

8.	SIMULATION OF RAILWAY VEHICLES IN UM	8-3
8.1.	General information.....	8-3
8.2.	Creating railway vehicle models	8-4
8.2.1.	Type of object – Railway Vehicle	8-4
8.2.2.	Base coordinate system.....	8-4
8.2.3.	Wheelset	8-5
8.2.3.1.	Structure of subsystem Wheelset	8-5
8.2.3.2.	Wheelset geometry	8-6
8.2.3.3.	Adding wheelsets and changing their parameters	8-7
8.2.4.	Modeling axle-boxes	8-10
8.2.4.1.	Axle-boxes as massless graphic images	8-10
8.2.4.2.	Axle-box as a rigid body	8-12
8.2.5.	Modeling linear springs of primary and secondary suspension	8-13
8.2.6.	Modeling dampers, traction rods	8-14
8.2.7.	Some features of development of locomotive models	8-14
8.2.7.1.	Model of traction	8-14
8.2.7.1.1.	Bodies and joints	8-15
8.2.7.1.2.	Traction torque	8-15
8.2.7.1.3.	Gearing	8-17
8.2.7.1.4.	Computation of initial angular velocities by fixation file.....	8-19
8.2.7.1.5.	Computation of initial angular velocities by constraints for initial values	8-21
8.2.7.2.	Support of braking mode.....	8-23
8.3.	Rail track.....	8-25
8.3.1.	Track geometry	8-25
8.3.1.1.	Geometry of rails in an ideal straight section	8-25
8.3.1.2.	Macro geometry of curve	8-26
8.3.1.3.	Switch geometry.....	8-30
8.3.1.4.	Track system of coordinates	8-31
8.3.1.5.	Track irregularities.....	8-32
8.3.2.	Elastic, dissipative and inertia properties of track	8-32
8.4.	Wheel-rail contact.....	8-33
8.4.1.	Algorithms for wheel-rail contact geometry	8-33
8.4.1.1.	Algorithm for computation of nearest points between two profiles	8-33
8.4.1.2.	Computing tables of contact points.....	8-34
8.4.1.3.	Simplified contact geometry. Equivalent conicity and contact angle parameter.....	8-38
8.4.2.	Tool for analysis of pairs of profiles.....	8-41
8.4.3.	Contact forces	8-44
8.4.3.1.	Method for computation of rail deflections and contact force	8-44
8.4.3.2.	Algorithms for computing creep forces.....	8-46
8.4.3.2.1.	Mueller's method.....	8-47
8.4.3.2.2.	FastSim	8-47
8.4.3.2.3.	FastSim analytic	8-48
8.4.3.2.4.	Minov's model	8-48
8.4.3.3.	Non-elliptical contact model	8-49
8.4.3.4.	Coefficient of friction in wheel/rail contact.....	8-52
8.5.	Simulation of railway vehicles.....	8-53
8.5.1.	Tools for preparing simulation process.....	8-53
8.5.1.1.	Creation of wheel and rail profiles.....	8-53
8.5.1.1.1.	Input as a set of points with successive spline approximation.....	8-54
8.5.1.1.2.	Input of profiles as a set of line segments and circle arcs.....	8-55
8.5.1.2.	Creation of track irregularities.....	8-55
8.5.1.2.1.	Creation of files with irregularities	8-56
8.5.1.2.2.	Programming irregularities in the Control file.....	8-61
8.5.1.3.	Creation of macro-geometry files	8-63
8.5.1.4.	Creation of files with switch parameters	8-67
8.5.2.	Setting parameters of railway vehicle dynamics simulation	8-68

8.5.2.1.	Solver and its parameters. Use of Jacobians.....	8-68
8.5.2.2.	Modes of longitudinal motion of vehicle	8-69
8.5.2.3.	Assignment of rail and wheel profiles.....	8-71
8.5.2.3.1.	Assignment of rail profiles	8-71
8.5.2.3.2.	Assignment of wheel profiles	8-73
8.5.2.3.3.	Rail profile evolution along the track.....	8-73
8.5.2.4.	Track parameters	8-75
8.5.2.4.1.	Assignment of track irregularities and rail geometry parameters (gauge, rail inclination).....	8-75
8.5.2.4.2.	Input track macro geometry parameters (tangent, curve, S-curve, switch, profile).....	8-76
8.5.2.4.3.	Track stiffness and damping.....	8-80
8.5.2.4.4.	Programming elastic and damping track parameters	8-80
8.5.2.5.	Parameters for computation of rail-wheel contact forces	8-81
8.5.2.6.	Setting in-train forces	8-83
8.5.2.6.1.	Setting forces and attachment points.....	8-83
8.5.2.6.2.	Creating files with force description.....	8-84
8.5.2.6.3.	Plotting and visualization of forces.....	8-84
8.5.2.6.4.	Example of in-train forces.....	8-85
8.5.2.7.	Additional contact parameters	8-88
8.5.2.7.1.	Difference in running circles radii	8-88
8.5.2.7.2.	Parameters of contact geometry computation.....	8-88
8.5.3.	Tools for visualization and analysis of railway vehicle dynamics.....	8-90
8.5.3.1.	Some features of creation of variables	8-90
8.5.3.1.1.	Rail/wheel contact variables.....	8-90
8.5.3.1.2.	Kinematic characteristics relative to track system of coordinates.....	8-94
8.5.3.1.3.	Use of <i>Path</i> variable	8-94
8.5.3.2.	Animation window	8-95
8.5.3.3.	Contact animation window.....	8-96
8.5.3.4.	Contact patch animation.....	8-97
8.5.3.5.	Table processor.....	8-98
	References.....	8-99

8. Simulation of railway vehicles in UM

8.1. General information

Program package “Universal Mechanism (UM)” has been developed at the Department of Applied Mechanics, Bryansk State Technical University. UM Loco is an extension of UM aimed at simulation of dynamics of railway vehicles (locomotives, cars, trains). In addition to the standard UM configuration it includes the following items:

- Wheelset as a standard subsystem;
- Automatic calculation of wheel/rail contact forces according to various models of creep forces (Mueller model, FastSim, Non-Hertzian contact model, etc.);
- 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 of the vehicles wheels);
- Database of profiles and track irregularities;
- Database of subsystems (wheels-motor blocks, bogies), 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 on straight track or 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 multivariant 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 Power-Point presentation at <http://umlab.ru/download/docs/eng/umloco.zip> (33Mb with animations)

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

8.2. Creating railway vehicle models

User creates UM models of railway vehicles similar to any other multibody systems, see Chapt. 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 railway vehicle contains the following rigid bodies: vehicle body, frames of bogies, wheelsets, axle-boxes (often can be removed from the model) etc. In the case of a locomotive, motor model consists often of two bodies: a motor case and a rotor.

See the *UMLoco_gse.pdf* file with detailed information about process of a rail vehicle model description.

8.2.1. Type of object – Railway 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. After that set its type: **railway vehicle** on the **Object** tab of the inspector. In this case

the standard UM subsystem *Wheelset* becomes available;

the standard identifier $v0$ for initial speed of the vehicle is added to the list of identifiers.

8.2.2. Base coordinate system

Base coordinate system (SC0, Base0) is the system of coordinate in which the object is described and simulated. For railway 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.

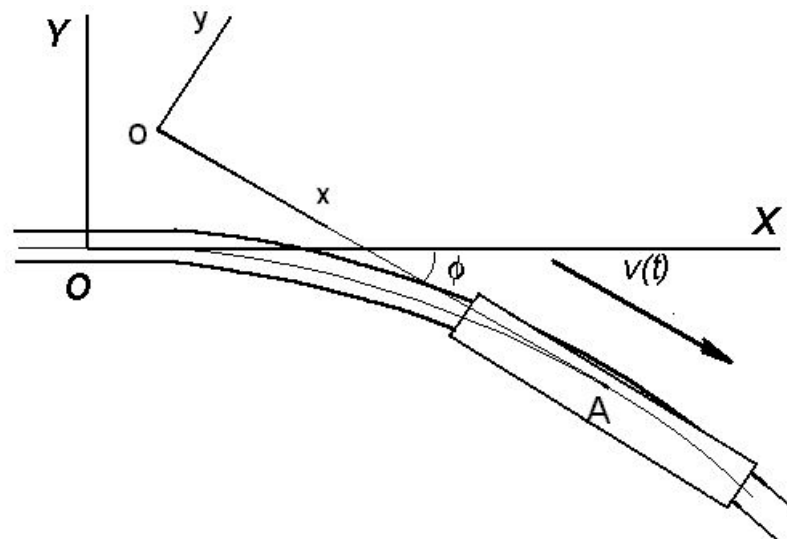


Fig.8.1. Reference frames

During the simulation the user can optionally choose two different types of SC0:

non-inertial system of coordinates;

inertial system of coordinates.

These types of reference frames differ while curving only (Fig.8.1). The inertial SC0 is fixed (OXY in Fig.8.1). The non-inertial SC0 (oxy in Fig.8.1) coincides with the inertial one at the initial state but when the vehicle runs in curve, the X-axis rolls without sliding in horizontal plane. So, X-axis is always tangential to the curve centerline at the position of the vehicle body

center of mass. X-coordinate of the vehicle body center of mass is equal to the length of the traversed path by the car body, if its initial X-coordinate is zero.

Description of models in Input module does not differ for both the SC0 because they coincide at $t = 0$. Use of the inertial SC0 is recommended in the following cases:

- for models with several vehicles (train);
- for running in sharp curves.

The inertial SC0 is automatically assigned in the case of switches and a track with an arbitrary macro-geometry, which can include several curves.

See Sect. 8.5.2.4.2 Input track macro geometry parameters.

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.

8.2.3. Wheelset

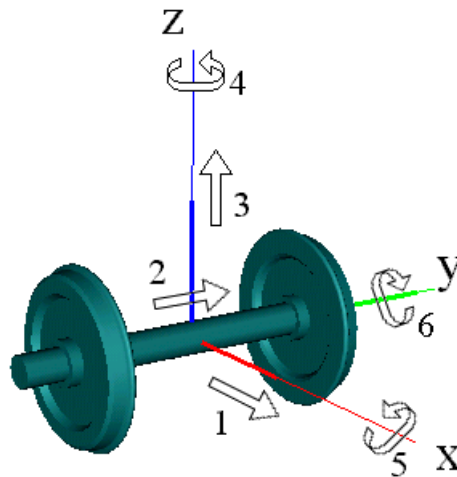


Fig.8.2. Wheelset degrees of freedom

8.2.3.1. Structure of subsystem Wheelset

A wheelset is a UM standard subsystem. Two types of wheelset with different number of degrees of freedom are available.

The model of **wheelset with 6 d.o.f.** includes two bodies, two joints and a wheelset image (Fig.8.2). The first body presents the *base* of the wheelset with 5 degrees of freedom. This body lacks for the rotation around wheelset axle. The standard name of the base is *Wset*. The second body with the standard name *WSetRotat* is a gyrostat. It has only one 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 a one body with six degrees of freedom. The goal of introducing the two bodies for one wheelset is as follows: it allows avoiding the introduction of axle-boxes as separate bodies and reduces considerably CPU expenses while simulation of the vehicle dynamics. Really, 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 (Fig.8.2):

- 1 – transition X;
- 2 – transition Y;
- 3 – transition Z;

- 4 – rotation Z;
- 5 – rotation X;
- 6 – rotation Y.

Wheelset with 7 d.o.f. is used sometimes in models of locomotives or wheelsets with freely rotating wheels. In comparison with the above model, the 7th degree of freedom is introduced as the rotation of the right wheel with respect to the left one. This degree of freedom allows the user to take into account the torsional axle compliance. The wheelset model consists of three bodies and three joints. The user should enter the joint force for the third joint *jWSetRightWheel*, which describes the torsional stiffness and damping of the axle. The user should set a value of the moment of inertia of the right wheel (*WSetRightWheel*) about the symmetry axis as well.

Subsystems of wheelsets can be corrected in a usual way, but anyway it is not recommended to change the sequence of rotations.

All wheelsets are added to the model of the vehicle as included subsystems.

8.2.3.2. Wheelset geometry

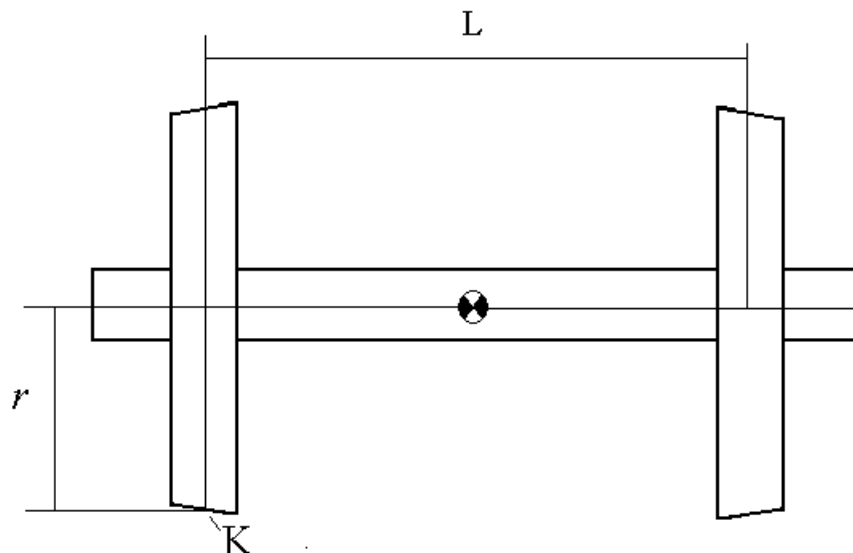


Fig.8.3. Geometrical parameters of wheelset

Geometrical properties of a wheelset are fully set with the following data (Fig.8.3, 8.4):
 wheelset semibase ($L/2$);
 running circle radius r ;
 reduction of running circles radii dr ;
 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 Input module as the parameters of the standard subsystem (wheelset).

The *difference in running circles radii* is the difference of the left and the right running circles radii (e.g. due to wear). This parameter is calculated according to the formula

$$dr = r_l - r_r$$

where r_l, r_r are the running circles of the left and right wheel respectively. The dr parameter is positive if the left radius is greater than the right one. If dr is nonzero, the radii r_l, r_r are equal to

$$r_l = r, r_r = r - dr,$$

i.e. the left radius is always equal to the value specified in the Input module.

Profiles are chosen from the UM database or created by the user with a special tool in the Simulation module (Sect. 8.5.1.1. *Creation of wheel and rail profiles*).

Let us introduce the notion of system of coordinate of a tread (wheel) profile Fig. 8.4, SCW. The SCW origin is located at point K on the running circle (Fig.8.3) 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.

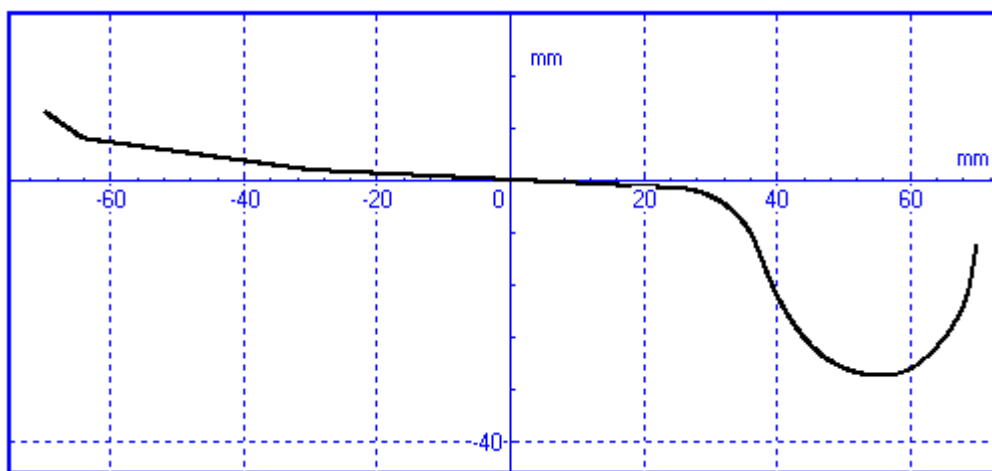


Fig.8.4. System of coordinates of a tread (wheel) profile

8.2.3.3. Adding wheelsets and changing their parameters

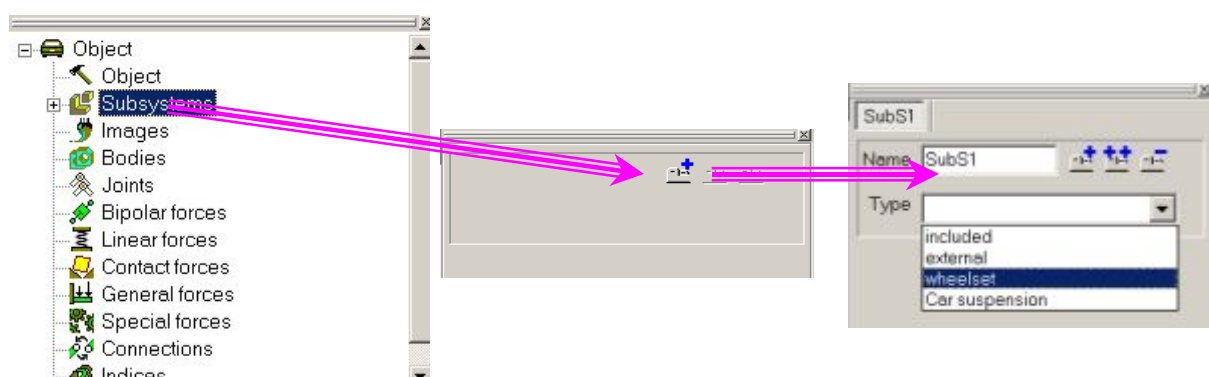



Fig.8.5. Adding the first wheelset to a vehicle or bogie

To add wheelsets (WS) to the model of a railway vehicle the Subsystems item of the element list is used. To create the first wheelset:

make sure that the object type is the railway vehicle (Sect.8.2.1);

add a new subsystem by clicking the  button (Fig.8.5);

Choose the type of the subsystem – **wheelset**.

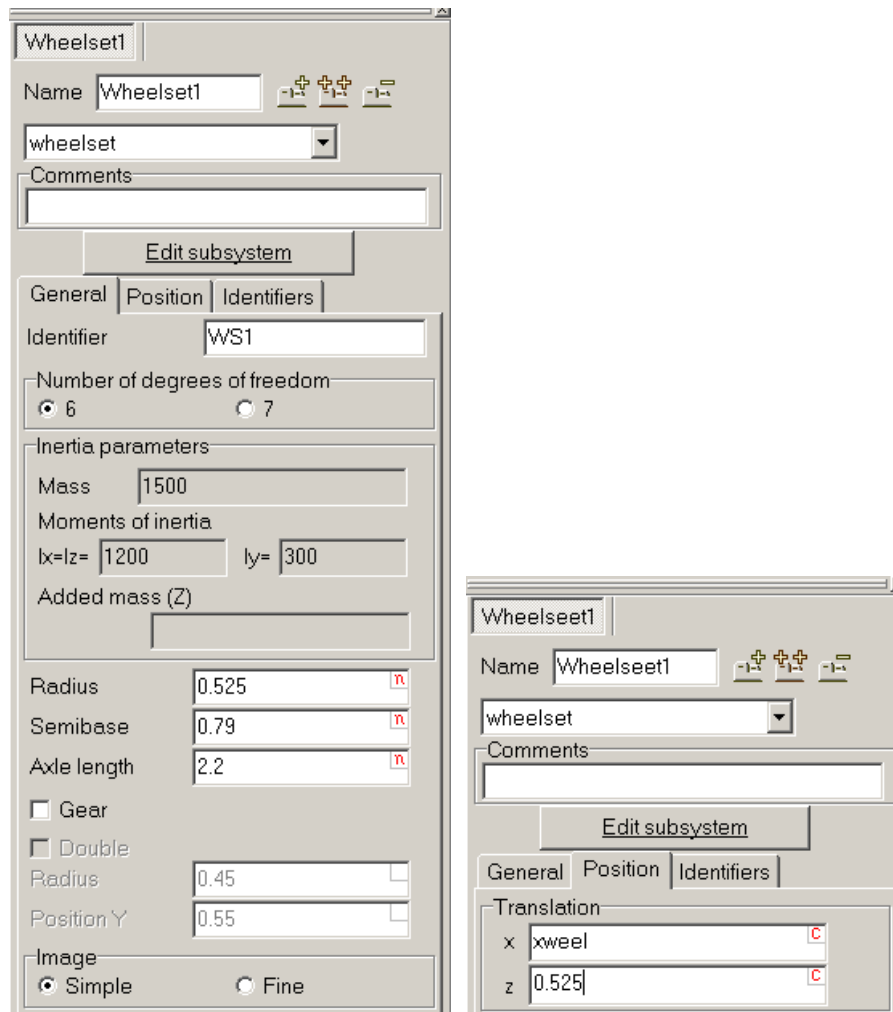


Fig.8.6. Wheelset parameters

The following parameters are available on the inspector window (Fig. 8.6):

Name of the subsystem, e.g. Wheelset1;

Identifier of the subsystem – set 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.

Number of degrees of freedom (Sect. 8.2.3.1 Structure of subsystem Wheelset).

Further parameters of the wheelset:

Radius of the running circle (Sect. 8.2.3.2);

Semibase of the wheelset (Sect. 8.2.3.2).

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

Gear – for a traction wheelset adds a simplified image of a gear (double if necessary) as well as the corresponding radius of the gear and its position on the axle

Image– Simple/Fine (differ in image discretization level).

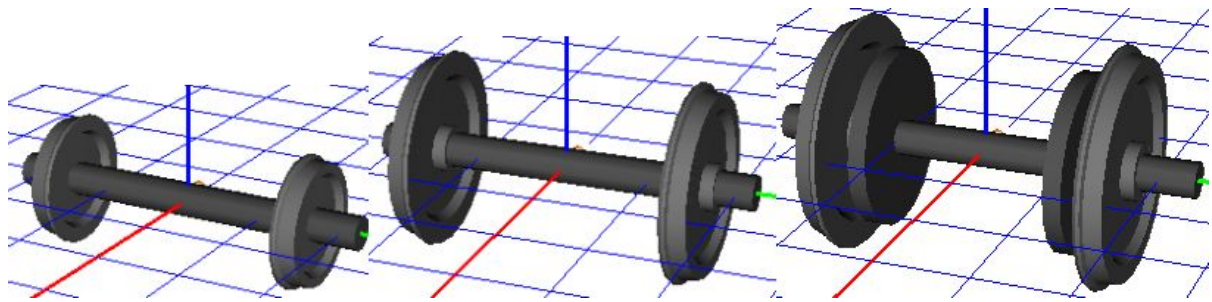


Fig.8.7. Images of wheelset

Changing geometric parameters of the wheelset results in automatic changing its image (Fig.8.7).


Note. An image is automatically assigned to the wheelset if the following files are available with UM:

- \bin\graph\rwsimple.umi – simple image ;
- \bin\graph\rwfine.umi – fine image.

Use the **Edit subsystem** button (Fig.8.6) 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 constructor where the necessary modifications of the wheelset such as changing parameters or renaming elements can be done in a usual manner (see Chapt.3, Chapt.9).

Use the **Position** tab (Fig.8.6) 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 simplified symbolic expression (Chapt.3, Sect. *Simplified 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. 8.2.2. Base coordinate system). Zero value of this coordinate corresponds to the wheelset center location of the SC0 origin.

To add all other wheelsets:

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

Visual adding a wheelset

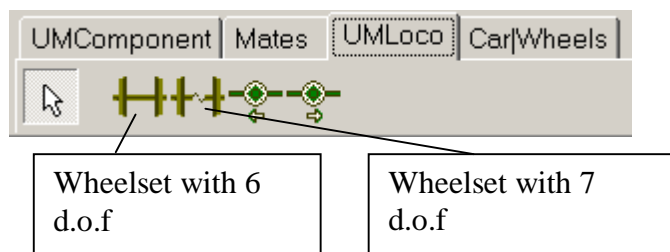


Fig.8.8. Visual components of UMLoco module

The UMLoco tab of the component lists is used for visual adding wheelsets, Fig. 8.8. The advantage of the visual adding: automatic full parameterization of the wheelset model as well as adding of connection points of the vector type for visual creation of axlebox revolute joints.

8.2.4. Modeling axle-boxes

Sometimes axle-boxes are not included in vehicle models as separate bodies (Sect. 8.2.3.1), and the bogie frame is connected directly with the base of the wheelset (the first body of the wheelset as a subsystem). 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 (non-rotating part 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.

8.2.4.1. 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\graph\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 (fig.8.6).
- Read image of the **left** axlebox from the `\bin\graph\axlebox.img` file by the  button on the tool panel. Rename the image, e.g. *Axlebox L*.
- Copy the image of the left axlebox, rotate it on 180 degrees about the vertical axis and rename the copy *Axlebox R*.

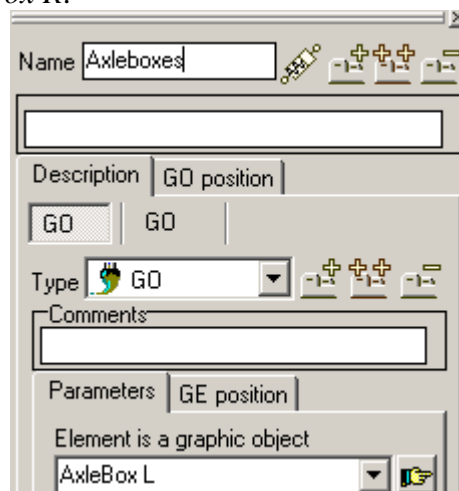


Fig.8.9. Graphic element of “GO” type

- Create a new graphic object with two graphic elements of “GO” type, assign images of the axle boxes to the elements, Fig. 8.9. Set lateral locations of the axleboxes and rename the image. The result is shown in Fig. 8.10.

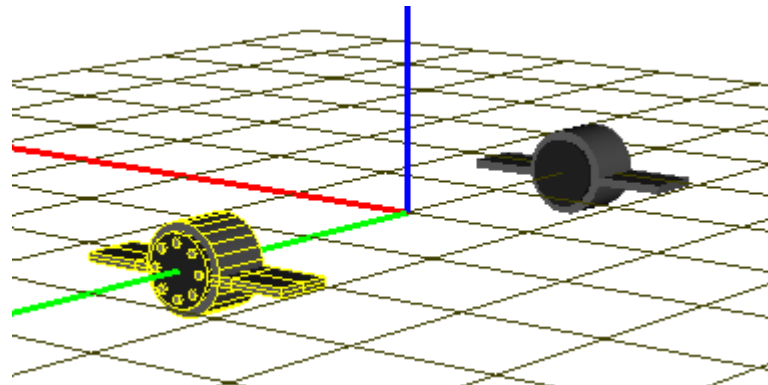


Fig.8.10. Image of fictitious axle-boxes

- Finally, select the *Wset* body in the list of bodies (the base of the wheelset, Sect.8.2.3.1), and assign to it the created image (Fig.8.11).

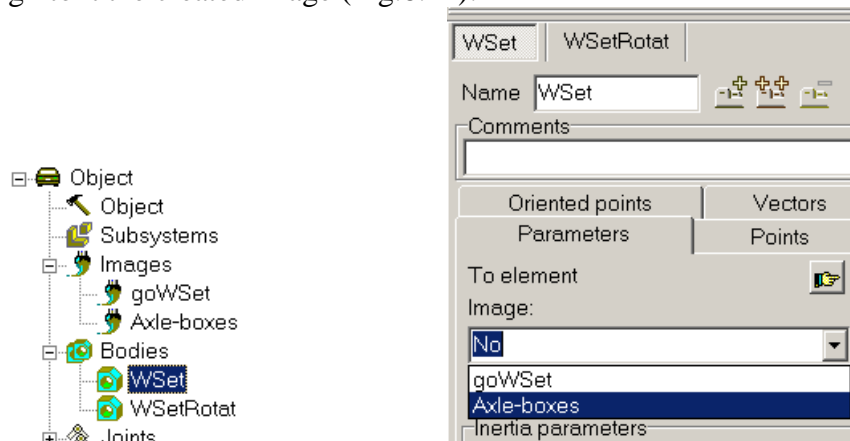


Fig.8.11. Selection of the *Wset* body in the list and assignment of an image

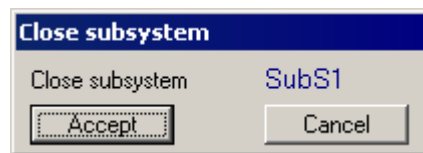


Fig.8.12. Close subsystem window

- Close editing the wheelset subsystem by clicking the **Accept** button of the close subsystem window or use the **X** button and confirm the acceptance of the modifications. As a result, the wheelset image will look like Fig.8.13.

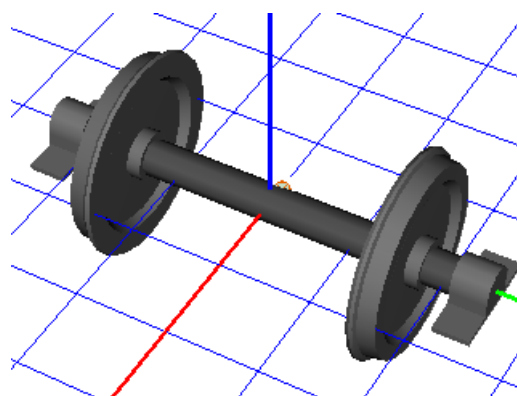



Fig.8.13. Wheelset with fictitious axle-boxes

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

8.2.4.2. 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 containing 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 using the  button on the tool panel).
3. Add a new body for the axle-box, rename it, assign the image a set inertia parameters. If the axle-box has one rotational degree of freedom relative to the axle, it is enough to set the moment of inertia relative to the Y-axis.

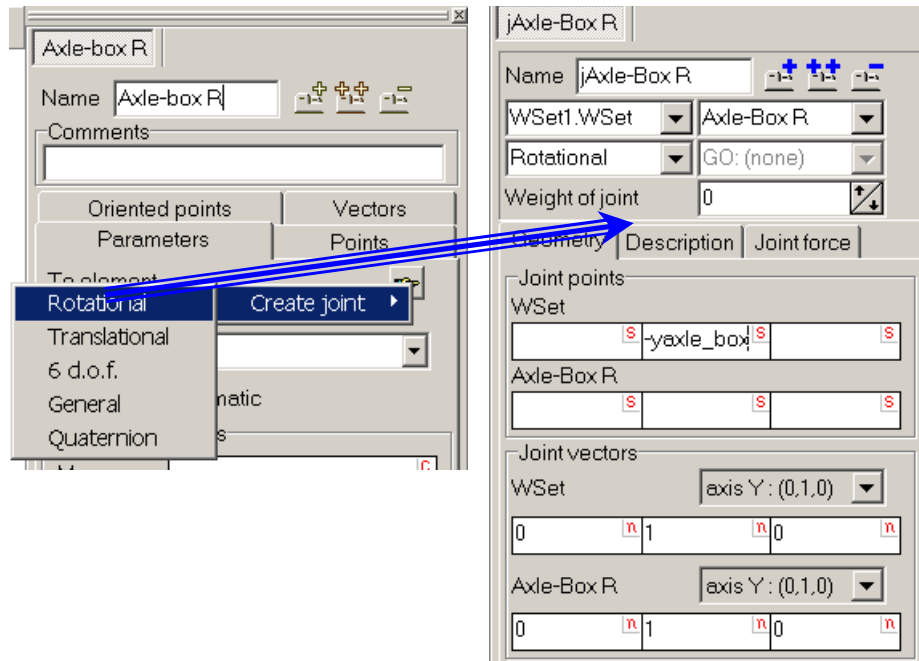



Fig.8.14. Adjustment of a joint to the right axle-box

4. Adjust a joint to the axle-box by the  button (Fig.8.14). 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 the 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 Fig.8.14 for the right axle-box).

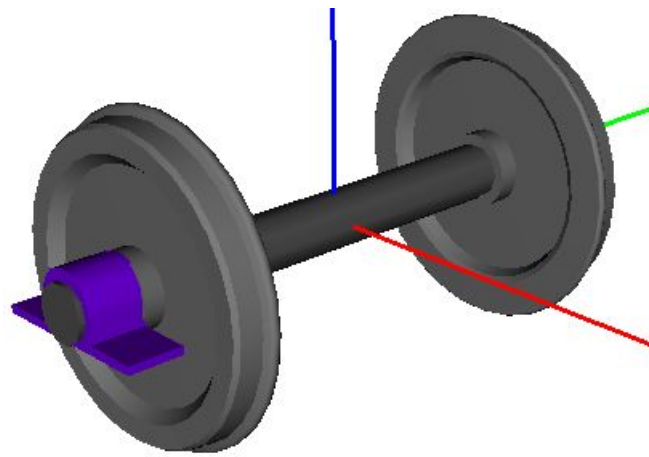


Fig.8.15. Wheelset with the right axle-box

Visual adding an axlebox

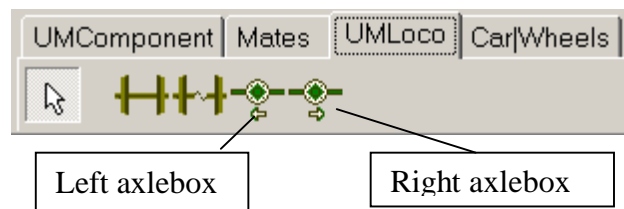



Fig.8.16. Visual components of UMLoco module

Axle boxes can be added by visual components

- Open the UMLoco tab of the visual component lists.
- Click on the axlebox button, then click somewhere on the grid for a temporary location of the axlebox.
- Assign a revolute joint using either method above or the visual component  of the joint. In the last case, connection points of vector type corresponding to the joint location should be created for the wheelset base. These points are automatically presented if the wheelset is added as a visual component.

8.2.5. 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 with identical shear stiffness.

Mathematic model of the element is described in Chapt.2, Sect. *Special forces/Spring*, *Generalized linear force element*.

Examples of description and/or usage:

- Chapt.7. Sect. *Models of Springs*
- Model: `\samples\Rail vehicles\ac4`
- Chapt. 8. Sect. *Springs*

- *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 is not identical in different directions. It can be used also for modeling bilinear springs.

Mathematic model of the element is described in Chapt.2, Sect. *Generalized linear force element*.

Examples of description and/or usage:

- Chapt.7. Sect. Models of Springs
- Model: `\samples\Rail vehicles\Manchester benchmarks\Vehicle1`
- Model: `\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 Chapt.2, Sect. *Bipolar force element*.

Examples of description and/or usage:

- Model: `\samples\Rail vehicles\Manchester benchmarks\Vehicle1`
- Model: `\samples\Rail vehicles\Manchester benchmarks\Vehicle2`

General recommendations:

- models should be created near the equilibrium state at zero values of object coordinates; stationary values of vertical loads should be set for springs;
- 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.

8.2.6. 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 Chapt.2, Sect. *Bipolar force element*.

Examples of description and/or usage:

- Model: `\samples\Rail vehicles\Manchester benchmarks\Vehicle1`
- Model: `\samples\Rail vehicles\Manchester benchmarks\Vehicle2`
- Model: `\samples\Rail vehicles\ac4`

8.2.7. Some features of development of locomotive models

Features of locomotive models are traction systems.

8.2.7.1. Model of traction

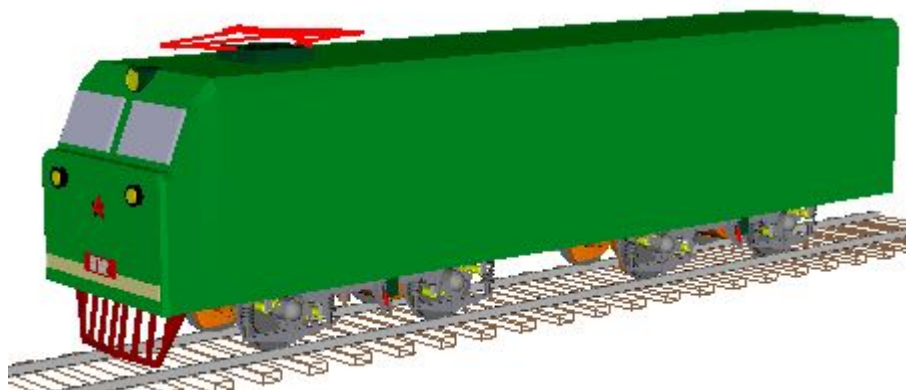


Fig.8.17. Model of one unit of VL80 electric locomotive

Simplified modeling of traction motors and wheel driving systems are 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, Fig. 8.17.

8.2.7.1.1. Bodies and joints

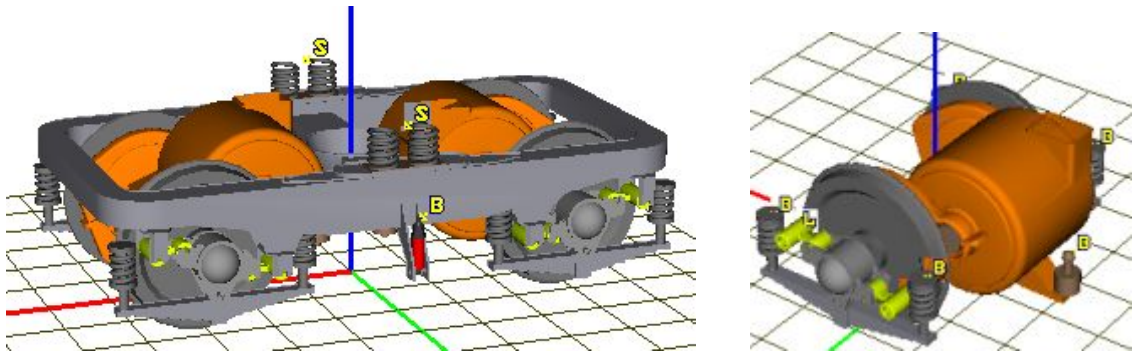


Fig.8.18. Subsystems “bogie” and “wheel-motor assembly”

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.

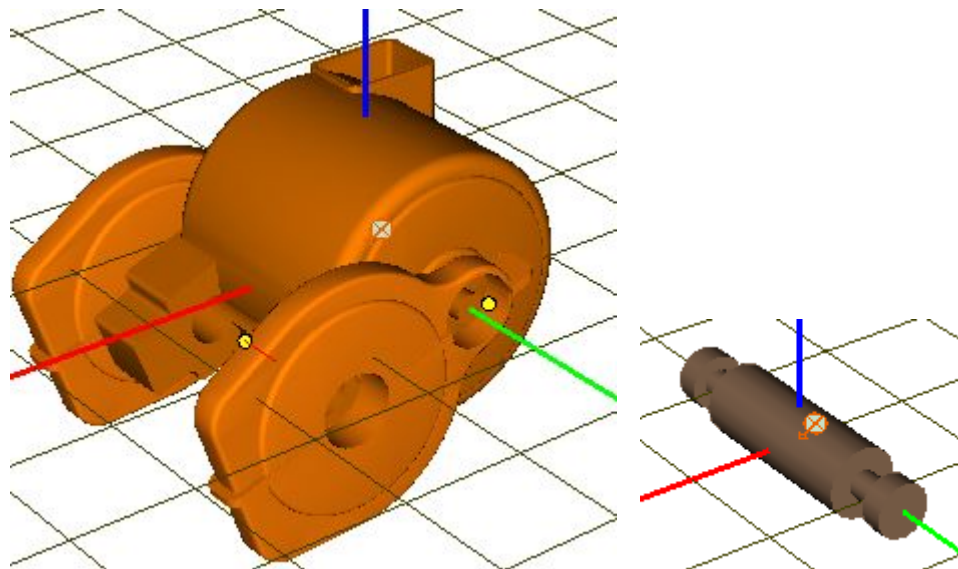


Fig.8.19. Bodies: motor and reducer casings and drive shaft

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, Fig.8.19.) as well as primary suspension force elements. All the suspension force elements are described as external ones (the *External* second body) and have the *autodetection* attribute on.

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.

8.2.7.1.2. Traction torque

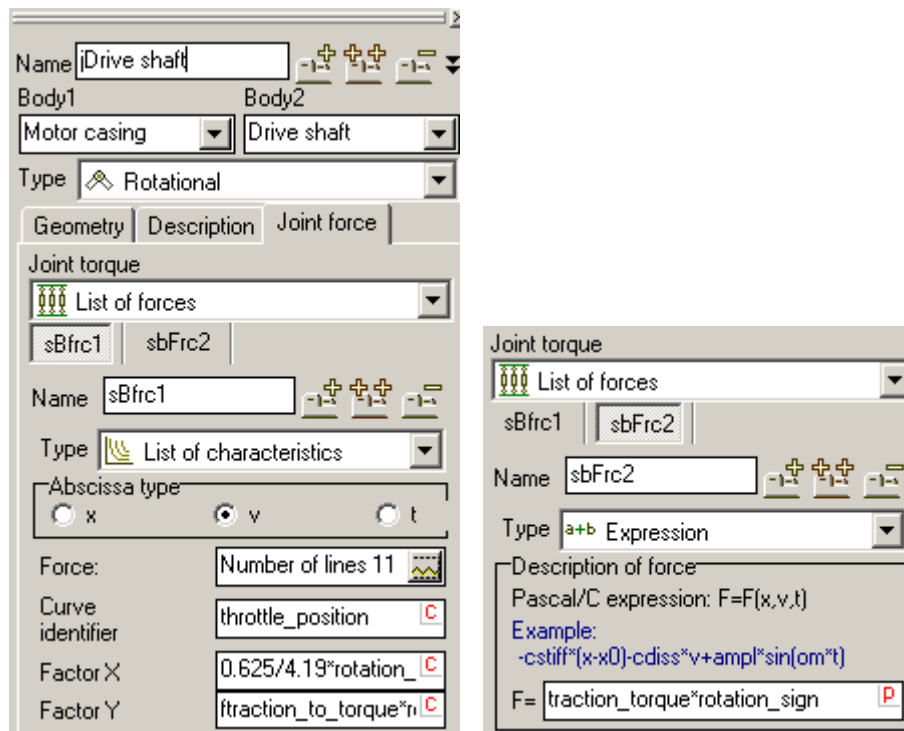


Fig.8.20. Setting traction torque

Driving torque is introduced as a joint torques in the revolute joint connecting the drive shaft with the motor casing, Fig. 8.20. 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.

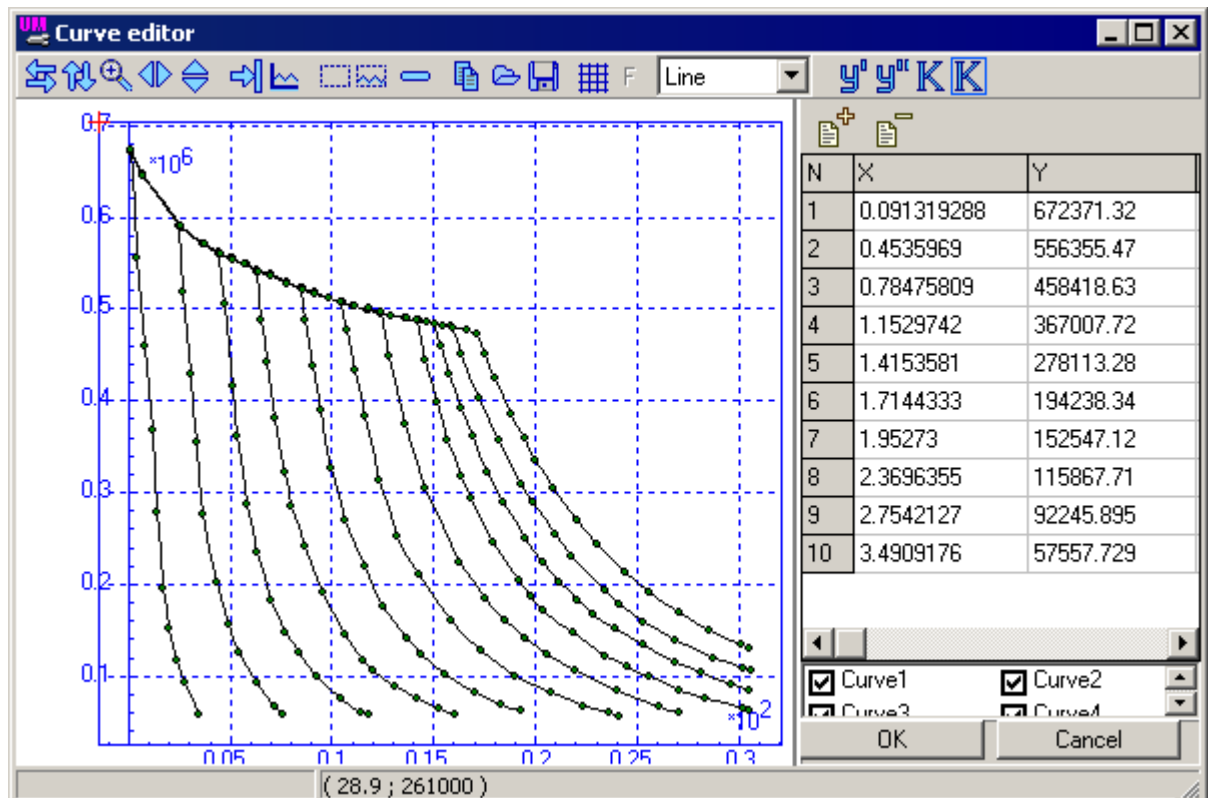


Fig.8.21. Total traction force versus unit speed

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 unit, Fig. 8.21. 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 (Fig. 8.20., left).

- **Curve identifier:** the identifies which numerical 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* identified, which automates the recognition of the identifier when the locomotive is included in a train 3D model.
- **Factor X:** the factor should differ from unity if abscissa in Fig. 8.20. 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 Fig. 8.20. the expression $0.625/4.19*rotation_sign$ is used for the factor, which, first, convert 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. 8.2.7.1.4 *Computation of initial angular velocities by fixation file*).

Factor Y: the factor should differ from unity if the ordinate in Fig. 8.20. is not an 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 Fig. 8.20. 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. Fig. 8.20. 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).

8.2.7.1.3. Gearing

Model of a gearing is the necessary part of the drive system.

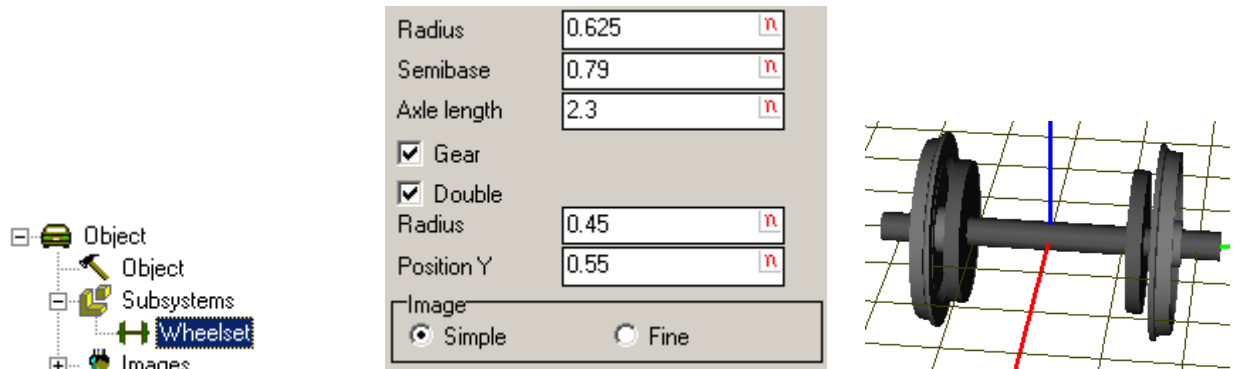


Fig.8.22. Adjustment of the wheelset image

To add bull gears to the wheelset image, open the wheelset subsystem parameters in the object inspector, check the **Gear** key and if necessary **Double** key, as well as set the gear radius (0.45m in the figure) and location on the axle (0.55m in the figure).

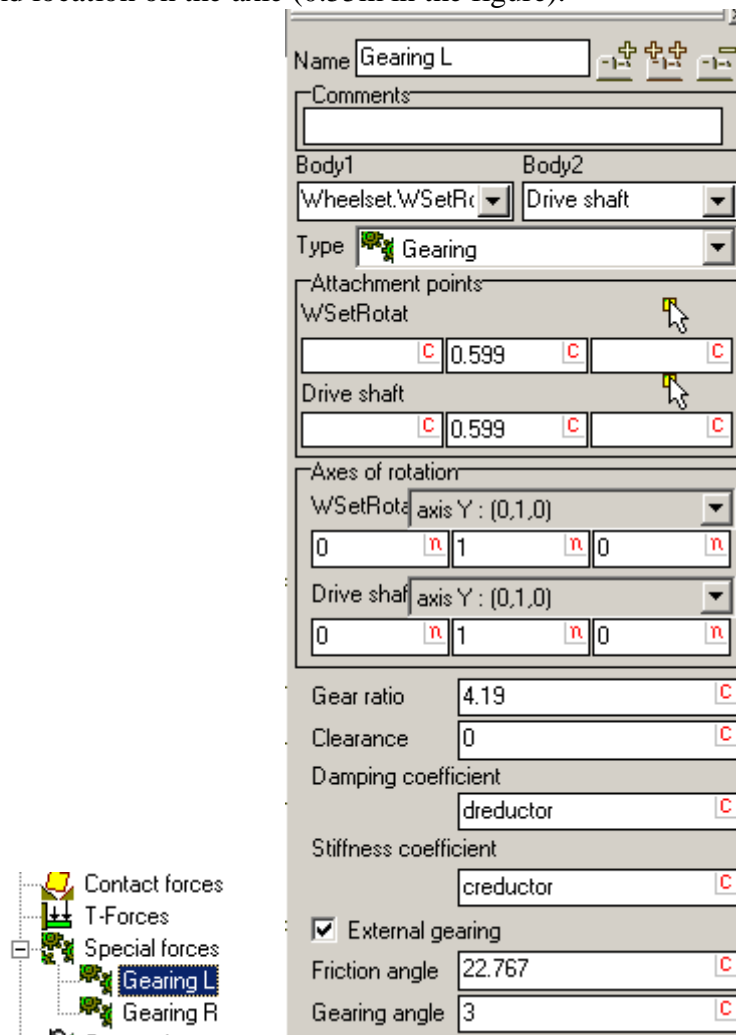


Fig.8.23. Gering force element description

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:

- **Attachment points** are coordinates of bull gear and pinion centers in system of coordinates of the wheelset and rotor;
- **Axes of rotation** are unit vectors (0, 1, 0);
- **Gear ration** 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.

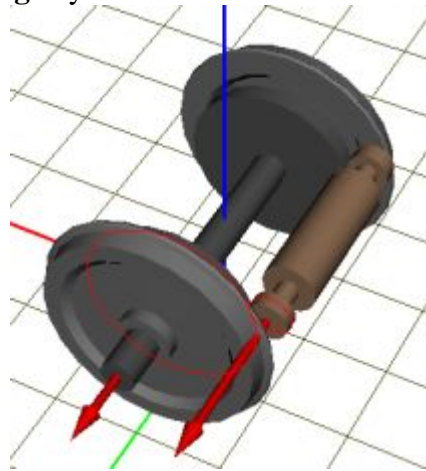



Fig.8.24. Visualization of the gearing force element

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, Fig. 8.24.

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

8.2.7.1.4. Computation of initial angular velocities by fixation file

Use of gearing makes setting the correct initial angular velocities of some bodies. Neglecting this requirement lead to large forces in drive system, to lifting wheels from rails at the beginning of simulation. One of the main tool for automatic computation of initial angular velocities (more precisely, joint velocities) is the creation by the user of a so-called **fixation file**, which prompt to the program which 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 ω_w is the angular velocity of the wheelset rotation, ω_m is the angular velocity of the motor casing relative to the wheelset base, ω_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 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 ω_w ω_r is forbidden (or the corresponding coordinates are *fixed*), the program automatically selects the only velocity ω_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.

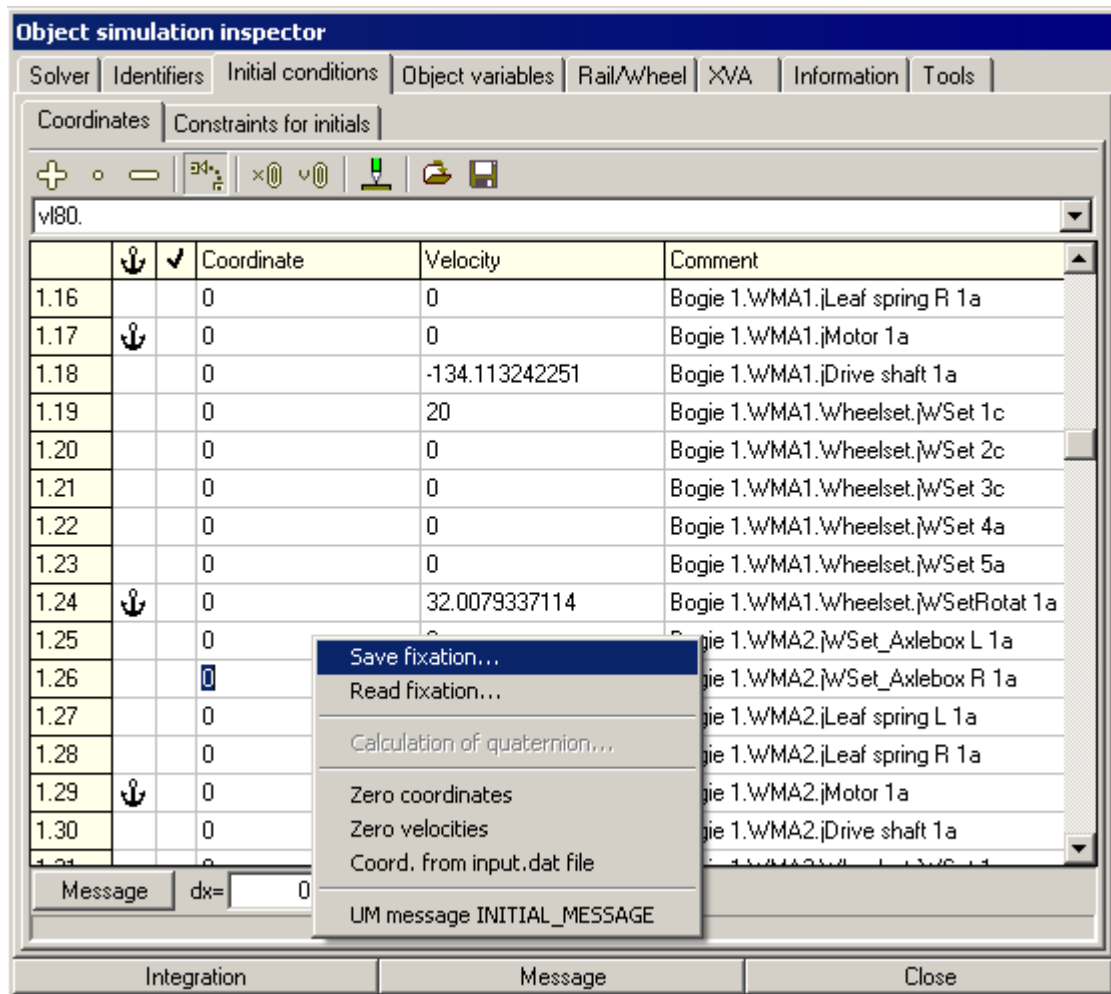




Fig.8.25. Fixation of coordinates and creation of fixation file

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 the 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 simulation module on the **Initial conditions** | **Coordinates** tab of the object simulation inspector, Fig. 8.25.

- Click on the upper button **Message**. 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 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 in 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 on 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 suspension (e.g. in case of quill drive) the fixation file is not sufficient for computation of all 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 Fig. 8.25. should be used for specifying the *rotation_sign* identifier value, see Sect. *Traction torque*.

8.2.7.1.5. Computation of initial angular velocities by constraints for initial values

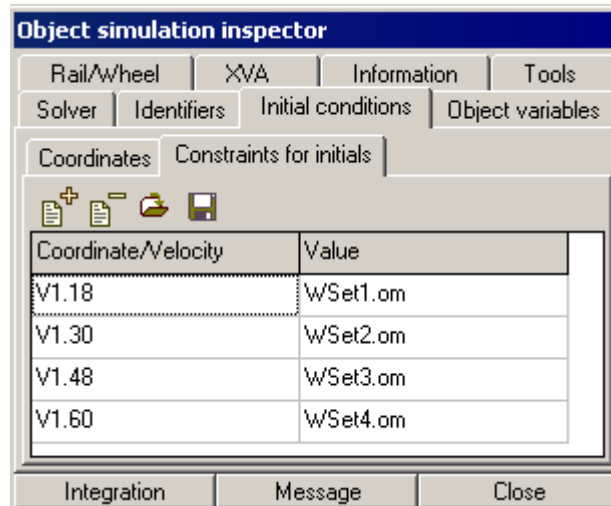


Fig.8.26. Tools for setting constraints for initial values

Constraints for initial values are an alternative of the fixation file, and give opportunities to some additional potential for setting initial values.



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. Chapt 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 by the **Tools | Wizard of variables** menu command or by then  button on the tool panel.

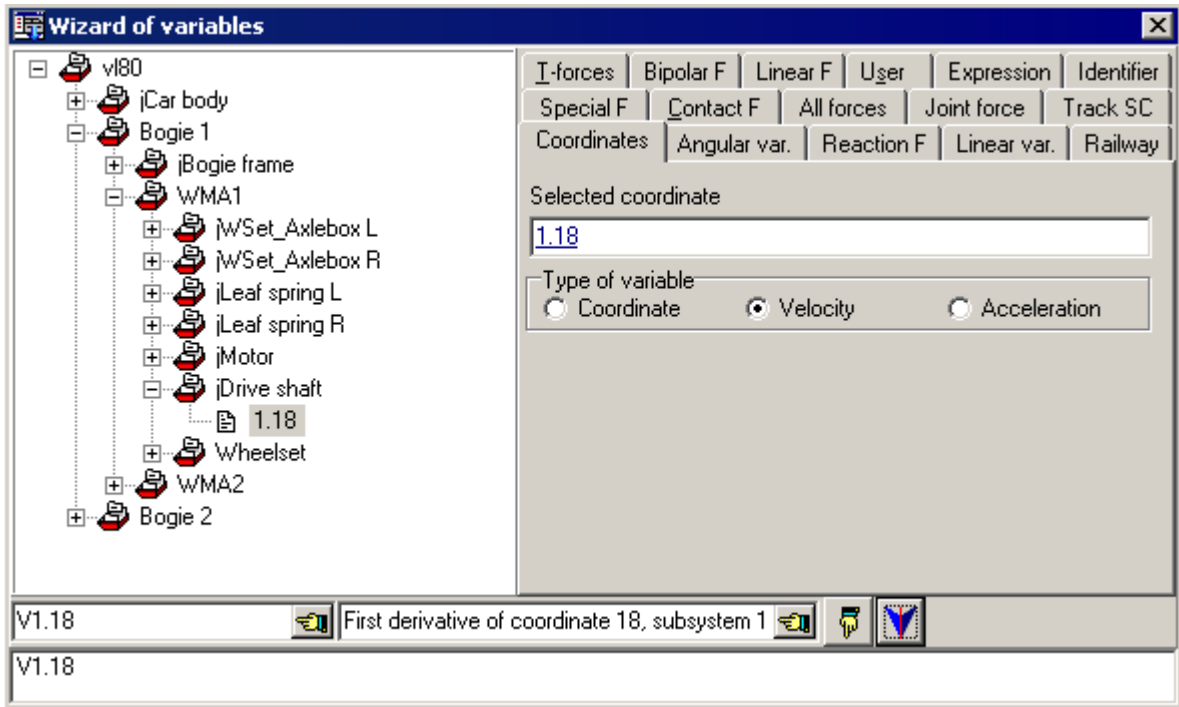



Fig.8.27. Creation of variable: velocity of drive shaft rotation relative to the motor casing

- Create a variable corresponding to joint velocity for setting the initial value: open the **Coordinates** tab, select the necessary joint coordinate in the list located in the right part of the wizard. Set the variable type **Velocity** and send it to the container by the  button, Fig. 8.27.
- Drag the variable by the mouse and drop it into the left cell of the co constraint data row, Fig. 8.26.

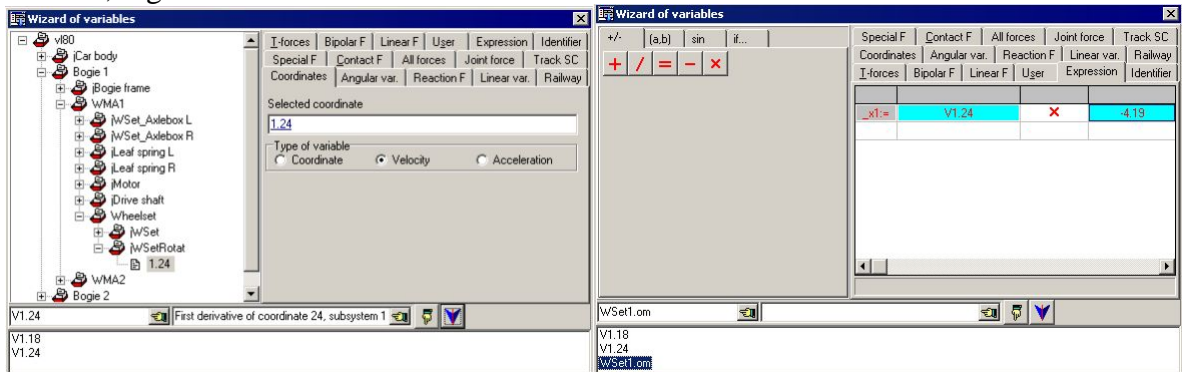





Fig.8.28. Creation of variables: angular velocity of wheelset and computed value of drive shaft angular velocity

- Create a variable corresponding to the desired value of the velocity and set it in the right cell of the constraint row. Consider an 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 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 about the wheels axis); this is the joint velocity in the joint jWSetRotat.

- Open the Expression tab of the wizard. Add an operation by the  button (multiplication).
 - 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, Fig. 8.26.
- 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 default value), the constraints are automatically loaded at each load of the object in the simulation module. Constraints are used at 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.

8.2.7.2. Support of braking mode

Simulation of braking process is discussed in the user's manual chapter 15 devoted to the longitudinal train dynamics. Here we consider only adding to a vehicle model (both car and locomotive) force elements, which allow 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.

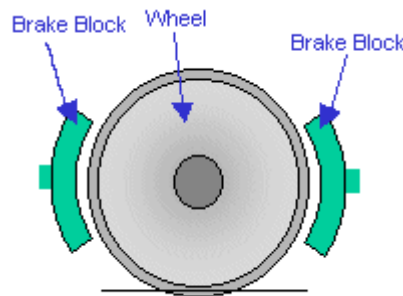


Fig.8.29. Bilateral brake blocks

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, Fig. 8.29. 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

$$\pm \text{braking_force} / N_w * R_w$$

где N_w – число колесных пар, R_w – радиус колеса. The + sign is set if the wheelset forward motion corresponds to the negative angular velocity (it is possible often 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 Fig. 8.25.

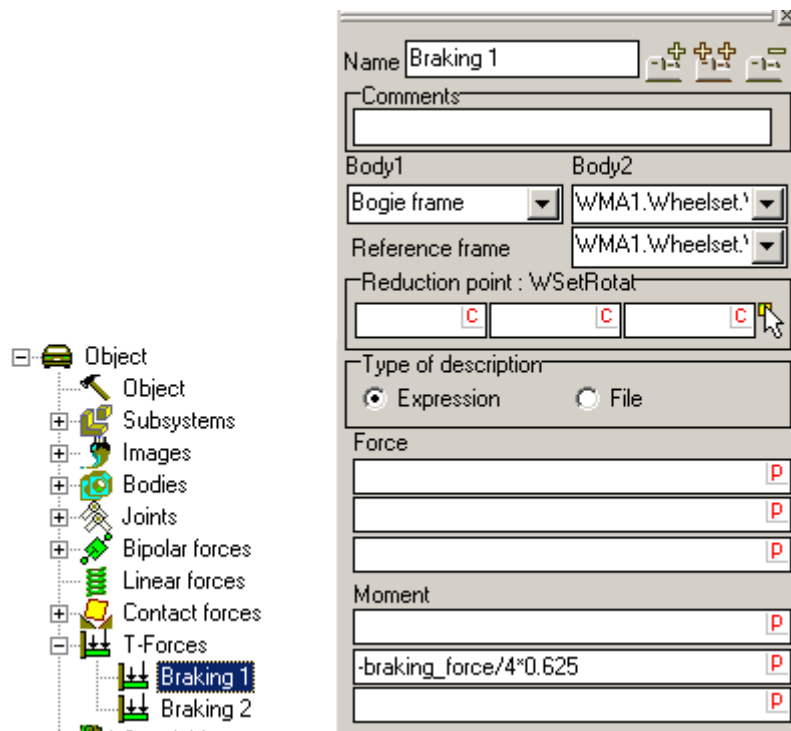


Fig.8.30. Setting braking torque for locomotive VL80

Such a moment should be added for each of the wheelsets using the *T-force* element, Fig. 8.30. 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 depend both on direction of rotation of the wheelset, and on position of the brake block relative to the wheelset.

8.3. Rail track

8.3.1. Track geometry

Track geometry includes the following components:

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

8.3.1.1. Geometry of rails in an ideal straight section

The geometry of rails in an ideal straight section includes:

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

Different profiles can be assigned to the left and right rails.

The rail profile can be changed with the distance along the track (Sect. Rail profile evolution along the track). 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 ($S_2 > S_1$). Then the program computes an intermediate profile R(S) for every necessary position $S \in [S_1, S_2]$ as a linear interpolation of profiles R1 and R2 such as $R(S_1) = R_1$, $R(S_2) = R_2$.

A rail profile in UM should be determined in a special system of coordinates (SCR), Fig.8.31. 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.

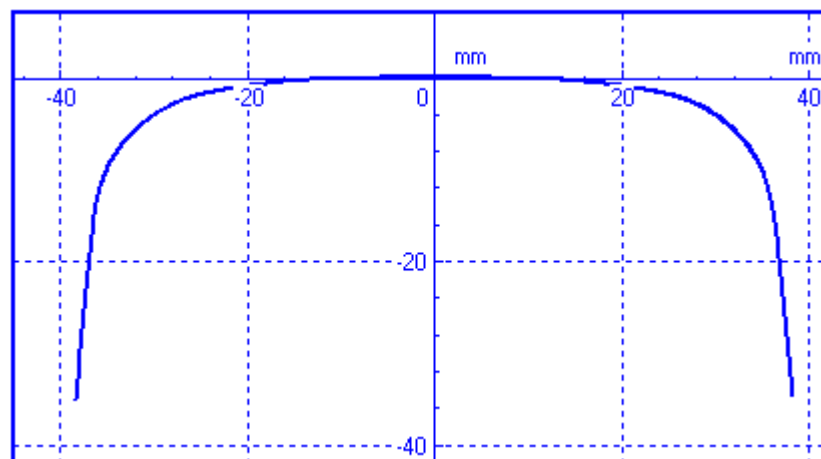


Fig.8.31. System of coordinates of rail

Lateral position of rails in an ideal straight section is set by the *standard track widening* Δy relative to the wheelset base (Fig.8.32)

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

where L_r is the distance between the rail head centers, L – wheel base (distance between running circles, Sect.8.2.3.3). In other words, the *standard track widening* is the lateral distance between the origins of two profile frames (wheel and rail) at ideal symmetric position of the wheelset.

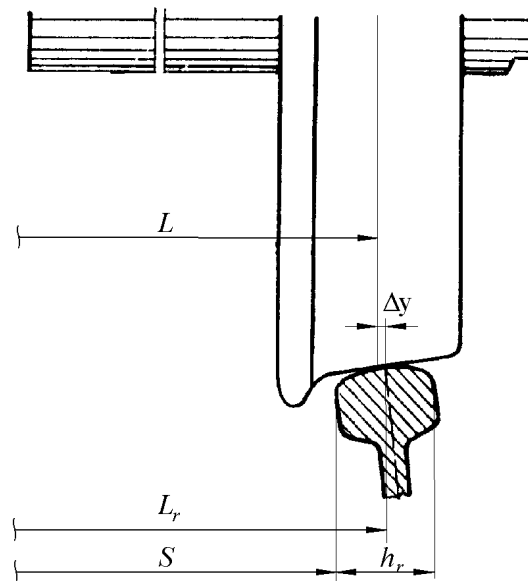


Fig.8.32. On standard track widening

Further expression for the widening:

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

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

Rail inclination α_{r0} is the angle between the rail profile axis of symmetry and the vertical in an ideal straight track. Unit for angle is radian. The angle is positive for inclination inside the track. Default value is $\alpha_{r0} = 0.05$ rad.

8.3.1.2. Macro geometry of curve

The following types of curves are available in UM (Fig.8.33):

- 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 Fig.8.33:

- $L0$ – straight section before the curve;
- $P11$ – first transition for the right curve;
- $S1$ – length steady curve;
- $R1$ - radius of steady curve;
- $H1$ – cant for the outer rail;
- $P12$ – second transition for the right curve;
- dy – additional gauge widening in curve;
- L – 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. The curvature in transitions changes according approximately linear law.

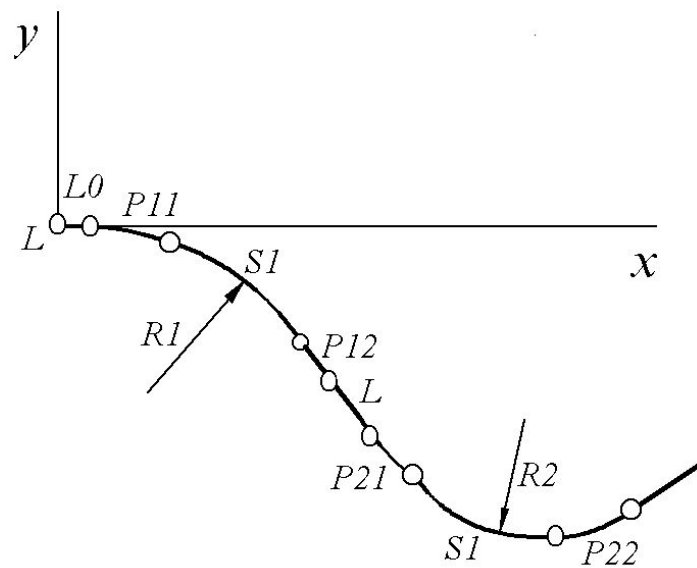


Fig.8.33. On macro geometry of curve

Growths and decreases of the cant at transient sections are linear. It is possible to smooth the vertical junctions at ends of the transition by arc of circle.

Additional widening dy of track in a curve is proposed automatically: $dy = 10$ mm for $R \in [300, 350]$, $dy = 15$ mm for $R < 300$ m. The widening is realized as symmetric lateral shift of both the rails on a half of the widening. The widening on transient sections is the linear function of the position. The user may set own widening.

See Sect. Input track macro geometry parameters

Let us introduce a number of designations for curve parameters computed by UM:

$X(s), Y(s)$ – equation for a curve, s is the arc coordinate (length along the curve);

$\psi_b(s)$ – angle between the X-axis and the tangent to the curve;

$\rho(s)$ – curvature radius;

s_o – traveled path.

Consider running of a vehicle in a curve when the non-inertial SC0 is used (Sect. 8.2.2)

Let ω_0, ε^I be the angular velocity and the angular acceleration of SC) relative to the vertical axis, a^I be a centrifugal acceleration. The following relations take place:

$$\omega_0 = V / \rho(s_0), \quad \varepsilon^I = V^2 \kappa'(s_0), \quad a^I = V^2 / \rho(s_0).$$

Here κ' is the derivative of the curvature of the curve.

The following inertia force and inertia moment are taken into account for every body when the non-inertial SC0 is used:

$$F^I = -m(a^I + x_0 \varepsilon^I),$$

$$M^I = -J \varepsilon^I,$$

where m, J are the mass and moment of inertia of a body relative to the vertical axis, x_0 is the longitudinal position of the body center of gravity relative to the vehicle body CG. Note that the angular acceleration ε^I is non-zero at transitions only.

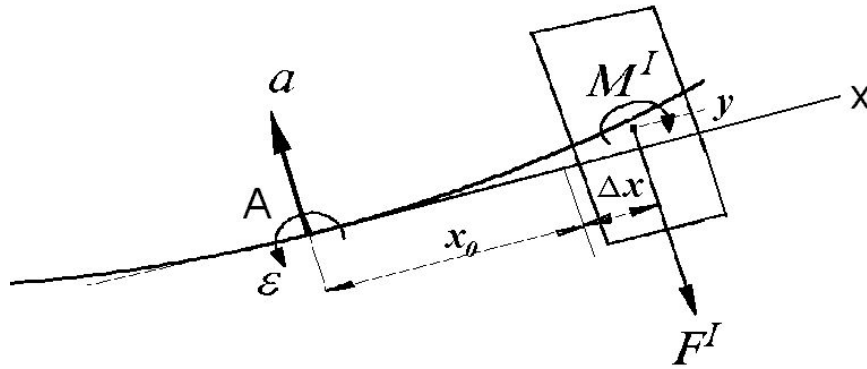


Fig.8.34. Inertia force for the non-inertial SC0

Consider an example of a S-curve for the following parameters:
 $L_0=10m$, $p_{11}=70m$, $s_1=150m$, $R_1=300m$, $p_{12}=60m$, $h_1=0.09m$,
 $L=10m$, $p_{21}=50m$, $s_2=140m$, $R_2=330m$, $p_{22}=70m$, $h_2=0.1m$.

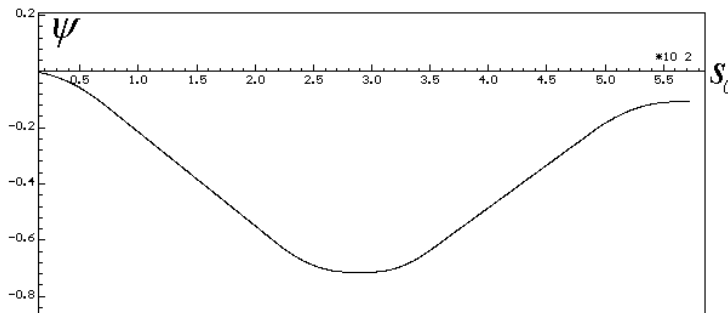


Fig.8.35. Angle between rail and X –axis

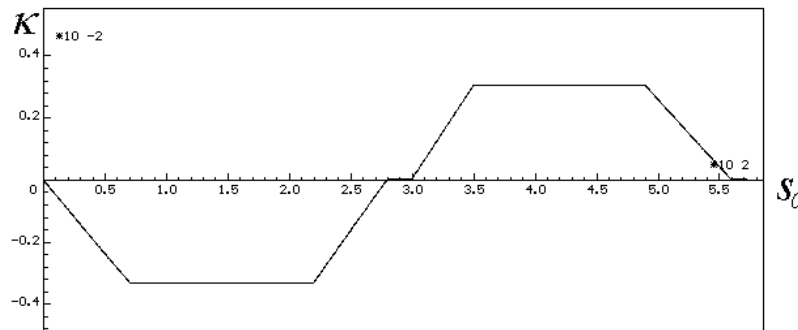


Fig.8.36. Curvature

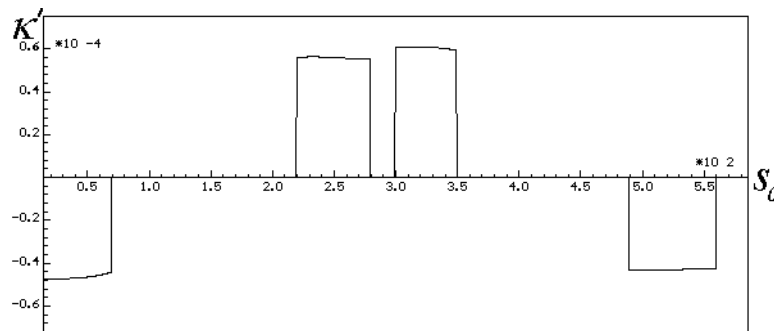


Fig.8.37. Derivative of the curvature

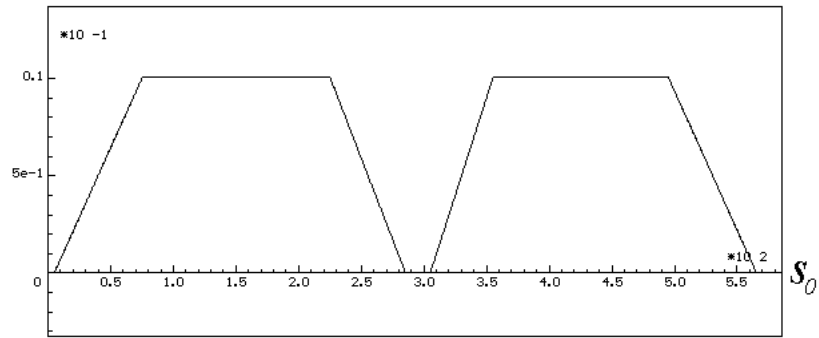


Fig.8.38. Gauge widening

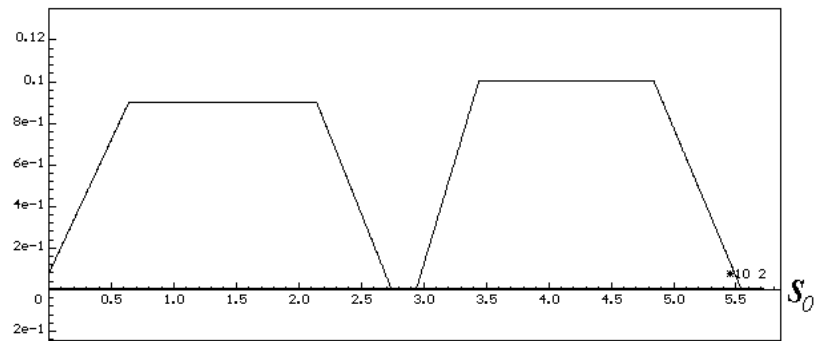


Fig.8.39. Cant for outer rail

8.3.1.3. Switch geometry

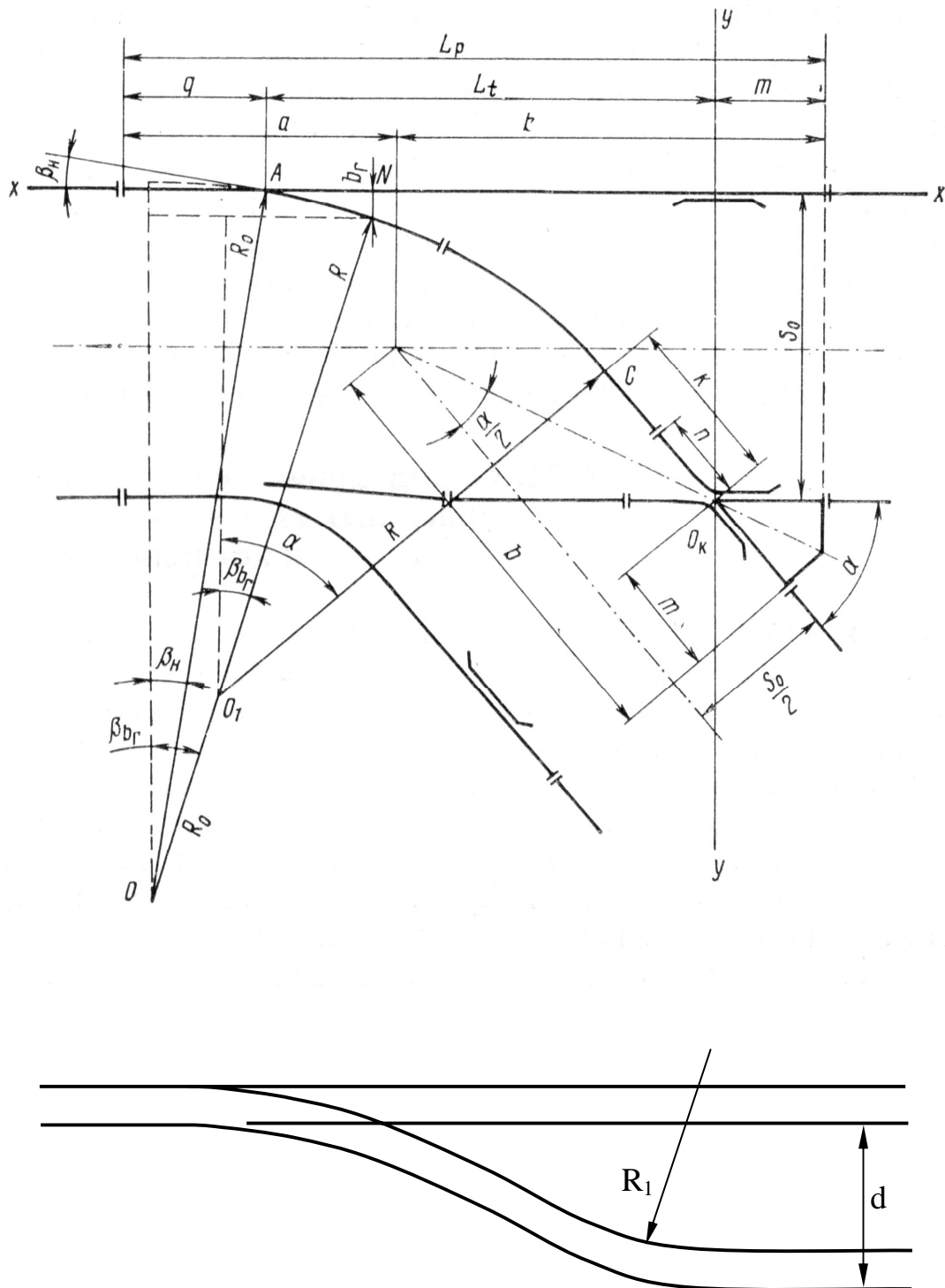


Fig.8.40. Geometric parameters of a switch

Motion in left and right switches are implemented in UM. The basic geometric parameters of a switch are shown in Fig.8.40. The following parameters are used for description of the switch.

q – stock rail overhang;

β_n - initial angle;

- α - switch angle;
- R_0 - radius of point;
- R - radius of switch;
- b_r - switch deviation for R_0 ;
- m - frog tail length;
- d - track spacing.
- R_1 - radius behind frog.

The parameters define fully the switch geometry and some additional parameters in Fig.8.40. like

- L_p - full switch length;
- L_t - theoretical length;
- κ - tangent section before frog.

UM50 includes the standard R65 1/11 and R65 1/9 switches. The user can create files with any switch parameters.

8.3.1.4. Track system of coordinates

A track system of coordinate is introduced for each body i of the vehicle (TSK). Its origin W_i coincides with the projection of the body center of mass on the ideal central axis of the track. 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.

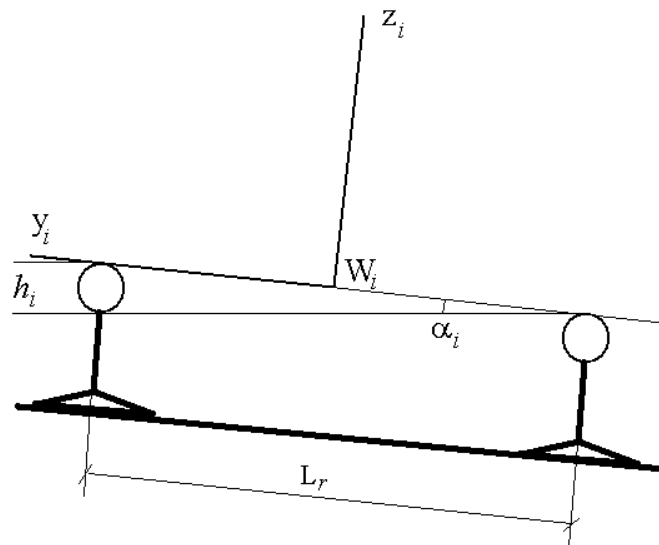


Fig.8.41. TCSi axes in presence of a cant

A cant of the outer rail h_i (Fig.8.41) 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. *Kinematic characteristics relative to track system of coordinates.*

8.3.1.5. Track irregularities

Vertical and horizontal irregularities of rails are stored in unformatted files *.way and assigned to rails before the simulation in the Simulation Module. Step size of the irregularities is 0.1 m (Sect.8.5.1.2).

If horizontal irregularities lead to excessively narrowed gauge and a simultaneous contact of two flanges of a wheelset takes place, the program sends a message about the error and stops the simulation.

Long raise or/and lowering of the track should be realized as irregularities.

It is possible to program user's irregularities in the Control File (8.5.1.2.2).

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 a growth is considered on the next 20 meters by a multiplier, which increases linearly from 0 to 1. As an example consider an irregularity of a constant height of 1 mm. In the reality the irregularity in Fig.8.42 will be applied. This is valid for file irregularity only, and not for the programmed one.

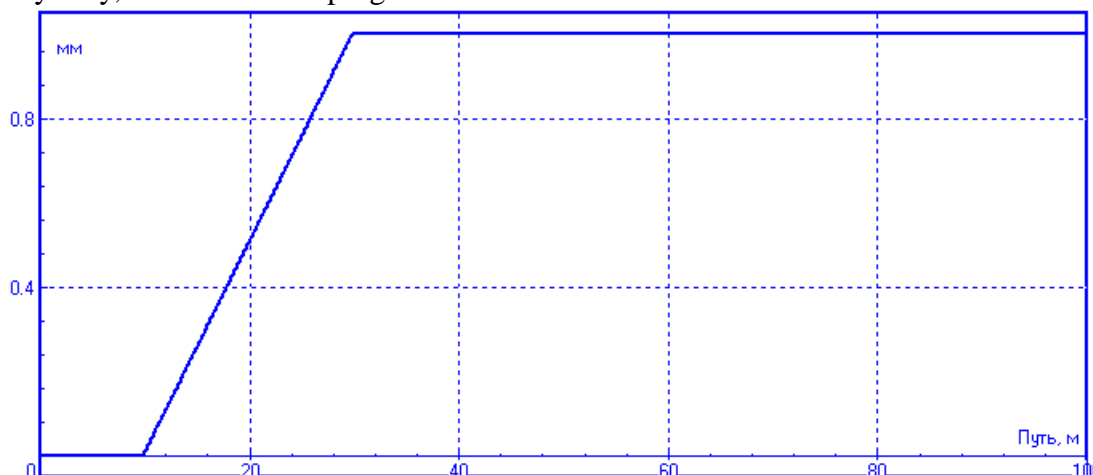


Fig.8.42. Smoothing an irregularity

8.3.2. Elastic, dissipative and inertia properties of track

Elastic and dissipative properties of a track are taken into account by introducing linear stiffness and linear damping for rails in vertical and lateral directions.

The user can introduce variable stiffness and damping along the way using programming in the Control File.

Inertia properties of the track can be taken into account by usage added masses in the vertical and lateral directions for wheelset bases.

8.4. Wheel-rail contact

Three main parts of contact computation can be pointed out

- 1) contact geometry, i.e. computing locations of contact points for a definite position of a wheelset;
- 2) computation of kinematical characteristics at contacts (creepages and spins, rail velocities);
- 3) 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.

8.4.1. Algorithms for wheel-rail contact geometry

To make computation of contact points faster and more reliable, two main ideas are realized. Firstly, if rail profile does not change along the track (Sect.8.3.1.1), computation of contact points is executed once for given rail and wheel profiles before the simulation start. UM creates tables of contact coordinates for different relative positions of the wheel profile relative to the rail profile (lateral displacement and rotation about the longitudinal direction). By simulation the coordinates of contacts are interpolated with the help of these data.

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.

8.4.1.1. Algorithm for computation of nearest points between two profiles

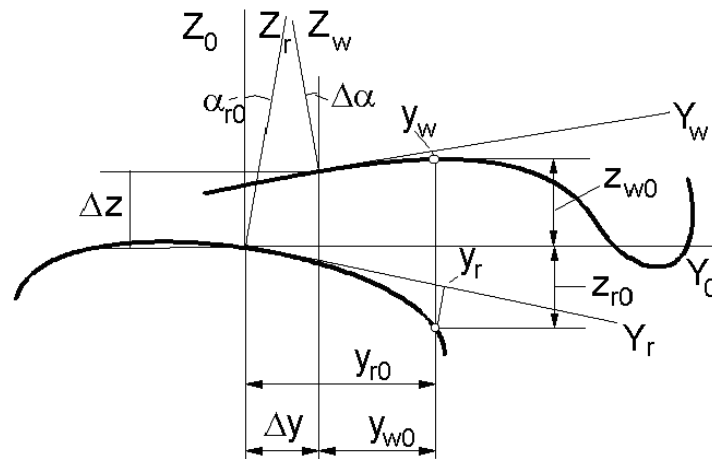


Fig.8.43. Relative position of profiles

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. 8.3.1.1), Fig.8.43. 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 α_{r0} is zero. The SCR $Y_r Z_r$ is inclined on α_{r0} relative to SCR0. Position of SCW (system of coordinates of the wheel profile, Sect.8.2.3.2) $Y_w Z_w$ relative to SCR0 is defined by coordinates of the origin Δy , Δz , and the angle $\Delta\alpha$. Coordinates in Fig.8.43 are positive. All angles are considered to be small.

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

$$\min \delta z = \min(z_{w0} - z_{r0})$$

Obviously, the solution does not depend on Δz .

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 δz .

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

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

$$z_{r0} = z_r - y_r\alpha_{r0},$$

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: $y_{w0} = y_{r0} - \Delta y$, Fig.8.43. 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

$$y_w^0 = y_{w0},$$

$$y_w^{i+1} = y_{w0} + z_w(y_w^i)\Delta\alpha, \quad i = 0, 1, 2, \dots$$

could be used.

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

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

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 δz_{\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

$$\delta z(y_{ri}), i = 1, 2, \dots, y_{ri+1} - y_{ri} = h.$$

and select the points with the minimal value δz . After that the process is repeated near the found point with smaller step size (0.1mm).

8.4.1.2. 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 h relative to the rail profile. UM

allows two kinds of pairs of profiles: profiles with one-point contact only and profiles with possible two-point contact (Fig.8.44).

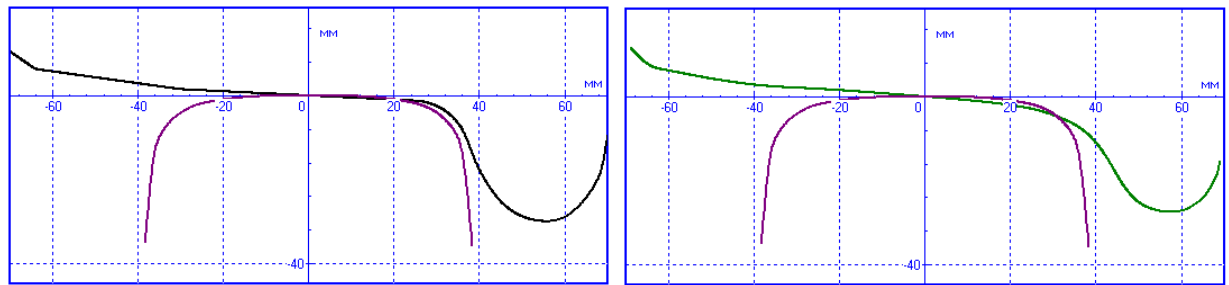


Fig.8.44. Pairs of profiles allowing two-point contact (left) and one-point contact only (right)

In a *one-point contact mode* the contact point position depends on the lateral shift Δy and the angle $\Delta\alpha$, Fig.8.43. Coordinates of the contact point correspond to the profile point with the minimal distance in the vertical direction (Sect. 8.4.1.1). Thus, the table contains coordinates of the contact point in SCR and SCW for a discrete set of variables $\Delta y_i, \Delta\alpha_j, i=1..N_y, j=1..N_\alpha$ with a proper step size of discretization.

Computation of one-point contact is realized according to the algorithm in Sect 8.4.1.1. Here we consider some features of computing a two-point contact.



Fig.8.45. Types of contact for a pair of profiles allowing two-point contact: one-point contact, two-point contact and creeping up

Fix the value $\Delta\alpha$ and give the maximal value of the lateral displacement (Fig. 8.45, left). In this position of the profile we have obviously a one-point contact. Now decrease Δy without changing $\Delta\alpha$, i.e. shift the wheel profile to the left. There exists some critical value of the lateral coordinate Δy^* for which a two-point contact occurs, if the profiles allow it (Fig. 8.45, center). *Further decreasing Δy 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 (Fig. 8.45, left). Secondly, the contact on the running surface disappears (the corresponding normal force becomes zero) and the wheel goes to the creeping up state (Fig. 8.45, right). This mental experiment is the base of an algorithm for computing the two-point contact. Let us start discussions of the algorithm.

Choose intervals for possible values of the $\Delta\alpha$, Δy coordinates with a definite reserve,

$$\Delta\alpha \in [\Delta\alpha_{\min}, \Delta\alpha_{\max}], \quad \Delta y \in [\Delta y_{\min}, \Delta y_{\max}]$$

and introduce a homogeneous discretization of the intervals, N_α and N_y are numbers of subintervals. Now for each fixed value $\Delta\alpha_j, j=1..N_\alpha+1$ compute coordinates of contact point on the running surface according to Sect.8.4.1.1 successively decreasing the lateral shift $\Delta y_i, i=N_y+1, \dots, 1$. If the profiles allow the two-point contact, the position of the contact point for some Δy_i will change by a large enough jump. This fact means that the flange contact occurs inside the latest change of Δy . Denote this value of Δy as Δy^* . 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, η_y, η_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.

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 0.0001 mm), and the critical value Δy_j^* is defined more exactly.

To model the process of creeping up the wheel, 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 ∇h^* 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 γ . UM uses an approximate analytic expression for the value of this parameter. To get it consider a simplified geometrical model of the flange-rail contact, Fig.8.46. 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.

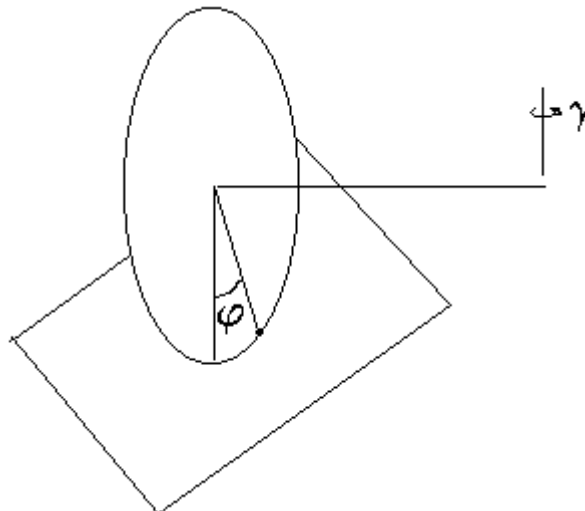


Fig.8.46. Simplified model of the flange contact for computing an overswing

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

$$\tau = (r_w \cos \varphi, 0, r_w \sin \varphi)^T,$$

where r_w is the wheel radius at the flange contact.

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

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

yields the following orthogonal condition

$$n^T A_{01} \tau = 0,$$

or, taking into account that angles γ and φ 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

$$x_c = k \gamma, \quad k = r_w n_{ry} / n_{rz}.$$

8.4.1.3. 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.

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 if 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 in dependence on the lateral shift is used for evaluation of the equivalent conicity of a pair of profiles.

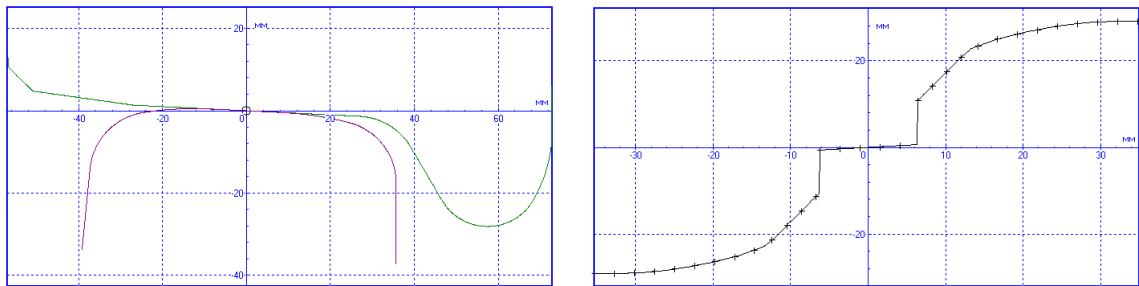


Fig.8.47. A pair of new profiles: R65 and Russian wagon wheel (left); the corresponding RRD (right)

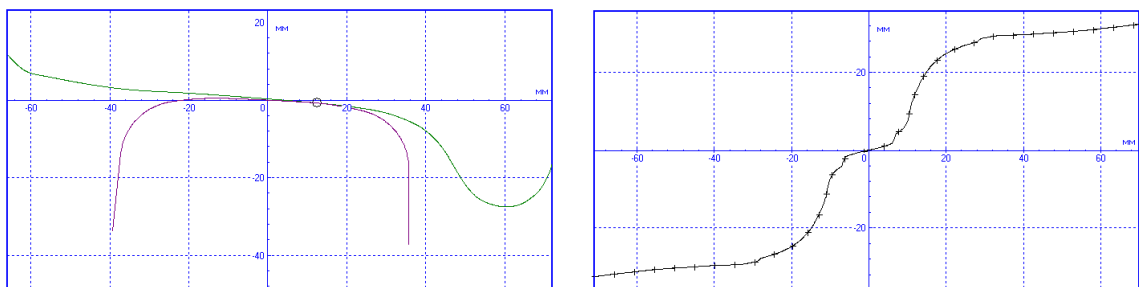


Рис.8.48. A Pair of profiles: R65 and DMetI (left), the corresponding RRD (right)

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 $\lambda=1/20$, Fig. 8.47. Really, the wheel profile function is

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

and

$$\Delta r = 2\lambda y .$$

For worn or curved profiles the RRD curve is non-linear even for small shifts of the wheelset, Fig.8.48. In this case the notion of equivalent conicity λ is introduced as a mean value for a definite lateral shift of the wheelset Δy according to the formula

$$\min_{\lambda} \int_0^{\Delta y} f(y)(\Delta r(y) - 2\lambda y)^2 dy, \tag{1}$$

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 Δy interval. Often Δy 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) in the current UM version. The default value of the standard deviation is $\sigma = 2.5$ mm.

Differentiating Eq.(1) with respect to λ yields the formula for the equivalent conicity as

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

Contact angle parameter

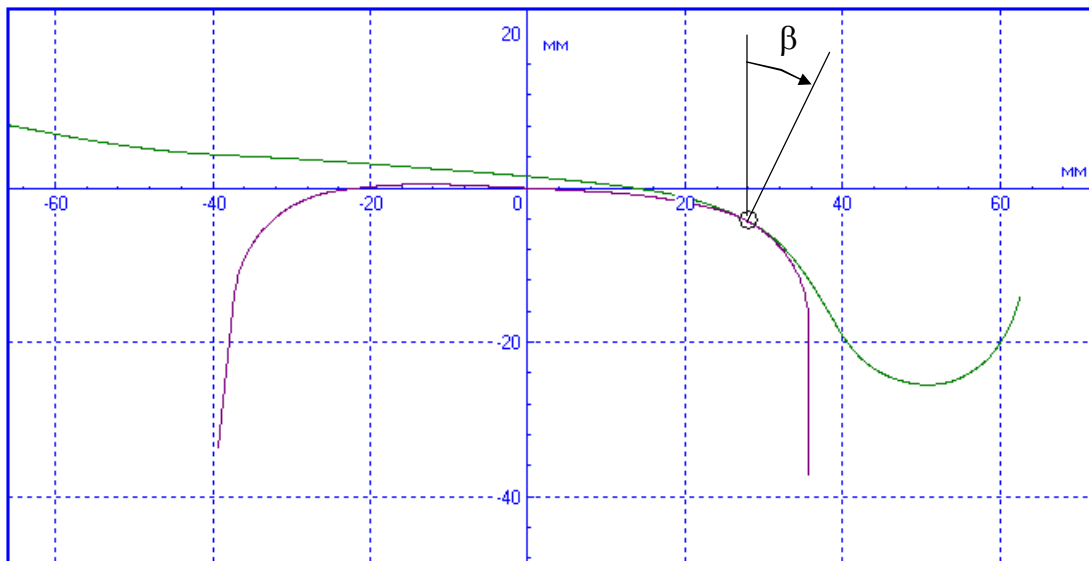


Fig.8.49. Contact angle

Contact angle β 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, Fig. 8.49.

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.

The dependence

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

on the lateral shift is used for definition of the *contact angle parameter*. Here L is the distance between the wheelset rolling radii, Fig. 8.6.

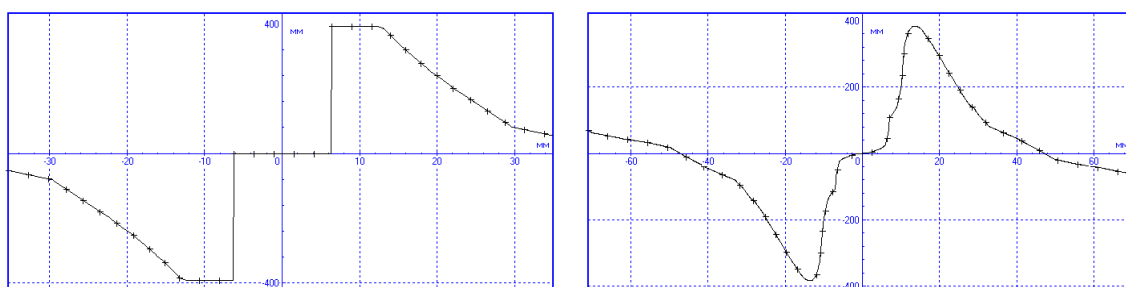


Fig.8.50. Function $E(y)$ for pairs of profiles in Fig. 8.47., 8.48.

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

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

The formula similar to the equivalent conicity is used for evaluation of ε

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

Simplified geometry of contact

The following simplified formulas 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:

$$r_l = r_0 + \lambda y_l, \quad r_r = r_0 - \lambda y_r,$$

$$\beta_l = \beta_0 + \frac{2\varepsilon y_l}{L/2}, \quad \beta_r = \beta_0 - \frac{2\varepsilon y_r}{L/2}.$$

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^* 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 the contact angle β^* . The longitudinal contact overswing is computed according to the results of the previous section.

The longitudinal and lateral creepages in the contact point on the running surface are computed by the formulas

$$\xi_x = \frac{v_{x0} - \Delta r \omega}{v}, \quad \xi_y = \frac{v_{y0}}{v},$$

where v_{x0}, v_{y0} are the longitudinal and lateral components of velocities of the point on the wheel profile corresponding to the running circle, v is the longitudinal velocity of the wheelset enter.

Remark. The Mueller's method of creep force computation is used in case of simplified geometry of contact, see Sect. 8.4.3.2.1. Mueller's method.

8.4.2. Tool for analysis of pairs of profiles

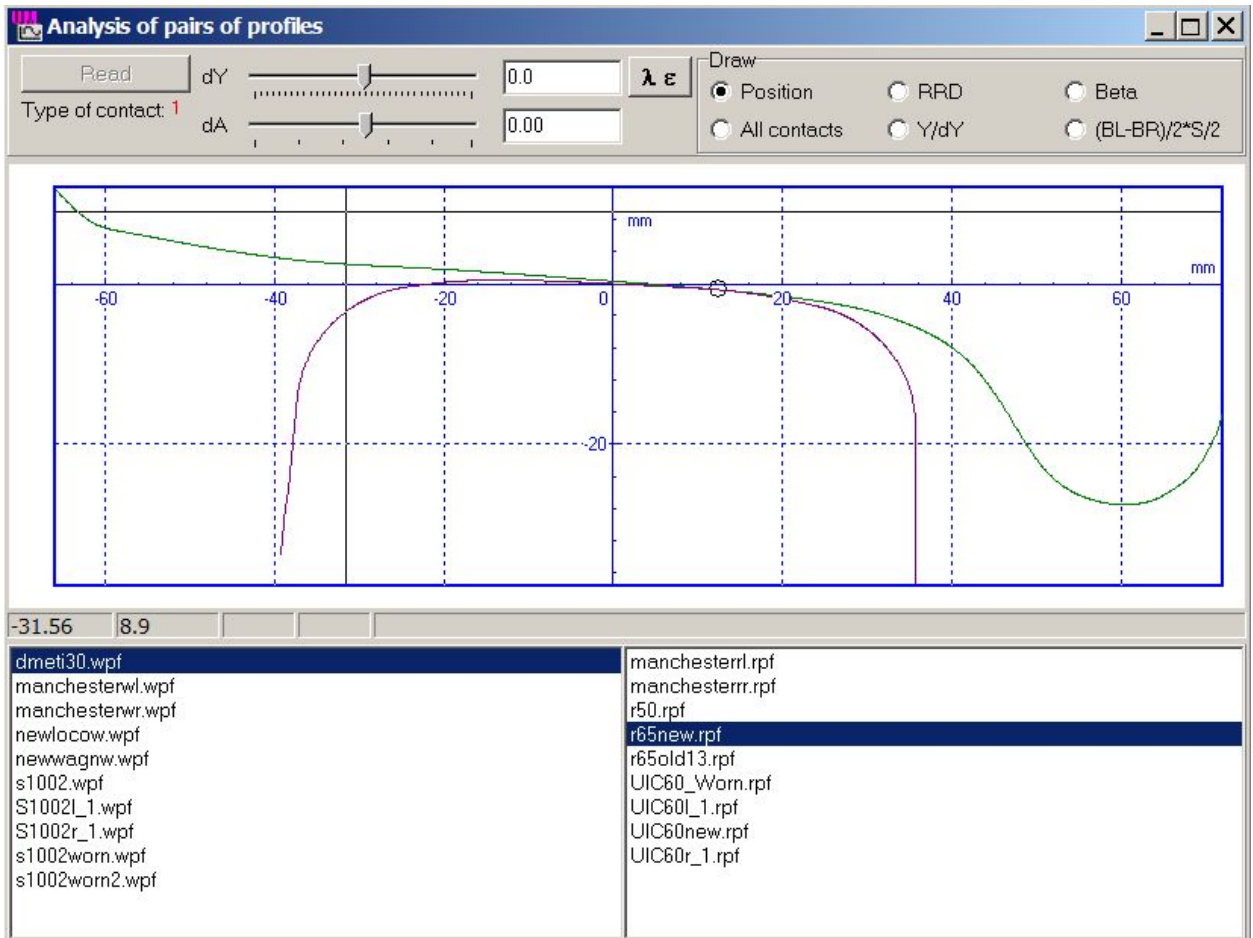



Fig.8.51. Profile analysis window

The tool for analysis of profiles is available in the simulation program for the current values of the track gauge and rail inclination (Sect. 8-25. Geometry of rails in an ideal straight section, 8.5.2.4.1. Assignment of track irregularities and rail geometry parameters (gauge, rail inclination)). Use the **Tools | Analysis of pairs of profiles** menu command or the  button on the tool panel to open the window.

Container with profiles

The container contains lists wheel (left) and rail (right) profiles located in the standard directory <Path to UM>\bin\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 select a pair of analyzed profiles, select them in the lists by the mouse Fig. 8.51., and click the Read button on the top of the window.

Analysis results

The following results are available with the tool.

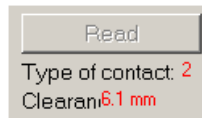


Fig.8.52. Type of contact and clearance on a side

- **Type of contact** (one or two-point contact), Fig. 8.52.
- **Clearance** (on a side) is the lateral shift of the wheel relative to the rail, which lead to the double contact (for profiles allowing two-point contact), Fig. 8.52.

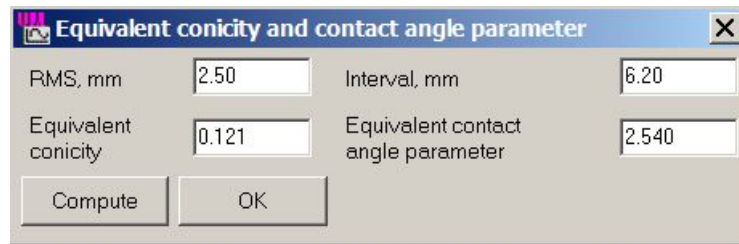


Fig.8.53. Equivalent conicity and contact angle parameter

- λ, ϵ - Equivalent conicity and contact angle parameter, Sect. 8.4.1.3. The parameters are computed for given values of the standard deviation of the lateral shift (RMS, mm) and averaging interval, Fig. 8.53.

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

- **Position** shows the contact point 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 , Fig. 8.51. 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 changes from -1.4 to 1.4 degrees.

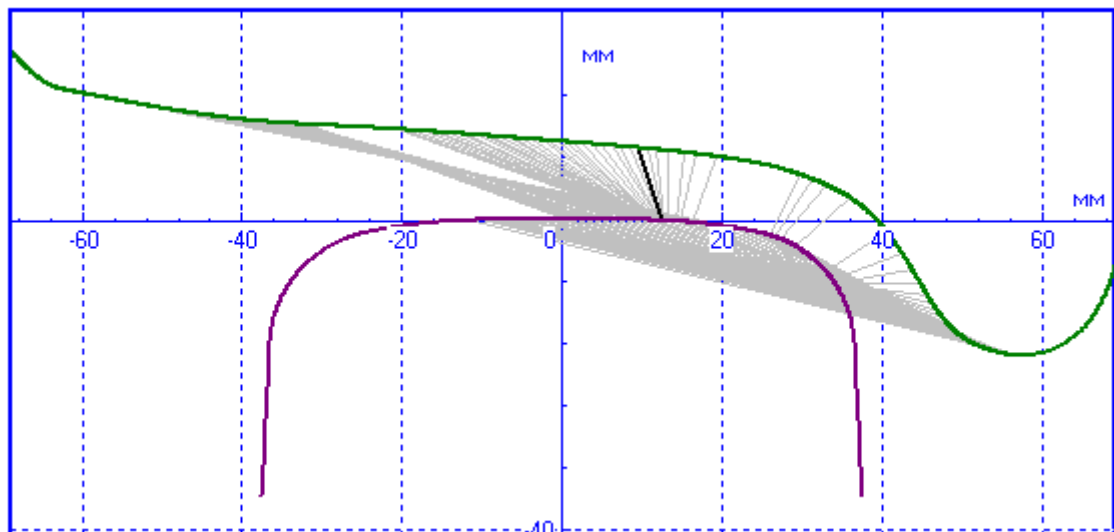


Fig.8.54. All contacts of a pair of profiles

- **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. 8.4.1.3.
- **dY/Y:** Y coordinates of contact point on rail and wheel profiles in SC of the corresponding profile depending on the lateral wheel shift.
- **Beta:** contact angle versus lateral wheel shift, Fig. 8.49.
- **(BL-BR)/2*L/2:** plot of the function

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

used in computation of contact angle parameter, Sect. 8.4.1.3.

8.4.3. Contact forces

8.4.3.1. Method for computation of rail deflections and contact force

A rail is considered in UM as a massless force element. This means, both stiffness and damping of the rail is taken into account, but not the inertia properties. Generalized coordinates are not introduced for the rail and its lateral and vertical deflections must be computed from the equilibrium equations.

The following assumptions take place:

- 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 Δy_r , vertical Δz_r deflections (Fig.8.55), which are parallel to the corresponding SC of the track;
- rail roll is not considered.
- 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.

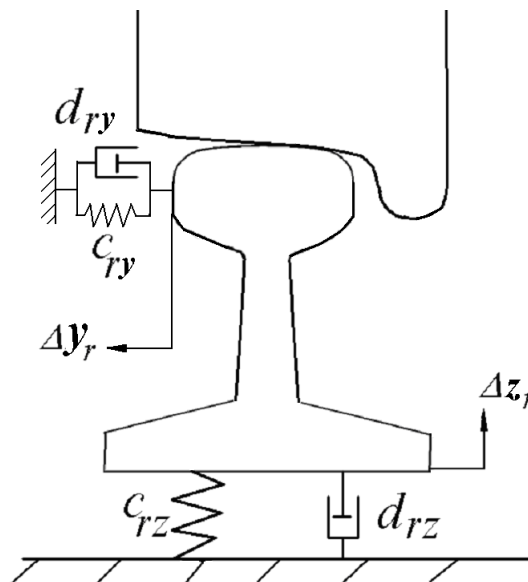


Fig.8.55. Rail as massless force element

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:

$$\begin{aligned} R_y &= -c_y \Delta y_r - d_y \Delta \dot{y}_r, \\ R_z &= -c_z \Delta z_r - d_z \Delta \dot{z}_r. \end{aligned} \tag{8.1}$$

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 Fig. 8.56. Longitudinal forces are not shown in this figure.

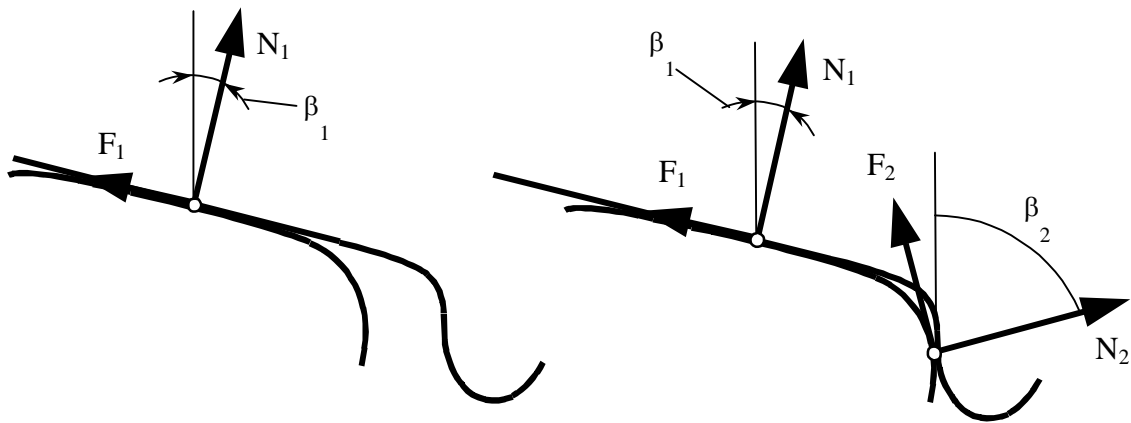


Fig.8.56. Forces acting on wheel at one- and two-point contacts

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

$$R_y - F_1 \cos\beta_1 + N_1 \sin\beta_1 = 0, \tag{8.2}$$

$$R_z - N_1 \cos\beta_1 - F_1 \sin\beta_1 = 0.$$

Analogous equations are valid for a two-point contact

$$R_y - F_1 \cos\beta_1 + N_1 \sin\beta_1 - F_2 \cos\beta_2 + N_2 \sin\beta_2 = 0, \tag{8.3}$$

$$R_z - N_1 \cos\beta_1 - F_1 \sin\beta_1 - N_2 \cos\beta_2 - F_2 \sin\beta_2 = 0.$$

Here β_1, β_2 are the angles between the normal to the rail at contact and the axis perpendicular to the track.

Eq. (8.2) for a one-point contact and (8.3) 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 not going into details.

1. 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 Δz_r is not an independent variable because when the lateral Δy_r deflection is known, the value of Δz_r can be obtained from the geometry of the contact. This value is equal to the value δz , Sect.8.4.1.1.
2. Calculation of the rail deflections and contact forces is an iterative process. Iterations include two cycles: internal end external. The internal iterations are used for solving equations (8.2), (8.3) 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 (8.2), (8.3) are solved for corrected values of creep forces.

Consider some features of realization of internal iterations.

1. Lateral deflection Δy_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 .

2. The previous value of the Δy_{r1} variable is used as the initial value for iterative solving the equilibrium equations. Iterations make the value more accurate.
3. 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. 8.4.1.2), computing the vertical rail deflection Δz_r , the angle β_1 (and β_2 for the two-point contact). The second part of the rail deflection Δy_{r2} is computed from the geometrical conditions.
 - Forces R_y, R_z are computed according to Eq. (8.1).

The normal force N_1 for a one-point contact is obtained from Eq. (8.2); in the case of a two-point contact the normal reactions N_1 and N_2 are computed from Eq. (8.3).

4. The new value of the deflection Δy_{r1} is computed and stored for the next iterations.

8.4.3.2. Algorithms for computing creep forces

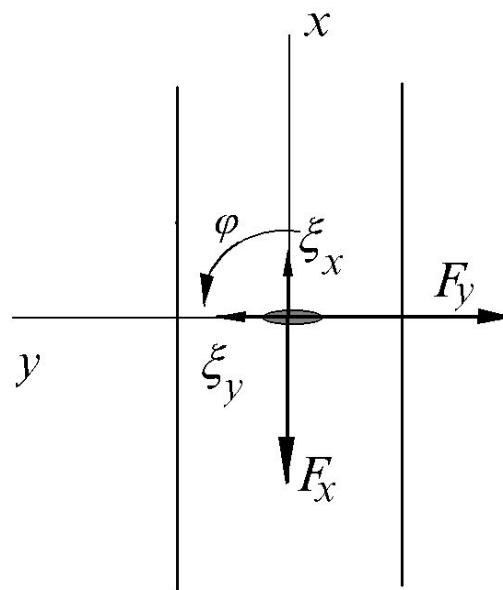


Fig.8.57. Creepages, spin and creep forces

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

$$F_x = F_x(N, \xi_x, \xi_y, \phi, p), \quad F_y = F_y(N, \xi_x, \xi_y, \phi, p)$$

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;

ξ_x, ξ_y are the longitudinal and lateral creepages;

ϕ is the spin;

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:

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

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

Models of the creep forces are used both for the one-point and for the two-point contact.

The following algorithms for computing the creep forces are available with UM.

8.4.3.2.1. Mueller's method

Mueller's method is the simplest one for computation of the creep forces according to the following analytic expressions [1]

$$\xi = \sqrt{\xi_x^2 + \xi_y^2}, \quad P = 0.001N, \quad k_c = P(235 - P(2.4 - 0.01P))$$

$$F_{xy} = - \frac{1000k_c}{\left(1 + (k_c \xi / f / P)^m\right)^{1/m}},$$

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

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.

8.4.3.2.2. FastSim

FASTSIM is the well-known and the most frequently used algorithm for creep forces by Kalker [2]. It takes into account both the spin and the geometry of the contact surfaces. The algorithm is used for calculation of creep forces both for one- and two-point contacts.

Consider some features of the algorithm, which are important for its understanding and a correct usage.

FASTSIM requires the following parameters and variables for computation of the creep forces:

- Rail and wheel material properties, which are supposed to be equal: elasticity modulus and Poisson ratio – set by the user;
- Current geometric characteristics of the contact point: principal curvatures of the contact surfaces – computed by the program;
- The normal force N in the contact – computed by the program.

This data is used by FASTSIM to compute the semi axes of the elliptic contact patch according to the Hertz theory:

- Current values of the longitudinal and lateral ξ_x, ξ_y , creepages and spin ϕ – computed by the program.

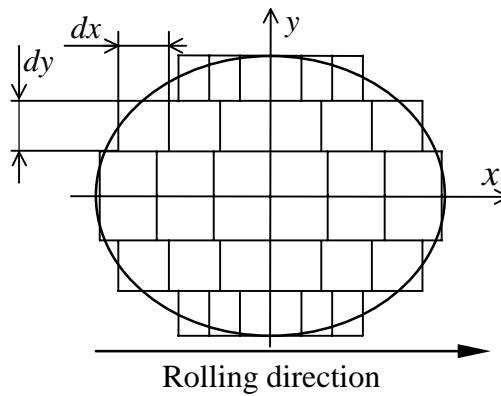


Fig.8.58. Discretization of the contact ellipse

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 (Fig.8.58). 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.

8.4.3.2.3. FastSim analytic

FASTSIM_A is our 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 $n=10$ is at least two times faster than FASTSIM and gives quite similar numeric results for computing creep forces in case of moderate spin values ($\varphi < 0.5$)

8.4.3.2.4. Minov's model

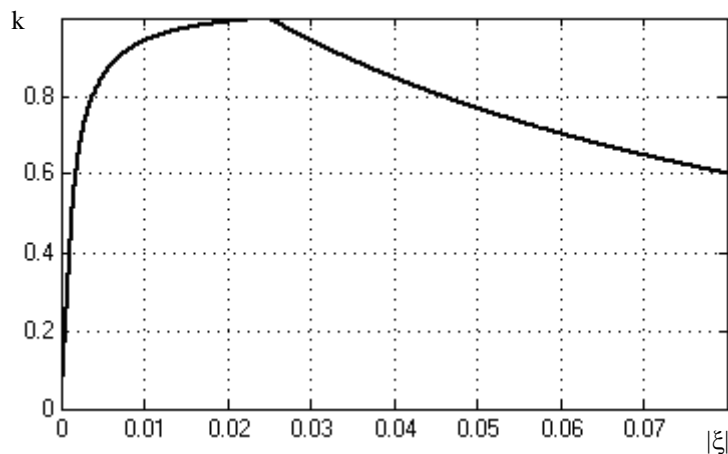


Fig.8.59. Sticking coefficient versus creep in the Minov's model

The method is used in simulation of locomotive in traction modes, and based on experimental dependence of sticking (friction) force on lateral creep, Fig. 8.59. 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

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

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

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

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

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

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

v_0 , km/h	0-5	5-20	20-40	40-120
χ	0,9	0,6	0,5	0,35

Creep forces are computed according to the formulas

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

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

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

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

8.4.3.3. Non-elliptical contact model

Taking into account that the bodies in contact are quasi-identical, i.e. material properties of wheel and rail are the same, the contact problem can be divided into two independent ones: the normal contact problem and the tangential contact problem.

Normal Contact Problem. According to [3], an approximate contact patch is defined in the following way. The wheel and the rail are considered as a body of revolution and a cylindrical surface, respectively. The surfaces are intersected in a depth δ as rigid bodies, fig. 8.60. The function $u(x, y)$ specifies the interpenetration of the surfaces at the point (x, y) . It satisfies the linearized equation

$$u(x, y) = \delta - z(x, y), \quad z(x, y) = \frac{x^2}{2R} + h(y) \quad (8.4)$$

where $z(x, y)$ is the function, which specifies the distance between the points of the bodies for $\delta = 0$, R is the wheel radius at the contact point, $h(y)$ is the distance between the points of profiles in $x = 0$ plane.

Edge of the approximate contact patch is determined as a line of intersection of the surfaces, fig. 8.60. The dependence of the intersection line on the lateral coordinate is

$$a(y, \delta) = \sqrt{2R(\delta - h(y))} \quad (8.5)$$

The roots y_i of the equation

$$\delta = h(y) \tag{8.6}$$

determine the length of the patch along the lateral axes. The number of separate zones of the contact is equal to a half of the number of the roots.

So the contact area is a function of the surfaces of the contacting bodies and the interpenetration δ . Since the surfaces are always given functions we have the only unknown value δ .

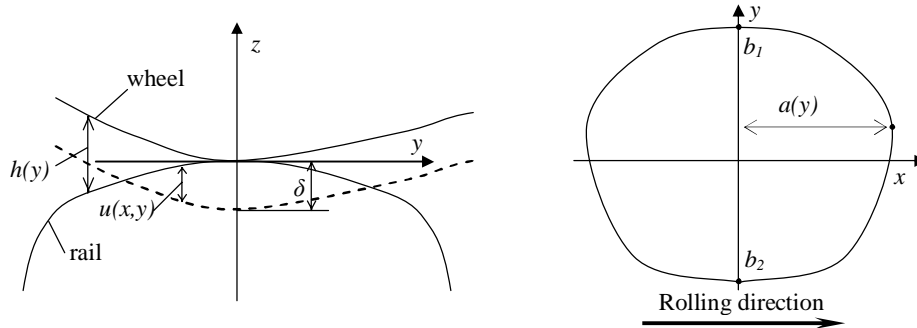


Fig.8.60. Wheel\rail contact

The materials of the bodies are assumed to be homogeneous, isotropic and elastic. Since the size of the contact patch is small in comparison to the characteristic sizes of wheel and rail, the contact stress does not depend on the shape of the contacting bodies distant from the contact patch. In this case, the contacting bodies can be considered as elastic half spaces. By using the half space method the value of δ can be estimated. The deflection at point (0,0) can be found with the help of the Boussinesq's influence function as

$$\omega(0,0) = \frac{1-\nu^2}{\pi E} \iint_C \frac{p(x,y)}{\sqrt{x^2 + y^2}} dx dy \tag{8.7}$$

where $p(x,y)$ is the distribution of the normal pressure, C – contact area. According to the assumption about materials, the wheel and rail deflections at contacting points are equal, thus $\delta = 2\omega(0,0)$. But in reality, the bodies at contact cannot interpenetrate and deflections occur, so the interpenetration region encloses the contact patch if the influence function is unidirectional. Granting this fact the bodies are interpenetrated in the depth $\delta_0 < \delta$, Figure 3.

To find the normal pressure distribution $p(x,y)$ in the contact area the elastic Winkler foundation model [4] i.e. the assumption about proportionality between stress and interpenetration $u(x,y)$ was used. So the distribution of the normal pressure is set as

$$p(x,y) = k_p u(x,y) \tag{8.8}$$

where k_p is a proportionality factor.

In UM rail-track model, the rail is considered as a massless body on a visco-elastic foundation and the normal force N at the contact point is available from the solution of rail equilibrium equations. The interpenetration of the bodies is only used to compute the normal pressure distribution and the approximate contact patch, and its small value can be neglected in the dynamic equations. Thus the normal force depends on the quite small vertical and lateral stiffness of the rail-track system and this model is not stiff.

Using Equations (8.4) – (8.8) the interpenetration in the case of single zone in the contact patch is

$$\begin{aligned} \delta = 2\omega(0,0) &= 2 \frac{1-\nu^2}{\pi E} k_p \int_{b_1-a}^{b_2} \int_{-a}^a \frac{\delta - \frac{x^2}{2R} - h(y)}{\sqrt{x^2 + y^2}} dx dy = \\ &= 2 \frac{1-\nu^2}{\pi E} k_p \int_{b_1}^{b_2} \left[\left(\delta - h(y) + \frac{y^2}{4R} \right) \cdot \ln \left(\frac{a + \sqrt{a^2 + y^2}}{|y|} \right) - \frac{a}{4R} \sqrt{a^2 + y^2} \right] dy. \end{aligned} \quad (8.9)$$

The normal contact force is calculated as

$$N = \iint_C p(x, y) dx dy = \iint_C k_p u(x, y) dx dy = k_p 2 \int_{b_1}^{b_2} \left(\delta a - h(y)a - \frac{a^3}{6R} \right) dy \quad (8.10)$$

Using Equations (8.9), (8.10) the following nonlinear equation are obtained:

$$N = \frac{\delta \pi E}{2(1-\nu^2)} \cdot \frac{\int_{b_1}^{b_2} \left(\delta a - h(y)a - \frac{a^3}{6R} \right) dy}{\int_{b_1}^{b_2} \left[\left(\delta - h(y) + \frac{y^2}{4R} \right) \cdot \ln \left(\frac{a + \sqrt{a^2 + y^2}}{|y|} \right) - \frac{a}{4R} \sqrt{a^2 + y^2} \right] dy}. \quad (8.11)$$

The solution of Equation (8.11) is the interpenetration δ . Taking into account that $\delta_0 < \delta$, the approximate contact patch is found and then using equation (8.10) the proportionality factor k_p is calculated. Substituting the value of k_p in equation (8.8) the distribution of the normal pressure over the contact area is obtained.

Tangential contact problem. The FASTSIM algorithm, which was adapted for non-elliptical contact area, was used to solve the tangential contact problem. To determine the value of flexibility an equivalent ellipse such that the area of the non-elliptic contact patch is equal to the area of the ellipse was calculated. The semi-axis of the ellipse in the rolling direction is set equal to the maximal half-length of the non-elliptical patch.

8.4.3.4. 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 - on the inward rail side f_s . These coefficients are numeric constants or functions of the longitudinal coordinates along the track $f_r(x)$, $f_s(x)$.

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.

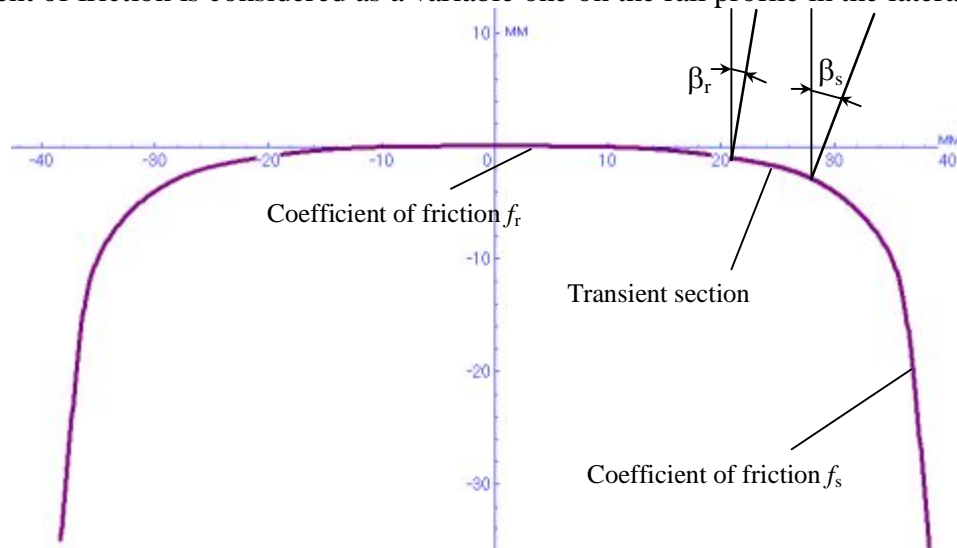


Fig.8.61. Changing the coefficient of friction along the rail profile

As a result, the profile is divided into three parts, Fig.8.61. 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, there the coefficient changes continuously from f_r to f_s . The transition is linear in the angle β . To divide the profile on these sections, the angles β_r, β_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.

8.5. Simulation of railway vehicles

8.5.1. Tools for preparing simulation process

8.5.1.1. Creation of wheel and rail profiles

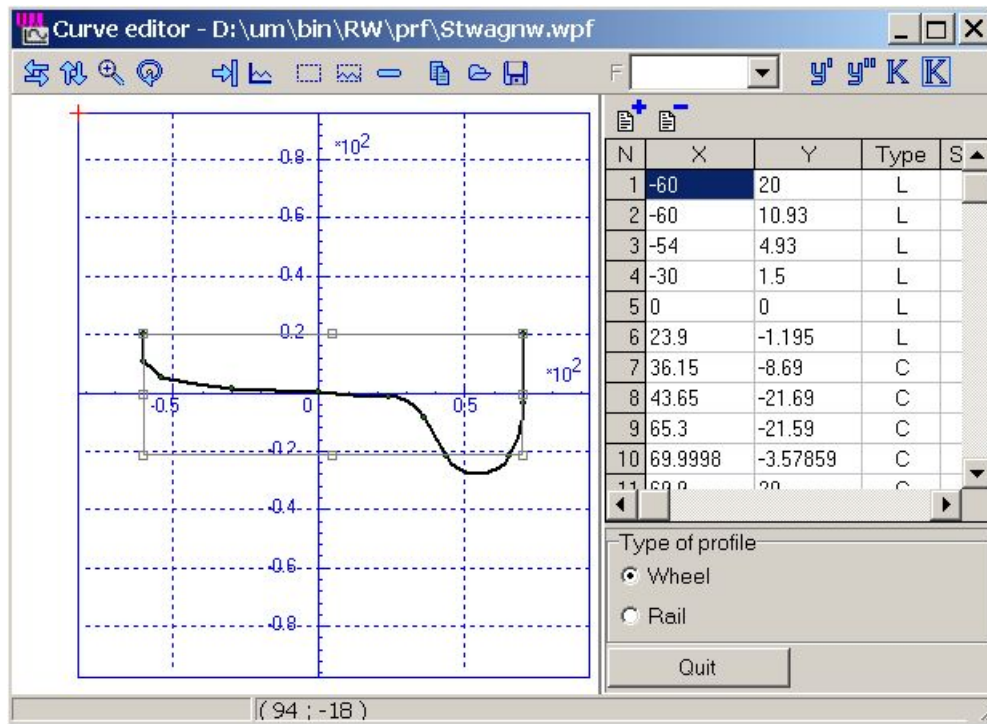



Fig.8.62. Editor of curves for creation of profiles

Wheel and rail profiles are located in the `\bin\rw\prf` directory in form of separate files with extension `*.wpf` (wheels) and `*.rpf` (rails). The profiles are described in special systems of coordinates (Sect.8.2.3.2, 8.3.1.1) with the help of a special tool, which is available in the **UM Simulation** program by clicking the **Tools | Create wheel/rail profile**, the `Ctrl+P` hot key or the  button on the tool panel. Creation of new profiles and modification of old ones are made with the help of the *Curve editor* (Fig.8.62). Detailed description of the editor can be found in Chapt.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.

8.5.1.1.1. Input as a set of points with successive spline approximation

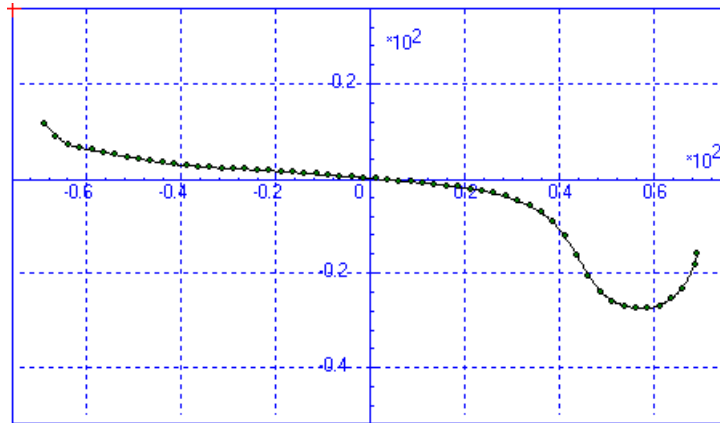


Fig.8.63. Input as a set of points

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.

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

- 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 curve by dragging the mouse and set the *Beta-spline* approximation (Fig.8.64).

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

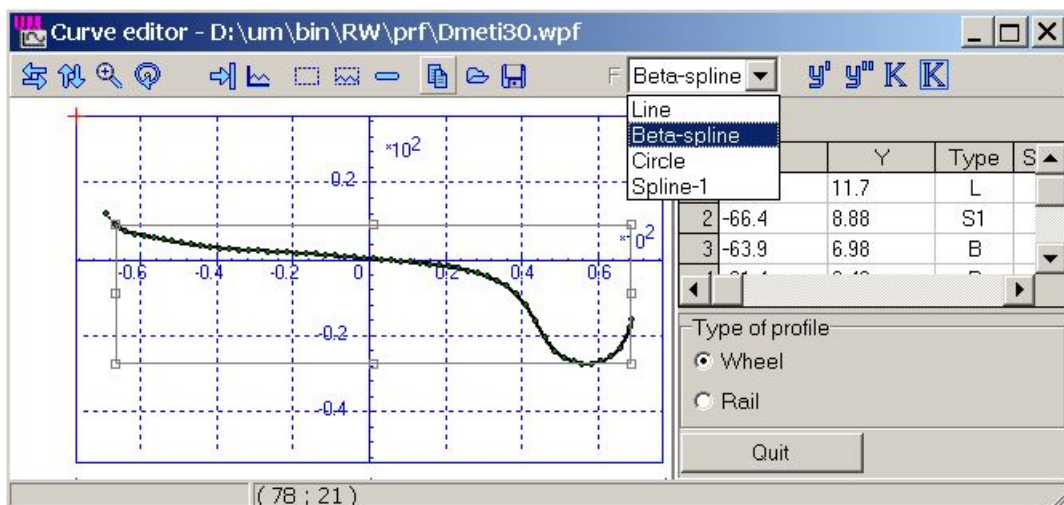


Fig.8.64. Selection of curve and setting the approximation

Note 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 (button **Y**), and the curvature plot might have large overfalls (button **K**). Such cases may lead to deterioration of the contact point computation process and to violation of continuity the contact point coordinate.

Note 2. The **K** button is used for plotting the smoothed profile curvature, which is used by computation of creep forces with the help of FASTSIM procedure.

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

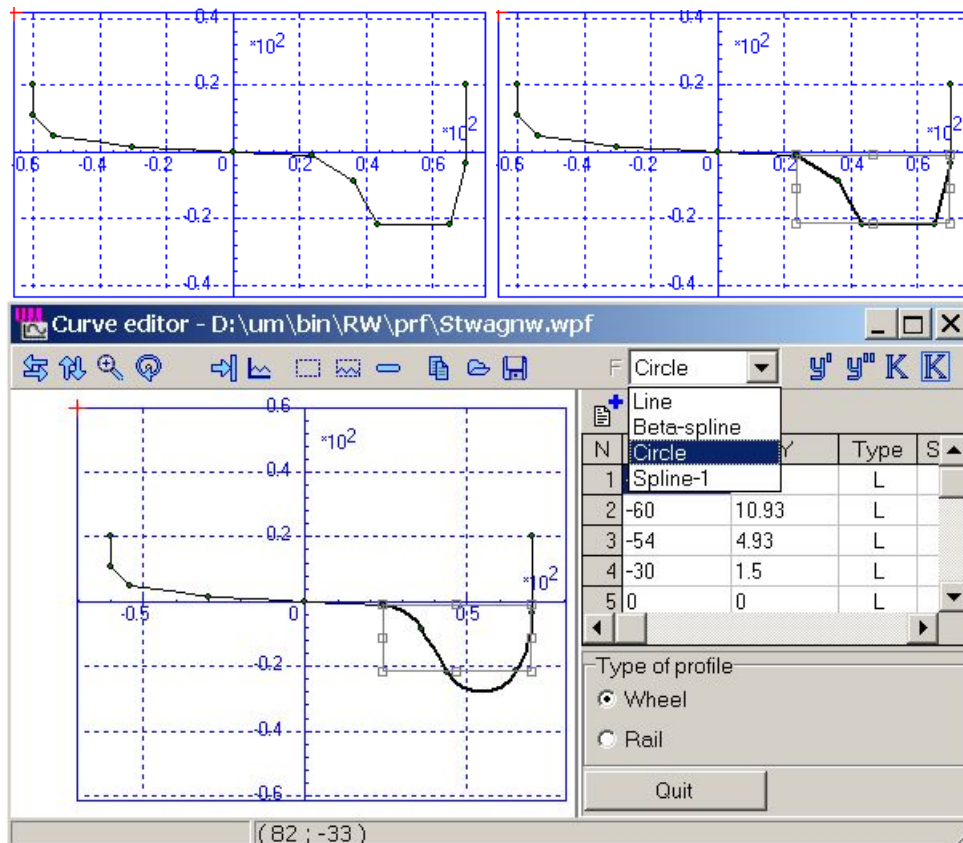


Fig.8.65. Creation of profile as a set of line sections and circle arcs

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 (Fig.8.65).
3. Save the profile with the help of the **S** button on the toolbar of the editor.

8.5.1.2. Creation of track irregularities

There exist two ways how to create track irregularities in UM

- as files
- using programming in the *Control file* of the model.

Use of files with irregularities is the main method. At the same time, programming of the irregularities in some cases is more effective, because it allows parameterizing some irregularity parameters such as height, length etc. In its turn, the parameterization allows scanning dynamic properties of a vehicle in dependence on the irregularity parameters.

8.5.1.2.1. Creation of files with irregularities

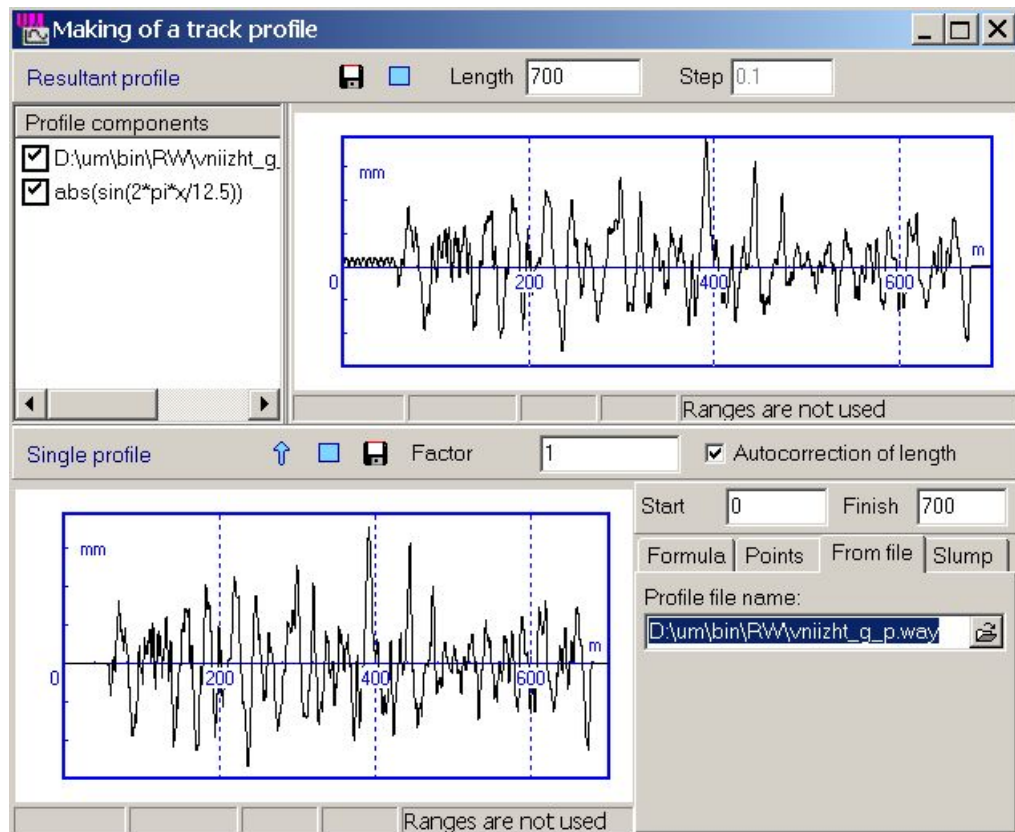


Fig.8.66. Tool for creation of irregularities



Files with irregularities *.way are stored in the \bin\rw directory. A file contains values of irregularities in meters with the constant step size 0.1 m in the *single* format (4-byte floating point numbers).

A new file of irregularities is created with a special tool, which is available in the **Simulation module** by clicking the **Tools | Create track irregularities** menu command.




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

Let us consider the structure of the tool and the meaning of its parts.

The resultant profile of irregularities is plotted in the top part of the tool window. It is created as a sum of separate irregularities of various types. The list of components is located in the left top part of the window (Fig.8.66). Deleting an element of the list removes the corresponding component from the resultant irregularity. Buttons and parameters at the top have the following functions.

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

Tabs in the right bottom part are used for creation separate irregularities of different types. The corresponding plot is located in the left bottom part of the window (Fig.8.66). Buttons and parameters at the top have the following functions.


- Button  adds the current separate irregularity to the resultant track profile.
- Button  saves the current separate irregularity to file.
- Buttons  clears the current separate irregularity.
- Parameter *Factor*: the current separate irregularity is added in the resultant one, it is multiplied by this factor. Consider an example. The user wants to convert some irregularity

in text format data into UM format. Let the data be given in meters. The tool with the help of the *Points* tab can accept the irregularity. However the factor 1000 should be set before adding the data to the resultant profiles to convert it in millimeters.

- The *Autocorrection of length* check box: if it is on, the length of the resultant profile is automatically increased to match the adding separate irregularity.
- The *Start* parameter shows where the separate irregularity begins when added to the resultant profile. Note that the plot of the separate irregularity in the bottom graphic window always starts with zero.
- The *Finish* parameter sets the length of the current irregularity. More exactly, the length is the difference between the finish and the start parameter values.

Consider types of separate irregularities.


- **Analytic expression** (the *Formula* tab)

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


- **Slump**

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

- **From file**

Here an already created file 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.

- **Points**

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

```
0 0
0.05 0.011
0.10 0.021
```

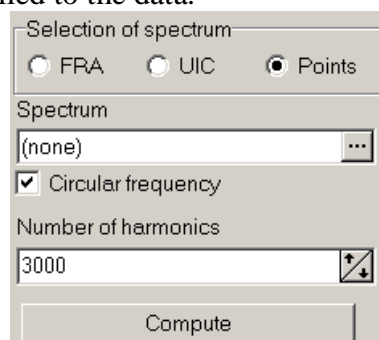
.....

To input this data with the help of the clipboard

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

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

Spline approximation can be applied to the data.




- **Spectrum (Points)**

Track irregularities can be generated according to specified SPD (spectral power density) function. The Rice-Pearson algorithm is used to generate the irregularity functions according to the formula

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

where Δs is the irregularity step size, m; M is the total number of harmonics in the sum; $S_x(\omega)$ is the SPD function, $m^2/(\text{osc./m})$; $\Delta\omega$ is the frequency increment, osc./m or rad/m ; $\varphi(m\Delta\omega)$ is the phase uniformly distributed on interval $[-\pi, \pi]$. The SPD can be function of frequency measured both in rad/m (circular frequency) and in oscillation/m . Use the Circular frequency check box to specify the necessary type.

Consider generation of irregularities be the SPD function shown in Fig. 8.67. Open the curve editor by the  button and specify four points on the SPD curve.

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.

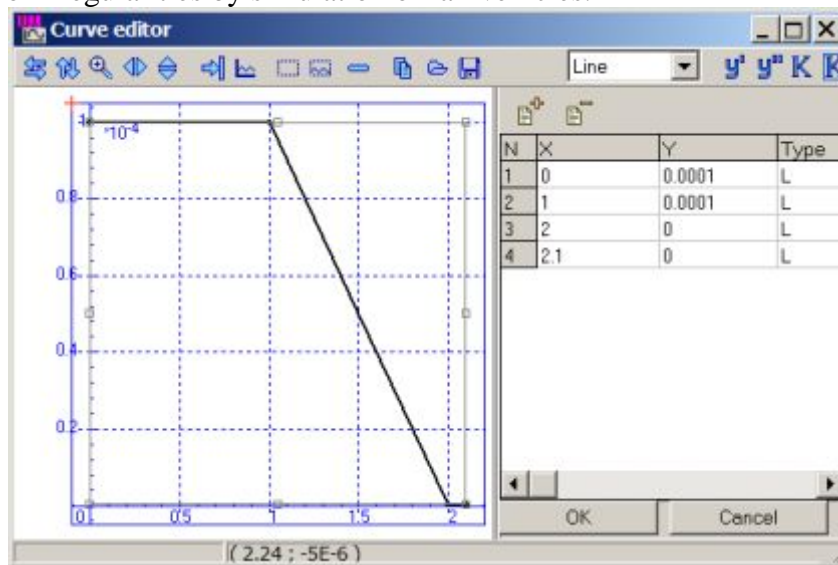




Fig.8.67. SPD function as a sequence of points

Set parameters like in Fig. 8.68, and add the function to the resultant profile by the  button and save in to a file by clicking the  button in the top of the window.

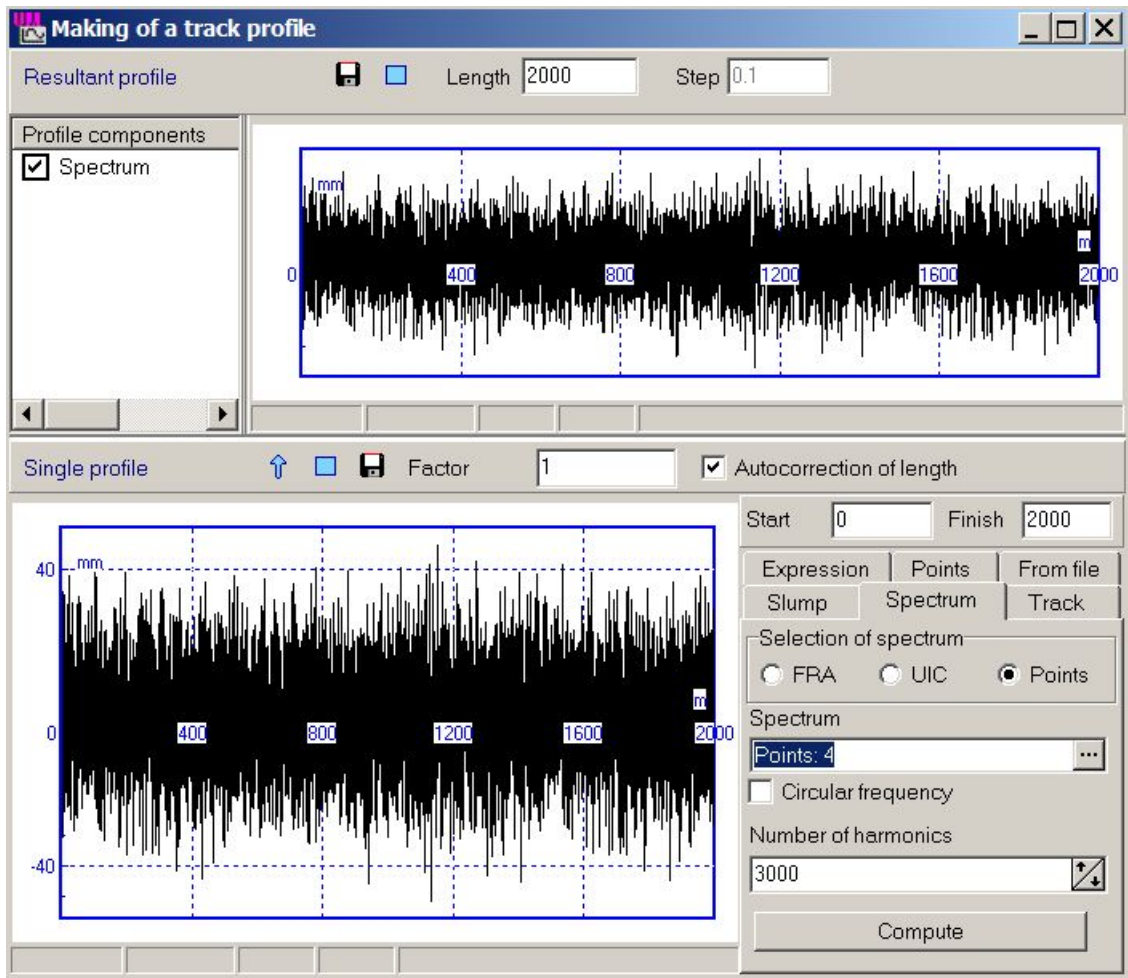



Fig.8.68. Track irregularity generation window

Reverse transformation of the irregularity to SPD function with the Statistics tool (**Tools | Statistics** menu command or the  button) is shown in Fig. 8.69, ($m/N = 0,002$).

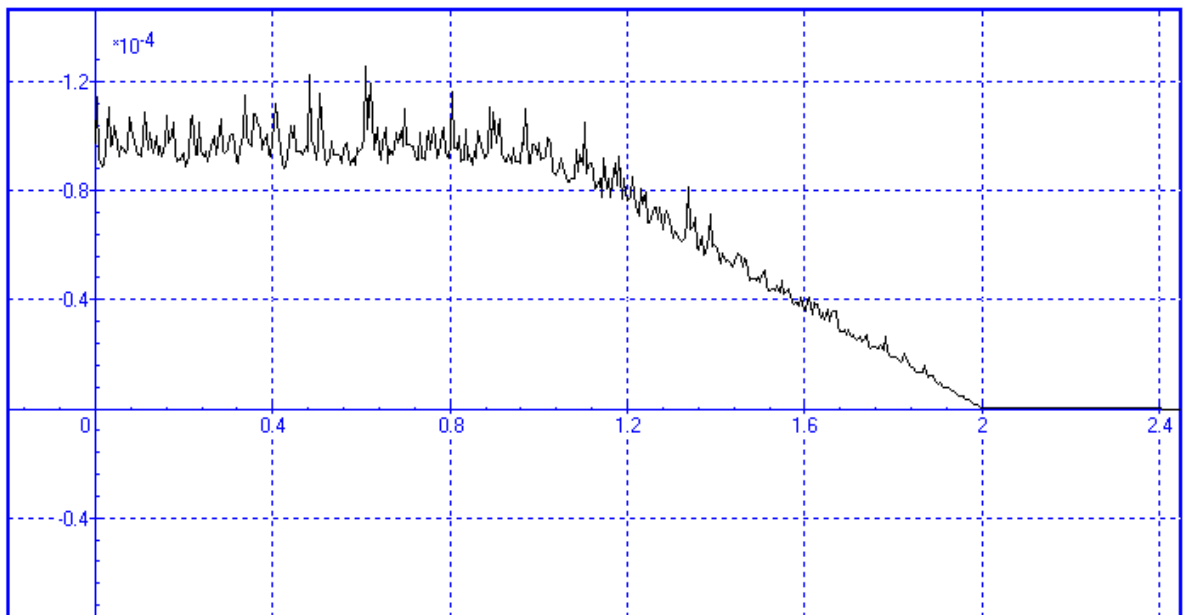


Fig.8.69. SPD function of generated irregularities

Remark. If necessary use the **Factor** value to convert irregularities in mm

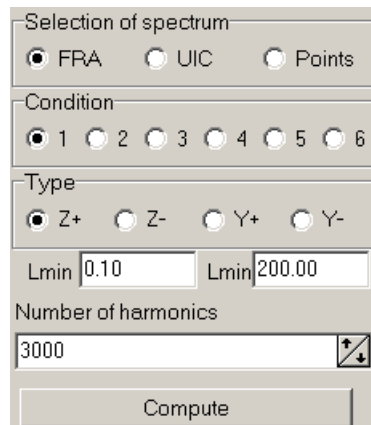




Fig.8.70. Standard SPD of FRA

- **Spectrum (FRA)**

With this option the user can generate irregularities by SPD corresponding to USA standards. Six option of the track **condition** allow generating irregularities of different quality: from very bad track (1) to very good track (6). The **Lmin** and **Lmax** parameters specify the minimal and the maximal length of irregularity in the realization (m).

It is possible to generate different realizations for left end right rails both for vertical and horizontal irregularities. To create two files for vertical irregularities of the left and right rails

- select the **Z+** option in the **Type** group, which denotes a SPD for a half sum of the vertical irregularities of the left and right rails;
- generate realization by the **Compute** button and add it to the resultant irregularity by the  button;
- select the **Z-** option in the **Type** group, which denotes a SPD for a half difference of the vertical irregularities of the left and right rails;
- generate realization by the **Compute** button and add it to the resultant irregularity by the  button keeping **Factor** = 1, Fig. 8.71.; save the result to file for the left rail;

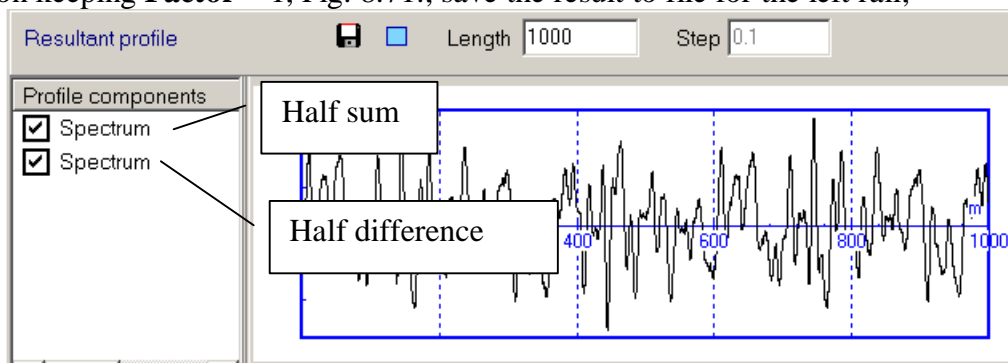
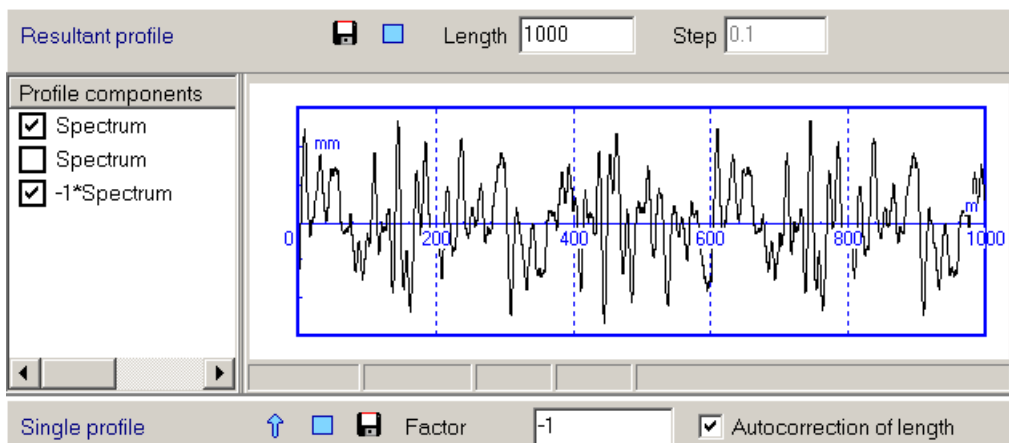



Fig.8.71. Resultant profile containing two realizations

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



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

Horizontal irregularities are generated in the same manner.

- **Spectrum (UIC)**

With this option the user can generate irregularities by SPD corresponding to UIC standards. Track conditions are of good or bad quality. Generation process for vertical irregularities is similar to that for the FRA standards. Note that horizontal irregularities in this case are equal for the left and right rails.

8.5.1.2.2. Programming irregularities in the Control file

Control file is the main tool for the user programming in UM environment (Chapt.5). The control file for railway vehicles contains the following function:

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

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

Input:

position – 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 – vertical irregularities of the left and right rails in meters;

LeftY, RightY – horizontal irregularities of the left and right rails in meters;

derLeftZ, derRightZ – derivatives of vertical irregularities of the left and right rails w.r.t. longitudinal coordinate;

derLeftY, derRightY – 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.

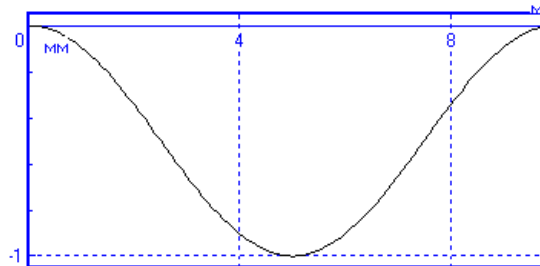


Fig.8.72. Slump

Consider a separate irregularity of the *Slump* type, Fig.8.72. 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 (Chapt.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;


```

8.5.1.3. Creation of macro-geometry files

The macro-geometry files *.mcg are used in the following cases:

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

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

Use the **Tool | Create macrogeometry** menu command or the  button to start the window where the track is described.

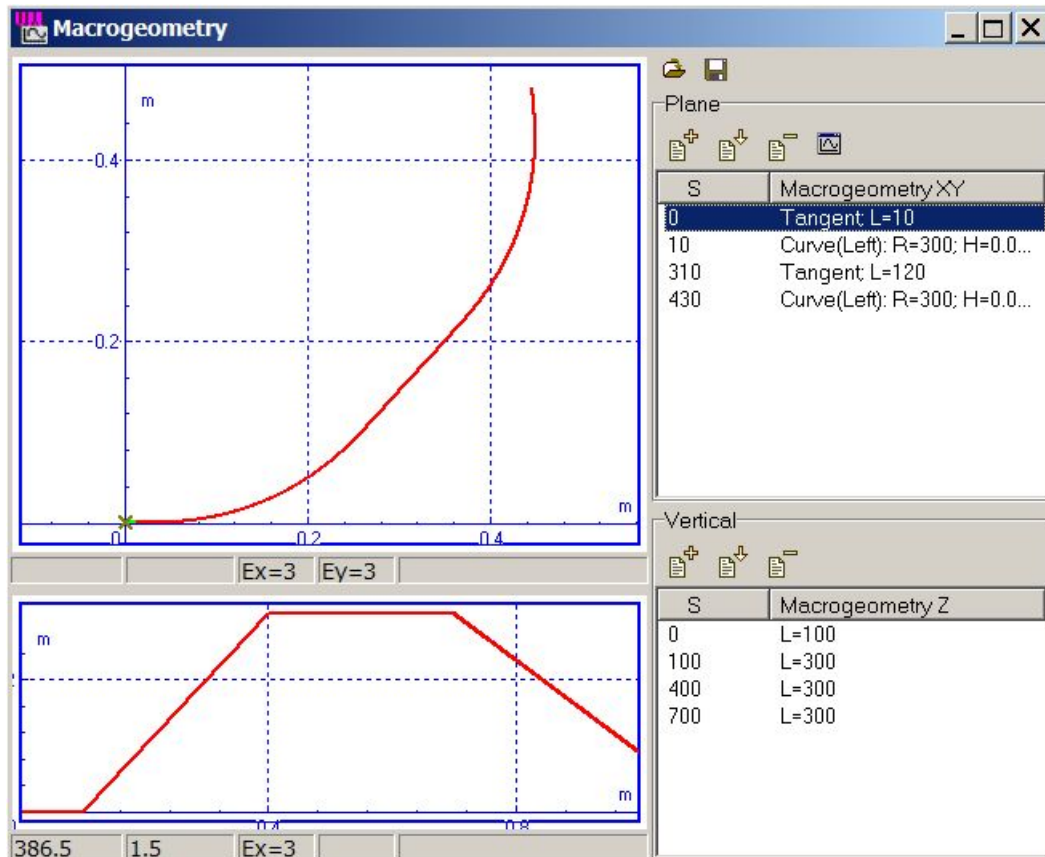


Fig.8.73. Marcogeometry window

The upper part of the window 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.

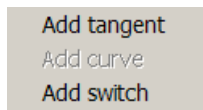


Fig.8.74. Adding section menu

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



Fig.8.75. Parameters of a tangent section

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

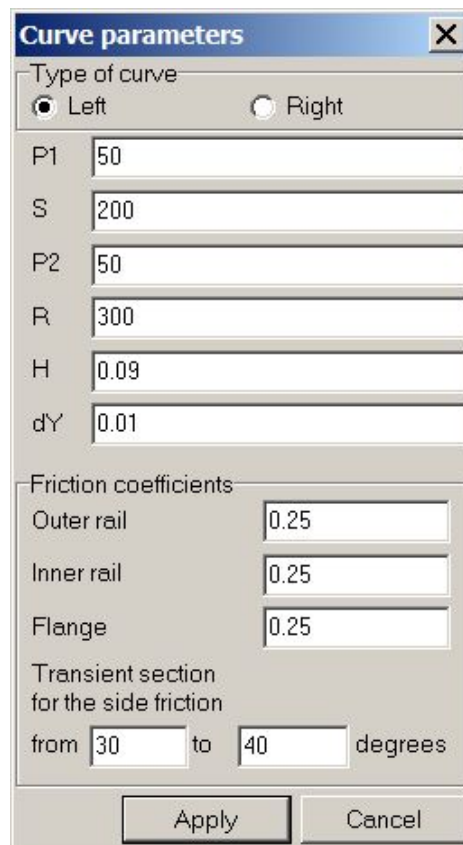


Fig.8.76. Window with curve parameters

- Curve parameter window includes (Sect. Macro 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 β_r, β_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. 8.4.3.4. Coefficient of friction in wheel/rail contact.

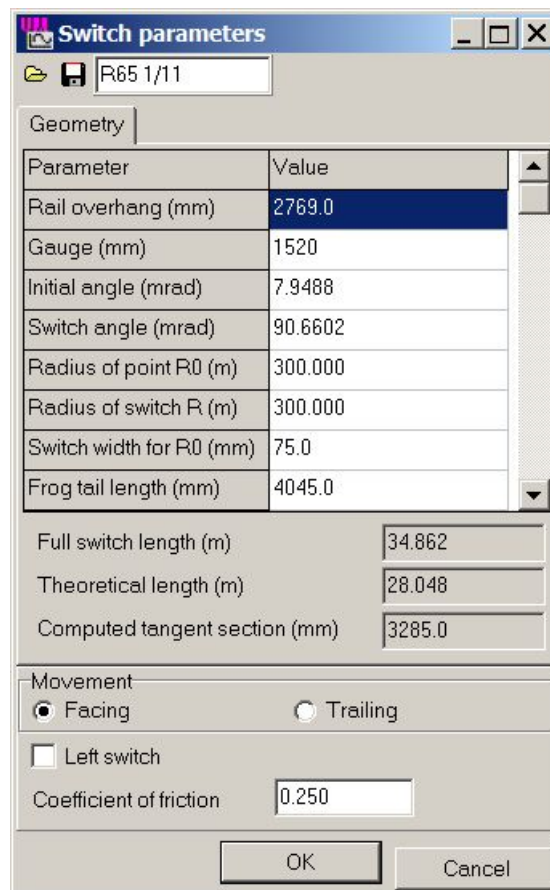


Fig.8.77. Switch parameters

- Switch parameter window includes values of (Sect. 8.3.1.3. Switch geometry)

q – stock rail overhang;

gauge;

β_n - initial angle;

α - switch angle;

R_0 - radius of point;

R - radius of switch;

b_f - switch deviation for R_0 ;

m - frog tail length;

d - track spacing.

R_1 - radius behind frog.

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

L_p - full switch length;

L_t - theoretical length;

κ - tangent section before frog.

Additional parameters

- direction of motion (facing, trailing)

- type of switch (left or right);

- coefficient of friction.

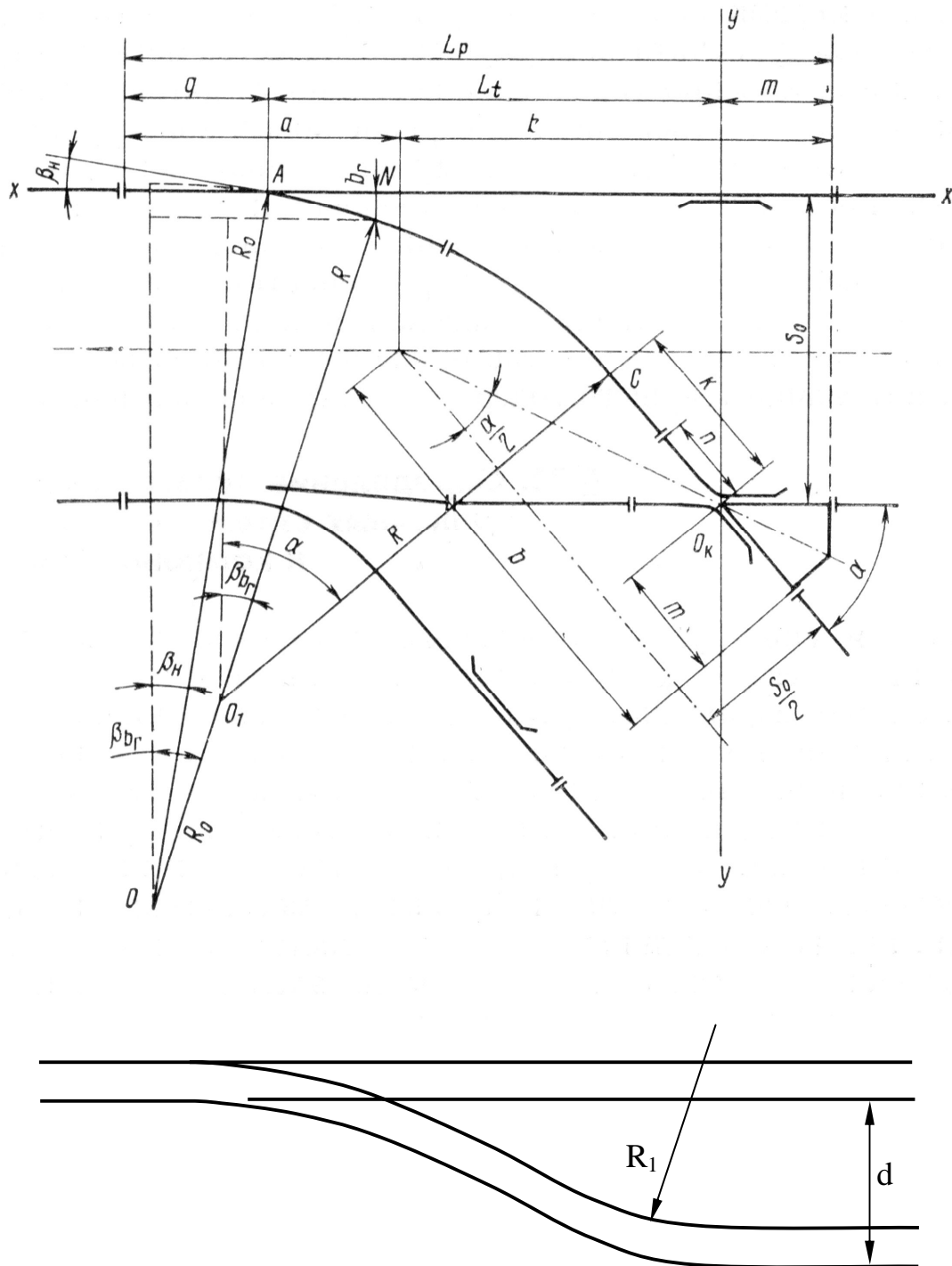



Fig.8.78. Switch geometry parameters

Use the buttons in the bottom part of the window to create the **vertical profile** of the track. Double click of the line of the list to edit the section length and inclination.

8.5.1.4. Creation of files with switch parameters

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

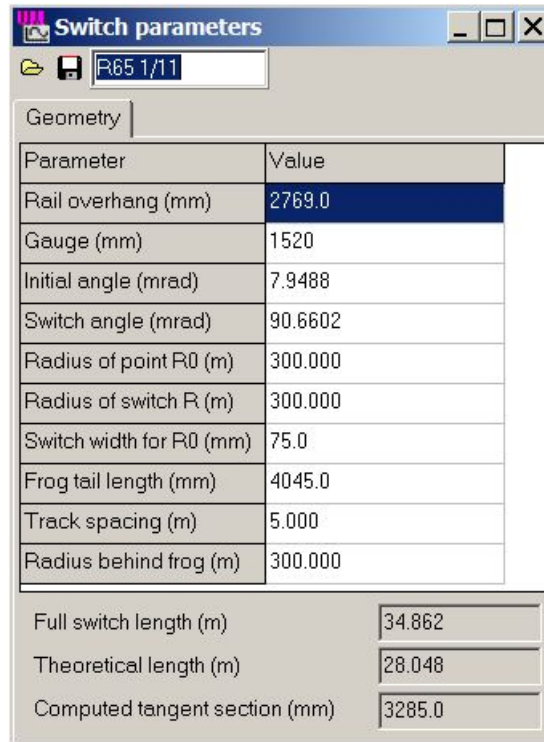





Fig.8.79. Switch window parameters

Use the   buttons to read data from *.swt file or to save data. The upper text box contains the name of the switch (optional parameter).

See the previous section as well as Sect. 8.3.1.3. *Switch geometry* for more details.

8.5.2. Setting parameters of railway vehicle dynamics simulation

Here we suppose that the model of a vehicle is ready for simulation and it is already loaded in the Simulation module. Parameters of modeling a railway vehicle are set with the help of the Object Simulation Inspector, which is called by means of

- the **Analysis | Simulation** menu command;
- the *Ctrl+F9* hot key;
- the  button of the tool bar.

A considerable part of the parameters can be set in a standard manner, Chapt.4, Sect. *Preparing for integration*. Here we consider some features of the parameters setting for a railway vehicle.

8.5.2.1. Solver and its parameters. Use of Jacobians

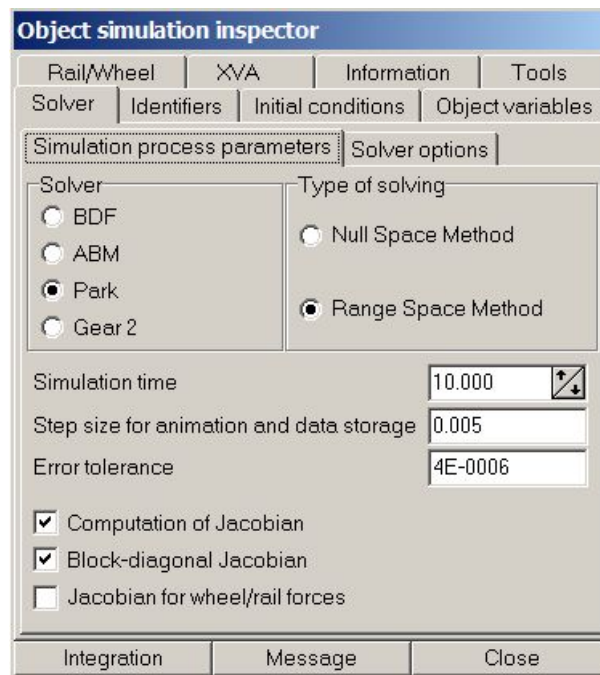


Fig.8.80. Solver parameters

Here we consider parameters in the *Solver* tab of the inspector, Fig . It is recommended to use the Park solver with an error tolerance value about $1 \cdot 10^{-6} \div 1 \cdot 10^{-7}$. The recommended step size for animations and data storage is $0.02 \div 0.005$ sec. If the integration process does not converge, i.e. a ‘beard’ is observed for some variable plots, mainly for accelerations, **Computation of Jacobian** should be used or the accuracy should be increased right up to disappearance of the ‘beard’. Note that increasing the accuracy corresponds to decreasing the **Error tolerance** parameter.

Use of the Jacobians allows a considerable decreasing the CPU expenses in the following cases.

- Small speed of the vehicle (less than 8..10 m/s); in this case both the **Computation of Jacobian** and **Jacobian for rail/wheel forces** boxes must be checked.
- The model contains stiff force element, which description contains large stiffness and/or damping coefficients (contacts, bump stops etc.). This may lead to large frequencies or damping level.

If computation of Jacobians is on, simulation process can be often made faster by

- Use of **block-diagonal Jacobians**;

- Switching of computation of Jacobians of non-stiff forces such as suspension springs or dampers on the **Tools | Forces** tab of the *Object Simulation Inspector*, Fig. 8.81.

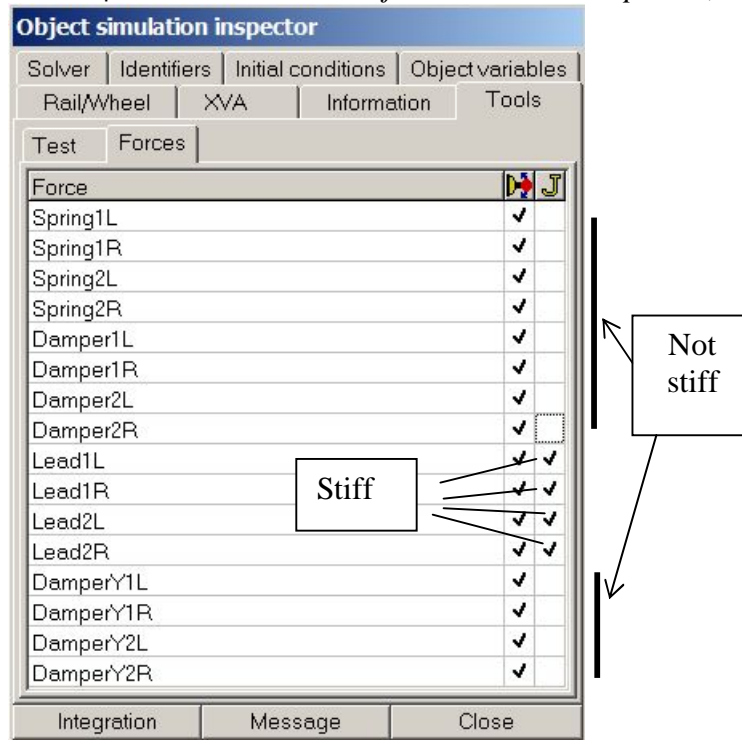


Fig.8.81. Switching on/off evaluation of Jacobian matrices for force elements

Note. Use of the Jacobian matrices and other recommendations are useful if they allow increasing the solver stability or decreasing CPU time. The user should optimize these parameters by studying the model behavior.

8.5.2.2. Modes of longitudinal motion of vehicle

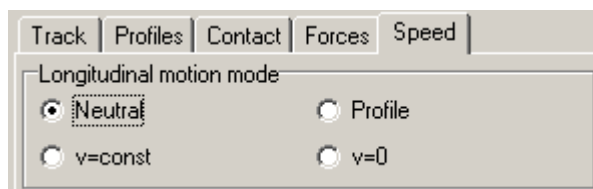


Fig.8.82. Longitudinal motion modes

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

- **Neutral**

In this mode the initial speed value is set by the v0 identifier. The speed decreases due to resistance forces.

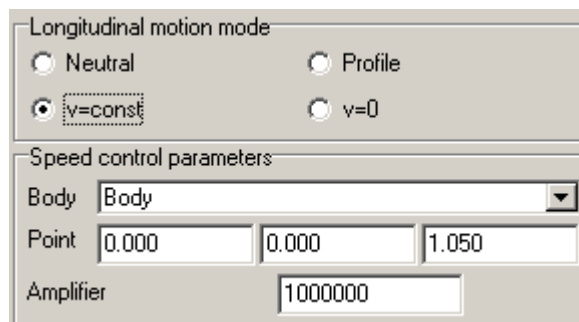


Fig.8.83. Constant speed mode

- **v=const**

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, Fig. 8.83. Usually the point lies on the coupling level; the longitudinal coordinate of the point is not important if the body is rigid.

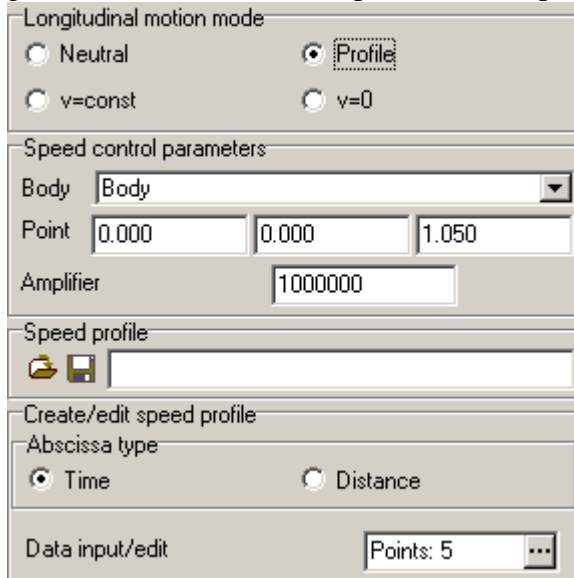


Fig.8.84. Speed profile mode

- **Profile**

The vehicle speed is controlled according to a dependence on a time or distance, the **Abscissa type** group. The force 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, Fig. 8.85. The and 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).

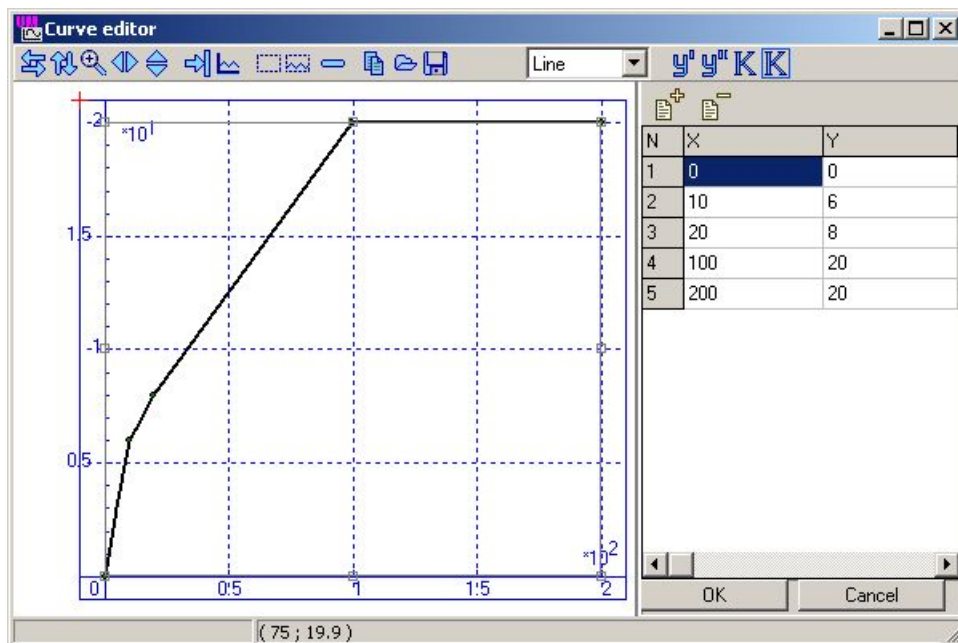


Fig.8.85. Example: vehicle speed graph is dependence on time

- $v=0$

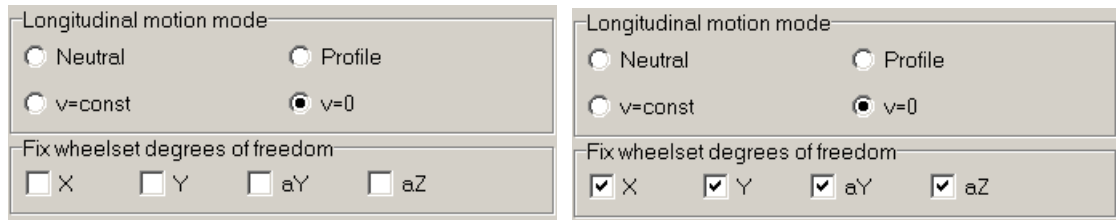





Fig.8.86. Parameters of zero speed mode. Degrees of freedom of wheelsets are not blocked (left) and blocked (right)

Zero velocity mode. This mode is used when

- the vehicle equilibrium is not computed automatically by the  button or
- the equilibrium is computed not quite correctly (e.g. when dry friction is presented in the model),
- oscillation under harmonic loads are analyzed.

In the last case the 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 (aY, aZ). To block a degree of freedom, check the corresponding box.


Coordinates of the nearly equilibrium position computed in this mode can be used as initial values in simulation with non-zero speed. The following steps are necessary for this.

- Run the simulation until the transient processes disappear;
- Save finish values of coordinates in file by the **Save** button in the pause mode;
- Open the **Initial conditions** tab of the inspector;
- Read coordinate values by the  button,
- Set zero velocity values by the  button.

8.5.2.3. 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.

The  button on the inspector of on the program tool bar let the user to view the assigned profiles.

8.5.2.3.1. Assignment of rail profiles

UM database of rail profiles includes the following files.

- r65new.rpf – new Russian rail R65 with running surface radius 500mm;
- r65_300n.rpf – new Russian rail R65 with running surface radius 300mm;
- r65old13.rpf – R65 with 13mm lateral wear;
- r65old15.rpf – R65 with 15mm lateral wear;
- r50.rpf – new Russian rail R50;
- Several new and worn rail profiles UIC60.

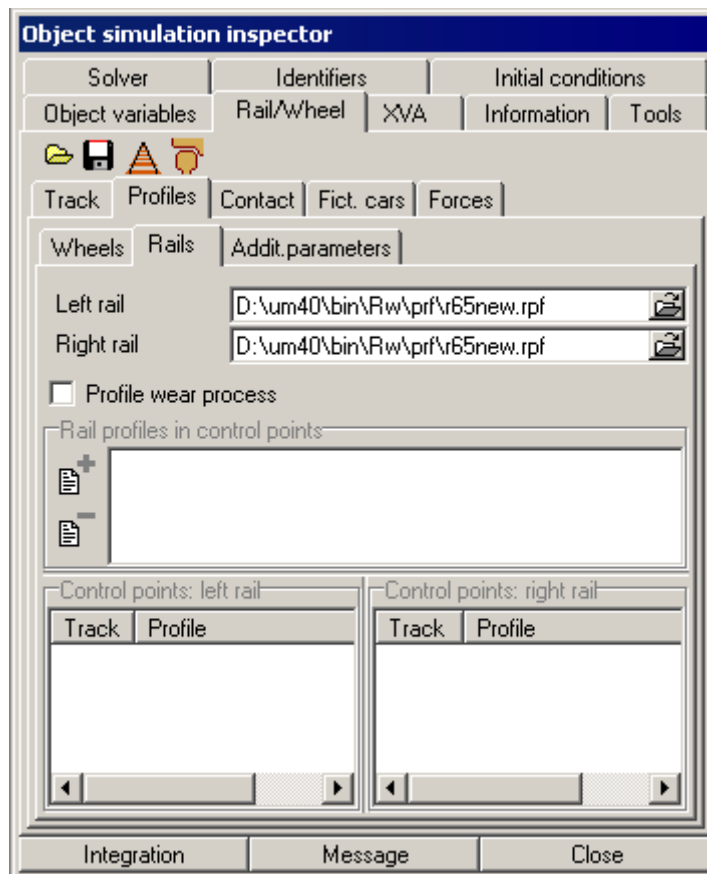



Fig.8.87. Assignment of rail profiles and creation of wheel profile list

Use the  buttons to assign profiles for the left and right rails (Fig.8.87).

Note. Rail profiles assigned here are not used if the rail profile evolution mode is switched on (Sect. *Rail profile evolution along the track*).

8.5.2.3.2. Assignment of wheel profiles

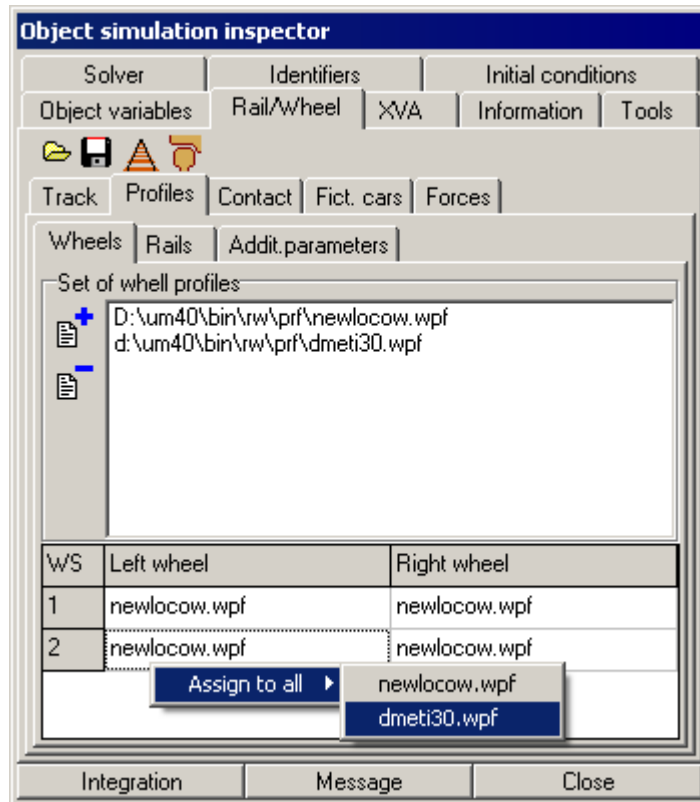


Fig.8.88. Assignment profiles to separate wheels

UM database of wheel profiles includes the following files.

- newlocow.wpf – new Russian locomotive wheel;
- newwagonw.wpf – new Russian freight car wheel;
- dmeti30.wpf – new DMETI profile;
- new and worn wheel profiles S1002.wpf.

Assignment of wheel profiles has two stages. At the first stage a list of wheel profile is created (Fig.8.87). The list should include at least one profile. The button on the pop up menu is used to add a profile to the list.

To assign profiles from the list to the wheels

- open the Rail/wheel / Profiles / wheels tab (Fig.8.88);
- call the pop up 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.

8.5.2.3.3. Rail profile evolution along the track

To describe a changing the profile of the left or/and right rail along the track (Sect.8.3.1.1), the following points should be used.

- Create the list of profiles with the Rail/Wheel / Profiles / Profile files / Rail tab. The list is created by adding and deleting of existing files. Use the buttons or the pop up menu to add and delete profiles.
- Use Rail/Wheel / Profiles / Profile files / Evolution of rail profile tab to set a sequence of profiles R_i the track as well as their positions along the track S_i (Sect. 8.3.1.1), $i=1,2,\dots$

Add profiles to the sequence with the help of the pop up menu. After that set the positions (Fig.8.89).

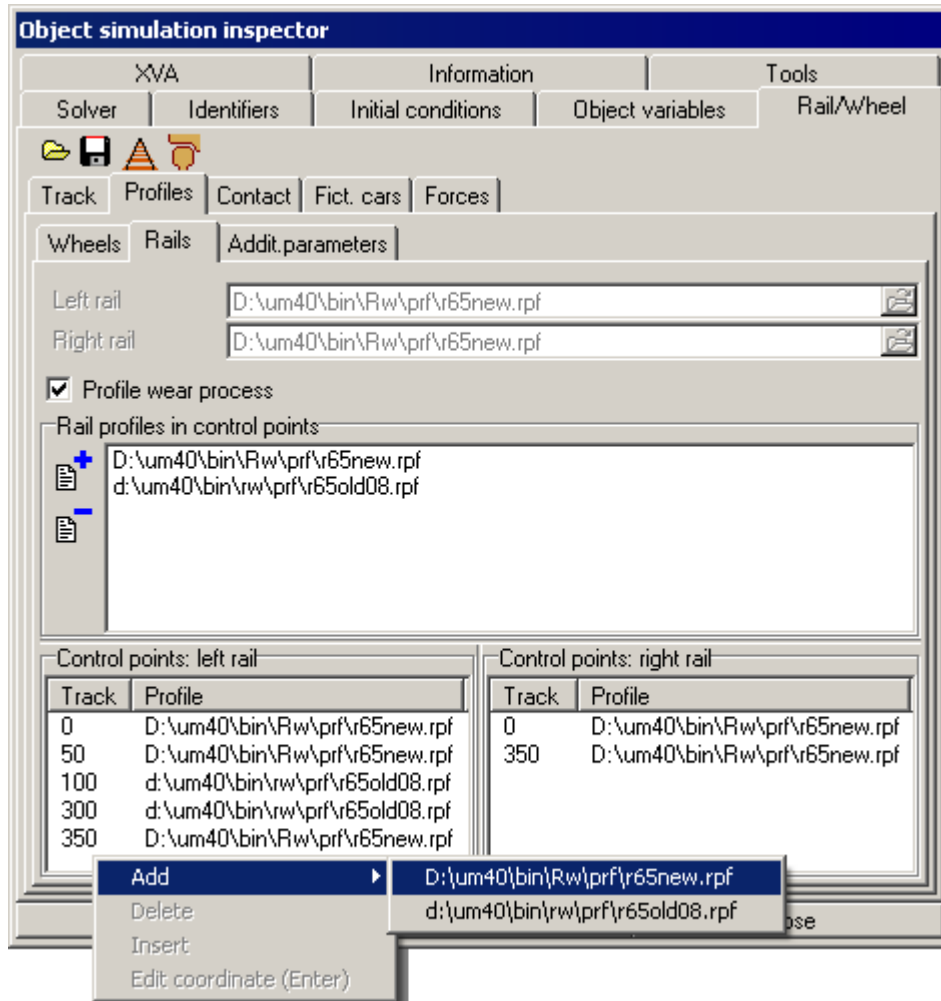


Fig.8.89. Creation of sequence of profiles along the track

- Switch on the mode of the profile evolutions by the *Variable profile* key (Fig.8.89).

Example. Fig. 8.89 show creation of sequence of profiles for the outer rail of a right curve with the following parameters (Sect. Macro geometry of curve)

$$L0=10m, P11=50m, S1=200m, P12=50m.$$

The new profile *r65new* (Fig. 8.90) is used at straight sections. At the first 50 m transition (section from 10m to 60m) the new profile is transformed to the worn one *r65old*. In the steady curve (from 60m to 260m) the profile does not change. At the second 50 m transition (from 260m to 310m) the profile is transformed from the worn to the new one.

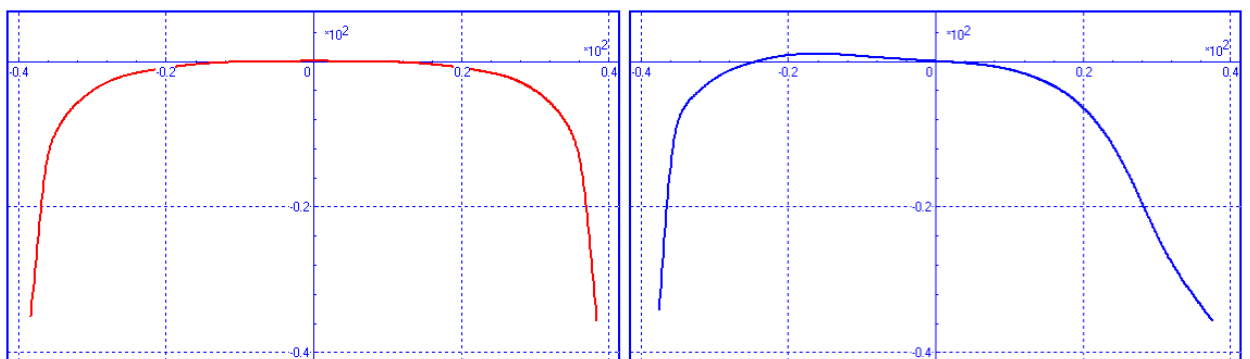


Fig.8.90. Profiles r65new (left) and r65old

8.5.2.4. Track parameters

8.5.2.4.1. Assignment of track irregularities and rail geometry parameters (gauge, rail inclination)

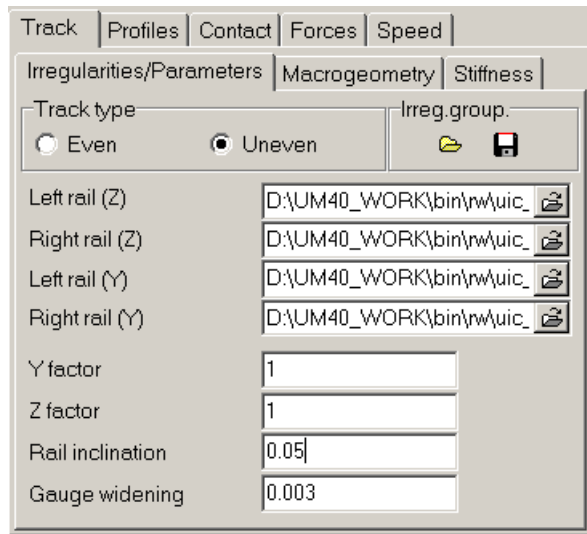


Fig.8.91. Track irregularities and parameters

Several files with track irregularities are delivered with UM.



1). Bad UIC track generated by SPD function, 1000 m length. The irregularity group file is uic_bad_1000.



uic_bad_1000_zleft.way
uic_bad_1000_zright.way
uic_bad_1000_y.way

2). Good UIC track generated by SPD function, 1000 m length. The irregularity group file is uic_good_1000.

uic_good_1000_zleft.way
uic_good_1000_zright.way
uic_good_1000_y.way

The NoIrregularities.way file is used if a rail has no irregularities or the user programs the irregularities in the control file.

The *Rail/Wheel | Track | Irregularities/Parameters* tab is used for assignment of irregularities from files. To set both the vertical (Z) and the horizontal (Y) irregularities to the left and right rails use the corresponding  button (Fig.8.91). Use the  button on the inspector and on the tool bar to visualize the chosen irregularities.

The *Irreg[ularity] group* contains buttons, which are used for read and write groups of four files. For example, the user created four files with irregularities assigns them to the track as above. After that he save 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.

The *Factor Y* and *Factor Z* parameters are 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 decreasing the horizontal irregularities come to 30%.

In addition the standard **rail inclination** α_{r0} in radians and the standard **gauge widening** (m) in the ideal straight track Δy should be set on this tab (Sect.8.3.1.1).

Use the *Even* parameter of the *Track type* group to simulation a vehicle in the ideal section of curve. If this case the irregularities are ignored.

Use the *Uneven* parameter of the *Track type* group to take the irregularities into account.

See also:

Track irregularities

Creation of files with irregularities

Programming irregularities in the Control file

References to files the assigned irregularities are stored in a vehicle configuration file *.rwc.

8.5.2.4.2. Input track macro geometry parameters (tangent, curve, S-curve, switch, profile)

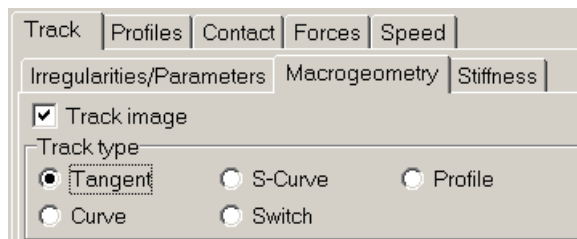


Fig. 8.92. Macro-geometry types

Use the *Rail/Wheel / Track / Macrogeometry* tab of the inspector to set the track macro geometry.

The *Track type* parameter specifies the current track.

- **Tangent**

Motion in a tangent section of unlimited length.

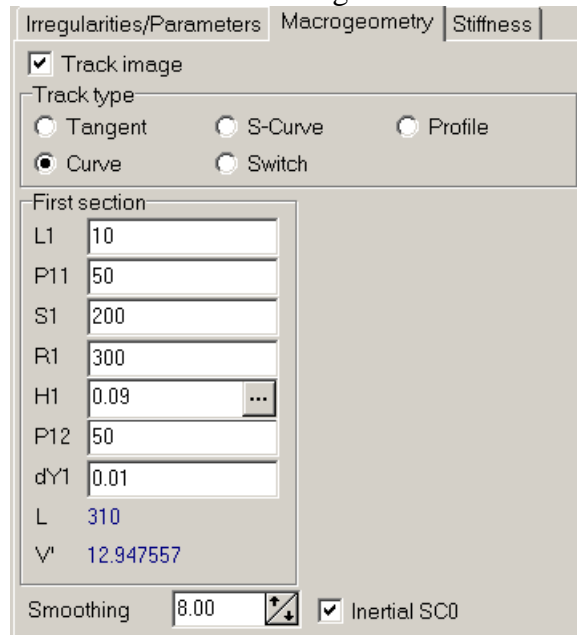


Fig.8.93. Right curve parameters

- **Right curve, Fig. 8.93**

Motion in a right curve 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 dy should be set in meters. See Sect. 8.3.1.2. *Macro geometry of curve* 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.

The *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.

The *Inertial SCO* check box allows the user to run the simulation either in the inertial (checked) or non-inertial (unchecked) system of coordinate, Sect. *Base coordinate system*.

Track type	
<input type="radio"/> Tangent	<input checked="" type="radio"/> S-Curve
<input type="radio"/> Curve	<input type="radio"/> Switch
<input type="radio"/> Profile	

First section		Second section	
L1	10	L2	50
P11	50	P21	50
S1	200	S2	200
R1	300	R2	300
H1	0.09	H2	0.09
P12	50	P22	50
dY1	0.01	dY2	0.01
L	310	L	350
V'	12.947557	V'	12.947557

Fig. 8.94. Parameters of S-curve

- **S-Curve**

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

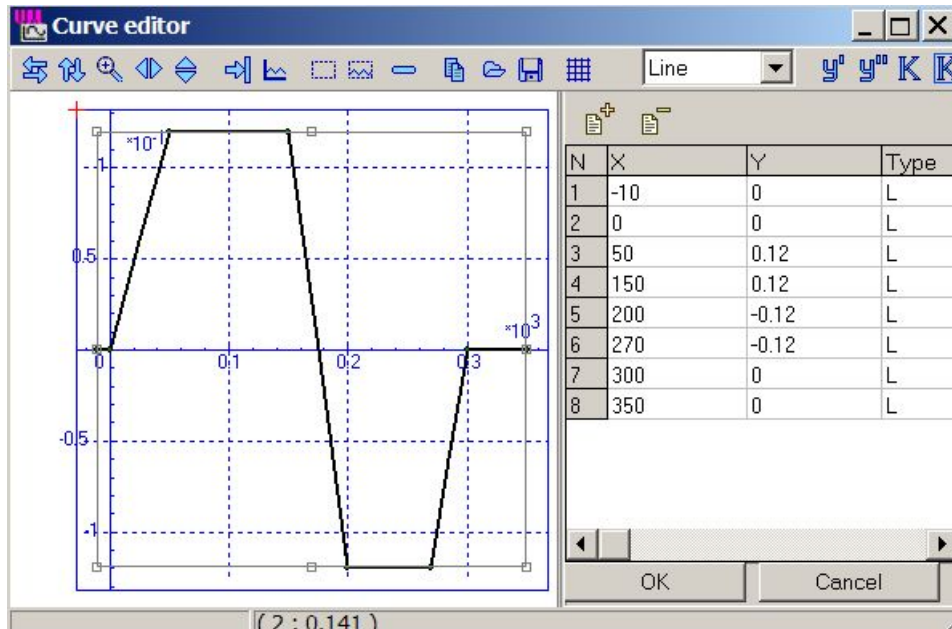

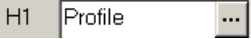


Fig. 8.95. Cant versus track distance

Features of setting cant values

In case of motion in a right curve and in S-curve (Fig. 8.93, Fig.8.94.) 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, Fig. 8.93, Fig.8.94. The curve editor appears, and the cant value (m) versus curve distance can be specified by a sequence of the points, Fig. 8.95. After that the box looks like . To set a constant cant back, the user should delete all points in the list in Fig. 8.95.

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 Fig. 8.95. 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.


How to model the motion in a left curve?

Fig. 8.96. Trick: left curve as S-curve

S-curve can be readily 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.

Fig. 8.97. Parameters of a switch

- **Switch**

Use the  button to open a *.swt file with the switch parameters (see Sect. 8.5.1.4. *Creation of files with switch parameters*). Other parameters allow the user to simulate both facing and trailing movements in left and right switches, Fig. 8.98. The **Tangent section** parameter sets the length of a tangent section before the switch.

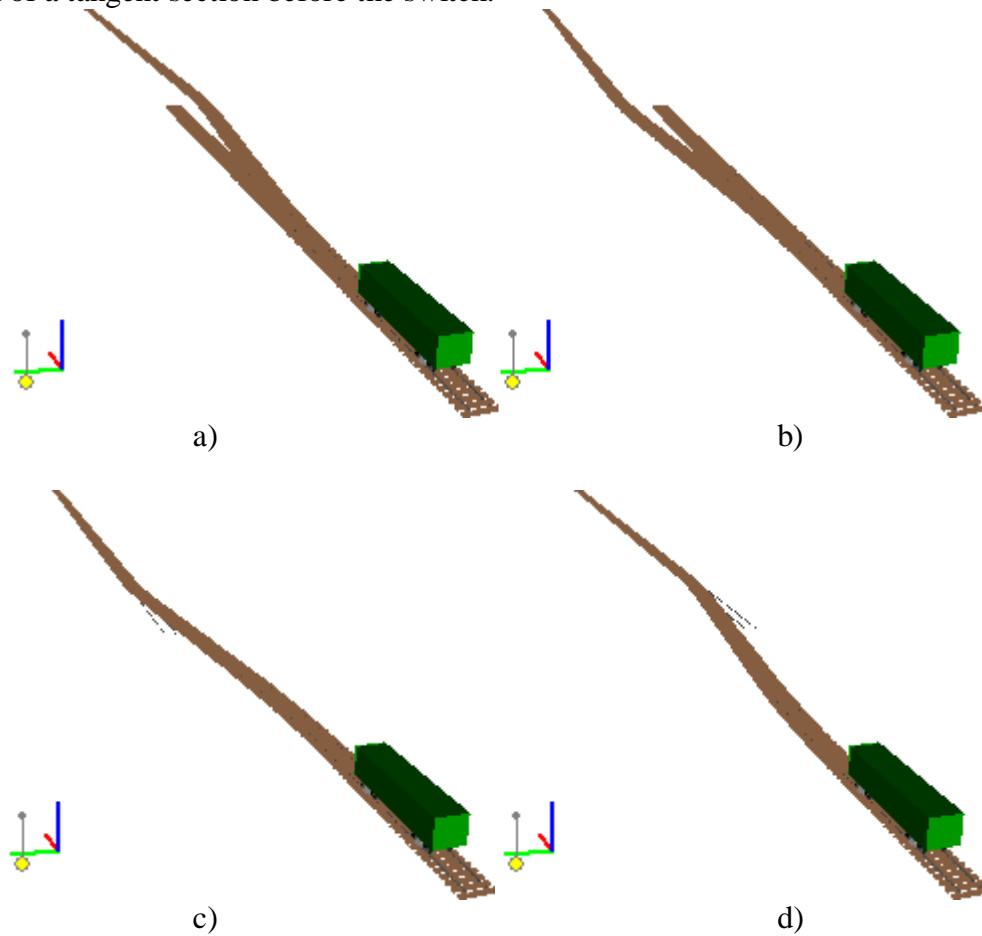
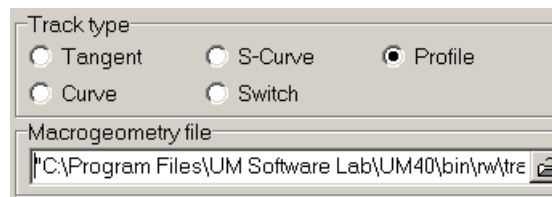



Fig. 8.98. Facing (a,b) and trailing (c,d) movements in left and right switches

- **Profile**



Use the  button to open a *.mcg file with the track macro-geometry parameters (see Sect. 8.5.1.3. *Creation of macro-geometry files*). In this case of the track the motion in inertial SC0 is always used.

8.5.2.4.3. Track stiffness and damping

Irregularities	Macrogeometry	Stiffness
		Vertical stiffness
		44 000 000
		Vertical damping
		400 000
		Lateral stiffness
		18 000 000
		Lateral damping
		100 000

Fig.8.99. Track stiffness and damping parameters

Stiffness and damping coefficients of the track for a wheel are here equal for the left and right rail and constant along the track. Stiffness is measured in N/m, damping coefficient – in Ns/m. The default values are given in Fig.8.99.

See also:

8.3.2. *Elastic, dissipative and inertia properties of track*

8.4.3. *Contact forces*

Method for computation of rail deflections and contact force

8.5.2.4.4. Programming elastic and damping track parameters

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 (Chapt.5). 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 – longitudinal coordinate for which the irregularities are computed.

Output:

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

cLeftZ, cRightZ – vertical stiffness of the left and right rails in N/m;

cLeftY, cRightY – lateral stiffness of the left and right rails in N/m;

dLeftZ, dRightZ – vertical damping coefficient of the left and right rails in Ns/m;

dLeftY, dRightY – lateral damping coefficient of the left and right rails in Ns/m.

Note 1. If the function returns 1, parameters entered in Fig.8.99 are ignored except the cases in the Note 2.

Note 2. Zero values of stiffness *cLeftZ, cRightZ, cLeftY, cRightY* are ignored and replaced by values from entered in Fig.8.99.

8.5.2.5. Parameters for computation of rail-wheel contact forces

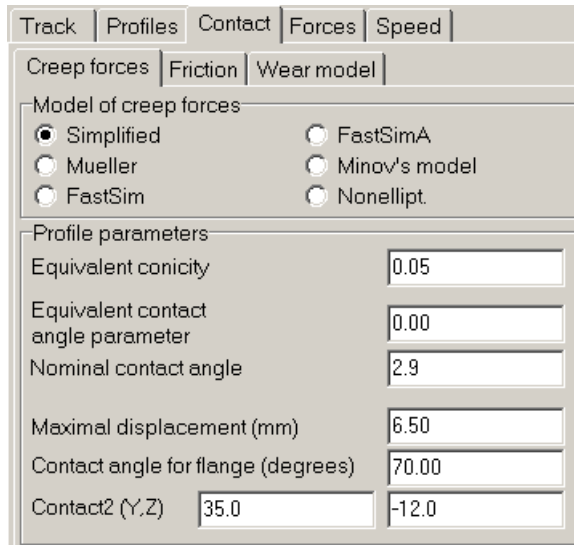


Fig.8.100. Choice of a creep forces calculation method and its parameters

A method for computation of creep forces (Sect. 8.4.3.2) can be chosen on the *Rail/wheel / Contact / Creep forces* tab.

• **Simplified**

Parameters of the simplified model of rail/wheel contact are shown in Fig. 8.100, see 8.4.1.3. Simplified contact geometry. Equivalent conicity and contact angle parameter:

- equivalent conicity λ
- contact angle parameter ε
- nominal contact angle β_0 (degrees) is the value of the contact angle for non-shifted wheelset;

- maximal lateral shift of a wheel relative to the rail y^* (start of two-point contact, mm)
- contact angle on flange by two-point contact β^* (degrees)

- **Contact 2 (Y, Z):** coordinates of flange contact by two-point contact in the rail coordinate system, y_r^*, z_r^* (mm).

Remark. Mueller's method is used for computation of creep forces.

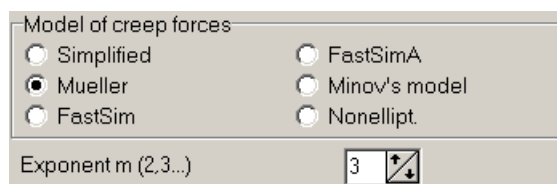


Fig. 8.101. Mueller's model

• **Mueller**

See Sect. 8.4.3.2.1. Mueller's method for the details. The only parameter is m , the default value is 3.

Model of creep forces	
<input type="radio"/> Simplified	<input type="radio"/> FastSimA
<input type="radio"/> Mueller	<input type="radio"/> Minov's model
<input checked="" type="radio"/> FastSim	<input type="radio"/> Nonellipt.
Contact parameters	
Elasticity modulus	200 000 000 000
Poisson's ratio	0.30
Number of strips	10
Number of elements	20

Fig. 8.102. FastSim and FastSimA model parameters

- **FastSim, FastSimA**

See Sect. 8.4.3.2.2, 8.4.3.2.3 for description of the models

- **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. 8.4.3.2.4. Minov's model.

Model of creep forces	
<input type="radio"/> Simplified	<input type="radio"/> FastSimA
<input type="radio"/> Mueller	<input type="radio"/> Minov's model
<input type="radio"/> FastSim	<input checked="" type="radio"/> Nonellipt.
Contact parameters	
Elasticity modulus	200 000 000 000
Poisson's ratio	0.30
Number of strips	10
Number of elements	20
Embedding coefficient	0.400000

- **Nonelliptic contact**


This model is usually used by evaluation of wheel and rail wear evolution.

The *Rail/wheel / Contact / Friction* tab is used for setting the friction in the wheel/rail contact.

Creep forces	Friction
Left rail, tread	0.33 ...
Left rail, flange	0.33 ...
Right rail, tread	0.33 ...
Right rail, flange	0.33 ...
Transient section for side friction	
from	30 to 40 degrees

Fig.8.103. Setting coefficients of friction

See Sect. *Coefficient of friction in wheel/rail contact* for the detailed information about the coefficient of friction.

The four upper edit boxes in Fig.8.103 contain constant values of coefficients of the rail running surface f_r and on its inward side f_s . If a coefficient of friction is variable along the track, click the corresponding  button to describe the function with the help of the *Curve editor* (Chapt.3, Sect. *Object constructor/Curve editor*).

Finally, the angles β_r, β_s , which define section with different coefficients of friction in the lateral direction for $f_r \neq f_s$ should be set in the two lower edit boxes.

8.5.2.6. Setting in-train forces

Use the *Rail/Wheel | Forces tab* specify in-train forces acting on the vehicles (Fig. 8.104).

8.5.2.6.1. Setting forces and attachment points

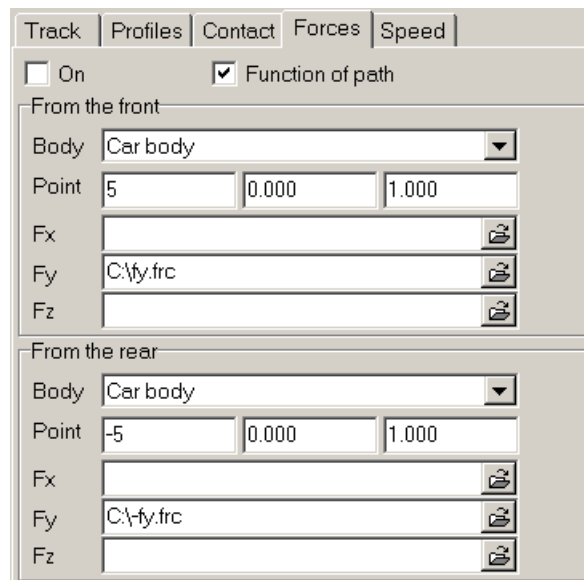



Fig.8.104. Window for specifying longitudinal forces

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 role this is a car body.
- Force application point. Coordinates X are Y set *in the body-fixed SC*, whereas the Z coordinate is set *from the rail head level*.
- Files with force component description. The force is described by three components in the body-fixed SC. To set a component, a preliminary created *.frc file should be selected by the  button, Fig. 8.104. File format description can be found in the next section.
- Use the *On* key for switch on/off the forces.
- If the *Function of path* key is on, the forces are functions of the distance else they are functions of time.

If files do not assign to some components of forces, they are zeroes.

The  button is used for plotting the assigned functions.

All the assigned data is stored in the configuration file *.rwc.

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.

8.5.2.6.2. Creating files with force description

Components of forces in Sect. 8.5.2.6.1 should be preliminary described in binary files *.frc. The force should be given in N as a function of distance or time. A 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 save format of files *.way (Sect. 8.5.1.2. Creation of track irregularities). 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 ../bin/rw directory.

8.5.2.6.3. Plotting and visualization of forces

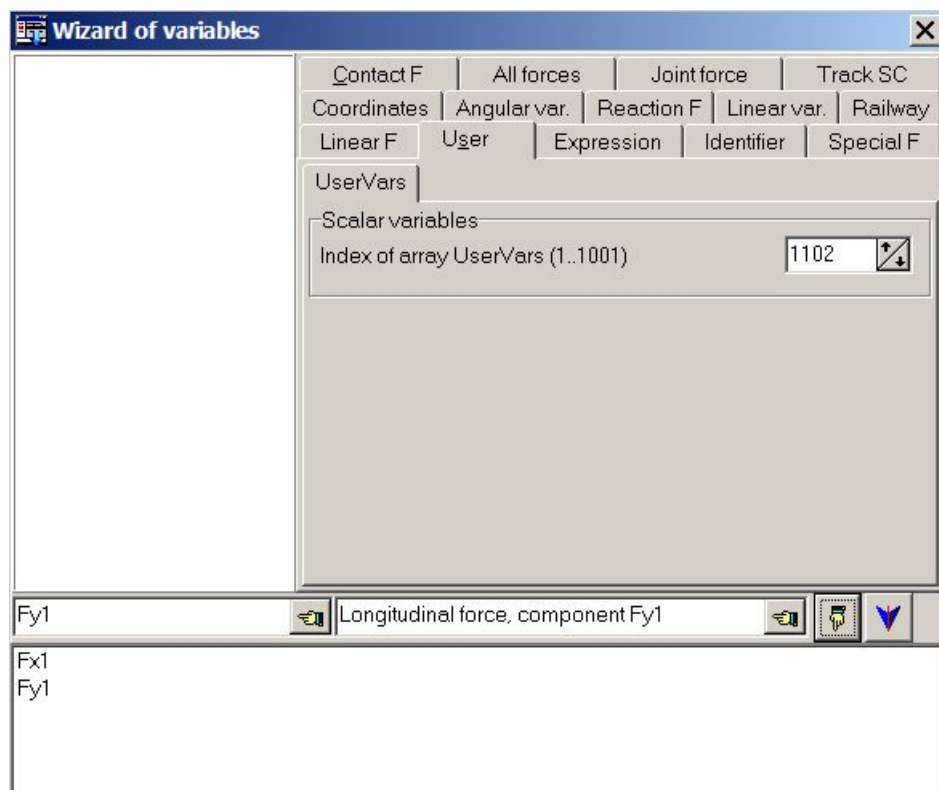



Fig.8.105. Variables corresponding to in-train forces

The *UserVars* array elements with standard numbers 1101-1106 (F_{x1} , F_{y1} , F_{z1} , F_{x2} , F_{y2} , F_{z2}) are used to create variables corresponding to force components. Use the *Tools / Wizard of variables* menu command to open the window for creating variables. Open the *UserVars* tab and set the necessary number of the element, variable name and comment, and click the  button to send the variable into the container, Fig. 8.105.

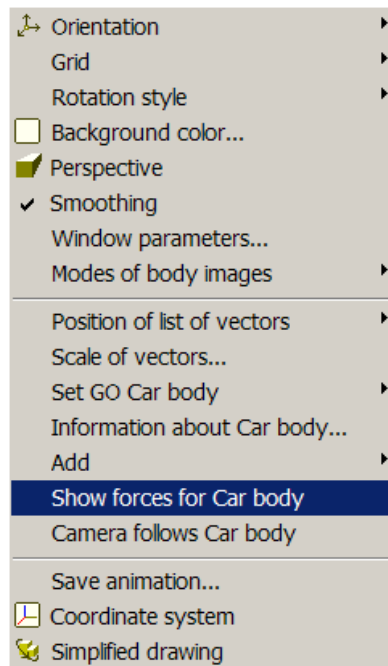


Fig.8.106. Visualization of forces for Car Body in animation window

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 Fig.8.106. If the forces are invisible by 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, Fig.8.106.

8.5.2.6.4. Example of in-train forces

Consider lateral forces applied to the front and rear couplers, Fig. 8.107. The forces are zero on the first 50 m of the vehicle motion, then they grow uniformly to 3000N on the interval from 50m to 70 m. The forces are constant till 150m, and decreases uniformly on the 20m interval. The front force is positive, the rear one is negative.

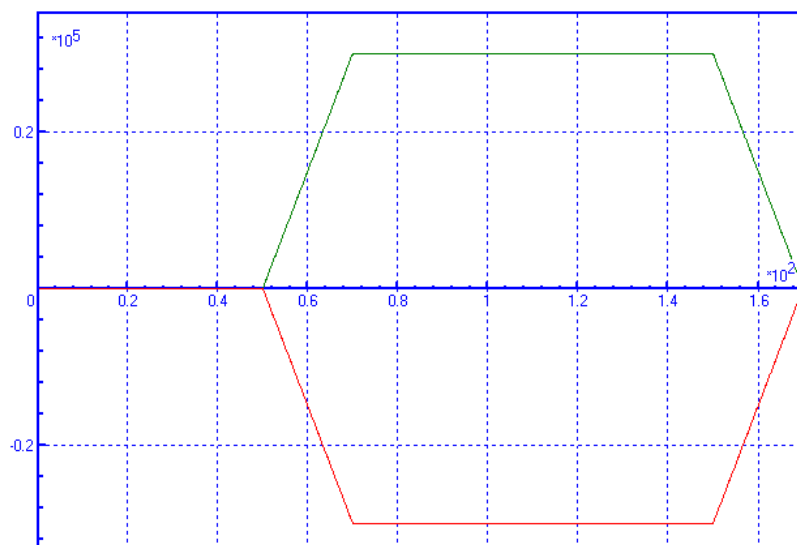



Fig.8.107. Lateral forces versus distance

Open the window for development of files with irregularities by the **Tools | Create track irregularities** menu command, open the *Points* tab and start the curve editor by the  button.

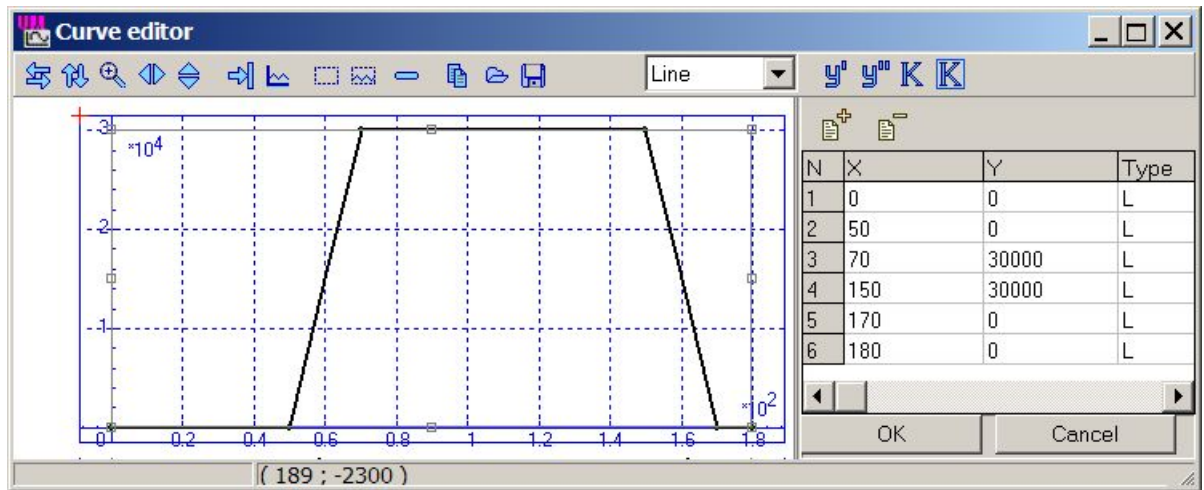


Fig.8.108. Force description by points

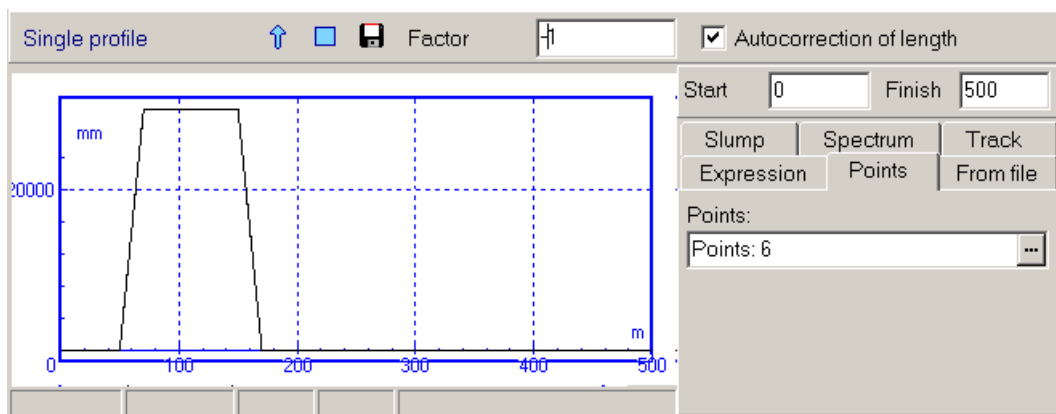
Add 6 points by the button, and set their coordinates as in Fig. 8.108. Optimize the polygon view by then button.



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 Fig. 8.108 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 save 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 , and rename the extension.

The created files can be immediately assigned to the force components, Fig.8.104, and used in simulation process.

8.5.2.7. Additional contact parameters

8.5.2.7.1. Difference in running circles radii

Track		Profiles		Contact		Forces		Speed	
Wheels		Rails		Addit.parameters					
Difference in radii		Contact options							
Wheelsets		Deviation of radii, mm							
1.		3							
2.		0							
3.		0							
4.		0							

Fig. 8.109. Difference in running circle radii of Wheelset 1

Difference in running circles radii dr (difference in running circles radii of the left and right wheel of a wheelset, Sect. *Wheelset geometry*) in millimeters are set on the *Rail/Wheel / Profiles / Addit. parameters / Difference in radii*. The parameter is assigned to the right wheels. It is positive if the left radius is greater than the right one. In the case shown in Fig. 8.109., the running circle radius of the right wheel of Wheelset 1 is decreased on 3 mm in comparison with the left wheel.

8.5.2.7.2. Parameters of contact geometry computation

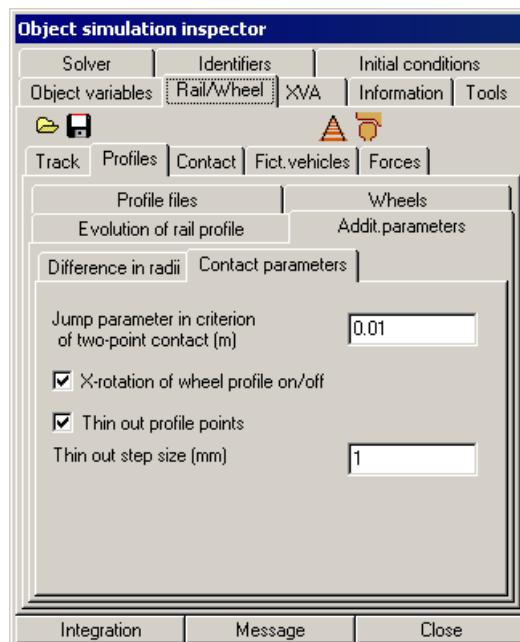


Fig.8.110. Additional parameters for contact geometry computations

When worn profiles defined by a set of points are created, 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 the *Rail/Wheel / Profiles / Addit. parameters / Contact parameters* tab.

- *Jump parameter in criterion of two-point contact.*

The parameter is used in the procedure of computing contact coordinates for verifying the fact of a two-point contact (see the η_y parameter in Sect. *Computing tables of contact points*). 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 a 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 and to divergence of solvers of equations of motion. 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 as well as the absence of profile smoothing, leads to violation of continuity of contact point positions in dependence on lateral displacements and relative rotations of profiles (see Sect. *Computing tables of contact points and Creation of wheel and rail profiles*). 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 step default value is 1mm.

8.5.3. Tools for visualization and analysis of railway vehicle dynamics

8.5.3.1. Some features of creation of variables

General information about creation and usage of variables as well as lists of variables can be found in Chapt. 4, Sect. *Variables*, *Wizard of variables*, *List of variables*. Here we consider variables, which concern dynamics of railway vehicles exclusively.

8.5.3.1.1. Rail/wheel contact variables

To get variables describing contacts between a rail and a wheel, use the *Railway* tab of the *Wizard of Variables*. For detailed information about the Wizard see Chapt.4, Sect. *Wizard of variables*.

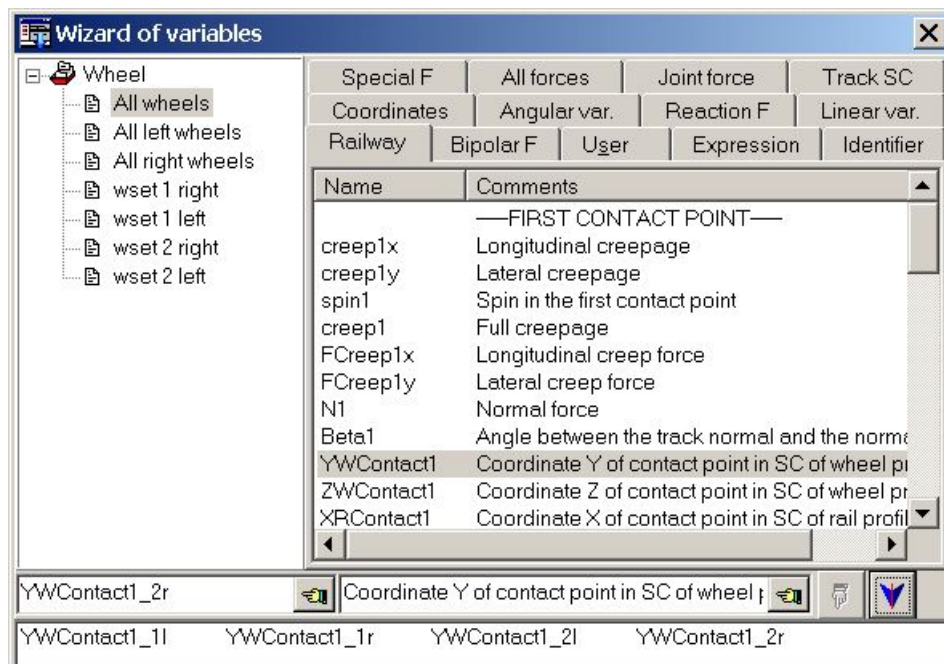




Fig.8.111. Variables for rail/wheel contacts

To create *one* variable corresponding to one of the wheel, the wheel should be selected in the list 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.

Note. The wheelsets are numbering 1,2... is ordered according to the decreasing their longitudinal coordinate.

To create *a group* of variables the ‘*All wheels*’, ‘*All left wheels*’, and ‘*All right wheels*’ items are used. Fig.8.111 shows an example when the four variables in the container are created by a single click on the  button because the ‘*All wheels*’ item is selected. If the ‘*All right wheels*’ item were selected, only two variables *YWContact1_1r* and *YWContact1_2r* corresponding to the right wheels were created by the  button click.

A *type* of a variable is selected in a list located in the right part of the wizard. The first column of the list contains standard identifiers of types. An identifier of a variable is constructed from the identifier 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 corresponds to the right wheel of Wheelset 1.

If one variable for a separate wheel is created, the used 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.

Identifier	Comments
FIRST CONTACT POINT	
One-point contact or contact on the running surface by two-point contact (Fig.8.56)	
Creep1x, Creep1y	Longitudinal and lateral creepages ξ_x, ξ_y , the variables are used for computing creep forces, Sect. 8.4.3.2.
Spin1	Spin φ . The variables are used for computing creep forces by FASTSIM, Sect. 8.4.3.2.
Creep1	Module of the creepage vector $\xi = \sqrt{\xi_x^2 + \xi_y^2}$
FCreep1x, FCreep1y	Longitudinal and lateral creep forces, Sect. 8.4.3.2. The lateral creep force corresponds to the F_1 force in Fig. 8.56. Unit – N.
N1	Normal force N_1 , Fig. 8.56. Unit – N.
Beta1	Contact angle: angle β_1 between the N_1 and the Z-axis of the track SC, Fig.8.56. Unit – rad. The angle is positive for inclination of the normal forces inward the track.
YWContact1, ZWContact1	Coordinates y (lateral) and z (vertical) of the contact point in SC of the wheel profile (Sect.8.2.3.2). Unit – m.
XRContact1, YRContact1, ZRContact1	Coordinates y (lateral) and z (vertical) of the contact point in SC of the rail profile (Sect.8.3.1.1). Unit – m.
Zlifting	Vertical lifting the wheel running surface for profiles allowing a two-point contact (Fig.8.45, left). Unit- m.
Mwear1, Awear1, Swear1	Flanges wear factors for a one-point contact of for the running surface contact in the two-point contact: power, work, and specific work. The factor are computed according to formulae $M_{wear} = -F_{1x}v_{1x} - F_{1y}v_{1y}$ - the scalar product of the flange creep force and the sliding velocity at the contact; unit - W; $A_{wear} = \int_0^t M_{wear} dt$; unit – J; $S_{wear} = A_{wear} / S$, where S is the path; unit – J/m;
Area1	Area of the contact patch. Unit – m ² .
Pressure1	Maximal pressure in the contact. Unit – Pa.
SECOND POINT CONTACT	
Flange contact for profiles allowing a two-point contact (Fig.8.56, right)	
Creep2x, Creep2y	Longitudinal and lateral creepages ξ_x, ξ_y , the variables are used for computing creep forces, Sect. 8.4.3.2.
Spin2	Spin φ . The variables are used for computing creep forces by FASTSIM, Sect. 8.4.3.2.

Creep2	Module of the creepage vector $\xi = \sqrt{\xi_x^2 + \xi_y^2}$
Fcreep2x, Fcreep2y	Longitudinal and lateral creep forces, Sect. 8.4.3.2. The lateral creep force corresponds to the F_2 force in Fig. 8.56. Unit – N.
N2	Normal force N_2 , Fig. 8.56. Unit – N.
Beta2	Contact angle: angle β_2 between the N_2 and the Z-axis of the track SC, Fig.8.56. Unit – rad. The angle is positive for inclination of the normal forces inward the track.
YWContact2, ZWContact2	Coordinates y (lateral) and z (vertical) of the contact point in SC of the wheel profile (Sect.8.2.3.2). Unit – m.
XRContact2, YRContact2, ZRContact2	Coordinates y (lateral) and z (vertical) of the contact point in SC of the rail profile (Sect.8.3.1.1). Unit – m.
N2y	Guiding force: projection of N_2 on y-axis of the track SC, $N_{2y} = \mp N_2 \sin \beta_2$, where the mines is set for the left wheel.
Mwear2, Awear2, Swear2	Flanges wear factors for a two-point contact: power, work, and specific work. The factor are computed according to formulae $M_{wear} = -F_{2x}v_{2x} - F_{2y}v_{2y}$ - the scalar product of the flange creep force and the sliding velocity at the contact; unit - W; $A_{wear} = \int_0^t M_{wear} dt$; unit – J; $S_{wear} = A_{wear}/S$, where S is the path; unit – J/m;
Area2	Area of the contact patch. Unit – m ² .
Pressure2	Maximal pressure in the contact. Unit – Pa.
GENERAL VARIABLES Variables are not directly connected with one of the contact point	
CSafety	Derailment quotient (Russian version). For a wheel with the two-point contact is computed according to the formula $\lambda = \frac{F_z}{ F_y } \frac{tg\beta - \mu}{\mu tg\beta + 1},$ where F_y, F_z are the total forces acting on the wheel from the rail in the lateral and vertical directions; μ is the coefficient of friction, β is the flange contact angle. If $\lambda > 5$ then the value $\lambda = 5$ is accepted. The value $\lambda = 5$ is set if there is no contact between a wheel and a rail (a full separation). The motion is safe if $\lambda > 1.2$.
CSafetyRefined	Refined derailment quotient (Russian version) $\lambda_e = \frac{F_z}{ F_y } \frac{tg\beta - \mu_e}{\mu_e tg\beta + 1}.$ Differs from λ in coefficient μ_e , which is less than coefficient of friction and computed as

	$\mu_e = \frac{f_y}{N},$ <p>where f_y is the lateral creep force in the flange contact, and N is the corresponding normal force. Less conservative than the Nadal criterion. Usually $\lambda_e > \lambda$</p>
CSafetyFx-Fy, CsafetyFx-N	<p>Variants of Refined derailment quotient (Russian version)</p> $\lambda_{F_y} = \frac{F_z}{ F_y } \frac{tg\beta - \mu_{F_y}}{\mu_{F_y} tg\beta + 1}, \lambda_N = \frac{F_z}{ F_y } \frac{tg\beta - \mu_N}{\mu_N tg\beta + 1}.$ <p>Coefficients are computed according to formulas</p> $\mu_{F_y} = \frac{\mu F_y^h}{\sqrt{(F_y^h)^2 + F_x^2}}, \mu_N = \mu \sqrt{1 - \left(\frac{F_x}{\mu N^h}\right)^2}.$ <p>Forces included in these formulas are</p> $F_y^h = F_z \sin \beta + F_y \cos \beta,$ $N^h = F_z \cos \beta \mp F_y \sin \beta,$ <p>where the mines sign corresponds to the left wheel. Less conservative than the Nadal criterion. Usually $\lambda_{F_y}, \lambda_N > \lambda$.</p>
Nadal	<p>Derailment quotient</p> $\lambda = \mp F_y / F_z = \left(\frac{Y}{Q}\right)$ <p>where the mines sign corresponds to the left wheels. The value $\lambda_2 = 0$ is set if there is no contact between a wheel and a rail (a full separation).</p>
Weinstock	<p>Derailment criterion proposed by Weinstock. Less conservative than the Nadal criterion</p>
psi	<p>Angle of attack: the yaw angle of wheels with respect to the rail.</p>
Fx, Fy, Fz	<p>Total forces acting on the wheel along axes of the track system of coordinates (Sect.8.3.1.3). Longitudinal, lateral and vertical forces.</p>
dyW/R	<p>Lateral displacement of the wheel relative to the rail.</p>
D.ratio (W/R)	<p>Dynamic ratio on the rail/wheel level/ The value is computed according to the formula</p> $(F_z - F_{z0})/F_{z0},$ <p>where F_z is the total vertical force in the track SC, F_{z0} is the static load for a wheel.</p>
dyTrack, dzTrack	<p>Horizontal and vertical rail irregularities at the wheel position.</p>
Path	<p>Path passing since start of simulation. The variable is used for plotting other variables in dependence on the path.</p>

8.5.3.1.2. Kinematic characteristics relative to track system of coordinates

Kinematical variables of bodies especially in the case of use the inertial SC0 (Sect. *Base coordinate system*) should be often projected on the track system of coordinates TSC (Sect. *Track system of coordinates*). Note that axes of the TSC and SC0 in a straight track are parallel, and projections of vectors on these SC are the same.

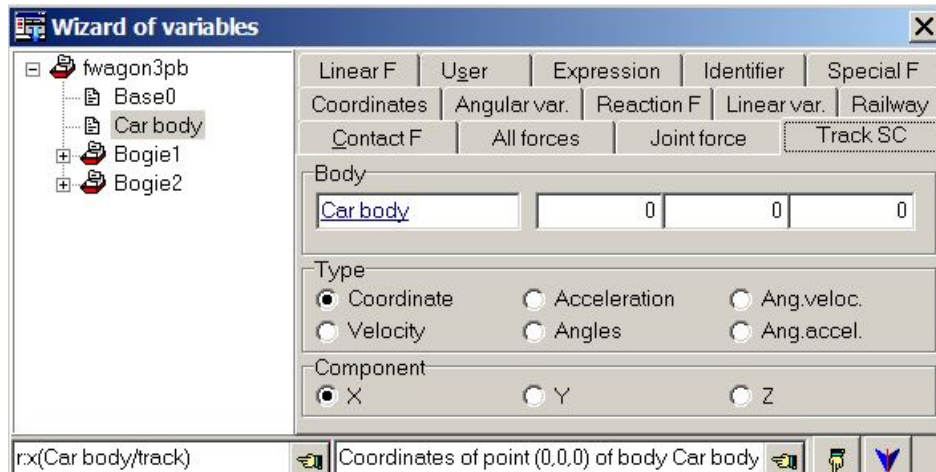


Fig.8.112. Kinematic characteristics of bodies in the track SC

Use the Track SC tab of the Wizard of variables to get any kinematic variable in projection of the TSC (Fig.8.112). To create the variable

- Select a body in the list in the left part of the wizard;
- 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 lateral component of acceleration, either an uncompensated acceleration or a usual acceleration is selected (Fig.8.112).

8.5.3.1.3. Use of *Path* variable

The *Path* variable is created on the *Railway* tab of the wizard of variables. It denotes the distance passed by the vehicle since start the simulation. This variable has the standard identifier, 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.

- **Use of *Path* variable in graphic windows**

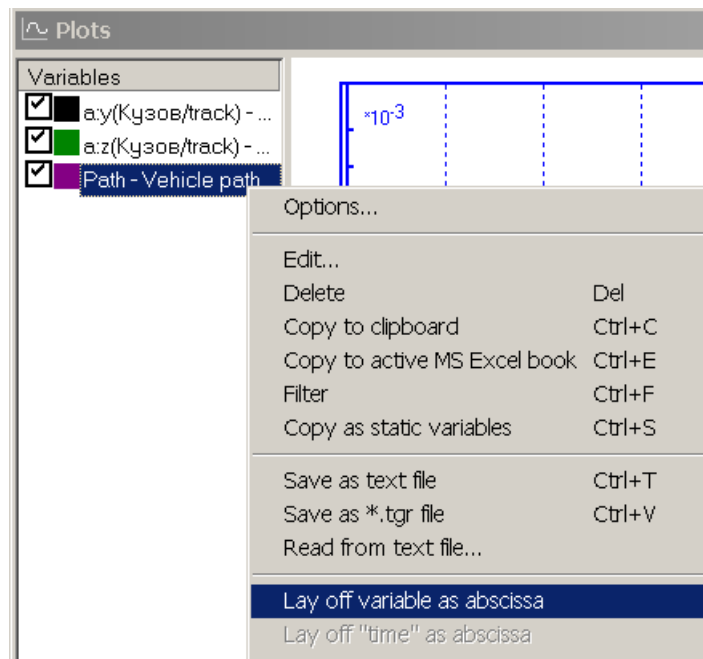


Fig.8.113. Plotting variables versus the path

Drag the *Path* variable into the graphical window list. Select the variable and click the right mouse button. Click the Lay off variable as abscissa in the pop up menu.

- Use of *Path* variable in list of calculated variables

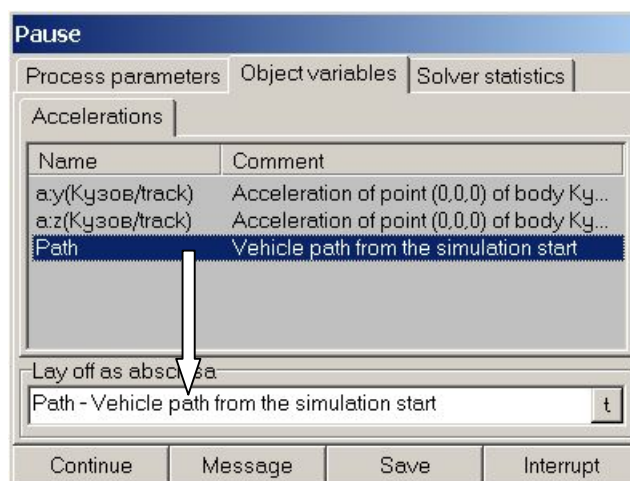


Fig.8.114. Setting dependence of calculated variables on the path


It is strongly recommended that list of object variables for a railway vehicle always contains the *Path* variable. If it does not include this variable, add it to the list before simulation of scanning the object in a standard manner (Chapt.4, Sect. *Assignment and usage of a list of automatically calculates variables*). To get dependence of computed variables on the path 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, tables processors, etc., are processed in dependence on the path.

8.5.3.2. Animation window

General information about usage of animation windows can be found in Chapt.3, Sect. *Animation window* and in Chapt.4, Sect. *Animation window in the Simulation module*.

In addition to standard functions, visualization of rails and slippers is available for railway vehicles (the *Show gauge* check in the *Rail/Wheel | Track | Macrogeometry* tab of the object simulation inspector).

8.5.3.3. Contact animation window

The *Contact animation window* 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, the  button on the tool panel or the *Ctrl+N* hot key to call the window.

Parameters of the contact animation are located in the bottom of the window. These parameters allow the user to turn on/off visualization of contact forces, change vector scales and profile positioning (horizontal or vertical manner).

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.

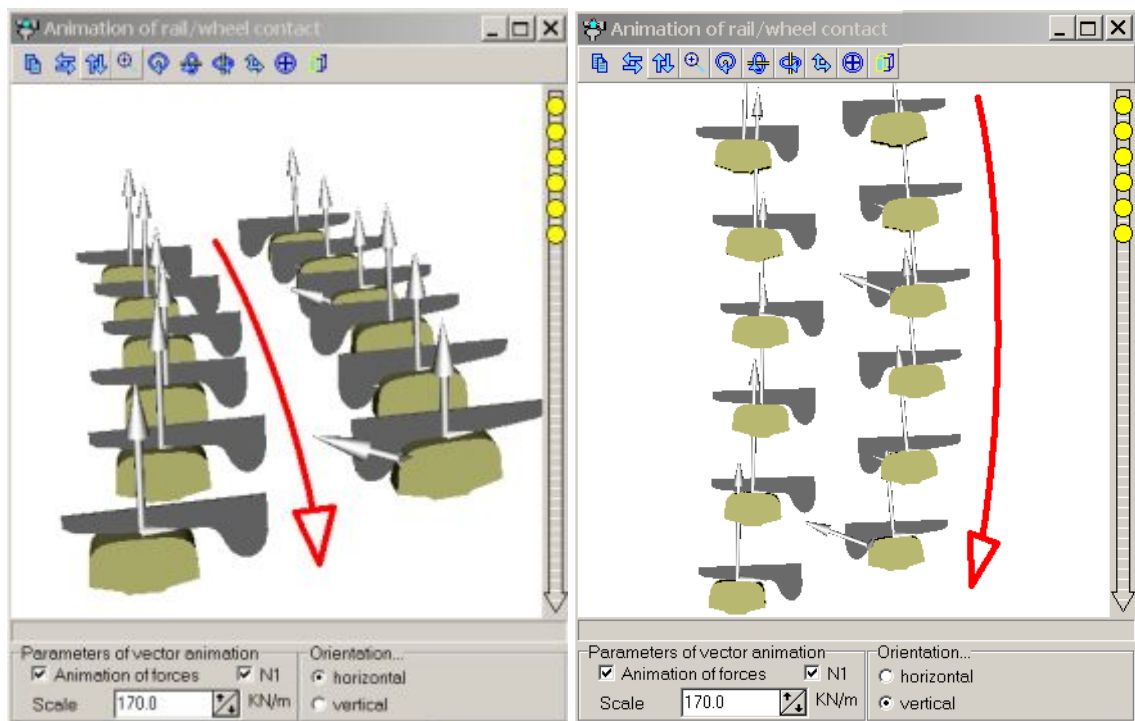



Fig.8.115. Direction of motion in the animation window

8.5.3.4. Contact patch animation

Use the **Tools | Contact patch** menu command or the  button on the tool bar to call the contact patch animation window. Only one contact patch for a wheel is presented in the window even if a two-point contact takes place. The patch is presented either for the single contact point or for the contact on the running surface in the case of a two-point contact.

The user can optionally watch the follows:

- elliptic contact patch (FASTSIM and FASTSIM_A);
- vector of creep forces;
- adhesion (dark color) and sliding (light color) regions within the patch (FASTSIM and FASTSIM_A);
- distribution of contact stresses within the contact patch (FASTSIM and FASTSIM_A).

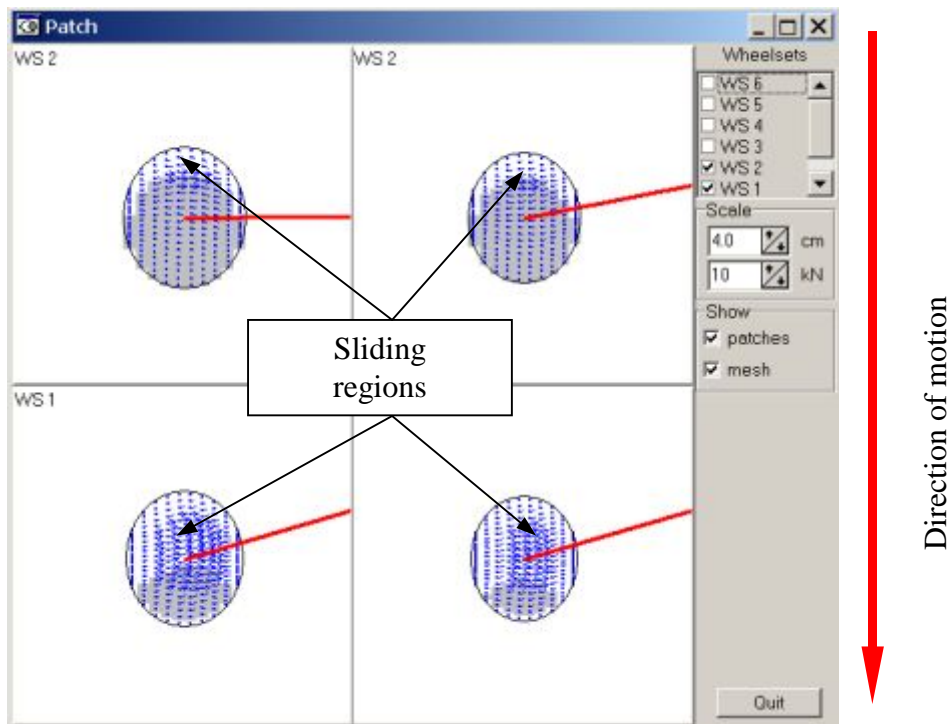


Fig.8.116. Contact patches while curving

8.5.3.5. Table processor

The following functional are added along with UM Loco module.

Functional for UM Loco module

Functional	Description
_3Min_Mean _3Max_Mean _3Max_Zero _4Min_Mean _4Max_Mean _4Max_Zero	Mean value of three/four minimums or maximums. All functional work according to the same algorithms with some minor differences. Firstly the list of local extremums are created. Local minimums are taken for _3Min_Mean and _4Min_Mean , local maximums are taken for _3Max_Mean , _3Max_Zero , _4Max_Mean and _4Max_Zero . 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 only plot intersects the abscissa axis (for functional with Zero postfix) or mean value (for functional with Mean postfix) between these extremums. Otherwise the only extremum is chosen. Then 3 or 4 extremums maximum/minimum extremums are extracted. For the _4Min_Mean , _4Max_Mean and _4Max_Zero functional the smallest/biggest one is deleted. Then the mean value for the rest extremums are calculated as a results of the functional.
Ride_Comfort_G	Ride comfort factor for horizontal direction according to Russian (ex-USSR) standard 24.050.16-85.
Ride_Comfort_V	Ride comfort factor for vertical direction according to Russian (exUSSR) standard 24.050.16-85.

References

1. Mathematical modeling vibrations of railway vehicles / V.F. Ushkalov et al; V.F. Ushkalov (Ed); AS USSR. Inst. of techn. mech. – Kiev: Nauk. dumka, 1989. – 240 p. (in Russian)
2. J.J. Kalker and J. Piotrowski, Some New Results in Rolling Contact, *Vehicle System Dynamics*, 18 (1989).
3. Kik W., Piotrowski J. A Fast Approximate Method to Calculate Normal Load at Contact between Wheel and Rail and Creep Forces During Rolling. *Proceedings of 2nd mini. conf. Contact Mechanics and Wear of Rail/Wheel Systems*, 1996, P. 52–61.
4. K.L. Johnson, *Contact Mechanics*, Cambridge University Press (1985).