Ansys
2025 R2

POWERING INNOVATION THAT DRIVES HUMAN ADVANCEMENT

Twin Builder® Components: Mechanical System

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
https://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 2025 R2
July 2025

Copyright and Trademark Information

© 1986-2025 ANSYS, Inc. Unauthorized use, distribution or duplication is prohibited.

ANSYS, Ansys Workbench, AUTODYN, CFX, FLUENT and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries located in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. All other brand, product, service and feature names or trademarks are the property of their respective owners. FLEXlm and FLEXnet are trademarks of Flexera Software LLC.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001: 2015 companies.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the legal information in the product help files for the complete Legal Notice for Ansys proprietary software and third-party software. If you are unable to access the Legal Notice, please contact ANSYS, Inc.

Table of Contents

1 - Mechanical System Library

The one-dimensional Mechanical System simulation module extends Twin Builder's strength's in simulating electric circuits and related fields toward complex drive systems.

The Mechanical System library pursues the multi-body approach for computing motion quantities along one motion coordinate (dimension). Some nonlinear properties include rigidity, friction, backlash, and stick.

The Mechanical System library is mainly created for rotational drive systems, but the also provides the corresponding component in the translational representation. Therefore, purely translational system can be modeled, as long as only one motion coordinate is being computed.

The Mechanical System Library consists of the following component groups:

- Engine
- Rotational_V
- Translational_V
- Transmission

Introduction

A mechanical system is decomposed into a finite number of rigid bodies connected by mass-less rigidities. The library provides two fundamental components:

- MAS - rigid mass
- STF - mass-less rigidity (spring), of which complex systems (gears, couplings, shafts, drive trains) can be designed

Principle

The mechanical power train is decomposed into concentrated equivalent bodies of finite inertia and infinite rigidity connected by concentrated, mass-less springs of finite rigidity. Each mass can be linked to an arbitrary number of springs (or none). Each rigidity usually joins two inertias but can also be connected to no body (idle shaft) or to mechanical ground (earth potential). Friction and damping parameters can be specified with both components. The system can be exerted on by arbitrary external forces/torques. This allows you to model complex structures of straight, branched, or nested mechanical power trains by predecessor-successor relations with nonlinear properties.

System Decomposition and Multi-Body Approach

The drive power train consists of a motor rotor, a motor shaft, coupling with soft rigidity, and a load driven by a load shaft.

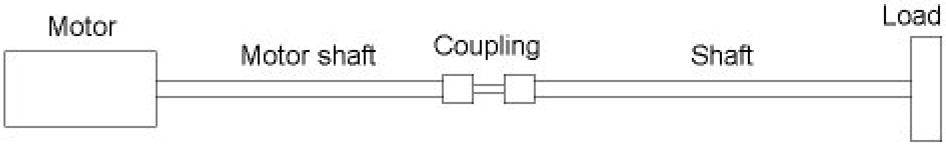


Figure 1. The drive power train system

Each material has the following properties:

- Inertia with corresponding mass parameter M or J.
- Rigidity with corresponding spring parameters C.
- Friction/damping with corresponding friction parameters K.

The drive power train system in Figure 1 consists of an infinite number of mass elements connected by an infinite number of rigidities, both with friction/damping properties. To reduce the number of elements, the multi-body approach decomposes the system into a finite (and possibly small) number of masses (bodies) connected by a finite (and possibly small) number of rigidities.

Mass concentrates in the following sections of the mechanical power drive train: the rotor, both coupling halves, and the load mass. These sections are linked by the following other sections where rigidity dominates: the motor shaft, the intrinsic coupling rigidity, and the load shaft.

Although different approaches are possible, it is preferable to assign a body to the location where inertia concentrates and to couple these masses by springs in the location where rigidity dominates. Consequently, decomposing the mechanical power drive train yields to a four-body system consisting of four masses connected by three springs.

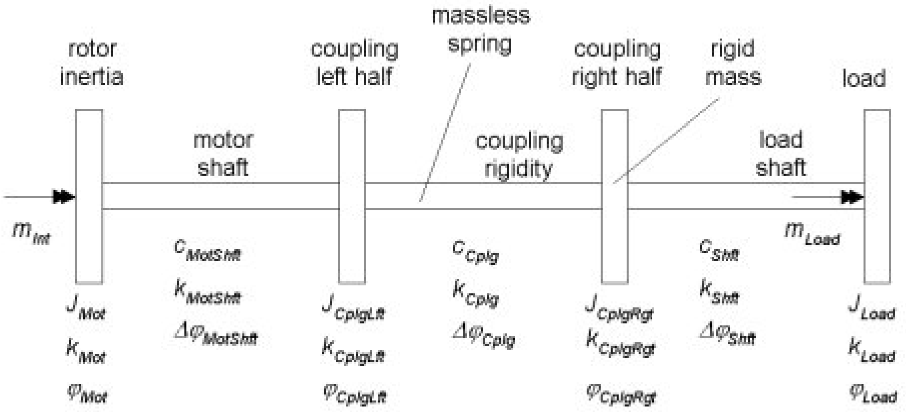


Figure 2. The four-body system of the drive power train

Each body incorporates the concentrated inertia properties of the corresponding section. For rotational systems, the moment of inertia J is used. Each spring retains the concentrated rigidity properties of its section with parameter C.

Friction/damping can occur in two different ways: external or internal. External friction describes the mechanical energy loss at a body (friction with the environment). Internal friction in a spring section is caused by particle friction inside the material.

Each body can be exerted on by external forces, in this case, a magnetic motor torque and a possible load torque function. The motion quantity is always the computed one. This is the position and all its derivatives.

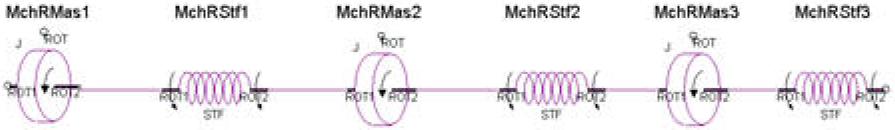


Figure 3. The rotational four-body system model in Twin Builder

The Mechanical System library also offers components for translational systems. The next example shows the converted system of the rotational four-body system.

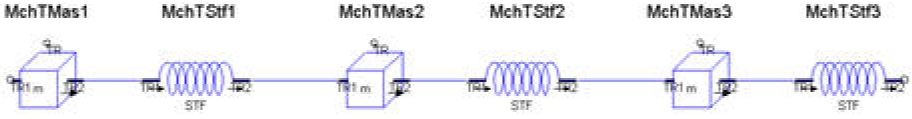


Figure 4. The translational four-body system model in Twin Builder

The above examples doesn’t include branches in the power train. However, the Mechanical System library allows the modeling of branched and nested mechanical power trains (e.g., gearboxes).

Mathematical Background

Motion Equation

Because the mathematical background is identical for both translational and rotational systems, more general coordinates and parameters are used in this section.

The motion coordinate (position quantity) is always q(t). MECHSIM computes all its derivatives as the following:

$$\frac{d^n q(t)}{dt^n} \qquad n = 0 \ldots 3$$

Motion quantities

Order (n)	Motor Quantity ($q^{(n)}$) [unit]	
	Rotational Systems	Translational System
0	Angular position [rad]	Position [m]
1	Angular velocity [rad/s]	Velocity [m/s]
2	Angular acceleration [rad/s^2]	Acceleration [m/s^2]
3	Angular jerk [rad/s^3]	Jerk [m/s^3]

Parameters

Parameter	Rotational Systems			Translational Systems		
	Name	Symbol	Unit	Name	Symbol	Unit
Inertia	Moment of inertia	J	[kgm^2]	Mass	M	[kg]
Rigidity	Spring constant	C	[Nm/rad]	Spring constant	C	[N/m]
Friction	Friction coefficient	K	[Nms/rad]	Friction coefficient	K	[Ns/m]
Excitation forces	Torque	M	[Nm]	Force	F	[N]

The following example shows a simplified straight three-body system in general coordinates. A coordinate transformation can be inserted between any two sections.

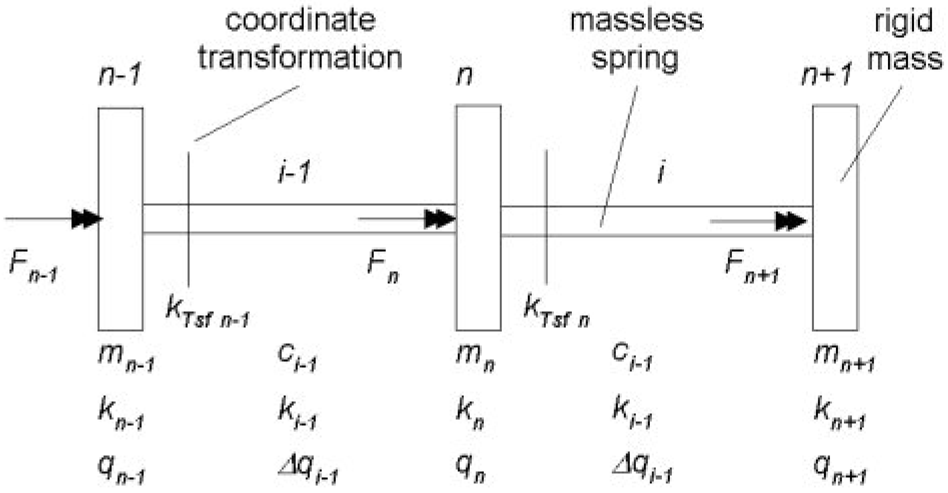


Figure 5. A three-body system in general coordinates

According to the energy balance, all forces/torques acting on body n must be zero (n = 1...N, i = 1...I). This yields the following motion equation:

$$\mathbf{m}_n \ddot{\mathbf{q}}_n + \mathbf{k}_n \dot{\mathbf{q}}_n - \mathbf{k}_{i-1} \Delta \dot{\mathbf{q}}_{i-1} - \mathbf{c}_{i-1} \Delta \mathbf{q}_{i-1} + \mathbf{k}_i \Delta \dot{\mathbf{q}}_i + \mathbf{c}_i \Delta \mathbf{q}_i = \mathbf{F}_n$$

The position difference is calculated as follows:

$$\Delta \mathbf{q}_i = \mathbf{q}_n - \mathbf{q}_{n+1}$$

The motion equation does not consider branches and coordinate transformations and does not represent the way the Mechanical System library is implemented. It is merely used to assist in the understanding of the mechanical principles.

Deriving the motion equation for each body in the system yields .a differential equation system consisting of n equations. This system can be written in matrix form as follows:

$$\mathbf{M}\ddot{\mathbf{q}}(\mathbf{t}) + \mathbf{K}\dot{\mathbf{q}}(\mathbf{t}) + \mathbf{C}\mathbf{q}(\mathbf{t}) = \mathbf{F}$$

where:

- M is the mass matrix, K the damping matrix, and C the nxn rigidity matrix
- $q^{(n)}(t)$ and F are the n x 1 motion coordinate and excitation force vectors.

Eigenvalues

System time constants are important for estimating simulation time steps. The time constants can be obtained by retrieving the eigenvalues of the system. Most rotational drive systems have rather low damping degrees. Therefore, it is sufficient to examine the frequency behavior for undamped systems.

Transforming the differential equation system into frequency domain and examining the natural frequencies of the undamped system by calculating the eigenvalues as following:

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{C}\mathbf{q}(t) = 0 \qquad \mathbf{M}\mathbf{q}(s)s^2 + \mathbf{C}\mathbf{q}(s) = 0$$

$$\lambda = -s^2 = \omega_0^{\,2}$$

$$\mathbf{M}^{-1}\mathbf{C} - (\mathbf{I} \cdot \lambda) \cdot \mathbf{q} = 0 \qquad \mathbf{A} - (\mathbf{I} \cdot \lambda) \cdot \mathbf{q} = 0$$

$$(\mathbf{A} - \mathbf{I} \cdot \lambda_m) \cdot \mathbf{q}_m = 0$$

$$\mathbf{A} = \mathbf{M}^{-1}\mathbf{C}$$

for straight systems,

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & 0 & \dots & 0 \\ 0 & m_2 & 0 & \dots & 0 \\ 0 & 0 & m_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & m_N \end{bmatrix}, \mathbf{C} = \begin{bmatrix} c_1 & -c_1 & 0 & \dots & 0 \\ -c_1 & -c_1 + c_2 & -c_2 & \dots & 0 \\ 0 & -c_2 & c_2 + c_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & -c_{I-1} \\ 0 & 0 & 0 & 0 & c_I \end{bmatrix}, \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dots \\ q_N \end{bmatrix}$$

The result is the diagonal matrix **λ** and the matrix q; the elements the matrix **λ** are the eigenvalues λ_m (they are the squares of the natural frequencies ω_{0m}), and the rows of the matrix q are the eigenvectors q_m, shown as following:

$$\lambda_1 = \omega_1^2 \quad \lambda_2 = \omega_2^2 \quad \lambda_3 = \omega_3^2 \quad \dots \quad \lambda_M = \omega_M^2$$

$$\mathbf{q} = \left(\begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ \dots \\ q_{N1} \end{bmatrix} \begin{bmatrix} q_{12} \\ q_{22} \\ q_{32} \\ \dots \\ q_{N2} \end{bmatrix} \begin{bmatrix} q_{13} \\ q_{23} \\ q_{33} \\ \dots \\ q_{N3} \end{bmatrix} \begin{bmatrix} \dots \\ \dots \\ \dots \\ \dots \\ \dots \end{bmatrix} \begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ \dots \\ q_{NM} \end{bmatrix} \right) \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \dots \\ q_N \end{bmatrix}$$

The eigenvectors q_m measure the relative displacements of the body n ($n \in [1, N]$) at frequency ω_m. The kinetic energy stored in the bodies at the corresponding eigen frequency can be computed as the following:

$$W_{kin} = \frac{1}{2} \cdot \mathbf{M} \cdot \mathbf{q}^2 \cdot \omega_0^2 \qquad \omega_0^2 = \lambda$$

This equation measures the amount of energy body n can feed into the system at frequency ω_m. Furthermore, beyond the displacement of the bodies, the excitability of the components is determined by the rigidity sections that transfer the largest amount of energy. This is equal to the energy that section i has to store at frequency ω_m. The potential energy stored in a rigidity is computed as the following:

$$W_{pot} = \frac{1}{2} \cdot \mathbf{C} \cdot \Delta\mathbf{q}^2 \qquad \Delta\mathbf{q}_i = \mathbf{q}_n - \mathbf{q}_{n+1}$$

This information can be used to assign the natural frequencies ω_m to the rigidity sections i. The most excitable section at the mth eigen frequency is the one with the largest potential energy. From that, time constants can be computed. In addition, decision can be made on system reductions or extensions regarding the degree of freedom (N).

Components of the Mechanical System Library

Mechanical Fundamental Components

- Mass component: Concentrated inertia in one section of the model (motion quantities are absolute values referred to ground); can be connected to Spring (STF), Torque Source (SRCF), etc.
- Rigidity: Concentrated rigidity in one section of the model (motion quantities are difference values referred to predecessor/successor; can be connected to Mass (MAS), Velocity Source (SRCV), Ground (GND), etc.

Mechanical Auxiliary Components

- Coordinate Transformation: Implements transformation of motion coordinates; can be connected to all mechanical components.
- Torque/Force Source: Feeds an external force/torque quantity into the mechanical system; can be connected to MAS.
- Velocity Source: Feeds an external velocity quantity into the mechanical system; can be connected to STF.
- Ground Connection: Connects a rigidity component to GND (foundation, earth); can be connected to STF, etc.

Note:

Apart from characteristics, all component parameters are dynamic. To define a parameter value, you can use the name of a constant or any other Twin Builder system quantity. However, you cannot specify a mathematical expression unless you have assigned the expression to a variable first and then specified this variable as a Mechanical System component parameter.

Each parameter must comply with its permissible range. Violating this range (either by specifying an out-of-range number or by assigning a time function yielding out-of-range values during simulation) results in the specified parameter value being ignored and the corresponding permissible parameter limit being used instead. All component parameters are also available as outputs.

Certain properties are activated by specifying related parameters. If not activated, the properties and the related equations are not simulated. As a demonstration, a one-body vibrator model is used as an example, as shown in Figure 1., Here, certain characteristics and time functions are used to define certain properties and parameters. Most of them have been acquired by simulations. The same excitation function has been used for all simulations.

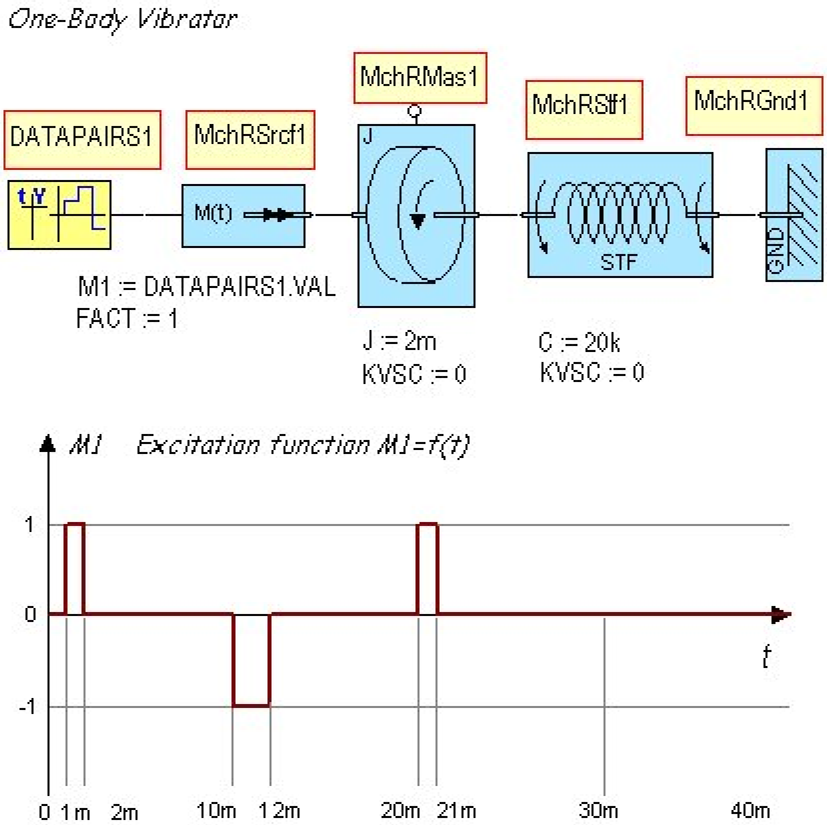


Figure 1. A three-body system in general coordinates

Properties and Restrictions

Nonlinear Properties

Components in Mechanical System library have many nonlinear properties, ranging from nonlinear frictions like Coulomb, progressive, degressive, and stick, to nonlinear rigidities like progressive, degressive, and backlash. A parameter can be nonlinear if it is assigned to an arbitrary system quantity (time functions or nonlinear characteristics).

Numerical Properties

To ensure high simulation accuracy and speed, especially when dealing with nonlinearities, all simulator capabilities of error control and dynamic time step adaptation are applied.

Note:

The choice of simulation time step can be important, especially for nonlinear systems. Trapezoidal integration method is required for optimum results.

Restrictions

The Mechanical System library computes only one-dimensional systems. For every component, only the dominant motion coordinate is computed. A body in space, however, can move along six coordinates, translation along the X-, Y-, and Z-axis and rotation around them. The motion coordinate can be assigned to any component. The Coordinate Transformation (TSF)-component transforms the motions between components. That means one body can rotate around the X-axis while the neighboring spring performs translational motions along the Z-axis. The Mechanical System library cannot compute rotations of a body around the X-axis while simultaneously rotating the same body around another axis.

If multiple motion coordinates are simultaneously active, components in the Mechanical System library can be used if the one of the following conditions applies:

- The other motion coordinates are negligible; then one can consider only the most dominant coordinate.

- The other motion coordinates can be mapped to the dominant one (e.g., by applying an equivalent friction characteristic or excitation force).

It is not sufficient to simply compute motions in different coordinates independently and then superimpose the results. Furthermore, be careful with translational systems. Although the motion coordinates of most of the components in rotational systems are aligned with the center of gravity of the components, this is not true for all rotational systems, or for most translational systems. If the center of gravity is displaced from the motion coordinate, extra torques/forces result that are not directly covered by components in the Mechanical System library.

Parameter Identification

The most essential mechanical parameters to be specified for components in the Mechanical System library are inertias (moments of inertia or masses), rigidities, and damping or friction coefficients.

Inertias

Inertia is the ability of a body to keep its motion and withstand accelerations. The inertia parameters, J [kgm^2] for rotational systems and M [kg] for translational systems, are defined according to the following:

$$\omega = \frac{1}{J} \cdot \int m_{Acc} \cdot dt \qquad v = \frac{1}{m} \cdot \int f_{Acc} dt$$

where ω is the angular velocity [rad/s], v is the velocity [m/s], m_{Acc} is the accelerating torque [Nm], and f_{Acc} is the accelerating force [N].

Rotational Moments of Inertia J

Since components in the Mechanical System library are designed mainly for rotational drive systems, rotational symmetric parts are mostly considered here. The parameter J [kgm^2] can be obtained by one of the following means:

- By measurements.
- From data sheet specifications (e.g., the rotor moment of inertia of electrical machines).
- As an approximation using simple geometric bodies.

Taking this into consideration, J [kgm^2] of a solid or hollow cylinder rotating around its longitudinal axis (center of gravity) can be calculated as follows:

$$J_{solid} = \rho \cdot l \cdot \frac{\pi}{32} \cdot d^4 \qquad J_{hollow} = \rho \cdot l \cdot \frac{\pi}{32} \cdot (d_o^4 - d_i^4)$$

where ρ is the density [kg/m^3] (ρ_{steel} = 7850 kg/m^3), l is the length [m], d_o/d_i the outer/inner shaft section diameter [m] .

In case a rotational symmetric body does not rotate centrally, but is displaced from its longitudinal axis/center of gravity in a parallel direction, Steiner's rule must be used:

$$J_A = J_G + ms^2$$

where J_A is the J of the working axis [kgm^2], J_G is the J in the center of gravity [kgm^2], m is the mass of the body [kg], and s is the displacement [m].

If several inertias, J_i, are connected together in series, the resulting moment of inertia, J_{res}, can be computed as the sum of the individual inertias:

$$J_{res} = J_1 + J_2 + ... + J_n = \sum_{i=1}^{n} J_i$$

Translational Masses m

If the motion coordinate is not displaced from the center of gravity of a translational moving body, or if the displacement is negligible, then only the mass parameter is necessary for translational systems.

The mass parameter m [kg] can be obtained by measurements, from data sheet specifications, or as an approximation using simple geometric bodies.

Given the volume (by geometric properties) and the material, the mass m [kg] can be calculated as follows:

$$m = \rho \cdot v$$

where ρ is the density [kg/m^3] (ρ_{steel} = 7850 kg/m^3) and V is the volume [m^3].

In case this body does not move along a coordinate axis aligned with its center of gravity, the same rule still holds as long as additional torques resulting from this displacement can be neglected.

If several inertias m_i are connected together in series, the resulting inertia m_{res} can be computed as the sum of the individual inertias:

$$m_{res} = m_1 + m_2 + \ldots + m_n = \sum_{i=1}^{n} m_i$$

Rigidities c

Rigidity is the stiffness of a spring. It specifies the amount of torque or force generated by linearly stretching, compressing or twisting a spring component. The following equations define the rigidity parameters:

$$c_{Rotational} = \frac{m_{Stf}}{\Delta\varphi} \qquad c_{Translational} = \frac{f_{Stf}}{\Delta s}$$

where m_{Stf} is the generated rigidity torque [Nm], f_{Stf} is the generated rigidity force [N], $\Delta\varphi$ is the angular displacement [rad], and Δs is the displacement [m].

Rotational Rigidities c

The parameter c [Nm/rad] can be obtained through vibration measurements, from data sheet specifications (the rigidity or rigidity characteristic of clutches/couplings), or as an approximation using simple geometric bodies. Always use the point of least rigidity (search for the softest point) for systems of more complex geometry and limited number of degrees of freedom (mass-rigidity number). The rigidity (spring stiffness) of a rotational symmetric part can be computed as following:

$$c = \frac{G \cdot \pi \cdot d^4}{l \cdot 32}$$

where c is the rotational rigidity [Nm/rad], d is the section diameter [m], l is the section length [m], and G is the shear modulus [N/m^2] ($G_{steel} = 8*10^{10}$ N/m^2).

When several rotational rigidities c_i are connected together in series, the resulting c_{res} can be computed as the following:

$$c_{res} = \frac{1}{\frac{1}{c_1} + \frac{1}{c_2} + \ldots + \frac{1}{c_n}} = \frac{1}{\sum_{i=1}^{n} \frac{1}{c_i}}$$

Translational Rigidities c

The parameter c [N/m] can be obtained by vibration measurements, from data sheet specifications (the rigidity or rigidity characteristic of springs), or as an approximation using simple geometric bodies. Always use the point of least rigidity (search for the softest point) for systems of more complex geometry and limited number of degrees of freedom (mass-rigidity number). There are various ways to determine translational rigidities analytically. One way is to assume a stretching cane, whose rigidity (spring stiffness) can be computed as follows:

$$c = \frac{E \cdot A}{l}$$

where c is the translational rigidity [N/m],l is the length [m], A is the cross-section area [m^2], E is the elastic (stress) modulus [N/m^2] (e.g., for a steel rope:E = 1011 N/m^2).

When several translational rigidities ci are connected together in series, the resulting c_{res} can be computed according to rotational rigidities.

Damping/Friction Coefficients and Characteristics

Friction and damping parameters describe how much resistive torque or force is generated depending on velocity.

Determining the exact damping coefficients, friction parameters, or characteristics is often rather difficult because they depend on various other factors. When tolerances allow, manufacturer specifications (data sheets) and handbooks (machinery handbooks) can be used. Furthermore, such parameters can be gained approximately by vibration studies.

The internal damping parameter (k_{Vsc} of a rigidity component) can be obtained through Lehr's attenuation ratio. Thus, from the resulting internal damping degree D_i, k_{Vsc} can be determined as follows:

$$k_{Vsc} = \frac{D_i \cdot \omega_{Dm}}{2 \cdot c_i}$$

where c_i is the ith section rigidity, ω_{Dm} is the natural frequency of section m. The natural frequencies ω of the n-body torsional vibrational system are determined by harmonic analysis (measurements) or by the eigenvalue problem solution.

See also Eigenvalues in Mechanical System Principle

The following equations calculate the natural frequencies for an undamped (ω_0) and damped (ω_D) one-body vibrator:

$$\omega_0 = \sqrt{\frac{c}{J}} \quad \text{or} \quad \omega_0 = \sqrt{\frac{c}{m}} \quad \text{and} \quad \omega_D = \omega \cdot \sqrt{1 - D_i^2}$$

Note the mutual dependency of k_{Vsc} and ω_D , i.e., it is impossible to compute one quantity without having determined the other one in a different way, such as via measurements.

Parameter Reduction

If neighboring (connected) mass and rigidity components do not have the same motion coordinate, then a coordinate transformation component, TSF, with a corresponding transformation coefficient, k_{Tsf}, can be placed in between.

Alternatively, the motion quantities of other mass or rigidity components can be reduced to their motion quantities. The motion quantities (jerks, torques and accelerations, velocities, and positions) are then transformed directly by k_{Tsf}. The parameters of the mass or rigidity, whose motion quantities are being transformed, are reduced by the square of the transformation coefficient, as follows:

$$J_{red} = \frac{J}{k_{Tsf}^{2}},\ m_{red} = \frac{m}{k_{Tsf}^{2}},\ c_{red} = \frac{c}{k_{Tsf}^{2}},\ k_{red} = \frac{k}{k_{Tsf}^{2}},\ \ldots$$

Engine Models

The engine models of the Mechanical System library provide basic behavioral representations of the internal combustion engine at two different levels. The mechanical connections of the machines are implemented using conservative nodes that follow the velocity-torque representation (velocity is the across quantity and torque is the trough quantity).

- Dynamic Engine Model (engine_dyn)
- Ideal Engine Model – Speed Source (engine_ss)

Engine — Dynamic Model

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

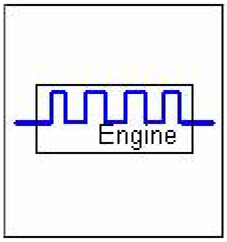


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example

Description

The dynamic engine model provides a basic behavioral model of the internal combustion engine.

The characteristic of the engine is taken from two files. The full load curve describes the maximum torque that can be produced internally at a given crankshaft speed. The losses curve describes the braking torque of the engine depending on the crankshaft speed if no fuel injection is present.

The instantaneous value of the internal engine torque is calculated from the losses curve and the output of a dynamic injection model. With the help of the input node, power request, a fraction of the full load torque can be specified. The product of this input value on the speed-dependent full load torque is fed into a first-order delay model of the injection system. The output of the injection system model is used to calculate the crankshaft acceleration. To consider the discontinuous function of the engine, the points when a cylinder fired are calculated. Only at these points, the value of the injection system model is imported into the mechanical model of the crankshaft.

Top

Assumptions and Limitations

1. The maximum torque that can be measured at the crankshaft for a given crankshaft speed is the difference of the value taken from the full load curve and the value taken from the losses curve.

2. When setting the minimum time step, the following parameters must be considered:

- The time constant of the injection control.

- The time between the firing of two cylinders (which depends on the number of cylinders, the number of strokes, and the engine speed).

Top

Mathematical Description

The typical full load curve and the losses curve are shown in Figure 2. These characteristics are decided by user-defined data files.

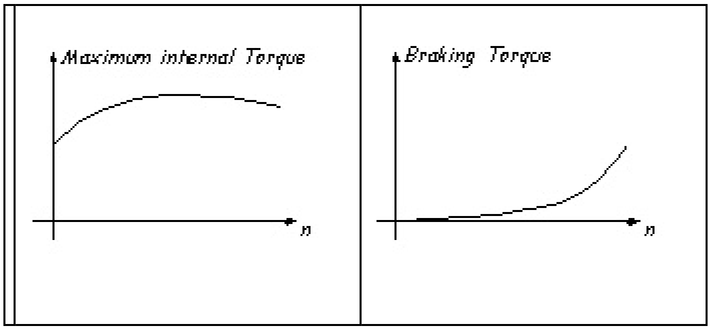


Figure 2. Typical full load curve and the losses curve of the Dynamic Engine Model

Top

Netlist Syntax

MODEL engine_dyn ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot2:= %0, rot1:= %1 (losses:= @losses, full_load:= @full_load, power_request:= @power_request, j:= @j, dead_time:= @dead_time, time_constant:= @time_constant, n0:= @n0) SRC: DB(Lib:-:=@ModelLibraryName) ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
rot1 (See Note)	Shaft – Side 1	Rotational_v
rot2 (See Note)	Shaft – Side 2	Rotational_v

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
j	Moment of Inertia	real	1 [kg*m^2]
power_ request	Relative Power Request	real	0.5
full_load	Characteristic Maximum Torque vs. Speed	file	—
losses	Characteristic Loss Torque vs. Speed	file	—
time_constant	Time Constant of Injection Control	real	10m [s]
dead_time	Delay Time due to Stroke	real	10m [s]
n0	Initial Speed at t=0	real	0.7k [rpm]

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
n	Speed [rpm]	Output	real
omega	Angular Velocity [rad/s]	Output	real
torque	Shaft Torque [Nm]	Output	real
torque_c	Combustion Torque [Nm]	Output	real
torque_l	Loss Torque [Nm]	Output	real
md	Internal Engine Torque [Nm]	Output	real

Top

Example

This example demonstrates the engine starting process. The switch S1 is closed at the beginning of the starting process, and connects the Battery (basic model) battery1 to the Starter through the Normally Open Relay rlyno1. When the engine speed exceeds 800 rmp, S1 is open and the starting process ends. The schematic of the system is show in Figure 3, the parameters are listed in Table 4, and the simulation results are shown in Figure 4.

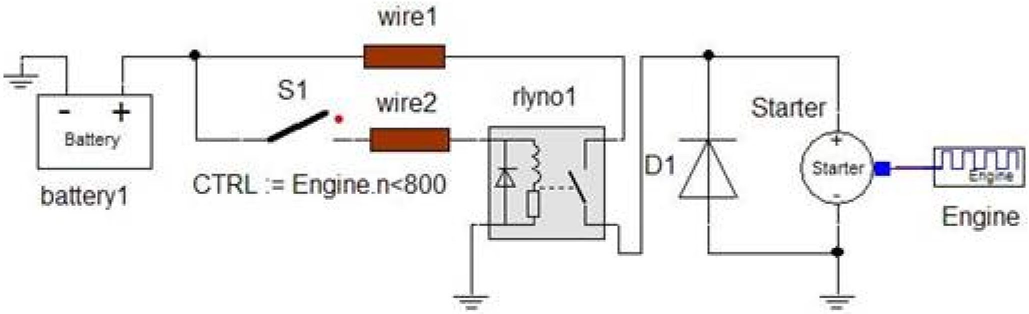


Figure 3. Application example of the Dynamic Engine model

Table 4. System Parameters

Component	Parameter	Value [unit]
Basic Battery battery1	rnom	14 [V]
	rin	10m [ohm]

wire1/wire2 (Level 1)	dia	2m [m]
	area	0.6793u [m^2]
	l	1 [m]
	lval	10n [H/m]
	i0	45 [A]
	tamb	20 [^{o}C]
	tinit	20 [^{o}C]
	tref	20 [^{o}C]
	enable_bread	0
	tdoff	1 [s]
	roff	1G [ohm]
	rho	1.72n [ohm m]
	alpha	3.93m [1/K]
	spgc	9k [kg/m^3]
	shcc	0.38k [J/(K kg)]
	kc	0.401k [W/(K m)]
	tcmelt	1084 [^{o}C]
	spgi	1.3k [kg/m^3]
	shci	1.5k [J/(K kg)]
	timelt	150 [^{o}C]
	ki	0.16 [W/(k m)]
Relay-Normally Open rlyno1	lcoil	0.5 [H]
	rcoli	5 [ohm]
	vpull	12 [V]
	vdrop	2 [V]
	ron	5m [ohm]
	roff	10Meg [ohm]
	tdmk	0.2 [s]
	tdbrk	3m [s]
	enable_ramp	1

Diode (Equivalent Line) D1	VF	0.8 [V]
	RB	1m [ohm]
	RR	100k [ohm]
Starter	ra	1m [ohm]
	la	1m [H]
	ke	100m
	j	10m [Kg m^2]
	cf	10m [Nms/rad]
	nclutch	800 [rpm]
	no	0 [rpm]
Engine	losses	1m [ohm]
	full_load	1m [H]
	power request	0.1
	j	1 [Kg m^2]
	dead_time	10m [s]
	time_constant	10m [s]
	no	0 [rpm]

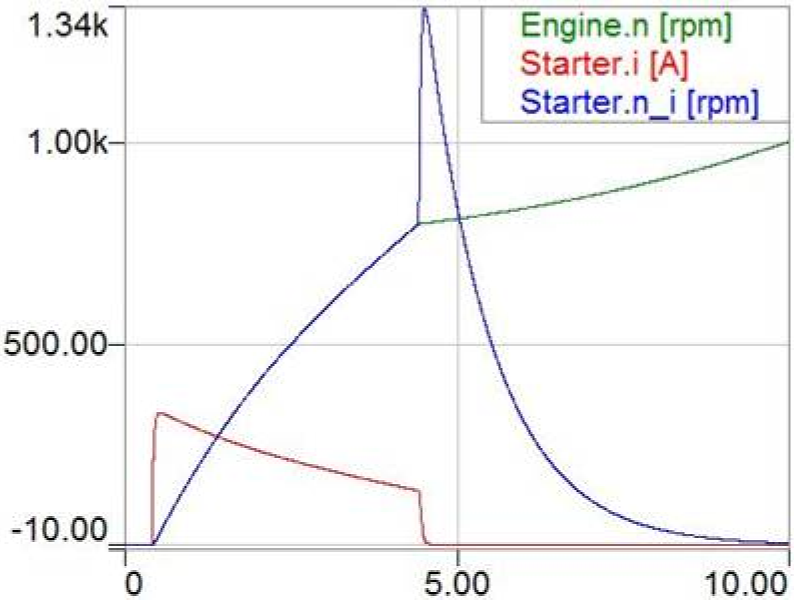


Figure 4. Simulation results

Engine — Speed Source

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

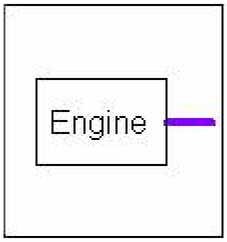


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

The engine speed source model forces the speed of the crankshaft to be on a level defined by the parameter n and measures the resulting load torque.

Top

Assumptions and Limitations

1. This model is applicable only when the engine has a constant load within its maximum load range.

2 The input parameter is in rpm, while the across quantity (angular velocity) at the terminal is converted to rad/s ($n \cdot 2\pi/60$).

Top

Mathematical Description

Top

Netlist Syntax

MODEL engine_ss ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot:= %0 (n:= @n) SRC: DB(Lib:=@ModelLibraryName) ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
rot	Shaft	Rotational_v

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
n	Speed	real	rpm

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
torque	Engine Torque	real	Nm

Top

Example

In this example, the Alternator alternator is connected to a PWM load LOAD. The speed of the alternator is decided by the Engine-Speed Source model engine. The schematic of the system is show in Figure 2, the parameters are listed in Table 4, and the simulation results are shown in Figure 3, 4 and 5.

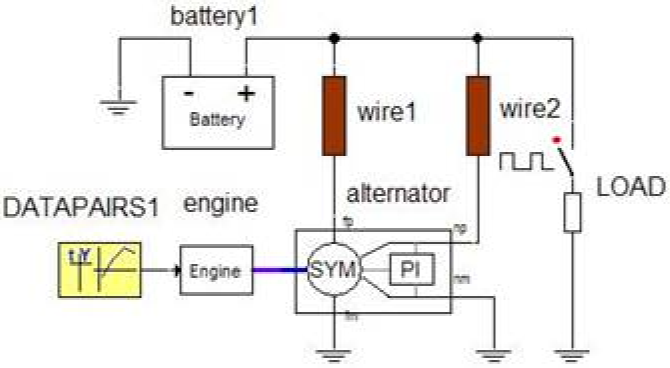


Figure 2. Application example of the Engine-Speed Source model

Table 4. System Parameters

Component	Parameter	Value [unit]
Basic Battery battery1	rnom	14 [V]
	rin	10m [ohm]

wire1/wire2 (Level 1)	dia	2m [m]
	area	0.6793u [m^2]
	l	1 [m]
	lval	10n [H/m]
	i0	45 [A]
	tamb	20 [^{o}C]
	tinit	20 [^{o}C]
	tref	20 [^{o}C]
	enable_bread	0
	tdoff	1 [s]
	roff	1G [ohm]
	rho	1.72n [ohm m]
	alpha	3.93m [1/K]
	spgc	9k [kg/m^3]
	shcc	0.38k [J/(K kg)]
	kc	0.401k [W/(K m)]
	tcmelt	1084 [^{o}C]
	spgi	1.3k [kg/m^3]
	shci	1.5k [J/(K kg)]
	timelt	150 [^{o}C]
	ki	0.16 [W/(k m)]

alternator	rs	30m [ohm]
	ls	0.2 [H]
	re	3.5 [ohm]
	le	0.8 [H]
	p	6
	k	50m
	p_gain	0.2
	i_gain	0.1
	vref	14 [V]
	vd	0.7 [V]
Engine-Speed Source	n	DATAPAIRS1.VAL
PWM Load	period	1 [s]
	dc	0.5
	r	1 [ohm]
	phase	0 [degree]
	td	0 [s]

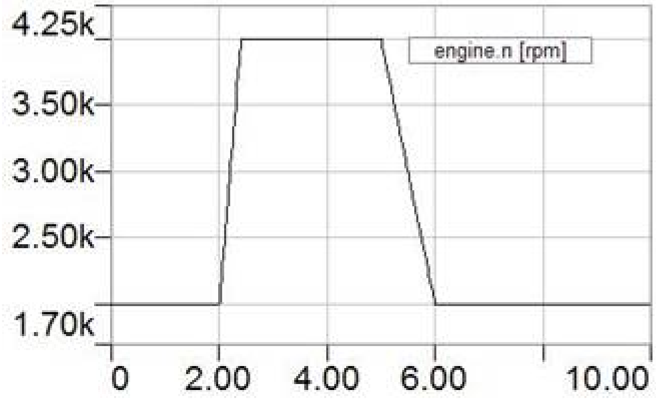


Figure 3. Simulation result-engine speed

Figure 4. Simulation result-load current

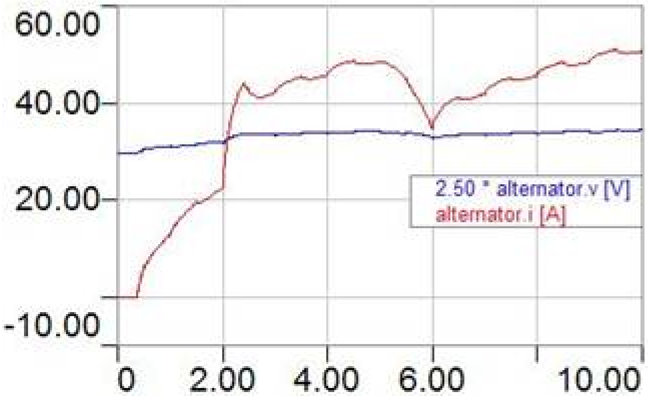


Figure 5. Simulation results-alternator current and voltage

Top

References

Rotational_V Components

- Fan Model (fan)
- Ground - rotational (MchRGnd)
- Mass -rotational (MchRMas)
- Torque Source - rotational (MchRSrcf)
- Velocity Source - rotational (MchRSrcv)
- Rigidity - rotational (MchRStf)

Fan Component

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

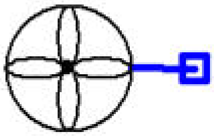


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

The fan model provides a load torque that consists of one part that depends linearly on the rotational velocity and a second part that increases with the square of the rotational velocity.

Top

Assumptions and Limitations

Top

Mathematical Description

Load Torque(T) = $cf*\omega + windage*\omega^2$

Where cf is the friction coefficient and windage is the windage coefficient.

Top

Netlist Syntax

MODEL fan ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1:= %0 (cf:= @cf, windage:= @windage) SRC: DB(Lib:=@ModelLibraryName) ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
rot1 (See Note)	Shaft1	rotational_v

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
windage	Windage Coefficient	real	0.1 [Nm*s^2/rad^2]
cf	Friction Coefficient	real	0 [Nm*s/rad]

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
n	Speed [rpm]	Output	real
OMEGA	Angular velocity [rad/s]	Output	real
TORQUE	Torque Output [Nm]	Output	real

Top

Example

In this example, the Mass (rotational) MASS_ROT1 is given an initial angular velocity of 400 rpm. Once the simulation starts, the fan friction and windage decelerate the mass. The schematic of the system is shown in Figure 2, parameters of the system are listed in Table 4, and the simulation results are shown in Figures 3 and 4.

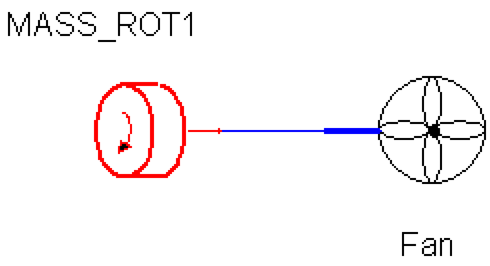


Figure 2. Application example of the fan model

Table 4

Component	**Parameter**	**Value [Unit]**
Rotational Mass (Velocity Force) MASS_ROT1	OMEGA0	400 [rpm]
Fan Model FAN	cf	0.01 [Nms/rad]
	windage	0.025 [Nms^2/rad^2]

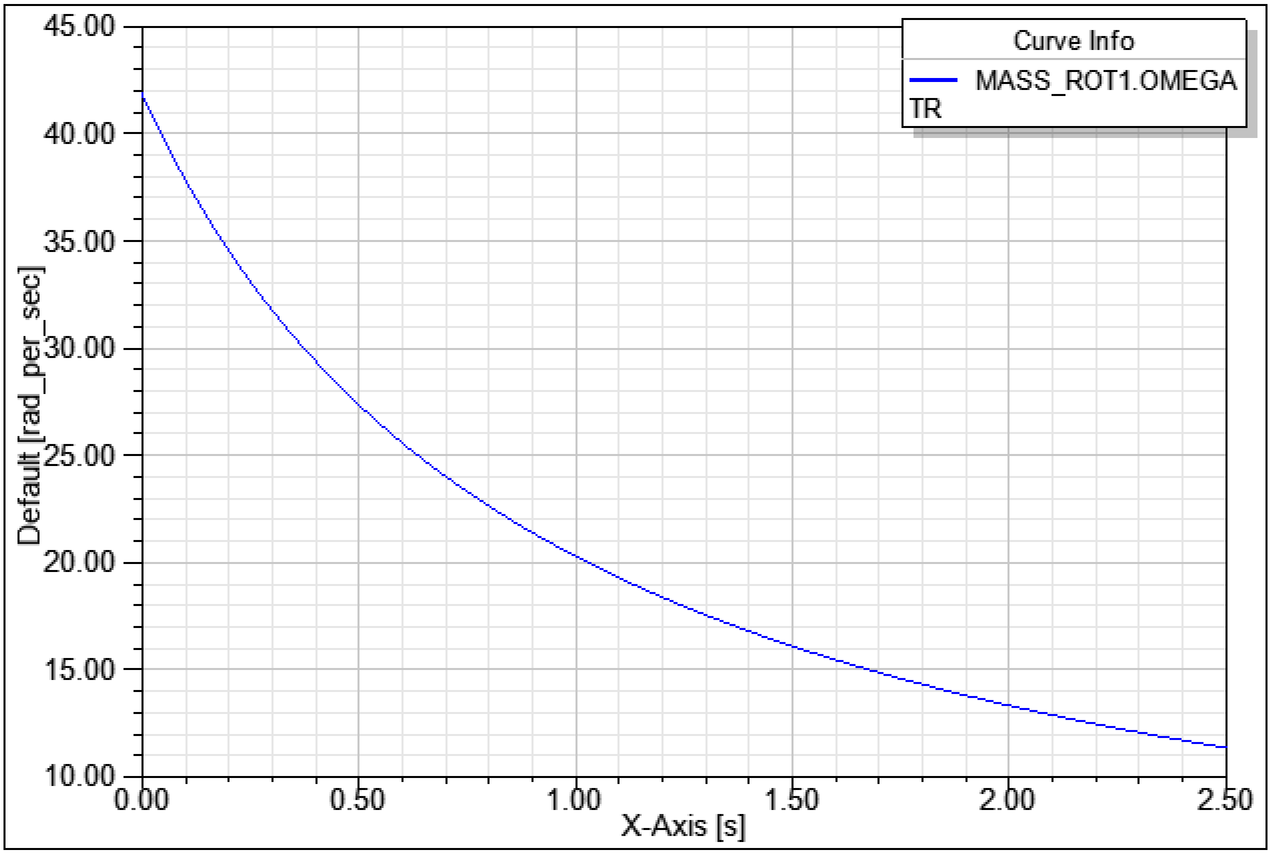


Figure 3. Simulation results - Rotational speed of mass MASS_ROT1.

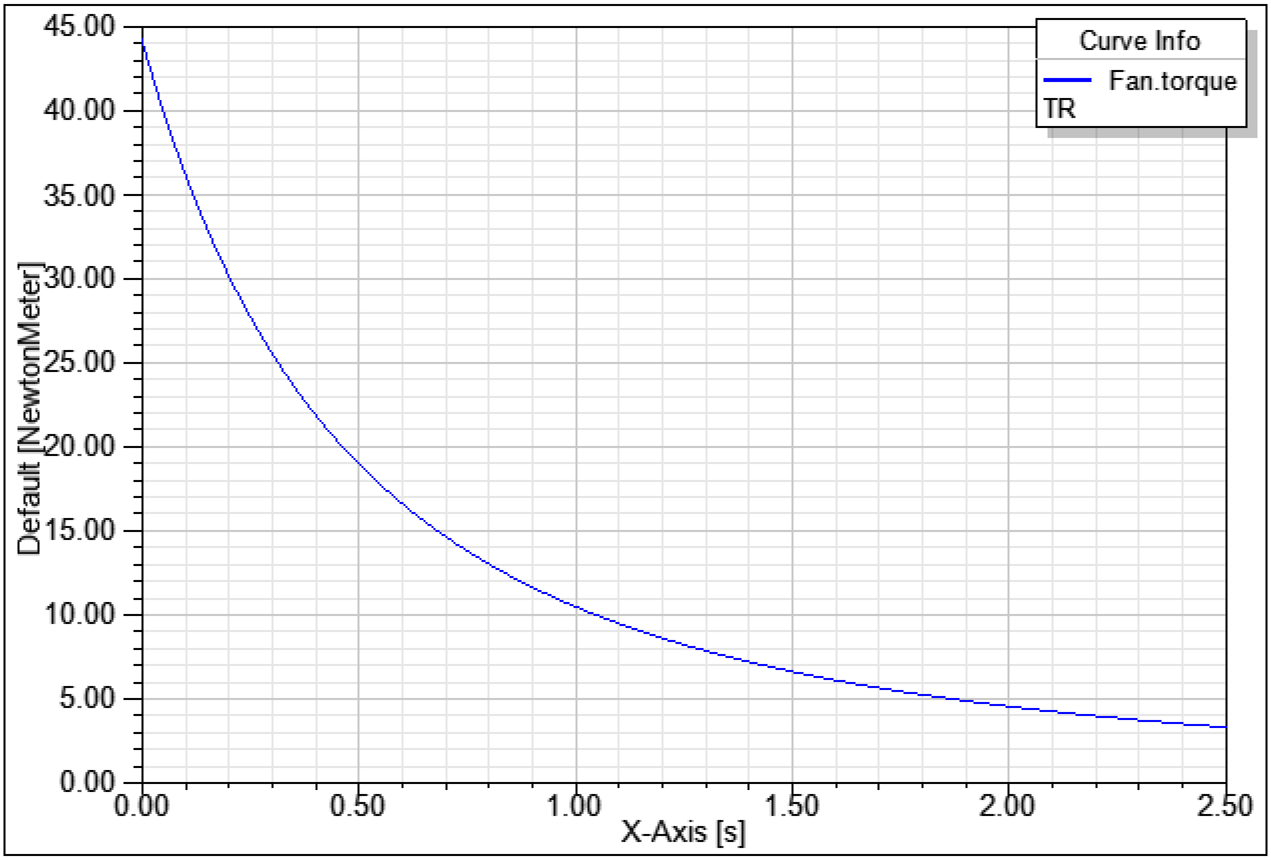


Figure 4. Simulation results - Fan torque.

Top

References

Ground Component (Rotational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

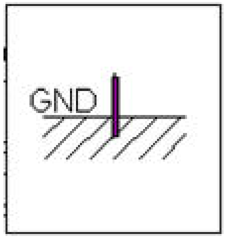


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Input/Output Quantities
- Example
- References

Description

This component enables the connection of a rigidity component (shaft section) with ground to transmit a torque into the foundation. The torque M[N], which is received by the ground, is provided as an output.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

UMODEL VM_ROT VM_ROT_Src_?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %0, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchRGnd ?InstanceName(@InstanceName):(@(Refbase)@(ID)) GND

:= %0, Nul := GND ( ) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
GND (See Note)	Ground Node (for connection to a rigidity component)	Rotational_V

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Input/Output Quantities

Table2

Name	Description [Unit]	Direction	Data Type
M	Torque output [Nm]	Output	real

Top

Example

Top

References

Mass Component (Rotational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

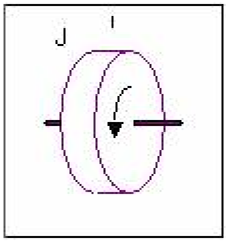


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

The mass component concentrates all inertial properties of a mechanical n-body system section. All motion quantities are absolute quantities referring to ground.

Top

Assumptions and Limitations

The mass component can only be connected to rigidity or torque source components. If the mass component is connected to a coordinate transformation component, make certain that the component after the coordinate transformation component is either a rigidity or torque source. You cannot connect two mass components directly. Several rigidity components, however, can be connected to a mass component.

Note:

Except for the moment of inertia J, all properties and corresponding parameters are optional. The related equations are not calculated unless the property has been activated.

Top

Mathematical Description

- Properties and Restrictions

Inertia

The parameter J [kgm^2] specifies the concentrated inertia of the mass component. For more information, refer to Inertias.

Friction Properties

General information on Friction property can be found at Friction properties.

Mass Component Viscous Friction

If Viscous Friction Coefficient KVSC [Nms/rad] > 0, the component calculates the linear speed dependent friction resistive torque MVSC [Nm], as follows,

$MVSC = KVSC \cdot \omega(t),\ 0 \leq KVSC < \infty$

The linear characteristic and typical time function (exponential envelope) of KVSC and MVSC are shown in Figure 2.

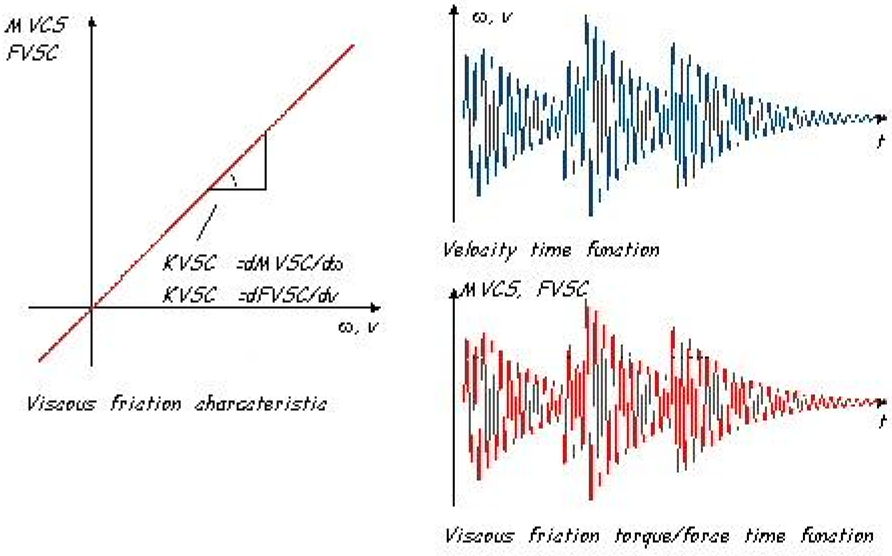


Figure 2. The linear characteristic and time function (exponential envelope) of KVSC and MVSC

Mass Component Coulomb Friction

If Constant Coulomb Friction Torque MCLB > 0, the component calculates a constantresistive torque MFRCCLB [Nm], which is independent of the speed, but dependent on the direction of the speed.

If $|\omega(t)| \geq OMEGACLB$, then $MFRCCLB = MCLB \cdot sign(\omega(t))$

If $|\omega(t)| < OMEGACLB$, then $MFRCCLB = MCLB/ OMEGACLB \cdot sign(\omega(t))$

where OMEGACLB [rad/s] is the angular velocity at which constant coulomb friction becomes active.

The constant friction characteristic and typical linear time function of ω and MFRCCLB are shown in Figure 3.

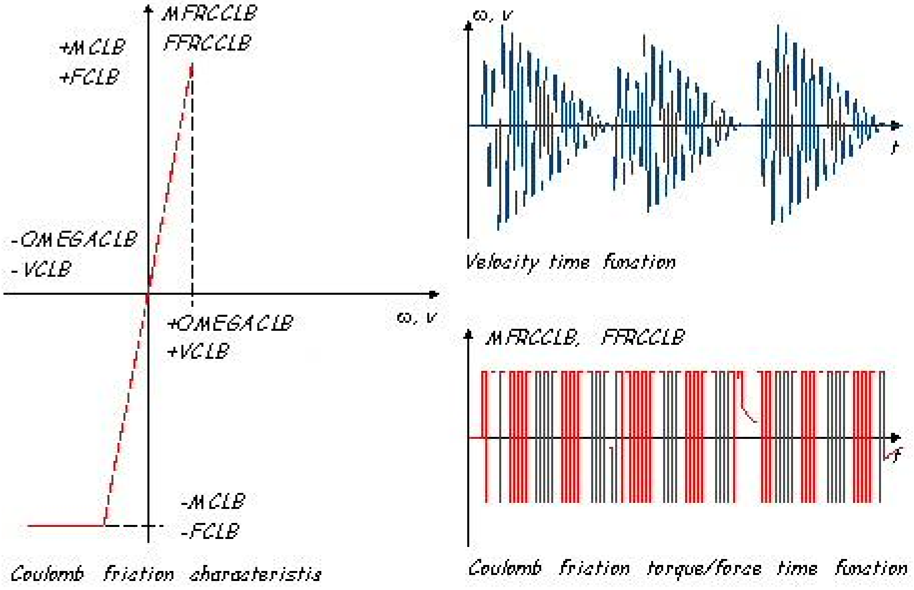


Figure 3. The constant friction characteristic and typical linear time function of ω and MFRCCLB

The simulator reduces the simulation time step to reach the limits of the viscous range given by $|\omega(t)| < \text{OMEGACLB}$. In case it is rather small and the minimum time step is rather large, such that it has not been hit, then simulation continues, and a simulator message appears with the model name, component name, and time instant.

Mass Component Polynomial Friction

If the Polynomial Friction Multiplier $\text{KMUL} < 0$, the component calculates a speed-dependent polynomial resistive torque MFRCPOL [Nm]. This property is typical for frictions with streaming media (e.g., windage).

$$\text{MFRCPOL} = \text{KMUL} \cdot |\omega(t)|^{\text{KEXP}} \cdot \text{sign}(\omega(t))$$

The characteristic and typical time function of MFRCPOL and FFRCPOL are shown in Figure 4. Notice that the first quadrant is always evoked into the third quadrant no matter what KEXP is.

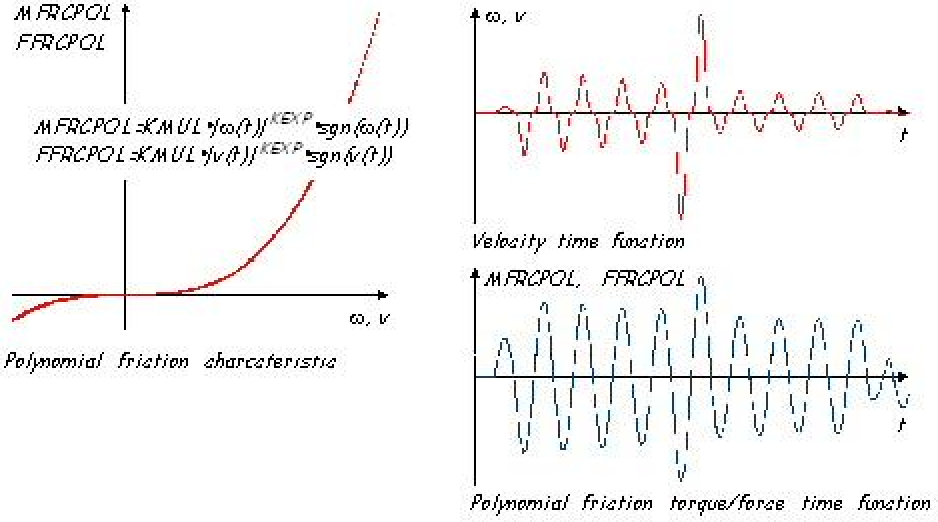
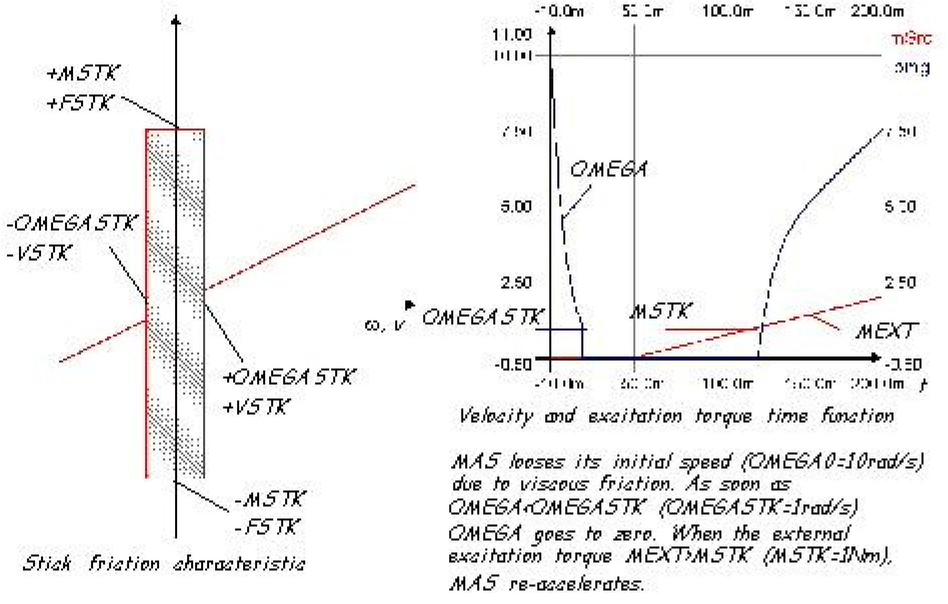


Figure 4. The characteristic and typical time function of MFRCPOL and FFRCPOL

Mass Component Stick Friction (“Stiction”)

If Stick Angular Velocity OMEGASTK > 0, the component calculates the behavior where a mass gets stuck (stick friction) at very low speeds. As soon as the speed falls below a certain limit OMEGASTK, the mass latches into stick friction state. Braking occurs with the viscous stick friction coefficient KSTK. The mass can only reaccelerate after the accelerating force exceeds the stick friction force ("break-off torque/force") MSTK.

The characteristic and typical time function of MSTK and FSTK are shown in Figure 5. Notice that the Speed suddenly goes to zero while within the hatched area.



MAS looses its initial speed (OMEGA0=10rad/s) due to viscous friction. As soon as OMEGA<OMEGASTK (OMEGASTK=1rad/s) OMEGA goes to zero. When the external excitation torque MEXT>MSTK (MSTK=1Nm), MAS re-accelerates.

Figure 5. The characteristic and typical time function of MSTK and FSTK

Usually OMEAGASTK is rather small. On the transition from positive to negative speeds or vice versa, the simulator reduces the simulation time step to reach the stick range given by (-OMEAGASTK, + OMEAGASTK) (hatched area). In case the required time step is smaller than the minimum time step HMIN, simulation continues without computing the stick property at this instant, and a simulator message appears with the model name, component name, and time value.

Mass Component Characteristic Friction

If the Characteristic Friction Torque Coefficient MFRCCH > 0, the component calculates an arbitrary, weighted, speed-dependent resistive torque MFRCCH = f(ω(t)), which can be specified with a characteristic (preferably a 2D Lookup Table characteristic).

MFRCCH_VAL = KFRCCH · MFRCCH (ω(t))

where KFRCCH is the weighting coeffient.

Figure 6 shows an arbitrary friction characteristic defined by distinct points.

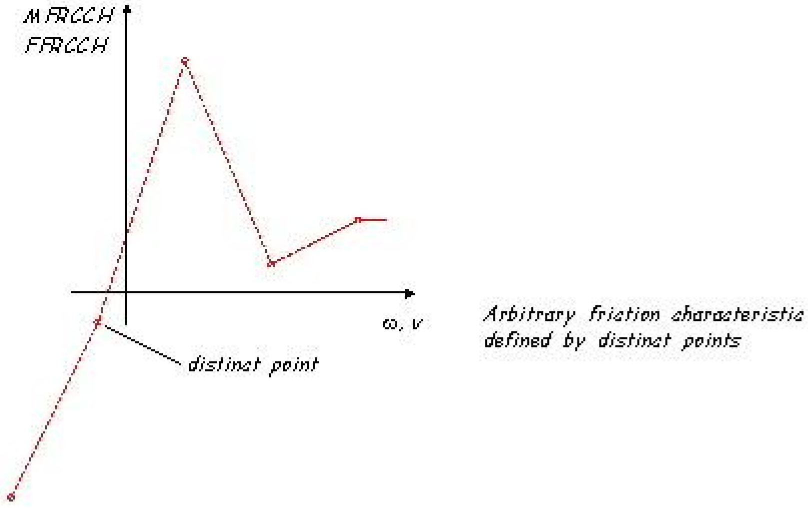


Figure 6. An arbitrary friction characteristic

Mass Component External Friction

If the External (arbitrary) Torque MEXT > 0, the component calculates an arbitrary, weighted, externally computed resistive torque (time function) MEXT = f(t), which can be any arbitrary Twin Builder quantity (e.g., the result of a formula, a block output quantity of the block diagram module, or a state graph quantity).

MEXT_VAL = KEXT · MEXT (t)

where KEXT is the weighting coefficient.

Figure 7 shows an arbitrary characteristic function for the External (arbitrary) Torque MEXT and the External (arbitrary) Force FEXT.

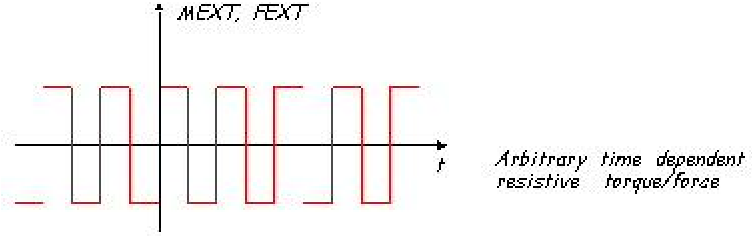


Figure 7. An arbitrary characteristic function for MEXT and FEXT

Limits and Initial Values

Mass Component Upper and Lower Position Limit

If the Upper Angular Position limit PHIU ≠ 0, the Lower Angular Position limit PHIL ≠ 0 and PHIU > PHIL, the component considers the current position φ(t). When the position exceeds the upper limit PHIU or φ(t) falls below the lower limit PHIL, then this position is retained, and the angular velocity becomes zero (or enters stick state if stick friction is activated). The system can only be restarted after reversing the motion or exceeding break-off torque MSTK. The limits are typical for driving a body against a "wall".

Figure 8 shows the typical time function of φ(t) and ω(t). Notice that as soon as φ(t) reaches its upper or lower limit, ω(t) suddenly goes to zero.

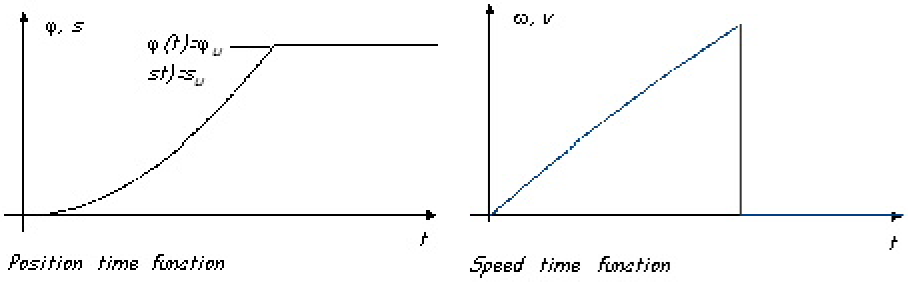


Figure 8. typical time function of φ(t) and ω(t)

On the transition from the linear position range to either the upper or lower limit, the simulator reduces the simulation time step to its minimum HMIN to reach the limit. If PHIL and PHIU are close together (the linear position range is rather small) and the required simulation time step is smaller than the minimum simulation time step HMIN, then a message appears during simulation, displaying the model name, component name, and time instant. The position limits are dynamic parameters (i.e., they can change during simulation). If the conditions are violated during simulation, the parameters stay at their permissible limits, the simulation continues, and a simulator message appears with the model name, component name, and time instant.

Mass Component Initial Velocity and Initial Position Value

The initial values for velocity OMEGA0 and position PHI0 can be specified to define the starting points; the default values are zero..

Top

Netlist Syntax

UMODEL VM_ROT VM_ROT_Src_?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %1, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_ROT VM_ROT_Pre_?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %1, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_ROT VM_ROT_Suc_?InstanceName (@InstanceName):(@(Refbase)@(ID)) ROT1 := %2, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchRMas ?InstanceName (@InstanceName):(@(Refbase)@(ID)) ROT := %0, ROT1 := %1, ROT2 := %2, GND := GND (J:=@J ,KVSC:=@KVSC ,KMUL:=@KMUL ,KEXP:=@KEXP ,MCLB:=@MCLB ,OMEGACLB:=@OMEGACLB ,OMEGASTK:=@OMEGASTK ,MSTK:=@MSTK ,KSTK:=@KSTK ,MFRCCH:=@MFRCCH ,KFRCCH:=@KFRCCH ,MEXT:=@MEXT ,KEXT:=@KEXT ,OMEGA0:=@OMEGA0 ,PHI0:=@PHI0 ,PHIU:=@PHIU ,PHIL:=@PHIL) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
ROT (See Note)	Source Node	Rotational
ROT1 (See Note)	Predecessor Node	Rotational
ROT2 (See Note)	Successor Node	Rotational

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
J	Moment of Inertia	real	1 [kg m^2]
KVSC	Viscous Friction Coefficient	real	0 [Nms/rad]
KMUL	Polynomial Friction Multiplier	real	0 [Nms/rad]
KEXP	Polynomial Friction Exponent	real	0
MCLB	Constant Coulomb Friction Torque	real	0 [Nm]
OMEGACLB	Coulomb Angular Velocity	real	1m [rad/s]

OMEGASTK	Stick Angular Velocity	real	0 [rad/s]
MSTK	Stick Torque (break-off)	real	0 [Nm]
KSTK	Stick Friction Coefficient	real	1e16 [Nms/rad]
MFRCCH	Characteristic Friction Torque (m=f(OMEGA))	real	0 [Nm]
KFRCCH	Characteristic Friction Torque Coefficient	real	1
MEXT	External (arbitrary) Torque (m=f(t))	real	0 [Nm]
KEXT	External (arbitrary) Torque Coefficient	real	1
OMEGA0	Initial Angular Velocity	real	0 [rad/s]
PHI0	Initial Angular Position	real	0 [rad]
PHIU	Upper Angular Position	real	0 [rad]
PHIL	Lower Angular Position	real	0 [rad]

Top

Input/Output Quantities

Table3

Name	**Description [Unit]**	**Direction**	**Data Type**
dALPHA	Angular jerk [rad/s^3]	Output	real
ALPHA	Angular acceleration [rad/s^2]	Output	real
OMEGA	Angular velocity [rad/s]	Output	real
PHI	Angular position [rad]	Output	real
M	Torque supplied by source [Nm]	Output	real
M1	Torque supplied by pre-decessor [Nm]	Output	real
M2	Torque supplied to successor [Nm]	Output	real
MACX	Accelerating external torques [Nm]	Output	real
MACC	Accelerating torque [Nm]	Output	real
MFRC	Resulting friction torque [Nm]	Output	real
MVSC	Viscous friction torque [Nm]	Output	real
MFRCPOL	Polynomial friction torque [Nm]	Output	real

MFRCCLB	Coulomb friction torque [Nm]	Output	real
MFRCCH_ VAL	Characteristic friction torque [Nm]	Output	real
MEXT_VAL	External (arbitrary) torque [Nm]	Output	real

Top

Example

In this example, the Mass (rotational) Mas1 is driven by theTorque Source Srcf1, whose output value is decided by the 2D Lookup Table mStk. The initial angular velocity of Mas1 is 10rad/s. The output torque of Srcf1 is 0 when simulation time t < 50ms, and increases linearly after t reaches 50 ms. The schematic of the system is shown in Figure 9, parameters of the system are listed in Table 4, and the simulation results are shown in Figure 10.

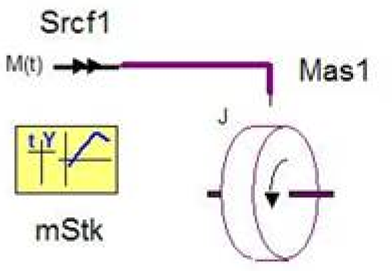


Figure 9. Application example of the mass (rotational) model

Table 4

Component	**Parameter**	**Value [Unit]**
Mass (Rotational) Mas1	J	2.1m [kg m^2]
	KVSC	0,25 [Nms/rad]
	KMUL	0 [Nms/rad]
	KEXP	0
	MCLB	0 [Nm]
	OMEGACLB	1m [rad/s]
	OMEGASTK	0 [rad/s]
	MSTK	0 [Nm]
	KSTK	1e16 [Nms/rad]
	MFRCCH	0 [Nm]
	KFRCCH	1

	MEXT	0 [Nm]
	KEXT	1
	OMEGA0	10 [rad/s]
	PHI0	0 [rad]
	PHIU	0 [rad]
	PHIL	0 [rad]
Torque Source Srcf1	M1	mStk.VAL [Nm]
	FACT	1

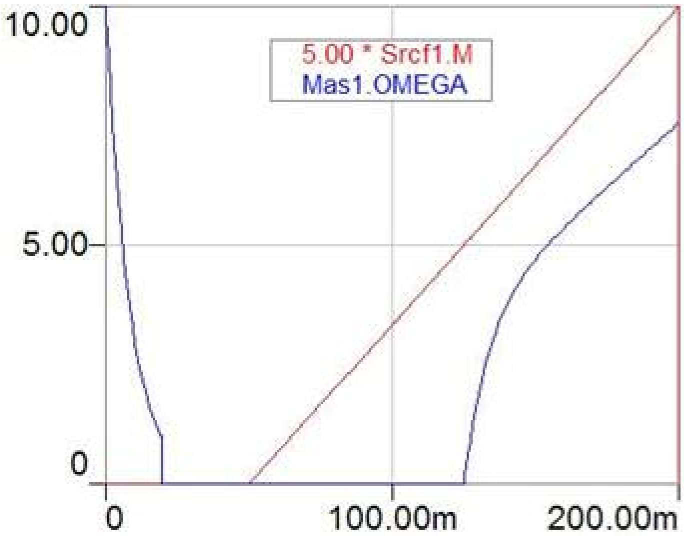


Figure 10. Simulation results

Top

References

Torque Source (Rotational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

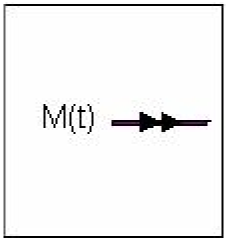


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

This component provides an acceleration torque, which is calculated from a Twin Builder quantity, for a mechanical mass node. The mechanical system can use external forces. Outputs include torque and angular velocity.

Top

Assumptions and Limitations

No internal inertia or losses are considered in this component. The Torque Output of this component is proportional to the External Torque Source at a ratio specified by the External Source Weighting Coefficient, and there is no output limits.

Top

Mathematical Description

$M = FACT \cdot M1$

Where M1 is the External Torque, FACT is the External Source Weighting Coefficient, and M is the Torque Output.

Top

Netlist Syntax

UMODEL VM_ROT VM_ROT_Src_?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %0, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchRSrcf ?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT := %0, GND := GND (M1:=@M1 ,FACT:=@FACT) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
ROT (See Note)	Source Node (for connection to a mass component)	Mechanical

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
M1	Source (arbitrary) torque quantity (m = f(t))	real	0 [Nm]
FACT	Source weighting factor	real	1

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
M	Torque Output [Nm]	Output	real
OMEGA	Angular velocity [rad/s]	Output	real

Top

Example

This example shows a constant Torque Source Srcf1 applied to a Mass (rotational) block Mas1 with upper position limit and stick friction. The angular velocity of Mas1 increase as a result of the applied torque output of Srcf1. Mas1 rotates till it reaches its upper position limit then stops, and its angular velocity drops to zero. The schematic of the system is shown in Figure 2, system parameters are listed in Table 4, and the simulation results are shown in Figure 3, 4 and 5.

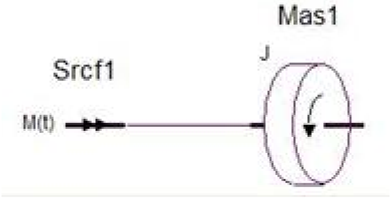


Figure 2. Application example of the Torque Source model

Table 4

Component	Parameter	Value [Unit]
Mass (Rotational) Mas1	J	10m [kg m^2]
	KVSC	50m [Nms/rad]
	KMUL	0 [Nms/rad]
	KEXP	0
	MCLB	0 [Nm]
	OMEGACLB	1m [rad/s]
	OMEGASTK	1u [rad/s]
	MSTK	5 [Nm]
	KSTK	1e16 [Nms/rad]
	MFRCCH	0 [Nm]
	KFRCCH	1
	MEXT	0 [Nm]
	KEXT	1
	OMEGA0	0 [rad/s]
	PHI0	0.5 [rad]
	PHIU	1 [rad]
	PHIL	-6 [rad]

Torque Source Srcf1	M1	6
	FACT	1

Figure 3. Simulation result – Accelerating External Torque MACX of Mas1

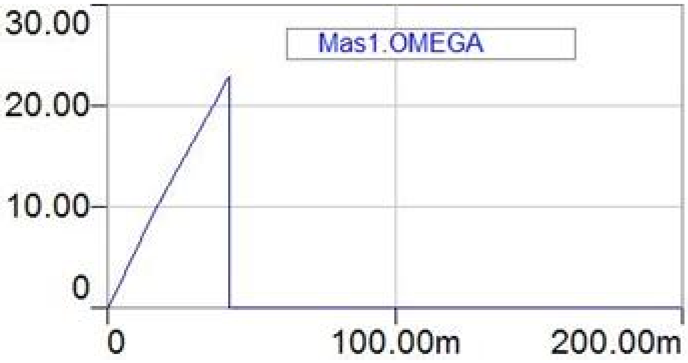


Figure 4. Simulation result – Angular Velocity OMEGA of Mas1

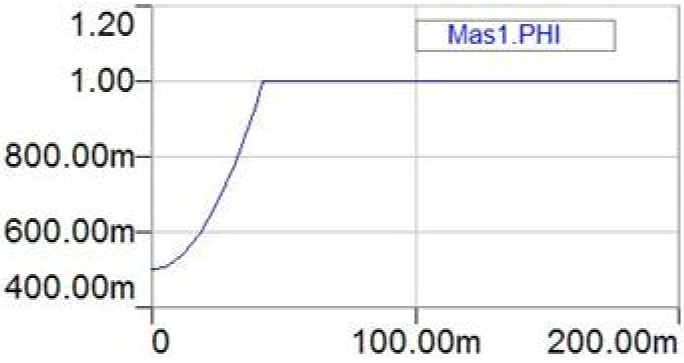


Figure 5. Simulation result – Position PHI of Mas1

Top

References

Velocity Source (Rotational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

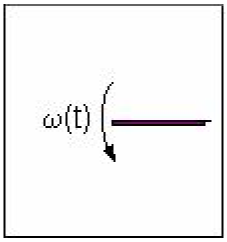


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

This component provides an angular velocity, which is calculated froma Twin Builder quantity, for a mechanical rigidity node. The mechanical system can use (system) external speeds.

Top

Assumptions and Limitations

No internal inertia or losses are considered in this component. The Velocity Output of this component is proportional to the External Velocity Source at a ratio specified by the External Source Weighting Coefficient, and there is no output limits.

Top

Mathematical Description

$\mathrm{OMEGA} = \mathrm{FACT} \cdot \mathrm{OMEGA1}$

Where OMEGA1 is the External Velocity Source, FACT is the External Source Weighting Coefficient, and OMEGA is the Velocity Output.

Top

Netlist Syntax

UMODEL VM_ROT VM_ROT_Src_?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %0, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchRSrcv ?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT := %0, GND := GND (OMEGA1:=@OMEGA1 ,FACT:=@FACT) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
ROT (See Note)	Source Node (for connection to a rigidity component)	Rotational

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
OMEGA1	Source Velocity Quantity	real	rad/s
FACT	Source Weighting Factor	real	

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
M	Torque Output [Nm]	Output	real
OMEGA	Angular velocity [rad/s]	Output	real

Top

Example

This example shows a one-body vibrator with the Rigidity Stf1 driven by a Velocity Source Srv1, as shown in Figure 2. System parameters are listed in Table 4. The torque output of the Velocity Source Srcv1 is decided by a 2D-Lookup Table omh_Ext, as shown in Figure 3. The simulation results, Angular Velocity OMEGA of Mas1 and the Difference Torque DM of Stf1, are shown in Figure 4 and 5, respectively.

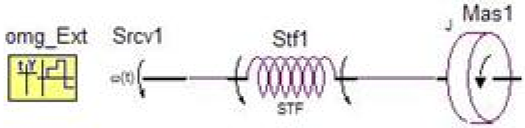


Figure 2. Application example of the Velocity Source (rotational) model

Table 4

Component	Parameter	Value [Unit]

Rigidity (Rotational) Stf1	C	20k [Nms/rad]

	KVSC	0.25 [Nms/rad]
	CMUL	0 [Nms/rad]
	CEXP	0
	KMUL	1 [Nms/rad]
	KEXP	5
	KVSCMUL	0.1k [Nms/rad]
	KVSCEXP	3
	CPGR	0
	KPGR	0
	CDGR	0
	KDGR	10
	DPHIBKLU	0 [rad]
	DPHIBKLL	0 [rad]
	CBKL	0 [Nm/rad]
	KBKL	0 [Nms/rad]
	KBKLEXP	0
	KBKLCND	1
	DPHISTPU	0 [rad]
	DPHISTPL	0 [rad]
	CSTP	0 [Nm/rad]
	KSTP	0 [Nms/rad]
	KSTPEXP	0
	KSTPCND	1
	MSTFCH	0 [Nm]
	KSTFCH	1
	MFRCCH	0 [Nm]
	KFRCCH	1
	MEXT	0 [Nm]
	KEXT	1
	DPHI0	0 [rad]

Mass (Rotational) Mas1	J	2m [kg m^2]
	KVSC	0 [Nms/rad]
	KMUL	0 [Nms/rad]
	KEXP	0
	MCLB	0 [Nm]
	OMEGACLB	1m [rad/s]
	OMEGASTK	0 [rad/s]
	MSTK	0 [Nm]
	KSTK	1e16 [Nms/rad]
	MFRCCH	0 [Nm]
	KFRCCH	1
	MEXT	0 [Nm]
	KEXT	1
	OMEGA0	0 [rad/s]
	PHI0	0 [rad]
	PHIU	0 [rad]
	PHIL	0 [rad]
Velocity Source(rotational) Srcf1	OMEGA1	omg_Ext.VAL
	FACT	1

Figure 3. Simulation result – Angular Velocity output OMEGA of Srcv1

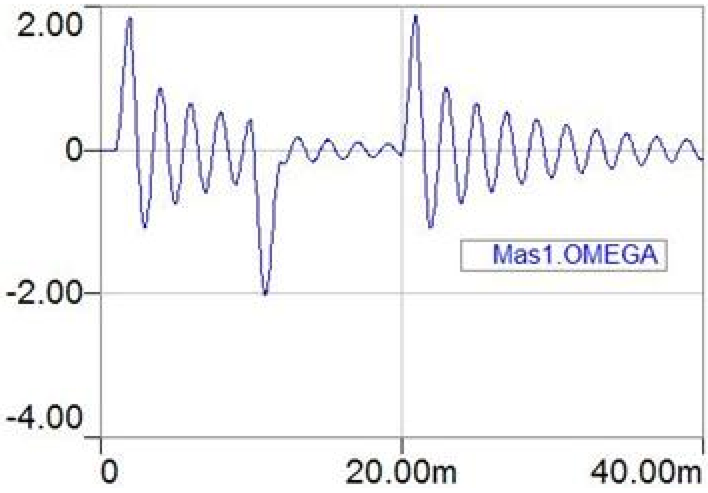


Figure 4. Simulation results – the Angular Velocity OMEGA of Mas1

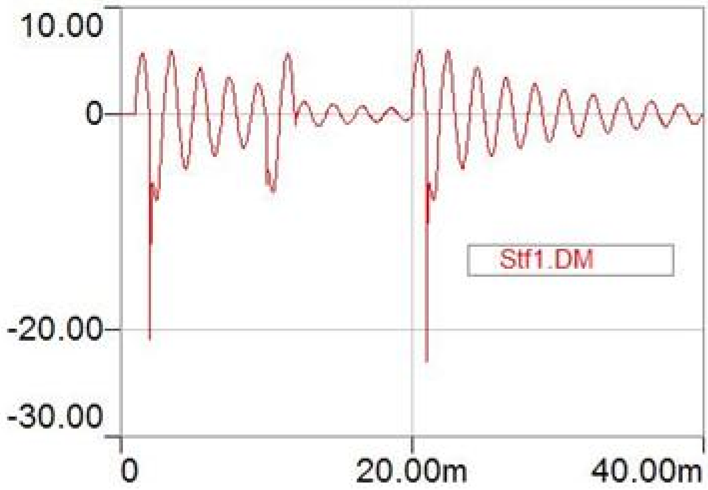


Figure 5. Simulation results – the Difference Torque DM of Stf1

Top

References

Rigidity Component (Rotational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

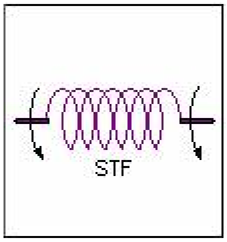


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

The rigidity component concentrates all spring properties of a mechanical n-body system section. All motion quantities are relative quantities that refer to the difference between the predecessor and successor masses.

Top

Assumptions and Limitations

The rigidity component can only be connected to a mass component, a velocity source, or a ground component. If the rigidity component is connected to a coordinate transformation component, make certain that the component beyond the coordinate transformation component is either a mass component, a velocity source, or a ground component. You cannot connect two rigidity components directly or several mass components to one rigidity.

Top

Mathematical Description

All torques resulting from the angular position difference $\Delta\varphi$ [rad] between predecessor and successor body are rigidity (stiffness) torques MSTF [Nm] with rigidity parameter C [Nm/rad]: MSTF = f(t, C, $\Delta\varphi$) .

All torques resulting from the angular velocity difference $\Delta\omega$ [rad/s] between predecessor and successor body are friction torques MFRC [Nm] with parameter K [Nms/rad]: MFRC = f(t, K, $\Delta\omega$).

The friction and damping parameter K can depend on the position difference K= f($\Delta\varphi$). The position dependent frictions result in MFRC = f(t, K, $\Delta\varphi$, $\Delta\omega$).

Note:

All properties and corresponding parameters are optional. The equations are not simulated unless the property has been activated.

- Rigidity properties
- Friction/Damping properties
- Backlash property
- Second step property
- External difference torque and initial values

Rigidity Properties

Rigidity Torque - Position Difference Plane Areas

The rigidity torque - the position difference plane, MSTF = f($\Delta\varphi$), is subdivided into five areas, which are separated by four $\Delta\varphi$ points, as shown in Figure 2.

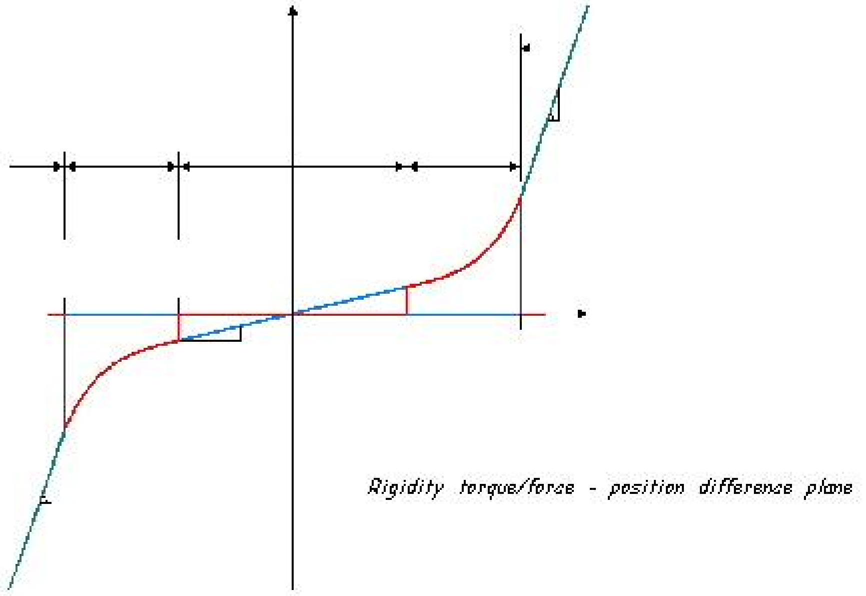


Figure 2. The 5 sub-areas of the rigidity torque - the position difference plane

The backlash area (BKL) is limited by the upper and lower backlash position parameters DPHIBKLU and DPHIBKLL. The linear rigidity CBKL is valid inside this area.

The normal areas, which follow into the backlash area, are limited by the parameters DPHISTPL and DPHIBKLL (to the left) and DPHIBKLU and DPHISTPU (to the right). The assigned rigidity behavior is valid only inside these areas, so the linear, progressive, degressive, or polynomial rigidity is only calculated within these areas.

The 2nd step rigidity areas, which follow into the normal areas, are limited by the upper and lower 2nd step position parameters DPHISTPU and DPHISTPL. The linear rigidity CSTP is valid inside this area.

Linear Rigidity Behavior

If linear rigidity coefficient $C > 0$, the component calculates the linear rigidity torque MSTF [Nm] in the normal area: $MSTF = C \cdot \Delta\varphi$. For $C = 0$, the rigidity component transfers only friction torques between the predecessor and successor body in the normal area.

Figure 3 shows the typical linear rigidity characteristic and the time function waveforms of $\Delta\varphi$ and MSTF/FSTF.

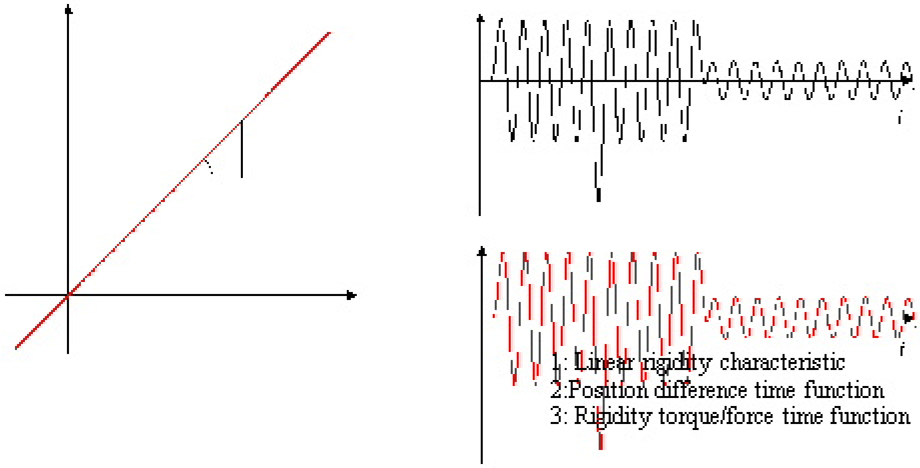


Figure 3. The typical linear rigidity characteristic and the time function waveforms of $\Delta\varphi$ and MSTF/FSTF

Progressive Rigidity Behavior

If the Progressive Rigidity Coefficient CPGR > 0, the component calculates a progressive rigidity torque MSTFPGR [Nm] in the normal area. The linear rigidity C is the initial rigidity at $\Delta\varphi = 0$. The property is typical for parts that become stiffer and stiffer with increasing torsion/stress (e.g., couplings).

$MSTFPGR = C/CPGR \cdot \sinh(CPGR \cdot \Delta\varphi)$

Figure 4 shows the typical CPGR characteristic and the time functions of MSTFPGR(t), $\Delta\varphi(t)$ and $\Delta\omega(t)$. Notice that the characteristic of CPGR is different from a polynomial.

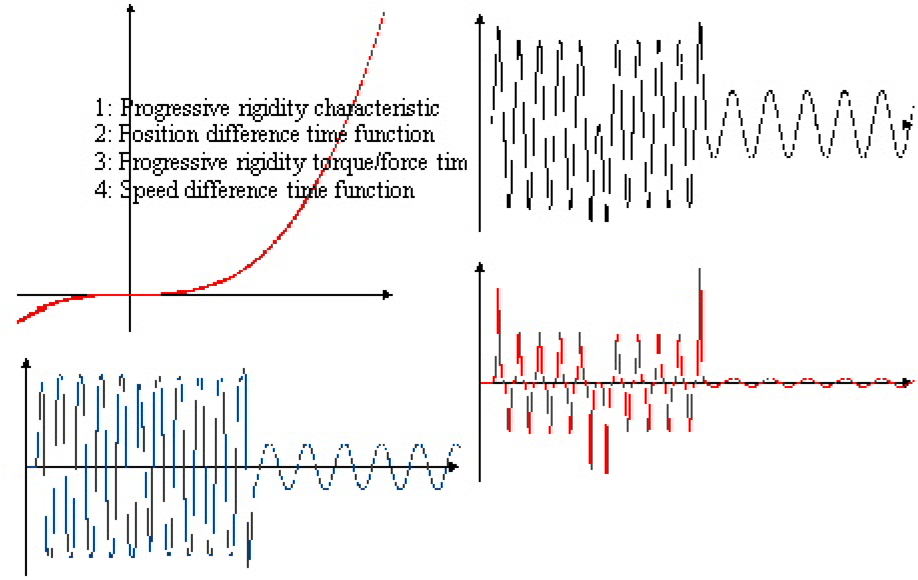


Figure 4. The typical characteristic of CPGR and the time function waveforms of MSTFPGR, Δφ and Δω

Degressive Rigidity Behavior

If the degressive rigidity coefficient CDGR > 0, the component calculates a degressive rigidity torque MSTFDGR [Nm] in the normal area.

$\mathrm{MSTFDGR} = C/\mathrm{CDGR} \cdot \operatorname{ashinh}(\mathrm{CDGR} \cdot \Delta\varphi)$

The linear rigidity C is the initial rigidity at $\Delta\varphi = 0$, and $0 \leq C < \infty$. The property is typical for parts (e.g. couplings) becoming softer and softer with increasing torsion/stress.

Figure 5 shows the typical CDGR characteristic and the time functions of Δφ(t), MSTFDGR(t)/ FSTFDGR(t) and Δω(t).

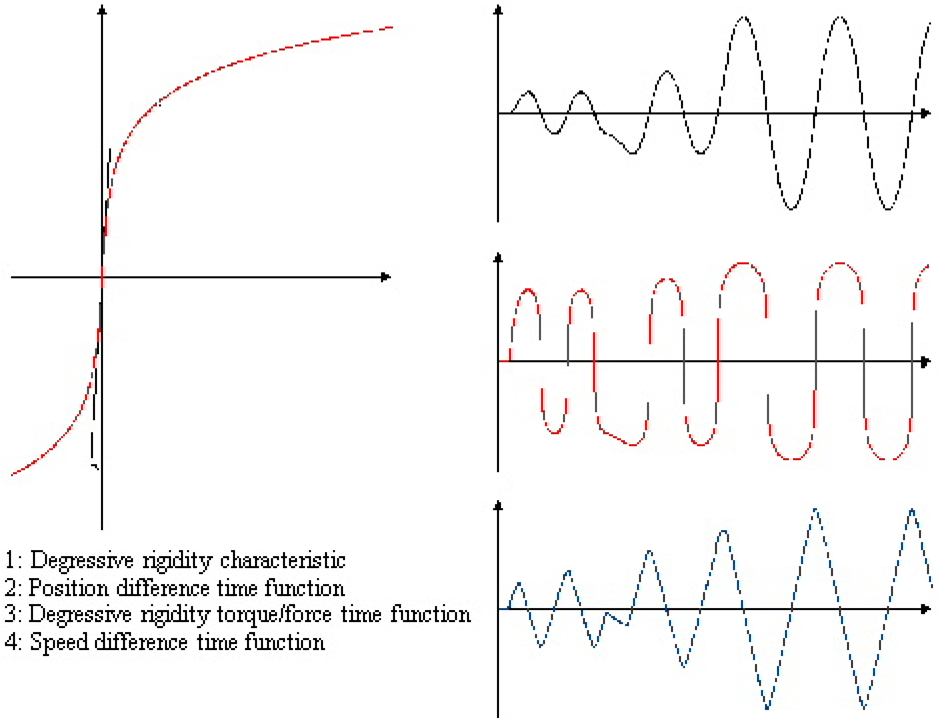


Figure 5. The typical CDGR characteristic and the time functions of Δφ(t), MSTFDGR(t)/ FSTFDGR (t) and Δω(t)

Polynomial Rigidity Behavior

If the Polynomial Rigidity Multiplier CMUL > 0, the component calculates a polynomial rigidity torque MSTFPOL [Nm] in the normal area.

$\mathrm{MSTFPOL} = C \cdot \Delta\varphi + \mathrm{CMUL} \cdot |\Delta\varphi|^{\mathrm{CEXP}} \cdot \operatorname{sgn}(\Delta\varphi)$

where the linear rigidity C is the initial rigidity at $\Delta\varphi = 0$, and $0 \le C < \infty$. $0 \le CMUL < \infty$, and the Polynomial Rigidity Exponent CEXP $\in (-\infty, \infty)$. The property is suitable to substitute for the progressive approach.

Figure 6 shows the typical polynomial rigidity characteristic and the time functions of $\Delta\varphi(t)$, MSTFPOL (t)/ FSTFPOL (t) and $\Delta\omega(t)$.

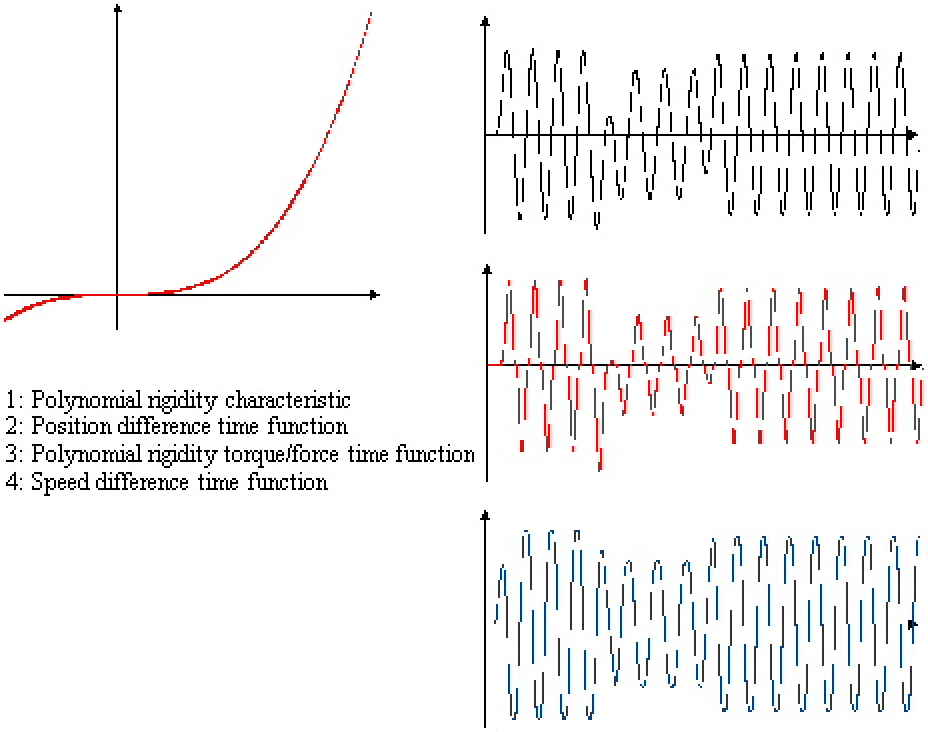


Figure 6. The typical polynomial rigidity characteristic and the time functions of $\Delta\varphi$(t), MSTFPOL (t)/ FSTFPOL (t) and $\Delta\omega$(t)

Characteristic Rigidity (Stiffness)

If the Characteristic rigidity torque MSTFCH $\neq 0$, the component calculates an arbitrary, weighted, position difference dependent rigidity torque MSTFCH = f($\Delta\varphi$(t)), which can be specified with a characteristic (preferably a 2D Lookup Table characteristic).

$$MSTFCH\_VAL = KSTFCH \cdot MSTFCH(\Delta\varphi(t))$$

where KSTFCH is the weighting coefficient.

Rigidity Component Friction/Damping Properties

Viscous Friction

If the Viscous friction coefficient KVSC **>** 0, the component calculates a linear speed difference dependent friction resistive torque MVSC [Nm].

$\mathrm{MVSC} = \mathrm{KVSC} \cdot \omega(t)$

Where KVSC is the viscous friction coefficient, and $0 \leq \mathrm{KVSC} < \infty$.

Progressive Friction

If the Progressivity friction coefficient KPGR > 0, the component calculates a progressive friction torque MFRCPGR [Nm]. The property is typical for parts (e.g., couplings) that become stiffer and stiffer with increasing torsion/stress. The linear KVSC is the initial friction coefficient at $\Delta\omega = 0$.

$\mathrm{MFRCPGR} = (\mathrm{KVSC}/\mathrm{KPGR}) \cdot \sinh(\mathrm{KPGR}(\Delta\omega(t)))$

where KVSC is theviscous friction coefficient, and $0 \leq \mathrm{KVSC} <\infty$.

Degressive Friction

If the Degressivity Friction Coefficient KDGR > 0, the component calculates a degressive friction torque MFRCDGR [Nm]. The property is typical for parts (e.g., couplings) that become softer and softer with increasing torsion/stress. The linear KVSC is the initial friction coefficient at $\Delta\omega =$ 0..

$\mathrm{MFRCPDR} = (\mathrm{KVSC}/\mathrm{KDGR}) \cdot \sinh(\mathrm{KDGR}(\Delta\omega(t)))$

Polynomial Friction

If the Polynomial Friction Multiplier KMUL > 0, the component calculates a speed-dependent polynomial resistive torque MFRCPOL [Nm]. This property is typical for frictions with streaming media (e.g., windage).

$\mathrm{MFRCPOL} = \mathrm{KVSC} \cdot \Delta\omega + \mathrm{KUMU} \cdot |\Delta\omega|^{\mathrm{KEXP}} \mathrm{sgn}(\Delta\omega)$

where KEXP is the Polynomial Friction Exponent, and $\mathrm{KEXP} \in (-\infty, \infty)$.

Characteristic Friction

If the Characteristic Friction Torque $\mathrm{MFRCCH} \neq 0$, the component calculates an arbitrary, weighted, speed dependent resistive torque $\mathrm{MFRCCH} = f(\omega(t))$, which can be specified with a characteristic (preferably a 2D Lookup Table characteristic).

$\mathrm{MFRCCH\_VAL} = \mathrm{KFRCCH} \cdot \mathrm{MFRCCH}(\omega(t))$

where KFRCCH is the weighting coefficient.

Rigidity Component Position Dependent Friction/Damping Properties

If the Position Dependant Polynomial Friction Multiplier KVSCMUL > 0, the component calculates a polynomial position difference or Position Dependent Viscous Friction Coefficient KVSCPOL = f($\Delta\varphi$)[Nms/rad], and the resulting viscous friction coefficient KVSCRES [Nms/rad]. The position dependence of friction and damping is accomplished by the position dependence of the viscous friction coefficient. This holds for the backlash area and the 2nd step area.

See also Backlash Property

See also Second Step Property

A polynomial approach for KVSC = f($\Delta\varphi$) can be specified. This property is typical for changing friction relations depending on position. The linear KVSC is the offset. KVSCRES is computed by superimposing KVSC, KVSCBKL, KVSCSTP, and KVSCPOL.

See also 2nd Step Friction/Damping Properties

$\mathrm{KVSCPOL} = \mathrm{KVSCMUL} \cdot |\Delta\varphi(t)|^{\mathrm{KVSCEXP}}$, $\mathrm{KVSCRES} = \mathrm{KVSC} + \mathrm{KVSCPOL}$

where KVSCEXP is the Position Dependant Polynomial Friction exponent, and $\in (-\infty, \infty)$.

Backlash Property

The backlash property is useful for specifying an area (backlash area) with linear rigidity and friction properties that might differ from the ones in the normal area. The property is typical for gear backlash, loose parts, etc.

If the Backlash Lower Limit of Position Angle Difference DPHIBKLL < 0 or/and the Backlash Upper Limit of Position Angle Difference DPHIBKLL > 0, then the backlash property is calculated. To reach the backlash limits, the simulator reduces the simulation time step to HMIN. However, if the backlash range given by (DPHIBKLL, DPHIBKLU) is rather small, it might be skipped due to a comparatively large minimum time step HMIN. If this occurs, simulation continues without computing the backlash property at this instant, and a simulator message appears with the model name, component name, and time value. All backlash limits are dynamic parameters (i.e., they can be changed during simulation). However, if the conditions are violated, the parameters stay at their permissible limits, simulation continues, and an appropriate simulator message appears.

Backlash Rigidity Properties

If the Backlash Rigidity (inside backlash range) CBKL > 0, the component calculates a linear rigidity torque MSTFBKL [Nm] in the backlash area. In most cases, the backlash rigidity tends to be zero. A residual rigidity, however, can be specified using CBKL. This parameter is useful in cases where stepped spring packages are modeled.

$\mathrm{MSTFBKL} = \mathrm{CBKL} \cdot \Delta\varphi$

where CBKL is the Backlash Rigidity Coefficient, and $\mathrm{CBKL} \geq 0$.

Backlash Friction/Damping Properties

If the Backlash Viscous Friction Coefficient $KBKL \neq KVSC$, the component calculates the resulting Viscous Friction Coefficient for Backlash KVSCBKL [Nms/rad]. Outside the backlash area (in the normal area), KVSC is used; inside the backlash area, KBKL is used. $KBKL > 0$ allows you to model a residual attenuation, which might result from lubricants.

$KVSCBKL = KBKL$, if $DPHIBKLL < \Delta\varphi < DPHIBKLU$; otherwise $KVSCBKL = KVSC$.

Smooth KVSC Transition Between Backlash Area and Normal Area

On the transition from backlash to normal area and vice versa, an abrupt change between KVSC and KBKL occurs. Specifying the Backlash Viscous Friction Exponent KBKLEXP and Backlash Viscous Friction Conditioner KBKLCND smoothes this transition.

If KBKLEXP > 0,, the component calculates the resulting viscous friction coefficients for backlash and normal area KVSCBKL [Nms/rad] and MFRCBKL [Nm]. The friction torque results from KVSCBKL. KBKL > 0 and allows you to model a residual attenuation, which might result from lubricants.

For a small KBKLEXP (e.g., KBKLEXP = 2) ,the transition is rather soft; whereas, for a higher one (e.g., KBKLEXP = 32), the transition is harder. KBKLCND affects the position of the medium transition from KVSC to KBKL (i.e., how well DPHIBKLU and DPHIBKLL are hit).

$KVSCBKL = KVSC - (KVSC - KBKL)EXP(-\delta^{KBKLEXP} \cdot \varphi^{KBKLEXP})$

$MFRCBKL = KVSCBKL \cdot \Delta\omega(t)$

where $\delta = 1/(\sqrt{KBKLCND} \cdot (\varepsilon/2))$, $\varphi = \Delta\varphi - (DPHIBKLU - \varepsilon/2)$ and $\varepsilon = DPHIBKLU - DPHIBKLL$.

Second Step

The 2nd step property is useful for specifying an area (2nd step area) with linear rigidity and friction properties that might differ from the ones in the normal area or in the backlash area. Using the backlash parameters, you can model stepped spring packages with up to five steps (four switching points) with equal rigidity and friction properties in the individual areas. The property is typical for Stepped spring packages. If the Upper Position Angle Difference of the 2nd step area DPHISTPL < 0 and/or the Lower Position Angle Difference of the 2nd step area DHISTPU > 0, the property is calculated.

To reach the 2nd step limits, the simulator reduces the simulation time step to HMIN. However, if the 2nd step range given by (DPHISTPL, DPHISTPU) is rather small, it might be skipped due to a rather large minimum time step HMIN. If this occurs, simulation continues without computing the 2nd step property at this instant, and a simulator message appears with the model name, component name, and time value. All 2nd step limits are dynamic parameters (i.e., they can be changed during simulation. However, if the conditions are violated, the parameters stay at their permissible limits, simulation continues, and an appropriate simulator message appears.

2nd Step Rigidity Properties

If the 2nd Step Rigidity (outside step range) CSTP $>$ 0, the component calculates a linear rigidity torque MSTFSTP [Nm] in the 2nd step area. This parameter is useful in cases where stepped spring packages are modeled.

$\mathrm{MSTFSTP} = \mathrm{CSTP} \cdot \Delta\varphi$

2nd Step Friction/Damping Properties

If the 2nd Step Viscous Friction Coefficient KSTP $\neq$ KVSC, the component calculates the resulting Viscous Friction Coefficient KVSCSTP [Nms/rad] in the 2nd step area. Outside the backlash area (in the normal area), KVSC is used; inside the backlash area, the friction coefficient KSTP is used.

KVSCSTP = KVSC, if $\mathrm{DPHISTPL} < \Delta\varphi < \mathrm{DPHISTPU}$; otherwise KVSCSTP = KSTP.

Smooth KVSC-Transition Between 2nd Step Area and Normal Area

On the transition from the 2nd step are to the normal area and vice versa, an abrupt change occurs between KVSC and KSTP. Specifying the 2nd Step Viscous Friction Exponent KSTPEXP and the 2nd Step Viscous Friction Conditioner KSTPCND smoothes this transition (analogous to backlash).

If KSTPEXP $>$ 0, the component calculates the resulting viscous friction coefficients for the 2nd step and normal areas, KVSCSTP [Nms/rad] and MFRCSTP [Nm]. The friction torque results from KVSCSTP.

For a small KSTPEXP (e.g., KSTPEXP = 2), the transition is rather soft; whereas, for a higher one (e.g., KSTPEXP = 32), the transition is harder. KSTPCND affects the position of the medium transition from KVSC to KSTP (i.e., how well DPHISTPU and DPHISTPL are hit).

In cases where KSTPEXP $>$ 0 and KBKLEXP $>$ 0, and the corresponding exponents are rather small (very soft transition), and the 2nd step area and backlash area are close together (small normal area), then the resulting friction coefficient KVSCRES is a result of superimposing (averaging) KVSC, KVSCBKL, KVSCSTP, and KVSCPOL (if the polynomial viscous friction coefficient is additionally specified).

$\mathrm{KVSCSTP} = \mathrm{KSTP} - (\mathrm{KSTP} - \mathrm{KVSC}) \cdot \mathrm{EXP}(-\delta^{\mathrm{KSTPEXP}} \cdot \varphi^{\mathrm{KSTPEXP}})$

$\mathrm{MFRCSTP} = \mathrm{KVSCSTP} \cdot \Delta\omega(t)$

where $\delta = 1/(\sqrt{KSTPCND} \cdot (\varepsilon/2))$, $\varphi = \Delta\varphi - (\mathrm{DPHISTPU} - \varepsilon/2)$ and $\varepsilon = \mathrm{DPHISTPU} - \mathrm{DPHISTPL}$.

Rigidity Component External Difference Torque and Initial Values

External Difference Torque

If MEXT $\neq$ 0, the component calculates an arbitrary, weighted, externally computed resistive torque (time function) MEXT = f(t), which can be any arbitrary Twin Builder quantity (e.g., the result of a formula, a block output quantity of the block diagram module, a state graph quantity, etc.).

$MEXT\_VAL = KEXT \cdot MEXT(t)$

where KEXT is the weighting coefficient.

Angular Position Difference Initial Value

The Angular Position Difference Initial Value PHI0 can be specified to define another starting point other than zero for pre-stretched springs.

$PHI0 = \varphi\,(t = 0)$

The difference between the PHI0 of the predecessor and successor mass components, including possible transformation coefficients, must be set manually.

Top

Netlist Syntax

UMODEL VM_ROT VM_ROT_Pre_?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %0, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_ROT VM_ROT_Suc_@?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %1, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchRStf ?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %0, ROT2 := %1 (C:=@C ,KVSC:=@KVSC ,CMUL:=@CMUL ,CEXP:=@CEXP ,KMUL:=@KMUL ,KEXP:=@KEXP ,KVSCMUL:=@KVSCMUL ,KVSCEXP:=@KVSCEXP ,CPGR:=@CPGR ,KPGR:=@KPGR ,CDGR:=@CDGR ,KDGR:=@KDGR ,DPHIBKLU:=@DPHIBKLU ,DPHIBKLL:=@DPHIBKLL ,CBKL:=@CBKL ,KBKL:=@KBKL ,KBKLEXP:=@KBKLEXP ,KBKLCND:=@KBKLCND ,DPHISTPU:=@DPHISTPU ,DPHISTPL:=@DPHISTPL ,CSTP:=@CSTP ,KSTP:=@KSTP ,KSTPEXP:=@KSTPEXP ,KSTPCND:=@KSTPCND ,MSTFCH:=@MSTFCH ,KSTFCH:=@KSTFCH ,MFRCCH:=@MFRCCH ,KFRCCH:=@KFRCCH ,MEXT:=@MEXT ,KEXT:=@KEXT ,DPHI0:=@DPHI0) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
ROT1 (See Note)	Predecessor Node	Rotational
ROT2 (See Note)	Successor Node	Rotational

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
C	Rigidity (stiffness)	real	0 [Nms/rad]
KVSC	Viscous friction coefficient	real	0 [Nms/rad]
CMUL	Polynomial rigidity multiplier	real	0 [Nms/rad]
CEXP	Polynomial rigidity exponent	real	0
KMUL	Polynomial friction multiplier	real	0 [Nms/rad]
KEXP	Polynomial friction exponent	real	0
KVSCMUL	Position dept polynomial friction multiplier	real	0 [Nms/rad]
KVSCEXP	Position dept polynomial friction exponent	real	0
CPGR	Progressivity rigidity coefficient	real	0
KPGR	Progressivity friction coefficient	real	0
CDGR	Degressivity rigidity coefficient	real	0
KDGR	Degressivity friction coefficient	real	0
DPHIBKLU	Backlash upper limit of position angle difference	real	0 [rad]
DPHIBKLL	Backlash lower limit of position angle diff	real	0 [rad]
CBKL	Backlash rigidity (inside backlash range)	real	0 [Nm/rad]
KBKL	Backlash viscous friction coefficient	real	0 [Nms/rad]
KBKLEXP	Backlash viscous friction exponent	real	0
KBKLCND	Backlash viscous friction conditioner	real	1
DPHISTPU	2nd step upper limit of position angle diff	real	0 [rad]
DPHISTPL	2nd step lower limit of position angle diff	real	0 [rad]
CSTP	2nd step rigidity (outside step range)	real	0 [Nm/rad]

KSTP	2nd step viscous friction coefficient	real	0 [Nms/rad]
KSTPEXP	2nd step viscous friction exponent	real	0
KSTPCND	2nd step viscous friction conditioner	real	1
MSTFCH	Characteristic rigidity torque (m = f(dphi))	real	0 [Nm]
KSTFCH	Characteristic rigidity torque coefficient	real	1
MFRCCH	Characteristic friction torque (m = f(dOMEGA))	real	0 [Nm]
KFRCCH	Characteristic friction torque coefficient	real	1
MEXT	External (arbitrary) torque (m = f(t))	real	0 [Nm]
KEXT	External (arbitrary) torque coefficient	real	1
DPHI0	Initial angular position difference	real	0 [rad]

Top

Input/Output Quantities

Table3

Name	**Description [Unit]**	**Direction**	**Data Type**
dDALPHA	Angular jerk difference [rad/s^3]	Output	real
DALPHA	Angular acceleration difference [rad/s^2]	Output	real
DOMEGA	Angular velocity difference [rad/s]	Output	real
DPHI	Angular position difference [rad]	Output	real
DM	Difference torque [Nm]	Output	real
OMEGA1	Predecessor angular velocity [rad/s]	Output	real
OMEGA2	Successor angular velocity [rad/s]	Output	real
MSTF	Resulting rigidity torque [Nm]	Output	real
MFRC	Resulting friction torque [Nm]	Output	real
MVSC	Viscous friction torque [Nm]	Output	real
KVSCRES	Resulting viscous friction coefficient [Nms/rad]	Output	real
MSTFPOL	Polynomial rigidity torque [Nm]	Output	real
MFRCPOL	Polynomial friction torque [Nm]	Output	real
KVSCPOL	Polynomial viscous friction coefficient [Nms/rad]	Output	real
MSTFPGR	Progressive rigidity torque [Nm]	Output	real
MFRCPGR	Progressive friction torque [Nm]	Output	real
MSTFDGR	Degressive rigidity torque [Nm]	Output	real
MFRCDGR	Degressive friction torque [Nm]	Output	real
MSTFBKL	Backlash rigidity torque [Nm]	Output	real
KVSCBKL	Backlash viscous friction coefficient [Nms/rad]	Output	real
MFRCBKL	Backlash friction torque [Nm]	Output	real
MSTFSTP	2nd step rigidity torque [Nm]	Output	real
KVSCSTP	2nd step viscous friction coefficient [Nms/rad]	Output	real

MFRCSTP	2nd step viscous friction torque [Nm]	Output	real
MFRCCLB	Coulomb friction torque [Nm]	Output	real
MSTFCH_VAL	Characteristic rigidity torque [Nm]	Output	real
MFRCCH_VAL	Characteristic friction torque [Nm]	Output	real
MEXT_VAL	External (arbitrary) torque [Nm]	Output	real

Top

Example

This example shows a one-body vibrator with stepped spring package with polynomial rigidity and “hard” viscous friction, as shown in Figure 7. System parameters are listed in Table 4. The torque output of the Torque Source Srcf1 is decided by a 2D-Lookup Table m_Ext, as shown in Figure 8. The simulation results, the Resulting Rigidity Torque MSTF, 2nd Step Rigidity Torque MSTFSTP and the Resulting Friction Torque MFRC are shown in Figure 9, 10, and 11, respectively.

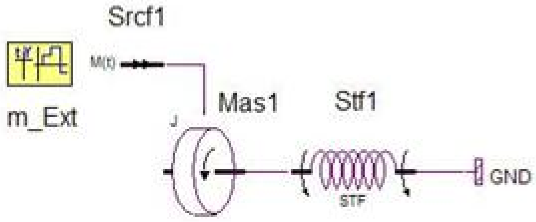


Figure 7. Application example of the Rigidity (rotational) model

Table 4

Component	Parameter	Value [Unit]

Rigidity (Rotational) Stf1	C	1k [Nms/rad]

	KVSC	0.25 [Nms/rad]
	CMUL	0.1G [Nms/rad]
	CEXP	2
	KMUL	0 [Nms/rad]
	KEXP	0
	KVSCMUL	0 [Nms/rad]
	KVSCEXP	0
	CPGR	0
	KPGR	0
	CDGR	0
	KDGR	0
	DPHIBKLU	10u [rad]
	DPHIBKLL	-10u [rad]
	CBKL	1k [Nm/rad]
	KBKL	25m [Nms/rad]
	KBKLEXP	0
	KBKLCND	1
	DPHISTPU	0.2m [rad]
	DPHISTPL	-0.2m [rad]
	CSTP	40k [Nm/rad]
	KSTP	0.5 [Nms/rad]
	KSTPEXP	0
	KSTPCND	1
	MSTFCH	0 [Nm]
	KSTFCH	1
	MFRCCH	0 [Nm]
	KFRCCH	1
	MEXT	0 [Nm]
	KEXT	1
	DPHI0	0 [rad]

Mass (Rotational) Mas1	J	0.2 [kg m^2]
	KVSC	0 [Nms/rad]
	KMUL	0 [Nms/rad]
	KEXP	0
	MCLB	0 [Nm]
	OMEGACLB	1m [rad/s]
	OMEGASTK	0 [rad/s]
	MSTK	0 [Nm]
	KSTK	1e16 [Nms/rad]
	MFRCCH	0 [Nm]
	KFRCCH	1
	MEXT	0 [Nm]
	KEXT	1
	OMEGA0	0 [rad/s]
	PHI0	0 [rad]
	PHIU	0 [rad]
	PHIL	0 [rad]
Torque Source Srcf1	M1	m_Ext.VAL
	FACT	10

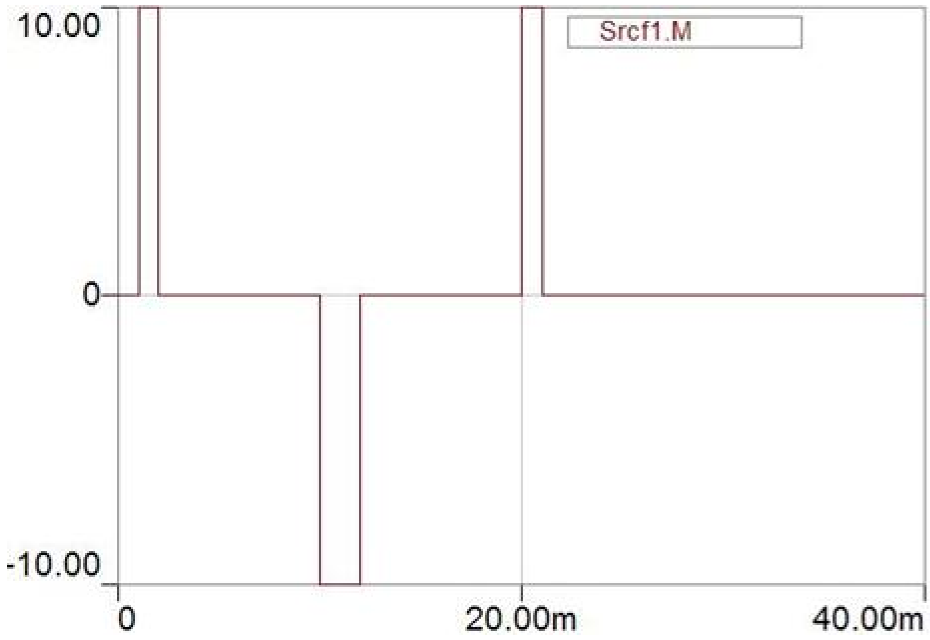


Figure 8. The output of the Torque Source Srcf1

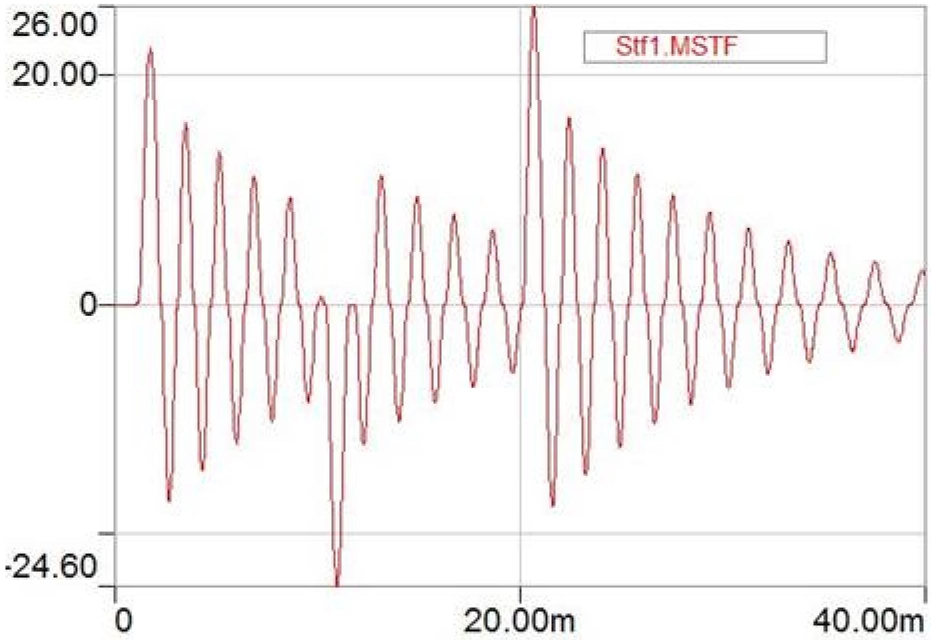


Figure 9. Simulation results -- the Resulting Rigidity Torque MSTF

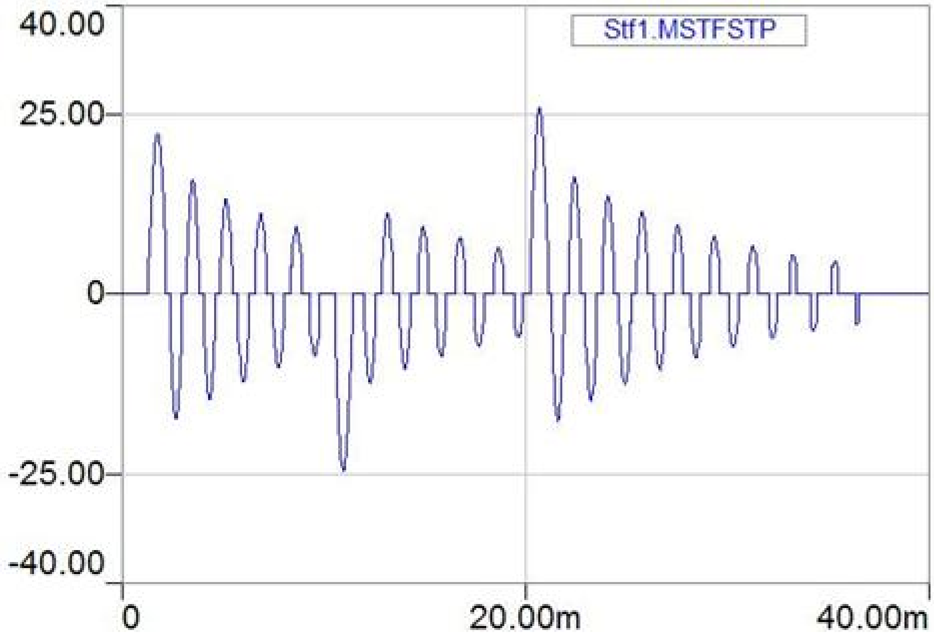


Figure 10. Simulation results – 2nd Step Rigidity Torque MSTFSTP

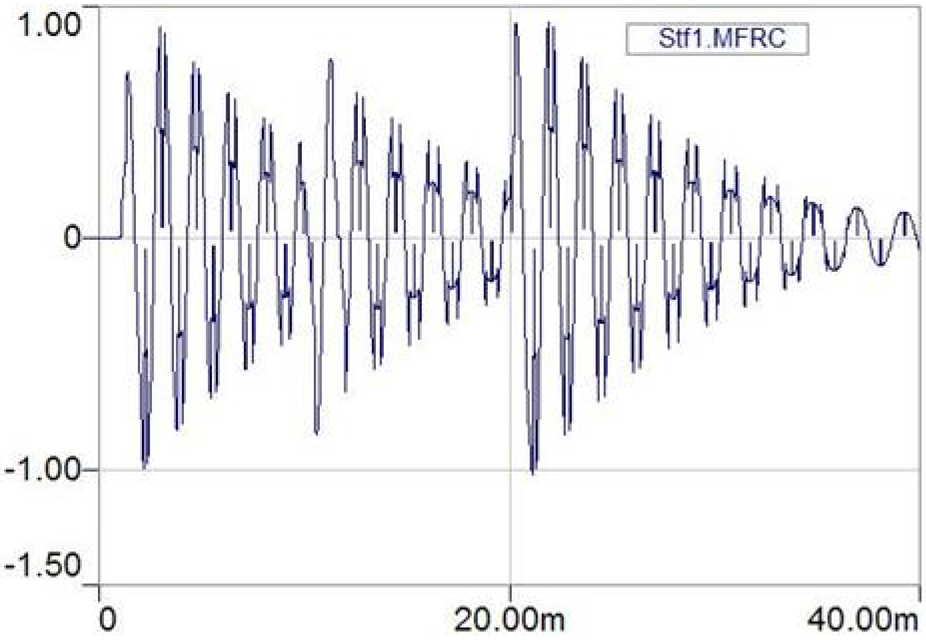


Figure 11. Simulation results – the Resulting Friction Torque MFRC

Top

References

Translational_V Components

- Ground -translational (MchTGnd)
- Mass - translational (MchTMas)
- Force Source - translational (MchTSrcf)
- Velocity Source - translational (MchTSrcv)
- Rigidity - translational (MchTStf)

Ground Component (Translational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

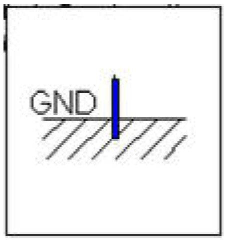


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Input/Output Quantities
- Example
- References

Description

This component enables the connection of a rigidity component (shaft section) with ground to transmit a force into the foundation. The force F[N], which is received by the ground, is provided as an output.

Top

Assumptions and Limitations

Top

Mathematical Description

Top

Netlist Syntax

UMODEL VM_TR VM_TR_Gnd_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchTGnd ?InstanceName(@InstanceName):(@(Refbase)@(ID)) GND := %0, Nul :=

GND ( ) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
GND (See Note)	Ground Node (for connection to a rigidity component)	Translational_V

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Input/Output Quantities

Table2

Name	Description [Unit]	Direction	Data Type
F	Force output [N]	Output	real

Top

Example

Top

References

Mass Component (Translational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

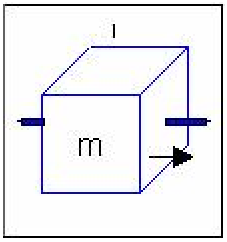


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

The mass component concentrates all inertial properties of a mechanical n-body system section. All motion quantities are absolute quantities referring to ground.

Top

Assumptions and Limitations

The mass component can only be connected to rigidity or force source components. If the mass component is connected to a coordinate transformation component, the component beyond the coordinate transformation component needs to be either a rigidity source or a force source. Two mass components cannot be connected directly. Several rigidity components, however, can be connected to a mass component.

Note:

Except for the moment of inertia J, all properties and corresponding parameters are optional. The equations are not calculated unless the property has been activated.

Top

Mathematical Description

- Inertia
- Friction/Damping properties
- Limits and Initial Values

Inertia

The parameter M [kg] specifies the concentrated inertia of the mass component. For more information, refer to Inertias.

Friction Properties

General information on Friction property can be found at Friction/Damping properties.

Mass Component Viscous Friction

If Viscous Friction Coefficient KVSC [Nms/rad] $>$ 0, the component calculates the linear speed dependent friction resistive torque MVSC [Nm], as follows,

$\mathrm{MVSC} = \mathrm{KVSC} \cdot \omega(t),\ 0 \le \mathrm{KVSC} < \infty$

The linear characteristic and typical time function (exponential envelope) of KVSC and MVSC are shown in Figure 2.

If Viscous Friction Coefficient KVSC [Nms/rad] $>$ 0, the component calculates the linear speed dependent Friction Resistive Force FVSC [N].

$\mathrm{FVSC} = \mathrm{KVSC} \cdot v(t),\ 0 \le \mathrm{KVSC} < \infty$

The linear characteristic and typical time function (exponential envelope) of KVSC and FVSC are shown in Figure 2.

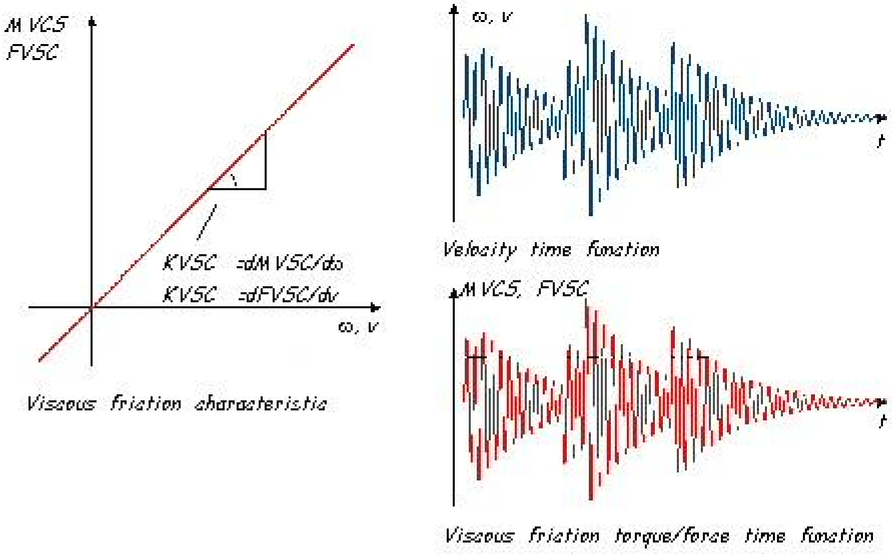


Figure 2. The linear characteristic and time function (exponential envelope) of KVSC and FVSC

Mass Component Coulomb Friction

If Constant Coulomb Friction Force FCLB $>$ 0, the component calculates a constantresistive forceFFRCCLB [N], which is independent of the speed, but dependent on the direction of the speed. This property is typical for dry bearing friction.

If $|v(t)| \geq \mathrm{VCLB}$, then $\mathrm{FFRCCLB} = \mathrm{FCLB} \cdot \mathrm{sign}(v(t))$

If $|\omega(t)| < \mathrm{VCLB}$, then $\mathrm{FFRCCLB} = \mathrm{FCLB} / \mathrm{VCLB} \cdot \mathrm{sign}(v(t))$

where VCLB [m/s] is the velocity at which constant coulomb friction become active.

The constant friction characteristic and typical linear time function of FFRCCLB are shown in Figure 3.

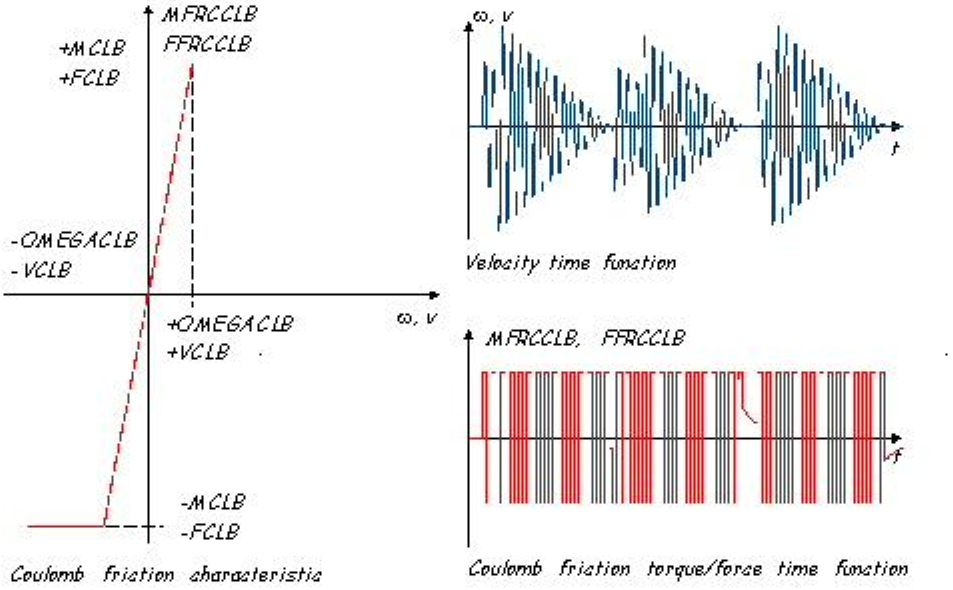


Figure 3. The constant friction characteristic and typical linear time function of v and FFRCLB

The simulator reduces the simulation time step to reach the limits of the viscous range given by $|\omega(t)| < VCLB$. In case it is rather small and the minimum time step is rather large, such that it has not been hit, then simulation continues, and a simulator message appears with the model name, component name, and time instant.

Mass Component Polynomial Friction

If the Polynomial Friction Multiplier KMUL < 0, the component calculates a speed-dependent polynomial resistive force FFRCPOL [Nm]. This property is typical for frictions with streaming media (e.g., windage).

$FFRCPOL = KMUL \cdot |v(t)|^{KEXP} \cdot sign(v(t))$

The characteristic and typical time function of v and FFRCPOL are shown in Figure 4. Notice that the first quadrant is always evoked into the third quadrant no matter what KEXP is.

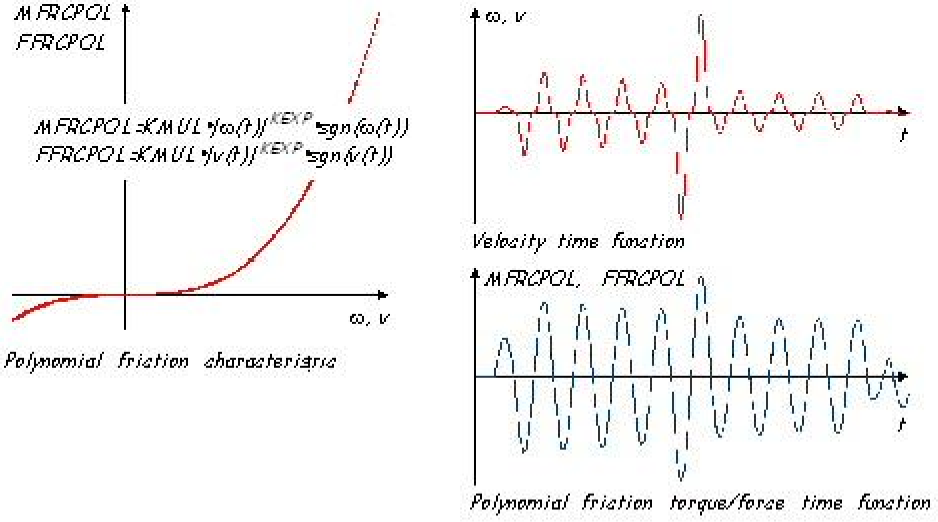
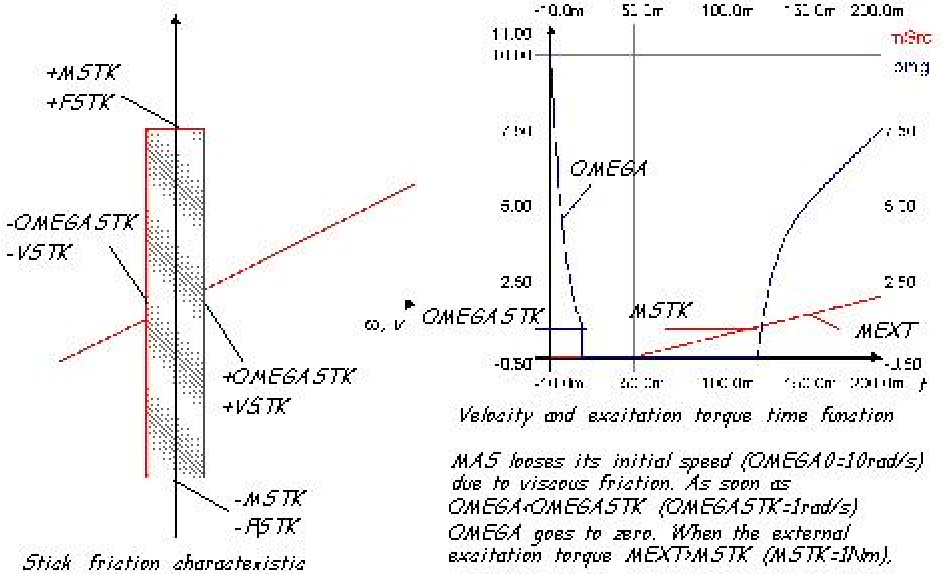


Figure 4. The characteristic and typical time function of v and FFRCPOL

Mass Component Stick Friction (“Stiction”)

If Stick Velocity VSTK > 0, the component calculates the behavior where a mass gets stuck (stick friction) at very low speeds. As soon as the speed falls below a certain limit VSTK, the mass latches into stick friction state. Braking occurs with the viscous stick friction coefficient KSTK. The mass can only reaccelerate after the accelerating force exceeds the stick friction force ("break-off torque/force") FSTK.

The characteristic and typical time function of v and FSTK are shown in Figure 5. Notice that the Speed suddenly goes to zero while within the hatched area.



MAS looses its initial speed (OMEGA0=10rad/s) due to viscous friction. As soon as OMEGA<OMEGASTK (OMEGASTK=1rad/s) OMEGA goes to zero. When the external excitation torque MEXT>MSTK (MSTK=1Nm), MAS re-accelerates.

Figure 5. The characteristic and typical time function of v and FSTK

Usually VSTK is rather small. On the transition from positive to negative speeds or vice versa, the simulator reduces the simulation time step to reach the stick range given by (-VSTK, + VSTK) (hatched area). In case the required time step is smaller than the minimum time step HMIN, simulation continues without computing the stick property at this instant, and a simulator message appears with the model name, component name, and time value.

Mass Component Characteristic Friction

If VFRCCH $>$ 0, the component calculates an arbitrary, weighted, speed-dependent resistive force FFRCCH = f(v(t)), which can be specified with a characteristic (preferably a 2D Lookup Table characteristic).

FFRCCH_VAL = KFRCCH · FFRCCH (v(t))

where KFRCCH is the weighting coeffient.

Figure 6 shows an arbitrary friction characteristic defined by distinct points.

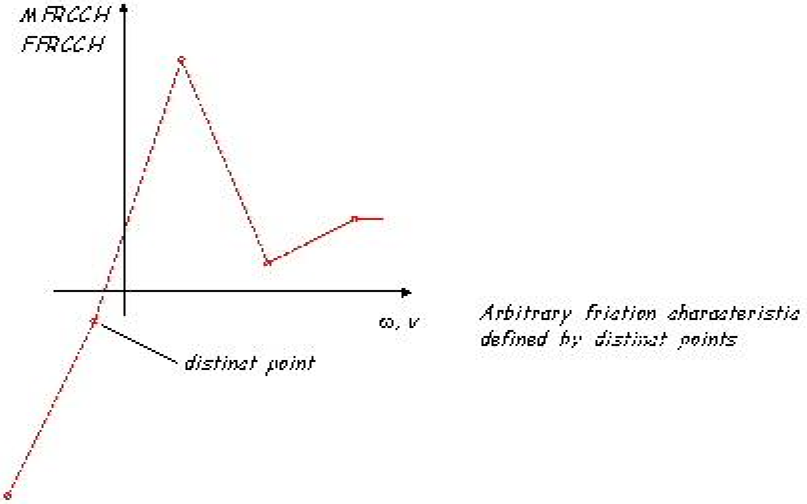


Figure 6. An arbitrary friction characteristic

Mass Component External Friction

If the External (arbitrary) Force FEXT $>$ 0, the component calculates an arbitrary, weighted, externally computed resistive force (time function) FEXT = f(t), which can be any arbitrary Twin Builder quantity (e.g., the result of a formula, a block output quantity of the block diagram module, or a state graph quantity).

FEXT_VAL = KEXT · FEXT (t)

where KEXT is the weighting coefficient.

Figure 7 shows an arbitrary characteristic function for the External (arbitrary) Force FEXT.

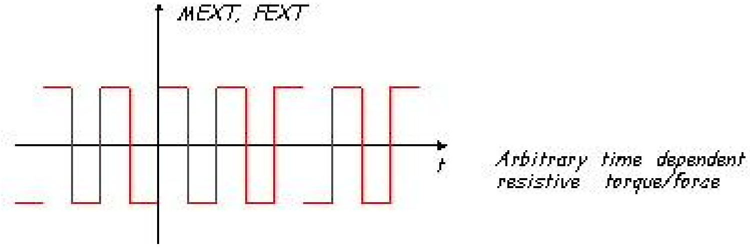


Figure 7. An arbitrary characteristic function for FEXT

Limits and Initial Values

Mass Component Upper and Lower Position Limit

If the Upper Position limit SU ≠ 0, the Lower Position limit SL ≠ 0 and SU > SL, the component considers the current position s(t). When the position exceeds the upper limit SU or s(t) falls below the lower limit SL, then this position is retained, and the angular velocity becomes zero (or enters stick state if stick friction is activated). The system can only be restarted after reversing the motion or exceeding break-off force FSTK. The limits are typical for driving a body against a "wall".

Figure 8 shows the typical time function of s(t) and v(t). Notice that as soon as s(t) reaches its upper or lower limit, v(t) suddenly goes to zero.

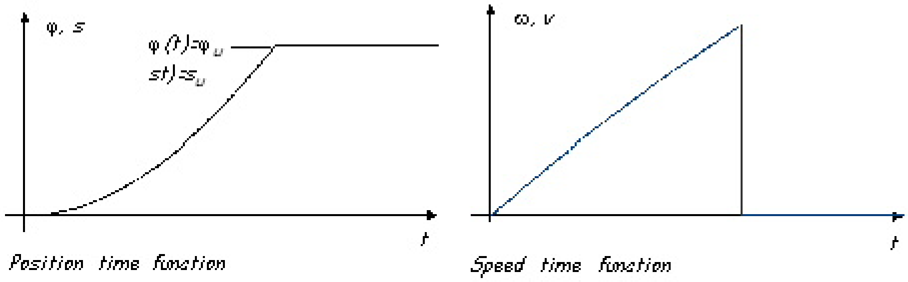


Figure 8. typical time function of φ(t) and ω(t)

On the transition from the linear position range to either the upper or lower limit, the simulator reduces the simulation time step to its minimum HMIN to reach the limit. If PHIL and PHIU are close together (the linear position range is rather small) and the required simulation time step is smaller than the minimum simulation time step HMIN, then a message appears during simulation, displaying the model name, component name, and time instant. The position limits are dynamic parameters (i.e., they can change during simulation). If the conditions are violated during simulation, the parameters stay at their permissible limits, the simulation continues, and a simulator message appears with the model name, component name, and time instant.

Mass Component Initial Velocity and Initial Position Value

The initial values for velocity v0 and position s0 can be specified to define the starting points; the default values are zero.

Top

Netlist Syntax

UMODEL VM_TR VM_TR_Src_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_TR VM_TR_Pre_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %1, TR2 := %2 () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_TR VM_TR_Suc_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %2, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchTMas ?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR := %0, TR1 := %1, TR2 := %2, GND := GND (M:=@M ,KVSC:=@KVSC ,KMUL:=@KMUL ,KEXP:=@KEXP ,FCLB:=@FCLB ,VCLB:=@VCLB ,VSTK:=@VSTK ,FSTK:=@FSTK ,KSTK:=@KSTK ,FFRCCH:=@FFRCCH ,KFRCCH:=@KFRCCH ,FEXT:=@FEXT ,KEXT:=@KEXT ,V0:=@V0 ,S0:=@S0 ,SU:=@SU ,SL:=@SL) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
TR (See Note)	Source Node	Mechanical
TR1 (See Note)	Predecessor Node	Mechanical
TR2 (See Note)	Successor Node	Mechanical

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
M	Mass, Inertia	real	1 [kg]
KVSC	Viscous Friction Coefficient	real	0 [Ns/m]
KMUL	Polynomial Friction Multiplier	real	0 [Ns/m]
KEXP	Polynomial Friction Exponent	real	0

FCLB	Constant Coulomb Friction Force	real	0 [N]
VCLB	Coulomb Velocity	real	1m [m/s]
VSTK	Stick Velocity	real	0 [m/s]
FSTK	Stick Force (break-off)	real	0 [N]
KSTK	Stick Friction Coefficient	real	1E16[Ns/m]
FFRCCH	Characteristic Friction Force	real	0 [N]
KFRCCH	Characteristic Friction Force Coefficient	real	1
FEXT	External (arbitrary) Force	real	0 [N]
KEXT	External (arbitrary) Force Coefficient	real	1
V0	Initial Velocity	real	0 [m/s]
S0	Initial Position	real	0 [m]
SU	Upper Position Limit	real	0 [m]
SL	Lower Position Limit	real	0 [m]

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
dACC	Jerk [m/s^3]	Output	real
ACC	Acceleration [m/s^2]	Output	real
V	Velocity [m/s]	Output	real
S	Position [m]	Output	real
F	Force supplied by source [N]	Output	real
F1	Force supplied by predecessor [N]	Output	real
F2	Force supplied to successor [N]	Output	real
FACX	Accelerating external forces [N];	Output	real
FACC	Accelerating force [N];	Output	real
FFRC	Resulting friction force [N]	Output	real

FVSC	Viscous friction force [N]	Output	real
FFRCPOL	Polynomial friction force [N]	Output	real
FFRCCLB	Coulomb friction force [N]	Output	real
FFRCCH_VAL	Characteristic friction force [N]	Output	real
FEXT_VAL	External (arbitrary) force [N]	Output	real

Top

Example

Top

References

Force Source (Translational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

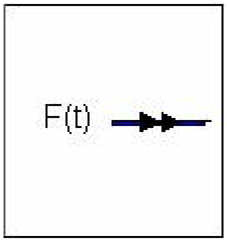


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

This component provides a force, which is calculated froma Twin Builder quantity, for a mechanical mass node. Thus, the mechanical system can use (system) external forces. The force and velocity are provided for output.

Top

Assumptions and Limitations

No internal inertia or losses are considered in this component. The Force Output of this component is proportional to the External Force Source at a ratio specified by the External Source Weighting Coefficient, and there is no output limits.

Top

Mathematical Description

$F = FACT \cdot F1$

Where F1 is the External Force, FACT is the External Source Weighting Coefficient, and F is the Force Output.

Top

Netlist Syntax

UMODEL VM_TR VM_TR_Src_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchTSrcf ?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR := %0, GND := GND (F1 := @F1 ,FACT := @FACT) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
TR (See Note)	Source Node (for connection to a rigidity component)	Mechanical

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
F1	Source velocity quantity	real	0 [N]
FACT	Source weighting factor	real	1 [rad/s]

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
F	Force Output [N]	Output	real
V	Source Velocity [m/s]	Output	real

Top

Example

Top

References

Velocity Source (Translational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

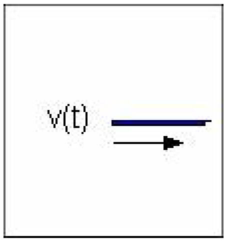


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

This component provides a velocity, which is calculated froma Twin Builder quantity, for a mechanical rigidity node. Thus, the mechanical system can use (system) external speeds. The velocity and force are provided for output.

Top

Assumptions and Limitations

No internal inertia or losses are considered in this component. The Velocity Output of this component is proportional to the External Velocity Source at a ratio specified by the External Source Weighting Coefficient, and there is no output limits.

Top

Mathematical Description

$V = FACT \cdot V1$

Where V1 is the External Velocity Source, FACT is the External Source Weighting Coefficient, and V is the Velocity Output.

Top

Netlist Syntax

UMODEL VM_TR VM_TR_Src_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchTSrcv ?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR := %0, GND := GND (V1:=@V1 ,FACT:=@FACT) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
TR (See Note)	Source Node (for connection to a rigidity component)	Mechanical

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
V1	Source velocity quantity	real	0 [m/s]
FACT	Source weighting factor	real	1

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
V	Source velocity [m/s]	Output	real
F	Force output [N]	Output	real

Top

Example

Top

References

Rigidity Component (Translational)

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

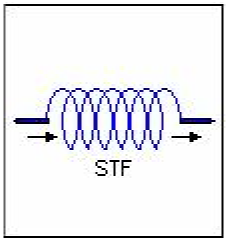


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

The rigidity component concentrates all spring properties of a mechanical n-body system section. All motion quantities are relative quantities and are referred to the difference between predecessor and successor mass.

Top

Assumptions and Limitations

The rigidity component can only be connected to a mass component, a velocity source, or a ground component. If the rigidity component is connected to a coordinate transformation component, make certain that the component beyond the coordinate transformation component is a mass component, a velocity source, or a ground component. You cannot connect two rigidity components directly or several mass components to one rigidity.

All torques resulting from the angular position differenceΔs [m] between predecessor and successor body are rigidity (stiffness) forces FSTF [N] with rigidity parameter C [N/m]: FSTF = f(t, C, Δs).

All forces resulting from the angular velocity difference Δv [m/s] between predecessor and successor body are friction forces FFRC [N] with parameter K [Ns/m]: FFRC = f(t, K, Δv).

The friction and damping parameter K can depend on the position difference K= f(Δs). The position dependent frictions result in FFRC = f(t, K, Δs, Δv).

Note:

All properties and corresponding parameters are optional. The equations are not simulated unless the property has been activated.

Top

Mathematical Description

- Rigidity properties
- Position dependent friction/damping properties
- Backlash property
- Second step property
- External difference torque and initial values

Rigidity Properties

Rigidity Force - Position Difference Plane Areas

The rigidity force - the position difference plane, FSTF = f(Δs), is subdivided into five areas, which are separated by four Δs points, as shown in Figure 2.

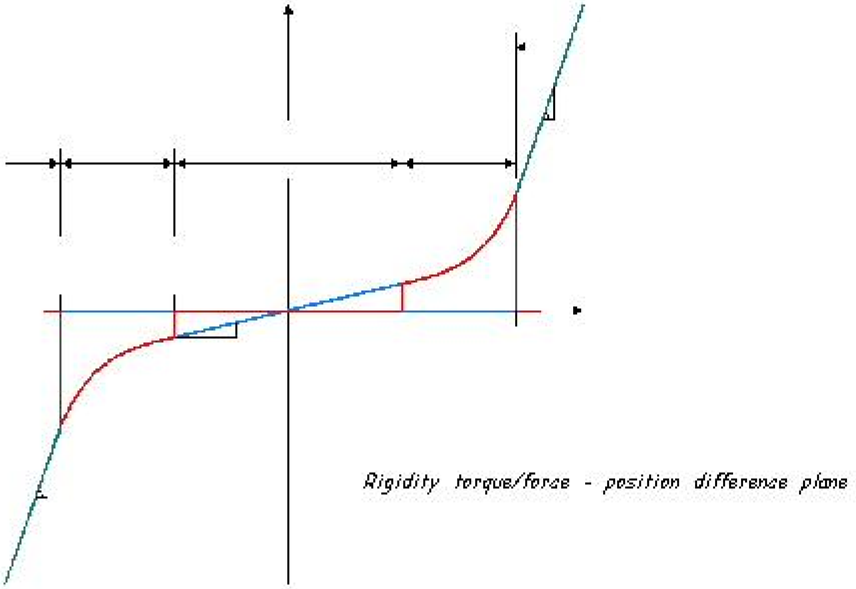


Figure 2. The 5 sub-areas of the rigidity force FSTF - the position difference (Δs) plane

The backlash area (BKL) is limited by the upper and lower backlash position parameters DSBKLU and DSBKLL. The linear rigidity CBKL is valid inside this area.

The normal areas, which follow into the backlash area, are limited by the parameters DSSTPL and DSBKLL (to the left) and DSBKLU and DSSTPU (to the right). The assigned rigidity behavior is valid only inside these areas, so the linear, progressive, degressive, or polynomial rigidity is only calculated within these areas.

The 2nd step rigidity areas, which follow into the normal areas, are limited by the upper and lower 2nd step position parameters DSSTPU and DSSTPL. The linear rigidity CSTP is valid inside this area.

Linear Rigidity Behavior

If linear rigidity coefficient $C > 0$, the component calculates the linear rigidity force FSTF [N] in the normal area: $FSTF = C \cdot \Delta s$. For $C = 0$, the rigidity component transfers only friction forces between the predecessor and successor body in the normal area.

Progressive Rigidity Behavior

If the Progressive Rigidity Coefficient $CPGR > 0$, the component calculates a progressive rigidity force FSTFPGR [N] in the normal area. The linear rigidity C is the initial rigidity at $\Delta s = 0$. The property is typical for parts that become stiffer and stiffer with increasing torsion/stress (e.g., couplings).

$FSTFPGR = C/CPGR \cdot \sinh(CPGR \cdot \Delta s)$

Degressive Rigidity Behavior

If the degressive rigidity coefficient $CDGR > 0$, the component calculates a degressive rigidity force FSTFDGR [N] in the normal area.

$FSTFDGR = C/CDGR \cdot \operatorname{ashinh} (CDGR \cdot \Delta s)$

The linear rigidity C is the initial rigidity at $\Delta s = 0$, and $0 \leq C < \infty$. The property is typical for parts (e.g. couplings) becoming softer and softer with increasing torsion/stress.

Polynomial Rigidity Behavior

If the Polynomial Rigidity Multiplier $CMUL > 0$, the component calculates a polynomial rigidity force FSTFPOL [N] in the normal area.

$FSTFPOL = C \cdot \Delta s + CMUL \cdot |\Delta s|^{CEXP} \cdot \operatorname{sgn}(\Delta s)$

where the linear rigidity C is the initial rigidity at $\Delta s = 0$, and $0 \leq C < \infty$. $0 \leq CMUL < \infty$, and the Polynomial Rigidity Exponent $CEXP \in (-\infty , \infty)$. The property is suitable to substitute for the progressive approach.

Characteristic Friction

If the Characteristic Friction Force $FFRCCH \neq 0$, the component calculates an arbitrary, weighted, speed dependent resistive force $FFRCCH = f(v(t))$, which can be specified with a characteristic (preferably a 2D Lookup Table characteristic).

$\mathrm{FFRCCH\_VAL} = \mathrm{KFRCCH} \cdot \mathrm{FFRCCH}(v(t))$

where KFRCCH is the weighting coefficient.

Rigidity Component Position Dependent Friction/Damping Properties

If the Position Dependant Polynomial Friction Multiplier KVSCMUL > 0, the component calculates a polynomial position difference or Position Dependent Viscous Friction Coefficient KVSCPOL = f(Δs)[Ns/m], and the resulting viscous friction coefficient KVSCRES [Ns/m]. The position dependence of friction and damping is accomplished by the position dependence of the viscous friction coefficient. This holds for the backlash area and the 2nd step area.

See also Backlash Property

See also Second Step Property

A polynomial approach for KVSC = f(Δs) can be specified. This property is typical for changing friction relations depending on position. The linear KVSC is the offset. KVSCRES is computed by superimposing KVSC, KVSCBKL, KVSCSTP, and KVSCPOL.

See also 2nd Step Friction/Damping Properties

$\mathrm{KVSCPOL} = \mathrm{KVSCMUL} \cdot |\Delta s(t)|^{\mathrm{KVSCEXP}}$, $\mathrm{KVSCRES} = \mathrm{KVSC} + \mathrm{KVSCPOL}$

where KVSCEXP is the Position Dependant Polynomial Friction exponent, and $\in (-\infty, \infty)$.

Backlash Property

The backlash property is useful for specifying an area (backlash area) with linear rigidity and friction properties that might differ from the ones in the normal area. The property is typical for gear backlash, loose parts, etc.

If the Backlash Lower Limit of Position Angle Difference DSBKLL < 0 or/and the Backlash Upper Limit of Position Angle Difference DSBKLL > 0, then the backlash property is calculated. To reach the backlash limits, the simulator reduces the simulation time step to HMIN. However, if the backlash range given by (DSBKLL, DSBKLU) is rather small, it might be skipped due to a comparatively large minimum time step HMIN. If this occurs, simulation continues without computing the backlash property at this instant, and a simulator message appears with the model name, component name, and time value. All backlash limits are dynamic parameters (i.e., they can be changed during simulation). However, if the conditions are violated, the parameters stay at their permissible limits, simulation continues, and an appropriate simulator message appears.

Backlash Rigidity Properties

If the Backlash Rigidity (inside backlash range) CBKL > 0, the component calculates a linear rigidity force FSTFBKL [N] in the backlash area. In most cases, the backlash rigidity tends to be zero. A residual rigidity, however, can be specified using CBKL. This parameter is useful in cases where stepped spring packages are modeled.

$\mathrm{FSTFBKL} = \mathrm{CBKL} \cdot \Delta s$

where CBKL is the Backlash Rigidity Coefficient, and CBKL ≥ 0.

Backlash Friction/Damping Properties

If the Backlash Viscous Friction Coefficient KBKL $\neq$ KVSC, the component calculates the resulting Viscous Friction Coefficient for Backlash KVSCBKL [Ns/m]. Outside the backlash area (in the normal area), KVSC is used; inside the backlash area, KBKL is used. KBKL $>$ 0 allows you to model a residual attenuation, which might result from lubricants.

KVSCBKL = KBKL, if DSBKLL < Δs < DSBKLU; otherwise KVSCBKL = KVSC.

Smooth KVSC Transition Between Backlash Area and Normal Area

On the transition from backlash to normal area and vice versa, an abrupt change between KVSC and KBKL occurs. Specifying the Backlash Viscous Friction Exponent KBKLEXP and Backlash Viscous Friction Conditioner KBKLCND smoothes this transition.

If KBKLEXP $>$ 0,, the component calculates the resulting viscous friction coefficients for backlash and normal area KVSCBKL [Ns/m] and FFRCBKL [N]. The friction torque results from KVSCBKL. KBKL $>$ 0 and allows you to model a residual attenuation, which might result from lubricants.

For a small KBKLEXP (e.g., KBKLEXP = 2), the transition is rather soft; whereas, for a higher one (e.g., KBKLEXP = 32), the transition is harder. KBKLCND affects the position of the medium transition from KVSC to KBKL (i.e., how well DSBKLU and DSBKLL are hit).

$\mathrm{KVSCBKL} = \mathrm{KVSC} - (\mathrm{KVSC} - \mathrm{KBKL})\ \mathrm{EXP}(-\delta^{\mathrm{KBKLEXP}} \cdot \varphi^{\mathrm{KBKLEXP}})$

$\mathrm{FFRCBKL} = \mathrm{KVSCBKL} \cdot \Delta\omega(t)$

where $\delta = 1/(\sqrt{KBKLCND} \cdot (\varepsilon/2))$, $\varphi = \Delta\varphi - (\mathrm{DPHIBKLU} - \varepsilon/2)$ and $\varepsilon = \mathrm{DPHIBKLU} - \mathrm{DPHIBKLL}$.

Second Step

The 2nd step property is useful for specifying an area (2nd step area) with linear rigidity and friction properties that might differ from the ones in the normal area or in the backlash area. Using the backlash parameters, you can model stepped spring packages with up to five steps (four switching points) with equal rigidity and friction properties in the individual areas. The property is typical for Stepped spring packages. If the Upper Position Difference of the 2nd step area DSSTPL < 0 and/or the Lower Position Difference of the 2nd step area DSSTPU > 0, the property is calculated.

To reach the 2nd step limits, the simulator reduces the simulation time step to HMIN. However, if the 2nd step range given by (DSSTPL, DSSTPU) is rather small, it might be skipped due to a rather large minimum time step HMIN. If this occurs, simulation continues without computing the 2nd step property at this instant, and a simulator message appears with the model name, component name, and time value. All 2nd step limits are dynamic parameters (i.e., they can be changed during simulation. However, if the conditions are violated, the parameters stay at their permissible limits, simulation continues, and an appropriate simulator message appears.

2nd Step Rigidity Properties

If the 2nd Step Rigidity (outside step range) CSTP > 0, the component calculates a linear rigidity force FSTFSTP [Nm] in the 2nd step area. This parameter is useful in cases where stepped spring packages are modeled.

$\mathrm{FSTFSTP} = \mathrm{CSTP} \cdot \Delta s$

2^{nd} Step Friction/Damping Properties

If the 2nd Step Viscous Friction Coefficient KSTP $\neq$ KVSC, the component calculates the resulting Viscous Friction Coefficient KVSCSTP [Nms/rad] in the 2^{nd} step area. Outside the backlash area (in the normal area), KVSC is used; inside the backlash area, the friction coefficient KSTP is used.

KVSCSTP = KVSC, if DSSTPL $< \Delta\varphi <$ DSSTPU; otherwise KVSCSTP = KSTP.

Smooth KVSC-Transition Between 2^{nd} Step Area and Normal Area

On the transition from the 2^{nd} step are to the normal area and vice versa, an abrupt change occurs between KVSC and KSTP. Specifying the 2^{nd} Step Viscous Friction Exponent KSTPEXP and the 2^{nd} Step Viscous Friction Conditioner KSTPCND smoothes this transition (analogous to backlash).

If KSTPEXP > 0, the component calculates the resulting viscous friction coefficients for the 2^{nd} step and normal areas, KVSCSTP [Ns/m] and FFRCSTP [N]. The friction force results from KVSCSTP.

For a small KSTPEXP (e.g., KSTPEXP = 2), the transition is rather soft; whereas, for a higher one (e.g., KSTPEXP = 32), the transition is harder. KSTPCND affects the position of the medium transition from KVSC to KSTP (i.e., how well DSSTPU and DSSTPL are hit).

In cases where KSTPEXP > 0 and KBKLEXP > 0, and the corresponding exponents are rather small (very soft transition), and the 2^{nd} step area and backlash area are close together (small normal area), then the resulting friction coefficient KVSCRES is a result of superimposing (averaging) KVSC, KVSCBKL, KVSCSTP, and KVSCPOL (if the polynomial viscous friction coefficient is additionally specified).

$\mathrm{KVSCSTP} = \mathrm{KSTP} - (\mathrm{KSTP} - \mathrm{KVSC}) \cdot \mathrm{EXP}(-\delta^{\mathrm{KSTPEXP}} \cdot s^{\mathrm{KSTPEXP}})$

$\mathrm{FFRCSTP} = \mathrm{KVSCSTP} \cdot \Delta v(t)$

where $\delta = 1/(\sqrt{KSTPCND} \cdot (\varepsilon/2))$, $s = \Delta s - (\mathrm{DPHISTPU} - \varepsilon/2)$ and $\varepsilon = \mathrm{DSSTPU} - \mathrm{DSSTPL}$.

Rigidity Component External Difference Force and Initial Values

External Difference Force

If the External Force FEXT $\neq$ 0, the component calculates an arbitrary, weighted, externally computed resistive force (time function) FEXT = f(t), which can be any arbitrary Twin Builder quantity (e.g., the result of a formula, a block output quantity of the block diagram module, a state graph quantity, etc.).

$\mathrm{FEXT\_VAL} = \mathrm{KEXT} \cdot \mathrm{FEXT}(t)$

where KEXT is the weighting coefficient.

Position Difference Initial Value

The Position Difference Initial Value S0 can be specified to define another starting point other than zero for pre-stretched springs.

$S0 = s\ (t = 0)$

The difference between the S0 of the predecessor and successor mass components, including possible transformation coefficients, must be set manually.

Top

Netlist Syntax

UMODEL VM_TR VM_TR_Pre_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_TR VM_TR_Suc_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %1, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchTStf ?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, TR2 := %1 (C:=@C ,KVSC:=@KVSC ,CMUL:=@CMUL ,CEXP:=@CEXP ,KMUL:=@KMUL ,KEXP::=@KEXP ,KVSCMUL:=@KVSCMUL ,KVSCEXP:=@KVSCEXP ,CPGR:=@CPGR ,KPGR::=@KPGR ,CDGR:=@CDGR ,KDGR:=@KDGR ,DSBKLU:=@DSBKLU ,DSBKLL:=@DSBKLL ,CBKL:=@CBKL ,KBKL:=@KBKL ,KBKLEXP:=@KBKLEXP ,KBKLCND:=@KBKLCND ,DSSTPU:=@DSSTPU ,DSSTPL:=@DSSTPL ,CSTP:=@CSTP ,KSTP:=@KSTP ,KSTPEXP::=@KSTPEXP ,KSTPCND:=@KSTPCND ,FSTFCH:=@FSTFCH ,KSTFCH:=@KSTFCH ,FFRCCH:=@FFRCCH ,KFRCCH:=@KFRCCH ,FEXT:=@FEXT ,KEXT:=@KEXT ,DS0::=@DS0) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
TR1 (See Note)	Predecessor Node	Mechanical
TR2 (See Note)	Successor Node	Mechanical

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
C	Rigidity (stiffness)	real	0 [N/m]
KVSC	Viscous friction coefficient	real	0 [Ns/m]
CMUL	Polynomial rigidity multiplier	real	0 [N/m]
CEXP	Polynomial rigidity exponent	real	0

KMUL	Polynomial friction multiplier	real	0 [Ns/m]
KEXP	Polynomial friction exponent	real	0
KVSCMUL	Position dept polynomial friction multiplier	real	0 [Ns/m]
KVSCEXP	Position dept polynomial friction exponent	real	0
CPGR	Progressivity rigidity coefficient	real	0
KPGR	Progressivity friction coefficient	real	0
CDGR	Degressivity rigidity coefficient	real	0
KDGR	Degressivity friction coefficient	real	0
DSBKLU	Backlash upper limit of position diff	real	0 [m]
DSBKLL	Backlash lower limit of position diff	real	0 [m]
CBKL	Backlash rigidity (inside backlash range)	real	0 [N/m]
KBKL	Backlash viscous friction coefficient	real	0 [Ns/m]
KBKLEXP	Backlash viscous friction exponent	real	0
KBKLCND	Backlash viscous friction conditioner	real	0
DSSTPU	2nd step upper limit of position diff	real	0 [m]
DSSTPL	2nd step lower limit of position diff	real	0 [m]
CSTP	2nd step rigidity (outside step range)	real	0 [N/m]
KSTP	2nd step viscous friction coefficient	real	0 [Ns/m]
KSTPEXP	2nd step viscous friction exponent	real	0
KSTPCND	2nd step viscous friction conditioner	real	1
FSTFCH	Characteristic rigidity force (f = f(ds))	real	0 [N]
KSTFCH	Characteristic rigidity force coefficient	real	1
FFRCCH	Characteristic friction force (f = f(dv))	real	0 [N]
KFRCCH	Characteristic friction torque coefficient	real	1
FEXT	External (arbitrary) force (f = f(t))	real	0 [N]

KEXT	External (arbitrary) force coefficient	real	1
DS0	Initial position difference	real	0 [m]

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
dDACC	Jerk difference [m/s^3]	Output	real
DACC	Acceleration difference [m/s^2]	Output	real
DV	Velocity difference [m/s]	Output	real
DS	Position difference [m]	Output	real
DF	Difference force [N]	Output	real
V1	Predecessor velocity [m/s]	Output	real
V2	Successor velocity [m/s]	Output	real
FSTF	Resulting rigidity force [N]	Output	real
FFRC	Resulting friction force [N]	Output	real
FVSC	Viscous friction force [N]	Output	real
KVSCRES	Resulting viscous friction coefficient [Ns/m]	Output	real
FSTFPOL	Polynomial rigidity force [N]	Output	real
FFRCPOL	Polynomial friction force [N]	Output	real
KVSCPOL	Polynomial viscous friction coefficient [Ns/m]	Output	real
FSTFPGR	Progressive rigidity force [N]	Output	real
FFRCPGR	Progressive friction force [N]	Output	real
FSTFDGR	Degressive rigidity force [N]	Output	real
FFRCDGR	Degressive friction force [N]	Output	real
FSTFBKL	Backlash rigidity force [N]	Output	real
KVSCBKL	Backlash viscous friction coefficient [Ns/m]	Output	real
MFRCBKL	Backlash friction force [N]	Output	real
MSTFSTP	2nd step rigidity force [N]	Output	real
KVSCSTP	2nd step viscous friction coefficient [Ns/m]	Output	real
FFRCSTP	2nd step viscous friction force [N]	Output	real
FFRCCLB	Coulomb friction torque [N]	Output	real
FSTFCH_VAL	Characteristic rigidity force [N]	Output	real
FFRCCH_	Characteristic friction force [N]	Output	real

VAL			
FEXT_VAL	External (arbitrary) torque [Nm]	Output	real

Top

Example

Top

References

Transmission

- Gearbox
- Transformation

Gearbox

- Ideal GearBox (gear_ideal)
- Gearbox with Losses (gear_loss)

Ideal Gearbox

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

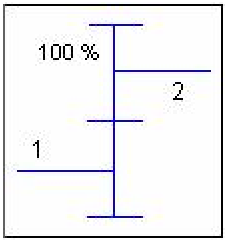


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

The ideal gearbox model provides the values of rotational speed and torque using the gear ratio. There is no calculation of losses. Both power flow directions can be simulated.

Top

Assumptions and Limitations

1. The gearbox model has no built-in inertia. If needed, inertia components may be placed at both shafts of the gearbox.

2. This model doesn’t take into account the losses.

3. Only basic mechanical properties are modeled.

Top

Mathematical Description

The transfer functions of the Ideal Gearbox model are:

n2 = n1 / ratio

omega2 = omega1 / ratio

torque2 = - torque1 · ratio

where n1, omega1, torque1 are the speed, angular velocity and torque of shaft 1, and n2, omega2 and torque2 are those of shaft 2.

Top

Netlist Syntax

UMODEL gear_ideal ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1:= %0, rot2:= %1 (ratio:= @ratio) SRC: DB(Lib:=@ModelLibraryName) ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
rot1	Shaft 1	Rotational_v
rot2	Shaft 2	Rotational_v

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
ratio	Gear Ration	real	I

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
n1	Speed – Shaft 1 [rpm]	Output	real
omega1	Angular Velocity - Shaft 1 [rad/s]	Output	real
torque1	Torque – Shaft 1 [Nm]	Output	real
n2	Speed – Shaft 2 [rpm]	Output	real
omega2	Angular Velocity - Shaft 2 [rad/s]	Output	real

torque2	Torque – Shaft 2 [Nm]	Output	real

Top

Example

Top

References

Gearbox with Losses

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

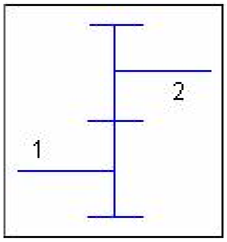


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

The Gearbox with Losses model provides the values of rotational speed and torque using the gear ratio and the gearbox efficiency. Both power flow directions can be simulated, i.e., if the power flows from shaft 2 to shaft 1, the reciprocal value of the gearbox efficiency is used to calculate the torque values.

Top

Assumptions and Limitations

This gearbox model has no built-in inertia. If needed, inertia components may be placed at both shafts of the gearbox.

Top

Mathematical Description

The transfer functions of the Gearbox with Losses model are:

n2 = n1 / ratio

omega2 = omega1 / ratio

torque2 = - torque1 · ratio · eta

where n1, omega1, torque1 are the speed, angular velocity and torque of shaft 1, n2, omega2 and torque2 are those of shaft 2, and eta is the gearbox efficiency.

Top

Netlist Syntax

UMODEL gear_loss ?InstanceName(@InstanceName):(@(Refbase)@(ID)) rot1:= %0, rot2:= %1 (ratio:= @ratio, eta:= @eta) SRC: DB(Lib:=@ModelLibraryName) ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
rot1	Shaft 1	Rotational_v
rot2	Shaft 2	Rotational_v

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
ratio	Gear Ration	real	I
eta	Efficiency	real	0.95

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
n1	Speed – Shaft 1 [rpm]	Output	real
omega1	Angular Velocity - Shaft 1 [rad/s]	Output	real
torque1	Torque – Shaft 1 [Nm]	Output	real
n2	Speed – Shaft 2 [rpm]	Output	real
omega2	Angular Velocity - Shaft 2 [rad/s]	Output	real

torque2	Torque – Shaft 2 [Nm]	Output	real

Top

Example

Top

References

Transformation

These components transform motion quantities between mechanical predecessor and successor components. All motion coordinates are absolute quantities referred to ground. The transformation coefficient K defines the ratio of predecessor to successor angular velocity/velocity, or the ratio of successor to predecessor torque/force. The components are used for gear ratios, wheel diameters in chain drives, and screw pitch in worm drives.

The transformation components can connect two mechanical components, which also could be connected directly. It must not be used, however, to connect two components that cannot be connected directly (i.e., for directly connecting two mass components).

The predecessor quantities are supplied by the predecessor component, whereas the successor quantities are supplied to the successor component.

Following are the available transformation components:

- Coordinate Transformation Rotational-Rotational (MchRrTsf)
- Coordinate Transformation Rotational-Translational (MchRtTsf)
- Coordinate Transformation Translational-Rotational (MchTrTsf)
- Coordinate Transformation Translational-Translational (MchTtTsf)

Rotational_V-Rotational_V

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

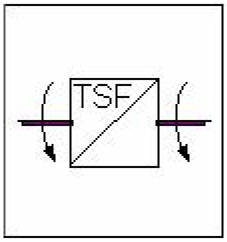


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

This component transforms the motion quantities between predecessor and successor in rotational motions. The transformation coefficient K defines the ratio of predecessor to successor angular velocity, and the ratio of successor to predecessor torque. In other words, K serves as the inverse gear ratio.

Top

Assumptions and Limitations

No internal inertia or losses are considered in this component. There is no output limits.

Top

Mathematical Description

The relation between the quantities of the successor and the predecessor are:

$M2 = K \cdot M1$

$OMEGA2 = OMEGA1 / K$

where M1 and M2 are the torque supplied by the predecessor and the torque supplied to the successor respectively, and OMEGA1 and OMEGA2 are the angular velocity of the predecessor and the successor, respectively. K is the Transformation Factor.

Top

Netlist Syntax

UMODEL VM_ROT VM_ROT_Pre_?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %0, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_ROT VM_ROT_Suc_@InstanceName ROT1 := %1, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchRrTsf ?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %0, ROT2 := %1 (K:=@K) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
ROT1 (See Note)	Predecessor Node	Mechanical
ROT2 (See Note)	Successor Node	Mechanical

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
K	Transformation factor	real	1

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
M1	Torque supplied by predecessor [Nm]	Output	real
M2	Torque supplied to successor [Nm]	Output	real
OMEGA1	Predecessor angular velocity	Output	real

	[rad/s]		
OMEGA2	Successor angular velocity [rad/s]	Output	real

Top

Example

Top

References

Rotational_V-Translational_V

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

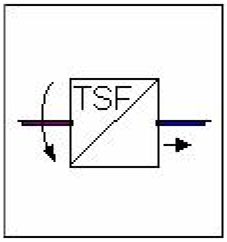


Figure 1. Component symbol

Description

This component transforms the reference coordinates of the motion quantities between predecessor and successor from rotational motions to translational. The transformation coefficient K defines the following two ratios:

- The ratio of predecessor angular velocity to successor velocity.
- The ratio of successor force to predecessor torque.

In other words, K serves as the inverse gear ratio.

Top

Assumptions and Limitations

No internal inertia or losses are considered in this component. There is no output limits.

Top

Mathematical Description

The relation between the quantities of the successor and the predecessor are:

F2 = K · M1

V2 = OMEGA1 / K

where M1 is the torque supplied by the predecessor, F2 is the force supplied to the successor, OMEGA1 is the angular velocity of the predecessor and V2 is the velocity of the successor. K is the Transformation Factor.

Top

Netlist Syntax

UMODEL VM_ROT VM_ROT_Pre_?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %0, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_TR VM_TR_Suc_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %1, TR2 := GND () SRC: SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchRtTsf @InstanceName ROT1 := %0, TR2 := %1 (K:=@K) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
ROT1 (See Note)	Predecessor Node	Mechanical
TR2 (See Note)	Successor Node	Mechanical

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
K	Transformation factor	real	1

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
M1	Torque supplied by predecessor [Nm]	Output	real
F2	Force supplied to successor [N]	Output	real
OMEGA1	Predecessor angular velocity [rad/s]	Output	real
V2	Successor velocity [m/s]	Output	real

Top

Example

Top

References

Translational_V-Rotational_V

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

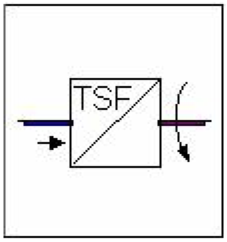


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

This component transforms the reference coordinates of the motion quantities between predecessor and successor from translational motion to rotational. The transformation coefficient K defines the ratio of predecessor velocity to successor angular velocity and the ratio of successor torque to predecessor force. In other words, K servers as the inverse gear ratio.

Top

Assumptions and Limitations

No internal inertia or losses are considered in this component. There is no output limits.

Top

Mathematical Description

The relation between the quantities of the successor and the predecessor are:

M2 = K · F1

OMEGA2 = V1 / K

Where F1 is the force supplied by the predecessor, M2 is the torque supplied to the successor, V1 is the velocity of the predecessor and OMEGA2 is the angular velocity of the successor. K is the Transformation Factor.

Top

Netlist Syntax

UMODEL VM_TR VM_TR_Pre_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_ROT VM_ROT_Suc_?InstanceName(@InstanceName):(@(Refbase)@(ID)) ROT1 := %1, ROT2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchTrTsf ?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, ROT2 := %1 (K:=@K) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
TR1 (See Note)	Predecessor Node	Mechanical
ROT2 (See Note)	Successor Node	Mechanical

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
K	Transformation Factor	real	1

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
F1	Force supplied by predecessor [N]	Output	real
M2	Torque supplied to successor [Nm]	Output	real

V1	Predecessor velocity [m/s]	Output	real
OMEGA2	Successor angular velocity [rad/s]	Output	real

Top

Example

Top

References

Translational_V-Translational_V

Library: Mechanical_System	Modeling Language: SML	Version Number: Twin Builder 2025.2

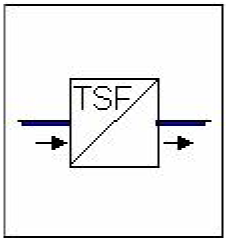


Figure 1. Component symbol

- Description
- Assumptions and Limitations
- Mathematical Description
- Netlist Syntax
- Conservative Pins
- Parameters
- Input/Output Quantities
- Example
- References

Description

This component transforms the reference coordinates of the motion quantities between predecessor and successor from translational motion to translational. The transformation coefficient K defines the ratio of predecessor to successor velocity, and the ratio of successor to predecessor force. In a simple case, K constitutes the inverse gear ratio. The predecessor and successor forces and velocities are provided for output.

Top

Assumptions and Limitations

No internal inertia or losses are considered in this component. There is no output limits.

Top

Mathematical Description

The relation between the quantities of the successor and the predecessor are:

$F2 = K \cdot F1$

V2 = V1 / K

where F1 and F2 are the force supplied by the predecessor and the force supplied to the successor respectively, and V1 and V2 are the velocity of the predecessor and the successor, respectively. K is the Transformation Factor.

Top

Netlist Syntax

UMODEL VM_TR VM_TR_Pre_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL VM_TR VM_TR_Suc_?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %1, TR2 := GND () SRC: DB(Lib:= "Simplorer Elements\\\\Basic Elements\\\\Basic Elements") ; UMODEL MchTtTsf ?InstanceName(@InstanceName):(@(Refbase)@(ID)) TR1 := %0, TR2 := %1 (K:=@K) SRC: DB(Lib:= "Simplorer Elements\\\\Multiphysics\\\\Mechanical System\\\\Mechanical System") ;

Top

Conservative Pins

Table 1

Name	Port/Terminal description	Nature/Data type
TR1 (See Note)	Predecessor Node	Mechanical
TR2 (See Note)	Successor Node	Mechanical

Note: Terminal set to No Action when unconnected. Terminal may remain unconnected without generating an error.

Top

Parameters

Table2

Name	Description	Data Type	Default Value [Unit]
K	Transformation factor	real	1

Top

Input/Output Quantities

Table3

Name	Description [Unit]	Direction	Data Type
F1	Force supplied by predecessor [N]	Output	real
F2	Force supplied to successor [N]	Output	real

V1	Predecessor velocity [m/s]	Output	real
V2	Successor velocity [m/s]	Output	real

Top

Example

Top

References

Index