Mechanical System as an Object for Modeling

General notions, definitions and mathematical models are considered in this section of the user's manual. The chapter can be considered as a scientific introduction in UM

2020

Contents

2. Modeling mechanical systems

Consider a physical pendulum i.e. a rigid body, which has a horizontal rotational axis not passing through its center of mass. It is a simple example of a mechanical system. Anybody could imagine the motion of the pendulum and suppose that the pendulum would swing if initially it were not in equilibrium. However, far not everyone can write down ‘off-hand’ the equation of motion and solve it. Anyway, even junior students of university natural faculties can do it. If the air resistance force is to be taken into account, the analysis can only be carried out by senior students. Moreover, if we attach a second body to the first one, even a professional mathematician cannot obtain the exact analytical solution (because it does not exist!). What then can be said about systems containing dozens, hundreds, and thousands of bodies?

In such cases, applying the following numerical methods for modeling is most effective:

- automatic generation of equations of motion;
- numerical analysis of equations of motion;
- treatment of the results of the equations analysis and their representation in a convenient form.

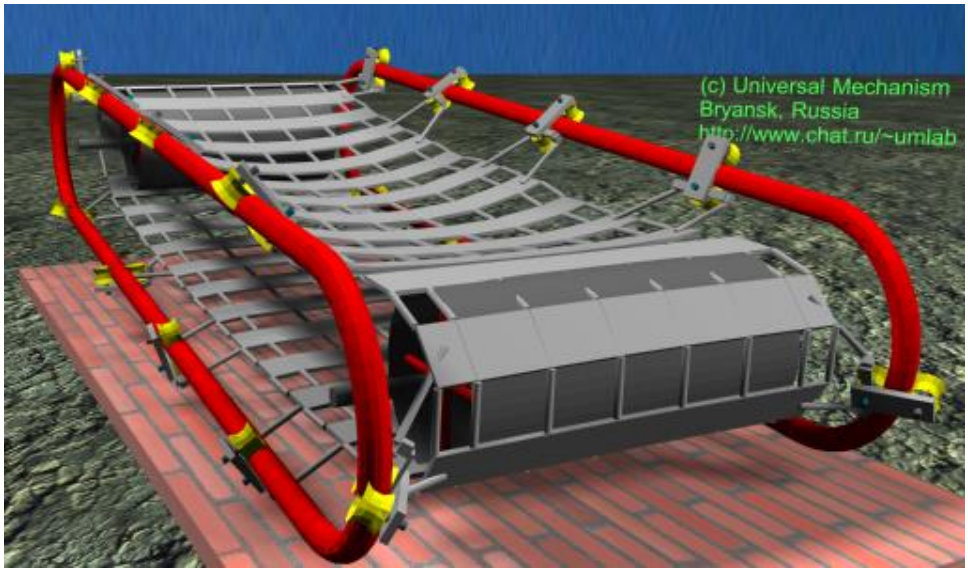
There are a lot of concepts for the equations analysis. So choosing one or another is, by far, determined by the nature of the analyzed system. This might be the numerical integration of the equations of an object in a complex spatial motion (e.g., for a robot manipulator). For a multibody system (MBS) whose bodies move slightly about a fixed in space position the determination of the natural frequencies and modes is often necessary. A considerable amount of information one could obtain through solving the problem of motion stability in the neighborhood of the equilibrium position or a steady motion.

The representation of the equation analysis results is most convenient when using computer graphics. So it is possible to simulate motion and to display time-varying charts.

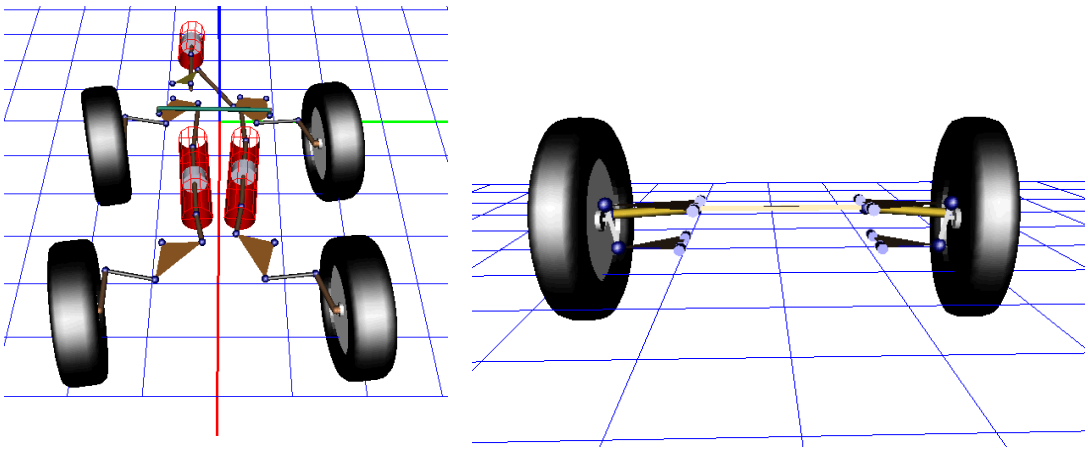
In the course of the system analysis, the designer often has to change its configuration and parameters (for example, the sizes of the bodies or any coefficients in the expressions for forces). These changes must be organized by the software as simply as possible. Once the configuration is changed, the equations of motion usually change as well. If the two-body physical pendulum is to be transformed into a three-body one, the equations will change totally. Here it matters how fast the equations are generated. The operational changing of parameters of an MBS is possible without generating the equations anew if they are derived in a fully symbolic form and the corresponding parameters take part in them as identifiers (data parameterization).

The program package UM has been designed for solving such and many other problems.

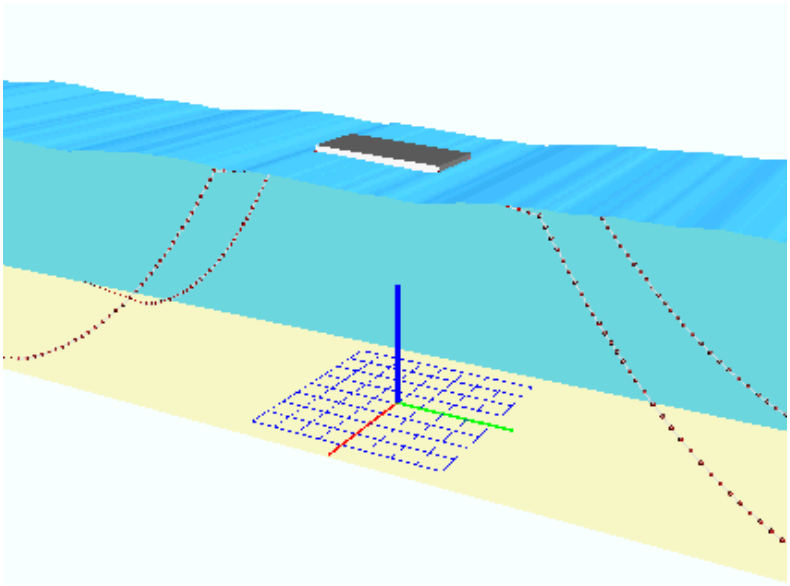
2.1. Rigid body systems



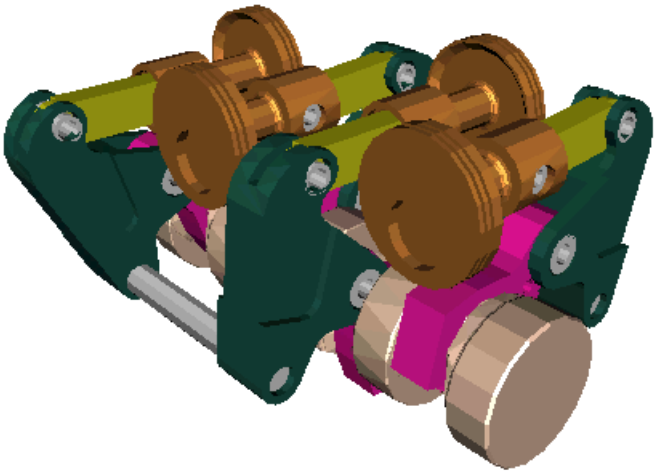
Belt conveyor



Double wishbone suspension of a vehicle (left); truck suspension with hydraulic actuators (right)

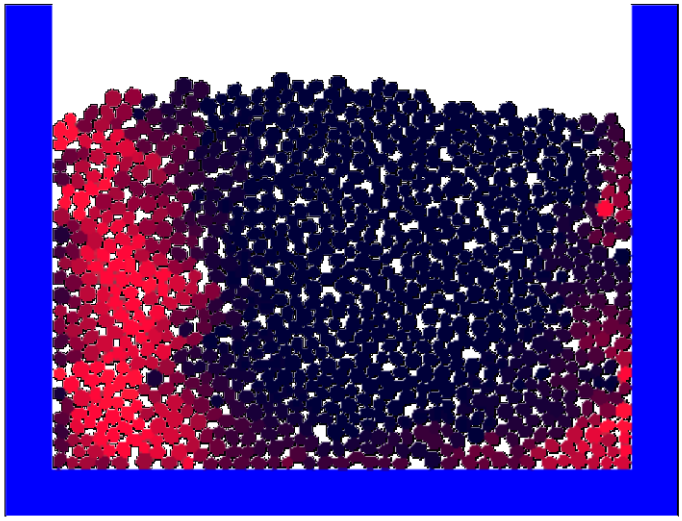


Mooring platform and its anchor system

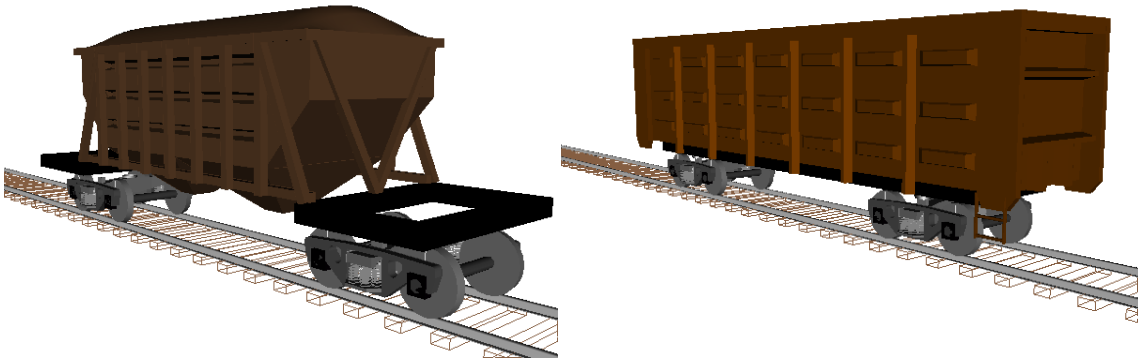


Expander

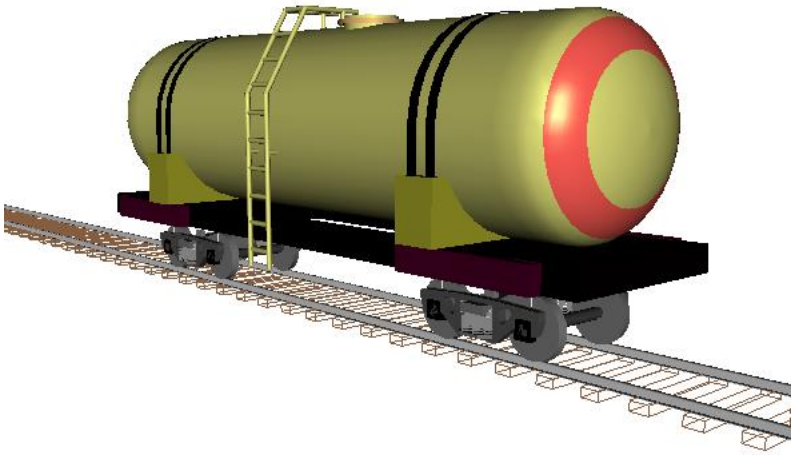
Tracked vehicle



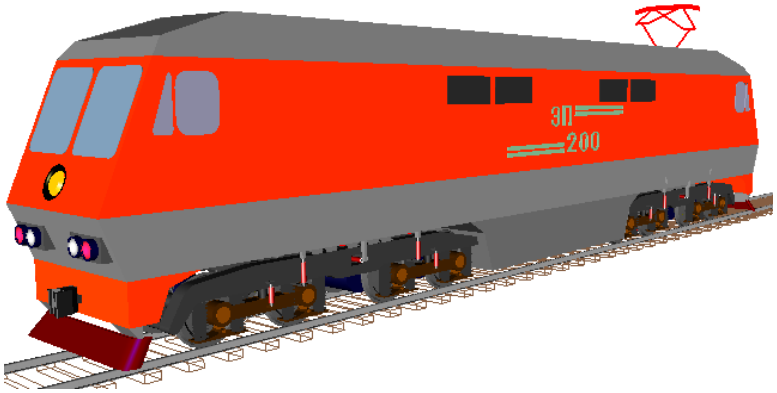
Ballast as a multibody system



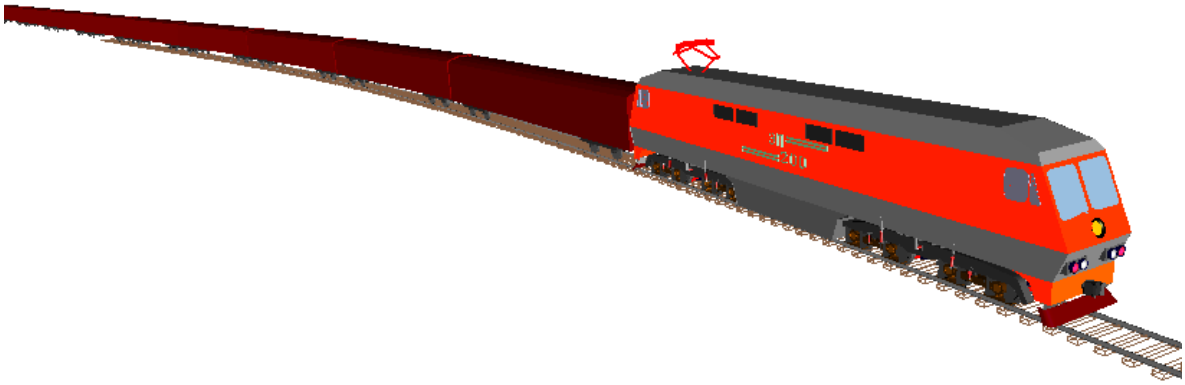
0



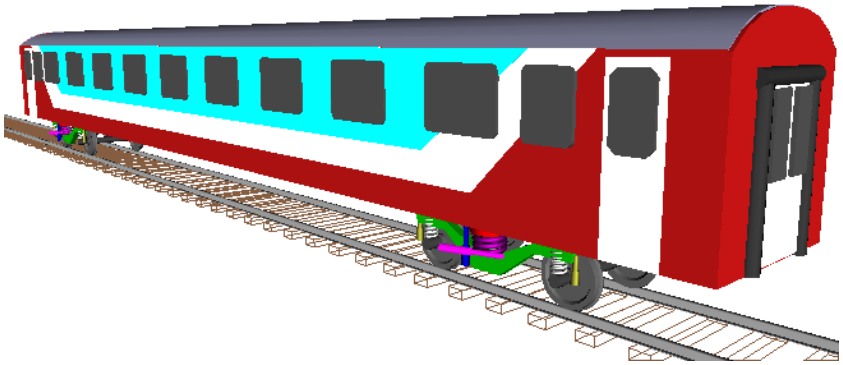
Freight railway vehicles



Electric locomotive EP200



Train model



Coach

Figure 2.1. Multibody systems

The objects to be dealt with by UM can be rigid and flexible body systems. The bodies of a system may or may not be connected with each other by joints and force elements. In particular, bodies may be mass points (Figure 2.1).

The motion of an MBS is studied with respect to *the basic body*, by which an inertial system of coordinates (SC) is meant. Often such a system can with enough accuracy be identified with the surface of the Earth. The basic body is considered fixed and therefore is not included in the

analyzed MBS, although takes an active part in the description of the system. *The basic SC is denoted SC0*. Usually, dividing a compound object into bodies is no problem. For example, a two-body physical pendulum consists of two bodies, whereas the Puma manipulator – of four. Sometimes, a deformable body, e.g., an elastic beam, can be represented as an MBS. For this representation to be valid the beam has to be divided into several rigid bodies. The separate bodies are connected with each other by massless elastic elements.

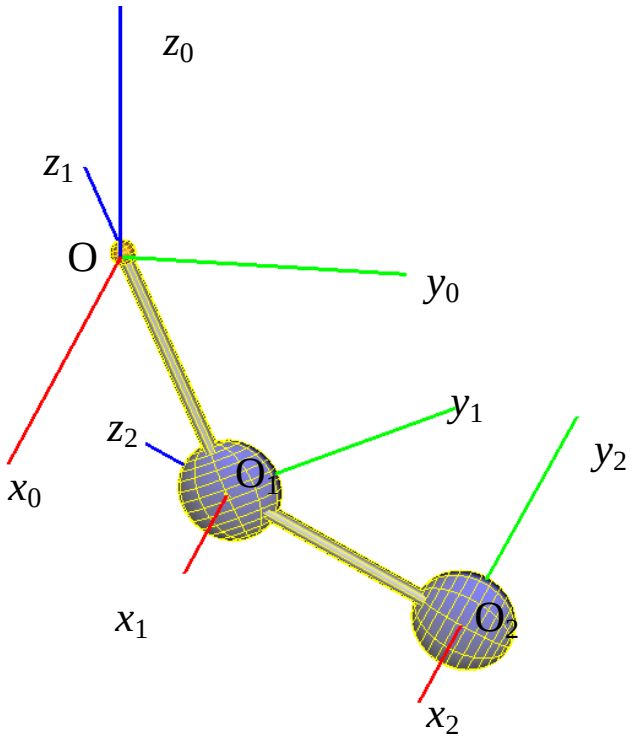


Figure 2.2. Double pendulum

To describe the motion of an MBS in terms of mathematics the right Cartesian system of coordinates is associated with each of the bodies. The origin can be located in any point of the body and the axes are fixed to it. Generally speaking, the orientation of the axes of the body-fixed SC may be chosen arbitrarily, but the equations of motion will be easier if the axes are the principal axes of inertia. Here and below such systems of coordinates are referred to as *the body-fixed systems of coordinates* and denote them as SC [the index of the body], e.g., SC1 is the system of coordinates fixed to body 1. In a particular case, when a body has axes of symmetry, the axes of the body-fixed SC are usually directed along the axes of symmetry.

For example, consider a model of the two-body pendulum, Figure 2.2, which consists of two homogenous rods connected by a rotational joint and attached by a joint to the immovable support. The axes of the joints are parallel and the motion of the MBS therefore is 2D, however in UM all coordinate systems *are assumed to be three-dimensional*. The basic SC0 Ox0y0z0 has the origin in O, which coincides with the center of the joint. The body-fixed systems of coordinate $O_1x_1y_1z_1$ and $O_2x_2y_2z_2$ originate in the centers of mass of the corresponding bodies, whereas the axes are directed along their axes of symmetry.

2.2. Joints

2.2.1. Connectivity of systems and definition of a joint

It is assumed that the modeled MBS satisfies a connectivity condition, i.e. each body of the system is connected by a joint at least with one other body or the basic body (SC0), and there exists a path from each body to SC0. This condition is very important and is automatically checked by UM. By what joint and with what body is an unrestrained body moving in space connected? And does it make sense at all to introduce joints for free bodies? It appears to be clear that to describe the position of a body in space it is sufficient to know the position of the body-fixed SC relative to the basic SC0. In terms of mathematics, for that it is sufficient to set the position of the origin and the orientation of the body-fixed SC with respect to the basic SC0, expressing them through certain variables called *coordinates*. For this purpose, UM uses the dependencies between, on the one hand, coordinates and, on the other, the radius vector of the body-fixed SC origin and the directional cosine matrix (the matrix of rotation). The term 'joint' is intended to describe the position of one body relative to another. To say 'describe the position of a body relative to the basic SC0 in terms of coordinates' means in UM 'introduce a joint between the given body and the basic one'.

Joints making it possible to describe the position of one body relative to another by means of introducing *the joint coordinates* are below denoted as *generalized, rotational, translational, quaternion and 6 d.o.f. joints*. UM uses another type of joints constraining the relative motion of bodies and which have certain problems in introducing the joint coordinates. In many systems the constraint represented by a massless rigid rod with joints at the ends is not a generalized joint either although it is also available with UM.

Certainly, generalized, quaternion and 6 d.o.f. joints can be introduced for any pair of bodies both kinematically connected and not. If any two bodies are connected by a joint in the usual sense – e.g., rotational or prismatic – its representation in UM assumes that the position of one body will be described relative to the other, namely the position of one body-fixed SC will be described relative to the other body-fixed SC. The joint coordinates, i.e. the variables setting this position, are introduced. The complete set of coordinates for the object as a whole is the result of unification of the local joint coordinates. Such a description takes place in the data input programs and is greatly automated. However, joints to connect each body with the basic one may not exist, although it is sufficient that there exists for each body a chain of bodies from this one to the basic. It is the urgent condition of *connectivity* of the modeled MBS

According to the condition of *connectivity*, it is also required that joints in the chain be not arbitrary but of certain types. Currently the following generalized types of joints are available:

- rotational;
- translation;
- six d.o.f. joint;
- generalized;
- internal body joint, based on 6 d.o.g.
- quaternion;
- massless rigid rod constraint;

- mates;
- convel (constant velocity) joint.

Rotational, translation, six degree of freedom and generalized joint belong to a certain group. They have identical internal representation and describe kinematical pairs with various translation and rotational d.o.f. (from zero to six). Rotational, translation and six degree of freedom joint can be described with the help of generalized joint.

Quaternion joint is often used for introduction of coordinates for a free body and for description spherical joint.

Massless rod, mates and convel joint do not introduce new coordinates. They are just kinematical constraints.

Joints of types

- rotational;
- translational;
- 6 degrees of freedom

have internal UM representation as generalized joints. Conversion of joints to generalized type is available in the **UM Input**. This tool can be used to create additional degrees of freedom, to describe joint forces, to parameterize inclination of axis and so on. See Chapter 3 of user's manual for additional information.

Example. User's manual, Chapter 7, Sect. *Joint type conversion. Parameterization of axis inclination*.

2.2.2. Description of joints

2.2.2.1. Translational and rotational joints

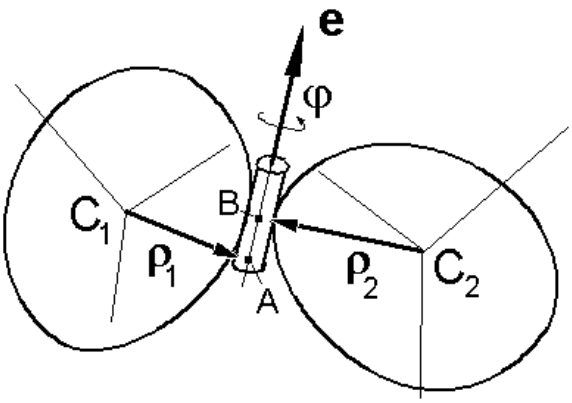


Figure 2.3. Joint with 1 d.o.f.

Translational and rotational joints allow describing one d.o.f. joints. Every joint of this type introduces one local coordinate (linear or angular). A sketch of such joints is given in Figure 2.3. There are following parameters:

- joint vector **e**, described by its projections on SC_1 and SC_2 (e_x^1, e_y^1, e_z^1), (e_x^2, e_y^2, e_z^2). Vector cannot be zero.
- coordinates of two joint points (connected with body 1 and body 2) situated on joint axis $(\rho_{1x}^1, \rho_{1y}^1, \rho_{1z}^1)$, $(\rho_{2x}^2, \rho_{2y}^2, \rho_{2z}^2)$.
- Some additional parameters, shift along the joint axis x_a and/or turning around the joint axis φ_a.

Vector (e_x^1, e_y^1, e_z^1) and point A $(\rho_{1x}^1, \rho_{1y}^1, \rho_{1z}^1)$ describe the joint axis position relative body 1, and vector (e_x^2, e_y^2, e_z^2) and point B $(\rho_{2x}^2, \rho_{2y}^2, \rho_{2z}^2)$ describe the joint axis position relative body 2.

If $x_a = 0$ then points A and B are coincide (have the same coordinate in SC0).

The x_a parameter in the case of a translational joint and the φ_a parameter in the case of a rotational joint are used for describing the relative positions of the bodies when the coordinate is zero.

Positive rotation corresponds to the right-hand screw rule.

There might be introduced joint force in a translational joint and joint torque in a rotational joint.

Local coordinates for both joint types might be described as explicit time functions. In that case joint coordinates is not included to the list of coordinates of object. For example, angle of rotation might be describes as $\varphi(t) = \omega t$, where ω is the angular velocity.

Examples

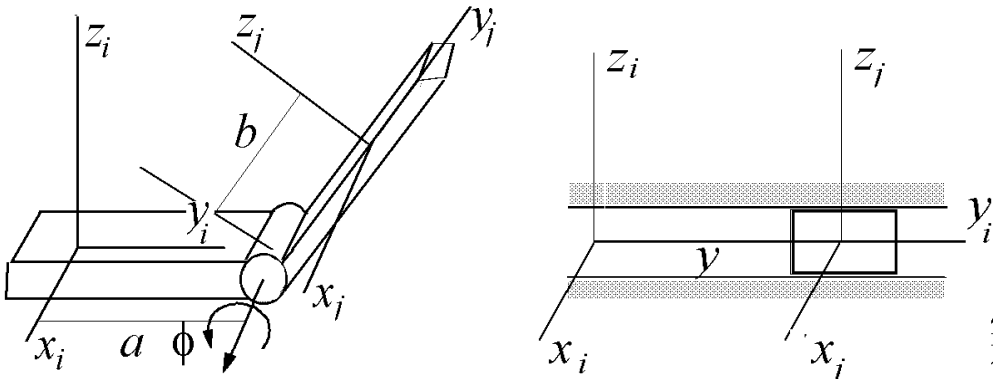


Figure 2.4. Rotational and translational joints

Here the rotational joint (Figure 2.4 left) is described by the following parameters:

$$\left(e_x^1, e_y^1, e_z^1\right) = (1,0,0), \quad \left(e_x^2, e_y^2, e_z^2\right) = (1,0,0),$$
$$\left(\rho_{1x}^1, \rho_{1y}^1, \rho_{1z}^1\right) = (0, a, 0), \quad \left(\rho_{2x}^2, \rho_{2y}^2, \rho_{2z}^2\right) = (0, -b, 0),$$
$$x_a = \varphi_a = 0,$$

And here the translational joint (Figure 2.4 right) is described by the following parameters:

$$\left(e_x^1, e_y^1, e_z^1\right) = (0,1,0), \quad \left(e_x^2, e_y^2, e_z^2\right) = (0,1,0),$$
$$\left(\rho_{1x}^1, \rho_{1y}^1, \rho_{1z}^1\right) = \left(\rho_{2x}^2, \rho_{2y}^2, \rho_{2z}^2\right) = (0,0,0),$$
$$x_a = \varphi_a = 0.$$

2.2.2.2. Six d.o.f. joint

Joints of such a type are often used to introduce kinematical pairs with various numbers of rotational and translation d.o.f. Consider two bodies: 1 and 2 (see Figure 2.5). Origins of body-fixed SC are located in points O1 and O2. Introduce two additional SC: SC1A and SC2B with its origins in some points A and B correspondently. The joint introduces coordinates, which describe the position of SC2B relative to SC1A.

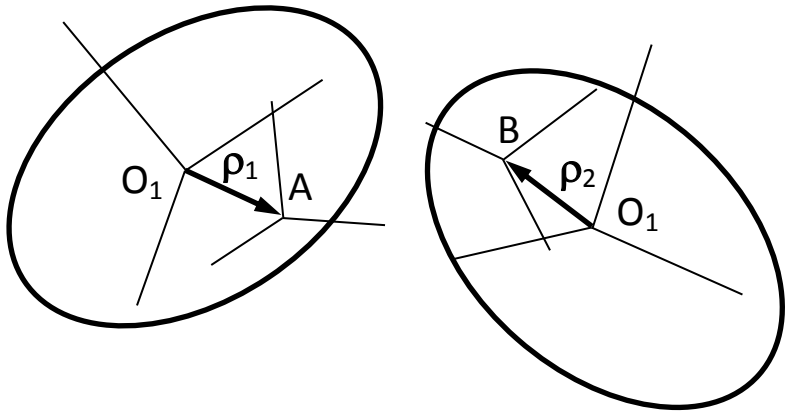


Figure 2.5.

By default the joint has six d.o.f.: three shifts SC_{2B} relative to SC_{1A} (x, y, z) and three angles of SC_{2B} orientation relative to SC_{1A} (a_1, a_2, a_3). In this case body 2 moves freely and joint does not introduce any constraints on relative movement.

There are 12 ways of introduction of orientation angles accordingly sequence of rotations around axes (x-1, y-2, z-3):

- Euler (3,1,3);
- (3,2,3);
- (2,1,2);
- (2,3,2);
- (1,2,1);
- (1,3,1);
- Cardan (1,2,3);
- (1,3,2);
- (2,1,3);
- (2,3,1);
- (3,1,2);
- (3,2,1).

Any of these six d.o.f. might be locked. It means that the d.o.f. will not be introduced in a model. A lot of various kinematical pairs might be described in this way.

The following parameters are necessary for description the six d.o.f. joint:

- coordinates of points A in SC1 and B in SC2 $(\rho_{1x}^1, \rho_{1y}^1, \rho_{1z}^1)$, $(\rho_{2x}^2, \rho_{2y}^2, \rho_{2z}^2)$ as well as orientation of SC1A and SC2B relative to SC1 and SC2;
- type of angle orientation;
- enabled/disabled d.o.f.

Consider some examples.

- Spherical joint (see Figure 2.5). Vectors ρ_1, ρ_2 describe the position of the center of the joint in SC1 and SC2, all translation d.o.f. are disabled. Here the spherical joint (Figure 2.6) is described by the following parameters: $\rho_1 = 0, \rho_2 = (0,0 - a)$, type of orientation angles is (3,1,3).

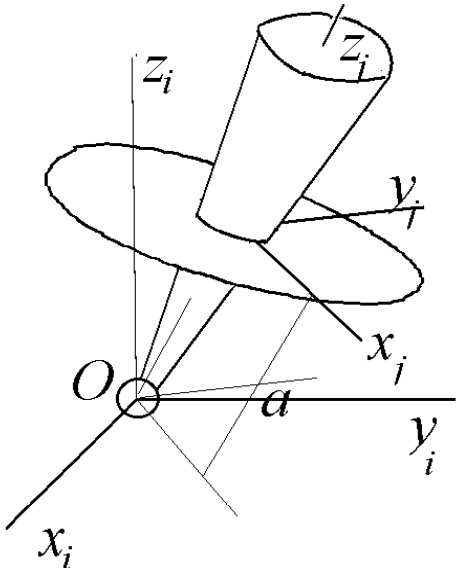


Figure 2.6. Spherical joint

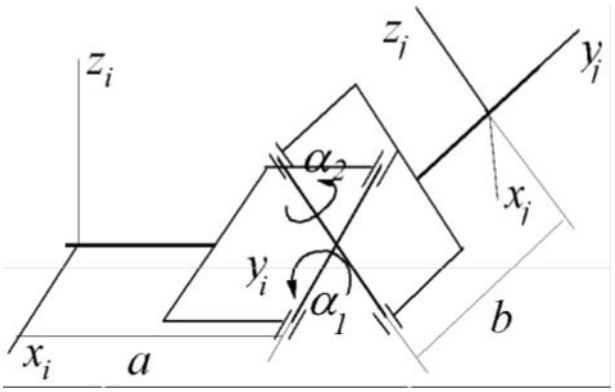


Figure 2.7. Hooke (universal) joint

- Hook joint with two d.o.f. (see Figure 2.2, Figure 2.7). All translations and one of rotational coordinate are disabled. Vectors ρ_1, ρ_2 describe the position of the center of the joint in SC_1 and SC_2. For the joint in Figure 2.2, Figure 2.7: $\rho_1 = (0, a, 0)$, $\rho_1 = (0, a, 0)$ type of orientation angles is (1,2,3), the angle a_2 is turned off (or the type is (1,3,2) and a_3 is turned off).
- One d.o.f. joints. Let us consider examples in Figure 2.4. For a rotational joint we have $\left(\rho_{1x}^1 \rho_{1y}^1 \rho_{1z}^1\right) = (0, a, 0)$, $\left(\rho_{2x}^2 \rho_{2y}^2 \rho_{2z}^2\right) = (0, b, 0)$, all translations are disabled and two rota-

tional ones, too. For example, these are angles a_2, a_3 for the orientation angles of type (1,2,3). For a translational joint $\rho_1 = \rho_2 = 0$, all the angles are disabled and translations x, z, too.

Remark. It is well known, that there are singular orientations of SC_2 relative to SC_1 for any type of three orientation angles. In the singular orientations, the numerical values of the orientation angles cannot be found uniquely, which results in failure of the simulation of motion. The first six types (axes of the first and the last rotations have the same indices) are singular at $a_2 = 0\pi$. The rest six types have singularities when $a_2 = \pm\pi/2$. This fact must be taken into account for spherical joints, if SC_2 can have an arbitrary orientation in respect to SC1 while simulation of motion. For example, do not use orientation angles being singular at $a_2 = 0, \pi$, if axes of SC_2 are parallel to those of SC1 at begin of simulation (when t=0). Generally, in case when a free body can have an arbitrary spatial orientation at runtime, use the quaternion joint instead of six d.o.f. one.

2.2.2.3. Generalized joint

Most of kinematical pairs can be modeled with the help of joints described above. However, there are a variety of kinematical pairs, which cannot be described with the help of those joints. Here is a list of examples, which is far from being complete:

- a Cardan joint, whose rotation axes are not perpendicular; or whose first axis is not parallel to some of axes of SC1; or whose second axis is not parallel to some of axes of SC_2;
- joints with more than one d.o.f. having joint forces and/or torques, corresponding some of d.o.f.;
- joints with more than one d.o.f. realizing predefined motion (some or all of d.o.f. are time-dependent functions).

All these examples, and also all the types of joints above, can be implemented using the *generalized joint*. Mathematical model of the joint is discussed in the scientific manual.

So, the *generalized joint*, by definition, is a constraint having an arbitrary number of both translational and rotational degrees of freedom (from 0 to 6). The transition from SC_1, connected with the first body, to SC_2, connected with the second one, can be described as a sequence of *elementary transformations* (ET) at an arbitrary relative position of the bodies. Each ET is either a translation along or a rotation about a certain direction. Let us introduce concepts of a *vector e* and a *parameter s* of an ET. The unit vector e defines the direction of the translation or rotation (depending on type of the ET). The parameter *s* (i.e. the value of the translation or rotation) is either a *constant value* or a certain *time-dependent function*, or a *variable value*, which must be calculated at simulation time. In the latter case, the parameter *s* is *a local generalized joint coordinate*.

Thus, there are six types of ET:

- tc – translation with a constant parameter;
- tv – translation with a variable parameter;
- tt – translation whose parameter is a known time function;
- rc – rotation with a constant parameter;
- rv – rotation with a variable parameter;
- rt – rotation whose parameter is a known time function.

Let us consider some examples of generalized joints.

a) A rotational kinematic pair (Figure 2.2, Figure 2.4 left). It is described by three ETs:

$$T_1 = \{tc, \underline{e}_y, a\}, T_2 = \{rv, \underline{e}_x, \varphi\}, T_3 = \{tc, \underline{e}_y, b\}, \underline{q}_{ij} = [\varphi].$$

Before starting the explanation of the expressions, it seems appropriate to make clear the introduced designations. T_1, T_2, T_3 stand for the consecutive ETs. Then in brackets, the type of ET, the vector e and the parameter s are given. For all the types of ETs, except constant translation *tc*, the vector e must be unit. For *tc* there are no parameters since the value of translation is directly included in the vector of transformation. The column q_{ij} contains the list of local joint coordinates.

The first ET translates SC_1 by *a* along axis *y*. The second ET corresponds to the rotation by φ about *x*. The final ET (transition along axis *y*) makes SC_1 coincide with SC_2.

b) A prismatic joint (Figure 2.4 right) in a simplest case can be described by a single ET

$$T_1 = \{tv, e = (0,1,0), s = y\}, q_{ij} = [s];$$

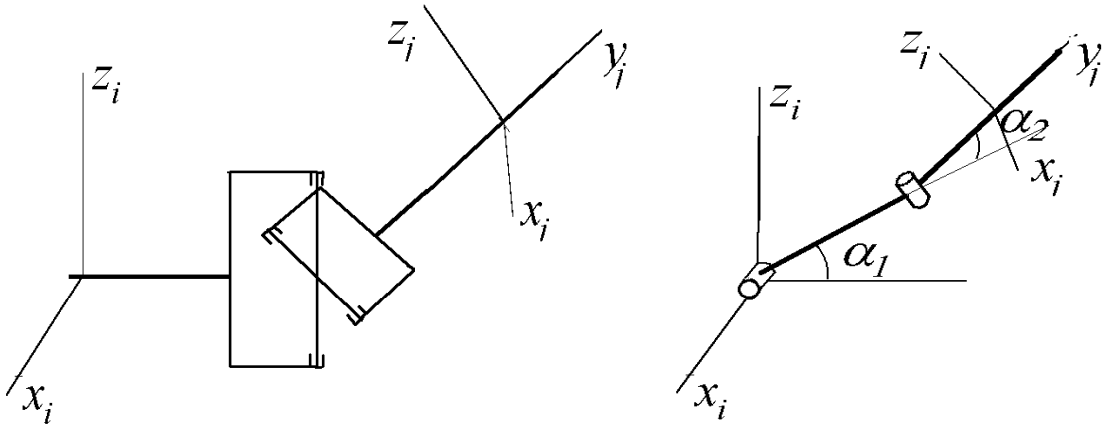
s is the coordinate which determines the translation of body 2 with respect to body 1.

c) A Cardan joint (Figure 2.2, Figure 2.7) is a kinematical pair having two rotational d.o.f. In the case sketched in the figure it is set by four ETs:

$$T_1 = \{tc, e = (0, a, 0)\},$$
$$T_2 = \{rv, e = (1,0,0), s = \alpha_1\},$$
$$T_3 = \{rv, e = (0,0,1), s = \alpha_2\},$$
$$T_4 = \{tc, e = (0, b, 0)\},$$
$$\underline{q_{ij}} = [\alpha_1, \alpha_2]^T.$$

T_1 translates SC_1 by a along the axis y and the axis x occupies the position of the first axis of rotation. The second ET (the rotation by α_1 about *x*) brings the axis *z* into coincidence with the axis of the second rotation. These are followed by the rotation by α_2, which makes the axes of the SC parallel to those of SC_2. The final ET – the translation along *y* – results in the coincidence with SC_2.

Note. Every next ET is done relative to the new position of SC, in which the preceding ETs resulted. As a rule, ETs cannot be exchanged, thus, in other words, ETs are not commutative.



a) b)

Figure 2.8. Two joints with the same set of ETs but with different their sequence

E.g., if, in the case of the Cardan joint shown in Figure 2.7, one exchanges the second and third ETs, it will result in a Cardan joint with a totally different kinematic structure (Figure 2.8 a). If one exchanges the first and second ETs, the resulting pair will not be a Cardan joint at all but a joint with two d.o.f. (Figure 2.8b). This pair is also available with UM, although it by no means corresponds to Figure 2.3. These examples show that the concept of common joint is far not trivial and while describing it there may appear serious and hardly detectable errors.

d) A spherical joint. Bodies connected by spherical joints may have arbitrary relative orientation. In the example of a Lagrange top shown in Figure 2.6 (a symmetrical body connected with

the support by a spherical joint whose center lies on the axis of symmetry), the transfer from SCI to SCJ is carried out in two stages. First, rotate SCI so that its orientation coincide with SCJ and then make a translation along the axis *z*. Bringing the two systems of coordinates into coincidence might be obtained by rotating about three axes. So three angles of orientation are introduced. To introduce the three angles one can use any of the twelve allowable combinations of the axes of rotation. However, in the case of a Lagrange top Euler angles are conventional. Thus, here the spherical joint is described by the following four ETs:

$$T_1 = \{rv, e = (0,0,1), s = \psi\},$$
$$T_2 = \{rv, e = (1,0,0), s = \vartheta\},$$
$$T_3 = \{rv, e = (0,0,1), s = \varphi\},$$
$$T_4 = \{tc, e = (0,0,a)\},$$
$$q_{ij} = [\psi, \vartheta, \varphi]^T$$

And the final example is a free body in space. It is a joint with six d.o.f. and six ETs: three variable translations along the axes *x*,*y*,*z* and three variable rotations about the axes *z*,*x*,*z* – Euler angles, or *x*,*y*,*z* – Cardan angles, or, at last, any series of rotations about three arbitrary axes (one has to bear in mind, however, that the axes of two consecutive rotations must not be parallel). For instance, for Cardan angles:

$$T_1 = \{tv, e = (1,0,0), s = x\},$$
$$T_2 = \{tv, e = (0,1,0), s = y\},$$
$$T_3 = \{tv, e = (0,0,1), s = z\},$$
$$T_4 = \{rv, e = (1,0,0), s = \alpha\},$$
$$T_5 = \{rv, e = (0,1,0), s = \beta\},$$
$$T_6 = \{rv, e = (0,0,1), s = \gamma\},$$
$$q_{ij} = [x, y, z, \alpha, \beta, \gamma]^T.$$

When solving various problems of kinematics and inverse problems of control ETs whose parameters are explicit time functions are frequently used. It means that how the angle of rotation or the parameter of translation changes with time is known a priori. E.g., while solving the kinematic problem of the crank-and-slide mechanism the motion of the input link – the crank – is given. Here the angle of rotation in the joint connecting the crank with the fixed support changes in a known manner, for example, it can be linear with respect to time.

Remark. The generalized joint with three rotational d.o.f. can be singular at some relative orientations of the bodies in pair (see remark in Sect. 2.2.2.2. *"Six d.o.f. joint"*, p. 2-14). In case if the bodies can be arbitrary oriented while moving, values of the orientation angles may become (almost) singular. This results in a strong deceleration or a total break of simulation of motion due to automatic decreasing the integration step. Because of this reason, in this case it is recommended to use the quaternion joint, which does not have any singularities.

2.2.2.4. Quaternion joint

The quaternion joint is similar to that with 6 d.o.f. The most significant difference consists in coordinates, which define the orientation of SC2B relatively to SC1A. In the case of a quaternion joint we have four coordinates (a quaternion) q_0, q_1, q_2, q_3. As it is known, the quaternion cannot be singular, but it satisfies the identity

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

at any moment.

Translational degrees of freedom can be turned off similar to the 6 d.o.f. joint, but rotational degrees of freedom are always presented.

The quaternion joint is mainly used for introducing coordinates of freely moved bodies, as well as for introduction of spherical joints.

Remark. If a spherical joint is cut (Sect. 2.2.3. *"System graph. Closed kinematical loops"* p. 2-27), its description by a quaternion joint is the most effective way from the numerical point of view.

2.2.2.5. Internal body joint

Internal body joint is often used by introducing 6 degrees of freedom to a body, which can move freely. Orientation angles in these case correspond to the sequence of rotations 1,2,3 (Cardan angle).

A large weight is automatically assigned to the internal body be the program. Due to this fact, if a joint or a chain of joints setting position of the given body relative to the Base0 is introduced, the internal joint is cut and automatically **removed**. This property of internal joints is frequently used in subsystems in particularly in specialized UM modules such as UM Caterpillar and UM Train3D.

Consider an example. In the UM Caterpillar module the model of a track is automatically generated as an included subsystem. All elements in the subsystem, which must be connected with the hull, are connected with a fictitious body. The fictitious body has six d.o.f. introduced by an internal joint. In the model of a tracked vehicle, the model includes a hull with 6 d.o.f. and two subsystems – tracks. Fictitious bodies from each of the subsystems are rigidly fixed relative to the hull by special zero d.o.f. joints. In this case the internal joints of fictitious bodies are cut and automatically removed.

2.2.2.6. Weightless rod constraint

This type of constraint corresponds to a weightless rod with spherical joints at the end points connecting two bodies. There is no friction in the joints. To describe the constraint it is enough to set the coordinates of the attachment points of the rod in the body-fixed SC of both bodies and a nonzero length of the rod. The length of the rod may be either constant or an explicit time function, which greatly expands the area of its application to analyze controlled systems. E.g., an actuator may be considered a massless rod with the length varying somehow in a number of problems where its inertial parameters are insignificant.

2.2.2.7. Mates

Mates are constraints, which limit relative position and motion of pairs of bodies. The notion ‘mate’ is introduced in CAD programs and appears in UM in connection with development interfaces to CAD programs such as SolidWorks, Autodesk Inventor, KOMPAS. As a rule, mates appear in UM models as a result of converting CAD assemblies into UM objects when some constraint cannot be converted automatically in joints described above.

Mates realized in UM a described by the following data.

1. Type of mate
   - Coincident
   - Concentric
   - Parallel
   - Distance
   - Angle
2. Type and parameters of manifolds connected with each of the in the pair of bodies
   - **Point.** Described by coordinates in the body-fixed SC.
   - **Line.** Described by coordinates of one point on the line and a unit vector along it in the body-fixed SC.
   - **Plane.** Described by coordinates of one point on the plane and a normal to the plane in the body-fixed SC.
3. Parameters, which depend on the mate type.
   - Distance. Distance between the manifolds should be specified.
   - Angle. Value of angle between the manifold must be set.

There exist limitations on type of manifolds for some types of mates. For instance, manifolds cannot be points for mates “coincident”, “parallel” и “angle”.

Example. Mate of the “coincident” type with points as manifolds for both of the bodies, i.e. the coincidence of a point of the first body with another point of the second body, introduces the same constraints on relative motion of the bodies like a spherical joint. The main difference consists in the fact that if the spherical joint is not cut, it introduced 3 angular degree of freedom defining orientation of the second body relative to the first one. The mate always adds constraint equations, 3 equations in the considered case:

$$r_1 + A_{01}\rho_1^1 - r_2 - A_{02}\rho_2^2 = 0.$$

Here r_1, r_2 are the radius-vectors of body-fixed system of coordinates, A_{01}, A_{02} are the direct cosine matrices of body-fixed SC, ρ_1^1, ρ_2^2 are the radius-vectors of coinciding points, which are specified in the body-fixed SC.

Number of constrain equations for all possible types of mates is specified in the table.

Type of mate	Types of manifolds	Number of constraint equations
Coincident	point-point	3
	point-line	2
	point-plane	1
	line-line	4
	line-plane	2
	plane-plane	3
Concentric	line-line	4
Parallel	line-line	2
	line-plane	1
	plane-plane	2
Distance	point-point	1
	point-line	1
	line-line	3
	line-plane	2
	plane-plane	3
Angle	line-line	1
	line-plane	1
	plane-plane	1

2.2.2.8. Convel joint

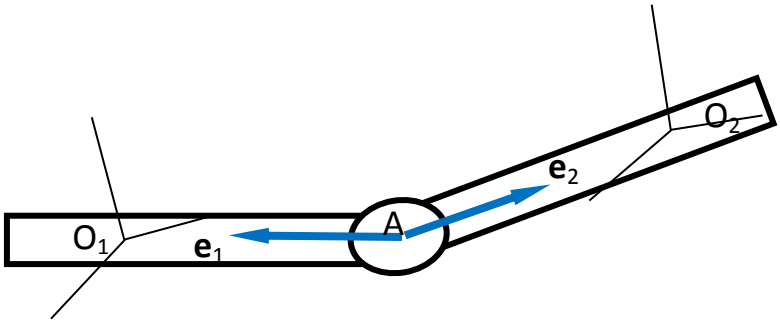


Figure 2.9. Scheme of ConVel joint

A ConVel (constant velocity, CV) joint provide equal angular velocities of a pair of inclined shafts. This constraint corresponds to a point-point coincidence mate as well as to equal shaft angles of rotation.

The joint is specified by the coordinates of the joint center A in SC of each of the bodies as well as by two unit vectors e_1, e_2 for shaft axes, Figure 2.9.

The joint adds *four constraint equations*. Three of them correspond to coincidence of two points A of shafts. The fourth equation provides the equality except for sign of angle of rotation of body 2 about vector e2 and the angle of rotation of body 1 about the vector e1.

Remark. In fact, the joint does not provide constant angular velocities of shafts. The name of the joint reflects its property to keep (nearly) constant velocity of the second shaft by the constant angular velocity of the first one, unlike the Hook joint. Besides, angular velocities of shafts are (nearly) equal.

To make the CV joint model correctly, the user should create a correct kinematic scheme of shafts by proper choice of joints and/or force elements.

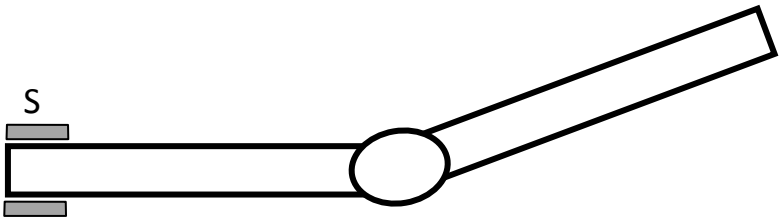


Figure 2.10. Model configuration with additional rotational joint

At the beginning consider a model, which does not correct with respect to the CV joint. In this model the first shaft is connected with the base by a rotation joint S, Figure 2.10, and 6 d.o.f. are introduced for the second shaft relative to SC0. In this model a CV joint reduces the degrees

of freedom of the second shaft to two rotational d.o.f. relative to shaft 1. In such the model, the second shaft does not keep it orientation along the axis e_2, and the shaft motion is not correct.

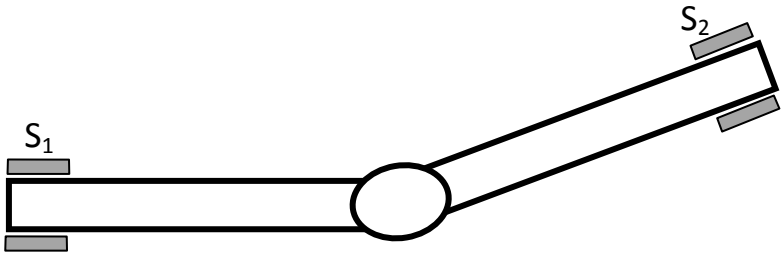


Figure 2.11. Model with two additional rotational joints

The first variant of the possible models is shown in Figure 2.11. In this model each of two shafts is connected with the base by rotational joints S_1 and S_2. If the joint axes lie exactly on straight lines specified by CV joint vectors e_1, e_2, Figure 2.10, the mechanism works properly. At the same time, the second shaft without CV joint has 1 d.o.f. only, and adding the CV joint introduces four constraints. This means, the model has redundant constraint and statically uncertain. Reaction joints cannot be computed properly. There are other reasons why this model cannot be recommended. For example, if joint axes do not coincide exactly with CV axes e_1, e_2, the mechanism cannot move at all. Most likely, the user gets a message about non-consistent equations of motion by the start of simulation. As a rule, this scheme is incorrect if the shafts are connected with different bodies, which can move relative to each other.

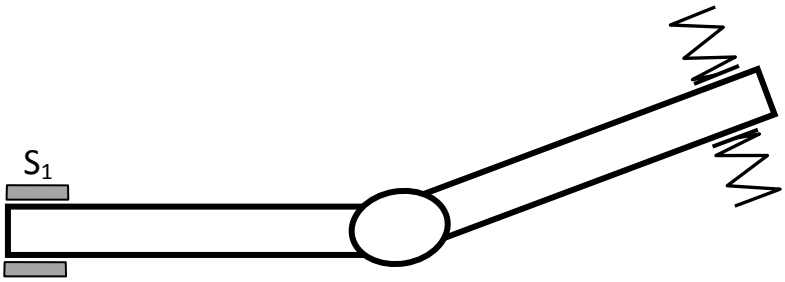


Figure 2.12. Model with one rotational joint and with one bushing

The model shown in Figure 2.12 is better than the previous one in many cases. Here we see a bushing for the second shaft instead of a joint. The bushing limits shifts of the body in directions perpendicular to the shaft axis. A special force of the ‘bushing’ type as well as a generalized linear force element can be used for the bushing model. In an alternative model the bushings are introduced for both shafts instead of rotational joints. The user must take care of the directions in which the bushings are blocks shaft displacements.

Example. See the User’s Manual, Chapter 7, Sect. *Convel joint*.

2.2.3. System graph. Closed kinematical loops

An important stage in dealing with complex mechanical systems is the analysis of their structure, which can be carried out with the help of the system graph. Its vertices correspond to the bodies including the basic one; its ribs correspond to the joints. The above conditions of connectivity being met, the graph of the system is connected, i.e. there is at least one path between any two of its vertices. That the graph satisfies the connectivity condition is the corollary of the condition that there is a chain for each body that connects it with body 0 (*base*, SC0). If there is but one path between any pair of vertices, the graph is a tree, and in this case the corresponding mechanical system has a tree structure. If at least one pair of bodies between which there exists more than one path can be found, the graph of the system has cycles and the MBS contains closed kinematical loops. Then it is clear that there are bodies in the systems, which are connected with body 0 by more than one chain.

A majority of mechanisms has closed loops. To analyze systems with closed loops is more difficult than if the system is a tree. The most effective strategy here consists in cutting a few joints so that a tree could be obtained. The number of cut joints equals that of independent cycles in the graph. It is easy to see that almost always to choose which joints to cut is ambiguous and there is a chance of picking them out best or optimally considering this or that criterion, e.g., to facilitate the equations of motion, reduce their size, decrease the number of arithmetic operations in the numerical modeling of motion, etc. In UM the choice of such joints is made automatically through the analysis of the graph. The optimal cutting is based on the Dijkstra algorithm for obtaining paths of minimal weight from the root of the graph to each vertex.

As was noted in the section devoted to system connectivity (Sect. 1.3.1), to generate the equations in a symbolic form requires that the joints in the chain between the basic body and anybody of the system must introduce coordinates. Thus, a rod joints must be cut. The condition of connectivity, for this reason, requires sometimes the introduction of additional joints with six d.o.f. for a spatial problem and three d.o.f. for a 2D one.

Remark 1. The user can have an influence on the choice of joints to be cut by means of large weight coefficient for the corresponding joint in the closed loop.

Remark 2. Cut joints with 6 degrees of freedom are conditionally removed, i.e. the corresponding coordinates are not included in the set of object coordinates.

2.3. Equations of motion

Equations of motion of a multibody system have the following form of differential-algebraic equations:

$$M(q,t)\ddot{q} + k(q,\dot{q},t) = Q(q,\dot{q},t) + G^T(q)\lambda, \qquad (2.1)$$
$$h(q,p) = 0,$$

where q is the column of basic coordinates of the system, p is the column of auxiliary coordinates (local joint coordinates in cut joints; M is the mass matrix, k,Q columns of generalized inertia and applied forces; λ are Lagrange multipliers corresponding to reactions in cut joints; the second equation in Eq. (2.1) is the algebraic constraint equation corresponding closure conditions of cut joints. Matrix G is the Jacobian matrix of the constraint equations after elimination of auxiliary coordinates.

UM generates the equations of motion in both a symbolic and in numeric-iterative form. Symbolic generation of equations makes often simulation faster.

Consider expressions generated by UM for each simple object. Let $q = \{q,p\}$ be the set of coordinates of the object. It contains the local joint coordinates for normal and fictitious joints (except the cut normal joints. UM generates in a symbolic form the following kinematic relations:

- The radius-vectors of mass centers and rotation matrices for each body,

$$r_i^0 = r_i^0(q,t), \quad A_{i0} = A_{i0}(q,t),$$

- The velocities of mass centers and angular velocities for each body,

$$v_i^0 = D_i^0\dot{q} + v'^0_i, \quad \omega_i^i = B_i^i\dot{q} + \omega'^i_i,$$

- Note that all quantities relating to body rotations are given in the body-fixed SC, whereas all translational quantities are set in the SC0;
- The matrices D_i^0 and B_i^i included in the expressions for the velocity of the center mass and angular velocities (Jacobean matrices);
- The vectors a'^0_i and ε'^i_i of the terms of accelerations and angular accelerations not depending on $\ddot{q}$,

$$a_i^0 = D_i^0\ddot{q} + a'^0_i, \quad \varepsilon_i^i = B_i^i\ddot{q} + \varepsilon'^i_i;$$

- Constraint equations for the cut joints

$$h_k(q,p_k) = 0.$$

- Elements of the matrices M,k,Q of the motion equations (2.1).

2.4. Theoretical foundations for solving constraint equations

Theory of numerical solving differential-algebraic equations asserts that the numeric solvers are sensitive to errors in initial conditions. This means that the initial values of coordinates and their first time derivative should satisfy the constraint equations as exactly as it is possible. To clarify the problem it is good to consider the procedure of forming and solving the constraint equations.

Let q_0 , p_0 be the current values of generalized (coordinates in joints, which belongs to the object tree, i.e. they are not cut) and auxiliary coordinates, respectively. These values, as a rule, do not satisfy the constraint equation

$$h(q,p) = 0,$$

which corresponds to the closure conditions for the cut joints. The corrections Δq, Δp must be set so that new coordinate values satisfies the equation

$$h(q_0 + \Delta q, p_0 + \Delta p) = 0. \tag{2.2}$$

The Newton-Raphson iterations are used by UM for solving the nonlinear Eq.(2.2). The following equations are solved at every the iteration:

$$H_q(q^k,p^k)\Delta q^{k+1} + H_p(q^k,p^k)\Delta p^{k+1} = -h(q^k,p^k),$$
$$q^{k+1} = q^k + \Delta q^{k+1}, p^{k+1} = p^k + \Delta p^{k+1}, k = 0,1\ldots$$

$$q^0 = q_0, p^0 = p_0.$$

Here H_q, H_p are the Jacobian matrices of the vector h. The k-th iteration is done according to the following pattern. The constraint equations

$$H_{q,i}\left(q^k,p_i^k\right)\Delta q^{k+1} + H_{p,i}\left(q^k,p_i^k\right)\Delta p_i^{k+1} = -h\left(q^k,p_i^k\right), \tag{2.3}$$

are generated for each cut joint j. This equation is in the generalized coordinates and local auxiliary variables (i.e. the local coordinates in the cut joint). The auxiliary variables Δp_i^{k+1} are eliminated from Eq.(2.3) with the help of the Gauss elimination procedure and the following two equations are obtained:

$$\Delta p_i^{k+1} = P_i\Delta q^{k+1} + \delta p_i^k, \tag{2.4}$$
$$G_i^k\Delta q^{k+1} = -g_i^k. \tag{2.5}$$

Then Eqs. (2.5) for separate cut joints are added to the matrix equation

$$G^k\Delta q^{k+1} = -g^k, \tag{2.6}$$

whose solution are obtained using the Gauss elimination procedure based on the row pivoting. The correction values for the pivotal elements (independent variables) are set to zero and non-zero values of the dependent variables are calculated according to Eq. (2.6). The obtained solution Δq^{k+1} is substituted into Eqs. (2.4) for evaluations of the auxiliary variables. If the norm of the calculated correction vector is less that the prescribed small tolerance ε, the iteration is stopped and the program exits from the procedure otherwise the next iteration is executed.

Note that according to this algorithm, an arbitrary solution to the equations is obtained in agreement with the constraint equations. Fixing coordinates (e.g., when it is necessary to obtain the configuration of a four bar mechanism for a certain angle of crank rotation, Figure 2.13), can

be done by the user but only for generalized coordinates (not auxiliary!). The **fixed** coordinate remains unchanged during the above iterations.

The process of automatically solving the constraint equations can fail. If it does not converge after 20 iterations, you receive the corresponding message and can either continue the process or quit.

Consider some situations when iterations do not converge.

- **The constraint equations have no solution**

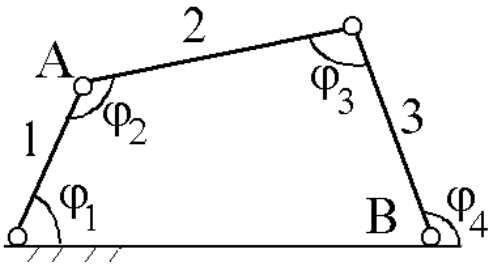


Figure 2.13. Four bar mechanism

That means the mechanism has not been assembled correctly. For instance, consider a planar mechanism shown in Figure 2.13. If one sets the lengths of bodies 1, 2 and 3 so that their sum is less than the distance between the hinges connecting body 1 and 3 to the support, the mechanism cannot be assembled. This occurs, too, when the axes of rotation of the joints are not parallel.

UM cannot see if your mechanism can or cannot be assembled. Although it sends a message notifying that iterations do not converge. Of course, you have to find out the reason why this happened and make changes in the object description using the Input Module or correcting the values of some identifiers.

- **Constraint equations have no solution for the given values of the fixed coordinates.**

If you have *fixed* some coordinates (see previous sections) so that their values cannot be changed during iterations, sometimes the solution cannot be found. It can be for two reasons. First, the set of fixed coordinates contains dependent coordinates. For example, if you have fixed any *two* coordinates for the mechanism shown in Figure 2.13, the solution cannot be found because the mechanism has one d.o.f. and any two coordinates are dependent. Second, if the current value of a fixed coordinate is outside of its interval. This occurs (for the case of mechanism in Figure 2.13) if you have fixed the angle φ_1 and the distance between points A and B is greater than the sum of the lengths of bodies 2 and 3.

- **Bad starting approximation**

It is well known that the Newton-Raphson procedure requires a good starting approximation q_0, p_0 for successfully solving non-linear equations. If your current coordinate values are far from the desirable solution, the following variants may take place:

  - Iterations do not converge. Try to continue iterations for several times. If the solution is not found jet, try to set the start values of coordinates manually.

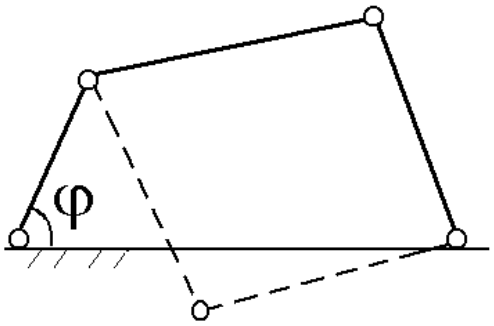


Figure 2.14.

- UM obtains an undesirable solution. This problem arises due to lack of uniqueness of nonlinear constraint equations. One example is given in Figure 2.14 (the dashed line). Use a manual choice of coordinates to improve the approximation.
- For the given values of coordinates q_0, p_0 the Jacobian matrix of the constraint equations (2.6) is singular. For the mechanism in Figure 2.14, a singularity of the given kind is encountered if the joint between bodies 1 and 2 is cut and the angles in joints satisfy $\varphi_3=0,\pi$, $\varphi_4=\varphi_1+\varphi_3$. Use a manual choice of coordinates to avoid the singular position.
- The Jacobian matrix $H_{p,i}$ in Eq. (2.4) is singular for the given values of q_0, p_0. Use a manual choice of coordinates to avoid the singular position.

2.5. Force elements

Forces acting between bodies are generally divided into applied and constraint reactive forces. In their turn, reaction forces are represented by two components: the tangential component performing mechanical work during motion (as a rule, these are friction forces) and an ideal – or normal – component. If all the constraints in the MBS are ideal, the first component is absent. In terms of data input applied forces and the reaction force non–ideal components have the following in common: they should be expressed through the variables and parameters of the system. However the ideal components of reaction forces are determined by the type of constraint and their computation is carried out automatically.

In UM the following active forces varying in the patterns of their description in the data input module, are available:

- gravity forces;
- joint forces (for translation, rotational and generalized joints);
- bipolar forces;
- scalar torque;
- generalized linear force elements;
- contact forces;
- special forces (gearing, bushing etc.);
- T-forces;
- externally formulated forces.

An applied force may be a function of time, coordinates and their first time derivatives. In simpler cases, (e.g., for gravity and generalized linear force element) these functions are automatically generated by the program. However, they are so often quite complicated and the user has to write his/her own procedures in a *control file* or use *external libraries*. Such forces are referred to as *externally formulated forces*.

2.5.1. Gravity

In UM the gravitation field is assumed homogenous. Thus it is only necessary to specify the unit vector of the gravity direction. The gravity acceleration is assumed 9.81 m/s^2. However, the user can leave gravitation out of account. The generalized forces corresponding to gravity are generated automatically.

2.5.2. Joint forces and torques

One way to set an active force or torque is to introduce these for the corresponding degree of freedom. Thus, one models an engine, actuator (transmission) and other systems where dynamic effects are insignificant. E.g., in a robot manipulator model the designer has the right not to take into account inertial properties of the transmission members and therefore not to solve the dynamic problem. So the model is idealized, and the designer assumes that the influence of the transmission on the object is reduced to the appearance of driving forces or torques in the kinematic pairs. Such force (torque) is directed along the axis of the pair.

To model such forces in the program the definition of a joint force (torque) for generalized, rotational and translational joints is introduced. In the case of a generalized joint, a force (a torque) can be introduced for any ET with a variable parameter (i.e. *tv* or *rv*, see Sect. 2.2.2.3. *"Generalized joint"* p. 2-17). The force (torque) vector is directed along the axis of ET according to the increase of the parameter values. It is assumed that the force is only a function of time, the corresponding ET parameter and its time derivative (that is velocity). This limitation is removed if the user applies the external description of the force, in other words, if the force computation is done while the modeling of motion with the help of a procedure written by the user in a *control file*.

For a joint with several d.o.f. it is possible to introduce a force for each d.o.f., or for some of them.

2.5.3. Bipolar forces

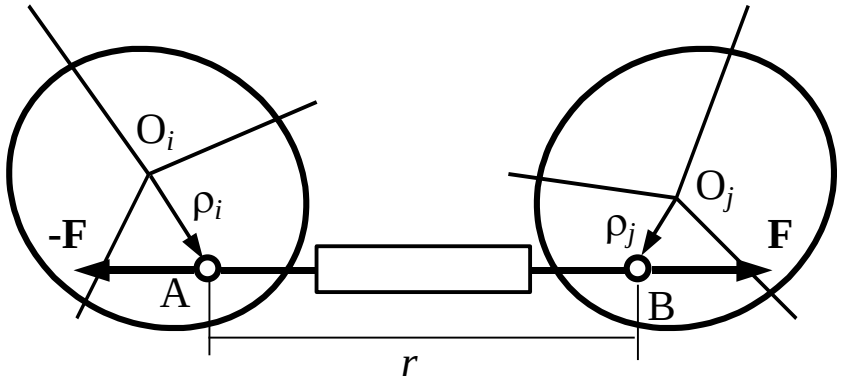


Figure 2.15.

A bipolar force element connects two chosen fixed points of a pair of bodies (attachment points O_i, O_j in Figure 2.15). The force acts along the straight line between the points and may depend on time *t*, the distance *r* between the points and its time derivative $\dot{r}$,

$$F = F(t, r, \dot{r}).$$

The force is positive in the repulsion case, for example, it is positive in the case shown in Figure 2.15.

In the **UM input** program some of the most often met types of the dependencies between the bipolar force and the variables are available: a linear function, an analytical expression, a set of points etc. Mathematical models for this dependence are described in Sect. 2.5.5. *"Types of scalar forces"* p. 2-37.

If the distance *r* equals zero, the degeneration of the force element occurs (due to the uncertainty of the force direction). Here, the force is assumed to be zero.

Example. Consider a bipolar force, which models a linear viscoelastic force element with c and *d* parameters as stiffness and damping coefficients. Let the force equals F_0 when the length of the element is x_0 and the velocities vanish. The analytic expression for the force looks like

$$F = F_0 - c(x - x_0) - dv.$$

2.5.4. Scalar torque

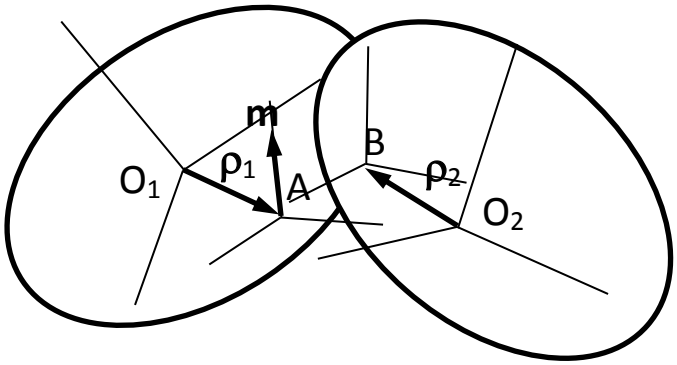


Figure 2.16. To the notion of scalar torque

This type of force element describes torque acting on the second body from the first one/ To clarify the model of the torque consider interacting bodies 1 and 2, Figure 2.16. Points O1 and O2 are the origins of the body-fixed SC1 and SC2. Let us introduce additional local system of coordinates SCA1 and SCB2, rigidly connected with bodies 1 and 2 respectively. Arbitrary points A and B are origins of local SC. A scalar torque **m** depends on orientation of SCB2 relative to SCA1 and does not depend on their origin positions. Moreover, it is supposed that **Z axes of SCB2 and SCB1 are nearly parallel**, i.e. the angle between these axes is small during motions of bodies and does not exceed 10 degrees. By these assumptions, let us shift axes of SCB2 to the origin A.

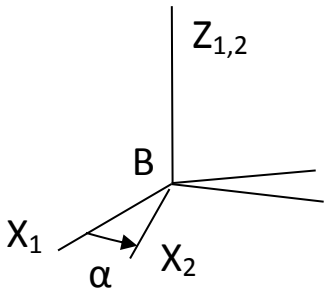


Figure 2.17. Angle of rotation of the second body relative to the first one

Let us introduce angle α of rotation of the second body relative to the first one about Z axis as an angle between the X-axes of SCB2 and SCA1, Figure 2.17. For simplicity directions of Z axes on this figure coincide.

Scalar torque acting on the second body from the first one is directed along Z axis of SCA1. Value of the torque depends on the angle α, its time derivative $\dot{\alpha}$, and time *t*

$$m = m(\alpha, \dot{\alpha}, t).$$

Mathematical models for this dependence are described in Sect. 2.5.5. *"Types of scalar forces"* p. 2-37.

Remark 1. If the second body is connected with the first one by a rotational joint, which axis coincides with the Z-axis of SCA1, and Z-axes of SCA1 and SCB2 coincides so that the angle α is equal to the joint coordinate, the scalar torque can be equivalently described by a joint torque, see Sect. 2.5.2 *"Joint forces and torques"* p. 2-34

Remark 2. In the description of the scalar torque, x, v letters are used as identifiers of the angle α and its time derivative $\dot{\alpha}$.

2.5.5. Types of scalar forces

Definition of a bipolar/joint/axle force as well as scalar torque includes its mathematical model as a scalar function $f = f(x, v, t)$. In the case of a joint force the arguments *x*, *v* are the joint coordinate and its time derivative, for a bipolar force element they are the element length and its time derivative. In case of a scalar torque they represent the relative angle of rotation and the corresponding angular velocity. Anyway, *t* is the current time value. The following types of description of the force model are foreseen in UM.

2.5.5.1. Linear force

The model corresponds to a linear viscoelastic interaction with a harmonic excitation:

$$f = F_0 - c(x - x_0) - dv + Q\sin(\omega t + \alpha).$$

Here F_0 is the constant component of the force, *c*, *d* are the stiffness and damping constants, x_0 is the value of the coordinate *x* for zero value of the elastic component, Q, ω, a are the amplitude, the frequency and the initial phase of the harmonic excitation.

The element is used for description of linear springs, damping elements, harmonic excitations and their combinations

Adding a new element of this type to a model see in Chapter 3, Sect. *Linear force element*.

2.5.5.2. Friction force

This type of force is mainly used for modeling frictional dampers. The force description includes two modes: sliding and sticking. In the sliding mode the force satisfies the formula

$$f = Fsgn(v)$$

analogously to the Coulomb friction with a constant friction force F, and v as a the sliding velocity. In the sticking mode, the force model looks like

$$f = f_0 - c(x - x_0) - dv,$$

i.e. presents a linear viscoelastic force with the c and d parameters as stiffness and damping coefficients.

The sliding-sticking transition occurs at the moment when the velocity *v* changes its sign. At this moment the force *f* and the coordinate *x* values are stored (the f_0 and x_0 parameters in the formula for the force in the sticking mode).

The sticking-sliding transition occurs when the force reaches its maximal value

$$|f| \geq \delta F,$$

where F_0 is the maximal value of static friction force.

The user should specify the parameters of the model F, c, d, F_0.

Force value in the sliding state F can be variable; in particular, it can depend on the reaction force in joint, see Section 2.5.10 *Friction in joints depending on reactions*.

Adding a new element of this type to a model see in Chapter 3, Sect. Friction and elastic-friction elements.

2.5.5.3. Elastic-frictional force

This is one of well-known models of friction, which consists of a dry friction with a linear spring in series. An additional damper is usually set in parallel with the spring (Figure 2.18).

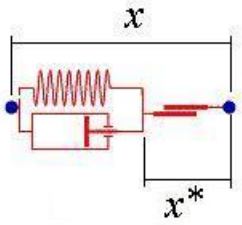


Figure 2.18.

The mathematical model of the element requires the introduction of an auxiliary variable x^* in addition to x, v variables. In a sliding mode the mathematical model includes the following differential equations relative to the x^* variable:

$$-d(v - \dot{x}^*) - c(x - x^*) = sgn\dot{x}^* \cdot F \qquad (2.7)$$

$$f = sgn\,\dot{x}^* \cdot F$$

Here *c*, *d* are the stiffness and damping constants, *F* is the constant friction force. The equation corresponds to a condition that the friction force is equal to the sum of the elastic and damping forces.

In a sticking the x^* variable is constant, and the force is equal to

$$f = -dv + c(x^* - x)$$

Stick-slip transition occurs when

$$|f| \geq \delta F,$$

where $\delta = \frac{\mu_0}{\mu} \geq 1$ if the static/dynamic coefficient of friction ratio.

Slip-stick transition occurs when

$$\dot{x}^* \dot{x}^{*-} < 0,$$

where $\dot{x}^{*-}$ is the value of the velocity $\dot{x}^*$ at the previous step of integration process. This condition corresponds to changing the sliding direction.

Consider a new variable

$$y = x - x^*,$$

which corresponds to an elastic deflection of the spring. In a sliding this variable satisfies the equation

$$\dot{y} + \frac{c}{d}y = -\frac{1}{d}sgn(v - \dot{y}) \cdot F, \tag{2.8}$$

and the force value produced by the element is $f = sgn(v - \dot{y}) \cdot F$. In a sticking the equations is

$$\dot{y} = v,$$
$$f = -dv - cy,$$

and the slip-stick transition occurs when

$$(v - \dot{y})(v^- - \dot{y}^-) < 0.$$

Note that in this formulation the model of the friction element corresponds to the compliant contact model [1].

Consider some properties of the variable y. At a sticking its behavior quite analogous to that of the variable x, so the most interesting is its value at slipping according to Eq. (2.8). Covert this equation to the following form

$$\dot{y} = \lambda y = \pm\Delta,$$

where $\lambda = \frac{k}{2\beta}, k = \sqrt{\frac{c}{m}}$ is the frequency of a body with the mass m on a spring with the stiffness c; β is the damping ratio of critical, $\Delta = \frac{F}{c}$ is the deflection of the spring by the friction force F. Usually the frequency k is large, and for a reasonable damping $(\beta < 1)$ the value of λ is large< and the differential equation is stiff. For example, if $k = 628$ rad/s (100 Hz) and $\beta = 0.1$ then $\lambda = 3040$.

If Δ is constant, the solution of the equation is

$$y = y_F + y_0\left(1 - e^{-\lambda t}\right), \;\; y_F = \pm\frac{\Delta}{\lambda}.$$

with $y(0) = y_0$ as an initial condition. If Δ is variable by virtue of variability of the friction force F, it can be shown by the singular perturbation method that the solution in the first approximation is analogous to the above one. It is a sum of the slow term y_F and a boundary layer function, which tents to zero very fast. The typical behavior of the variable *y* is shown in Figure 2.19. The left figure is obtained for $y_0 = 1$. In fact, we see here a fast tending the variable to zero. The right figure shows the $y(t)$ dependence for, $y_0 = 0$ for a process when sliding changes its direction several times. The variable is nearly constant at sliding and changes rapidly at sticking. Figures were obtained for the friction ratio δ = 1,2.

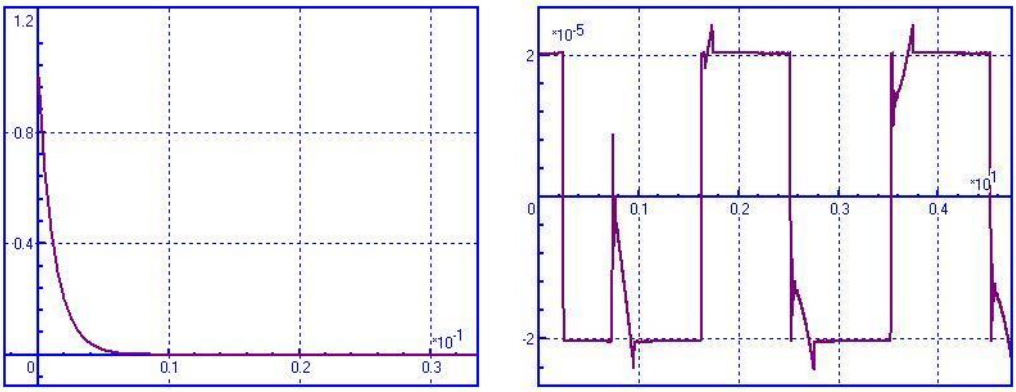


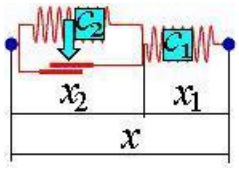
Figure 2.19. Variable y (elastic deflection) versus time for different initial condition

Note 1.	'Friction' and 'elastic-friction' force show in simulations similar results. The 'elastic-friction' force introduces an additional variable and makes equation of motion stiff. So, we recommend using the 'friction' force.
Note 2.	Model of the force contains a first order differential equation, but its internal representation is replaced by an equivalent second order differential equation $\ddot{z} + \frac{c}{d}\dot{z} = -\frac{1}{d} sgn(v - \ddot{z}) \cdot F, y = \dot{z}$

This replacement serves the unification of numerical methods.

Adding a new element of this type to a model see in Chapter 3, Sect. Friction and elastic-friction elements.

2.5.5.4. Elastic-frictional force 2



The force element is a spring (c_1) in series with a parallel combination of the second spring (c_2) and a Coulomb friction element. In contrary to the previous two model of the frictional element, here the friction force is not constant. Its value depends on a deflection of the spring in parallel c_2.

Consider mathematical model of the element. Let x_1, x_2 be the length of the springs, and $x_2(0) = 0$, $x_1(0) = x(0)$ be initial values of these variables. This means that at start of simulation the second spring c_2 has zero length and this position corresponds to its undeformed state. Let L_0 be the length of the spring c_1 in an undeformed state. Then the forces produced by the springs can be computed from the expressions

$$f_1 = -c_1\Delta x_1 = -c_1(x_1 - L_0) = -c_1(x - L_0 - x_2) = -c_1(\Delta x - x_2),$$
$$f_2 = -c_2\Delta x_2 = -c_2 x_2.$$

Here $\Delta x = x - L_0$.

As usual, the friction has two models. At sticking we accept a proportionality of the friction force to the force produced by the spring c_2

$$F_{fr} = -\mu sgn(\dot{x}_2)|f_2| = -\mu sgn(\dot{x}_2)|c_2 x_2| = -\mu sgn(\dot{x}_2)sgn(x_2)c_2 x_2$$

where μ is the dynamic coefficient of friction. The x_2 deflection can be computed from equality of two forces: the force in the spring c_1 and the sum of friction force and the force produced by the springc_2

$$f = f_1 = f_2 + F_{fr},$$

i.e.

$$-c_1(\Delta x - x_2) = -c_2 x_2 - \mu sgn(\dot{x}_2)sgn(x_2)c_2 x_2,$$

and finally

$$x_2 = \frac{c_1 \Delta x}{c_1 + c_2\big(1 + \mu sgn(\dot{x}_2)sgn(x_2)\big)} = \frac{c_1 \Delta x}{c_1 + c_2(1 \pm \mu)},$$
$$\Delta x_1 = x - x_2.$$

From the expression for x_2 we obtain that at sliding the variables Δx and x_2 have equal signs at least for $\mu < 1$. The force value is

$$f = \frac{c_1 c_2 \Delta x(1 \pm \mu)}{c_1 + c_2(1 \pm \mu)}.$$

The estimation $c_1 \gg c_2$ often takes place, which yields

$$f \approx c_2 \Delta x(1 \pm \mu).$$

In such cases the friction coefficient is a good approximation for the “relative (effective) friction coefficient” according to the estimate

$$\varphi = \frac{f_c - f_s}{f_c + f_s} \approx \mu,$$

where $f_c \approx c_2 \Delta x(1 + \mu)$, $f_c \approx c_2 \Delta x(1 - \mu)$ are forces at compression and stretching.

At sticking the deflection x_2 is a constant and the resultant force produced by the element is computed from the formula

$$f = f_1 = c_2(\Delta x - x_2),$$

Finally, slip-stick transition occurs when the velocity $\dot{x_2}$ changes its sign. The sign of the velocity is estimated on the difference $x_2 - x_2^-$, where x_2^- is the value of the coordinate at the previous integration step. Stick-slip transition occurs when

$$\big|F_{fr}\big| = |f_1 - c_2 x_2| > \mu_0 |c_2 x_2|,$$

where μ_0 is the static coefficient of friction.

List of parameter of the model:

c_1, c_2 – stiffness,

μ, μ_0 – static and dynamic coefficients of friction,

L_0 – length of element in an undeformed state.

A typical hysteresis (force versus coordinate) as well as coordinate x versus time are shown in Figure 2.20. The amplitude of vibrations decreases exponentially like a viscous damper, but the element realizes a frequency independent damping like rubber.

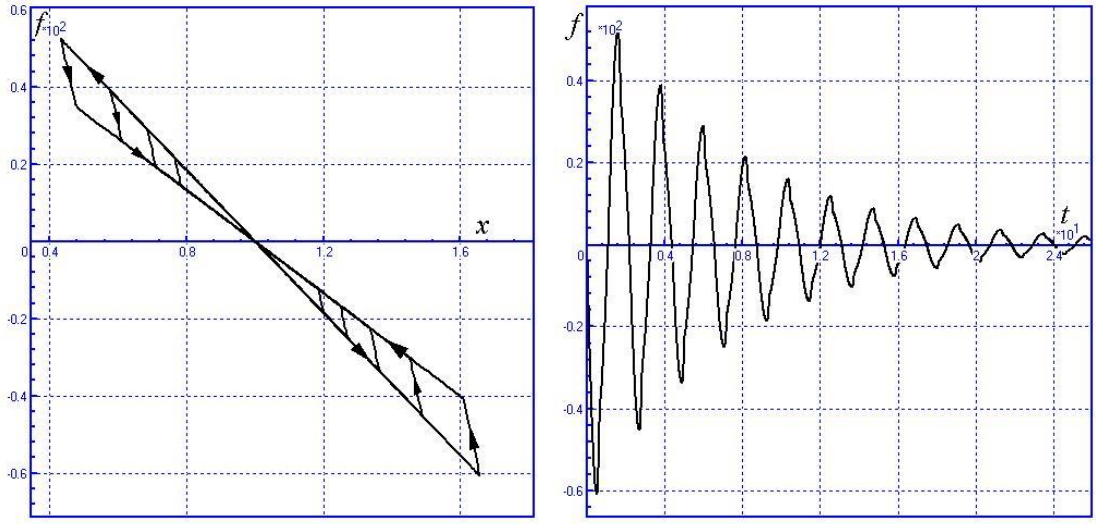


Figure 2.20. Force versus coordinate *x*; coordinate *x* versus time

Note 1.	Force element of this type can be used for modeling leaf springs, internal friction in rubber elements etc.
Note 2.	Element is automatically disabled if at least one of the parameters c_1 or c_2 is zero.
Note 3.	When friction is zero, the element corresponds to two springs in series.
Note 4.	Bipolar force element degenerates at zero length. If it passes through zero length, the simulation results are incorrect.
Note 5.	Do not set coefficient of friction more than 1.
Note 6.	Static and dynamic coefficients of friction are usually equal for elements of this type.
Note 7.	At sticking the element has no dissipates. If necessary, add dissipation in parallel (Sect. 2.5.5.16. *"List of forces"* p. 2-56).

Adding a new element of this type to a model see in Chapter 3, Sect. *Elastic-friction element 2*.

2.5.5.5. Stiffness and damping in series and parallel

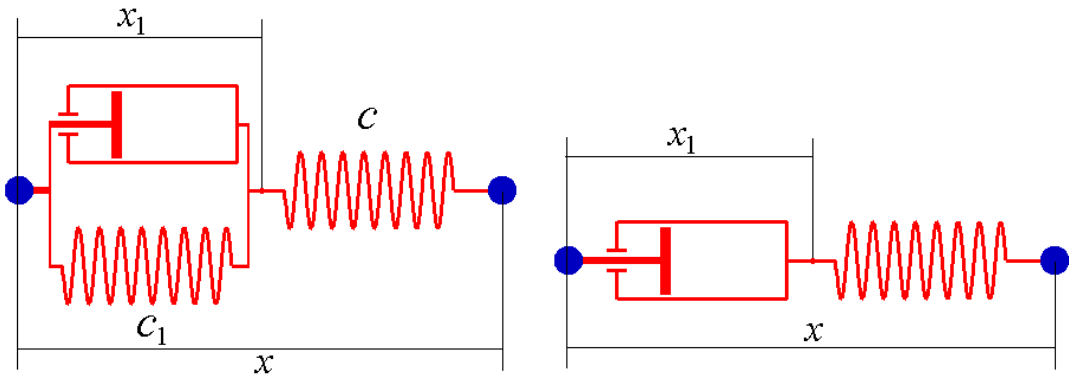


Figure 2.21. Scheme of the force element

The scheme of the element includes linear spring and in series linear spring damping in parallel (Figure 2.21, left). This element is used, e.g. in the Nishimura model of an air spring. In a particular case $c_1 = 0$ (Figure 2.21, right) the element is a linear spring and damper in series, and such elements are used for modeling of dampers, as a part of models of rubber, elastomer, etc.

Mathematical model of the element is obtained from equality of elastic and viscoelastic forces and includes the following differential equation

$$v\dot{x}_1 + c_1 x_1 = c(x - x_0 - x_1) = c(\Delta x - x_1),$$

where x_0 is the length of the unloaded element (if $c_1 \neq 0$) or the initial value $x_0 = x|_{t-0}$ (if $c_1 = 0$).

Thus, the element adds a new variable x_1 to the model and the corresponding differential equation. If the time constant

$$T = \frac{v}{c}$$

is small, the differential equation is stiff. In such cases the Park solver with computation of Jacobian matrices is recommended. It is worth to note that if *T* is small, and the analyzed object motion is slow, the element is equivalent to a simple linear damping with the same damping constant.

Note 1. Model of the force contains a first order differential equation, but its internal representation is replaced by an equivalent second order differential equation

$$\ddot{z} + \frac{c + c_1}{v}\dot{z} = \frac{c}{v}\Delta x,$$

This replacement serves the unification of numerical methods.

Note 2. Usually the initial value of x_1 variable (more accurately, initial for *z* and $\dot{z}$) is zero. Adding a new element of this type to a model see in Chapter 3, Sect. *Viscoelastic element*.

2.5.5.6. Nonlinear spring and damper in series

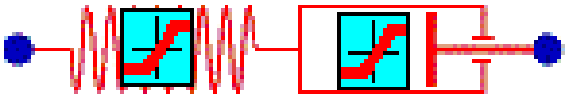


Figure 2.22. Scheme of force element

Here a generalization of the linear spring and damper in series is considered, when the spring and damper characteristics are specified by point curves, Figure 2.22. This element is used in cases when one characteristic or both of the characteristics are nonlinear.

The following recommendations are important.

1. Increase of the abscissa corresponds to compression of bipolar force element or decrease the joint coordinate, Figure 2.23.

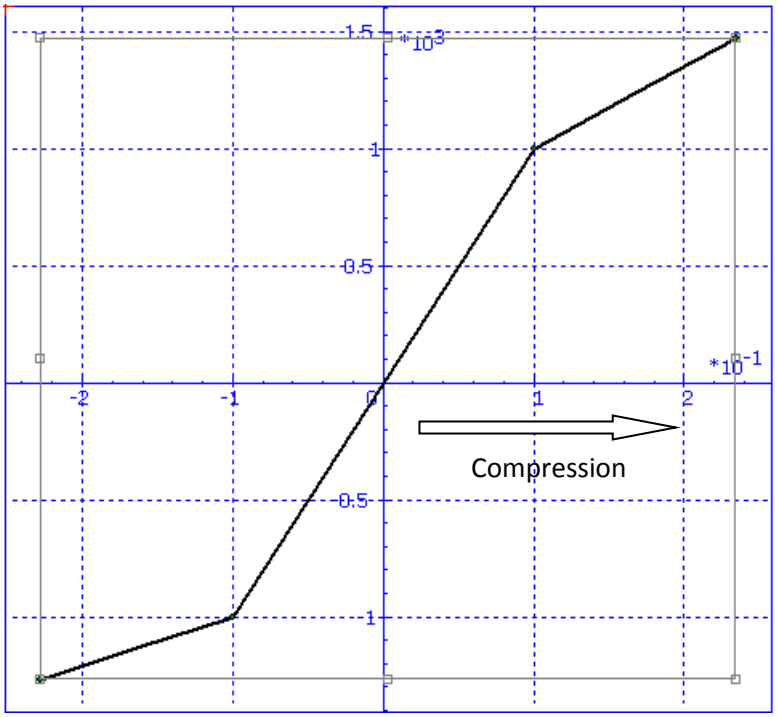


Figure 2.23. Example of a damper characteristics

2. The spring force versus deflection plot as well as damper force versus velocity plot must by **strongly monotone increasing** functions. The plots must pass through the origin (0,0).
3. Point coordinates can be parameterized by identifiers. Spline or B-spline interpolation can be applied.
4. If abscissa value exceeds the definitional domain during the simulation, the linear approximation if applied.

Like in the previous section, the element introduces an additional coordinate x_1 in the model as well as an ordinary differential equation, which implicit form is

$$F_v(\dot{x}_1) + F_x(\Delta x - x_1) = 0,$$

where F_v, F_x are functions specifying the spring and the damper.

The differential equation is often stiff, and the Park integration method with the use of Jacobian matrices is recommended. It is worth to note that if the spring stiffness is large, and the analyzed object motion is slow, the element is equivalent to a nonlinear damping.

2.5.5.7. Pointwise model

This type of the force requires the description of the $f(x), f(v)$ or $f(t)$ functions in a as a set of points. Coordinates of points can be both numbers and expressions. UM uses linear interpolation and extrapolation for calculation of force values in an arbitrary point. Figure 2.24 shows some force models, which can be easily realized with the help of this method. Each of the tree models requires four points for description.

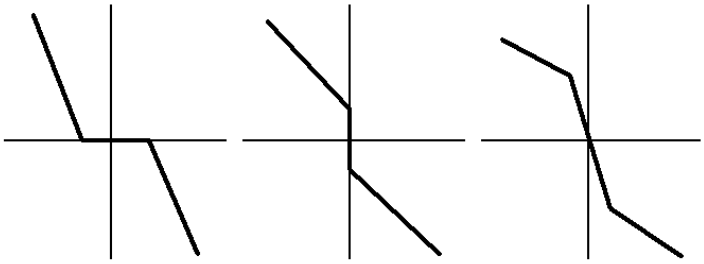


Figure 2.24. Example of nonlinear pointwise force models

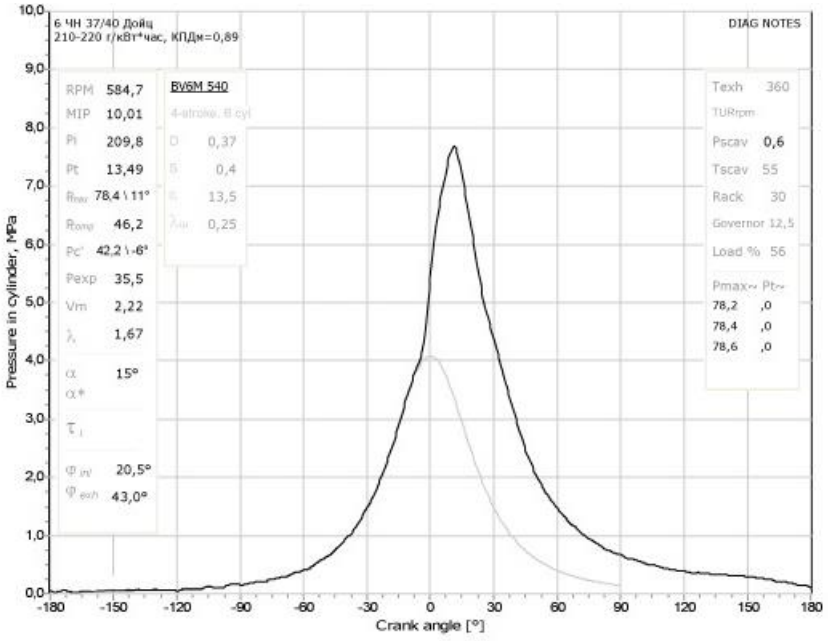


Figure 2.25. Example of indicator diagram

It is available the description of periodic forces, when the force is specified on one period only, Figure 2.25. As a rule, periodic forces are used for description of indicator diagrams of engines, compressors and so on.

Adding a new element of this type to a model see in Chapter 3, Sect. *Points (numbers), Points (expressions).*

2.5.5.8. Expression

The force model is an expression including standard elementary functions. For instance,

$$f = f_0 - c(x - x_0) - dv + a\,sin(15t).$$

Adding a new element of this type to a model see in Chapter 3, Sect. *Data types* | *Expression – explicit function*.

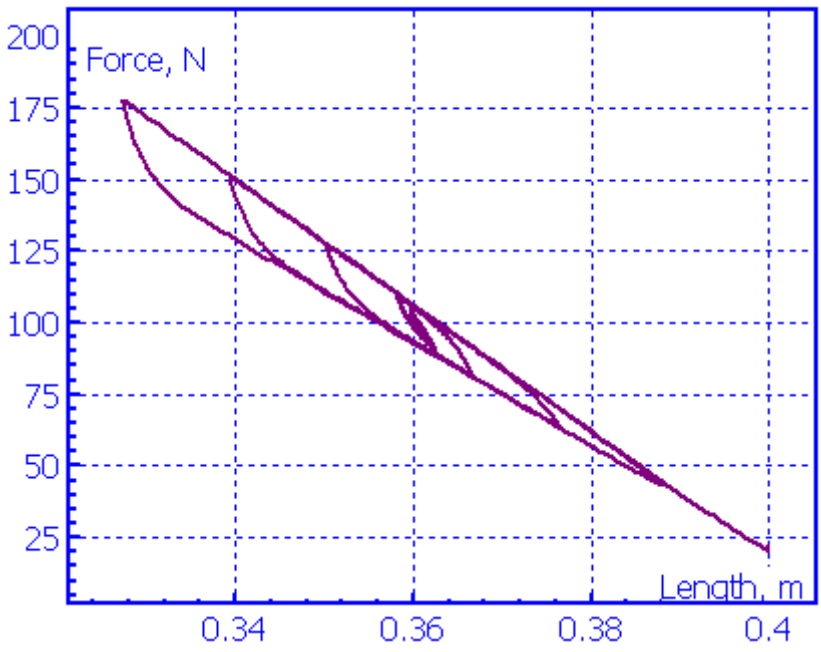
2.5.5.9. Fancher leaf spring

This type of force is used for simulation of leaf massless springs. The mathematical model of the force is the following:

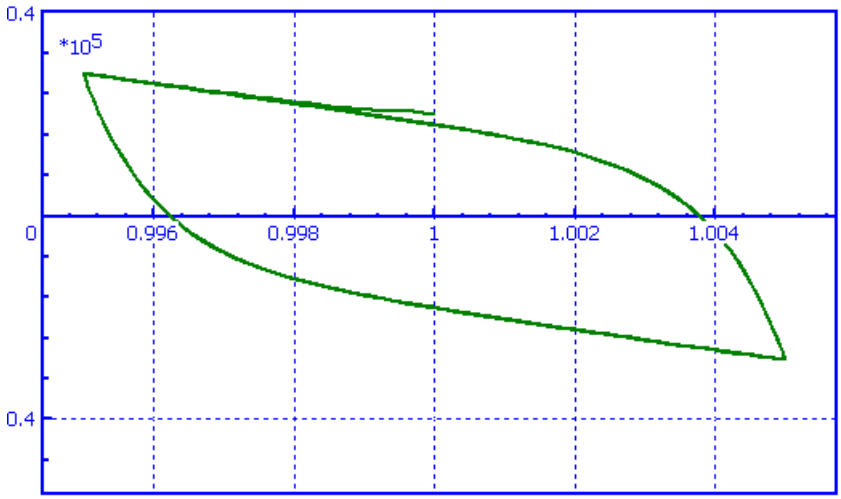
$$F_i = F_{env,i} + \left(F_{i-1} - F_{env,i-1}\right)e^{-|\Delta x_i - \Delta x_{i-1}|/\beta},$$
$$F_{env,i} = -c\Delta x_i - F_{fr}sign\{\Delta x_i - \Delta x_{i-1}\},$$
$$F_{fr} = fc\Delta x_i,$$
$$\Delta x = x - x_0.$$

Here i is the number of the integration step, f is the friction coefficient, c is the stiffness of the spring, F_{fr} is friction force, β is the exponential suspension parameter, x_0 is the height of the spring in the undeformed state. Parameters c and f may depend on the spring state: stretched ($dx < 0$) or compressed ($dx > 0$).

The plot in Figure 2.26a shows an example of free vertical vibration of a body connected with the base by the Fancher leaf spring element, β =0.002 m.



a)



b)

Figure 2.26. Typical behavior of the Fancher model

Note. The model is unstable for big β values. The recommended interval for this parameter is 0,001÷0,004. If $\beta \to 0$, the model is similar to the force of kind elastic – frictional 2, see Sect. 2.5.5.4. *"Elastic-frictional force 2"*, p. 2-40, with equal static and dynamic coefficients of friction.

Adding a new element of this type to a model see in Chapter 3, Sect. *Fancher leaf spring*.
Model: {UM Data}\SAMPLES\LIBRARY\Fancher, Figure 2.26b.

2.5.5.10. External function

The function should be written by the user with the help of programming in the UM environment.

Adding a new element of this type to a model see in Chapter 3, Sect. *Data types* | *External functions*.

2.5.5.11. List of characteristics

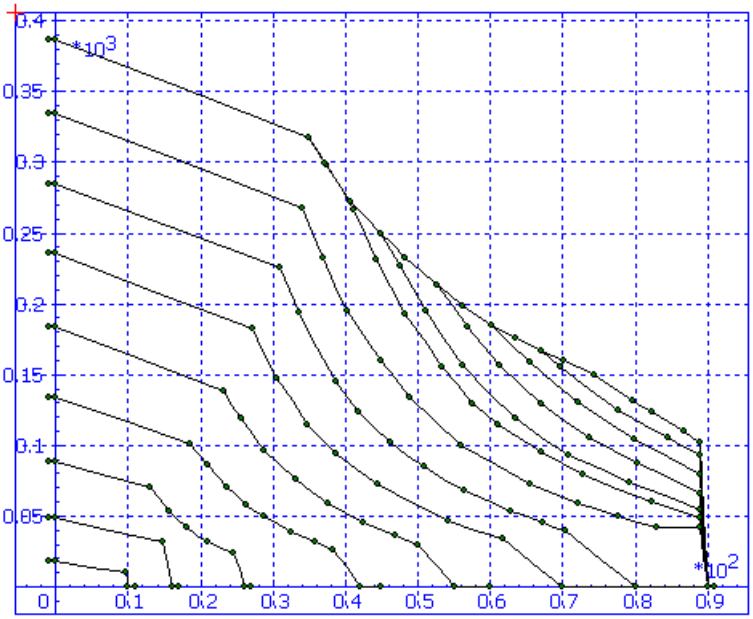


Figure 2.27. Example of a list of characteristics of a traction motor

A list of characteristics is a set of n curves defining dependences of the force on an argument. The argument can be *x*, *v* or *t*. Each of the curves is specified be a set of points, which number can be different for different curves. Curves are numbered 1...*n* in the input order.

During the simulation a force characteristic is selected by an identifier, which value lies in the interval from 0 to n. Zero value of the identifier corresponds to zero value of force, i.e. the force is excluded.

Two modes are available for identifier values: integer and real. In the first case, the force is exactly follows the curve with the index equal to the identifier value. In the second case, the force characteristic is a linear interpolation of the two nearest curves.

The user can specify multipliers *kx*, *ky* for the abscissa and ordinate values for the purpose of change of units, variation of abscissa direction and so on. It is supposed that abscissa in plot is the force argument multiplied by *kx*, i.e. *v*kx*. The force value corresponds to the ordinate multiplied by *ky*.

An example of the element usage is the modeling of electric motor torque versus rotor angular velocity for different throttle positions, Figure 2.27.

Force description in the input program can be found in Chapter 3, Sect. *List of characteristics*.

2.5.5.12. Hysteresis

2.5.5.12.1. Notions and definitions

Force of this type is used for modeling force elements, which mathematical model contains a hysteresis like cushioning device in a train.

The scheme of the force description curves is shown in Figure 2.28 (force versus element length or coordinate value). The force description includes definition of four curves:

- *preloading*, which is a nonlinear elastic curve including possible clearance; this section can be omitted, in this case only one point (as a rule this points is (0,0)) should be assigned to it;
- loading curve, containing two parts: *the start loading* and *loading*;
- unloading curve, containing two parts: *the start unloading* and *unloading*;
- *stop* is a nonlinear elastic curve.

The start loading/unloading curves are used in particularly for computation of the curve of *intermediate loading/unloading* shown in the figure by the dashed curve. The transition of the force law to the intermediate curve occurs in the following two cases. First, when the force is on the loading curve and the coordinate starts decreasing. Second, when the force is on the unloading curve and the coordinate starts increasing. After transition to the intermediate curve, the force follows it up and/or down until the loading or unloading curves are reached.

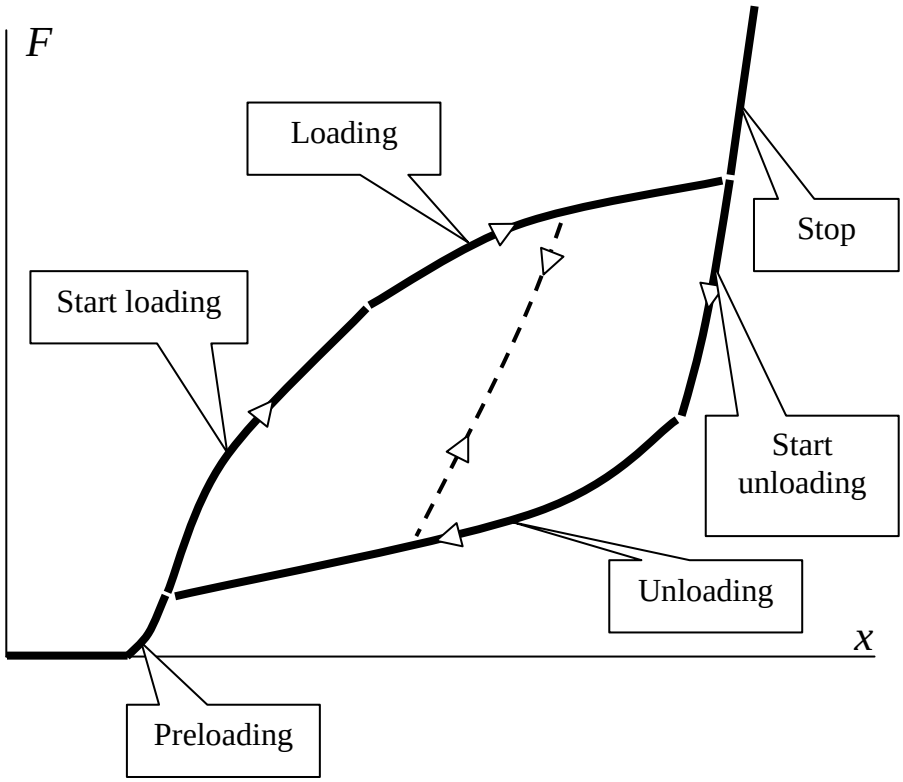


Figure 2.28. Scheme of hysteresis curve

2.5.5.12.2. Details of curve descriptions

1. The user should enter a set of points, which lie on the curves.
2. Each of the curves is described by numbers of points; abscissas of the points must be *in ascending (strictly increasing) order*. The curves must be single-valued functions.
3. Points connecting different curves must be the same. For instance, the first points on unloading and start loading curves or the first point on the loading curve and the last point on the start loading must coincide.

4. A curve can be a polyline (set of straight sections) or an interpolation polynomial, which order should be set by the user. Note that the number of points setting a line must be greater than the order of polynomial.

Formulas for evaluation of intermediate loading-unloading curve

Suppose that the transition to the intermediate curve occurs from the loading curve at point V (Figure 2.29). The unitless parameter $\alpha \in [0,1]$ is computed as

$$\alpha = \frac{x_V - x_C}{x_D - x_C}$$

Point U on the unloading curve is defined by the same value of the parameter α. Abscissa of point U is

$$x_U = x_B\alpha + x_A(1-\alpha)$$

The intermediate curve passes through points U, V. Let us compute its arbitrary point u, which abscissa corresponds to the unitless parameter $s \in [0,1]$

$$s = \frac{x_u - x_U}{x_V - x_U}$$

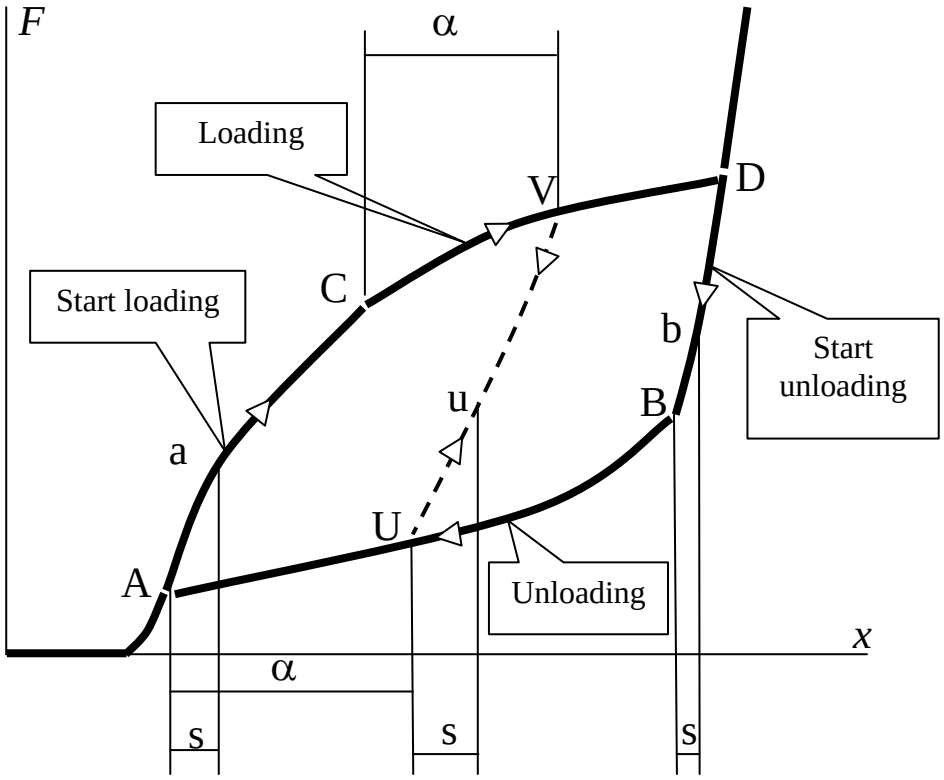


Figure 2.29. Definition of intermediate loading-unloading curve

Points a, b on the start loading and start unloading curves for the given values of s have the following abscissa values:

$$x_a = x_C s + x_A(1-s), x_b = x_D s + x_B(1-s).$$

Then abscissa and ordinate of point u are computed according to the formulas

$$x_u = \big(x_U - x_B\alpha - x_A(1-\alpha)\big)(1-s) + \big(x_V - x_D\alpha - x_C(1-\alpha)\big)s + x_b\alpha + x_a(1-\alpha),$$
$$F_u = \big(F_U - F_B\alpha - F_A(1-\alpha)\big)(1-s) + \big(F_V - F_D\alpha - F_C(1-\alpha)\big)s + F_b\alpha + F_a(1-\alpha).$$

The intermediate curve defined by these formulas has the following properties:

- by $\alpha = 0$ it coincides with the start loading curve;
- by $\alpha = 1$ it coincides with the start unloading curve;
- for other values of α it passes through points U, V.

Thus, the intermediate loading/unloading curve is a combination of start loading/unloading curves.

An example of a hysteresis force is shown in Figure 2.30. The preloading curve is a polyline passing through three points; it specifies the 10 mm clearance. The loading, unloading and start unloading curves are set by 3 points each; the second order interpolation polynomials are used. The start loading curve is specified by two closed points; it is invisible on the plot. Three intermediate curves and the stop curve (straight section) are presented in the figure as well.

Elastic-frictional force with load dependent friction is shown in Figure 2.31. All curves are polylines.

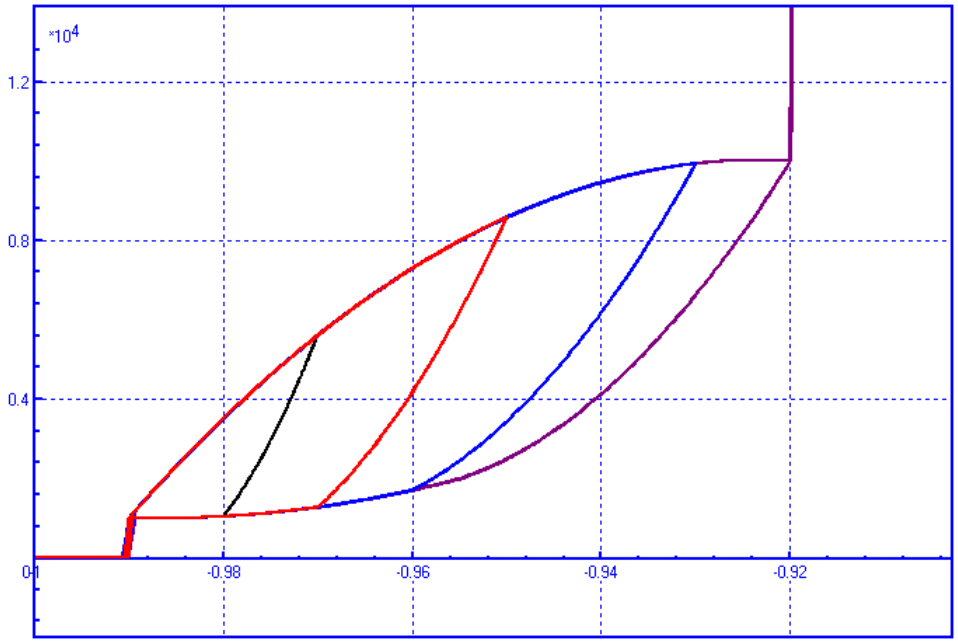


Figure 2.30. Example of hysteresis model with a soft characteristic

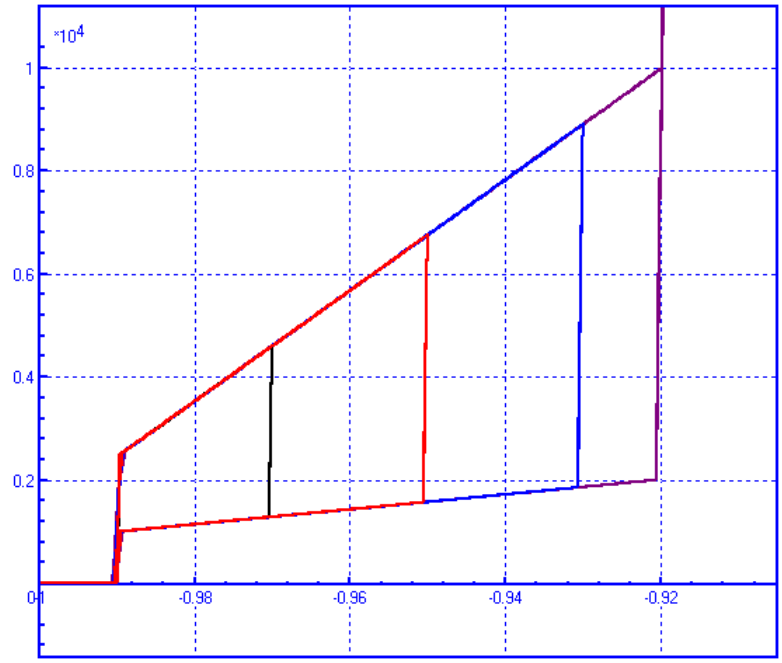


Figure 2.31. Example of friction hysteresis model

2.5.5.12.3. Modes: stretching, compression, symmetric

The force element supports the following modes:

- Symmetric

In this case, the force curve is symmetric for stretching and compression of the element like in Figure 2.32.

- Stretching

The hysteresis curves are set for the stretching of the element. Extrapolation of the *preloading* curve for the compression is used.

- Compression

The hysteresis curves are set for the compression of the element. Extrapolation of the *preloading* curve for the stretching is used.

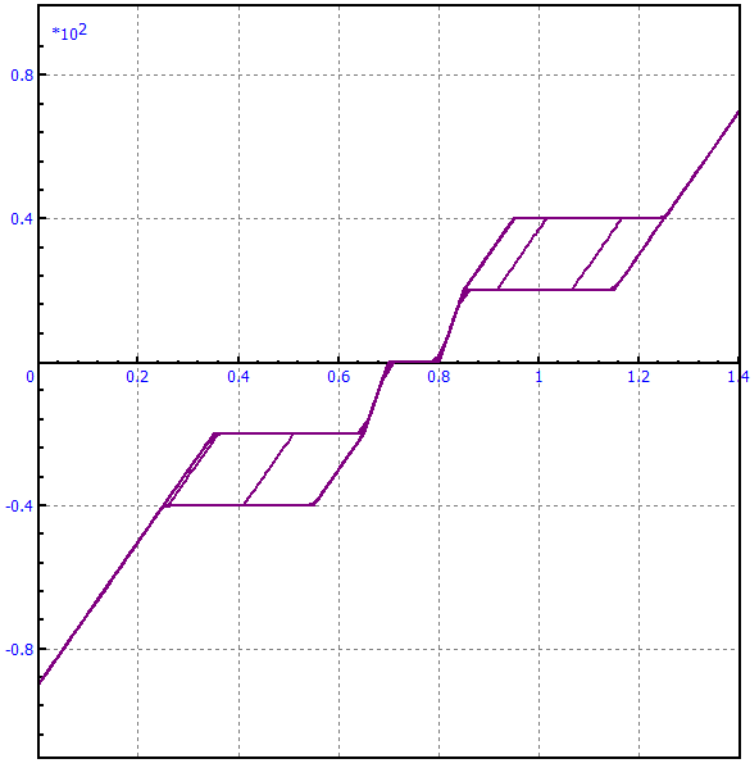


Figure 2.32. Example of symmetric hysteresis

2.5.5.12.4. Extrapolation of the hysteresis curve

In the compression and stretching modes, an interpolation of the preloading curve is used for the coordinate shift in the negative region, i.e. for the stretching in the compression mode, and for the compression in the stretching mode. To set zero force value for this opposite shift, a section with zero force should be defined in the preload curve, Figure 2.33. Input of the element parameters see in Chapter 3, Sect. Hysteresis.

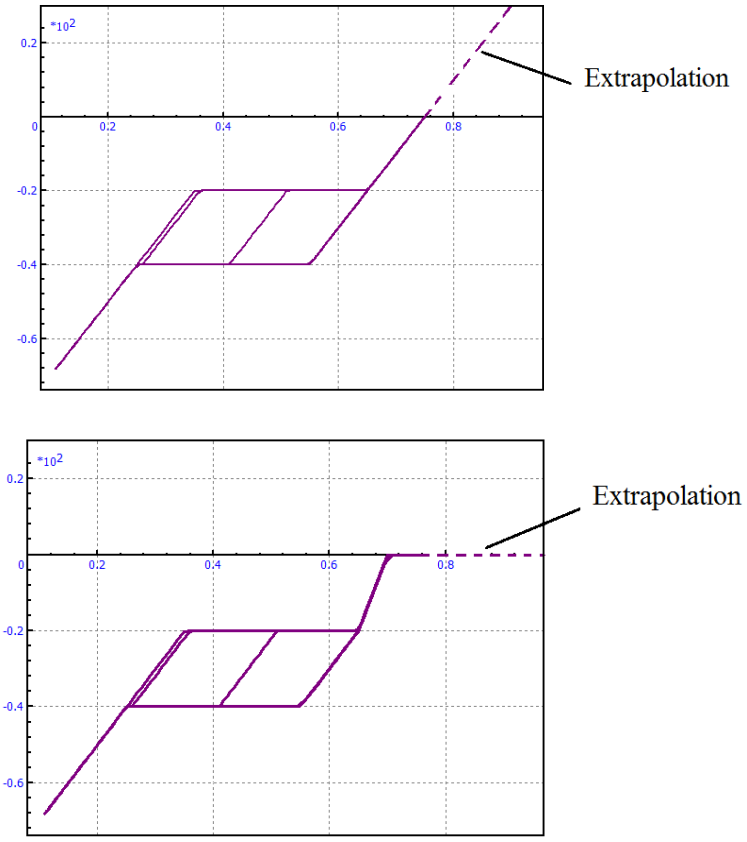


Figure 2.33. Examples of extrapolation in compression mode

Note	It is recommended to add a linear dissipative element in parallel with the hysteretic one because the intermediate curve does not dissipate energy.

2.5.5.13. Impact (bump stop)

Forces of this kind works as bump stops and use the following mathematical model:

$$\Delta = x - l$$
$$F = -c\Delta^b - step(\Delta, 0, \Delta_d, 0, d)v,$$

where

x is the current length of the force elements for bipolar forces or the current value of joint coordinate for joint forces;

l is the length of the element at zero clearance when force element starts to work;

c is the stiffness coefficient in a contact;

b is the force curve exponent, is not used in the current version of UM software, assumed to be 1;

Δ_d is the contact deflection where damping coefficient reaches its maximal value d;

d is a damping coefficient in a contact,

step is a special function, see below for details.

Due to using *step* function effective damping coefficient changes smoothly from zero at zero deflection to maximal damping coefficient *d* at Δ_d deflection.

Description of step function

Step function smoothes switching between two levels of a function Y (Y_1, Y_2) within some (X_1, X_2) interval.

$$step(x, x_1, x_2, y_1, y_2) = \begin{cases} y_1, x \le x_1 \\ y_1 + (y_2 - y_1)\left(\frac{x - x_1}{x_2 - x_1}\right)^2 \left(3 - \frac{2(x - x_1)}{x_2 - x_1}\right), x_1 < x < x_2 \\ y_2, x \ge x_2 \end{cases}$$

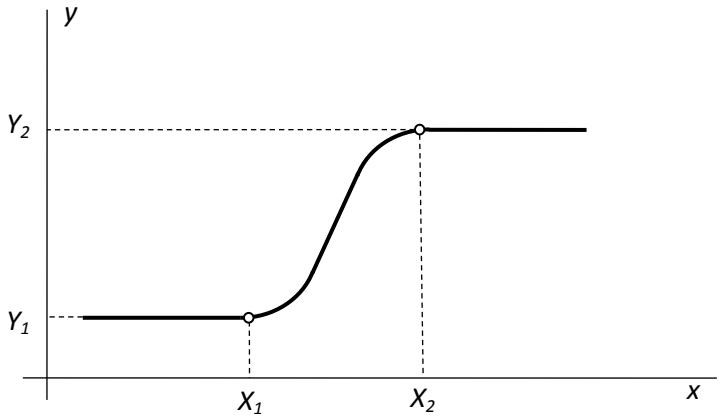


Figure 2.34. STEP function

2.5.5.14. Ratchet

A simplified model of a ratchet gear is implemented. The force/torque may only shift/turn in one direction. As a rule, the element is used in a joint torque description, which locks rotation in the direction specified by the user, and the torque is zero by the rotation in the opposite direction. The torque model is quite similar to that described in the “Impact” force type Sect. 2.5.5.13. *"Impact (bump stop)"*, p. 2-54.

In particular, the torque model of this type is used in the model of the driven sprocket of the bike chain gear.

Models:

{UM Data}\SAMPLES\LIBRARY\Ratchet;

{UM Data}\SAMPLES\LIBRARY\ChainGear.

Data input: Chapter 3, Sect. *Ratchet*.

2.5.5.15. Library (DLL)

Forces of this type are calculated in an external Dynamic-Linked Library (DLL). It helps the user to describe any mathematical model of force to include in a considered mechanical system. To develop your own external library the user can use any software environment and any program language that supports creating Dynamic-Linked Libraries (DLL). Ready-to-use DLLs for scalar forces of the Library (DLL) type should be placed in the **{UM Data}\[x32|x64]\lib\BFrc** folder.

Please turn to the Chapter 5 of UM User's Manual, Sect. *Creating and using external libraries*, for detailed guide into developing and using external libraries in UM models.

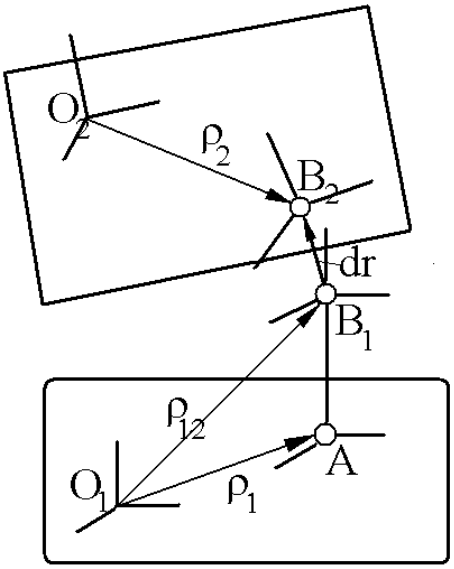
2.5.5.16. List of forces

List of forces is a set of force elements of above types in parallel. The total force is the sum of forces produces by separate elements in the list/ The element allows creating useful combinations of forces such as nonlinear damper in parallel with nonlinear spring, rubber-metal elements etc.

Adding a new element of this type to a model see in Chapter 3, Sect. *List of forces*.

2.5.6. Generalized linear force element

When modeling various technical systems, especially those used in transport, force elements linearly depending on the relative displacements of bodies and the velocities of their relative motion are so often to be obtained whereas the influence of their dynamics is insignificant. A spring and a damper belong to this type of elements. To model such forces automatically UM uses the generalized linear and damping force elements. An element of this type connects two different bodies (one of which may be the basic one). One body is considered the first, the other – the second.



Let us consider the model of a linear elastic force element in more details (see the figure). Two bodies are shown. The A and B2 points of the bodies are connected by the linear elastic force element. The body-fixed systems of coordinates SC1 and SC2 have O1 and O2 as origins. The element attachment points are set by two vectors ρ_1^1, ρ_2^2, each of them should be set in the SC of the corresponding body.

And additional body1-fixed system of coordinates (SC *of the element*) has point A as the origin. Axes of SCA are arbitrary oriented relative to SC1, in a simples case they are *parallel* to the axes of SC1.

An additional point B_1 is assigned to the first body. The ρ_{12}^1 vector specifies its position in SC1. Point B_1 is the origin of a system of coordinate (SCB1), which axes are *parallel* to SCA. Another additional SCB2 is fixed *in* body 2, its axes are arbitrary oriented relatively to SC2. Point B_1 corresponds to the position of the second end of the force element (i.e. it coincides with B_2) when *dr* vanishes (dr=0). The relative rotation $d\pi$ is equal to zero when the axes of SCB1 and SCB2 are parallel.

Thus, the force is equal to the stationary value F0, and the moment vanishes when SCB1 and SCB2 coincide. Displacement of B2 relatively to B1 sets the dr vector, whereas the rotation of SCB2 relatively to SCB1 (SCA) sets the $d\pi$ vector. *Both* dr *and* $d\pi$ *are assumed to be small* and define the force and the moment values.

Now consider the mathematical model of the force element. The force *F* and the moment applied to the *second* body at B_2 and resolved in SCB1 (SCA) are:

$$F = F_0 - C_{rr}dr - C_{ra}d\pi - D_{rr}v - D_{ra}\omega,$$
$$M = -C_{ar}dr - C_{aa}d\pi - D_{ar}^T v - D_{aa}\omega$$

for an viscoelastic, and

$$F = -D_{rr}v - D_{ra}\omega,$$
$$M = -D_{ar}^T v - D_{aa}\omega$$

for a damping element. Here $v = dr, \omega$ is the angular velocity of the second body relative to the first one.

The formulas can be rewritten as:

$$G = G_0 - CdR - DV,$$
$$G = -DV,$$

with *C* and *D* as the 6 × 6 stiffness and damping matrices, and

$$G = \begin{pmatrix} F \\ M \end{pmatrix}, G_0 = \begin{pmatrix} F_0 \\ 0 \end{pmatrix}, dR = \begin{pmatrix} dr \\ d\pi \end{pmatrix}, V = \begin{pmatrix} v \\ \omega \end{pmatrix}.$$

Here is the list of parameters describing the element:

- $\rho_1, \rho_{12}, \rho_2$ are the coordinates of points A, B_1, B_2; points A, B1 should be defined in SC1, whereas point B_2 should be defined in SC2;
- orientations of SCA (SCB1) relative to SC1, and SCB2 relative to SC2; in the **UM Input** program the oriented connection point are used to define both the point and the orientation of system of coordinates;
- stiffness *C* and/or damping *D* matrices as well as the stationary force F_0 (optionally); these data are defined in SCA (SCB1).

Force element can be also used for modeling a *bilinear* spring, i.e. a set of two springs where the internal spring is lower than the external one. The internal springs works when the longitudinal deflection of the external spring is greater that the difference between the spring heights. A bilinear element is described by two stiffness matrices and by the height difference.

Generalized linear force element can be only used if relative displacements of interacting bodies are small.

See Sect. 2.5.9.6. *"Spring"*, p. 2-85.

Examples of description and/or usage:

Chapter 7. Sect. Models of Springs;

{UM Data}\SAMPLES\Rail_Vehicles\Manchester benchmarks\Vehicle1;

{UM Data}\SAMPLES\Rail_Vehicles\wedgetest.

2.5.7. Contact forces

2.5.7.1. General information about contact interactions

2.5.7.1.1. Compliant contact

Contact interaction in UM is implemented as a *compliant contact*, where the contact forces depend on a penetration of a contact element of one body (points, circle, sphere and so on) into a surface connected with another body (plane, Z-surface, sphere, etc). Types of contact force elements depend on the contact element geometry and discussed in details below.

Most of the contacts are related to the *unilateral* interaction, when contact forces appear for a positive penetration only. Two forces are computed for each of the contacts: a normal force N and a friction force. The normal force is an elastic-dissipative function of the penetration depth and its time derivative. The friction force is located in the contact tangential plane.

Rational choice of contact stiffness and damping parameters is discussed in Sect. 2.6. *"Methodology of choice of contact parameters"*, p. 2-100.

2.5.7.1.2. Models of friction force

The friction force model must realize both sliding and sticking modes. In the sliding mode, the friction force is proportional to the normal force $F = f \cdot N$ and directed opposite to the sliding velocity. The constant of proportionality *f* is the friction coefficient, which value can depend on the sliding velocity, Sect. 2.5.7.1.3. *"Dependence of coefficient of friction on sliding velocity"* p. 2-60.

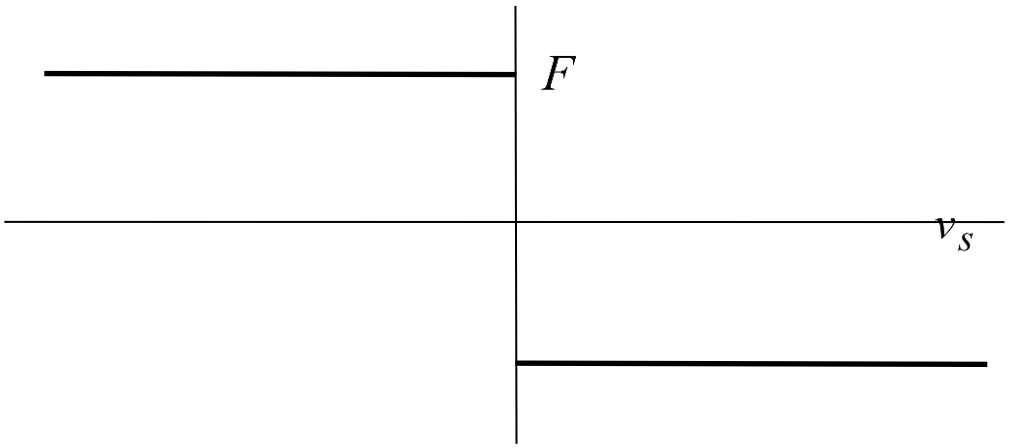


Figure 2.35. Classical Coulomb friction

It is known that the classical Coulomb friction model for dependence of the friction force on the sliding velocity v_s (Figure 2.35) cannot be correctly used in the sticking mode numeric. It leads to oscillations in friction force and acceleration and because of discontinuity at $v_s = 0$. Two different models are implemented in UM for the sticking: a model of regularized friction law and a flexible sticking.

In the case of the **regularized friction**, the F vs. v_s characteristic is smoothed for small sliding velocities. This method is used by simulation of rolling processes, Sect. 2.5.7.5. *"Other types of contact forces"* p. 2-68.

In the case of the flexible sticking, the contact point is flexibly connected to the contact surface. This connection is broken if the friction force exceeds its maximal value. This model is applied to contacts points-surface and point-curve, Sect. 2.5.7.2. *"Points-Plane and Points-Z-surface types"* p. 2-61 and Sect. 2.5.7.3. *"Point-Curve contact"* p. 2-65. See Sect. 2.5.7.2. *"Points-Plane and Points-Z-surface types"*, p. 2-61 for more details.

2.5.7.1.3. Dependence of coefficient of friction on sliding velocity

A combination of an exponential model describing the Stribeck effect for lubricated surfaces and a viscous friction is considered as a dependence of the coefficient of friction f on the value of the sliding velocity [2] [3] [4]

$$f(v_s) = f_\infty + (f_0 - f_\infty)e^{-(v_s/v_{str})^\delta} + vv_s,$$

where

f_0, f_∞ are the values of coefficient of friction for zero and infinite velocity;

v_{str} is the Stribeck velocity specifying the velocity range in which the Stribeck effect is effective;

$\delta \in [0.5, 1]$ is the empirical exponent depending of materials;

v is the viscous friction constant.

Stribeck effect [2] consists in the experimental fact that the drop from static friction to Coulomb sliding friction is not discontinuous for lubricated surfaces but that it is a continuous function of the velocity like in Figure 2.36e.

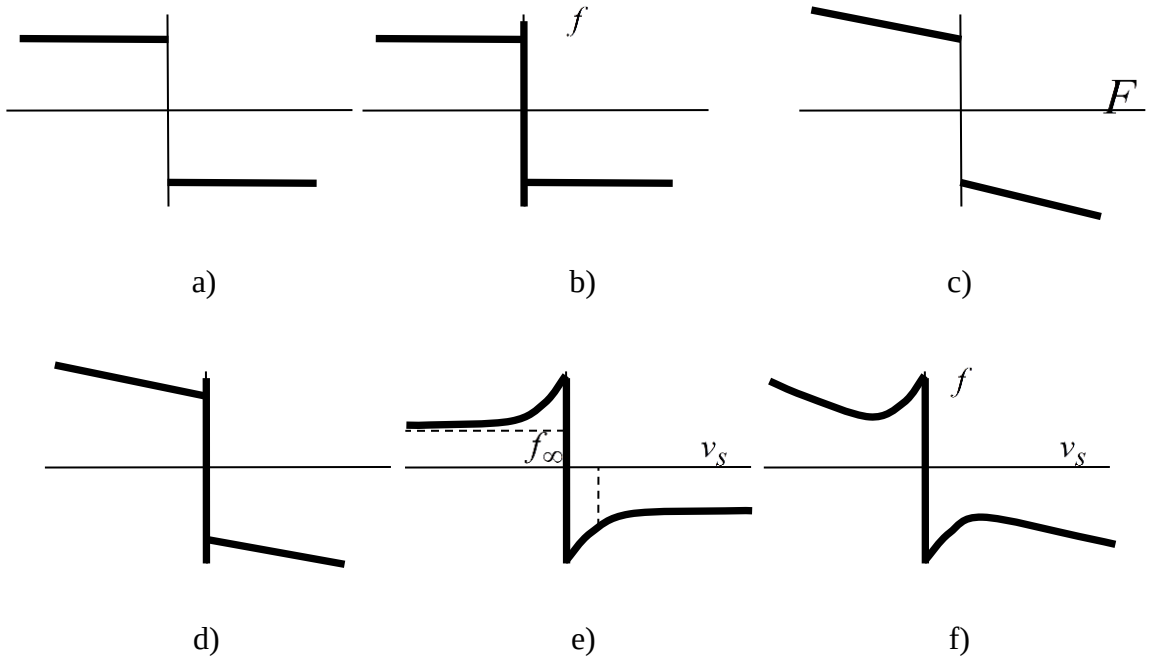


Figure 2.36. Different models of coefficient of friction verses sliding velocity characteristics

The above model allows the user to describe the most of the frequently used friction models, Figure 2.36:

a) $f_\infty = f_0, v_{str} = 0, v = 0$ is the classical dry friction model with equal static and dynamic coefficients;
b) $f_\infty < f_0, v_{str} = 0, v = 0$ is the classical dry friction model with static coefficient greater than the dynamic one;
c) $f_\infty = f_0, v_{str} = 0, v > 0$: in addition to Figure 2.36a the viscous friction is added;
d) $f_\infty > f_0, v_{str} = 0, v > 0$: in addition to Figure 2.36b the viscous friction is added;
e) $f_\infty > f_0, v_{str} > 0, v = 0$ is the model describing the Stribeck effect;
f) $f_\infty > f_0, v_{str} > 0, v > 0$ in addition to Figure 2.36e the viscous friction is added.

2.5.7.2. Points-Plane and Points-Z-surface types

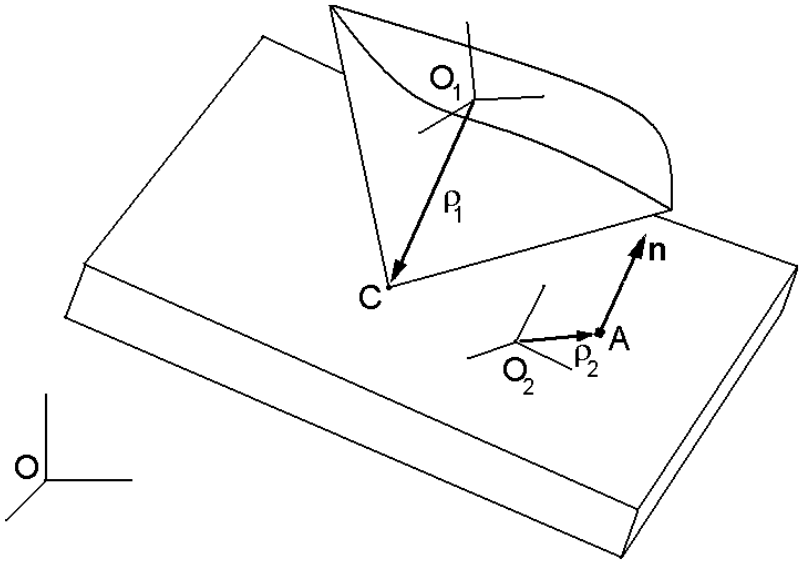


Figure 2.37. Point-Plane contact model

UM uses the Points-Plane contact force element for modeling the simplest contact interactions between two bodies (Figure 2.37). A set of body-fixed points is assigned to the first body (body 1, a single point C is shown in Figure 2.37). A body-fixed plane is assigned to the second body (body 2). The plane is specified by a body fixed point (A) and by an external normal **n**. The contact points, point A and normal n should be given in SC of the corresponding bodies. Number of contact points is unlimited.

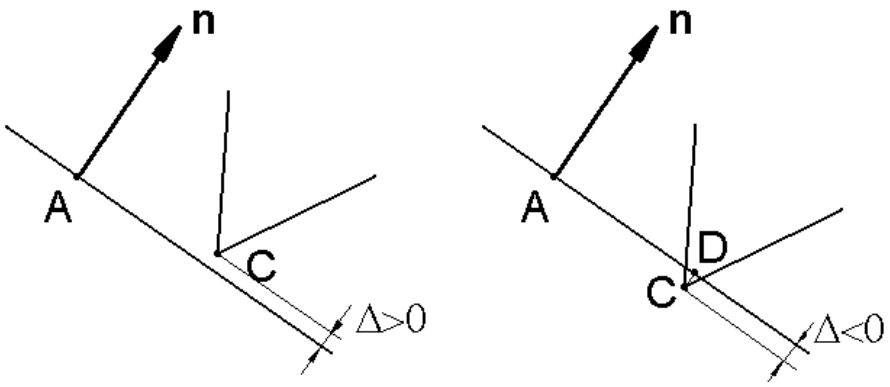


Figure 2.38.

If the distance Δ between the contact point and the plane is positive and the contact is a unilateral one, the contact force is equal to zero (no contact, Figure 2.38 left). If Δ<0 (Figure 2.38 right) the force appears (unilateral contact). The contact force has two components: the normal force N directed along the normal n, and the friction force $\mathbf{F}_f$ situated in the contact plane. In case of a bilateral contact, the force appears by deviation of a point in both direction with a possible gap.

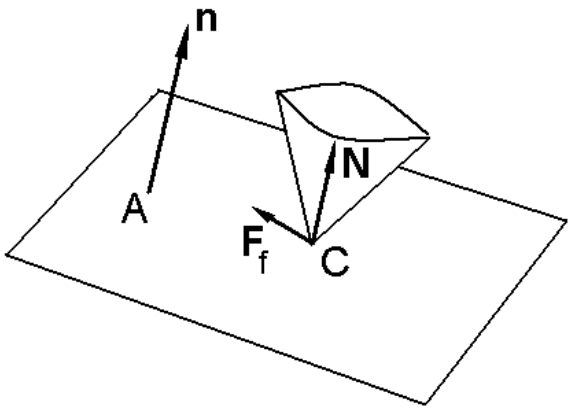


Figure 2.39.

A linear viscoelastic model is used for the normal force

$$N = -c\Delta - d\dot{\Delta},$$

If this value is negative and the contact is a unilateral one, the force is equal to zero (no adhesion). Parameters *c*, *d* are constant stiffness and damping coefficients.

There exist two modes for the Coulomb friction low: sliding and sticking. The force model in the sliding mode is

$$\mathbf{F}_f = -fN\boldsymbol{v}_s/\|v_s\|,$$

where $\boldsymbol{v}_s$ is the sliding velocity (projection on the contact plane of velocity of point C relative to body 2), f is the dynamic friction coefficient. The sliding-sticking transition occurs when the

sliding velocity changes its direction on an opposite one. The program realization compares the scalar product of the velocity on the current integration step and the velocity on the previous step. If the product is negative, the sticking occurs, and the friction force on the previous step $\boldsymbol{F}_g$ and the $\boldsymbol{r}_{g0}$=AD vector are stored, where D is the projection of point C on the contact plane (Figure 2.38).

The friction force model in the sticking mode looks like this:

$$F_f = F_g - c(r_g - r_{g0}) - dv_s,$$

where $\boldsymbol{r}_g$ is the current value of the vector AD. Thus, the point D is connected with the contact plane by a linear viscoelastic element at the sticking mode. The force F_g in the formula ensures the continuity of the friction force at the sliding-sticking transition.

The sticking mode is over when the friction force exceeds its maximal value

$$\|\boldsymbol{F}_f\| > f_0 N,$$

where f_0 is the static friction coefficient, $f_0 > f$.

The following parameters specify the Points-Plate force element:

- Coordinates of contact points in SC of the first body;
- Coordinates of a point and an external normal to the contact plane in SC of the second body;
- Parameters characterizing contact forces: dynamic and static friction coefficients, contact stiffness and damping coefficients c, d.

An additional mode of the contact description is implemented in UM, a so-called mode of close contact. In this mode, the normal **n** and/or the point **A** are determined automatically. The mode is enabled by the following conditions:

- Number of contact points is greater than 2 (for the normal detection only);
- Contact points lie in a plane (for the normal detection only);
- At the moment the simulation starts, the contact point plane (belongs to the first body) and the contact plane (the second body) are close, i.e. a small clearance δ between the planes is allowed. A small deviation of the normal to the planes is allowed as well.

The close contact description requires specifying the clearance and the normal deviation $\Delta\mathbf{n}$. The following actions are made by UM to define the contact plane (the second body):

- The external normal $\mathbf{n}_2$ to the plane is computed as

$$\mathbf{n}_2 = -\mathbf{n}_1 + \Delta\mathbf{n},$$

where $\mathbf{n}_1$ is the normal to the points plane (the first body), $\Delta\mathbf{n}$ is the normal deviation. The $\mathbf{n}_1$ vector is determined by three first contact points, entered by the user, according to right-hand screw rule, the points cannot lie on a line. The normal $\mathbf{n}_1$ should be external for the first body contact (the opposite vector $-\mathbf{n}_1$ is assumed to be external to the contact plane of the second body).

Point A on the contact plane (the second body) is computed according to the formula

$$\mathrm{r}_A = \rho_1 + \mathbf{n}_1\delta,$$

where δ is the clearance, ρ_1 is the vector to the first contact point specified by the user.

Before the simulation of the object starts, the contact plane position is calculated anew according to initial positions of the bodies.

The contact can be a *bilateral* one (the contact forces appears independent on the direction or penetration of the point). In this case the clearance can be introduced even if the option of a close contact is not applied.

The *limited plane* contact is available. In this mode, the contact is possible in a closed region on the plane. The forces are zero outside the region. The region boundary curve can be rectangle, circle or any curve described by a set of points with spline smoothing.

Note. **Points–Z-surface** contact element is mathematically similar to the above one. In the case of this element, the contact plane is replaced by a surface, which is described by the functional relation $z = f(x, y)$. No close contact mode is available for this contact.

Some features of *linear analysis* of models in presence of points-plate contacts are described in Chapter 4, Sect. *Equilibrium in presence of contact forces*.

2.5.7.3. Point-Curve contact

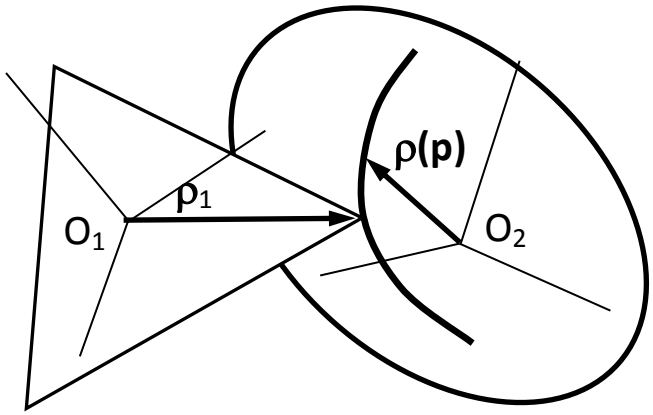


Figure 2.40. Point-curve contact

This type of contact interaction realizes sliding with a small deviation of a point belonging to the first body on a curve, which is fixed relative to the second body, Figure 2.40.

Position of the contact point relative to SC1 is set by a constant vector ρ_1.

In general the curve in SC2 is specified by a dependence of coordinates of points on the curve on a scalar parameter *p* (the upper index in the equation corresponds to the number of SC)

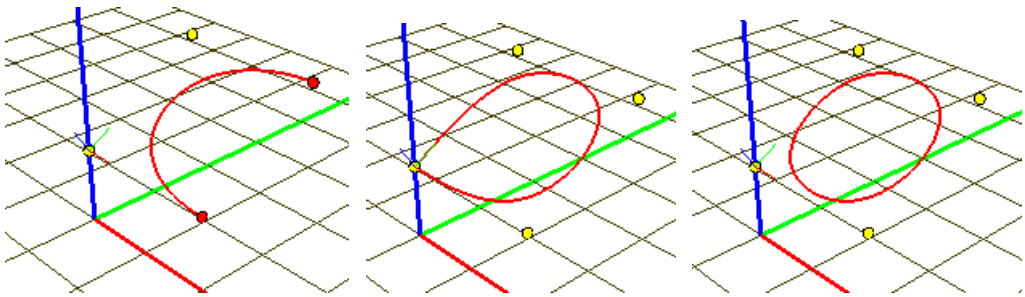
$$\rho^2 = \rho^2(p), p \in [p_{min}, p_{max}],$$

where ρ^2 is the radius vector of a point on the curve in SC2. The equivalent scalar form of the curve equation is

$$x_2 = x_2(p), y_2 = y_2(p), z_2 = z_2(p),$$
$$\rho^2 = (x_2, y_2, z_2)^T$$

The following classification of curves is used:

- open curve, which end points differ, $\rho^2(\rho_{min}) \neq \rho^2(\rho_{max})$;
- closed curve, end points are equal, $\rho^2(\rho_{min}) = \rho^2(\rho_{max})$;
- periodic curve is a closed curve, which have smooth derivative at the end points (tangents coincide) $\rho'^2(\rho_{min}) = \rho'^2(\rho_{max})$; the stroke here corresponds to derivative with respect to the parameter p.



a b c

Figure 2.41. Examples of different curves

Figure 2.41 shows an open (a), closed (b) and periodic (c) curves.

Each of the two end points of an open curve can be either locking or unlocking. The locking end point keeps the contact point on the curve while by passing through the unlocking point the contact disappears.

Singular points are locking end points of an open curve as well as points on it where he derivative discontinues (jump in the tangent or edge points, see Figure 2.41b).

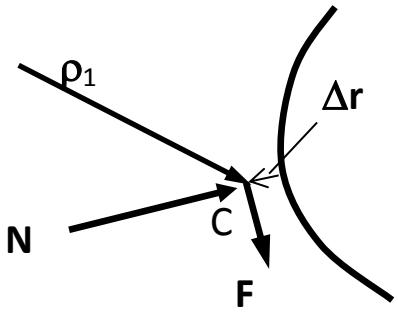


Figure 2.42. To model of contact forces in non-singular points

In case of a contact in non-singular points, two contact forces appear: a normal force **N** perpendicular to the curve and a tangent friction force **F**. The normal force vector applied to the contact point C, Figure 2.42, is computed by the formula

$$N = -c\Delta r - \nu\Delta\dot{r}$$

where Δr is the vector of the minimal deviation of the point from the curve; c and ν are the contact stiffness and damping constants.

The friction force supports both sliding and modes similar to the point-plane contact model.

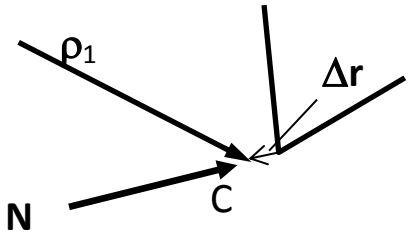


Figure 2.43. To model of contact forces in singular point

In case of contact in a singular point, Figure 2.43, the friction force vanishes, and the normal force depends on the vector of deviation Δr of the contact point C from the singular point according to the above equation.

Remark. If the contact stiffness is big enough, the deviation of the contact point from the curve is small. Values of contact stiffness and damping constants should be chose

according to recommendations in Sect. 2.6. *"Methodology of choice of contact parameters"*, p. 2-100.

2.5.7.4. Contact of circle with cylinder with curved axis

The element models a compliant contact of a circle with a cylinder, which axis is set by a smooth curve. It is assumed that the axis curve intersects the circle plane under the angle, which differs from the right angle less than 30°. The normal force depends on the penetration depth and rate. Friction force model is described in Sect. 2.5.7.5. *"Other types of contact forces"*, p. 2-68.

The force element can be applied in simulation of a pipe dynamics in cylindrical holes.

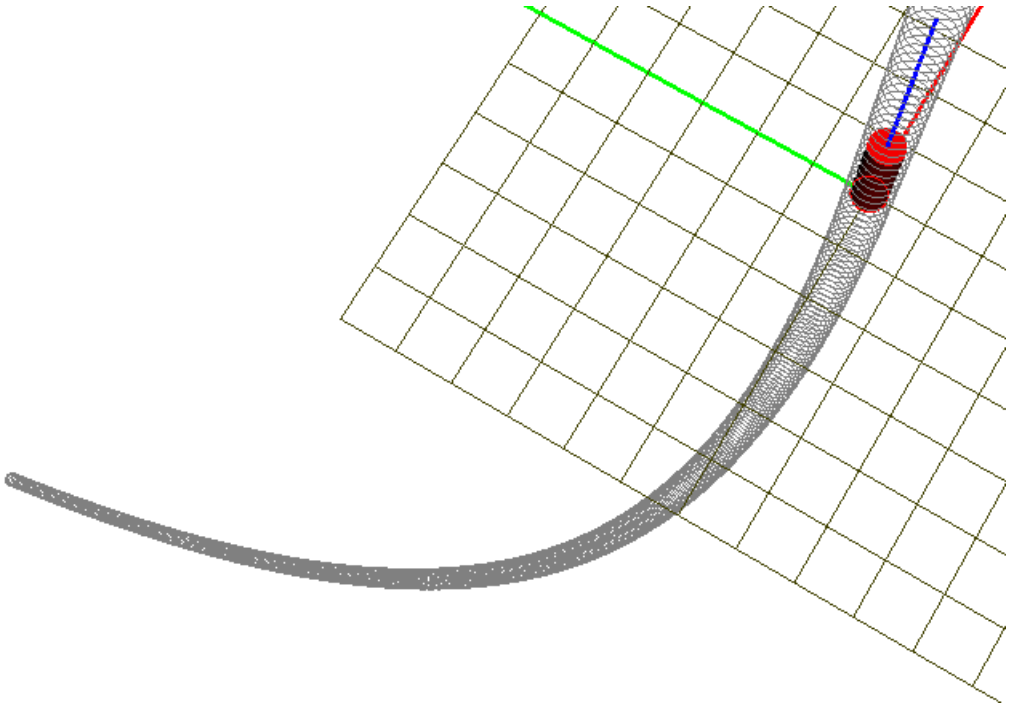


Figure 2.44. Modeling of cylinder motion in a cylindrical hole with curved axis

Model (Figure 2.44): {UM Data}\SAMPLES\LIBRARY\CylCircle.

2.5.7.5. Other types of contact forces

UM allows other types of contact:

- Sphere-Plane;
- Circle-Plane;
- Sphere-Sphere;
- Circle-Z surface
- Sphere-Z surface.

All of these contacts describe rolling one body on a surface of another one.

Consider mathematical models of the contacts. First of all, the minimal distance between the surfaces is determined. If the distance is positive, contact forces are zero, if not, the normal force N is computed (Sect. 2.5.7.1. *"General information about contact interactions"*, p. 2-59). The friction force model differs from that in Sect. 2.5.7.1. *"General information about contact interactions"*, p. 2-59:

$$\mathbf{F}_f = \begin{cases} -\dfrac{fN\mathbf{v}_s}{\|\mathbf{v}_s\|}, & \|\mathbf{v}_s\| > v_s^*, \\ -\dfrac{fN\mathbf{v}_s}{v_s^*}, & \|\mathbf{v}_s\| \le v_s^*, \end{cases}$$

where f is the dynamic friction coefficient, N is the normal force, $\mathbf{v}_s$ is the sliding velocity, v_s^* is the empirical (small enough) value of sliding velocity. If the sliding velocity as not small, the classical model of friction is used, else the viscous damping is considered. The v_s^* should be set by the user.

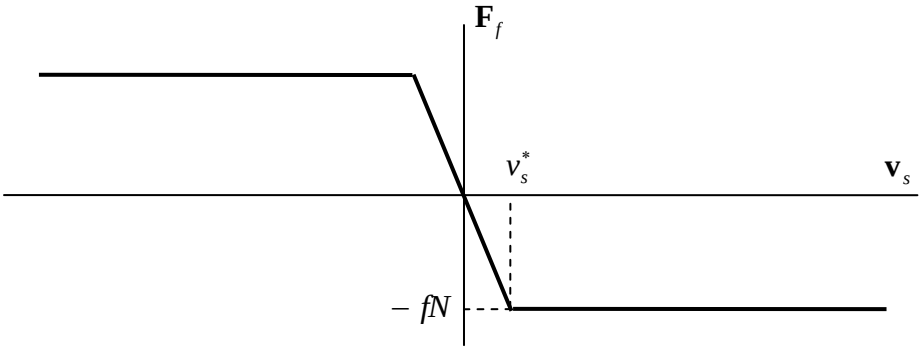


Figure 2.45. Regularized friction model

Rolling and spin friction

Rolling and spin friction are available for circle and sphere contact models. Let us consider a body angular velocity $\boldsymbol{\omega}$ as a sum

$$\boldsymbol{\omega} = \boldsymbol{\omega}_{\text{spin}} + \boldsymbol{\omega}_{\text{roll}}.$$

Here the first summand $\boldsymbol{\omega}_{\text{spin}}$ is perpendicular to the contact surface and corresponds to a spin, the second summand $\boldsymbol{\omega}_{\text{roll}}$ lies in the tangent plane and corresponds to the rolling. The spin

M_{spin} and rolling M_{roll} friction torques are computed according to formulas, which are similar to the sliding friction

$$M_{\text{spin}} = \begin{cases} -k_{\text{spin}} N \, \boldsymbol{\omega}_{\text{spin}} / \left| \boldsymbol{\omega}_{\text{spin}} \right|, \left| \boldsymbol{\omega}_{\text{spin}} \right| > \omega^* \\ -k_{\text{spin}} N \, \boldsymbol{\omega}_{\text{spin}} / \omega^*, \left| \boldsymbol{\omega}_{\text{spin}} \right| \le \omega^* \end{cases}$$

$$M_{\text{roll}} = \begin{cases} -k_{\text{roll}} N \, \boldsymbol{\omega}_{\text{roll}} / \left| \boldsymbol{\omega}_{\text{roll}} \right|, \left| \boldsymbol{\omega}_{\text{roll}} \right| > \omega^* \\ -k_{\text{roll}} N \, \boldsymbol{\omega}_{\text{roll}} / \omega^*, \left| \boldsymbol{\omega}_{\text{roll}} \right| \le \omega^* \end{cases}$$

The spin k_{spin} and rolling k_{roll} coefficients of friction are measured in meters, ω^* is the empirical (small enough) value of angular velocity similar to velocity v_s^* in Figure 2.45.

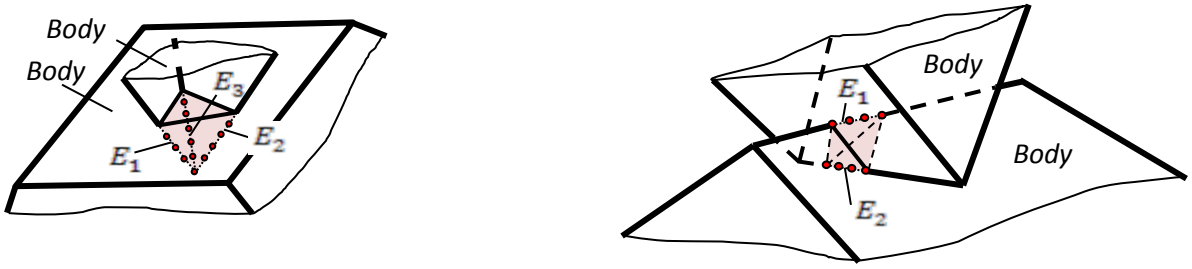
2.5.8. 3D contact

The common approach for simulation of contact between rigid bodies based on its contact manifolds defined through graphical objects is implemented in UM. This approach treats nondeformable 3D objects with small overlaps at the contact. The presented approach consists of two parts: collision detection for arbitrary polyhedrons and then a contact force calculation. Collision detection deals with generalized three-dimensional clipping algorithm by Cyrus and Beck [2]. Contact force calculation is based on a point-plane model and computed as a sum of normal viscous-elastic and tangential dry friction forces. Several examples of application of this approach for simulation of multibody system dynamics are given.

Collision Detection for Polyhedrons

To accelerate computational processes the collision detection is typically divided into so-called far and near collision detection problems. Far collision detection is usually a fast algorithm that should select polyhedrons for the following, usually more time-consuming, near collision detection. On the first stage of the far collision detection circumscribed spheres around polyhedrons are created and its intersection is checked. The polyhedrons that passed through the far collision detection are treated by a near collision detection algorithm.

The well-known in computer graphics generalized three-dimensional clipping algorithm by Cyrus and Beck is used as the near collision detection algorithm. The algorithm deals with two convex polyhedrons and gives as a result clipped edges of one polyhedron that lie within another one and vice-versa, see Figure below. For example, the algorithm gives the set of edges {E_1, E_2, E_3} of *Body*$_2$ and the empty set of edges for *Body*$_1$ in Figure 2.46a and {E_1} of *Body*$_1$ and {E_2} of *Body*$_2$ in Figure 2.46b.



a) b)

Figure 2.46. Vertex-face and edge-edge penetration

Contact force calculation

After all collisions between the neighboring polyhedron pairs have been detected, the contact forces have to be determined. Let us consider a pair of polyhedrons. Having a set of clipped edges that belong to each polyhedron the algorithm of calculation of contact forces arranges contact points on each edge, see Figure 2.47a. The step size between neighboring contact points is a parameter of the mathematical model that depends on a characteristic dimension of polyhedrons. Then for each contact point the nearest face on another polyhedron from the pair is determined. As soon as pairs of points and planes (faces) are obtained the contact force $\overline{R}$ can be calculated

as a superposition of normal viscous-elastic force $\bar{N}$, depending on penetration Δ and its derivative, and tangential dry friction force $\bar{F}_f$, see Figure 2.47b, [2].

Special control procedure keeps position of contact points on an edge during a contacting phase even the length of clipped part of the edge is changed. Such strategy provides smooth changing a resultant vector of contact force between two bodies with time.

a) b)

Figure 2.47. Contact points and forces

The presented approach perfectly manages both basic contact situations: vertex-face and edge-edge penetrations, Figure 2.46. Since the Cyrus and Beck algorithm deals with convex polyhedrons, the presented approach for contact force simulation is also applicable for a case of convex polyhedrons only. However the suggested algorithms can be enlarged for non-convex case if non-convex polyhedrons would be preliminarily divided into convex ones. In practice this strategy works well when a convex decomposition is available with a moderate number of pieces, it breaks down for utterly non-convex objects.

2.5.9. Special forces

The following types of force elements are implemented in UM as special forces:

- Gearing;
- Combined friction;
- Cam;
- Spring.

2.5.9.1. Gearing

The gearing is realized in UM as a simplified model of contact iterations of gears. The force model can be used for simulation of plane gear (internal and external) and bevel gearing taking into account possible clearance between teeth as well as a compliance reduced to the contact point.

The gearing is specified by the following parameters (Figure 2.48):

- two points A and B on the gear axes coinciding with the gear centers (coordinates should be given in body-fixed frames);
- two unit axle vectors (in body-fixed frames);
- gearing ratio (radius of the first gear R_1 divided by the radius of the second one R_2);

- clearance δ;
- gearing stiffness coefficient reduced to the contact point c (N/m);
- gearing damping coefficient reduced to the contact point d (Ns/m);
- type of gearing internal or external, for plane gearing only.

Gear radii are calculated by UM automatically.

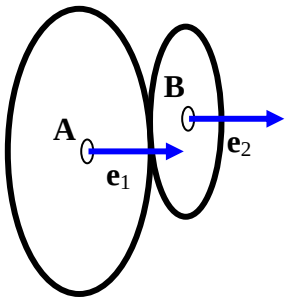


Figure 2.48.

The gearing is modeled by a tangential force at the contact as

$$F = \begin{cases} -c\left(\Delta - \frac{\delta}{2}\right) - d\dot{\Delta}, \Delta > \frac{\delta}{2} \\ -c\left(\Delta + \frac{\delta}{2}\right) - d\dot{\Delta}, \Delta < -\frac{\delta}{2} \\ 0, |\Delta| < \frac{\delta}{2}, \end{cases}$$

$$\Delta = R_2\varphi_2 \pm R_1\varphi_1.$$

Here the force acts on the second body (the opposite force acts on the first one). φ_1, φ_2 are the angles of gear rotations. The minus sign in Δ corresponds to an external gearing.

Remarks.

The gear axes must lie in a plane.

Use the *Initial conditions* tab of the *Object simulation inspector* to compute initial velocities before the simulation starts (Chapter 4).

Models:
{UM Data}\SAMPLES\LIBRARY\Gears;
{UM Data}\SAMPLES\TUTORIAL\Crusher.

2.5.9.2. Chain gear

The force element is used for a simplified description of chain gears. It is assumed that the massless chain connects two gears, which radii must be specified. Gear axes must be parallel on near parallel. The mathematical model of the chain is similar to that for the gearing force element, Sect. 2.5.9.1. *"Gearing"*, p. 2-71: the chain produces a viscoelastic force, which depends on differences in rotation angles and rates taking into account the gear ratio. It is assumed as well that only one chain strand is loaded at each moment, Figure 2.49.

In particular, the element is used for description of chain gears of bicycles and motorcycles. To implement the unilateral chain gear, the torque of the ratchet type is introduced in the joint connecting the driven gear with the rear wheel.

Model: {UM Data}\SAMPLES\LIBRARY\ChainGear.

Description of the element parameters see in Chapter 3, Sect. *Chain gear*.

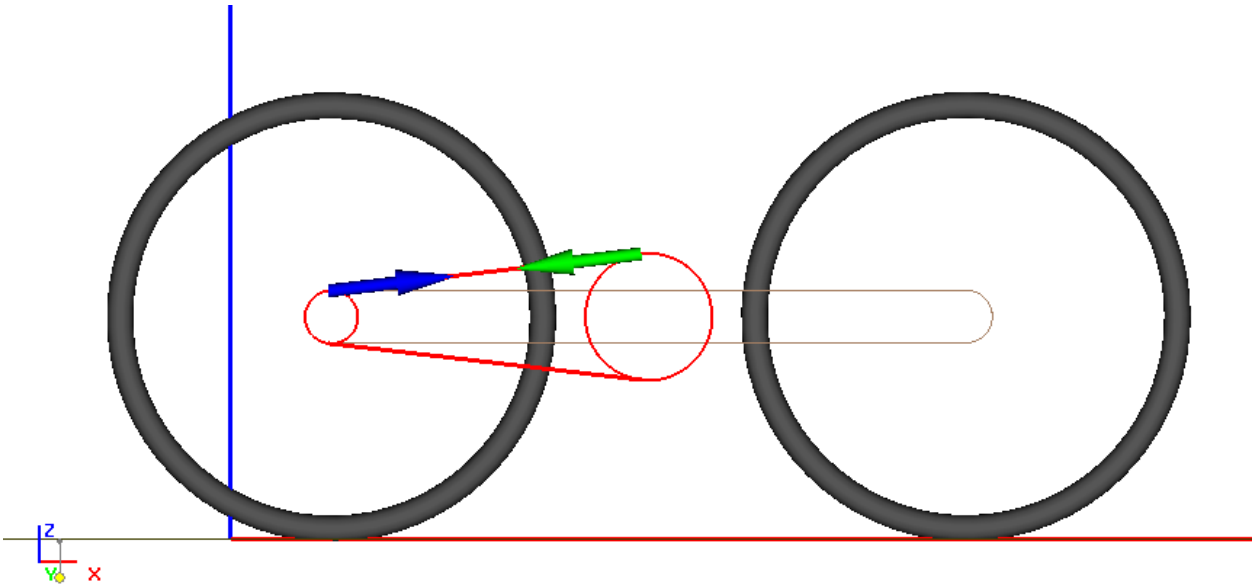


Figure 2.49. The model of chain gear

2.5.9.3. Rack and pinion

Rack and pinion force element is a special case of gearing described above (see in Sect. 2.5.9.1. *"Gearing"*, p. 2-71). The only difference is that the second body (rack) moves translational.

2.5.9.4. Combined friction

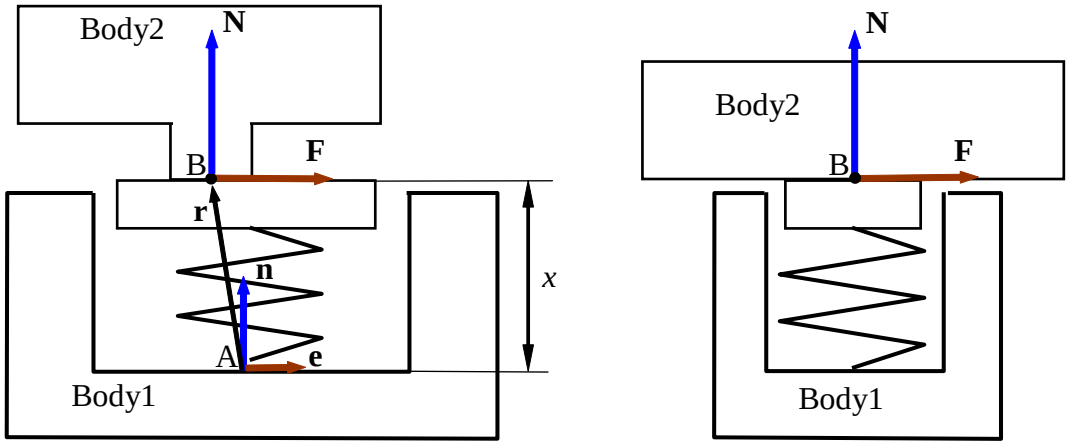


Figure 2.50. Model of force element “Combined friction”: contact point B belongs to Body 2 (left) or massless Body 1 (right)

The “combined friction” force element is a generalization of the point-plane contact. Body2 is in the contact with a massless body. Contact point B belongs either to Body2 or to the massless body and slides on a contact plane, which is perpendicular to the normal n, Figure 2.50. The massless body is connected with Body1 by a force element and moves relative to Body1 along the force element axis n. In the *fictitious body mode*, the massless body can move relative to Body1 in one or two tangential directions. The model of interaction of bodies 1 and 2 includes a normal force N directed parallel to the normal **n**, and the friction force. In the sliding mode, the friction force depends on the normal force according to the usual dry friction law $F = \mu|N|$ with μ as the coefficient of friction.

Main differences of point-plane contact and combined friction element consist in the following features.

- Normal force law can be any nonlinear.
- Lateral displacements are available for the fictitious body.
- Limitations on lateral shift both fictitious body relative to Body1 and Body2 relative to the fictitious one.

The force element connecting body 1 with the massless body is attached to point A of Body1. It is assumed, that during the motion relative to point A the contact plane is stay in the positive direction specified by the vector **n**. The distance $x = \mathbf{n} \cdot \overline{AB} = \mathbf{n} \cdot \mathbf{r} > 0$ from point A to the contact plane is variable.

The normal force in this model is a function of distance x, the first time derivative of the distance v, and of time t (see Sect. 2.5.5. *"Types of scalar forces"*, p. 2-37),

$$N = N(x, v, t).$$

The following modes of the normal force are available:

- *unilateral*, when the interaction forces vanishes by $N < 0$, that means, the contact point leaves the contact plane at $N = 0$.
- *bilateral*, when the normal force can be negative.

As a rule, the element is used in the *unilateral* mode.

Two modes are realized for the **friction force**: sliding and sticking. In the 2dD model fo the force element, the friction force is directed along a line rigidly couple with Body1. In a 3D case, the friction force lies in the plane perpendicular to the element axis **n**.

In any case, the body on which slides Body2 is massless, but two modes should be taken into account: *without the fictitious body and with it*. In its turn, it is possible to set a limitation of shift of the fictitious body relative to Body1 or Body2 relative to fictitious body.

Here is the list of parameters:

- coordinates of points A, B in SC1 and SC2, respectively;
- projections of vector n in SC1;
- model of normal force N;
- type of contact: unilateral or bilateral;
- type of element: 2D or 3D;
- dynamic and static coefficients of friction for friction force F;
- unit vector e in SC1 specifying the direction of friction force for 2D type of element;
- coefficients of lateral contact stiffness and damping c, d in the friction sticking mode (are not required for some element modes);
- coefficients of lateral stiffness and damping c_y, d_y for connection of the fictitious body with Body1 (for the mode with fictitious body only);
- type of lateral shift limitations and gap parameters (are not required for some element modes).

2.5.9.4.1. Mode without fictitious body

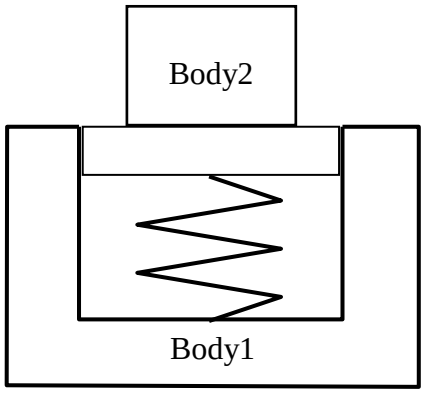


Figure 2.51. Scheme of “combined friction” force element without fictitious body

This mode is available both for **2D** and **3D** element models.

It is assumed that the massless body on which plane Body2 slides, can move along the force element axis **n** only, and cannot move in the lateral direction, Figure 2.51. In this case, the stand-

ard friction force in lateral direction is realized by the element, which in the simplest form is presented by the point-plane contact, Sect. 2.5.7.1. *"General information about contact interactions"*, p. 2-59. The main difference from the point-plane contact consists in the possible nonlinear dependence of the normal force N from the vertical displacement and velocity.

The combined friction in the considered mode can be realized by an equivalent multibody model. Third body 3 with a small mass value with one degree of freedom relative to Body1 is introduced. Body3 is connected with Body1 by a force element shown in Figure 2.51 as a spring. One or several contact points are introduced between Body2 and Body3. Such the model is analogous to the combined friction force element in dynamic sense. The disadvantage of this model consists in introduction of additional body, degree of freedom and a contact force element, i.e. it makes simulation slower. Thus, the combined friction element allows simplifying the modal and making simulation faster.

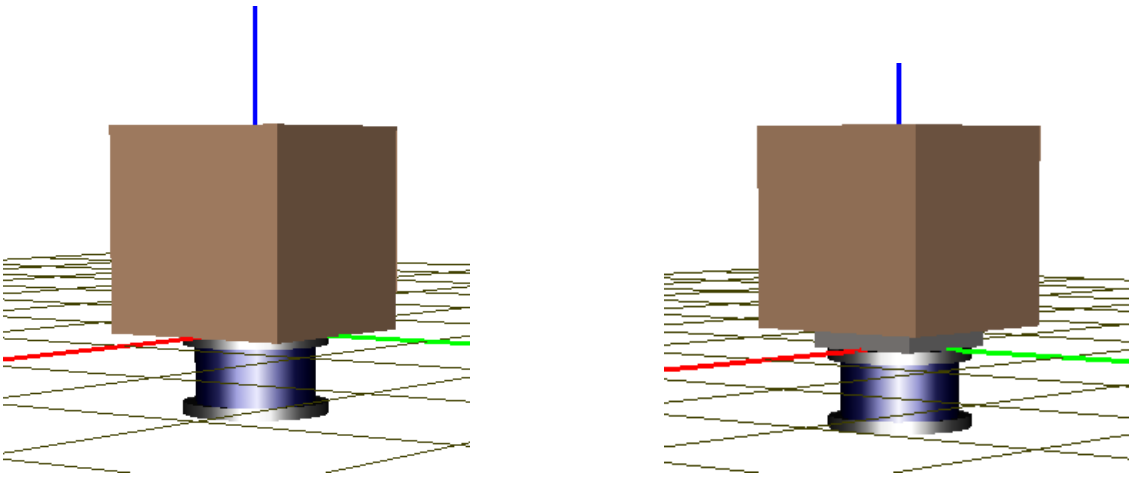


Figure 2.52. Examples of combined friction without fictitious body (left) and its equivalent scheme with the third body (right)

Example
Models:
{UM Data}\SAMPLES\LIBRARY\CombFriction\CF2D_without_fict;
{UM Data}\SAMPLES\LIBRARY\CombFriction\CF2D_without_fict_MBS.

The example illustrates the use of the combined friction force element without fictitious body, and makes proof of correctness of its mathematical model. Two models to be compared are shown in Figure 2.52. The models look quite similar, but the first model includes the combined friction element, whereas the second one is the multibody analogue of the combined friction. The upper body is under horizontal harmonic oscillations; it is supported by the force element (the first model) or by the body with a small mass through a contact force (the second model). The body possesses a vertical degree of freedom, along which a periodic force can act. A detailed description of the models can be found in Chapter 7. Here we consider the main result.

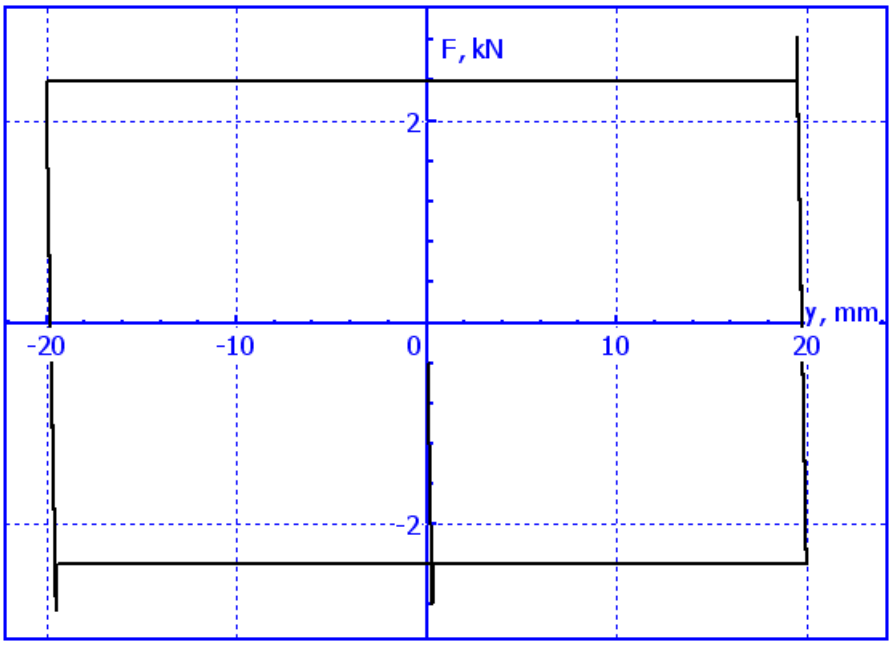


Figure 2.53. Friction force hysteresis for a constant normal force

In the case of the constant load the two models show practically the same results in dependence of friction force on the lateral displacements, Figure 2.53.

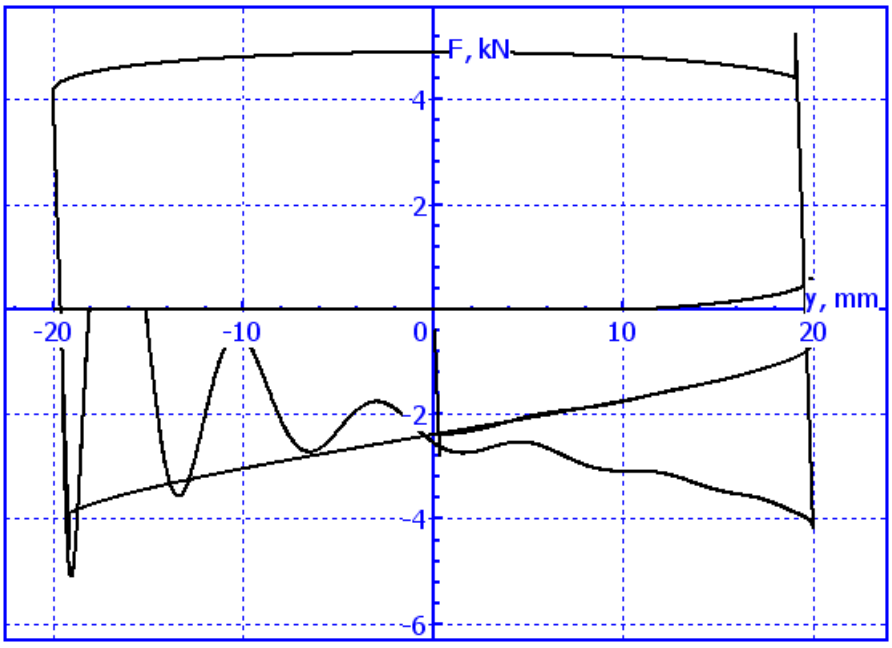


Figure 2.54. Hysteresis of friction force vs. normal force

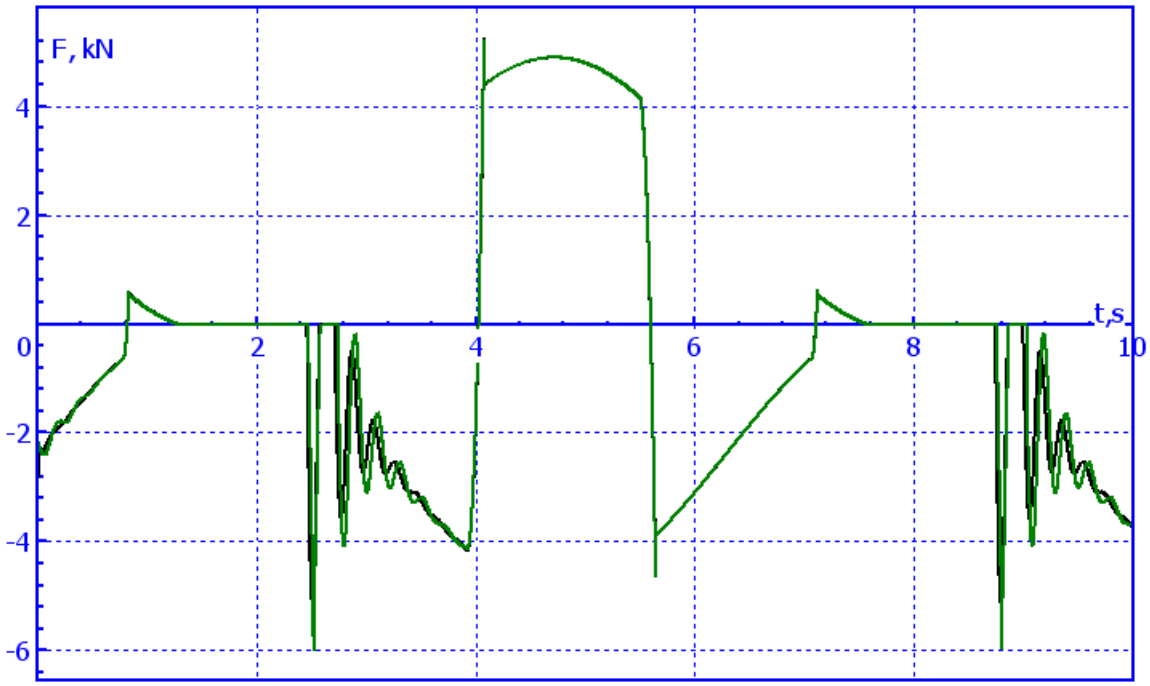


Figure 2.55. Comparison of dependences of friction force on time

By the variable vertical load under the action of periodic vertical force, dependence of force on lateral shift shows a very small difference, Figure 2.54 (the plot is obtained for the first model). The most significant differences are take place be the lifting of Body2 and successive impacts, Figure 2.55.

The results of simulation of the models are similar. This fact gives proof for correctness of the combined friction element.

2.5.9.4.2. Mode with fictitious body without lateral constraints

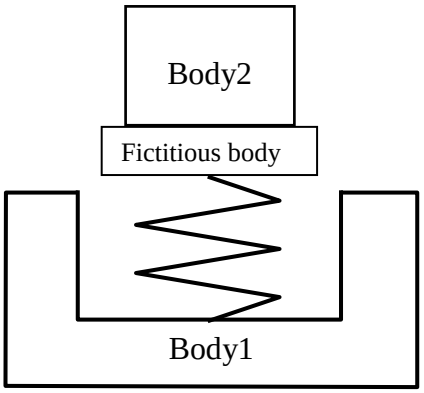


Figure 2.56. Model with fictitious body without constraints

This mode is available both in **2D** *and in* **3D** *element model.*

It is assumed that the fictitious body (FB) with zero mass on which Body2 slides, can additional move parallel to the contact plane, Figure 2.56. Thus, the FB has two degrees of freedom in the contact plane in case of 3D element, and one d.o.f. in the case of 2D element.

A linear viscous-elastic lateral force R with stiffness and damping constants c_y, d_y appears by shifts on the FB relative to Body1. Lateral shifts of the FB take place *in the tangential plane*.

In the sticking mode, the shift of the FB is fully defined by motion of point B in the tangential plane of Body2.

In the sliding mode, the shift is specified from the equilibrium equation for the FB

$$R = F,$$

where R, F are the viscous-elastic force and friction force applied from the FB to Body2.

For **3D** element, this equation has the form of two differential equations

$$-d_y\dot{x} - c_y x = \mu N \frac{(\dot{x} - v_{2x})}{v_s},$$
$$-d_y\dot{y} - c_y y = \mu N \frac{(\dot{y} - v_{2y})}{v_s}.$$

Here

x, y are the coordinates of the FB in the tangential plane;

v_{2x}, v_{2y} are the projections on the tangential plane of velocity of point B of Body2 relative to Body1;

$\dot{x} - v_{2x}$, $\dot{y} - v_{2y}$ is the sliding velocity of point B on the FB;

$v_s = \sqrt{(\dot{x} - v_{2x})^2 + (\dot{y} - v_{2y})^2}$ is the absolute value of the sliding velocity.

These nonlinear differential equations must be solved together with equations of motion of the modeled object. Assuming the normal force N constant during the single integration step and making the approximate substitution

$$x = x_p + v_x h, y = y_p + v_y h,$$
$$v_x = \dot{x}, v_y = \dot{y}$$

the differential equations are replaced by nonlinear algebraic equations relative to the unknown velocities v_x, v_y of the FB. Here h is the step size, x_p, y_p are coordinates of the FB on the previous integration step. Note that the described procedure is equivalent to applying the implicit Euler method to solving the differential equations.

For the **2D** element, one differential equation takes place corresponding to the shift of point B along the element axis **e**, Figure 2.51.

$$-d_y\dot{x} - c_y x = \mu N \frac{(\dot{x} - v_{2x})}{|\dot{x} - v_{2x}|} = F.$$

Assuming a constant value of the normal force N during one integration step and sliding velocity does not change its sign during this interval, force F in the right hand side of the equation is constant, and the relation valid

$$x = \left(x_p + \frac{F}{c_y}\right) e^{-\frac{c_y}{d_y}(t-t_p)} - \frac{F}{c_y}.$$

Here 2 is the value of coordinate *x* on the previous integration step at the moment t_p.

The combined friction in the discussed mode can be realized by an equivalent multibody model similar to that described in the previous section. The body with a small mass must have in this case either three (3D) or two (2D) translational degrees of freedom relative to Body1.

Example.

Models:

{UM Data}\SAMPLES\LIBRARY\CombFriction\CF2D_with_fict;

{UM Data}\SAMPLES\LIBRARY\CombFriction\CF2D_with_fict_MBS.

Consider an example illustrating the use of the combined friction with FB without limitations on shift and proving the correctness of its mathematical model and implementation, Figure 2.52. The general structure of the models is analogous to that for the example in Sect. 2.5.9.4.1. *"Mode without fictitious body"*, p. 2-75. Comparison of models in Figure 2.57 confirms the correctness of mathematical models described in this section.

Remark. Sometimes a multibody model has an important advantage in comparison with the combined friction because it lets to simulation of self-excited vibrations, which can appear is such the systems.

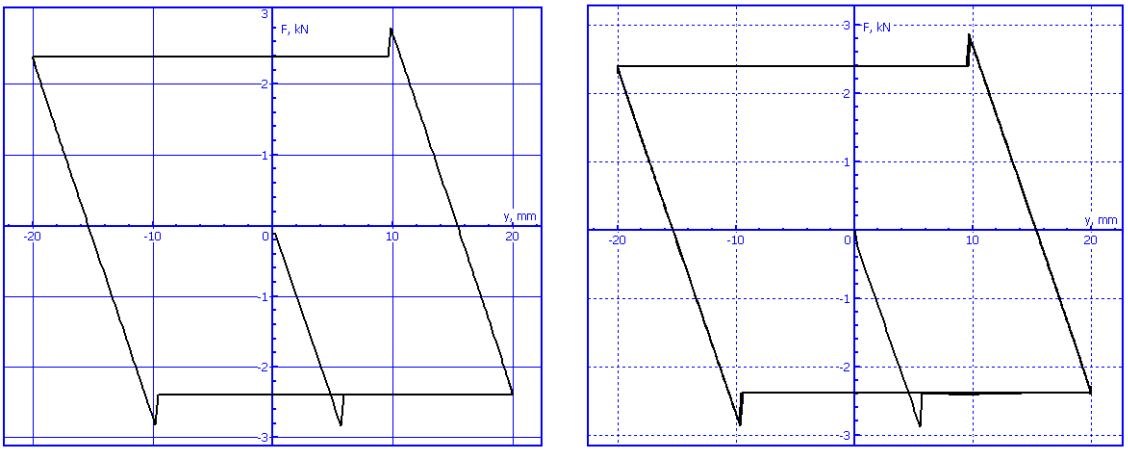


Figure 2.57. Hysteresis of friction force for a constant normal force for the combined friction (left) and for the multibody model (right)

2.5.9.4.3. Mode with restrictions on shift of fictitious body

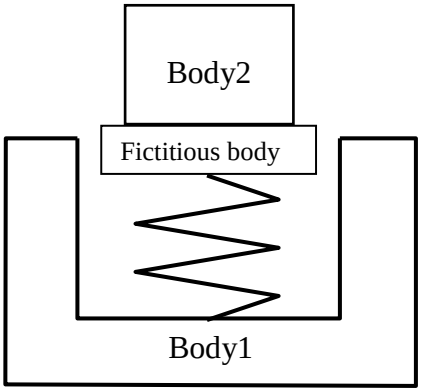


Figure 2.58. Restriction on lateral shift of FB

This mode is implemented for **2D** *element with FB.*

In this mode, the lateral shift of the FB has limits such as (Figure 2.58)

$$x \in [x_{min}, x_{max}], x_{min} < 0, x_{max} > 0,$$

where x_{min}, x_{max} are limit values of the lateral coordinate of the FB.

Example.

Model: {UM Data}\SAMPLES\LIBRARY\CombFriction\CF2D_with_fict_limit_fict.

Consider an example illustrating the use of the combined friction with the FB. The general structure of the model is similar to that in Sect. 2.5.9.4.1. *"Mode without fictitious body"*, p. 2-75. The test computation of frictions force by harmonic lateral oscillations of Body2 relative to Body1 with amplitude 20 mm is presented in Figure 2.59. A 10 mm gap for a side is specified.

Remark. If the FB does not reach the limits, the force element model coincides with than in Sect. 2.5.9.4.2. *"Mode with fictitious body without lateral constraints"*, p. 2-78.

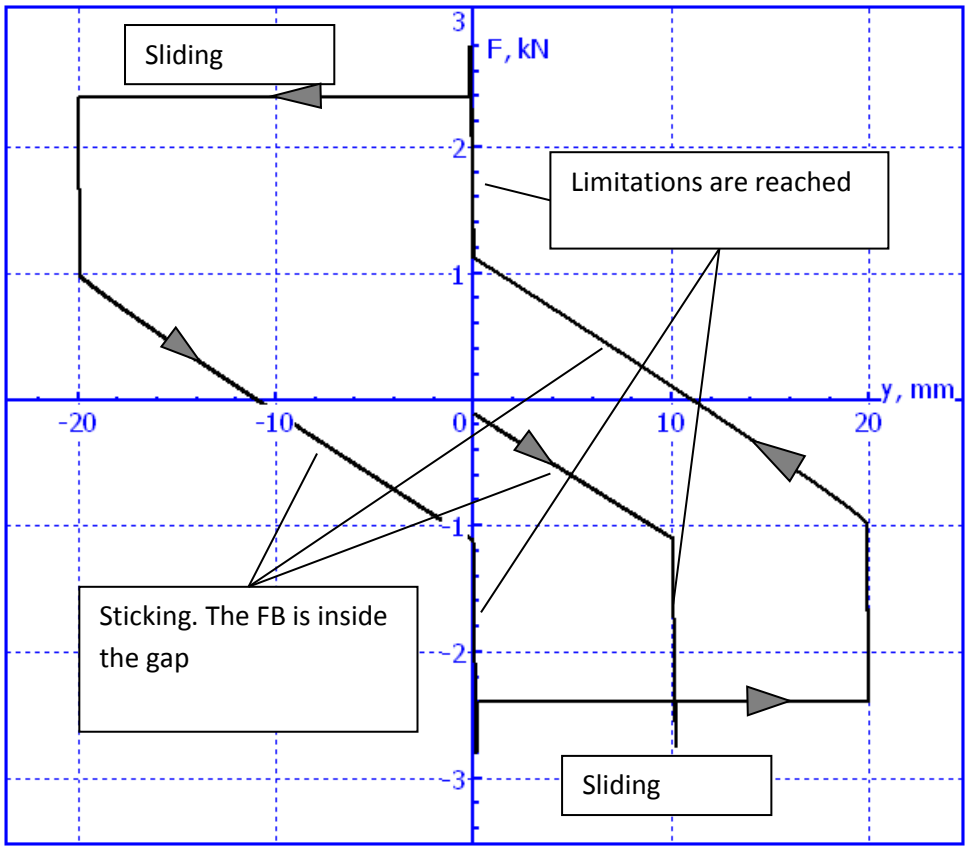


Figure 2.59. Hysteresis of force element with restrictions of the lateral shift of FB

2.5.9.4.4. Mode with restriction on Body2

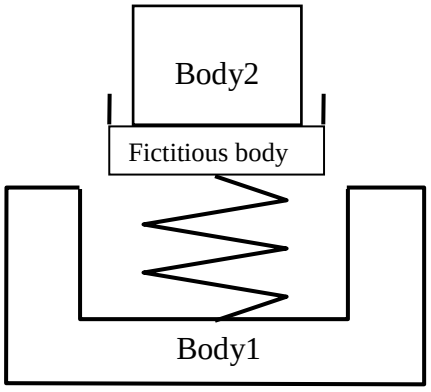


Figure 2.60. Limitations on lateral shift of Body2 relative to FB

The model is implemented for 2D element with the FB.

In this model, the lateral shift Δx_2 of Body2 relative to FB is limited as (Figure 2.60)

$$\Delta x_2 \in [x_{min}, x_{max}], x_{min} < 0, x_{max} > 0,$$

where x_{min}, x_{max} are limit values of lateral shift.

Example.

Model: {UM Data}\SAMPLES\LIBRARY\CombFriction\CF2D_with_fict_limit_body2.

Consider an example illustrating the use of the combined friction with restriction on lateral shift of Body2 relative to FB. The general structure of the model is similar to that in Sect. 2.5.9.4.1. *"Mode without fictitious body"*, p. 2-75. The test computation of frictions force by harmonic lateral oscillations of Body2 relative to Body1 with amplitude 20 mm is presented in Figure 2.60. A 10 mm gap for a side is specified.

Remark. If Body2 does not reach the limits, the force element model coincides with than in Sect. 2.5.9.4.2. *"Mode with fictitious body without lateral constraints"*, p. 2-78.

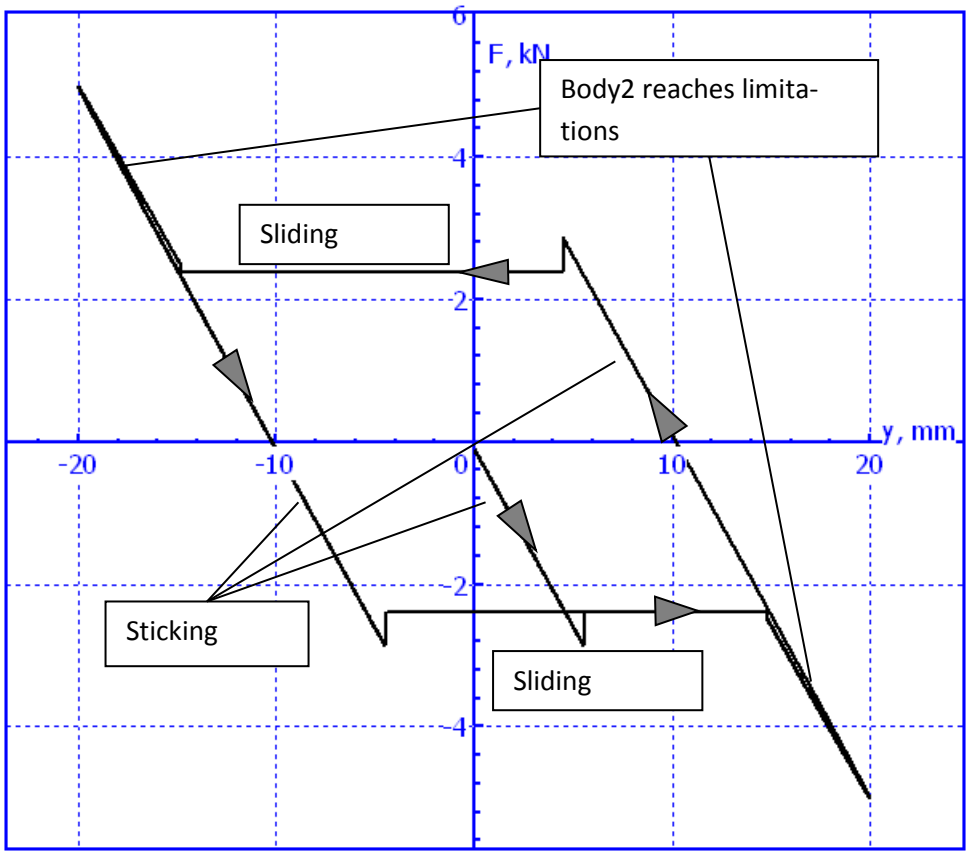


Figure 2.61. Hysteresis of force element with restrictions of the lateral shift of Body2 relative to FB

2.5.9.5. Cam

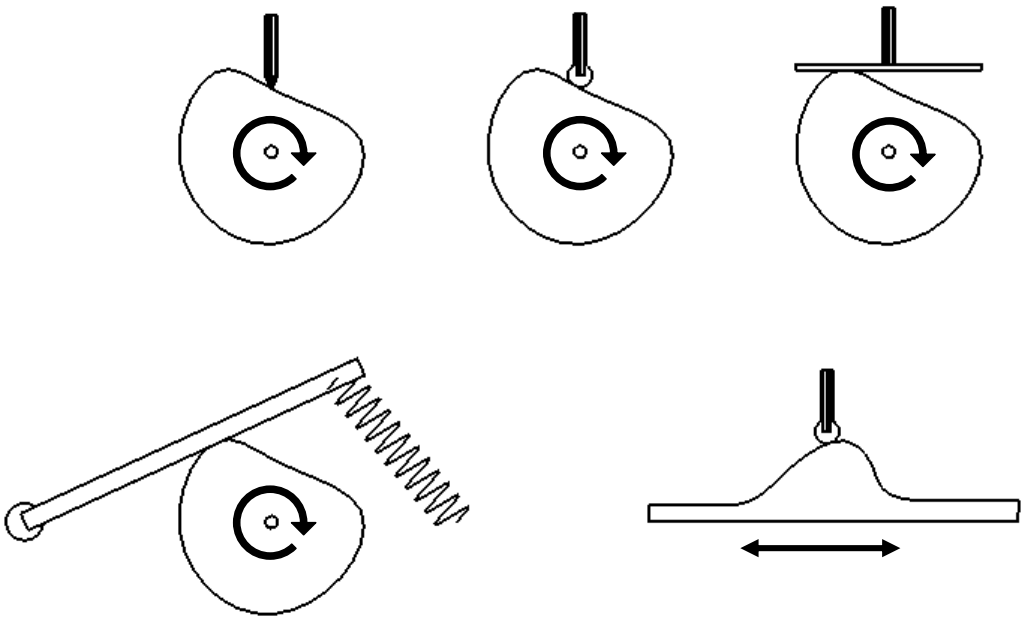


Figure 2.62. Types of cam models

The special case of contact interaction is an interaction of a cam and a piston or a link. The mathematical model of this interaction is similar to one described above (Sect. 2.5.7.1.1. *"Compliant contact"*, p. 2-59). Types of cam-piston (cam-link) couples and examples of their using are shown in the Figure 2.62.

Three types of cam pairs are implemented in UM depending on the contact type:

- **point** (the first figure);
- **roller** (2nd and 5th figure);
- **plane** (3rd and 4th figure).

The Coulomb (dry) friction is taken into account for the plane and point contact types, whereas the roller contact is considered as an ideal one (without friction).

Note. There is only one contact point at each moment of time. In the multipoint case actually the point with the maximal penetration is considered.

The examples of the description and/or usage:
Chapter 3. Sect. *"Cam"*;
{UM Data}\SAMPLES\Mechanisms\cams.

2.5.9.6. Spring

Special force element *Spring* is a special case of a general linear force element (Sect. 2.5.6. *"Generalized linear force element"*, p. 2-57), which is used for modeling linear helical springs with equal shear stiffness in any direction perpendicular to the spring axis and with the symmetric rigid fixing of ends. It is supposed also that the spring axis is parallel to one of the axes of the first body-fixed system of coordinates.

Stiffness matrix, which is introduced in Sect. 2.5.6. *"Generalized linear force element"*, p. 2-57, for a spring parallel to Z-axis of the body1-fixed SC, has the following form:

$$C = \begin{pmatrix} c_s & 0 & 0 & 0 & -c_s H/2 & 0 \\ 0 & c_s & 0 & c_s H/2 & 0 & 0 \\ 0 & 0 & c_l & 0 & 0 & 0 \\ 0 & c_s H/2 & 0 & c_\varphi & 0 & 0 \\ -c_s H/2 & 0 & 0 & 0 & c_\varphi & 0 \\ 0 & 0 & 0 & 0 & 0 & c_a \end{pmatrix}$$

Here c_s, c_l, c_φ, c_a are shear, longitudinal (axial), bending and torsion stiffnesses of the spring, H is the current length of the spring.

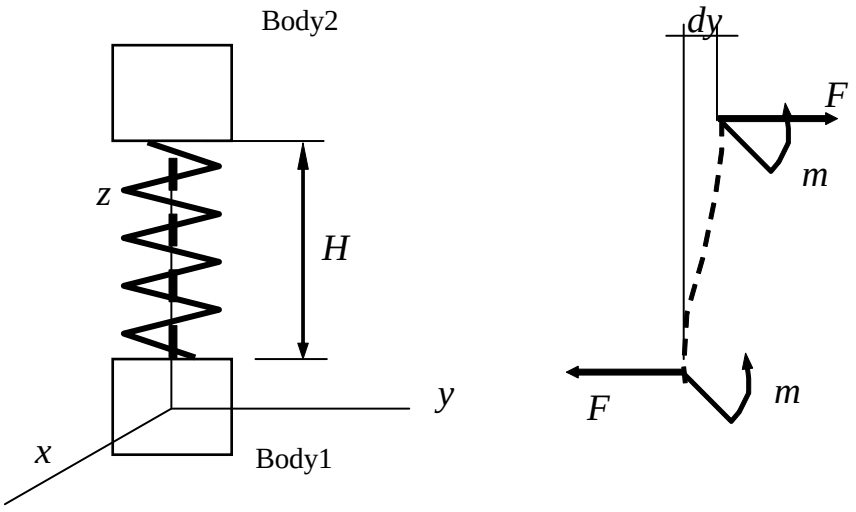


Figure 2.63. Spring forces

Let us consider how we can get the values for non-diagonal elements of this matrix. Let the second body be over the first one. Shift body 2 along Y-axis on *dy*, Figure 2.63. The force $F_y = -c_s dy$ appears. An opposite force acts on the upper end of the spring and the same force acts on the lower end. Equilibrium of the spring requires a balancing pair of forces with the moment $c_s dy H$. This pair must be realized by two equal (due to symmetry) pairs $c_s dy H/2$ in the upper and lower ends. Thus, the moment with the X-projection $c_s dy H/2$ acts on the second body, which results in the non-diagonal element of the stiffness matrix

$$C_{42} = c_s \frac{H}{2}$$

as well as in the element C_{24} which is equal to C_{42} due to symmetry of the matrix. Elements $C_{52} = C_{25}$ can be obtained analogously.

If the spring is parallel to the Y-axis, its stiffness matrix is

$$C = \begin{pmatrix} c_s & 0 & 0 & 0 & 0 & \pm c_s\, H/2 \\ 0 & c_l & 0 & 0 & 0 & 0 \\ 0 & 0 & c_s & \mp c_s\, H/2 & 0 & 0 \\ 0 & 0 & \mp c_s\, H/2 & c_\varphi & 0 & 0 \\ 0 & 0 & 0 & 0 & c_a & 0 \\ \pm c_s\, H/2 & 0 & 0 & 0 & 0 & c_\varphi \end{pmatrix}.$$

This matrix is obtained from the previous one permutations $x \to z, y \to x, z \to y$. Finally, here is the matrix for a X-parallel element.

$$C = \begin{pmatrix} c_l & 0 & 0 & 0 & 0 & 0 \\ 0 & c_s & 0 & 0 & 0 & \pm c_s\, H/2 \\ 0 & 0 & c_s & 0 & \pm c_s\, H/2 & 0 \\ 0 & 0 & 0 & c_a & 0 & 0 \\ 0 & 0 & \mp c_s\, H/2 & 0 & c_\varphi & 0 \\ 0 & \mp c_s\, H/2 & 0 & 0 & 0 & c_\varphi \end{pmatrix}.$$

Two types of specifying stiffness parameters are implemented: theoretical evaluation of a coil spring and experimental data.

2.5.9.6.1. Theoretical spring stiffness

The stiffness parameters are computed according to the following formulas:

$$c_l = \frac{Gd^4}{64nR^3}, c_a = \frac{Ed^4}{128Rn},$$
$$c_s = \frac{1}{\frac{1}{c_{1s}} + \frac{H^2}{12c_{2s}}}, c_\varphi = \frac{c_{2s}}{1 - \frac{H^2}{4c_{2s}\left(\frac{1}{c_{1s}} + \frac{H^2}{3c_{2s}}\right)}}.$$

Here $c_{ls} = 2(1+\mu)c_l, c_{2s} = 2c_a/(2+\mu)$, *G*, *E* are the shear and Young modulus, μ is the Poisson's ratio, *d* in the diameter of spring wire, *R* is the spring radius, *n* is the number of active coils.

Thus, the shear and bending stiffnesses depend on the current spring length *H*, so they are not constant at simulation. In particular, the shear stiffness increases with the compression of the spring.

2.5.9.6.2. Setting stiffnesses by experimental data

The user can set both constant stiffness parameters and variable depending on the axial compression.

Note. If it is necessary to describe a spring, which is not parallel to one of the body1-fixed axis, a special fictitious body should be introduced. This body must be fixed

to body1 one of its coordinate axis must have the desirable direction. Joint fixing the fictitious body and body1 can be a generalized joint, which allows constant rotations as elementary transformations (Sect. 2.2.2.3. *"Generalized joint"*, p. 2-17), and can provide any orientation of the fictitious body relative to body1.

Examples of description and/or usage:
Chapter 7. Sect. *Models of Springs*;
{UM Data}\SAMPLES\Rail_Vehicles\AC4.

2.5.9.7. Bushings

Element of this type is used for modeling linear and nonlinear compliant joints (bushings).

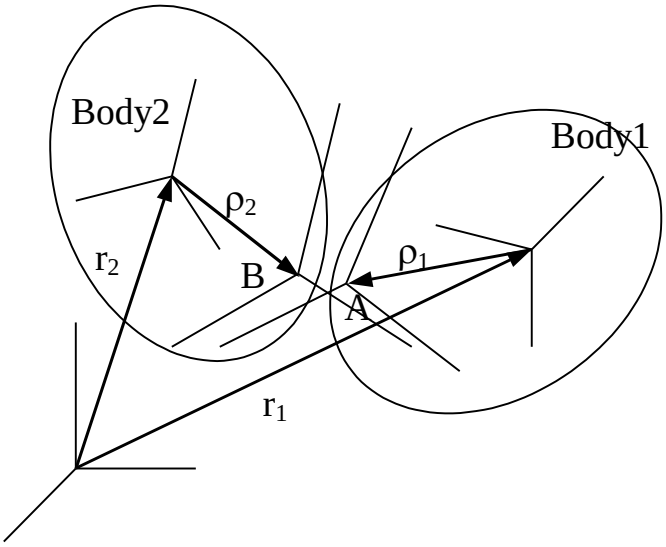


Figure 2.64. Systems of coordinates of a bushing

Consider the mathematical model of a bushing. Let us introduce two systems of coordinates. The first system SCB1 is fixed relative to body 1, the second one SCB2 is fixed relative to body 2. Vectors ρ_1, ρ_2 and constant rotation matrices A_{B1}, A_{B2} determine the positions of SCB1 and SCB2 relative to the bodies.

It is supposed that deviation of SCB2 relative to SCB1 is small.

In general the mathematical model of the bushing as a force element is expressed in terms of the displacement of point B relative to SCB1

$$\Delta r = r_2 + \rho_2 - r_1 - \rho_1$$

and the direct cosine matrix, which sets the orientation of SCB2 relative to SCb1

$$A_{B1B2} = A_{B1}^T A_{10} A_{02} A_{B2}$$

This matrix is nearly the identity one for small deviations of SCB2 from SCB1. In this case a vector of small rotation of SCB2 relative to SCB1 can be introduced as

$$\Delta\pi = \begin{pmatrix} \Delta\pi_x \\ \Delta\pi_y \\ \Delta\pi_z \end{pmatrix}.$$

Components of this vector correspond to small rotation angles of SCB2 about axis of SCB1.

The skew-symmetric matrix corresponding to the rotation vector depends on the direct cosine matrix according to the formula

$$\Delta\tilde{\pi} = \begin{pmatrix} 0 & -\Delta\pi_z & \Delta\pi_y \\ \Delta\pi_z & 0 & -\Delta\pi_x \\ -\Delta\pi_y & \Delta\pi_x & 0 \end{pmatrix} = \frac{A_{B1B2} - A_{B1B2}^T}{2}.$$

This expression is used for evaluation of the rotation vector by the simulation.

Input of the element parameters see in Chapter 3, Sect. *Data Input* | *Input of force elements* | *Special forces* | *Bushings.*

2.5.9.7.1. Linear model of bushing

In case of a linear bushing its mathematical model is:

$$\begin{pmatrix} F^{B1} \\ M^{B1} \end{pmatrix} = \begin{pmatrix} F_0 \\ M_0 \end{pmatrix} - C \begin{pmatrix} \Delta r - \Delta r_0 \\ \Delta\pi - \Delta\pi_0 \end{pmatrix} - D \begin{pmatrix} v_{12} \\ \omega_{12} \end{pmatrix}$$

Here F^{B1}, M^{B1} are the force and the torque produced by the element expressed in SCb1 and reduced to point B. The following constant parameters specifying the bushing model must be set by the user (all vectors are resolved in SCB1):

- C and D are the constant diagonal matrices, which specify stiffness and damping constants by movement/rotation along/about axes of SCB1;

- the vectors $\Delta r_0, \Delta\pi_0$ set small offset, i.e. a 'constant' deviation of CSB2, for instance in stationary position of the model;

$v_{12} = \Delta\dot{r}$ is re velocity of the SCB2 origin relative to the second body;

$\omega_{12} = \omega_2 - \omega_1$ is the relative angular velocity;

F_0, M_0 are constant values of force and torque for zero relative velocities and $\Delta r = \Delta r_0$, $\Delta\pi = \Delta\pi_0$; these vectors are usually used for setting the stationary or static values of forces, often by $\Delta r_0 = 0, \Delta\pi_0 = 0$.

2.5.9.7.2. Pointwise model of bushing

In case of a pointwise bushing its mathematical model is:

$$\begin{pmatrix} F^{B1} \\ M^{B1} \end{pmatrix} = \begin{pmatrix} F_0 \\ M_0 \end{pmatrix} - \begin{pmatrix} F_e \\ M_e \end{pmatrix} - D \begin{pmatrix} v_{12} \\ \omega_{12} \end{pmatrix}$$

Nonlinear part of the model is set by points and concentrated in dependence of the force and torque F_e, M_e on the corresponding components of vectors $\Delta r - \Delta r_0, \Delta\pi - \Delta\pi_0$.

$$F_{ex} = F_{ex}(\Delta x - \Delta x_0), F_{ey} = F_{ey}(\Delta y - \Delta y_0), F_{ez} = F_{ez}(\Delta z - \Delta z_0),$$
$$M_{ex} = M_{ex}(\Delta\pi_x - \Delta\pi_{0x}), M_{ey} = M_{ey}(\Delta\pi_y - \Delta\pi_{0y}), M_{ez} = M_{ez}(\Delta\pi_z - \Delta\pi_{0z}).$$

Nonlinear dependencies of these components on displacements must be set by the user as plots.

In particular, a pointwise bushing is often used for modeling or contacts with gaps on translational and rotational degrees of freedom.

2.5.9.7.3. Generalized bushing

The generalized bushing is the most powerful element for description of nonlinear force connecting bodies. Projections of the force and torque in SCB1 are set with the help of models of scalar forces described in Sect. 2.5.5. *"Types of scalar forces"*, p. 2-37. A projection of the force depends on the relative displacement along the corresponding axis, on the time derivative of this

displacement, and time *t*. Analogously a torque projection depends on the angle of relative rotation about the axis, on the projection of the relative angular velocity and *t*.

$$F_{ex} = F_{ex}(\Delta x, \Delta\dot{x}, t), F_{ey} = F_{ey}(\Delta y, \Delta\dot{y}, t), F_{ez} = F_{ez}(\Delta z, \Delta\dot{z}, t),$$
$$M_{ex} = M_{ex}(\Delta\pi_x, \omega_{12,x}, t), M_{ey} = M_{ey}(\Delta\pi_y, \omega_{12,y}, t), M_{ez} = M_{ez}(\Delta\pi_z, \omega_{12,z}, t).$$

The linear and pointwise bushings are particular cases of the generalized type corresponding to the linear (Sect. 2.5.5.1. *"Linear force"*, p. 2-37) and pointwise (Sect. 2.5.5.7. *"Pointwise model"*, p. 2-45) scalar forces.

Use of sets of forces in parallel allows describing connections with complex behavior, which can differ for different directions.

2.5.9.8. Air springs

The following models of air springs are available: tabular, Nishimura's, Berg's, thermodynamic.

List of symbols:

n – Polytropic index;

R – Gas constant for air;

A_e – Effective area of air spring;

dA_e/dz – Effective area gradient;

P_a – Atmospheric pressure;

P_0 – Internal pressure in air spring at equilibrium state;

P_b – Bellow pressure;

P_t – Auxiliary tank pressure;

V_b – Bellow volume;

V_t – Auxiliary tank volume;

L_p – Pipe length;

d_p – Pipe diameter;

λ – Distributed pressure drop coefficient;

ζ – Lumped pressure drop coefficient;

C – Sonic conductance;

b – Critical pressure ratio.

2.5.9.8.1. Tabular model

The tabular model of the air spring uses experimental static tabular data on the dependence of the F force and V volume on the height of the air spring and the air pressure inside its p. These data are obtained by isobaric loading of the air spring. The detailed mathematical model of the tabular air spring is described in Chapter 31 of the user's guide. The tabular model is the only model of the air spring that allows you to create pneumatic systems.

2.5.9.8.2. Nishimura model

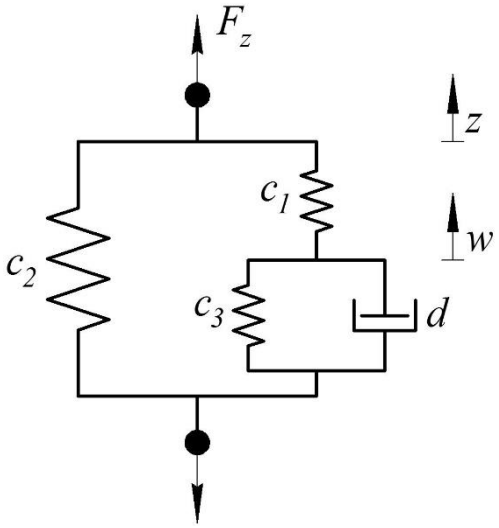


Figure 65. Nishimura model

$$c_1 = nA_e^2\frac{P_0}{V_b}, \qquad c_2 = (P_0 - P_a)\frac{dA_e}{dz}, \qquad c_3 = nA_e^2\frac{P_0}{V_t}, \qquad d = \frac{0{,}126}{d_p^3}A_e^2\frac{P_0}{RT}g$$

$$\begin{cases} d\dot{w} = c_1(z - w) - c_3 w \\ F_z = (P_0 - P_a)A_e + c_2 z + c_1(z - w) \end{cases}$$

2.5.9.8.3. Berg model

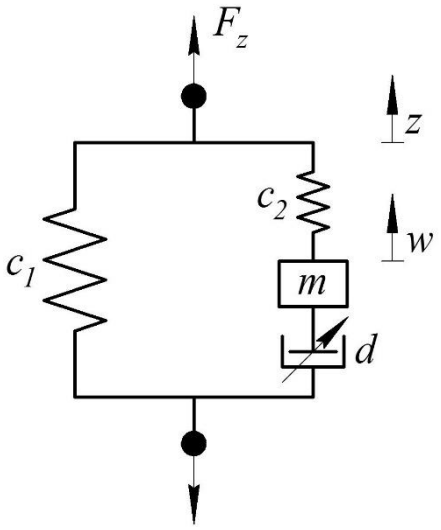


Figure 66. Berg model

$$c_1 = nA_e^2\frac{P_0}{V_b + V_t}, \qquad c_2 = c_1\frac{V_t}{V_b}, \qquad d = \frac{1}{2}A_p\rho k_T\left(\frac{A_e}{A_p}\frac{V_t}{V_b + V_t}\right)^3,$$

$$m = L_p A_p \rho\left(\frac{A_e}{A_p}\frac{V_t}{V_b + V_t}\right)^2$$

$$k_T = \lambda\frac{L_p}{d_p} + \zeta, \qquad \rho = \frac{P_0}{RT}$$

$$\begin{cases} m\ddot{w} = c_2(z-w) - d\dot{w}^2\text{sing}(\dot{w}) \\ F_z = (P_0 - P_a)A_e + c_1 z + c_2(z-w) \end{cases}$$

The Berg's model takes into account the inertia effect of air in the pipeline.

2.5.9.8.4. Thermodynamic model

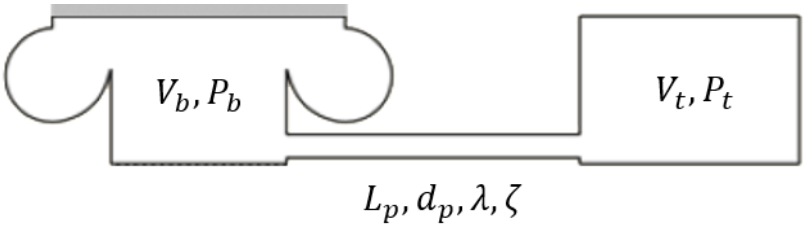


Figure 67. Thermodynamic model

It is considered that the connection between pressure, volume and mass of air in the bellow and auxiliary tank is described by the polytropic equation, so:

$$\begin{cases} \dot{P}_b = \dfrac{n}{V_b}\left[RT\left(\dfrac{P_b}{P_0}\right)^{\frac{1-n}{n}} q - P_b\dfrac{dV_b}{dz}\dot{z}\right] \\ \dot{P}_t = \dfrac{n}{V_t}RT\left(\dfrac{P_t}{P_0}\right)^{\frac{1-n}{n}} q \\ F_z = (P_b - P_a)A_e(z) \end{cases}$$

Equations of a pipeline mass flow q:

- Algebraic

$$q = \sqrt{\frac{2\rho_p A_p^2}{k_T}|P_t - P_b|}\text{sign}(P_t - P_b)$$

$$q = \begin{cases} P_1 C\rho_{ref}\sqrt{\dfrac{T_{ref}}{T_1}}\sqrt{1 - \left(\dfrac{P_2/P_1 - b}{1-b}\right)^2} & \dfrac{P_2}{P_1} > b \\ P_1 C\rho_{ref}\sqrt{\dfrac{T_{ref}}{T_1}} & \dfrac{P_2}{P_1} \le b \end{cases}$$

- Differential

$$\dot{q} = \frac{A_p}{L_p}\left[(P_t - P_b) - \frac{1}{2\rho_p A_p^2}\left(\lambda\frac{L_p}{D_p} + \zeta\right)q^2\text{sign}(q)\right]$$

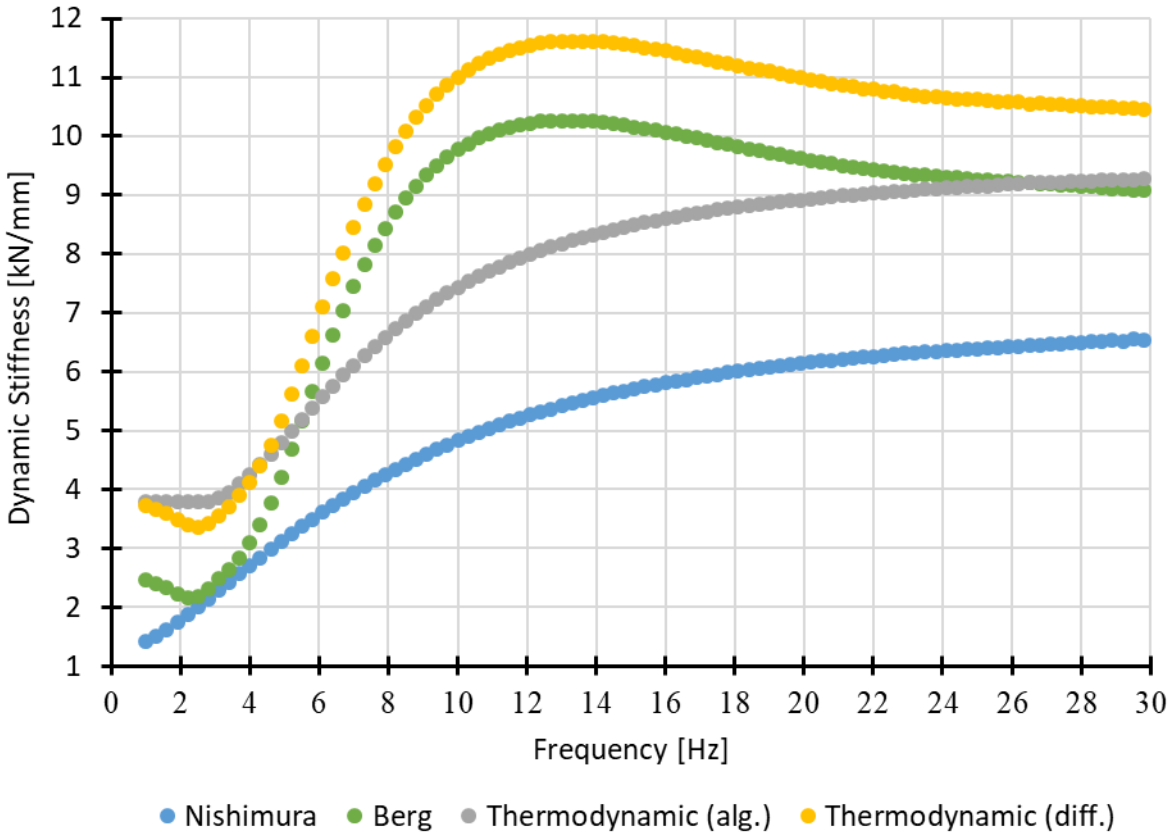


Figure 68. Comparison of models

Models of air springs:
{UM Data}\Samples\Library\AirSpring\Nishimura;
{UM Data}\Samples\Library\AirSpring\Berg;
{UM Data}\Samples\Library\AirSpring\Thermodynamic.

2.5.9.9. T-forces

The element defines a force and a moment, which components can be defined in two modes:

- explicit analytic expressions of time and kinematic functions;
- functions of time stored in columns of a text file (the first column contains time).

Remark. The user can introduce additional applied forces acting on bodies in the control file without describing the forces in the Input module.

2.5.10. Friction in joints depending on reactions

There exists a simple method how to model a friction in rotational and translational joints depending on reaction forces. For instance, a frictional torque in rotational joint in a sliding state can be proportional to the module of a component of reaction force perpendicular to the joint axis.

The friction joint force is used for this purpose, see Section 2.5.5.2 *Friction force*. The sliding force magnitude must be parameterized by an identifier. Using the **Identifier control** tool in the simulation module a necessary variable is assigned to this identifier.

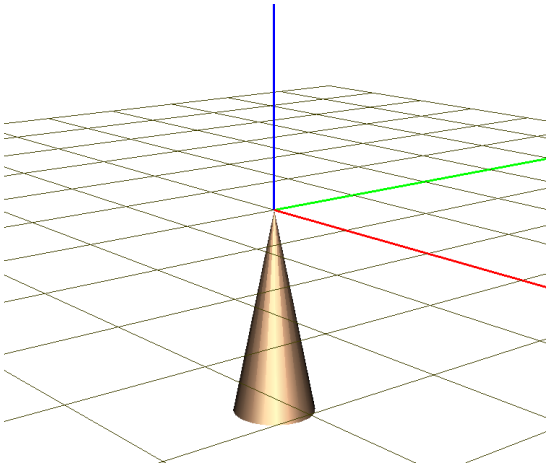


Figure 2.69. Pendulum

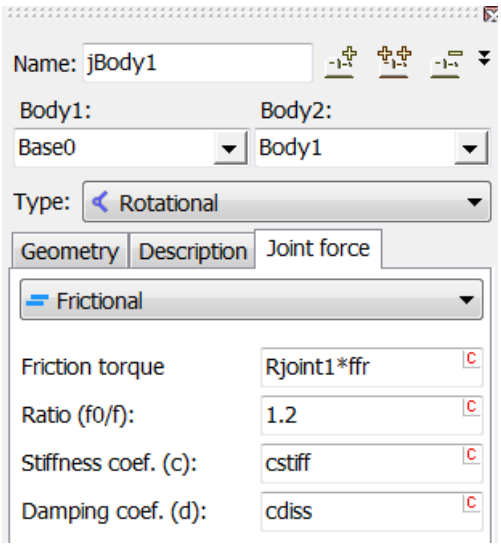


Figure 2.70. Frictional torque in joint

Example: {UM Data}\Samples\Library\Joint with friction.

The model contains a pendulum with a cone shape (Figure 2.69) connected with the base by a rotational joint with friction. A frictional torque is applied to the joint (Figure 2.70), which sliding value is set by the expression *Rjoint*1**ffr*. The identifier *ffr* corresponds to the friction coefficient and *Rjoint*1 corresponds to the module of reaction force in joint. The value of the identifier *Rjoint*1 must be assigned in the simulation module.

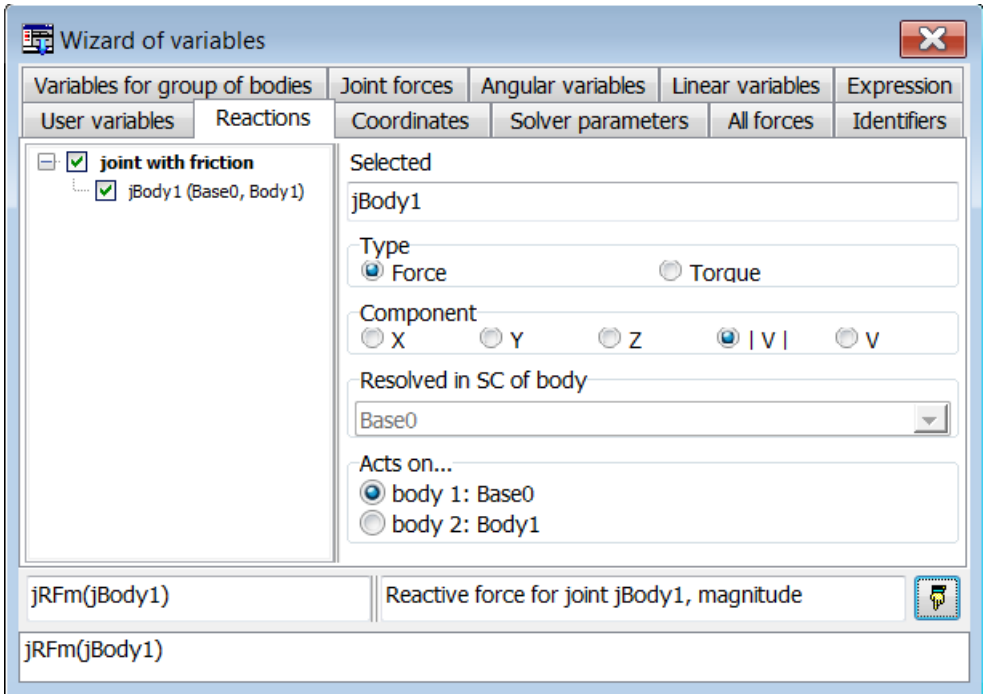


Figure 2.71. Module of reaction force

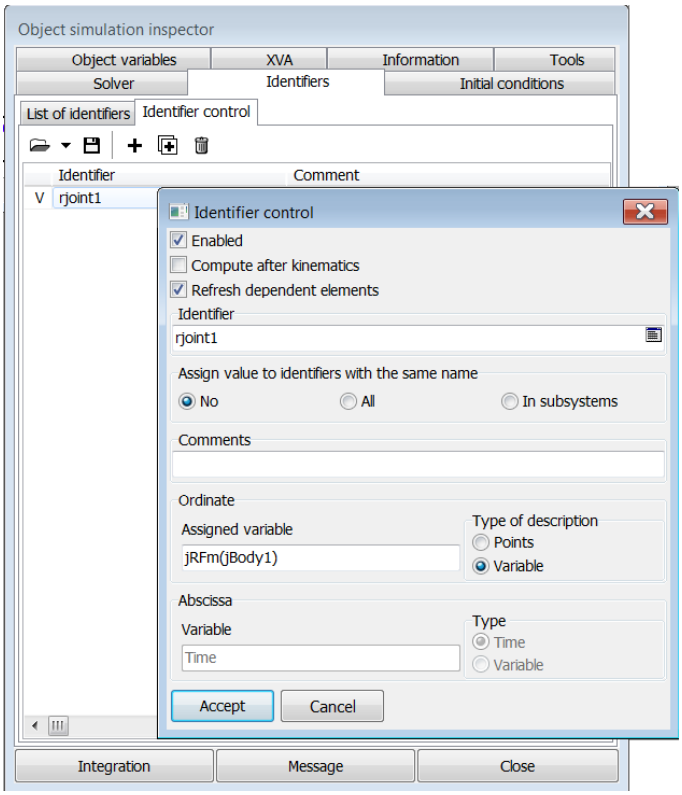


Figure 2.72. Assignment of variable to identifier

In the simulation program, a variable equal to the reaction force module is created with the wizard of variables, Figure 2.71. Then this variable is assigned to the identifier *Rjoint*1 on the **Identifier control** tab of the inspector, Figure 2.72. As a result, the friction force in sliding mode will be proportional to the module of the reaction force like in Figure 2.73.

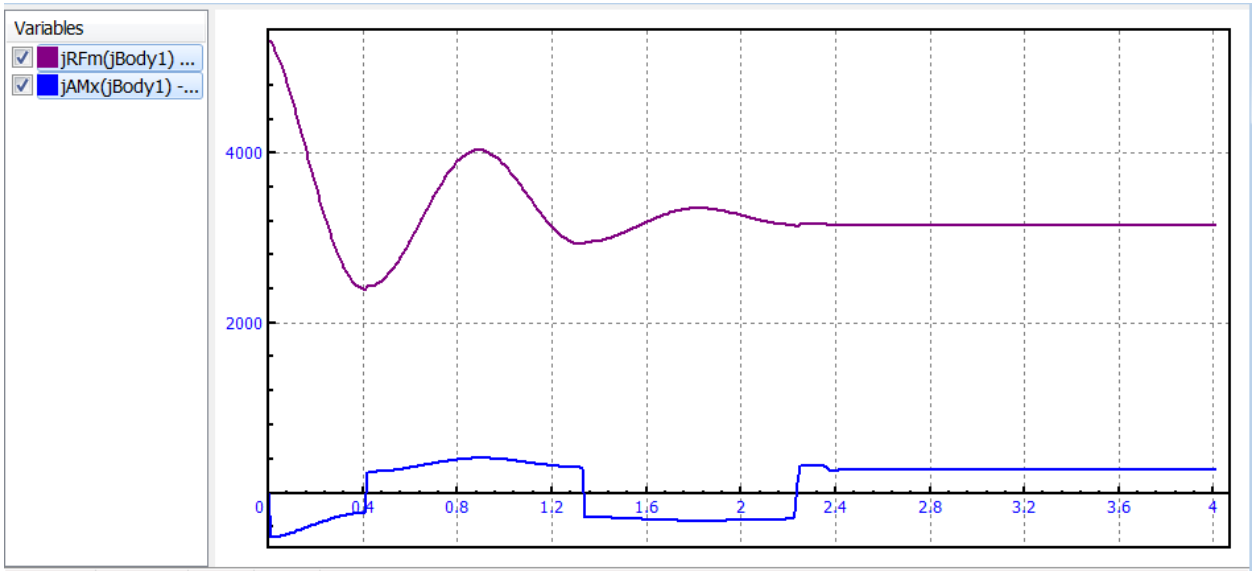


Figure 2.73. Plots: module of reaction force and friction torque

2.5.11. Force element response in the frequency domain

Force element response in the frequency domain is a useful utility for a scalar force (a bipolar force element or a joint force). Consider a scalar force element, which dependence on a scalar variable *x* and its time derivative *v* is the following (Sect. 2.5.5. *"Types of scalar forces"*, p. 2-37):

$$F = F(x, v).$$

Consider a harmonic excitation of the force

$$x(t) = x_0 + asin(2\pi ft)$$
$$v(t) = 2\pi fasin(2\pi ft),$$

where x_0, a, f are the center of excitation, its amplitude and frequency in Hz, *t* is the time variable. The response is the periodic function

$$F(t) = F\big(x(t), v(t)\big).$$

Expand this function into Fourier series keeping zero and first order terms

$$F(t) \approx F_0 + F_K(f)sin(2\pi t) - F_C(f)cos(2\pi ft) =$$
$$= F_0 + F_1(f)sin\big(2\pi ft - \delta(f)\big).$$

Main characteristics of the force in the frequency domain are introduced according to this expansion:

- Dynamic stiffness $K(f) = F_K(f)/a$,
- Equivalent damping $C(f) = F_C(f)/(2\pi fa)$,
- Phase or damping angle $\delta(f)$,
- Amplitude of the response $F_1(f)$,
- Dependence of the response function on the coordinate $F\big(x, v(x)\big)$ and ty $F\big(x, v(v)\big)$ for a fixed excitation frequency.

For description of a tool for force element analysis in the frequency domain see Chapter 4, Sect. *Force element response in the frequency domain.*

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

2.6. Methodology of choice of contact parameters

Contact stiffness c and damping μ parameters are used in description of a number of force elements (Sect. 2.5.7. *"Contact forces"*, p. 2-59, Sect. 2.5.5.2. *"Friction force"*, p. 2-37, Sect. 2.5.5.3. *"Elastic-frictional force"*, p. 2-38, Sect. 2.5.9.5. *"Cam"*, p. 2-84). Let us consider some methods for estimation their numeric values.

The real contact stiffness due to elastic deformations in contacts is usually very high (say, 10^{11}N/m). It is clear, that for reasonable values of contacting body masses such stiffness introduces a very high frequency in the model (about $\sqrt{c/m}$, where c is the stiffness, and a m is the mass). This makes the model oscillatory stiff and increases CPU expenses considerably due to decreasing the integration step size. In practice much less values the stiffness can be set in the model. Consider the main reasons for that.

Applied theory of ordinary differential equations states (a strict proof can be obtained with the help of theory of singular degenerate equations) that if a system has two groups of frequencies of different order, e.g. 1–10Hz in the first group and 200–1000Hz in the second one, the high frequency processes does not affect practically the low frequency processes. Moreover, changing parameters, which determine high frequency processes, does not lead to considerable changes of the low frequency processes. That means, if the stiffness c is 'large enough', its further increase does not affect analyzed processes, if they are 'slow'. The stiffness is 'large enough' if the introduced local frequencies are at least by order of magnitude greater than the frequencies of analyzed object. The frequency k introduced by the contact stiffness may be estimated by the formula $k = \sqrt{c/m}$, where m is the lower mass of interacting bodies. Thus, if the user choose a 'large enough' local contact frequency $f = \frac{k}{2\pi}$ Hz, the corresponding contact stiffness can be computed according to the formula

$$c = 4\pi^2 f^2 m. \qquad (2.9)$$

Consider an example. Let $m = 20$ kg, $f = 200$ Hz (i.e. the main frequencies of the object are 1–10 Hz). Then $c = 3.16 \times 10^7$ N/m.

A stiffness coefficient computed according to this methodology should be verified and corrected by the user. To do this, run simulations for different values of the stiffness and plots of object performance variables should be compared. If 2÷10 times increasing the coefficient does not affect the results, the stiffness is 'large enough'. In this case it is recommended to try decreasing its value. This may reduce CPU expenses. The boundary of the parameter is its value when the plots of performances are changed.

Let us discuss now methodology for choosing a dissipation coefficient μ. If this coefficient is 'too small', high frequency undamped oscillations may appear in the model, which introduce large accelerations. Moreover, frequencies of these oscillations according to the methodology above have nothing in common with the reality. If the damping is 'too large', the equations of motion become stiff, and CPU expenses increase. A correct choice of the damping coefficient is especially important for systems with unilateral constraints, i.e. for systems with gaps and im-

pacts, because the damping is responsible for a value of coefficient of restitution. Finally, the value of the damping must correlate with the value of the contact stiffness.

To get a justified value of the contact damping, a very important notion of damping ratio of critical β and a damping factor β^* is used. To clarify this notion consider the equation of free linear damped oscillations

$$m\ddot{x} + \mu\dot{x} + cx = 0. \tag{2.10}$$

As it is well known, the solution of this equation depends on ratio of two parameters: the frequency of free undamped oscillations $k = \sqrt{c/m}$ and the damping coefficient $n = \mu/(2m)$. If $n < k$, the motion is damped oscillations with the frequency $k^* = \sqrt{k^2 - n^2}$. If $n \geq k$, an aperiodic solution takes place. The boundary value $n = k$ corresponds to a critical damping. The damping ratio and the damping factor satisfy the following relations

$$\beta = \frac{n}{k}, \tag{2.11}$$

$$\beta^* = \frac{n}{k^*} = \frac{n}{\sqrt{k^2 - n^2}} \in [0, \infty],$$

Oscillations are undamped if $\beta = \beta^* = 0$, and by $\beta = 1$ ($\beta^* = \infty$) the damping is critical. The parameters can be expressed in terms of each other,

$$\beta^* = \frac{\beta}{\sqrt{1 - \beta^2}} > \beta, \beta = \frac{\beta^*}{\sqrt{1 + \beta^{*2}}}$$

If $\beta < 1$, a decrement factor is computed as

$$D = e^{-\pi\beta^*} = e^{-\frac{\pi\beta}{\sqrt{1-\beta^2}}}.$$

The decrement factor determines an amplitude fall on a half of the period of oscillations. If the damping is small, $\beta \approx \beta^*$ and

$$D \approx e^{-\pi\beta}.$$

Values of the decrement factor for different degrees of damping are presented in Table1.

Table 1

Decrement factor versus damping ratio

β	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
D	1	0.729	0.527	0.372	0.254	0.163	0.095	0.046	0.015	0.0015	0

Figure 2.74 shows solutions for different β.

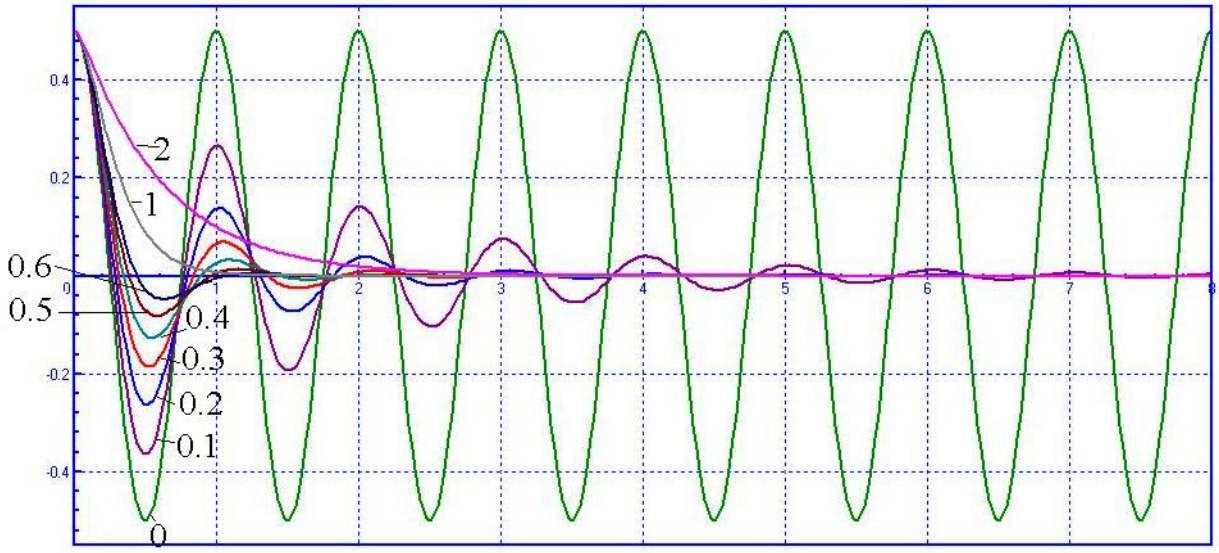


Figure 2.74. Damped vibrations for various damping ratios

If $\beta \leq 0.1$, the damping is often considered as a small one. The value $\beta = 0.1$ can be recommended for an approximate in rubber-metal elements. The value $\beta = 0.3$ is recommended for dampers of transport vehicles. If $\beta \geq 1$, the system is strongly overdamped.

The above analysis clarifies the notions 'too small' or 'too large' damping coefficient. We recommend the damping $\beta = 0{,}1 \div 0{,}4$ for the contact forces. Thus, if you choose some values for stiffness and damping ratio, the coefficient of damping is computed according to the following formula

$$\mu = 2\beta\sqrt{mc} \tag{2.12}$$

For the above example $c = 3.16 \times 10^7$ N/m with the damping ratio $\beta = 0{,}2$ we get $\mu \approx 10^4$ Ns/m.

The next important problem is, how collisions may be likely modeled? If we answer this question, we could find a more proved value of damping ratio β in cases of unilateral contacts.

Is it is known, one of the most simple and frequently used model of a collision of bodies is based on the Newton's coefficient of restitution *e*, which can be illustrated by the following example. Consider a mass point, which strikes against a fixed plane. The point velocity before the collision be v^-, and after the collision – v^+. The coefficient of restitution is

$$e = \frac{v^+}{v^-}.$$

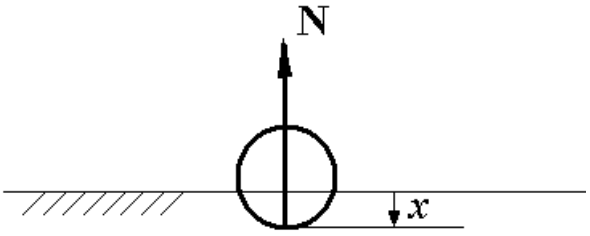


Figure 2.75. Collision process

Consider the model of viscoelastic interaction of the point with the plane (Sect. 2.5.7.1. *"General information about contact interactions"*, p. 2-59). The coordinate x in Figure 2.75 is the penetration, which determines the normal contact force

$$N_x = -cx - \mu\dot{x}.$$

During the process of the contact the point motion is described by the differential equation (2.10) with the following initial conditions: $x(0) = 0, v(0) = v^-$. At the compression stage of the collision the normal force increases, and at the relaxation phase it decreases. At some moment τ the normal reaction vanishes and the impact process is over. This, $v(\tau) = v^+$.

To get the velocity after the impact, consider the solution of Eq. (2.10)

$$x = \frac{v^-}{k^*} e^{-nt} sin(k^* t),$$
$$v = v^- e^{-nt}\left(-\frac{n}{k^*} sin(k^* t) + cos(k^* t)\right).$$

At the end of the collision we have

$$0 = N(\tau) = \mu v(\tau) + cx(t) = m\big(2nv(\tau) + k^2 x(t)\big)$$
$$= v^- e^{-nt} mk^* \left(2\frac{n}{k^*} sin(k^*\tau) + \left(1 - \frac{n^2}{k^{*2}}\right) cos(k^*\tau)\right)$$

so the collision duration τ satisfies the equation

$$tg(k^*\tau) = \frac{2\beta^*}{\beta^{*2} - 1}$$

or

$$\tau = \begin{cases} \pi + \frac{1}{k^*} arctan\left(\frac{2\beta^*}{\beta^{*2} - 1}\right), \beta^* < 1 \\ \frac{1}{k^*} arctan\left(\frac{2\beta^*}{\beta^{*2} - 1}\right), \beta^* \geq 1 \end{cases}$$

The result can be simplified is we find that

$$sin(k^*\tau) = \frac{2\beta^2}{\beta^*}$$

or finally

$$\tau = \frac{1}{k^*} arcsin\left(\frac{2\beta^2}{\beta^*}\right)$$

Now the velocity after the collision is:

$$v^+ = |v(\tau)| = \frac{k^2}{2n} x(\tau) = \frac{v^- k^2}{2nk^*} e^{-n\tau} sin(k^*\tau) = v^- e^{-\beta^*\varphi^+}.$$

Here a *lifting off phase* $\varphi^+ = k^*\tau = \arcsin(\frac{2\beta^2}{\beta^*})$ is introduced. This phase depends of the ratio β (or β^*) only.

So, we have obtained the following fine formula for the coefficient of restitution:

$$e = e^{-\beta^* \varphi^+}$$

Illustration of dependence of the coefficient of restitution on the damping ratio is given in Table 2 and Figure 2.76.

Table 2

Coefficient of restitution versus damping ratio

β	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
e	1	0.859	0.744	0.65	0.572	0.506	0.451	0.404	0.364	0.329	0.298
β	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1	
e	0.272	0.249	0.228	0.21	0.194	0.18	0.167	0.155	0.145	0.135	

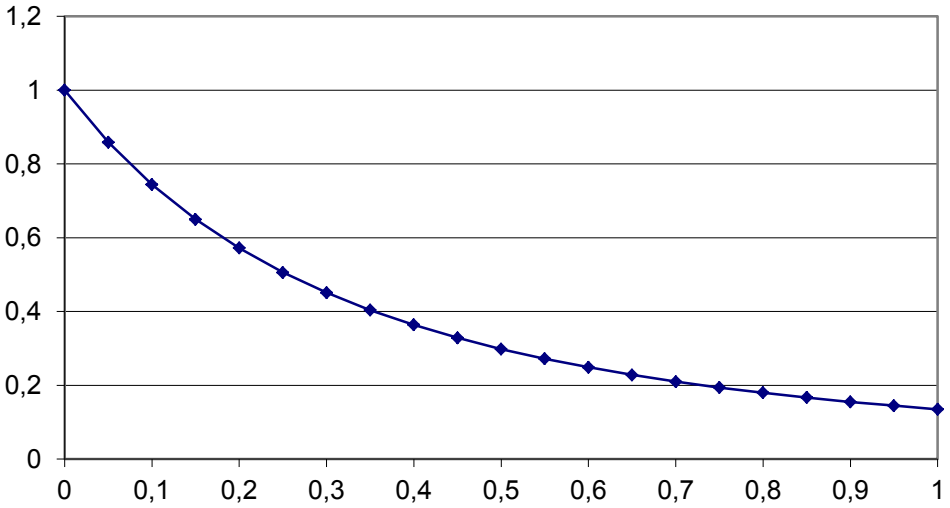


Figure 2.76. Coefficient of restitution versus damping ratio

If the user knows the coefficient of restitution, he can estimate the damping ration β and, respectively, the damping coefficient μ.

2.7. Subsystems

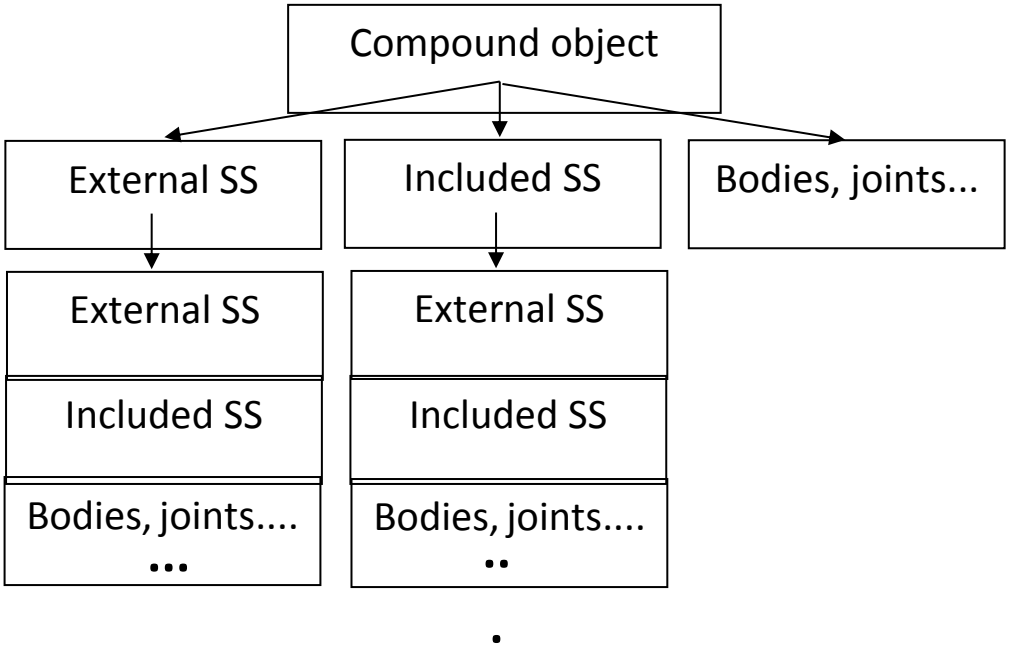
2.7.1. Subsystem technique

If a mechanical system containing hundreds or thousands of bodies is modeled, the equations of motion are so huge that problems with their compilation might arise. For such a system *the subsystem technique* is convenient. In this case, the MBS is preliminarily (conditionally) divided into a few parts – subsystems – and is called *a compound object*. Separate subsystems may or may not be connected with each other by joints or force elements.

The subsystem technique is especially effective when several subsystems are *kinematically identical* or, in other words, have the same kinematical scheme. Here is a rigorous definition of *kinematic identity*.

Two subsystems are kinematically identical if, their configuration being arbitrary, it is possible to bring them into spatial coincidence in the basic SC by means of the following transformations:

- translation (the vector is constant);
- rotation about any axis by a constant angle;
- changing of the parameters of the kinematic pairs (e.g., changing the lengths of the members);
- changing the values of the local joint coordinates.



For the equations of motion of the kinematically identical subsystems to be the same the following conditions should be satisfied: all the constant parameters (masses, lengths of the structural elements, etc.), which may vary numerically must be represented in *a symbolic form* (i.e. by identifiers). While modeling motion, different values may be attributed to the parameters of different subsystems.

UM model of an object can be structured as a tree of *external* and *included* subsystems.

An **External subsystem** is generated by any preliminarily created UM object – ancestor. All modifications made in description of the ancestor are automatically taken by all external subsystems – descendants. Moreover, modification in an external subsystem can be made exclusively through modification of the corresponding ancestor. A compound object, which contains external subsystems, does own neither structure, nor parameters of external subsystems, but refers to them. At the same time, the compound object can add joints and force elements connecting bodies from different subsystems. Thus, in the case of kinematically identical subsystems, the user can create an ancestor, generate the equation library for it (dll), and add the ancestor to the object as many times as it is necessary.

Included subsystems, unlike the external ones, belong to the compound object. The object owns their structures (bodies, joints, forces, etc.) and parameters. Several included subsystems

generated by one ancestor can be arbitrary modified. Equations of motion for included subsystems are generated as a part of the object equations.

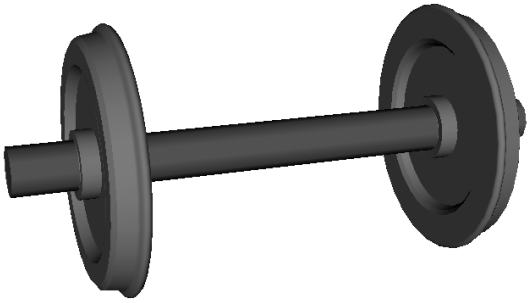
Subsystems have a tree structure. An object can contain any number of subsystems (both external and included). Each of the subsystem, in its turn, may be a compound object containing subsystems.

The subsystem technique is the foundation for development data basis for modeling different technical systems (such as rail vehicles).

2.7.2. Standard subsystems

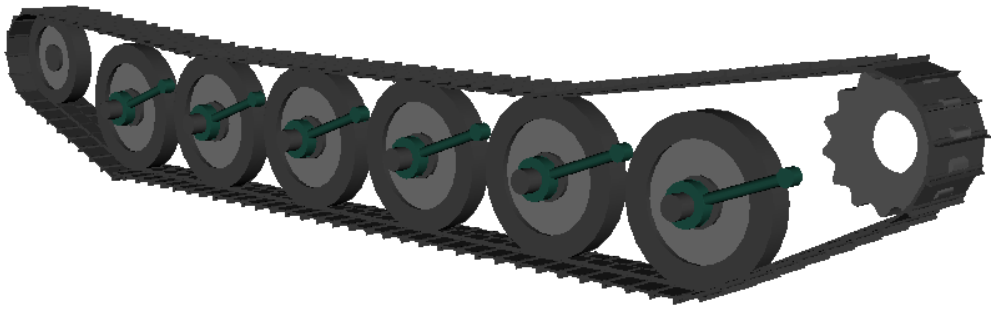
Standard subsystems are automatically generated subsystems, which are basis for creation of models of technical systems taking their features. Standard subsystems are used in the main um modules UM: simulation of rail vehicles, automotive vehicles, caterpillar models, and ballast.

2.7.2.1. Wheelset as standard subsystem



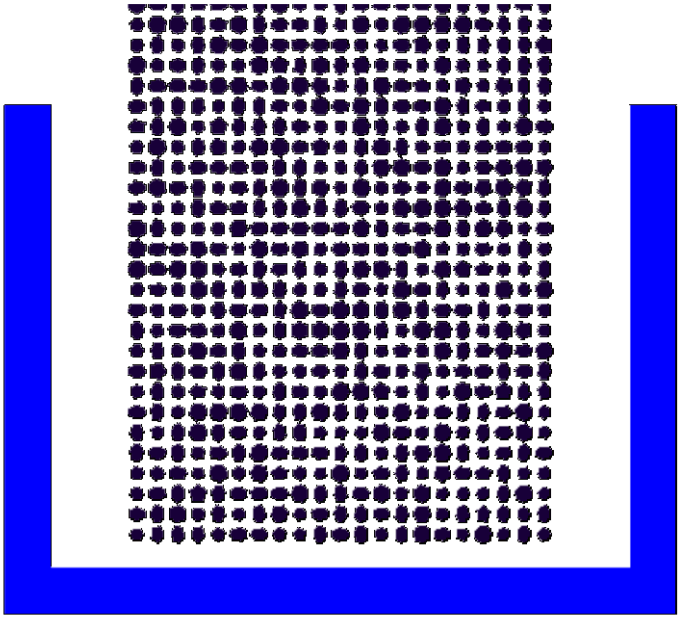
A standard subsystem *wheelset* is the base of the UM Loco module for simulation of railway vehicles. Simultaneously to an automatic adding wheelsets to vehicle models, UM computes rail-wheel contact forces as well as special interface abilities are available with this module.

2.7.2.2. Caterpillar



The UM/Caterpillar module is based on a standard subsystem “caterpillar”, which allows the user to get a model with any number of tracks, supporting wheels, etc.

2.7.2.3. Ballast



Generated subsystem ‘ballast’. Simulation of the ballast backfilling process

The ‘ballast’ subsystem is developed for analysis of dynamic properties of broken stone ballast. The UM ballast model can include thousands of bodies of different shape.

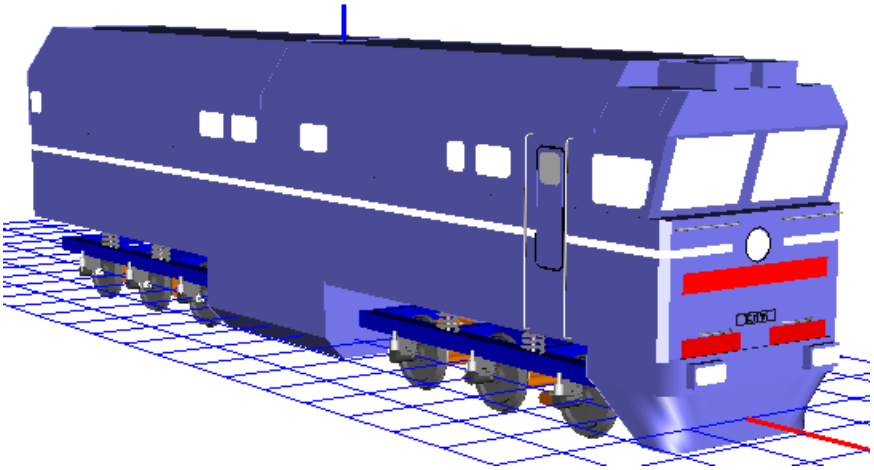
2.7.3. Examples of compound objects

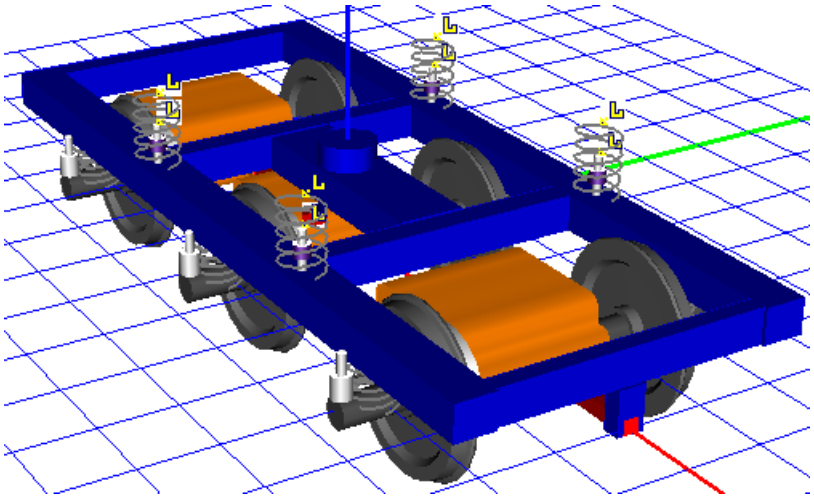
2.7.3.1. Dynamically independent subsystems

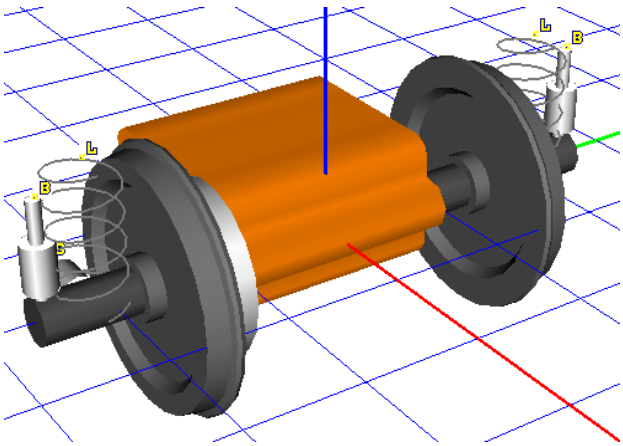
If it is necessary to compare in one time scale the behaviors of the same object in different situations or various objects, the method of subsystems is also useful. In this case, the compound object containing several subsystems generated by one or many UM objects is formed. For instance, two Puma robots or one Puma robot and a Stanford manipulator can be dealt with as compound objects. Here, the subsystems are not dynamically connected. However, the integration of equations of motion is done for both subsystems simultaneously, and the results are displayed out in the same time scale.

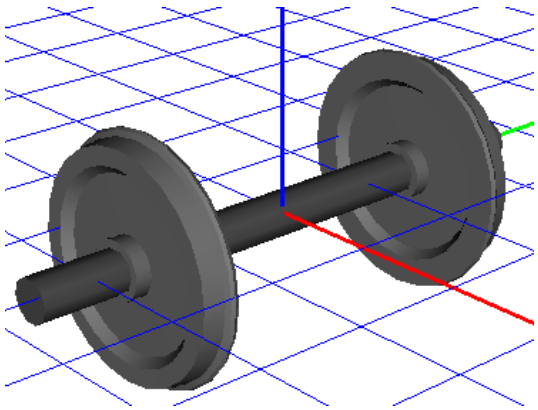
The number of dynamically independent subsystems in a compound object is limited only by the resources of the user’s PC.

2.7.3.2. Locomotive as a compound object



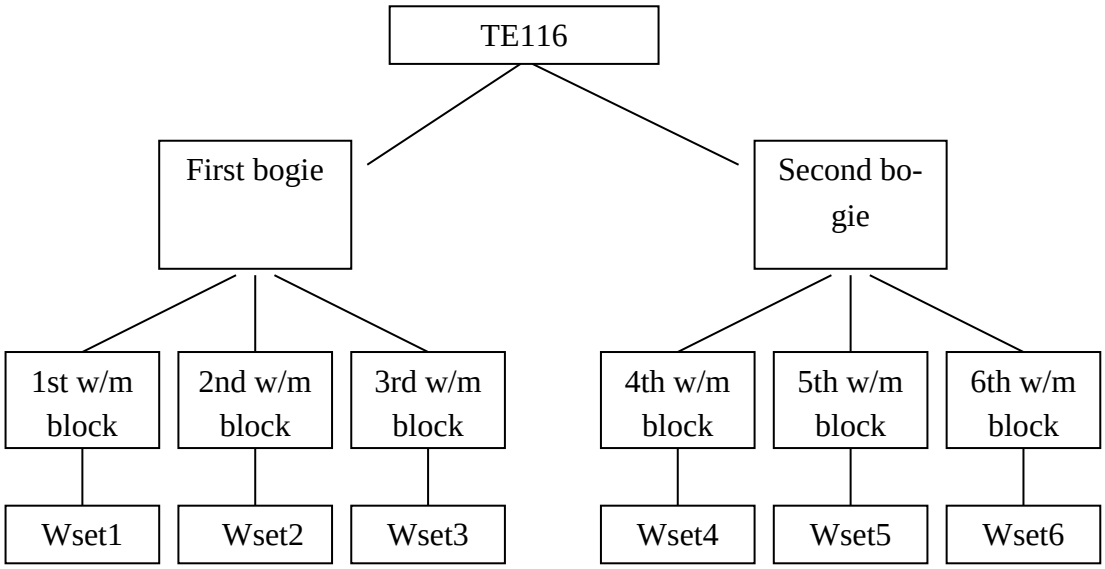






The model of the TE116 locomotive as a compound object consists of a body and two included subsystems – bogies. A bogie contains three included subsystems – wheel-motor blocks, one body (a bogie frame), and elements of the secondary suspension (springs and dampers). A wheel-motor block consists of one included subsystem (a wheelset is the standard UM subsystem), a motor, elements of the primary suspension (springs, dampers, propulsion rods), gearing.

The subsystem tree has four levels.



The model description has three stages.

Stage 1. Creation of an object-ancestor: wheel-motor block, which contains a wheelset as an included subsystem.

Stage 2. Creation of an object-ancestor: a bogie, which includes three *wheel-motor block* subsystems.

Stage 3. Creation of the locomotive object, which includes two *bogie* subsystems.

Each of the included subsystems is available for separate modification.

2.8. Linearization of equations and equilibrium positions

2.8.1. Equilibrium equations and their solving

Consider equations of motion for a multibody system with stationary constraints. If the system has closed loops, its equations of motion are

$$M(q)\ddot{q} + k(q,\dot{q}) = Q(q,\dot{q}) + G^T(q)\lambda,$$
$$g(q) = 0$$

At equilibrium by $\dot{q} = 0, \ddot{q} = 0$, the following equations take place:

$$Q(q,0) + G(q)^T\lambda = 0, \quad g(q) = 0 \tag{2.13}$$

Here we have nonlinear algebraic equations relative to unknown values of coordinates q=q0 and Lagrange multipliers $\lambda = \lambda_0$. The Newton-Raphson method is used for solving the equations. The following linear equations are solved at each the iterations:

$$\left(Q_q\left(q_0^k,0\right) + \Phi\left(q_0^k,\lambda_0^k\right)\right)\Delta q_0^{k+1} + G\left(q_0^k\right)^T\lambda_0^{k+1} = 0, \qquad G\left(q_0^k\right)\Delta q_0^{k+1} = -g\left(q_0^k\right) = 0,$$
$$q_0^{k+1} = q_0^k + R\Delta q_0^{k+1}$$

Here Q_q is the Jacobian matrix $Q_q = \partial Q/\partial q^T$, the matrix Φ is defined by the relation

$$\Phi = \frac{\partial(G^T\lambda)}{\partial q^T},$$

$R \le 1$ is the relaxation factor, which is usually equal to 1.

The iterations are considered as successfully converged if the discrepancies in Eqs. (2.13) satisfy the condition

$$\sum|\delta_i| < \varepsilon,$$

with an error tolerance ε.

User's initials $q_0^0 = q^0$ are used as a starting approximation. The Jacobian matrices $Q_q\left(q_0^k,0\right), \Phi\left(q_0^k,\lambda_0^k\right)$ are evaluated by finite differences as

$$Q_{qi}\left(q_0^k,0\right) = \left(Q\left(q_0^k + \delta q_i,0\right)^T - Q\left(q_0^k,0\right)^T\right)/\delta,$$
$$\Phi_i\left(q_0^k,\lambda_0^k\right) = \left(G\left(q_0^k + \delta q_i\right)^T - G\left(q_0^k\right)^T\right)\lambda_0^k/\delta.$$

Here Q_{qi}, Φ_i are the columns i of the corresponding matrices, δ is a small real number; the column δq_i has one nonzero value at position i, which equals to δ.

2.8.2. Natural frequencies, modes, eigenvalues and eigenvectors

QR algorithm is used for computing both natural frequencies and eigenvalues. Reverse iterations are applied to get modes and eigenforms of linearized equations.

2.9. Units of measure

All the calculations in UM are carried out in the International System of Units: kg, m, s. While solving dynamic problems using other units is not recommended (cm, kN, etc.). In the data input unit angles are usually introduced in degrees (when describing initial conditions and limits), whereas when modeling motion their numerical values are obtained in radians.

2.10. Generation and analysis of equations of motion

Equations of motion can be generated in two forms.

1. **Symbolic generation of equations**

Kinematical expressions and the motion equation generation of an object are done in a symbolic form with the help of a built-in specialized computer-algebra system. To decrease the number of operations more significantly fracturing and substituting procedures are used.

2. **Numeric-iterative generation of equations**

Often symbolic generation of equations makes the simulation process faster.

The equations of motion of a system are generated according to the Newton-Euler formalism and are differential-algebraic. The equations analysis is carried out by means of the ABM, BDF numerical multistep methods with the automatic choice of the step size and the order of the method, as well as the Park and Gear methods for stiff equations.

The Park Parallel solver is the powerful numeric-iterative solver, which allows usage of multithreading on the multi-core processors.

While integrating the equations computing kinematical characteristics and constraint reaction forces is possible.

The Park and Gear methods with computing of Jacobian matrices are recommended for solving stiff differential-algebraic equations of motion.

The obtaining of equilibrium positions of non-linear objects and the linearization of the equations of motion in the neighborhood of the equilibrium position can also be performed. For linear systems, there are standard analysis procedures: the obtaining of natural frequencies and vibration modes, root locus and so forth.

See Chapter 4 for more information on the model simulation process.

2.11. The innovative capacity of the program and programming in UM environment

To raise its universality, UM has been left *open* for changes in anything concerning motion modeling. The user may include his modules and influence the process of modeling by means of the program messages. Programming in the environment of UM assumes using some internal procedures and data types, described in the current manual.

Developing special modules to be linked to the program makes it possible to take into account the features of various technical system types.

References

[1] Kraus P.R., Fredriksson A., Kumar V.S Modeling of frictional contacts for dynamic simulation / Proc. of IROS 1997 Workshop on Dynamic Simulation: Methods and Applications, Sept. 1997.

[2] Stribeck R., 1902. “Die wesentlichen eigenschaften der gleit- und rollenlager”. Zeitschrift.

[3] Bo L.C., Pavelescu D., 1982. “The friction–speed relation and its influence on the critical velocity of the stick–slip motion”. Wear, 82(3), pp. 277–289.

[4] Armstrong-H´elouvry B., 1991. Control of Machines with Friction. Kluwer Academic.

[5] Cyrus M., Beck J. Generalized Two- and Three-Dimensional Clipping. Computers & Graphics, Vol. 3, pp. 23-28, 1978.