UM Simulation Program

User's manual

2014

Contents

4. UM Simulation program

UM Simulation program is a separate application (**UM Simulation**). It loads the object equations of motion using the created dynamic-linked library of the object (*umtask.dll*) or generates equations of motion in the numeric-iterative form, see Chapter 2.

To run this program, use the **Object | Simulation…** menu command in the UM Input program or run the **UM Simulation** file directly. You can change the active object in the UM Simulation with the help of the **File | Open** or **File | Reopen** menu command.

Numerical analysis of the equations of motion includes:

- numerical integration of the equations of motion;
- computation of equilibrium and linear analysis.

4.1. Options of UM Simulation program

Use the **Tools | Options…** command of the main menu (or F10 key) to open a window with the **UM Simulation** options. Consider tabs of the dialog box.

4.1.1. General

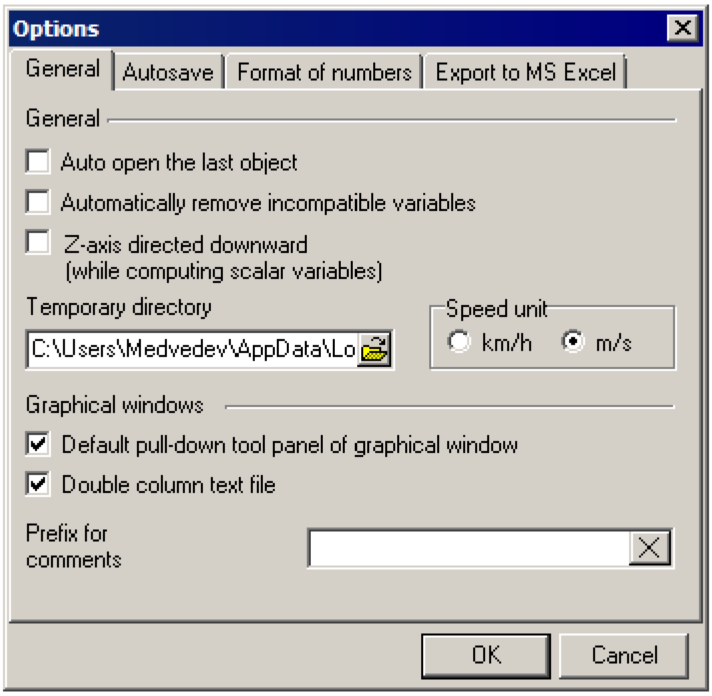


Figure 4.1. General options

- **Auto open the last object** – always open the latest open object.
- **Automatically remove incompatible variables**. If the switch is turned on, all variables incompatible with the current object are automatically removed from animation and graphical windows while reading a configuration file, as well as from lists of variables. The option is ignored for lists of calculated variables (Sect. 4.3.3.3).
- **Z-axis directed downward** – change signs of vector projections of variables on the Z- and Y-axis.
- **Temporary directory** used for temporary files, e.g. while creating animation files (*.avi).
- **Speed unit**: here the user can select the unit for initial vehicle speed specified by the identifier **v0**. The option is valid for UM modules: **UM Loco, UM Automotive, UM Tracked Vehicle** only.
- **Default pull-down panel of graphical window tool** – turns on/off the corresponding mode of the graphical window tool panel (Sect. 4.3.4)

Double column text file: the option sets one of two formats for saving simulation results from a graphic window. If the option is not checked, *n* variables are stored in *n*+1 columns: time and *n* variables. If the option is checked, n successive groups are stored in the format time as the first column and the i^{th} variable as the second one, i=1,…,n, Sect. 4.3.4.1.

- **Prefix for comments** – a character or a set of characters, which will be set at initial parts of comment lines while copying calculated variables in the clipboard and a text file (Sect. 4.3.4.1). The option is useful for the export of computed variables into external programs (Maple, MATHEMATICA etc.).

4.1.2. Autosave

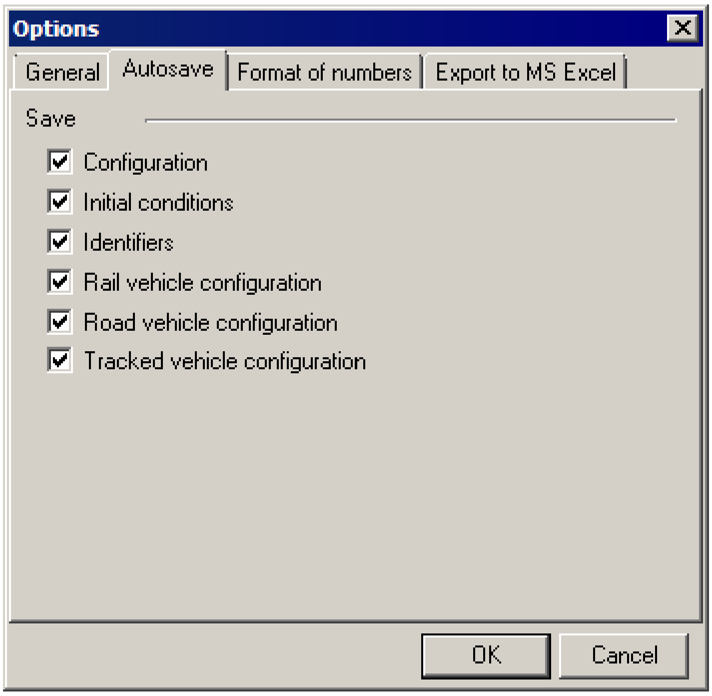


Figure 4.2. Options of automatic saving

The **Autosave** options are used for storage of **UM Simulation** program desktop and object state after the latest simulation.

Switches:

- **Configuration** – saves the simulation program desktop: positions, parameters and variables of graphical and animation windows, integration method and its parameters (the *last.icf* file);
- **Initial conditions** – saves the latest used coordinates and their time derivatives (the *last.xv* file);
- **Identifiers** – saves the latest object identifier values (the *last.par* file);
- **Rail vehicle configuration** – saves the latest options for a rail vehicle model (profile files, track parameters etc., the *last.rwc* file, **UM Loco** module).
- **Road vehicle configuration** – saves the latest options for a road vehicle model (the *last.car* file, **UM Automotive** module).
- **Tracked vehicle configuration** – saves the latest options for a tracked vehicle model (the *last.tvc* file, **UM Tracked Vehicle** module).

4.1.3. Format of numbers

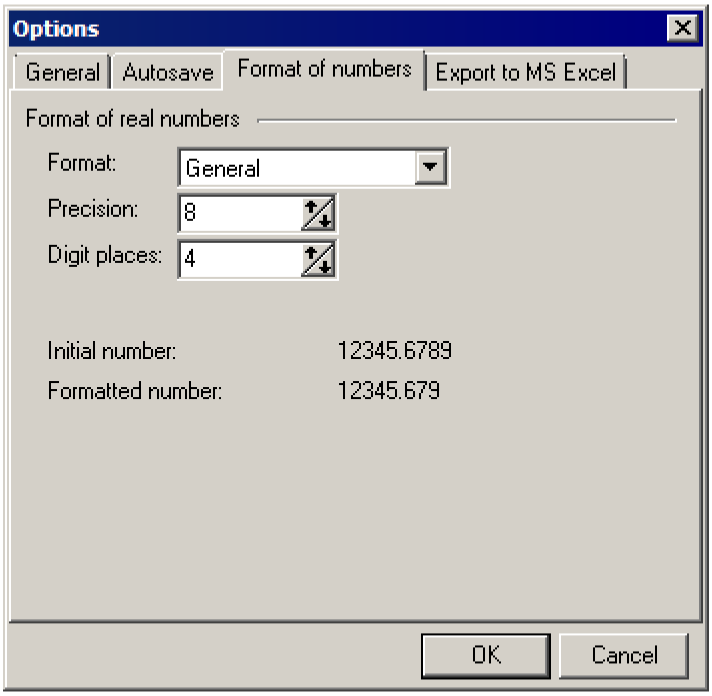


Figure 4.3. Format of numbers

The string format of floating point numbers is valid for the following tools

Table processor (Sect. 4.3.7);

Graphical window (while copying calculated variables in the clipboard or in a text file, Sect. 4.3.4.1).

Formats:

- General

General number format. The value is converted to the shortest possible decimal string using fixed or scientific format. Trailing zeros are removed from the resulting string, and a decimal point appears only if necessary. The resulting string uses fixed point format if the number of digits to the left of the decimal point in the value is less than or equal to the specified **Precision**, and if the value is greater than or equal to 0.00001. Otherwise the resulting string uses scientific format, and the **Digits** parameter specifies the minimum number of digits in the exponent (between 0 and 4).

- **Exponential**

Scientific format. The value is converted to a string of the form "-d.ddd...E+dddd". The resulting string starts with a minus sign if the number is negative, and one digit always precedes the decimal point. The total number of digits in the resulting string (including the one before the decimal point) is given by the **Precision** parameter. The "E" exponent character in the resulting string is always followed by a plus or minus sign and up to four digits. The **Digits** parameter specifies the minimum number of digits in the exponent (between 0 and 4).

- **Fixed**

Fixed point format. The value is converted to a string of the form "-ddd.ddd...". The resulting string starts with a minus sign if the number is negative, and at least one digit always precedes the decimal point. The number of digits after the decimal point is given by the **Digits** parameter--it must be between 0 and 18. If the number of digits to the left of the decimal point is greater than the specified **Precision**, the resulting value will use scientific format.

4.1.4. Data export to MS Excel

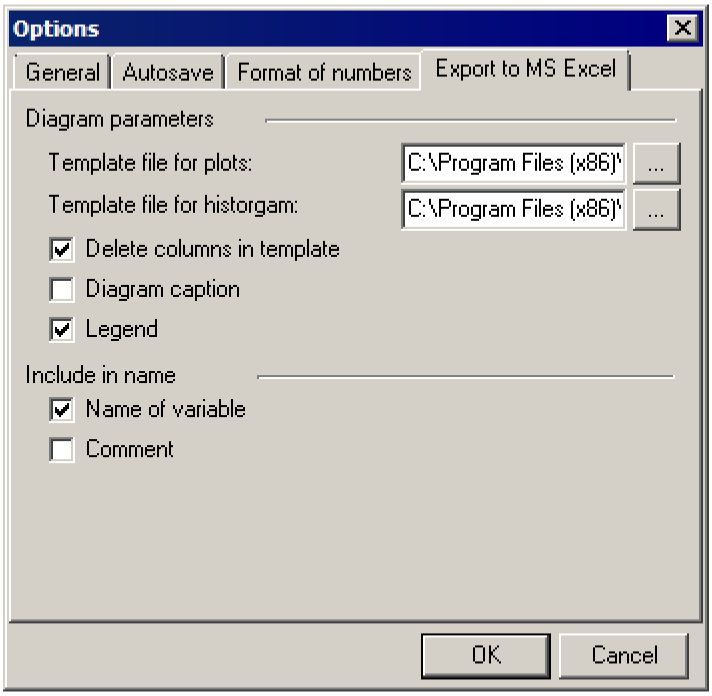


Figure 4.4. Options of export in MS Excel

Directories settings:

- **Template file for plots** – choose a template graph file for MS Excel diagrams (the standard templates are located in the .*Templates* directory, see Sect. 4.3.4.5).
- **Template file for histogram** – choose a template histogram file for MS Excel diagrams (the standard templates are located in the .*Templates* directory, see Sect. 4.3.4.5).

Switches:

- **Delete columns in template** – temporary columns in templates will be deleted while exporting data from UM.
- **Diagram caption** – if turned on, the graphical window caption is assigned to the diagram caption, if not – the template caption is kept.
- **Legend** – if turned on, a legend according to the Include in name of column switches will be added.
- **Name of variable** – include the variable name in the column name (ignored if the Legend switch is turned off).

- **Comments for variable** – include variable comments in the column name (ignored if the switch **Legend** if turned off).

Note.	If there are several diagrams in the template file then active one is used for export. On default the first diagram is active.

4.2. Menu of UM Simulation program

4.2.1. File

- **Open**

  Opens an object for simulation. Choose an object from the list. The current object will be <u>replaced</u> with the new one. So you can simulate different objects without leaving the UM Simulation program (Figure 4.5).

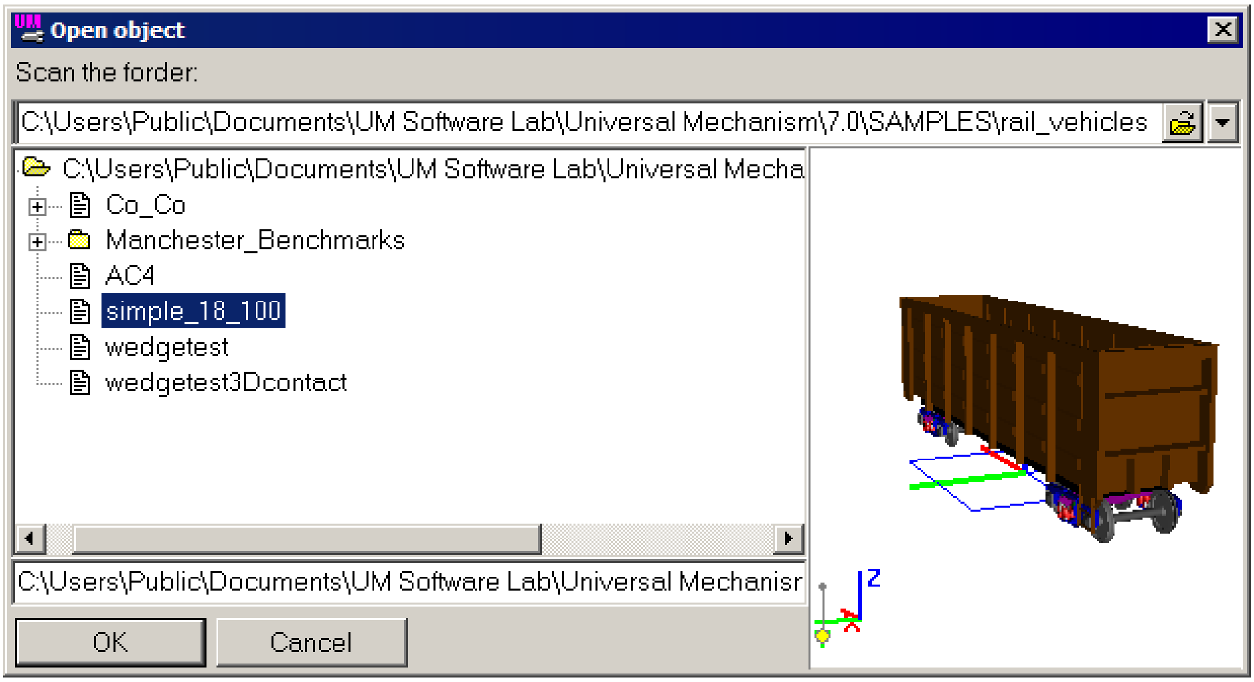


Figure 4.5. Open object dialog

- **Reopen**

  Allows opening one of the recently used objects.

- **Read configuration**

  a) *Desktop*: reads the **.icf* file, which includes information concerning animation and graphical windows, numerical method and so on.

  b) *Other menu subitems*: reads a *full* configuration, which includes desktop, initial conditions, values of identifiers, as well as a special configuration if presented (e.g. a configuration for rail vehicles, car, tracked vehicle), Figure 4.6.

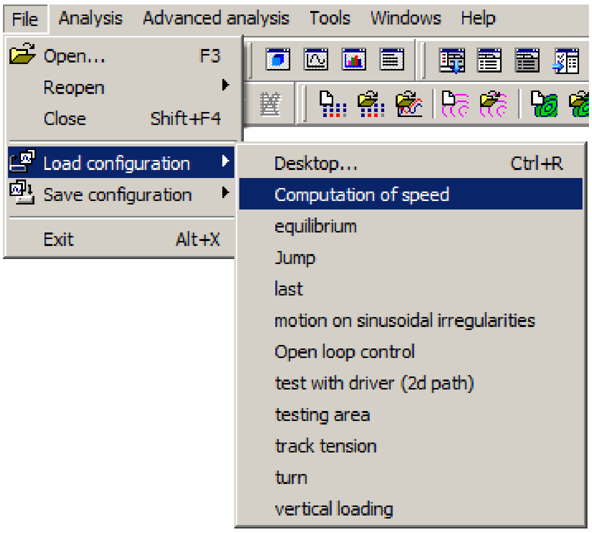


Figure 4.6. Reading of a full configuration

- **Save configuration**

  a) *Desktop*: save desktop information to the **.icf* file;

  b) *All parameters*: save desktop, initial conditions, current values of identifiers, as well as a special information (e.g., for a rail vehicle). The stored configuration might be read with the help of the *Read configuration* item.

4.2.2. Analysis

This menu commands give the user access to the procedures of the object dynamics analysis.

- **Simulation**

Numerical integration of the active object equations of motion; the command opens the simulation inspector (Sect. 4.4.1).

- **Linear analysis**

Calculation of equilibrium positions, linearization of equations and their analysis: natural frequencies and modes, eigenvectors and eigenvalues, root locus, etc. (Sect. 4.5. Linear analysis).

4.2.3. Advanced analyses

This menu give you access to the procedures of advanced analyses of the object: scanning, optimization and approximation. Exhaustive information about these procedures you can find in Chapter 6. Requires UM Experiments.

4.2.4. Tools

- **Animation window** (Ctrl+A)

Opens a new *animation window* (Sect. 4.3.5). The number of animation windows is unlimited.

- **Graphical window** (Ctrl+G)

Opens a new *graphical window* (Sect. 4.3.4). The number of animation windows is unlimited.

- **Text editor** (Ctrl+Alt+T)

Opens a built-in *text editor*.

- **Wizard of variables** (Ctrl+M)

Opens a new *wizard of variables* (Sect. 4.3.2). Use the *wizard of variables* to create new variables. Drag and drop variables (vectors, trajectories or a list of variables) in an animation window, a graphical window, and a list of variables.

- **List of variables** (Ctrl+L)

Opens a new *list of variables* (Sect. 4.3.2.16). To avoid creating lots of variables every time you simulate an object use a *list of variables* to save the dynamic characteristics of the object you are interested in.

- **List of calculated variables**

Calls a list of computed variables for analysis and comparison (Sect. 4.3.3.3).

- **Table processor**

Opens a new *table processor*. The table processor (Sect. 4.3.7) lets you carry out statistical analysis (minima, maxima, mean, root mean square, etc.).

- **Symbolic calculator**

Opens a new *symbolic calculator*. The symbolic calculator lets you calculate complicated relations mechanical systems parameters.

- **Calculator of orientation**

Opens the *Calculation of orientation tool*. This tool lets you calculate orientation angles, vectors/angles of turning, quaternion and direct cosine matrices.

- **Statistics** (Ctrl+I)

Opens a new Statistics window (Sect. 4.3.8).

- **Identifier macros**

Tool for creating macro-commands for simultaneous assignment of numeric values for groups of identifiers, Sect. 4.3.11.

- **Force analysis** (Ctrl+F)
  Force element response in the frequency domain (Sect. 4.3.9).

- **Create track irregularities** (Ctrl+T)
  Opens *Track irregularities* tool (Chapter 12).

- **Tire model parameters**
  Opens *Tire model* tool (Chapter 12).

- **Control Panel editor**
  Opens the *Control Panel editor* window (Sect. 4.3.10.2). Use the editor to create a new *Control Panel* (Sect. 4.3.10).

- **Open Control Panel**
  Opens a previously created control panel from file (*.cp).

- **Options**
  Enables a window with parameters of the system (Sect. 4.1).

- **Tool panels**
  List of visible groups of buttons on the toolbar.

4.2.5. Windows

- **List of windows** (Alt+0)

  Opens a list of available (unclosed) windows (Figure 4.7).

Figure 4.7. List of windows

- **Previous**

  Sets in front the previous window.

- **Next**

  Sets in front the next window.

- **Current inspector**

  Sets in front current inspector (*Linear analyses* (Sect. 4.5) or *Object simulation inspector* (Sect. 4.4).

4.3. UM Simulation program notions and tools

4.3.1. Variables

A *variable* is any characteristic of an object, which can be computed with the help of UM, for example, a reaction force in a joint, module of a velocity of a point, vector of an applied force, coordinate of an arbitrary point and so on.

The variables are created with the help of the *wizard of variables* (Sect. 4.3.2) and can be stored in a *list of variables* (Sect. 4.3.2.16). Variables can be scalars or vectors. *A graphical window* (Sect. 4.3.4) and a list of variables operate with scalar variables, *an animation window* (Sect. 4.3.5) with vectors and trajectories.

4.3.2. Wizard of variables

The *wizard of variables* is used for creating variables. Use the **Tools | Wizard of variables…** menu command or **Ctrl+M** hot key to open it.

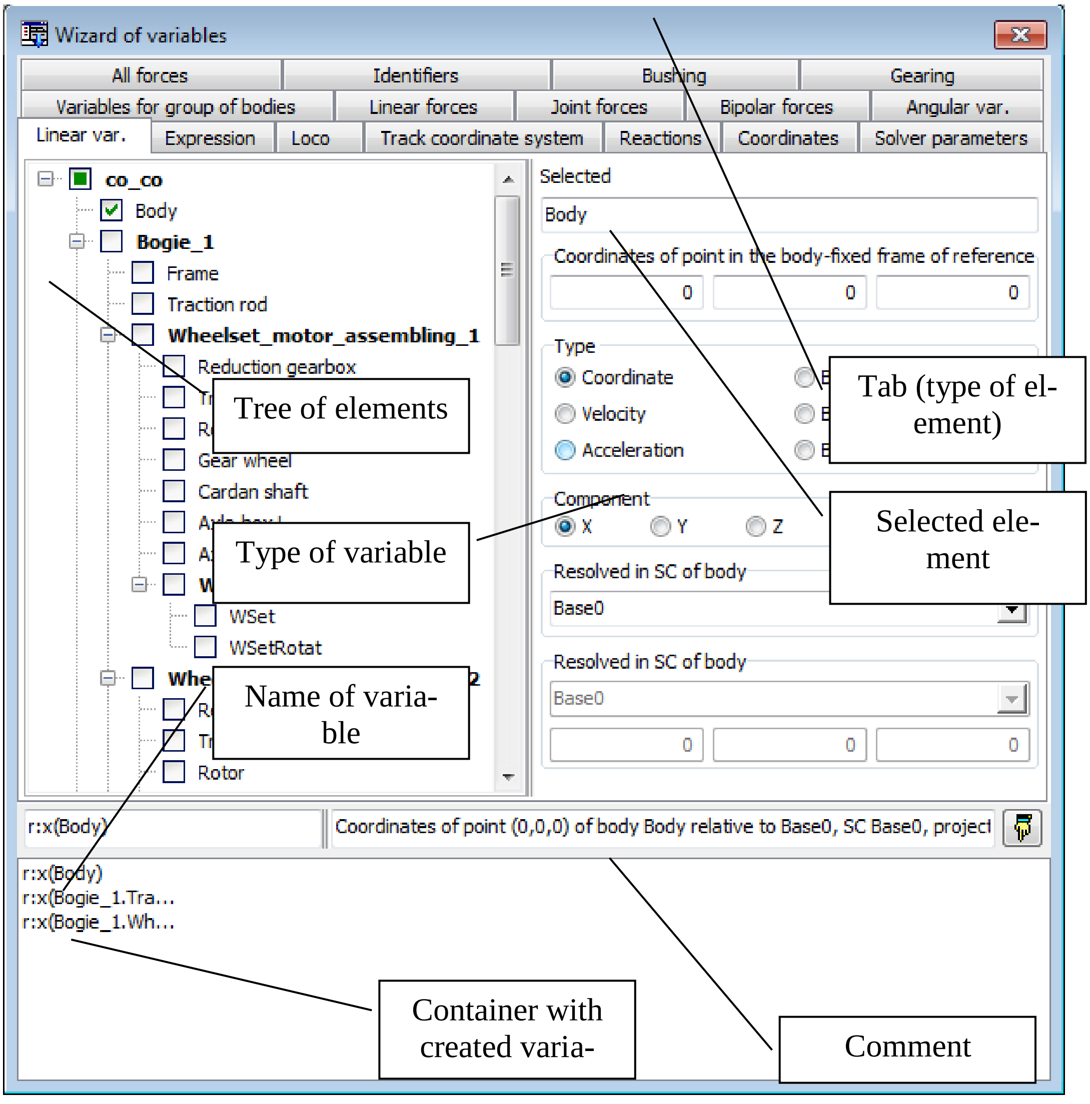
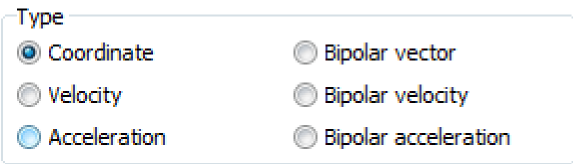


Figure 4.8. Wizard of variables

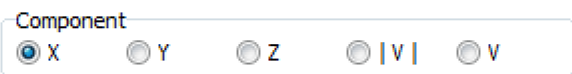
Tabs are the main elements of the wizard tool. Each the tab corresponds to an object element or variable of certain type (coordinates, kinematical variables, forces etc.). The corresponding list or tree of elements depending on the tab is located in the left part of the wizard.

Creation of a new variable is executed in the following sequence:

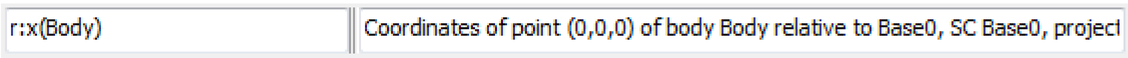
- choose a necessary tab (for instance, *Linear var[iables]*);
- choose an element from the tree by clicking the left mouse button on an element name (body, joint, force element…);
- select a type of the variable (for example: coordinate, velocity, acceleration or: force, moment);



- specify additional parameters defining the variable (for example, coordinates of a point, which velocity should be calculated);
- specify the *Component* parameter for a vector variable (velocity, acceleration, force etc.): projection (X,Y or Z), module (|V|) or the vector itself (V); if a projection is selected, the corresponding system of coordinate should be specified (the *Resolved in SC of body* parameter);



- user can change a standard name and comment of the variable



- send the variable into the container clicking the  button.

0After that drag the variable (or a group of variables) into the corresponding window (a graphical window or an animation window for vectors) or into a list of variables (for scalar variables).

Remark. In general, the variables created for an object are not valid for another one.

Now consider creating different variables.

4.3.2.1. Coordinates

The *Coordinates* tab (Figure 4.9) allows creating a variable, which is a joint coordinate or its first and second derivative with respect to time. The tree of element for this tab contains all object joints and the corresponding coordinates. Subsystem and local indices for every coordinate are pointed out.

Figure 4.9 contains the description of the following variable: coordinate 21 subsystem 1 – the local coordinate in the joint *Bogie_1.Wheelset_motor_assembly1.jRotor_Cardan_shaft*.

The variable can be added either to a graphical window or to a list of variables.

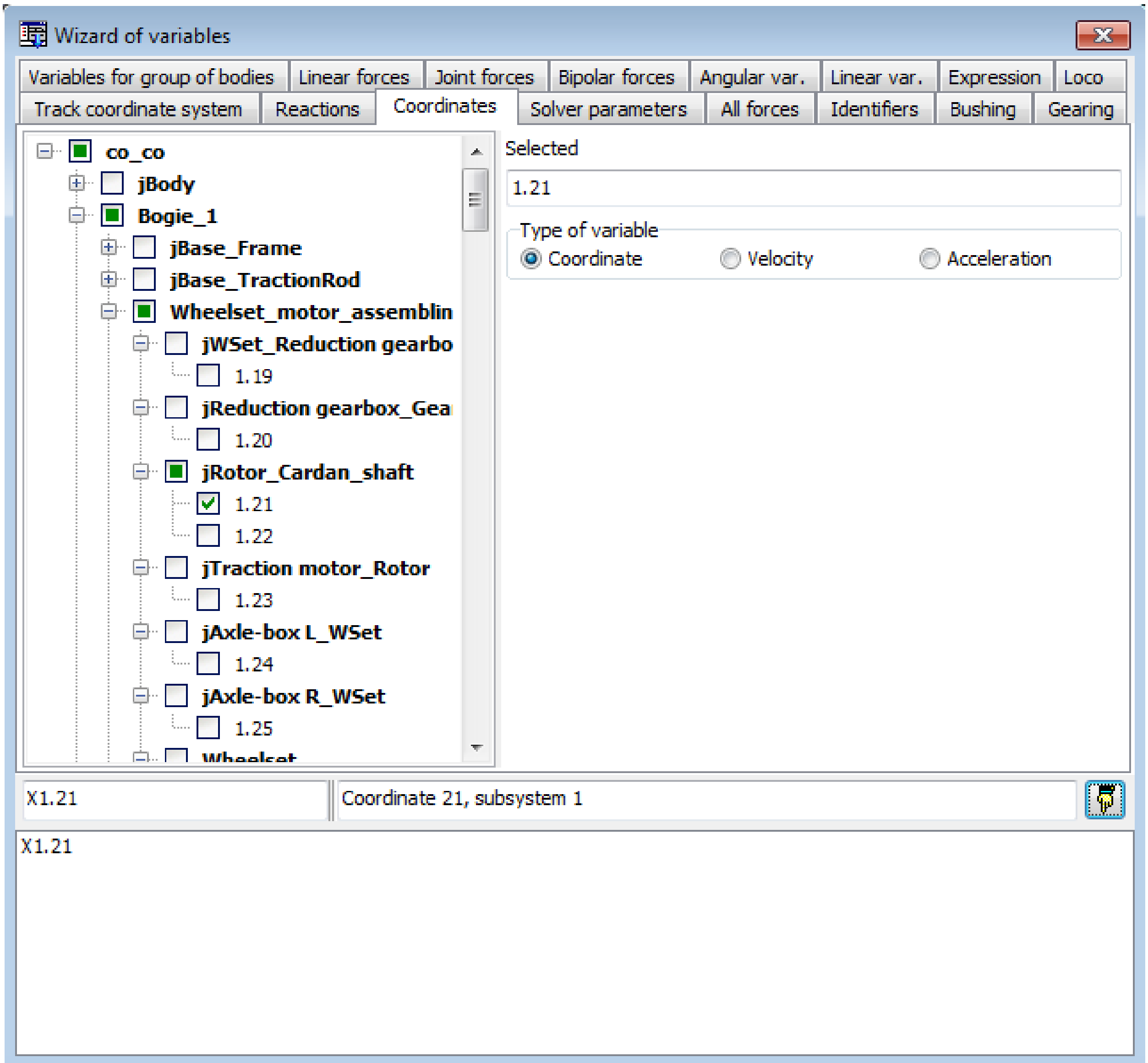


Figure 4.9. Coordinates

4.3.2.2. Angular variables

The *Angular var*[iables] tab creates variables, which specify angular orientation and angular motion of one body relative to another one: projections or module of rotation vector, angular velocity and acceleration. Projections of a vector can be resolved in SC of any body. The element tree for the tab contains bodies.

Figure 4.10 corresponds to the following variable: vector of the angular acceleration of the *Bogie_1.Wheelset_motor_assembling_1.Rotor* body relative to the SC0 (absolute motion). The variable should be transferred into an animation window only.

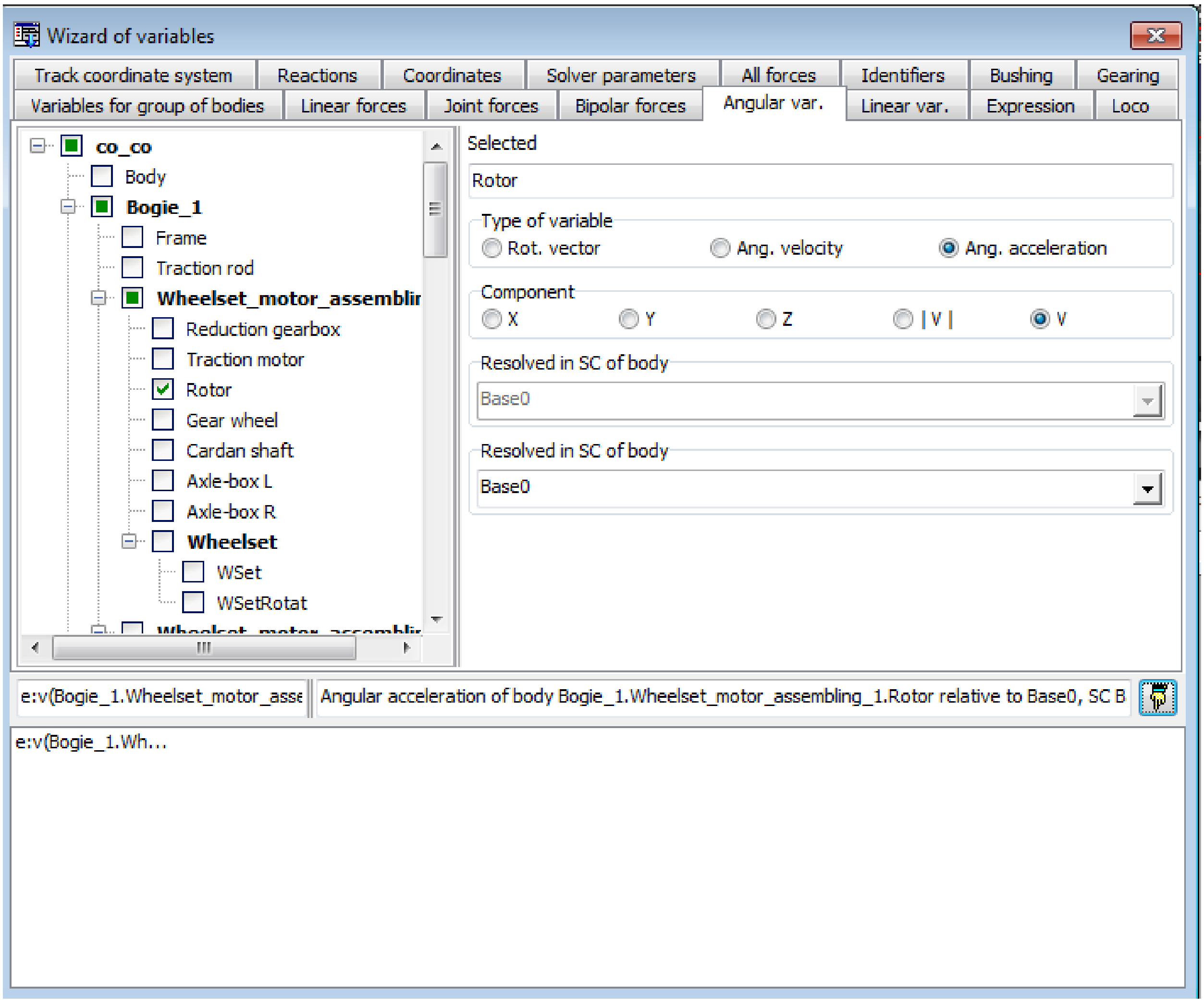


Figure 4.10. Angular variables

The **Use orientation for zero coordinates** key (not shown in Figure 4.10) is applied for the *Rotation vector* variable only. **Use orientation for zero coordinates key** means that the variable is computed for rotations of bodies relative to their positions at zero values of object coordinates. If the key is off, the variable is computed for positions of body-fixed SC.

Remark. The rotation vector is computed according to the theorem on finite rotation. It is equal to the rotation vector multiplied by the rotation angle. If one body orientation differs slightly from another one, the vector components determine small turning angles of the first body around the corresponding axes of the second one.

4.3.2.3. Linear variables

The *Linear var*[iables] allows creating variables, which characterize 'translational' position and motion of one body relative to another one: coordinates (r), trajectory (vector *r*), velocity (v), acceleration (a) of a first body point relative to SC of the second body as well as so called bipolar variables: bipolar vector (r12), velocity (v12) and acceleration (a12) of a pair of points. Projections of a vector can be resolved in SC of any body.

Figure 4.12 illustrates the 'bipolar vector', 'bipolar velocity' and 'bipolar acceleration' notions. Consider a pair of bodies and two points A (for the reference body) and B (for the analyzed body). Let $\mathbf{e}_{12}$ be a unit vector directed from A to B. The bipolar vector between the pair of points is by definition

$$\mathbf{r}_{12} = r\mathbf{e}_{12},$$

where r is the distance between the points. The bipolar velocity v_{12} and acceleration a_{12} are the following vectors:

$$\mathbf{v}_{12} = \dot{r}\mathbf{e}_{12},$$

$$\mathbf{a}_{12} = \ddot{r}\mathbf{e}_{12}.$$

The values of bipolar velocity and acceleration $\dot{r}, \ddot{r}$ characterize changes the distance.

Figure 4.11 defines the following variable: projection of the bipolar vector on the Y axis of SC of the *Frame*. The vector connects the point (5.3, -1, -2.18) of the *Frame* with the point (0.1, –1, -0.18) of the *Traction Rod*. The variable can be transferred into a graphical window or added to a variable list.

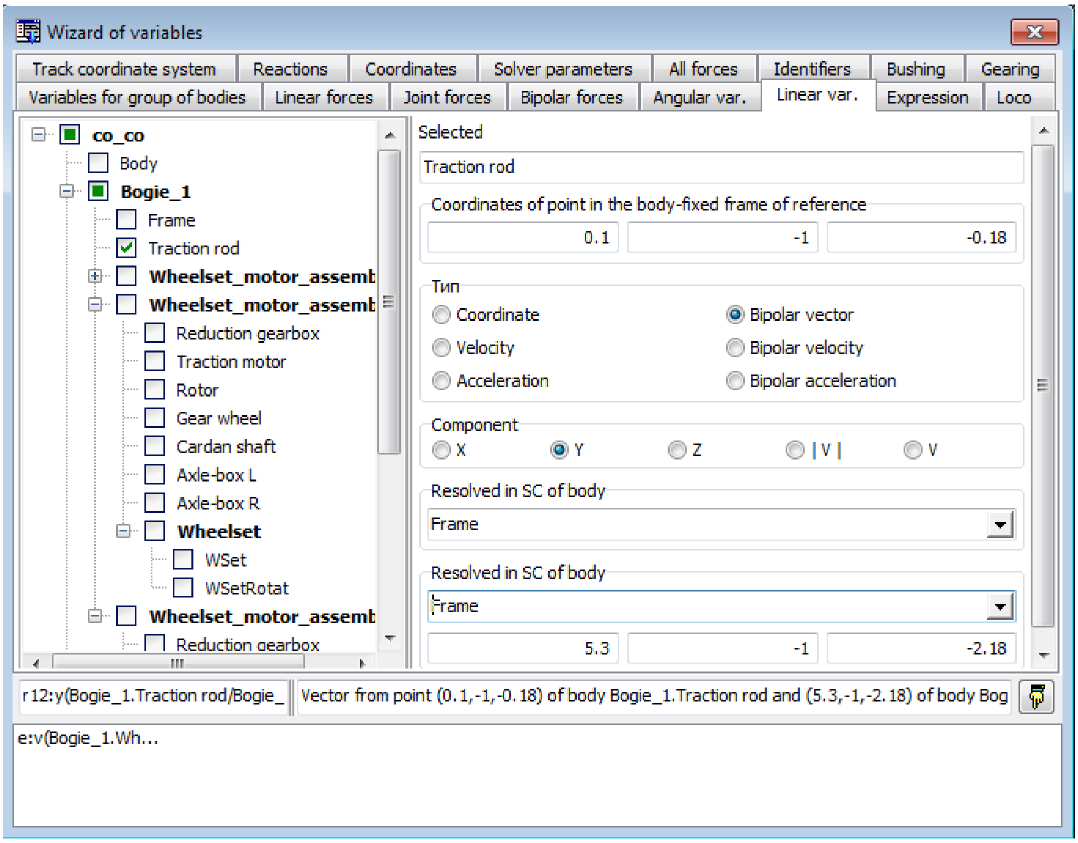


Figure 4.11. Linear variables

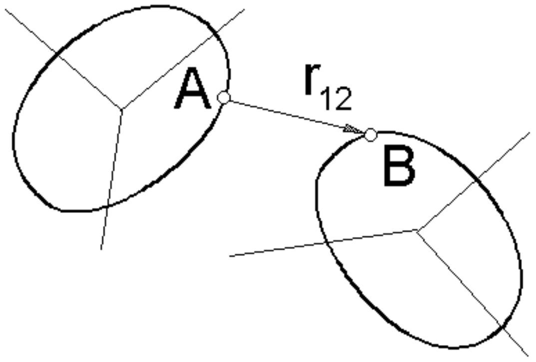


Figure 4.12. Bipolar vector

Remarks

To get a variable for drawing a trajectory of a body point in an animation window, select a body in the tree, specify the point coordinates in SC of the body, set the *r* variable type (coordinates), and select the V (vector) *Component*.

The module (|V|) of a bipolar velocity or acceleration specifies the scalar values $\dot{r}, \ddot{r}$ rather than the length of vector.

4.3.2.4. Variables created by the user in UM Input module

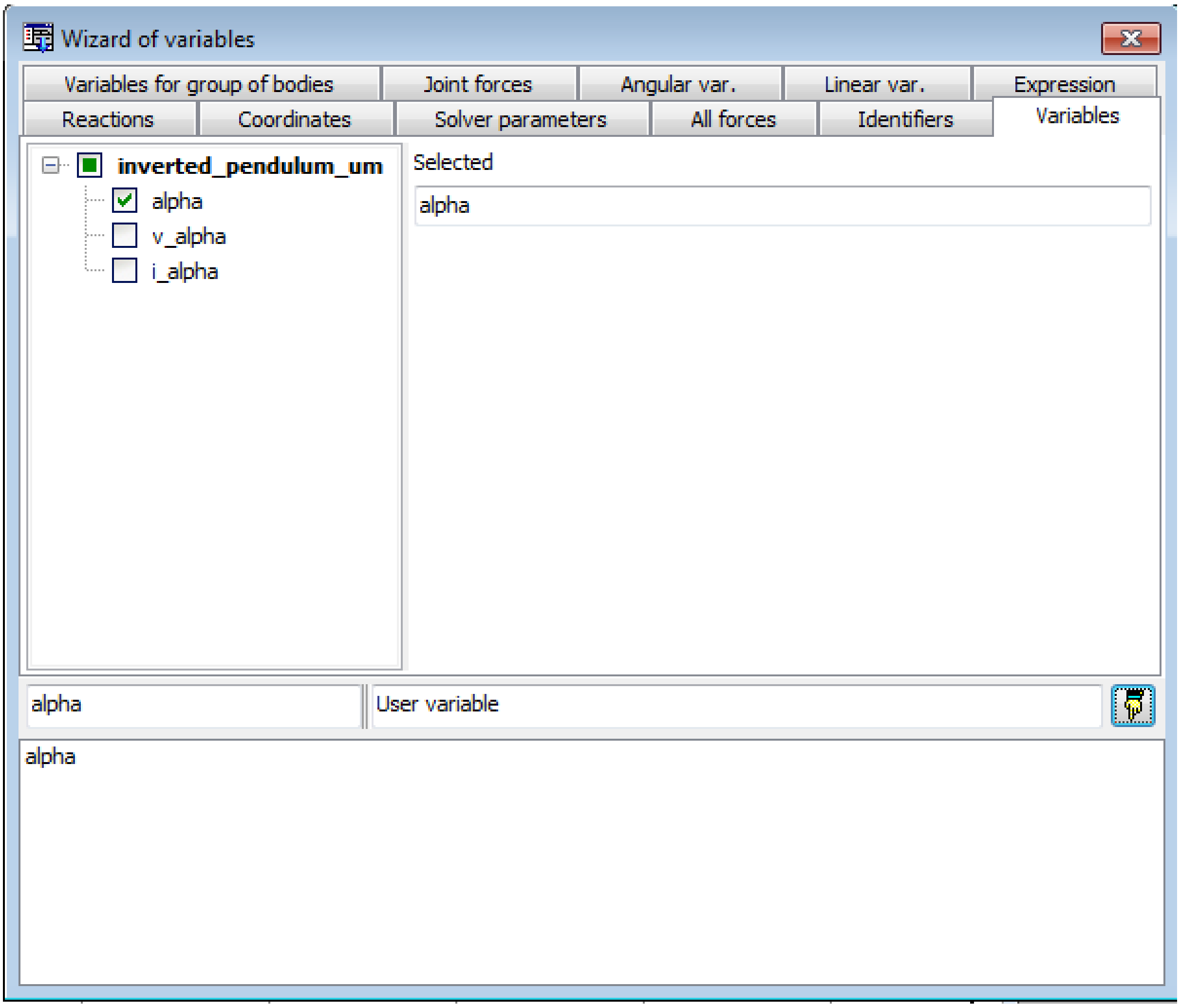


Figure 4.13. List of variables from UM Input

The user may create a computable variable with the help of a list of variables developed for the current model in UM Input, Chapter 3, Sec. Data types | List of variables.

The list of variables from UM Input is located on the **Variables** tab of the wizard, Figure 4.13.

Models containing lists of variables:

{UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_functions\Inverted_pendulum_UM;

{UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_functions\Euler angles;

{UM Data}\SAMPLES\LIBRARY\Variables_and_Kinematic_functions\Yaw Pitch Roll.

4.3.2.5. Sensors

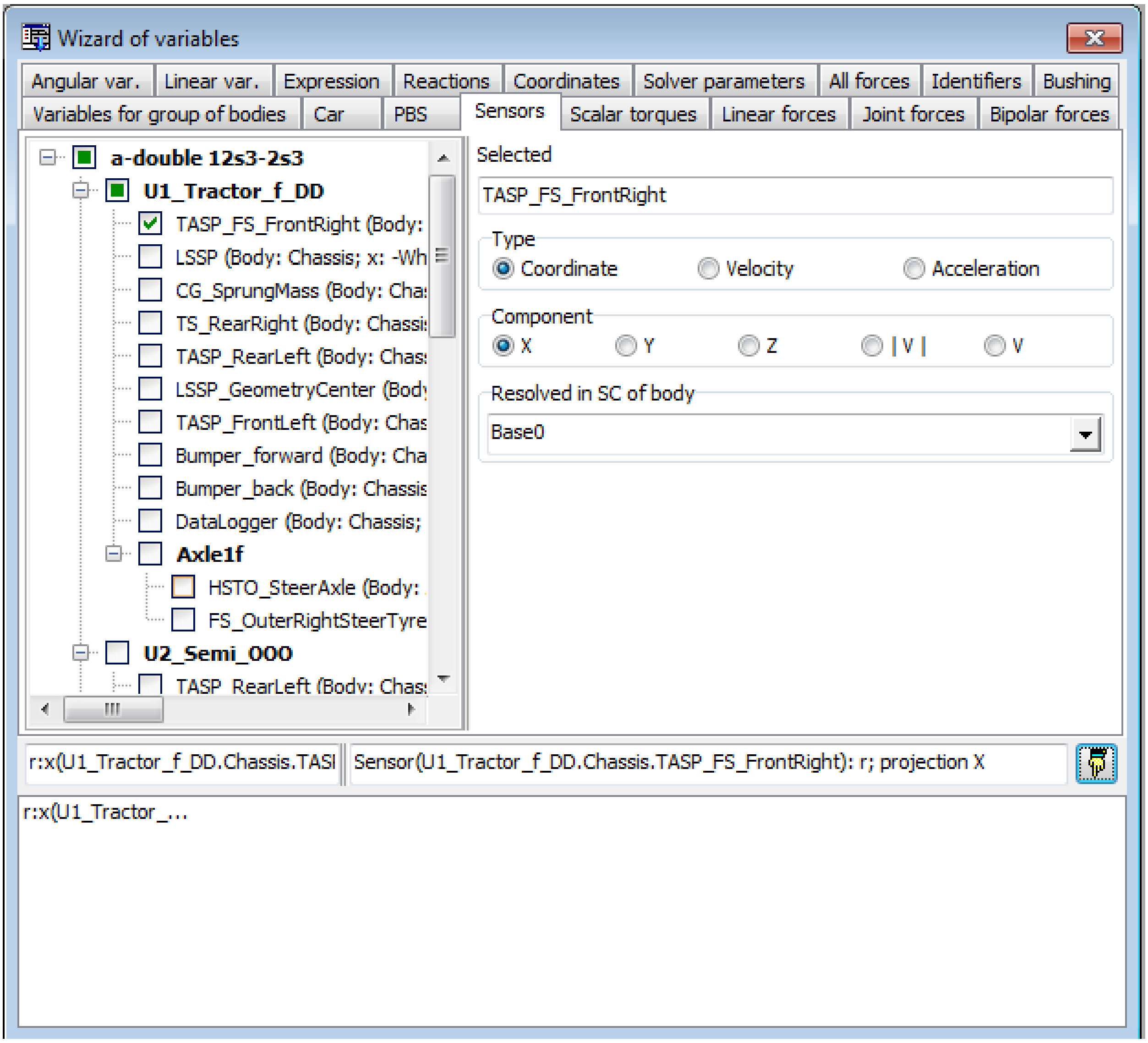


Figure 4.14. Sensors

This type of variables is available if the user has created a set of sensors for the current object in the input module, see Chapter 3, Sect. *“Sensors/LSC” tab*.

The sensor variable allows getting a coordinate, velocity or an acceleration of a body point like the Linear variable, Sect. 4.3.2.3. The only difference consists in the fact that the sensor co-ordinates in the body-fixed SC can be **parameterized** by expressions. Therefore, the coordinates will change if the corresponding identifiers are changed.

The sensor body is entered in the Body box, Figure 4.14. Coordinates of the sensor in SC of the body are shown next to its name.

The list of sensor is shown in the right part of the wizard.

4.3.2.6. Forces: general, bipolar, joint, linear, special, reactive

Creation of variables for the listed force types (Figure 4.8) is quite similar. In addition to the standard data (element, component, type (force of moment), reference SC), the *Acts on body* parameter should be specified, Figure 4.15.

If the object does not contain any force element of a definite type, the corresponding tab is invisible.

Consider some features variable descriptions.

4.3.2.6.1. T-force

The variable corresponds to T-force (Figure 4.15). Type of the variable: force or moment.

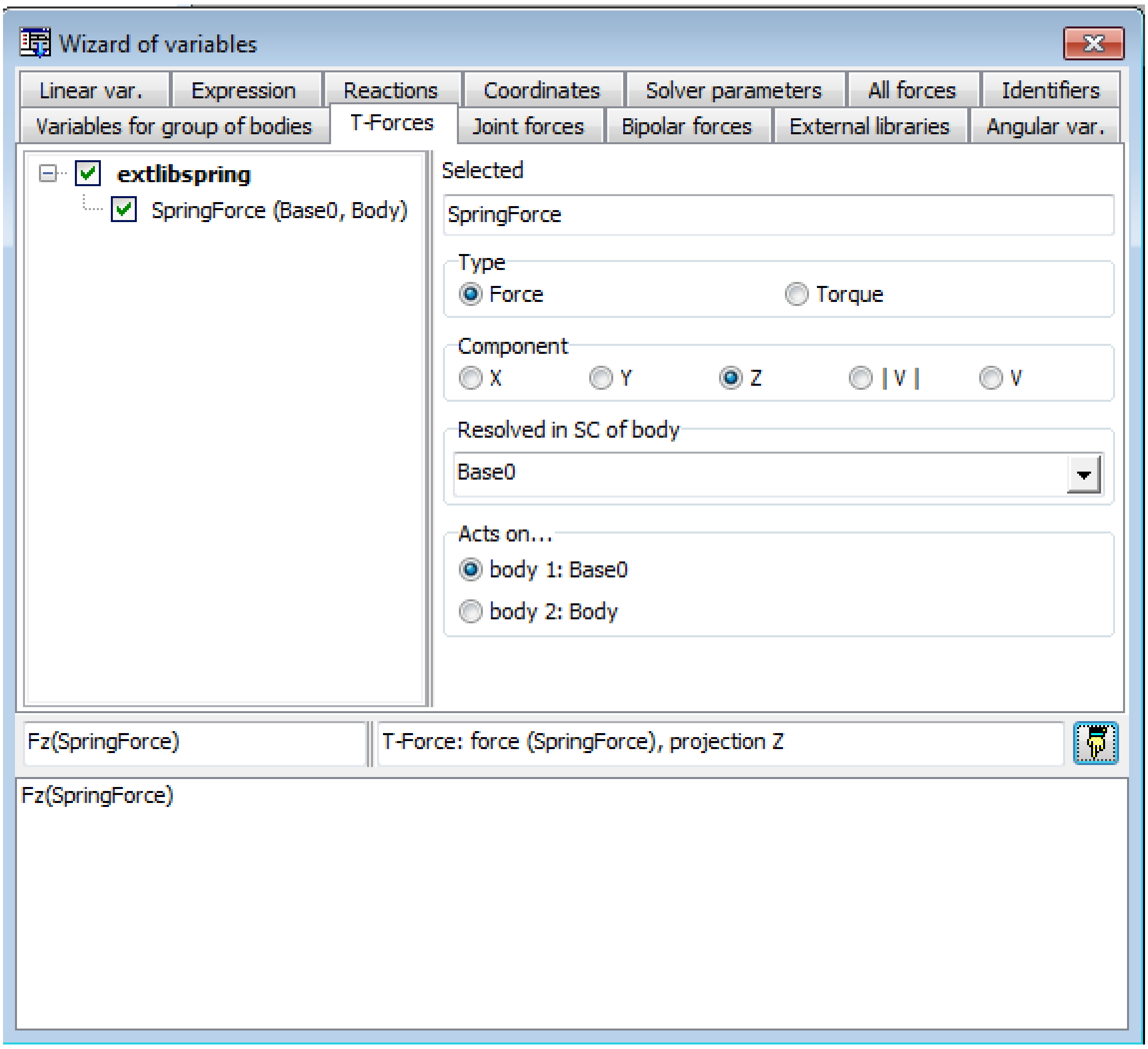


Figure 4.15. Parameters of T-force variable

4.3.2.6.2. Bipolar force

The variable corresponds to bipolar force elements added to the object. Some features of the variable, Figure 4.16.

- The **Force Magnitude** *Component* parameter specifies the scalar force value (signed) rather than its module.
- The **Length** is the length of the bipolar element. In particular, this variable is used for getting plots force versus length.
- The **Velocity** is the time derivative of the length. In particular, this variable is used for getting plots force versus velocity.

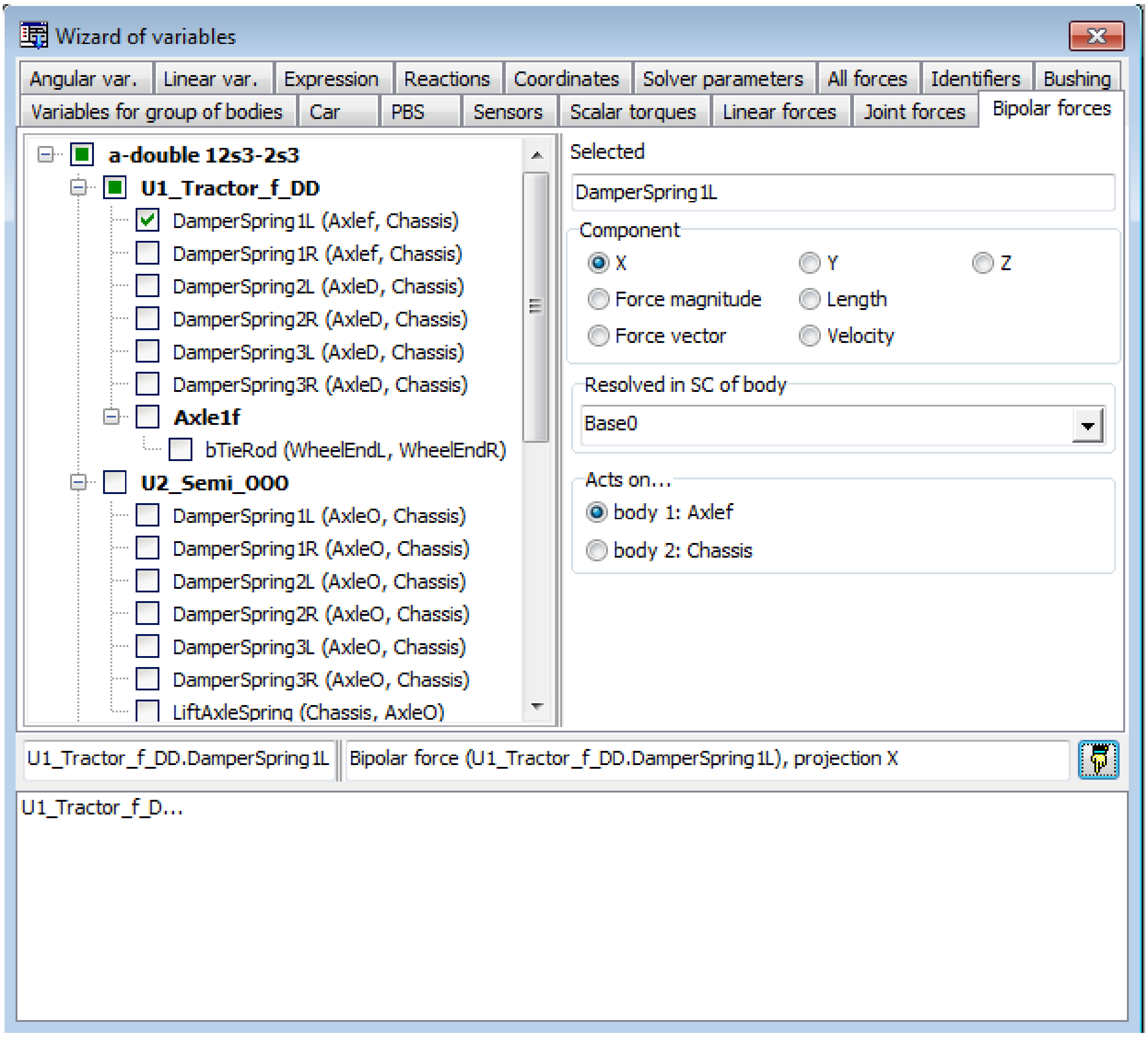


Figure 4.16. Bipolar force

4.3.2.6.3. Scalar torque

The 'Scalar torque' variable is quite similar to the bipolar force, Figure 4.16. The only difference consists in variables **Angle** and **Velocity**, which denotes angle of rotation and the corresponding angular velocity, which are used for computation of the torque.

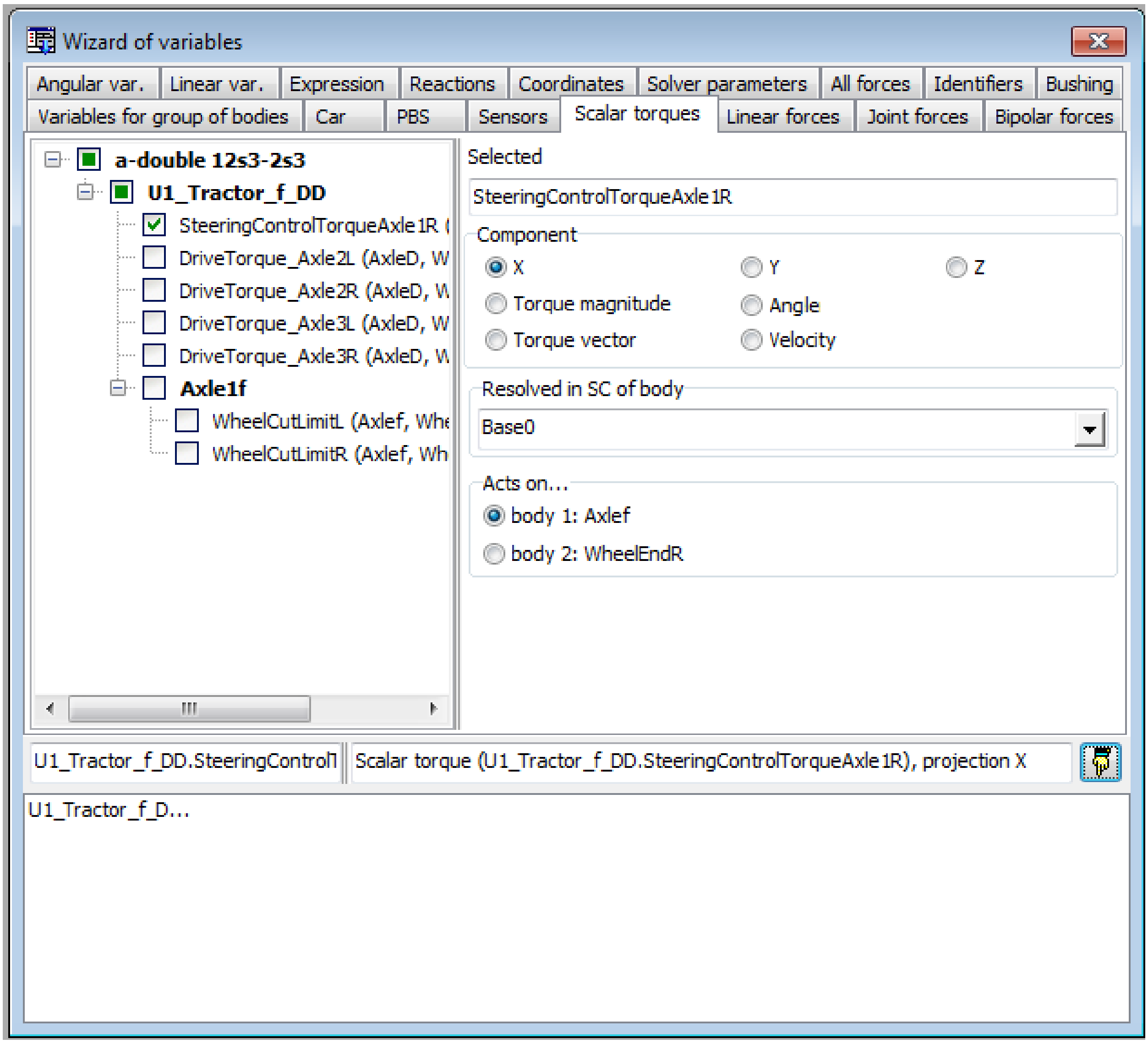


Figure 4.17. Scalar torques

4.3.2.6.4. Joint force

This type of variable corresponds to applied joint forces and torque (rotational, translational joints, joints of generalized type). The element tree presents the list of joint for the current object. The variable is zero for all joint types except for rotational, translational joints and joints of generalized type.

If description of a joint of the generalized type includes several forces and torques (corresponding to different joint coordinates), the variable forms a principal vector reduced to the joint point.

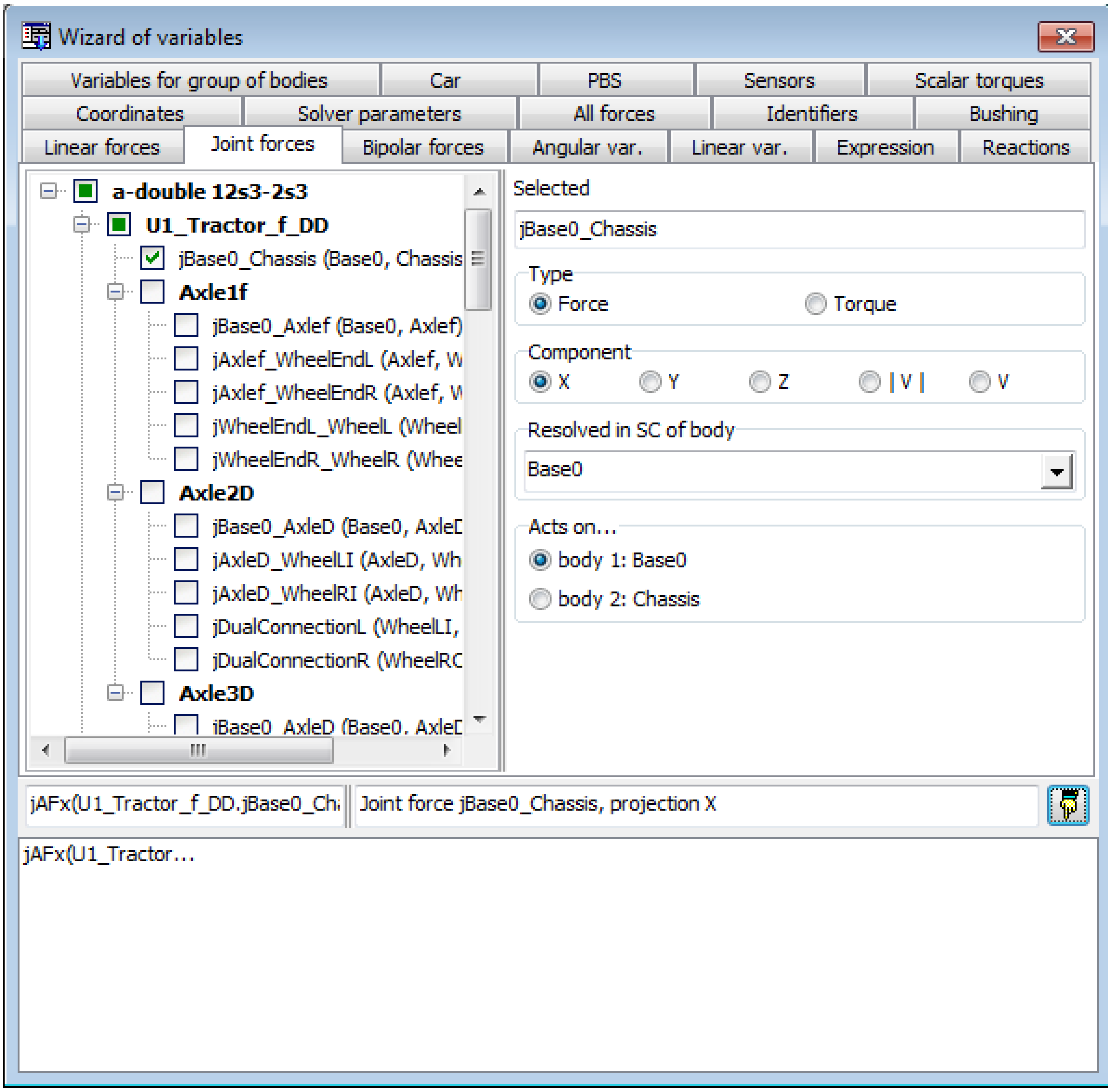


Figure 4.18. Joint forces

4.3.2.6.5. Linear force

Linear force variable corresponds to generalized linear force elements.

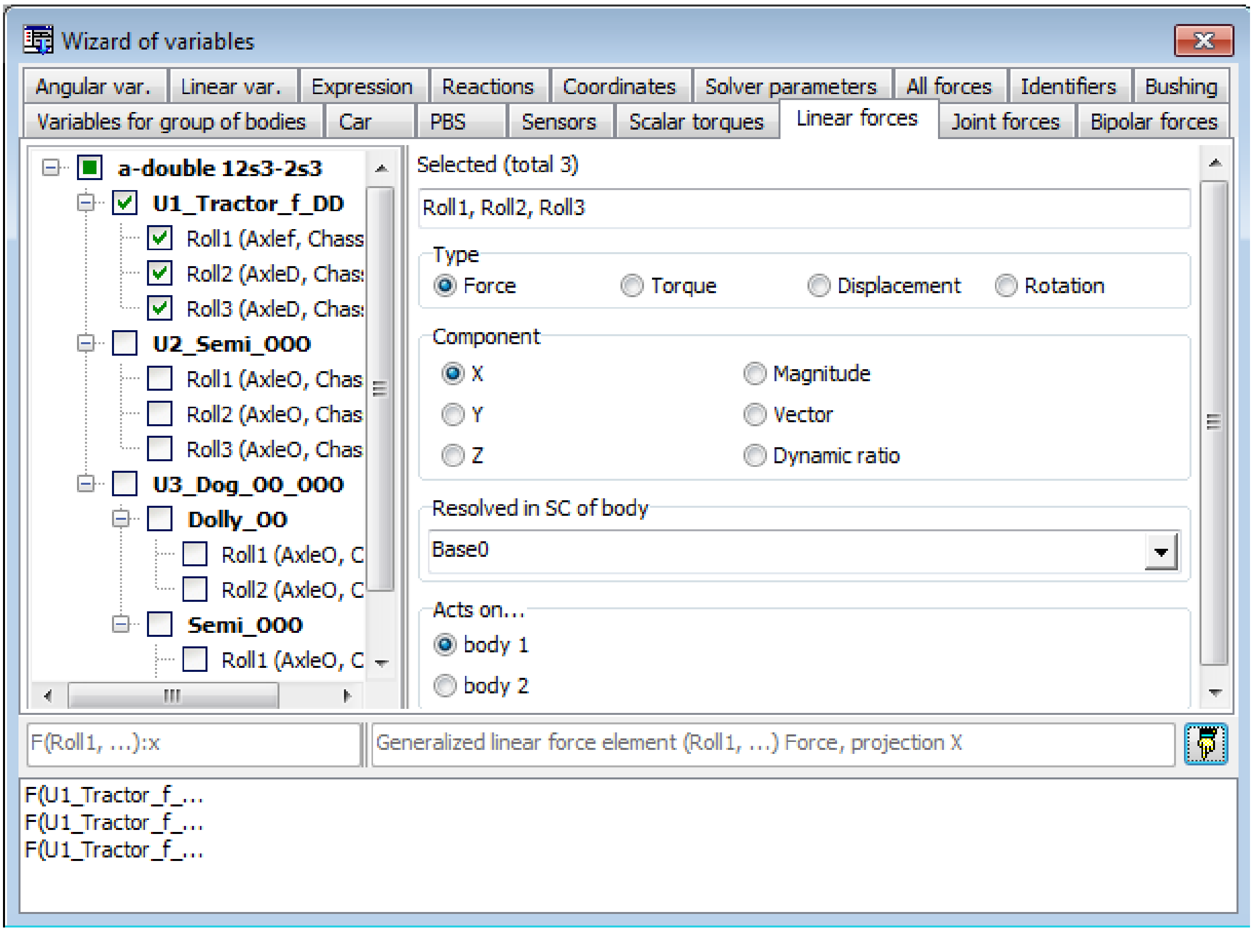


Figure 4.19. Linear force

The *Analyzed variable* group allows choice of the variable type:

- *Force*,
- *Torque*,
- *Displacement* – displacement (for elastic elements) or V – velocity (for dissipative elements),
- *Rotation* – rotation (for elastic elements) or *Omega* – angular velocity (for dissipative elements).

The last two types correspond to relative motions of interacting bodies, which are used in computation of the force and the moment according to Chapter 2, Sect. *Generalized linear force element*.

The *Component* parameter has an additional item Dynamic ratio, which is related to the Force type of variable only. The dynamic ratio is computed by the following assumption: description of the corresponding linear *elastic* force element includes a nonzero stationary value with a single nonzero component. The dynamic response is equal to the difference between the nonzero component of the stationary force and the corresponding dynamic value, divided by the stationary value. For example, if the Z-component F_{z0} is nonzero,

$$k_D = \frac{F_z - F_{z0}}{F_{z0}}.$$

4.3.2.6.6. Special forces

The variable corresponds to one of the special force elements.

Consider all types of special force elements and features of the corresponding variables

4.3.2.6.6.1. Gearing, rack and pinion

For a ‘gearing’ or ‘rack and pinion’ force element, the force is computed, and there is no torque presented in the wizard, Figure 4.20.

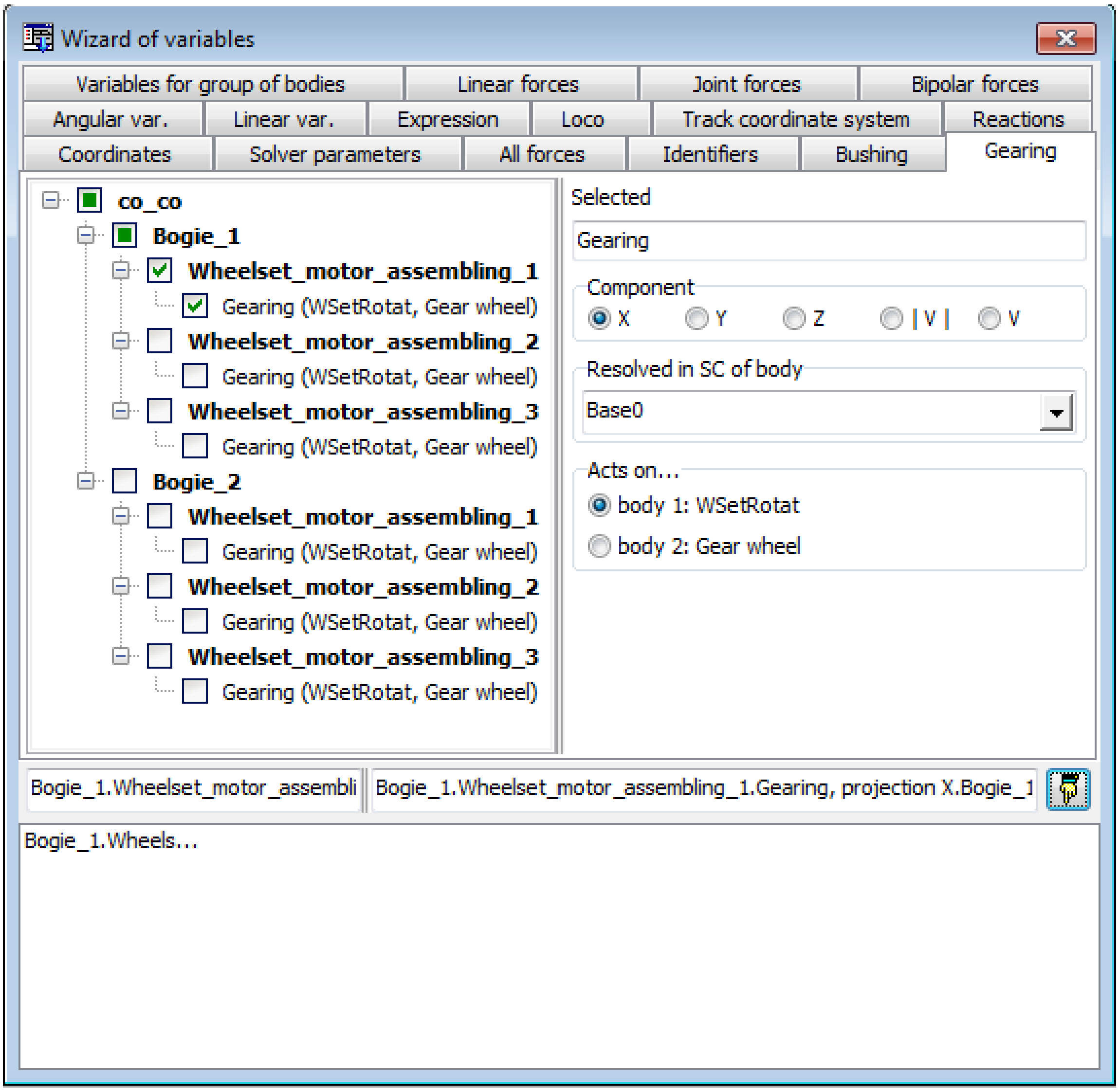


Figure 4.20. Master of variables for ‘Gearing’ force element

4.3.2.6.6.2. Bushing

The tab of the master has the standard interface like in Figure 4.15.

4.3.2.6.6.3. Cam

Normal and frictions forces are available as variables in the case of a cam element.

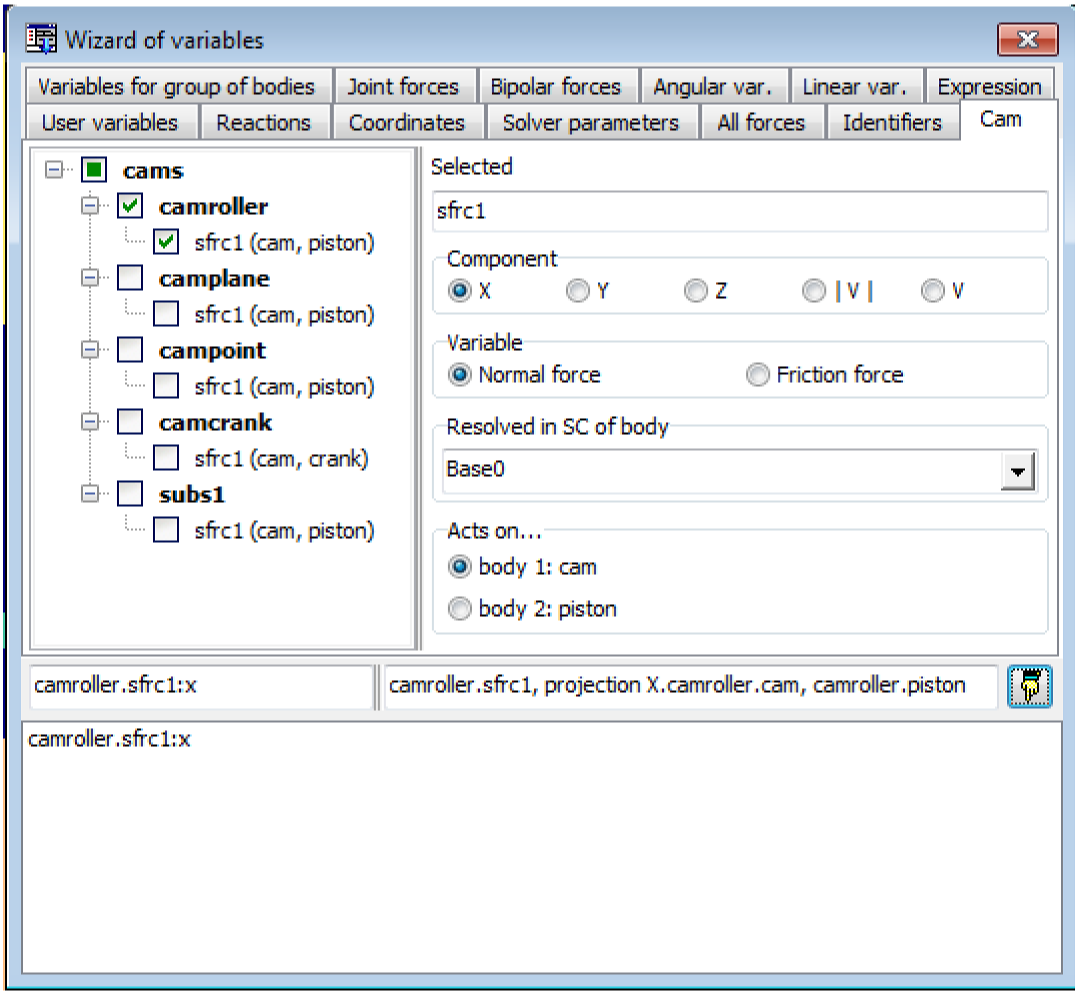


Figure 4.21. Master of variables for Cam force element

4.3.2.6.6.4. Spring

In the case of a ‘Spring’ special force element an additional variable the ‘dynamic ratio’ is introduced, Figure 4.22, see Sect. 4.3.2.6.5.

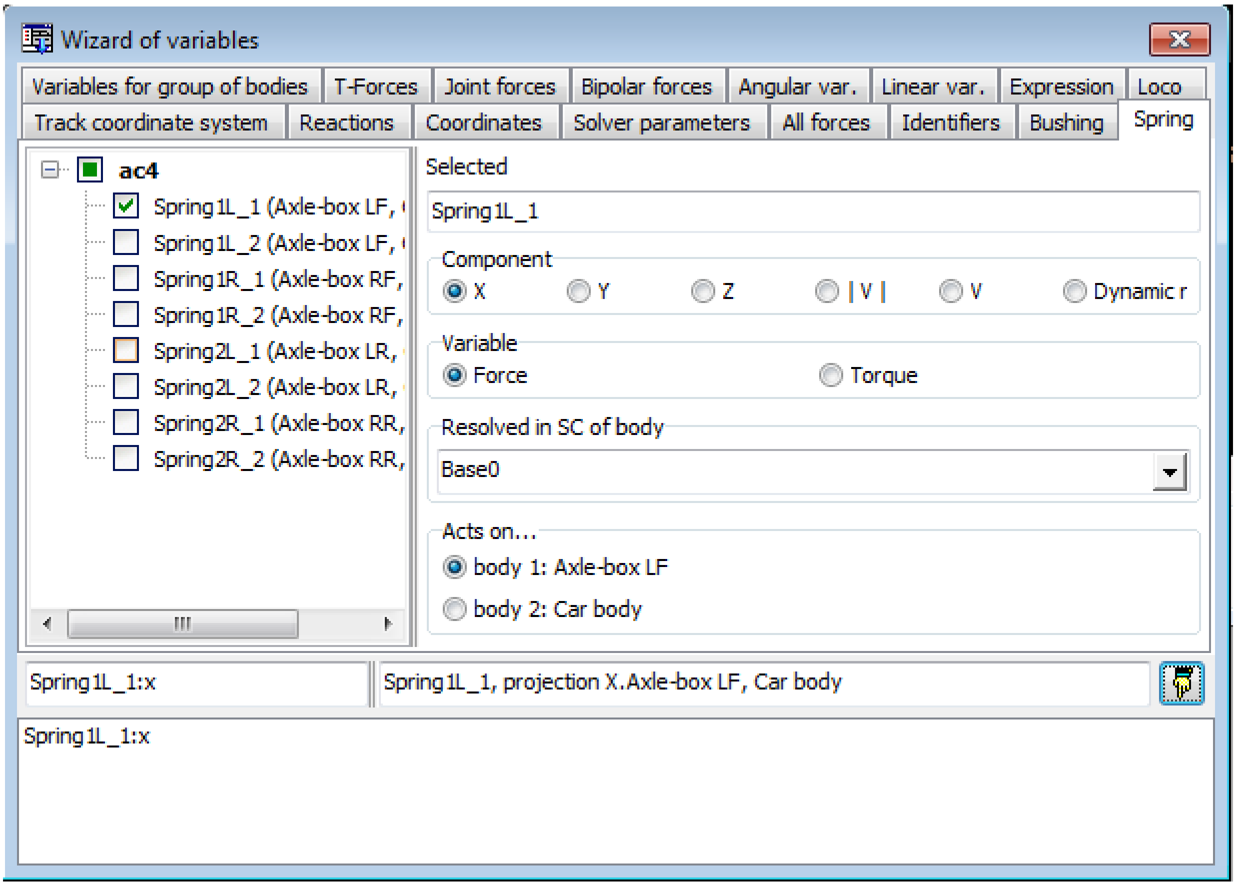


Figure 4.22. Master of variables for a spring force element

4.3.2.6.6.5. Combined friction

For a **combined friction** the variable corresponds to the axle force or to the friction force.

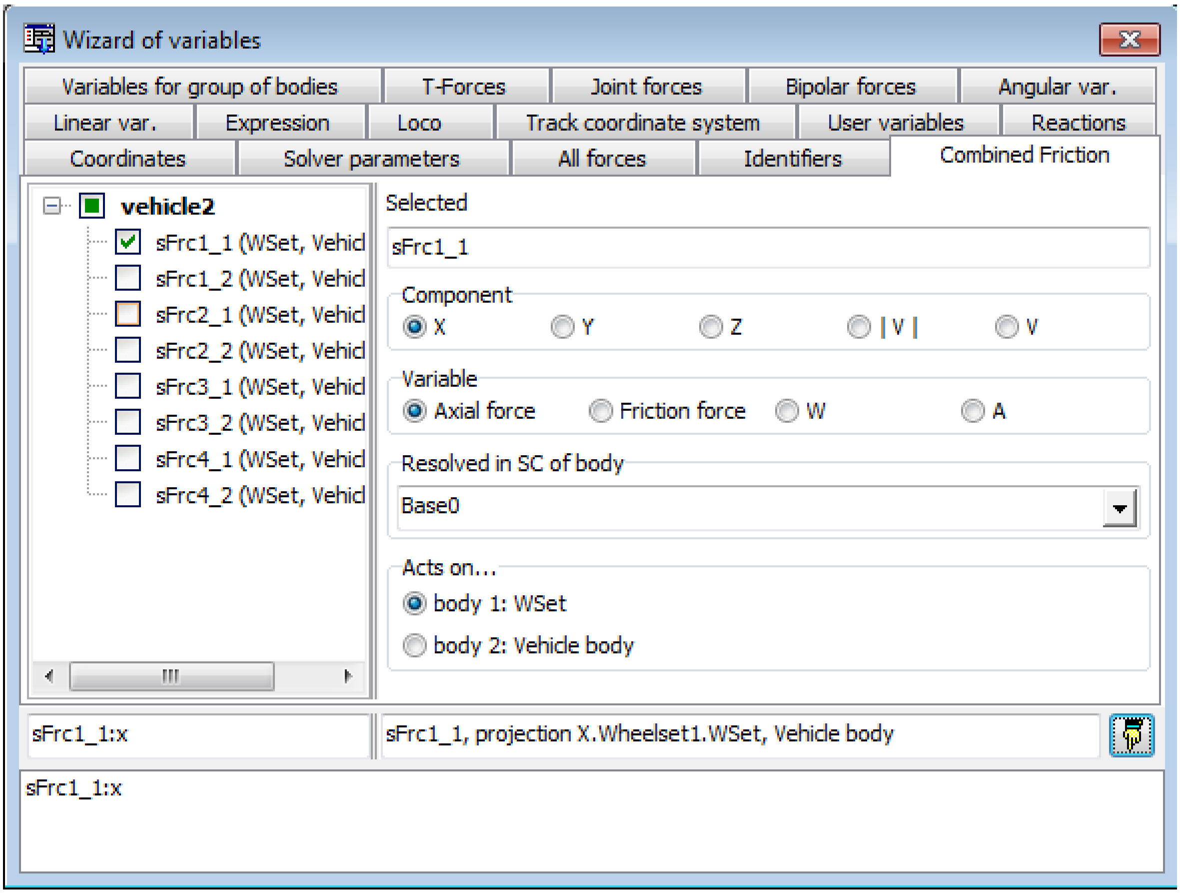


Figure 4.23. Master of variables for a combined friction force element

4.3.2.6.6.6. Tire

For a tire force element, the wizard formed the same variables as in the list of special variables of a road vehicle, Figure 4.24, see Sect. 4.3.2.16.

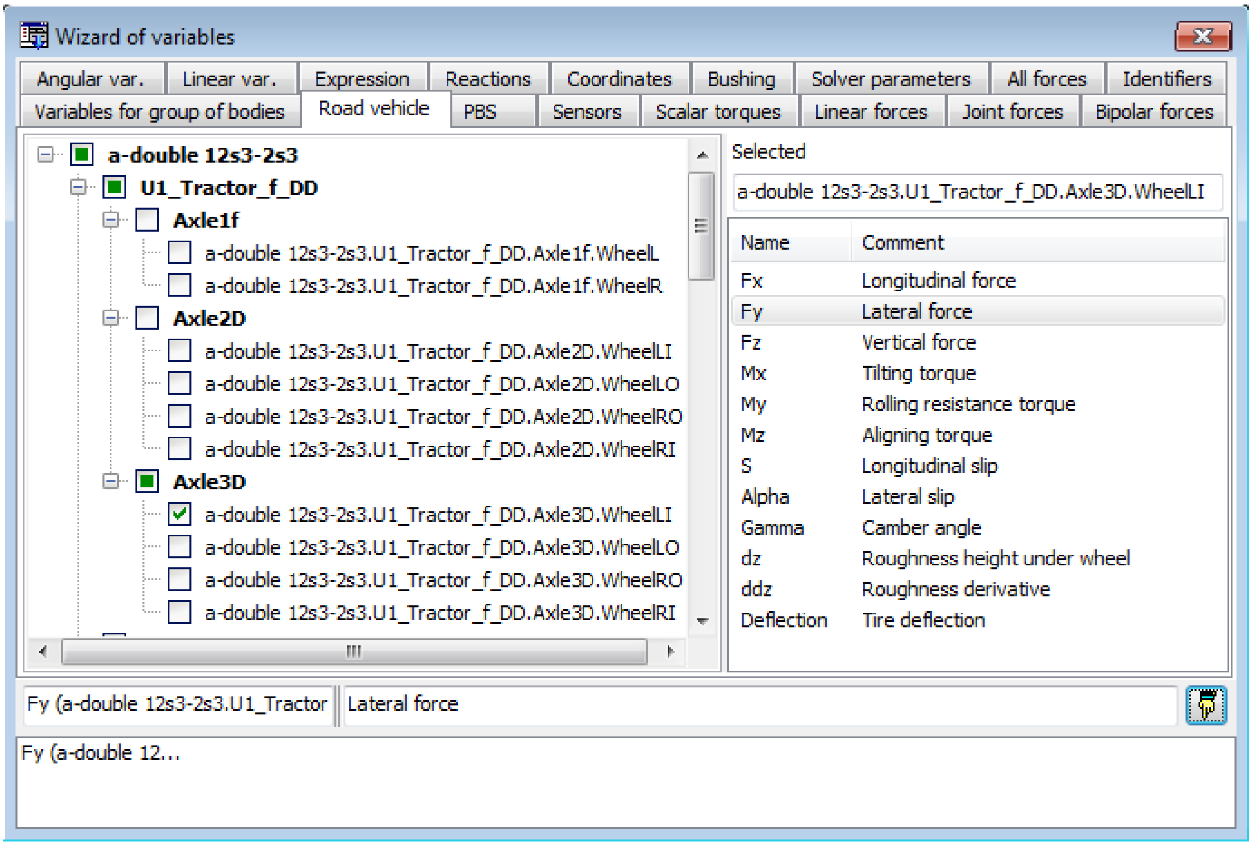


Figure 4.24. Master of variables for a tire force element

4.3.2.6.7. Reaction force

The variable corresponds to reaction forces in object joints reduced to joint points.

Remarks

If a joint has 6 d.o.f. (or 3 for a plane mechanism), the reaction must be zero. De-facto, the value differs from zero due to integration errors. This value can be used for evaluation of errors in calculation of reaction forces and moments, and it decreases with the growth of the integration accuracy.

Calculation of the reaction forces is correct for statically determinate objects only, e.g. for objects without closed kinematical loops. It is incorrect for the over-constrained systems.

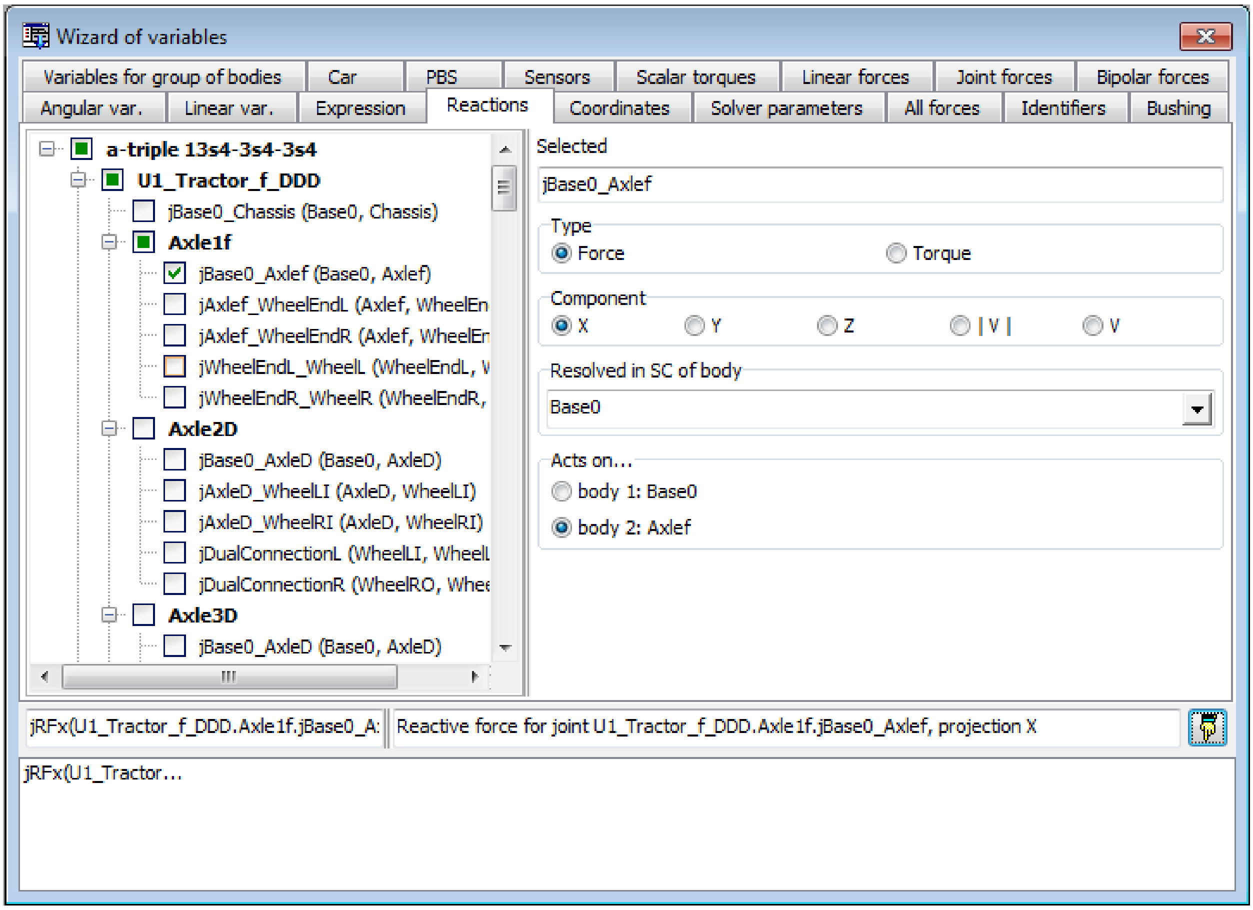


Figure 4.25. Reactions

4.3.2.7. All forces

The variable is intended for animation of all forces acting on a separate body or on a group of bodies.

Check a body or a group of bodies in the element list, specify force types (applied, reaction, inertia forces). Inertia forces are reduced to the centers of mass of bodies.

Checking the *Show internal applied forces* and *Show internal reactions* boxes are not supported for the current UM version.

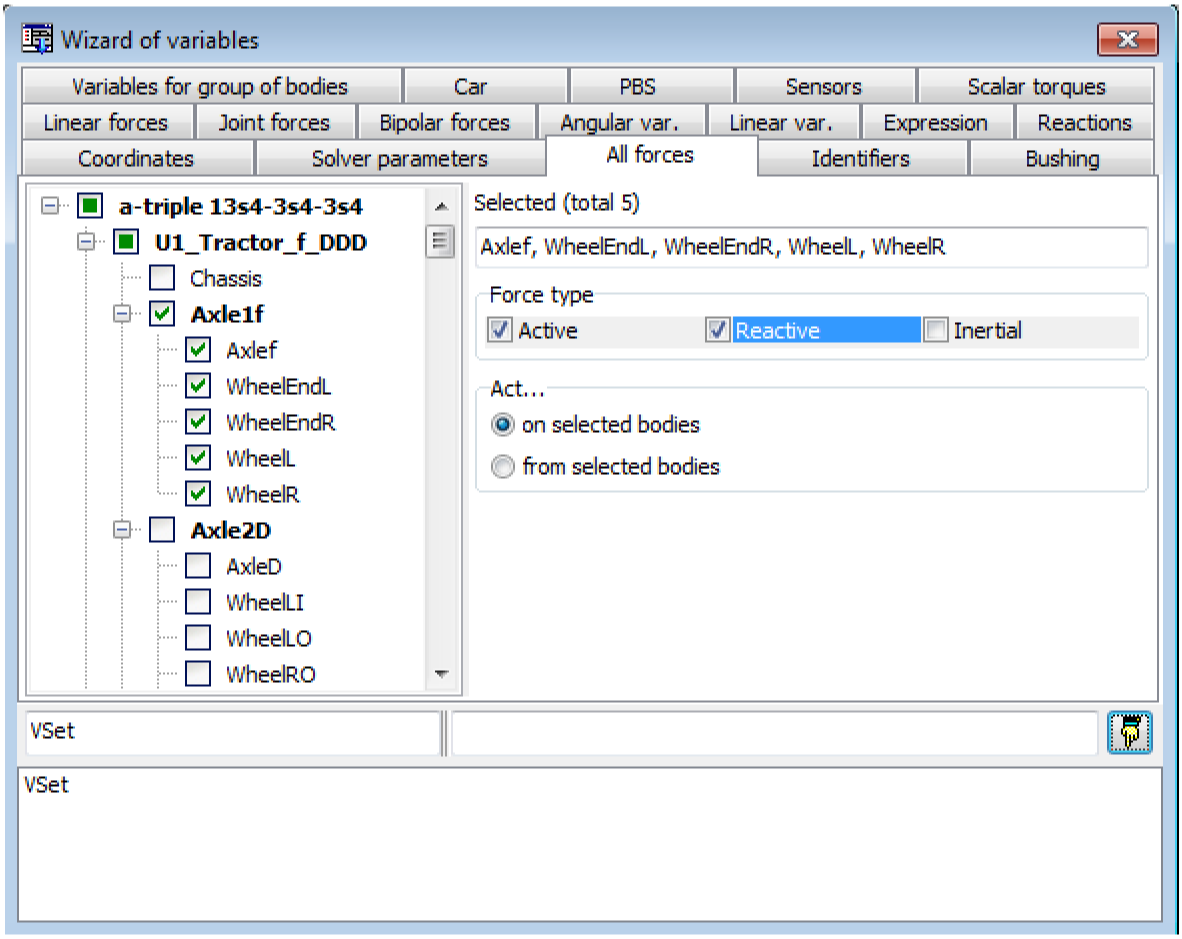


Figure 4.26. The All forces tab

Figure 4.26 displays the following variable: all applied and reaction forces acting on the *Axlef, WheelEndL, WheelEndR, WheelL, WheelR*.

4.3.2.8. Contact forces for bodies

The *Contact* Forces *for bodies* tab is intended for creation of variables, which correspond to contact force elements.

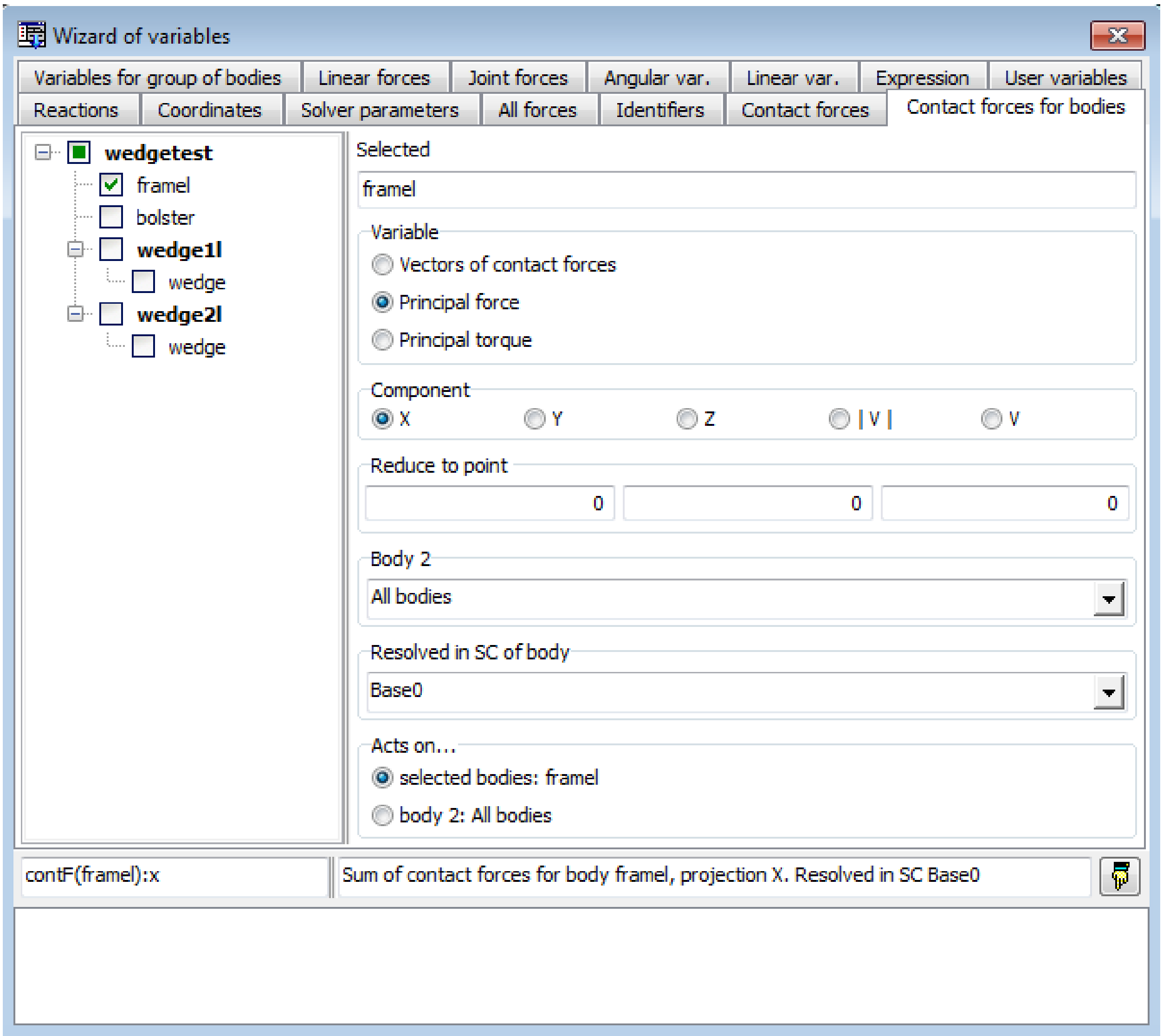


Figure 4.27. *Bodies* tab

All Points-Plane forces are considered, which act on the selected body. The variable can be one of three types:

Vectors of contact forces

The variable can be transferred into an animation window only. It contains a set of all contact forces (normal and friction forces are presented by separate vectors). Forces are applied to the corresponding contact points. An example of contact forces acting from the friction wedges on the side frame and bolster is shown in Figure 4.28. We have created two variables of the above type to obtain all the vectors.

Principal force

The principal force specifies a geometrical sum of all contact forces acting on the selected body including normal and friction forces. The *Component* parameter can be set in the standard manner.

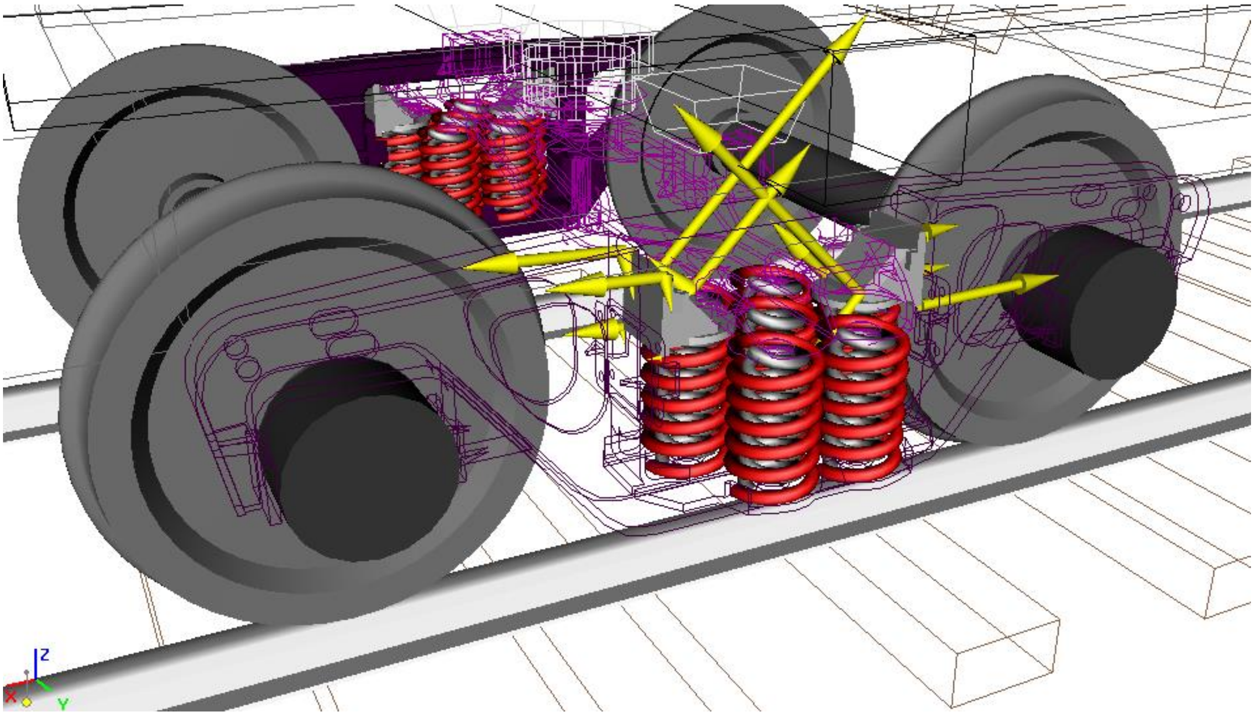


Figure 4.28. Vectors of contact forces

Principal moment

The principal moment specifies a geometrical sum of moments of all contact forces acting of the selected body including normal and friction forces. The moment point is specified by its body-fixed coordinates:

The Component parameter can be set in the standard manner.

Figure 4.29. *Single forces* tab for a multipoint contact presents the following variable: Y-projection (SC *Body*) of the principal force vector for all contact forces acting on the *Body*.

4.3.2.9. Single forces tab: Points-Plane and Points-Z Surface force elements

Principal vector

Total actions of all the contact points of the elements are evaluated. The following variables are specified by the **Type of variable** group:

R is the vector sum of all contact forces including normal and friction forces;

N is the vector sum of normal forces;

Ffr is the vector sum of friction forces;

D is the average depth of penetration of points, a scalar value;

W is the vector of the total power of sliding friction forces, the definition see below;

A is the vector of the total work of sliding friction forces, the definition see below.

For all of the vectors, variables as projections and vector module are available.

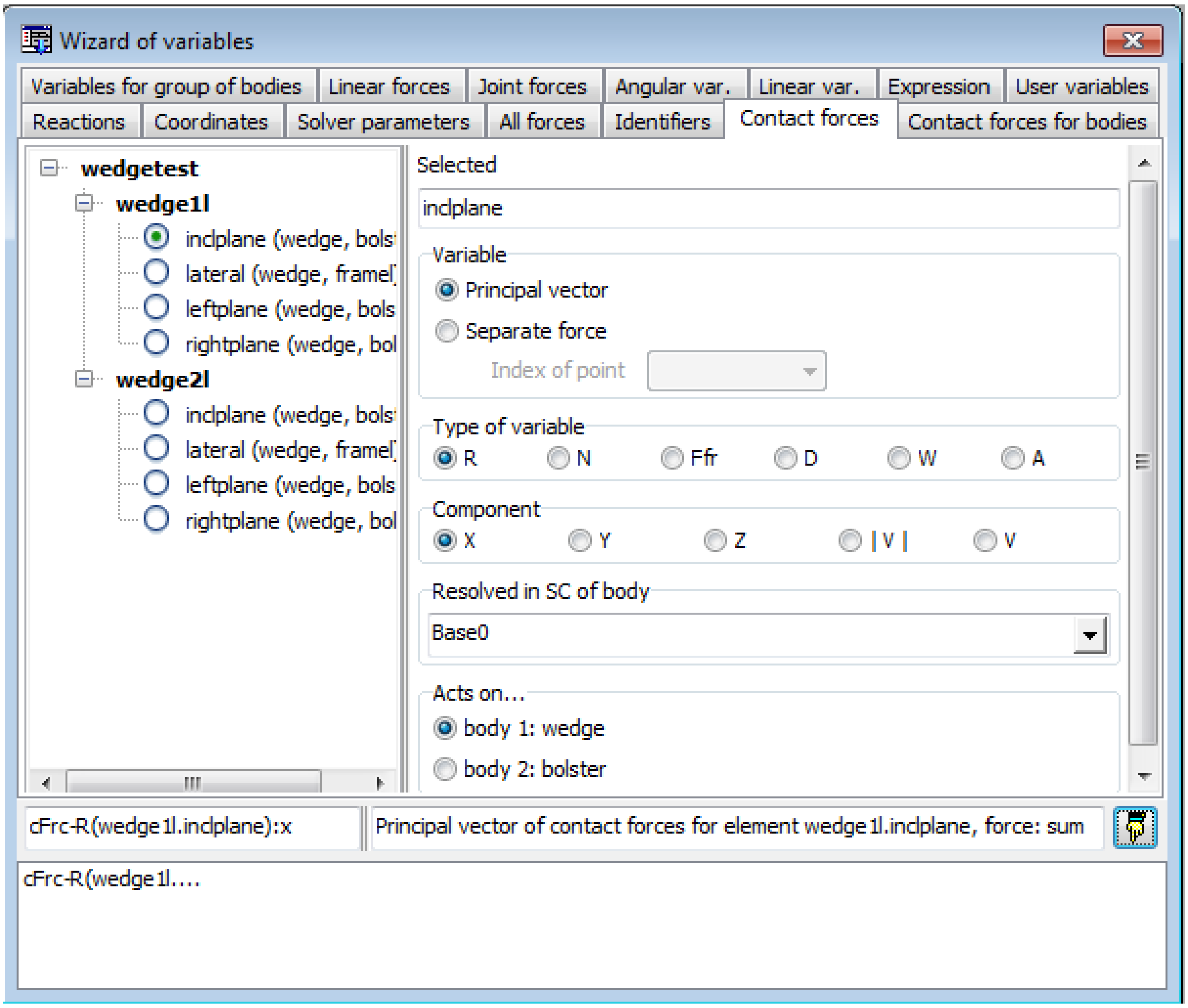


Figure 4.29. *Single forces* tab for a multipoint contact

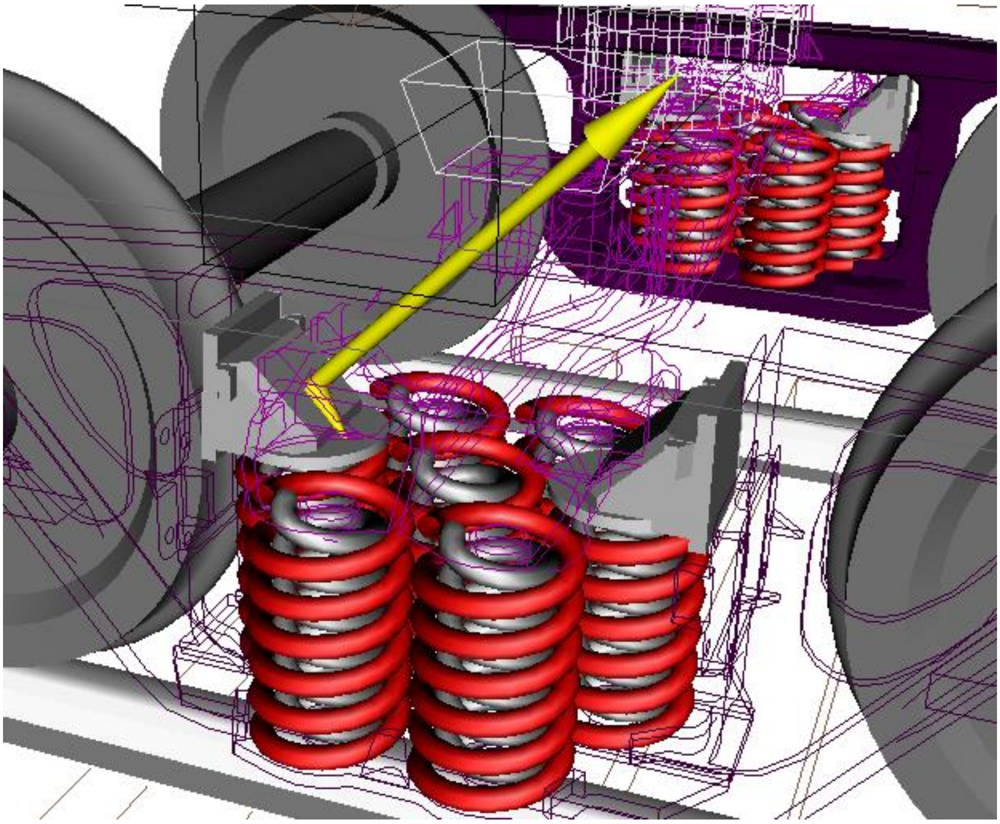


Figure 4.30. Principal vectors of normal and friction forces acting on the bolster from the wedge

Consider variables, which can be created for *Points-Plane and Points-Z surface* contact elements.

If vectors **R, N** are **Ffr** put into an animation window, the begin of the vector is placed in the geometric center of contact points computed in SC of the body according to the formula

$$\rho = \frac{\sum_{i-1}^{N} \rho_i}{N},$$

where ρ_i are the radius vectors of contact point relative to the body-fixed SC origin, N is the number of points. For instance, Figure 4.30 shows the vectors of total forces N, Ffr.

Power vector of sliding friction forces is computed by three projections in SC of the second body (the body with contact plane) according to the relation

$$\vec{W} = \begin{pmatrix} W_x \\ W_y \\ W_z \end{pmatrix} = \begin{pmatrix} \sum F_{ix} v_{ix} \\ \sum F_{iy} v_{iy} \\ \sum F_{iz} v_{iz} \end{pmatrix},$$

where F_{ix}, F_{iz}, F_{iz} are the projections of single force for contact point i of axis of SC of the second body, and v_{ix}, v_{iz}, v_{iz} are the projections of the sliding velocity on the same SC. In the sticking mode, the force power of a friction force is zero.

The power vector is not transforms to other SC.

Projections of the power vector can be interpreted as a part of the total power, which is dissipated in the given direction.

If the component |V| is selected, the total power $W = W_x + W_y + W_z$ is computed.

Vector of work of sliding friction forces is computed by three projections in SC of the second body (the body with contact plane) according to the relation

$$\vec{A} = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} = \begin{pmatrix} \int W_x dt \\ \int W_x dt \\ \int W_z dt \end{pmatrix},$$

The vector of work is not transforms to other SC.

Projections of the vector of work can be interpreted as a part of the total energy, which is lost in the given direction.

If the component |V| is selected, the total work $A = A_x + A_y + A_z = \int W dt$ is computed.

Single force

The user can select contact data corresponding to a single contact point in the **Index of force** list. The following variables are specified for the contact point by the **Type of variable** group:

R is the vector of the sum of normal and friction forces;

N is the vector of normal force;

Ffr is the vector of friction force;

D is the depth of penetration on the contact point into the plane, a scalar value;

W is the vector of power of the sliding force;

A is the vector of work of the sliding force.

The same formulas and comments are valid made above for the principal vectors, if set the number of points equal to N=1 in the formulas.

4.3.2.10. Single forces tab: other types of contact force

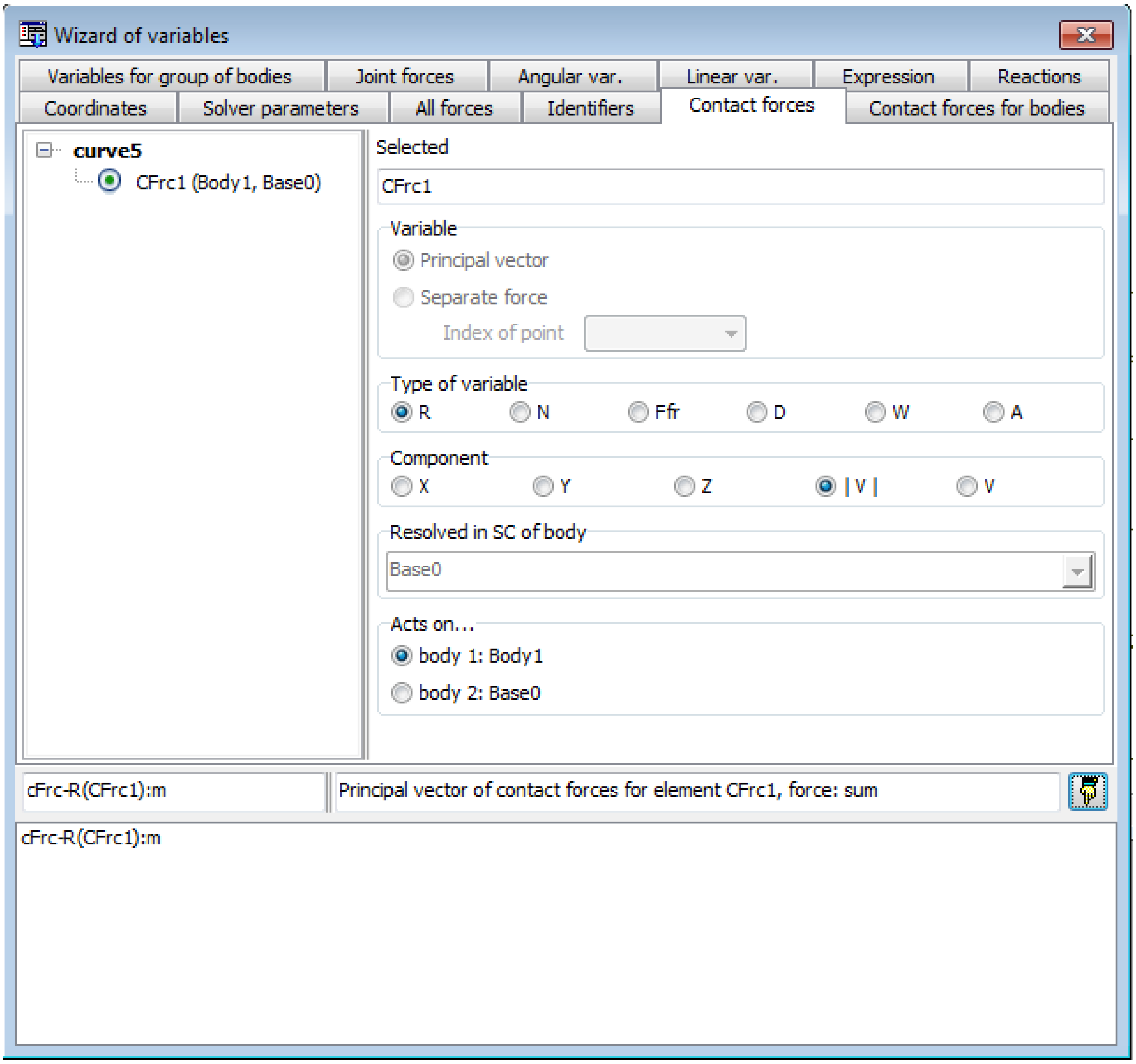


Figure 4.31. Single forces tab

The following variables can be created:
R is the vector of the sum of normal and friction forces;
N is the vector of normal force;
Ffr is the vector of friction force;
D is the depth of penetration in the contact.

Projections and module values can be considered as variables for all of the vectors.

4.3.2.11. Variables defined by user in control file

The **User variables** tab (Figure 4.32) allows the user to create variables, which numeric values are computed in the control file, see Chapter 5 of the user's manual. The variable can be either a scalar or a vector.

A scalar variable value should be assigned to an element of the *UserVars* array [0..1001], the corresponding index should be set as the *Index of array…* parameter in the **User variables** tab.

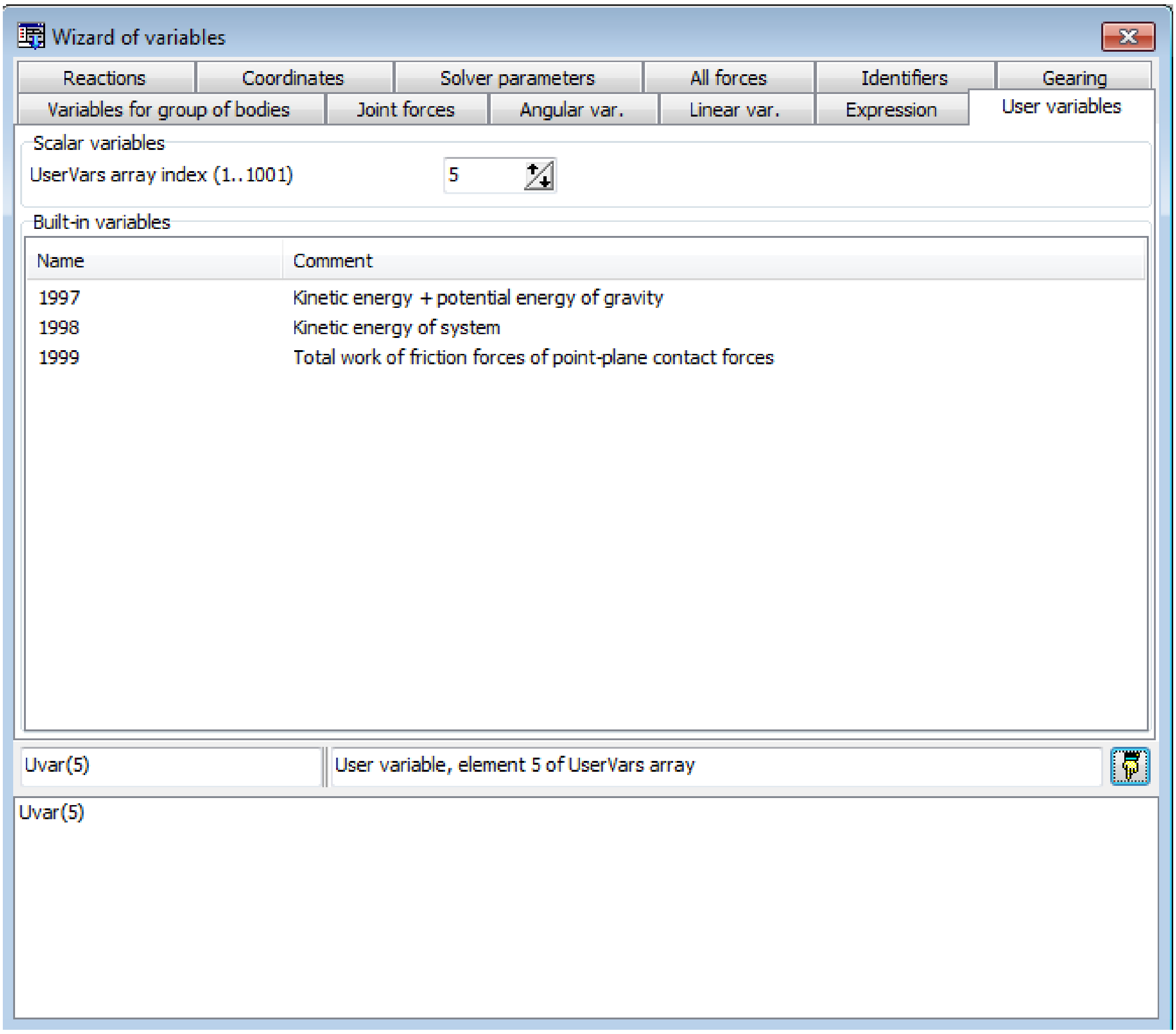


Figure 4.32. Scalar variable defined by the user

4.3.2.12. Vectors defined by user in control file

The **User vectors** tab (Figure 4.33) specifies one of the user-defined vectors (Sect. 5.6.8).

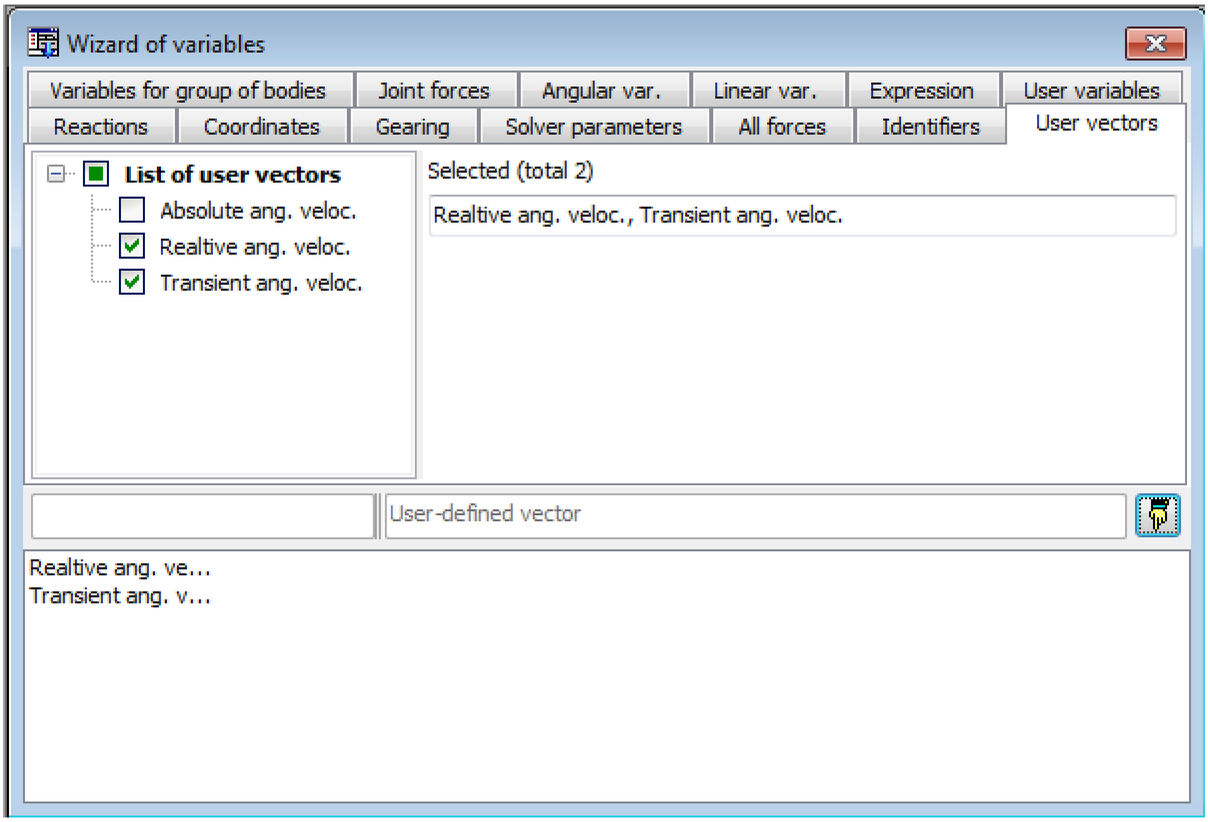


Figure 4.33. Vector defined by the user

4.3.2.13. Identifiers

The Identifier tab (Figure 4.34) creates a scalar variable, which value is identical to the value of an identifier. This variable is useful if the user changes the identifier value in the control file or by a control panel.

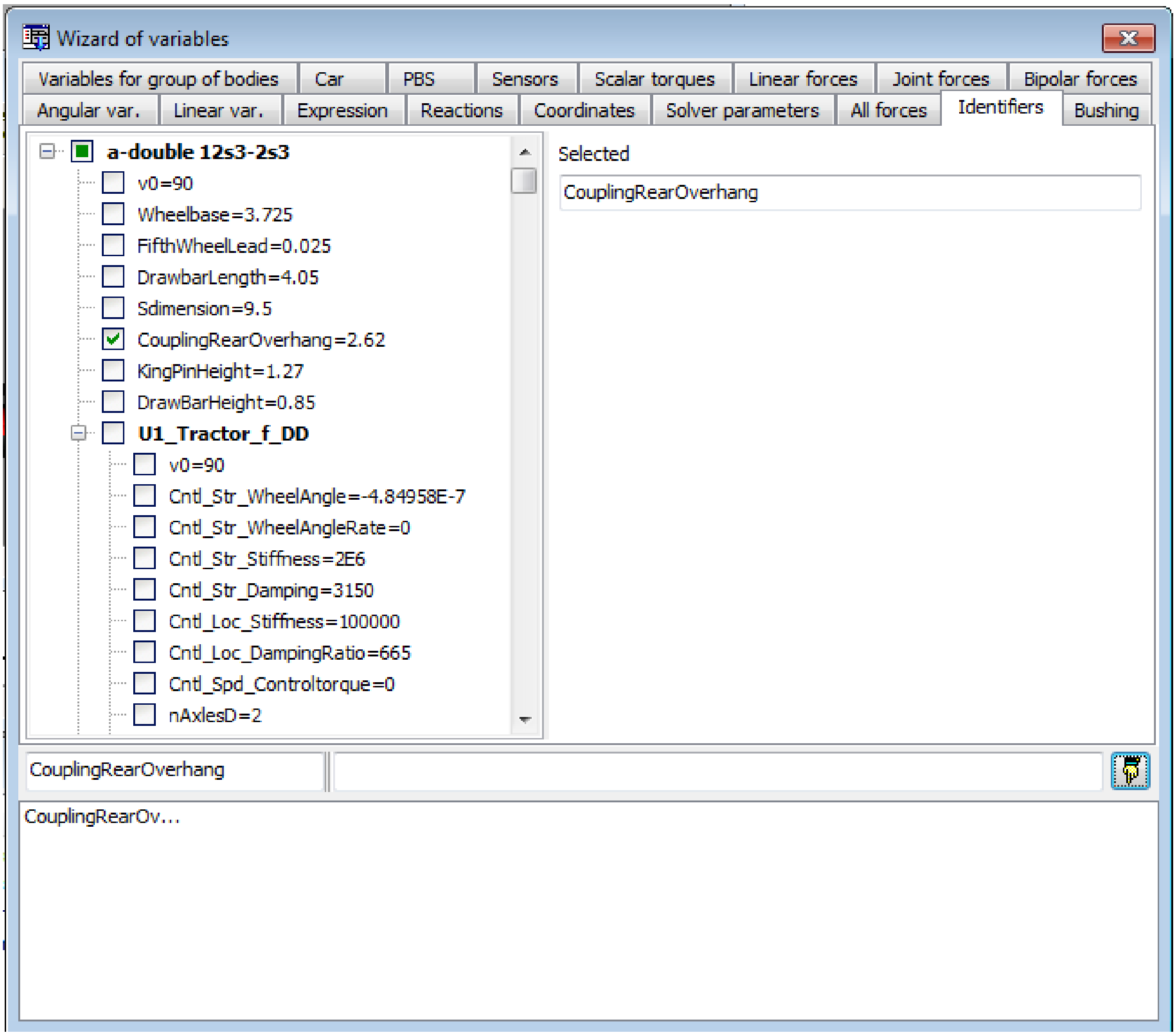


Figure 4.34. Identifiers

4.3.2.14. Variable - expression

Use the **Expression** tab of the wizard to create new variables by applying a series of arithmetic operation, functions, if statements etc. to preliminary created variables.

Consider examples of variables, which can be created here.

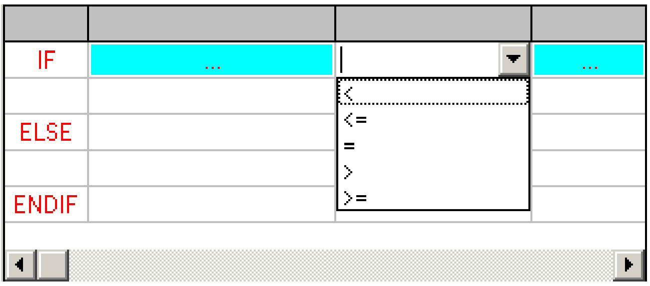
- Principal vector of any set of forces as well as its magnitude and components.
- Multiplication of any variable by a number, e.g. to get a non-standard units such as inches, pounds.
- Function of time, for example, *sin*(10*t*).
- Difference between two variables.

4.3.2.14.1. Notions and rules

The following actions should be done to create a variable - expression.

1. Prepare variables, which are used in the new variable (for example, open a list of variables). If it is necessary, create new variables with the help of the wizard and put them into the container.
2. Programming consists in adding operators with one or two operands.

- Each operator (function or operation) is located in one row of the table except the IF statements. The operator row contains a name of the result in the first column, an image or name of the operator (e.g., sin), and light blue boxes for one or two operands.
- A new operator can be either written in an empty active row or replaces an existing operator in an active row by clicking the button corresponding the operator (e.g. +).
- An operand is
  - a number (should be directly written in the box),
  - time identifier t (should be directly written in the box),
  - a variable from the wizard container, list of variables or graphical window (drag the variable with the mouse and drop it in the operand box),
  - a result of one of the upper operators (drag the name of the result with the mouse and drop it in the operand box),
  - a standard identifier (pi, rtod, dtor, mtoi, ntop), see Chapter 3, Sect. Standard functions and constants.
- A new operand can replace an old one in the same manner, i.e. by a direct input or by drag-and-drop.
- Use the pop-up menu to delete or insert a row to the table or to clear the table.
- Use the drop-down list to select a condition (<, <=, >…) in the IF statements. The list appear after clicking on the corresponding cell of the IF row.



- Deleting one of the row with the main element of an IF statement (IF, ELSE, ENDIF) leads to deleting the entire operator and all internal operators.

3. To add the created variable to the wizard container

- enter the variable name;
- select an operator, which result corresponds to the new variable;
- send the variable into the container by clicking the  mouse.

4.3.2.14.2. Operators

Operator	First operand	Second operand	Comments
+ - *	Vector or scalar	Vector or scalar	Multiplication of two vectors is the cross product
/	Vector or scalar	Scalar	
=	Vector or scalar	-	
sin, cos, abs,ln,exp sqrt,atan,sign	Scalar	-	Elementary functions
χ	Scalar	-	Heaviside function (1 for positive argument, 0fo other cases)
pow	Scalar	Scalar	Power function, the first operand is the base; the second one is the exponent.
Px, Py, Pz	Vector	-	Component of a vector
||•||	Vector	-	Norm of a vector
•	Vector	Vector	Scalar product of vectors

IF	Scalar	-	IF statement
			IF (condition)
			[group of operators]
			ENDIF

IF…ELSE	Scalar	-	IF statement
			IF (condition)
			[group of operators]
			ELSE
			[group of operators]
			ENDIF

4.3.2.14.3. Example

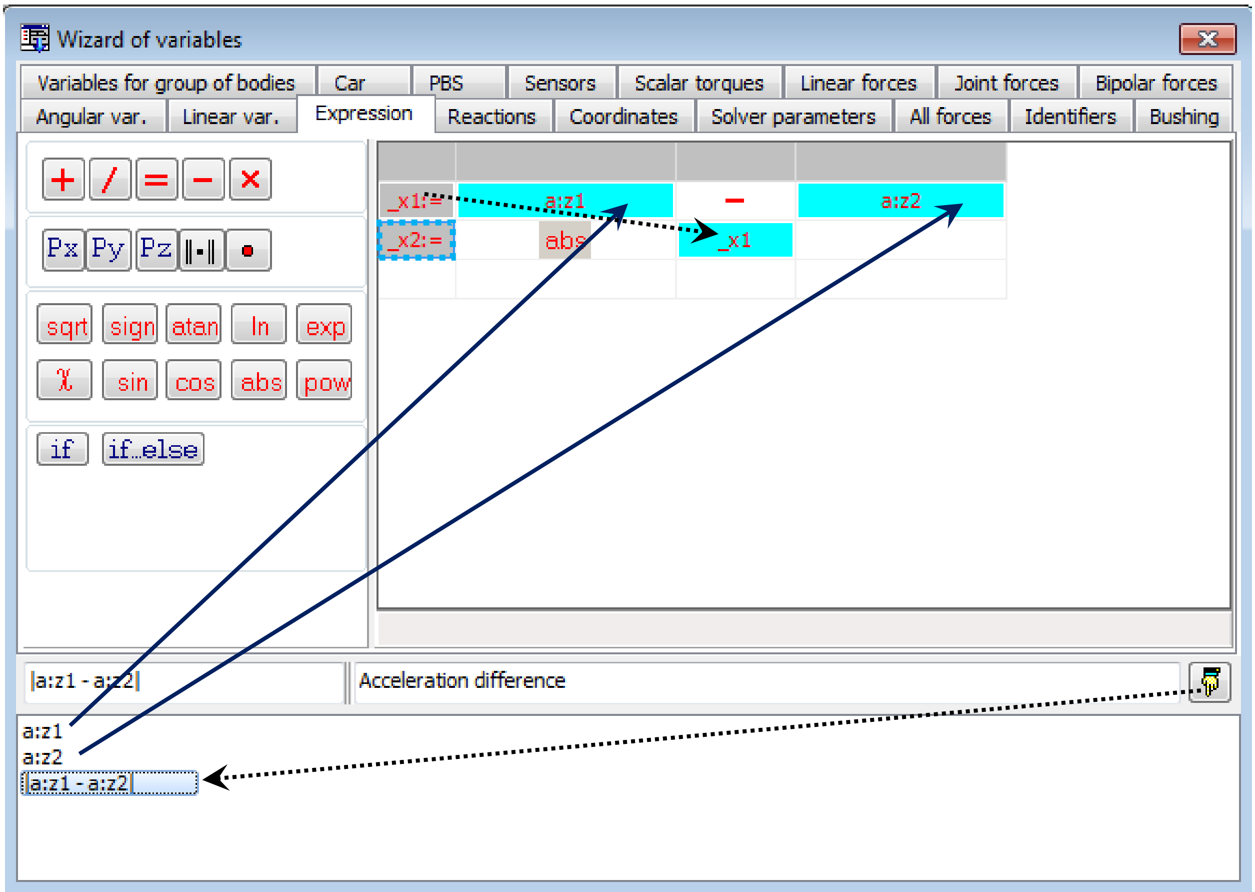


Figure 4.35. Example for programming a variable

Consider a variable, which is an absolute value of difference of two scalar variables, e.g. Z-projections of accelerations of two points $|a_{z1} - a_{z2}|$.

Here is a sequence of actions to create the variable (Figure 4.35).

- Create the variables az1, az2 with the help of the wizard and send them into the container.
- Click the button to add the subtraction operator.
- Drag the variable az1 and drop it into the box of the first operand, and the variable az2 into the box of the second operand.
- Click the mouse on the second (empty) row of the table to make the row active. Click the abs button to create this operator in the second row of the table.
- Drag the name of the result of the first operand (_x1) and drop it into the operand box of the operator abs.

Enter a name of the variable, e.g. |az1-az2| and click the button to send it into the container. Note, that the second row must be active.

4.3.2.15. Special variables for rail vehicles: tabs Railway and Track coordinate system

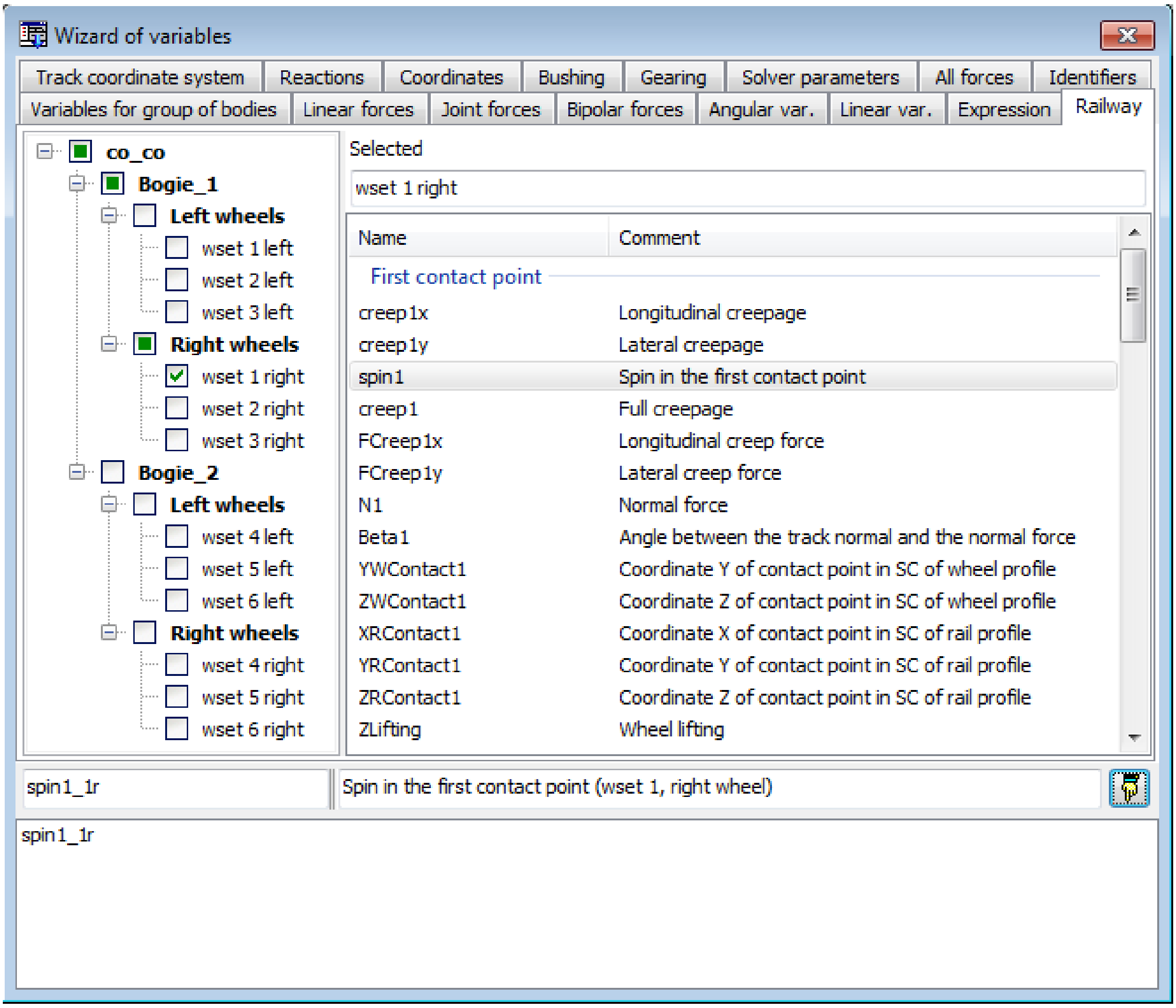


Figure 4.36. Special variables for rail vehicles: Railway

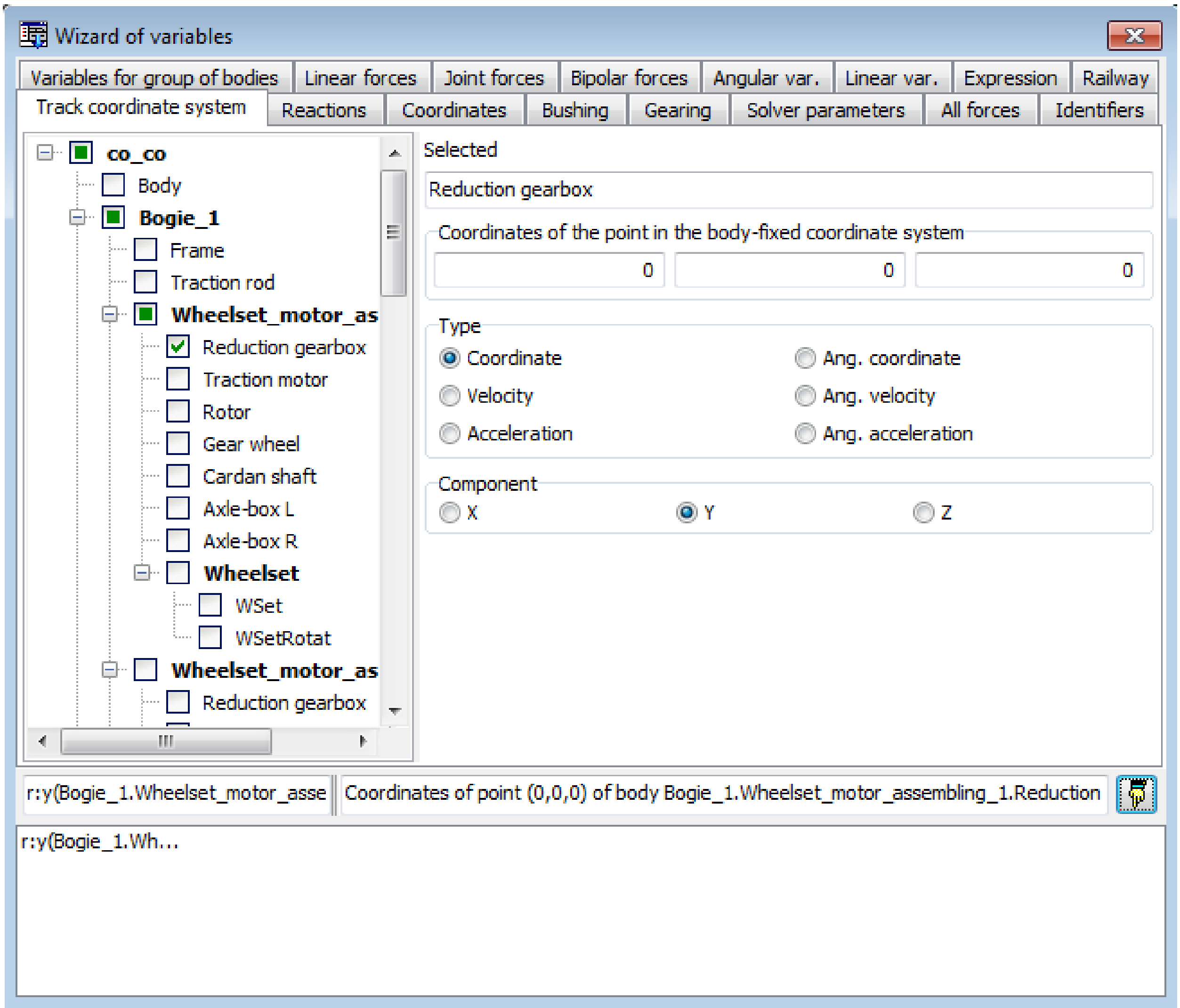


Figure 4.37. Special variables for rail vehicles: Track coordinate system

These tabs are visible for models of rail vehicles only. Description of the corresponding variables is located in Chapter 8, Sect. *Some features of creation of variables*.

4.3.2.16. Special variables for road vehicles: tab Road Vehicle

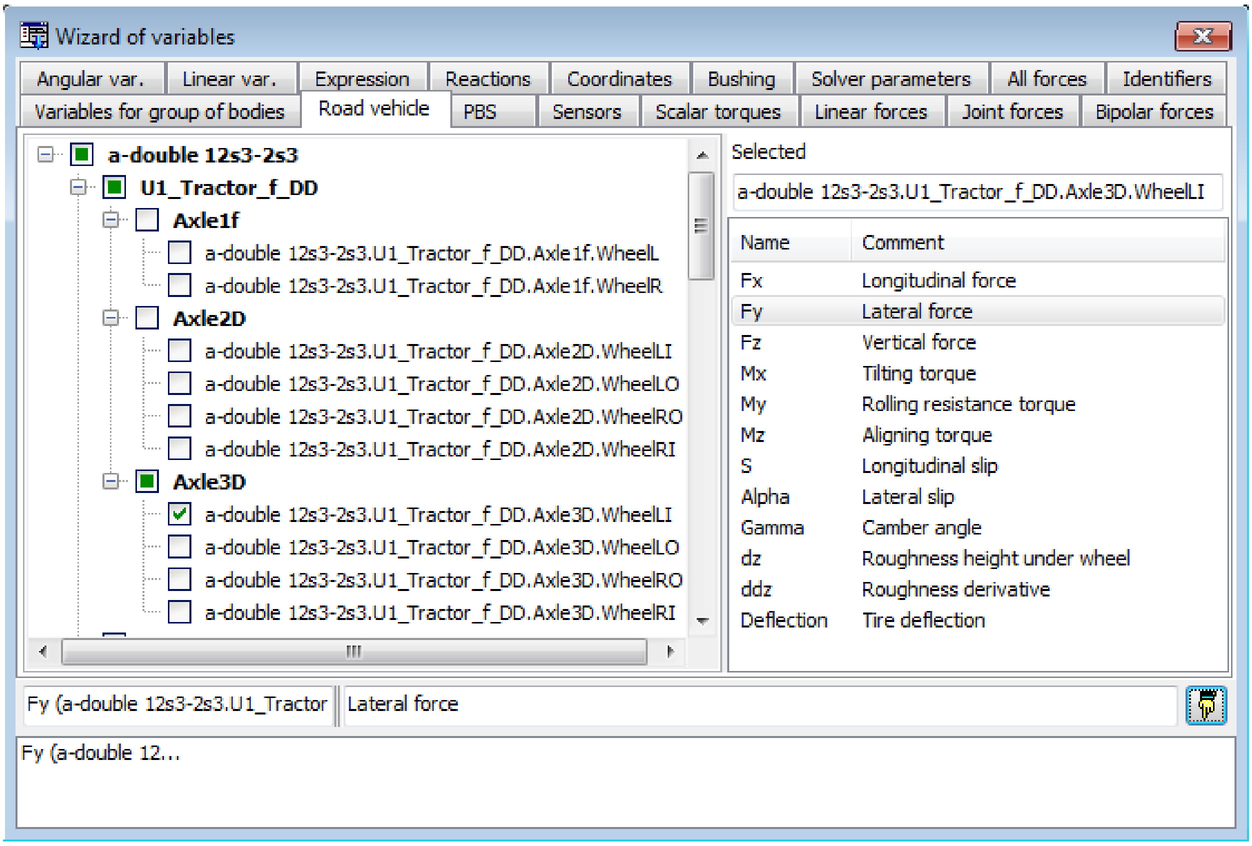


Figure 4.38. Special variables for a road vehicle

These tabs are visible for models of road vehicles only, the UM Automotive module. Description of the corresponding variables is located in Chapter 12, Sect. *Road vehicle specific variables*.

4.3.2.17. Special variables for tracked vehicles: tab Tracked Vehicle

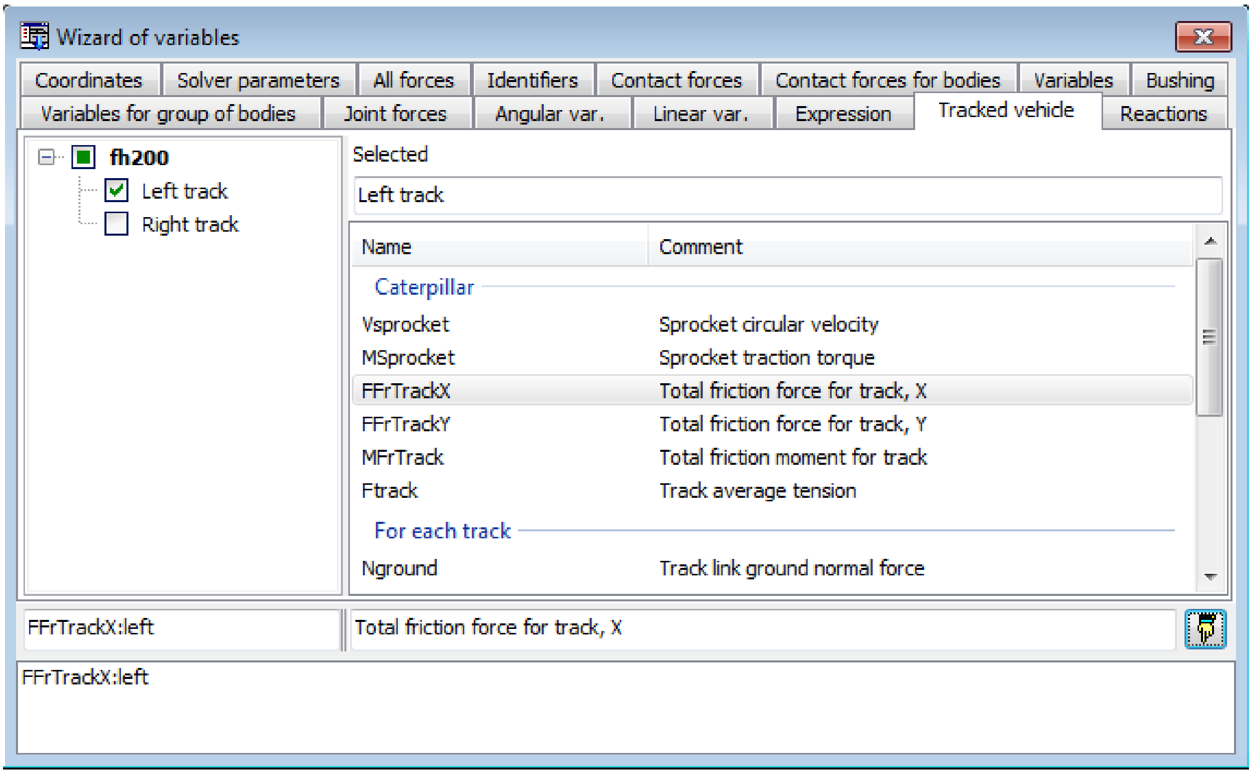


Figure 4.39. Special variables for a tracked vehicle

These tabs are visible for models of tracked vehicles only. Description of the corresponding variables is located in Chapter 18, Sect. *List of special variables for tracked vehicles*.

4.3.2.18. Special variables for simulation of longitudinal train dynamics: tab Train

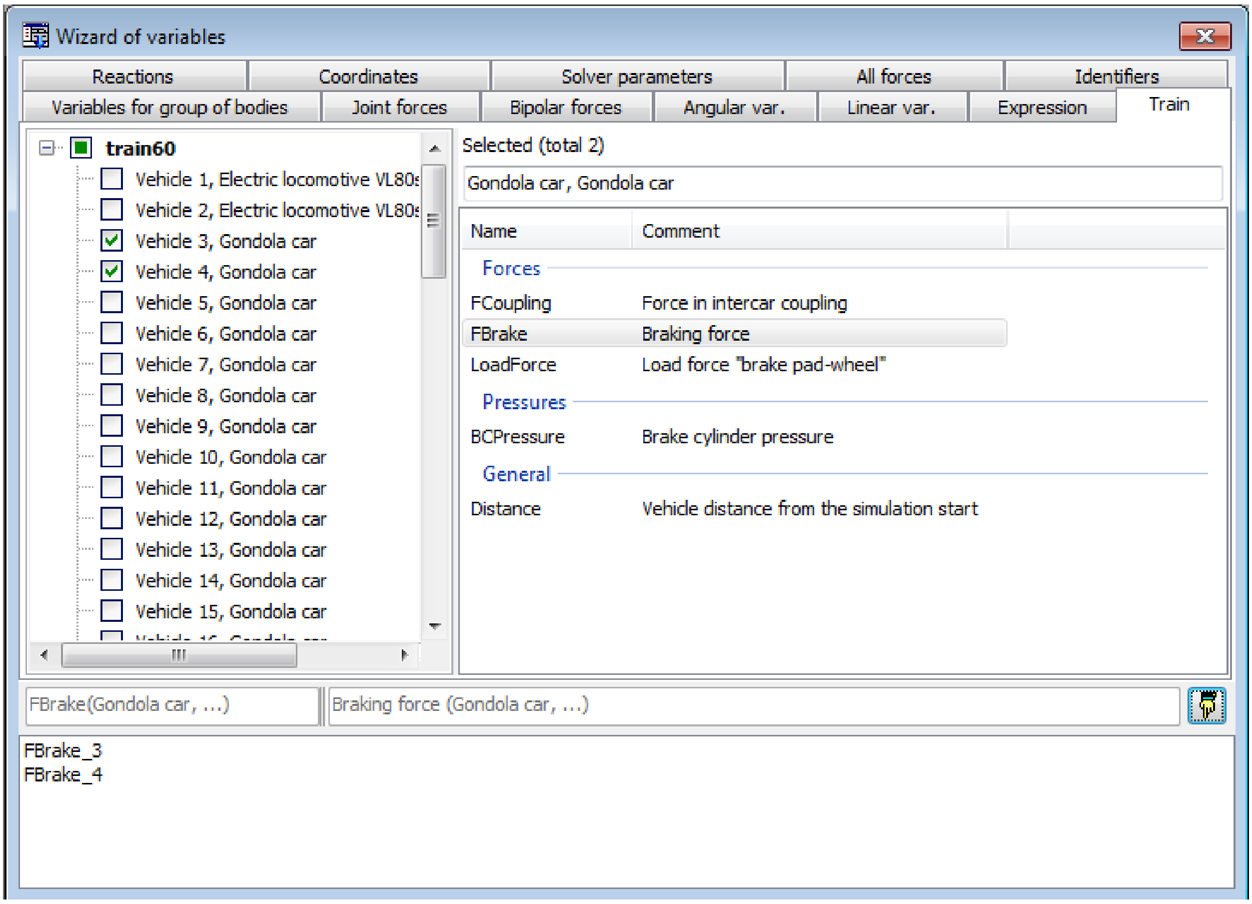


Figure 4.40. Special variables for a train

These tabs are visible for models of trains only, see Chapter 15.

4.3.2.19. Special variables: External libraries

This tab is visible if the loaded in UM Simulation model uses external libraries. It might be libraries imported from Matlab/Simulink as well as user defined libraries. Detailed information about developing, connecting and using external libraries please find in the “Sect. *Creating and using external libraries*”, Part 5 of UM User’s Manual.

Element tree in the left part of the window of **the Wizard of variables** includes all *input* and *output signals* for all connected external libraries, see figure below.

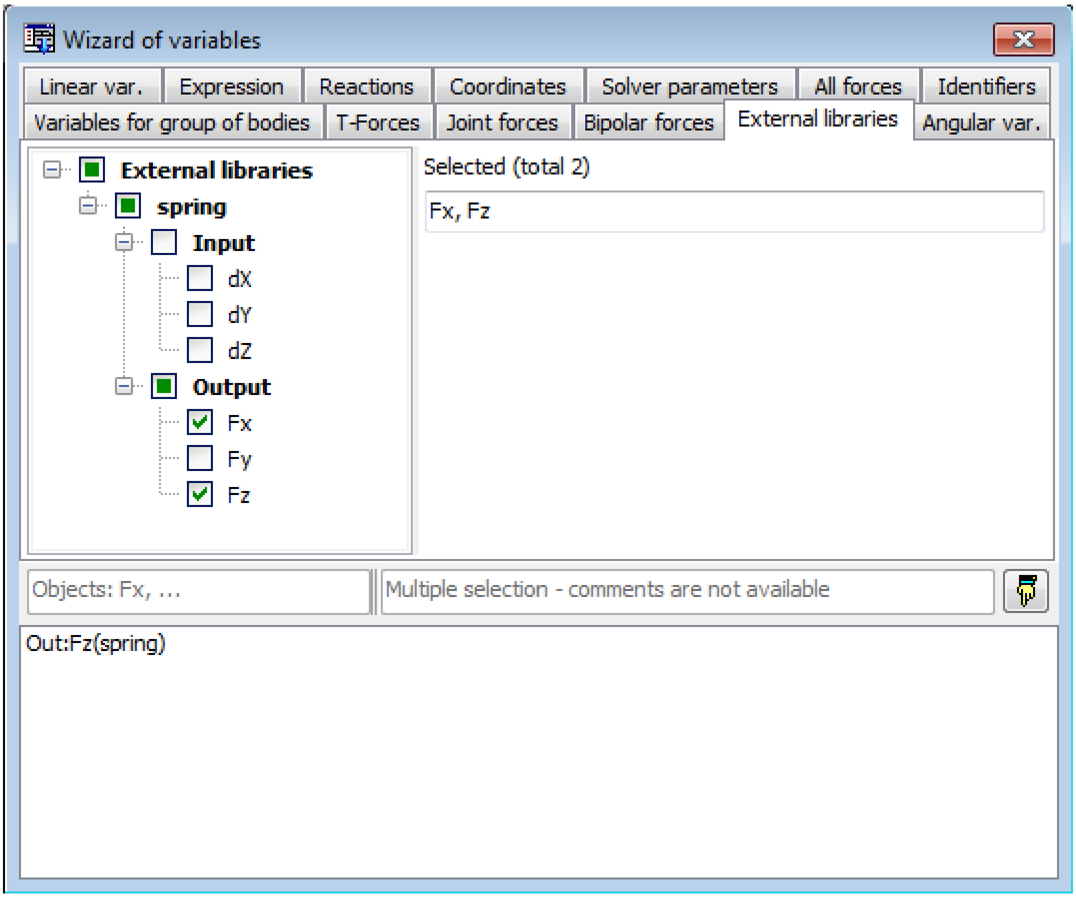
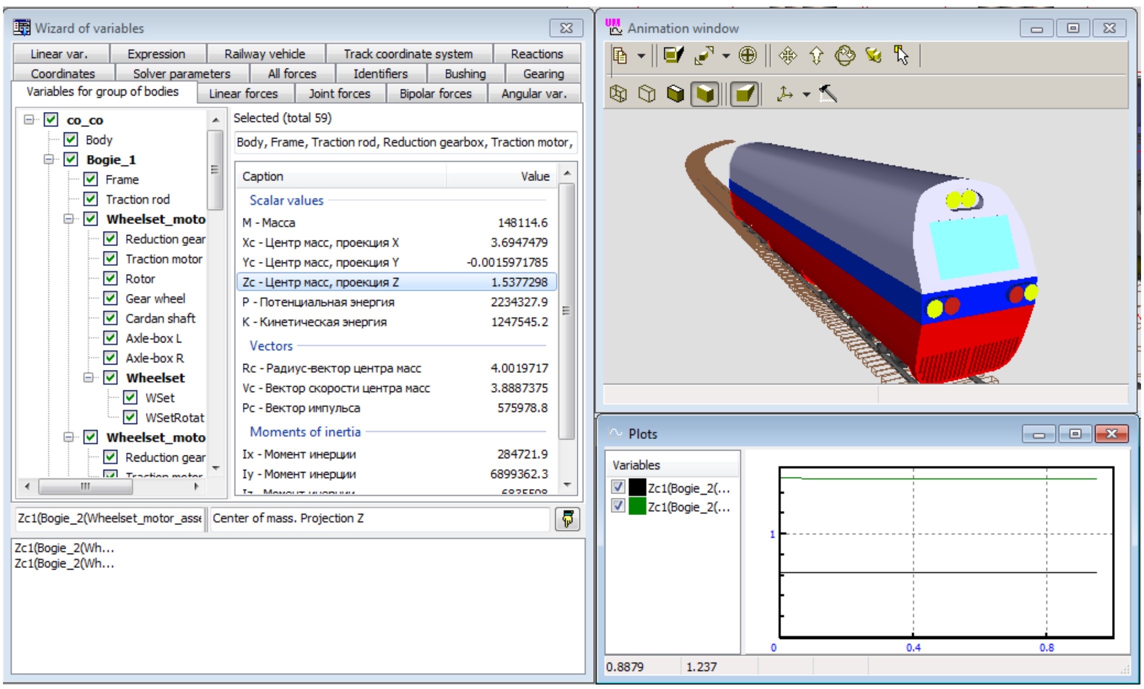


Figure 4.41. Input and output signals for external libraries

4.3.2.20. Variables for group of bodies

In various applications there might be needed to calculate certain parameters for a selected group of bodies, e.g.: the mass, the position of the center of gravity, the kinetic energy and so on. For example, the mass and the center of gravity might be useful in railroad applications to check the accuracy of positioning of a locomotive or its parts (e.g. bogies); kinetic and potential energy are useful for examining the accuracy of numerical integration of a conservative multibody system since the sum of those energies should be preserved in time, and so on.

For selecting the group of bodies and creating the above-mentioned variables, tab **Variables for group of bodies** is used. The general view of the dialog window is shown in Figure 4.42, left. In the upper part of the window, the bodies and subsystems of the simulation object are displayed in the form of a tree; It is possible to select either single particular body, or particular subsystem, or the whole simulation object. To select the body, click the left mouse button in a checkbox near a body's or a subsystem's name. On changing the selection of a group of bodies, the list of available parameters for group calculations is displayed in the bottom part of the window, together with their current numerical values. This container of variables can be used as described above in the Wizard of Variables, and all operations for standard variables are available. For example, the user can drag and drop the variables into a graphical window to plot a time history of the scalar variables, right bottom, or drag and drop the vector-valued variables into an animation window to plot a trajectory (and/or velocity) of the center of gravity. Besides that, those variables can be used in expressions to perform the arithmetic operations on them, e.g. for summation of kinetic and potential energy.



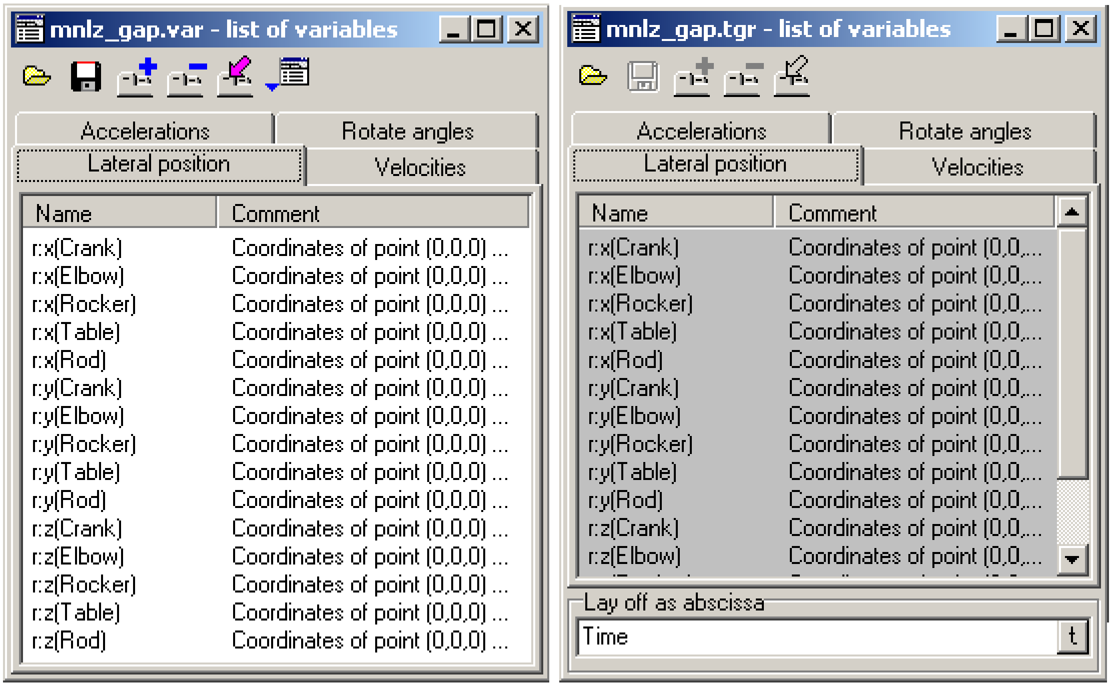
a) b)

Figure 4.42. Variables for group of bodies: a) tree of subsystems and bodies; b) windows that might use the variables

4.3.3. List of variables

In modeling complex technical systems the user is often interested in certain kinematic and dynamic characteristics (variables) of an object such as the acceleration of a point, reaction forces in some joints, etc. A *list of variables* allows the user to store the variables whose behavior should be analyzed. For example, a *list of variables* in the typical design of a railway vehicle with the UM can have a few hundreds of variables. It is recommended to create *lists of variables* in advance.

As it will be shown below, this method allows the user to reduce drastically his efforts in simulation of objects. For instance, analysis of rail vehicle dynamics requires evaluations of hundreds of variable, and even description of the list of necessary variables is often a time consuming process. The considered technique allows grouping variables, using special technical terminology, applying lists of variables prepared for one object to a similar one etc.



a) b)

Figure 4.43. List of variables (a), list of calculated variables (b)

During the simulation you can save the numerical characteristics of the different variables in in *file of calculated variables* process (Sect. 4.3.3.3). All variables stored in a file are available for subsequent analysis with the help of various tools such as table a processor, a window for statistical analysis and a graphical window.

4.3.3.1. Creating a list of variables

Every list of variables contains elements, which are kinematic or dynamic characteristics of an object. As a rule, lists created for quite different object are not compatible.

Each element of the list (a variable) is characterized by the name and comment. Both the name and the comment may be an arbitrary set of symbols.

To create a list, load the corresponding object (the **File | Open** menu command) and call a list window with the help of the **Tools | List of variables…** main menu command or click the button in the tool panel.

The appeared window contains an empty tab (*No name*). The buttons on the top are used for:

– reading a list of variables and calculated lists (files *.var or *.tgr);

– saving a list (*.var);

– adding a new tab to the list;

– deleting a tab;

– renaming a tab;

– options of statistical processing groups of variables (scale factor, filter options etc.The options are used at processing of scanning results (Chapter 6, Sect. *Processing results of scanning*).

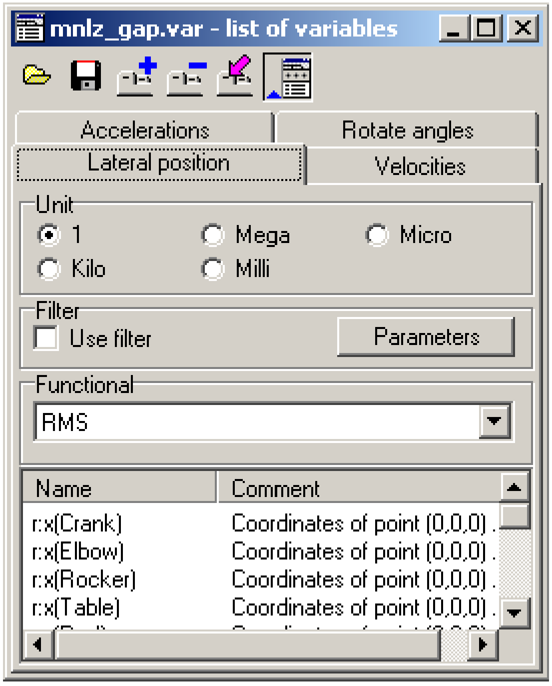


Figure 4.44. Options of statistical processing groups of variables

One or several variables can be selected in the list in the standard manner.

The selected variables can be deleted (the **Delete** key) or transferred into another tool window with the help of the *Drag&Drop* technique:

- graphical window;
- table processor (calculated variables only);
- another list of variables;
- window for statistical analysis (calculated variables only);

To transfer the variables from the list into other tool windows, select them in the list, press the left mouse button and drag.

To change the order of variables in the tab, select them in the list, press the left mouse button and drag them to the new position.

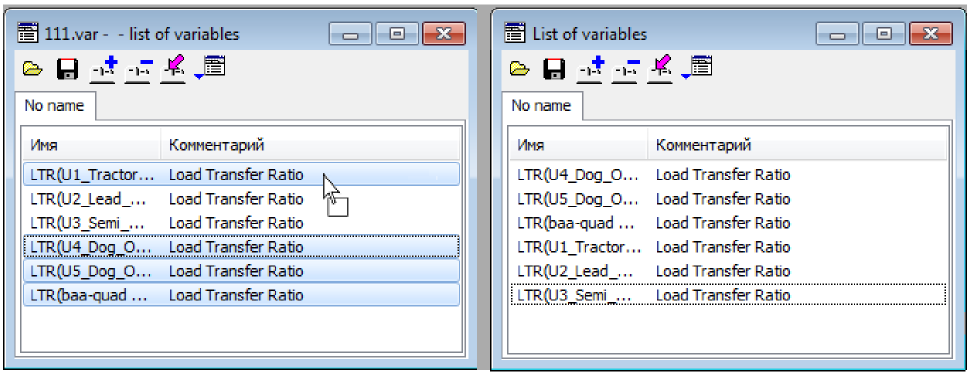


Figure 4.45. Changing order of variables in the tab

To transfer variables from current tab to another, select them in the list, press left mouse button, drag it to the target tab (it will cause to open variable list of target tab), then drop them.

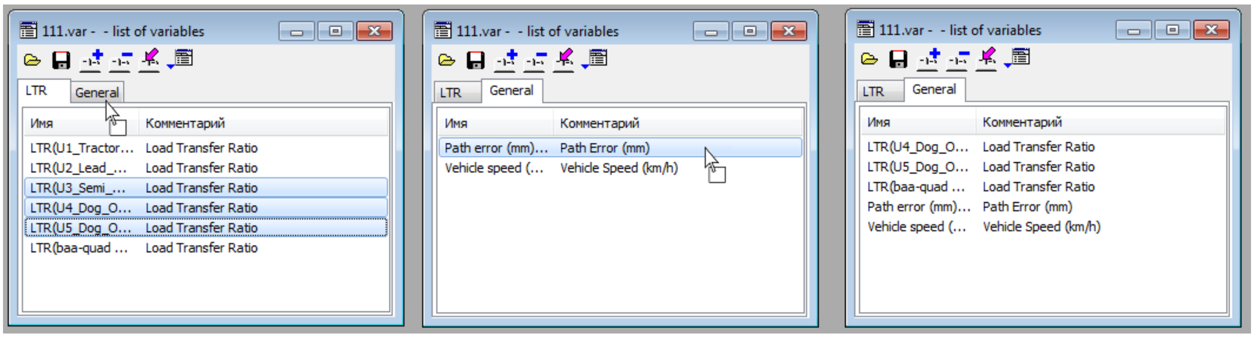


Figure 4.46. Moving variables from one tab to another

4.3.3.2. Filling a list of variables

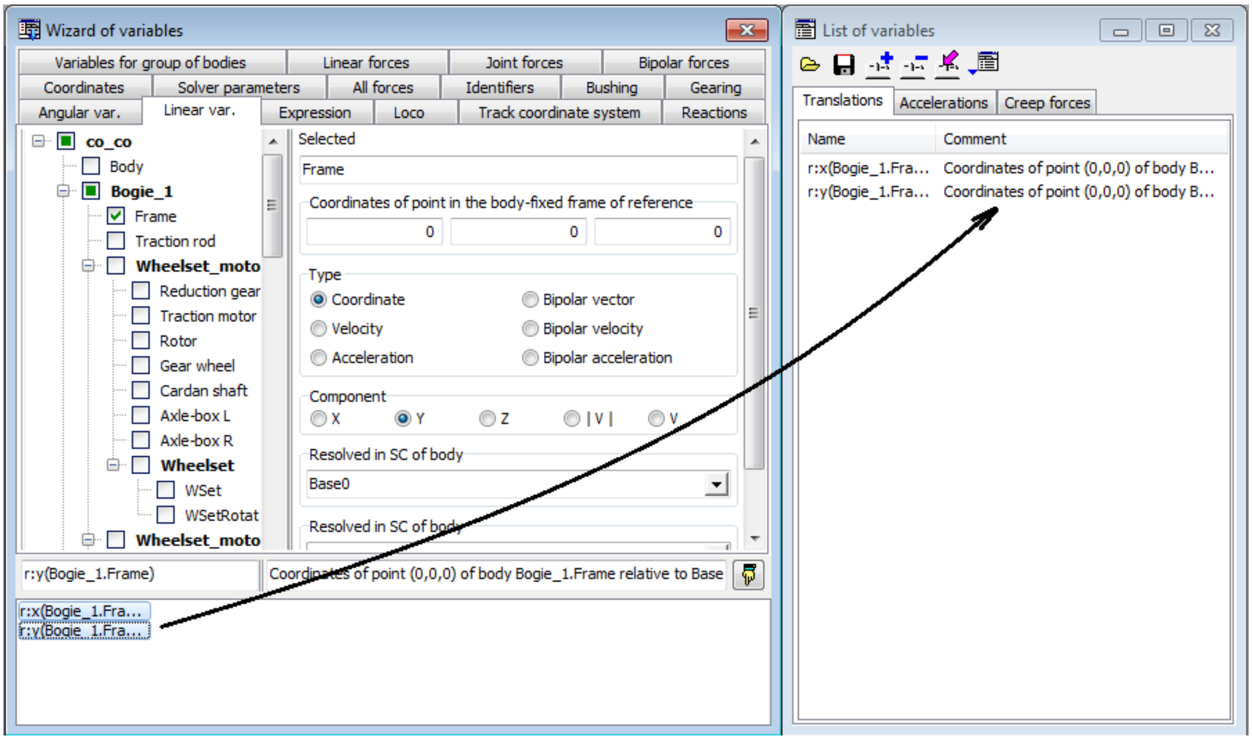


Figure 4.47. Filling a list of variables

Filling a list of variables is carried out in two ways. The first method consists in transferring variables from other lists and graphical windows. The second one is the creation of a new variable with the help of the wizard. Every variable or a set of variables adding to the list can be located on separate tabs according to the type of variables.

To add a new variable using the wizard

- open the *Wizard of variables* (Sect. 4.3.2) with the help of the main menu command **Tools | Wizard of variables** or by clicking the button in the tool panel;
- create variables and drag them into the corresponding tabs of the list window;
- save the list.

4.3.3.3. Processing calculated lists

If the list contains calculated variables, any variable from the list can be transferred into the processing tools

- graphical window;
- table processor;
- window for statistical analysis.

After receiving the variable (or a set of variables), the tool processes immediately the corresponding data and visualizes results.

Processing of data may be carried out in dependence on time (the default independent variable) or on any other variable from the same list of calculated variables. To assign a variable as an abscissa, drag it into the *Lay off as abscissa* box in the bottom of the list window (Figure 4.43).

Use the *data interval* to process a part of calculated data for a variable. To setup the interval

- click the right mouse button within the list;
- select the *Options of interval* pop-up menu command;
- set the left and the right interval limits for the variable, which is laid off as abscissa.

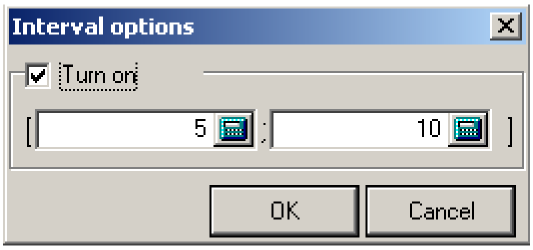


Figure 4.48. Interval options dialog box

4.3.3.4. Import file of calculated variables to MATLAB

You can import any file of calculated variables to MATLAB workspace by **LoadUMData-File.m** utility.

The utility load file of calculated variables to MATLAB workspace as two arrays.

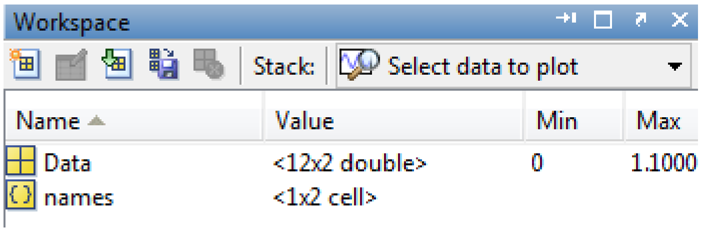


Figure 4.49. Interval options dialog box

Utility **LoadUMDataFile.m** is found in folder {Data UM}\Utils. You should type following command for loading file of calculated variables:

[*Data*, *names*] =LoadUMDataFile('*filename*'),

where *Data* is name of array with values of variables;

names is name of array with names of variables;

filename is name of file of calculated variables (without extension).

Fig. show results of working of the utility. Each column of "Data" matrix is a separate variable. First column is always an abscissa.

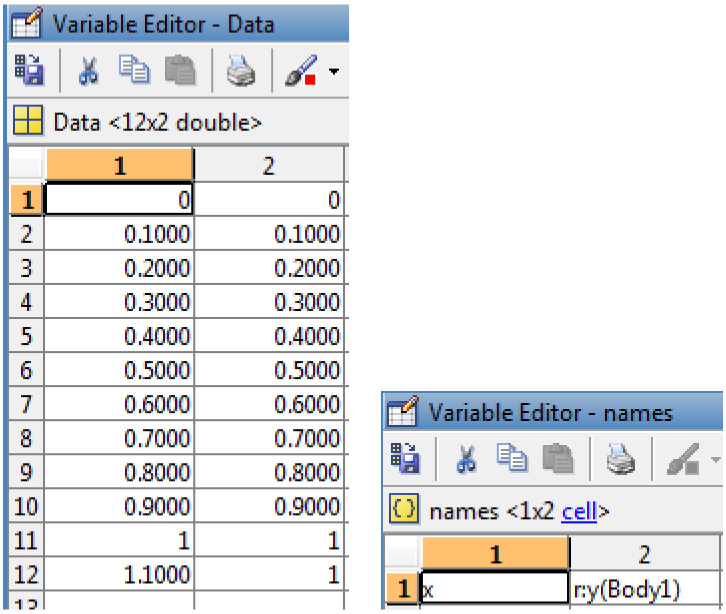


Figure 4.50. Interval options dialog box

4.3.4. Graphical window

A graphical window (Figure 4.51) is one of the most important tools for visualization of simulation results. Number of windows as well as the number of plots within a window is unlimited. Select the **Tools | Graphical window…** main menu command to open a window. Positions of windows on the program desktop and assigned lists of variables (except variables transported from lists of calculated variables) are stored in a configuration file (the **File | Save configuration** menu command).

A graphical window consists of

- container of variables,
- plotting area,
- tool panel.

Marking of axes is made automatically. If necessary, *scale coefficients* are drawn.

The status bar contains coordinates of the mouse cursor in the variable scale.

The container location within the window can be changes (the *Position* item of the pop-up menu).

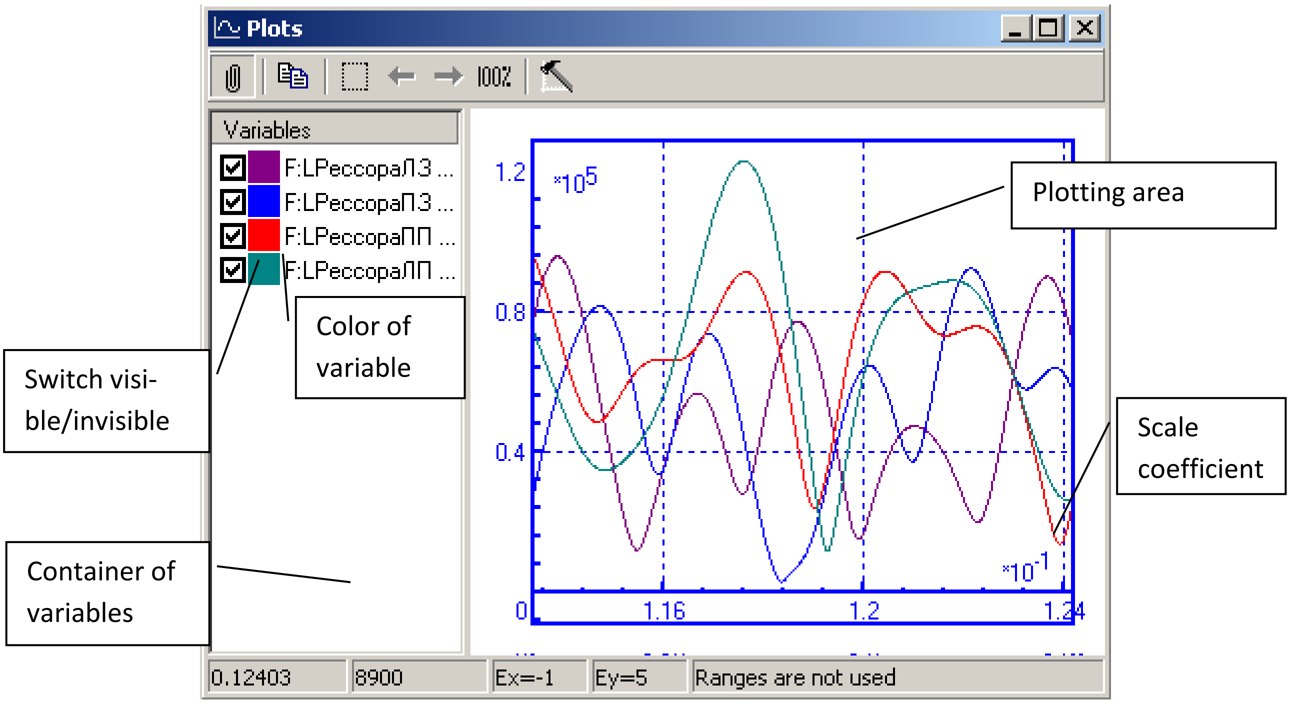


Figure 4.51. Graphical window and its elements

Selected graphs are plotted by thick lines or markers.

Use the buttons on the tool panel:

 – to fix the panel;

 – to copy graphs into the clipboard;

 – to zoom in a rectangular area;

 – to show the whole graphs (only visible graphs ate taken into account);

 – to open the **Window parameters** dialog box.

Press and keep the *left* mouse button within the plotting area to move graphs according to the mouse cursor motion.

Press and keep the *right* mouse button and move the cursor within the plotting area to zoom in/out graphs.

Press and keep the **Shift** key, then select a rectangular area by the mouse cursor (keep pressed simultaneously the right mouse button) to zoom in this area.

There exist fixed and drop-down modes of the tool panel. In the drop-down mode move the mouse cursor to the window caption area until the panel drops down. Use the option window to fix the panel by default (Sect. 4.1.1).

Graphs can be transported from one graphical window to another, from lists of variables and from lists of calculated variables (Sect.4.3.2.16). Any plotted graph from a graphical window can be processed by a *table processor* (Sect. 4.3.7), and by a *statistic tool* (Sect.4.3.8).

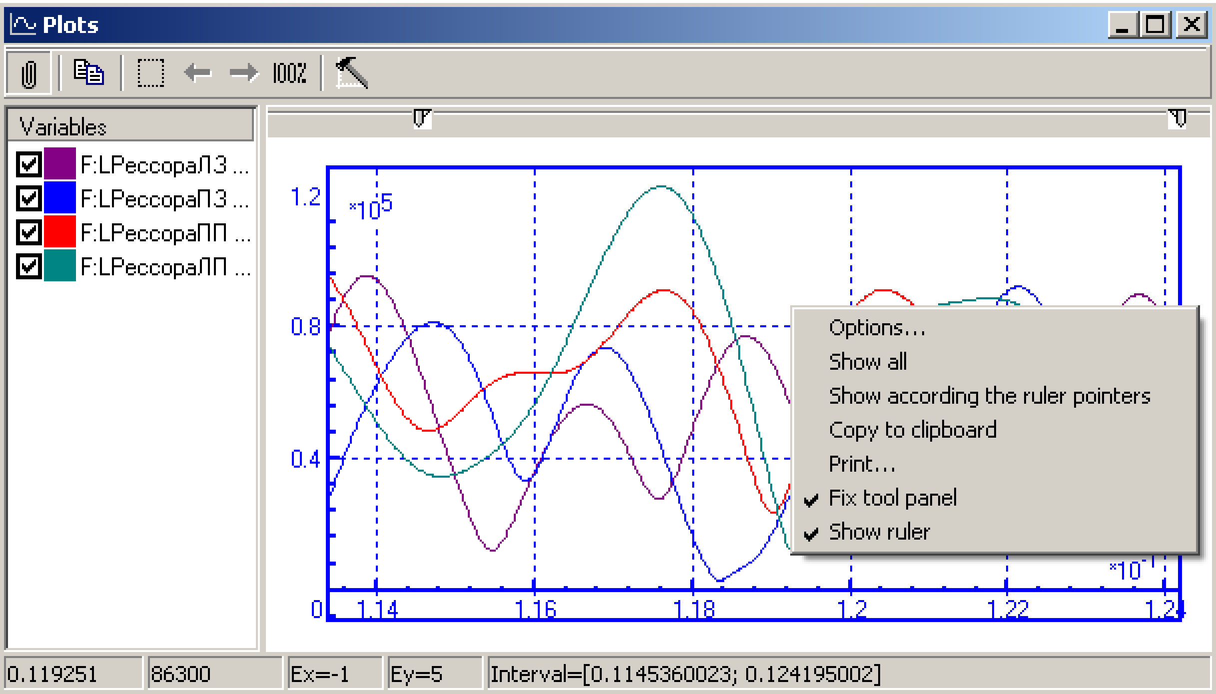


Figure 4.52. Pop-up menu for plotting area

The items of pop-up menu of plotting area shown in Figure 4.52 (many of the items duplicate the command of the tool panel):

- **Options…** – to open the **Window parameters** dialog box,
- **Show all** – to show the whole graphs (only visible graphs ate taken into account),
- **Show according to the ruler pointers** – to show graphs according to the ruler pointers,
- **Copy to clipboard** – to copy graphs from the plotting area to the clipboard as a picture,
- **Print…** – to print the plotting area,
- **Fix tool panel** – fix/unfix the tool panel,
- **Show ruler** – show/hide ruler.

It is often necessary to cut a part of a plot for the further analysis, for instance to remove a transient process. Use the **Axes | Style** tab in the *Options* dialog box (Sect. 4.3.4.2) or a *ruler* (the Show ruler item of the pop-up menu, Figure 4.52) to specify an interval.

4.3.4.1. Copying graphs to clipboard, text file and file of calculated variables

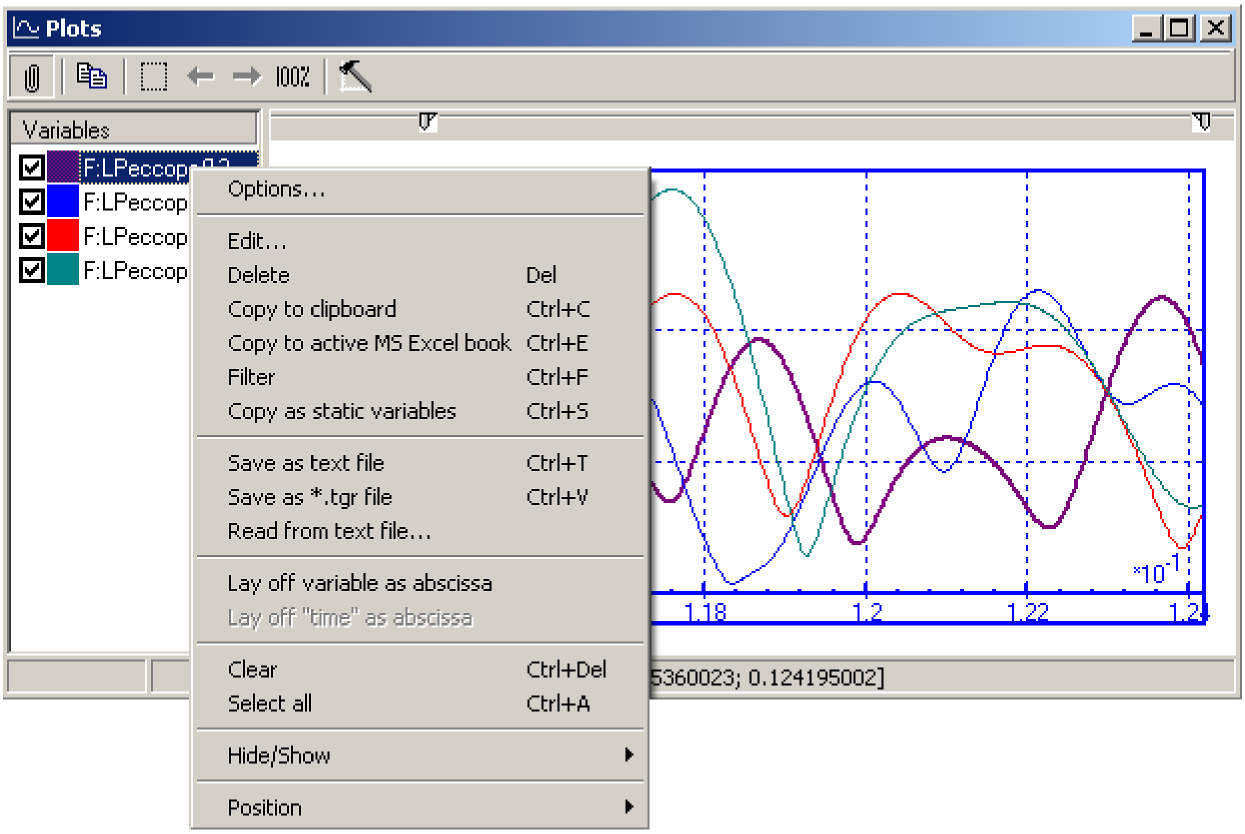


Figure 4.53. Pop-up menu for container

Select one or a group of variables in the container (except those, which have been transferred from the list of calculated variables, Sect. 4.3.3.3), then click the right mouse button on one of the selected variables. The pop-up menu appears (Figure 4.53).

- Use *Edit* to set color and style setting of variable plot.
- Use *Delete* (**Del**) to remove variable from the container.
- Use the *Copy to clipboard* (**Ctrl+C**) item to copy the selected graph as a text.

  Example:

  %Description of variables:
  %Column 1 - Array of values laid off as abscissa
  %Column 2 - v:z(Crusher) [Velocity of point (0,0,0.1) of body Crusher relative to Base0…
  % Column 3 - a:y(Crusher) [Acceleration of point (0,0,0.1) of body Crusher relative …
  %
  9.99999997475243E-7 5.00000000000000E-1 1.35000002384186E+0
  3.02056260406971E-2 5.02337813377380E-1 1.34784018993378E+0
  4.30080145597458E-2 5.04678130149841E-1 1.34569251537323E+0
  6.30080178380013E-2 5.09747564792633E-1 1.34109389781952E+0

  The first % character (a prefix) is inserted according to the program option (Sect. 4.1.1).
- Use the *Copy to clipboard* (**Ctrl+C**) item to copy the selected graph as a text.

- Use the *Save as text file* item to save the data in a text file. The output format is specified by the options, Sect. 4.1.1.

n variables can be stored in *n*+1 columns: time and n variables. Otherwise, *n* successive groups are stored in the format: time as the first column and the *i*-th variable as the second one, *i*=1,…n, Sect. 4.3.4.1.

Example of saving two variables in a text file in two columns:

```
%
% name=v:y(Pendulum.Sensor1)
% comment=Pendulum.Sensor1; v; projection Y
%
0               0
0.02            -0.062579758
0.039999999     -0.12598488
0.059999999     -0.19101849
…………………………………
%
% name=v:z(Pendulum.Sensor1)
% comment=Pendulum.Sensor1; v; projection Z
%
0               0
0.02            -0.097109251
0.039999999     -0.19339079
0.059999999     -0.28800511
………………………………….
```

Example of saving the same variables in three columns:

```
%Description of variables:
%Column 1 - Time
%Column 2 - v:y(Pendulum.Sensor1) [Pendulum.Sensor1; v; projection Y]
%Column 3 - v:z(Pendulum.Sensor1) [Pendulum.Sensor1; v; projection Z]
%
0               0                0
0.02            -0.062579758     -0.097109251
0.039999999     -0.12598488      -0.19339079
0.059999999     -0.19101849      -0.28800511
```

- Use the *Save as *.tgr file* to save the data in a file of calculated variables (Sect. 4.3.3.3).
- Use the *Filter* item to apply a frequency filter to selected variables.

4.3.4.2. Graphic window parameters

Click the  panel tool button or the *Options* item of the pup-up menu (Figure 4.53) to change the window parameters.

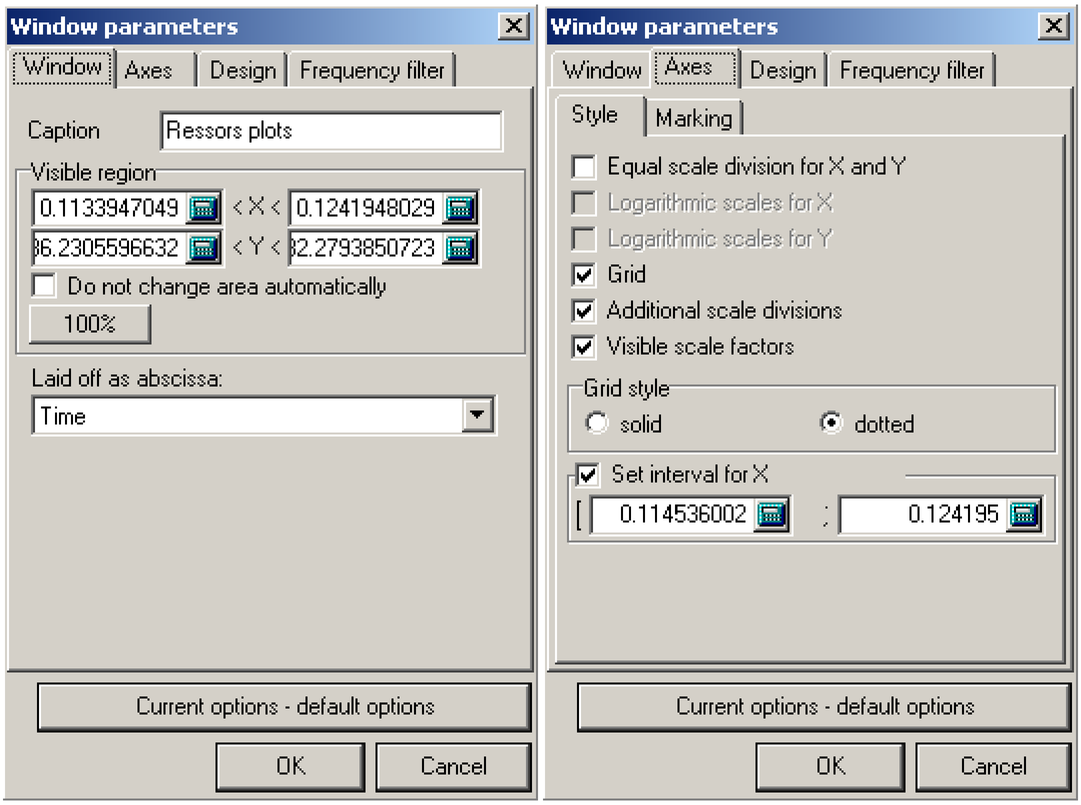


Figure 4.54. Parameters of a graphical window

The sheet **Window** includes:

- *Caption* – sets the window caption;
- *Visible region* - sets the plotting area ranges;
- *Laid off as abscissa* - sets a variable, which will be laid off as the abscissa (this action will be ignored for variables transported from lists of calculated variables), time is the default independent variable;
- *Do not change area automatically* mark means, that plotting area will not be resize automatically during simulation process.

The sheet **Axes** allows following:

- Set style the grid visible/invisible;
- Set equal scales for the abscissa and ordinate exes (the *square scales* option);
- Set logarithmic scales (statistical analysis only, Sect. 4.3.8);
- Set a filter interval.

The sheet **Design** lets user to turn up graphic window style.

4.3.4.3. Frequency filter

To carry out frequency filtration of the variable one has to choose some variables from the container and select **Filter...** in popup menu. Selected variables will be filtered accordingly the current preferences (popup menu command **Window parameters** – sheet **Frequency filter**) and add to the list of variables with name: _ + variable name. The filters of the following types are available:

- Rectangular LPF;
- Low-pass filter;
- High-pass filter;
- Band-stop filter;
- Band-pass filter;
- Butterworth low-pass filter;
- Butterworth high-pass filter.

Input strings **Low frequency** and **Higher frequency** set wave range of the filter. Low-pass filter passes wave data in the range from 0 Hz to the **Higher frequency** value (**Low frequency** value is ignored). High-pass filter passes wave data in the range from the **Low frequency** value to the Nyquist‘s frequency. (**Higher frequency** value is ignored). Band-pass filter lets pass throw frequencies in the selected wave range. Band-stop filter excludes waves, which frequencies belong to the selected wave range, from the process.

Frequencies can be assigned to the absolute value (Hz) or part of the Nyquist‘s frequency.

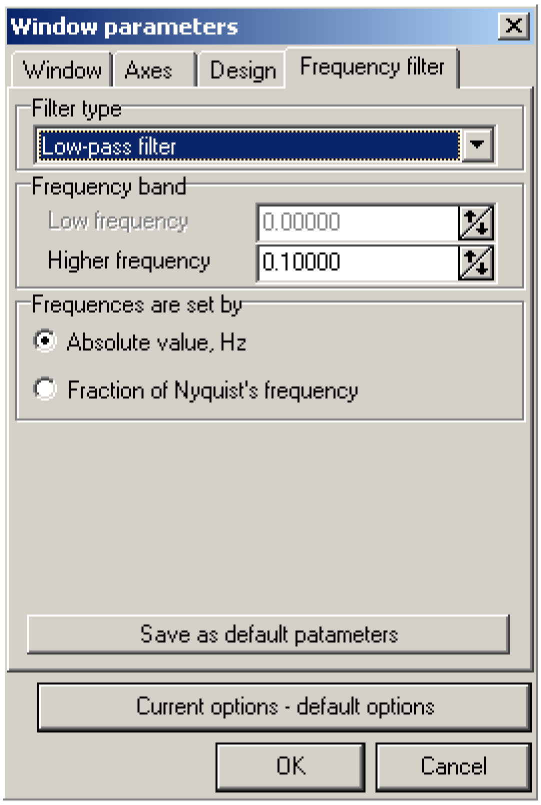


Figure 4.55. Frequency filter parameters

Save as default parameters – sets current parameters of the filter as default ones.

4.3.4.4. Change of a variable parameters

Double click the variable in the container. The dialog box for changing some graph parameters appears (Figure 4.56).

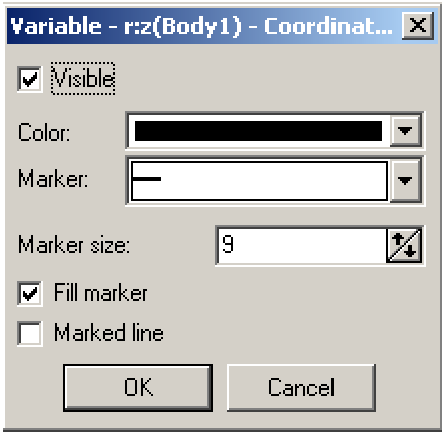


Figure 4.56. Parameters of a variable in a graphical window

Specify the dialog box parameters to

- make the graph visible/invisible;
- change the graph color;
- choose a marker, its size and filling option.

4.3.4.5. Export to MS Excel

To export graphs/histograms from graphical/histogram windows select necessary variables in a window and use the **Copy to active MS Excel Book** pop-up menu command. Selected graphs/histograms will be exported to MS Excel active book according to export settings (Sect. 4.1.4).

4.3.5. Histogram window

A histogram window (Figure 4.57) is an additional tool for visualization of simulation results. The number of histogram windows as well as the number of histograms within the window is unlimited. Select the **Tools | Histogram…** from the main menu or  button to open a window. Positions of the window on the program desktop and assigned lists of variables (except variables transported from lists of calculated variables) are stored in a configuration file (the **File | Save configuration** menu command).

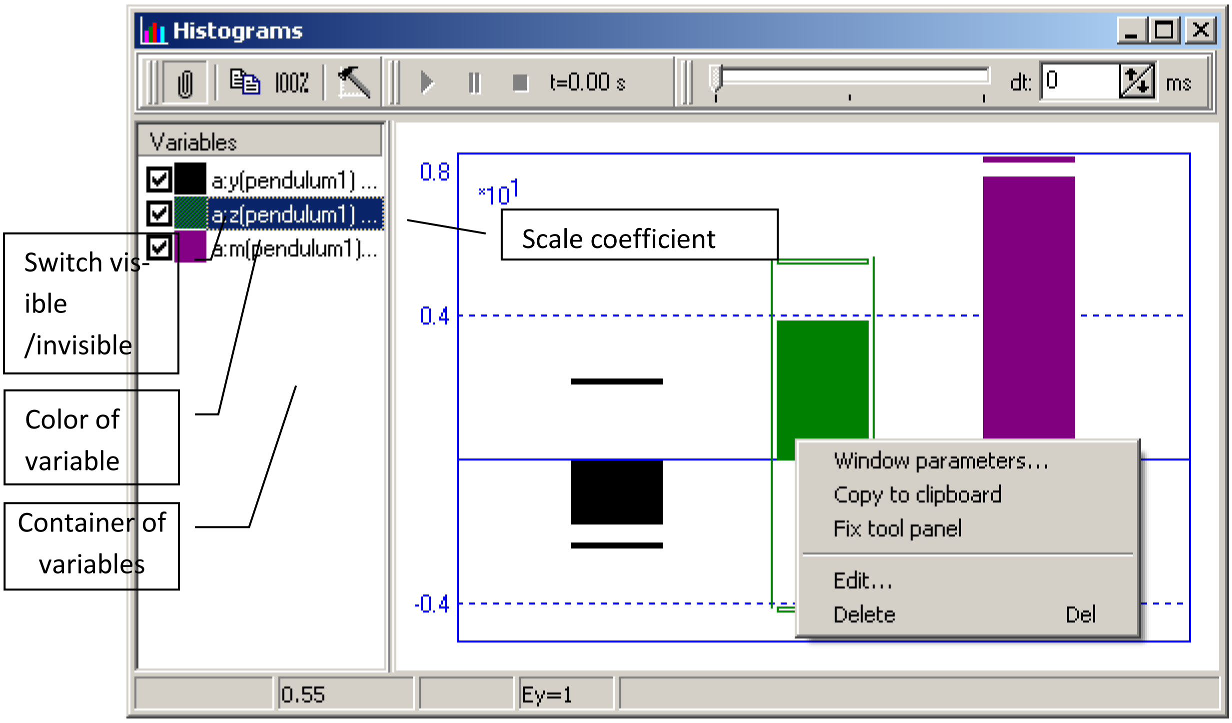


Figure 4.57. Histogram window

The histogram window is organized as well as the graphical one. It consists of the following elements:

- container of variables,
- tool panel,
- plotting area.

The user can put variables to container from **Wizard of variable** window (Sect.4.3.2) or **List of variables** (Sect.4.3.2.16).

The tool panel gives access to the following instruments:

 – to fix the panel,

 – to copy histograms into the clipboard as a picture,

 – to set an optimal scale in the plotting area,

 – to open the *Histogram window parameters* dialog box.

Controls of calculated variables animation (Figure 4.58):

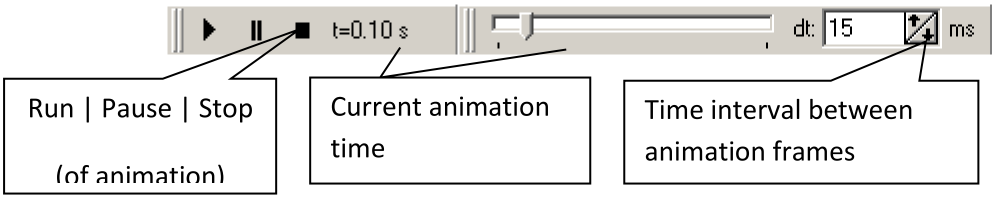


Figure 4.58. Animation tools of histogram window.

Remark. If you want to animate calculated variables, you have to place the variables to the container before the simulation starts.

Each variable (marking for visible) is animated as a *histogram* at the plotting area.

A histogram looks like a rectangle (Figure 4.59) and gives the graphical information of the variable:

- Current value of variable – coordinate of the moving top/lower side of rectangle,
- Current maximum of variable – upper line,
- Current minimum of variable – lower line.

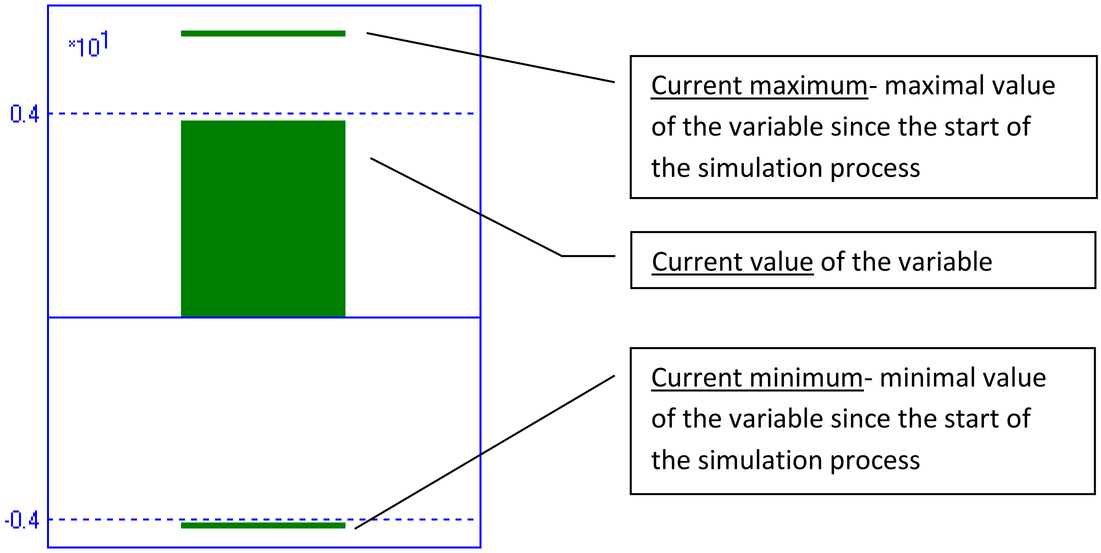


Figure 4.59. Graphic information reflected on histogram.

The user can set graphical preferences for each histogram. Call popup menu by right click at the histogram area and select **Edit…** command to open *Histogram graphic parameters* window (Figure 4.60).

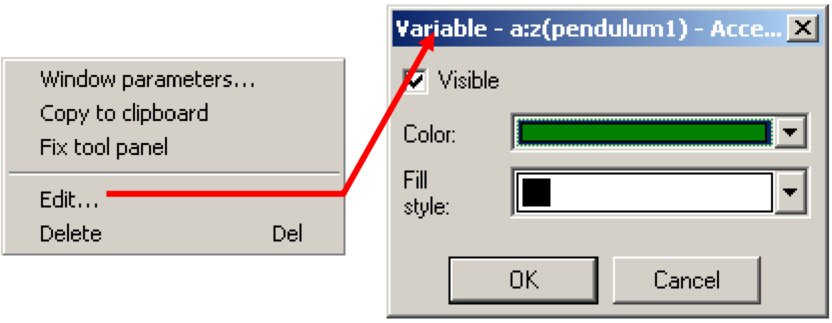


Figure 4.60. Histogram parameters

Plotting area style looks like the style of a graphic window.

Marking of axes is made automatically. If necessary, *scale coefficients* are drawn.

The status bar contains coordinates of the mouse cursor in the variable scale.

The container location within the window can be changed (the *Position* item of the pop-up menu).

There exist fixed and drop-down modes of the tool panel. In the drop-down mode move the mouse cursor to the window caption area until the panel drops down. Use the option window to fix the panel by default (Sect. 4.1.1).

Graphs can be transported from one graphical or histogram window to another, from lists of variables and from lists of calculated variables (Sect.4.3.2.16). Any plotted histogram from a histogram window can be processed by a table processor (Sect. 4.3.7), and a statistic tool (Sect.4.3.8).

Select one or a group of variables in the container (except those, which have been transferred from the list of calculated variables, Sect. 4.3.3.3), then click the right mouse button on one of the selected variables. The pop-up menu appears (Figure 4.61).

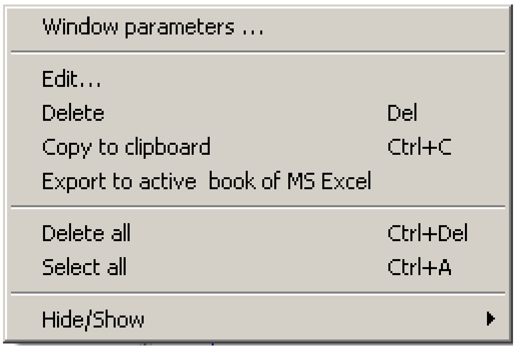


Figure 4.61. Pop-up menu for container

Pop-up menu for container (Figure 4.61) lets the following operations:

- **Window parameters** - calls the *Histogram parameters* window (Figure 4.62).
- **Copy to clipboard (Ctrl+C)** – copies the selected histogram to clipboard as text.
- **Copy to active book of MS Excel** – exports selected variables from the histogram window to MS Excel active book according to export settings (Sect. 4.1.4).

Histogram window parameters

Click the  panel tool button or the **Options** item of the pop-up menu (Figure 4.53) to change the window parameters.

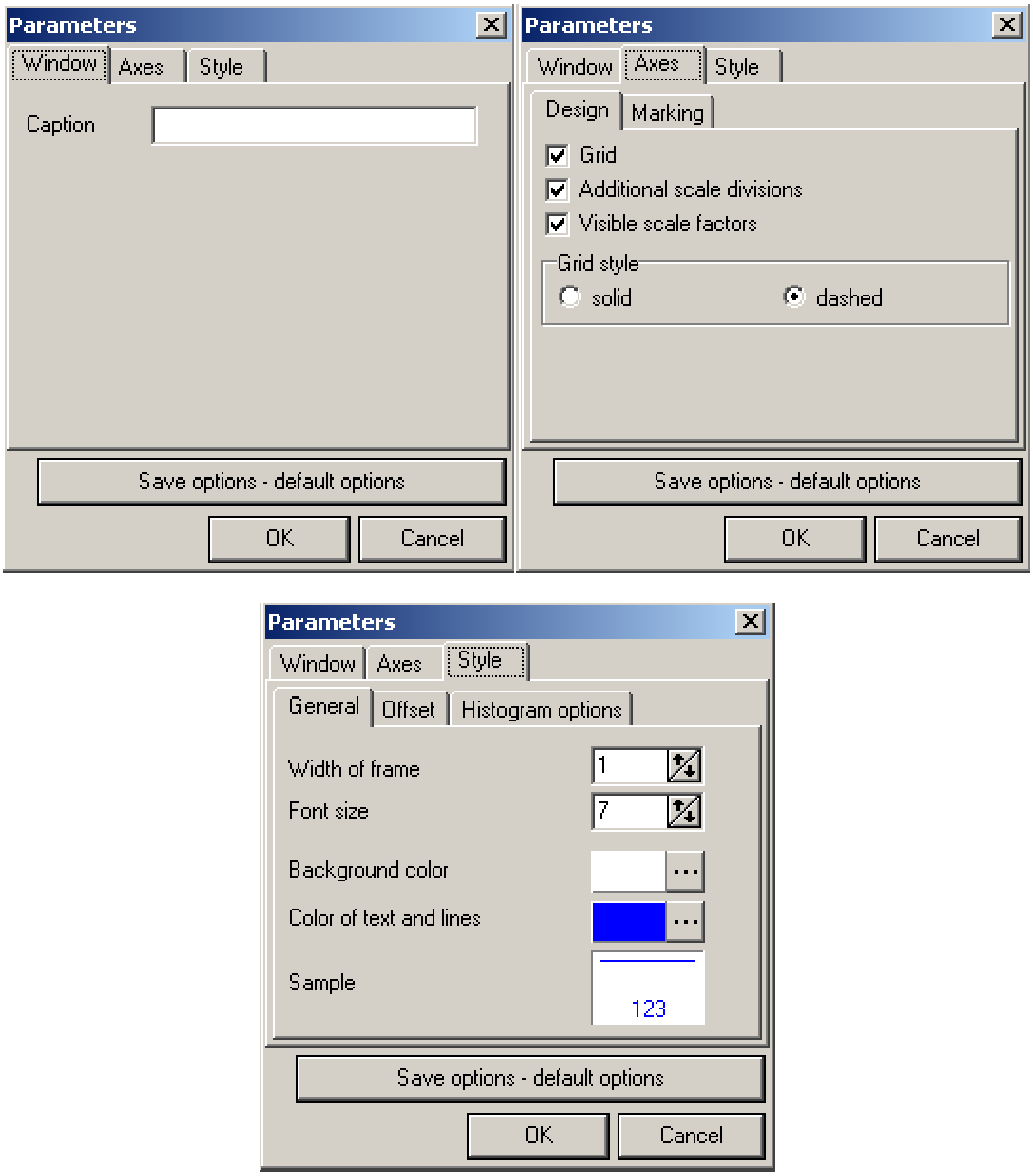


Figure 4.62. Histogram parameters window

The **Window** sheet includes:

- **Caption** – sets the window caption;

The **Axes** sheet allows following:

- **Grid** – sets the grid visible/invisible;
- **Additional scale divisions** – sets additional scale divisions;
- **Visible scale factor** – sets scale factors visible;

The sheet **Style** lets user to turn up graphic window style and set common style for all histograms.

4.3.6. Animation window in UM Simulation program

Animation windows are used for visualization of motion of objects. General information about the animation window can be found in Chapter 3.

Pop-up menu, mouse operations

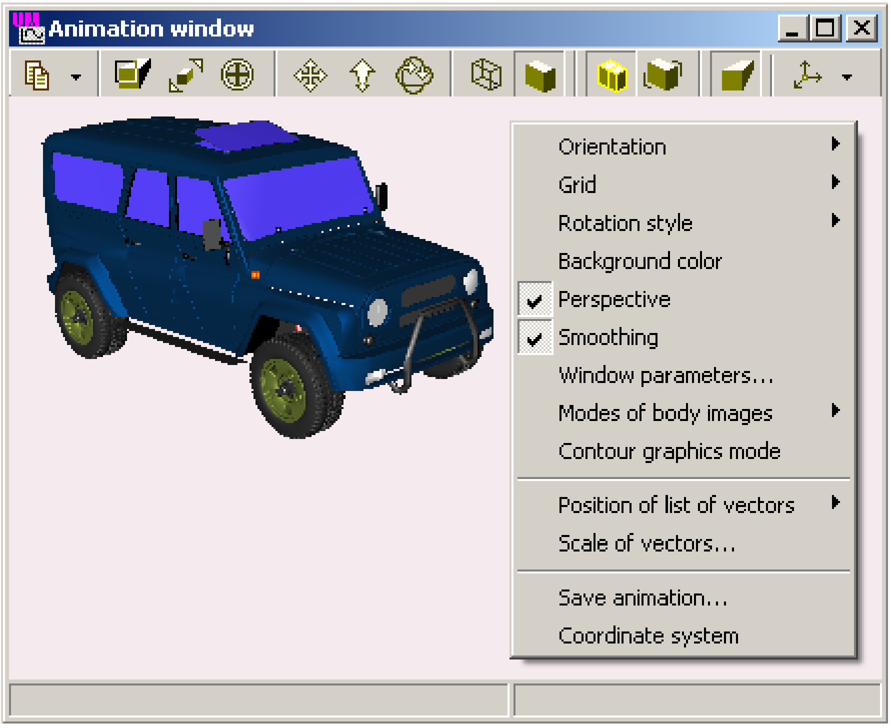


Figure 4.63. Animation window

Additional properties of an animation window in the UM Simulation:

- **Window parameters**

With the help of additional parameter – *Follow the body*, one can select the body from the list of object bodies. The option fixes the orientation of the camera according to the coordinate system of the selected body.

- **Modes of body images**

One can set visible/invisible or wired/solid modes for image of the body by checking/unchecking the body in the list (Figure 4.64).

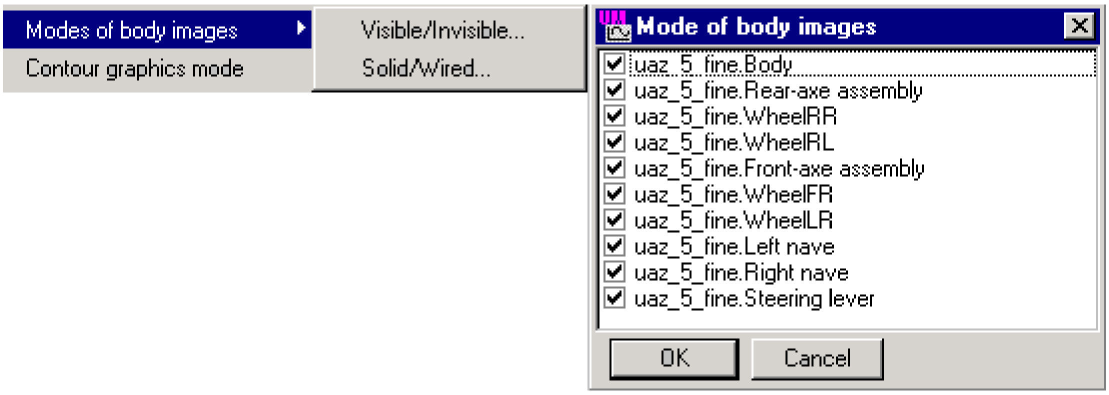


Figure 4.64. Settings of body visualization mode

- **Position of list of vectors**

The user can animate vectors by placing them to the vector list of the animation window. A vector variable can be defined with the help of *Wizard of variables* (Sect.4.3.2). Vectors list can be positioned in the left/right/top/bottom part of the window. Also, it can be hided.

- **Scale of vectors**

The user can select the scale of vectors, which animate different types of physical magnitudes. Size of vectors is defined with the help of scale coefficients and *Characteristic size* parameter (Chapter 3, Sect. *Object parameters and options | "General" tab*).

- **Save animation**

The user can save animation of the object motion in a file (Sect. 4.3.6.2).

If the mouse cursor stands over a body image, five additional items in the pop-up menu are available (Figure 4.65):

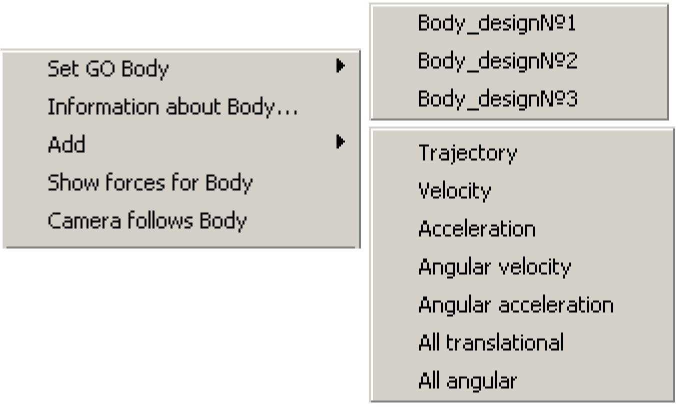


Figure 4.65. Additional options of pop-up menu

- **Set GO to Body** option lets you to select the body GO from the list of graphical objects. For example, if you have some variants of the body image, you can change GO in the UM Simulation (don't have necessity to start the UM Input program).

Information about the body

- **Add**: the command is used for animation of one or several vectors/trajectory for the point under the cursor (**Trajectory, Velocity, Acceleration, All translational** (Trajectory

+Velocity + Acceleration), **Angular velocity, Angular acceleration, All angular** (Angular velocity, Angular acceleration), Figure 4.65.

- **Show forces for body.** The command adds to the vector list of animation window vectors of all forces applied to the body.
- **Camera follows body.** The option fixes the orientation of the camera according to the coordinate system of the selected body.

Double click the left mouse button on

- a body image to get information about the body,
- a vector image to change its color (for simulation/XVA pause only),
- a trajectory image to change its parameters (for simulation/XVA pause only).

Move the mouse cursor over an element image or a vector to get information about it:

- name,
- current information (components of a vector, coordinates velocities and accelerations of a body point at the cursor position etc.). Table 1 contains useful key-modifiers (use the keys simultaneously with the mouse cursor positioning).

Table 1.

Key-modifiers for animation windows

Key	**Type of information about a body point under the mouse cursor**
No	Coordinates (SC0)
Ctrl	Velocity (SC0)
Ctrl+Alt	Acceleration (SC0) for simulation/XVA pause only
Shift	Coordinates (body-fixed SC)

4.3.6.1. Visualization of vectors and trajectories

Animation windows can be used for visualization of vectors (velocities, accelerations, forces) and trajectories of body points.

Use the *Position of list of vectors* item to make the list visible (Figure 4.65). Use the master of variables (Sect. 4.3.2) to add new vectors/trajectories to the list.

Double click on an element of the list to set its parameters.

Vector scales are specified according to their types

- Velocity (m/s);
- Acceleration (m/s^2);
- Angular velocity (rad/s);
- Angular acceleration (rad/s^2);
- Force (N);
- Moment (Nm)

Use the *Scale of vectors* item of the pop-up menu (Figure 4.63) to specify scales as well as vector image sizes.

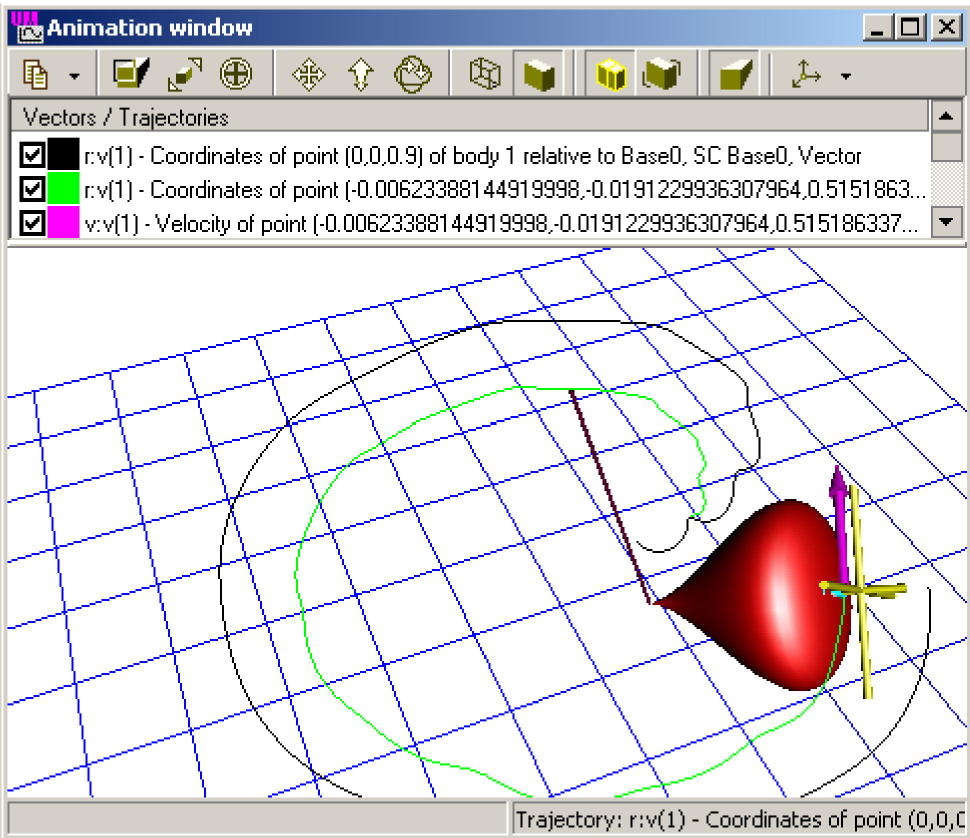


Figure 4.66. Trajectory and vectors in animation window

4.3.6.2. Creation of animation files

Simulation and XVA-analysis processes allow creating animation files as

- a set of BMP files;
- an animating GIF file;

- an animating AVI file.

Use the *Save animation…* pop-up menu item (Figure 4.63) and the *Copy of animation* dialog box (Figure 4.67) to choose a type of output file(s) and to specify some parameters.

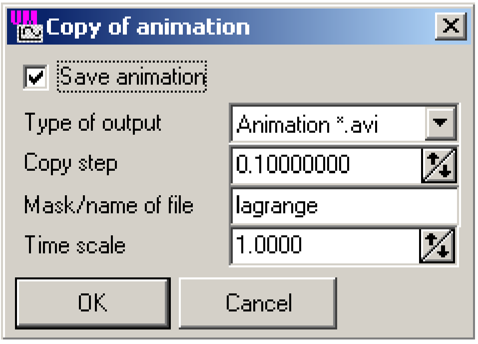


Figure 4.67. Copy of animation dialog

Check the *Save animation* key to create animation file(s).

The *Type of output* drop-down menu contains the following items:

- **Set .bmp** – a set of BMP files (frames) with the names [Mask]0.BMP – [Mask][Index of the last frame].BMP will be created in the object directory, the *Mask* should be entered in the *Mask/name of file* edit box;
- **Animation .avi** – an AVI file will be created; file name should be specified in *Mask/name of file* edit box;
- **Animation .gif** – an animated GIF file will be created; file name should be specified in the *Mask/name of file* edit box;

The *Copy step* parameter specifies the animation step in seconds.

Set the *Time scale* parameter to make the AVI animation faster (<1) or slower (>1) the real time.

Remark. Output files are located in the object directory. File(s) are created after breaking the simulation/XVA process in the pause mode (Sect. 4.4.2.1). Size of temporary files in some cases can increase tens or hundreds of megabytes. Before starting animation-writing process make sure that you have enough disk space.

4.3.7. Variable processor

Variable processor consists of two tools for variables processing: a table processor and a tool for transformation of variables. The processor is called by the **Tools | Table processor…** menu command, a *Table processor* window appears.

4.3.7.1. Table processor

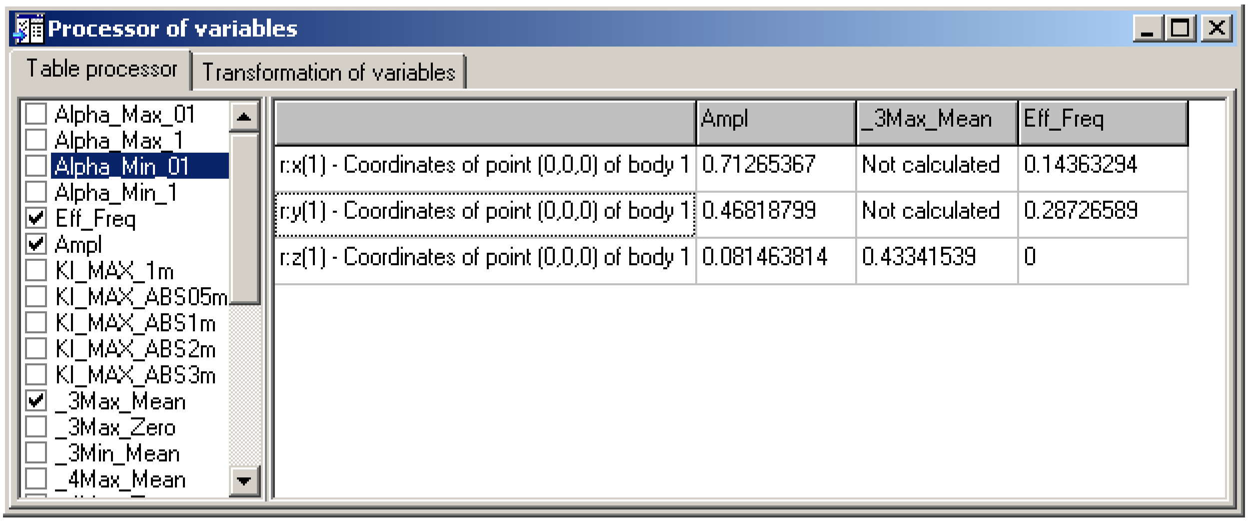


Figure 4.68. Table processor, dragging variables

In the left part of the *Table processor* window the list of all available functionals is located. One can check a functional or a group of functionals to process calculated variables. Calculated variables can be dragged into the table processor from a graphical window (Sect. 4.3.4) or from a list of calculated variables (Sect. 4.3.3.3).

Use the *Delete function* command of the pop-up menu or the **Shift+Delete** keys to remove a functional (a column of the processor). Use the **Delete** key to remove a variable (a row of the processor).

Use the popup menu (Figure 4.69) to

- clear list of functionals
- clear list of variables
- order a column by decrease/growth
- copy the results into the active MS Word document

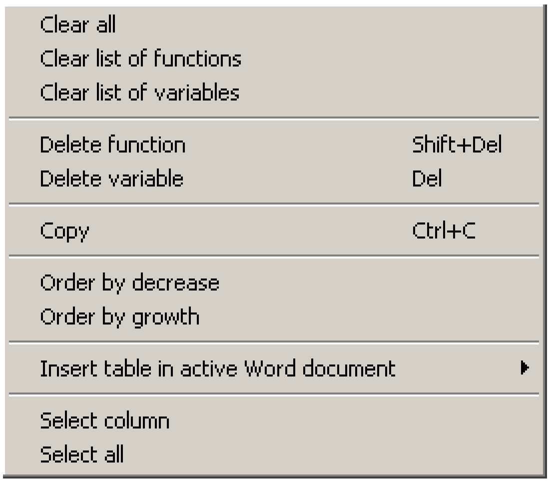


Figure 4.69. Popup menu of table processor

Table 4.1

List of the standard functionals (.\plugins\standard.dll, percentile.dll files)

Functional	Comment
Percentile_0_point_1 (0.1%) **Percentile_0_point_15 (0.15%)** **Percentile_1 (1%)** **Percentile_99 (99%)** **Percentile_99_point_85 (99.85%)** **Percentile_99_point_9 (99.9%)**	Percentile. A percentile is the value of a variable below which a certain percent of observations fall. So the 1st percentile is the value below which 0.1 percent of the observations may be found.
Integral	Integral of the variable taking into account the sign of the variable. Areas between a curve and abscissa axis are taken into account as positive for positive sections of the curve and as negative for negative ones. For the periodic functions like *sin(t)* for a integer number of periods this functional returns zero.
IntegralAbs	Integral of the variable without taking into account a sign of the variable. Areas between a curve and abscissa axis are taken into account as positive for both positive and negative sections of the curve.
LastAbscissa	The last value of abscissa
LastOrdinate	The last value of ordinate
Max	Maximal value
MaxAbs	Maximal absolute value
Max_Min	Difference between maximal and minimal values of variable **Max_Min = Max-Min**, double amplitude
Max_Min_2	Difference between maximal and minimal values of variable divided 2, **Max_Min = (Max-Min)/2**, amplitude

Mean	Mean value
Mean_plus_Std_Dev3	Sum of a mean value and triple standard deviation, **Mean_plus_Std_Dev3 = Mean + 3*Std_Dev**
Min	Minimal value
MinAbs	Minimal absolute value
Std_Dev	Standard or root mean square deviation, which is calculated according to the following formula $\sigma = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \bar{X})^2}{N-1}}$, where N is a number of points, $\bar{X}$ is the mean value.
Std_Dev3	Triple root mean square deviation, **Std_Dev3 = 3*Std_Dev**

Functionals are realized as export functions in DLL. The **standard.dll** and **percentile.dll** libraries contain the above functionals. The user can add own functional. Use the *..\com\Plugin.pas* file as a template and the *..\Plugins\standard.dpr* project as an example for creating a library (DLL) with new functionals. Then copy the new library into the *.\plugins* directory.

4.3.7.2. Transformation of variables

The tool **Transformation of variables** (Figure 4.70) is used for processing with calculated variables.

Example. There are two calculated variables in a graphical window. With the help of **Transformation of variables** instrument was formed the new variable as sum of the two variables (Figure 4.70).

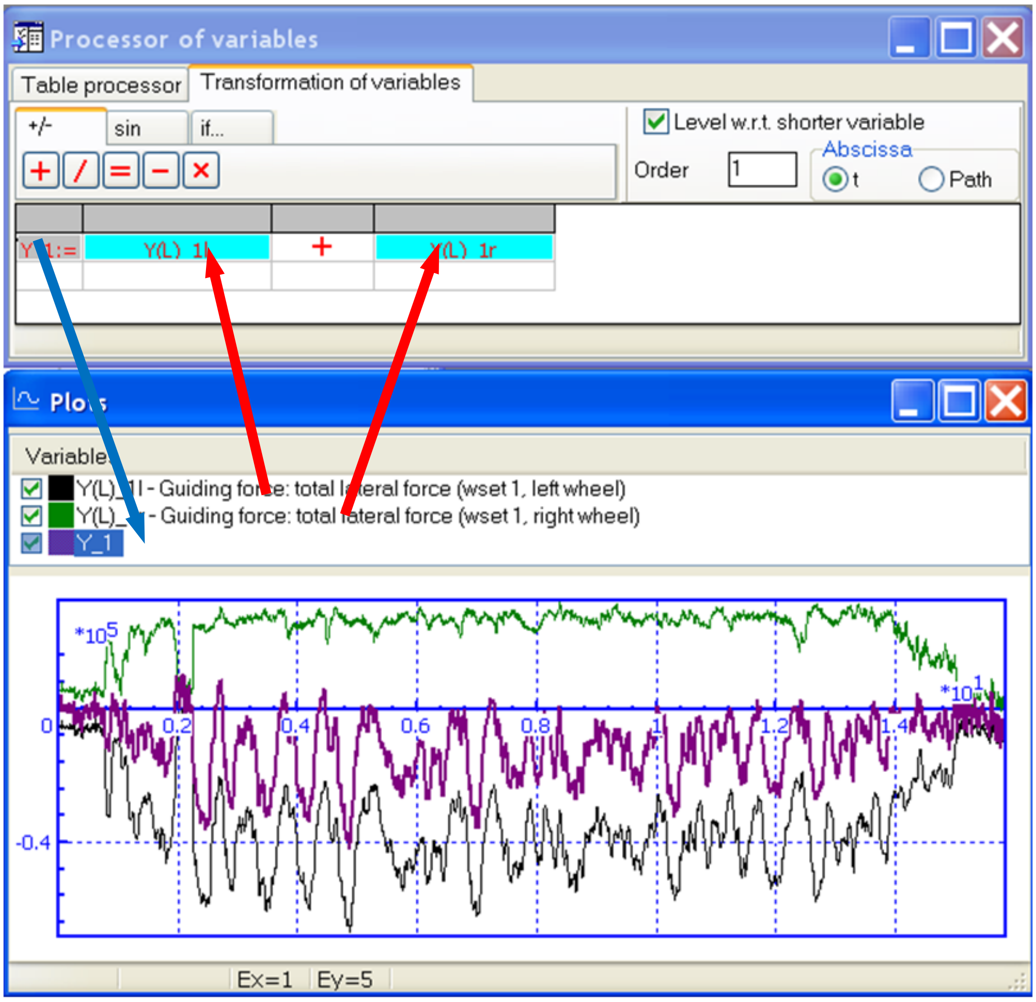


Figure 4.70. Operations with calculated variables

4.3.8. Statistics

The *Statistics* tool is used for statistical analysis of *calculated variables*. Use **Tools | Statistics** or the  button to open it. The number of statistical windows is unlimited.

The *Statistics* window is divided into two parts: the container of variables (on the left), the plot area (on the right).

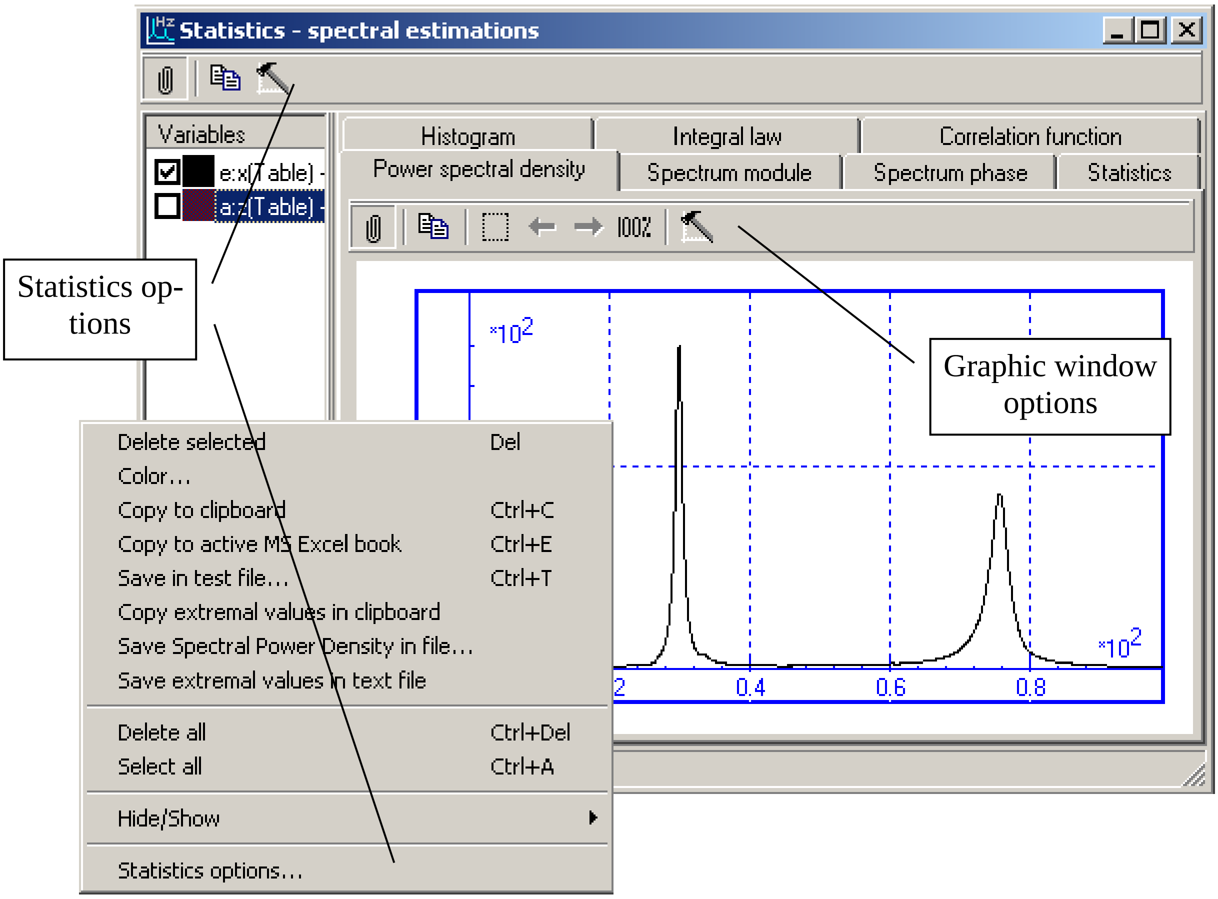


Figure 4.71. Statistics

The plot area contains a set of tabs with histograms, graphs and other characteristics. The following statistics are presented on the tabs:

- **Histogram** – histogram of probability density;
- **Integral law** – histogram of probability distribution;
- **Correlation function** – graphs of autocorrelation function estimate;
- **Power spectral density** – graphs of power spectral density (PSD) estimate;
- **Real and imaginary parts of spectrum** (modulus and phase) – complex spectral density of time series (spectrum).
- **Statistics** – table of statistics.

Histograms and graphs are drawn for active variables only (checked in the container of variables). Graphs are drawn in graphic windows situated on tabs (Sect.4.3.4).

Use the **Statistics options** dialog box (Figure 4.72, the *Statistics options* pop-up menu item or the tool panel button) for changing statistics parameters. Use the **General** tab to specify the format of numbers and histograms:

- **Number of columns** – number of columns of histograms (**Auto** – number is defined automatically).
- **Marks** – the switch turns on/off sampling probability values on histograms.

You can set fast Fourier transform (FFT) sample forming mode at the same tab:

- **Adding with zeroes** – sample is added with zeroes to the nearest greater value 2^n.

- **Cutting to nearest** – sample is cut to the nearest smaller value 2^n.

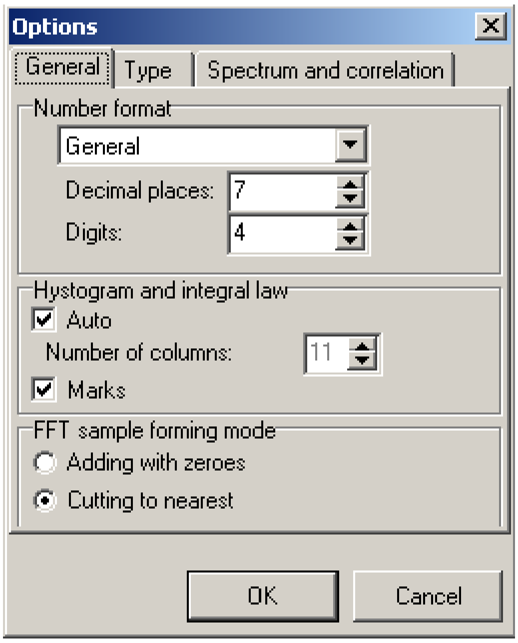


Figure 4.72. Statistics options

Type of complex spectral density (spectrum) representation as well as included statistics is specified on the **Type** tab. The complex spectral density is calculated with the direct FFT. The following model of the discrete Fourier transform (DFT) is realized

$$x(i) = \frac{1}{m}\sum_{n=0}^{m-1} a(n)\cdot e^{j\cdot 2\pi \frac{in}{m}};$$

$$a(n) = \sum_{i=0}^{m-1} x(i)\cdot e^{-j\cdot 2\pi \frac{in}{m}}.$$

The complex spectral density may be presented either by real and imaginary parts of spectrum or by its modulus and phase.

The **Spectrum and Correlation** tab is used for choice of window function, autocorrelation function estimate and the *m/N* ratio.

Graphs of power spectral density are one-sided power spectral density estimate, which uses different window functions. The next window functions are available: Bartlett, Hanning, Parzen, Hamming and periodogram.

The algorithm of the FFT is used for calculating of the DFT.

Both biased and unbiased estimates are used for autocorrelation function.

In the **m/N** ratio we have m as the number of points of autocorrelation function estimate, which are used for computing the PSD, and N as the sample size.

4.3.9. Force element response in the frequency domain

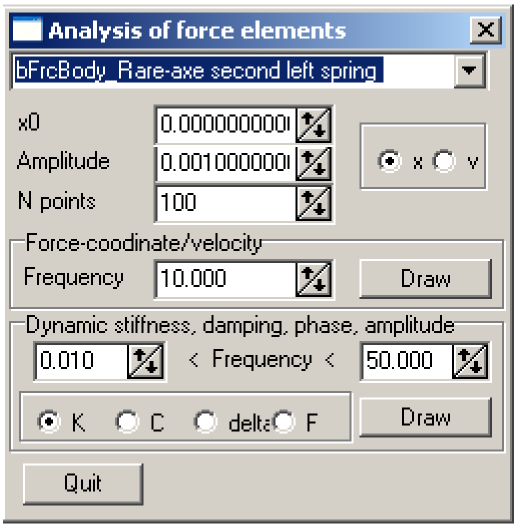


Figure 4.73. Force element analyses window

See the mathematical description to this section in Chapter 2, *Force element response in the frequency domain*.

Use the **Tools | Force analysis** menu item to call the corresponding window.

Select a force element for analysis with the help of a pull-down menu. If the element is a list of forces, both the whole force is available and every its component are available for analysis.

Set excitation parameters:

- center of excitation x_0 (m)
- amplitude *a* (m)
- number of points on a plot
- To get the response force versus the coordinate x or velocity v
- select a type of abscissa variable (*x*,*v*)
- set an excitation frequency in the **Force-coordinate/velocity** group (Hz)
- select the **Draw** button in the **Force-coordinate/velocity** group to get the plot

To get characteristics of the force in the frequency domain (dynamic stiffness, equivalent damping, dissipation angle, response amplitude versus excitation frequency).

- set lower and upper frequency (Hz)
- click on the lower **Draw** button

An example for a force element analysis can be found in Chapter 7, Sect. *Elastic-friction element 2*.

4.3.10. Control panel

4.3.10.1. Use of control panel

A control panel gives an opportunity to vary model parameters simultaneously with the simulation process. The user can effect to the model behavior interactively during the simulation of motion. For example, control panel allows taking into account human-driver actions for cars, locomotives, robots and manipulators.

With the help of the control panel (Figure 4.74) the user can drive the transport robot (see sample model {UM Data}\SAMPLES\robots\atr): select direction of the motion (left, right, forward, backward) and control the value of driving moment at the wheels. One can also set the amplitude of irregularities and the friction coefficient between wheels and the surface.

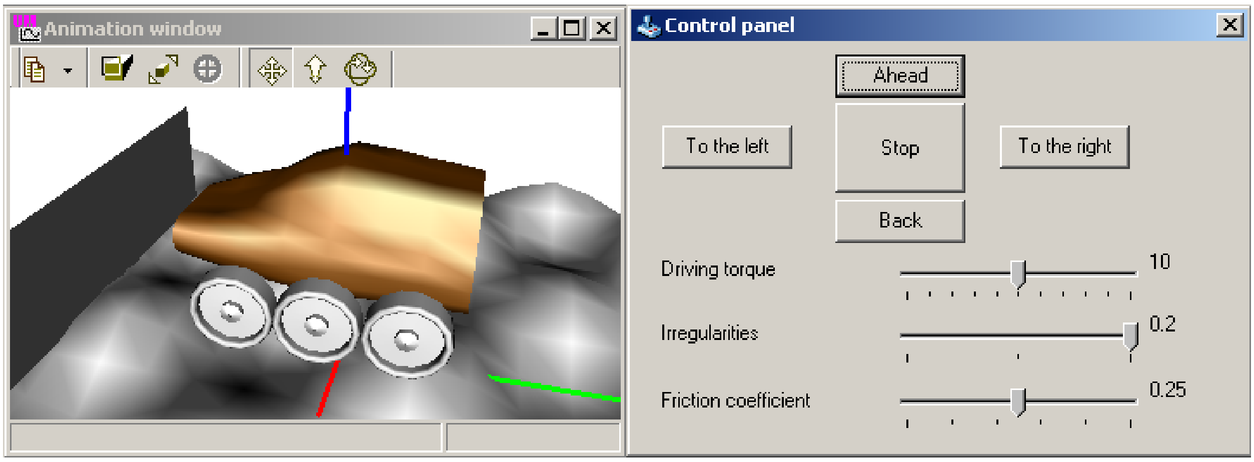


Figure 4.74. Control panel (model …\demo\ Atr).

Control panel can be created with the *Control panel editor* (Sect.4.3.10.1).

Use **Tools | Control panel …** or  **button** to open an existing control panel (*.cp).

The user can work with any number of *Control panel editor* windows at one time, but the only *Control panel* can be opened.

Control panel can be assigned with identifiers that parameterize force elements of any kind and graphical objects. Identifiers that parameterize properties of bodies (masses, moments of inertia, position of center of mass) and joints cannot be used within control panel.

4.3.10.2. Control panel editor

Use **Tools | Control panel editor…** or  button to start the visual constructor – *Control panel editor* (Figure 4.76). Constructor permits to set different control elements at the panel and connect them with identifiers.

To add control element to the control panel click the element icon at the *Control panel editor menu*. Then click at the *input panel* of the window. New element will be added at the pointed position.

The control panel can include the following elements, see Figure 4.75:

- static text;
- edit box (the field for numerical input);
- check box;
- track bar;
- button;
- radio group;
- biaxial joystick.

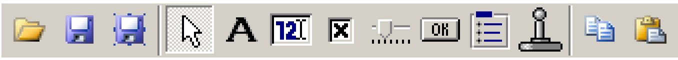


Figure 4.75. Menu of control panel editor

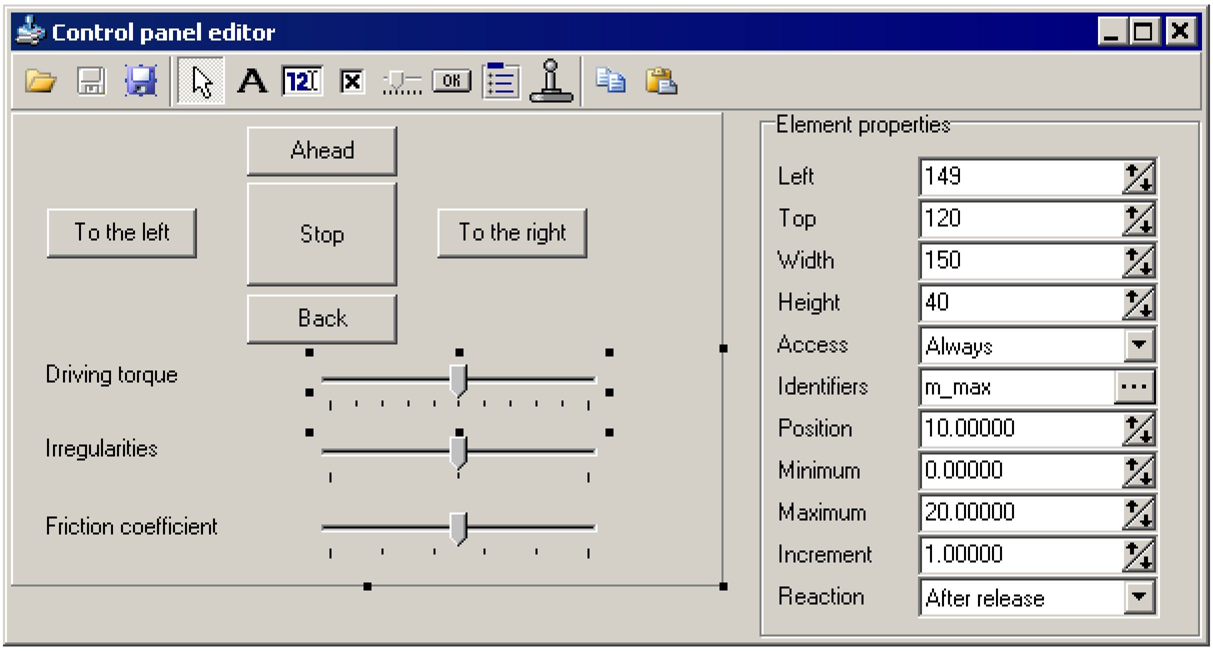


Figure 4.76. Control panel editor

Let us study properties of control elements. The following group of properties is common for all elements.

- **Left, Top** describe the position of control elements;
- **Width, Height** describe sizes of control elements.

All elements, besides the static text, have the following properties:

- **Access** regulates accessibility of the element. Available parameter values are **«Always», «Before simulation», «While simulation only», «In pause mode»**.
- **Identifier(s)** sets one or several identifiers, which are connected with the control element. Identifiers are selected from the identifier list of the object. Specific realization depends on the type of the control element.

Specific element properties:

Static text

- **Text** is element text;
- **Word wrap** switches the wrap mode **(Yes / No)**;
- **Align**. Available values: **«Left», «Right», «Center»**.

Edit box

- **Value** is the current (initial) identifier value;
- **Maximum, Maximum** describes enabled interval; set zero for both **Maximum** and **Maximum** to ignore the limits;
- **Increment** is step value for increasing / decreasing of the identifier (only for visual changing by the  button).

Check box

- **State** is the current (initial) state of the switcher **(On/Off)**;
- Use **Identifiers** to describe identifiers and their values for on/off modes.

Track bar

- **Minimum, maximum** are limits of identifier changing;
- **Reaction** sets mode of changing the identifier(s). Available values are **«Always»** and **«After release»**. **«Always»** supposes that the identifier value is changing during the process of track bar changing; **«After release»** supposes that new value of the identifier is set after the track bar is released.

Radio group

- **Values** describes relations between names in the list and values of the identifier (Figure 4.77);
- **Title** describes the title of the radio group;
- **Columns** is the number of columns in the list;
- **Active** is the index of the active string (identifier value), -1 if string is not selected.

Note that identifiers that are connected with the **Check box** or **Radio group** cannot possess the arbitrary values when the control panel is active. The only allowed by **Check box** or **Radio group** values can be assigned. If you will change their values manually via **Object simulation**

inspector (see Sect. 4.4.1) the identifiers will be rounded automatically to the nearest allowed values.

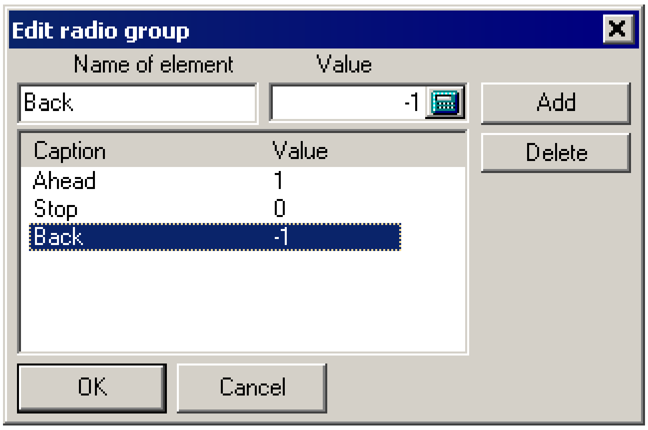


Figure 4.77. Edit radio group

Biaxial joystick

- **Hor.[izontal] axis** describes the identifiers that will be connected with horizontal joystick displacement, [-1..1].
- **Vert.[ical] axis** describes the identifiers that will be connected with vertical joystick displacement, [-1..1].
- **Start/Stop** describes the identifier that reflects the joystick mode (start/stop). **Start** value is set on start of joystick moving. **Stop** value is set on stop of joystick moving. It is usually used to turn off the automatic brake on start moving the joystick and turn it on again on stop moving the joystick for robots, manipulators and similar models.

Note. You can omit any identifier (for horizontal and vertical axes and start/stop identifier). If you will omit an identifier for horizontal or vertical axes your biaxial joystick becomes the single-axial one.

4.3.11. Identifier macros

Identifier macros are a tool for assignment of numeric values for a group of identifiers by one operation.

In every UM model, the user can create a list of macros. A macro contains a group of identifiers selected by the user as well as a table of identifier values. A column of the table has a name and includes numeric values for each of the identifiers from the group.

The list of macros created by the user is stored in the standard text file with the name *Macros*. The file is read automatically by the loading of the model into UMSimul program.

Before start of the integration process, the user can assign the desired values of identifiers by selecting a macro and the table column by its name, Sect. 4.3.11.

To create the macros, a window for macros editing is used, Figure 4.78. Select the **Tools | Identifier macros** menu command to call the window. By the first open of this window, the identifier choice window appears first, where the user may select a group of identifiers for the first macro, Figure 4.79.

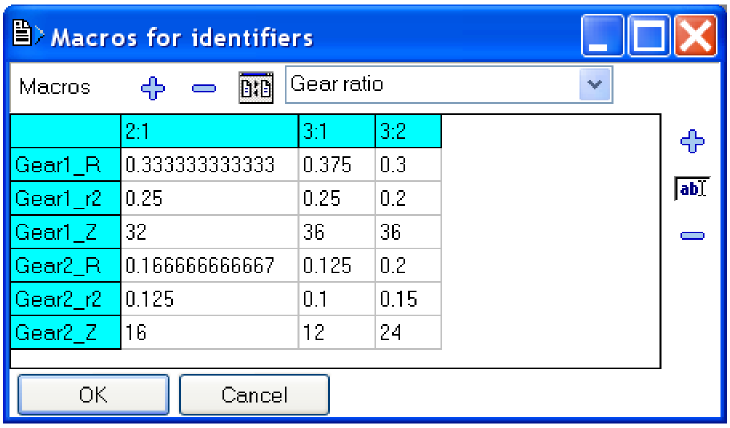


Figure 4.78. Window for macros editing

The tools in the top of the window are used for adding, deleting, renaming of macros as well as for selection of the current one.

The button adds a new macro. At the same time the window for selection of a group of identifiers appears, Figure 4.79.

The button deletes the current macro.

The button calls the window for editing the identifier group in the current macro, Figure 4.79.

The drop-down menu contains the list of macro names for selection of the current macro. In the edit box, the name of the current macro can be set.

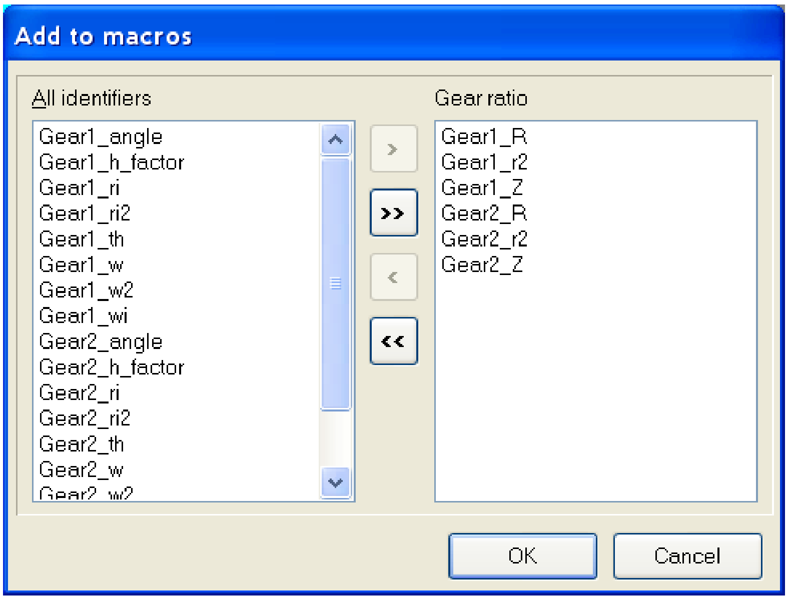


Figure 4.79. Window for selection of identifier group

The right tool panel contains buttons for editing the table of the identifier values of the current macro.

The button adds a new column to the table.

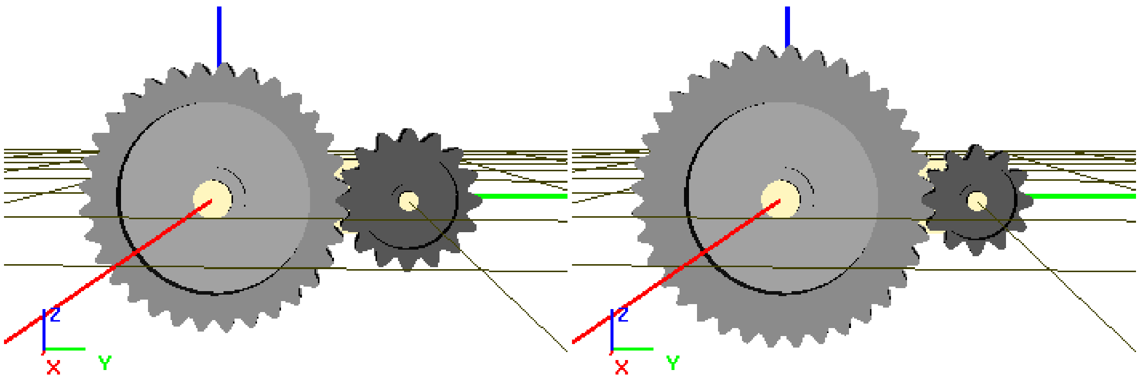
The button calls the window for editing the column name.

The button deletes the current column.

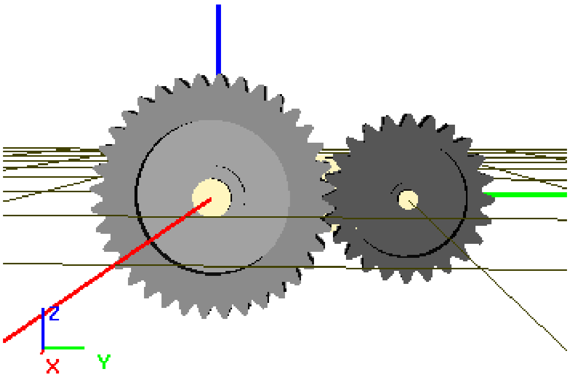
Examples:

{UM Data}\SAMPLES\LIBRARY\Planetary gear. The model contains one macro with three columns of identifier values, see. Figure 4.78. The macro contains identifiers, which parameterize the number of teeth as well as some geometric parameters of gears.

{UM Data}\SAMPLES\LIBRARY\ElastFriction



2:1 3:1



3:2

Figure 4.80. Change of the gear ratio with the marco

4.4. Integration of equations of motion (single mode)

The numerical integration is one of the basic tools for analysis of the mechanical system behavior. One can use single mode or multi-variants mode of motion simulation. Multi-variant mode is available if the **UM Experiments** module is available in the current UM configuration.

The user can choose a numerical method and accuracy of the integration, the results plotting and saving step. It facilitates the supervision of the object motion with the help of as many animation windows as necessary. One can easily copy information displayed in the graphical windows, interrupt integration at the instant of concern, influence the integration process in an interactive manner by processing messages in the control file.

Here we consider the mode of single integrations. Select the **Analysis | Simulation…** menu item to start this mode. The *Object simulation inspector* appears (Figure 4.80).

The inspector allows the user to specify some parameters before the simulation process starts:

- to choose a numeric method and its parameters;
- to change initial conditions (coordinates and their time derivatives);
- to modify identifier values;
- to assign a list of automatically calculated variables (Sect.4.3.2.16).

4.4.1. Preparing for integration

The **Object simulation inspector** (Figure 4.81) is used for preparing the mechanical system to the integration process.

4.4.1.1. Solver

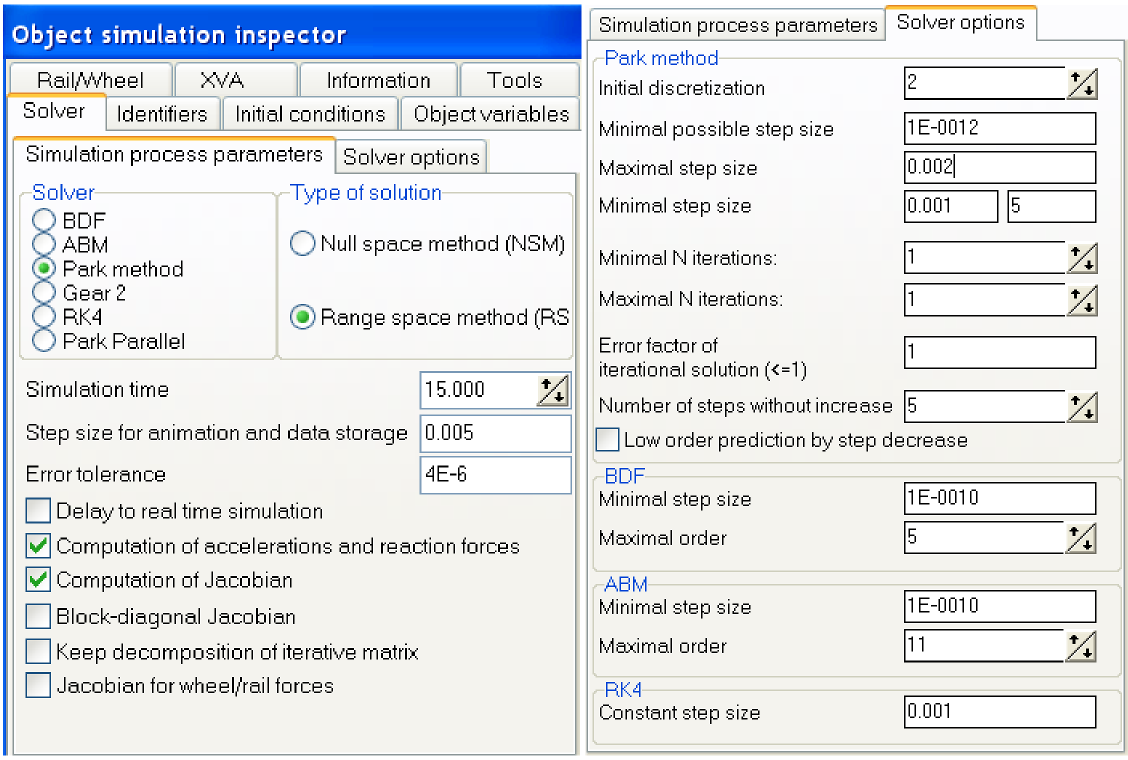


Figure 4.81. Solver parameters

Various numerical methods for solving equations of motion are implemented in UM. First consider the solution of nonlinear equations.

If the object has closed kinematical loops, the equations are ordinary differential-algebraic (DAE). The following methods are available for integration of both ODE (ordinary differential equations) and DAE:

- **BDF** (the Backward Differentiation Formula) is an explicit PEC (i.e. prediction-evaluation-correction) solver; up to 5^{th} order, with a variable step size and order; only for non-stiff equations.
- **ABM** (the Adams-Bashfort-Moulton method) is an explicit PECE (prediction-evaluation-correction-evaluation) solver; up to 11^{th} order, with a variable step size and order; only for non-stiff equations.
- **PARK** is an implicit solver of the second order with a variable step size; only for stiff ODE and DAE.
- **GEAR 2** is an implicit method of the second order with a variable step size.
- **RK4** is the 4th order Runge-Kutta method with a constant step size. The solver is not used for DAE.

- **Park parallel** is the solver based of the Park finite-difference methods, which uses a special method for generation of equations of motion and realizes the efficient parallel computations of multi-core processors.

If the equations of motion are non-stiff, both the DBF and ABM methods have the similar efficiency for not very high integration accuracy. For integration with a high accuracy, the ABM method is more efficient.

If the equations of motion are stiff, the PARK method is much more efficient both for ODE and DAE, especially if the *Calculation of Jacobian matrices* option is turned on.

Usually, the BDF and ABM methods are more efficient than RK4 for non-stiff ODE, and the Park method is faster than RK4 for stiff ODE.

The most efficient methods for industrial applications such as rail, road, tracked vehicles, and hybrid models and so on, the Park and Park parallel solvers are proved to be the most efficient ones. Consider these methods in more details.

4.4.1.2. General solver parameters

The *Solver* tab allows specifying the following parameters:

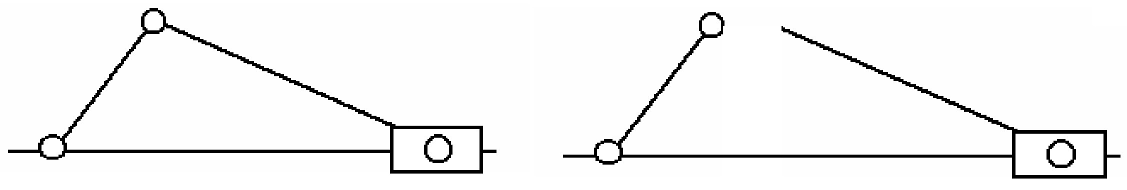
- **Solver:** the group box specifies an integration method
- **Simulation time.** The simulation time can be changed during the integration process.
- **Step size for animation and data storage** sets the time step size for refreshing animation windows, adding new points in graphic windows, and computing variables in the lists of variables. The recommended value is 0.005 s. Note that the default value is 0.02s.
- **Error tolerance** is the accuracy parameter.
- For the *BDF* and *ABM* methods: this is the absolute error tolerance corresponding to the time interval of 1s. The step size is automatic chosen to ensure the given relative error *e* of the calculated coordinate values. For the PARK methods: the local absolute error tolerance, therefore *the Park method requires higher value of the accuracy parameter* (e.g. $10^{-5} \div 10^{-7}$).
- The **Delay to real time simulation** option is activated if simulation is faster than real time, and the user wants to see the motion in the real time in an animation window.
- **Computation of accelerations and reaction forces** key is usually checked. In this case acceleration and reactions are evaluated. Sometimes for real-time applications these values are not important, and the option can be unchecked.
- The **Type of solution** group: the Null-Space Method **(NSM)** or the Range-Space Method **(RSM)**

This parameter is valid for systems with closed loops only. The user must test both methods, and choose the fastest one.

Additional information about NSM and RSM

The Null-Space Method (NSM): the Lagrange multipliers are eliminated from the motion equation using transfer to the local generalized coordinates. The Range-Space Method (RSM): at the first step the Lagrange multipliers are calculated and then the coordinates.

Both RSM and NSM have a disadvantage, which can be clarified using a model of the crank-and-slide mechanism (Figure 4.82a).



a) b)

Figure 4.82. Slider-crank mechanism

Suppose the mass of one of the bodies is much less than the mass of the others. You set it to zero. The mechanism has a closed loop and one joint is cut (Figure 4.82b). The mass matrix of the mechanism is block-diagonal and one of the two blocks is singular! That means the inverse matrix cannot be calculated. If you try to do this, you will receive a message that the mass matrix is singular. *Of course, to enter zero masses is not recommended!*

4.4.1.3. Solver: Park method

Recommended values of parameters for the Park solver are show in Figure 4.80 right. For getting precise computation, it is recommended to set zero value for the **Minimal step size**.

If equations of motion are stiff and/or error tolerance is not small enough, the Park method shows instable solutions. As a rule, if a solver is instable, plots of some of accelerations include large oscillations with a high frequency. In such cases, it is recommended to do the follows:

- activate use of **Jacobian matrices (JM)**,
- set smaller **error tolerance**,
- set **Minimal step size** to zero.

Use of **Jacobian matrices** leads to a considerable acceleration of simulation process in the following cases:

- low speed of rail vehicle (less 8-10 m/s); in this case the **Jacobian for wheel/rail forces** key should be activated; *this method helps in the case of motion with a nearly constant speed*;
- the model includes stiff forces, i.e. forces with large gradients due to big stiffness and damping coefficients; examples: contact forces, force element with a successive connection of spring and damper (the viscous-elastic force element).

If computation of **Jacobian matrices** is on, simulation process can be often made faster with the help of **block-diagonal Jacobians** and switching off computation of **Jacobian matrices** for non-stiff forces such as suspension springs or dampers on the **Tools | Forces** tab of the **Object Simulation Inspector**, Figure 4.83.

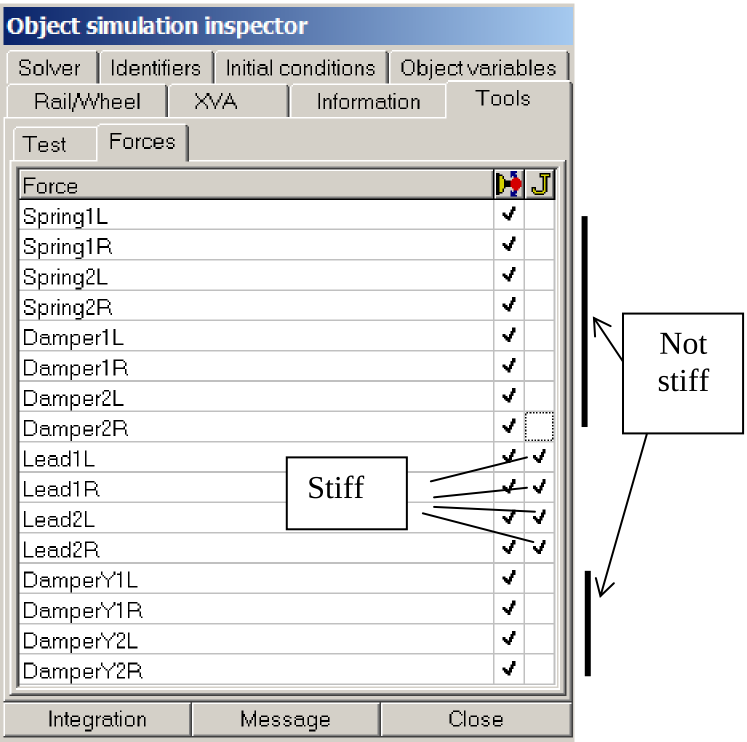


Figure 4.83. Switching on/off evaluation of JM for force elements

Keep decomposition of iterative matrix option is not used for simulation of rail, road and tracked vehicles.

Remark. The main criterion for use of **Jacobian matrices** is very simple. If simulation with this option is faster and stable, it should be used. It is recommended to optimize parameters and options of the solver for each of the models.

4.4.1.4. Solver: Park Parallel

The new solver **Park parallel** is implemented in UM 6.0. In fact, this method is a combination of a special algorithm for numeric-iterative generation of equations of motion, and the Park solver for stiff ordinary differential equations. One of the main features of this method is use of multi-core processors for parallel computations, which make simulation faster in many cases.

To estimate the efficiency of the solver for a computer with multi-core processors, we recommend studying the following standard UM models:

{UM Data}\SAMPLES\Tracked_Vehicles\gsTV (tracked crawler);
{UM Data}\SAMPLES\Tracked_Vehicles\m1a1 (tank);
{UM Data}\SAMPLES\Tracked_Vehicles\fh200 (tracked excavator);
{UM Data}\SAMPLES\Rail_Vehicles\tgv (TGV train with ten vehicles);
{UM Data}\SAMPLES\Rail_Vehicles\heavyhaultrain (freight train with ten vehicles);
{UM Data}\SAMPLES\Rail_Vehicles\simple_18_100 (simplified model of a freight car with three-piece bogies 18-100).

It is recommended to compare the simulation rate for different number of parallel threads. For train models as well as for the freight car model, it is useful to compare the Park Parallel and Park solvers. Minimize or close all animation windows during simulation tests.

4.4.1.4.1. Conditions for use of Park Parallel solver

The Park parallel can be used if some requirements are met.

- Numeric iterative method of generation of motion equations must be set for the vehicle model in the **UM Input** program, Figure 4.84.
- Mass and moments of inertia relative to the X, Y, Z axes for all of the bodies must be positive.
- Using Park Parallel for simulation of hybrid models, which include flexible bodies (UM FEM) and interfaces with Matlab\Simulink is not supported.
- Park Parallel does not support simulation of road vehicles, mates and some other elements.
- 3D contact elements are ignored (see Chapter 3 of the user’s manual).

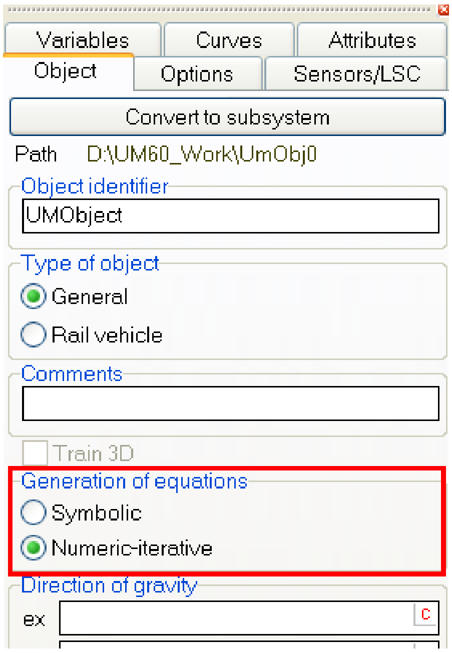


Figure 4.84. Numeric-iterative method of generation of equations; the Object tab of the inspector in UM Input program

If the model does not satisfy these requirements, the corresponding message appears, and the list of violations is stored in a text file, Figure 4.85.

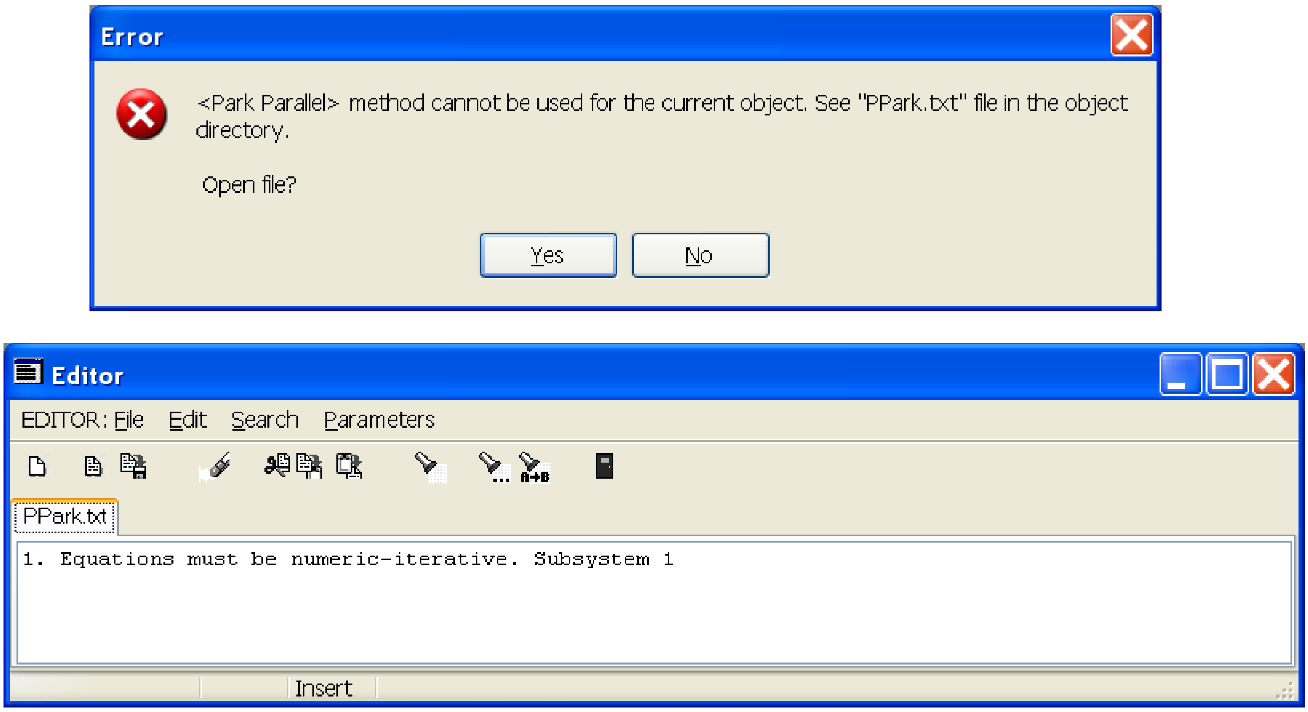


Figure 4.85. Message about impossibility of use the Park Parallel, and list of violations

4.4.1.4.2. Solver parameters

CG iterations (use of iteration of conjugate gradient method)

The CG method is used for more precise solving stiff equations. If the key is active, the full JM are used otherwise the block-diagonal JM are applied. There are no exact recommendations for use of CG iterations. The user should follow the simple method: the key is activated if the simulation becomes faster.

If the CG iterations are used, the **CG error** must be specified. As a rule, results are good even for low accuracy 0.1.

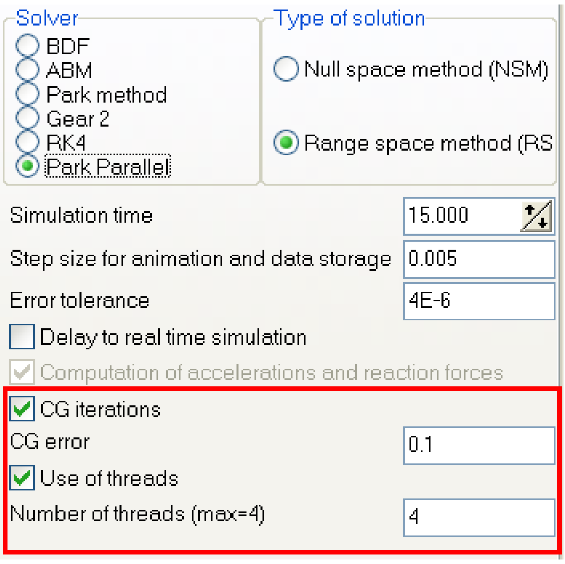


Figure 4.86. Parameters of solver Park Parallel

Use of threads

Threads can be used for computers with multi-core processors. The number of threads for parallel computations cannot exceed the maximal number of processor cores (physical and logical).

Optimal number is determined by the experience.

Nowadays, four- and eight-core processors are the most efficient. In some cases, use of parallel computations improves the solver performance by factor 2.5-3.

4.4.1.5. Changing values of identifiers

To change the values of identifiers use the *Identifiers* tab shown in Figure 4.87.

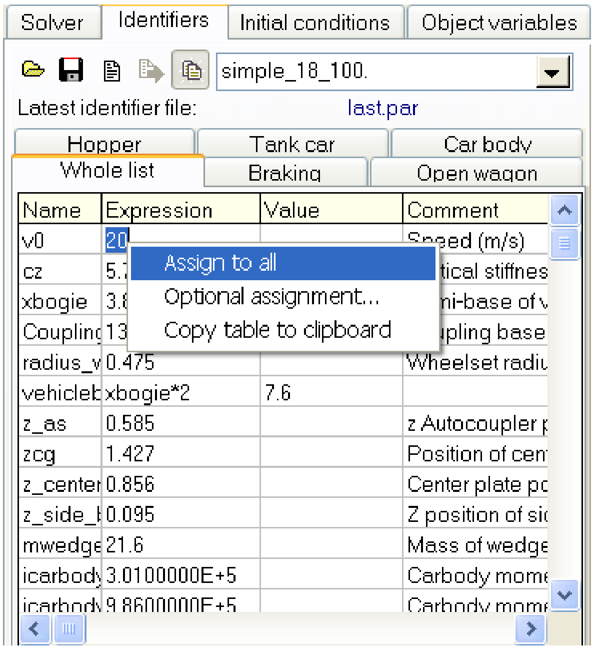


Figure 4.87. Object identifiers, the tab and the pop-up menu

Use the second column to change the identifier value. **Press Enter after modification!**

The current identifier values can be saved with the help of the buttons:

save to file (*.par);

save to text file (*.txt).

The button allows assignment values to a group of identifiers if macros are created for the current model.

The button switches the modes of immediate/postponed refresh of the object elements after change of numeric value of an identifier. The refresh is recommended to be postponed if it is made too slow in case of large models.

The button reads identifier values from a file *.par. Identifiers can be read from a *.par file of any model. Before changing the current values of identifiers, an auxiliary window appears, Figure 4.88. The left part of the window contains the list of identifiers with the different old (in brackets) and new values of identifiers. Uncheck some identifiers from this list to cancel the assignment of values from the file to these identifiers. The right part of the window contains identifiers from file, which are not found in the list of the current model.

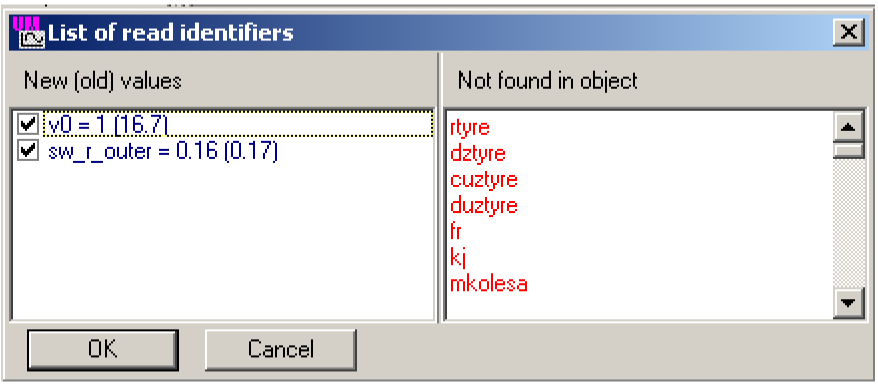


Figure 4.88. Adding identifiers from file (*.par).

The pop-up items of the identifier list:

- **Assign to all** – assigns the value of the selected identifier to all identifiers with the same name in other subsystems of the object.
- **Optional assignment** – assigns the value of the selected identifier to some (checked) identifiers with the same name in other subsystems of the object.
- **Copy table to clipboard** – copies the current group of identifiers to clipboard

Remark. It is not allowed changing dependent identifiers (identifiers-expressions).

Changing values of identifiers of the same name in subsystems

If the user changed the value of an identifier and subsystems of the model have identifiers of the same name, a special window with the list of these identifiers appears, Figure 4.89. The user may assign the new value to the selected identifiers from the list. In the case shown in Figure 4.89, the new value 82000 will be assigned to all identifiers.

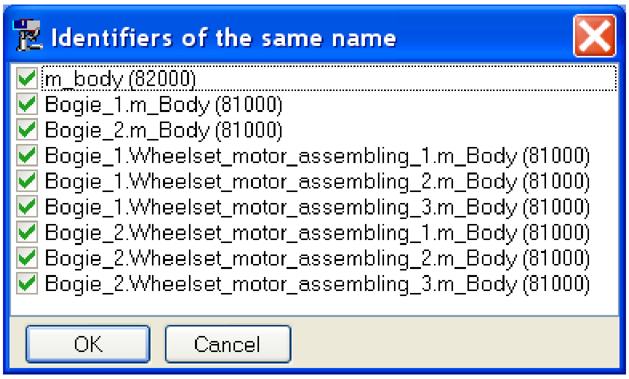


Figure 4.89. Example of a list of identifiers of the same name

Access to identifiers in subsystems

To change identifiers in subsystems of the current model, the user should select the subsystem in the pull-down tree located in the top of the tab, Figure 4.90.

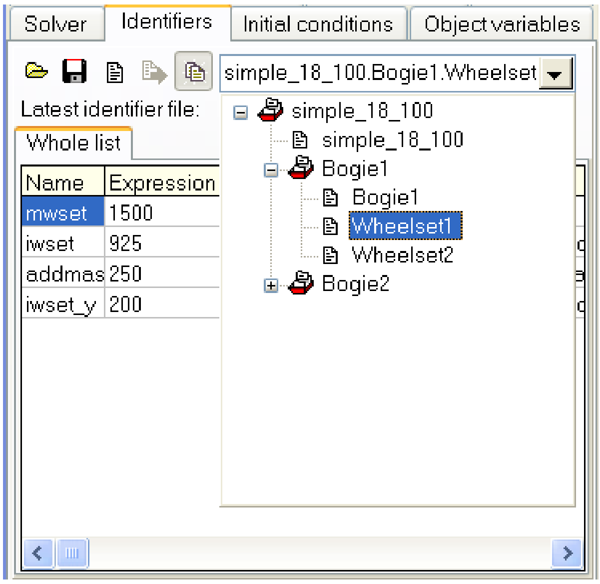


Figure 4.90. Selection of identifiers in a subsystem

Assignment identifiers by macros

If the current UM models contains a list of identifier macros developed by the user (Sect. 4.3.11), the button become enabled, and the user get access to the menu with macros by click on this button, Figure 4.91. Selection of a menu command leads to the change of values for an identifier group.

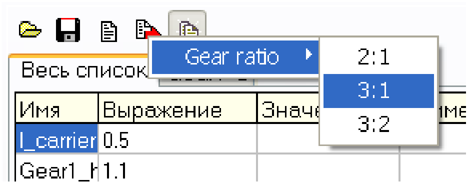


Figure 4.91. Setting identifier values by macros

Automatic save of identifiers

By loading a model in the **UM Simulation**, parameters of the model are read from the *last.par* file, if this file is presented in the directory of the model. If the *last.par* file not found, the parameter values are the same as in the **UM Input** module.

If the option for automatic saving of the latest identifier values is active (see Sect. 4.1.2), the *last.par* file is created or replaced automatically by every close of the model in the **UM Simulation**.

Note that even if you change some identifier values in the **UM Input**, but the *last.par* file exists in the directory of the object, the new values of identifiers will be replaced by the values from the last.par file. To assign the identifier values from *input.dat* file, the *last.par* file must be removed or renamed.

4.4.1.6. Choice and automatic calculation of the initial conditions

4.4.1.6.1. General notions

Choice of the initial values of coordinates and velocities is an important part of preparing the object for simulation. This operation is very simple for systems with a tree structure because the set of Lagrangean coordinates is not redundant and the coordinates are independent with an exception of models with quaternion joints. If the system has closed loops, the coordinates are dependent and some constraint equations must be satisfied both for coordinates and velocities. The algebraic constraint equations are non-linear and their solution is nontrivial.

Let us introduce some notions.

An object has redundant coordinates if it has closed kinematical loops or quaternion joints. In this case the object coordinates cannot be arbitrary numbers, i.e. they are *dependent* and satisfy some algebraic equations (*constraint equations*). As a rule, the constraint equations are nonlinear and difficult to solve (even numerically). If an object has closed loops, some joints are *cut*, and the constraint equations are the joint closure conditions. Local coordinates in the non-cut joints are *generalized* coordinates, whereas coordinates in cut joints are *auxiliary* coordinates. Constraint equations are almost always transcendental equations, therefore, initial values of coordinates are unknown a priory. They can be only calculated numerically. Automatic calculation of initial conditions means numerical calculation of coordinates and their time derivatives satisfying constraint equations.

UM makes it possible to automate the process of calculation of the initial conditions satisfying the constraint equations using the Newton-Raphson iterations. Coordinate values entered by the user are used as initial approximation. The situation is not trivial because the non-linear equations may have several solutions or none. Moreover, the solution may not be found if the initial approximation set by the user is far from the exact values. The calculation of the initial conditions is nevertheless possible if the recommendations below are followed and the problem is stated correctly.

4.4.1.6.2. Window for assignment initial coordinate values

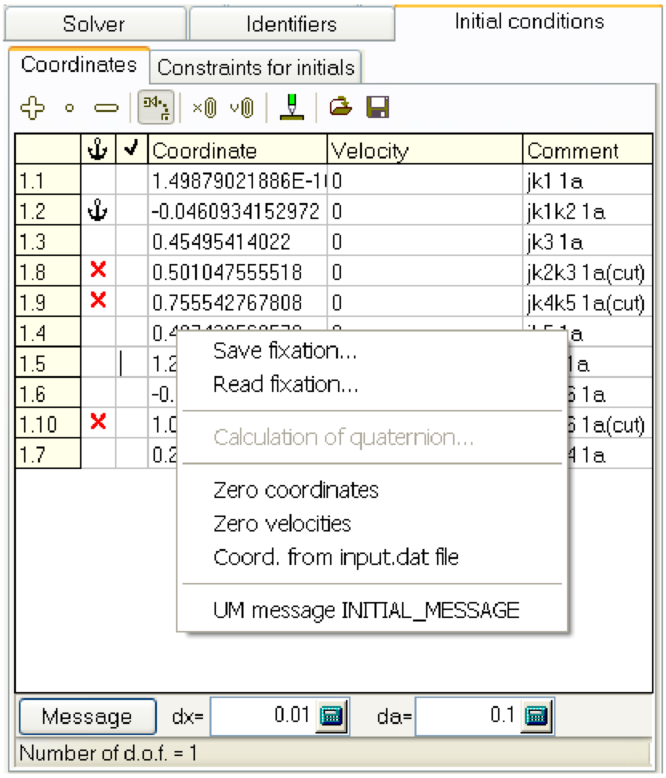


Figure 4.92. Initial conditions tab and the corresponding pop-up menu

Working with the master of initial conditions requires some experience. Consider the elements in Figure 4.92.

- **Parameters and icons in the table of initial coordinates and velocities**

The ⚓ column contains marks of coordinate *fixing*. It is used for objects with redundant coordinates as well as gearing force elements and ignored otherwise. To fix a coordinate, click the right mouse button in a cell of the column which corresponds to the coordinate. The ⚓ mark appears in the cell. Auxiliary coordinates cannot be fixed. *Fixed coordinate values are kept during the Newton-Raphson iterations*.

The × icon marks joint coordinates in cut joints.

The ✓ column is used for *selection* of one or a group of coordinates. Selection is used for stepwise changing coordinates by clicking the + − buttons. Steps of changing are different for angular and translational coordinates (the **da** parameter for angular and **dx** – for translational coordinates).

The **Coordinate** and **Velocity** columns contain the current values of coordinates and their time derivatives. Direct input of values is allowed.

- **Setting zero values of coordinates and velocities**

The ×0 v0 buttons assign zero value for all coordinates and velocities.

- **Computation of coordinates in equilibrium**

The button calls the procedure for computation of equilibrium position of the model.

- **Computation of current coordinate values**

The , and buttons are used for computing and visualization of the object current position (the button), as well as for stepwise decreasing () or increasing () selected coordinates. If redundant coordinates are presented, the constraint equations are solved taking the current coordinates as start value in iterations.

- **Number of degrees of freedom of the model**

The number of degrees of freedom in written in the status bar of the coordinate table, Figure 4.92. Use the buttons to get this number. The button must be in the ‘down’ state.

- **Saving initial conditions to file and reading from file**

The buttons read from file and write to file initial coordinate and velocity values. Unformatted coordinate files have *.xv extension.

- **Automatic saving initial conditions to file last.xv**

If the corresponding option is enabled, Sect. 4.1.2, the initial values of coordinates and velocities are saved into the last.xv file by closing the model.

- **List of commands of the pop-up menu**, Figure 4.92.

Save fixation. The command save in a file *.fix (fixation file) indices of coordinates, which the fixation state is assigned to. The tool is used mainly for automatic computations of initial angular velocities of models including gearing force elements, Sect. 4.4.1.6.5.

Read fixation. The command reads a preliminary created fixation file.

Zero coordinates. The command sets zero values for all coordinates like the button .

Zero velocities. The command sets zero values for all velocities like the button .

Coordinates from input.dat file. The command sets value of coordinates from the input.dat file of the model.

Message INITIAL_MESSAGE. The command sends a message with the key **INITIAL_MESSAGE** to a control file like the top button Message in Figure 4.92. The tool is used by programming using the control file, see Chapter 5.

Remark.	Fix a coordinate or a group of coordinates to forbid their random changing during computation of the constraint equations.

4.4.1.6.3. Specifying initial conditions for objects without redundant coordinates

Enter desirable values of coordinates or press *Enter* on click the button. The object will be redrawn in all animation windows. To get a stepwise decreasing/increasing a coordinate or a group of coordinates, select them, click and keep one of the buttons.

4.4.1.6.4. Specifying initial conditions for objects with redundant coordinates

The choice of the initial conditions in the case of the object with closed kinematics loops is much more difficult because nonlinear constraint equations should be solved.

If the button is down, the automatic calculation of constraint equations is executed after every modification of a coordinate. If the button is released, the calculation does not start.

The button is used to start the Newton-Raphson iterations. After the program has computed the mechanism configuration, a stepwise decreasing/increasing coordinates is available. Fix and select a coordinate or a set of coordinates: , . Click one of the , buttons.

Number of fixed coordinates cannot exceed the number of degrees of freedom.

Advice. Save successfully computed initial conditions to file.

4.4.1.6.5. Computation of initial conditions for models with gearing. Fixation file

If a model contains the gearing force elements, the so called **fixation file** can be useful for automatic computation of initial angular velocities of bodies. The fixation file contains indices of coordinates, which are fixed by computation of initial conditions.

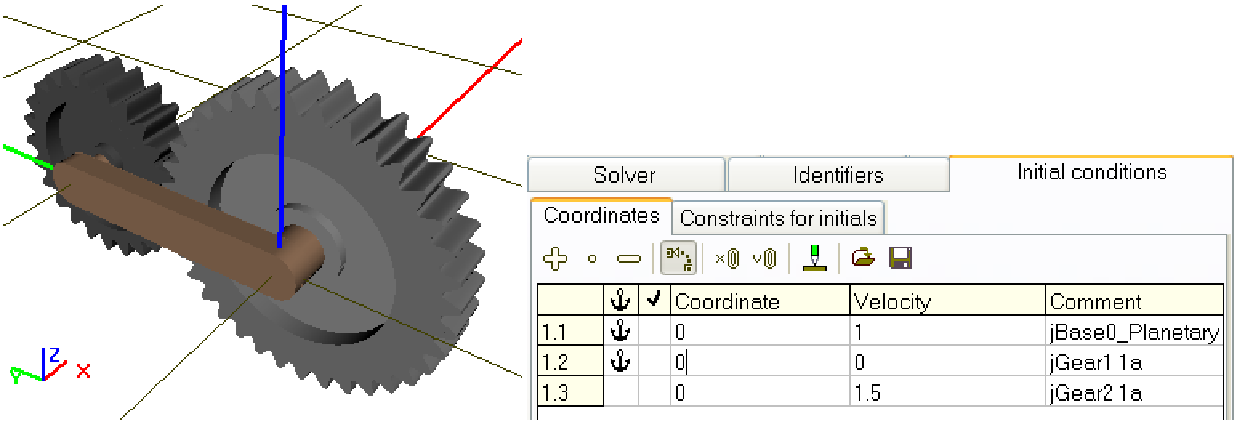


Figure 4.93. Planetary train model and table of coordinates

Consider a simple example to clarify the sense of the fixation file. The mechanism under consideration is the simplest planetary train consisting of a carrier and two gears, Figure 4.93, the model {UM Data}\SAMPLES\LIBRARY\Planetary_gear. Three joints introduce three rotational degrees of freedom

- rotation of the carrier relative to the SC0 (jBase0_Planetary carrier 1a),
- rotation of the sun gear relative to the SC0 (jGear1 1a),
- rotation of the planet gear relative to the carrier (jGear2 1a).

Time derivatives of the rotation angles satisfy the equation

$$\omega_1 r_2 - \omega_2 r_2 - \omega_3 r_3 = 0,$$

where $\omega_1, \omega_2, \omega_3$ are the angular velocities; here the indices correspond to the position on the angle in the table in Figure 4.93; ω_3 is the angular velocity of the planetary wheel relative to the carrier; r_2, r_3 are the radii of the sun and planet gears.

The program tries to solve automatically this equation relative to one of the three velocities. If the user does not point out, which of the velocity must be computed, the program chooses it in an arbitrary way. For instance, the carrier angular velocity will be computed for the given values of angular velocities of the gears.

As a rule, a quite definite velocity must be computed, for example the planet gear angular velocity ω_3 must be computed by the given carrier and sun gear velocities ω_1, ω_2. In fact this means that velocities ω_1, ω_2 cannot be changed during calculation of initial values, and they must be fixed by the ⚓ sign in the first column of the table like in Figure 4.93.

The indices of fixed coordinates must be saved in the fixation file with the name of the model by the **Save fixation** command of the popup menu. The program loads such the file automatically and uses the fixation by computation of correct initial conditions.

Remark 1.	Using the fixation file in locomotive models is described in Chapter 8, Sect. *Computation of initial angular velocities by fixation file*.
Remark 2.	If the user changes the number of coordinates in the model, the old fixation file becomes incompatible with the new model and must be created anew.

4.4.1.6.6. Constraints on initial conditions

The tab allows specifying relations for automatic computation of initial values of coordinates and velocities, in particular in dependence of identifiers.

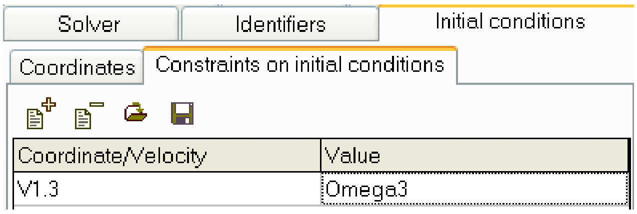


Figure 4.94. List of constraints on initial conditions

The buttons in the top of the tab are used to development of the list of constraint, its saving and reading, Figure 4.94. The table contains two columns, in which variables created with the wizard of variables are dragged, Sect. 4.3.2:

- the left column accepts variables, corresponding to coordinates and their time derivatives, Sect. 4.3.2.1;
- a variable corresponding to the desired initial value of the coordinate or velocity is placed into the right column.

Consider an alternative to the fixation file as an example. Let us create a variable foe computation of initial value of the planet gear from Sect. 4.4.1.6.5.

The angular velocity ω_3 is computed by the formula

$$\omega_3 = (\omega_1 - \omega_2)\ \frac{r_2}{r_3}.$$

The right part of this equation in programmed on the Expression tab of the wizard as the variable **Omega3** and dragged by the mouse into the right column in Figure 4.94. The following variables are used as operands in the expressions

- V1.1, V1.2 are the joint velocities;
- r2, r3 are the variables-identifiers corresponding to the gear radii, Sect. 4.3.2.13.

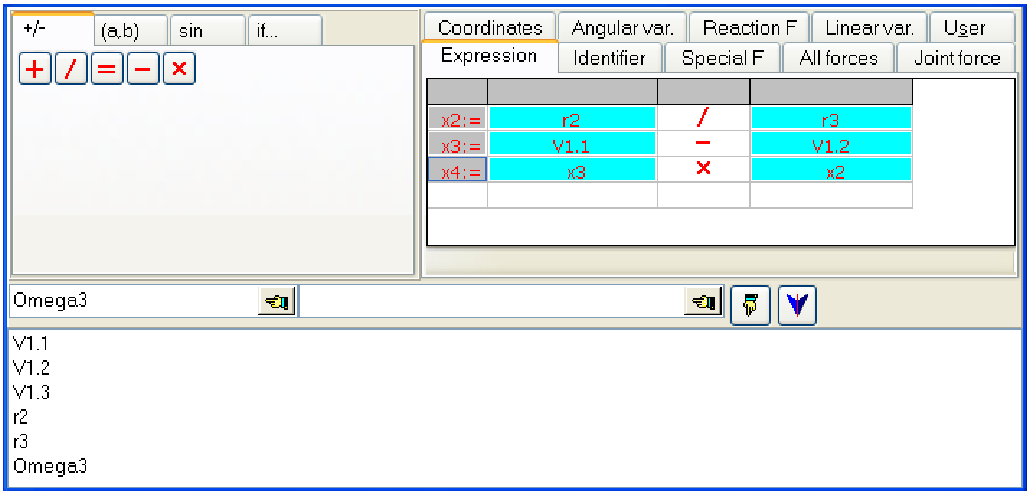


Figure 4.95. Expressions for computation of angular velocity of the planet gear

Remark 1. Use of constrains on initial conditions by computations of initial velocities for locomotive models is discussed in Chapter 8, Sect. *Computation of initial angular velocities by constraints on initial values*.

4.4.1.6.7. Computation of boundary values of a joint coordinates

The automatic computation of model configuration can be applied to estimation of the coordinate interval bounds, if the model has closed loops. Compute configuration with the , buttons until the program send a message about impossibility of computation of constraint equations.

Remark. The method cannot be used for evaluation of bounds of coordinates in cut joints.

4.4.1.7. Test for force start values

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

To run the test, open the **Tools | Test** tab of the inspector and click the Compute button. The program computes the forces and displays the force component values on SC0, Figure 4.96. The user compares the values with the expected ones.

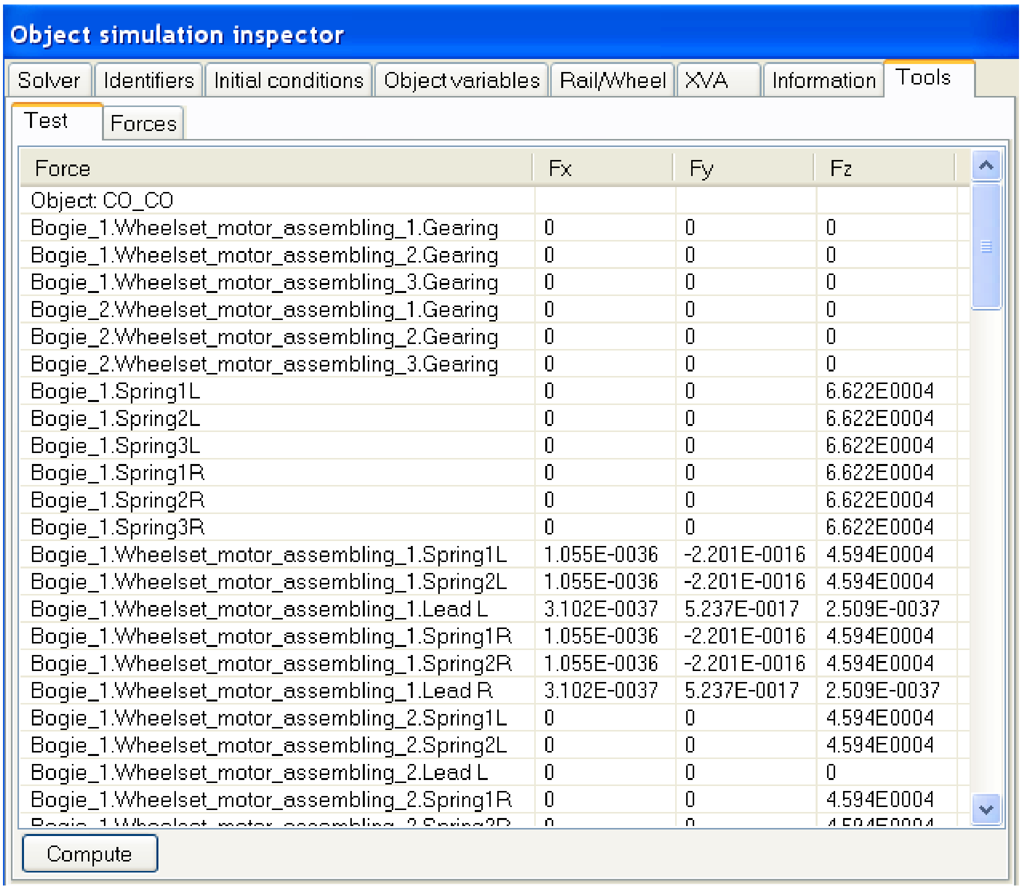


Figure 4.96. Test for forces

4.4.1.8. Disabled and enabled forces. Key for stiff forces

The user can disable any forces in the model. Disabled forces are not computed during simulations. The disabled/enabled key is available on the **Tools | Forces** tab. To disable a force, the user must uncheck it in the first column marked by the image, Figure 4.97. The list of disabled forces is stored in the model configuration file *.icf.

If computation of Jacobian matrix is checked, it is recommended to skip the computation of Jacobians for some non-stiff force elements, which makes simulation faster. Usually, suspension springs and dampers, traction torques, indicator diagrams are not stiff. To skip the computation of Jacobians, the non-stiff forces must be unchecked in the second column marked by the image J. The list of non-stiff forces is stored in the model configuration file *.icf.

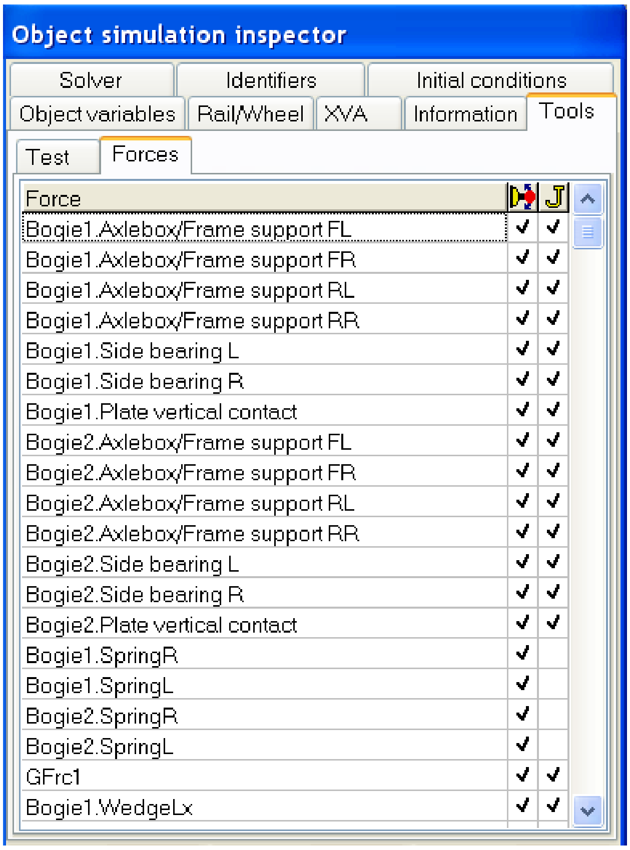


Figure 4.97. List of forces for disabling and setting the stiffness option

4.4.1.9. Assignment and usage of a list of automatically calculates variables

Use the automatically calculated variables (the **Object variables** tab) for access to one of the basic form of data storage and analysis the object by simulation (Figure 4.98).

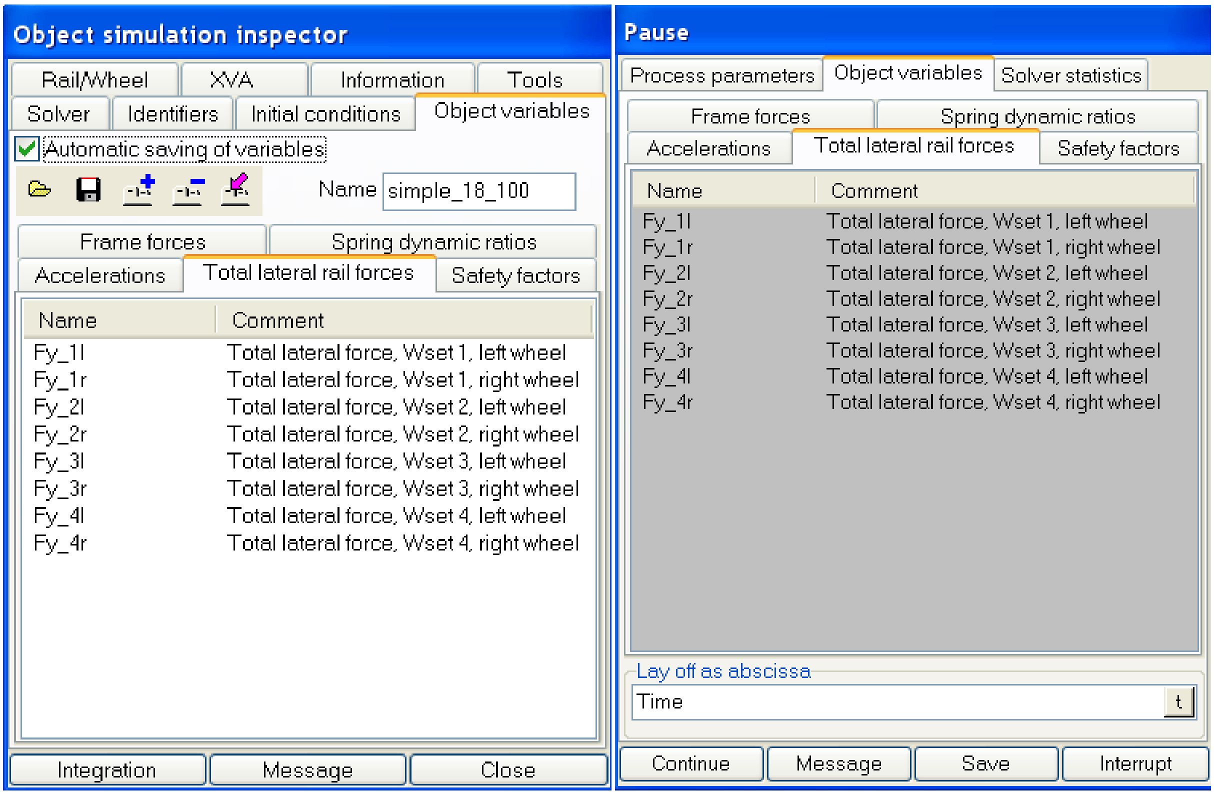


Figure 4.98. List of automatically calculated variables
before start the simulation (left) and in the pause mode (right)

For saving the calculated data during the simulation, check the **Automatic saving of variable** key. Then read or fill the list (Sect.4.3.3.2). All variables in the list will be saved in the files *.sgr and *.tgr. The file with extension *tgr* contains text information about the variables, and the file with extension *sgr* contains numeric values of variables.

Use the button to rename the output files.

The calculated variables are enabled for analysis in the pause mode and after the simulation. Use the *table processor* (Sect. 4.3.7), *graphical windows* (Sect. 4.3.4) and a *window for statistics* (Sect. 4.3.8) to analyze variables. The calculated variables of one object can be analyzed when *another* object is active, e.g. to compare results.

Consider the methodology of usage of the list of automatically calculated variables.

- A list should be chosen, modified or created before the simulation start.
- Variables cannot be deleted from the list after start of the simulation process.
- In the pause mode and after the simulation, the variables can be dragged into a table processor and into a graphical window. The variables dragged into a graphical window obtain the status of *calculated variables* (with some restriction on saving, Sect.4.3.4.1).
- One and the same list can be used for several starts of simulations. Rename the list in the **Name** box to keep the old files with calculated variables.

- Use the **Tools | List of calculated variables** menu item to open any file with preliminary computed variables; the **.tgr* files must be selected. Variables from these lists can be analyzed in a table processor, in a graphical window or in a window for statistics.

4.4.1.10. 3D Contact interaction parameters

3D Contact tab appears if there are at least two rigid bodies with assigned contact manifold in the considered model. Please turn to Sect. *Input of bodies / 3D Contact* of the Chapter 3 of this manual to read in details about creating contact manifolds for rigid bodies.

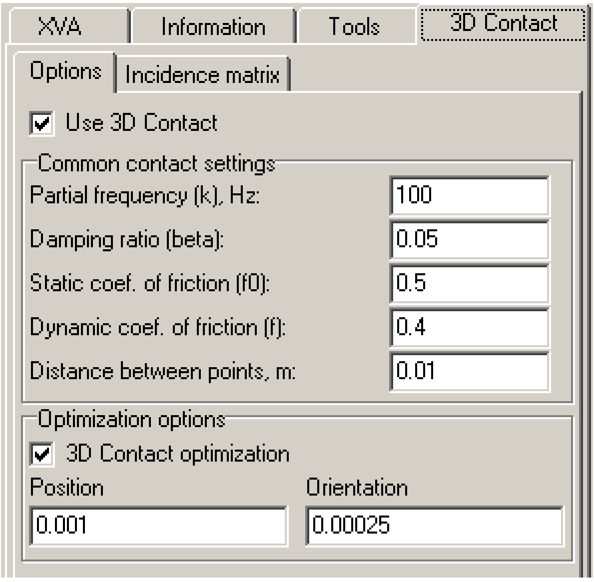


Figure 4.99. 3D contact parameters

Common contact settings

Partial frequency and **damping ratio** parameters define contact stiffness and damping coefficients for each pair of bodies (per contact point) as follows:

$$c = 4\pi^2 k^2 m, \quad \nu = 2\beta\sqrt{cm}$$, where

k is the partial frequency in Hz; c is contact stiffness, N/m; *m* is the mass of the smaller body, kg; ν is damping coefficient, Ns/m; β is damping ratio of critical. Detailed information about the partial frequency and the damping ration please find in Sect. *Methodology of choice of contact parameters* in Chapter 2.

Stiffness and damping coefficients are calculated for each pair of bodies that have assigned contact manifolds. Recommended values of **Partial frequency** lie in the range of 50÷500 Hz. Small value of this parameter may lead to big contact penetration bodies into each other. High partial frequency reduces numerical integration process. Recommended values of **Damping ratio** are 0.05÷0.5. Zero and small value correspond to nearly perfectly elastic impact, higher values rather corresponds to perfectly inelastic impact.

Distance between points means distance between neighbor contact points on edges. It is recommended to set this parameter so as it would be 2÷6 contact points on the edge of character length.

Assigning values of the **partial frequency** and **damping ratio** it needs to take into account the **distance between points**. The more count of contact points the higher effective frequency between two bodies. Contact points in fact work as parallel springs and dampers. That is why it is recommended to decrease the **partial frequency** and the **distance between points** simultaneously. Please also note that effective damping coefficient between two bodies increases faster than count of contact points because of non-linear dependence between damping and contact coefficients, see formulas above.

Optimization options

The main point of **3D Contact optimization** consists in ignoring the most time-consuming *near contact* if relative position and orientation between two bodies changed slightly. In fact procedure of *near contact* calculates 3D clipping according to Cyrus-Beck algorithm and places contact points on clipped sections. The procedure that calculates contact forces in contact points goes after the *near contact* procedure. So all, including small, changing in relative position and orientation will be take into account on the stage of contact force calculation even if **3D Contact optimization** is on.

Position and **orientation** parameters are threshold values that determines a necessity to recalculate a *near contact* procedure. **Position** parameter corresponds to linear relative position of bodies. **Orientation** limits the maximal change in elements of the relative cosine matrix. High values of these parameters can lead to probable artifacts of contact interaction – contact points will appear with considerable penetration that will raise unrealistically big contact forces. Small values of the parameters will decrease effect of **3D Contact optimization** and increase CPU efforts for simulation of **3D Contact**.

Incidence matrix

By default the contact mode **All to all** is used. In this contact mode the general contact settings are used. If necessary contact interaction between some of pairs can be turned off as well as contact parameters for each pair can be set particularly. For that you should replace **All to all** mode with **Manual settings** mode in the **Contact mode** drop down list. Then using the following dialog window you can assign contact parameters for any pair of bodies.

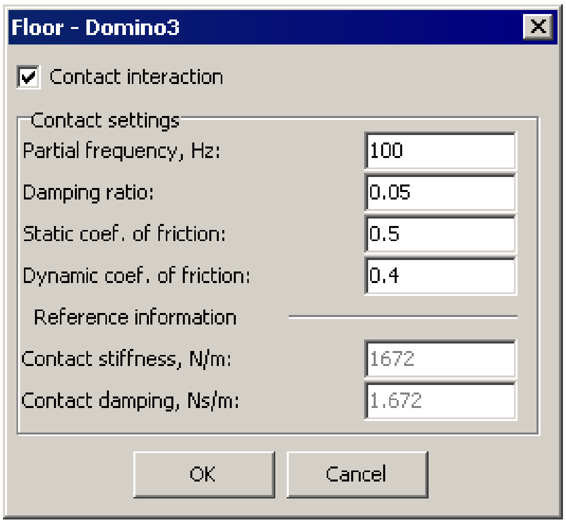


Figure 4.100. Parameters of contact interaction between two bodies

Options

If contact manifolds of bodies have considerable penetration then the will be very high contact forces on simulation starts. In such case big penetrations will lead to burst of contact forces that will move the system in incorrect/ undesirable state. To prevent such critical situations it needs to control initial positions of interacting bodies or turn off contact interaction between such bodies. The following buttons help you to check or turn off contact interaction between penetrated bodies automatically:

 shows the list of interpenetrating bodies;

 shows the list of interpenetrating bodies and turn off contact interaction between them.

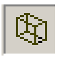
You should use this buttons before start simulation to prevent the burst of contact forces.

There is one more possibility to prevent the burst of contact forces. You can turn off contact interaction between bodies if there is interpenetration between them during first T seconds of simulation of system dynamics, see **Object simulation inspector | 3D Contact | Incidence matrix | Options**.

Recommendations.

For better understanding the processes in 3D contact, for determination of position of contact forces, as well as values and directions of contact normal and tangential forces it is recommended to use the follows tricks.

- Please note that it is possible to show contact forces in animation windows of **UM Simulation**. There are two ways to do that. (1) Firstly make sure that the **Mouse pick mode** () is on then point mouse at interesting body and click **Show forces for [Body]** menu item in the context menu. (2) Show **Wizard of variables**, select **All forces | Sets of vectors** tab, choose the interesting body in the left list of bodies and select necessary options in groups **Types of forces** and **Act**, see Figure 4.101. Please note that contact forces are **Applied** ones, that is why to show contact forces it is necessary to check on just **Applied** flag in **Types of forces** group. Please also note that in that case all applied forces will be shown including gravity and other applied force, see Sect. 4.3.2.7.
-

- Visualization *of vectors and* trajectories, page 4-78.
- For better view you can turn on/off visualization of bodies with the help of **Modes of body images** context menu command, Figure 4.102.
- Use the **Wire frame** mode () on the tool panel of an animation window to see contact force clearly even they are inside of body image, Figure 4.102.

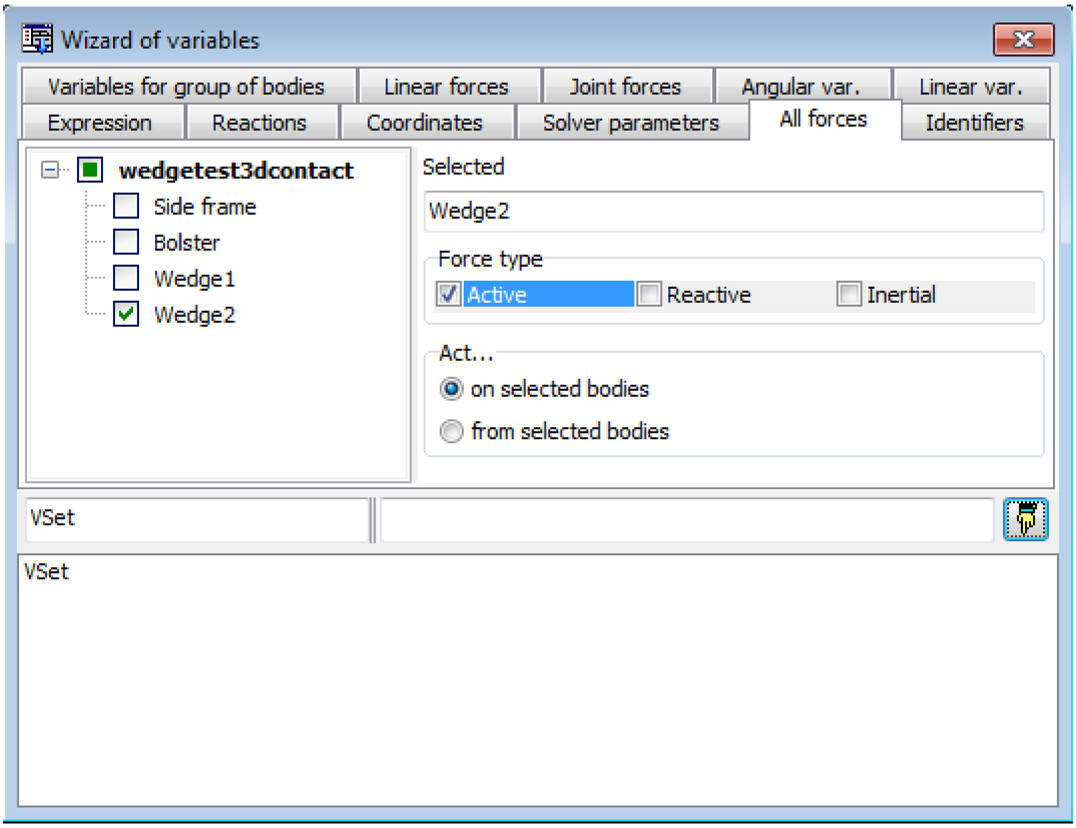


Figure 4.101. Wizard of variables: all forces

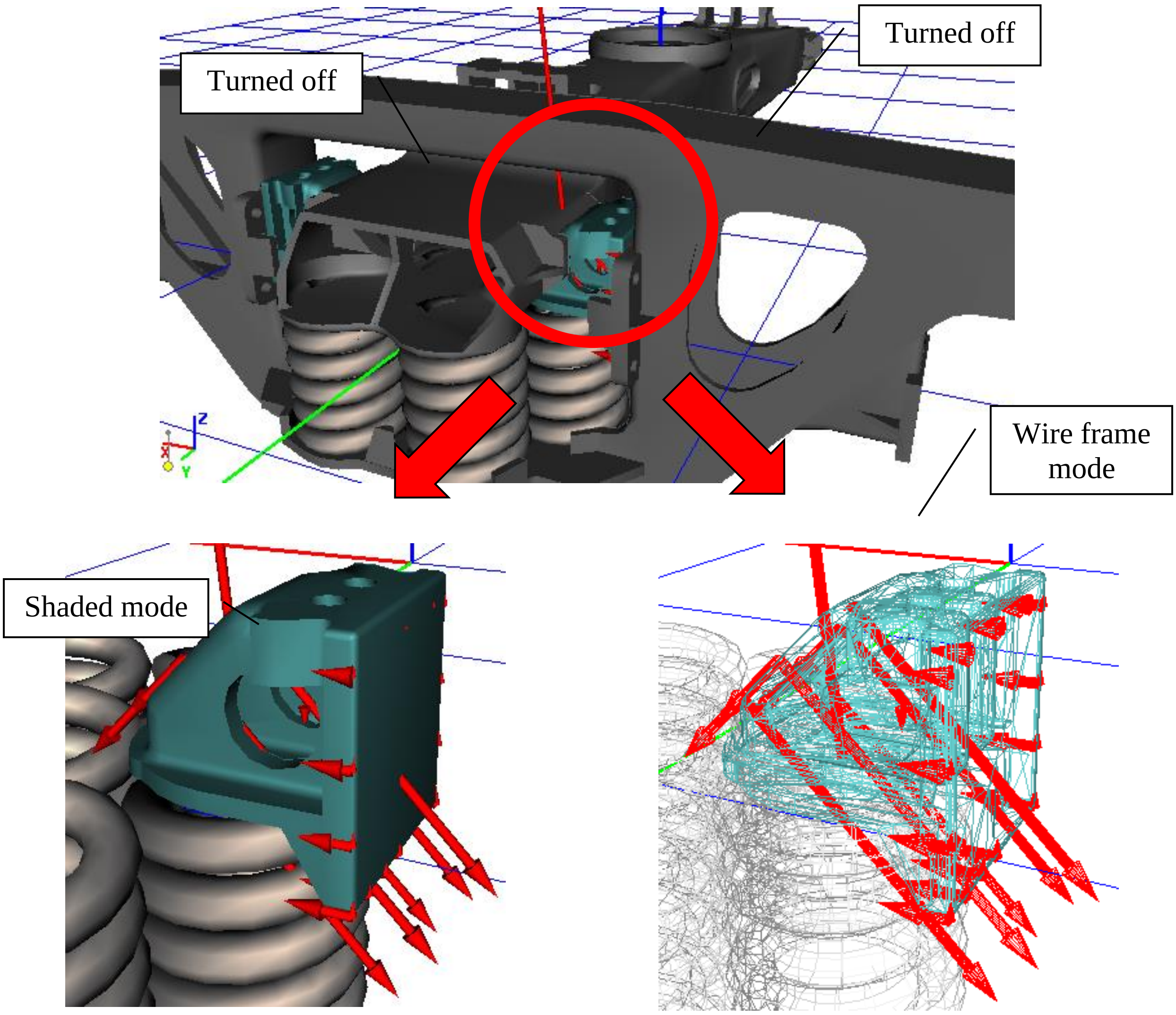


Figure 4.102. Animation window: tips and tricks

4.4.2. Integration of equations of motion (simulation)

Use the **Integration** button to run the integration process (Figure 4.102).

Redrawing animation windows, adding new points to plots, saving calculated variables and adding a frame to a file with animation are done with the **Animation step** size.

4.4.2.1. Pause mode

Press the space bar or the **Esc** key to start the pause mode (Figure 4.103).

At the pause mode the user can:

- Change the solver and some simulation parameters (simulation time, accuracy etc.) and continue the simulation with new parameters (the **Continue** button);
- Break simulation (the **Interrupt** button)
- Save the current object coordinates to use them later as initials (the **Save** button);
- Sent the message PAUSE_MESSAGE to the control file (the **Message** button);
- Analyze calculated variables (the **Object variables** tab, Figure 4.97).
- Get solver statistic (Figure 4.104)

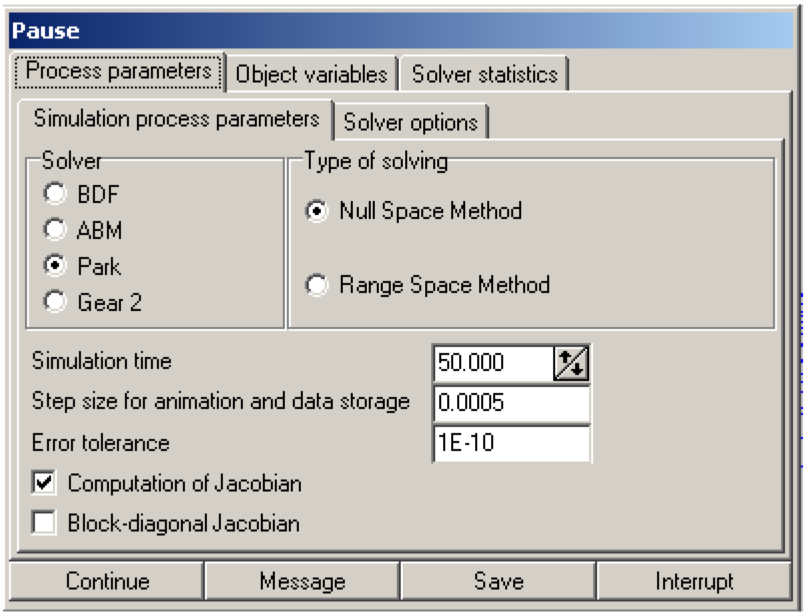
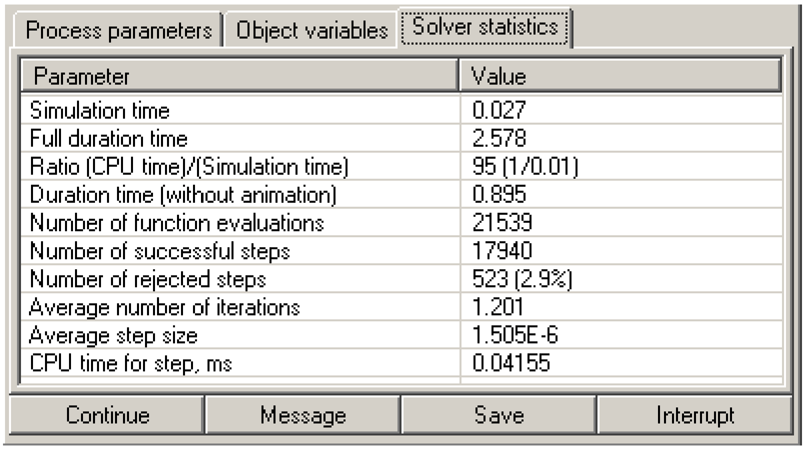


Figure 4.103. Pause dialog box



Parameter	Value
Simulation time	0.027
Full duration time	2.578
Ratio (CPU time)/(Simulation time)	95 (1/0.01)
Duration time (without animation)	0.895
Number of function evaluations	21539
Number of successful steps	17940
Number of rejected steps	523 (2.9%)
Average number of iterations	1.201
Average step size	1.505E-6
CPU time for step, ms	0.04155

Figure 4.104. Solver statistic window

4.4.2.2. Current parameters of simulation process

The following panel is available during the simulation process (Figure 4.105).

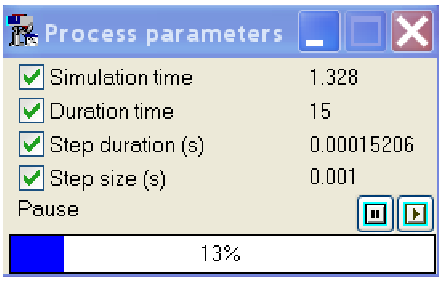


Figure 4.105. Parameters of simulation process

The dialog box contains the following parameters:

- **Simulation time** – model time from the start moment.
- **Duration time** – CPU time from the start moment.
- **Step duration (s)** - mean CPU time for a step (without expenses for animation and plotting graphs).
- **Step size (s)** – current time step in a method of integration. Time step of the method in case of nonlinear motions equations is variable. It is selected in purpose to reach goal accuracy of the result.

Use the  button to start the *Pause mode* (4.4.2.1). In the pause mode, the  button continues the integration process.

4.4.2.3. XVA-Analysis of simulation results

Running XVA-files (X stands for ‘coordinate’, V – for ‘velocity’, A – for ‘acceleration’) imitates integration. The basic difference consists in the fact that the values of coordinates and their derivatives are not the result of numerical integration but are read from the preliminarily created file of the integration results **.xva*. This type of analysis is used, e.g. for fast demonstrations.

That analyses type is used for demonstrations of earlier modeled motions. It is useful in case of slowly integration process. One can use graphical and animation windows during the XVA-analysis. It gives abilities of additional analyses of once calculated problem.

For creating XVA-file the user have to turn on the switch on at the **XVA** tab, enter XVA-file name and start the simulation (Figure 4.106).

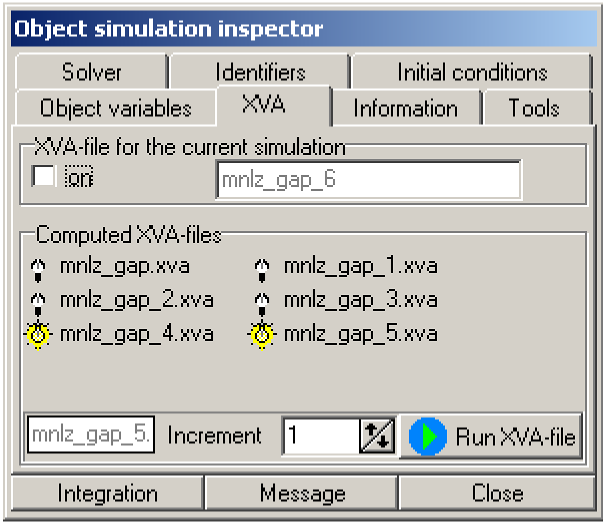


Figure 4.106. XVA analysis

The **XVA** tab contains a list of computed XVA–files.

If the object structure has been modified, an old XVA-file can include wrong information. The XVA-files created after the last modification of the object (file *input.dat*) marked with , created before it – with .

To run an XVA-file, double click its name in the list or select and click the Run XVA-file button. Changing of the parameter **Increment** allows to admit frames and to make the XVA-analysis faster.

Use **Esc** or **Space** key to stop the modeling process. XVA pause window (Figure 4.107) lets you to change **Increment** parameter, continue or interrupt XVA process.

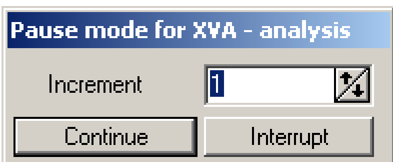
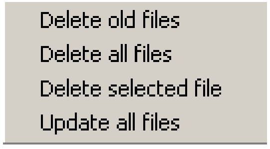


Figure 4.107. Pause window for XVA.

Editing of XVA-files list is available with the help of pup-up menu.



4.5. Linear analysis

Linear analysis solves problems of computing equilibrium positions of multibody systems, linearization of equations near the equilibrium and analysis of these linear differential-algebraic equations (Chapter 2, Sect. *Linearization of equations and equilibrium positions*).

Use the Analysis | Linear analysis menu item to get the corresponding regime of the UM Simulation program. Tools of linear analysis are located on tabs of the Linear Analysis window.

All forces like friction forces, which cannot be linearized, are not taken into account at the linear analysis mode.

Remark. Linear analysis is not supported for tracked vehicles.

4.5.1. Dependence of equilibrium on parameter

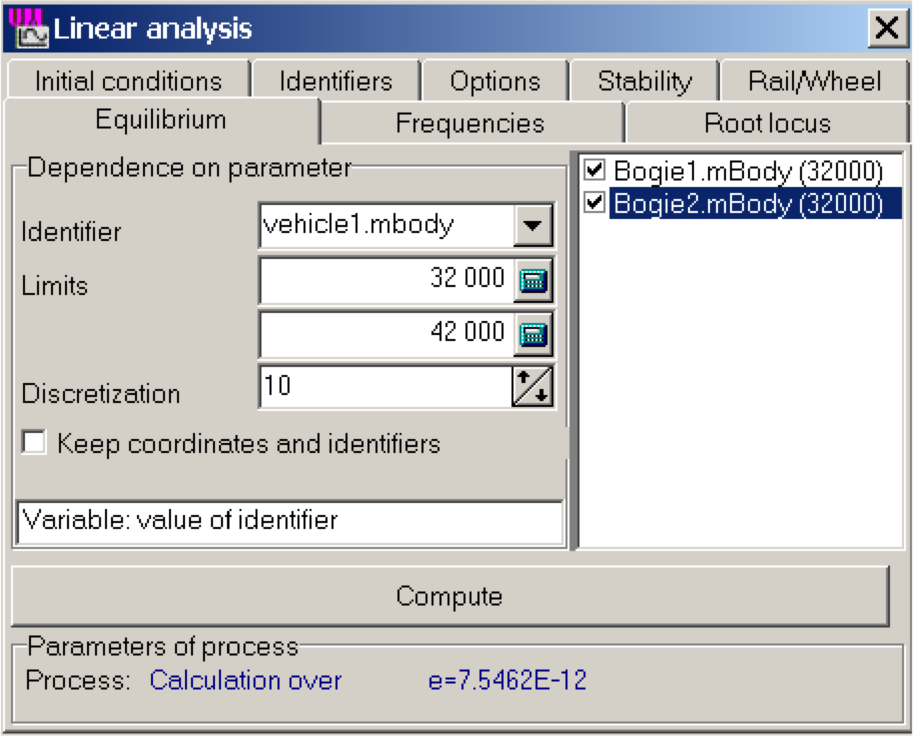


Figure 4.108. Preparing equilibrium computation in dependence on identifier

The *Equilibrium* tab is used for computation of equilibrium (Figure 4.108). Two modes of equilibrium computations are available.

Single computation for current values of identifiers

Use the *Identifiers* tab to set values of identifiers, and click the *Compute* button. The *Keep coordinates and identifiers* option is on if the computed should be accepted as the current ones. After the computation of equilibrium, vectors of forces can be visualized in an animation window, if the corresponding variables are placed in the animation window (Sect. 4.3.6).

Visualization of vectors and trajectories

Computation of equilibrium depending on an identifier

The mode is used for computation of coordinates, reactions and active forces in equilibrium position depending on identifiers.

To prepare the computation

- Select an identifier in the pull-down menu; if identifiers of the same name are presented in subsystems, check those which are changes simultaneously with the selected identifier (Figure 4.108 shows two checked identifiers of the same name *mBody*);
- Set boundary values of the identifier (a start value in the upper, and a finish value in the lower edit box) as well as a number of computation with a constant step (the *Discretization* parameter);
- Use the *Keep coordinates and identifiers* option if coordinates and value of the identifier at the last computation should be accepted as current ones;
- To get some plots of scalar variables at equilibriums versus the identifier in a graphic window, place the necessary variables in one or several graphic windows, and then drag by the mouse the *"Variable: value of identifier"* in the graphic windows;
- To get animation of vectors or trajectories during the computation, place the corresponding variables into an open animation window;
- To get an animation file for sequence of equilibrium positions, use the standard utility of the animation window (Sect. 4.3.6.2);
- After finishing the preparing process click the *Compute* button to run the computation.

If the computation of equilibrium fails, use the Initial condition to set values of coordinates, which are near the supposed equilibrium position. Sometimes a preliminary integration of equations of motion for models with dissipation is useful to bring the model in the position near to equilibrium.

4.5.2. Natural frequencies and eigenvalues

The *Frequencies* tab is used for calculation of natural frequencies and eigenvalues.

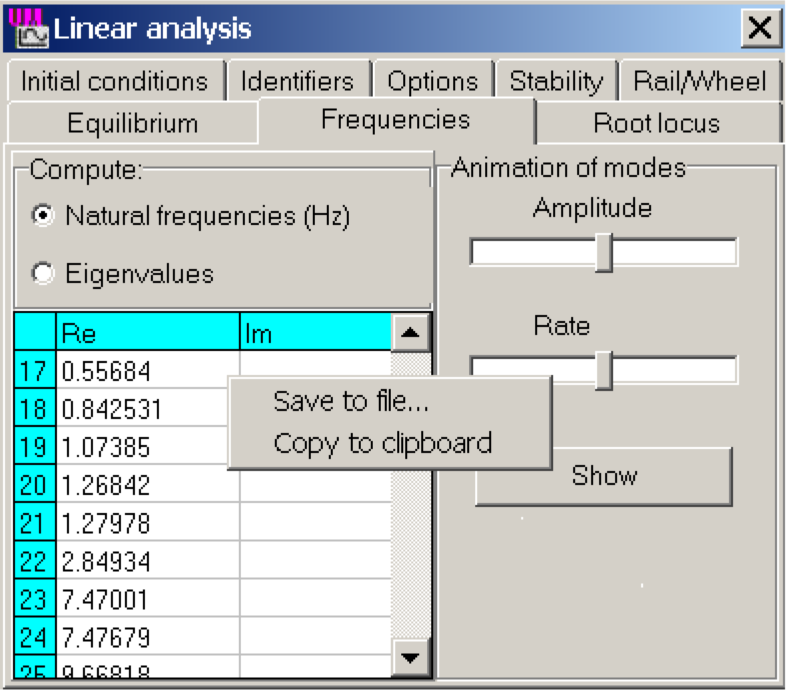


Figure 4.109. Natural frequencies

Computation is possible if equilibrium position is found. It is recommended to compute the equilibrium before evaluation of frequencies and eigenvalues (Sect. 4.5.1). If the equilibrium is found, equations are linearized in near it. Depending on computed values (eigenvalues or natural frequencies), dissipative and other non-conservative forces are taken into account or not.

Natural frequencies

- Imaginary frequency corresponds to instability of the equilibrium.
- Use a popup menu to save computed values in a text file or into the clipboard.
- Select a frequency in the list and click the Show button to get animation of a natural mode in an open animation window.
- To create an animation file for a natural mode, use the standard utility of the animation window (Sect. 4.3.6.2);
- Use the Amplitude and Rate track bars to change the corresponding properties of the animation.

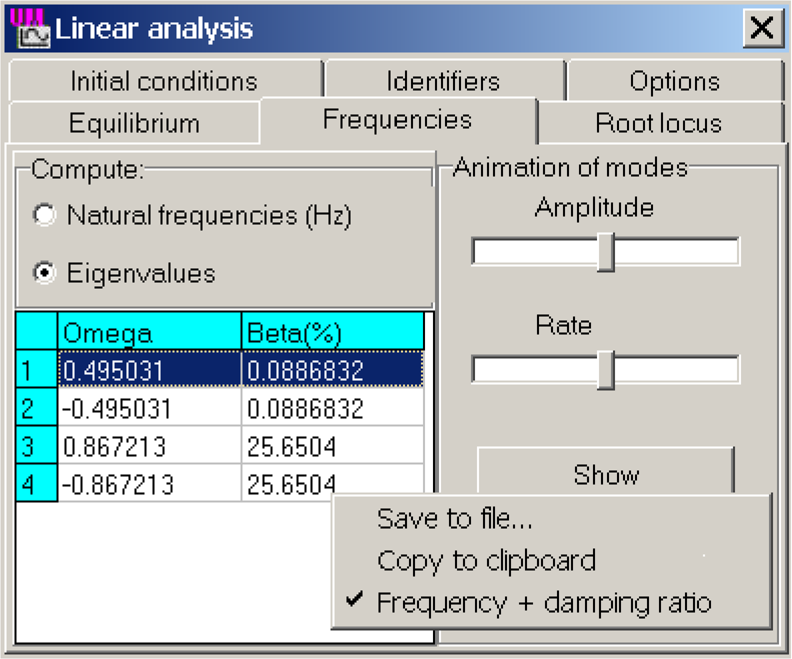


Figure 4.110. Eigenvalues in form frequency/damping ratio

Eigenvalues

Select the *Eigenvalues* parameter of the *Compute* group, Figure 4.110. Eigenvalues can be presented either in a *Real/Imaginary* form ordered by decrease of the real part or in a *Frequency/Damping ratio* (in % of critical) ordered by increase of the frequency. Use a popup menu to switch the forms.

Use the same approaches as for natural frequencies to animate eigenvectors as well as to create the corresponding animation files.

4.5.3. Root locus and dependence of frequencies on parameter

Use the *Root locus* tab to get a dependence of eigenvalues on a parameter plotted in a complex plane.

To prepare the computation

- Select an identifier, an interval and a discretization (see Capt.2 Sect. Dependence of equilibrium on parameter);
- List of indices should be set for natural frequencies (no more than 15 frequencies);
- Use the *Keep coordinates and identifiers* option if coordinates and value of the identifier at the last computation should be accepted as current ones;
- Window for plots appears automatically;
- Use the *Show* group parameters to place plots in the same window with deleting or keeping previous results or in a new window;
- Use an animation of root locus to select eigenvalues for a definite value of identifier.

Examples frequency lists for computation:

1-12

1..12,15

1,2,3,7,9,11-14

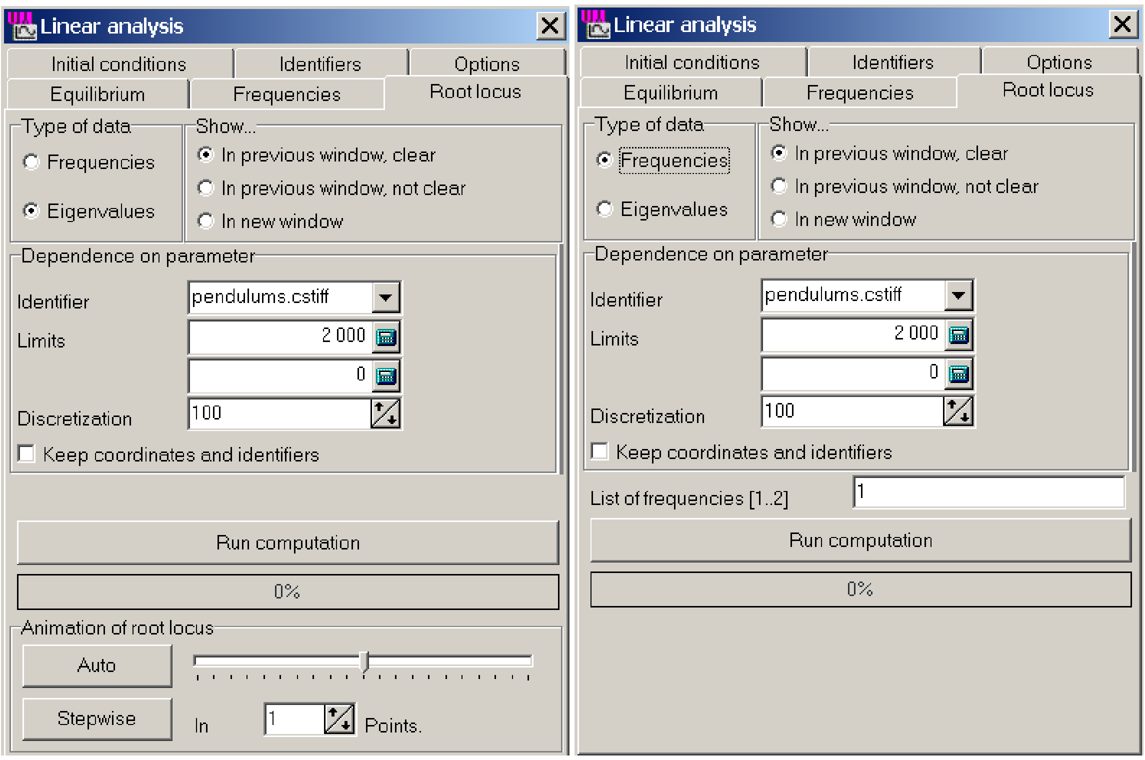


Figure 4.111. Computation of eigenvalues and natural frequencies in dependence on identifier

4.5.4. Options

Use the **Options | Parameters** tab to modify parameters of equilibrium computation procedure. The parameters *relaxation coefficient(R), Accuracy* (error tolerance ε), and *Increment of coordinates* (δ) are introduced in Chapter 2, Sect. *Equilibrium equations and their solving*.

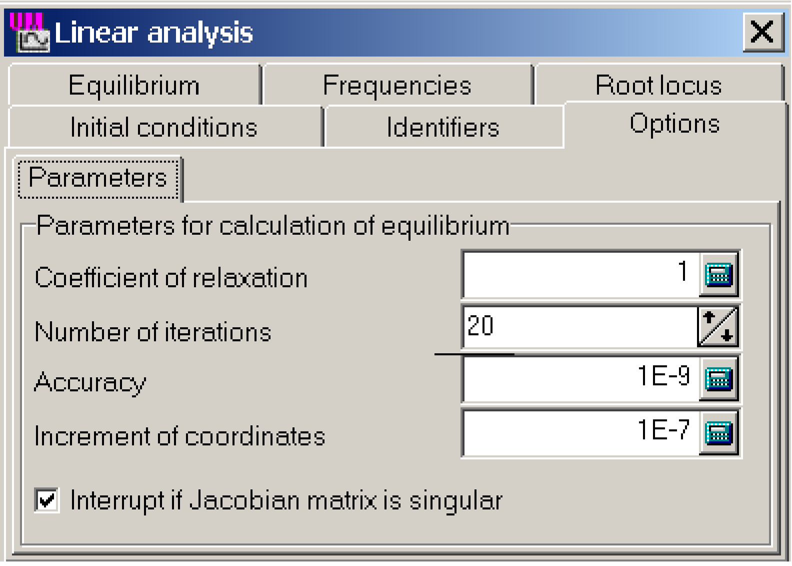


Figure 4.112. Parameters of equilibrium computation process

Turing off the **Interrupt if Jacobian matrix is singular** key could be useful for computation of neutral equilibriums.

4.5.5. Equilibrium in presence of contact forces

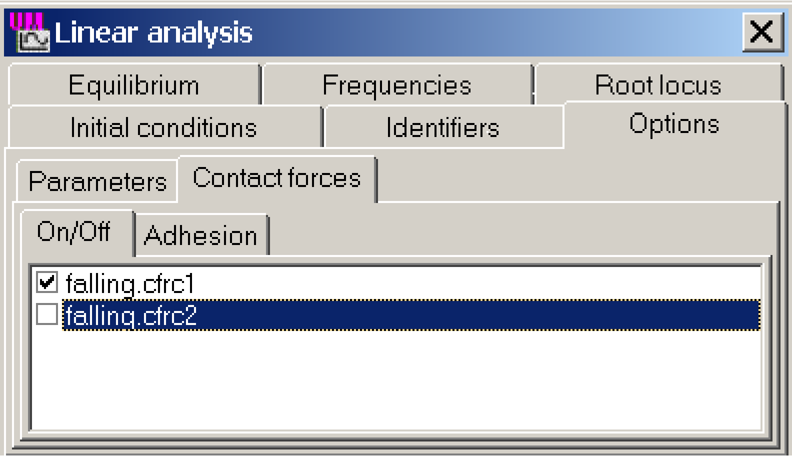


Figure 4.113. Turning on/off contact force

Tree modes are available for Point-Plane and Point-Z-surface contacts in linear analysis (Chapter 2, Sect. *Contact forces*).

- A full switching-off on the **Options | Contact forces | On/Off** tab.
- A two-lateral mode without sticking (friction is not taken into account); the force element should be switched on in the **On./Off.** tab, and it should be off in the **Adhesion** tab.
- A two-lateral mode with adhesion; the force element should be switched on both in the **On/Off** tab and in the **Adhesion** tab. The mode is similar to the previous one. The only difference consists in computation of eigenvalues (not the natural frequencies) when the linearized equations are formed on the assumption of sticking mode of contacts. At sticking a tangential stiffness equal to the normal one is introduced.

Use the pop-up menu to save the state of the **On/Off** tab in a *.chc file. An existing chc-file with the name of the model will be loaded automatically.