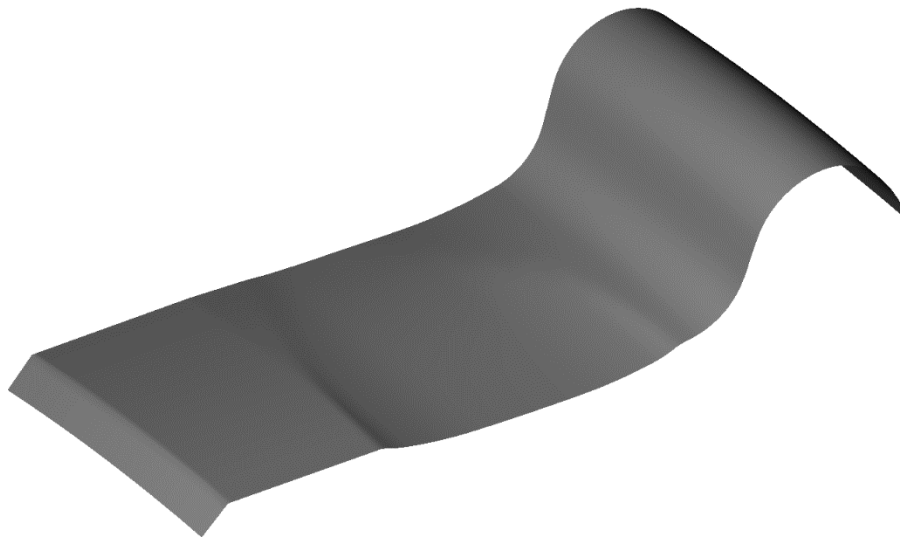


Figure 1.217. 3D geometry of wheel flat

**Note** 3D model of the wheel flat is supported only in Kik-Piotrowski or CONTACT contact models.

**1.4.2.7.1.2. Ellipse**

The form of an elliptic wheel is computed according to the formula

$$\frac{x^2}{(R + dA)^2} + \frac{y^2}{(R + dB)^2} = 1,$$

where parameters *dA* and *dB* set deviation of ellipse semiaxes from the ideal wheel radius *R*, which is specified for the given wheelset in the **UM Input** program, Sect. 1.1.3.6. "Editing wheelset parameters", p. 1-15, Figure 1.219.

Type of defect

No defect
  Harmonic

Flat
  Pointwise

Ellipse

Show wheel form -->

Phase (degrees)

Ellipse parameters

dA (mm)  dB (mm)

Figure 1.218. Parameters of elliptic wheel

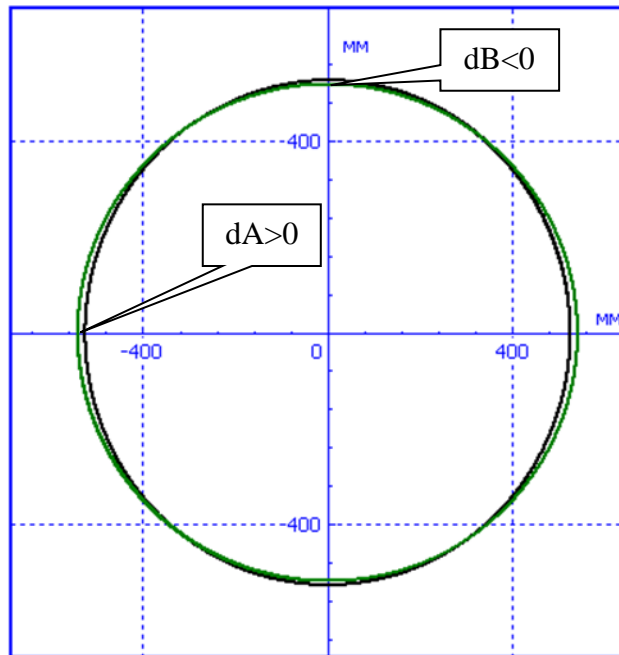


Figure 1.219. Example of elliptic wheel

**1.4.2.7.1.3. Harmonic deviation**

The form of the wheel is computed according to the formulas

$$\begin{aligned}
 x &= (R + A \sin N\varphi) \sin \varphi \\
 y &= (R + A \sin N\varphi) \cos \varphi \\
 \varphi &\in [0, 2\pi]
 \end{aligned}$$

where deviation of the variable radius from the constant value R is set by the harmonic function the amplitude A, Figure 1.220. The number of humps on the curve is equal to N, Figure 1.221.

Type of defect

No defect
  Harmonic
  Pointwise

Flat

Ellipse

Show wheel form -->

Phase (degrees)

Harmonic parameters

A (mm)  N

Figure 1.220. Parameters of harmonic deviation of wheel from circle

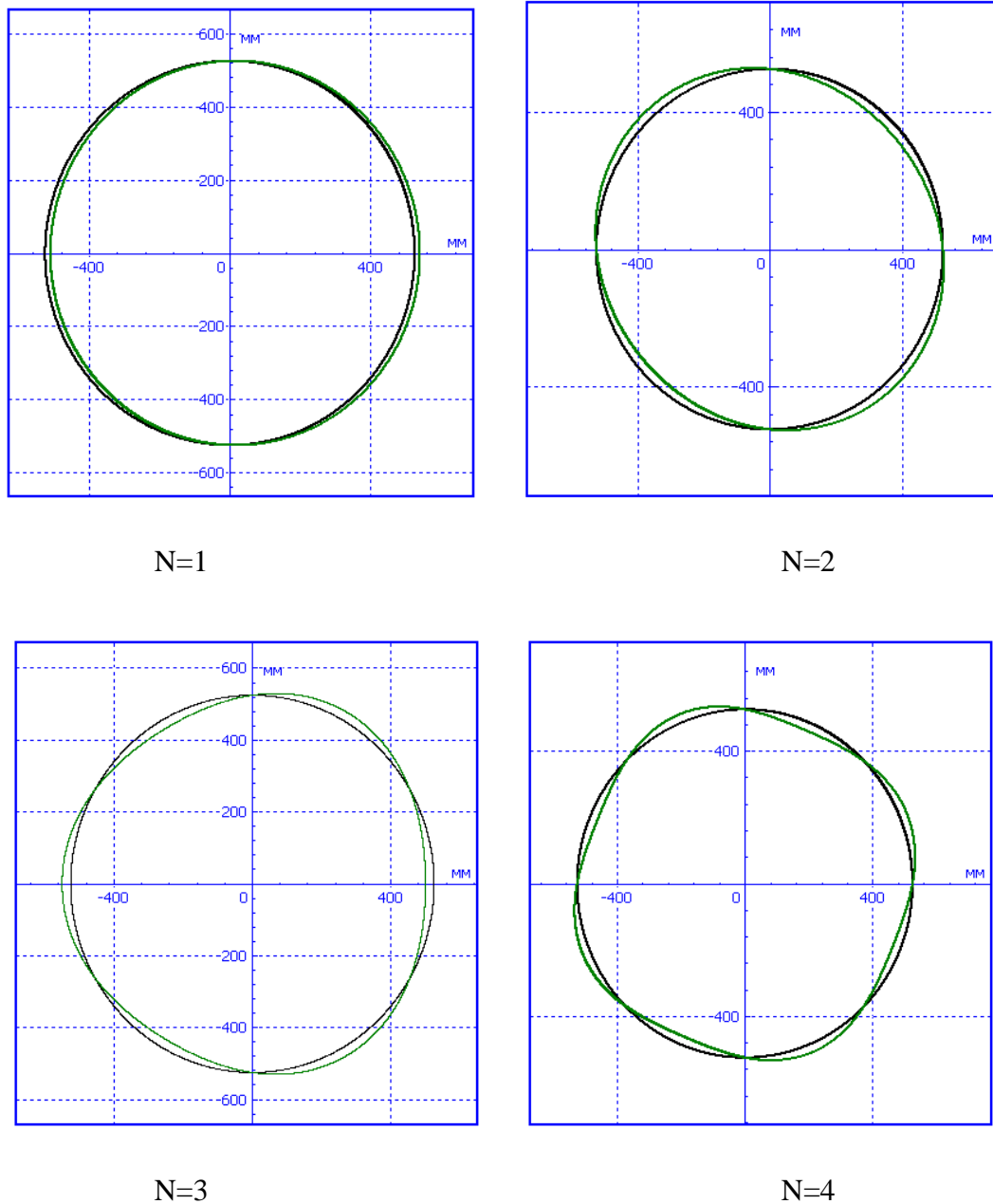


Figure 1.221. Examples of wheels with different number of humps

**1.4.2.7.1.4. Points**

The form of the wheel is computed according to the formulas

$$x = (R + dr(\varphi)) \sin \varphi$$

$$y = (R + dr(\varphi)) \cos \varphi$$

$$\varphi \in [0, 2\pi]$$

where deviation  $dr$  of the variable radius from the constant value  $R$  is set by a curve, that must be specified by the user in a text file, Figure 1.223. The corresponding form of the wheel is shown in Figure 1.224.

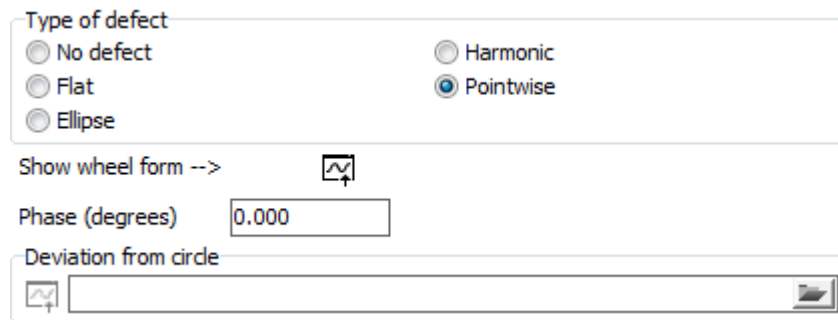


Figure 1.222. Setting wheel form by points

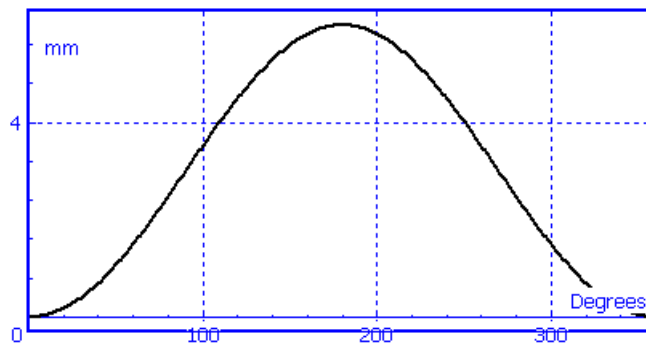


Figure 1.223. Deviation of radius vs. angle

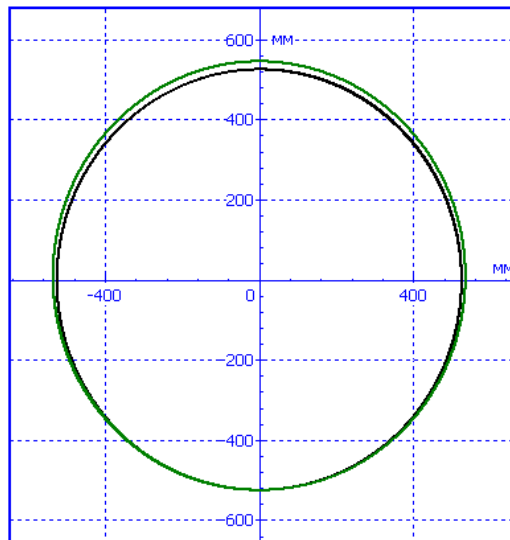


Figure 1.224. Example of setting form of wheel by points


The text file contains two columns. The first column is the angle in degrees; the second one is the deviation of radius in mm. Example:

```

0 0
1 0.000456917
2 0.001827527
3 0.004111415
4 0.007307883
5 0.011415959
6 0.016434391
7 0.02236165
8 0.02919593
9 0.036935151
10 0.045576954
11 0.055118707
12 0.065557503

```

.....

The button  is used for assignment of a file. Continuous function of  $dr$  vs.  $\phi$  is obtained by use of the Lagrangian interpolation polynomial of the 3<sup>rd</sup> order.

#### 1.4.2.7.2. Wheel radii difference

Defect of wheel radii is the difference between the real radius of running circle, and the radii of wheels specified in the **UM Input** program, Sect. 1.1.3.6. *"Editing wheelset parameters"*, p. 1-15. This difference is specified in **Rail/Wheel | Profiles | Wheels | Radii difference** tab in millimeters. In Figure 1.225a, the radius of the left wheel is 2 mm less than that for the right wheel. In Figure 1.225b, the wheels of the first WS are 3 mm less than that for the second wheel.

Radii differences have internal parameterization for scanning projects, Sect. 1.6.1. *"List of internal identifiers parameterizing operation conditions of rail vehicles"*, p. 1-244.

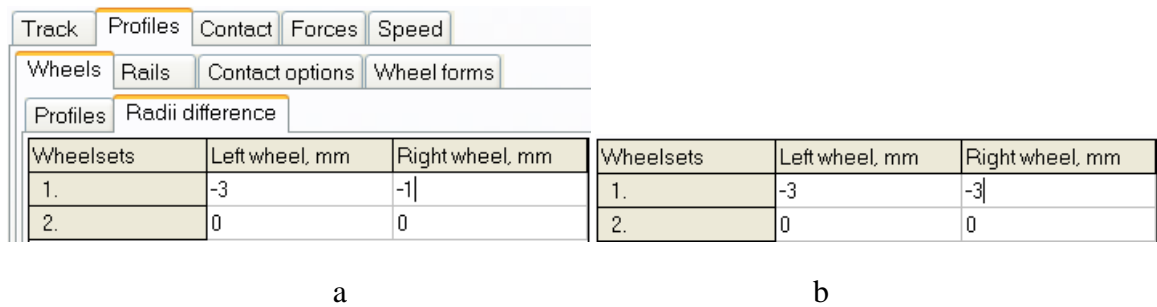


Figure 1.225. Setting difference in radii

### 1.4.2.7.3. Rail/wheel contact options

If worn profiles defined by a set of points are created or if profile evolution is computed, some problems with the contact geometry computations might appear. Let us consider some of these problems as well as their solving with the help of parameters in **Rail/Wheel | Profiles | Contact options** tab.

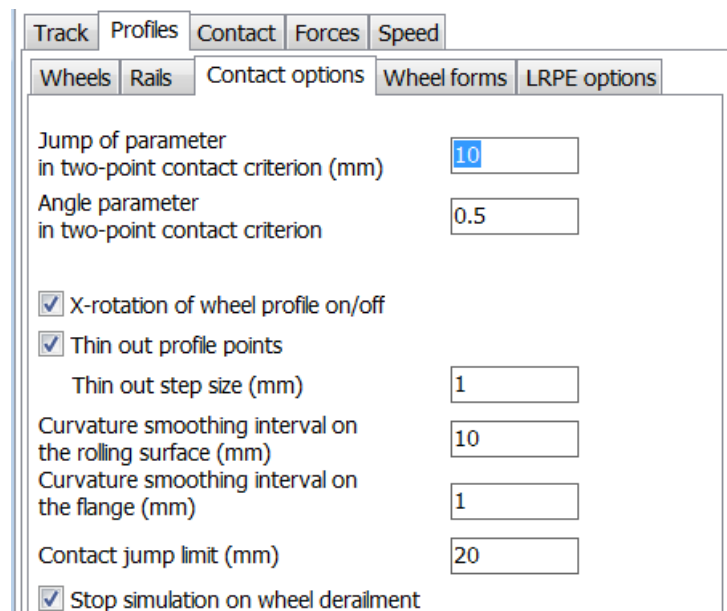


Figure 1.226. Additional parameters for contact geometry computations

- **Jump of parameter and angle parameter in two-point contact criterion**

These parameters are used in the procedure of computing contact coordinates for verifying the fact of the two-point contact (see the  $\eta_y, \eta_\eta$  parameters in Sect. 1.3.1.1.2. "Computing tables of contact points", p. 1-78). Sometimes variation of this parameter can improve the detection of two-point contact.

- **Key X-rotation of wheel profile on/off**

Sometimes worn profiles of wheel and rail are in a very close contact near the flange region so that a small relative rotations  $\Delta\alpha$  of the profiles about the longitudinal X-axis may cause a big jump in coordinates of contact points or transfer from profiles with one-point contact only to the profiles, which allow a two-point contact and back. Such cases lead usually to large jumps in

values of contact forces. To stabilize the simulation, switching off the variation of the contact positions in dependence on the  $\Delta\alpha$  angle is useful very often (see Sect. *Computing tables of contact points*). Switching off the key means that the coordinates for  $\Delta\alpha = 0$  are used.

- **Key Thin out profile points**

A very small abscissa step size in defining profiles leads to violation of continuity of contact point positions in dependence on lateral displacements and relative rotations of profiles (see Sect. 1.3.1.1.2. "*Computing tables of contact points*", p. 1-78 and Sect. 1.4.1.1. "*Creation of wheel and rail profiles*", p. 1-97). If the key is checked, the step size is decreased up to the value, which is set as the *Thin out step size* parameter. The default value of the step is 1mm.

- **Contact jump limit**

The table of contact coordinates versus relative lateral shift of profiles has a discrete nature. If a difference between the neighbor coordinate values exceed this parameter, a jump or discontinuity of the curve take place. Otherwise, a linear interpolation within the interval is used.

- **Stop simulation on wheel derailment**

If the option is checked, the simulation is broken when the lateral displacement of a wheelset exceeds some definite value. This value is computed automatically according to the rail and wheel profiles. Usually it is about 100mm.

### 1.4.3. Tools for visualization and analysis of railway vehicle dynamics

#### 1.4.3.1. Some features of creation of variables

General information about creation and usage of variables as well as lists of variables can be found in [Chapter 4](#), Sect. *Variables, Wizard of variables, List of variables*. Here we consider variables, which are related to rail vehicles exclusively.

##### 1.4.3.1.1. Rail/wheel contact variables

For variables which describe wheel-to-rail interaction use the **Rail/Wheel** tab of the **Wizard of variables**, Figure 1.227. See [Chapter 4](#), Sect. *Wizard of variables* for detailed information about the **Wizard of variables**.

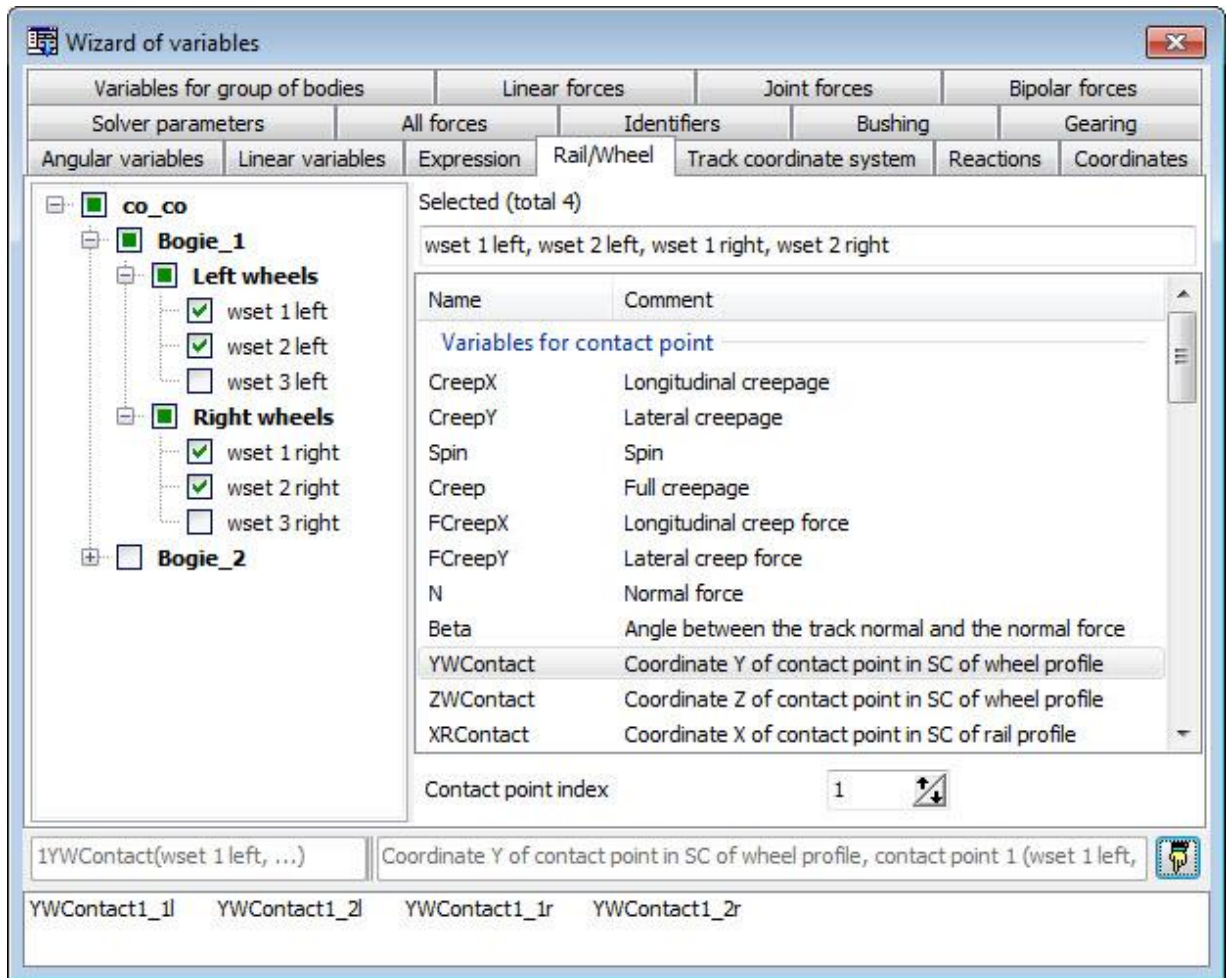


Figure 1.227. Variables for rail/wheel contacts

**Note.** The wheelsets are numbered 1, 2... in compliance with the decreasing of their longitudinal coordinate.

The variables on **Rail/Wheel tab** are formed in two groups: **Variables for contact point** and **Variables for wheel**. The first group contains the variables which describe wheel-to-rail contact points. It should be noted that the *Simplified* and *two-point* contact forces models assume two contact points as maximum. In this case the first contact point corresponds to the one-point contact or a contact on a wheel rolling surface (Figure 1.228), and the second contact point corresponds to a contact on the ledge of profiles, allowing a two-point contact (Figure 1.228). *Multi-point* and *CONTACT* models allow an unlimited number of contact points. The points are numbered in order correspondent its lateral position on the rail, from outer side to inner side of the railway track (Figure 1.228). The variables in this group are described in Table 1.4. The second group contains variables for the wheel which are not directly connected with any of the contact points. The description of these variables is given in Table 1.5.

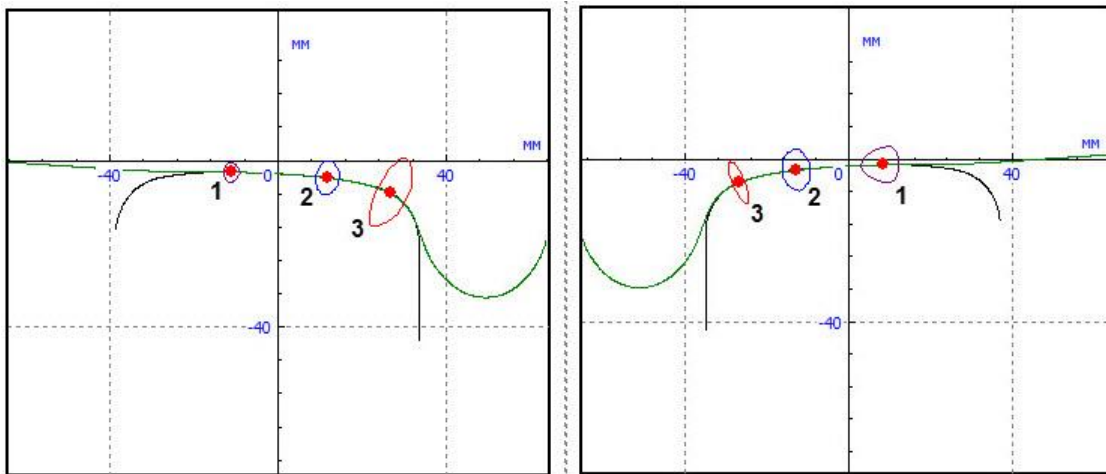



Figure 1.228. Contact points numeration

To create *one* variable corresponding to one of the wheel, the wheel should be selected in the tree located in the left part of the wizard. For example, selection of the ‘wset 1 left’ item corresponds to the choice of the left wheel, wheelset 1.

Figure 1.227 shows an example when the four variables in the container are created by a single click on the  button.

A *type* of a variable is selected in the list located in the right part of the wizard. The first column of the list contains standard names of variables. An identifier of a variable is constructed from the name of the corresponding type by adding the number of a wheelset and the *l* (left) or *r* (right) for wheel identification. For instance, the **YWContact1\_1r** identifier of a variable (coordinate Y of the first contact point in SC of the rail profiles – Y Wheel Contact 1) is obtained from the standard identifiers of the type **YWContact1** and relates to the right wheel of wheelset 1.

If one variable for a separate wheel is created, the user may change both the identifier and the comment in a standard manner. However, if a group of variables is created, standard identifiers and comments are assigned automatically.

Table 1.4

**Variables for the contact point**

Identifier	Comments
Creep[i]x, Creep[i]y	(unitless) Longitudinal and lateral creepages $\zeta_x, \zeta_y$ the variables are used for computing creep forces, Sect. 1.3.1.2.2. <i>"Algorithms for computing creep forces"</i> , p. 1-87.
Spin[i]	(rad/m) Spin $\varphi$ . The variables are used for computing creep forces with the FASTSIM algorithm, Sect. 1.3.1.2.2. <i>"Algorithms for computing creep forces"</i> , p. 1-87.
Creep[i]	(unitless) Module of the creepage vector $\xi = \sqrt{\zeta_x^2 + \zeta_y^2}$
FCreep[i]x, FCreep[i]y	(N) Longitudinal and lateral creep forces, Sect. 1.3.3.2. The lateral creep force corresponds to the $F_l$ force in Figure 1.96.
N[i]	(N) Normal force $N$ , Figure 1.96.

Beta[i]	(rad) Contact angle: angle $\beta_l$ between the normal force $N$ and the Z-axis of the track SC, Figure 1.96. The angle is positive for inclination of the normal forces inward the track.
YWContact[i], ZWContact[i]	(m) Coordinates Y (lateral) and Z (vertical) of the contact point in SC of the wheel profile (Sect. 1.1.3.5. "Wheelset geometry", p. 1-13).
XRContact[i], YRContact[i], ZRContact[i]	(m) Coordinates Y (lateral) and Z (vertical) of the contact point in SC of the rail profile (Sect. 1.2.1.1. "Geometry of rails in an ideal track", p. 1-66).
MWear[i]	Wear factor is power of frictional forces (W): $M_{wear} = \int_A \mathbf{v}\boldsymbol{\tau} dA,$ where $\mathbf{v}$ is "true" sliding, $\boldsymbol{\tau}$ is tangential traction.
Tgamma[i]	Wear factor is $T\gamma$ number (N): $T\gamma =  T_x \xi_x  +  T_y \xi_y $
Iw[i]	Wear index (N/mm <sup>2</sup> ): $I_w = \frac{T\gamma}{A},$ where A is area of the contact patch.
FIs[i]	Surface fatigue index (N): $FI_{surf} = \frac{\sqrt{T_x^2 + T_y^2}}{N} - \frac{k}{p_0} > 0,$ where $T_x$ and $T_y$ are longitudinal and lateral creep forces, $N$ is normal contact force, $k$ is yield stress in pure shear, $p_0$ is maximal pressure in the contact.
Area[i]	(m <sup>2</sup> ) Area of the contact patch.
Pressure[i]	(Pa) Maximal pressure in the contact.
TauMax[i]	(Pa) Maximal tangential traction in the contact.
a[i], b[i]	(mm) Longitudinal and lateral semi-axis of the elliptical contact patch or equivalent non-elliptical.
AdhArea[i]	(m <sup>2</sup> ) Area of adhesion zone in the contact patch.
Penetration[i]	(mm) Maximal geometrical penetration.

Table 1.5

**Wheel variables**

Identifier	Comments
Csafety	Derailment quotient (Russian version) $S_{RU}$ , Eq. (1.16). If $S_{RU} > 5$ then the value $S_{RU} = 5$ is accepted. The value $S_{RU} = 5$ is set if there is no contact between a wheel and a rail (a full separation). The motion is safe if $S_{RU} > 12$ .
CSafetyRefined	Refined derailment quotient (Russian version), $S_{RU}$ , Eq. (1.17). Usually $S_{RU,r} > S_{RU}$ .

Nadal	Nadal derailment criterion $S_{L/V}$ , Eq. (1.14). According to UIC 518, $Y/Q < 0.8$ for $R > 250m$ Zero value is set if there is no contact between a wheel and a rail (a full separation).
Weinstock	Derailment criterion proposed by Weinstock $S_w$ , Eq. (1.15).
SFC	Combined safety factor (Russian version), $\lambda_c$ , Eq. (1.19). Is recommended for analysis of derailment accidents by simulation.
ZLifting	(m) Vertical lifting the wheel running surface, Figure 1.234. For profiles with one-point contact (Figure 1.88, left): raise of the wheel over the critical position.
ZLiftingRelative	(unitless) Relative lifting of a wheel over the rail, Eq. 1.18).
psi	(rad) Angle of attack: the yaw angle of wheels with respect to the rail.
Fx	(N) Total longitudinal force acting on the wheel along the X-axes of the track system of coordinates (Sect. 1.2.1.3. "Switch geometry", p. 1-71).
Y(L)	(N) Total lateral force acting on the wheel along the Y-axes of the track system of coordinates (Sect. 1.2.1.3. "Switch geometry", p. 1-71).
Q(V)	(N) Total vertical force acting on the wheel along the Z-axes of the track system of coordinates (Sect. 1.2.1.3. "Switch geometry", p. 1-71).
dYRail, dZRail	(m) Lateral and vertical deflection of rail under the wheel
dPhiRail	(rad) Torsion deflection of rail under the wheel
dyW/R	(m) Lateral displacement of the wheel relative to the rail.
D.ratio (W/R)	Dynamic factor on the rail/wheel level. The value is computed according to the formula $(F_z - F_{z0})/F_{z0}$ , where $F_z$ is the total vertical force in the track SC, $F_{z0}$ is the static load for a wheel.
dyTrack, dzTrack	(m) Horizontal and vertical rail irregularities at the wheel position.
DerivYTrack, DerivZTrack	(unitless) Derivative of lateral and vertical irregularities with respect to the longitudinal coordinate
dyRailFull	(m) Lateral rail position taking into account flange clearance, flexible deflection and lateral irregularities, Sect. 1.4.3.1.8. "Example: draw plots for lateral motion of wheelset in the rail gangway", p. 1-202.
MWear, AWear, SWear	Summed on all contact points wear factors, namely, power (W), work (J), specific work (J/m) of friction forces. $A_{wear} = \int_0^t M_{wear} dt$ $S_{wear} = A_{wear}/S$ , where $S$ is the traveled distance.

### 1.4.3.1.2. Variables related to the wheelset

Variables related to the wheelset are shown on the **Wheelsets** tab (Figure 1.229). Variables description is in Table 1.6.

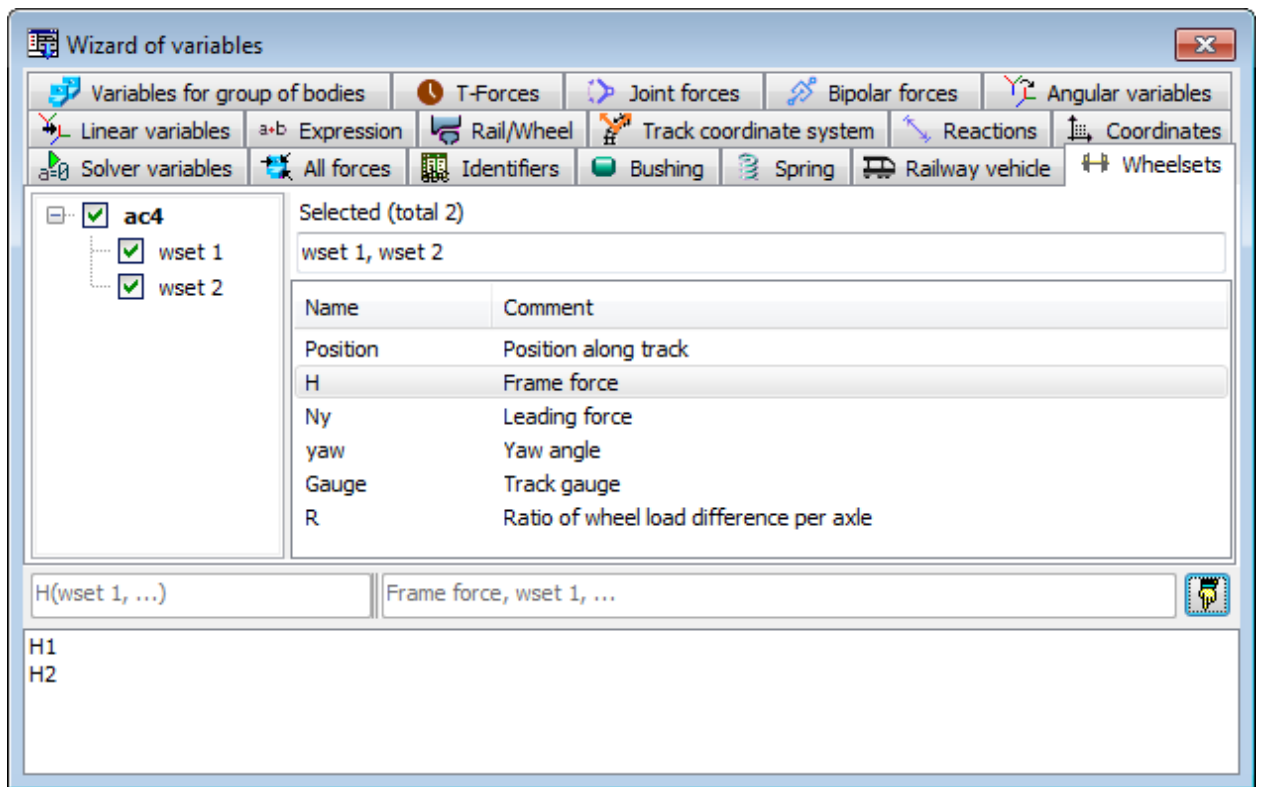


Figure 1.229. Variables related to the wheelset

Table 1.6

**Variables related to the wheelset**

Identifier	Comments
Position	(m) Wheelset position along the track
H	(N) Frame force. Frame force or total force acting on the frame from the wheelset, see Sect. 1.4.3.1.5. "Lateral and frame forces", p. 1-197.
Ny	(N) Leading force: a projection of normal reaction of N on axis-related system of coordinates for the given wheel set. To compute the projection use a formula $N_y = \pm N \sin(\beta)$ , take minus sign for the left wheel.
yaw	(rad) Yaw angle of the wheelset
Gauge	(m) Track gauge under the wheelset, Figure 1.170.
R	Ratio of wheel load difference per axle according to the standard EN 14363: $R = \frac{ Q_l - Q_r }{Q_l + Q_r},$ where $Q_l$ and $Q_r$ are a total vertical force acting on the left and right wheel respectively.

**1.4.3.1.3. Variables related to the railway vehicle**

The variables, which relate to the railway vehicle, are shown on tab **Railway vehicle** (Figure 1.230). The description of variables is shown in Table 1.4.

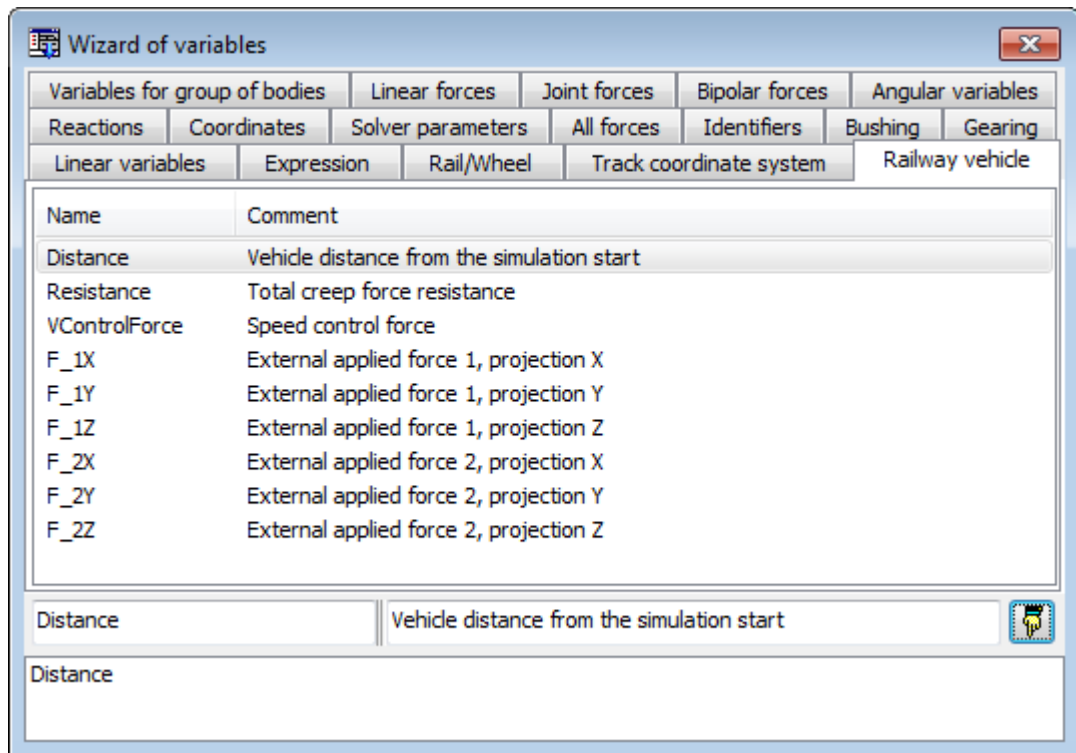


Figure 1.230. Related to the railway vehicle variables

Table 1.7

**Related to the railway vehicle variables**

<b>Identifier</b>	<b>Comment</b>
Distance	(m) A distance which vehicle has run since the start of simulation. The variable is used for plots construction of other variables depending on the distance.
Resistance	(N) Resistance force caused by creep forces. It is calculated as a summarized power of creep forces of all wheel sets divided by longitudinal velocity.
VControlForce	(N) Speed control force (See Sect. 1.4.2.2. "Modes of longitudinal motion of vehicle", p. 1-139, Sect. 1.4.2.2.2. "v=const", p. 1-139 and 1.4.2.2.3. "Profile", p. 1-140).
F_1X, F_1Y, F_1Z, F_2X, F_2Y, F_2Z	(N) Forces, applied to the automatic couplers (See 1.4.2.6. "Setting in-train forces", p. 1-173).

**1.4.3.1.4. Wheel climb derailment criteria**

**1.4.3.1.4.1. Nadal and Weinstock criteria**

Derailment criteria are variables which allow estimating a level of derailment danger by wheel climb. Often these variables are different for field test and computer-aided simulations because of lack in data available for experimental evaluation, for example, friction coefficient.

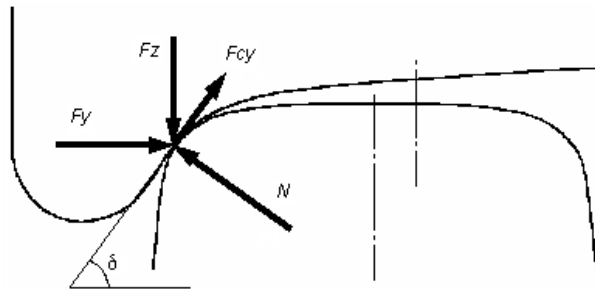


Figure 1.231. Wheel contact forces

Derailment criteria are mainly based on the following assessment of the ratio of the total lateral (L or Y) and vertical (V or Q) forces applied to a wheel from the rail in case of one-point flange contact, Figure 1.231:

$$\frac{L}{V} = \frac{Y}{Q} = q(\delta, \mu_y) = \frac{\tan \delta - \mu_y}{1 + \mu_y \tan \delta} \tag{1.5}$$

Here  $\delta$  is the contact angle,  $\mu_y = F_{cy}/N$  is the ratio of the lateral creep force to the normal force in the contact,  $L = F_y, V = F_z$ .

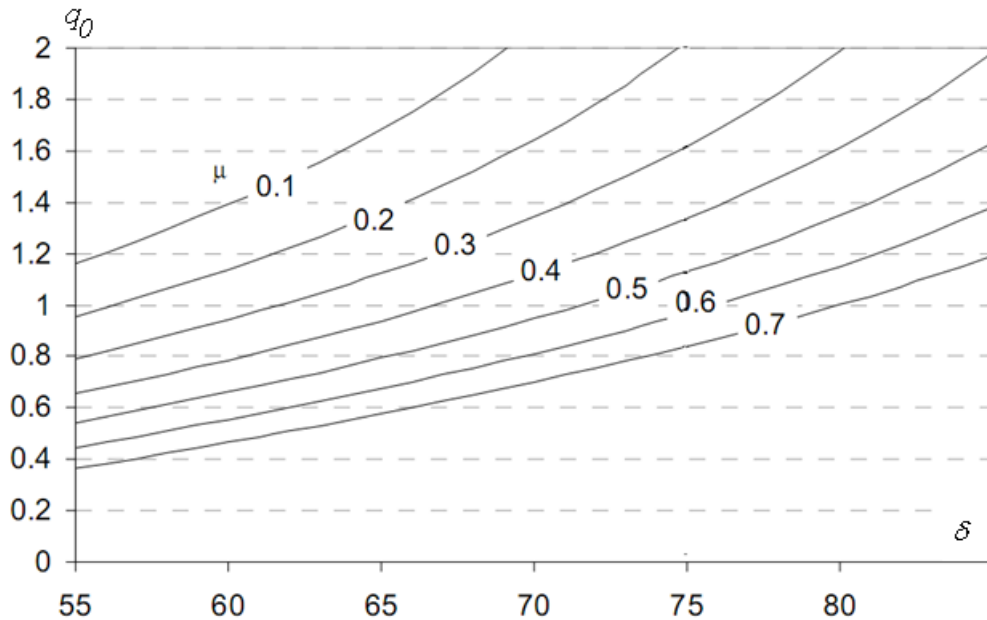


Figure 1.232. Nadal value versus flange angle for different values of coefficient of friction

If we suppose that the longitudinal creep force is small and a pure sleep in contact takes place, i.e.  $F_{cy} \approx \mu N$  with  $\mu$  as the coefficient of friction, the formula is converted to the known Nadal criterion

$$S_{L/V} = \frac{L}{V} < q_0 = q(\delta_0, \mu) = \frac{\tan \delta_0 - \mu}{1 + \mu \tan \delta_0}, \tag{1.6}$$

where  $q_0$  is the Nadal value computed for the maximal value of the contact angle on flange  $\delta_0$  (flange angle), Figure 1.232. This criterion is very conservative for small and negative angles of attack. Nevertheless it is often used in field tests because of difficulties with measurement of angle of attack or the lateral creep forces.

Weinstock [10] proposed an alternative and less conservative derailment criterion based of evaluation of L/V ratio for both left and right wheels

$$S_W = \frac{\left(\frac{L}{V}\right)_l - \left(\frac{L}{V}\right)_r}{\mu + q_0} < 1. \tag{1.7}$$

**1.4.3.1.4.2. Safety factor in Russian railways. Refined and combined factors**

A slightly different from Nadal derailment criterion is used by Russian Railways, the so called ‘‘safety factor’’ is

$$S_{RU} = \frac{V}{L} q_0 > 1. \tag{1.8}$$

It is clear that the criteria are correlated and satisfy the identity

$$S_{L/V} \cdot S_{RU} = q_0$$

Some advantage of the  $S_{RU}$  factor is the independence of its critical value on the coefficient of friction and flange angle.

To make the assessment of derailment danger by simulation more realistic, wheel lift is often used as the main criterion of wheel climb derailment.

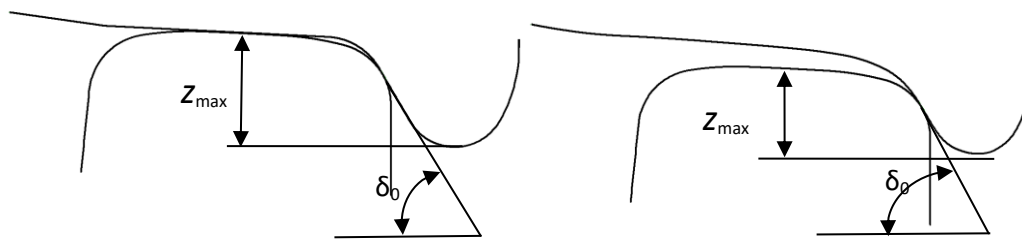


Figure 1.233. Critical positions for two- and one-point contacts

The refined safety factor is

$$S_{RU,r} = \frac{V}{L} q(\delta_0, \mu_y) = \frac{V}{L} \frac{\tan \delta_0 - \mu_y}{1 + \mu_y \tan \delta_0} \geq 1. \tag{1.9}$$

Its value is exactly equal to the critical value 1 when the wheel ‘starts to climb’. In case of a pair of profiles allowing the two-point contact, this position corresponds to the two-point contact with zero value of contact forces on the top of rail, Figure 1.233 left. In case of profiles with one-

point contact, the critical position corresponds to the flange contact in the point with the contact angle equal to the flange angle, Figure 1.233 right. In both cases the further lift of the wheel leads to increase of the contact angle which often results in the growth of the safety factor (1.17). Therefore, it cannot be used for assessment of wheel climb beyond the critical position shown in Figure 1.233.

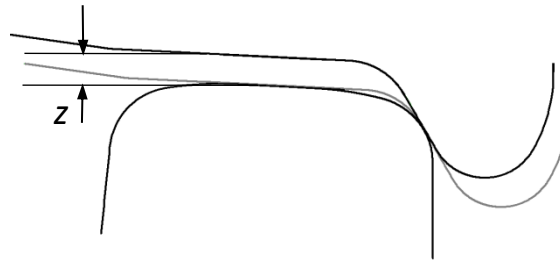


Figure 1.234. Wheel lift over the rail head

Second, the relative vertical displacement of a wheel over the critical position is evaluated according to the formula

$$S_z = \frac{z_{\max} - z}{z_{\max}} \in [0,1], \quad z > 0, \tag{1.10}$$

where  $z$  is the lift of the wheel relative to the position in Figure 1.205. Like the refined safety factor  $S_{RU,r}$  the lifting factor  $S_z$  is equal to unity at the critical position, Figure 1.234.

The formula for the factor combination of  $S_{RU,r}$  and  $S_z$  in Eqs. (1.17), (1.18) is

$$S_{\text{comb}} = \begin{cases} S_{RU,r} & \text{if } z \leq 0, \\ S_z & \text{if } z > 0. \end{cases} \tag{1.11}$$

The combined safety factor decreases continuously by passing the wheel through the critical position and subsequent vertical displacement by climb. Zero value of the factor (1.19) corresponds to the wheel flange position on the top of the rail in Figure 1.235 and, in fact, to the derailment.

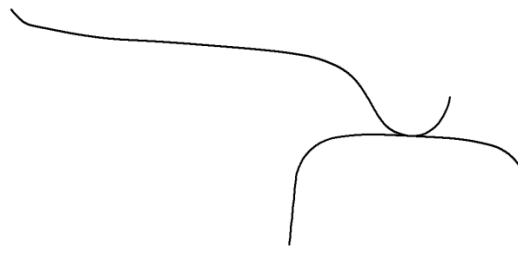


Figure 1.235. Wheel top climb

Value of the combined factor below unity gives a quantitative estimation of the wheel lift. For instance, the value  $S_{\text{comb}} = 0.85$  corresponds to 15% lift.

**1.4.3.1.4.3. Simulation of motion of a freight car over dangerous combination of irregularities**

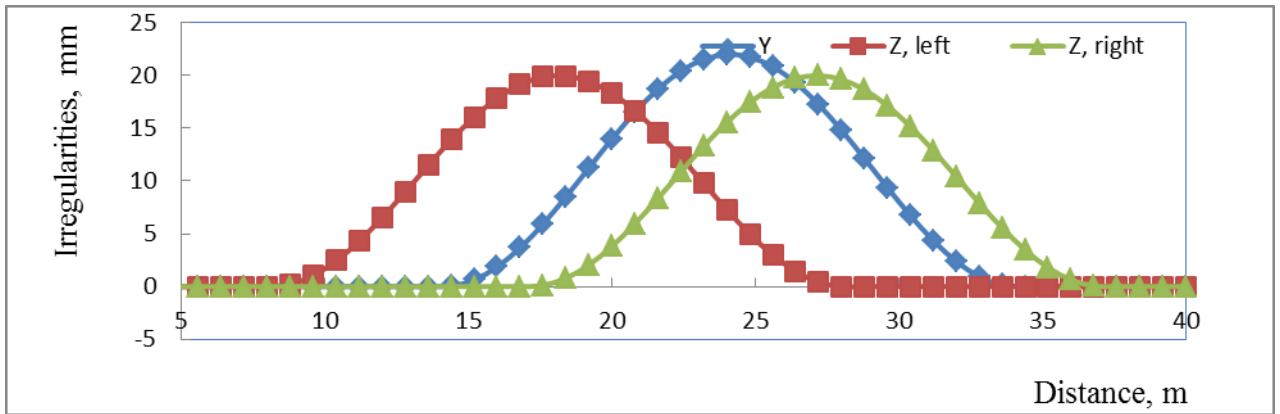


Figure 1.236. Track irregularities at longitudinal position of Wheelset 1

Consider a simulation example. An empty freight hopper with three-piece bogies runs on a tangent section of a track with speed 20 m/s. Vertical and horizontal irregularities are shown in Figure 1.236. The horizontal irregularities are the same for the left and right rail. The corresponding wave has 20 m lengths and 22 mm height. Vertical waves of 20 m length and 20 mm height differ 9 m in phase. The horizontal irregularities begin 6m after the vertical left one. Coefficient of friction in wheel/rail contacts is 0.4. Flange angle is 600. The Nadal value in this case is  $q_0 = 0.79$ .

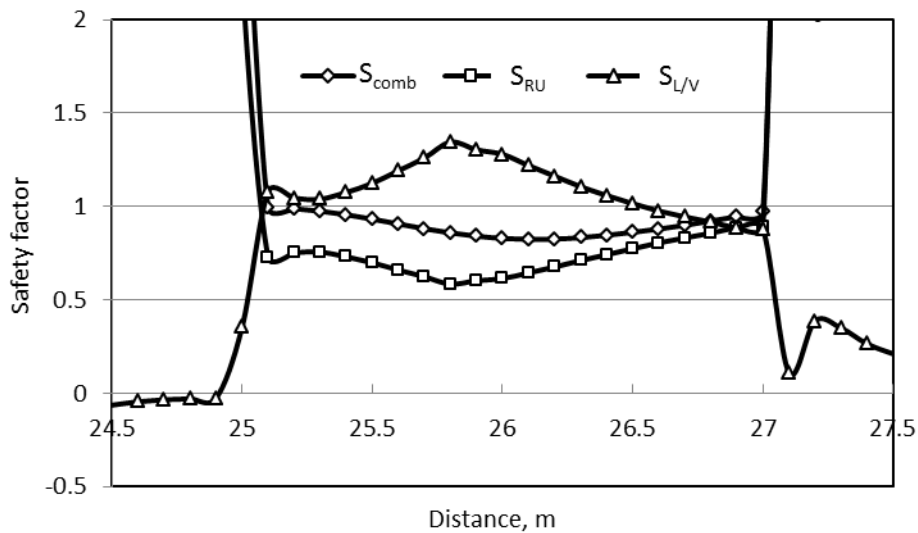


Figure 1.237. Comparison of derailment criteria for Wheelset 1

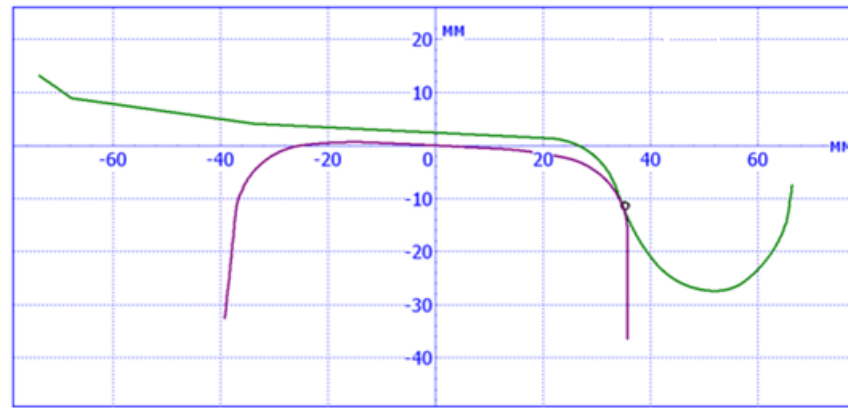


Figure 1.238. Maximal raise of the wheel in simulation

Safety factors (1.14), (1.16), (1.19) for the wheelset 1 are drawn in Figure 1.237. All of them show a dangerous situation. The maximal  $L/V$  ratio 1.35 is much greater than the Nadal value 0.79, the Russian safety factor is less 1, and falls down to 0.58. Finally, the combined safety factor gives the most informative result showing that the maximal wheel lift over the rail is 18%, Figure 1.238. Lateral motion of Wheelset 1 is shown in Figure 1.239 by the thick line. Thin lines depict rail positions taking into account their elastic deflections. Constant lateral shifts of the rails are equal to the flange clearance so that touching the wheelset displacement curve and a rail line indicates the state of two-point contact. Shift of the wheelset curve outside of the gangway indicates the wheel lift over the critical position.

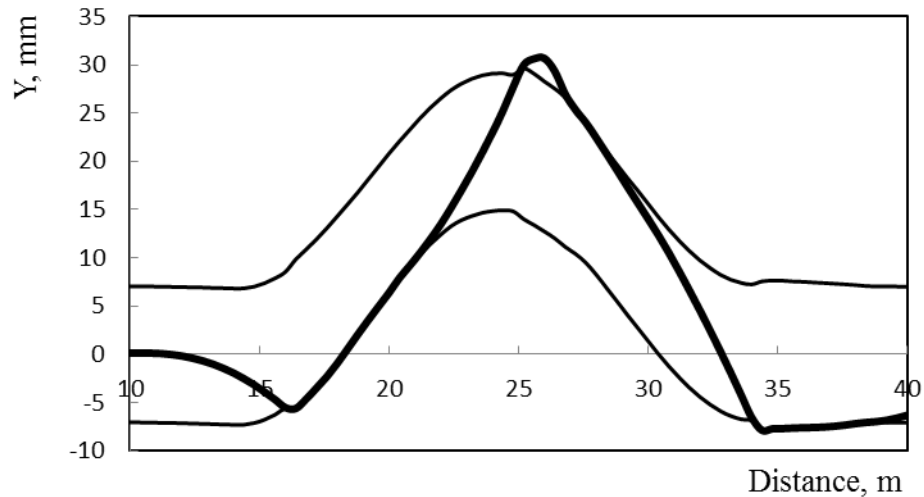


Figure 1.239. Lateral motion of Wheelset 1

#### 1.4.3.1.5. Lateral and frame forces

Lateral (Y or L) and frame (H forces) present important variables using for evaluation of load on the track from the vehicle in lateral direction.

Lateral force. Total lateral force acting on the wheel along the Y-axes of the track system of coordinates (Sect. 1.2.1.3. "Switch geometry", p. 1-71).

Frame force or total force acting on the frame from the wheelset (H-force according to UIC 518).

According to the Russian regulations, the allowed frame forces are:

- empty freight car  $H_p/P_0 \leq 0.38$ ;
- loaded freight car  $H_p/P_0 \leq 0.3$ ;
- locomotive  $H_p/P_0 \leq 0.4$ .

Here  $P_0$  is the static axle load.

Allowed guiding forces should be less than 100 kN.

**1.4.3.1.6. Example: creating variable  $\Sigma Y$**

Let us create a variable  $\Sigma Y$  according to UIC 518 is, that in fact is the total force acting from the track on the wheelset in lateral direction. This force differs from the H (‘frame’) force on the value of lateral wheelset inertia force. Thus, these variables are nearly equal for small speeds.

1. Create three variables for the first wheelset in **Wizard of variables | Rail/Wheel** tab sheet: two total lateral forces  $Y(L)_{1l}$  and  $Y(L)_{1r}$  and frame force  $H$ , Figure 1.240.

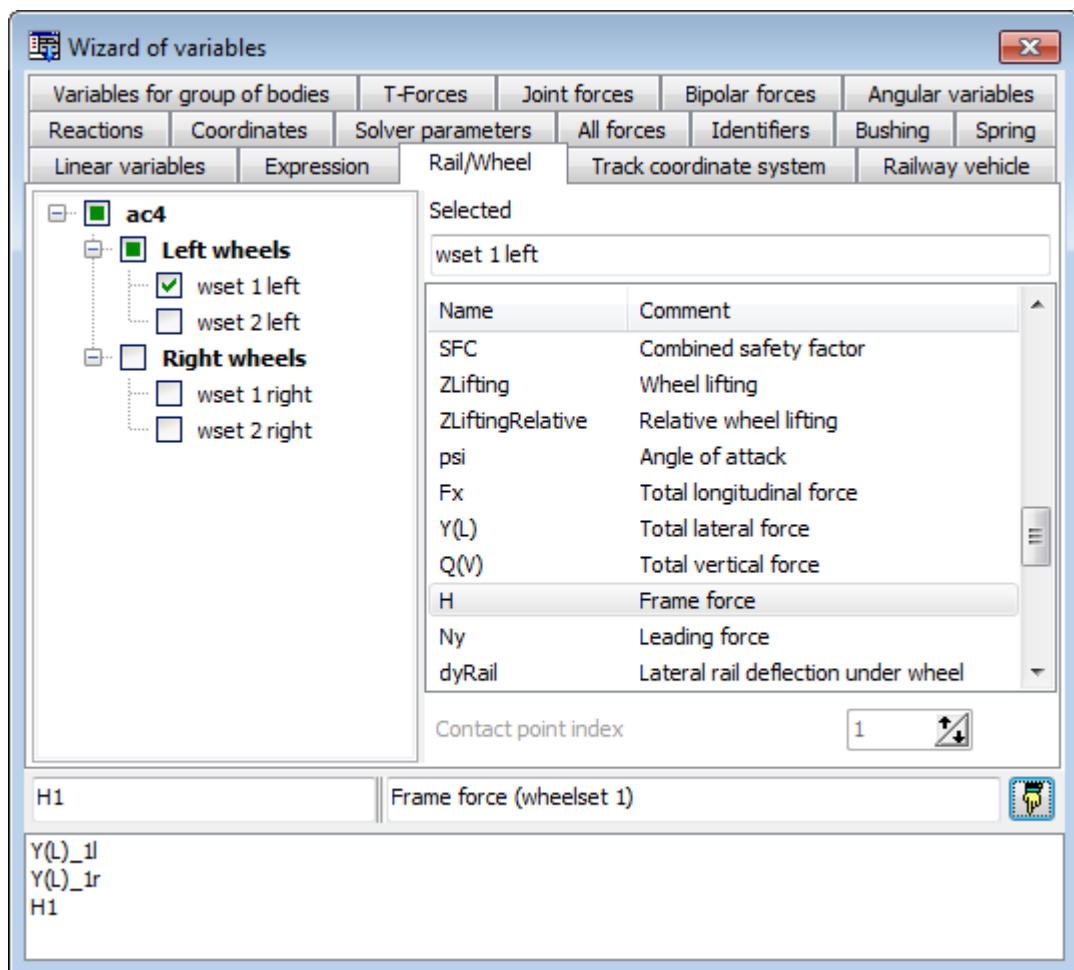




Figure 1.240. Total lateral and frame forces for wheelset 1

2. on the **Expression** tab (Figure 1.241):

- add an operator by the  button;
- drag lateral forces to the operand boxes;
- set a name of variable and comment (*Fy1* in this example);
- send the variable to container by clicking the  button.

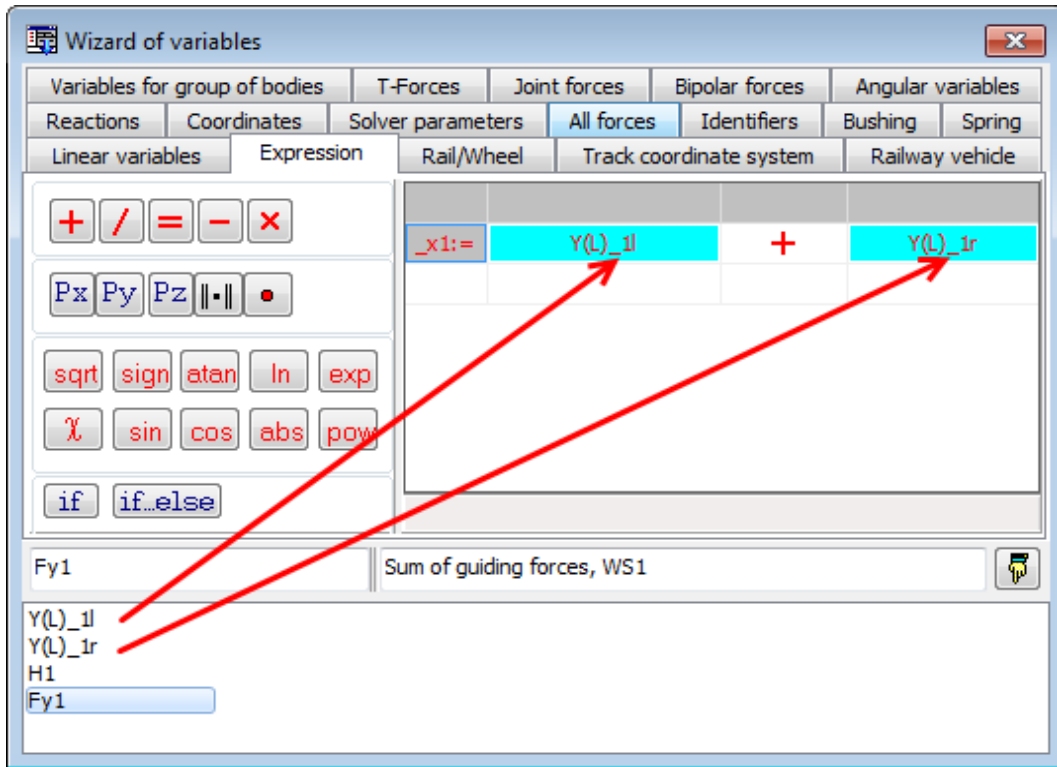


Figure 1.241. Sum of guiding forces for WS1

Comparison of the created variable with the corresponding H force for the rail car AC4 is shown in Figure 1.242.

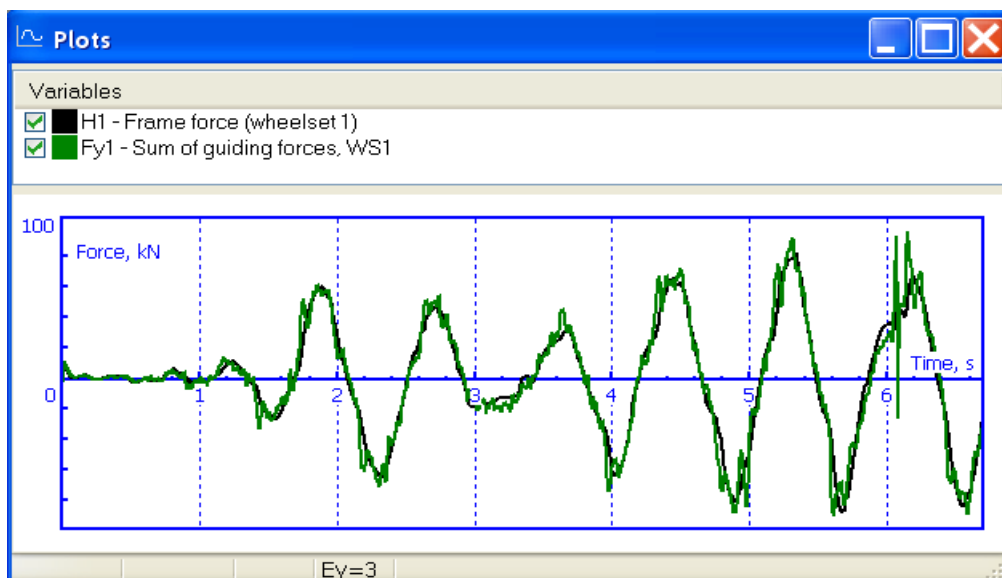




Figure 1.242. Comparison of H force with the sum of lateral forces

### 1.4.3.1.7. Example: creating variable $\Sigma H$

To create a variable, that is the sum of H (frame) forces for the leading bogie, the following steps are required:

1. create  $H$  forces for all of the wheelset of the bogie ( $H1$ ,  $H2$ ,  $H3$  variables) with the help of **Wizard of variables** | **Rail/Wheel** tab, Figure 1.243;

2. on the **Expression** tab (Figure 1.244)

- add two sum operator by the  button;
- drag variables as operands;
- set name and comment for the variable;
- send the variable to container by the  button.

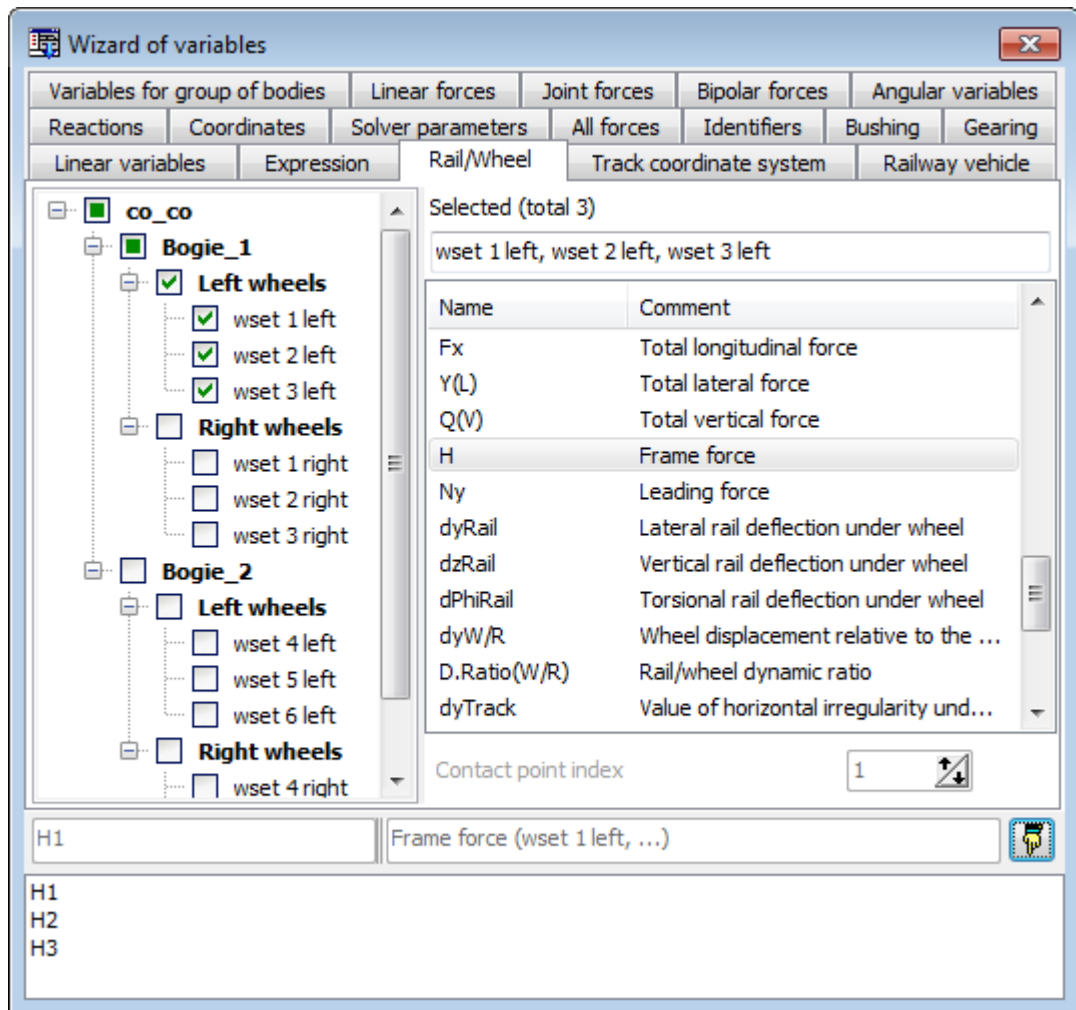


Figure 1.243. Creating H forces for the leading bogie

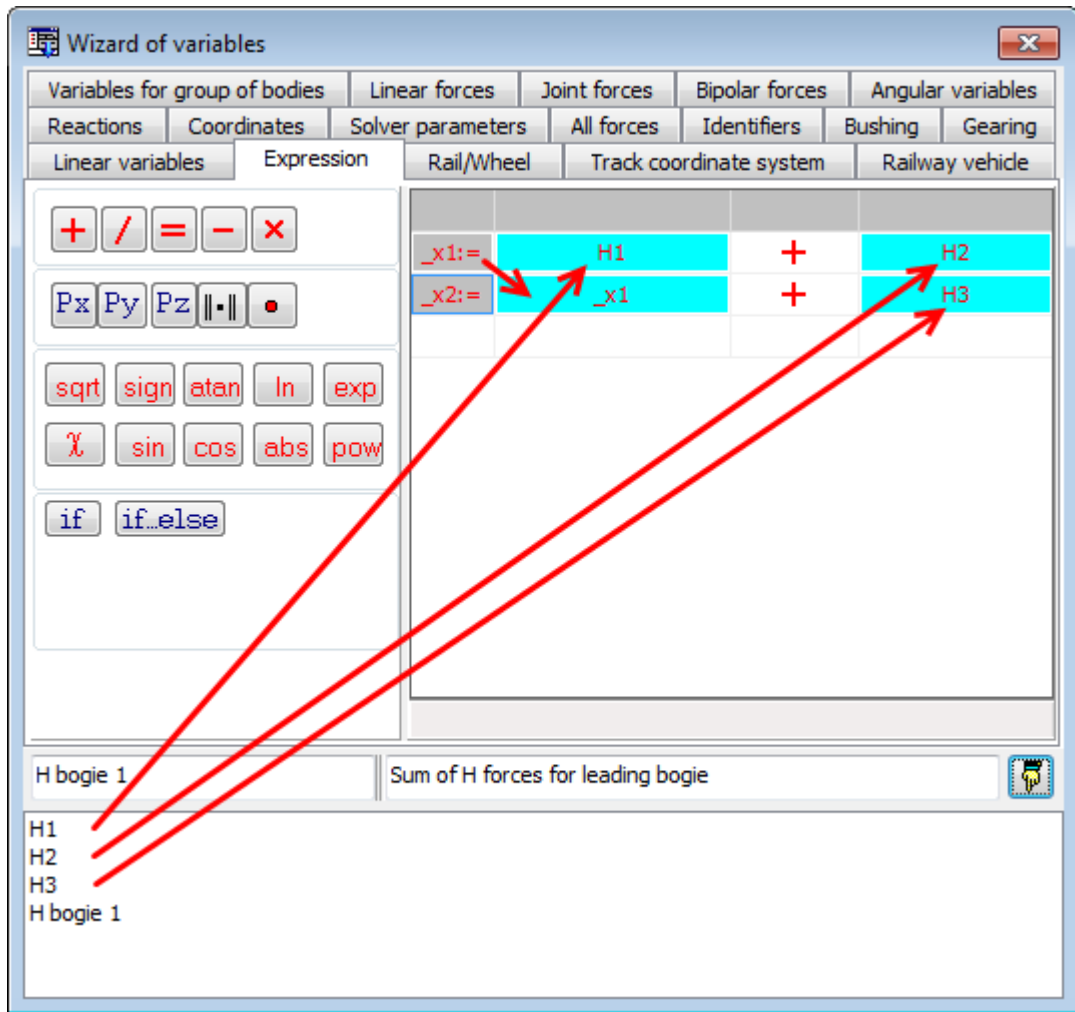


Figure 1.244. Creating a variable "Sum of H forces for leading bogie"

**1.4.3.1.8. Example: draw plots for lateral motion of wheelset in the rail gangway**

By simulation of the rail vehicle dynamics, a plot of lateral movement of a WS in the rail gangway is highly useful. The gangway is obtained by plots of lateral position of the rails under the WS taking into account flange clearance, elastic deflection and irregularities, Figure 1.245, Figure 1.246 (see Figure 1.239 as well).

In case of profiles with two-point contact, the states of contact in two points correspond to touching the WS and rail plots. In case of profiles with one-point contact, the touching of plots corresponds to the critical position of the wheel (see. Figure 1.233 right). For all types of profiles, shift of the wheelset curve outside of the gangway indicates the wheel lift over the critical position, Figure 1.239.

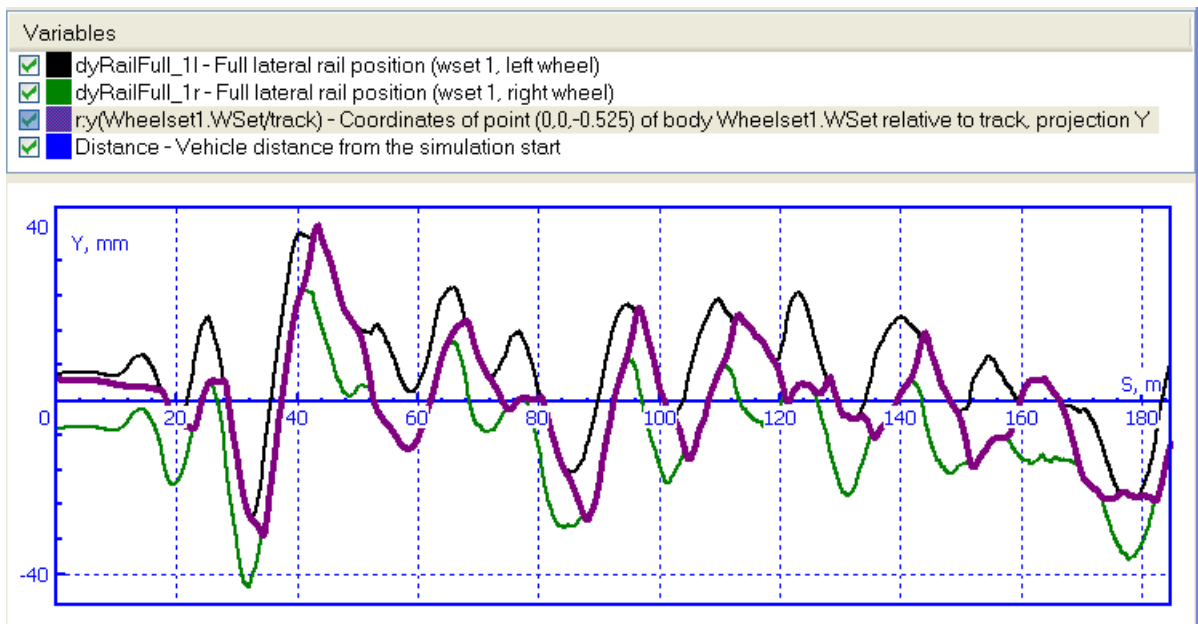


Figure 1.245. Lateral movement of WS1 in the rail gangway. Profiles of two-point contact

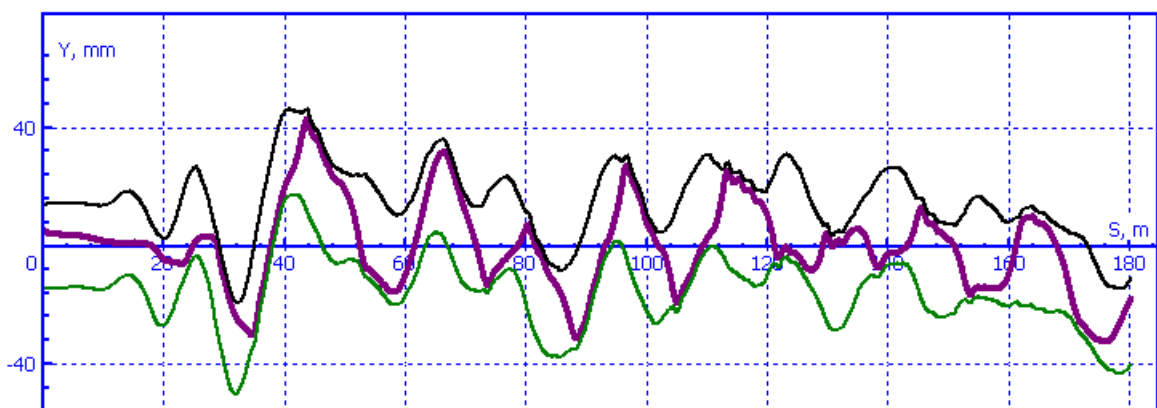


Figure 1.246. Lateral movement of WS1 in the rail gangway. Profiles of one-point contact

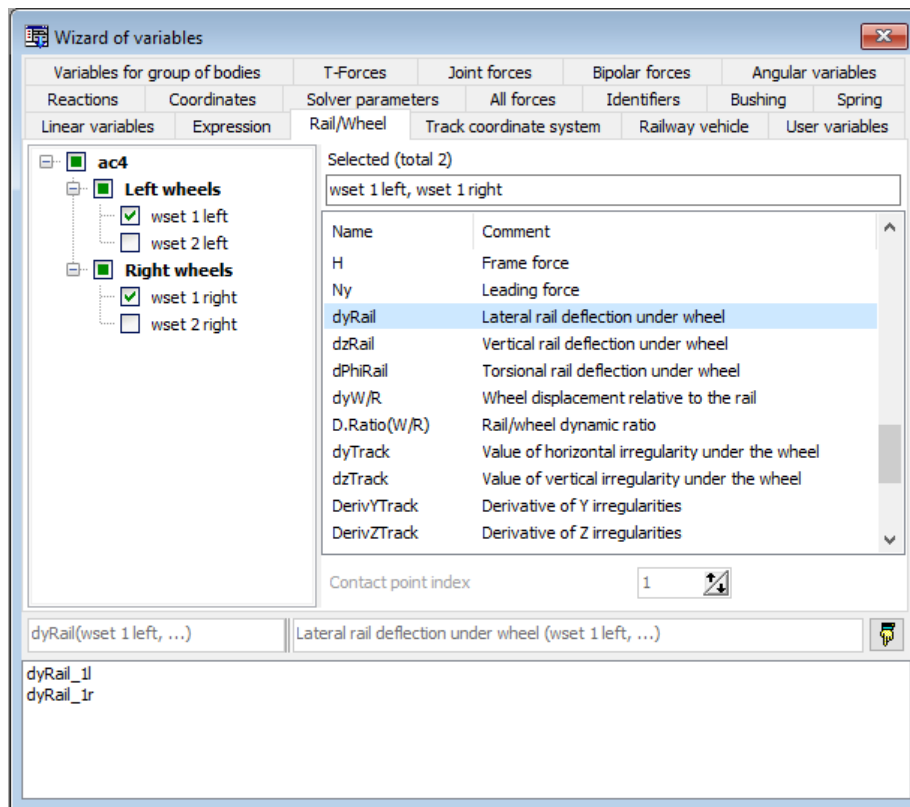


Figure 1.247. Rail position variables

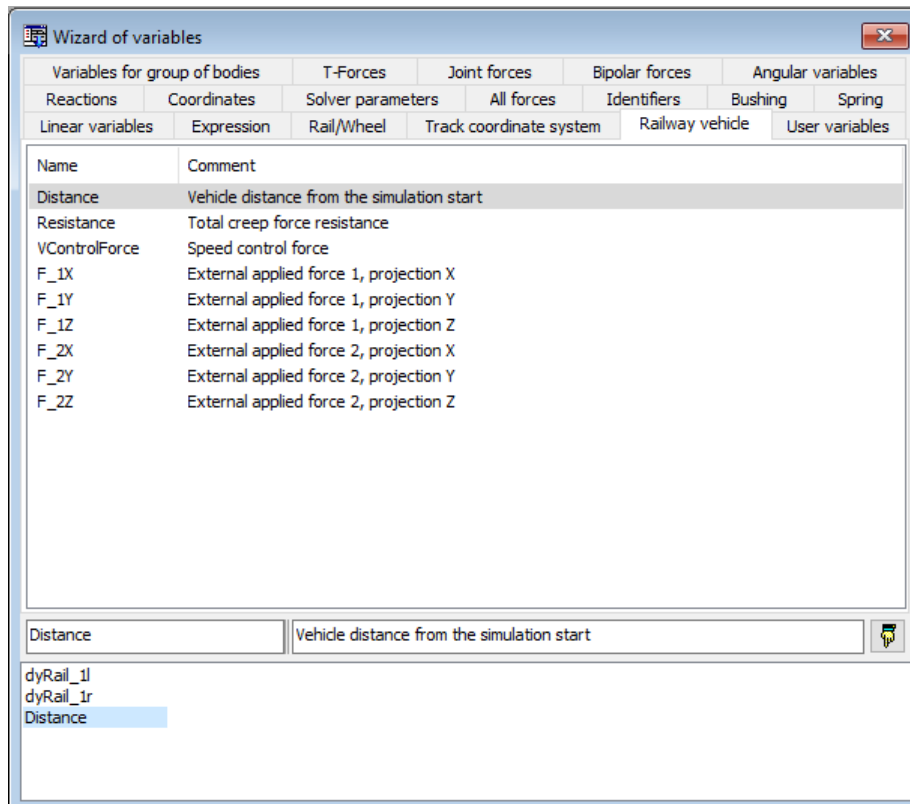


Figure 1.248. Distance variable

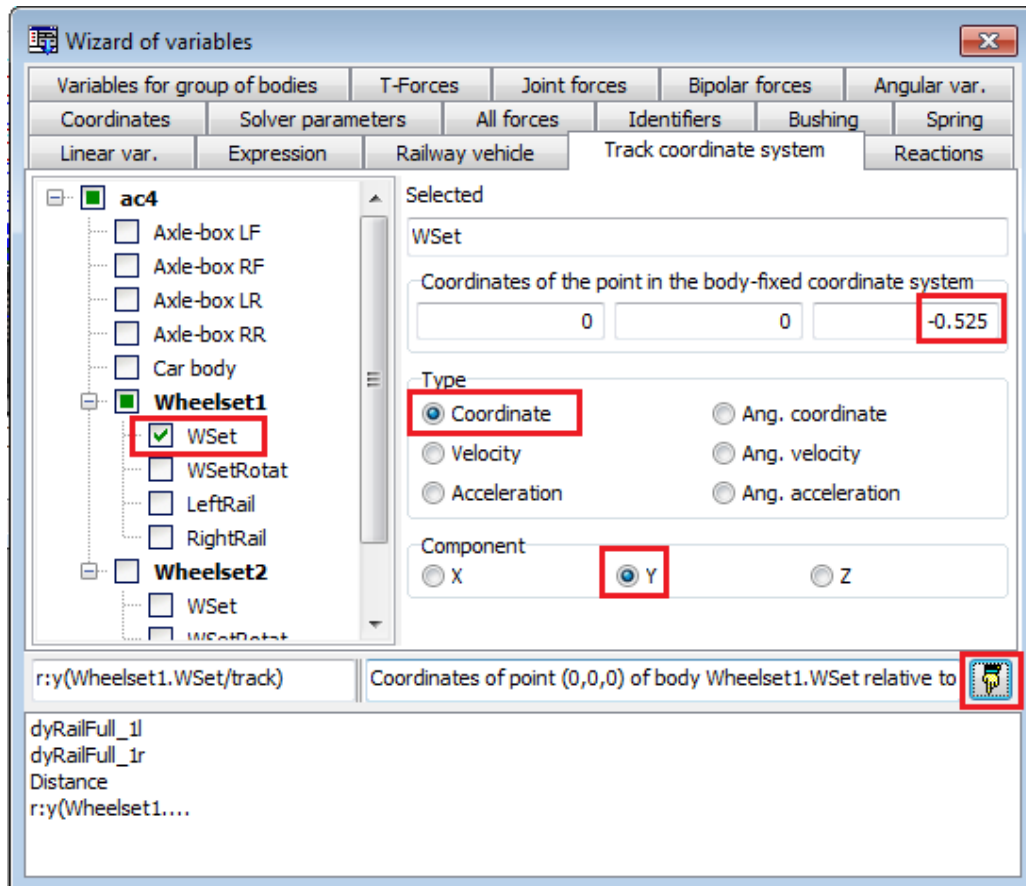


Figure 1.249. Variable ‘Lateral movement of WS1’

To get such plots, the corresponding variables are to be created with the **Wizard of Variables**, Figure 1.247 (rail positions), Figure 1.248 (distance), Figure 1.248 (lateral movement of a wheelset). Note that the lateral shift of a wheelset must be computed relative to the track system of coordinates, Sect. 1.4.3.1.9. *"Kinematic characteristics relative to track system of coordinates"*, p. 1-205. Z coordinate of the point on the wheelset base is equal to the wheel radius, Figure 1.247.

It is recommended to use **Distance** variable as an abscissa, Sect. 1.4.3.1.10. *"Use of Distance variable"*, p. 1-206.

### 1.4.3.1.9. Kinematic characteristics relative to track system of coordinates

Kinematical variables of bodies should be often projected on the track system of coordinates (TSC, Sect. 1.2.1.4. *"Track system of coordinates"*, p. 1-72). Note that axes of the TSC and SC0 in a straight track are parallel, and projections of vectors on these SC are the same.

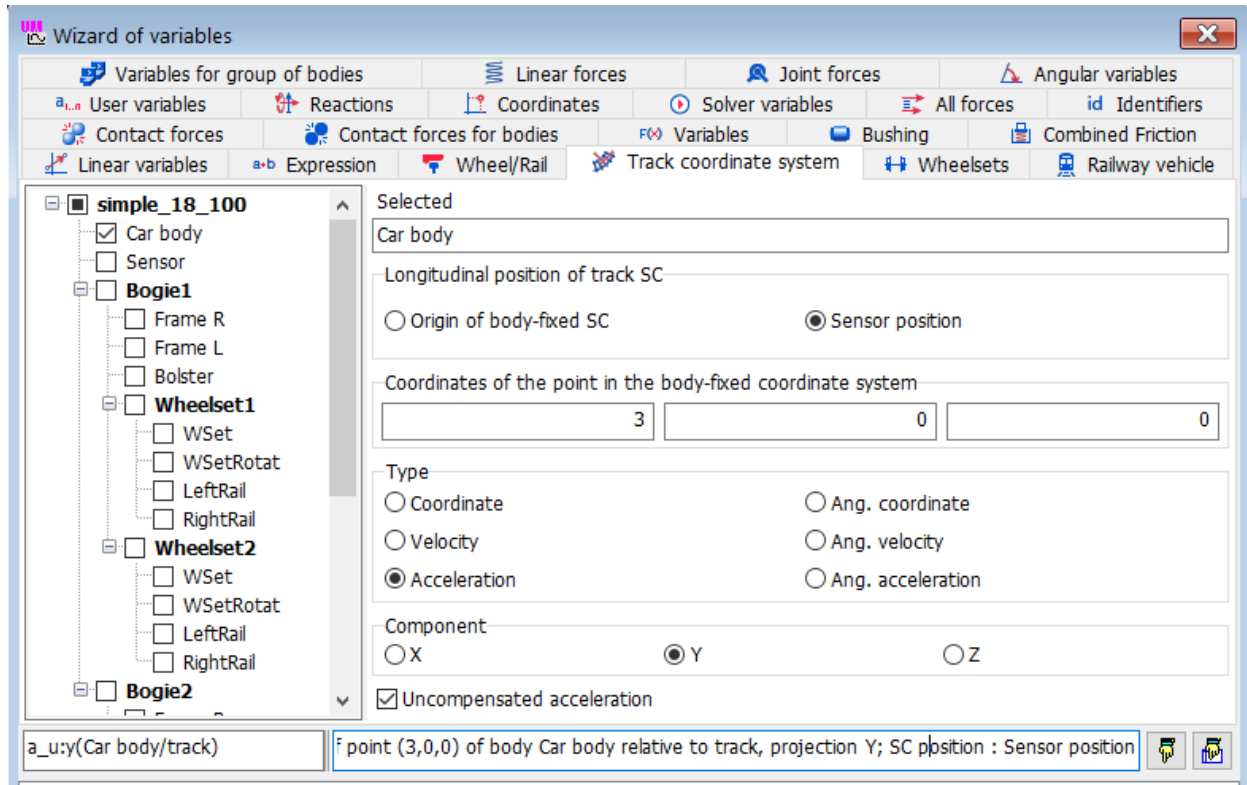


Figure 1.250. Kinematic characteristics of bodies in the track SC

The position of the TSC is associated with the body for which the kinematic variable is constructed. There are two options for determining the position of the origin of the SC on the central trajectory of the path: the longitudinal position of this point can be determined by the position of the origin of the body-fixed coordinate system, or by the position of the point, the kinematic characteristics of which are determined (sensor).

Use the **Track coordinate system** tab of the Wizard of variables to get any kinematic variable in projection of the TSC (Figure 1.250). To create the variable perform the following steps:

- select a body in the list in the left part of the wizard;
- select the type longitudinal positioning the TSC origin either on the origin of the body-fixed SC or on the sensor;
- select the type of variable: a linear variable (Cartesian coordinates, velocity or acceleration) or an angular variable (angles, angular velocity and angular acceleration);
- set a point in SC of the body, which coordinate, velocity or acceleration should be computed, if a linear variable is selected;
- set an axis of the TSC for projection.

For the lateral component of acceleration, either the uncompensated acceleration or the usual acceleration is selected (Figure 1.250).

### 1.4.3.1.10. Use of Distance variable

**Distance** variable is created on the **Railway vehicle** tab of the **Wizard of variables**, see Figure 1.251. It denotes the travel of vehicle since start the simulation. This variable has the standard name, which should not be modified by the user. Using this variable, simulation results obtained for railway vehicles running with different speed can be easily compared. Consider variants of the variable usage.

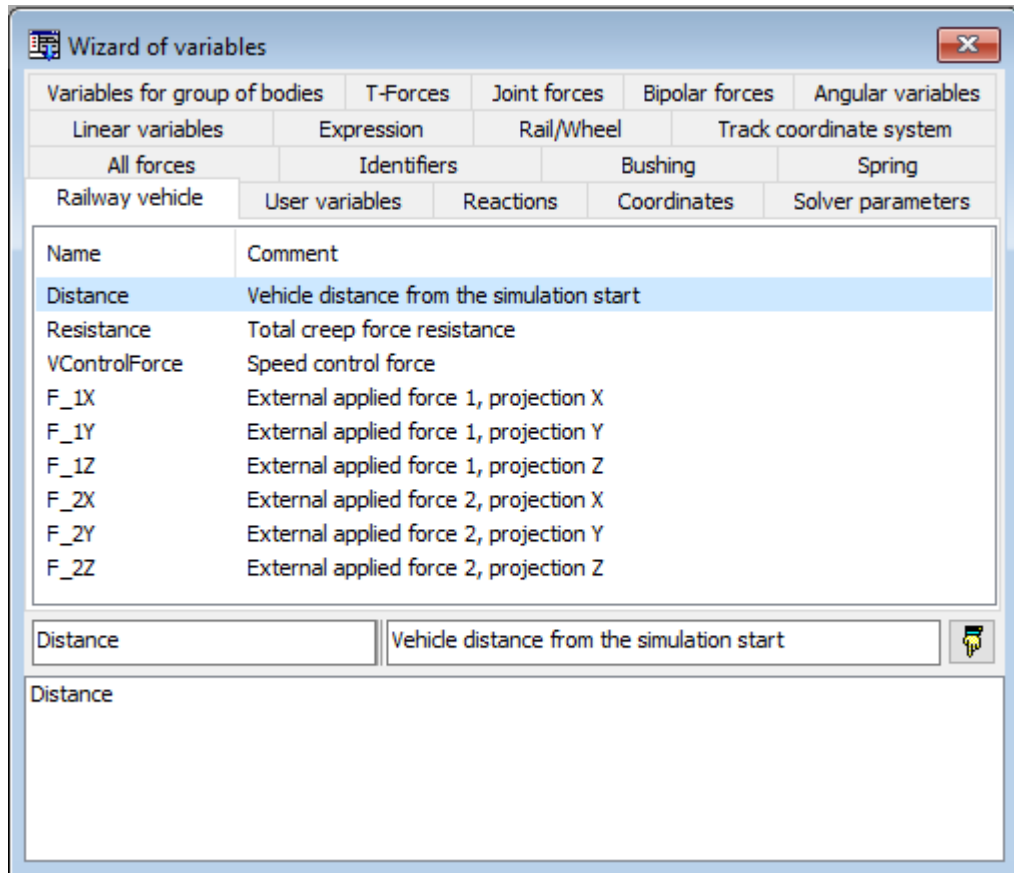


Figure 1.251. Distance variable

#### 1.4.3.1.10.1. Use of Distance variable in graphic windows

Drag the *'Distance'* variable into the graphical window. Select the variable and click the right mouse button. Click the **Lay off variable as abscissa** command in the popup menu.

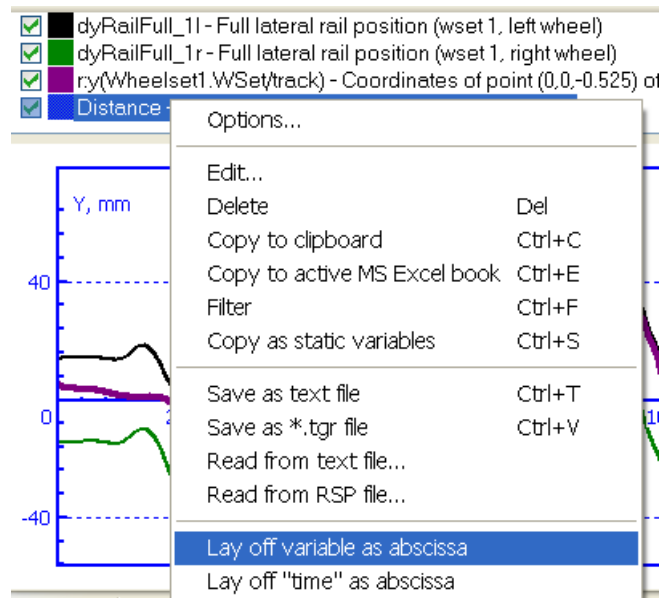


Figure 1.252. Plotting variables versus the distance

**1.4.3.1.10.2. Use of *Distance* variable in list of calculated variables**

It is strongly recommended that list of object variables for a railway vehicle always contains the ‘Distance’ variable. If it does not include this variable, add it to the list before simulation or scanning the object ([Chapter 4](#), Sect. *Assignment and usage of a list of automatically calculates variables*). To get dependence of computed variables on the distance after the simulation, drag it into the box **Lay off as abscissa**. After that all the variables dragged from the list into the graphic windows, table processors, etc., are processed in dependence on the distance.

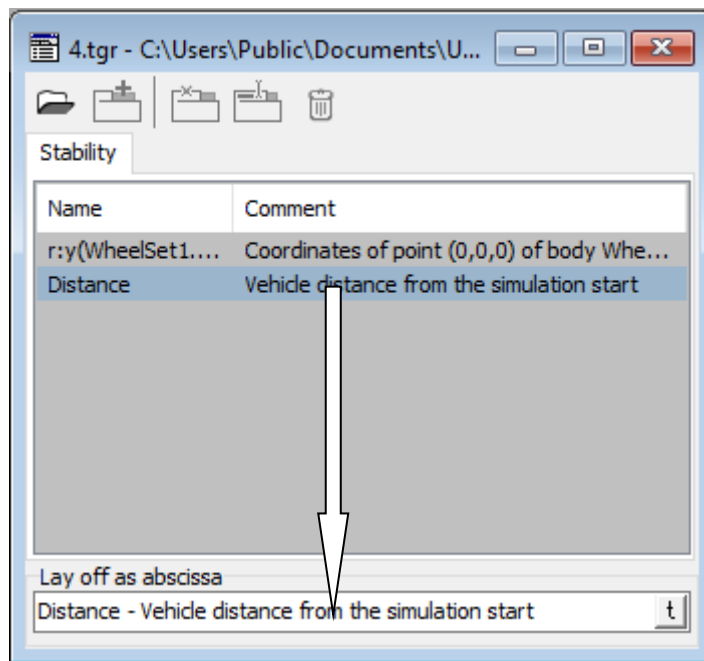


Figure 1.253. Setting dependence of calculated variables on the distance

### 1.4.3.2. Animation window

General information about usage of animation windows can be found in [Chapter 3](#), Sect. *Animation window*, and in [Chapter 4](#), Sect. *Animation window in the UM Simulation program*.

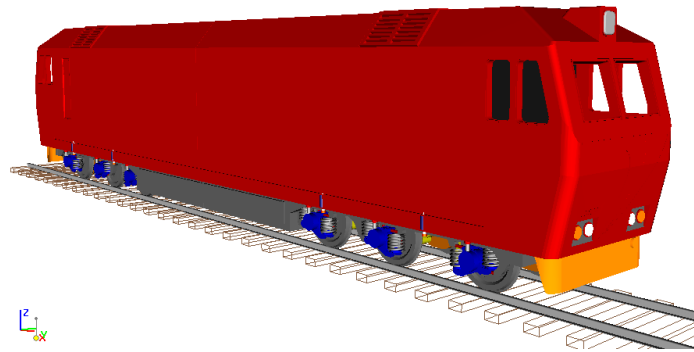


Figure 1.254. Diesel locomotive TEP70 in animation window

In addition to standard functions, visualization of rails and slippers is available for railway vehicles (the **Show gauge** check box in the **Rail/Wheel | Track | Macrogeometry** tab of the object simulation inspector).

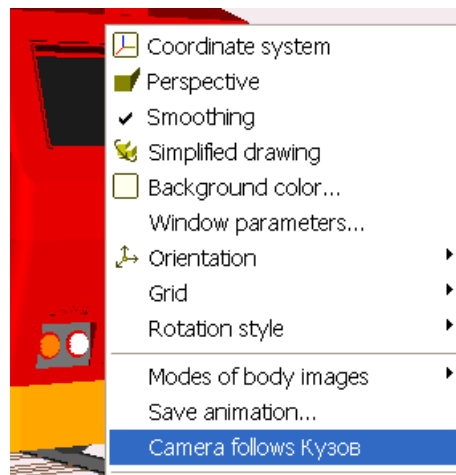


Figure 1.255. Assignment of a body to be followed by the camera

In animation window, the camera can follow a body position during the simulation. To assign the body, move the mouse cursor to the body image and click the right mouse button; select the corresponding menu command, Figure 1.255.

Figure 1.256. Contact forces in animation window. The wheel is drawn in a wired mode

Rail-wheel contact forces can be drawn in the animation window, Figure 1.256. Use the **All forces** tab of the **Wizard of variables** to create the variable for the wheelset base, and drag it into the animation window, Figure 1.257. As a rule, vector scales for forces can be corrected to get a proper length of vectors, Figure 1.258.

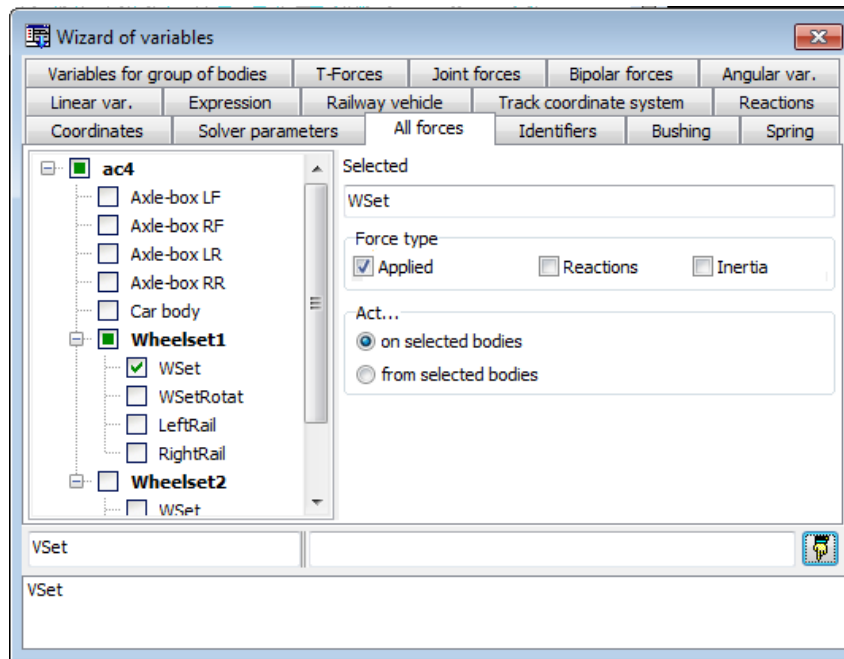


Figure 1.257. Creating variable for vectors of contact forces

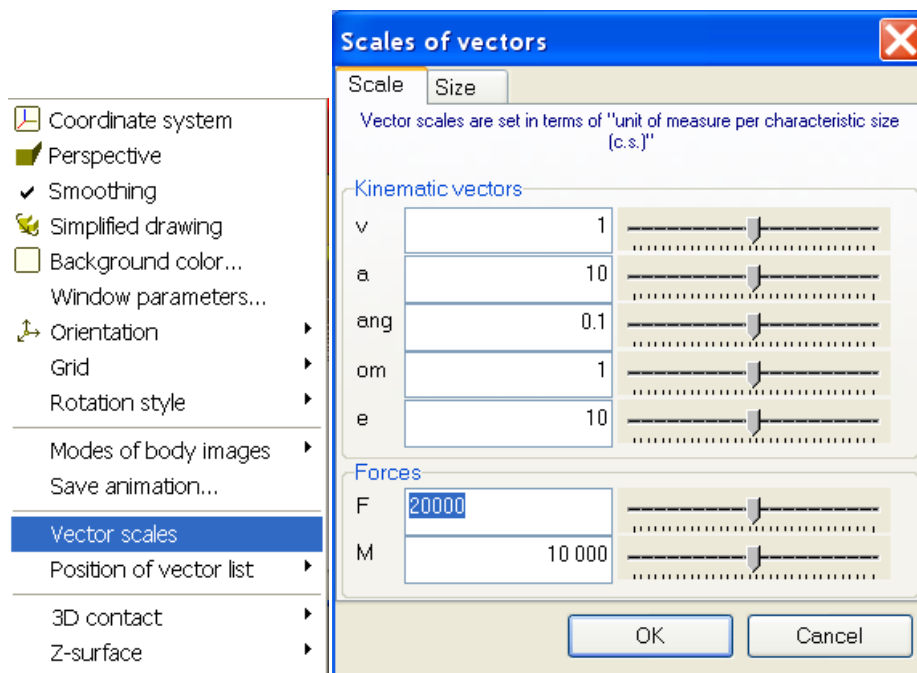


Figure 1.258. Call of window with vector scales using popup menu of animation window

### 1.4.3.3. Contact animation window

The **Contact animation window** (Figure 1.259, Figure 1.260) is a special animation window for visualization of relative positions of rail and wheel profiles as well as rail/wheel interaction forces while simulation of a vehicle dynamics. Use the **Tools | Contact animation** menu command or the *Ctrl+N* hot key to call the window.

Window parameters are located in the bottom part. These parameters allow the user to turn on/off visualization of contact forces, change vector scales and profile positioning (horizontal or vertical orientation).

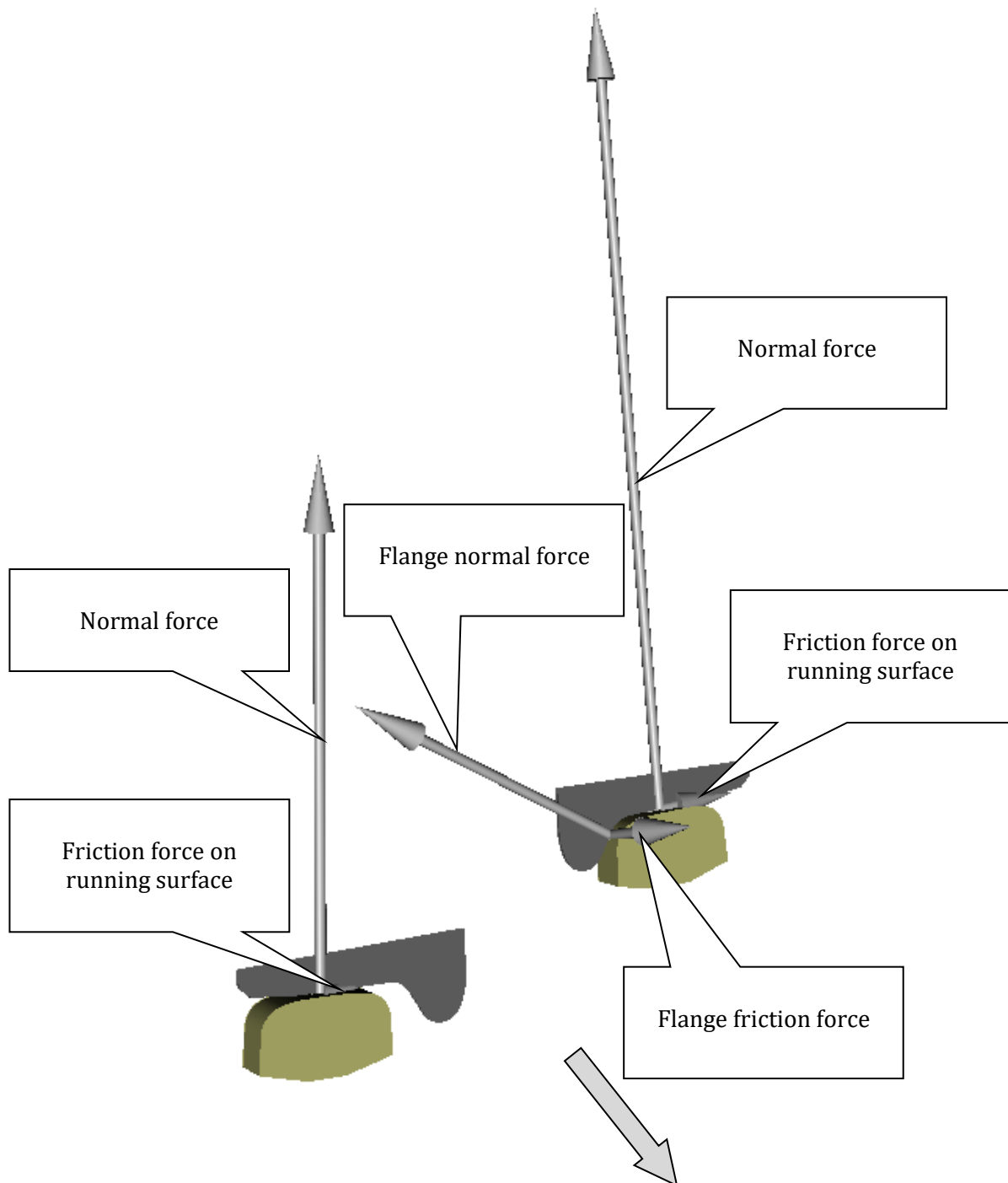


Figure 1.259. Classification of contact forces in contact animation window

Pointing the mouse cursor at a vector allows the user to get its current value. Clicking the vertical positioning circles switch on/off profiles corresponding to one of wheelsets.

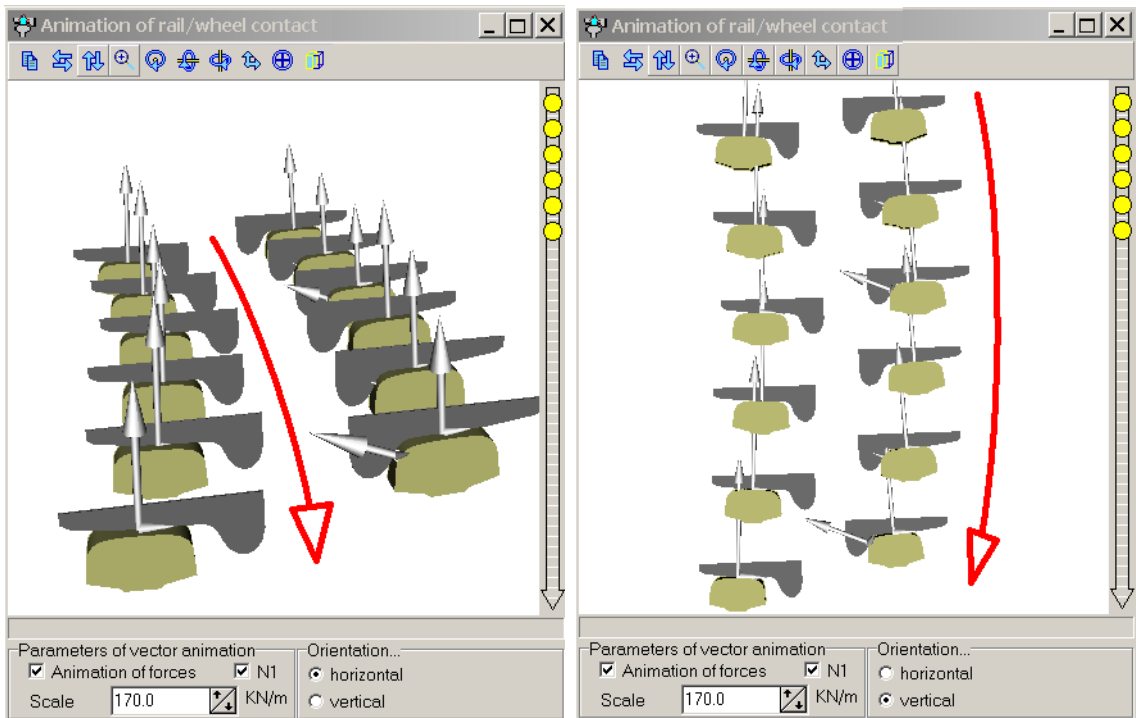


Figure 1.260. Direction of motion in the animation window

#### 1.4.3.4. Contact patch animation window

Use the **Tools | Contact patch viewer** menu command to call the contact patch viewer window (Figure 1.261).

In the **Contact patch viewer** window in the process of simulation of vehicle dynamics the user can see the following variables:

- relative position of wheel and rail profiles in the track coordinate system;
- contact points;
- vectors of normal forces;
- contact patches;
- coordinate system of the contact patch;
- adhesion zone;
- distribution of tangential forces within the contact patch in the coordinate system of the contact patch.

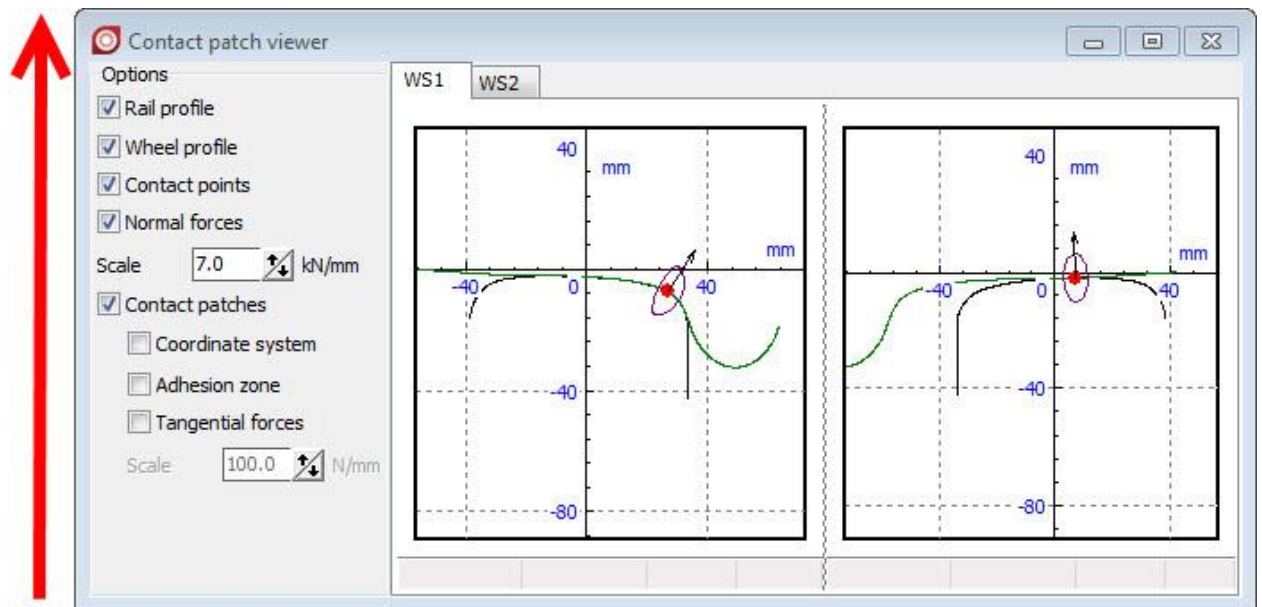


Figure 1.261. Contact patches while a vehicle is in motion in a curve. The arrow shows the direction of motion.

### 1.4.3.5. Table processor

The following functionals are distributed along with UM Loco module.

#### Functional for UM Loco module

Functional	Description
_3Min_Mean _3Max_Mean _3Max_Zero _4Min_Mean _4Max_Mean _4Max_Zero	<p>Mean value of three/four minimums or maximums. All functionals are evaluated according to the same algorithms with some minor differences. Firstly, a list of local extremums is created. Local minimums are taken for <b>_3Min_Mean</b> and <b>_4Min_Mean</b>, local maximums are taken for <b>_3Max_Mean</b>, <b>_3Max_Zero</b>, <b>_4Max_Mean</b> and <b>_4Max_Zero</b>. The window of 1/20 of the length of realization is used during creating the list of local extremums. Two adjacent extremums are added into a list of extremums if the plot intersects the abscissa axis (for functional with <b>Zero</b> postfix) or mean value (for functional with <b>Mean</b> postfix) between these extremums. Otherwise the only extremum is chosen. Then <b>3</b> or <b>4</b> extremums maximum/minimum extremums are extracted. For the <b>_4Min_Mean</b>, <b>_4Max_Mean</b> and <b>_4Max_Zero</b> functionals the smallest/biggest one is removed. Then the mean value for the rest extremums is calculated as a value of the functional.</p>

### 1.4.3.6. Ride comfort functionals

The additional functionals for ride comfort evaluation are available with **Ride Comfort Tool**.

#### Passenger comfort according to UIC513

**Functionals:** *UIC513\_Nmv\_axyp95*, *UIC513\_Nmv\_azp95*

**Library:** *Ride\_Comfort\_UIC513.dll*

**UIC513\_Nmv\_axyp95** calculates  $a_{XP95}^{Wd}$  or  $a_{YP95}^{Wd}$  from the simplified ride comfort formula (1) (see p. 7.2.1 of [11]):

$$N_{MV} = 6 \sqrt{(a_{XP95}^{Wd})^2 + (a_{YP95}^{Wd})^2 + (a_{ZP95}^{Wb})^2} \quad (1)$$

**UIC513\_Nmv\_azp95** calculates  $a_{ZP95}^{Wb}$  from the simplified ride comfort formula.

**Note.** The estimated variable must have time as abscissa and be not shorter than 5 seconds duration.

These functionals are intended for processing variables created in **Wizard of variables** which contain values of acceleration components of different points of railway vehicles. When calculating these functionals, the estimated variable is divided into segments of 5 seconds. So the estimated variable must have time as abscissa and be not shorter than 5 seconds duration. The respective effective accelerations  $a^{Wd}$  are calculated for every segment by using the method from [9]. Since the high frequency of using filters is 100 Hz, the time step of the estimated variable must be not higher than 0.005 seconds (use **Variable calculation step** parameter of the **Object simulation inspector**). After all effective accelerations are calculated, they are processed by using 95<sup>th</sup> percentile. The value of the percentile is the result of the functional.

So user can calculate the ride comfort by using results of these functionals by formula (2):

$$N_{MV} = 6 \sqrt{\left( UIC513\_Nmv\_axyp95(a_x(t)) \right)^2 + \left( UIC513\_Nmv\_axyp95(a_y(t)) \right)^2 + \left( UIC513\_Nmv\_azp95(a_z(t)) \right)^2} \quad (2)$$

where  $UIC513\_Nmv\_axyp95(a_x(t))$  is the result of **UIC513\_Nmv\_axyp95** functional for X component of the estimated acceleration,

$UIC513\_Nmv\_axyp95(a_y(t))$  is the result of **UIC513\_Nmv\_axyp95** functional for Y component of the estimated acceleration,

$UIC513\_Nmv\_azp95(a_z(t))$  is the result of **UIC513\_Nmv\_azp95** functional for Z component of the estimated acceleration.

### Sperling's ride comfort indices

**Functionals:** *Vertical\_Sperling\_Index, Lateral\_Sperling\_Index*

**Library:** *Ride\_Comfort\_Sperling.dll*

**Vertical\_Sperling\_Index** and **Lateral\_Sperling\_Index** calculate Sperling's ride comfort indices for vertical and lateral directions [10]. The ride comfort index for the lateral direction is calculated as

$$W_z = 10 \sqrt{\int_{0.5}^{30} a^3 B_w^3 df}, \quad (1)$$

where  $a$  is acceleration in frequency domain ( $\text{cm/s}^2$ ),  $f$  is the frequency (Hz),  $B_w$  is defined by the following equation:

$$B_w = 0.737 \left[ \frac{1.911f^2 + (0.25f^2)^2}{(1 - 0.277f^2)^2 + (1.563f - 0.0368f^3)^2} \right]^{\frac{1}{2}} \quad (2)$$

The ride comfort index for the vertical direction is calculated as

$$W_z = 10 \sqrt{\int_{0.5}^{30} a^3 B_s^3 df}, \quad (3)$$

where  $B_s$  is defined as

$$B_s = 0.588 \left[ \frac{1.911f^2 + (0.25f^2)^2}{(1 - 0.277f^2)^2 + (1.563f - 0.0368f^3)^2} \right]^{\frac{1}{2}} \quad (4)$$

**Note.** The estimated acceleration values in UM must be in m/s.

These functionals are intended for processing variables created in **Wizard of variables** which contain values of acceleration components of different points of railway vehicles. These variables must be in m/s. Since the high frequency of using filters is 30 Hz, the time step of the estimated variable must be not higher than 0.0167 seconds (use **Variable calculation step** parameter of the **Object simulation inspector**).

**Ride comfort factor according to Russian (exUSSR) OST 24.050.16-85  
and GOSTP 55513 – 2013**

**Functionals:** *Ride\_Comfort\_G*, *Ride\_Comfort\_V*

**Library:** *Ride\_Comfort.dll*

**Ride\_Comfort\_V** and **Ride\_Comfort\_G** calculate ride comfort indices for vertical and lateral directions according to Russian (exUSSR) branch standard 24.050.16-85 and GOST P 55513-2013, [12] and [13].

These functionals are intended for processing variables created in **Wizard of variables** which contain values of acceleration components of different points of railway vehicles. These variables must be in m/s.

**Note.** Since the high frequency of using filters is 20 Hz, the time step of the estimated variable must be not higher than 0.025 seconds (use **Variable calculation step** parameter of the **Object simulation inspector**).

### 1.4.3.7. Methods for quantifying passenger ride comfort index

This section describes methods for quantifying the assessment of the effect of accelerations on passenger ride comfort index in rail vehicles according to the European standard EN12299 [14].

This standard includes four ride comfort index:

- mean ride comfort standard method  $N_{MV}$ ;
- continuous ride comfort  $C_{C_x}, C_{C_y}, C_{C_z}$ ;
- ride comfort on curve transitions  $P_{CT}$ ;
- ride comfort on discrete events  $P_{DE}$ .

Mean comfort is a perceived comfort level, continuously adjusted, as evaluated through measurement on a long-time basis (at least some minutes). The mean ride comfort standard method  $N_{MV}$  is implemented in the UM software in the form of the **UIC513\_Nmv\_axyp95** and **UIC513\_Nmv\_azp95** functionals in the table processor (see Sect. 1.4.3.6). Below, we consider the continuous ride comfort  $C_{C_x}, C_{C_y}, C_{C_z}$ , the comfort on curve transitions  $P_{CT}$  and ride comfort on discrete events  $P_{DE}$ .

#### 1.4.3.7.1. Continuous ride comfort

Continuous comfort is a level of accelerations, ISO frequency weighted continuously evaluated as a set of root mean square values in vertical, lateral and longitudinal direction over a short time period (typical 5 s). This method can be used to assess passenger ride comfort index in straight and curved sections of the track to account for the short-term impact, which is estimated by the values of three acceleration values: in the longitudinal, lateral and vertical directions. The continuous ride comfort is a time function of  $C_{C_x}, C_{C_y}(t), C_{C_z}(t)$  five-second RMS frequency-weighted accelerations:

$$\begin{aligned} C_{C_x}(t) &= a_X^{W^d}(t), \\ C_{C_y}(t) &= a_Y^{W^d}(t), \\ C_{C_z}(t) &= a_Z^{W^b}(t), \end{aligned}$$

where  $a$  are RMS values of accelerations,  $m/s^2$  for a period of 5 s;

$W_i$  are superscript, refers to frequency values in accordance with a certain direction  $i$ : ( $i = d, d, b$ ):

$d$ : longitudinal direction,  $W_d$ ;

$d$ : lateral direction,  $W_d$ ;

$b$ : vertical direction,  $W_b$ .

Five-second RMS frequency-weighted accelerations  $a_X^{W^d}(t), a_Y^{W^d}, a_Z^{W^b}(t)$  calculated using the following formulas:

$$\begin{aligned} a_X^{W^d}(t) &= \sqrt{\frac{1}{T} \int_{t-T}^t (\ddot{x}_{W_d}(t))^2 d\tau}, \\ a_Y^{W^d}(t) &= \sqrt{\frac{1}{T} \int_{t-T}^t (\ddot{y}_{W_d}(t))^2 d\tau}, \end{aligned}$$

$$a_z^{Wd}(t) = \sqrt{\frac{1}{T} \int_{t-T}^t (\ddot{z}_{Wd}(t))^2 d\tau}$$

where  $T = 5$  s, and the value of  $t$  is a multiple of 5 s.

**Remark.** The estimated variable must use time as an abscissa (not travelled distance) and the signal length must be longer than 5 seconds.

The continuous ride comfort index is evaluated in accordance with the Table.

Table 1.8

**Continuous comfort level**

The range of values of continuous comfort	The level of comfort in terms of influence accelerations
The "sitting position in/against the direction of travel"	
$C_{C_y}(t), C_{C_z}(t) < 0,2 \text{ m/s}^2$	Very comfortable
$0,2 \text{ m/s}^2 \leq C_{C_y}(t), C_{C_z}(t) < 0,3 \text{ m/s}^2$	Comfortable
$0,3 \text{ m/s}^2 \leq C_{C_y}(t), C_{C_z}(t) < 0,4 \text{ m/s}^2$	Borderline feeling
$0,4 \text{ m/s}^2 \leq C_{C_y}(t), C_{C_z}(t)$	Less comfortable
The "sitting across the direction of travel" position	
$C_{C_x}(t), C_{C_z}(t) < 0,2 \text{ m/s}^2$	Very comfortable
$0,2 \text{ m/s}^2 \leq C_{C_x}(t), C_{C_z}(t) < 0,3 \text{ m/s}^2$	Comfortable
$0,3 \text{ m/s}^2 \leq C_{C_x}(t), C_{C_z}(t) < 0,4 \text{ m/s}^2$	Borderline feeling
$0,4 \text{ m/s}^2 \leq C_{C_x}(t), C_{C_z}(t)$	Less comfortable
The "lying on/against the direction of travel" position	
$C_{C_y}(t), C_{C_z}(t) < 0,2 \text{ m/s}^2$	Very comfortable
$0,2 \text{ m/s}^2 \leq C_{C_y}(t), C_{C_z}(t) < 0,3 \text{ m/s}^2$	Comfortable
$0,3 \text{ m/s}^2 \leq C_{C_y}(t), C_{C_z}(t) < 0,4 \text{ m/s}^2$	Borderline feeling
$0,4 \text{ m/s}^2 \leq C_{C_y}(t), C_{C_z}(t)$	Less comfortable
The "lying across the direction of travel" position	
$C_{C_y}(t), C_{C_z}(t) < 0,2 \text{ m/s}^2$	Very comfortable
$0,2 \text{ m/s}^2 \leq C_{C_y}(t), C_{C_z}(t) < 0,3 \text{ m/s}^2$	Comfortable
$0,3 \text{ m/s}^2 \leq C_{C_y}(t), C_{C_z}(t) < 0,4 \text{ m/s}^2$	Borderline feeling
$0,4 \text{ m/s}^2 \leq C_{C_y}(t), C_{C_z}(t)$	Less comfortable

To calculate the continuous ride comfort  $C_{C_x}(t)$ ,  $C_{C_y}(t)$  or  $C_{C_z}(t)$ , select the **Calculate Continuous Comfort Index** context menu item in the graphical window for the graph of longitudinal, lateral or vertical accelerations, Figure 1.262. As a result of the calculation, a new variable will be created. The indicators are calculated for each time interval of 5 seconds, Figure 1.263.

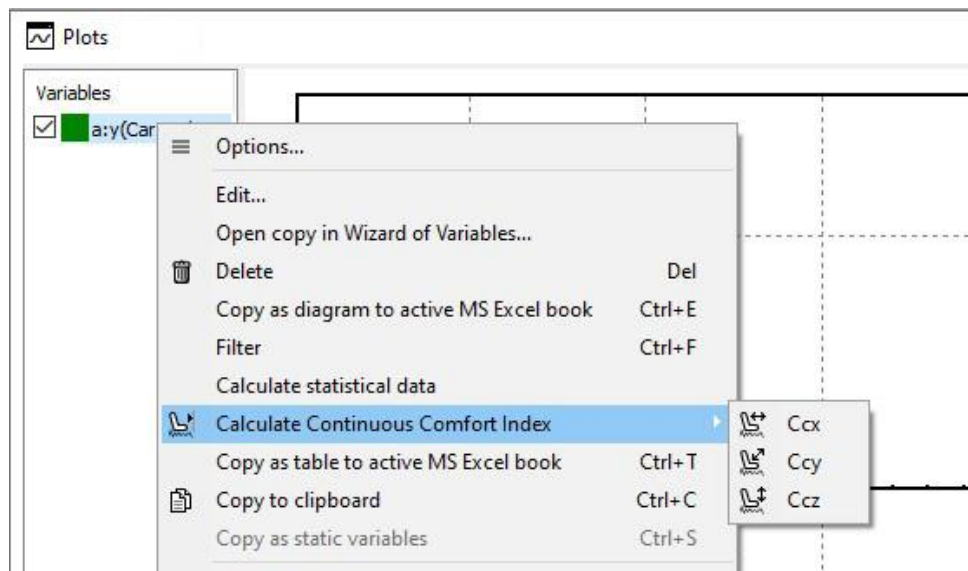


Figure 1.262. Continuous comfort index

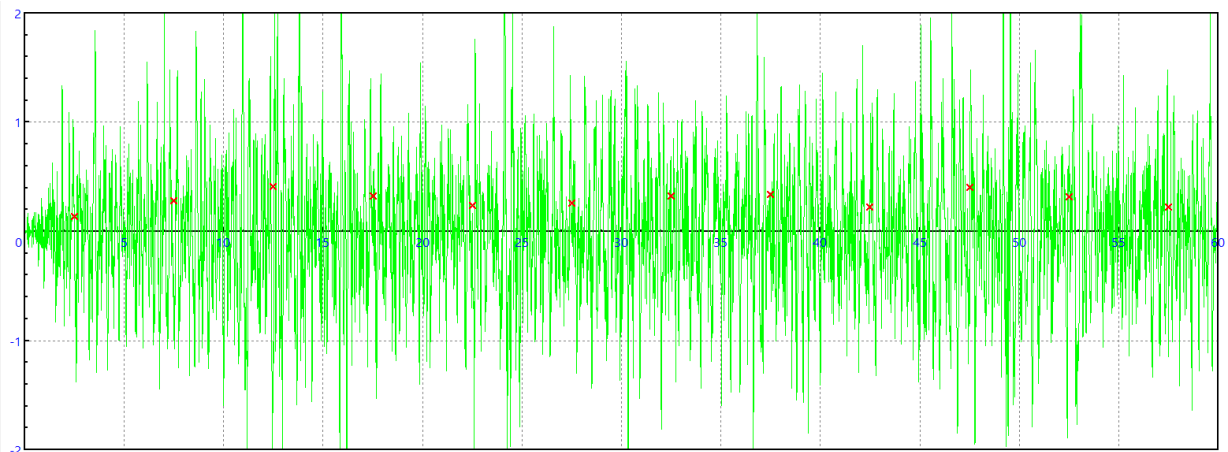


Figure 1.263. Continuous comfort index  $C_{C_y}(t)$

### 1.4.3.7.2. Ride comfort on curve transitions

This method concerns the assessment of ride comfort on curve transitions  $P_{CT}$  which is instantly perceived by passengers as a sudden change in the average feeling of comfort while the vehicle is moving. This type of assessment makes it possible to determine the effect of accelerations on passenger comfort when passing a separate curve, excluding the assessment of cumulative effects. The ride comfort index for passing the  $P_{CT}$  is calculated based on the lateral acceleration of the body  $\ddot{y}(t)$  and the roll rate of the body  $\dot{\phi}(t)$  of the vehicle. The measured signals are filtered using a low-pass filter  $W_p$  and averaged in a one-second window:

$$\begin{aligned} \ddot{y}_{1s}(t) &= \frac{1}{T} \int_{t-0.5T}^{t+0.5T} \ddot{y}_{W_p}(\tau) d\tau, \\ \dot{\phi}_{1s}(t) &= \frac{1}{T} \int_{t-0.5T}^{t+0.5T} \dot{\phi}_{W_p}(\tau) d\tau, \\ \ddot{y}_{1s}(t) &= \frac{1}{T} \left[ \dot{y}_{1s} \left( t + \frac{T}{2} \right) - \dot{y}_{1s} \left( t - \frac{T}{2} \right) \right], \end{aligned}$$

where  $\ddot{y}_{1s}(t)$  is the filtered value of the lateral acceleration of the body, averaged in a one-second window;

$\dot{\phi}_{1s}(t)$  is the filtered value of the body roll velocity, averaged in a one-second window;

$\ddot{y}_{1s}(t)$  is the filtered value of the rate of increase of lateral acceleration, averaged in a one-second window.

The ride comfort index is the percentage of passengers indicating discomfort that is calculated using the following formula:

$$P_{CT} = 100 \cdot \{ \max[0.2854 \cdot |\dot{y}_{1s}|_{max} + 0.2069 \cdot |\ddot{y}_{1s}|_{max} - 0.111; 0] + (3.64 \cdot |\dot{\phi}_{1s}|_{max})^{2.283} \},$$

where

$|\dot{y}_{1s}|_{max}$  is the maximum absolute value of the lateral acceleration  $\dot{y}_{1s}(t)$  in the curve, including the transition curves, plus 1,6 s;

$|\ddot{y}_{1s}|_{max}$  is the maximum absolute value of the rate of increase of lateral acceleration  $\ddot{y}_{1s}(t)$  in the curve, including transition curves;

$|\dot{\phi}_{1s}|_{max}$  is the maximum absolute value of the body roll velocity  $\dot{\phi}_{1s}(t)$  in the curve, including transition curves.

To calculate the comfort on curve transition  $P_{CT}$ , follow these steps.

1. For the considered model, calculate or load plots for the lateral acceleration of the body and the roll rate of the body in the curve.

2. Select the menu item **Tools | Ride comfort indexes | Ride comfort on curve transitions...**, Figure 1.264. After that, the **Calculator of comfort index on curve transitions** appears, Figure 1.265.

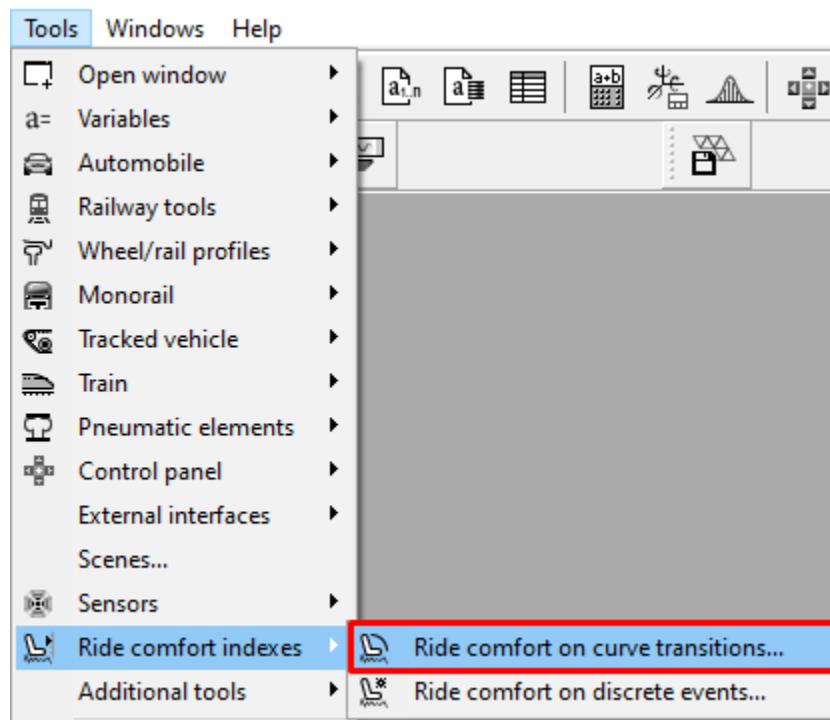


Figure 1.264. Ride comfort on curve transitions

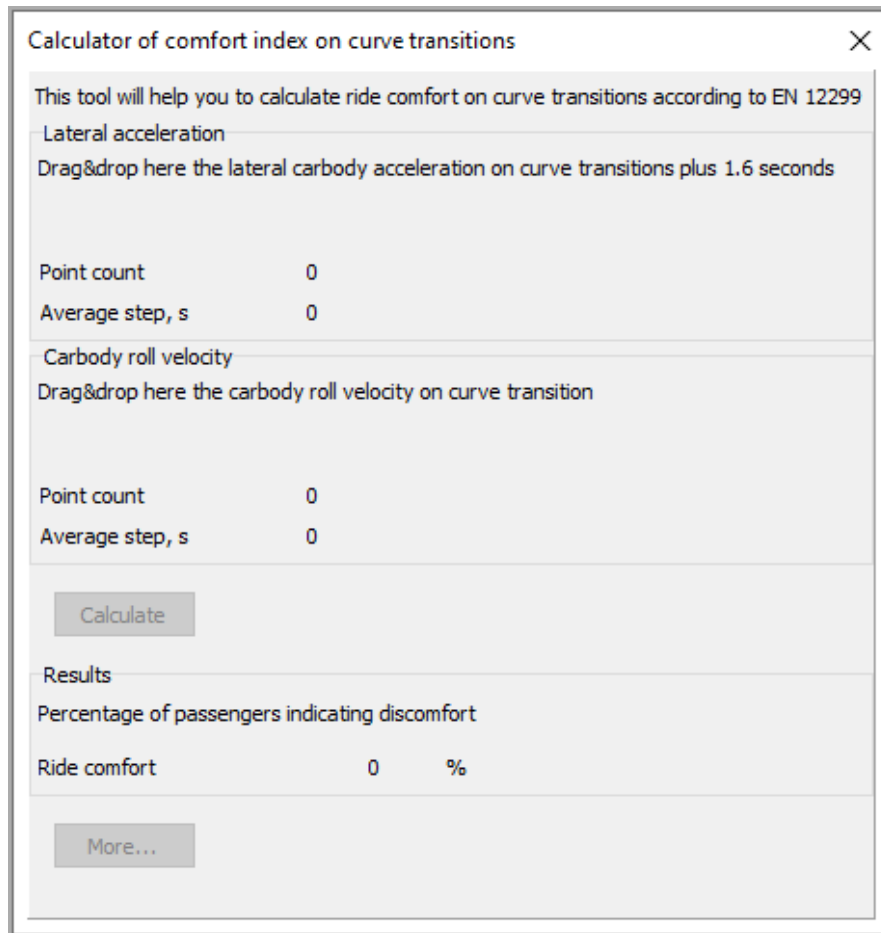


Figure 1.265. Calculator of comfort index on curve transitions  $P_{CT}$  window

3. In the graphical window, open the context menu and activate the **Show ruler**, Figure 1.266. After that, use the sliders of the ruler to select a section of the curve, including the transition sections, and plus 1.6 seconds for lateral acceleration.

**Remark.** The time period can be extended if it is obvious that acceleration reaches its maximum value after passing the end point of the curve plus 1.6 seconds.

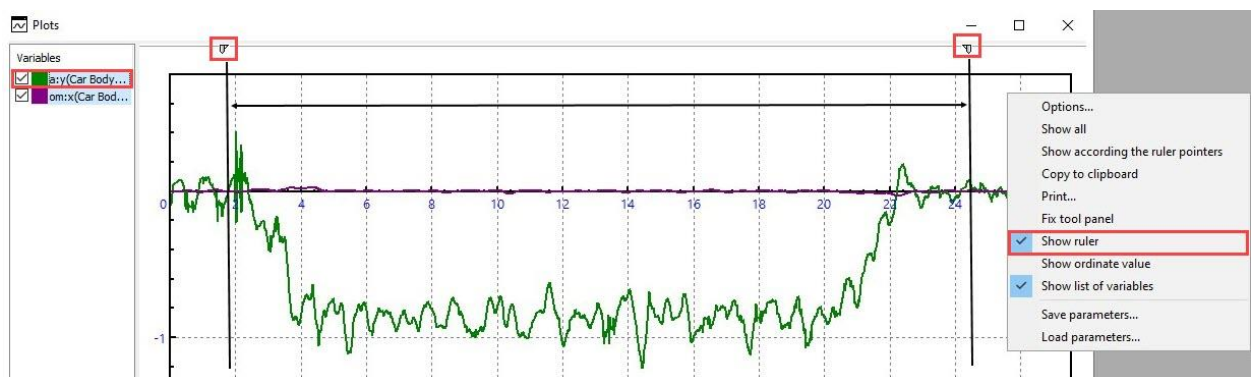


Figure 1.266. Selecting a section of the carbody lateral acceleration graph

4. Drag the lateral acceleration variable to the **Lateral Acceleration** calculator field, Figure 1.265.

5. In the graphical window, use the sliders of the ruler for the roll rate of the body to select a section of the curve, including transition curves, Figure 1.267, and drag the variable into the **Carbody roll velocity** calculator field. When loading graphs into the comfort indicator calculator, the name of the loaded variable, the number of points, and the average step are displayed in the fields of **Lateral acceleration** and **Carbody roll velocity**, figure 1.268.

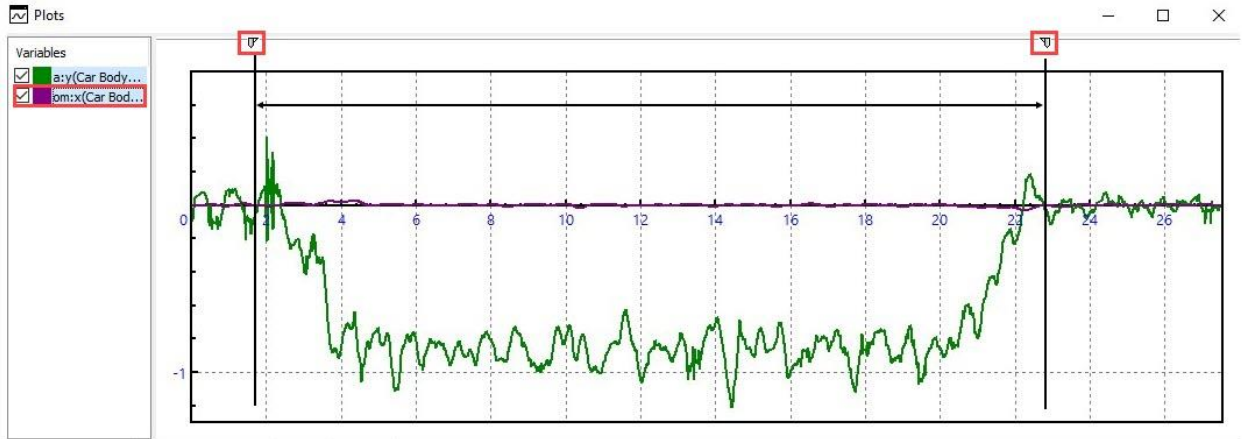


Figure 1.267. Selecting a section of the body roll velocity graph

6. After loading the graphs, click **Calculate**, Figure 1.268. The **Results** field will display the percentage of passengers indicating discomfort.

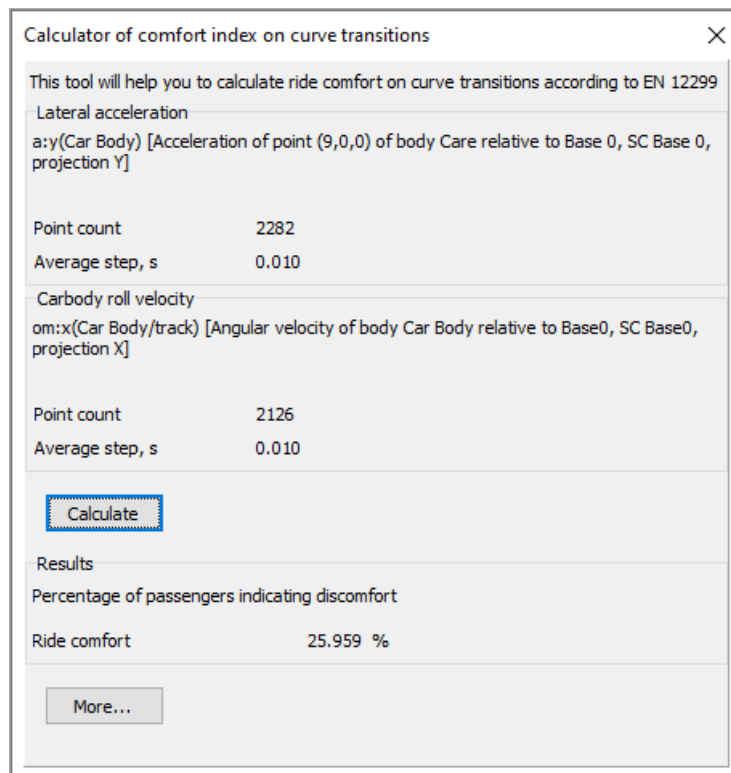


Figure 1.268. Calculator of comfort index on curve transitions the result window

7. By clicking the **More** button you can find the intermediate calculated values: maximal absolute value of lateral acceleration  $|\dot{y}_{1s}|_{max}$ , maximal absolute value of lateral jerk  $|\ddot{y}_{1s}|_{max}$ ,

maximal absolute value of car body roll velocity  $|\dot{\phi}_{1s}|_{max}$ , figure 1.269. The calculated intermediate filtered variables can be displayed in the graphical window, figure 1.270, figure 1.271.

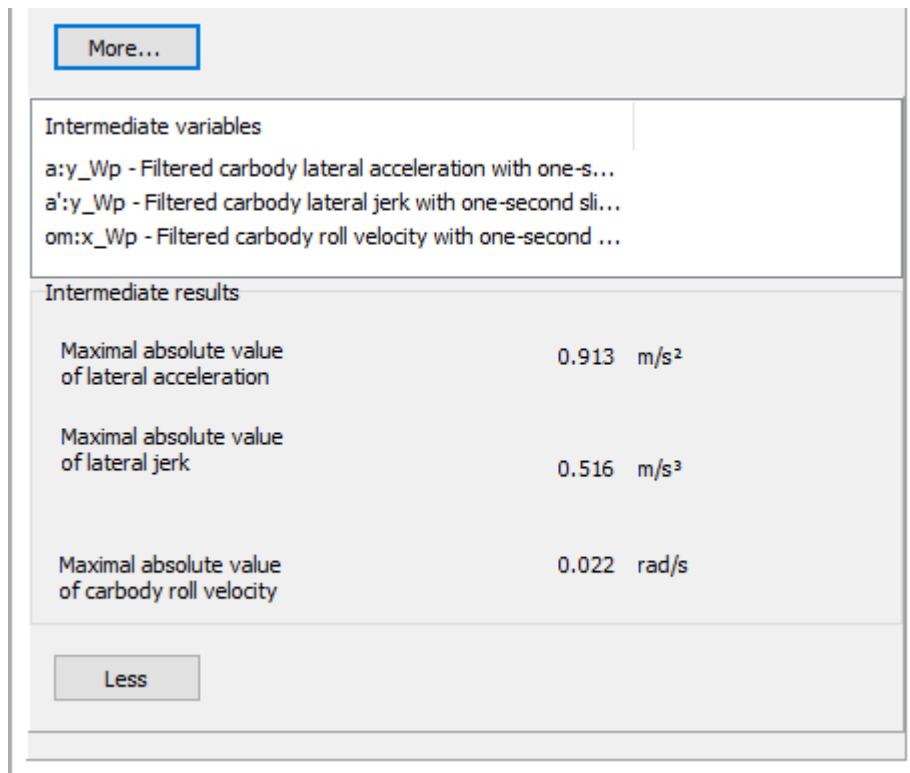


Figure 1.269. Intermediate result

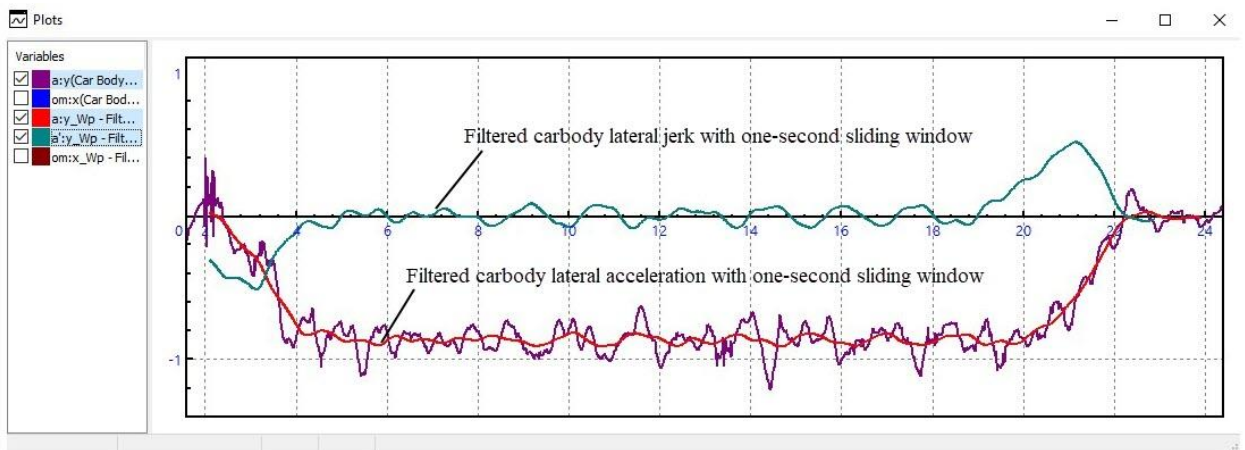


Figure 1.270. Filtered carbody lateral acceleration and filtered carbody lateral jerk, with one-second sliding window

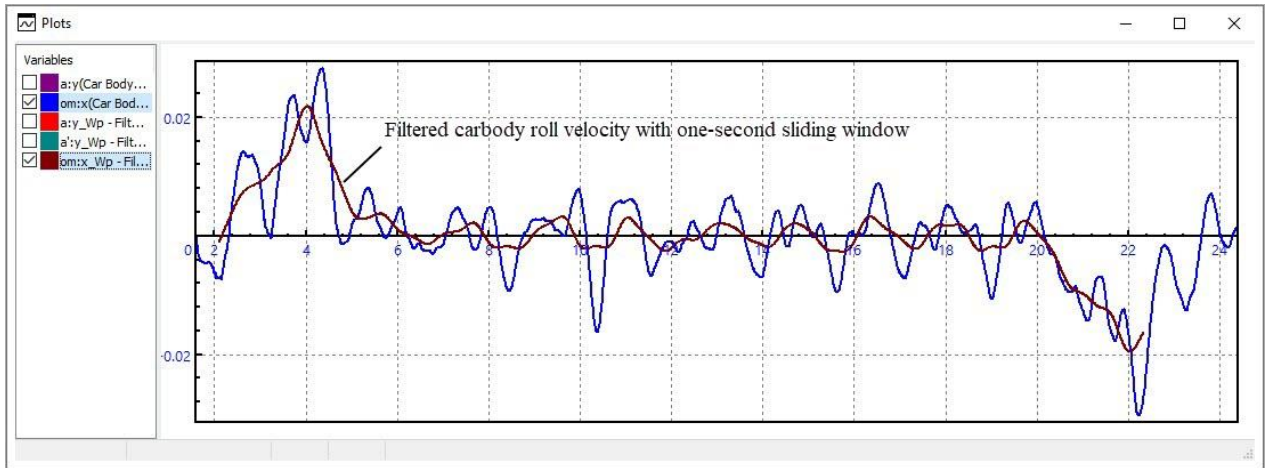


Figure 1.271. Filtered carbody roll velocity with one-second sliding window

### 1.4.3.7.3. Ride comfort on discrete event

Comfort on discrete event is discomfort due to a perceived transient oscillation. This method concerns measuring and evaluating the ride comfort on discrete event  $P_{DE}$ , which is instantly perceived by passengers as a sudden change in the feeling of comfort while the vehicle is moving. This method allows us to evaluate the effect of accelerations on the degree of comfort for passengers in a separate discrete event, excluding the assessment of cumulative effects.

The comfort on discrete event is calculated based on the longitudinal  $\ddot{x}(t)$ , lateral  $\ddot{y}(t)$  and vertical  $\ddot{z}(t)$  accelerations measured in the center of the carbody. The measured signals are filtered using a low pass filter  $W_d$  for longitudinal acceleration  $\ddot{x}_{W_d}(t)$ , a low pass filter  $W_p$  for lateral acceleration  $\ddot{y}_{W_p}(t)$  and a low pass filter  $W_b$  for vertical acceleration  $\ddot{z}_{W_b}(t)$ . The filtered acceleration values determine the total acceleration:

$$\ddot{a}^W(t) = \sqrt{\ddot{x}_{W_d}^2(t) + \ddot{y}_{W_p}^2(t) + \ddot{z}_{W_b}^2(t)}.$$

The total acceleration  $\ddot{a}^W(t)$  is averaged over a two-second window, the result of which is the dependence

$$|\ddot{a}_{2s}|(t) = \left| \frac{1}{T} \int_{t-0.5T}^{t+0.5T} \ddot{a}^W(\tau) d\tau \right|.$$

During a period  $T$  of 2 seconds, the oscillation range of the total acceleration is calculated as  $\ddot{a}_{pp}(t) = \max(\ddot{a}^W(\tau), \tau \in [t - 0.5T, t + 0.5T]) - \min(\ddot{a}^W(\tau), \tau \in [t - 0.5T, t + 0.5T])$ .

The comfort on discrete event  $P_{DE}$  indicates the percentage of passengers indicating discomfort and it is calculated using the formula:

$$P_{DE}(t) = 100 \cdot \max(0.0846 \cdot \ddot{a}_{pp}(t) + 0.1305 \cdot |\ddot{a}_{2s}|(t); 0).$$

**Remark** Ride comfort on discrete event  $P_{DE}$  cannot be calculated if the number of points or the step-size of the input signals do not match. The step should not exceed 0.1 seconds.

To calculate the comfort on discrete event  $P_{DE}$ , follow steps below.

1. For the considered UM model, calculate or load ready-made plots of the longitudinal, lateral and vertical accelerations of the carbody of the same length into the graphical window. If you need to explore a particular section, use the **Ruler** tool.

2. Select the main menu item **Tools | Ride comfort indexes | Ride comfort on discrete events**, Figure 1.272. After that, the **Calculator of comfort index on discrete events** appears, Figure 1.273.

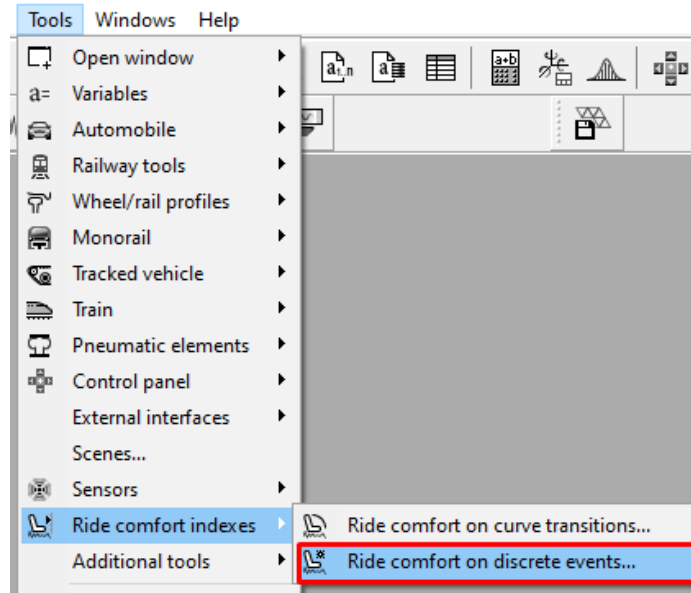


Figure 1.272. Ride comfort on discrete events

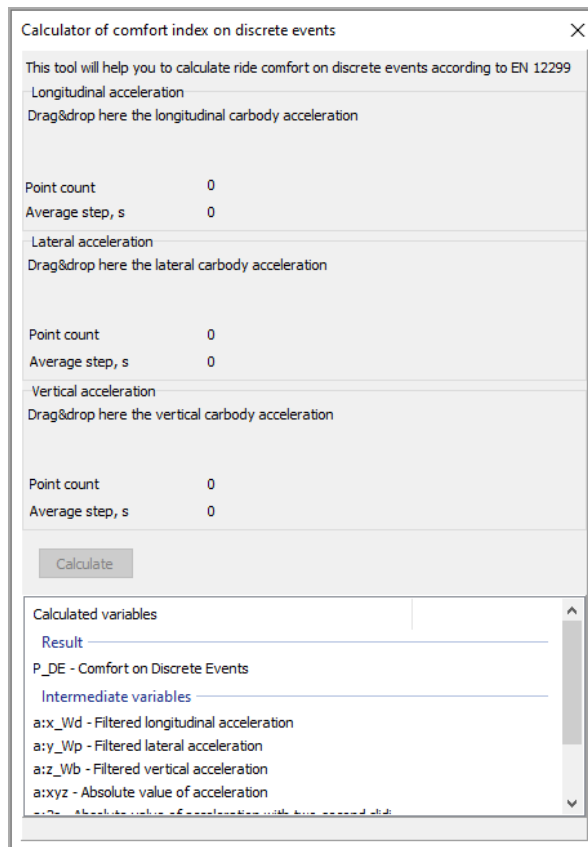


Figure 1.273. Calculator of comfort index on discrete events  $P_{DE}$  window

3. Drag the longitudinal acceleration variable to the **Longitudinal acceleration** field. Similarly, drag the plots of lateral and vertical acceleration into the corresponding fields.

4. Make sure that the number of points and the step-size of the signals match to each other, Figure 1.274.

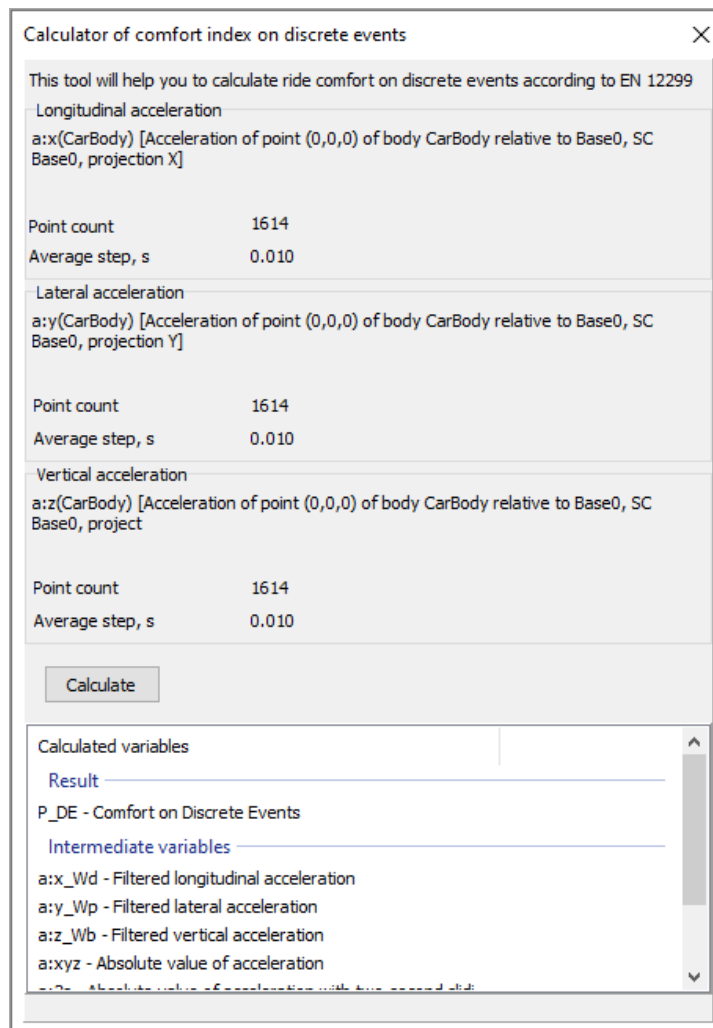


Figure 1.274. Calculator of comfort index on discrete events  $P_{DE}$  window with loaded signals

5. When you drag&drop all input signals (longitudinal, lateral and vertical accelerations), click **Calculate** button, Figure 1.274. After that, drag the **P\_DE** variable from the **Calculated variables / Result** group to the graphical window and estimate the comfort index, Figure 1.275.

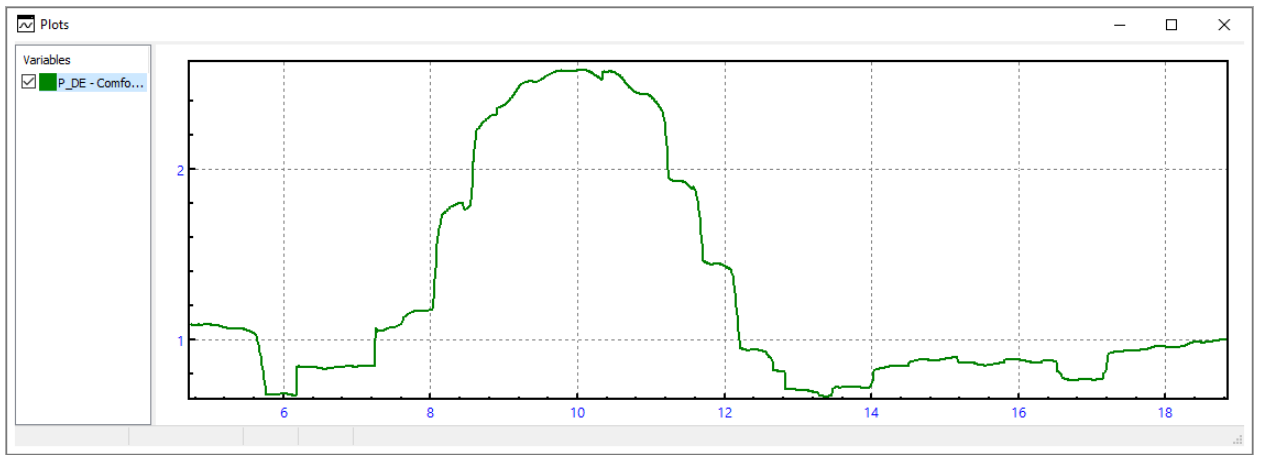


Figure 1.275. The comfort index  $P_{DE}$ , calculated based on acceleration on a straight section of the road with small irregularity

If necessary, you can view the intermediate calculated variables: filtered longitudinal  $\ddot{x}_{W_d}(t)$ , lateral  $\ddot{y}_{W_p}(t)$  and vertical  $\ddot{z}_{W_b}(t)$  acceleration, absolute value of acceleration  $\ddot{a}^W(t)$ , absolute value of acceleration with two-second sliding window  $|\ddot{a}_{2s}|(t)$  acceleration range with two-second sliding window  $\ddot{a}_{pp}(t)$ , Figure 1.273.

### 1.4.4. Solvers

Park solver in its two implementations is the only solver recommended for simulation of a rail vehicle dynamics, Figure 1.276 (Left panel): without use of multithread computation (**Park method**) and with use of multithread computation (**Park Parallel**). Choice of the solver is made by comparison of computational efficiency of these two solvers on the local computer for a rail vehicle model.

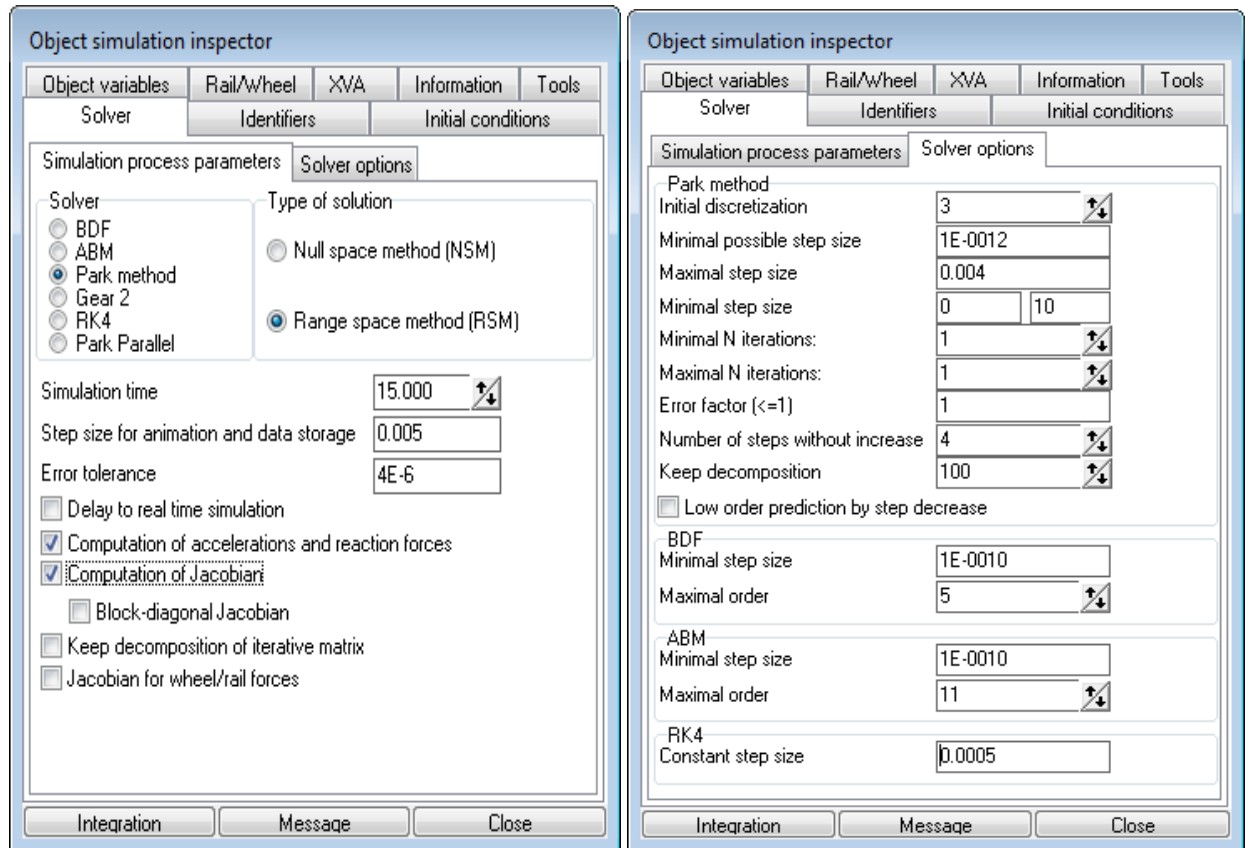


Figure 1.276. Park method parameters

Let us consider some general recommendations related to the choice of rational parameters of solvers.

The recommended value of the **error tolerance** in the case of Park solver is 4E-6...1E-7.

The **Delay to real time simulation** option is activated if simulation is faster than real time, and the user wants to see the motion in the real time in an animation window.

**Step size for animation and data storage** sets the time step size for refreshing animation windows, adding new points in graphic windows, and computing variables in the lists of variables. The recommended value is 0.005 s for rail vehicles. Note that the default value cannot be more than this value.

The recommended values of parameters for the Park solver are shown in Figure 1.276 (Right panel). For getting precise computation, it is recommended to set zero value for the **Minimal step size**.

**1.4.4.1. Solver: Park method**

Recommended values of parameters for the Park solver are show in Figure 1.276 right. For getting precise computation, it is recommended to set zero value for the **Minimal step** size.

If equations of motion are stiff and/or error tolerance is not small enough, the Park method shows instable solutions. As a rule, if a solver is instable, plots of some of accelerations include large oscillations with a high frequency. In such cases, it is recommended to do the follows:

- activate use of **Jacobian matrices (JM)**,
- set smaller **error tolerance**,
- set **Minimal step size** to zero.

Use of **Jacobian matrices** leads to a considerable acceleration of simulation process in the following cases:

- low speed of vehicle (less 8-10 m/s); in this case the **Jacobian for wheel/rail forces** key should be activated; *this method helps in the case of motion with a nearly constant speed*;
- vehicle model includes stiff forces, i.e. forces with large gradients due to big stiffness and damping coefficients; examples: contact forces, force element with a successive connection of spring and damper (the viscous-elastic force element).

If computation of **Jacobian matrices** is on, simulation process can be often made faster with the help of **block-diagonal Jacobians** and switching off computation of **Jacobian matrices** for non-stiff forces such as suspension springs or dampers on the **Tools | Forces** tab of the **Object Simulation Inspector**, Figure 1.277.

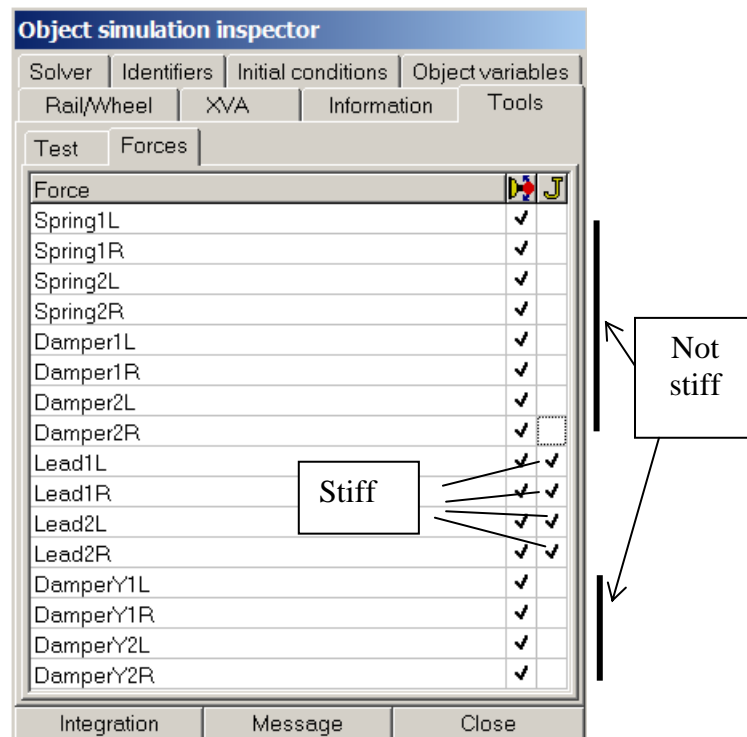


Figure 1.277. Switching on/off evaluation of JM for force elements

**Keep decomposition of iterative matrix** option is not used for simulation of rail vehicles.

**Remark.** The main criterion for use of **Jacobian matrices** is very simple. If simulation with this option is faster and stable, it should be used. It is recommended to optimize parameters and options of the solver for each of the rail vehicle models.

#### 1.4.4.2. Solver: Park Parallel

The new solver **Park parallel** is implemented since UM 9.0, Figure 1.280. In fact, this method is a combination of a special algorithm for numeric-iterative generation of equations of motion, and the Park solver for stiff ordinary differential equations. One of the main features of this method is use of multi-core processors for parallel computations, which make simulation faster in many cases.

##### 1.4.4.2.1. Conditions for use of Park Parallel solver

The **Park parallel** can be used if some requirements are met.

- Numeric iterative method of generation of motion equations must be set for the vehicle model in the **UM Input** program, Figure 1.278.
- Mass and moments of inertia relative to the X, Y, Z axes for all of the bodies must be positive.
- Using **Park Parallel** for simulation of hybrid models, which include flexible bodies (UM FEM) and interfaces with Matlab\Simulink is not supported.
- 3D contact elements are ignored (see [Chapter 3](#) of the user's manual).

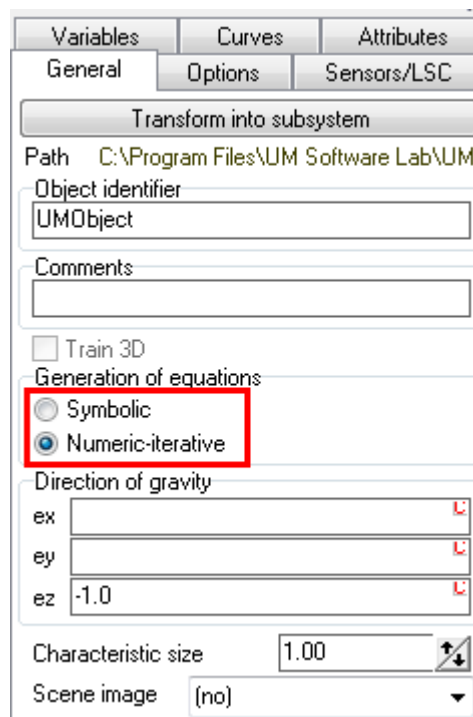


Figure 1.278. Numeric-iterative method of generation of equations; the **Object** tab of the inspector in **UM Input** program

If the model does not satisfy these requirements, the corresponding message appears, and the list of violations is stored in a text file, Figure 1.279.

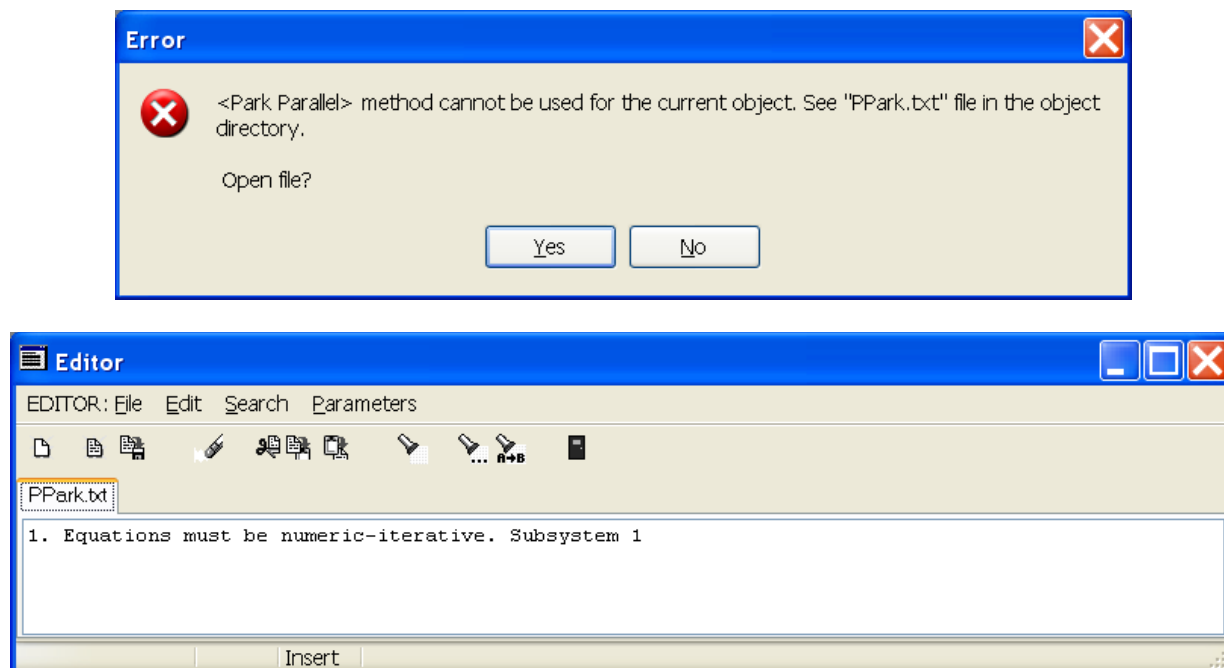


Figure 1.279. Message about impossibility of use the Park Parallel, and list of violations

#### 1.4.4.2.2. Solver parameters

##### Solution method

**BDJ** is the use of Jacobi block-diagonal matrices (Block Diagonal Jacobians).

**CGM** is iteration of Conjugate Gradient Method. The **CGM** is used for more precise solving stiff equations. If **Solution method** is set to **CGM**, the full JM are used otherwise the block-diagonal JM are applied. There are no exact recommendations for use of CG iterations. The user should follow the simple method: the key is activated if the simulation becomes faster.

If the CG iterations are used, the **CGM error** must be specified. As a rule, results are good even for low accuracy 0.1.

##### Use of threads

Threads can be used for computers with multi-core processors. The number of threads for parallel computations cannot exceed the maximal number of processor cores (physical and logical). Optimal number is determined by the experience. Nowadays, four- and eight-core processors are the most efficient. In some cases, use of parallel computations improves the solver performance by factor 2.5-3, see Sect. 1.4.4.3. *"Comparison of solver performances in simulation of freight car with three-piece bogies"*, p. 1-232.

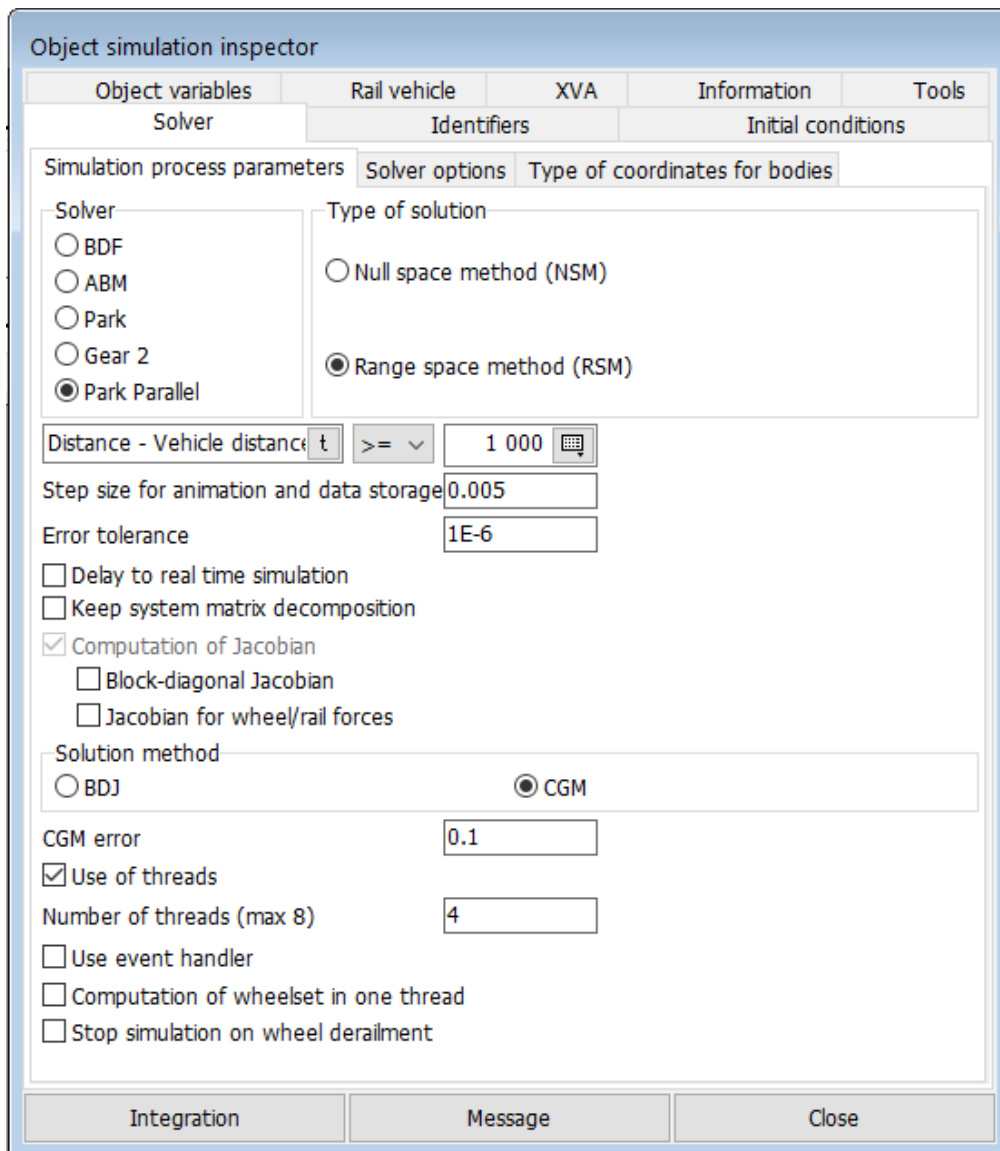


Figure 1.280. Parameters of solver Park Parallel

### 1.4.4.3. Comparison of solver performances in simulation of freight car with three-piece bogies

Consider an example of simulation of an empty hopper with three-piece bogies 18-100, Figure 1.281. The advanced model of a freight car is used compared to simplified one delivered with UM ([UM Data\SAMPLES\Rail\\_vehicles\simple\\_18\\_100](#)). The advanced model includes friction wedges as rigid bodies with six degrees of freedom. The model has 114 d.o.f., and it is accepted by Russian Railways as the standard by safety problem analysis [15].

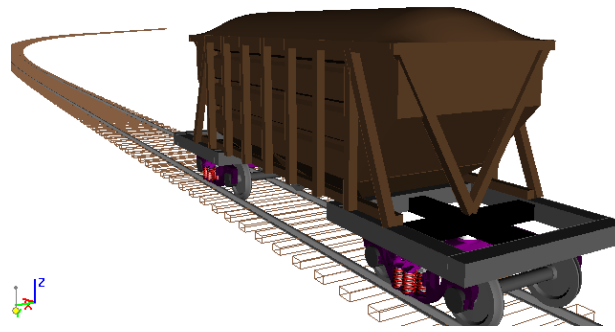


Figure 1.281. Model of hopper

**Remark.** The model of a freight car [{UM Data}\SAMPLES\Rail\\_vehicles\simple\\_18\\_100](#) uses programming in UM environment for evaluation of friction wedge forces, and cannot be simulated with the Park Parallel solver.

Consider simulation of motion of the hopper in curve  $R=600$  m (Figure 1.282) speed 20 m/s, simulation time 15 s. Solver parameters and options correspond to Figure 1.276.

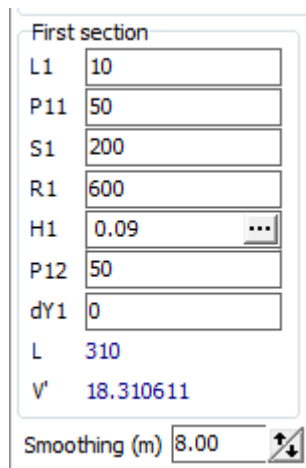


Figure 1.282. Parameters of curve

Comparison of solver performance for different forms of generation of equations of motion and variants of use of JM is summarized in Table 1.5. Computations were made on the four-core processor (all cores are physical): Intel Core 2 Quad CPU Q6700 @ 2.67 GHz

Table 1.8

**Comparison of solver performances**

Solver	Number of threads	CPU time, s		
		Without use of JM	With use of JM	Block-diagonal JM
Park method Symbolic equation	1	40.7	16.1	15.4
Park method, numeric-iterative equations	1	-	20.4	23.2

Park Parallel	1		8.4	11.8
	2		5.6	7.6
	3		4.5	6.1
	4		3.7	5.0

The following conclusions can be drawn from these results.

1. The model is stiff because the use of JM makes the simulation much faster. This conclusion is quite evident; the model includes a lot of contacts with high stiffness.
2. If the **Park method** is used as a solver, generation of equations of motion in symbolic form is recommended. The user must set the corresponding option in Figure 1.283 and follow the instructions of [Chapter 3](#), Sect. *Compilation of equations of motion*.
3. Sometimes use of block-diagonal JM is recommended.
4. The **Park Parallel** solver in this case is much faster than the Park method, even if the parallel computation are not used
5. Use of multithread computations improves the performance up to factor 2.3.
6. For the considered model, the use of block-diagonal JM (CG iterations is not active, Figure 1.280) in **Park Parallel** solver makes simulation slower. Other models of rail vehicles may show an opposite result.
7. The main conclusion is as follows. The combination of the new solver and a computer with a high performance leads to efficient simulation of the freight car, which is four times faster than real time!

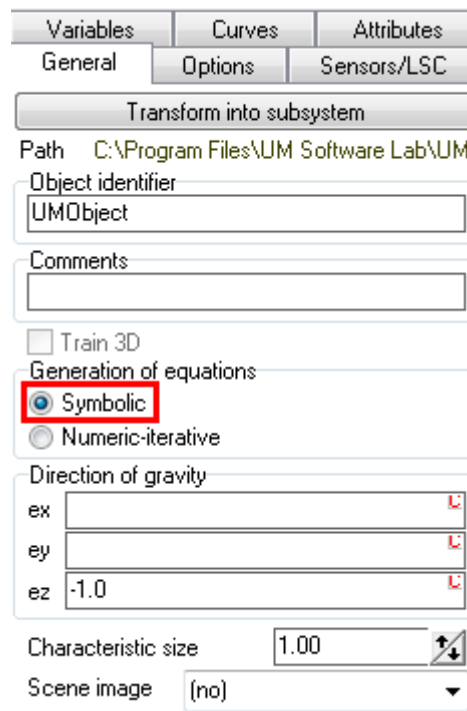


Figure 1.283. Generation of equations in symbolic form

**Remark.** In the case of scanning projects, the parallel computations are not used because a simultaneous run of several applications makes the total simulation faster (see the manual to UM Cluster, [Chapter 6](#)).

## 1.5. Simulation of gear (cog) rail trains

### 1.5.1. General information

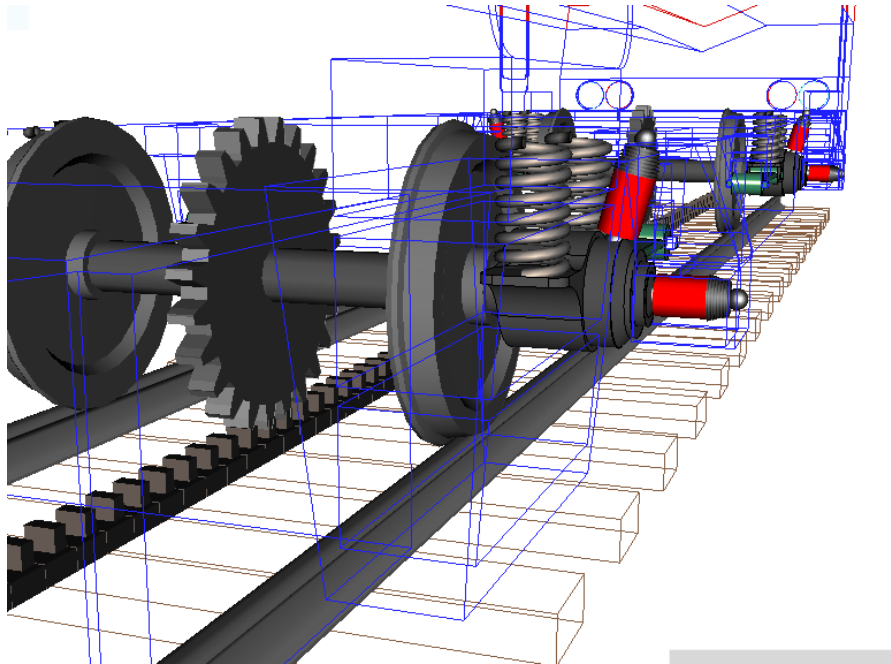


Figure 1.284. Example of cog rail track

In these sections, we consider some features for simplified simulation of cog rail vehicles and trains, Figure 1.284. The following features and simplification should be taken into account.

1. The generalization of the rack-pinion force element is used for the train propulsion, that is, the real contact interaction between the teeth of the gear wheel and the rack is not taken into account.

2. When modeling, the cog rail is taken into account over the entire simulation interval or it is optionally disabled completely. That is, there is no way to run on the rack and move out from it during the simulation.

3. The powertrain for the propulsion system can be modelled either simplified by the direct application of the control torque to the gear wheels as it is described below, or the UM Driveline module can be used for an advanced modeling of the propulsion system.

4. The UM Simulation program detects a cog rail train if the model includes one or several force elements for interaction between the cog rail and gear wheels, see Section 1.5.3 *Force element for cog rail train propulsion*.

5. In the current UM version, the cog rail image in UM Simulation program is visible in the old animation window (Figure 1.284) and not available in the new one.

An example of the simplified model of a cog rail vehicle can be found in the directory [{UM Data}\SAMPLES\Rail\\_Vehicles\as4 gear track](#)

### 1.5.2. Gear wheels for propulsion system and speed control

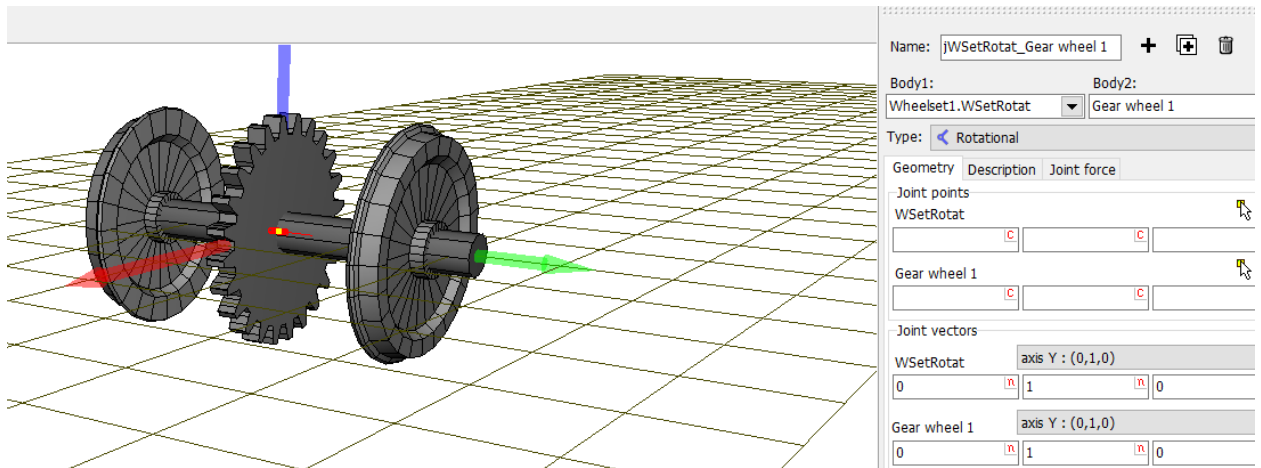


Figure 1.285. A joint connecting wheelset and gear wheel

For modeling the propulsion system, the vehicle or train model must include a number of gear wheel interacting with the cog rail. Each of the gear wheel is usually connected with a wheelset by a rotational joint, Figure 1.285.

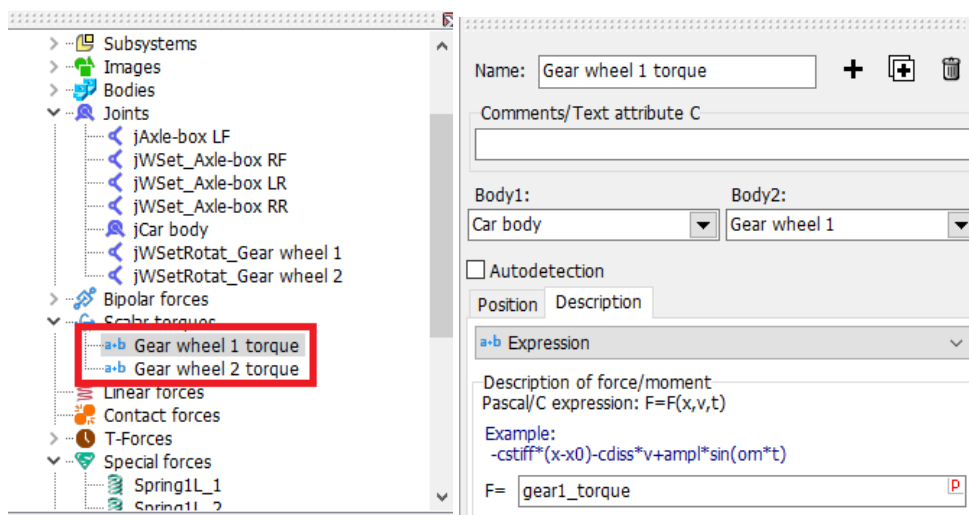


Figure 1.286. Propulsion modeling by scalar torque

For simplified modeling the propulsion, it is recommended to introduce a scalar torque of the **Expression** type for each of the gear wheel, Figure 1.285. In the above example, the torque is parameterized by the identifier *gear1\_torque*. The identifiers are used for speed control of the vehicle.

Consider a PI speed control, which uses these torque identifiers. Such type of the control described below is implemented in the model [{UM Data}\SAMPLES\Rail\\_Vehicles\as4\\_gear track](#). The PI control is

$$M = K(v^* - v) + K_I \int_0^t (v^* - v) dt$$

Its implementation in the model “AS4 gear track” can be found in the UM Input program on the **Object | Variables** tab, Figure 1.287.

Type	Name	Expression
var	vx_carbody	vx( "Car body.cg",, "Car body.LSC" , )
var	dv	v_desired-vx_carbody
dy/dt=v:	Idv	integral(dv)
var	pi_speed_control	sck*dv+sckT*Idv

Figure 1.287. Programming of speed control variable

Here are some comments to the implantation of the speed control.

1. The variable  $vx\_carbody$  is the longitudinal speed of the car body center of gravity. It is defined as a projection of the vector of velocity on the X axis of the system of coordinates connected with the car body.
2. The variable  $dv$  is the difference between the desired speed specified by the identifier  $v\_desired$ , and the vehicle speed.
3. The variable  $Idv$  is the integral of the speed difference.
4. The variable  $pi\_speed\_control$  is the computed control torque, which should be assigned to the identifiers, parameterizing the torque ( $gear1\_torque$  and  $gear2\_torque$  in the model).

**Remark.** The simplified PI speed control can be instable for big values of the parameter  $K$ . Decrease its value to get a stable motion of the vehicle or train.

Assignment of the torque control is done in the UM Simulation program basing on the **Identifier control** tool, Figure 1.288. Double click on the line with the identifier assignment to get the window with the corresponding identifier assignment, Figure 1.289.

Identifier	Comment
<input checked="" type="checkbox"/> v_desired	
<input checked="" type="checkbox"/> gear1_torque	
<input checked="" type="checkbox"/> gear2_torque	

Figure 1.288. Speed control assignments

The desired speed identifier is equal to the speed identifier  $v0$ , Figure 1.289, left. The  $pi\_speed\_control$  variable is assigned to the scalar torque identifiers  $gear1\_torque$  and  $gear2\_torque$ . This method allows the user to control both the constant speed and the speed specified by a time function, Figure 1.290.

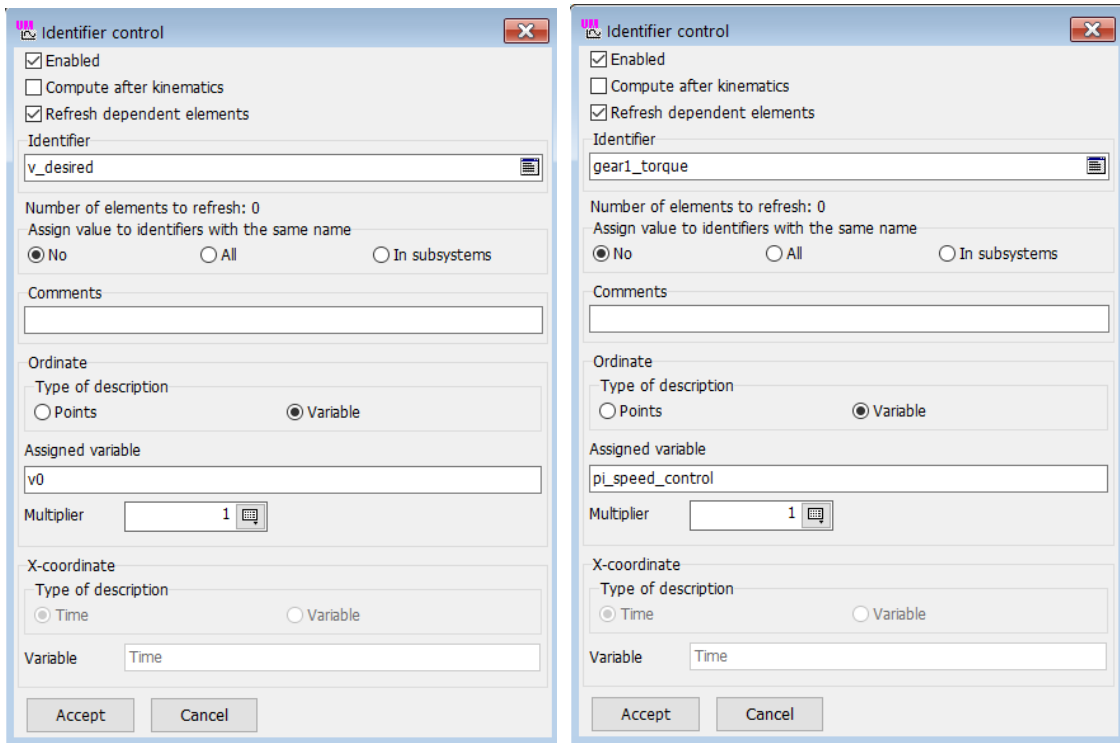


Figure 1.289. Identifier assignment window

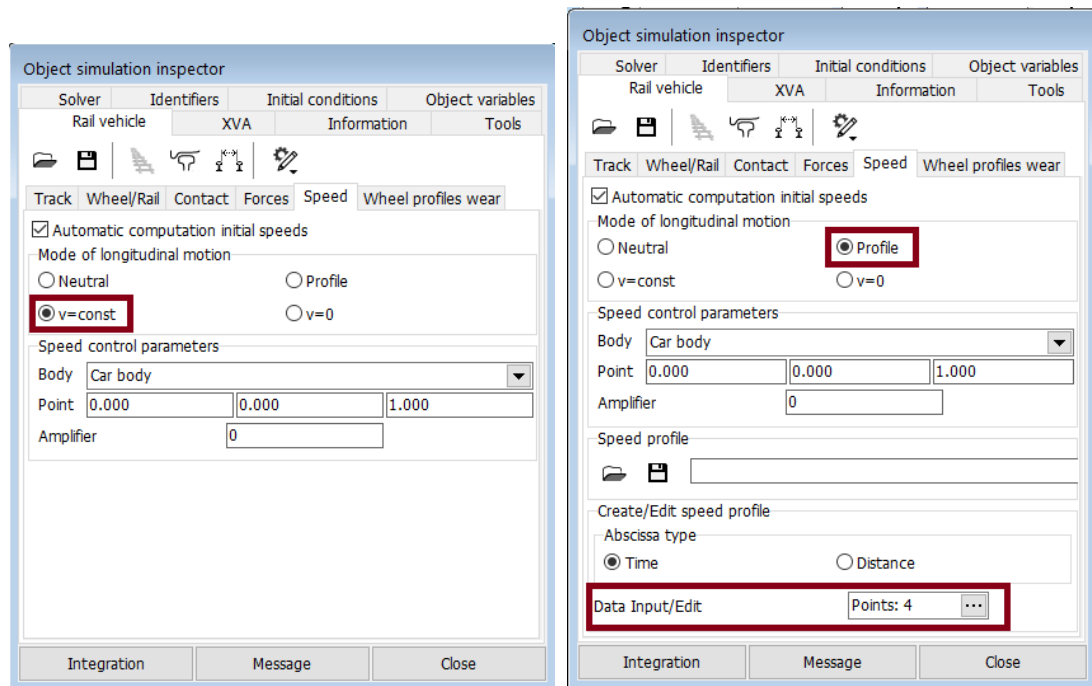


Figure 1.290. Motion with constant (left) and variable (right) speeds

### 1.5.3. Force element for cog rail train propulsion

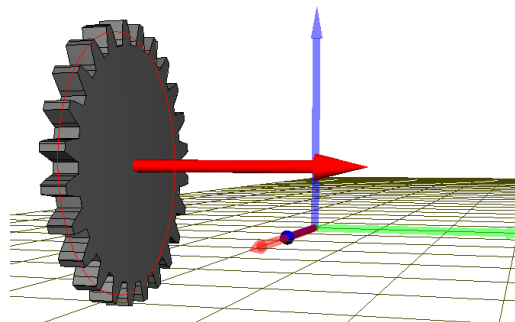
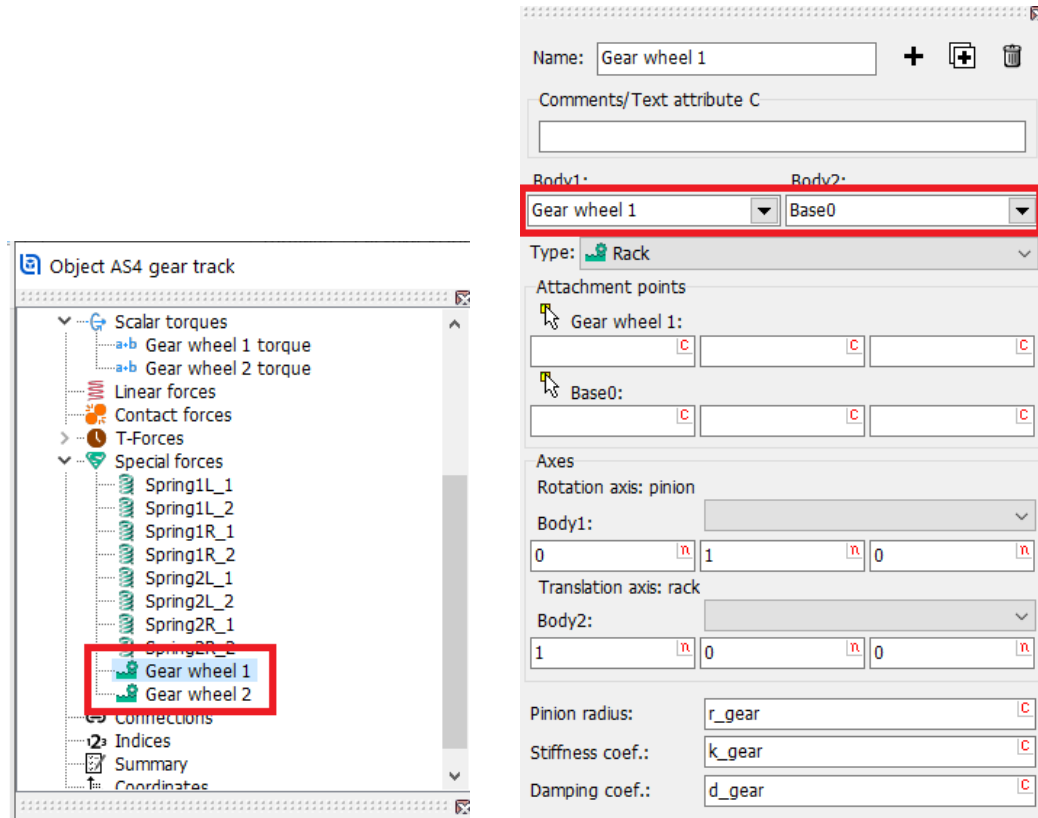


Figure 1.291. For element for interaction between gear wheel and cog rail

Model of the propulsion system must include of a number of special force element of the **Rack** type for each of the propulsion gear wheels, Figure 1.291, see [Chapter 3](#), Section *Rack and pinion*. The first body in the element must be a gear wheel the second one must be **Base0**.

During the simulation, the force vector is directed parallel to the tangent to the cog rail line coinciding with the track centerline, Figure 1.292. The force acts on the gear wheel at the point corresponding to the given pinion radius in the force description (Figure 1.291).

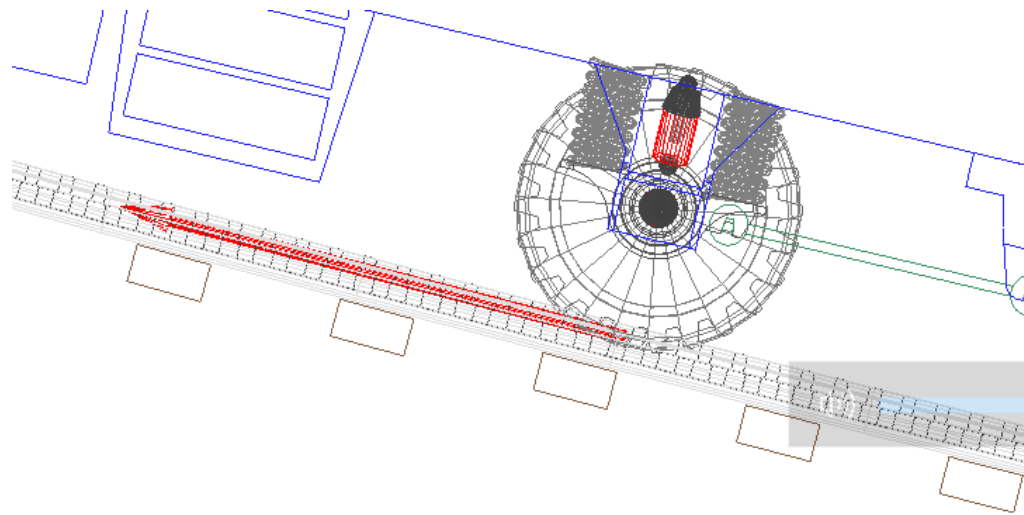


Figure 1.292. Propulsion force vector acting on the gear wheel

### 1.5.4. Cog rail image

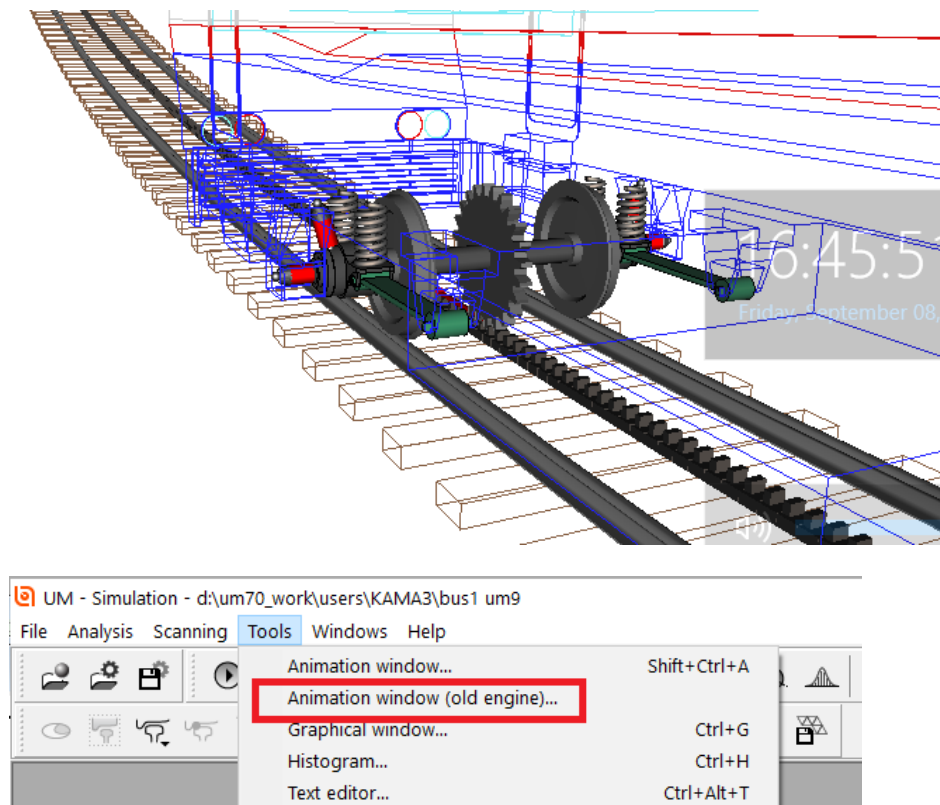



Figure 1.293. Cog rail image in the old animation window

The cog rail image is visualized in the old animation window using a template file \*.img created by the user in the UM Input program. The template file is the image of a short part of the cog rail containing only one cog and oriented along the X axis. For example, the template *Rack Cog.img* used in the model [{UM Data}\SAMPLES\Rail Vehicles\as4 gear track](#) includes a graphic object consisting of two boxes oriented along X axis, Figure 1.294. The bigger box corresponds to the section of the rail, and the smaller one represents the cog. To develop a template,

create an own GO and save it to a file \*.img by the button . It is recommended to locate the cog top in the origin of the GO like in Figure 1.294.

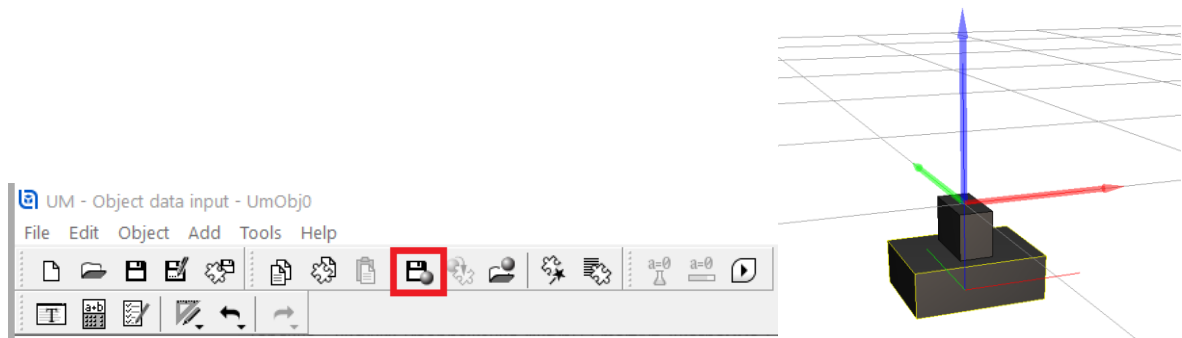



Figure 1.294. Cog rail template

The cog rail template should be assigned to the model in the UM Simulation program, Figure 1.295. Open the **Rail vehicle | Track | Gear rail** tab and use the button  to assign the template. On the same tab, the user can set the distance between the cogs (**Gear step** parameter) and well as the vertical position of the cog rail relative to the track system of coordinates (**Image shift Z**).

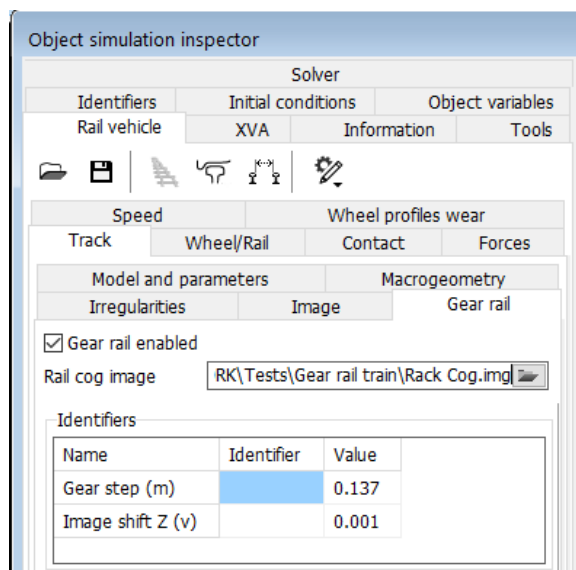


Figure 1.295. Assignment of cog rail template and the cog rail geometrical parameters

### 1.5.5. Simulation of vehicle on a track with cog (gear) rail

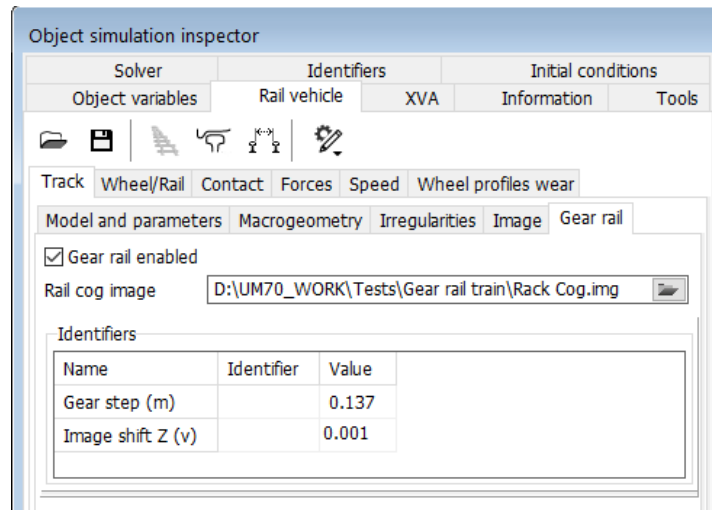


Figure 1.296. Tab with cog rail parameters

Simulation of a vehicle on a cog rail track can be run in two modes.

- **Gear (cog) rail**

To run the simulation in this mode, the following steps should be done.

- Set the **From file** type of the track description and assign a macrogeometry file, Figure 1.297.
- Assign a file with the cog rail image template and set the geometry parameters, Figure 1.296, see Sect. 1.5.4 *Cog rail image*.
- Check the **Gear rail enabled** option.
- Assign the v0 identifier to the desired speed as well as the control torque variable to the corresponding identifiers, Sect. 1.5.2 *Gear wheels for propulsion system and speed control*.
- Set the speed control parameter values, Sect. 1.5.2 *Gear wheels for propulsion system and speed control*.

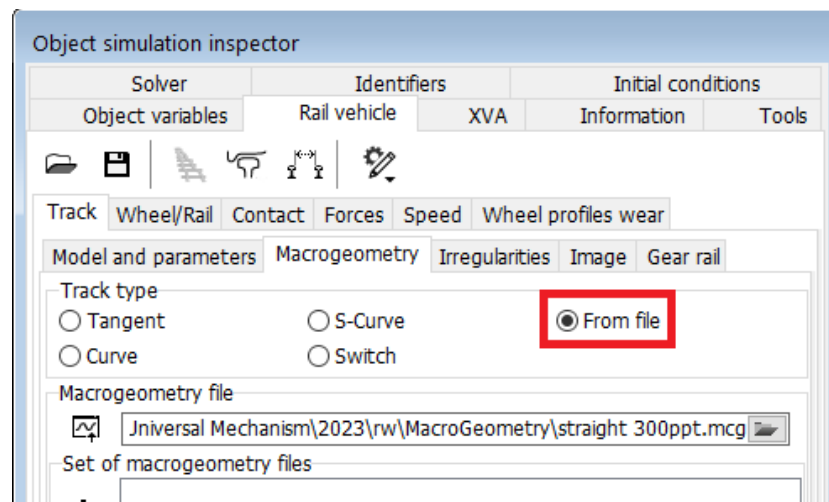


Figure 1.297. Type of track description

- **Standard rails**
- The option **Gear rail enabled** is not checked (Figure 1.296) or the track type does not set as **From file** (Figure 1.297).
- The control torque assignment in Figure 1.288 must be disabled, the corresponding identifiers (*gear1\_torque* and *gear2\_torque* in our example) must have zero values.

**Remark.** The simplified PI speed control described in Sect. 1.5.2 can be instable for big values of the parameter  $K$ . Decrease its value to get a stable motion of the vehicle or train.

## 1.6. Scanning projects with models of rail vehicles

Here we suppose that the **UM Experiments** module is available in UM configuration, Figure 1.298.

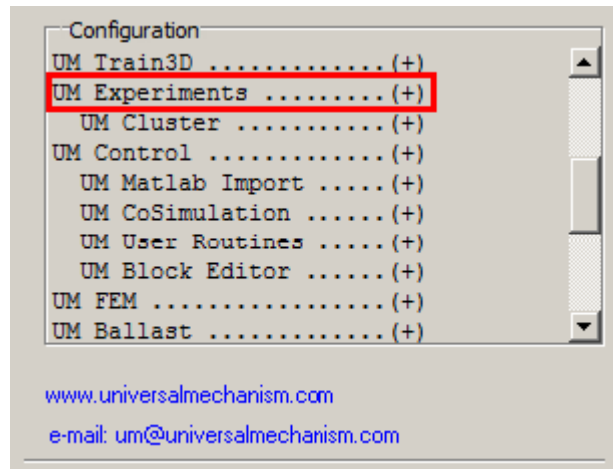


Figure 1.298. List of available UM modules

### 1.6.1. List of internal identifiers parameterizing operation conditions of rail vehicles

Analysis of the rail vehicle dynamics with **UM Experiments** module has some features, that we discuss in these section. To improve the efficiency of scanning projects with rail vehicle, standard **internal identifiers** for operation conditions of a rail vehicle are introduced (macro-geometry, irregularities, profiles of wheels and rails end so on). These identifiers can be used for variation of the corresponding conditions in numeric experiments.

The list of internal identifiers is available on the **Alternatives | Hierarchy of parameters** tab, the **List of parameters** group, the **RVPParameters (Rail Vehicle Parameters)** branch, Figure 1.299.

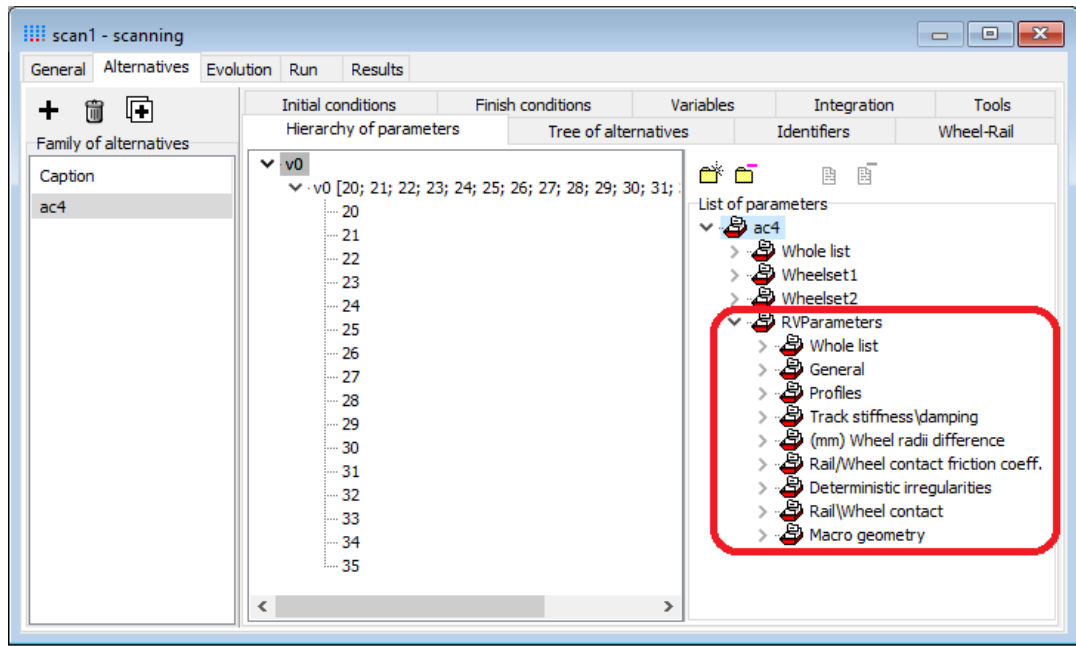


Figure 1.299. List of railway-related identifiers

The full list of parameters is collected in Table 1.6.

Table 1.9

**List of standard internal identifiers**

Branch of identifiers tree	Name of identifier	Comments
General	_irr_type	Type of irregularities. 0: no irregularities, 1: file, 2: deterministic, Sect. 1.4.2.4.3. <i>"Assignment of track irregularities"</i> , p. 1-158.
	_irr_group	Number of irregularity group starting with 1, Sect. 1.4.2.4.3.1. <i>"Irregularities from file. Group of irregularities"</i> , p. 1-158.
	_scr_scw_distance	(mm), distance between SC rail and SC wheel. The parameter specifies gauge, Sect. 1.2.1.1. <i>"Geometry of rails in an ideal track"</i> , p. 1-66.
	_y_irr_factor	Scaling factor for lateral file irregularities, Sect. 1.4.2.4.3.1. <i>"Irregularities from file. Group of irregularities"</i> , p. 1-158.
	_z_irr_factor	Scaling factor for vertical file irregularities, Sect. 1.4.2.4.3.1. <i>"Irregularities</i>

		<i>from file. Group of irregularities"</i> , p. 1-158.
	_rail_inclination	Rail inclination, Sect. 1.2.1.1. " <i>Geometry of rails in an ideal track</i> ", p. 1-66.
Profiles	_i_wheel_profile	Number of wheel profile in the list starting with 1, Sect. 1.4.2.3.2 " <i>Assignment of wheel profiles</i> ", p. 1-146.
	_i_rail_left_profile	Number of left rail profile in the list starting with 1, Sect. 1.4.2.3.1. " <i>Assignment of rail profiles</i> ", p. 1-145.
	_i_rail_right_profile	Number of right rail profile in the list starting with 1, Sect. 1.4.2.3.1. " <i>Assignment of rail profiles</i> ", p. 1-145.
Track stiffness/damping Sect. 1.4.2.4.1.1. <i>"Track stiffness and damping"</i> , p. 1-151.	_z_rail_stiffness	(N/m) Vertical rail stiffness constant
	_y_rail_stiffness	(N/m) Lateral rail stiffness constant
	_z_rail_damping	(Ns/m) Vertical rail damping constant
	_y_rail_damping	(Ns/m) Lateral rail damping constant
	_torsional_rail_stiffness	(Nm/rad) Torsional rail stiffness constant
Wheel radii difference, Sect. 1.4.2.7.2. <i>"Wheel radii difference"</i> , p. 1-184.	_dr_wheel_[n]l _dr_wheel_[n]r	(mm) Radius differences for left and right wheels, wheelset n
Rail/wheel contact friction coefficients, Sect. 1.4.2.5.4. <i>"Setting coefficients of friction in rail/wheel contact"</i> , p. 1-170.	_cfriiction_left	Running surface of left rail
	_cfriiction_right	Running surface of right rail
	_cfriiction_left_s	Side face of left rail
	_cfriiction_right_s	Side face of right rail
Deterministic irregularities, Sect. 1.4.2.4.3.2. <i>"Deterministic irregularities"</i> , p. 1-160.	_type_z	Type of vertical irregularity. 0: $h/2(1-\cos x)$ ; 1: $h \sin x $
	_l_z	Wave length of vertical irregularity L
	_h_z	Height of vertical irregularity H
	_s0_z	Start of left vertical irregularity
	_ds0_z	Shift of the right vertical irregularity
	_count_z	Number of waves of vertical irregularity N
	_l_y	Wave length of lateral irregularity L
	_h_y	Height of lateral irregularity H
	_s0_y	Start of left lateral irregularity
	_count_y	Number of waves of lateral irregularity N
Rail/wheel contact, Sect. 1.4.2.5.	_contact_model	Model of creep forces. 0: simplified

<p>"Parameters for computation of rail-wheel contact forces", p. 1-167</p>		<p>1: Mueller, 2: FastSim, 3: FastSimA, 4: Minov</p>
	_sc_eq_conicity	Equivalent conicity $\lambda$
	_sc_eq_cont_ang_par	(rad/m) Equivalent contact angle parameter
	_sc_cont_ang0	(deg.) Nominal contact angle $\beta_0$
	_sc_dy2_max	(mm) Maximal lateral displacement $y^*$
	_sc_cont_ang2	(deg.) Flange contact angle $\beta^*$
	_sc_y2	(mm) Y coord. of flange contact $y_r^*$
	_sc_z2	(mm) Z coord. of flange contact $z_r^*$
	_fastsim_nstrips	FastSim : number of strips
_fastsim_nelem	FastSim : number of elements	
<p>Macro-geometry, Sect. 1.4.2.4.2. "Track macrogeometry", p. 1-153</p>	_mg_type	<p>Type of macrogeometry 0: tangent, 1: curve, 2: S curve, 3: switch, 4: file</p>
	_mg_file	Index of macrogeometry file starting with, Sect. 1.4.2.4.2.7. "Track type: from file. Set of macrogeometry files", p. 1-157
	_mgc_l1	(m) Curve, L1 (tangent before curve)
	_mgc_p11	(m) Curve, P11 (transition 1)
	_mgc_s1	(m) Curve, S1 (steady curve length)
	_mgc_r1	(m) Curve, R1 (radius)
	_mgc_h1	(m) Curve, H1 (cant)
	_mgc_p12	(m) Curve, P12 (transition 2)
	_mgc_dy1	(m) Curve, dy1 (widening)
	_mgsc_l2	(m) S Curve, L2 (tangent before curve)
	_mgsc_p21	(m) S Curve, P21 (transition 1)
	_mgsc_s2	(m) S Curve, S2 (steady curve length)
	_mgsc_r2	(m) S Curve, R2 (radius)
	_mgsc_h2	(m) S Curve, H2 (cant)
	_mgsc_p22	(m) S Curve, P22 (transition 2)
_mgsc_dy2	(m) S Curve, dy2 (widening)	

### 1.6.2. Example: factorial experiment

An example of a factorial experiment is shown in Figure 1.300. In this experiments, influence of nine factors is estimated five of which are specified by internal identifiers.

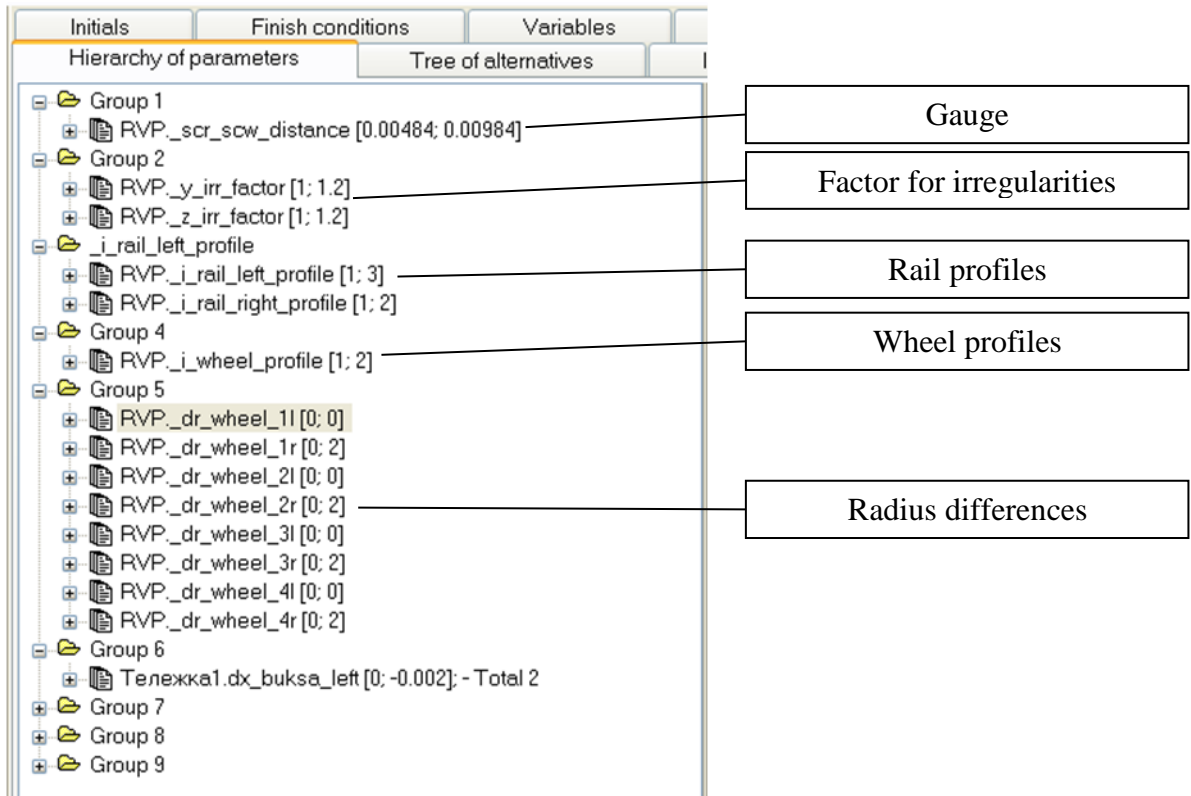


Figure 1.300. Hierarchy of parameters in factorial experiment

## 1.7. Methods for evaluation of critical speed

It is recommended to use two methods for evaluation of the critical speed of rail vehicle. In both cases it is recommended to apply a *short lateral irregularity*, which initiates the perturbation of motion of a rail vehicle in lateral direction, as well as **vertical irregularities** along the full travel distance.

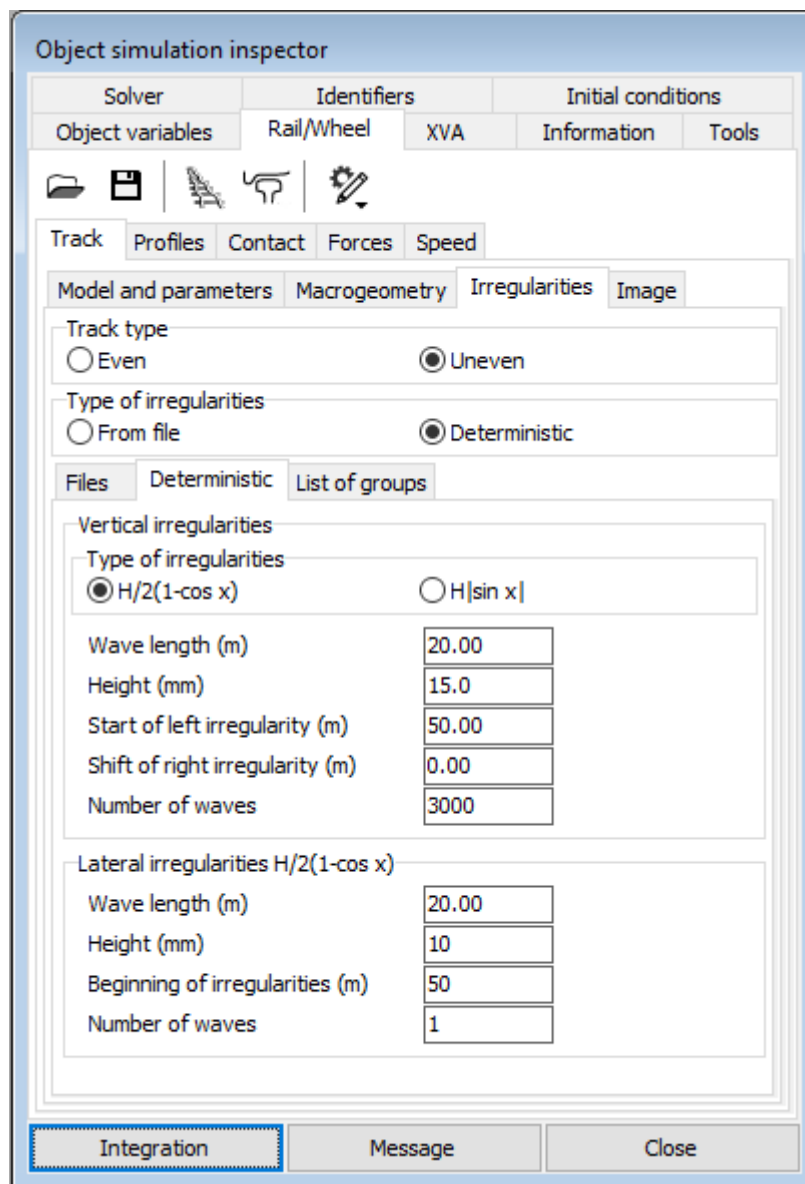


Figure 1.301. Typical parameter values for deterministic irregularities in evaluation of critical speed

Deterministic irregularities are suitable for setting both lateral and vertical irregularities, Sect. 1.4.2.4.3.2. "*Deterministic irregularities*", p. 1-160. An example of such irregularities is shown in Figure 1.301. The following requirements should be met:

- vertical irregularities must be harmonic, equal for the left and right rail, and long enough due to large number of waves  $N$ ;
- horizontal irregularity must be short; the recommended number of waves is 1.

### 1.7.1. Method of slow decrease of speed

The **neutral** mode of longitudinal motion should be set, Sect. 1.4.2.2.1. "Neutral", p. 1-139. Initial speed  $v_0$  must be greater than the critical speed. It is to impart a small deceleration  $0.1-0.2 \text{ m/s}^2$  to a rail vehicle. To do this, a constant force

$$F_x = -Ma$$

is to be applied to the car body of a rail vehicle, where  $M$  is the mass of a rail vehicle, and  $a$  is the desired value of the deceleration.

The force must be preliminary described in the **UM Input** program. Add a  $T$ -force and set its longitudinal component by an identifier with zero default value. The force vertical application point should be at the automatic coupler level, Figure 1.302.

To estimate the stability of a rail vehicle, a plot of lateral motion of the first wheelset versus speed is used. Two variables are created in the **Wizard of variables** and dragged into a graphic window, see Figure 1.303, Figure 1.304. The speed must be taken as abscissa: select the speed variable in the graphic window, call the popup menu by the right mouse button, and run the corresponding command, Figure 1.305.

Then set (Figure 1.306):

- a desired and negative (!) value for the identifier of the braking force,
- value of the initial speed  $v_0$ ,
- big enough simulation time,
- neutral mode of longitudinal motion.
- track type: tangent.

It is recommended to deactivate drawing rails in animation window, because computation of a long track image requires a lot of time.

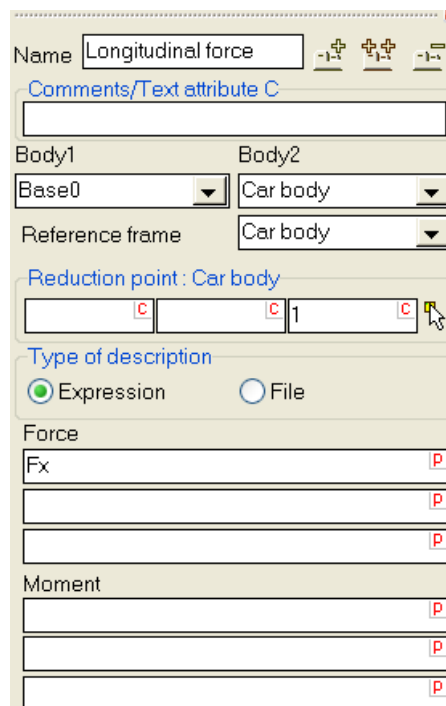


Figure 1.302. Longitudinal force

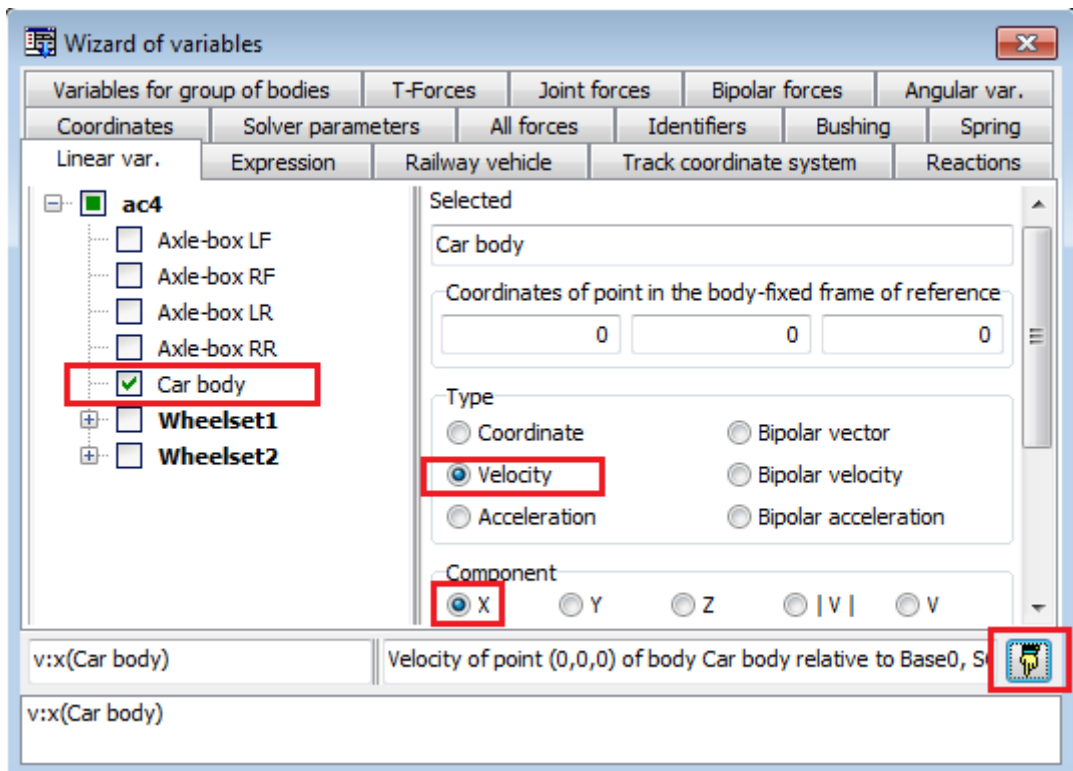


Figure 1.303. Creating variable: longitudinal speed of the vehicle

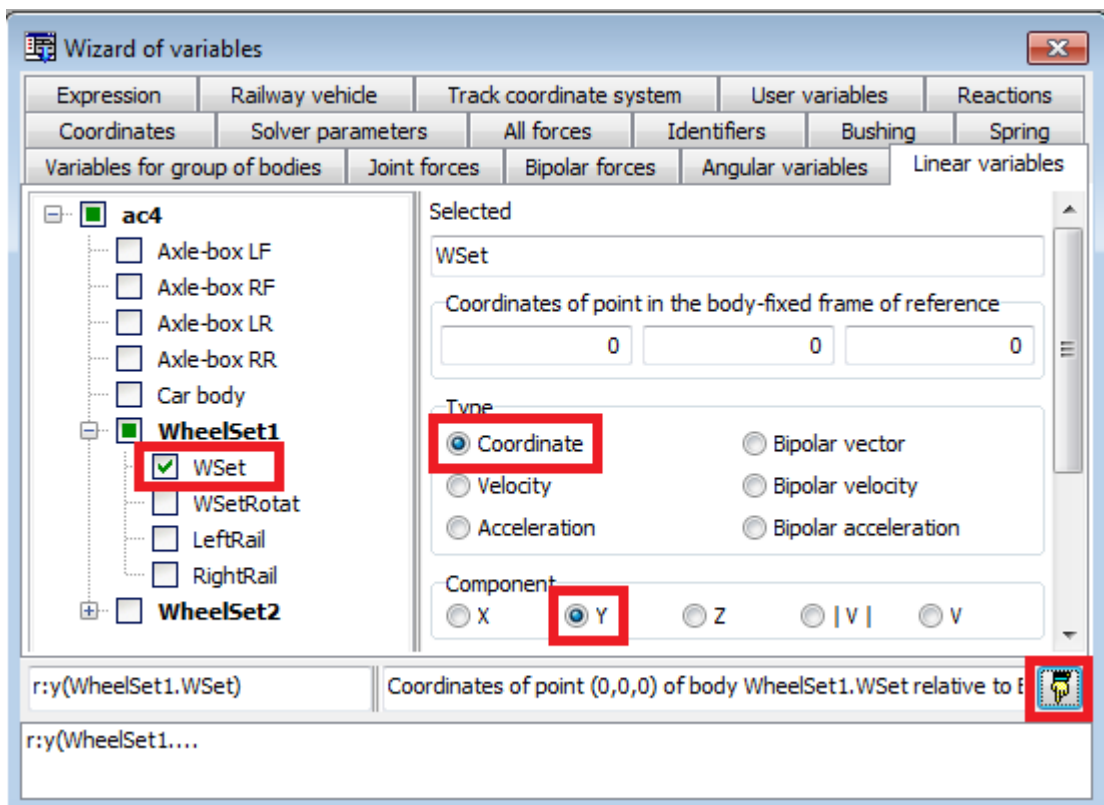


Figure 1.304. Creating variables: lateral displacement of the first wheelset

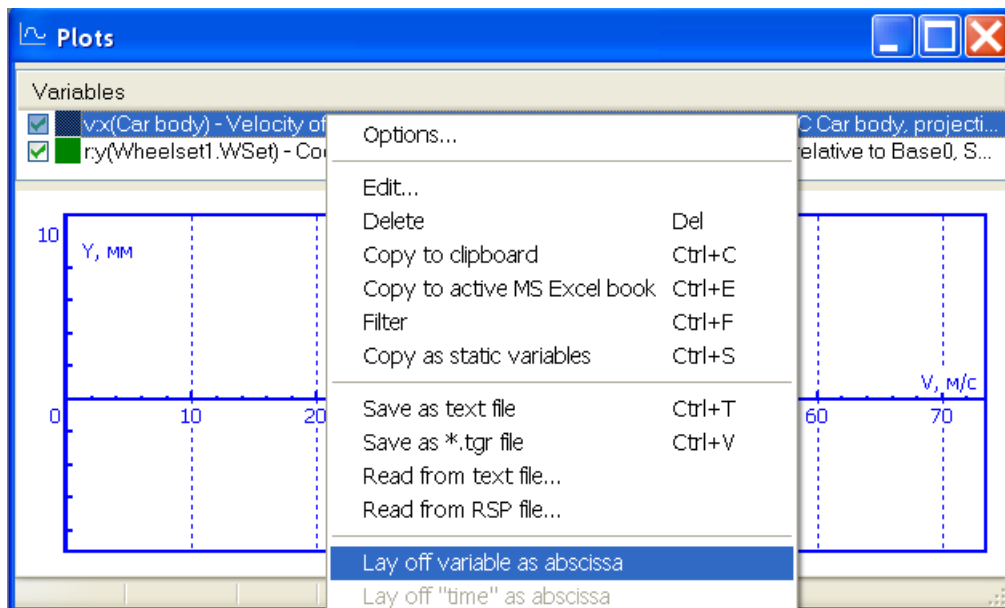


Figure 1.305. Variable vx is taken as abscissa

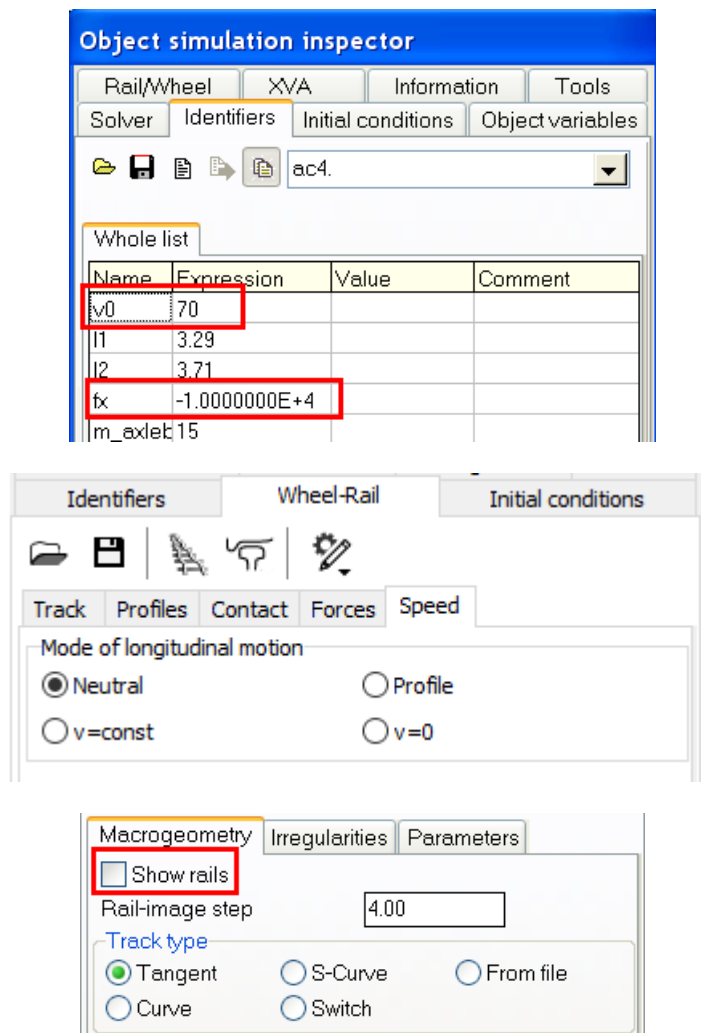


Figure 1.306. Preparing evaluation of critical speed

Critical speed can be estimated on the plot like in Figure 1.307. We have obtained this plot for the model of rail car AC4 ([\UM Data\SAMPLES\Rail Vehicles\AC4](#)). Critical speed is about 30 m/s as a boundary value between stable motion of the wheelset and its oscillations.

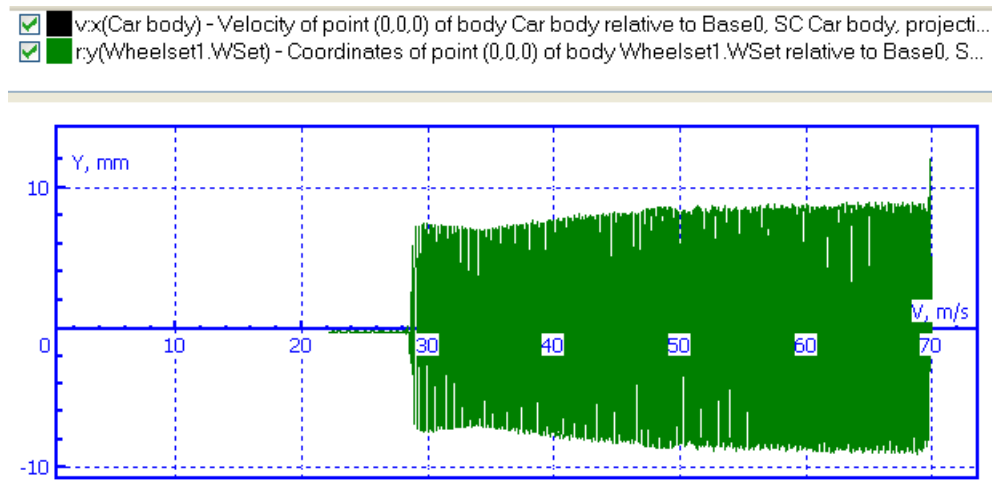


Figure 1.307. Lateral oscillations of WS1 versus speed

Note that if we set zero value of vertical irregularities, the critical speed will be about 42 m/s according to the plot in Figure 1.308. This value of critical speed is wrong. The large difference in results appears because of inclined friction dampers in the model of the rail car. In the case of simulation without vertical irregularities, the friction dampers come to the sticking mode when the speed was big enough, and we have got quite different behavior of the vehicle.

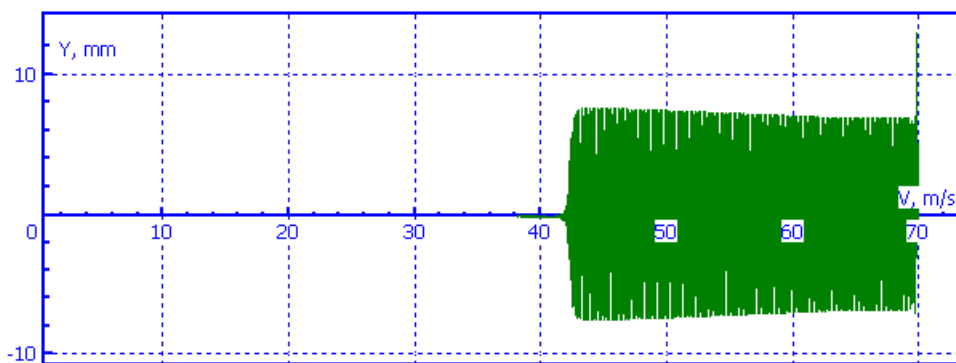


Figure 1.308. Lateral oscillations of WS 1 versus speed without vertical irregularities

## 1.7.2. Evaluation of critical speed by multivariant calculations

We suppose in this section that the **UM Experiments** is available, see Figure 1.298.

The following steps are required for evaluation of critical speed by a scanning project.

1. Create a new scanning project by the **Scanning | New project...** menu command.
2. Load the rail vehicle model by the **+** button, Figure 1.309.

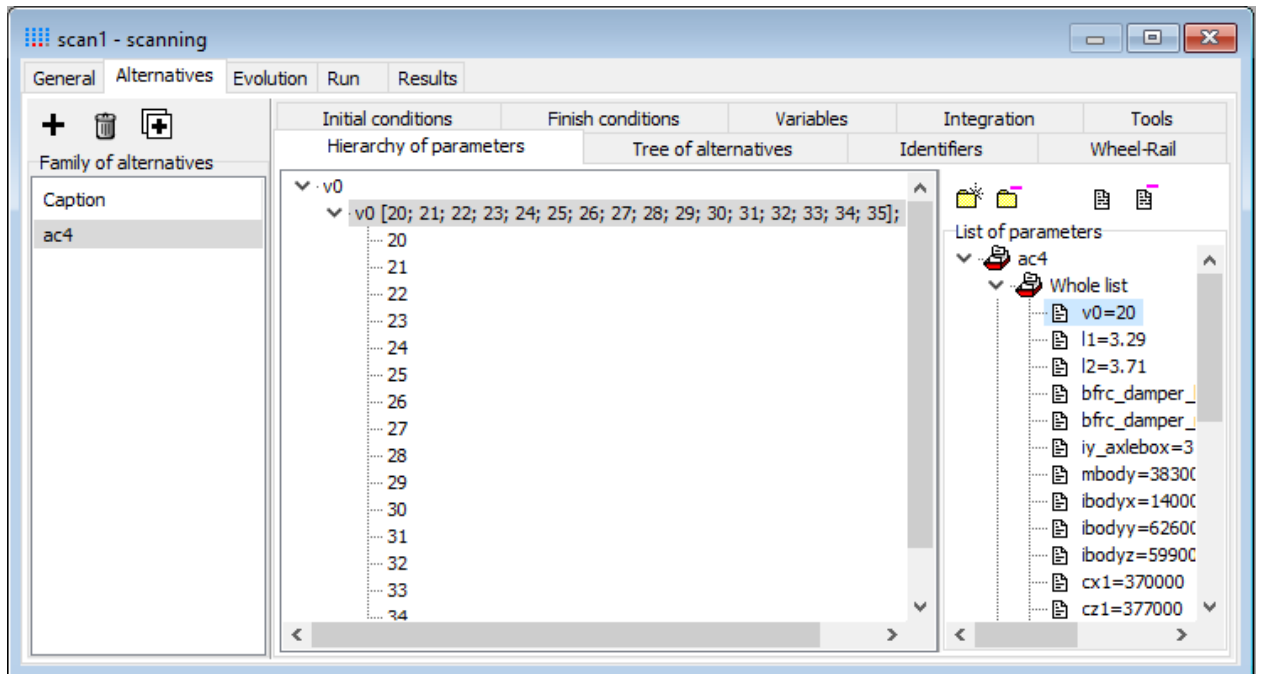


Figure 1.309. List of speed values for evaluation of critical speed

3. Create a list of speeds; the lowest speed must correspond to the stable region, the highest one must be greater than the critical speed.
  - click on the speed identifier **v0** in the identifier list, Figure 1.309;
  - create a list of speeds with some step size, e.g. in 1 m/s.

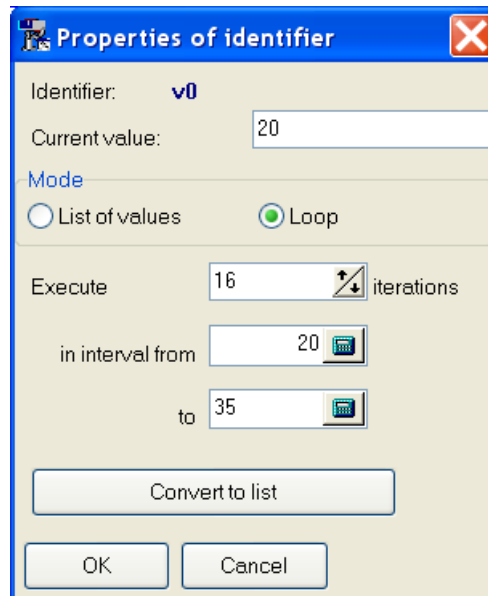


Figure 1.310. List of speeds

4. Set deterministic irregularities in the **Alternatives | Rail/Wheel | Track | Irregularities** tab, Figure 1.301. Assign wheel and rail profiles, set other parameters and options. If necessary, assign identifier values.

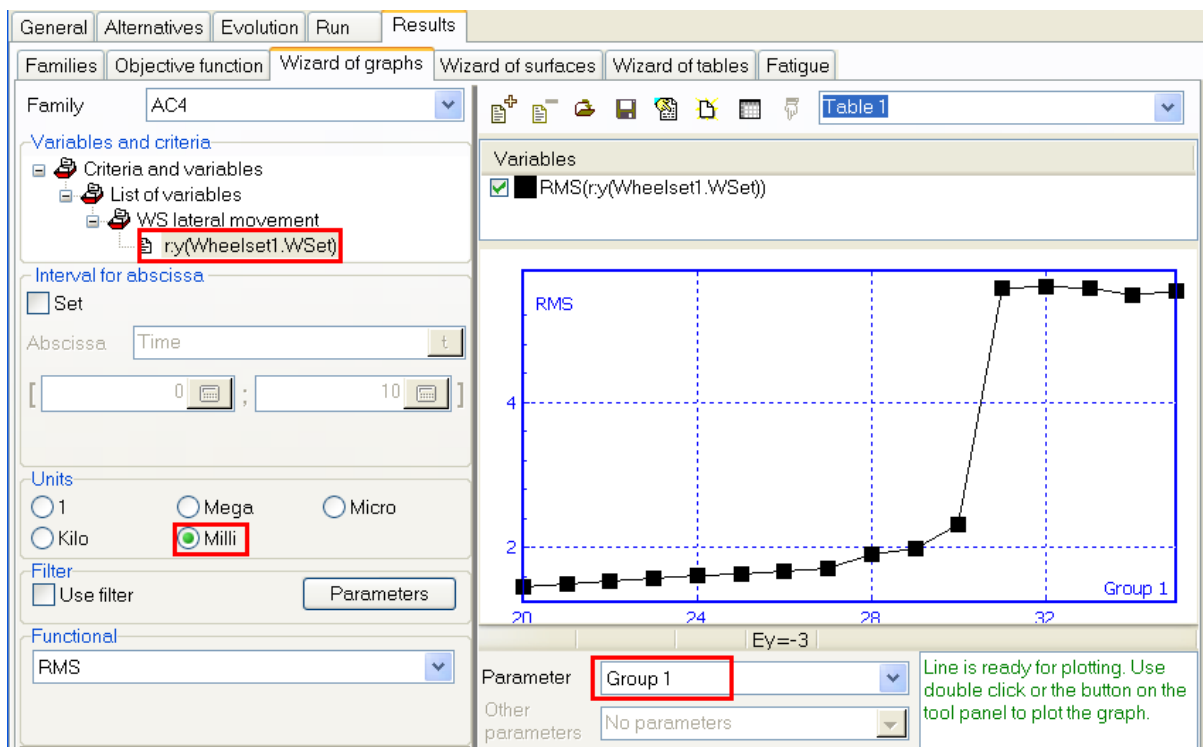


Figure 1.311. Processing results

5. Use the **Wizard of variables** to create a variable corresponding to lateral motion of the leading wheelset. Drag the variable in the **Alternative | Variables** tab.
6. Start calculations of the **Run** tab.

7. After end of the computations, open the **Results | Wizard of graphs** tab and create a plot of RMS of the lateral WS motion vs. speed, Figure 1.311:
- select **Group1** in the list **Parameter**;
  - set **Units** to **Milli**;
  - double click on the variable **ry**;
  - analyze the results; the critical speed lies in the region of growth of the RMS value.

In our case the critical speed lies in the interval from 30 to 31 m/s, which is close to the result obtained by the slow decrease the a speed.

## References

- [1] E. Jie, J.Y. Kim, D.H. Hwang, J.H. Lee, K.J. Kim, J.C. Kim, An Experimental Study of Squeal Noise Characteristics for Railways using a Scale Model Test Rig // Proceedings of the Third International Conference on Railway Technology: Research, Development and Maintenance, Stirlingshire, 2016.– Paper 231..
- [2] J.J. Kalker and J. Piotrowski, Some New Results in Rolling Contact // Vehicle System Dynamics. – 1989. – Vol.18(4). – P.223-242.
- [3] D.Yu. Pogorelov, Simulation of rail vehicle dynamics with universal mechanism software. In: A. Sladkowski (ed.) Rail vehicle dynamics and associated problems, Gliwice: Silesian University of Technology, 2005 – P.13-58.
- [4] J. Piotrowski, W. Kik, A simplified model of wheel/rail contact mechanics for non-Hertzian problems and its application in rail vehicle dynamic simulations // Vehicle System Dynamics. – 2008. – Vol.46(1-2). – P.27-48.
- [5] Kalker J.J., Three-dimensional elastic bodies in rolling contact. Solid and its Applications // Kluwer academic publishers, Dordrecht, Netherlands, 1990. – 314 p..
- [6] O. Polach, Creep forces in simulation of traction vehicles running on adhesion limit // Wear. – 2005. – Vol.258(7-8). – P.992–1000.
- [7] V. Quarz, Die Generierung von Fahrwegstörungen für vorgegebene Spektraldichten mit Hilfe orthogonaler Funktionen // Dissertation, Dresden: Universitaet Dresden, 2004.
- [8] ERRI B176/DT 290: B176/3 Benchmark Problem, Results and Assessment. Technical report, European Rail Research Institute, 1993.
- [9] Pieringer A., Kropp W., Nielsen J.C.O, The influence of contact modelling on simulated wheel/rail interaction due to wheel flats // Wear – 2014. – V.314(1-2) – P.273-281.
- [10] Weinstock H., Wheel Climb Derailment Criteria for Evaluation of Rail Vehicle Safety // Proceeding of ASME Winter Annual Meeting. – New Orleans, 1984. – Paper no.84-WA/RT-1.
- [11] UIC 513R. Guidelines for Evaluating Passenger Comfort in Relation to Vibration in Railway Vehicle, International Union of Railways, 1994.
- [12] OST 24.050.16-85. Passenger cars. Routine to specify ride comfort.
- [13] GOST P 55516-2013 Locomotives. Requirements for strength and dynamic properties.
- [14] EN 12299 Railway applications. Ride comfort for passengers. Measurement and evaluation.
- [15] Zavertaluk A.V., Cherkashin Yu.M., Pogorelov D. Yu., Simonov V.A., Yakovleva E.V., Technology for simulation of derailment of rolling stock (with use of Universal Mechanism Software). Report 19.5.018.N. Moscow. Russian Railways. VNIIZHT. 2009. (in Russian).
- [16] Ushkalov V.F., Mathematical modeling oscillations of rail vehicles; Institute of Technical Mechanics. Kiev. Naukova dumka. 1989. (in Russian).
- [17] Nadal, M.J., "Theorie de la stabilite des Locomotives, part 2, Mouvement de Lacet", Annales des Mines, Vol.10, 1896, p.232.

