

FEDeM

User's Guide

Release 5.0

Trondheim, April 2009

Copyright © 1995-2009 by Fedem Technology AS

Published 2009. Printed in Norway.

All rights reserved. No part of this document may be reproduced in any form or distributed in any way without prior written permission from Fedem Technology AS.

The information provided in this document is subject to change without notice.

The Software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement, which accompanies the software.

Fedem is a registered trademark of Fedem Technology AS.

Acrobat Reader is a registered trademark of Adobe Systems, Inc.

Altair and Hypermesh are registered trademarks of Altair Engineering, Inc.

I-DEAS is a registered trademark of Structural Dynamics Research Corporation.

MATLAB and Simulink are registered trademarks of The MathWorks, Inc.

MATRIXx is a registered trademark of Wind River Systems, Inc.

MSC.Nastran is a registered trademark of The MSC.Software Corporation.

MSC.Patran is a registered trademark of The MSC.Software Corporation.

Nastran is a registered trademark of NASA.

OpenGL is a registered trademarks of Silicon Graphics, Inc.

Pro/ENGINEER is a registered trademark of Parametric Technology Corporation.

All other brand or product names mentioned herein are trademarks or registered trademarks of their respective companies or organizations.

Fedem Technology AS

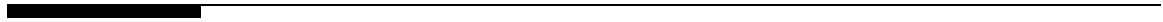
Prinsens gate 32, N-7011 Trondheim, Norway

Phone: +47 72 90 03 00, Fax: +47 72 90 03 01

Web: <http://www.fedem.com>

Technical Support: support@fedem.com

Sales, Pricing, and General Information: sales@fedem.com



Fedem Release 5.0 User's Guide

Table of Contents

Chapter 1 Introduction to Fedem

1.1	What is Fedem?	1-2
1.2	Non-linear structural dynamics	1-3
1.3	Control systems in mechanical analysis	1-3
1.4	What is a Fedem model?	1-4
1.5	Using FE models	1-4
1.6	CAD integration	1-5
1.7	Fedem solver modules	1-5
1.7.1	Reducer	1-6
1.7.2	Dynamics Solver	1-6
1.7.3	Stress Recovery	1-6
1.7.4	Mode Shape Recovery	1-6
1.7.5	Strain Rosette Analysis	1-6
1.7.6	Strain Coat Analysis	1-7
1.7.7	Curve Export Utility	1-7

Chapter 2 Learning the Basics

2.1	System requirements	2-2
2.2	Software requirements	2-2
2.3	Storing models and results	2-3
2.3.1	FTL format	2-3
2.3.2	FTC format	2-3
2.3.3	Other supported formats	2-4
2.4	Starting Fedem	2-4

2.4.1	Template model file	2-5
2.4.2	Console window	2-5
2.5	Touring the interface	2-6
2.5.1	Fedem's main window	2-6
2.5.2	Menus and toolbars	2-7
2.5.3	Model Manager	2-9
2.5.4	ID and Topology panel	2-10
2.5.5	Property Editor	2-12
2.5.6	Workspace	2-13
2.5.7	Output List	2-16
2.6	Executing commands	2-17
2.6.1	Select	2-17
2.6.2	Done	2-19
2.6.3	Cancel	2-19
2.7	Visualizing the model	2-19
2.7.1	3D Navigation	2-19
2.7.2	3D View controls	2-22
2.7.3	Zoom and Pan	2-23
2.7.4	General Appearance	2-24
2.7.5	Item Appearance	2-27
2.7.6	Element face visibility	2-28
2.7.7	Visualization of special finite elements	2-29
2.8	Opening and saving model files	2-30
2.8.1	Opening a file	2-30
2.8.2	Saving models	2-32
2.8.3	Starting a new model	2-33
2.9	Loading and unloading FE-Data	2-33
2.9.1	FE-Data Settings	2-33
2.9.2	Skipping FE-Data when opening a model file	2-34

2.9.3	Modeling with unloaded links	2-34
2.9.4	Postprocessing unloaded links	2-35
2.10	Printing and exporting	2-35
2.10.1	Printing	2-35
2.10.2	Exporting	2-35
2.11	License information	2-38
2.11.1	Available modules	2-38
2.11.2	License denial	2-38
2.11.3	License file/server	2-39
2.11.4	Managing license files/servers	2-39
2.12	Using the CAD integration	2-39
2.12.1	Overview	2-39
2.12.2	Model association	2-40
2.12.3	How parts and subassemblies translate to Fedem links	2-41
2.12.4	Process communication	2-41
2.12.5	Commands in CAD system (The <i>Fedem</i> menu)	2-42
2.12.6	Commands in Fedem	2-43

Chapter 3 Mechanism Modeling

3.1	Basic assembling techniques	3-2
3.2	Mechanism modeling environment	3-2
3.2.1	Modeler window	3-3
3.2.2	Modeling toolbars	3-3
3.3	Mechanism modeling tools	3-4
3.3.1	Reference Plane	3-4
3.3.2	Interactive Odometer and 3D Point Marker	3-5
3.3.3	Stickers	3-6
3.4	Creating mechanism elements	3-7
3.4.1	Selecting position and orientation	3-8

3.5	Moving mechanism elements	3-10
3.5.1	Smart Move	3-10
3.5.2	Align CS and rotations	3-12
3.5.3	Move To Center	3-12
3.5.4	Origin property	3-13
3.6	Attaching and detaching elements	3-15
3.6.1	Attaching using <i>Attach</i>	3-16
3.6.2	Surface Connectors	3-17
3.6.3	Surface connector commands	3-18
3.6.4	Attachment rules and restrictions	3-20
3.6.5	Detaching	3-21
3.6.6	Color of attached and unattached elements	3-21
3.6.7	Invalid attachments	3-21
3.7	Deleting mechanism elements	3-22
3.7.1	Deleting in the Modeler	3-22
3.7.2	Deleting in the Model Manager	3-22
3.8	Using file references in mechanism elements	3-23
3.9	Model preferences	3-24
3.9.1	Model database units	3-25
3.9.2	Modeling tolerance	3-25
3.9.3	Buoyancy	3-25
3.9.4	Gravitation	3-26
3.9.5	Initial translational velocity	3-26

Chapter 4 Mechanism Elements

4.1	Links	4-2
4.1.1	Creating links by file import	4-3
4.1.2	Creating links from hard points	4-5
4.1.3	Copying links	4-6

4.1.4	Link properties	4-6
4.1.5	Using link repositories	4-14
4.2	Element groups.....	4-16
4.2.1	Element group properties.....	4-17
4.3	Triads.....	4-18
4.3.1	Triads in joints	4-18
4.3.2	Triad symbols.....	4-18
4.3.3	Triad properties.....	4-19
4.4	Joints.....	4-20
4.4.1	Joint variables	4-20
4.4.2	Joint properties.....	4-21
4.4.3	Point-to-point joints	4-27
4.4.4	Point-to-path joints	4-30
4.5	Joint pair constraints.....	4-37
4.5.1	Gears.....	4-37
4.5.2	Rack-and-Pinion	4-38
4.6	Frictions	4-38
4.7	Springs and Dampers	4-39
4.7.1	Spring properties	4-40
4.7.2	Damper properties	4-41
4.7.3	Axial spring symbol.....	4-42
4.7.4	Axial damper symbol	4-42
4.7.5	Spring and damper characteristics	4-42
4.7.6	Advanced spring characteristics.....	4-44
4.8	Loads.....	4-46
4.8.1	Load symbols.....	4-46
4.8.2	Load properties.....	4-47
4.8.3	Target point	4-47
4.8.4	Direction	4-48

4.9	Tires and Roads	4-48
4.9.1	Tire	4-48
4.9.2	Road	4-49
4.9.3	Road elevation	4-50
4.10	Functions	4-50
4.10.1	Creating a function	4-51
4.10.2	Function properties	4-51
4.10.3	Preview	4-52
4.10.4	Extrapolation	4-53
4.10.5	Function Types	4-53
4.10.6	Time history input files	4-63
4.11	Sensors	4-64
4.11.1	Simple sensors	4-65
4.11.2	Relative sensors	4-65
4.11.3	Managing sensors	4-65
4.12	Strain rosettes	4-66

Chapter 5 Control System Modeling

5.1	Control modeling environment	5-2
5.1.1	Control Editor	5-2
5.1.2	Control toolbars	5-2
5.2	Input and output	5-3
5.3	Control blocks	5-4
5.3.1	Amplifiers	5-4
5.3.2	Binary-input blocks	5-4
5.3.3	Integrator and limited derivator blocks	5-5
5.3.4	Time-dependent blocks	5-5
5.3.5	Non-continuous blocks	5-5
5.3.6	PI, PD, and PID controllers	5-6

5.3.7	General-transfer functions	5-7
5.4	Building control modules	5-7
5.4.1	Setting Grid and Snap	5-8
5.4.2	Inserting blocks.....	5-8
5.4.3	Moving blocks.....	5-8
5.4.4	Editing block properties	5-9
5.4.5	Connecting blocks.....	5-9
5.4.6	Rotating blocks.....	5-10
5.4.7	Deleting blocks or connections.....	5-10
5.5	External control systems	5-11
5.5.1	Requirements	5-11
5.5.2	Data exchange	5-11
5.5.3	Connecting Simulink models.....	5-12
5.5.4	Performance.....	5-13
5.5.5	Limitations	5-13

Chapter 6 Mechanism Analysis

6.1	Overview of Fedem analyses	6-2
6.1.1	Model reduction.....	6-2
6.1.2	Dynamics analysis	6-2
6.1.3	Stress recovery	6-4
6.1.4	Mode shape recovery	6-4
6.1.5	Strain rosette recovery	6-4
6.1.6	Strain coat recovery.....	6-4
6.1.7	Duty cycle analysis.....	6-4
6.2	Solver tools.....	6-5
6.2.1	Solvers Toolbar	6-5
6.2.2	Controlling placement of temporary files	6-5
6.2.3	Additional solver options	6-6

6.2.4	Link- and group-wise solving	6-8
6.3	Model reduction	6-10
6.3.2	Using component modes.....	6-11
6.3.3	Using lumped mass matrix.....	6-12
6.3.4	Handling singularities during the model reduction	6-12
6.3.5	Eigenvalue analysis of the reduced links.....	6-14
6.3.6	Visualization of eigenmode shapes from the model reduction	6-15
6.3.7	Reduction of applied load vectors.....	6-15
6.4	Model reduction in Nastran	6-16
6.5	Dynamics analysis.....	6-18
6.5.1	Dynamics Solver Setup	6-18
6.5.2	Result output control.....	6-26
6.5.3	Monitoring the most problematic DOFs during time integration	6-29
6.5.4	Starting the analysis.....	6-30
6.5.5	Handling singularities during the dynamics analysis	6-30
6.6	Stress recovery analysis	6-31
6.6.1	Stress recovery options.....	6-31
6.6.2	Result output control.....	6-32
6.6.3	Import of residual stresses from external analyses	6-32
6.6.4	Starting the analysis.....	6-33
6.7	Mode shape recovery analysis	6-33
6.7.1	Mode shape options.....	6-33
6.7.2	Starting the analysis.....	6-34
6.8	Strain rosette analysis	6-35
6.8.1	Strain rosette options.....	6-35
6.8.2	Starting the analysis.....	6-36
6.8.3	Result output	6-36
6.8.4	Strain rosette definition file format.....	6-38
6.9	Strain coat analysis.....	6-40

6.9.1	Generating strain coat	6-40
6.9.2	Strain coat analysis options	6-41
6.9.3	Import of residual stresses from external analyses.....	6-42
6.9.4	Starting the analysis	6-43
6.9.5	Strain coat recovery on element groups or individual links	6-43
6.9.6	Using nCode FE-Fatigue.....	6-43
6.9.7	Licensing needs	6-44
6.10	Duty cycle analysis.....	6-44
6.10.1	Getting started	6-45
6.10.2	Licensing needs	6-45
6.10.3	Setting up a Duty Cycle analysis.....	6-45
6.10.4	Running Duty Cycle.....	6-46
6.10.5	Viewing Duty Cycle results	6-47
6.11	Interaction during processing	6-47
6.11.1	Simultaneous viewing and processing.....	6-47
6.11.2	Stop processing	6-48
6.12	Deleting results.....	6-48
6.12.1	Deleting specific results.....	6-48
6.13	Automated curve export from multiple result database files.....	6-49
6.14	Batch execution of solver processes	6-49
6.14.1	Batch solving trough the User Interface.....	6-49
6.14.2	Preparing for batch solving on remote computers	6-50
6.15	How to read error messages from the solvers	6-51

Chapter 7 Postprocessing Results

7.1	Postprocessing environment	7-2
7.2	Graphs	7-3
7.2.1	Creating graphs and curves	7-4
7.2.2	Moving curves to a new graph	7-6

7.2.3	Showing a graph	7-6
7.2.4	Graph properties	7-6
7.2.5	Curve properties	7-7
7.2.6	Fourier analysis.....	7-13
7.2.7	Scale and Shift	7-14
7.2.8	Appearance.....	7-15
7.2.9	Curve Statistics.....	7-15
7.2.10	Fatigue calculation from standard S-N curves.....	7-16
7.2.11	View control	7-17
7.2.12	Export of Curve Data	7-18
7.2.13	Importing Curves and Graphs.....	7-20
7.2.14	Exporting to picture files	7-20
7.2.15	Printing graphs.....	7-21
7.3	Animations	7-21
7.3.1	Managing animations	7-22
7.3.2	Animation properties.....	7-23
7.3.3	Available animation results	7-29
7.3.4	Performance of animation loading.....	7-32
7.4	Viewing animations	7-32
7.4.1	Play panel.....	7-33
7.4.2	Animation controls.....	7-35
7.4.3	Contour legend control.....	7-36
7.4.4	Exporting animations	7-39

Chapter 8 Managing Results

8.1	Model and Result file handling	8-2
8.1.1	Discarding unsaved changes.....	8-2
8.1.2	Saving a model.....	8-2
8.2	Result File Browser	8-3

8.2.1	The Result File Browser dialog	8-3
8.2.2	Result manipulation	8-5
8.2.3	Result files from restart simulations.....	8-7
8.3	RDB directory structure	8-8
8.3.1	Link database.....	8-9
8.3.2	Response directory structure.....	8-9

Appendix A Using the MF-Tyre Model

A.1	About MF-Tyre.....	A-2
A.2	MF-Tyre Version 5.2	A-2
A.3	Tire-Road Interaction.....	A-3
A.4	Axis Systems and Definitions	A-5
A.4.1	W-Axis System.....	A-5
A.4.2	Units	A-6
A.4.3	The Contact-Point C and the Normal Load	A-7
A.4.4	The Effective Tire Rolling Radius.....	A-8
A.4.5	Tire Slip Quantities.....	A-11
A.5	The Magic Formula Tire Model (MF-Tyre).....	A-11
A.5.1	Introduction.....	A-11
A.5.2	History of the Magic Formula.....	A-12
A.5.3	Learning the Basics of the Magic Formula.....	A-13
A.5.4	User Scaling Factors.....	A-15
A.6	Steady-State: Magic Formula	A-18
A.6.1	Steady-State Pure Slip.....	A-18
A.6.2	Magic Formula Steady-State Combined Slip	A-25
A.6.3	Transient Behavior	A-30
A.6.4	Transient Model Equations	A-30
A.6.5	The Gyroscopic Couple	A-31
A.6.6	Switching from a Simple to a Complex Tire Model	A-32

A.7	Some Practical Aspects.....	A-33
A.7.1	Rolling Resistance Torque.....	A-33
A.7.2	Typical Tire Characteristics.....	A-33
A.7.3	Effect of Camber Angle.....	A-34
A.7.4	Tire Model Output at Extreme Input Values.....	A-35
A.8	Standard Tire Interface (STI).....	A-35
A.9	MF-Datasets and MF-Tool.....	A-36
A.9.1	Tire Measurements.....	A-37
A.9.2	Calculation of Magic Formula Parameters.....	A-37
A.10	Definitions.....	A-38
A.11	References.....	A-40

Appendix B Using the SWIFT-Tyre Model

B.1	Introduction.....	B-2
B.1.1	Dynamics.....	B-2
B.1.2	Slip Force Calculation.....	B-2
B.1.3	Road Input.....	B-2
B.1.4	References.....	B-3
B.2	Notation.....	B-3
B.3	Force Evaluation.....	B-8
B.3.1	Rigid Ring Model.....	B-8
B.3.2	In-Plane Characteristics.....	B-8
B.3.3	Out-of-Plane Characteristics.....	B-9
B.3.4	Vertical Force Characteristics.....	B-10
B.3.5	The Effective Tire Rolling Radius.....	B-14
B.3.6	Effective Road Input.....	B-14
B.3.7	Contact Model.....	B-19
B.3.8	Switching from Simple to Complex Tire Model.....	B-21
B.4	Tire Model Parameters.....	B-22

B.5	Tire Property File Example.....	B-28
B.6	Road Property File Example	B-34

Appendix C FE Link Interface

C.1	Fedem Technology Link format.....	C-2
C.1.1	Syntax.....	C-2
C.1.2	Nodes.....	C-3
C.1.3	Structural elements.....	C-3
C.1.4	Properties	C-5
C.1.5	Loads	C-10
C.1.6	Strain Coat Elements.....	C-11
C.1.7	Strain Coat Properties.....	C-11
C.1.8	Other identifiers	C-12
C.2	Nastran Bulk Data File format.....	C-13

Appendix D File Types and Usage

D.1	File types	D-2
D.1.1	Input files	D-2
D.1.2	Intermediate files	D-3
D.1.3	Results files.....	D-3
D.1.4	Other files	D-3
D.2	File usage for each program module.....	D-4

Appendix E Command line options

E.1	Fedem UI Options (fedem)	E-2
E.2	Link reducer options (fedem_reducer)	E-3
E.3	Dynamics solver options (fedem_solver).....	E-6
E.4	Stress recovery options (fedem_stress)	E-12
E.5	Mode shape recovery options (fedem_modes).....	E-14

E.6	Strain rosette recovery options (fedem_gage).....	E-16
E.7	Strain coat recovery options (fedem_fpp)	E-18
E.8	Curve export options (fedem_graphexp).....	E-20

Appendix F Beta feature documentation

F.1	Joints.....	F-2
F.1.1	Universal Joint	F-2
F.1.2	Constant Velocity Joint	F-2
F.1.3	Rigid Joint	F-3
F.1.4	Axial Joint.....	F-3
F.1.6	Prismatic Joint and Cylindric Joint.....	F-4
F.1.7	Cam Joint	F-4
F.2	Links.....	F-5
F.2.2	Component modes.....	F-5
F.3	Springs	F-6
F.4	Frictions	F-7
F.5	Tires.....	F-7
F.6	Roads.....	F-8
F.7	Additional masses.....	F-8
F.8	Sensors	F-9
F.9	Generic database objects.....	F-9
F.10	Prescribed Triad motion	F-10
F.11	Initial conditions for dynamics analysis.....	F-10
F.11.1	Initial velocities in the model file	F-11
F.11.2	Initial velocities using description field commands	F-11
F.11.3	Turning initial conditions off	F-12
F.12	Boundary conditions for dynamics analysis	F-12

Index



Chapter 1 Introduction to Fedem

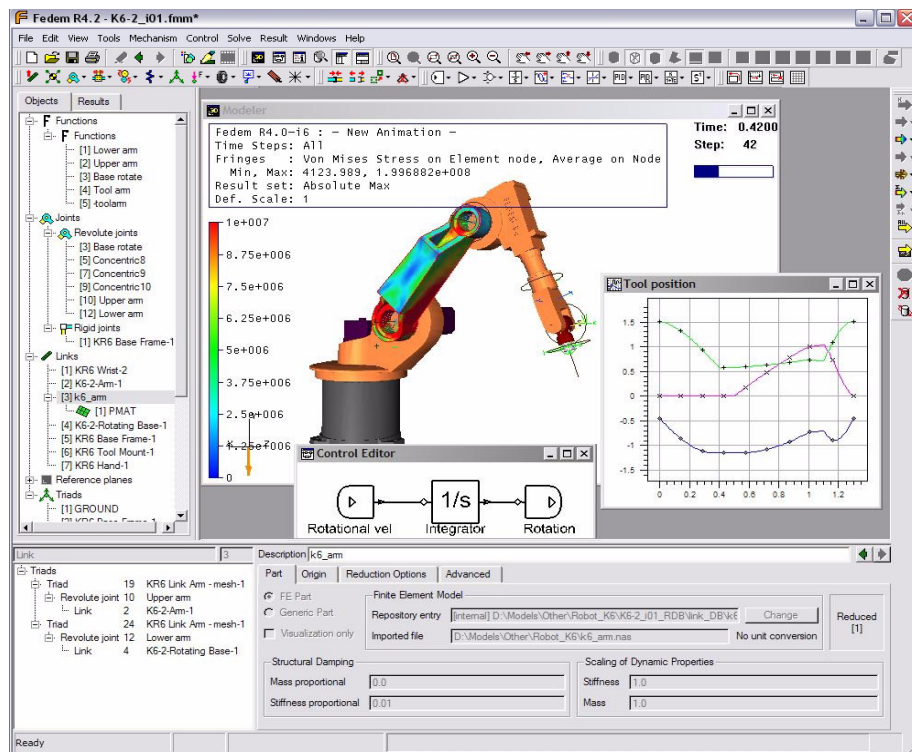
Welcome to Fedem! This chapter gives an overview of the Fedem program and its technological foundation. It explains what a Fedem model is, and outlines the different program modules.

Sections in this chapter address the following topics:

- [What is Fedem?](#)
- [Non-linear structural dynamics](#)
- [What is a Fedem model?](#)
- [Control systems in mechanical analysis](#)
- [Using FE models](#)
- [CAD integration](#)
- [Fedem solver modules](#)

1.1 What is Fedem?

Fedem, an acronym for Finite Element Dynamics in Elastic Mechanisms, provides both a technology platform and an engineering framework for virtual testing of complex mechanical assemblies. It provides a complete set of features to create, solve and post-process a model in a 3D graphical environment. Dynamic results in the form of curves and animations are available during and after model solution. Combined with the fast and numerically stable Fedem solvers, the user interface facilitates an engineering process with shortened turnaround times and quick access to simulation results for a clearer understanding of the physical behavior of the model. Fedem also provides intuitive and high-performing postprocessing capabilities, including full stress analysis, eigenmode solutions, strain gage solutions and fatigue analysis for selected time steps.



1.2 Non-linear structural dynamics

In multibody applications such as suspension systems, axle systems, car bodies, satellite appendages, industrial manipulators, medical equipment, high-speed mechatronic systems and so on, some of the mechanism components can be flexible and can experience large elastic deflections and coupling effects. To ensure sufficient accuracy, the simulation solver must account for the mutual dependencies between dynamic properties at the system level and structural flexibility at the component level. These requirements can be efficiently satisfied through a non-linear structural dynamics approach.

In Fedem, a non-linear structural dynamics approach is utilized in order to simultaneously solve structural deformations and 3D motion dynamics in the time domain. The mechanical assembly to be simulated is comprised of parts, each represented by a linear elastic finite element model or a simplified stiffness description and coupled together with linear or non-linear joints. After a DOF reduction of each finite element part based on a dynamic super element formulation, the system equations are assembled and solved with respect to finite element degrees of freedom, allowing large translations and rotation due to a co-rotated theory.

1.3 Control systems in mechanical analysis

A mechanical system is often in a control loop including sensors measuring states in the mechanism, compensators representing control algorithms, and servo, hydraulic, or electrical actuators that generate the energy to drive the mechanism. These control modules and the mechanism must be integrated simultaneously in order to ensure sufficient accuracy of the simulation results.

To model your control systems, Fedem provides the Control Editor modeling environment, which combines the Fedem look-and-feel with graphical modeling tools similar to those of MATLAB®/Simulink®. These modeling tools are crucial when creating models of mechanical systems such as robots, milling machines, and space mechanisms. (See [Section 2.5, "Touring the interface"](#) and [Chapter 5, "Control System Modeling"](#) for more information about the Control Editor.)

As an alternative to the internal control system, Fedem provides an interface for co-simulation with MATLAB/Simulink models (see [Section 5.5, "External control systems"](#)).

1.4 What is a Fedem model?

A *Fedem model* is a virtual test model designed by you to simulate your mechanical systems. The virtual test model is an assembly of individual parts, also called *links*. The mechanical properties of the links can be represented by either an FE model or a simplified stiffness description. You connect links, virtual joints, springs, and other elements to create an accurate virtual test model of a movable mechanical system or *mechanism*.

To minimize the time needed to calculate and simulate mechanisms, Fedem's model reduction process reduces each FE model to a super element with external nodes. These external nodes are defined at FE nodes that serve as connections between the various model entities in the assembly.

Once constructed, a Fedem model retains the FE characteristics of its component links, and can therefore be treated as a *geometrically nonlinear Finite Element model*. The mechanism is then allowed to experience large translations, rotations, and nonlinear, behavior-dependent loads in the simulation of its dynamic motion.

Functions, measured time history input data and control systems can be used to model virtual test-events by controlling loads, motion, spring lengths etc.

Fedem's solvers can then calculate the motion kinematics, dynamics, structural flexibility, stresses, strains, and varying loads in one consistent model.

1.5 Using FE models

Fedem imports FE models created in external systems by interfacing with the MSC.Nastran® Bulk Data File (.*bdf* and .*nas*) format (see [Section 4.1.1, "Creating links by file import"](#)). Many FE modeling systems (including I-DEAS®, Pro/ENGINEER®, Altair® HyperMesh®, NEiWorks/NEiFusion and MSC/Patran®) can export data to the Bulk Data File format, enabling you to use your favorite FE modeling program with Fedem.

Fedem can also import models which are saved using the ABAQUS(*.inp*) format and the Ansys(*.anf* and *.cdb*) format.

Each FE model must include descriptions of its nodal coordinates, element topologies, element properties, material data, nodal attributes, and so on.

The results obtained in the simulation depend to a large extent on the FE models used. To create a satisfactory model of a real structure, the analyst must combine insight into the nature of the problem, experience with the *Finite Element Method*, and knowledge of the general rules of FE modeling.

1

1.6 CAD integration

Fedem has an integration with several SolidWorks based cad systems. This includes native SolidWorks, NEiFusion and our own CAD editor add-on: FedemSolid. This integration makes it possible to transfer a complete CAD assembly to Fedem, and use it as a starting point for the Fedem model. The Fedem model can then be updated when the CAD assembly changes.

See [Section 2.12, "Using the CAD integration"](#) for more details.

In addition (see [Section 4.1, "Links"](#)), CAD geometry from VRML files (*.wrl*, *.vrml*, *.vrl*, *.wrz*) can be imported directly.

1.7 Fedem solver modules

Fedem uses separate program modules to perform the different type of calculations on the model. Thus, there are modules for model reduction, dynamic mechanism simulation, stress and mode shape recovery, strain rosette analysis, strain coat analysis, and automated curve export. The main Fedem application manages the execution of each module. However, they may also be run separately as batch processes. The Fedem solver modules are described briefly below.



TIP: To display a list of all the modules and their version and build date, press the About Fedem entry in the Help menu.

1.7.1 Reducer

The Fedem Reducer performs a super element reduction of the FE model representing the mass and stiffness of a link. A super element reduction reduces the required DOF to a minimum. The retained DOFs, also called external DOFs, are the results of a dynamic super element reduction technique called Component Mode Synthesis reduction, also known as CMS reduction.

More information on this topics can be found in Finite Element textbooks.

1.7.2 Dynamics Solver

The Fedem Dynamics Solver module performs a nonlinear dynamics simulation. This means a simulation of the motion and deformations of the Super Elements and the Joints, as they respond to load and displacement time histories and Control System output.

In any time step of the simulation the model can be linearized whereby an eigenvalue analysis can be performed.

Link deformations and mode shapes are recovered by the *Stress Recovery* module and *Mode Shape Recovery* module.

1.7.3 Stress Recovery

The Fedem Stress Recovery module recovers the internal DOFs from the deformations of the external DOFs simulated by the Dynamics Solver. The element stresses, strains and beam forces are then calculated.

1.7.4 Mode Shape Recovery

The Mode Shape Recovery module recovers the links mode shapes from Eigen values calculated in the Dynamic Solver.

1.7.5 Strain Rosette Analysis

The Fedem Strain Rosette Analysis module recovers the stresses and strains on virtual strain gages on the links, based on the Dynamic Solver results. The output is time history data of stresses and strains similar to output from real strain gages.

1.7.6 Strain Coat Analysis

The Fedem Strain Coat Analysis recovers the stresses and strains on the strain coat elements in the model, and calculates a summary of the recovered results over its entire time history. The output from this analysis is maximums of certain stress/strain quantities over time.

1.7.7 Curve Export Utility

The Fedem Curve Export Utility module allows you to automatically export a set of curves to a single ASCII/RPC-file. The result data on which the curves are defined can be distributed on several results database files. This program module can only be run separately as a batch process.

Chapter 2 Learning the Basics

This chapter outlines system requirements, explains where models and results are saved, and introduces the Fedem user interface and common commands.

Sections in this chapter address the following topics:

- [System requirements](#)
- [Software requirements](#)
- [Storing models and results](#)
- [Starting Fedem](#)
- [Touring the interface](#)
- [Executing commands](#)
- [Visualizing the model](#)
- [Opening and saving model files](#)
- [Loading and unloading FE-Data](#)
- [Printing and exporting](#)
- [License information](#)
- [Using the CAD integration](#)

2.1 System requirements

The following are minimum system/hardware requirements and recommended settings for optimal Fedem performance.



NOTE: Use of resolution settings other than those recommended below may require you to move or resize windows and dialogs to fit your screen.

Table 2-1: Windows 2000/XP/Vista and recommended settings

Processor	Recommended: Pentium 4 @ 2 GHz Minimum: Pentium III @ 850 MHz
RAM	Recommended: 2 GB Minimum: 256 MB
Free disk space	Dependent upon the size and complexity of mechanism models *
Graphics card	Recommended: high end graphics card Required: OpenGL® API support
Pointing device	Three-button mouse with wheel
Resolution setting	1,280 x 1,024 pixels or higher

* Larger models may require several GBs.

2.2 Software requirements

Fedem interacts with other softwares in different ways. To utilize these integrations, you need the correct versions of those 3rd party systems.

The following versions of these softwares are supported by Fedem:

- CAD integration
 - SolidWorks 2008 (32 bit). Optionally with the NEiWorks 1.4.2 add-in.
 - NEiFusion 1.4.2 (32 bit)
- nCode fatigue integration:
 - nSoft 5.3.1/FE-Fatigue 5, ISR 3 or ISR 4

- nSoft 2004/FE-Fatigue 6
- Simulink integration for external control system
 - MATLAB 6.1/Simulink 4 or
 - Matlab 6.5/Simulink 5
- FTire Tire/road integration
 - FTire v2.7 (1 October 2008) and later

2.3 Storing models and results

Fedem uses the following files and directories to store the contents of a Fedem model:

- The mechanism assembly description and all simulation parameters are saved in the Fedem Mechanism Model (*.fmm*) format. This file is called the model file and contains the complete description of your model except the link files and the simulation results.
- The link files and all the simulation results are saved in a directory named *<modelfilename>_RDB* (the model name specified by the user is substituted for "*<modelfilename>*"). Fedem creates this directory in the same location as your model file. Within this directory, the link files are stored in a directory named *link_DB*, unless a *link repository* is used (see [Section 4.1.5, "Using link repositories"](#)).

[Appendix D, "File Types and Usage"](#) contains more information about Fedem file types, whereas the directory structure is shown in [Section 8.3, "RDB directory structure"](#).

2.3.1 FTL format

The Fedem Technology Link (*.ftl*) format is used to store FE models as sets of data in text format (see [Section C.1, "Fedem Technology Link format"](#) for more information about the *.ftl* format).

2.3.2 FTC format

The Fedem Technology Cad (*.ftc*) format is used to store CAD geometry in text format.

2.3.3 Other supported formats

Fedem can import FE models (see [Section 4.1.1, "Creating links by file import"](#)) in the MSC.Nastran Bulk Data File (*.bdf* or *.nas*) format (see [Section C.2, "Nastran Bulk Data File format"](#)), the ABAQUS(*.inp*) format and the Ansys(*.anf* and *.cdb*) format.

Earlier versions of Fedem used the Fedem Link Model (*.flm*) format for FE models. These files can also be imported and saved to the *.ftl* format.

Fedem can also import CAD geometry from VRML files (*.wrl*, *.vrm1*, *.vrl*, *.wrz*) for visualization of *Generic Parts* (see [Section 4.1, "Links"](#)).

2.4 Starting Fedem

Fedem can be started from a compatible CAD system, from the start menu, or from the command prompt.

On windows systems:

- To start Fedem with a new, empty model, press the Fedem icon on the windows desktop or in the Programs menu.
- To start Fedem and open an existing model, double-click the icon representing the desired model file in windows explorer.
- Open a command prompt, navigate to the installation directory and enter *fedem* optionally followed by some command line options. (See [Section E.1, "Fedem UI Options \(fedem\)"](#))
- To start Fedem from the CAD system, refer to [Section 2.12, "Using the CAD integration"](#)

On unix systems:

- To start Fedem with a new, empty model, just type *fedem* at the command prompt.
- To start Fedem and open an existing model, type *fedem -f <modelname>.fmm* at the command prompt, substituting "*<modelname>*" with the path and name of your model file.

When Fedem is started with a new, empty model, the model file is named *untitled_<#>.fmm*, where *<#>* indicates a unique running number. This empty model file will contain only some default settings, animations and graphs. The file will be created in your default login directory when Fedem is started from the windows desktop icon or Programs menu.

Command-line options

Fedem has several command-line options that can be used to achieve different tasks. These options can be invoked when starting Fedem from the command prompt. See also [Appendix E.1, "Fedem UI Options \(fedem\)"](#).



TIP: You can create a new model file with a specific name by specifying a path and filename that does not yet exist, using the `-f` option. The file will be created for you and opened when Fedem starts.

2.4.1 Template model file

The default settings of a new model are stored in a template model file which is loaded whenever a new model is created (either when Fedem is started, or when using the `New` command, see [Section 2.8.3, "Starting a new model"](#)). The template file can be modified or changed at will, and is located in `$FEDEMDIR/Templates/default.fmm`.

If a different location for the template file is wanted, the environment variable `FEDEM_TEMPLATE_FILE` can be defined to contain the desired path to the template file.

Predefined template files

In the Templates directory in the Fedem installation you will find tree files. `default.fmm`, `default_m.fmm` and `default_mm.fmm`. The `default.fmm` and the `default_m.fmm` are identical and suitable for models using SI unit set, while the `default_mm.fmm` contains default settings that are suitable for models using millimeter, newton and mega gram as unit set. Simply copy the `default_mm.fmm` to `default.fmm` if you prefer mm/N/Mg as your default unit set.

2.4.2 Console window

When Fedem is started on a windows system, a separate console window can be opened in addition to the Fedem main window, by specifying the command-line option `-console`. This window lists some startup messages, and may also contain some low-level error messages during a Fedem session, that normally can be ignored by users, but sometimes might help understanding an abnormal incident.



WARNING! Once opened, the separate console window must remain open throughout the whole Fedem session; closing it explicitly closes the entire Fedem session immediately, without the option to save unsaved work.



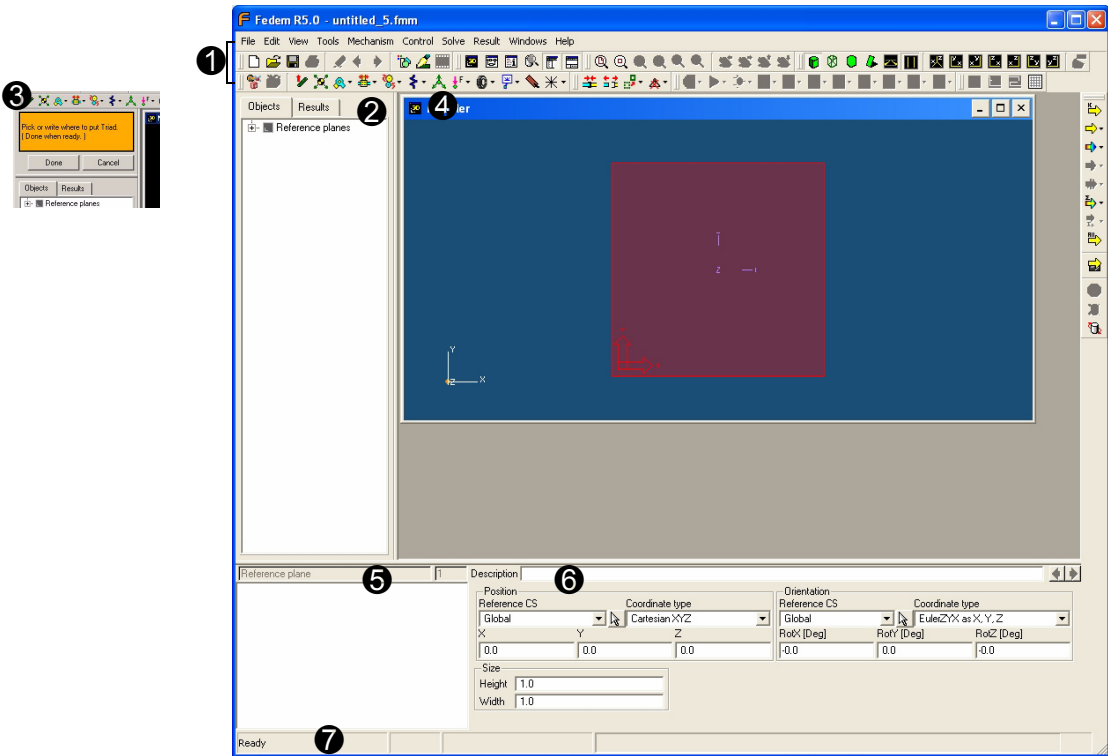
TIP: If abnormal behavior occurs, or if Fedem stops responding, check the console window (if activated) and the output list (discussed later) for important information. Take note of any errors or warnings before contacting Fedem technical support (support@fedem.com).

Once the initial tasks are completed, the Fedem main window is displayed. The following sections describe Fedem’s user interface.

2.5 Touring the interface

2.5.1 Fedem’s main window

Starting with an empty model file, the Fedem main window appears as shown below.



- ❶ *Menus and toolbars* – contain buttons used to initiate commands.
- ❷ *Model Manager panel* – contains the Objects and Results tabs, which allow you to create, manage, and delete the objects in your model and define animations and graphs of your results.
- ❸ *Guide panel* – This guide will pop up above the Model Manager panel when you invoke a command. It will tell you what you are expected to do during the different steps of the command. It also provides a Done and a Cancel button used to accept choices or to cancel the command.
- ❹ *Workspace area* – contains the Modeler, Control Editor, and Graph View windows for constructing and viewing models and results.
- ❺ *ID and Topology panel* – contains a list of objects related to the selected item.
- ❻ *Property panel* – allows you to view and edit the properties of individual objects in your model.
- ❼ *Status bar* – provides information of the status, progress information and whether some solver is running.

2.5.2 Menus and toolbars

Fedem commands are initiated from buttons on the toolbars and menus. All Fedem commands can be accessed from the menus, while the toolbars display only some of the most commonly used commands. The menus are arranged from left to right in logical order of task performance, starting with standard file functions, then continuing with viewing commands, mechanism modeling, control system modeling, analysis/solution tools, and finally results management.

Command sensitivity

Menu and toolbar buttons are sensitive to the active window and object selection. For example, if an object in the modeling window is selected, the graph view controls appear dimmed (grayed-out) and cannot be selected.

Fedem toolbars

Fedem uses the following toolbars (a toolbar handle to the left (or on top) marks the beginning of a toolbar):

- *Standard*: provides standard file operations such as Open, Save, Exit, and selecting and deleting objects.

- *Windows*: provides commands for controlling the active window selection (Modeler, Control Editor, Output List, Result File Browser) and hiding/showing the Model Manager and Property Editor panels.
- *Zoom and Pan*: provides commands for zooming and panning the two-dimensional display of graph views. (See [Section 2.7.3, "Zoom and Pan"](#) for more information about these commands.)
- *3D View Control*: includes commands for rotating the view and changing the view perspective in the Modeler window. (See [Section 2.7.2, "3D View controls"](#) for information about manipulating the view.)
- *Solvers*: these commands are used to set up, start, and stop mechanism analyses and calculate specific results. (See [Chapter 6, "Mechanism Analysis"](#) for information about these commands.)
- *Mechanism Creation*: contains commands for importing links and creating mechanism entities such as joints, springs, dampers, forces, and sensors (see [Chapter 4, "Mechanism Elements"](#)).
- *Mechanism Tools*: provides commands used in modeling such as moving, attaching, copying objects, and applying motion constraints. (See [Chapter 4, "Mechanism Elements"](#) for information about modeling tools.)
- *Control Creation*: provides commands for creating control objects. (See [Chapter 5, "Control System Modeling"](#) for detailed descriptions of control objects.)
- *Control Tools*: contains commands for manipulating the control system. (See [Chapter 5, "Control System Modeling"](#) for information about modeling control systems.)

Manipulating toolbars

Only some of the commands accessible from the toolbars are displayed. You will see that there are arrows (▼) beside some of the icons. To access other commands, click and hold down the ▼ button. A pull-down list of additional, related commands is displayed (example at right). Selecting a command from the list initiates the command and replaces the button's function with that of the new command. You can select another option at any time by clicking, holding down the ▼ button, and selecting a different command.



Fedem provides several ways to manage toolbars:

- Right-click a toolbar handle and select an option to relocate, show, or hide the toolbar.
- Right-click an empty space in the toolbar area and select an option from the list to show or hide a toolbar.
- Double-click a toolbar handle to show or hide the toolbar.
- Drag a toolbar handle to the left, right, top, or bottom of Fedem's main window to relocate the toolbar.

2.5.3 Model Manager

The Model Manager panel contains lists of all the objects and result views that make up your model. This includes mechanism, modeling, and control system objects, along with animations and graphs. In each list, objects are grouped by type and sorted by identification numbers or names. In the Model Manager, right click menus can access several commands that can be applied to the selected objects.



NOTE: The Objects and Results lists are empty (or nearly empty) until you create items.

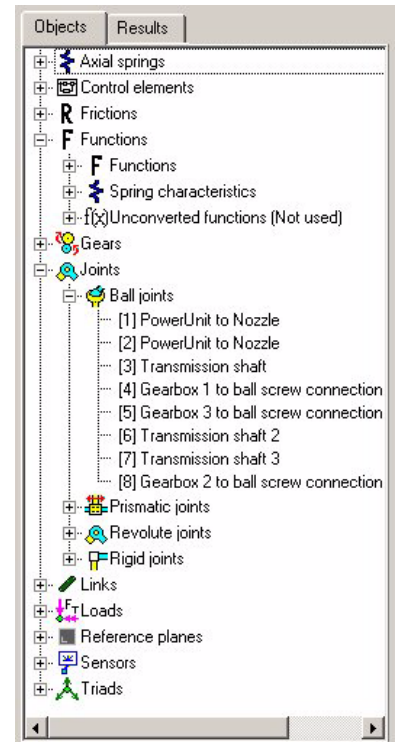
Selecting items

In the *Model Manager*, you can select items in several ways:

- Highlight a single item.
- Hold down the **Shift** key and click multiple items.
- Hold down the **Ctrl** key and click multiple items (or a single item to select/deselect).
- Click and drag the mouse over multiple items.

Deselecting items

To deselect all items, right-click an empty space in the Model Manager panel.



Sorting

The objects in the *Model Manager* can be sorted, either by ID-number or by item name. To switch sorting mode, right click inside the *Model Manager* area, and select either *Sort by ID* or *Sort by Name* in the menu. Default sorting in the *Model Manager* is based on the ID-numbers.



TIP: Right-clicking in the *Model Manager* will display a pop-up menu with commands that applies to the current selection. The contents of this menu depend on the type of the selected object.

Objects

The *Objects* tab displays a list of all modeling objects in your model. Selecting an item from this list highlights it in the *Modeler* or the *Control Modeler* and displays its properties in the *Property Editor* panel.

Results

The Results tab displays a list of the result views you have created. The available result views are graphs which contains curves (individual sets of plotted data), and animations.

Selecting a graph from this list will cause the window containing the selected graph to pop up, if loaded. Selecting a curve will highlight it (the curve is rendered red) and raise the graph view in which it resides as well.

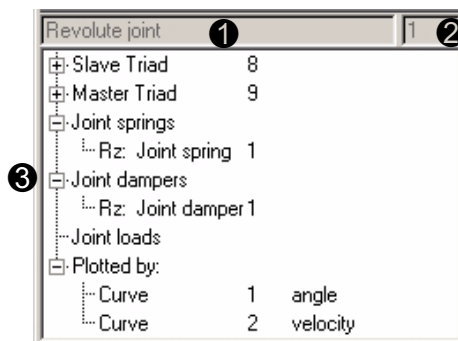
The properties of the different objects is shown in the *Property Editor* panel when they are selected.

See [Chapter 7, "Postprocessing Results"](#) for more information about graphed and animated results.

2.5.4 ID and Topology panel

When an object in the model is selected, the ID and Topology panel displays information related to the object as described below. If multiple items are selected, only the item selected last is displayed.

- ❶ *Item Type* – the type of the selected object (for example, revolute joint, gear, or spring).



- ② *ID Number* – a unique integer that distinguishes one item of the same type from another.
- ③ *Topology View*– a list that displays the mechanism objects related or connected to the selected item.



TIP: The *Plotted by branch* in the *Topology View* lists all curves that plot result quantities from the selected structural object.

ID numbers are assigned automatically to new objects in numerical order. If you delete an object such as a ball joint with ID Number 3, this number is free and the next ball joint created is then assigned ID Number 3.

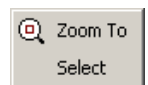
Topology View and browsing

Most mechanism elements are related to other objects; for example, joints consists of triads that are connected to links, and sensors are measuring variables from other mechanism elements.

These relations are shown by the *Topology view* in a hierarchical fashion, indicating their topological relationships. This list can then be used to investigate these related objects, and to browse through the mechanism model through the topological connections. This can be done by using the browsing features offered by the *Topology View* described below.

Temporary highlight: The items in the topology view will be highlighted in the Modeler when you select an item keeping the mouse button pressed. This is useful to see exactly which objects in the 3D view that corresponds to the listed item.

Right clicking an item in the list will show a tiny pop-up menu that allows you to either *Zoom to* or *Select* the item.



Zoom To: This command zooms to the item, making it easy to locate it in a complex model. See also [Section 2.7.3, "Zoom and Pan"](#).

Select: This command will select the right clicked element from the topology view, and thus jump to it showing the properties of that item instead. See also [Section 2.6.1, "Select"](#) on how to get back to your previous selection.



TIP: Double clicking an item in the topology view will also select it.

2.5.5 Property Editor

The Property Editor panel is used to view and edit the properties of mechanism items. The appearance of properties is different depending on the object selected. The image below is an example showing properties that are common to some of the Fedem modeling objects.

① *Description* – an optional user-supplied name or identifying remarks

② *Properties* – editable attributes specific to the selected object



NOTE: The description field may contain any ASCII-character, except for the "-" character which is reserved for text string delimiters in the model file.



NOTE: The description field is also used to activate beta features, see [Appendix F, "Beta feature documentation"](#).



CAUTION: After editing a value in the Property Editor by typing, you must press the **Enter** key to apply the change.

Property menus

Many mechanism items have internal properties that can depend on the simulation time, measurement of a system variable or some internal variable in the item in question. The *Property menus*, are used to set up such properties in a simple way. These menus consists of an option menu that in some cases are editable, and an edit button.

The menu contains references to other entities in your model that can be used as input for the property in question. A non-linear spring characteristic (e.g. a force displacement curve) will be listed in the stiffness property menu of an axial spring, functions and control outputs will be listed in the magnitude property menu of a load etc. Press the menu button ▼ to access the list.



By pressing the *Edit button* you will select the item shown in the menu, as if you had selected it in the *Model Manager*. This is a convenient way of navigating through the relations of the model, giving a simple way of finding the details of a complex model.

In some property fields a numeric value can be entered instead of selecting a reference from the menu. The spring stiffness property is a good example. Entering a numeric value will assign that value as a constant spring stiffness to the spring in question.



TIP: Property menus that accept a numeric value always have a numeric value as default, while "None" is normally the default for Property menus not accepting a number.

To change a property from referring to constant, select the top entry in the pull-down list (which is either *None* or the last numerical value that was entered in the same box), or delete the current contents of the box, type in a numerical value and then press *Enter* to apply the change.

2

2.5.6 Workspace

The Workspace area is used for constructing, manipulating, and viewing mechanism models, control systems, graphed results, and animations. The Workspace can contain several windows, including the Modeler, Control Editor, and multiple windows for viewing graphed results. These windows are described in the following sections.

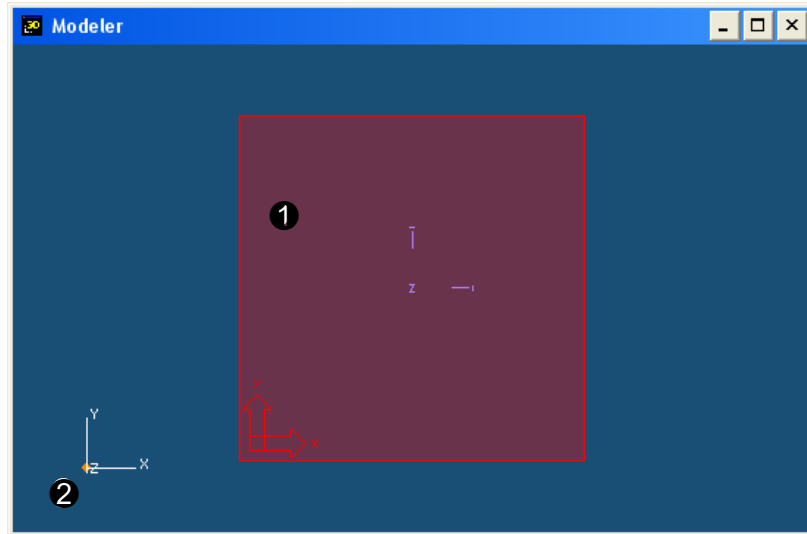


TIP: Windows in the Workspace can be managed using the *Tile* and *Cascade* commands from the *Windows* menu.

Modeler

This window displays a 3D view of your mechanism and provides dynamic viewing capabilities such as zooming, panning, and 2- and 3-dimensional rotation (see [Section 2.7.1, "3D Navigation"](#)). The Modeler window is also used to view your animated simulation results. Select this window to view, create, or edit a mechanism model. To open the Modeler window, select **Show Modeler** from the Windows toolbar or menu. The Modeler is displayed as shown below. (See also [Section 3.2.1, "Modeler window"](#).)





- ❶ *Reference Plane* – The shaded area in the center of the Modeler window represents a plane, which can be considered the ground or base for your models.
- ❷ *Global Directions* – The arrows located in the lower left corner of the Modeler show the orientation of the global coordinate system and the direction of the gravity vector, **g**.



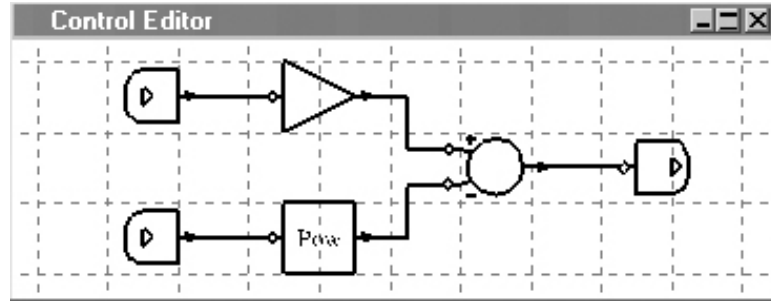
TIP: The Modeler window can be viewed at almost full screen size by hiding the Model Manager and Property Editor panels. To hide these panels, click the **Model Manager** and **Property Editor** buttons on the Windows toolbar (or View menu). Hiding the toolbars also increases the viewing area of the Modeler window (see ["Manipulating toolbars"](#) in [Section 2.5.1](#)).

Control Editor

The Control Editor is a workspace for configuration of the control system. The graphical representation is a block-based diagram that consists of a series of control blocks that can be connected to simulate your control system. This editing environment allows you to create and manipulate the control system using drag-and-drop functionality. It also features grid and snapping tools (see [Section 5.4.1](#), ["Setting Grid and Snap"](#)).



To open the Control Editor, click the **Show Control Editor** button on the Windows toolbar (or Windows menu). The Control Editor displays the control system (an example is shown below). See [Chapter 5](#), ["Control System Modeling"](#) for more information about control systems.

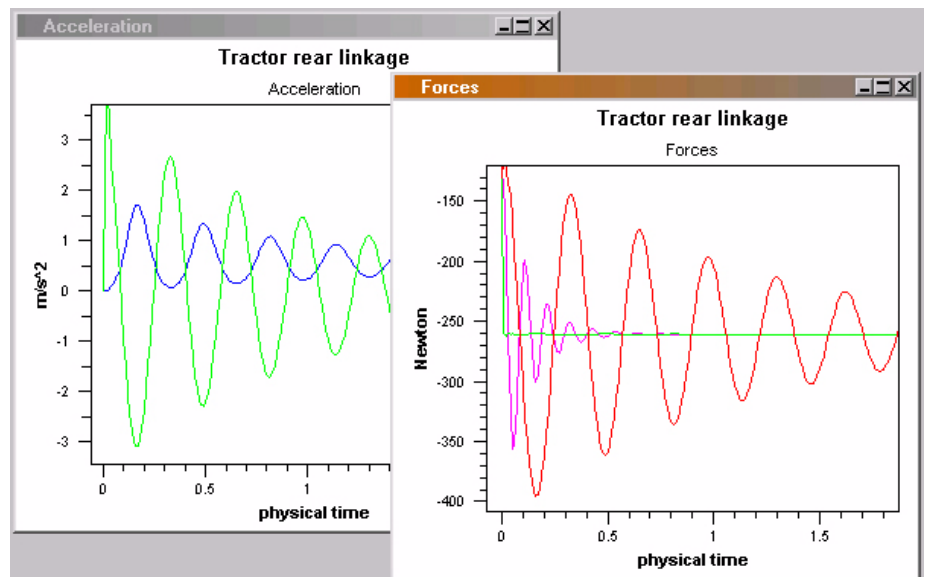


NOTE: The Control Editor is empty until you create control elements.

Graph Views

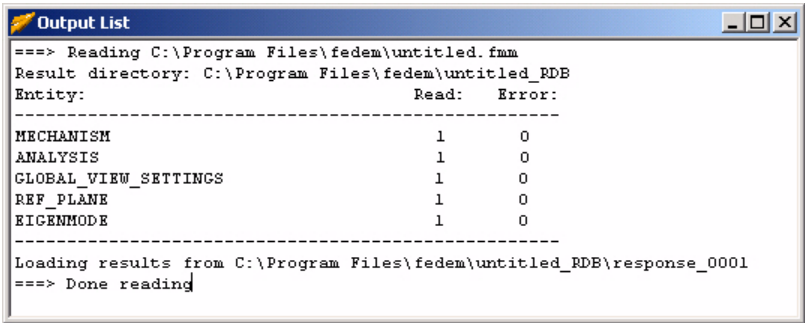


The graph windows can display various views of your results. You can customize graphs of selected simulation results and manipulate the view in the graph window. To open a graph window, right-click the graph from the Model Manager Results list and select **Show Graph**. Graph Views are displayed in the Workspace area as shown below.



2.5.7 Output List

The Output List window displays written output from Fedem, such as a log of commands and solution, and error messages. This window allows you to observe the commands performed by Fedem. To open the Output List window, select **Show Output List** from the Windows menu or toolbar.



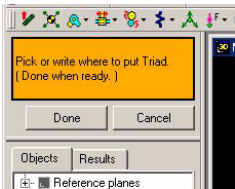
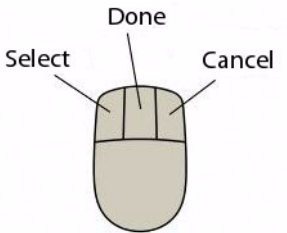
The text in the Output List window is also written to a log file. The name of this log file is the same as the current model file name, but with extension *.log* instead of *.fmm*. Therefore, a new log file is opened whenever you **Open** a new model, or perform a **Save As...** command.



NOTE: If a log file already exists for the model you open from an earlier Fedem session, the output from the current session is appended to that file. That means that the entire history of the Output List content for the model is recorded. In addition, the date and time of the model open and close operations, as well as the Fedem version used, are recorded to the log-file.

2.6 Executing commands

When performing commands in Fedem, the *Guide panel* prompts you with instructions for completing each command. You may be asked to select mechanism objects or locate points to place or move objects. Performing commands makes use of three actions: Select, Done, and Cancel.



2.6.1 Select

To select items in your model, or points on objects as references for moving or creating items, place the cursor over the object or position and press the left mouse button (left-click). The item is highlighted and its properties can then be edited in the Property Editor panel.



NOTE: Some commands require that an object is selected from windows in the Workspace area only, such as the Modeler or Control Editor. Instructions regarding these commands are provided in the Guide panel.



TIP: To deselect an item, simply click an empty space within the Modeler window.

Snapping

When selecting objects in your model, the selection automatically snaps to a point on the object such as the nearest node on an FE link, the center point of a joint, and so on. This makes your selection quick and accurate.

Snapping behaves differently on different types of objects. FE parts, VRML models, CAD parts and mechanism symbols all have different snapping policies.

Selection snaps to FE nodes on FE parts, to vertices on a VRML part, to geometry features such as center points and edges on CAD parts, and to important points on mechanism symbols.

Multiple selection

Some Fedem commands, such as Smart Move and Delete, allow you to select several items at once. To select more than one item, press and hold down the **Ctrl** key and then click the items you want to add to your selection.

If you accidentally add the wrong object to the selection, simply release the **Ctrl** key and click an empty space within the Modeler window. The last selected item is deselected.

Selection history

Fedem maintains a history of the items you select in the current session. This history can be accessed using the Select Backward and Select Forward commands.



- To choose a previous selection, press the **Select Backward** button. You may need to press it several times to cycle back through the selections until the desired object or selection is reached.



- To select a recent selection, press the **Select Forward** button once or as many times as necessary until the desired selection is reached.

Co-located items

Sometimes several items in a model are located very close together or on top of each other; in Fedem, these are called *co-located items*. To select a co-located item, click the same spot several times to cycle through the items. Fedem cycles from the item closest to the viewer to the one furthest from the viewer.

Selection Filter

Some Fedem commands allow you to select only certain types of items. These restrictions are automatically imposed and based on the type of command in use. For example, sensors cannot be applied directly to links, and Fedem will therefore limit your selection to other types of mechanism elements. To make the selection even easier, you can filter the selectable items by limiting the types of items displayed in the Modeler.



TIP: To limit the display of mechanism elements, click the **General Appearance** button on the Standard toolbar, then disable Mechanism Symbols as necessary. (See also [Section 2.7.4, "General Appearance"](#).)

Selecting ground

To select the ground during modeling, simply click anywhere on the Reference Plane.

2.6.2 Done

When executing a command in Fedem, the *Guide panel* prompts you to select items or locate points. When you have achieved the desired selection, press the *Done* button in the *Guide panel*.



TIP: You can also press the center mouse button within one of the modeler windows or the Enter key on the keyboard to accept the selection.

2.6.3 Cancel

To abort or escape a command procedure, press the *Cancel* button on the *Guide Panel*.



TIP: You can also press the right mouse button within one of the modeler windows or the Esc-key to cancel a command.

2.7 Visualizing the model

2.7.1 3D Navigation

The 3D navigation commands enables you to change the view without interrupting the current command or procedure. There are several different sets of viewpoint control commands:

- Mid mouse button/wheel + mouse motion
- Function keys (F1, F2, F3 and F4 + mouse motion)
- Keyboard keys
- Predefined view tool buttons

Middle mouse button/wheel

Pressing and holding the middle mouse button while moving the mouse will rotate the view around the rotation center. If the button was pressed near the edge of the 3D view, the rotation will be restricted to the viewport's normal axis.

Rolling the mouse wheel will zoom in and out. When the mouse wheel is used to zoom in, the view will be zoomed towards the position of the mouse pointer giving a combined zoom and pan behavior.

Using the middle mouse button commands is the most used 3D navigation in Fedem.

Function keys

The function key based navigation can be used when you need added control over the navigation. The commands include:

- Pan  (F1)
- Zoom  (F2)
- Rotate  (F3)
- Select Rotation Center  (F4)

These functions are often useful while working in the Modeler. We recommend that you keep your left hand near these function keys while you work.

To use the function key commands, press and hold the function key, and move the mouse to manipulate the view. The manipulation will only

occur as long as the mouse is inside the Modeler view. By pressing the left mouse button, you may avoid this restriction.

When the function key is released, the manipulation stops.



CAUTION: When pressing the left mouse button while using the function keys, Fedem grabs the mouse and keyboard control.

Pan - (F1)

The Pan  command shifts the view left, right, up or down.

Zoom - (F2)

The Zoom  command moves the scene closer or further away from the camera. It pays attention to the rotation center, and will zoom towards it (see "[Select Rotation Center \(F4+Select\)](#)" below). This is useful when you need to examine an object or its components closely.



TIP: To achieve maximum zoom at a specific point, select the point using **Select Rotation Center (F4)** and then zoom in on the point using **Zoom (F2)**.

Rotate (F3)

The Rotate  command enables you to dynamically rotate your model around a point or an axis at the rotation center (see "[Select Rotation Center \(F4+Select\)](#)" below). The rotation can be performed in two different ways, depending on the position of the cursor when you press F3.

- Axis rotation: With the cursor near the edge of the Modeler window, the view rotates around an axis that is perpendicular to the screen.
- Point Rotation: With the cursor near the center of the window, the view rotates around a point located at the rotation center of the Modeler some distance into the model. This allows rotation of the view in any direction around the point.



TIP: The rotation motion is sensitive to the speed of the mouse. If the mouse is moved slowly the control of the rotation becomes finer and makes it possible to accurately control the view along long constructions.

Select Rotation Center (F4+Select)

The Select Rotation Center  command enables you to select a new center for zooming and rotation. When you use the F4 function to select a point, the selected target point shifts to the center of the window and becomes the new dynamic center used by the Zoom (F2) and Rotate (F3)

commands. This target point remains the dynamic center until the model is moved by some other view command.

To select a new rotation point, press and hold down the **F4** key, move the cursor to the target point, and click the left mouse button.



TIP: To closely examine a part of your model, set the dynamic center (**F4**) at the point of interest, and use **Zoom (F2)** to magnify the view. You can then easily examine the point from many directions using **Rotate(F3)**.

Keyboard keys

To rotate the model by increments of 15 degrees, use the arrow keys. Rotation and panning may also be done by pressing Shift (rotate 90 degrees), Alt (rotate around screen normal) or Ctrl (panning) in combination with arrow keys. To zoom in and out, press z or Shift + z.

Predefined view tool buttons



Isometric

To display an isometric view of your mechanism, click the **Isometric** button.



Top

To display the top view of your mechanism, click the **Top** button.



Right

To display the side view of your mechanism, click the **Right** button.



Front

To display the front view of your mechanism, click the **Front** button.



Bottom

To display the bottom view of your mechanism, click the **Bottom** button.



Left

To display the left side view of your mechanism, click the **Left button**.



Back

To display the back view of your mechanism, click the **Back view** button.

2.7.2 3D View controls

Fedem provides several 3D viewing commands for use in the Modeler window. The following commands can be accessed on the 3D View Control toolbar (or View menu):



Solid View

To display all mechanism elements as solid/shaded objects, click the **Solid View** button. This is on by default.



Line View

To speed up graphic performance and display all mechanism elements as outlines, click the **Line View** button.



Flat Colors

To render the model without shading, click the **Flat Colors** button. This is especially useful when viewing color plots. This option is off by default.



Show Top Faces

To distinguish the top and bottom faces of shell elements, click the **Show Top Faces** button. The top faces will then be rendered normally while the bottom faces will be rendered dark/black. If some of the links have the detail level set to Reduced Surface or Reduced Surface and Internals, only a rough indication on the states of the faces is given. To see the exact top and bottom of every face on a link, set the detail level to Surface or Surface and Internals. This is off by default.



Perspective

To display a perspective view of your mechanism, click the **Perspective** button. This command controls the appearance of mechanisms in depth as perceived by normal binocular vision.



Parallel Projection

To display a parallel view of your mechanism, click the **Parallel Projection** button. This is the default projection.

2.7.3 Zoom and Pan

Fedem provides zooming and panning controls for use in the active view - for example, graph views. The following commands can be accessed on the Zoom and Pan toolbar (or View menu):



NOTE: Some of these commands cannot be used in all windows. When commands cannot be used in the current view, their buttons become unavailable (grayed out) on the menus and toolbars.



Zoom All

To scale the active view so that all objects (for graph views, every curve on the graph) fit within the view, press the **Zoom All** button. When working in graph views, this can also be achieved by pressing the F5 key.



Zoom To

This command pops up the correct view, zooms to the selected object, and places the *Dynamic Center* of rotation at the center of the object. This is very useful when trying to locate a certain triad or joint in a large model. This command is also available from the *Topology* view and the *Model Manager* (see [Section 2.5.3, "Model Manager"](#) and [Section 2.5.4, "ID and Topology panel"](#)). It is also applicable on Control Elements and Control Lines in the Control Editor view (see [Section 5.1.1, "Control Editor"](#)).



Zoom Window

To enlarge a rectangular area, press the **Zoom Window** button. The command can also be activated by pressing the Z key.



Zoom Window With Auto scale

To enlarge a rectangular area, press the **Zoom Window With Auto scale** button. The contents will be auto scaled to fit the entire plotting area. This command can also be activated by pressing the X key.



Zoom In

To enlarge the active view by a predefined scale factor, press the **Zoom In** button.



Zoom Out

To reduce the active view by a predefined scale factor, press the **Zoom Out** button.



Pan Left

To move the active view to the left, press the **Pan Left** button.



Pan Right

To move the active view to the right, press the **Pan Right** button.

**Pan Up**

To move the active view up, press the **Pan Up** button.

**Pan Down**

To move the active view down, press the **Pan Down** button.

2.7.4 General Appearance

The General Appearance command can be used to control which entities are displayed in the Modeler window. This command also provides control of the size and appearance of mechanism symbols. Click the **General Appearance** button to display the window.



- ❶ *Mechanism symbols* – allows you to toggle on/off the display of mechanism elements, edit the color used to specify each item type, and change the size of symbols and line widths (see "*Mechanism symbols*" below).
- ❷ *Default colors* – controls the colors used for Grounded triads, unattached mechanism items and the modeler background (see "*Default colors*" below).
- ❸ *Viewer options* – controls 3D rendering options such as visibility, transparency type, and line-smoothing (see "*Viewer options*" below).

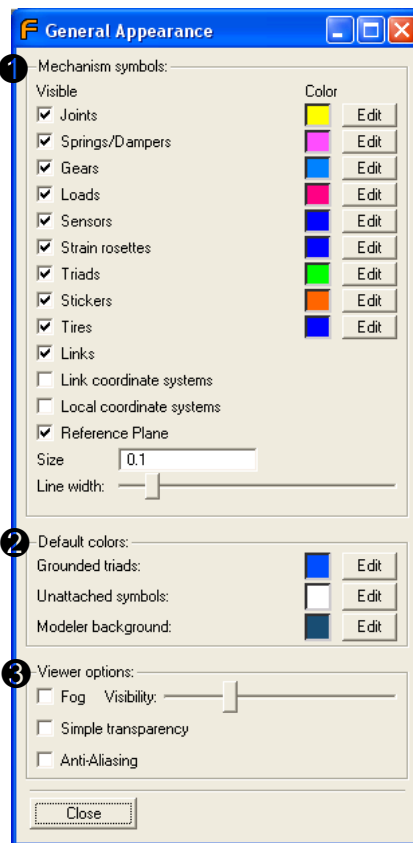
Mechanism symbols

This area provides the following controls:

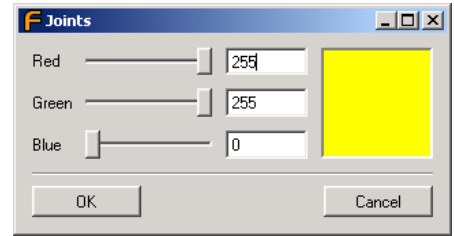
- *Visible* – Enables/disables the display of each item type. Click the box next to an item type to change the setting.



TIP: Turning off items speeds up and simplifies the display of complex mechanisms, and provides a useful way to limit selection to specific items.



- **Color** – Allows editing of the RGB settings for each item type. Press the **Edit** button next to an item type to edit the default display color for that item. In the item window (shown at right), move the sliders to change the settings or enter the desired values directly in the number fields.
- **Size** – Size is a scale factor for sizing the display of mechanism entities. To change the size of items, enter a new number in the size field.
- **Line Width** – This is a scale factor for the line-width of all mechanism symbol lines, together with all 1D elements and *Surface Connectors* in the FE parts (see [Section 3.6.2, "Surface Connectors"](#)). To change the line-width, move the slider right or left.



Default colors

This area enables the user to edit the colors used on the modeling background and unattached mechanism items. It also allows you to set the colors on triads that are attached to the ground, to distinguish them from triads that are free to move. The default colors may be changed in a similar ways as changing the colors for *"Mechanism symbols"*.

Viewer options

This area enables the user to modify the way that models are rendered.

- **Fog** is an option that enables you to create a fog-like effect around your model that appears as fog or darkness (or even an underwater scene). The distant parts of the model appear to fade into the background. The Visibility slider controls the distance at which the model is completely hidden in the "Fog."



TIP: You can create the following effects with the Fog option:

- To create the effect of a foggy day, set the Modeler background color to light gray (Red 180, Green 180, Blue 180), enable the **Fog** option, and adjust the **Visibility** slider until the model almost fades into the background.
- To create the effect of an underwater scene, set the Modeler background color to sea green (Red 20, Green 125, Blue 130), enable the **Fog** option, and adjust the **Visibility** slider until the model nearly fades into the background.
- **Simple Transparency** is a dithering algorithm used to speed graphic performance when displaying transparent objects in your model. The effects of this option depend on the type of graphics card you have.



- *Anti-Aliasing* enables/disables line-smoothing for symbols.

TIP: If *Anti-Aliasing* does not function properly, try enabling the **Simple Transparency** option.



NOTE: Graphics cards do not all have the same optimal settings. In general, disabling the **Fog** and **Anti-Aliasing** options and turning on **Simple Transparency** gives the best performance, but with some systems the performance gain is insignificant.

2.7.5 Item Appearance



The *Item Appearance* command can be used to change the level of detail and the appearance of individual links and the Reference Plane. To open the Item Appearance window, click the *Item Appearance* button on the Standard toolbar, and select a link or the Reference Plane. The Item Appearance window is displayed as shown below.

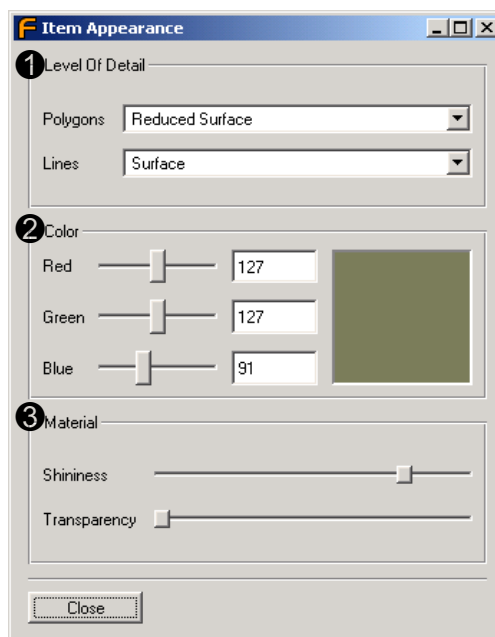


TIP: To change the appearance of a hidden link, select the link in the Model Manager Objects list after clicking the **Item Appearance** button.

- ① **Level of Detail** – controls the level of complexity displayed in the model.
- ② **Color** – controls the RGB settings of the selected link or Reference Plane.
- ③ **Material** – controls the shininess and transparency of the selected link or Reference Plane.

In the Level of Detail area, the *Polygons* and *Lines* settings allow you to change the complexity of models displayed in the Modeler. Changing these settings can improve the graphic performance of 3D rendering.

Polygons can be displayed at five levels of detail: *Surface and Internals*, *Reduced Surface and Internals*, *Surface*, *Reduced Surface*, and *Off*. The default level is *Surface*.



1. *Surface and Internals* - With polygon detail set to Surface and Internals, element faces from solid elements inside the links are shown together with the surface faces of the elements. All element faces are shown as single polygons.
2. *Reduced Surface and Internals* - Setting the detail level to Reduced Surface and Internals displays a simplified polygon representation of the surface and internals of a link. This is faster than using *Surface and Internals*, but shows a less accurate representation of the links.
3. *Surface* - This option turns off the internal faces in a link, and will only show the surface element faces of a link. All the surface faces are shown as separate polygons.
4. *Reduced Surface* - Setting the polygon detail to Reduced Surface provides the most efficient way to visualize the shaded view of a mechanism link. The surface is displayed using a simplified polygon model and the internal faces are turned off.
5. *Off* - Setting the polygon detail to Off turns all polygons off.

Lines can be displayed at six levels of detail: *Full*, *Surface*, *Outline*, *Outline No 1D-elements*, *Simplified*, and *Off*. The default level is *Outline*.

1. *Full* - With line detail set to Full, mesh lines from solid elements inside the links are shown together with the surface mesh of the elements.
2. *Surface* - Setting the line detail level to Surface displays only the mesh lines on the surface of the FE model.
3. *Outline* - This option leaves only the mesh lines on the surface of the link with neighboring element faces with a relative face-angle above a certain threshold. The default threshold is $\pi/4$. (It is possible to edit this value for each link by editing the model file.)
4. *Outline No 1D-elements* - Same as Outline except that all *Surface Connectors* and 1-D elements such as RGD, WAVGM, BEAM2 are also removed from the display.
5. *Simplified* - This option generates a simple line visualization of the link based on the Triads attached to it. One line is drawn from each Triad to their geometrical center. This option makes most sense if the polygons are turned off. This visualization is the same that is used if the link is not loaded, see [Section 2.9.1, "FE-Data Settings"](#).
6. *Off* - Setting the line detail to Off turns all mesh lines off.



TIP: To edit the appearance or level of detail on several links at the same time, select multiple links in the Model Manager Objects list after clicking on the **Item Appearance** icon.

2.7.6 Element face visibility

The visibility of element faces can be controlled using the *Hide Element Faces* and *Show Element Faces* commands in the right click menu in the Objects View. These commands can be applied to the entire link, or to a selection of the element groups listed in the Objects view. The icon in front of each group entry in the list indicates the current visual state of the elements in that group: Either all visible , some visible  or all hidden .

These commands will only be active when the polygon detail level is set to *Surface* or *Surface and Internals*, or when color contour results are loaded for the link.

This feature can also be utilized to load color contour results only for small parts of a big link because the color values will not be loaded on hidden elements. See also [Section 7.3.4, "Performance of animation loading"](#).



NOTE: The visibility of the mesh is not affected by the Show/Hide commands.

2.7.7 Visualization of special finite elements

There are several finite element types that are treated differently from normal elements like shells and solids, when it comes to visualization. Those elements are listed in the table below.

Table 2-1: Line style of special finite elements

Element type	Visualization	Comments
Beams	Dash dot lines	The eccentricity is ignored in the visualization
Rigids	Dashed lines	
Constraint elements (RBE3, WAVGM, Distributed coupling)	Dotted lines	
Concentrated mass	No visualization	
Springs	No visualization	
Bush elements	No visualization	

Color

The color of those elements are set automatically to black, white or a grayish color to achieve a good contrast to both the color of the link, and the viewer background. If the FE part is displayed using lines only, the color of those elements is set to the color of all the other lines.

Line width

The line width is adjusted according to the Line Width parameter set for the Mechanism symbols. See [Section 2.7.4, "General Appearance"](#).

2.8 Opening and saving model files

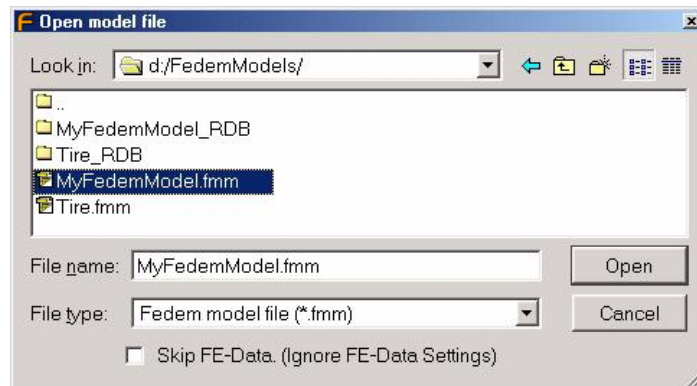
2

2.8.1 Opening a file

You can open a Fedem model file created in this version (or any of the previous versions) of Fedem through the following steps:



1. Chose **Open** in the File menu. You can also use the file open icon in the toolbar.
2. Locate the Fedem model you want to open and click **Open**.



The model file selection dialog normally displays all files with the Fedem model file extension (. *fmm*). However, you may open a file with any extension by selecting the *All files* filter in the *File type* pull-down.



CAUTION: If you choose to open a file without the .fmm extension you should make sure it is a proper Fedem model file. Attempting to open a non-model file will usually result in an empty model, but unpredictable behavior may also occur, depending on the actual contents of the file.

In the file selection dialog, you can skip the loading of FE data for the links. See also [Section 2.9.2, "Skipping FE-Data when opening a model file"](#).

If the opened model file contains results, information about the result files in the model (. *frs* files) is reported in the "Output List" window. However, results files belonging to unloaded links (see [Section 2.9, "Loading and unloading FE-Data"](#)) and disabled result files (see [Section 8.2.2, "Result manipulation"](#)) are not included in this report.

If any problems are encountered during model loading, Fedem displays an alert message box and provides additional information in the "Output List" window.



NOTE: *There might be minor changes in the model file format from one Fedem version to subsequent versions of Fedem. The newer versions are always backward compatible such that you may safely open a model that was created in one particular version in any of the subsequent versions, without losing model consistency. The model is automatically converted to the new format while reading it.*



CAUTION: *The Fedem model files are not necessarily forward compatible. If you open a model in an older version of Fedem than it was created in, there might be changes in the model file format that makes the imported model incorrect or inconsistent. In some cases it may also make the Open operation fail or hang. Refer to the Fedem R5.0 Release Notes, Chapter 3, "Notes", for a summary of the forward compatibility issues that might need to be manually resolved in such cases.*

Loading links

When Fedem opens a model file and the loading of FE data is enabled, the link information (. *ftl* files, reduced matrix files, etc.) is read from the link repository, see [Section 4.1.5, "Using link repositories"](#).

If the link repository is missing, for example when a model file is moved, Fedem uses the following search path to locate the links:

1. The name and location of the originally imported FE link.
2. The name of the original FE link, located in a sub-directory of the current model file directory, with the same name as in the original FE file path (only if the original path was an absolute path).
3. The name of the original FE link, located in a parallel directory of the current model file directory, with the same name as in the original FE file path (only if the original path was an absolute path).
4. The name of the original FE link, located in the same directory as the model file.

5. The base name of the original FE link with the extension `.ftl` located in the same directory as the model file.

2.8.2 Saving models

You can save the model using its current name and location, or save a copy of the model using a different name and/or location.

To save the model, do one of the following:



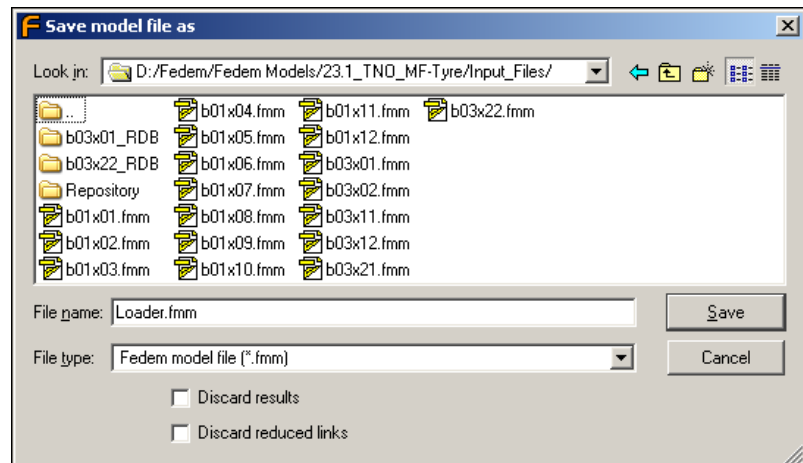
- To replace the current version on disk, choose **Save** in the File menu or click on the save icon in the toolbar.



NOTE: The previous version of the model file will be renamed to `<filename>.bak` before writing the new file, such that you can always go back the previously saved version by renaming that file, in case the last save operation failed due to full disk or other reasons.

- To save the file in a different location and/or with a different name, choose **Save As...** in the File menu.

When saving a copy of the file using **Save As...**, you can choose to discard results and/or reduced link information by setting the toggles in the **Save As...** dialog:



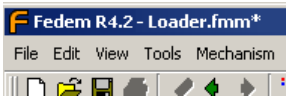
When saving a new model for the first time, you are prompted to give it a name different from the default name `untitled_<#>.fmm`, which was assigned when Fedem was started (see [Section 2.4, "Starting Fedem"](#)). If you also have performed some solver tasks before saving the model, the existing results database will then be moved to the correct location associated with the new model file name.



CAUTION: If you do not save a new model still named `untitled_<#>.fmm` before you open another model or **Exit** Fedem, all solver results associated with this model will be deleted, if any. This also includes the results of any Link reductions performed, unless link repositories were used (see [Section 4.1.5, "Using link repositories"](#)).

Indication on whether a Save is needed

When the current model has been changed compared to the previously saved version, Fedem will indicate this with an asterisk (*) after the model file name in the title bar of the main window. Only if this asterisk is present, you will be asked if you first want to save the current model when you **Open** another model, create a **New** model, or **Exit**.



2.8.3 Starting a new model



You can at any time start modelling on a completely new model in the current Fedem session by choosing **New** from the File menu, or by clicking the new file icon in the toolbar. This is equivalent to exiting the current Fedem session and then starting a new one (see [Section 2.4, "Starting Fedem"](#)) except that the new model now by default will be located in the same directory as the current model.



NOTE: If the current model has been modified after the last **Save**, you will be asked whether you want to save those changes before starting on the new model.

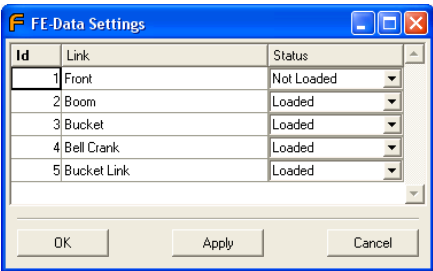
2.9 Loading and unloading FE-Data

When a model contains several large FE-models, unloading some of the FE-data from memory can be necessary in order to reduce the amount of resources used. This is particularly useful when you want to load contour plots for a particular link, or to free up resources for the solvers.

2.9.1 FE-Data Settings



Finite Element models tend to be large. The amount of data needed for visualization and lookup is indeed significant. In some cases it will be convenient or necessary to unload this data, to free up RAM. To control the loading and unloading of FE-data the *FE-Data Settings* dialog is used.



To open the dialog select the *FE-Data Settings* command in the Tools menu.

The status of each link is set using the drop down menu in the *Status* column. Set the status you want for each link, and press *OK* or *Apply*.

The links that are not loaded will be shown using the *Simplified* line shape, as described in [Section 2.7.5, "Item Appearance"](#).

The loading and unloading of links does not affect the simulation. The solver processes will read the necessary FE-Data from the FE-model files.



NOTE: Loading unloaded links can take several minutes if the links are big.

TIP: To change the status of all or several links quickly, it is convenient to use the arrow keys to navigate between the drop down menus, and the N-key (Not Loaded) or L-key (Loaded) to set the status.

2

2.9.2 Skipping FE-Data when opening a model file

When opening a big model for inspection or to start a calculation process, it is sometimes convenient to override the FE-Data Settings and skip the FE-models completely. In that way you can open a model containing extremely large links in a few seconds, without using any significant amount of memory.

To do this, toggle the *Skip FE-Data* toggle in the File Open dialog when opening a model file, or use the `-noFEData` command-line option together with `-f <modelfilename>` when starting Fedem.

The model will then be loaded with all the links unloaded. To load the links, open the FE-Data Settings dialog. This dialog now shows the status on the links last time you saved your model. If you simply press *OK* or *Apply*, those settings will be applied, and the links marked as *Loaded* will be loaded.

2.9.3 Modeling with unloaded links

An unloaded link can generally be used as any link while modeling, but there are some restrictions.

When selecting points on the link, the points will not snap to the closest node, because the node information is not loaded. The only points on the link that are available for selecting and modeling are the external nodes represented by the triads.

Triads can generally not be detached or deleted from the unloaded link. This is done to protect the reduced matrices from being accidentally invalidated. If you try to remove or detach a triad on an unloaded link you will get an error. When detaching joints from an unloaded link, a triad will be left on the link where the joint was attached.

2.9.4 Postprocessing unloaded links

Unloaded links will be completely skipped when loading an animation, except for the rigid body motion. This makes it possible to focus the computer resources on the parts of your model that are interesting, and skip everything else, see also "[Disabling and Enabling results](#)" in [Section 8.2.2](#).

2.10 Printing and exporting

2.10.1 Printing



The Print View command is available only for graph views. Clicking the **Print View** button prints the active graph view from the Workspace area regardless of the selection in the Model Manager panel.

2.10.2 Exporting

Six types of objects can be exported from Fedem: links (FE models), individual curves, graphs, graph views, the 3D modeler view and animations.

- Links can be exported in Fedem's *.ftl* format.
- Curves and graphs can be exported in ASCII (*.asc*, *.txt*), nCode DAC (*.dac*) or MTS RPC time history (*.rsp*, *.drv*, *.tim*) format.
- Graph views and the 3D modeler view can be exported as binary image files in a variety of formats.
- Animations can be exported as movies in mpeg-1, mpeg-2, and avi formats.

Exporting a link

To export a link, complete the following steps:

1. Right-click the link in the Model Manager Objects list to access the shortcut menu.
2. Select **Export Object...** to open the Save As dialog.
3. Specify a filename and location, then click **Save**.

Exporting curves and graphs

To export a curve, complete the following steps:

1. Select one or more curves in the Model Manager Results list, and right-click your mouse to access the shortcut menu.
2. Select **Export Curves...** to open the Save As dialog
3. Depending on your selection, you will either be prompted for a file name, or a directory. If you selected multiple curves, the curve files will be named automatically. Remember to select what **file type** you want to export to.

To export one or more graphs, complete the following steps:

1. Select one or more graphs or curves in the Model Manager Results list and right-click your mouse to access the shortcut menu.
2. Select **Export Graphs...** to open the Save As dialog.
3. If you have selected only one graph, you will be prompted to supply a file name for the exported graph. If you have selected several graphs, you will be prompted to select a directory to export your graphs to. In this case, the graph files will be given names automatically.

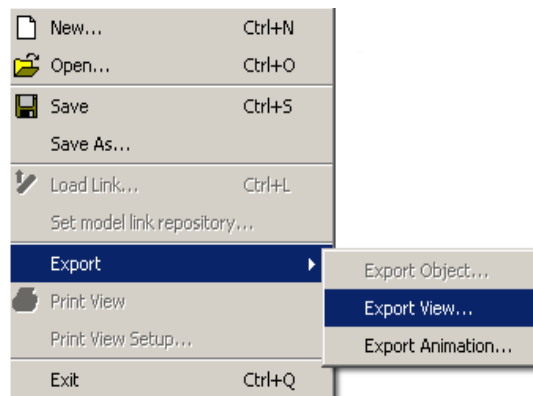
For more detailed information on how to export and import curves and graphs, see [Section 7.2.12, "Export of Curve Data"](#) and [Section 7.2.13, "Importing Curves and Graphs"](#), respectively.

Exporting the 3D modeler view

To export this view, make the window of the view active, and then select Export -> Export View... from the File menu.

When exporting the modeler view, you may choose between the following image formats:

- bmp
- jpeg
- png
- rgb
- 3D inventor snapshot, iv



The different file formats have different quality. *Jpeg* is a widely recognized format. The compression reduces the quality somewhat, but the files are small. *Png* and *bmp* have better quality. We recommend using the *png* format for high quality images.

The *iv* format enables dynamic 3D-viewing of your models using an external viewer. Viewers are available for several platforms.

Stills of animations, e.g. contour plots, can also be exported. Simply pause your animation where you want the picture or 3D-snapshot taken, then export the modeler view.

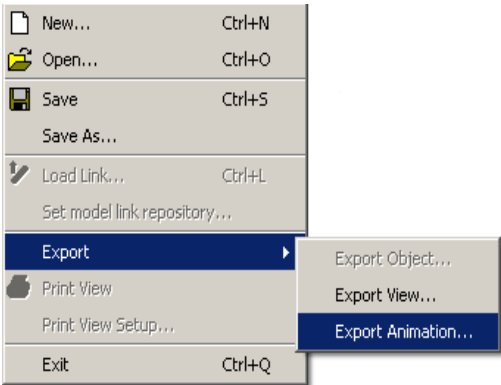
Exporting graph views

To export a graph view, follow the steps described in the section above. You may choose between these image formats:

- bmp
- jpeg
- png

Exporting animations

Animations can be exported using the mpeg-1, mpeg-2 and avi (Windows only) formats. The animation may be viewed in any standard video player (e.g. Windows Media Player or Eleccard MPEG Player (www.elecard.com) Note that the Windows Media Player has a size limit of 720x480 pixels).



After loading the animation, select Export->Export Animation... from the File menu, to open a file dialog where you can select location, file name and format of your exported animation. See also [Section 7.4.4, "Exporting animations"](#).

2.11 License information

The Fedem software consists of a foundation module and several add-on modules. The actually installed and available modules depend on your license contract with Fedem Technology.

2.11.1 Available modules

The currently installed licenses are listed to the console window when Fedem is started, if the command-line option *-licenseinfo* is specified. The list may also be printed in the *"Output List"* at any time during a Fedem session by selecting the *License Information* entry in the *Help* menu. The list indicates the licenses that are required upon startup, and the add-on licenses that are available to the user:

License information:

Module	Name	Version
FF-MDC	Modeler Core	4.00
Available, but not checked out:		
FA-CTI	Control interface add on	4.00
FA-CTR	Control add on	4.00
FA-DRB	Durability add on	4.00
FA-TII	Tire interface add on	4.00

```
-----  
License file/server:  
-----
```

```
27000@devpc7 (devpc7)  
-----
```

For example, when loading a model that has control elements, the control interface add-on license (FA-CTI) is also checked out. This license is held by the application until the model file is closed again.

The license information list is available at any time from the *License information* command in the *Help* pull-down menu.

2.11.2 License denial

If the requested add-on license is unavailable (either in use by another session, or not a part of the license contract), then modeling, manipulation, and solving of objects covered by that license is denied (e.g. control block editing is not allowed while the control interface add-on is missing).

Post-processing (including graph and animation handling) is always allowed as long as the modeler core license is available.

2.11.3 License file/server

The current license file/server used is also shown in the license information listing.

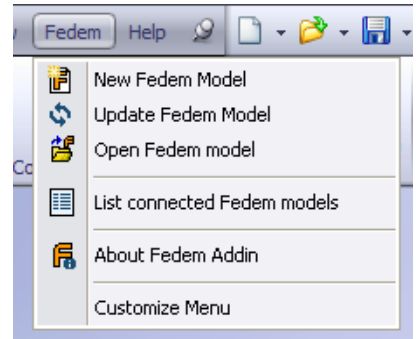
2.11.4 Managing license files/servers

By selecting the *License Manager* from the *Help* menu, you are able to add or remove license files/servers.

You may also paste a new license text directly into the default license field (See the Fedem R5.0 Installation Guide).

2.12 Using the CAD integration

When Fedem is installed on Windows, it registers an *add-in* with all the supported CAD systems. This add-in makes it possible for the CAD program and Fedem to communicate through the *Fedem* menu that is made available in the CAD system. This menu is shown in the picture to the right.



The supported CAD systems are all based on SolidWorks, and they include the CAD editor add-on to Fedem, named *FedemSolid*.

In this section you will learn how to make use of the integration from the *Fedem* menu in the CAD system. To learn how to use the CAD system itself, you will have to refer to its own documentation.

The description below is for FedemSolid, but it is valid for SolidWorks and NEiFusion as well.

2.12.1 Overview

FedemSolid and Fedem work together in order to easily create and update assembly-based simulation models in Fedem.

Mass properties, geometry, positions and the assembly structure is extracted from the FedemSolid assembly and used to produce a starting point for a simulation model in Fedem.

There are two sets of commands that control the CAD integration features. One set is in the CAD system, and the other one in Fedem itself.

In the CAD system, the *Fedem* menu provides the commands, while in Fedem itself there are two commands located in Mechanism Creation toolbar. (See [Section 2.5.2, "Menus and toolbars"](#))



The *Fedem* menu is where the different commands to create, open, update and keep track of the Fedem simulation models associated with a FedemSolid assembly model can be found. This menu is only visible when the active document is an assembly.

The following topics cover how the interoperation works.

- Model association: How CAD assembly configurations relate to the Fedem Analysis models
- How parts and subassemblies translate to Fedem links
- Process communication
- Commands in CAD system (the *Fedem* menu)
 - *New Fedem Model*
 - *Open Fedem Model*
 - *Update Fedem Model*
 - *List connected Fedem Models*
- Commands in Fedem
 - *Connect to CAD system*
 - *Open associated CAD assembly*

2.12.2 Model association

Each configuration can have an independent Fedem model associated with it. The Fedem commands (*New Fedem Model*, *Update Fedem Model* and *Open Fedem Model*) are working with respect to the current configuration and the Fedem model file, if any, which name is stored within the configuration.

The Fedem models are stored in a directory called `<assemblyname>.FEDEM` in the same directory as the FedemSolid assembly file (`.sldasm`). The names of the fedem models are `<assemblyname>_<configuration_name>.fmm`.

A list of all the associated Fedem models of an assembly can be shown using the command *List connected Fedem models*.

The model file names of the connected Fedem models are saved along with the FedemSolid assembly file, so you are advised to save the assembly model when you save a newly created Fedem model, to keep the association alive.

2.12.3 How parts and subassemblies translate to Fedem links

A Fedem Link is one rigidly moving body in the dynamics model. It is composed of all the parts and subassemblies that move as one rigid body.

When a Fedem model is created from an assembly, one Link is created for each of the top level parts and for each subassembly marked to be "Solved as Rigid".

(Right click a subassembly, and select the *Component Properties* command).

"Solved as Rigid" is the default setting for that property. If it is set to "Solved as Flexible" Fedem creates Links from the parts and subassemblies inside the subassembly in question after the same pattern; Top level parts and rigid subassemblies become Links, while the flexible subassemblies are recursively traversed.

2.12.4 Process communication

Fedem and FedemSolid are two executables that run side by side, and communicates using COM. When a Fedem command is executed, Fedem is started and connects to FedemSolid., unless it is already running and connected.

If Fedem is running but is not connected to the FedemSolid session, a question appears asking whether it is ok to start a new Fedem session.

If your intention was to use the Fedem session already running, you will have to answer *No* and then invoke the *Start or reconnect to CAD system* command in the already running instance of Fedem.

FedemSolid can also be started from within Fedem by using the same command.

2.12.5 Commands in CAD system (The *Fedem* menu)



New Fedem Model

This command creates a new Fedem model from the current assembly configuration.

If a Fedem model already exists for this configuration, you are asked whether you want to update it, or really create a new one.

This command also allows you to use an existing Fedem model as a starting point for the new model. This would typically be a Fedem model associated with one of the other configurations in your assembly.

If there exist Fedem models for some of the other configurations in the assembly, you are prompted with a list, and can chose to use one of those models as a starting point, or simply choose None to create a completely new Fedem model.

When no associated Fedem models can be found, you get the option of browsing for one.



Open Fedem Model

This command simply opens the associated Fedem model in Fedem. If no associated model exists, you are prompted whether you want to create a new one.



Update Fedem Model

This command opens the associated Fedem model, and updates it with the changes from the CAD assembly.

Geometry, mass properties and positions of all the links are then updated.

New Fedem Links will be created for new parts and subassemblies, while parts and subassemblies that have been removed will need some more attention:

Parts or subassemblies that have been moved from one position in the assembly hierarchy to another will not be recognized as "moved". It will rather appear as if they are deleted and that some completely new parts/subassemblies have appeared. This is somewhat inconvenient regarding all entities in Fedem that could have been associated (attached) to the links.

To make updating a bit more convenient in such cases, an Update Wizard is used to allow you to reassociate the Links in Fedem that no longer have a valid reference to parts in the CAD assembly. The Update Wizard pops up when Fedem discovers that some Links have an invalid reference to the CAD assembly. It steps through all the Links that have lost their referenced CAD part or subassembly and lets the user choose to either delete it, or merge all its connections onto a different link.



List associated Fedem Models

This command lists all the configurations in the assembly along with the file name of any associated Fedem model.

**About Fedem Add-in**

This command provides version information and information on how and where the Fedem executable is found.

2.12.6 Commands in Fedem**Connect to CAD system**

This command starts the available CAD system if it is not running. If it is, it connects the current Fedem session to the running instance of the CAD system.

**Open associated CAD assembly**

This command does the same as the *Connect to CAD system* command, but in addition it opens and activates the CAD assembly model that is associated with the current Fedem model.

Chapter 3 Mechanism Modeling

Now that you have been introduced to Fedem's user interface, you can begin the modeling process.

This chapter describes how to perform the various commands you need to build mechanism models, such as creating, moving, attaching, and detaching elements. It also describes how to apply motion constraints to the model.

Fedem's mechanism elements (links, triads, joints, sensors, and so on) and their properties are discussed in detail in [Chapter 4, "Mechanism Elements"](#).

Sections in this chapter address the following topics:

- [Basic assembling techniques](#)
- [Mechanism modeling environment](#)
- [Mechanism modeling tools](#)
- [Creating mechanism elements](#)
- [Moving mechanism elements](#)
- [Attaching and detaching elements](#)
- [Deleting mechanism elements](#)
- [Using file references in mechanism elements](#)
- [Model preferences](#)

3.1 Basic assembling techniques

There are three main approaches to assemble a Fedem model.

- *From a SolidWorks, FedemSolid or NEiFusion CAD assembly* - If you have a CAD assembly of your model and have one of the mentioned CAD systems available, you can directly transfer the complete assembly to Fedem. Then all the links will be in place, so what is left is to connect the parts by adding joints, etc.
- *With FE models or VRML geometry* - If you have VRML geometry or FE-models of your parts, assembling a Fedem model means to import the links, fit the parts together by moving and/or rotating the links as necessary, and then connecting them by creating and attaching joints.
- *With hardpoint positions* - When you only have hardpoint information, it is more convenient to place the joints in space at the hard points, and then connect the joints by creating generic parts from the triads in each joint.

Other mechanism elements such as springs, dampers, loads, and so on can be added in the same manner, either by placing triads on an FE-model, or placing them in space, and attach them to FE-model-less links.

When assembling a model in Fedem, triads are the system model counterpart of the FE-nodes and represent the connection between the system model and the links. See [Section 4.3, "Triads"](#) and [Section 3.6, "Attaching and detaching elements"](#) for a description of how connections are made using triads.

As you assemble the model and move things around, Fedem tries to help you by setting the *movability* of objects to match the constraining of your model. This means that you will be unable to move an object (such as a ball joint) that is fixed in some way. If an object is constrained from translating, you will be able to rotate but not translate it, and so on.

3.2 Mechanism modeling environment

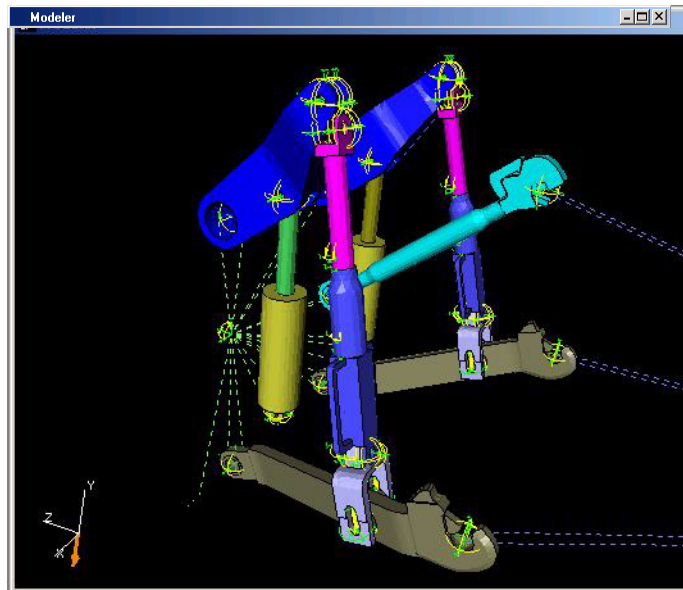
Fedem's modeling environment combines a powerful, 3D graphic interface and dynamic viewing capabilities with quick and easy management tools. The Model Manager tabs provide shortcuts for creating, selecting, and deleting elements.

3.2.1 Modeler window

To build a mechanism model, Fedem provides the Modeler window; a three-dimensional, graphical environment in which your model can be viewed and edited. The mechanism elements are selected from menus and toolbars for placement in the Modeler window. They can then be moved and manipulated using Fedem's modeling tools. This editing environment also features dynamic viewing tools, defined-view commands, and appearance settings (see [Section 2.7, "Visualizing the model"](#)).



To open the Modeler window, click the **Show Modeler** button on the Windows menu or toolbar. The Modeler window is shown below with an example mechanism assembly.



3.2.2 Modeling toolbars

In Fedem, there are three major tasks performed by the user: 1) creating the mechanism and control system; 2) setting up and starting the analysis; and 3) setting up and viewing the results. Each task has a different set of associated commands. The mechanism modeling tools used to create and edit models are covered by the *Mechanism Creation* toolbar and the *Mechanism Tools* toolbar.

Mechanism Creation toolbar

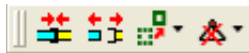
The Mechanism Creation toolbar (shown below) contains the mechanical elements used to build Fedem mechanisms (see [Section 3.4, "Creating mechanism elements"](#) for instructions on using these commands, and [Chapter 4, "Mechanism Elements"](#) for a detailed description of each element).



NOTE: An arrow (▼) beside a button indicates that more options can be accessed by clicking and holding down the button.

Mechanism Tools toolbar

The Mechanism Tools toolbar (shown below) consists of modeling tools. (Each of these commands is described in the following sections.)



3.3 Mechanism modeling tools

To help you position items with greater accuracy and to simplify the modeling process, Fedem provides some helpful modeling tools, including a reference plane, an interactive point locator, point markers, and *movability* constraints.

3.3.1 Reference Plane

The Reference Plane is the shaded area in the center of the Modeler window. It serves both as a visual reference, and as a representation of the ground.

You can move it around, change its color and size, or turn it off so that it is not visible in the Modeler window.

To disable, enable or change the appearance of the reference plane, see [Section 2.7.4, "General Appearance"](#) and [Section 2.7.5, "Item Appearance"](#).

Changing the size

To change the size of the Reference Plane, select the Reference Plane in the Modeler window (or Model Manager Objects list), then edit its *Height*

and *Width* fields in the Property Editor panel (shown below). Remember to press *Enter* after typing the values to apply the changes.

Position

Reference CS

Global

Coordinate type

Cartesian XYZ

X

0.0

Y

0.0

Z

0.0

Orientation

Reference CS

Global

Coordinate type

EulerZYX as X, Y, Z

RotX [Deg]

-0.0

RotY [Deg]

0.0

RotZ [Deg]

-0.0

Size

Height

1.0

Width

1.0

Moving

The Reference plane is moved by editing the *Position* and *Orientation* fields in the Property Editor panel. See [Section 3.5.4, "Origin property"](#) for a detailed description of these data fields.

It is also possible to move the reference plane by aligning it to a specified coordinate system in your model. To do so, use the *Align CS* or *Align rotation* commands. See [Section 3.5.2, "Align CS and rotations"](#).

3.3.2 Interactive Odometer and 3D Point Marker

Many Fedem commands require you to select a point in your model. To help you locate specific points, Fedem provides the Interactive Odometer and the 3D Point Marker (shown at right). These are displayed in the Modeler window each time you select a point. The odometer shows the coordinates of the selected point, and the marker shows the location of the point in the Modeler window.

Reference

Global

X

0.08612320

Y

0.07296870

Z

0.0



When using the Smart Move command to move or rotate links and other mechanism elements (see [Section 3.5, "Moving mechanism elements"](#)), the Interactive Odometer allows you to edit the selected point or enter a new 3D point using global or local coordinates. The local coordinate system used is the coordinate system of the item you selected when the point was picked.



TIP: You can use the Interactive Odometer with the Smart Move command to place a mechanism element (link, joint, triad, and so on) at a point in free space. The object can then be used as a reference when moving other objects.

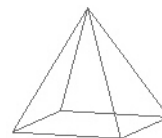
To edit a point or enter a new point using the Interactive Odometer, complete the following steps:

1. Select a point in the Modeler window. The coordinates of the point (given in the local coordinate system for the selected element) are displayed in the Interactive Odometer, and the 3D Point Marker shows the location of the point selected.
2. Select *Local* or *Global* coordinates from the Reference pull-down menu.
3. Type new values for the *X*, *Y*, and *Z* coordinates in the Interactive Odometer, and press the *Enter* key after editing the values. The 3D Point Marker is updated to show the new position.
4. When you are satisfied with the new location, press *Done* to confirm the selected point.

3.3.3 Stickers

Stickers are *movability* constraints that are applied automatically when moving mechanism objects with the Smart Move command (see [Section 3.5, "Moving mechanism elements"](#)). Stickers can also be applied manually (see "[Manually applying stickers](#)" below). Each sticker applies the same constraint as a ball joint; in other words, it constrains all translational motion.

Stickers are displayed in the Modeler as small pyramids (shown at right). The tip of the pyramid is located at the constrained point.



When using Smart Move, the motion allowed, or *movability*, for a selected object or group depends on the number and location of applied stickers. Each move using Smart Move automatically applies an additional sticker. Therefore, three successive moves of an initially free object (without stickers) first results in a translation, then in rotation about a point, and lastly rotation about the remaining axis. The object is then locked in place. (See also "[Movability](#)" in [Section 3.5.](#))



IMPORTANT! *Stickers function as modeling aids only. They are not considered part of the mechanism model, and therefore do not influence the mechanism motion during simulation.*

Manually applying stickers

Stickers are created automatically when using Smart Move, but you can also create them manually when you need a certain type of movability. To rotate about a point in space, apply one sticker at the rotation center. To rotate about an axis, apply two stickers somewhere along the rotation

axis. Stickers are applied to Triads and Links, and can be positioned at any point in space. They are *not* restricted to the geometry of visualization or the nodal points of a link. To create a sticker, perform the following steps:



1. Click the *Sticker* button on the Mechanism Tools toolbar. The Guide bar prompts you to select an application point for the sticker on an object.
2. Place the cursor over the point on the object you want and press the left mouse button. The selection snaps to the nearest node or point on the object.
3. If necessary, edit the position using the Interactive Odometer as described in [Section 3.3.2, "Interactive Odometer and 3D Point Marker"](#).
4. Confirm the point by clicking *Done*. The sticker is created, and the sticker symbol appears in the Modeler window at the selected point.

Deleting stickers

You can delete stickers individually or all in a single operation.

- To delete a single sticker, complete the following steps:



1. Select the sticker you want to delete in the Modeler window or from the Model Manager Objects list. The sticker symbol in the Modeler turns red when selected.
2. Click the **Delete** button on the Standard toolbar or use the Delete-key. The sticker is removed from the model.



- To delete all the stickers applied to your model, click the **Delete All Stickers** button on the Mechanism Tools toolbar or the Mechanism menu.



NOTE: You may have to click and hold down the **Sticker** button on the Mechanism Tools toolbar to access the Delete All Stickers command.



WARNING! There is no undo option after deleting all stickers. To replace them in your model, you must recreate each of them individually.

3.4 Creating mechanism elements

Objects such as Spring/Damper characteristics, Functions and Frictions, which do not need to be positioned are created in the Model Manager Objects list. Right-click an empty space in the Model Manager panel and select *Create* to access the full list of elements that can be created using the shortcut menu.

All mechanism elements that need to be positioned in the Fedem model are created by a different method. Normally, you first need to select the position(s) needed to place the new element. Then you sometimes need to orient the element properly, and then you finally have to attach it. See [Section 3.5, "Moving mechanism elements"](#) and [Section 3.6, "Attaching and detaching elements"](#).

The only exception to this is the links which are treated completely different. Please have a look at [Section 2.12, "Using the CAD integration"](#), [Section 4.1.1, "Creating links by file import"](#) and [Section 4.1.2, "Creating links from hard points"](#).

To create a mechanism element, complete the following steps:

1. Click the button for the item on the *Mechanism Creation* toolbar (see ["Mechanism Creation toolbar"](#) in [Section 3.2](#)).
2. Follow the instructions in the *Guide panel* while selecting one or more positions or related objects in the Modeler window. Positions can also be entered by using the Interactive Odometer. (See [Section 3.3.2, "Interactive Odometer and 3D Point Marker"](#).)
3. When you are satisfied with a selection, click *Done* to confirm. When all positions/selections are completed, The element is created using the selected position(s). A sticker is normally applied to the new object automatically to make it easy to rotate using Smart Move.
4. You can then use *Smart Move* or some other means to adjust the object's orientation (see [Section 3.3.3, "Stickers"](#) and [Section 3.5, "Moving mechanism elements"](#)).
5. As the last operation you need to attach the object to a Link using the Attach command. See [Section 3.6, "Attaching and detaching elements"](#).



TIP: To edit the properties of a new mechanism element, select the item in the Modeler or Model Manager Objects list. The properties of the item are then displayed in the Property Editor panel.

3.4.1 Selecting position and orientation

When creating mechanism elements, you are asked to select their position. Revolute joints, Free joints, Loads and Cam joint master triads will be created with a default **orientation** as well.

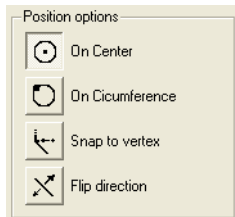
As you select a point, Fedem will snap to geometric features and also extract a default orientation if necessary, by different rules depending on what type of part/object you hit.

Snapping and default orientation on FE Parts

When picking an FE Element surface the default orientation is set to be perpendicular to that surface and the position snaps to the closest node. If the exact position of the mouse is on an FE mesh line, however, the default orientation is aligned with the direction of that line. It is the exact mouse position that decides whether you hit a line or a surface, even when the 3D Point Marker snaps to the same node.

Snapping and default orientation on CAD Parts

If the mouse position is on a CAD part, the geometry of the face or edge is used to extract a default orientation and a snap point. If the face or edge in question is a revolved geometry, the options shown to the left pop up in the *Guide panel*. These options allow you to control how the snap point and the default orientation is extracted from the geometry.



First of all, you can choose whether the point shall snap to the center axis of the revolved face (*On Center*), with the default orientation along the axis, or to the circumference of the revolved geometry (*On Circumference*), with the default orientation perpendicular to the face.

The default orientation can be flipped to the opposite direction using the *Flip Direction* option.

The *Snap To Vertex* option controls whether or not the selected point shall snap to the closest vertex. If you have chosen *On Circumference*, the point will snap to the closest vertex on the surface. If *On Center* is chosen, the point will snap to the point on the center axis which is closest to the vertex.

Default direction notes

The default direction is visualized as a yellow arrow starting from the selected point. The different object types utilizing the default direction and how they use it are listed here:

- *Force and Torque* - the attack direction.
- *Revolute joint* - the axis of rotation.
- *Cam joint master triads* - the x-direction (up) of the masters are aligned with the default direction.
- *Free joint* - Z-Axis of the master.



3.5 Moving mechanism elements

Fedem provides several commands and tools for moving parts of your model: The *Smart Move*, *Align CS*, *Align rotations* and *Move To Center* are useful commands while the *Origin property*, that several mechanism objects share, can be used to access the position and orientation of a single component directly.

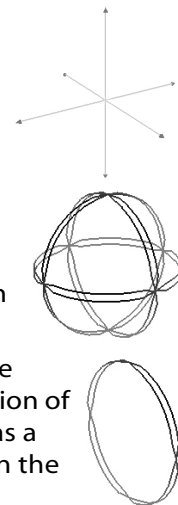
3.5.1 Smart Move

Smart Move is an integral part of 3D modeling in Fedem. It is a sophisticated way to position and orient mechanism entities such as links, triads, joints, springs, dampers, and so on. The command enables you to move or orient an object or a group of objects according to the selection's current *movability* (see below). During a move, *Smart Move* also automatically adds a movability constraint called a *sticker* (see [Section 3.3.3, "Stickers"](#)).

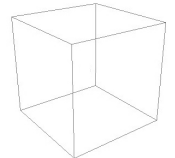
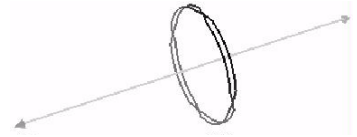
Movability

The movability of an object or a group of objects is determined by examining not only the stickers that are applied to the selection, but also the joints between the selected object/group and other objects. Each joint or sticker that constrains the group reduces its movability. The following symbols represent the six types of motion allowed when using the *Smart Move* command:

- *Free* – when an object without stickers or attached joints is moved with the Smart Move command, it can move in translation in any direction. The symbol for free movement is depicted in the Modeler as shown to the right.
- *Ball* – when one sticker or one ball joint has been applied to a mechanism entity, it can rotate about the point at which the sticker/joint is applied. The symbol for ball movement is depicted in the Modeler as shown to the right.
- *Revolving* – when two stickers (or one ball joint and one sticker, or two ball joints) have been applied to a selection of mechanism elements, the stickers/joints act together as a revolute joint with the axis defined by the line between the two stickers/joints. The symbol for revolving motion is depicted in the Modeler as shown to the right.



- *Cylindric* – when a selection of mechanism elements is constrained by one cylindrical joint, the selection can be translated along and rotated about the joint axis. The symbol for cylindrical motion is depicted in the Modeler as shown to the right.
- *Prismatic* – when a selection of mechanism elements are constrained by one prismatic joint, the selection can be translated along the joint axis. The symbol for prismatic motion is depicted in the Modeler as shown to the right.
- *Rigid* – when three stickers or ball joints (not located on a straight line) are applied to a selection of mechanism elements, the selection is fully constrained and cannot be moved with the Smart Move command. The symbol for rigidity (no movement allowed) is depicted in the Modeler as shown to the right.



3

Performing a *Smart Move*

To move an object or group of objects using the *Smart Move* command, complete the following steps:



1. Click the *Smart Move* button from the Mechanism Tools toolbar. The Guide bar prompts you to select objects to move.
2. To select an object and indicate the from-point, place the cursor over a point on the object and press the left mouse button. The selection snaps to the nearest node or point on the object. This point becomes the from-point and a symbol is shown that depicts the movability of the current selection.



TIP: Several objects can be selected by pressing and holding the *Ctrl* key while selecting objects. To change the last selected object only, release the *Ctrl* key and select until you hit the right object. To remove several of the last selected objects from the selection, release the *Ctrl* key, and press the left mouse button on some empty space in the modeler until all the objects in question is deselected.



TIP: You can also type in a discrete point or edit the point using the *Interactive Odometer* (see [Section 3.3.2, "Interactive Odometer and 3D Point Marker"](#)).

3. When you are satisfied with the object and the from-point selected, press *Done* to confirm it. The Guide bar then prompts for you to select a to-point.

4. Select the to-point in the same way you selected the from-point, and if necessary, edit it using the Interactive Odometer to specify a discrete point.
5. When you are satisfied with the to-point, press *Done* to confirm it. The move operation is animated in the Modeler window.

3.5.2 Align CS and rotations



The two align commands can be used to align one or several objects to an existing coordinate system in your model. The *Align CS* command will both translate and rotate the selected objects to match their coordinate systems with the selected one, while the *Align rotations* command only rotates the selected objects.

Performing an Align command

To move an object or group of objects using one of the align commands, complete the following steps:

1. Click the correct Align button in the *Mechanism Tools* toolbar. They are located under the *Smart Move* icon.
2. Select the objects to move by picking them in the Modeler view. Press *Done* to confirm the selection.



TIP: Several objects can be selected by pressing and holding the *Ctrl* key while selecting objects. To change the last selected object only, release the *Ctrl* key and select until you hit the right object. To remove several of the last selected objects from the selection, release the *Ctrl* key, and press the left mouse button on some empty space in the modeler until all the objects in question is deselected.

3. Select the coordinate system to align to by picking an object that is defined in that coordinate system. The selected coordinate system will be displayed in red. Press *Done* to confirm the selection and execute the move.



TIP: The *Align* commands can be used to align objects to local FE coordinate systems, if present. The visibility of local coordinate systems can be toggled using the *General Appearance* dialog. See [Section 2.7.4, "General Appearance"](#).

3.5.3 Move To Center



Move To Center is a useful tool if you want to position an object at the center of some geometry. The object will move to the center of a circle you define, or somewhere along its axis. The new orientation of the object aligns with the circle. The x-axis is defined by the center and the first point defining the circle, the z-axis is perpendicular to the circle.

Performing a Move To Center

To move an object or group of objects using the *Move To Center* command, complete the following steps:

1. Click the Move To Center button in the *Mechanism Tools* toolbar. It is located under the *Smart Move* icon.
2. Select the objects to move by picking them in the model view. Press *Done* to confirm the selection.

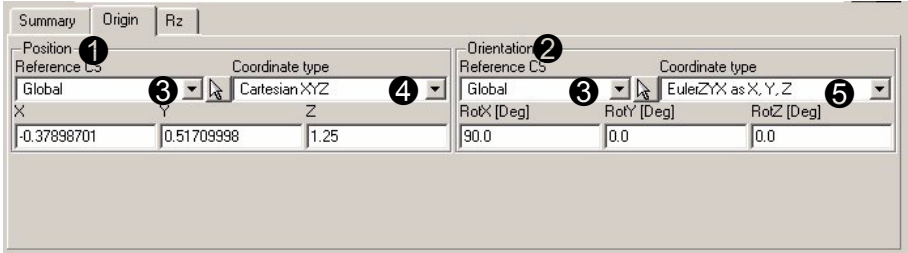
TIP: Several objects can be selected by pressing and holding the Ctrl key while selecting objects. To change the last selected object only, release the Ctrl key and select until you hit the right object. To remove several of the last selected objects from the selection, release the Ctrl key, and press the left mouse button on some empty space in the modeler until all the objects in question is deselected.
3. Select three points of the perimeter defining a circle. Confirm each point by pressing *Done*. After setting the third point the defined circle will appear in the Modeler view.
4. You may now select a point to place your objects along the circles axis, or press *Done* once more to move the objects to the center of the circle.



3

3.5.4 Origin property

Triads, Links, and point-to-point joints have a property tab called *Origin*. (Shown below). This property tab is used to display and edit the position and orientation of the mechanism element in question. The sensitivity of the fields will reflect whether the selected object is allowed to move considering its attachments etc., without corrupting the model.



- ① *Position* - This frame displays the data for the translational part of the position.
- ② *Orientation* - This frame displays the data for the rotational part of the position.

- ③ *Reference CS* - The translation and rotation can both be displayed and edited in any coordinate system in the model. These pull-down menus allows the reference CS's to be selected. The reference CS's can also be selected by picking in the *Modeler* or selected from the *Objects Browser*. To do so, you must first click the arrow button next to the pull-down menu. The guide bar tells you to select a reference CS. Select a triad or a link, when satisfied, press *Done*.
- ④ *Coordinate type* - This menu controls whether to display the translation in cartesian coordinates or cylindrical coordinates. The cylindrical coordinates can use either X, Y or Z as the rotational axis.
- ⑤ *Coordinate type* - This menu controls how to edit and display the orientation.

The Orientations input type options are:

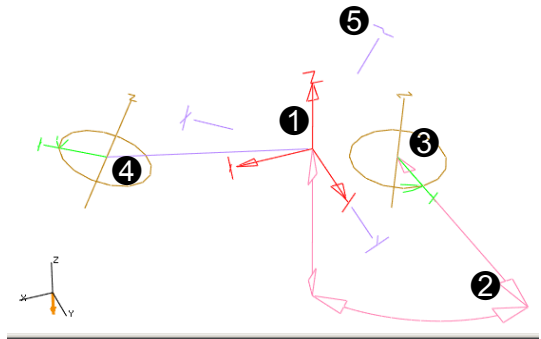
- Angles about the X-, Y- and Z-axis in degrees. The parameterization used is the one called Euler-ZYX, which means a rotation about the Z-axis of the reference CS first, then the Y-axis of the reference CS, and finally the X- axis of the reference CS. (This can also be understood as a rotation about the axes of the with-rotated (or local) X-axis first, then local Y and finally local Z.)
- A point on the X-axis, and a point in the XY plane. The X and Y direction is then computed from the given point and the translation of the object.
- A point on the Z-axis, and a point in the XZ plane.
- A vector in the X direction and a vector in the XY plane. In this mode, the X and the Y vectors from a 3x3 rotation matrix can be used directly.



NOTE: *The center of the applied rotations are always the origin of the Coordinate system in question (The position controlled by the Position options) and not at the origin of the selected reference CS.*

Visualization

The sizes displayed in the *Position* frame are visualized, along with the reference CS for the orientation, whenever the *Origin Property* is visible. The visual appearance of this visualization is shown in the picture.



- ❶ *The object to move (red)* - In this example; a Triad.
- ❷ *Position arrows (pink)* - Shows the various dimensions corresponding to the selected Coordinate Type as arrows extending from the reference CS for the position.
- ❸ *Reference CS for the position.*
- ❹ *Reference CS for the Orientation* - The reference CS for the Orientation is shown by the purple line from the object to move to the reference CS.
- ❺ *Orientation Reference direction* - The reference directions for the orientation are shown as purple lines and letters indicating the orientation of the reference superimposed on the position of the object to move.

3

3.6 Attaching and detaching elements

Nearly all mechanism elements created in Fedem needs to be attached to a link or two. The concept of attaching is to connect the joint constraints, loads etc. to the parts they affect.

When attaching, two things happen: Firstly a triad in the element in question is noted to be connected to the link (see also [Section 4.3, "Triads"](#)). All the constraints loads etc. that the element introduces will then be working on that particular link.

Secondly, if the part is an FE model, the triad is connected to the FE mesh of the part, either by being directly associated with an existing FE node in the part, or by using a *Surface connector* to distribute the forces in some way.

When an element is attached, it can generally not be moved relatively to the object it is attached to. The *Detach* command is used to disconnect a mechanism element from the link it is attached to, making it possible to move it around.

3.6.1 Attaching using *Attach*

The *Attach* command can be used when an element is to be connected to ground, to a *Generic Part*, or to an *FE part* with an existing FE-node at the attach point.

To attach an item to a link or to the ground, complete the following steps:



1. Click the *Attach* button on the Mechanism Tools toolbar (or select from the Mechanism menu). The Guide panel prompts you to select a mechanism element to attach to the model.
2. Select the element in the Modeler window.
3. When you have made your selection, press *Done* to confirm it. The Guide panel then prompts you to select a link onto which to attach the object.
4. Select a link or the reference plane in the Modeler window and press *Done* to confirm the selection. The object becomes attached to the selected link or to the ground if the reference plane was selected.



TIP: Watch the Output List for error messages during the attachment process. If Fedem cannot complete an attachment, an error message is displayed in the Output List.



NOTE: Triads in axial springs, dampers, and loads are automatically attached when created, because the orientation of such triads is not important.

Attaching joints

Joints consists of one slave triad and one or more master triads. The slave triad is normally attached to one FE-model, and the master triad(s) to another. This is done by attaching one part of the joint first, and then the other one. (See [Section 4.4, "Joints"](#) for more information about master and slave triads in joints) When attaching by selecting the joints directly, Fedem automatically selects which triad (master or slave) to attach first. In most cases, the slave is attached first.

To control whether a joint's slave or master is attached to a specific node, select only the part of the joint symbol that represents the particular (master/slave) triad when selecting the object to attach during the attach

command. See [Section 4.4, "Joints"](#) for more information about joint symbols.



TIP: To attach multiple joints to a single FE node, you must align the master triads of the joints. You can then attach the master triads of each joint to the FE node. The two master triads will then be merged into one triad shared by the two joints.

3.6.2 Surface Connectors



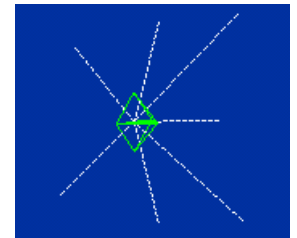
The *Surface connector* commands are used to attach mechanism elements to FE parts at positions without existing FE-nodes, (E.g. center of holes etc.) or in such a way that the Surface connector distributes the forces from the joint, load etc. onto some area on the FE model.

Surface connectors connects a triad to an FE model using two different connection types, *Rigid Surface* or *Flexible Surface*.



Flexible surface

The flexible surface connector acts as a force distributor. It does not introduce stiffness or constraints between the FE nodes it connects to, but distributes the forces from the triad onto the FE model.



Each nodal DOF gets an equal share of the translational forces it the triad. The moments acting on the Triad, and the moment created by the translational forces about the geometrical center of the nodes, is balanced by an additional force in each node, weighted by the nodes distance from the geometrical center of the nodes in the Surface connector. In the case where the nodes also have rotational degrees of freedom (shell nodes) those rotational degrees of freedom also gets an equal share of the moment.

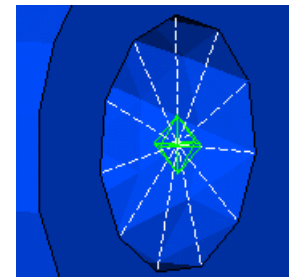
The Flexible surface connector is visualized with dotted lines.



Rigid surface

The rigid surface connector is a rigid connection between the triad and all the FE-nodes it connects. That means that all the FE-nodes connected with the surface connector becomes one rigid block.

The rigid surface connector is visualized with dashed lines.



3.6.3 Surface connector commands

There are two ways of creating a Surface connector: *By selecting nodes* or *By cylinder surface*.

The commands can either create a new connected triad at a user defined position, or connect an existing triad to the FE mesh, in somewhat the same manner as the *Attach* command.

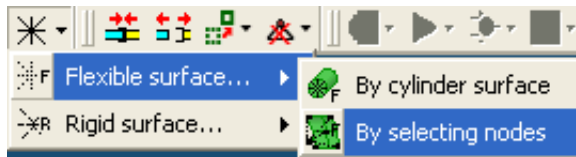


By selecting nodes

These commands are used to select arbitrary nodes or areas of nodes to use for the surface connector.

To create a Surface connector by selecting nodes, complete the following steps:

1. Select either the rigid or flexible version of the command.



2. First you have the option to either select an existing triad, which might be embedded in a mechanism element, or to select a position where you want a new triad to be created.

If you pick a triad, that triad will become selected and will be attached through the surface connector. If you pick something else, the snap-point will be used as the position for a new triad.

If you get it wrong, try again until you have selected what you want, then accept by pressing *Done*.

3. Now you need to select all the FE nodes to be connected to the triad. You can add single nodes to the selection by picking, or all visible surface nodes within a rectangle by dragging a window. If you press and hold the Ctrl-key, picked or window selected nodes will be removed from the selection instead of being added.

When finished, press *Done*.



By cylinder surface

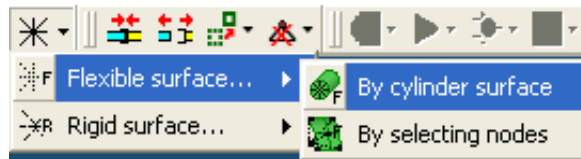
These two commands create a Surface Connector from nodes on the surface of a cylinder volume. It is convenient to use them to attach a

mechanism element to the inside of a hole, a circular edge etc. The command is also able to place a new triad along the axis of the cylinder, making it easy to get the hardpoints. This triad can then be used for further modeling.

The cylinder is defined by a 3 point circle, together with points on each end of the cylinder.

To create a Surface connector by cylinder surface, complete the following steps:

1. Select either the rigid or flexible version of the command.



2. First you have the option to select either an existing triad that will be attached by the Surface connector, or a position where a new triad will be created. If you rather want the command to create a triad along the cylinder axis, start selecting the tree nodes that define the circle.

If you pick a triad, that triad will become selected and will be attached through the surface connector. If you pick an FE-node, the node will be used as the first of three nodes that defines the cylinder circle. If you pick something else, the snap-point will be used as the position for a new triad.

If you get it wrong, try again until you have selected what you want, then accept by pressing *Done*.

3. Then select the rest of the tree points defining the cylinder circle, accepting each node by pressing *Done*. A cylinder/circle visualization will show the resulting cylinder as you select the last node.
4. When the cylinder circle is defined, you can now or after any of the following steps press *Done* to complete the command using the definition of the cylinder that is shown. The optionally new triad will then be created in the center of the circle, if you did not define a position for it in the start of the command.
5. The next steps is to select the start and end of the cylinder. Do this by selecting an FE-node for the start, and one for the end. Accept each of them by pressing *Done*.

6. Finally the position of the optionally new triad along the cylinder axis can be selected. Press *Done* to accept.

When the command is completed, Fedem selects all the nodes from the FE-model in question that is on the surface of the cylinder volume defined. These nodes is now connected to the new or existing triad by the Surface connector.

Deleting or redefining Surface connectors

A surface connector is actually an attribute of the Triad it connects. This means that if the triad is detached, the connector is deleted.

If a connector needs to be edited or changed, simply use one of the Surface Connector commands to redefine it. Invoke the command, and select the triad with the misdefined connector at the start of the command sequence.

3.6.4 Attachment rules and restrictions

There are several restrictions and rules that apply to the connections between FE models and triads. These restrictions do not apply when the link is a *Generic Part*.

- Triads can not be attached to 3-DOF FE nodes. To attach a triad to a solid model, you need to add an element and create a 6-DOF node at the correct position.



TIP: Normally a weighted average motion element, (Nastran RBE3) or a rigid element (Nastran RBE2) will be a good way to create a 6 DOF node at a hard point position on a solid mesh.

- The triad and the FE node must be within the distance set by the modeling tolerance (see [Section 3.9.2, "Modeling tolerance"](#)). If several 6 DOF nodes exist within the modeling tolerance, the one closest to the triad will be selected.
- Slave triads can not be attached to ground. Joints must be attached to the ground by their master triads.
- Two or more slave triads can not be attached to the same FE node.
- Master triads or triads where the triad directions is referenced must be aligned before they can be attached to the same FE node.



NOTE: When several elements are attached to the same node, the triads in those elements are merged into the triad already attached. This resulting triad is then shared by all the attached elements. (The Redundant triads are removed.)



NOTE: When a triad is attached to a FE model at the position of a slave FE-node, Fedem will automatically add a 6-DOF node, a spring and a mass element to the FE model at that point. The triad is then attached to the new 6-DOF node instead of the slave node to overcome limitations in the mathematical methods used. The spring stiffness and mass is set automatically by the Link Reducer to a very high stiffness, and a very low mass compared to the actual model in question. However, if the link is completely rigid (e.g., it consists of a single RGD element), the value $2e11$ is used for stiffness and no mass is added. See also the Fedem R5.0 Theory Guide, Sections A.10 "BUSH" and A.12 "CMASS".

3.6.5 Detaching

To detach mechanism elements from your model, complete the following steps:



1. Click the *Detach* button on the Mechanism Tools toolbar (or select from the Mechanism menu).
2. Select the item(s) to detach (hold down the *Ctrl* key while selecting multiple items in the Modeler window).
3. Click *Done* to confirm your selection. The object(s) are detached from your model.



NOTE: When detaching a joint, both the master and slave triads are detached. If you want to detach only one of them, press the **Detach** button, then select the part of the joint symbol that represents either the master or the slave triad and press **Done**.

3.6.6 Color of attached and unattached elements

The color of a mechanism element indicates whether or not it is attached to the model. If the element's symbol appears white (the default color), it is not completely attached. A colored symbol indicates that an element is attached.



TIP: The colors for attached and unattached elements can be changed in the General Appearance window (see [Section 2.7.4, "General Appearance"](#)).

3.6.7 Invalid attachments

At some points Fedem may find that some Triads in your model does not correspond to an underlying FE-node even if it should have. Fedem will then warn you with a dialog box, and mark all the invalidly attached triads with a red exclamation mark in the Model Manager Objects list.



The model will not be solvable as long as you have invalidly attached triads. To resolve this you have several options:

- Use a surface connector to connect the triad to a set of existing FE nodes.
- You can move the triads to the correct positions.
- Update the FE-mesh with nodes at the correct positions.
- Turn the FE-link into a Generic Part.
- Delete the Triads.

3.7 Deleting mechanism elements

Fedem uses the Delete command to remove mechanism elements from a model. The Delete command can be used in two different ways; in the Modeler window or in the Model Manager panel.

3.7.1 Deleting in the Modeler

To delete elements in the Modeler window, complete the following steps:

1. In the Modeler window, select the element to be deleted, or hold down the *Ctrl* key and select multiple items. The selected items are highlighted in red in the Modeler window.



2. To remove the selected element(s) from the model, click the *Delete* button on the Standard toolbar or hit the *Delete*-key.



WARNING! *There is no undo option after deleting objects. To replace mechanism elements after deleting them, you must recreate each of them individually.*

3.7.2 Deleting in the Model Manager

To delete mechanism elements in the Model Manager panel, complete the following steps:

1. In the Objects list, select the item to be deleted, or hold down the *Shift* or *Ctrl* key and select multiple items. The selected items are highlighted in red in the Modeler window.



TIP: *You can also click and drag the cursor to select multiple items for deletion.*

2. To remove the element(s) from the model, right-click and select *Delete* from the shortcut menu (or hit the *Delete*-key). The items are removed from both the model and the list at the same time.



NOTE: When deleting Links, you have the option to retain triads that are attached to each deleted link (except for joint triads - they are always retained), to also delete those triads, or to cancel the Delete command. If no such triads exist for a selected link, you must anyway confirm the deletion of the link. This choice must be made for each link in the selection. However, by selecting the Yes to all, No to all or Ok to all button, you automatically repeat the same choice for each link in the current selection.



NOTE: If any of the objects you delete are referred to by a curve, you will be able to choose to either delete that specific curve definition, leave the curve definition intact while still deleting the object, or cancel deletion of the selected object.



WARNING! If the deletion of a selected object also causes deletion of other objects connected to it, and any of these other objects also are referred to by a curve, you will be notified and can choose whether to delete that specific curve definition or not, too. However, you can not choose to cancel the Delete operation at this stage.

3

3.8 Using file references in mechanism elements



Some mechanism elements in Fedem need input from external files. For such elements the use of file references may be beneficial. The file reference replaces the file name in the element definition. As the contents of the file reference, the file it is referring to, is changed, so is the element input. Thus, if several mechanism elements receive their input from the same file reference, and the contents of the file reference changes, so does the input of *all* elements using it.

File references are created either by choosing *File reference* from the *Mechanism* menu, or by right-clicking an empty space in the Model Managers Objects list, choosing *Create* and then *File reference*. Then select one or more files in the *File Open* dialog that appears. One file reference will be created for each selected.

File references are set as input in mechanism elements by choosing the wanted reference from the pull-down menu of the input file field.



TIP: If all tires in a model are created from the same file reference their tire definitions may all be switched simultaneously.



TIP: If the several polyline from file functions use the same RPC file reference, the input for all functions may be changed quickly.

3.9 Model preferences

In the Model Preference dialog you can adjust several parameters regarding the model.

- ❶ *Model description* - This field lets you add notes to your model file.
- ❷ *Units* - The consistent unit set to use. See [Section 3.9.1, "Model database units"](#).
- ❸ *Modeling tolerance* - The maximum allowed distance between the FE-node and its attached Triad. See [Section 3.9.2, "Modeling tolerance"](#).
- ❹ *Sea environment* - If the mechanism model, or parts of it is submerged in water, or any other liquid, you may account for buoyancy if the *Water density* and *Sea water level* is specified. See [Section 3.9.3, "Buoyancy"](#).
- ❺ *Gravitation* - The gravitation vector direction and magnitude. See [Section 3.9.4, "Gravitation"](#).
- ❻ *Initial translational velocity* - The initial translational velocity of all triads in the model. See [Section 3.9.5, "Initial translational velocity"](#).
- ❼ *Model Link Repository* - Switch between external and internal model link repositories. You may also **Change** the external model link repository. See ["Setting the model link repository"](#) in [Section 4.1.5](#).



NOTE: Changes to the Model description notes will be saved also when pressing Cancel or closing the Model Preferences window. Therefore, you may edit these notes at any time, also when you have results.

The screenshot shows the 'Model Preferences' dialog box with the following fields and callouts:

- 1** points to the 'Model description' text area.
- 2** points to the 'Model database units' dropdown menu, which is set to 'SI'.
- 3** points to the 'Absolute modeling tolerance' text field, which contains '0.0001'.
- 4** points to the 'Sea environment' section, specifically the 'Water density' and 'Sea water level' fields.
- 5** points to the 'Gravitation' section, specifically the X, Y, and Z input fields.
- 6** points to the 'Initial translational velocity' section, specifically the X, Y, and Z input fields.
- 7** points to the 'Model link repository' dropdown menu, which is set to '[internal]'.

At the bottom of the dialog are 'OK', 'Apply', and 'Cancel' buttons.

3.9.1 Model database units

All modeling in Fedem is unit independent. However, some external interfaces, like tire and road, uses SI units in their calculations.

When modeling in other units than SI, you will need to change the model database units. Do this by selecting the model database units corresponding to your model. The chosen units are then used to properly scale calculated data before communicating with external modules that require a specific unit set (like tire/road data files).



TIP: You can add your own modeling units by editing the file `units.fcd` in any ASCII editor. This file is located in the Fedem installation directory.

3.9.2 Modeling tolerance

The modeling tolerance is a tolerance that controls how strict Fedem enforces that triads and their corresponding FE nodes are coincident. This tolerance needs to be strict, because an offset will introduce an error and inconsistency in the numerical model which might have serious impact on the reliability of the results.

The seriousness is however depending on the size of the offset compared to the size of the FE models in question and the overall size of the model. If you experience problems when trying to attach triads to a link, you might need to increase the modeling tolerance.

The default tolerance is set to $1e-4$ of the length unit you are using. That is a good tolerance when using meters as model database length unit, but is probably too strict when using millimeters. You will have to adjust this tolerance to some sensible value according to the size of your FE models and the units you work in.



WARNING! When decreasing the tolerance in a model that is built using a large modeling tolerance, some of the triads/joints might become invalidly attached when reopening the model.

3.9.3 Buoyancy

Buoyancy forces (and associated load correction stiffness) may be included for *Generic Part* links, if the link is assigned a geometry description file in the *Visualization* field in the Link property (see "[Part tab](#)" in [Section 4.1.4](#)). This geometry file has to define a closed volume that represents the total displaced fluid volume when submerged, and can be either on the VRML-format or Fedem's internal Cad format (`.ftc`).

The buoyancy force is computed from that part of the volume that is below the specified sea water level surface and is applied in the opposite direction of the gravitation vector. The water surface is a plane, with surface normal vector equal the opposite of the gravitation vector, and that passes through the point

$$-\frac{s_0}{\|\vec{g}\|} \vec{g}$$

where s_0 denotes the specified *Sea water level*, and $\vec{g}$ is the *gravitation* vector. The sea water level may either be a constant, or a Function of time (or any other response quantity) in order to simulate a wave event.



NOTE: *Simulating waves using a non-constant sea level assumes that the cross section between the buoyant volume and the water surface is small compared with the wave length, since the change in water surface normal is not accounted for.*

3.9.4 Gravitation

The gravitation size and direction can be adjusted in the Model Preference dialog. Remember to edit this in order to correspond to the units you are using.

The direction of the gravitation vector is displayed by the orange arrow in the lower left corner of the 3D modeler window (shown at right).



3.9.5 Initial translational velocity

The complete mechanism can be given an initial translational velocity by entering a velocity vector in the Model Preferences dialog. This velocity is distributed to all the Triads in the model.

This is useful if your event actually is describing the mechanism moving at some speed different from zero.

Chapter 4 Mechanism Elements

Now that you know how to create and assemble mechanism elements, you need to know how Fedem defines the properties of elements and how you can customize them to suit your design requirements. This chapter presents each of the mechanical and modeling elements used in Fedem mechanisms. It also describes each element's properties and how they can be edited once the element is created.

Sections in this chapter address the following topics:

- [Links](#)
- [Element groups](#)
- [Triads](#)
- [Joints](#)
- [Joint pair constraints](#)
- [Frictions](#)
- [Springs and Dampers](#)
- [Loads](#)
- [Tires and Roads](#)
- [Functions](#)
- [Sensors](#)
- [Strain rosettes](#)

4.1 Links

As described in [Chapter 1, "Introduction to Fedem"](#), links are the basic components of Fedem models. You can connect links with various types of joints to create a moveable *mechanism*. Each link is either an *FE Part* represented by an FE model with nodes, elements, materials and physical property data, or a *Generic Part* represented by a simplified model forming a semi-rigid connection between other mechanism entities.

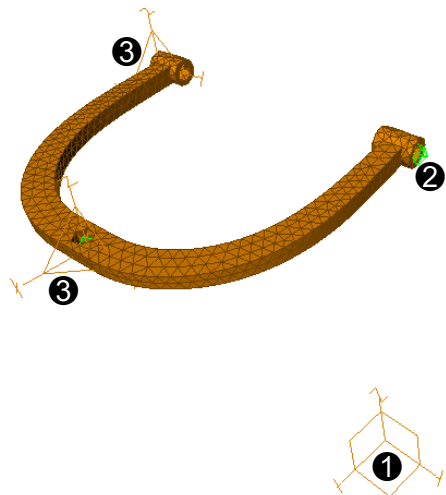
FE Parts

The mass properties, stiffness properties and dynamic properties of an FE Part are defined through its FE model, defined by nodes, elements, materials and physical property data.

The FE model must be constructed in an external FE modeler, such as one of those described in [Section 1.5, "Using FE models"](#), and then imported into Fedem.

You can use simple or complex links in your models, depending on your needs and modeling capabilities. Shown below is a simple FE link modeled with solid elements.

- ① *Link coordinate system* – The FE model representing the link is defined in the link coordinate system. When the link is imported into Fedem this coordinate system is aligned with the global coordinate system.
- ② *Triads* (optional) – nodal points that are defined as external during the construction of an FE model (in an external modeler) are automatically connected to a triad when the link is imported into Fedem. If external nodes are not defined in the FE model file, no triads will appear when the model is imported (see [Section 4.3, "Triads"](#) for more information about triads).
- ③ *Local coordinate systems* – Local FE coordinate systems present in the imported FE mesh is read and displayed for reference.

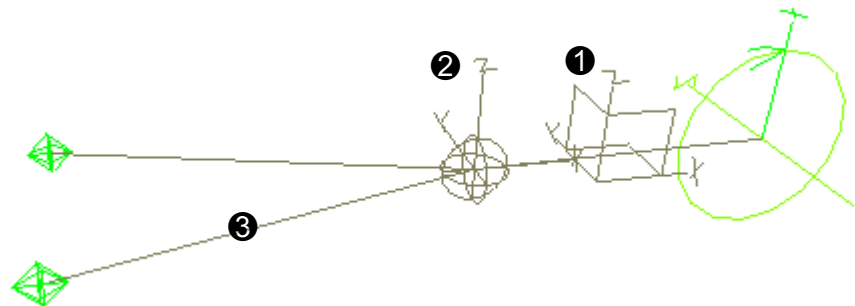


Generic Parts

A Generic Part is a simplified flexible body. It is purely defined by its connection points, mass properties and stiffness at the connection points. The stiffness can either be defined manually, or automatically set to some very high value, mimicking a rigid body. See the Fedem R5.0 Theory Guide, *Appendix A.16 "Generic Part element"*, for details on how a Generic Part is represented in the Dynamics Solver.

Generic Parts can be used when you have no FE model for the link, when trying to optimize hardpoint positions, or when the flexibility of the part is considered to be insignificant. They can also have a VRML geometry attached, to give better visualization of the link.

Shown below is a link defined as a Generic Part with two triads and a Revolute Joint connected to it.



- ① *Link coordinate system* - At the time of creation, the link coordinate system is in the global origin.
- ② *Centre of Gravity* - The Generic Part's centre of gravity can be positioned independently from the link coordinate system.
- ③ *Simplified visualization* - The lines extending from the centre of gravity to each of the triads attached provides a coarse visualization of the Generic Part.

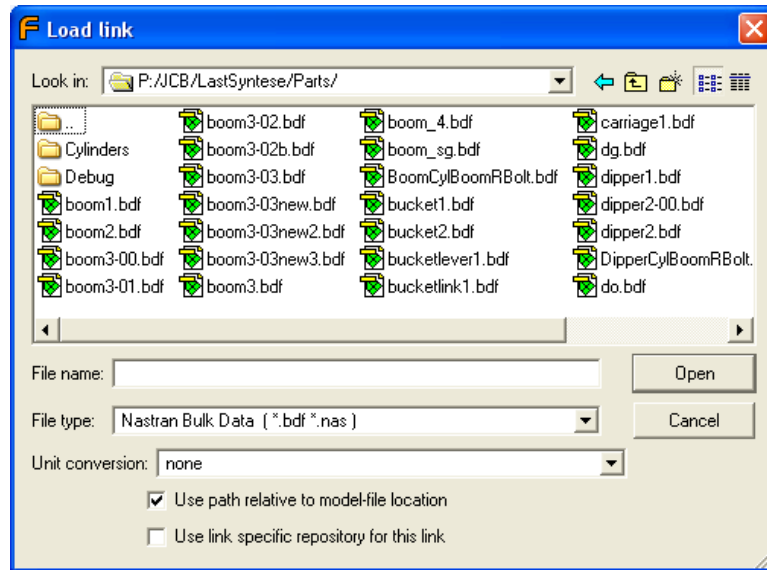
4.1.1 Creating links by file import

Links can be created by importing FE model files or by importing CAD geometry as VRML files. The available file formats are listed in [Section 2.3, "Storing models and results"](#).

Importing FE models will create a link and set it up as an FE Part, while importing a VRML file will create a link, and set it up as a Generic Part. To do so, complete the following steps:



1. Click the *Load Link* button on the Mechanism Creation toolbar (or select from the File menu). The dialog shown below then pops up.



2. Select the file type you are looking for from the *File type* pull-down menu.
3. Browse for or enter the path and filename of the link file in the *File Name* box.



TIP: You can import more than one link at the same time by holding down the **Ctrl** or **Shift** key and selecting multiple files in the Open dialog.

4. Select the units conversion you need from the *Unit Conversion* pull-down list. All units used in the file for dimensions and properties are converted according to your selection.



WARNING! There is no connection between this unit conversion and the model database units. You must be careful to choose the conversion that fits your needs.



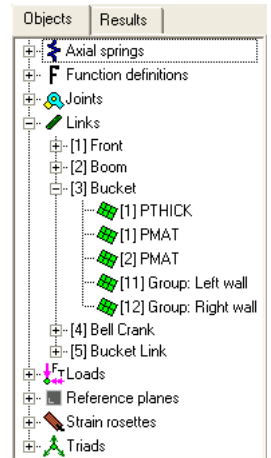
TIP: You can add your own unit conversions by editing the file `units.fcd` in any ASCII text editor. This file is located in the Fedem installation directory.

5. Make sure that you want to store the relative path to the original FE model or the VRML model. This setting is relevant when copying and moving the model across file systems, and has most impact on VRML files. The path to the original FE model file is only used if the internal FE model copies are lost.

6. Indicate if you want this link to be a part of the model link repository, or if it should be stored with a link specific repository. See [Section 4.1.5, "Using link repositories"](#).
7. Once you have selected the file(s) and a unit conversion, click **Open**. The link files are imported, and the coordinate system of the links are aligned with the global coordinate system.

When an FE model is imported, it is converted to the internal Fedem format. During this conversion, several element groups might be created as well. These element groups can either be user-defined explicit groups, or implicit groups based on the properties of the finite elements in the link.

All element groups associated with a FE link are listed in the Objects list in the Model Manager panel, as illustrated to the right. See [Section 4.2, "Element groups"](#) to learn more about element groups in Fedem.



4

4.1.2 Creating links from hard points



If you only have the hard point information of a link, you can create a link from triads positioned at the hard points.

To do this, follow these steps:

1. Select some or all of the triads that represents the hard points of the link using the multi select features of Fedem. See [Section 2.5.3, "Model Manager"](#) and [Section 2.6.1, "Select"](#) on how to select multiple objects either from the *3D Modeler Window* or in the *Model Manager Objects* list.
2. Click the *Create Generic Part* button on the *Mechanism Creation* toolbar. A Generic Part will be created with its origin in the global origin, and with its centre of gravity in the geometric centre of the triads you selected.
3. Set up the centre of gravity and mass properties of the link.
4. Optionally add triads to the link by attaching them.



TIP: You can at any time during modeling attach or detach new triads to the Generic Part. See [Section 3.6, "Attaching and detaching elements"](#).

4.1.3 Copying links

If you need to use the same FE model or Generic Part more than once, you can duplicate an existing link by completing the following steps:



1. Select the link you want to copy from the Modeler window (or Model Manager Objects list).
2. On the Edit menu, select *Copy Link*.

The new link is placed offset from the original.



TIP: You can also copy links using the shortcut menu in the Model Manager Objects list. Right-click the link you want to copy and select **Copy Link** from the shortcut menu. The new link is placed offset from the original in the Modeler window.

4.1.4 Link properties

Links, both FE Parts and Generic Parts, are the basic components of any Fedem mechanism. It is therefore essential to understand the link properties that are displayed in the Property Editor panel when you select a link.



TIP: In addition to the settings found in the Property Editor panel, you may also find some information on the link's underlying FE model by selecting the link in the Result File Browser (see [Section 8.2.1, "The Result File Browser dialog"](#)).

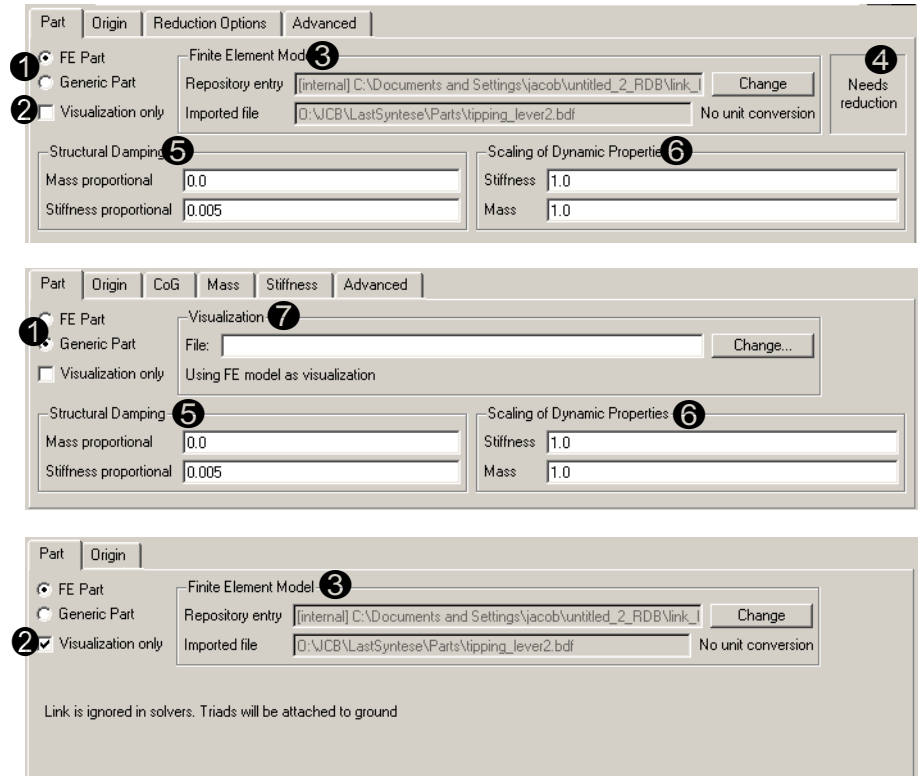
The link properties are separated into several tabs to better organize the different settings. The number of tabs and their content depend on whether the link is defined as a Generic Part or an FE Part, or if it is a link used for visualization only (grounded links).

The different tabs are as follows:

- *"Part tab"* - Always present
- *"Origin tab"* - Always present
- *"Reduction Options tab"* - Present for FE Parts (not for grounded links)
- *"Reduced Loads tab"* - Present for FE Parts, if element or nodal point loads are present in the FE data file (not for grounded links)
- *"CoG tab"* - Present for Generic Parts only (not for grounded links)
- *"Mass tab"* - Present for Generic Parts only (not for grounded links)
- *"Stiffness tab"* - Present for Generic Parts only (not for grounded links)
- *"Hydrodynamics tab"* - Always present unless the link is grounded
- *"Advanced tab"* - Always present unless the link is grounded

Part tab

The *Part* tab displays some basic settings and information about the link. The actual options that are displayed depend on whether the link is a Generic Part, an FE Part or if it is used for visualization only. Below, three versions of this panel are displayed.



- ① **FE Part/Generic Part** - Link type selector. You can switch between the FE model or the Generic Part model at any time during modeling.



NOTE: When switching between FE Part and Generic Part, Fedem tries to use the supplied FE data and visualization data in a sensible way. If your Link is defined as an FE Part and you switch to Generic Part, the FE model is retained in memory and used as visualization unless a VRML model file is specified. Fedem will not drop the FE data from memory until you actively enter or change the VRML model file name. When switching back to FE Part, Fedem will check that all the Triads are validly attached to the FE model. See also [Section 3.6.7, "Invalid attachments"](#).

- ② **Visualization only** - You can define the link to be used for visualization only. The link will then be ignored by the solvers and actually serve as

an extension of the *Reference plane* (see [Section 3.3.1, "Reference Plane"](#)). The link and all triads attached to it will thus be *grounded*.



NOTE: You can toggle a link as Visualization only at any time during modeling, but it is disallowed if the link has slave triads attached.

- ③ **Finite Element Model** (FE Parts only) - This group of options concerns the FE model used.
 - *Repository entry* indicates the repository type and the name of the selected .ftl file in Fedem's link database repository (for details on the link repository, refer to [Section 4.1.5, "Using link repositories"](#)). If the model file has not yet been saved, this entry will state the file name Fedem will use when you save your model.
 - *Imported file* indicates the name and location of the originally imported FE model file. The unit conversion that was applied during the import is also displayed.
 - The *Change* button allows you to replace the FE model of the link with a new one. The button triggers a file dialog in which the new link file can be chosen. Note that all mechanism entities attached to the link, that do not have corresponding nodal points in the new FE model, will be detached.
- ④ **Needs reduction** (FE Parts only) - This label is a flag that signals if your FE model has been reduced or not. If some data for the reduced link is present and is recognized to match the link, this entry will read *Reduced [n]*, where "*n*" is a number referring to the directory in the link database containing the reduced matrices.
- ⑤ **Structural Damping** – Allows you to change the values of both the mass and stiffness proportional damping for the link. These parameters are described in the Fedem R5.0 Theory Guide, [Section 7.5, "Structural damping"](#).
- ⑥ **Scaling of dynamic properties** – Allows you to scale stiffness and mass of each individual link in the dynamics simulation. The scaling is done during initialization and stays in effect for the entire analysis. This option is useful for sensitivity studies of deflection and stiffness.



WARNING! The mass and stiffness scaling is not accounted for in any of the recovery analyses, and the recovered result will thus be misleading on FE links using mass and/or stiffness scaling.



WARNING! The damping matrix and the associated force vector are not affected by the mass- and stiffness scaling parameters. That is, when using mass- and/or stiffness-proportional damping, it is the unscaled mass and stiffness matrix that contributes to the damping matrix and force vector.



WARNING! The mass- and stiffness scaling is not accounted for during FE link reduction. Therefore, the component mode shapes are always computed from the unscaled mass and stiffness matrix. Using stiffness and/or mass scaling on an FE link having component modes might thus yield inaccurate results unless the two scaling factors are equal (because the component modes then are computed from a different set of matrices than the one used in the dynamics simulation).

- 7 **Visualization** (Generic Parts only) - This frame contains options and information regarding the visualization of the Generic Part. The *File* field can contain a path to a VRML file to use as a visualization for the Generic Part. Press the *Change* button to browse for a file. If a valid FE model file is already referenced, fedem will use that as a visualization until a VRML model file name is entered.

Origin tab

The *Origin* tab is used to display and edit the position and orientation of the link. See [Section 3.5.4, "Origin property"](#) for a description of the data fields in this tab.

Reduction Options tab

The settings on the *Reduction Options* tab affect how the link is reduced.

- 1 **Singularity criterion** - This is the tolerance used to decide whether the the stiffness and mass matrices are singular when they are factorized during model reduction. See ["Singularity tolerance"](#) in [Section 6.3.4](#).

- ② *Component modes* - Allows you to specify the number of component modes representing the internal (eliminated) nodal degrees of freedom after CMS model reduction. See [Section 6.3.2, "Using component modes"](#).
- ③ *Eigenvalue tolerance* - This is the maximum acceptable relative error in the computed eigenvalues in the fixed boundary eigenvalue analysis.
- ④ *Consistent mass matrix* - Enables the use of consistent mass matrix in the model reduction process. If disabled, a lumped mass matrix approach is used. See [Section 6.3.3, "Using lumped mass matrix"](#).
- ⑤ *Ignore check-sum test* - Disables the check on whether the reduced link data, found in the link repository, is consistent with the link file currently used. Due to some rare numerical inconsistencies between reduced file data and the read link data file, Fedem may signal that a link file needs reduction even though the reduced data are present.



CAUTION: Do not enable the Ignore check-sum test toggle unless you are sure that the reduced link data found on disk are compatible with the current model. The consequence of using incompatible link data may be a diverging model or incorrect results. A warning is issued whenever this toggle is enabled to stress this.

- ⑥ *Expand mode shapes* - Enables the expansion of component mode shapes and free-free mode shapes of the reduced link, for subsequent visualization. See [Section 6.3.6, "Visualization of eigenmode shapes from the model reduction"](#) and [Section 7.3, "Animations"](#).
- ⑦ *Eigenvalue Factorization* - Allows you to specify which matrix to be Choleski-factorized during the eigenvalue analysis that is performed in the component modes computation. Default is the mass matrix.
- ⑧ *Recovery matrix storage precision* - Allows you to switch to *Single precision* storage of the recovery matrix (a.k.a. the B-matrix) on disk. This will reduce the needed disk space for this matrix by 50%, and might be advantageous for very large links with many triads that will result in a big B-matrix. The default is to use *Double precision* storage.
- ⑨ *Reduced/Needs reduction* - See ["Part tab"](#) above.



CAUTION: Switching to single precision storage of the B-matrix should normally have no influence on the dynamics simulation results. However, if the link's FE model is poorly conditioned (e.g., there is a large span in the stiffness properties over the link) there might be minor loss of accuracy in the recovery results due to the truncation of the B-matrix elements stored on file.



NOTE: For links that are reduced with component modes (see bullet 2 above), the single/double precision storage option also applies to the file containing the component mode shapes (the E-matrix file).

Reduced Loads tab

On this tab you can assign time histories for load cases that are defined in the FE link data file. The associated reduced consistent load vectors are computed by the link reducer and stored in the link repository together with the reduced stiffness and mass matrices.

Part	Origin	Reduction Options	Reduced Loads	Advanced
Load Case	Delay	Load Amplitude		
1	0.0	2.3		
2	0.0	[1] Load amplitude (Function)		

- 4
- ❶

Load Case - This column contains the user-defined load case ID for each load set defined in the FE link data file.
- ❷

Delay - If the load amplitude is defined by a Function, this value is used as a shift to the function argument, i.e., if the amplitude is specified as a function of time, $f(t)$, then the actual amplitude becomes $f(t-\text{Delay})$. This is useful when each load case define the load state at different times in a transient simulation. The *Delay* can then be set equal to the time where each load case is active, and the same function can then be applied to each load case to obtain a smooth transition from one load case to the next.
- ❸

Load Amplitude - You can either enter a constant value or select a Function (or Time history input file) from the pull-down menu. This value or function will then be used as a scaling of the reduced load vector associated with this load case in the dynamics simulation.

CoG tab

On this tab you can edit the position of the centre of gravity for a Generic Part. You can also enter the orientation of the principal axis of inertia to be used as the reference for the inertias entered on the *"Mass tab"*.

Part	Origin	CoG	Mass	Stiffness	Advanced
Center of Gravity			Principal Axes of Inertia		
Reference CS		Coordinate type	Reference CS		Coordinate type
[1] (Link)		Cartesian XYZ	[1] (Link)		EulerZYX as X, Y, Z
X	Y	Z	RotX [Deg]	RotY [Deg]	RotZ [Deg]
-0.080933712	-0.080933712	0.0	0.0	0.0	0.0
☐ Condense out Center of Gravity DOFs					

- ❶

This toggle enables the elimination of the DOFs associated with the centre of gravity in the dynamics simulation. It is used to remove potential artificial internal vibrations in the Generic Part, and thus increase the numerical stability of the model.

Mass tab

The settings on this tab concern the mass and inertia properties of a Generic Part, and is used to establish the link's mass matrix (see the Fedem R5.0 Theory Guide, *Appendix A.16 "Generic Part element"*).

The screenshot shows the 'Mass' tab of a software interface. It contains a 'Calculate from FE model' checkbox (callout 4) and an 'Inertia Reference' dropdown menu (callout 3) currently set to 'CG Orientation'. The 'Mass and Inertias' section contains input fields for 'Mass' (callout 1) set to 0.1, and a lower triangle of inertia matrix fields (callout 2) with values 0.0 for Ixx, Ixy, Iyy, Ixz, Iyz, and Izz.

- ❶ *Mass* - The total mass of the link.
- ❷ *Inertias* - The lower triangle of the inertia matrix at the centre of gravity of the link.
- ❸ *Inertia Reference* - Select whether to specify the inertia in the directions of the link coordinate system or in the directions specified as *Principal Axes of Inertia* on the "*CoG tab*".
- ❹ *Calculate from FE model* - The link mass and inertias, as well as the centre of gravity in the "*CoG tab*", will be calculated from the FE model, when available, if this toggle is on. The *Inertia Reference* will also be set to *Link Orientation* and editing of the fields are disabled.

Stiffness tab

The stiffness properties of a Generic Part can be controlled on this tab.

The screenshot shows the 'Stiffness' tab of a software interface. It features a 'Type' section with 'Automatic' and 'Manual' radio buttons (callout 1). The 'Properties' section (callout 2) contains two input fields: 'Translational stiffness at each triad' and 'Rotational stiffness at each triad', both set to 1 000 000.0.

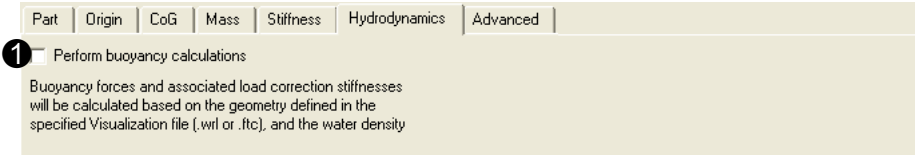
- ❶ *Type* - These radio buttons choose whether to set the overall Generic Part stiffness manually, or to have Fedem calculate a near rigid stiffness for you.
- ❷ *Properties* - This frame contains the stiffness values for the manually selected stiffnesses.

When setting the stiffness calculations to *Automatic*, Fedem uses the mass and a high target eigen frequency to calculate a sensible high stiffness. This will work as long as the mass of the link is set to something sensible. Thus Fedem is not able to calculate a good stiffness for a link with no mass at all, or with a mass that does not correspond to the actual use of the link. In such cases you will need to set the stiffness manually.

See the Fedem R5.0 Theory Guide, *Appendix A.16, "Generic part element"* for details on how the Dynamics Solver derives a link stiffness matrix from the manually specified stiffness values.

Hydrodynamics tab

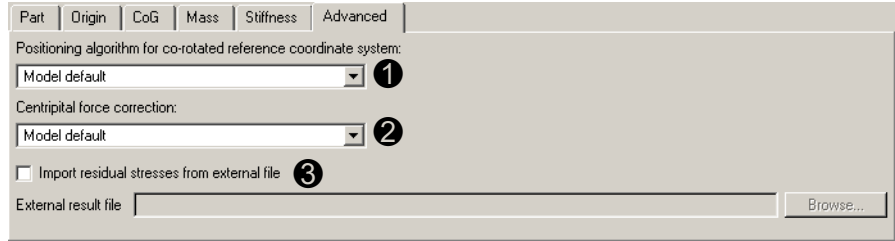
This tab concerns the calculation of hydrodynamic forces on a link.



- 1** *Perform buoyancy calculation* - Enables the calculation of buoyancy forces for this link, provided a link geometry and sea environment is defined. See [Section 3.9.3, "Buoyancy"](#).

Advanced tab

This tab contains drop-down menus for selection of positioning algorithm for the co-rotated reference coordinate system of the link, and for using centripetal force correction for the selected link. It also contains a toggle and field for specification of an external result file for import of residual stresses in the Stress and Strain Coat Recovery processes.



- 1** The options for the co-rotated reference coordinate system are:
- *Model default* - The global setting defined in the *"Integration tab"* of the Dynamics Solver Setup is used (see [Section 6.5.1, "Dynamics Solver Setup"](#)).

- *Max triangle, with unit offset when necessary* - This is the original algorithm, the only one available in Fedem R3.1.1 and earlier.
- *Max triangle, with link scaled offset when necessary* - The same as above, but with adjustments of the offset to better fit the link size.
- *Mass based nodal average* - Algorithm based on equilibrium of a rigid shadow element with averaged stiffnesses at the triads.

See the Fedem R5.0 Theory Guide, *Section 4.1, "Superelement local coordinate system"* for a detailed description of these algorithms.

- ② The options for the centripetal force correction are:
 - *Model default* - The global setting defined in the *"Integration tab"* of the Dynamics Solver Setup is used (see [Section 6.5.1, "Dynamics Solver Setup"](#)).
 - *On* - Turns centripetal force correction on for this link.
 - *Off* - Turns centripetal force correction off for this link.
- ③ By enabling this toggle, you may browse for an external result file from which a residual stress state is imported in the Stress and Strain Coat recovery processes. The supported file formats are:
 - SDRC universal file (*.unv*)
 - ABAQUS result file (*.fil*)
 - ANSYS result file (*.rst .rth*)
 - NASTRAN Output2 file (*.op2*)

See [Section 6.6.3, "Import of residual stresses from external analyses"](#) for further details on the usage of this feature.

4.1.5 Using link repositories

A link repository is a directory structure containing all the files related to one or more FE links. This includes the finite element model files, reducer input option files, the reduced matrix and load files, the displacement recovery matrix files, and log-files with text-based output from the link reduction processes.

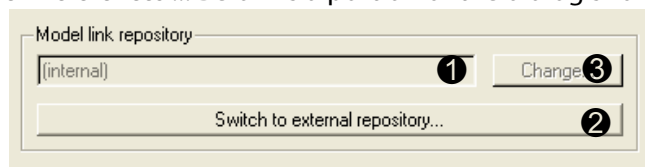
The term link database is also used when referring to a link repository. For detailed information about link databases, see [Section 8.3.1, "Link database"](#).

Fedem can handle the storage of a link repository in three different ways:

- *Internal link repository* (default) – The default link repository is placed inside the model results database in the `link_DB/` directory. This link repository will follow the model, and be copied along with the results if saving the model as a new name.
- *External link repository* – Sometimes it is useful to be able to share, and reuse the link repository among several model files. Using an external link repository enables such sharing. If this is set, using *Save As* will not copy the link repository, but the original and the new model will point to the same link repository and thus share any identical information.
- *External single link repository* – It is possible to use a specific link repository for an individual link. This is used to import and reuse a link from an existing link repository.

Setting the model link repository

The model link repository can be set in the *Model Preferences* dialog. *Edit->Model Preferences ...* Below is a portion of the dialog shown.



- ❶ The position of the current model link repository.
- ❷ This button switches between internal and external repository.
- ❸ The *Change* button changes the external repository to a new directory. This might be an empty directory, or a directory used as link repository by another model.

When changing the repository, Fedem will copy the active content of the current link repository to the new destination prompting on whether to overwrite if necessary. The old repository will be left untouched, unless it is an internal repository. Internal repositories will be deleted when switching to an external repository.

If there exists files with the same names at the new destination, Fedem will try to find out whether the reduced data match the existing FE model. If it does the reduced data will not be copied, as identical data is assumed to exist at the new destination.

Reusing a link from an existing link repository

To import and reuse the reduced data for a link, the link can be imported from the existing link repository using the *Load Link* command. See [Section 4.1.1, "Creating links by file import"](#). Select the *.ftl* file you want, and toggle on the *Use link specific repository* toggle in the *Import Link* dialog. You will then need to add triads and reducer options for that particular link, that match the options and triad positions used by the previously reduced link. When done, Fedem will detect the reduced files, and flag the link as *Reduced*.

4.2 Element groups

When a link is created by importing a FE model into Fedem, several element groups might be created as well, see [Section 4.1.1, "Creating links by file import"](#). The element groups can be of the following two types:

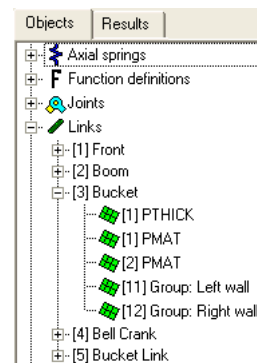
- **Explicit groups** User-defined group through Nastran SETs on bulk data files or Fedem GROUPs on *.ftl* files.
- **Implicit groups** An implicit element group consists of all elements referring to one particular material (PMAT) or thickness (PTHICK) property record in the FE model file. The ID numbers of these groups correspond to the ID numbers of the associated property record. Implicit groups are created only for property records that are in use.



NOTE: When the imported FE model file is a Nastran bulk data file, the created PMAT and PTHICK groups correspond to the MAT1 and PSHELL bulk entries, respectively, with corresponding ID numbers. No implicit groups are created for PSOLID bulk entries.

The element groups are visible in *Objects* list of the Model Manager panel under each link node, as shown to the right. They can be used to control component appearance (see [Section 2.7, "Visualizing the model"](#)), calculation focus (see [Section 6.2, "Link- and group-wise solving"](#)).

The element groups are also used to assign properties needed in Fatigue analyses during Strain Coat Recovery simulations (see [Section 4.2.1, "Element group properties"](#) below, and [Section 6.9, "Strain coat analysis"](#)).



Some Nastran bulk data files may also contain a user-defined name of an element set or physical property as a comment line before the set/property definition itself. When found, such comments are parsed and used in the default description of the created element group when the bulk data file is imported into Fedem.



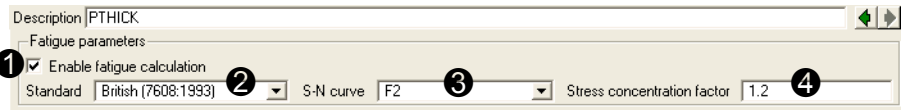
NOTE: The syntax of the comment lines containing names on element sets and properties depends on the software package that produced the actual Nastran bulk data file. Currently the syntax of the following packages are supported: I-DEAS, Hypermesh and NX.



TIP: The description field can be edited both for explicit and implicit element groups. The modified description is then stored in the link file (.ftl file). To revert to the original description (e.g. PTHICK for implicit groups based on the thickness element property), delete the description text completely. Any user-defined name parsed from a Nastran bulk data comment line is not restored, however.

4.2.1 Element group properties

When an element group is selected in the Model Manager panel, its properties are displayed in the Property Editor panel (shown below). It contains parameters and settings that are used in Fatigue calculations during a Strain Coat analysis on this element group. See [Section 6.9, "Strain coat analysis"](#) to learn more about such fatigue analyses.



- ① This toggle enables rainflow analysis and fatigue calculations for the selected element group.
- ② *Standard* - Select the fatigue standard to use for in the fatigue calculations.
- ③ *S-N curve* - Select an S-N curve from the selected fatigue standard.
- ④ *Stress concentration factor* - The computed stresses are scaled by this value before they are used in the fatigue calculation.



TIP: The S-N curve standards available in the pull-down menu are defined in the file *sn_curves.fsn* located in the installation directory of Fedem. The syntax of the S-N curve definitions is description in the header of this file, and it is possible to add your own S-N curve definitions to that file.

For details on how damage is calculated from a given time history response, see the Fedem R5.0 Theory Guide.

4.3 Triads



To construct a working mechanism, links are connected to each other using elements such as joints, springs, dampers, loads, and so on. Fedem uses a modeling object called a *triad* to make these connections. The triad defines a set of three mutually perpendicular coordinate axes originating from the connection point.

Triads enable links to be connected to other mechanism elements using the links' FE nodes as connection points. When links are joined together in your model, the connection points are retained after the model reduction process as external FE nodes (see the Fedem R5.0 Theory Guide, *Chapter 3, "Model Reduction"*). During simulation, a triad must move rigidly in translation and rotation with the FE node to which it is attached. This means that triads (and therefore connections) can only be placed on FE nodes with six degrees of freedom (DOFs), since three-DOF solid nodes allow random rotation.

4.3.1 Triads in joints

Triads are used to connect joints to links in the model in the same way that a door hinge uses one hinge-plate to attach the hinge to the door and the other to attach the hinge to the frame. Each type of joint may use a different number of triads to make the required connections. When a joint is created, its triads are created along, and positioned automatically. (See [Section 4.4, "Joints"](#) for more information about how triads are used in joints.)

4.3.2 Triad symbols

Because triads can be used for several different purposes—"building blocks" for joints; attachment points for springs, dampers, and loads; and measuring points for sensors—different symbol is used to visualize the triads based on its usage.



TIP: When attached, all triads are the same color, regardless of the symbol used to represent them. To identify a triad, simply look for the triad color (default = green). (Click the **General Appearance** button to access options for changing the color of triads and other mechanism elements.)

Diamond

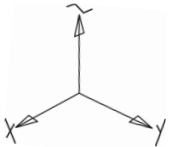
A triad is depicted as a small diamond (shown at right) when the triad's *coordinate system* is *not* used in your mechanism—for example, when a triad is placed on an



FE node and no other mechanism element is attached, or when it is only used to attach a load, torque, axial spring, or damper.

Coordinate system

A triad is visualized as a coordinate system (shown at right) when the triad stands alone, and its coordinate system is referenced in one of the following ways:



- The triad’s coordinate system is used by a sensor to measure a variable with components defined in the local (triad) coordinate system (see [Section 4.11, "Sensors"](#)).
- The triad’s coordinate system is used to define mass or inertia components for the triad (see [Section 4.3.3, "Triad properties"](#) below).
- The triad’s coordinate system is used to define boundary conditions for use in the initial equilibrium analysis and/or the system eigenvalue analysis (see [Section 4.3.3, "Triad properties"](#) below).

Member of a Joint

The triads that are members of joints are visualized as integral parts of the different joint symbols. Refer to [Section 4.4, "Joints"](#) for details.

4.3.3 Triad properties

To edit the properties of an existing triad, select the triad in the Modeler window (or Model Manager Objects list) and make your changes in the Property Editor panel (shown below).

- ❶ *FE Node* – This area provides the number of the FE node to which the selected triad is attached and the triad’s position in local (link) or global coordinates.
- ❷ *Additional Masses* – These options enable you to apply additional mass and inertia to the triad.

- ③ *Additional Boundary Conditions* – These options enable you to restrain the triad's movement during the initial equilibrium analysis and optionally in the eigenmode analysis. (See also "[Eigenmode tab](#)" in [Section 6.5.1](#) and the Fedem R5.0 Theory Guide, [Section 7.8](#) "Quasistatic equilibrium" and [Section 9.6](#), "Eigenvalue results".)
- ④ *Origin* – This tab contains the *Origin properties* of the triad. See [Section 3.5.4](#), "[Origin property](#)".



TIP: The FE node number can be useful if you edit the link (.`link`) file manually (described in [Appendix C](#), "[FE Link Interface](#)").

4.4 Joints

As with real mechanisms, you connect each link to the others using *joints*. A joint introduces motion and/or spring constraints between the two links it is acting between. These constraints are applied on the *joint degrees of freedom* (Joint DOFs) also called *Joint Variables*. Each Fedem joint uses at least two triads to connect the joint to links. One or more of the joint's triads are labeled "master" while one triad is labeled "slave", with the constrained DOFs of the slave triad following the movement of the master(s). (See the Fedem R5.0 Theory Guide, [Chapter 6](#), "[Modeling of Joints](#)".)

To attach a joint to links, the joint's master triad is attached to an FE node on one link and the slave triad to an FE node on another link. This means that when the mechanism moves, the FE node (and link) on the slave side of the joint follows the motion of those on the master side. FE nodes and links can, therefore, also be referred to as masters and slaves. See [Section 3.6](#), "[Attaching and detaching elements](#)" and [Section 3.6.1](#), "[Attaching using Attach](#)" about how to attach joints.



TIP: To determine which triad is the master and which is the slave, select one of the links or the joint to examine the *Topology List* of master and slave triads connected to the link/joint. You can then select (click and hold down the mouse button) the master/slave triad to highlight it in the *Modeler* window.

4.4.1 Joint variables

The joint variables are the accessible or controllable degrees of freedom for each joint. As an example, the Revolute Joint normally has one accessible degree of freedom, namely the rotation about one axis. The other degrees of freedom are fixed. For most joints the DOFs that are not accessible are fixed, but for Prismatic and Cylindrical joints that is not the

case. Refer to [Section 4.4.4, "Prismatic joint"](#) and [Section 4.4.4, "Cylindric joint"](#) for further details.

The behavior of the joint variable can be controlled or customized in several ways. There are four main options.

- Fixed - This DOF is fixed, and can not be moved. It is removed from the system of equations (condensed out).
- Free - This DOF is free to move. No constraints are applied. This is the default setting.
- Prescribed - This DOF can be assigned a prescribed motion, and is thus condensed out from the system of equations.
- Spring-Damper - This DOF is free to move, but a spring and a damper may be applied to assign stiffness and/or damping properties to it.

Integrated springs and dampers

When setting a joint variable to be spring and damper controlled, the joint springs and joint dampers are initially inactive (their properties—including spring stiffness and damper coefficient—are initially set to zero). You can then assigning values to the joint's spring and damper properties in the Property Editor panel (see [Section 4.4.2, "Joint properties"](#) and [Section 4.7, "Springs and Dampers"](#) for more information about the behavior of springs and dampers.)

The integrated springs and dampers are listed in the Topology List of the joint as separate items; they are however not listed in the Objects list in the Model Manger. You can access the full property panel for the joint springs and dampers by double-clicking on the entry in the Topology List, but the normal way of editing their properties is through the DOF tabs in the Joint property panel (see ["Joint variable properties"](#) below).

4.4.2 Joint properties

You can select a joint to display its properties in the *Property Editor panel* (shown below for a Revolute joint). The *Property Editor panel* for a joint consists of one tab with a summary table of all the major joint properties, and additional tabs for each of the *joint variables* where their properties are displayed in detail. The summary table shows a non-editable summary of all joint variables along with other editable joint properties. Finally, there is an *Origin* tab for the point-to-point joint types (Rigid, Revolute, Ball and Free joints) and an *Advanced* tab with further properties for Ball and Free joints.

Since the number of joint variables depends on the joint type, you may see from zero (Rigid joint) to six (Free joint) sets of joint variables listed in the summary table and a similar number of joint variable tabs.

123

SummaryOriginTzRz

	Constraint	Load	Model length	Init disp.	Length change	Spring	Spr. scale	Damper	D	Dmp. scale
Tz	Free	0.0	0.0							
Rz	Free	0.0	0.0							

Friction4

None

☒ Z translation DOF

- 1 Summary tab – This tab displays a "Summary table" over all properties of the joint.
- 2 Origin tab - This tab contains the definition of the joint coordinate system, i.e., the position and orientation of its origin (see Section 3.5.4, "Origin property"). The joint variables are defined in this coordinate system. See also "Moving point-to-point joints" in Section 4.4.3.
- 3 Joint variable tabs – These tabs display the properties related to the joint variable in question (see "Joint variable properties" below).
- 4 Friction – Some joint types allow you to add friction properties to the joint by selecting from the list of frictions in your model (see Section 4.6, "Frictions").
- 5 Advanced tab – This tab (not shown above) displays properties related to rotation formulation and spring inter-connectivity of the joint (see "Advanced joint properties" below).

Joint variable properties

The joint variable tabs display the different options and settings for each joint variable. The displayed options depend on the Constraint Type.

Fixed

SummaryOriginTzRz

Constraint Type

☒ Fixed☐ Free☐ Prescribed☐ Spring-Damper

Rotation control

Angle in model0.0

- 1 Length/Angle in model - This field shows the fixed value of this joint variable as modeled. For rotational DOFs the value can be edited to

set a different fixed rotation state. The 3D view will then update instantly, showing the new rotation in the joint symbol.



TIP: You may plot the reaction force associated with a fixed joint DOF by selecting the "Force/Moment value" item under the joint variable node in question from the RDB selector (see "Selecting RDB results" in Section 7.2.5, "Curve properties").

Free

- ① **Load magnitude** - You can apply a *Load* on the joint variable. This will be a torque or a force depending on whether the joint variable is a translational or a rotational DOF. The actual force value will be saved as a results quantity and thus available for plotting in a graph.
- ② **Length/Angle in model** - This field shows the initial value of this joint variable as modeled. The value defines the initial configuration of this joint variable in the dynamics simulation. For rotational DOFs the value can be edited to set a different initial rotation. The 3D view will then update instantly, showing the new rotation in the joint symbol.

4

Prescribed

- ① **Prescribed quantity** - You may choose whether you want to prescribe the *Deflection*, the *Velocity* or the *Acceleration* for the joint DOF.
- ② **Length/Angle in model** - This field shows the initial value of this joint variable as modeled, and has the same interpretation as when the joint variable is *Free* (see above).
- ③ **Initial length/angle, Initial displacement/rotation** - These radio buttons and fields work together to allow you to set the value that defines the initial configuration of the joint variable in the dynamics simulation.

You can choose to enter the initial length/angle either as an absolute value, or relative to the *Length/angle in model*.



CAUTION: If you define an Initial length/angle that differs from the Length/Angle in model, this difference will be accounted for in the very first iteration of the dynamics simulation and thus lead to a dynamic shock effect. However, when the initial *"Static equilibrium analysis"* is switched on, the force due to this difference is taken as a pure static load and the transient shock should be avoided.

- ④ **Length/Angle change** - You can prescribe the motion by a function. The function controls the *change* of the variable relative to the *Initial length/angle*. If you rather would like to prescribe the absolute value of the variable, set the *Initial length/angle* to zero.



TIP: You may plot the reaction force and input energy associated with a prescribed joint DOF by selecting the "Force/Moment value" and "Input Energy" items under the joint variable node in question from the RDB selector (see *"Selecting RDB results"* in [Section 7.2.5, "Curve properties"](#)).

Spring-Damper

These options enable you to add elastic and damping behavior to the joint variable by entering values for the spring and damper properties.

- ① **Load magnitude** - You can apply a Load in the joint variable. This will be a Torque or a Force depending on whether the joint variable is a translational or a rotational degree of freedom.
- ② **Stress free length/angle control** - This group of options concerns spring deflection calculation similar to the *Length/Angle control* of a *Prescribed* joint variable. See also [Section 4.7.1, "Spring properties"](#).
- ③ **Spring properties** - This group of options concerns the spring characteristics, namely the relation between deflection and force. See [Section 4.7.1, "Spring properties"](#) for details.
- ④ **Damper force/coefficient** - This group of options concerns the damper characteristics, namely the relation between velocity and force. See [Section 4.7.2, "Damper properties"](#) for details.

Summary table

The summary table displays an overview of the settings for each joint variable. The columns relate to the fields in the *"Joint variable properties"* described above.

Summary	Origin	Rx	Ry	Rz	Advanced						
	Constraint	Load	Model length		Init disp.	Length change	Spring	Spr. scale	Damper	D	Dmp. scale
Rx	Free		0.0	0.0							
Ry	Free		0.0	0.0							
Rz	Free		0.0	0.0							

- *Constraint* – Shows the *Constraint type* selected for the joint variable.
- *Load* – Shows the load applied to the joint variable.
- *Model length* - Shows the modeled length/angle of the joint variable.
- *Init. disp.* – Initial deflection: This column shows the initial deflection set up for the variable.
- *Length change* – Displays the function, if any, that controls the change of length/angle of the joint variable. If the constraint type is set to *Prescribed* this change is directly applied to the joint DOF. If the Constraint type is set to *Spring-Damper*, however, this function controls the *Stress free length/angle change* of the spring.
- *Spring* – Spring characteristic: Either a number describing a constant stiffness of the spring, or a description of the spring characteristics used as a non-linear stiffness- or force-deflection relationship.
- *Spr. scale* – The description of the optional function that can be used to scale the force developed in the spring. Empty if no scaling is done.
- *Damper* – Damper characteristic: Either a number describing a constant coefficient of the damper, or a description of a function used as a non-linear coefficient- or force-velocity relationship.
- *D* – Deformational velocity: A label indicating whether the damper is using deformational velocity. "d" if on, empty if not.
- *Dmp. scale* – The description of the function that is used to scale the force developed in the damper. Empty if no scaling is specified.

Advanced joint properties

For the Ball joint and Free joint, you have possibility to alter the numerical formulation of how the rotational DOFs are represented internally. You can also control the *spring inter-connectivity*, a feature that can be used to describe the circular or cylindrical stiffness behavior of rubber bushings, etc. This is done through the Advanced tab shown below.

Summary | Origin | Tx | Ty | Tz | Rx | Ry | Rz | Advanced

Rotation formulation

1 Formulation Sequential rotation, Follower axis

Sequence ZYX

2 Rotation of +/- 90 degrees about Y-axis gives a singularity

3 Spring inter-connectivity

Translation None

Rotation None

- 1 You may change the rotational formulation of the joint. The following choices are available:

- *Sequential rotation, Follower axis* - Euler angle parametrization.
- *Sequential rotation, Orthogonal axis* - Euler angle parametrization.
- *Rotational vector* - Singularity free Rodriguez parametrization.

See the Fedem R5.0 Theory Guide, *Section 2.3 "Finite rotations"* for further details on these choices.

- 2 You may alter the update sequence of the Euler angle parameters. The default sequence is Z-Y-X.



CAUTION: The Sequential rotation formulation may lead to singularities in the rotation update computations if the joint undergoes a 90 degrees rotation about the local Y-axis. If you have such behavior in the joint, you must use the Rotational vector formulation to achieve proper results.

- 3 You may specify how the translational- and rotational joint springs should be inter-connected (Cylindrical or Spherical coordinates). This can be used to describe the cylindrical or spherical behavior of rubber bushings, or pin joints with clearances. If you want the spring characteristics in the joint variables to be interpolated resulting in a cylindrical/spherical behavior, select the proper setting from the drop down menu. Please refer to the Fedem R5.0 Theory Guide, *Section 5.1.1 "Interconnected Spring Elements"* for details on how this affects the stiffness matrix.

4.4.3 Point-to-point joints

With point-to-point joints, the motion constraints of the joint are applied between two points represented by the slave and the master triad with their corresponding FE nodes. Point-to-point joints are found on the Mechanism Creation toolbar (shown at right).



Each of the point-to-point joint types are described in the following.

Revolute joint



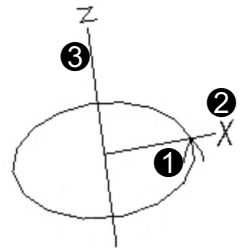
The revolute joint has a single DOF that allows rotation of one link with respect to another about a common axis. Its joint variable is the angle from the master triad to the slave triad about the common z-axis (defined by the right-hand rule).

The Revolute joint has an optional Joint variable too; namely the translation along the common z-axis. This Joint variable can be toggled on or off on the summary tab of the revolute joint property pane.

☒ Z translation DOF

The symbol for a revolute joint is displayed in the Modeler window as shown below.

- ① The arrow represents the slave triad and indicates the positive direction for the joint angle.
- ② The straight line (labeled X) represents the master triad.
- ③ The revolute axis is the common z-axis of the master and slave triads. Together with the circle it represents the joint itself.

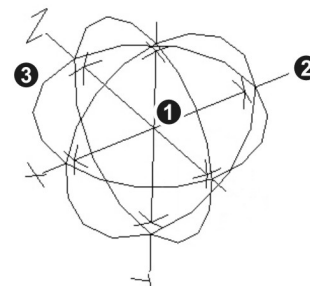


You can add friction to a revolute joint by selecting one from the list of frictions in your model in the *Friction* pull-down list, located on the Property Editor panel. (See also [Section 4.6, "Frictions"](#) and the Fedem R5.0 Theory Guide, [Section 6.5 "Joint Friction"](#).)

Ball joint



The ball joint has three DOFs that allow rotation of one link with respect to another about three axes. The joint variables are defined by the angles between the master triad and the slave triad in the x-, y-, and z-directions. The symbol for a ball joint is displayed in the Modeler window as shown to the right.



- ❶ The cross in the middle of the sphere represents the slave triad.
- ❷ The lines extending out of the sphere represent the master triad.
- ❸ The circles represent the joint itself.

You can add friction to a ball joint by selecting one from the list of frictions in your model in the *Friction*

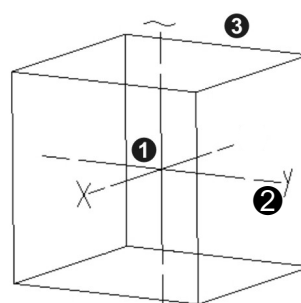


pull-down list. Then you also need to select which one of the three joint DOFs that shall receive the friction moment. The effective normal load in the friction is then computed from the other two joint DOFs that are orthogonal to the selected DOF. (See also [Section 4.6, "Frictions"](#) and the Fedem R5.0 Theory Guide, [Section 6.5 "Joint Friction"](#).)

Rigid joint



The rigid joint constrains all displacement between two links, and is therefore used as a stiff connection. It has no joint variables. The symbol for a rigid joint is displayed in the Modeler window as shown to the right.



- ❶ The cross in the middle of the cube represents the slave triad.
- ❷ The lines extending out of the cube represent the master triad.
- ❸ The cube represents the joint itself.

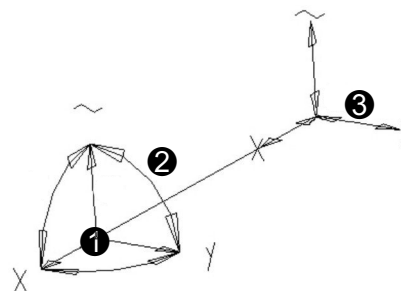
Free joint



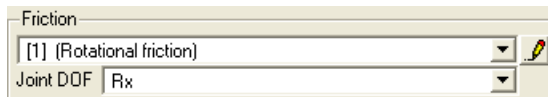
The free joint has six joint variables. The free joint can thus be used to introduce any type of mechanism motion by setting the constraint type of each joint variable to fit your needs.

The symbol for a free joint is displayed in the Modeler window as shown to the right.

- ❶ The coordinate system in the lower left (straight arrows) represents the master triad.
- ❷ The rounded arrows together with the line between the two coordinate systems represent the joint itself.
- ❸ The coordinate system in the upper right with double arrows represents the slave triad.



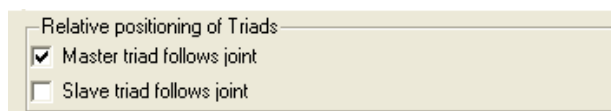
You can add friction to one of the free joint DOFs by selecting one from the list of frictions in your model in the *Friction* pull-down list. The list of selectable frictions depends on whether you have selected a translational or a rotational dof in the *Joint DOF* pull-down list.



The effective normal load in the friction is then computed from the two joint DOFs that are orthogonal to the selected DOF. (See also [Section 4.6, "Frictions"](#) and the Fedem R5.0 Theory Guide, [Section 6.5 "Joint Friction"](#).)

Moving point-to-point joints

The point-to-point joint types have three parts that either can be moved independently, or as a whole. To turn on and off this behavior a group of options are available on the joints *Origin* tab. The two toggles (shown below) control whether the slave and/or the master triad will move along with the joint symbol if the joint itself is moved. (See also [Section 3.5.4, "Origin property"](#) and [Section 3.5.2, "Align CS and rotations"](#).)



The sensitivity of the *Position* and *Orientation* fields in the Origin tab of the joint and its triads will reflect the movability of the selected object, and may change when changing these options. E.g., triads attached to FE nodes can not be moved, and thus if the triad is set to follow the joint, the joint can not be moved either.

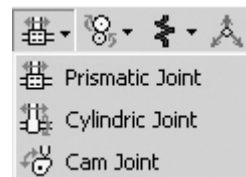


NOTE: These settings do not apply when you are using the Smart move command to move the joint (see [Section 3.5.1, "Smart Move"](#)). When applicable, the Smart move command will always move the master triad along with the joint.

The *Slave triad follows joint* toggle affects the *Position* of the slave triad only. The *Orientation* of the slave triad in a point-to-point joint will always follow that of the joint itself. The joint rotation variables are defined as the rotation between the joint coordinate system and the slave triad coordinate system and thus the triad rotation is controlled by the rotational joint variables alone. When creating a point-to-point joint, the default value of the rotational joint variables is zero.

4.4.4 Point-to-path joints

Point-to-path joints are more complex than point-to-point joints as they require more than one master triad for each slave. The motion is defined by at least two master triads in a straight or curved path. Point-to-path joints are found on the Mechanism Creation toolbar (shown at right).



NOTE: The same master triads can be used in more than one point-to-path joint.

Each of the point-to-path joint types is described in the following.

Prismatic joint



A flexible prismatic joint consists of a slave triad sliding along a straight path defined by two or more master triads. The local coordinate system of the joint is defined with its z-axis directed along the slide path. The x- and y-axes are defined from the coordinate systems of the master triads.

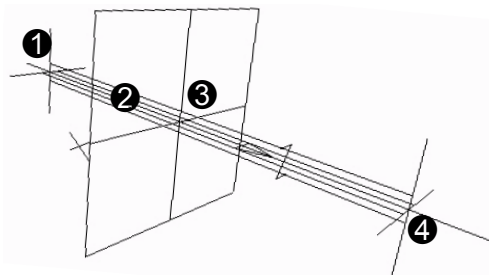
The joint has three unconstrained DOFs, but only a single joint variable (the slider variable) that allows you to control the translational displacement of the slave along the local z-axis. Rotation is constrained about the z-axis, but not in the other two directions (the slave can rotate about the local x- and y-axes independently of the masters).



TIP: You can attach two prismatic joints to make a stiff translating connection by attaching the masters for the two joints to the same nodes (in the same order) and attaching the two slave triads to the same link on different nodes.

The joint variable for prismatic joints is the distance from the first master to the slave triad in the direction of the local z-axis.

The symbol for a prismatic joint is displayed in the Modeler window as shown to the right.



- ❶ First master triad
- ❷ The slider path (represented by a line from the first to the last master)
- ❸ Slave triad
- ❹ Last master triad

Adding masters

A prismatic joint is created by selecting the position of the first and last master triad. The slider path is then defined as the straight line between the two triads. However, the slider path may be redefined by adding more master triads along that line. This improves load distribution during the simulation as the forces from the slave triad are distributed to the two masters closest to the current position of the slave.

To add master triads to a joint, click the *Add Master* button located below the Topology panel (shown at right) and select additional FE nodes along the slider path.



NOTE: You can add master to a prismatic joint only after it has been attached to a link (see [Section 3.6.1, "Attaching using Attach"](#)). It is not possible to add masters to a joint that is attached to ground.



Adding friction

You can add friction to prismatic joints by selecting one from the list of frictions in your model in the *Friction* pull-down list, located on the Property Editor panel. (See also [Section 4.6, "Frictions"](#) and the Fedem R5.0 Theory Guide, [Section 6.5 "Joint Friction"](#).)



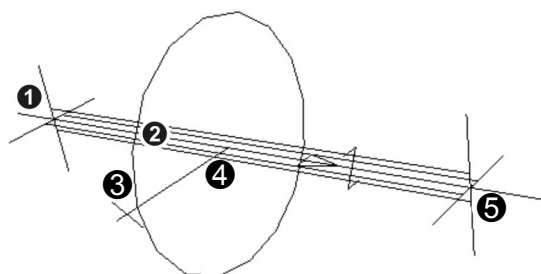
Cylindric joint

A flexible cylindric joint has four unconstrained DOFs that allow both translational displacement along the local z-axis, and rotation about the local z-axis. As with prismatic joints, cylindric joints do not constrain motion in the other two rotational directions. The joint's local coordinate system is defined in the same manner as for the prismatic joint.

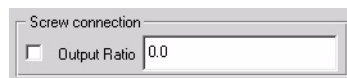
The cylindric joint has two joint variables. They are the translational distance along the local z-axis from the first master to the slave (the slider variable) and the angle of rotation of the slave about the local z-axis. The rotation angle is measured between the x-axis of the first master triad and the x-axis of the slave triad.

The symbol for a flexible cylindric joint is displayed in the Modeler as shown below.

- ❶ First master triad
- ❷ The slider path
(represented by the line from the first master to the last)
- ❸ Rotational joint variable
(represented by the angle of the x-axis)
- ❹ Slave triad
- ❺ Last master triad



You can constrain the two joint variables of a cylindric joint in a screw-like connection by defining a ratio of translational to rotational motion, called the *screw ratio*. This ratio determines how fast the slave rotates as it translates along the joint. To constrain the translational and rotational DOFs of a cylindric joint, enable the *Screw Connection* option in the Property Editor panel and assign a value to the *Screw Ratio*. (See the Fedem R5.0 Theory Guide, Section 6.4.3, "Screw joint" for more information about the screw ratio.)



TIP: You can refine the slider path by adding master triads in the same way as for prismatic joints (see "[Prismatic joint](#)" above).



TIP: A zero screw ratio makes the cylindric joint equivalent to a prismatic joint.

Cam joint



A cam joint has six unconstrained DOFs that allow the slave triad (called the *follower*) to move over a curved surface (called the *cam surface*). The cam surface is defined by a curve consisting of three-point circular arcs. Each arc is defined by the location of three master triads, also called *cam triads*. A cam joint must consist of one slave/follower triad and at least three master/cam triads. (See also the Fedem R5.0 Theory Guide, Section 6.3.3, "Cam joint.")

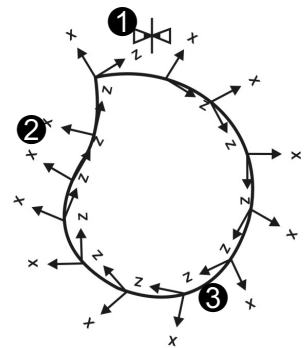
It is recommended to use at least an arc segment per quarter of a circle to make the solution more stable. That means that you will need at least 8 master triads for a complete circle.



TIP: You can use the same cam triads in several different cam joints, making it possible to constrain several follower triads to the same cam surface.

An example cam joint is shown to the right.

- ① Slave/follower triad
- ② Master/cam triads (represented by the sets of x-, y-, and z-axes extending from the curve)
- ③ Cam curve (represented by the curve)



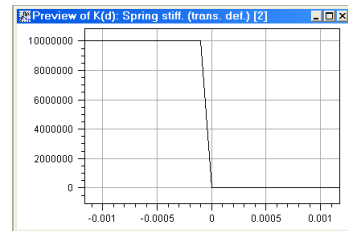
Creating cam joints

The cam curve is defined by circular arcs and straight lines. Each three-point arc is defined by three triads. If the triads are located on a straight line, a straight line will be defined (circular arc with zero curvature).

To create a cam joint, complete the following steps:

1. Click the cam joint icon.
2. Select a position for a new follower triad, or select an existing triad.
3. Confirm by pressing Done. If an existing triad was selected, this triad will become the follower triad, otherwise a new triad is created.
4. Select a position for the first master, or select an existing triad. You can also select an existing cam curve.
5. Confirm by pressing Done. If an existing triad was selected, this triad will become the first cam triad, otherwise a new triad is created. If an existing cam curve was selected, the new cam joint is complete, and the selected cam surface will be used by the new follower triad.

6. Repeat steps 4 and 5 until a sufficient number of masters has been added. As you add triads, they will be oriented automatically to have sensible orientations related to the cam curve. The z-direction is set to point along the cam curve, while the x-direction, considered to be "up", is calculated from the direction going from the master triad closest to the follower and to the follower triad.
7. To close the cam loop, add the first master triad as the last one.
8. Fedem tries to set sensible directions on the master triads created, but should any of the directions be inconvenient, rotate them using one of the tools to move mechanism elements. See [Section 3.5, "Moving mechanism elements"](#).
9. Define the spring characteristics you need for the contact behavior, and assign them to the correct joint variables. Normally, a non-linear spring with a stiffness-deflection curve as shown in the picture to the right will provide a decent contact behavior when assigned to the x-translation DOF.



Local coordinate system

The local coordinate system for a cam joint has its origin on the cam curve at a point calculated as the closest point to the follower; this point is referred to as the *contact point*. The local x-axis is then defined to be perpendicular to the cam surface and the z-axis tangential to the cam curve. The orientation of the local coordinate axes depends thus on the location of the contact point along the cam curve.

Cam joint variables

Cam joints display all the six DOFs as joint variables in the Property Editor panel, but have some restrictions on the *Constraint Type* setting that is unique for cam joints. The only legal settings are *Free* and *Spring-Damper*. The *Fixed* and *Prescribed* settings are not available because the cam joint uses a different formulation than the other joints.

The three main joint variables, defined in the x-, y- and z-directions of the cam joint's local coordinate system, are:

- **X-position:** The distance from the contact point to the follower in the direction normal to the cam surface (the "thickness" direction).



TIP: If no stiffness is assigned to the X-translation DOF, the whole cam joint will be completely ignored by the Dynamics Solver. This might be used as a simple tool to toggle a cam joint on and off during testing and modeling of complex models. The solver issues a warning when the X-translation spring is missing.

- Y-position: The distance from the contact point to the follower in the direction tangential to the cam surface and normal to the cam curve ("width" direction).
- Z-position: The distance along the cam curve from the first cam triad to the contact point (the slider variable).

You are also allowed to Spring-Damper constrain the rotational DOFs of the cam joint. Such rotational stiffness/damping might be beneficial as a stabilization tool in some cases.



WARNING! The rotational DOFs in a Cam joint are not suited for representing large rotations. However, this affects the solution only when some of these DOFs are Spring-Damper constrained. Therefore, when Spring-Damper constraining the rotational DOFs, you must ensure that the added stiffness is high enough to keep the rotations "small", typically $R_x < 0.3$, $R_y < 0.6$ and $R_z < 3.0$ radians. If not, the solution will probably diverge.

4

The initial values of the cam joint variables are interpreted differently compared with the other joint types. The *Length/Angle in model quantity* is always zero for all variables, regardless of the modelling position of the follower. For the Tx and Ty DOFs, this means that the deflection is always calculated as the distance from the contact point to the follower in the local x and y directions, respectively. However, for the Tz, Rx, Ry and Rz DOFs, the deflection is measured relative to the modelling position of the follower. The stress free length/angle of any springs associated with these DOFs are then also defined relative to these initial positions.



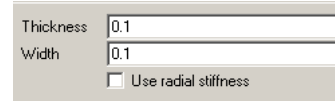
WARNING! If the follower is not within the contact domain of the cam joint (see "[Cam thickness and width as contact domain](#)" below) at the beginning of the first time step, the rotational springs as well as the slider spring, if any, are ignored throughout the simulation. This happens because the stress free length of these springs then are undefined. A warning is issued from the dynamics solver if this occurs.

Cam friction

The friction parameters for cam joints are the same as those for prismatic joints with the exception of the equivalent force, which is the sum of the x-spring and x-damper force in the cam joint. The friction state depends on the slider variable only.

Cam thickness and width as contact domain

The thickness and width parameters shown in the Property Editor panel define a rectangular domain in the xy-plane of the local coordinate system and is used to determine whether it is necessary to test if the follower is in contact or not. The springs and dampers associated with joint variables are activated only when the follower is located within the distance $Thickness/2$ from the cam curve in the local x-direction and within the distance $Width/2$ in the local y-direction. Use of a "reasonable" thickness is of great importance to ensure that the contact springs are attached to the correct cam segment (a cam segment is the part of a cam curve between two triads). One should avoid having the cam thickness so large that two cam segment have overlapping contact domains.



Thickness	0.1
Width	0.1
☐ Use radial stiffness	



CAUTION: When assigning highly non-linear spring characteristics to the cam joint variables to model contact behavior, it is often necessary to assign some associated damping to reduce fictitious oscillations due to sudden activation and deactivation of contact spring forces. A constant damping coefficient is then sufficient as long as the follower is within the contact domain throughout the simulation. However, if the follower enters the contact domain once or several times during the simulation, numerical instabilities may occur due to the sudden activation of the joint variable dampers, because they are active only when the follower is within the contact domain. To avoid this, it might be necessary to scale the damping coefficient with a function (see [Section 4.7.2, "Damper properties"](#)), that varies gradually from zero as the follower enters the contact domain, to one as the contact stiffness is activated.

Radial contact springs

By enabling the *Use radial stiffness* toggle, the springs associated with the x- and y-variables are referred to local polar coordinates in the xy-plane instead. Thus, the x-coordinate is then the radial distance from the cam curve to the follower, and the y-coordinate is the angle between the local Cartesian x-axis and the axis extending from the contact point through the follower. The contact domain will consequently be a circular cylinder instead of a rectangular one, and the *Thickness* and *Width* parameters above will now define the radial and the angular (in degrees) extension of the contact domain. This can be used to simulate contact in pipes, etc.



NOTE: The *Use radial stiffness* toggle does not affect the dampers (if any) that are assigned to the joint variables. They are still applied in the local Cartesian coordinate system. It is therefore advisable to apply the same damping characteristics to the x- and y-variables when using radial stiffness, to ensure a proper damping behavior in the cam joint.

Cam with spherical or cylindrical follower

Quite often the follower in a cam joint has some sort of spherical or cylindrical shape. This is not fully supported by Fedem, but this section describes how you can do it.

The radius of the sphere or cylinder must be entered as an *Initial stress free length* for the spring in the X-translation DOF (see [Section 4.7.1, "Spring properties"](#)). A normal contact stiffness function can then be used. The *Thickness* of the cam must also be set to a value greater than the roller radius in this case.

This will work as expected as long as the follower never is supposed to be in contact with the cam curve at more than one location simultaneously. This means that the follower can not pass the inside of a v-shaped cam curve, or curve segments that have a radius equal to, or less than the roller radius. By trying to do so, the numerical simulation will normally fail to converge when two simultaneous contact locations would be expected.

If the cam curve to be modeled has this kind of features, you will need to model the different parts of the contact curve as *separate cam joints* instead, and re-use the same triad as follower in all those cam joints. You will also have to set them up with the same contact spring characteristics and *Initial Stress free length*.



CAUTION: When using a radius on the follower, even **small** discontinuities of the cam tangent between curve segments might result in a v-shaped curve. The v's can cause numerical problems if the follower is on the inside of it.

4.5 Joint pair constraints

Joint pairs available in Fedem include both Gears and Rack-and-Pinion.

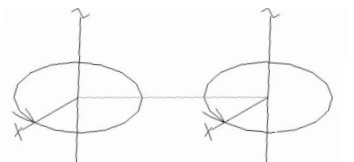
4.5.1 Gears



A gear is a rotational constraint between two revolute joints. The gear constrains the two joints to rotate at a given transmission ratio.

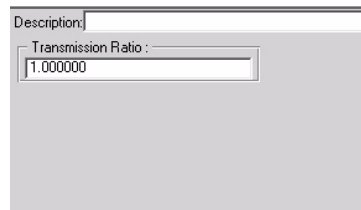
Gear symbol

The gear symbol (shown at right) is displayed in the Modeler window as a line between two revolute joints.



Transmission ratio

You can specify the gear transmission ratio (dimensionless rate) in the Property Editor panel (shown at right). For information about the gear transmission ratio, see the Fedem R5.0 Theory Guide, *Section 6.4.1, "Gear joint."*



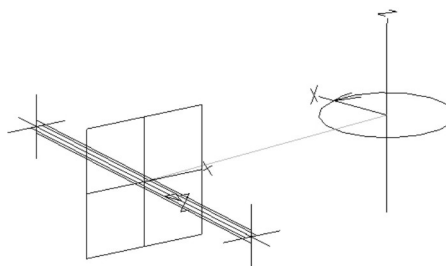
4.5.2 Rack-and-Pinion



A Rack-and-Pinion is a constraint between a prismatic and a revolute joint. The complete system is considered a five-DOF joint that constrains only a rotational input displacement to a translational output displacement. The joint also has a transmission ratio similar to that of gears.

Rack-and-Pinion symbol

The Rack-and-Pinion symbol (shown at right) is displayed in the Modeler as a line between a prismatic and a revolute joint.



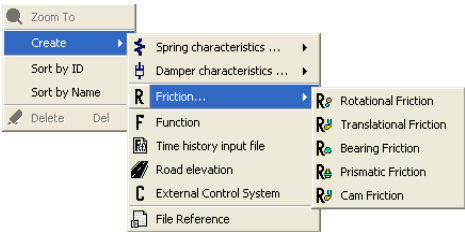
Transmission ratio

You can specify the transmission ratio (dimensionless rate) for a Rack-and-Pinion constraint in the Property Editor panel (see [Section 4.5.1, "Gears"](#)).

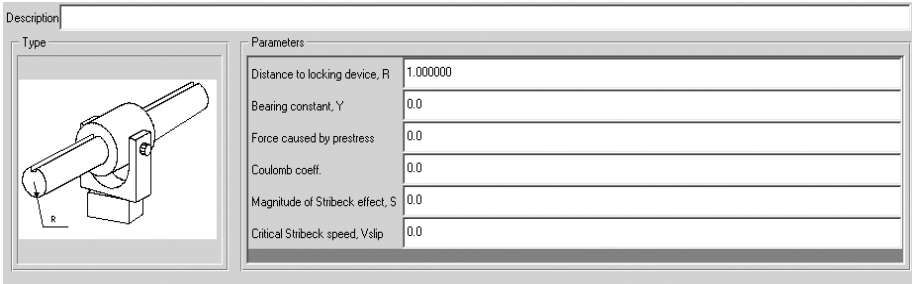
4.6 Frictions

Joint friction is based on the forces, moments, and velocity in a joint. These forces and moments give an equivalent load, which is the basis for computing the friction force. For a detailed description of friction behavior, see the Fedem R5.0 Theory Guide, *Section 6.5, "Joint friction"*.

You can create a friction by right-clicking an empty space in the Model Manager Objects list, selecting *Create, Friction* and then the desired friction type. You can also access this command from the Mechanism menu in the main window.



Frictions are managed in the Model Manager Objects list. If you have created frictions, you can expand the *Friction* group to see a list of the frictions in your model. Selecting a friction from the Objects list displays the friction properties in the Property Editor panel. Each type of friction has a different image and parameters associated with it. The figure below shows the Property Editor panel with prismatic joint friction selected.



4

To edit the friction, enter new values for each of the friction parameters listed in the Property Editor panel. See the Fedem R5.0 Theory Guide, Section 6.6, "Joint friction" for a description of friction parameters.



TIP: To associate the friction with the appropriate joint, select the joint and edit its friction properties in the Property Editor.

4.7 Springs and Dampers

There are two types of springs and dampers in Fedem: Axial- and joint springs and dampers. An axial spring or damper applies relative forces between two triads along the direction between the triads. Joint springs and dampers are integrated in the joint, and act on the joint triads along one of the joints unconstrained DOFs (see [Section 4.4.1, "Joint variables"](#)).

Both types of spring and dampers have the same options, and they will be described below. The joint springs and dampers are accessed through

the joint property panel (see [Section 4.4.2, "Joint properties"](#)) while the axial springs and dampers are separate items with 3D symbols and their own property panels.

4.7.1 Spring properties

The spring properties (shown below) consist of the following options:

Stress free length/angle control

This group of options concerns the calculation of spring deflection. The deflection is defined as positive when it is increasing the spring length.

- ❶ *Length/Angle in model* – The current distance (for translation) or angle (for rotation) measured in the model as you have made it. For joint springs, this is the measured value of the joint DOF that the spring acts on. For axial springs, it is the distance between the two triads.
- ❷ *Initial stress free length/angle, Initial deflection* - These radio buttons and fields work together allowing you to introduce prestress in the spring, by setting an initial stress free length/angle different from the *Length/Angle in model*. You can choose to enter this property either as an absolute value, using the *Initial stress free length/angle* option, or relative to the *Length/angle in model* by selecting the *Initial deflection* option.



CAUTION: If you introduce a spring prestress in this manner, it will be accounted for in the very first iteration of the dynamics simulation and thus lead to a dynamic shock effect. However, when initial *"Static equilibrium analysis"* is switched on, the prestress force is taken as a pure static load and the transient shock should be avoided.



NOTE: The initial deflection is positive when it increases the spring length/angle.

- ❸ *Stress free length/angle change* – You can select a function to change the stress free length/angle of the spring during the simulation (see [Section 4.10, "Functions"](#)). The value of the function will be used as an addition to the initial stress free length/angle defined above.



TIP: You can introduce motion into your system by using this option to change the length of a very stiff spring. However, an alternative and probably better way (for joint variables) is to use the Prescribed constraint type (see "[Joint variable properties](#)" in [Section 4.4.2](#)). In that case the stiff spring is avoided and the DOF is eliminated as an unknown from the system of equations. In most cases, this yields a more stable solution.

Spring properties

This group of options controls how the deflection is evaluated to produce the spring force or torque. The spring can be either a linear spring with a constant stiffness, or a non-linear spring with a non-linear relationship between the deflection and the force/torque or the stiffness.

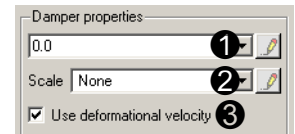
The spring force/torque is reckoned to be positive when it is working in the opposite direction of the increasing spring length/angle.

- ④ In this field you can enter a constant spring stiffness or select a defined spring characteristic from the pull-down menu.
- ⑤ *Scale* – The spring force or torque can be scaled by a function. This can for instance be used to switch the spring on and off during the simulation. When no function is selected the scale is set to 1.0.

4

4.7.2 Damper properties

This group of options controls the evaluation of a damper's force or torque from its velocity. Both linear and non-linear dampers are allowed. A linear damper uses a constant damping coefficient. A non-linear one uses a function to control how the damper force or coefficient depends on its velocity.



- ① In this field a constant damping coefficient can be entered, or you can select a damper characteristic from the pull-down menu.
- ② *Scale*– The damper force or torque can be scaled by a function. This can for instance be used to switch the damper on and off during the simulation. When no function is selected the scale is set to 1.0.



NOTE: In Fedem version 2.5m3 or lower, non-linear dampers were modeled using a function to change the damper coefficient. When opening such models in version 3.0 or higher, those dampers are converted by setting the damper functions as scale functions, and the coefficient to 1.0.

- ③ *Use deformational velocity* – This option is available only if the damper is acting together with a spring with a forced change in its stress free

length. The option enables the usage of the deformational velocity of the connected spring, when evaluating the damper. The deformational velocity is the spring velocity without the velocity component coming from a forced change in the stress free length.



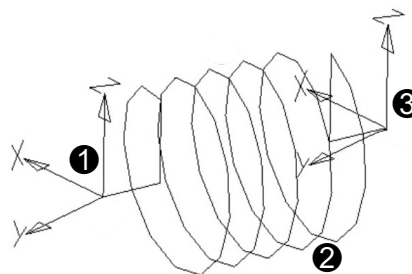
NOTE: The Use deformational velocity toggle is not visible for axial dampers unless there is a parallel axial spring connected to the same triads.

4.7.3 Axial spring symbol



The symbol for an axial spring is displayed in the Modeler as shown to the right.

- ❶ First triad
- ❷ Axial spring
- ❸ Second triad

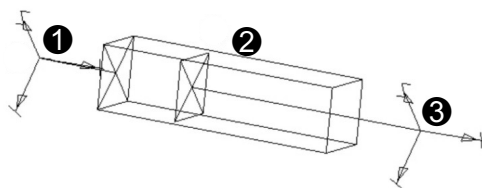


4.7.4 Axial damper symbol



The symbol for an axial damper is displayed in the Modeler as shown to the right.

- ❶ First triad
- ❷ Axial damper
- ❸ Second triad



4.7.5 Spring and damper characteristics

Non-linear springs and dampers are defined by creating a spring or damper characteristics and selecting them in the spring or damper property panel. There are four basic types of characteristics available both for springs and dampers. The differences between them are whether they define a rotational or translational spring/damper behavior, and whether they define a stiffness/damping coefficient curve or a force/torque curve. In addition, there are two types of advanced spring characteristics available (for translational and rotational springs, respectively), see [Section 4.7.6, "Advanced spring characteristics"](#).

Spring characteristics

The four basic spring characteristics types are:

- Force - Translation
- Torque - Rotation
- Stiffness - Translation
- Stiffness - Rotation

Force-Translation/Torque-Translation - These characteristics describe the relationship between displacement and spring force/torque directly. The spring stiffness is then computed as the derivative of the provided curve.

$$k(\Delta) = \frac{dF(\Delta)}{d\Delta}$$

Stiffness-Translation/Stiffness-Rotation - These characteristics describe the relationship between the displacement and the spring stiffness directly. The spring force /torque is then computed as the integral of the provided curve from 0 to the current deflection:

$$F = \int_0^{\Delta} k(x) dx$$

4

The definition of the curves used can be done using one of the following function shape types (see [Section 4.10.5, "Function Types"](#)):

- Polyline and Polyline from file
- Constant
- Linear
- Ramp
- Limited Ramp

A more detailed description of the spring characteristics can be found in the Fedem R5.0 Theory Guide, *Section 5.1, "Spring Elements."*

Damper characteristics

The four types of damper characteristics are:

- Force - Velocity
- Torque - Angular velocity
- Coefficient - Velocity
- Coefficient - Angular velocity

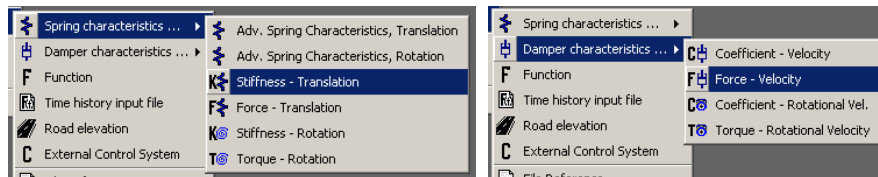
Force-Velocity/Torque Angular velocity - These characteristics describe the relationship between the damper velocity and damper force/torque directly. If a function $g(v)$ is used, the damper force is $F(v) = g(v)$ for all v . The damping coefficient is computed as the derivative $g'(v)$. A regular damper will have a $g(v)$ that is positive for positive v , and vice versa.

Coefficient-velocity/Coefficient-Angular velocity - These characteristics are interpreted as the derivative of the force/torque-velocity function with respect to v . The damper force at a specific v is thus $F(v) = \int_0^v g(w)dw$ for a given function $g(v)$, and the damping coefficient is the function value directly. A regular damper will have a coefficient-velocity function with positive values only.

The damper characteristics can be defined using the same function types as for the spring characteristics (see "[Spring characteristics](#)" above). A more detailed description of the damper characteristics can be found in the Fedem R5.0 Theory Guide, *Section 5.2, "Damping Elements."*

Creating spring and damper characteristics

To create a spring or damper characteristic, right click in the Model Manager Objects browser, and select Create -> Spring/Damper Characteristic and then the type you want from the menus shown below.:



The characteristics you have created, will be displayed in the proper pull-down menus in the spring and damper property panels. Only the ones of correct type will be listed to avoid using characteristics defined for rotation in translational DOFs, and vice versa.

4.7.6 Advanced spring characteristics

In addition to the basic spring characteristics types described above, there are also some more advanced characteristics available with further

options for defining the non-linear behavior of a spring. The Property Editor panel for the advanced spring characteristics is displayed below.

- ❶ **Spring function** - In this field you may either enter a constant spring stiffness, or select an existing basic spring characteristics function from the pull-down menu.
- ❷ **Failure criteria** - Failure of the spring can be defined through max and min. forces, and deflections. You can enable all four criteria, and whichever failure criterion is satisfied first will switch the spring (permanently) off (i.e. both spring force and stiffness vanishes).
 - *Max Deflection*: Spring is active until its deflection becomes greater than this value.
 - *Min Deflection*: Spring is active until its deflection becomes less than this value.
 - *Max Force*: Spring is active until force becomes greater than this value.
 - *Min Force*: Spring is active until force becomes less than this value.
- ❸ **Yield criterion** - Hysteretic behavior and/or permanent deflection after unloading can be introduced in springs by this options. The yield criterion will limit the force of the spring to the specified max and min. forces. When the spring force reaches any of these limits, the spring stiffness vanishes and any further deflection of the spring is defined as the *yield deflection*.
 - *Max Yield Force*: Spring force is always less than this value if this option is enabled.
 - *Min Yield Force*: Spring force is always greater than this value if this is option is enabled.
 - *Max Yield Deflection*: If the yield deflection exceeds this value (either on tension or compression) the spring is switched permanently off.



NOTE: The Max Yield Force and Min Yield Force can also be defined through functions giving the spring variable yield limits. This can be used to model "clutch-like" behaviors in a spring coupling, where you can smoothly (or abruptly) engage/disengage the motion coupling.

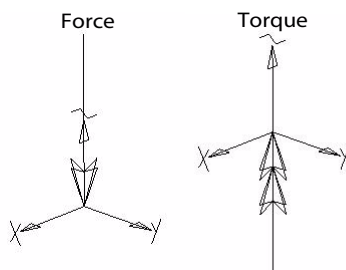
To define a non-linear spring with sudden failure and/or yield limits, you first have to define the non-linear elastic force-deflection (or stiffness-deflection) curve through the *Spring characteristics* menu (see "[Creating spring and damper characteristics](#)" above). Then you create an *Advanced Spring Characteristic* via the same menu, select the newly created *Spring Characteristic* in the Spring function pull-down, and then add the failure/yield criterions. The advanced spring characteristic is then available for selection in the "[Spring properties](#)" field of the Spring objects.

4.8 Loads



Two types of loads can be applied to triads or links: forces and torques. Both types are applied as point-force vectors on FE nodes. These loads can be used to introduce motion into your mechanism. During simulation, the magnitude of forces and torques can be constant or controlled by functions (see [Section 4.10, "Functions"](#)). When creating them, it is possible to add a load directly to an existing triad or to an FE node.

4.8.1 Load symbols



The symbols for forces and torques are displayed in the Modeler as shown to the left.

4.8.2 Load properties

The magnitude and direction of a force/torque vector can be edited in the Property Editor panel (shown below). Select the force or torque to show its properties.

- ① **Magnitude** – to change the load magnitude, enter a constant value or select one from the list of functions in your model (see [Section 4.10, "Functions"](#)).
- ② **Load Target Point** – the point is given in either global or local coordinates. You can also select a new target point using the *Mouse* button (see [Section 4.8.3, "Target point"](#) below).
- ③ **Direction** – The *From* and *To* options allow you to edit the orientation of the load vector (see [Section 4.8.4, "Direction"](#) below).
- ④ **Mouse button** – You can select a new point for the *Load Target Point*, or *From* or *To* *Directions* using the *Mouse* buttons (then select a new point in the Modeler window).
- ⑤ **View button** – You can click and hold down any of the *View* buttons in the Property Editor to highlight the corresponding point in the Modeler window.



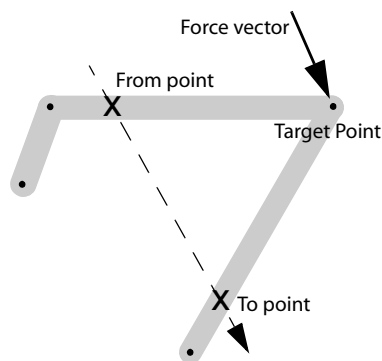
4.8.3 Target point

To specify a new target point, click the *Mouse* button and use the cursor to select a link in the Modeler. If the target point does not coincide with an FE node, the target point will snap to the closest node. Press *Done* to confirm the selection. A triad is created at that position.



4.8.4 Direction

The direction of the input load vector can be specified by two points moving together with the selected links, or by two fixed points given in global coordinates. The direction is given by the vector pointing from the *From* point to the *To* point (shown at right).



4.9 Tires and Roads

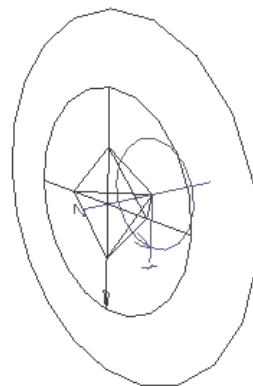
Tires and roads are used to simulate the behavior of rubber tires found on many vehicles such as cars, trucks and planes. Fedem currently interfaces tire models developed by TNO through the STI interface. For more information on the properties of these tire formulation, please refer to the [Appendix A, "Using the MF-Tyre Model"](#) and [Appendix B, "Using the SWIFT-Tyre Model"](#).

Fedem also interfaces the FTIRE tire model through the CTI interface defined by Cosin Consulting. Details on FTIRE is available on www.ftire.com.

4.9.1 Tire



To create a tire, press the tire icon, and select a *Revolute Joint* that acts as the bearing between the wheel carrier link and the spindle link. The master triad of the joint must be attached to the wheel carrier link, the slave triad must be attached to the spindle. The forces and torques from the tire will be applied to the slave triad of this joint.



To complete the tire, you have to edit some of the tire properties. The tire property panel is shown below.

- **Tire file** – Select a tire data file that describes the properties of the tire. This can be done by pressing the browse button and select a tire property file (*.tpf or *.tir) or select a file reference that refers to a tire file. Please contact Fedem Technology or your tire supplier to get tire property files for your particular tires
- **Road** – You also have to select which road you want the tire to run on. Do that by selecting the correct road in the Road drop-down menu.
- **Tire model** – The tire models currently available are:
 - MF-tyre delivered by TNO ([Appendix A, "Using the MF-Tyre Model"](#))
 - SWIFT-tyre developed by TNO ([Appendix B, "Using the SWIFT-Tyre Model"](#))
 - FTIRE developed by Cosin Consulting (www.ftire.com)
- **Z-offset** – If the wheel bearing joint is not located in the mid plane of the wheel, this can be accounted for by setting the Z-offset to the distance between the wheel mid plane and the bearing joint.

4.9.2 Road



The road surface is defined by either a *Road elevation* function, or by a road definition file. Currently, roads defined by a function can only be used by TNO tires (MF-Tyre and SWIFT), while road definition files can only be used by the FTIRE tire model.

The Property Editor panel for a road is shown below.

- ❶ Select the *Road elevation* function that describes the road surface.
- ❷ The road alignment, vertical shift and horizontal offset can be adjusted for the individual roads when defined by a Road elevation function. The road alignment angle is defined as degrees from the positive x-axis.
- ❸ Select a road definition file by choosing Browse or select a file reference.



NOTE: *There is currently no visualization of the road in the modeler.*

4.9.3

Road elevation



Road elevations are dedicated functions that are used to describe a road surface. To create a Road elevation, right click in the Model Manager Objects view, and select Create-> Road elevation. You can then edit the road elevation properties in the Property Editor panel in the same manner as for functions (see [Section 4.10.2, "Function properties"](#)).

4.10 Functions



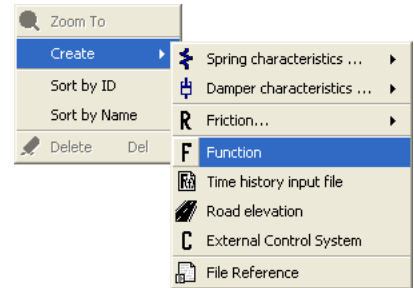
Functions can be used to control the magnitude of loads, length of springs, prescribe motion in joints etc. The function defines an input variable and a function shape that is used to transform the input value into the output value of the function, The output will thus change during the simulation depending on the variations in its input value.

The input value can be a system variable measured by a sensor, the output of a control system, the output of a different function, or simply the simulation time. The function shape can be defined in several different ways, and uses a common way of defining function shapes across different objects needing to do so.

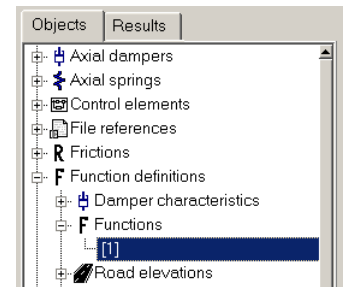
Road elevations, Control inputs, Control outputs, Spring and Damper characteristics are all examples of objects using a similar way of defining function like relationships. The description found here is thus valid for several other objects as well.

4.10.1 Creating a function

You create a function by right-clicking an empty space in the Model Manager Objects list, selecting *Create* and then *Function*. You can also access the command from the Mechanism pull down menu in the main window menu.



The new Function is automatically selected, and its properties are shown in the Property Editor panel. It will also be added to the list of Functions maintained in the Model Manager Objects list (shown at right).



4

4.10.2 Function properties

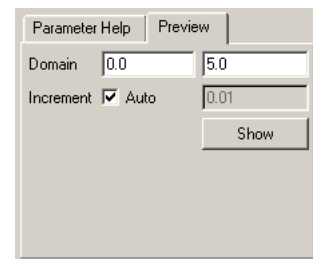
When a function is selected in the Objects list, its properties are displayed in the Property Editor panel (shown below) which is divided in three parts. The left part contains fields for defining the function type and argument and the middle part contains a list of parameters associated with the chosen function type. The right part contains two tabs; one for displaying an image explaining the function definition and another with options for previewing the function in a graph.



- ❶ *Function Type* - You can change the function type by selecting the new type from the pull-down list. The parameters and the help image shown in the panel are updated to reflect the new type.
- ❷ *Argument* - You can select any of the objects in your model that already is used as an argument by this or another function; namely those having a sensor attached, from the pull-down list. When an object has been selected, specify which quantity you want to access on that object by selecting from the *DOF* and *Var* pull-down lists.
- ❸ *Argument selection button* - By pressing this button you can select any object in your model to use as argument. When the button is pressed, the *Guide Panel* will prompt you to select an object. Do so and press *Done* to accept the selection. A sensor will then be created on the selected object which will appear in the *Argument* pull-down list. In addition to physical objects like Triads, Joints, etc., you may also select Control output elements and other Functions as arguments.
- ❹ *Parameters* - This frame contains the parameters of the selected function type.
- ❺ *Parameter Help* - This tab displays a reference picture to easier remember the meaning of the different parameters.
- ❻ *Preview* - This tab has options to control preview of the function shape (see [Section 4.10.3, "Preview"](#)).

4.10.3 Preview

To get an impression of the function shape you may preview it as a curve in a graph. Specify the argument *Domain* and *Increment* in the preview tab of the Property Editor, then push the *Show* button to plot. A preview graph containing the preview curve is then created and displayed. The displayed curve is updated automatically when changing any of the function properties.



Most functions have the option to set the preview increment automatically. This is enabled by default, but can be disabled by toggling the *Auto* toggle.

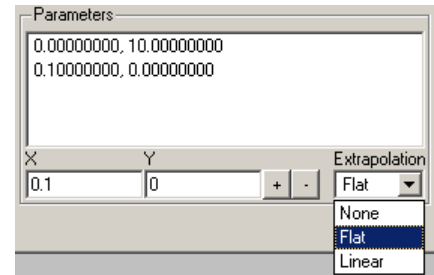
The created graph and curve is automatically added to the Model Manager Results list and may be handled like any regular graph and curve. (Refer to [Section 7.1, "Postprocessing environment"](#) and [Section 7.2, "Graphs"](#).)



TIP: The actual values of a function can also be plotted directly in a Graph when you have run a simulation or during a simulation. To enable such plotting, you must first specify the Additional Solver Option `-allEngineVars` for the Dynamics Solver (see [Section 6.2, "Additional solver options"](#)) before starting the simulation, such that the function values are saved to the results database files for the computed time steps.

4.10.4 Extrapolation

For functions defined on a user-specified finite domain, the option to extrapolate the function outside this domain exists (i.e., functions of type Polyline, Linear derivative or Spline). The default is no extrapolation (None).



If the *Extrapolation* option is set to Flat the function retains the end point values when outside its domain. That is:

- For all $v < x_1$ the function evaluates to $f(x_1)$.
- For all $v > x_n$ the function evaluates to $f(x_n)$.

If the *Extrapolation* option is set to Linear the function is continued along the tangent line of the nearest end point, that is:

- For all $v < x_1$ the function evaluates to $f(x_1) + f'(x_1) \cdot (v - x_1)$.
- For all $v > x_n$ the function evaluates to $f(x_n) + f'(x_n) \cdot (v - x_n)$.

4.10.5 Function Types

All function types available in Fedem are presented in the following.

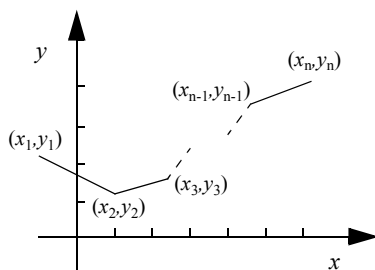
1:1

This represents the identity function. Its value equals the argument value.

Polyline

This is linear interpolation between user-specified points (x_i, y_i) .

$$f(v) = y_i + \frac{v - x_i}{x_{i+1} - x_i} (y_{i+1} - y_i), \quad x_i < v \leq x_{i+1}, \quad x_i, y_i, i \in [1, n]$$

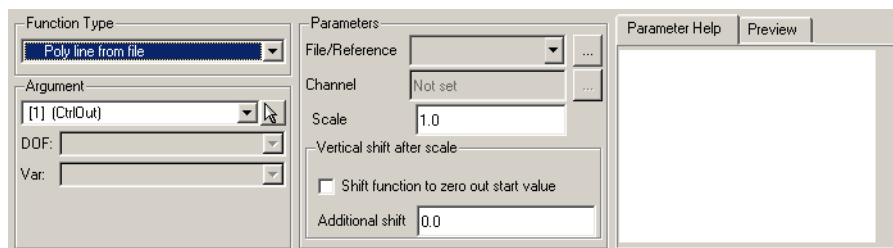


To add many numbers to a polyline function, copy and paste is feasible. Copy the numbers from the application where they are present (e.g. a spreadsheet or a text editor). On win32 based systems they have to be copied to the clip board by pressing Ctrl+C. The numbers are then inserted into the polyline function by clicking the list and pressing Ctrl+V.

Refer to [Section 4.10.4, "Extrapolation"](#) to learn about the extrapolation of a polyline function.

Polyline from file

This function is similar to the polyline function except that the points now are read from a file. The file format can either be single- or multi-channel ASCII (.asc, .txt), DAC (.dac) or RPCIII time history (.rsp, .drv, .tim). For multi-channel ASCII and RPCIII files, the channel to read is chosen by pressing the *Select* button. Additional parameters for scaling and vertical shift of the function may also be specified in the Property Editor panel of this function type (shown below).



For abscissa value v the returned ordinate $f(v)$ of the polyline from file function is

$$f(v) = \left(y_i + \frac{v - x_i}{x_{i+1} - x_i} (y_{i+1} - y_i) \right) s + k, \quad x_i < v \leq x_{i+1}, \quad x_i, y_i, i \in [1, n]$$

s - scale

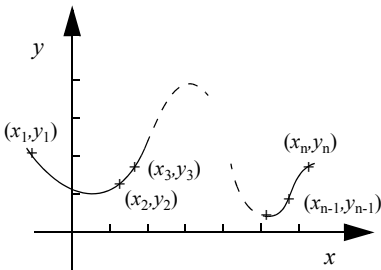
k - vertical shift

This function interpolates linearly between the user-specified points (x_i, y_i) . The interpolated ordinate value is then scaled by a factor s , and shifted by k . The vertical shift value k is set to the scalar κ entered in the "Additional shift" field. If "Shift function to zero out start value" is checked as well, the start value x_1 is also subtracted, so that $k = \kappa - x_1$.

For abscissa values outside the function domain an extrapolated ordinate is assigned, such that

$$v < x_1 \Rightarrow f(v) = f(x_1) \text{ and } v > x_n \Rightarrow f(v) = f(x_n)$$

Spline



$$x_i, y_i, i \in [1, n]$$

A third order spline approximation is calculated from a set of user-specified points (x_i, y_i) , which may be entered in the same way as for Polyline functions. At least 4 points are required.

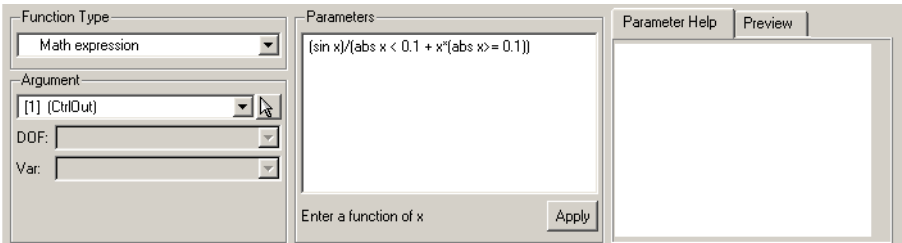
Refer to [Section 4.10.4, "Extrapolation"](#) to learn about the extrapolation of a Spline function.



TIP: If you have a "Polyline" function consisting of at least four points, you can change its type into Spline without losing the entered curve point data. This is useful just to see how the same set of points appear when they are interpolated with a cubic spline basis instead of the piece-wise linear interpolation. You can then change back to Polyline again, if you want to retain the linear interpolation in the simulation.

Math Expression

The math expression allows a user-defined function expression to be entered as free text. The function is entered in the Parameters section of the Property Editor panel as shown below. The Apply button *must* be pushed to check and register the expression entered.



Rules for Fedem math expressions:

- The expression must be mono variable, and the independent variable must be named x .
- The expression may only consist of the intrinsic functions and operators listed below, along with the independent variable x .
- Both x and other functions may be used as function input. E.g:
 - $\sin(\cos x)$ - is a valid expression.
- Precedence is set by parentheses in the usual manner. E.g:
 - $\sin(x^2)$ - x is squared before being input to the sine function.
 - $(\sin x)^2$ - the value of sine of x is squared.
- The logical operators have a return value of 0 if FALSE and 1 if TRUE, and are used by multiplying them with the function expressions. E.g:
 - $\sin x + (x > 2) * x$ equals $(\sin x)$ if $(x \leq 2)$, and $(\sin x + x)$ if $(x > 2)$.

Below are the available intrinsic functions and operators. The symbols a and b may both be numbers, functions or the independent variable x .

Function	Description
$abs\ a$	absolute value
$a+b$	addition
$a-b$	subtraction
$a*b$	multiplication
a/b	division
a^b	power
$\sqrt{a}$	square root
$n\#a$	n th root
$\sin a$	sine
$\cos a$	cosine
$\tan a$	tangent
$\arcsin a$	arcsin
$\arccos a$	arccos
$\arctan a$	arctan
$\ln a$	logarithm, base e
$\log a$	logarithm, base 10

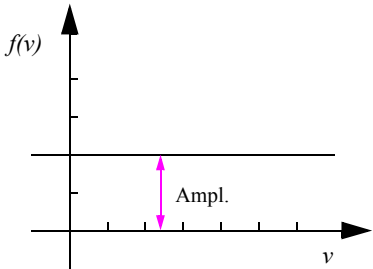
Function	Description
<i>exp a</i>	exponential, e^a
<i>aEb</i>	$a \cdot 10^b$
<i>pi</i>	constant value of π

Available logical operators:

Function	Description
<i>!a</i>	not
<i>a<b</i>	less than
<i>a>b</i>	greater than
<i>a<=b</i>	less than or equal to
<i>a>=b</i>	greater than or equal to
<i>a==b</i>	equal to
<i>a!=b</i>	not equal to
<i>a b</i>	or
<i>a&& b</i>	and

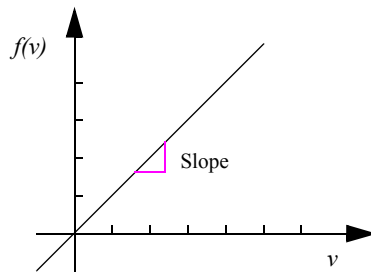
4

Constant



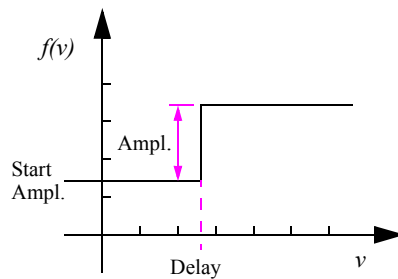
$$f(v) = C$$

C - constant (amplitude)

Linear function

$$f(v) = kv$$

k - slope

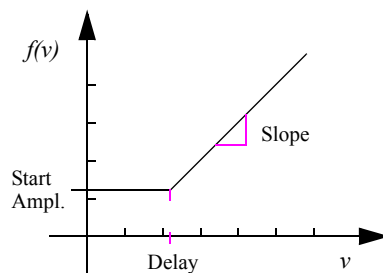
Step

$$f(v) = \begin{cases} C & , v \leq v_0 \\ C + A & , v > v_0 \end{cases}$$

C - offset (start amplitude)

A - amplitude

v_0 - step abscissa value (delay)

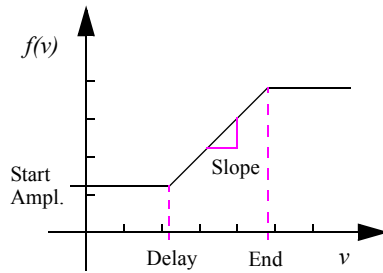
Ramp

$$f(v) = \begin{cases} C & , v \leq v_0 \\ C + k(v - v_0) & , v > v_0 \end{cases}$$

C - offset (start amplitude)

k - slope

v_0 - ramp start abscissa value(delay)

Limited ramp

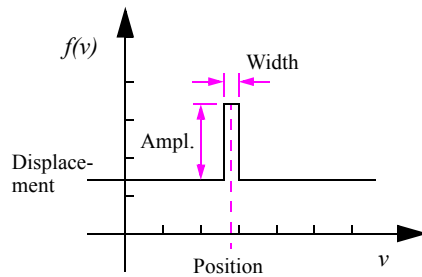
$$f(v) = \begin{cases} C & , v \leq v_0 \\ C + k(v - v_0) & , v_0 < v < v_e \\ C + k(v_e - v) & , v \geq v_e \end{cases}$$

C - offset (start amplitude)

k - slope

v_0 - ramp start abscissa value(delay)

v_e - ramp end abscissa value(end)

Pulse

$$f(v) = \begin{cases} C & , v \leq v_0 - \frac{\delta}{2} \\ C + A & , v_0 - \frac{\delta}{2} < v < v_0 + \frac{\delta}{2} \\ C & , v \geq v_0 + \frac{\delta}{2} \end{cases}$$

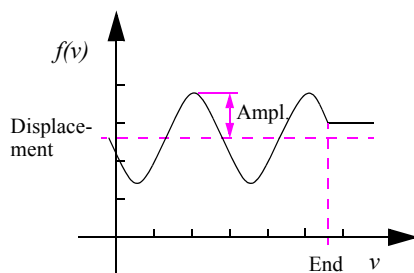
C - offset (displacement)

A - amplitude

δ - width

v_0 - pulse center abscissa value (position)

Sine



$$f(v) = \begin{cases} C + A \sin(2\pi(f_0 v - \theta)) & , v \leq v_e \\ C + A \sin(2\pi(f_0 v_e - \theta)) & , v > v_e \end{cases}$$

C - offset (displacement)

A - amplitude

f_0 - frequency (Hz)

q - phase shift (fraction of period)

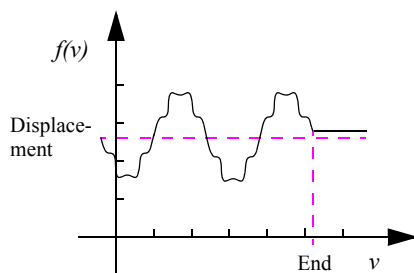
v_e - end of sinusoidal



NOTE: If the End value in a Sine function is specified less than or equal to zero, that is interpreted as infinity.

Combined sine

$$f(v) = \begin{cases} C + A_1 \sin(2\pi(f_1 v - \theta_1)) + A_2 \sin(2\pi(f_2 v - \theta_2)) & , v \leq v_e \\ C + A_1 \sin(2\pi(f_1 v_e - \theta_1)) + A_2 \sin(2\pi(f_2 v_e - \theta_2)) & , v > v_e \end{cases}$$



C - offset (displacement)

A_1, A_2 - amplitude

f_1, f_2 - frequency (Hz)

q_1, q_2 - phase shift (fraction of period)

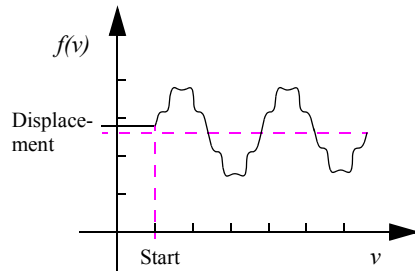
v_e - end of sinusoidal



NOTE: If the End value in a Combined sine function is specified less than or equal to zero, that is interpreted as infinity.

Delayed combined sine

$$f(v) = \begin{cases} C + A_1 \sin(2\pi(f_1 v_0 - \theta_1)) + A_2 \sin(2\pi(f_2 v_0 - \theta_2)) & , v \leq v_0 \\ C + A_1 \sin(2\pi(f_1 v - \theta_1)) + A_2 \sin(2\pi(f_2 v - \theta_2)) & , v > v_0 \end{cases}$$



C - offset (mean value)

A_1, A_2 - amplitude

f_1, f_2 - frequency (Hz)

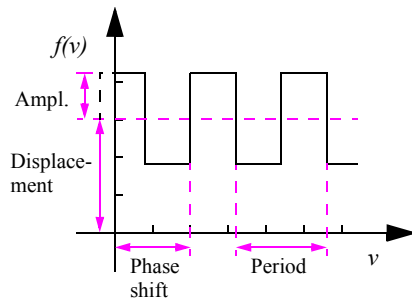
q_1, q_2 - phase shift (fraction of period)

v_0 - delay

4

Periodic square pulse

$$f(v) = \begin{cases} C + A, & \frac{n-1}{f_0} < v - \theta \leq \frac{2n-1}{2f_0} \\ C - A, & \frac{2n-1}{2f_0} < v - \theta \leq \frac{n}{f_0} \end{cases} \quad n \in 0, 1, 2, \dots$$

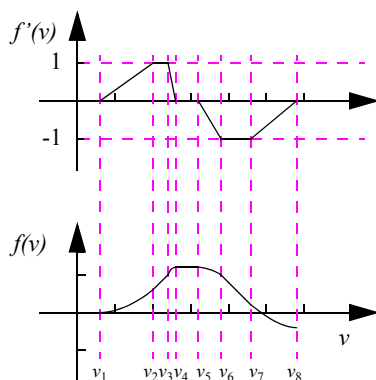


C - offset (displacement)

A - amplitude

f_0 - frequency(inHz when v is time)

θ - phase shift

Linear derivative function

$$f'(v) = \begin{cases} 0 & , 0 \leq v < v_1 \\ \frac{v - v_1}{v_2 - v_1} & , v_1 \leq v < v_2 \\ 1 & , v_2 \leq v < v_3 \\ \frac{v_4 - v}{v_4 - v_3} & , v_3 \leq v < v_4 \\ 0 & , v_4 \leq v < v_5 \\ \frac{v_5 - v}{v_6 - v_5} & , v_5 \leq v < v_6 \\ -1 & , v_6 \leq v < v_7 \\ \frac{v - v_8}{v_8 - v_7} & , v_7 \leq v < v_8 \\ 0 & , v_8 \leq v < v_9 \\ \frac{v - v_9}{v_{10} - v_9} & , v_9 \leq v < v_{10} \\ : & \end{cases}$$

$$f(v) = \int_0^v f'(\tau) d\tau$$

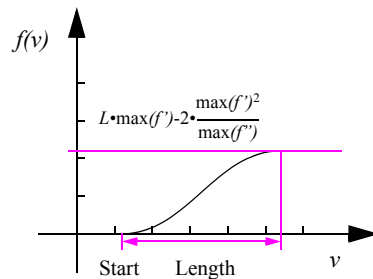
To add many numbers to a linear derivative function, copy and paste is feasible. Copy the numbers from the application where they are present. (e.g. a spreadsheet or a text editor). On win32 based systems they have to be copied to the clip board by pressing Ctrl+C. The numbers are then inserted into the polyline function by clicking the list and pressing Ctrl+V.

Refer to [Section 4.10.4, "Extrapolation"](#) to learn about the extrapolation of a Linear derivative function.

Smooth trajectory

$$f(v) = \begin{cases} 0 & , t \leq 0 \\ \frac{A_{max}}{4} \left(t^2 - \frac{1}{2\omega^2} (1 - \cos(2\omega t)) \right) & , 0 < t \leq \frac{\pi}{\omega} \\ V_{max} \left(t - \frac{\pi}{2\omega} \right) & , \frac{\pi}{\omega} < t \leq T - \frac{\pi}{\omega} \\ V_{max} \left(T - \frac{\pi}{\omega} \right) - \frac{A_{max}}{4} \left((T-t)^2 - \frac{1}{2\omega^2} (1 - \cos(2\omega(T-t))) \right) & , T - \frac{\pi}{\omega} < t \leq T \\ V_{max} \left(T - \frac{\pi}{\omega} \right) & , t > T \end{cases}$$

where $t \equiv v - v_0$ and $\left(\omega \equiv \frac{\pi A_{max}}{2 V_{max}} \right)$



v_0 - start abscissa value(delay)

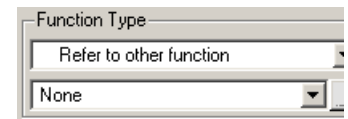
V_{max} - $\max(f')$ (maximum velocity)

A_{max} - $\max(f'')$ (maximum acceleration)

T - length of trajectory

4**Refer to other function**

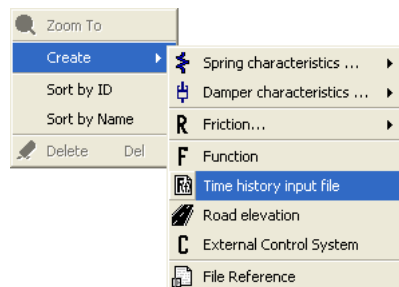
This function type reuses a previously defined function shape within this function. When this type is selected, a new drop-down menu appears from which you can select the function you want to reuse the function shape definition from. The definitions become linked, and the changes you make to the referenced function will be reflected in the referring function directly.

**4.10.6 Time history input files**

A special type of function definitions are the *Time history input file* object. This object behaves essentially as a function of time, but is optimized to be used for input of time history data from an external files.

Creating

To create a *Time history input file* object, right click in the *Model Manager Objects list*, select *Create* and *Time history input file*. A new object will then be created and its properties are displayed in the *Property panel*.



Properties

The behavior and options for this object is the same as for the *"Polyline from file"* function type, except that it is always a function of time. See [Section 4.10.5, "Function Types"](#).

4.11 Sensors

Sensors are used to measure movement and other variables associated with mechanism elements during the dynamics simulation. They work mostly as Tags to show what objects are being measured. Objects that have a sensor attached will appear in the *Argument* drop down list in *Functions* and *Control Inputs*. They are created automatically when selecting an *Argument* by using the *Select Argument* button in the *Function* and *Control Input* property panel. See [Section 4.10.2, "Function properties"](#) and [Section 5.2, "Input and output"](#). They can also be created manually using the *Simple Sensor* and *Relative Sensor* commands.

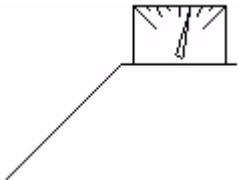
Among the quantities that can be measured are triad positions and joint variables and the associated velocity and acceleration, triad forces, spring/damper force, length and deflection/velocity, and so on. Triad rotations can also be measured in terms of Euler-Z-Y-X angles in the global coordinate system, or relative to another Triad with relative sensors. The data obtained from a sensor can be processed by a function and used in the model or a control system (see [Section 4.10, "Functions"](#) and [Chapter 5, "Control System Modeling"](#)).

There are two types of sensors, simple sensors and relative sensors:

4.11.1 Simple sensors



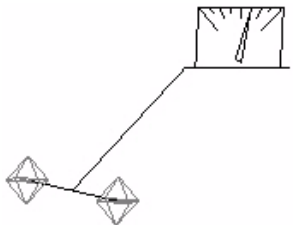
Simple sensors are used to tag a single mechanism element, such as a joint or triad, in order to extract measurements from it. When an object is tagged with a sensor, it will appear in the Argument drop-down menu in Functions and the Input menu in Control Inputs. In addition a 3D symbol will be created to show that this particular object is being measured. The symbol for a simple sensor is displayed in the Modeler window as shown to the right.



4.11.2 Relative sensors

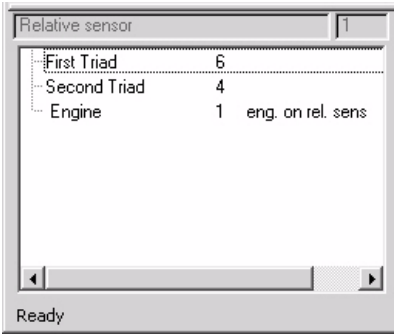


Relative sensors are used to tag two triads in order to extract relative measurements from them. The sensor will be displayed in the Argument drop-down menu in Functions and the Input menu in Control Inputs. The symbol for a relative sensor is displayed in the Modeler window as shown to the right.



4.11.3 Managing sensors

Sensors are managed in the *Model Manager Object* list. If you have created sensors, you can expand the **Sensors** group to view the list of sensors in your model. When a sensor is selected in the Objects lists, it is highlighted in the Modeler window and its description is displayed in the *Property Editor* panel. The ID and Topology Panel (shown at right) shows the triad(s) to which the sensor is attached and lists the functions or control inputs using it.



4.12 Strain rosettes

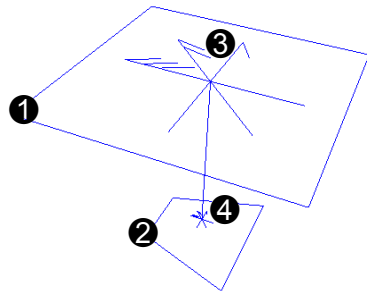


Strain rosettes are used to recover stresses and strains on a particular spot on your mechanism. The output is similar to the output from real strain gages in addition to standard strain and stress quantities like Von Mises, principal stresses/strains, and angle of max/min. principals. They can be placed on any of the FE models in the mechanism both before and after solving the dynamics. The strain rosette recovery is done by the strain rosette analysis (see [Section 6.8, "Strain rosette analysis"](#)), which recovers the strain rosettes data for one FE model at a time. See also the Fedem R5.0 Theory Guide, *Section 9.5 "Virtual strain gauges"* for the theoretical basis of strain rosettes.

Strain rosettes are defined by selecting 3 or 4 FE nodes and the direction of the first leg of the rosette. The nodes defines a strain element in which the strains and stresses is calculated at the centroid of the element based on the deformation of the nodes. Material properties and shell thickness are extracted from the underlying FE mesh, but can be overridden manually.

As the picture to the left indicates, the strain rosette symbol consists of different parts, and visualizes several aspects of the virtual strain rosette.

- ① An enlarged symbol of the strain rosette to make it easier to find.
- ② Lines connecting the FE nodes used, showing a wireframe of the strain element.
- ③ Enlarged arrows showing the directions of the particular legs within the strain rosette. Leg 1, 2 and 3 is distinguished by the number of lines used to draw the head of the arrows.
- ④ A small symbol showing the exact position of the virtual strain rosette. The position shown also includes the shell thickness or layer position, and will thus often be above the strain element wireframe.



The different features of the strain rosettes is accessed through the strain rosette property panel shown below.

The screenshot shows the strain rosette property panel with the following fields and controls:

- 1 Rosette type:** A dropdown menu currently set to "Single qage".
- 2 Nodes:** A text field containing "416, 387, 516, 494" and an "Edit" button.
- 3 Orientation:** A section containing an "Angle offset [Deg]" field set to "0.0" and an "Edit reference direction" button.
- 4 Layer position:** A section with a checked "Use thickness from FE Mesh" checkbox, a "Height" field set to "0.005", and a "Change side" button.
- 5 Set start strains to zero:** A checked checkbox.
- 6 Material:** A section with a checked "Use material from FE Mesh" checkbox, an "E-Module" field set to "2.10000000e+011", and a "Poissons ratio" field set to "0.3".

- 1 Rosette type** – Menu to select between different rosette configurations. Single Gage, Double gage 90, Triple Gage 60 and Triple Gage 45.
- 2 Nodes** – This field displays the FE node numbers used by the strain gage. The *Edit* button lets you select a different set of nodes.
- 3 Orientation** – The orientation of the strain rosette is calculated from a reference direction and an angle offset. The reference direction can be selected by pressing the *Change reference direction* button. You can select either an edge, or two points. The angle offset is entered in the field as degrees.
- 4 Layer position** – This is a feature that applies to strain rosettes on shell elements. The strains and stresses will vary through the thickness of the shell elements, with the extreme values on the top and bottom. Normally, when *Use Thickness from FE Mesh* is toggled, the strains are calculated at either the top or bottom. The *Height* field will then display the position of the strain rosette above the mid-plane of the shell element, e.g. half the thickness. The *Change side* button can be used to switch the position of the strain rosette from one side to the other. To calculate the strains and stresses at another level through the thickness of the shell element, toggle off *Use Thickness from FE Mesh* and enter the new position above the element mid-plane.
- 5 Set start strains to zero** – This toggle will zero out the strains at the first time step, eliminating strains due to gravitation or prestress effects when using the optional initial equilibrium iteration (see [Section 6.1.2, "Dynamics analysis"](#) and [Section 6.5.1, "Dynamics Solver Setup"](#)).
- 6 Material** – The stress calculations will normally use the material data from the FE mesh. If those values are inadequate for some reason, they can be overridden by toggling off the *Use material from FE Mesh* toggle, and enter proper values for E-Module and Poisson's ratio.

Chapter 5 Control System Modeling

Real mechanisms are often connected to or acted upon by control elements such as sensors, controllers, and actuators. A control system is therefore necessary to simulate these effects for Fedem mechanisms. This chapter describes the control elements available in the Fedem Control System library, and explains how to model your control systems.

In addition to the internal control system, a connection to MATLAB/Simulink models is enabled through a MATLAB-plugin module. A description is given on how to link this type of external control system to the Fedem model.

Sections in this chapter address the following topics:

- [Control modeling environment](#)
- [Input and output](#)
- [Control blocks](#)
- [Building control modules](#)
- [External control systems](#)

5.1 Control modeling environment

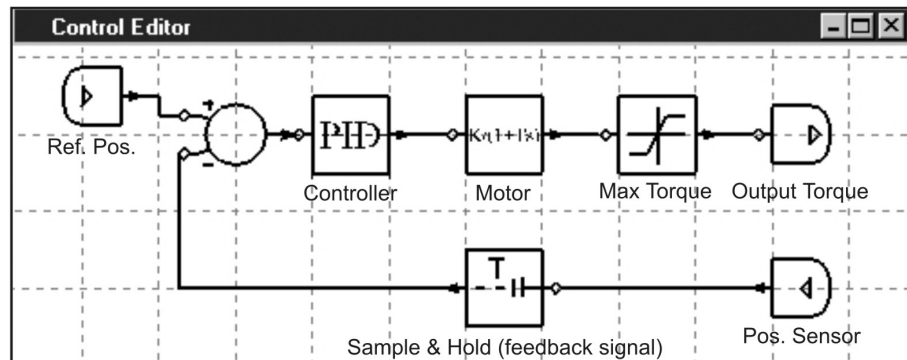
Fedem's block-diagram presentation of control systems closely resembles that given in most textbooks about basic control theory. The graphical representation consists of a series of connected control blocks, which you can model to simulate your control requirements.

5.1.1 Control Editor

To create a control system, Fedem provides the Control Editor window in which control systems can be created and manipulated. Control blocks are selected from the Control menu or the Control Creation toolbar for placement in the Control Editor window. They can then be moved and manipulated with drag-and-drop functionality. This editing environment also features grid and snap functionality (see [Section 5.4.1, "Setting Grid and Snap"](#)).



To open the Control Editor, click the *Show Control Editor* button on the Windows menu or toolbar. The Control Editor window is shown below with an example control system.



5.1.2 Control toolbars

Fedem provides two toolbars for modeling control systems:

Control Creation toolbar

The Control Creation toolbar (shown below) consists of the Fedem control blocks that are used to build control modules. (See [Section 5.3, "Control blocks"](#) for descriptions of each control block.)



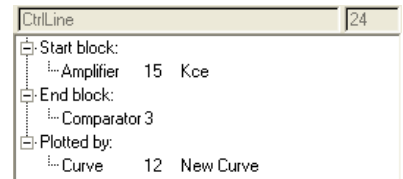
Control Tools toolbar

The Control Tools toolbar (shown below) consists of drawing and editing tools. (See [Section 5.4, "Building control modules"](#) for use of these commands.)



5.1.3 Control system topology

In a similar manner as for the structural mechanism, the topology of the current control system may be browsed in the Topology panel in the lower left corner of the Fedem main window. (See also [Section 2.5.4, "ID and Topology panel"](#).)



5.2 Input and output

The input and output blocks are the connections through which the control system and the rest of the mechanism model interact.

Control Input



The input block is used to set up input values from the mechanism, either measurements, or functions of time. The options of the Control Input element is nearly identical to the options of a *Function*. See [Section 4.10.2, "Function properties"](#). The output value of the Control Input is then connected to other control elements.

Control Output



The output block is used to make a response variable from the control system available to the mechanism. When created, it will automatically be available in the various drop-down menus that allow values from a control system. The variable can then be used to control the magnitude of loads, changes in spring length, and so on.

The Control output has the option to use an embedded function to alter the output from the control system before it is applied to the mechanism. This function is edited in the same way as *Functions*. See [Section 4.10.2, "Function properties"](#).

5.3 Control blocks

Control blocks (also called *control elements*) calculate the output as a function of one or more inputs and the block's internal state. When a control element is selected in the Control Editor, its mathematical equation and properties are shown in the Property Editor panel (see [Section 5.4.4, "Editing block properties"](#)). For more information about individual control elements, see the Fedem R5.0 Theory Guide, *Chapter 8, "Control System"*.

The following sections describe the control elements available for use in Fedem control systems.

5.3.1 Amplifiers

The control system supports the two types of amplifiers described below:



Amplifier block

The Amplifier block amplifies the input signal with a user-defined value.



Power block

The power block calculates and outputs the power (n) of the input signal (x), where n is specified by the user.

5.3.2 Binary-input blocks

Binary-input control elements have two input signals. The binary input blocks used in Fedem have no editable parameters. Each block is described below. (See also the Fedem R5.0 Theory Guide, *Section 8.4.1, "Basic elements."*)



Comparator block

The comparator block calculates and outputs the difference between the two input signals.



Adder block

The Adder block adds the two input signals and outputs the sum.



Multiplier block

The multiplier block multiplies the two input signals and outputs the product.

5.3.3 Integrator and limited derivator blocks

Fedem supports the use of the integrator and limited derivator blocks described below. (See also the Fedem R5.0 Theory Guide, *Section 8.4.1, "Basic elements"*.)



Integrator block

The Integrator block outputs the integral of the input signal.



Derivator block

The Derivator block outputs the derivative of the input signal within a specified bandwidth.

5.3.4 Time- dependent blocks

The control system supports two time-dependent control blocks described below. (See also the Fedem R5.0 Theory Guide, *Section 8.4.2, "Time-dependent elements"*.)



Delay block

The Delay block outputs a delayed input signal.



Sample-and-Hold block

The Sample-and-Hold block outputs a sampled input signal.

5.3.5 Non-continuous blocks

Non-continuous control blocks are used for simulating nonlinear behavior. They are used when the system response cannot be described by a linear model. (See also the Fedem R5.0 Theory Guide, *Section 8.4.3, "Piecewise continuous elements"*.)



Logical-Switch block

The Logical-Switch block returns one of two predefined constant inputs, both of which are dependent upon the value of a third input, called the control input. If the signal of the control input is greater than or equal to the threshold parameter, the block returns the user-specified upper limit; otherwise, it returns the user-specified lower limit.



Limiter block

The Limiter block imposes upper and lower bounds on a signal. When the input signal is within the user-specified range of the upper and lower parameters, the input signal passes through unchanged. When the input signal is outside these limits, the signal is limited to the upper or lower limit.



Dead-Zone block

The Dead-Zone block generates zero output within a specified range called its *dead zone*. The lower and upper limits of the dead zone are specified as the *Left* and *Right* parameters in the Property Editor panel. The block output depends on the input and the dead zone as follows:

- If the input is within the dead zone (greater than the lower limit and less than the upper limit), the output is zero.
- If the input is greater than or equal to the upper limit, the output is the input minus the upper limit.
- If the input is less than or equal to the lower limit, the output is the input minus the lower limit.



Hysteresis block

The Hysteresis (backlash) block controls output in such a way that a change in input causes an equal change in the output. However, changes in direction of the input signal have no effect on the output. The amount of side-to-side play in the system is referred to as the *deadband*. The deadband is centered about the output.

5.3.6 PI, PD, and PID controllers

PI, PD, and PID controllers are used to control output in such a way that the given input source forces a desired result. PI, PD, and PID controllers are described in detail in the Fedem R5.0 Theory Guide, *Section 8.4.4, "Compensator elements"*. The following are controllers used in Fedem:



- PID Controller block



- PI Controller block



- PD Controller block



- P Limited I block (limited PI controller)



- P Limited D block (limited PD controller)



- PI Limited D block (limited PID controller in serial form)



- P Limited I and D block (limited controller in serial form)

5.3.7 General-transfer functions

General transfer functions are continuous mathematical functions used to describe differential equations. The time response of the system is characterized by poles (the roots of the functions). Fedem uses the general transfer functions described in the following paragraphs. For a detailed description, see the Fedem R5.0 Theory Guide, *Section 8.4.5, "General transfer functions"*.



Real-Pole block

The Real-Pole block is a first-order transfer function. The Real-Pole block has a pole positioned on the real, negative axis in the s-plane.



Complex Conjugate Pole block

The Complex Conjugate Pole block is described by a second-order differential equation.



First-Order Transfer Function block

The First-Order Transfer Function block is described in detail in the Fedem R5.0 Theory Guide (see *"First-order element"* in *Section 8.4.5*).



Second-Order Transfer Function block

The Second-Order Transfer Function block is described in detail in the Fedem R5.0 Theory Guide (see *"Second-order element"* in *Section 8.4.5*).

5.4 Building control modules

The control module is fully defined when all blocks are connected and the inputs and outputs are attached to sensors and actuators in the mechanism model.

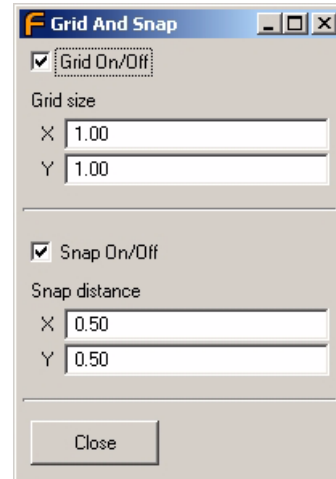
5.4.1 Setting Grid and Snap



You can manipulate the Grid and Snap settings in the Control Editor by selecting the *Control Editor Grid/Snap* button from the *Tools* menu or the *Control Tools* toolbar. The *Grid and Snap* dialog window is displayed (shown at right).



NOTE: *Grid and Snap options are applicable only for the Control Editor window.*



5.4.2 Inserting blocks

To insert control system blocks, complete the following steps:

1. Click the button that represents the block you want to insert.



TIP: *You may have to hold down the arrow found next to a block of a similar type to access the drop-down menu with other block selections.*

2. Move the cursor into the Control Editor. You will see the block following the mouse. Place the block where you want it and click the left mouse button to drop it.



NOTE: *When a block is inserted in the Control Editor window, it is automatically added to the Model Manager Objects list.*

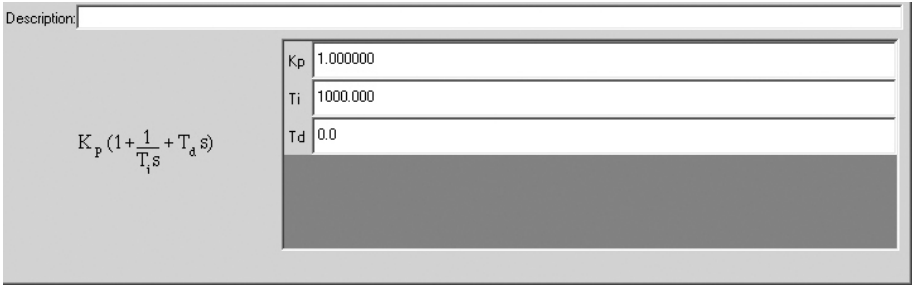
5.4.3 Moving blocks

You can adjust the block position by positioning the cursor over the block, pressing the left mouse button, and dragging the block to the new position. When the block is correctly positioned, release the mouse button.

Several blocks can also be moved together. To do that, press and hold the Ctrl-key, then select the blocks you want to move and finally drag them to the new position.

5.4.4 Editing block properties

Once a block is inserted in the Control Editor, you can edit its properties by selecting the block and editing the properties in the Property Editor window. For example, the equation and properties of a PID control block are shown below.

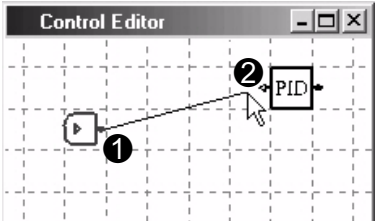


CAUTION: Changing a value in the Property Editor does not automatically apply the change. You must press the **Enter** key after editing each value to apply the change.

5.4.5 Connecting blocks

Once you have placed blocks in the Control Editor, you must draw connection lines between them to define the control module (shown at right). To connect two control blocks in your control module, complete the following steps:

- ❶ Click and drag from the arrow (output) of one control block —
- ❷ —to the circle (input) of another block and release the mouse button.



A connection line is drawn between the two blocks.

You can adjust the path of the line by dragging and dropping the line. Click and drag either a corner or a line segment between the corners.

Adding break points



This command adds lines or breaking points to connection lines. To use it, click the **Add Breakpoint** button on the Control Tools toolbar, then click and drag an existing line. A new corner is created.

Removing break points



This command removes lines or breaking points on connection lines. To use it, click the **Remove Breakpoint** button on the Control Tools toolbar, then select one of the break points on a line. Press **Done** to confirm the selection.



NOTE: A line consists of at least three perpendicular line segments. It is not possible to remove line points/segments if doing so reduces the line to less than three segments.



NOTE: Changing the path or breakpoints of a line is for display purposes only and has no influence on control system performance.

5.4.6 Rotating blocks



You can rotate control blocks 180°, so that the input connections are on the right side of the block, and the output connections are on the left side of the block. To rotate a block, click the **Flip Element Direction** button on the Control Tools toolbar and select a block. Press **Done** to confirm the selection.



NOTE: This command is for display purposes only and has no influence on control system performance.

5.4.7 Deleting blocks or connections

To delete a block or connection line in the Control Editor, complete the following steps:

1. Select the line or block in the Control Editor window.



NOTE: You can select multiple blocks at the same time by selecting them in the Model Manager Objects list.



2. Click the **Delete** button on the Standard toolbar, or the delete key on the keyboard. The selected objects are removed from the control system.



NOTE: When you delete a block, all connection lines to and from the block are also removed.

5.5 External control systems

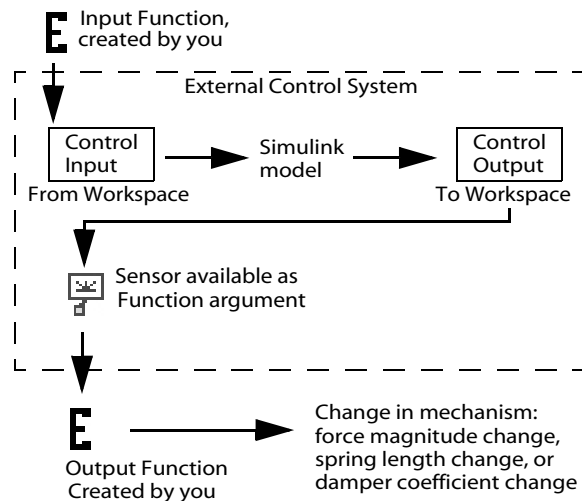
5.5.1 Requirements

The use of external control systems modeled in Simulink requires MATLAB to be installed on your computer. The external control plug-in module supports release 12.1 and 13, i.e. version 6.1 and 6.5 of MATLAB and version 4 and 5 of Simulink, respectively. If MATLAB is installed, the needed library routines will be dynamically loaded run time.

5.5.2 Data exchange

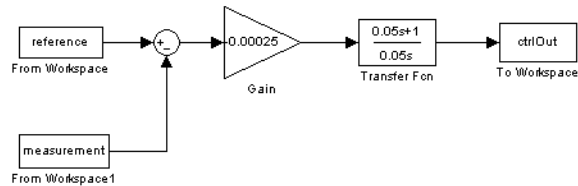
The data exchange between Fedem and the Simulink model utilizes the MATLAB workspace. A MATLAB session is therefore started when a simulation involving an external control system is initiated. The inputs and outputs of the Simulink model must be modeled as "From Workspace" and "To Workspace" blocks, respectively.

The data flow in Fedem is different from the data flow in case of internal control system (see [Section 5.2, "Input and output"](#)). Thus, mechanism variables are set up using functions, and passed to the control system by selecting the proper functions for each of the control system input values. The responses are read by sensors and through functions utilized in loads, etc. The data flow is visualized in the figure below.



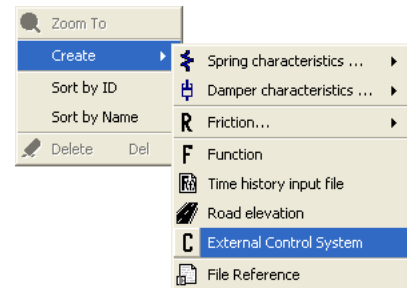
As an example, suppose a simple PI-controller has been configured in Simulink. This requires the blocks *Transfer Fcn*, *Gain* and *Sum*, as shown in

the figure below. There are two inputs - reference and measurement- and one output called ctrlOut.



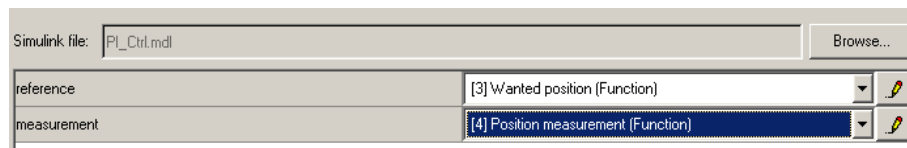
5.5.3 Connecting Simulink models

The interface to an external control system is created in a right-click pop-up menu in the Model Manager (see illustration). The empty object will then appear under the *External Control Systems* group.



In the *Property Editor* panel you may now specify the Simulink file containing the external model (see below).

The Simulink file is parsed by Fedem in order to find the input and output of the model. In the example below “reference” and “measurement” were found. The user must then specify the functions to be connected to these *From Workspace* blocks.



Detected *To Workspace* blocks are taken as outputs. An *External Ctrl Sys Sensor*, which is available for referencing by functions, is automatically created in the *Simple Sensors* group. To each output there will be a corresponding variable in this sensor. Thus, in this example, a variable *ctrlOut* will appear as the only variable of the new sensor.



NOTE: The format of the output data from the Simulink model, specified on the *Workspace I/O* tab, in the *Simulation Parameters* window, under the *Simulation* menu, must be “Array”.



NOTE: The output options, specified on the Solver tab, in the Simulation Parameters window, under the Simulation menu, should be "Refine output" with the "Refine factor" set to 1.

5.5.4 Performance

The co-simulation with an External Control System is based on the same principles as with the internal control module. Thus, the Simulink model is invoked on each iteration and re-integrated over the Fedem time step. However, the Simulink integration may take several steps for each time step in Fedem. The number of steps, together with several other choices, is set in the *Simulation parameters* window of the Simulink model view. A variable-step algorithm determines its own step length depending on specified limitations. By default the static equilibrium of the Simulink model is found and inserted upon startup.

The complete integration for each time step in Fedem, together with the data exchange via the MATLAB workspace, limits the total performance. Some improvement may be obtained by setting the Simulink time step fixed, but care should be taken not to deteriorate the accuracy.

5.5.5 Limitations

In general, the co-simulation is restricted to non-discrete blocks with scalar input and outputs. In addition, the method fails for elements depending on older values than those from the previous step. This is the case for the Transport Delay, Derivative, Backlash and Relay blocks.

Simulink elements that have been tested and found to work properly are summarized in the table below.

Group	Elements
<i>Sources</i>	From Workspace, Clock, Constant, Step, Ramp, Sine Wave, Pulse Generator, Signal Generator, Chirp Signal
<i>Sinks</i>	To Workspace
<i>Continuous</i>	Integrator, State-Space, Transfer Fcn
<i>Discontinuous</i>	Coulomb & Viscous Friction, DeadZone, Quantizer, Saturation
<i>Discrete</i>	Zero-Order Hold

Group	Elements
<i>Math Operations</i>	Abs, Gain, Math Function (exp, log, 10 ^u , log10, magnitude ² , square, sqrt, pow, reciprocal, rem, mod), MinMax (min, max), Product, Rounding Function (floor, ceil, round, fix), Sign, Sum, Trigonometric Function (sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, asinh, acosh, atanh)
<i>Signal Routing</i>	Switch

Refer to the latest Release Notes for continuous and discontinuous Simulink blocks that do not work.

Chapter 6 Mechanism Analysis

Now that you can assemble a model and implement a control system, you are ready to analyze the mechanism. This chapter describes Fedem's methods of mechanism analysis, and includes descriptions of setting up, starting, processing, and stopping each analysis.

Sections in this chapter address the following topics:

- Overview of Fedem analyses
- Solver tools
- Model reduction
- Model reduction in Nastran
- Dynamics analysis
- Stress recovery analysis
- Mode shape recovery analysis
- Strain rosette analysis
- Strain coat analysis
- Duty cycle analysis
- Interaction during processing
- Deleting results
- Automated curve export from multiple result database files
- Batch execution of solver processes
- How to read error messages from the solvers

6.1 Overview of Fedem analyses

6.1.1 Model reduction

To speed up the dynamics simulation and other mechanism analyses, Fedem first performs the model reduction process. During model reduction, individual links (FE models) in the mechanism assembly are reduced to superelements with external nodes at those points which connect links and other mechanism elements.

Fedem uses a Component Mode Synthesis (CMS) model reduction method that replaces the internal nodal DOFs with a set of static and component modes. The model reduction process for each link generates superelement mass and stiffness matrices, which are then assembled into the system mass and stiffness matrices in the dynamics simulation. See the Fedem R5.0 Theory Guide, *Chapter 3, "Model Reduction"* for more information about Fedem's model reduction process.

You can initiate the model reduction process manually at any time (see [Section 6.3, "Model reduction"](#)). Alternatively, you may also perform the model reduction in Nastran (see [Section 6.4, "Model reduction in Nastran"](#)).

6.1.2 Dynamics analysis

The dynamics analysis provides the time-history solution for all displacements, velocities, accelerations, control system variables, and derived secondary quantities (such as internal reaction forces) in the mechanical system that are driven by external forces and/or prescribed displacements.

To achieve second-order accuracy, the dynamic solution to the full system of equations is found using Newmark time-integration and Newton-Raphson equilibrium iterations at each time step. For the control system, an implicit second-order Runge-Kutta method (Lobatto IIIC) is used with Backward Euler for local error estimation. The latter is an implicit first-order method. See the Fedem R5.0 Theory Guide, *Chapter 7, "Dynamics Simulation"* for more information about the dynamics analysis.

See [Section 6.5, "Dynamics analysis"](#) to set up and perform the dynamics analysis.

Static equilibrium analysis

Before performing the dynamics simulation, Fedem may calculate an initial static equilibrium state for the mechanism model. The static

equilibrium analysis establishes a starting point for the dynamics analysis, and eliminates the initial system transients such as the sudden effect of applying gravity or other external loads. In the dynamics analysis, such unbalanced forces in the initial configuration can generate undesirable effects such as vibration of the mechanism during the first time steps. See the Fedem R5.0 Theory Guide, *Section 7.8, "Quasistatic equilibrium"* for more information about the initial static equilibrium analysis.

The static equilibrium analysis determines a state for the system in which all internal and external forces are in balance in the absence of any system motions or inertial forces. All system velocities and accelerations are set to zero during this analysis.

To set up the static equilibrium analysis, see [Section 6.5.1, "Dynamics Solver Setup"](#).

Dynamic Initial Conditions

When using the static equilibrium analysis described above, the subsequent dynamics simulation will start from a resting position with zero velocities and accelerations in all degrees of freedom. In some cases, however, it is more relevant to start the dynamics simulation from a known velocity state at all triad and joint degrees of freedom instead. Since Fedem R3.1 this is possible by entering the initial velocities directly in the model file or through certain description field commands in the object property panel, see [Section F.11, "Initial conditions for dynamics analysis"](#) for details.

If you do a static equilibrium analysis prior to the dynamics simulation, any initial conditions specified will not affect the static equilibrium analysis itself. However, the subsequent dynamics simulation will then start from a configuration that is in static equilibrium, but with a given non-zero velocity state.

Modal analysis

The dynamics analysis gives you the option of calculating the eigenmodes at different mechanism positions during the simulation. These eigenmode solutions can then be used to expand the model's mode shapes for later use in animations. See the Fedem R5.0 Theory Guide, *Section 9.6, "Eigenvalue results"* for more information about the modal analysis.

To specify the parameters for modal analysis, see [Section 6.5.1, "Dynamics Solver Setup"](#).

6.1.3 Stress recovery

After performing the dynamics analysis, a stress analysis can be conducted on the mechanism. The stresses, strains and elastic displacements can then be calculated at different time steps and/or mechanism positions of the dynamics analysis. See the Fedem R5.0 Theory Guide, *Section 9.4, "Finite element stress analysis"* for more information about the stress analysis. See [Section 6.6, "Stress recovery analysis"](#) to setup and perform the stress analysis.

6.1.4 Mode shape recovery

A mode shape recovery analysis enables you to expand the system mode shapes calculated during the dynamics simulation. These mode shapes can later be animated. See [Section 6.7, "Mode shape recovery analysis"](#) to setup and perform the mode shape analysis.

6.1.5 Strain rosette recovery

Fedem enables you to recover strains and stresses from virtual strain gages which are defined in a separate file. The results are similar to those from real strain gages. See the Fedem R5.0 Theory Guide, *Section 9.5, "Virtual strain gauges"* for more information about the strain rosette analysis. See [Section 6.8, "Strain rosette analysis"](#) to setup and perform the strain rosette analysis.

6.1.6 Strain coat recovery

The strain coat recovery process calculates the stresses and strains on the strain coat elements in the model, in a similar manner as in the strain rosette analysis. This is primarily used as input for subsequent fatigue calculations. See [Section 6.9, "Strain coat analysis"](#) to setup and perform the strain coat analysis.

6.1.7 Duty cycle analysis

Running a duty cycle analysis means summing up the damage done to any set of links from several load events, or load cases. The results obtained are summaries of the damage from the combination of all the events. See [Section 6.10, "Duty cycle analysis"](#) to setup and perform the duty cycle analysis.

6.2 Solver tools

Several dialogs and tools can be utilized to control the calculation processes in Fedem. In this section we describe those applying to all the solver processes.

6.2.1 Solvers Toolbar

The Solvers toolbar (shown below) contains the commands to set up and start each of Fedem's mechanism analyses, including the pre- and postprocessing of individual mechanism links. The toolbar is organized from left to right in the order of logical task performance, i.e. model reduction is performed first, then the dynamics simulation, stress recovery, and so on.



The setup commands (Dynamics Solver Setup, Stress Recovery Setup, Mode Shape Recovery Setup, Strain Rosette Recovery Setup, Strain Coat Recovery Setup, Duty Cycle Setup and Additional Solver Options) enable management of all analysis options.

Each of the Solver commands are described in the following sections of this chapter.



TIP: To access all commands on the Solvers toolbar, click and hold down those buttons with an arrow (▼) next to the icon.



NOTE: You can also access all Solver tools from the Solve menu.



Tip: Once you have set up the parameters for each of the solvers (as described later in this chapter), you can click the **Solve All** button on the Solvers toolbar (or Solve menu) to execute all analyses in consecutive order.

6.2.2 Controlling placement of temporary files

Some of the Fedem solvers use temporary files during computations, which are automatically deleted upon completion of the process.

On UNIX systems, these files are placed in the directory pointed to by the environment variable TMPDIR, if set. If TMPDIR is not set or points to a non-existing directory, they are placed in the directory /var/tmp instead. On Windows, they are placed in the directory pointed to by the environment variable TMP, if set. If TMP is not set or points to a non-existing directory, they are placed in the directory C:\ instead.



CAUTION: Some of the temporary files may become very large for big models. Make sure that the TMP or TMPDIR variable points to a directory with sufficient amount of free disk space.

6.2.3 Additional solver options

In the Additional Solver Options dialog, advanced users can fine tune the behavior of the solver modules through options that are not available elsewhere from the respective setup dialogs or the various Property Editor panels.



Select **Additional Solver Options...** from the Solve menu to open this dialog (shown below).

- ❶ In each of the first six fields, you may enter text strings of command-line options for the respective solver module. Refer to [Appendix E, "Command line options"](#) for a complete list of options for all Fedem modules.
- ❷ You can set the maximum number of concurrent processes that will be run during a simulation task. This is useful if you have a multi-processor machine and want to run several link reductions or recovery processes in parallel.
- ❸ You can fine-tune the memory usage of the Link Reducer through these options. See ["Optimizing the Link Reducer memory usage"](#) for further details.
- ❹ You can specify a command prefix to be applied on all solve tasks (except for FE-Fatigue and Duty cycle analysis). This can be used to launch the simulation tasks on another computer in your local network, than the Fedem UI is executed on. See ["Running solver processes on a remote computer"](#) below for further details.



CAUTION: Many of the solver options listed in [Appendix E, "Command line options"](#) may already have been provided to the corresponding solver through their respective Solver Setup dialogs. Specifying any such option also in the Additional Solver Options dialog will then override the setting in the Setup dialog and should therefore be avoided. Consult Fedem technical support if you are in doubt on the usage of a particular command-line option.

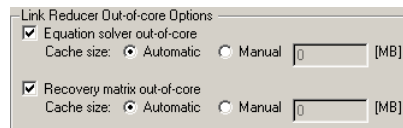


NOTE: If you mis-spell a command-line option in the Additional Solver Options dialog, or specify options that do not exist, the solver process will run as if the invalid options were not specified. A warning for each unrecognized option is issued in the ["Output List"](#) in that case, after the solver process has terminated.

Optimizing the Link Reducer memory usage

Perhaps the most memory critical solve process in Fedem is the Link Reduction for large models. On 32-bit platforms, the amount of in-core memory that one process may address is 2 GB (usually the practical limit is lower due to other processes sharing the same CPU). In Fedem, a linear equation solver is used in the Link Reducer which works out-of-core when necessary. This makes it possible to solve much larger FE models on a 32-bit platform than would be possible using an in-core solver. The equation solver reserves an in-core buffer (cache) for the numerical data of a certain size, and goes out-of-core only when this buffer is not large enough. The performance of the equation solver depends on the size of this buffer and it may therefore be optimized by fine-tuning this size.

The default is to let the Link Reducer automatically set the size of the in-core buffer (as shown to the right). It then reserves a fixed percentage of the free memory currently available (the actual size is written to the `.res` file). This size may be overridden by switching to *Manual* and entering the desired cache size in the corresponding field.



You may also switch off the out-of-core feature completely by toggling off the *Equation solver out-of-core* option. The Link Reducer will then abort if the problem does not fit in core.

There is also a similar set of memory usage options that affects the displacement recovery matrix, and they work in a similar way. Using the *Manual* setting here corresponds to the `-Bramsize` command-line option.

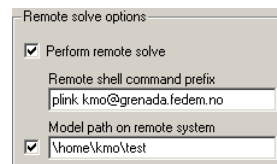


NOTE: You may switch back to the old in-core equation solver used in previous Fedem versions by specifying the command-line option `-sprSolver` to the Link Reducer. The out-of-core equation solver settings discussed above will then be ignored, but the options for the recovery matrix still apply.

Running solver processes on a remote computer

If you are using a workstation connected to a file server in a local network together with other computers, it may be advantageous to perform the simulation tasks on one of the other computers, such that the local workstation can allocate its resources fully to the Fedem UI process.

To enable such remote solving, use the *Perform remote solve* toggle in the *Remote solve options* part of the Additional Solver Options dialog (shown to the right). Enter the appropriate *Remote shell command prefix* needed to run the solver command on the remote computer, and optionally the *Model path on remote system*. The latter is necessary when, e.g., your local workstation is a Windows PC and the remote computer is a UNIX machine. You will then need to specify the path to the current model file, as it looks from the UNIX computer.



CAUTION: When specifying a remote shell command prefix, the input files needed by the solver tasks are still created locally within the Fedem UI and not explicitly copied on to the remote computer. Thus, a remote execution will work only if the local and remote computers use a shared file system. Similarly, it is assumed that the Fedem UI on the local computer can access the output files created by the solver task directly from where they are written by the remote computer.



NOTE: A server program accepting remote shell commands must be running on the specified remote computer for this feature to work (e.g., *sshd* or *rshd*). If using *ssh* (or the equivalent windows client program *plink*), you will also need to define proper identification keys in your login directory such that you are able to login to the remote computer without being prompted for a password. Please ask your system operator for assistance on such issues. For further information on the *ssh* and *PuTTY/plink* programs, consult, e.g., <http://www.openssh.com> and <http://www.chiark.greenend.org.uk/~sgtatham/putty>.

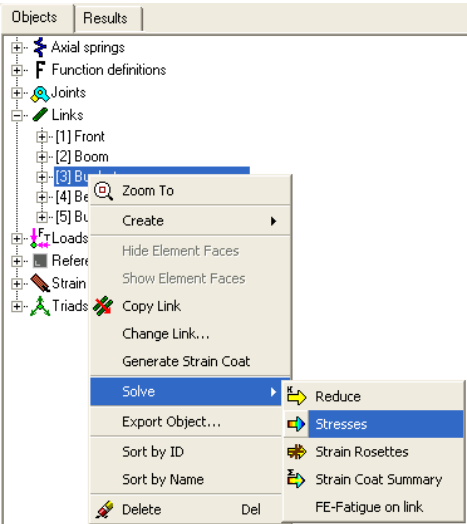
6.2.4 Link- and group-wise solving

Fedem has the capability of running some of its solvers on individual parts of the mechanism (links and element groups). This is beneficial when dealing with big models where the solver execution may be both time- and disc space consuming.

Note that these solver tasks act upon each part independently, and do not affect the global response in any way. The user is encouraged to identify critical parts of the model and recover just the results he wants on these parts. Be aware that each time you recover results they are added to the result database, regardless of their previous existence.

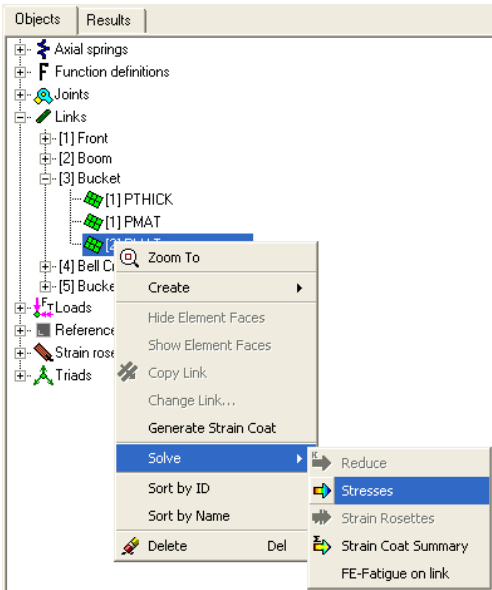
Link-wise solving

Solver processes that may be run on links individually are: Reducer, Stress recovery, Strain Rosette recovery, Strain Coat recovery and FE-Fatigue. A link-wise solve process is started by right-clicking on a link in the Model Managers Objects list, choosing *Solve*, and then the wanted process (see illustration to the right). Note that you may multi-select links in the Objects list to solve for two or more links simultaneously. As always, when trying to run the Reducer, a link will only be reduced if necessary.



Group-wise solving

Solver processes that may be run on individual element groups are: Stress recovery, Strain Coat recovery and FE-Fatigue. A group-wise solve process is started in a manner similar to that of a link-wise process (see illustration to the right). Right-click the group for which you want to solve and then choose the wanted process. Again, multi-selection is possible to solve for two or more groups simultaneously. For further information on element groups, refer to [Section 4.2, "Element groups"](#).



NOTE: When running Strain Coat recovery on individual links or element groups, a set of strain coat elements are automatically created on all shell and solid finite elements in the current selection, unless such elements have been created in a previous run (see also [Section 6.9.5, "Strain coat recovery on element groups or individual links"](#)).



NOTE: Running FE-Fatigue on element groups has the same effect as running FE-Fatigue on the link(s) the groups belong to.



NOTE: When solving on individual links or element groups, the current settings in the corresponding solver setup dialog are used.

6.2.5 Viewing the progress of long-duration analyses

When solving a large model that takes a considerable amount of time, it is often informative to know exactly how far in the simulation process we have reached at any time. This can be done efficiently by viewing a res-file that is continuously being written by the running solver in the Result File Browser (see [Section 8.2, "Result File Browser"](#)). When such a res-file is selected, the Info view is automatically scrolled to the bottom of the file and then continuously updated as the file is being written by the solver.



TIP: For the model reduction and all of the recovery modules, a dedicated progress file called `progress_info.res` is created. This file is updated more frequently than the conventional res-file associated with the process. Viewing this file while the solver is running will therefore give you the best update on its progress.

It is often wise to also keep an eye on the *Output List* view while a solver process is running, as important messages produced by the solver (errors, warnings and notes) are written here while the process is running (see also [Section 6.14.2, "How to read error messages from the solvers"](#)). All such messages are also written to the res-file, but in the *Output List* they are prefixed by the solver name and the process ID in brackets, e.g.

```
fedem_solver [2716]: Started.
fedem_solver [2716]: Warning : Indication of a poor centripetal force on the rotating Link 3
fedem_solver [2716]: Try distributing the triads better or use centripetal moment correction.
fedem_solver [2716]: Finished. Wall time elapsed: 00:04:07
```

The first message of a solver process is always *Started* and the last message is *Finished* followed by the consumed wall time. The process ID number is used to distinguish messages from possibly multiple simultaneously running reduction or recovery processes.

6.3 Model reduction

The model reduction process requires no user setup apart from the settings found in the *"Reduction Options tab"* of the Property Editor panel for each link (see [Section 4.1.4, "Link properties"](#)). Some of these options are discussed in further detail in the sub-sections below.

6.3.1 Starting the model reduction

Model reduction is performed automatically for the links needing it when you start the dynamics simulation. Fedem determines automatically which links need to be reduced based on the triad configuration and the connection to the rest of the model. It also checks whether any of the settings in the *"Reduction Options tab"* that affects the results have been changed since the last reduction of that link.



You can also initiate the model reduction process manually at any time. To do this, simply click the **Reduce All Links** button on the Solvers toolbar (or Solve menu). You may also initiate the model reduction for only one, or a selection of links, by using the *"Link-wise solving"* command (see [Section 6.2.4, "Link- and group-wise solving"](#)).



NOTE: If element calculations fail during model reduction due to bad element shapes, etc., all bad elements in the link are reported before the model reduction process exits.



NOTE: Once a link is reduced, that link will not be reduced again, even if the Reduce All Links button is clicked again, unless the link has been modified in the meantime (by for instance, adding or removing external nodes or altering the material properties).

6.3.2 Using component modes

Fedem uses a Component Mode Synthesis (CMS) model reduction method that replaces the internal nodal DOFs with a set of static and component modes. The static modes corresponds to the external nodal DOFs (i.e. at the Triads attached to the link), whereas the component modes are calculated as the eigenmode shapes of the link with all external nodes fully constrained.

Component modes describe the internal vibrations in the link. You should normally include a sufficient number of modes such that the frequencies within the time step size used in the dynamics simulation are covered. The frequencies of the computed component modes are found in the output file *fedem_reducer.res* which can be viewed using the *Result File Browser* (see [Section 8.2, "Result File Browser"](#)).

The number of component modes is specified in the Property Editor panel for the selected link (see *"Reduction Options tab"* in [Section 4.1.4](#)). The default is 0 (no component modes), i.e., only static modes are used. If the internal vibrations are not important for the overall response, you can save computation time by using static modes only. More details on component modes and CMS model reduction can be found in the Fedem R5.0 Theory Guide, Section 3.2, *"Component mode synthesis reduction"*.

6.3.3 Using lumped mass matrix

To increase the computational efficiency of the model reduction process, a lumped mass matrix approach is used by default. The FE mass matrices are then represented by one diagonal matrix each, where each diagonal term is the sum of the associated row (or column) of the corresponding consistent mass matrix. The assembled mass matrix of the link will then also be diagonal, except for some off-diagonal terms created by the constraint equations¹, if any. Thus, using lumped mass will reduce the memory requirements significantly compared to using a consistent mass matrix, and will also speed up the reduction process as some of the matrix operations involved are simplified (see equation (3.16)-(3.19) in the Fedem R5.0 Theory Guide).



NOTE: To some degree, using lumped mass does affect the results, as it represents a simplification of the FE model. The difference in the results may be significant for links with a coarse FE mesh. However, as the FE mesh is refined, the difference becomes smaller. Therefore, it is recommended to use consistent mass on links not having a sufficiently fine FE mesh. If you are in doubt, try a test run using both consistent and lumped mass and compare the results.

The use of consistent mass matrix is toggled on/off for the selected link through a button in the Property Editor panel (see "[Reduction Options tab](#)" in [Section 4.1.4](#)). The default is *off* for new links imported into a model.

6.3.4 Handling singularities during the model reduction

In complex FE modelling, it is inevitable not to have one or more defects in the FE model from time to time. In the model reduction process, this will typically result in a singular mass and/or stiffness matrix. Fedem recognizes two types of singularities, which are handled slightly different in the matrix triangularization (factorization):

- DOFs that have not received any stiffness/mass contribution at all and thus have an exact zero pivot before the triangularization is started, are detected *a priori*. The zero pivot is then replaced by the value 1.0, which implies that the singular DOF is constrained to zero.
- DOFs that initially have a non-zero pivot, but are reduced to a value close to zero during the triangularization, will receive some small

1. You have constraint equations in the link if it contains RBAR, RGD, WAVGM (RBAR, RBE2, RBE3 in Nastran terminology) and/or BEAM elements with end release. Thus, as the number of such elements in the link increases, the advantage of using lumped mass will decrease.

value on the diagonal, allowing the triangularization to continue for the detection of other potential singularities. The judgement on whether such singularities have occurred is based on the user-provided "*Singularity tolerance*" (see below).

When the reduction process has terminated, a list of all singularities is written to the Output List and the `.res` file. Each singular DOF is here identified by the node ID and local DOF number. If only singularities of the first kind are detected, the reduction process completes successfully and it should be safe to use this link in a dynamics simulation. However, if one or more singularities of the second kind are detected, the reduction is aborted with no results, and the FE model has to be manually fixed before a new attempt is made.



CAUTION: *If more than one singularity of the second kind are detected, there is a possibility that some of them, except the first one, are fictitious. On the other hand, there is also a possibility that not all singularities are detected. Thus, after having fixed all the reported singularities manually, other singularities may be revealed in the next run. This behavior is due to the insertion of a small value on the diagonal which actually changes the mechanical property of the FE model.*

It is possible to switch off the above treatment of singularities during model reduction. This is done by specifying `-singularityHandler 0` as an additional option to the Link Reducer (see [Section 6.2.3, "Additional solver options"](#)). Only the first occurring singularity, regardless of its kind, will then be printed to the Output List and `.res` file, and then the reduction process will abort. This corresponds to the behavior in Fedem R3.1 and earlier.

Singularity tolerance

The criterion used to determine if a stiffness or mass matrix is singular during the reduction is specified through a threshold value in the Property Editor panel for the link (see ["Reduction Options tab"](#) in [Section 4.1.4](#)). The default value is 1.0e-12. If the ratio between the current and previous value of a diagonal matrix element becomes less than the threshold value during the matrix triangularization, the matrix is assumed to be singular.

The default threshold value is usually sufficient for well-conditioned FE models. However, there might be situations where you have a natural high ratio between the stiffness (or mass) properties in different parts of a model. In this case you may need to reduce the value to avoid a false error exit. If you do get a singularity exit but is quite sure your model is sane (although not that well-conditioned), you should check the actual

diagonal decay value which is printed in the `.res` file. You may then try changing the *Singularity criterion* to a value less than this value and rerun the model reduction. Note that the lowest admissible value is 1.0e-20. Any value lower than that will be ignored and 1.0e-20 will be used.

Negative pivots

A model reduction may also reveal negative pivot elements in the stiffness or mass matrix. This may happen if, e.g., the link contains several poorly shaped elements (especially shell elements). However, since the linear equation solver is able to handle negative definite matrices, the reduction is not aborted when only negative pivots are encountered. Nevertheless, it is good practice to go over the FE model once again and check for bad elements if you get warnings on negative pivots.



WARNING! Using a reduced link with many negative pivots in its stiffness matrix may lead to instabilities in the subsequent Dynamics simulation, and should be avoided. Only a few negative pivots in a large link is normally not a problem, however. That has at most only local influence on the results in the vicinity of the elements with the negative pivots.

6.3.5 Eigenvalue analysis of the reduced links

To assess the dynamic properties of the reduced link matrices, you can perform an eigenvalue analysis of the reduced system, i.e.

$$(\mathbf{K} - \omega^2 \mathbf{M})\phi = 0$$

where $\mathbf{K}$ and $\mathbf{M}$ denote the reduced stiffness and mass matrices from equation (3.15) in the Fedem R5.0 Theory Guide. Since the reduced system does not include any constrained DOFs¹, the first six eigenvalues should always be zero. The remaining eigenvalues can then be used to assess the dynamic properties of the reduced link.

The n lowest eigenfrequencies, i.e. the quantities

$$\frac{\omega_i}{2\pi}, i = 1 \dots n$$

for each reduced link can be written to the corresponding `.res` file by specifying `-nevred n` in the Link Reducer field in the Additional Solver Options window (see [Section 6.2.3, "Additional solver options"](#)). The

1. DOFs that should be constrained are retained during the link reduction and are constrained only during the system analysis by the dynamics solver.

eigenfrequencies are printed in Hertz. The default value of this option is 12 for all links. However, the option is not effective for massless links. For other links, the option may be turned off by specifying `-nevred 0`.

6.3.6 Visualization of eigenmode shapes from the model reduction

To further verify the results of the reduction process and to increase the understanding of the link's dynamics properties, it is often useful to visualize the computed mode shapes of the link. This is possible if you toggle on *Expand mode shapes* in the ["Reduction Options tab"](#) of the Property Editor panel for the links in question, before they are reduced (see [Section 4.1.4, "Link properties"](#)). Both the component mode shapes (see [Section 6.3.2, "Using component modes"](#)) and the mode shapes associated with the eigenvalues of the reduced system (see [Section 6.3.5, "Eigenvalue analysis of the reduced links"](#)) will then be computed during the model reduction, and be subsequently available for viewing in an Eigenmode animation (see ["Eigen Modes tab"](#) in [Section 7.3.2](#)).



TIP: It is particularly useful to study the mode shapes of the reduced system, if you for instance get more than six modes with (close to) zero eigenfrequency. That is usually due to an internal mechanism in the link caused by an error in the FE-mesh and can be revealed when animating the corresponding mode shape.



TIP: If you get less than the required six zero rigid mode modes in the reduced system for a link, that can be caused by some over constraining in the FE model (due to bad modeling, etc.). By animating the six first mode shapes you can then see which rigid body modes are present and which are not, and also see the mode shapes that replaced the missing rigid body modes. All this might then give a hint towards the actual model error or weakness.

6.3.7 Reduction of applied load vectors

Along with the reduced mass- and stiffness matrices, the Fedem link reducer always computes reduced (unit) gravity load vectors of the link, which are applied as a constant static load in the direction of the defined gravitation vector in the dynamics simulation.

In addition, a set of load vectors corresponding to defined load cases in the FE data file are computed, when such load definitions are present. This includes both concentrated point loads and distributed surface loads on shell and solid elements.

These reduced load vectors may then be assigned time history scaling functions in the link property panel before the dynamics simulation is run (see ["Reduced Loads tab"](#) in [Section 4.1.4](#)).

6.4 Model reduction in Nastran

As an alternative to the Fedem Model Reduction, the model reduction may also be performed in Nastran. Nastran supports a similar CMS reduction procedure as in Fedem, and is able to export the reduced mass- and stiffness matrix as binary files that may be imported into Fedem.

The advantage of doing the model reduction in Nastran is that you then have a broader range of element and material properties available for use on the link level, such as orthotropic materials, composites, etc.



NOTE: When the model reduction is performed in Nastran, you also have to perform link-level recovery in Nastran. You can not perform Stress Recovery, Mode Shape Recovery, etc., on a link that has been reduced in Nastran.

6.4.1 Nastran DMAP

The Nastran model reduction is performed using a *DMAP* script to facilitate the generation of the reduced matrices needed in Fedem. A Nastran DMAP is a script program (kind of API and programming language) that modifies the execution of the Nastran solver. The script is model independent and must be *included* in the Nastran bulk data file when Nastran reduction is desired. The script, called *nastran_dmap.dat*, is located in the Template folder of the Fedem installation. Refer to the Nastran documentation for further details on the DMAP script language.

6.4.2 Nastran bulk data entries for CMS reduction

In addition the entries describing the FE model, the Nastran bulk data file must contain some additional entries when the link is going to be reduced in Nastran. These entries are listed in the following.

Before the CEND keyword, the following commands must be added to define the output files, solution type and the DMAP script:

- Define OP2-files used to store the reduced mass (m), stiffness (s) and gravity (g) matrices from Nastran:

```
ASSIGN, output2='<name>_m.op2',UNIT=71
ASSIGN, output2='<name>_s.op2',UNIT=72
ASSIGN, output2='<name>_g.op2',UNIT=73
```

<name> is some arbitrary string identifying this link.



NOTE: The OP2-files are automatically converted to *.fmx* files when the Nastran bulk data file is imported into Fedem, and the link is recognized as Reduced.

- Specify modal analysis:

```
SOL 103
```

- Include DMAP script:

```
INCLUDE 'nastran_dmap.dat'
```

After the BEGIN BULK keyword, the following commands must be added to perform the Nastran CMS-reduction:

- If you need to attach triads or joints to shell element nodes, you need to turn on the drilling dof on the Nastran shell elements:

```
PARAM, K6ROT, <value>
```

<value> = 10 or a small value is recommended. (If it is set to 0 the drilling dof is omitted, which is not what is desired). The value is used to calculate an artificial stiffness that might influence the overall stiffness. This might introduce undesirable effects, and the resulting component modes should therefore be compared with component modes calculated without this artificial stiffness.

- To get consistent instead of lumped mass representation the following bulk data command may be added:

```
PARAM, COUPMASS, 1
```

If coupled mass is not used, the DMAP script can be simplified since $M_{ie} = M_{ei} = 0$.

- Definition of static modes (external dofs). Use ASET or ASET1 entries to define the external dofs. To get 6-DOF Triads, all nodes in the ASET (active dof set) must have 6 active DOFs. A typical bulk entry is:

```
ASET, node_id1, 123456, node_id2, 123456, ...
```

Adding component modes (generalized dofs)

- If component modes (generalized dofs) are desired, the following bulk entries must be specified:

```
EIGRL, 1, , , <idn>
```

```
SPOINT, <id1>, <id2>...<idn>
```

```
QSET1, 0, <id1>, <id2>...<idn>
```

Here <idn> is the number of component modes.

6.5 Dynamics analysis



To control the dynamics analysis parameters, click the *Dynamics Solver Setup* button on the Solvers toolbar (or Solve menu). The Dynamics Solver Setup panel will appear and allow you to adjust the process setup.

In-depth information about the time integration process may be found in the Fedem R5.0 Theory Guide, *Section 7.3, "Newmark integration algorithm"*.

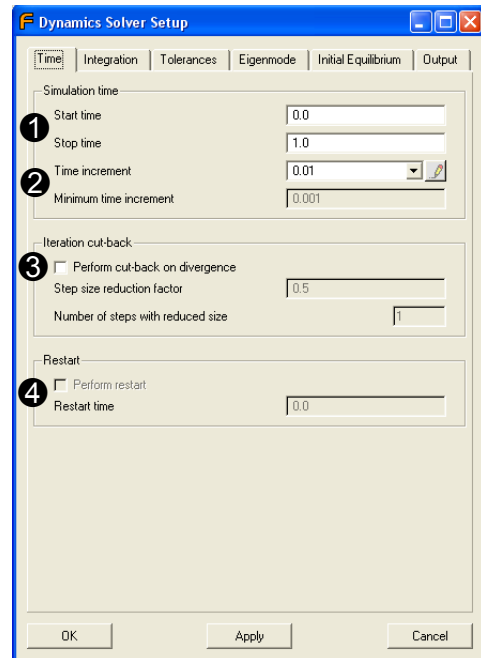
6.5.1 Dynamics Solver Setup

The control parameters for the Dynamics Solver are placed on six tabs labeled *Time*, *Integration*, *Tolerances*, *Eigenmode*, *Initial Equilibrium*, and *Output*.

Time tab

The time domain of the dynamics simulation is controlled through the following parameters:

- ❶ You can define the *Start time* and the *Stop time* of the dynamics simulation.
- ❷ You can define the size of the *Time increment* to be used by the time integration algorithm. In addition to a constant value, you may also select a *Function* or a *Time history input file* from the pull-down menu, in order to obtain a varying time increment size (see [Section 4.10, "Functions"](#)). In that case, the *Minimum time increment* is used as a lower bound on the step size.
- ❸ You can enable/disable the use of *Iteration cut-back* when the dynamics simulation diverges, and adjust the *Step size reduction factor* defining the size of the new time step to use in the cut-back, and the *Number of time steps with reduced*



size before the normal step size is resumed. If the cut-back iterations also diverge, another cut-back is attempted by applying the *Step size reduction factor* again. This procedure is then repeated until convergence is obtained or the *Minimum time increment* is reached. In the latter case, the simulation is aborted.

- ④ You can enable a *Restart* simulation if you already have some simulation results, and specify the *Restart time* defining the time step to restart from. If the specified time does not match an existing time step, the closest step after the specified time is used.



NOTE: In a restart simulation, you are allowed to adjust any of the other Solver Setup parameters and options, as well as adjusting the additional solver options for the Dynamics solver (see [Section 6.2.3, "Additional solver options"](#)). However, you can not change any properties of the mechanism model itself.

Restarting a Dynamics simulation is very handy if you discover you need to continue a simulation that was terminated abnormally, or you just want a longer event than was originally defined. You may restart a simulation as many times you wish.

Each restart adds a new set of result files to the existing results database such that subsequent post-processing and recovery runs can be conducted over the overall time domain spanned by both the initial run and the restart(s). If the time domain of the individual runs overlap, only the latest produced results will be used in post-processing and recovery. See [Section 8.2.3, "Result files from restart simulations"](#) for more information on the management of results from restart simulations.

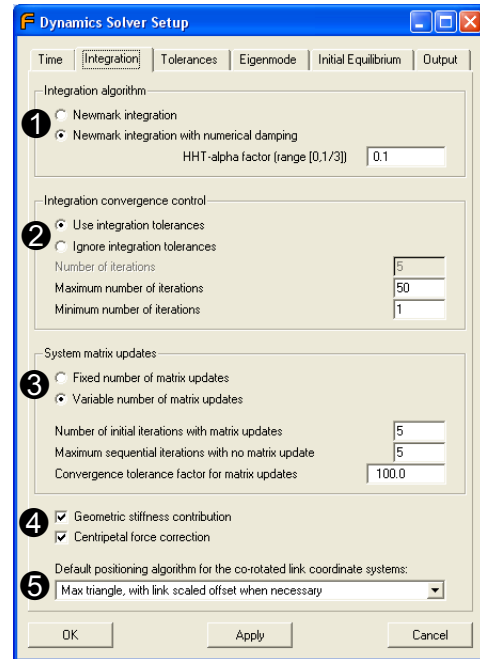
Integration tab

You can optimize numerical performance of the time integration by adjusting the following parameters:

- ① You can select *Newmark integration* with or without *numerical damping* (numerical damping, $\alpha = 0.1$, is recommended).
- ② You can enable/disable the use of *Integration tolerances*, and specify the *Maximum* and *Minimum number of iterations* for each time increment. When *Ignore integration tolerances* is set, the fixed *Number of iterations* is specified instead.
- ③ The nonlinear equations are solved using Modified Newton-Raphson iterations meaning that the system matrices are not necessarily recalculated in each iteration.

For efficiency reasons, the number of matrix updates per time step should be as low as possible. However, if the increments in the input variables are large during a time step, the system matrices need to be updated more often, to ensure the nonlinear iterations converge.

You may choose between *Fixed number of matrix updates* or *Variable number of matrix updates*. In either case you can also specify the *Number of initial iterations with matrix updates* and the *Maximum sequential iterations with no matrix updates*. The first number defines how many iterations in the beginning of each time step should be performed with updated matrices. If the iterations have not converged before reaching that number, the subsequent iterations are performed with a matrix update frequency defined by the *Maximum sequential iterations with no matrix updates*. However, if *Variable number of matrix updates* is chosen, a convergence based threshold is used in addition to determine when to do further system matrix updates. The factor entered in *Convergence tolerance factor for matrix updates* is then multiplied with the active convergence



tolerances specified on the "[Tolerances tab](#)". The resulting tolerance is compared to the error norms corresponding to the active convergence tolerances and the matrices are updated as long as the error norm is higher than this tolerance.

- ④ You can enable/disable both the *Geometric Stiffness Contribution* and the *Centripetal Moment Correction* during the non-linear iterations.

The geometric stiffness option accounts for stress stiffening in links, and axial springs and dampers. It may therefore improve the convergence of the nonlinear iterations if the mechanism contains links with large membrane forces or axial springs and dampers with large axial forces. This is because the forces alters the bending- or rotational stiffness of these elements. A tensile membrane/axial force will effectively increase the bending/rotational stiffness and a compressive force will reduce the bending/rational stiffness. For information on the computation of the geometric stiffness on links, see the Fedem R5.0 Theory Guide, *Section 4.4 "Superelement tangent stiffness"*.

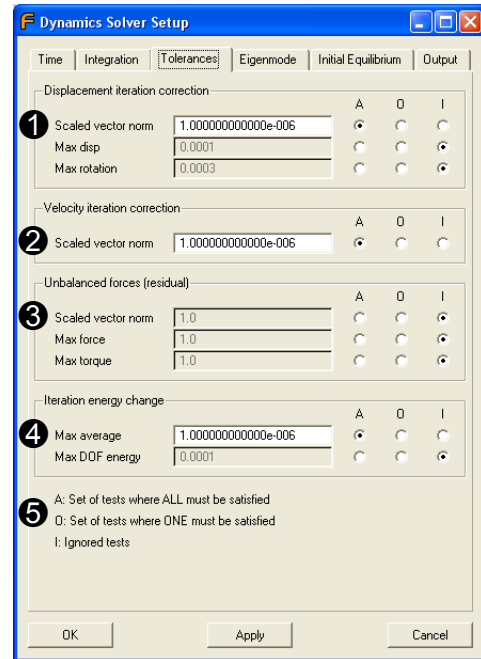
The centripetal moment correction option enables an improved representation of the inertia forces on links that have only a few Triads, but experience high-speed rotations (see the Fedem R5.0 Theory Guide, *Section 3.3, "Inertia forces and high-speed rotation"* for details).

- ⑤ You can specify the default algorithm for calculation of the co-rotated link coordinate systems during the simulation. The selections available in this pull-down menu correspond to those of the similar pull-down menu in the "[Advanced tab](#)" of the Link property panel (see [Section 4.1, "Link properties"](#)). The setting here applies to all links in the model, where the corresponding setting on the link level is "*Model default*".

Tolerances tab

Convergence criteria for the dynamics analysis are defined by enabling one, or more, convergence tolerances:

- ❶ This allows you to enable/disable and set convergence tolerances for the *Displacement iteration corrections*.
- ❷ You can define a convergence criterion on *Velocity iteration correction*
- ❸ You can define various tolerances for *Unbalanced forces*.
- ❹ You can define convergence tolerances for iteration energy changes.
- ❺ All available convergence tests can be ignored or defined into one of two sets of tests: In set A, all tests must be satisfied for convergence to be satisfied. In set O, only one of the tests must be satisfied (in addition to all the tests in set A) for convergence to be satisfied.



The various norms used in the above convergence criteria have different dimension properties. Some are dimensionless, whereas others depend on the model units. The default values defined are suitable for the SI unit set. If you model in a different unit set (see [Section 3.9, "Model preferences"](#)), you will need to adjust some of your active tolerance values accordingly.

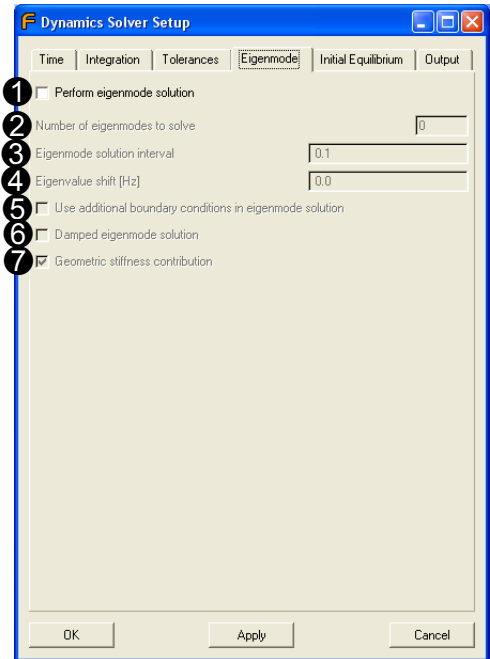
The scaled vector norm of the displacement correction is dimensionless, whereas the same norm for the velocity correction has dimension $1/[time]$ and for the force residual it is $[force][length]$. The Max norms have the dimension corresponding to the quantity that they measure.

For more information about convergence criteria, see the Fedem R5.0 Theory Guide, [Section 7.2.1, "Convergence criteria"](#).

Eigenmode tab

You can set up the calculation of eigenmode solutions (see "[Modal analysis](#)" in [Section 6.1.2](#)) by adjusting the following parameters:

- ① This option allows you to enable/disable calculation of the eigenmode solutions.
- ② You can specify the number of eigenmodes to be computed.
- ③ You can specify the time interval between each eigenvalue analysis.
- ④ You can specify an *Eigenmode Shift* (see the Fedem R5.0 Theory Guide, [Section 9.6.3, "Using shift when solving the eigenvalue problem"](#)).
- ⑤ You can enable/disable the application of the *Additional Boundary Conditions* specified for triads (see [Section 4.3.3, "Triad properties"](#)).
- ⑥ You can enable/disable computation of damped eigenmodes by accounting for structural damping (see the Fedem R5.0 Theory Guide, [Section 9.6.2, "Damped eigenvalue problem"](#)).
- ⑦ You can enable/disable the *Geometric Stiffness Contribution* in the eigenvalue analyses (see the similar option in the "[Integration tab](#)"). If the mechanism contains structural members that experience large tensile or compression forces at certain time steps, the geometric stiffness contribution may have a significant effect on the accuracy on the computed eigenvalues at those time steps.



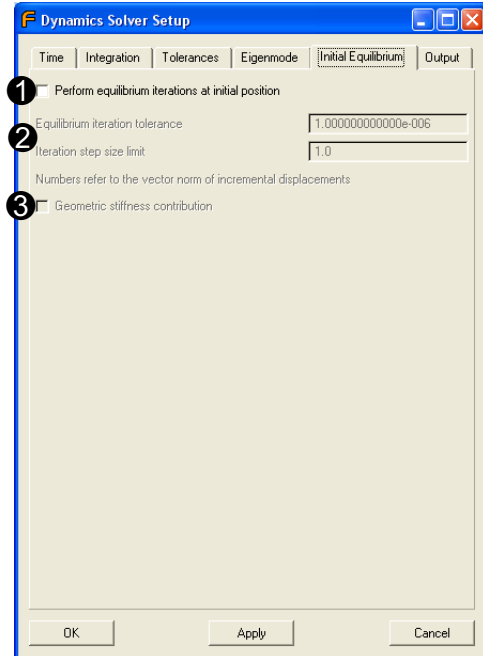
CAUTION: Computation of damped eigenmodes takes considerably longer time than computing the undamped modes.

Initial Equilibrium tab

If, during modeling, your model is not positioned at its static equilibrium position, it is recommended that you perform the *Initial Equilibrium* analysis to move the mechanism to a resting position before simulating the dynamics.

- ❶ You can enable this option to perform initial equilibrium iterations.
- ❷ You can adjust the iteration tolerance and the step-size factor for the initial equilibrium iterations.
- ❸ You can enable/disable the *Geometric Stiffness Contribution* in initial equilibrium iterations (see the similar option in the "[Integration tab](#)").

The *Equilibrium iteration tolerance* is the convergence tolerance on the norm of the iterative displacement correction during the equilibrium iterations (equivalent to the *Scaled vector norm tolerance* for *Displacement iteration correction* in the "[Tolerances tab](#)").



The *Iteration step size limit* is an upper limit on the norm of the displacement correction vector within one iteration. If the norm is higher than this value, the correction vector is scaled down such that its norm equals this number. Each time this happens, the iteration counter that is compared to the *Maximum* and *Minimum Number of Iterations* parameters of the "[Integration tab](#)", is reset to zero.



NOTE: *Maximum* and *Minimum Number of Iterations* set in the "[Integration tab](#)" also apply in the *Initial Equilibrium* analysis.

TIP: The defaults for the *Equilibrium Iteration Tolerance* and *Iteration Step-Size Factor* are usually acceptable. However, if the mechanism is modeled far from the equilibrium position, reducing the *Iteration step-size limit* may improve performance.



CAUTION: The Iteration step size limit must always be larger than the Equilibrium Iteration Tolerance.



CAUTION: To perform the equilibrium analysis, you may have to apply Additional Boundary Conditions before starting the dynamics analysis. (For information about applying such boundary conditions, see [Section 4.3.3, "Triad properties"](#).)

Output tab

On this tab you can control the automatic curve and animation export from the Dynamics Solver to file. The automatic curve export is useful if you want to run Fedem in an iterative loop with some external software, and need to process selected solver output in order to calculate new input for subsequent runs. The automatic animation export writes a GLview VTF-file with the rigid body motion of the computed response. This file may then be opened in the Ceetron GLview software for further viewing (see www.ceetron.com for further details on GLview).

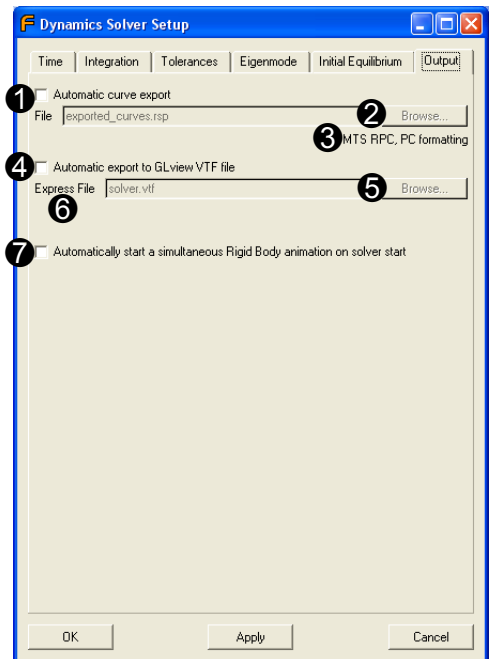
① This toggle enables export of all curves in the model with *Export curve automatically* toggled on in the Property Editor panel (see [Section 7.2.5, "Curve properties"](#)).

② This field shows path and name of the file the curve data will be written to. Press the *Browse* button to change file name or file format.

③ This label shows the selected format for the curve export. Available formats are MTS RPC (UNIX or PC formatting), and tab-separated multi-column ASCII.

Only curves plotting results produced by the Dynamics

Solver can be exported in this manner. If you need to include curves with results from other Fedem solvers in the same output file (e.g. results from subsequent strain rosette analyses), you have to run the Fedem Curve Export Utility module instead (see [Section 6.13, "Automated curve export from multiple result database files"](#)).



- ④ This toggle enables export of the rigid body motion of the computed response to a GLview VTF-file.
- ⑤ This field shows path and name of the VTF-file that will be written. Press the *Browse* button to change file name or file format.
- ⑥ This label shows the selected VTF file format. Available formats are Express, Binary and ASCII.
- ⑦ You can enable/disable the automatic start of a simultaneous rigid body animation when the Dynamics Solver is started. This is useful when doing rapid prototyping simulations of short duration, when it is essential to get quick feedback on the dynamic response.



CAUTION: When the automatic VTF export is enabled, the FE data for all links in the model is written to the specified VTF file before the Dynamics Solver is started. For large models this may take some time (especially if the FE data is unloaded), and the Fedem UI is blocked while the FE data is being exported.

6.5.2 Result output control

To control the size and contents of a dynamics analysis result database, several additional options exist.

The results data from a dynamics analysis is divided into primary and secondary variables. The primary ones consist of triad- and superelement (link) position matrices, and, if any, generalized displacements for super-elements having component modes. All other variables are secondary variables. The output frequency, accuracy and the amount of these variables may be controlled using Additional Solver Options for the Dynamics Solver (see [Section 6.2.3, "Additional solver options"](#)). These options are discussed below. Refer to [Appendix E, "Command line options"](#) for a complete list of solver options.

Output start-up time

Before presenting the options to format the output, we note that the time of output start-up may also be controlled. This is done using the "-savestart" solver option. The option applies to both the primary and secondary variables, as well as internal control system data.

-savestart Time for first save to response database



NOTE: The default value of this option is 0.0. Therefore, if you are using a "Start time" less than zero and want to save the results also for the time steps with physical time less than zero, you have to specify "-savestart <start time>" as an additional solver option for the Dynamics Solver.

Output frequency

Primary variables are output for all time steps of a dynamics simulation and, while using default settings, so are the secondary ones. The output frequency of secondary variables, however, may be lowered by using the "-saveinc2" solver option.

-saveinc2 Time between each save of secondary variables

For models with an internal control system, the "-saveinc2" option applies by default to control system data that is needed in a restart simulation as well (see "*Time tab*" in [Section 6.5.1](#)). However, it is possible to specify a separate output frequency for those data through the option

-saveinc4 Time between each save of control system data

Output accuracy

Primary variables are by default output in double precision (64-bit real). However, the output precision may be set to single by specifying the solver option "-double1-" (the appending minus sign indicates a *false* setting). This will make the primary results file created by the Dynamics Solver half the normal in size.

-double1- Save primary variables in single precision



CAUTION: *You should not switch off double precision output for primary variables unless you are particularly low on disk space, as this may affect the accuracy of the subsequent recovery runs. This is particularly true for models experiencing large global displacements but small local deformations.*

Secondary variables are by default output in single precision (32-bit real). This is sufficient for most purposes, and contributes to keeping the disk space needed by the secondary results file at a low level in long simulations. However, if you plan to use some of secondary variable results as input functions in subsequent simulations, it can be essential to retain full precision in the saved results to obtain satisfactory accuracy in the latter simulations. For this purpose, the solver option "-double2" can be used.

-double2 Save secondary variables in double precision

When specified, most of the secondary variables will be saved in double precision. Some quantities that are not likely to serve as input in subsequent runs are always saved in single precision. Thus, the "-double2" option currently affects only the following types of quantities:

- Triad velocities, accelerations and forces
- Spring stiffnesses, lengths, deflections and forces
- Damper coefficients, lengths, velocities and forces
- Friction forces
- External force values and vector components
- Joint variables, velocities and accelerations
- Tire contact point and wheel carrier forces
- Control line variables

Output selection

While using default settings all primary and secondary variables available are output to file. The selection of secondary variables to output, however, may be customized. The "*-allSecondaryVars*" option is used to switch on/off the output of secondary variables altogether, while a number of options of the form "*-allNameOfEntityVars*" are used to control the output of a range of entities. (E.g. "*-allTriadVars*" requests the output of all variables related to triads, while "*-allForceVars*" requests the output of all force variables in the simulation.)

All available solver options are listed in [Appendix E.3, "Dynamics solver options \(fedem_solver\)"](#).

File buffering and flush frequency

As stated above, the dynamics solver writes the results to (up to) three different results files. Since the amount of data output per time step and the output frequency are different for the three files, it is important to be able to synchronize the actual data output at given time steps. This is needed especially when doing a simultaneous simulation and animation/curve plotting. For this purpose, the "*-flushinc*" option can be used.

<i>-flushinc</i>	Time between each database file flush
	< 0.0 : Do not flush results database (let the OS decide)
	= 0.0 : Flush at each time step, no external buffers (default)
	> 0.0 : Flush at specified time interval, use external buffers

The default action (*flushinc* = 0.0) will do a physical flush of all open results files at the end of each time step. Each open file is associated with an internal file buffer of fixed length. For small models the amount of data per time step is usually less than the size of this buffer. The data will

therefore not be written physically to disk at the end of the time step, unless we force a physical flush.

Instead of relying on the fixed-size internal buffer, you may also instruct the solver to do a physical flush at fixed time intervals ($\text{flushinc} > 0.0$). An external buffer is then allocated for each result file which is big enough to just hold the number of time steps corresponding to the given flush interval. This can be used to reduce the number of file IO operations to a minimum, and may speed up the overall solution time if you have a slow disk (or data network if using a remote device). Moreover, if the model is so big that a single time step does not fit in the internal buffer¹, it will write data physically to the file in several steps during a time step, and you may risk synchronization problems in a simultaneous simulation animation/plotting. Specifying a flushinc greater than or equal the simulation time step size will prevent this.

By specifying a flush interval less than zero, you let the operating system decide when it wants to physically write data to file. When running the solver batch, or through the user interface with no graphs or animations loaded, this is normally sufficient (unless you have a slow disk drive, then $\text{flushinc} > 0.0$ may work better). This is also equivalent to the behavior of the R3.0 solver (or earlier versions).



TIP: For small models (i.e. few system degrees of freedom) that run over a long time span using a huge number of time steps, the default value ($\text{flushinc} = 0.0$) may severely hamper the performance, as it is forced to write small data amounts to disk at a very high frequency. In such cases, using a value less than zero or (much) greater than the time step size is recommended.

6.5.3 Monitoring the most problematic DOFs during time integration

For complex models, it is not likely that the dynamics simulation will run smoothly in the first attempt. Usually, some fine tuning and /or model correction will be required before a converged solution can be obtained. To aid the debugging of problematic models, the degrees of freedom that get the largest solution increments during the non-linear iterations will be listed in the `.res` file when the simulation experiences convergence problems. Often, the problem can then be identified by studying the elements related to these DOFs and verifying that their geometry and/or properties are sane.

1. The size of the internal file buffer is platform dependent. Consult the technical documentation on your computer operating system to find out how big it is.

Fedem will produce those messages when the convergence criteria employed (see "[Tolerances tab](#)" in [Section 6.5.1](#)) increases in two or more consecutive iterations, or the number of iterations is getting close to the maximum number of iterations specified. It is then assumed that a convergence problem is encountered, and the specified number of the worst DOFs will then be printed along with local DOF number, object entity information, and the actual solution increment in that DOF.

Finally, you can specify the Dynamics Solver to abort the simulation after a certain number of such poor convergence warnings have been issued.

The DOF monitoring is controlled by means of Additional Solver Options (see [Section 6.2.3, "Additional solver options"](#)). The following options are available for this purpose (default value in parentheses):

- monitorWorst* Number of DOFs to monitor on poor convergence (6)
- monitorIter* Number of iterations to monitor before *maxit* (2)
- stopOnDivergence* Number of warnings on possible divergence before the dynamics simulation is aborted (0 = no limit)

6.5.4 Starting the analysis



Once you have set up the dynamics solver options, you can start the dynamics simulation by clicking the *Solve Dynamics* button on the Solvers toolbar or in the Solve menu.

6.5.5 Handling singularities during the dynamics analysis

The Dynamics Solver has a similar treatment of singular equations during the solution of linear equation systems, as the Link Reducer (see [Section 6.3.4, "Handling singularities during the model reduction"](#)). All singular DOFs will thus be attempted found in one go, letting you fix them all before a new analysis is attempted. However, in the Dynamics Solver, neither type of singularities (true zero pivots, or reduced-to-zero pivots) are permitted. The solver will therefore abort on any occurrence of singularities after the triangularization is completed. All singularities found are then listed in the Output List and on the *.res* file. They are identified with the internal node number and the mechanism entity it is associated with (Triad DOF, Joint DOF or Link component mode).

6.6 Stress recovery analysis

6.6.1 Stress recovery options



To specify parameters for the stress analysis, click the *Stress Recovery Setup* button on the Solvers toolbar (or Solve menu).

Each time stress recovery is run, the results are added to the existing stress recovery results. This means you could solve stresses for one time interval first, and subsequently for another interval, while stresses from both intervals could be animated in the same animation. You could also solve stress first, view them, and solve strains later, making both stress and strain available for post-processing.

This also means that if stress recovery is performed more than once using identical settings, the same results will be stored multiple times on disk. It is not checked whether the results you asked for already exist or not.

- ① To specify the time steps at which stress is recovered, a *Start* time, *Stop* time and a time *Increment* should be provided. However, if the *Use all time steps* option is enabled, stress recovery will be performed for all computed time steps between the specified start and stop time. The *Reset* button restores the default *Time Interval* values, which are equal to the start and stop times of the simulation as specified in the "*Time tab*" of the *Dynamics Solver Setup* dialog, and 10 times the (initial) time increment used in the simulation.

The **Stress Recovery Setup** dialog box is shown with the following settings:

- Time Interval** (labeled ①):
 - Start: 0.0
 - Stop: 1.0
 - Increment: 0.1
 - ☐ Use all time steps
 - Reset button
- Output Options** (labeled ②):

☒ deformation	☒ stress resultant tensor
☒ strain tensor	☒ stress tensor
☐ von Mises strain	☐ von Mises stress
☐ max principal strain	☐ max principal stress
☐ min principal strain	☐ min principal stress
☐ max shear strain	☐ max shear stress
- Automatic export to GLview VTF file** (labeled ③):
 - ☐ Automatic export to GLview VTF file
 - Express File: stress.vtf (with Browse... button)
 - Contour range for Express VTF file (von Mises stress only):

Min: 0.0	Max: 0.0
----------	----------
- Buttons: OK, Apply, Cancel

- ② You can choose which type of results to recover by activating the appropriate toggles. For very large models it is recommended not to

recover more results than you actually want to animate, as the efficiency of the animation process, and the size of the results database files, is dependent on the amount of data computed.



NOTE: If you choose to recover the stress and/or strain tensors, there is no need to also toggle on the derived quantities (von Mises, max principal, etc.). You may always animate the derived quantities as long you have recovered the stress/strain tensors using the Operation menu in the Property Editor panel of the animation (see "[Contours tab](#)" in [Section 7.3.2](#)).

- ③ You may enable direct export of a GLview Express VTF-file with von Mises stress contours and optionally the deformed shape, for further viewing in the Ceetron GLview environment (see www.ceetron.com). You may also need to specify the contour range to be used in the exported VTF-file (the max and min. values of the exported von Mises stress will be used if no range is specified).



NOTE: The deformation toggle in the Output Options frame also affects the GLview VTF export from the Stress Recovery. The deformed shape will thus be exported to the VTF file only when this toggle is on. None of the other toggles influence the VTF export.

6.6.2 Result output control

As explained above, the Stress Recovery Setup panel may be used to specify the result types wanted as output from the stress analysis. There are, however, some additional options for customizing the size and contents of the results database that may be specified in the *Additional Solver Options* dialog (see [Section 6.2.3, "Additional solver options"](#)). Some of these options are discussed below. The complete list of options to the Stress Recovery solver module is found in [Appendix E.4, "Stress recovery options \(fedem_stress\)"](#).

Output accuracy

By default the results from a stress recovery analysis are output in single precision (32-bit real). The output precision may, if so wanted, be set to double precision (64-bit real) by using the "-double" solver option.

6.6.3 Import of residual stresses from external analyses

Sometimes one or more parts in a model may be subjected to stresses from molding or other temperature processes that have been computed in other simulation packages. These stresses may then be regarded as a residual stress state on which the stresses from the dynamics simulation in Fedem are superimposed. You may specify an external results file with such residual stresses for each link in the "[Advanced tab](#)" of the Property Editor panel (see [Section 4.1.4, "Link properties"](#)).

The stress state from the external results file is imported into the stress recovery module and added to the computed stresses from the dynamics simulation in Fedem for each time step, before the stresses are output to the results database for subsequent animation. The residual stress state is also output separately as a time step 0, for reference. The *Start* time of the stress recovery needs to be specified greater than 0.0 when using this feature, if not the residual stress field will be ignored.



NOTE: The imported residual stress state have no influence on the dynamics simulation in Fedem. Consequently, any potential non-linear effect caused by the residual stress will not be accounted for in the simulation.



NOTE: The imported residual stress state is also applied in the Strain Coat Recovery (see [Section 6.9.3, "Import of residual stresses from external analyses"](#)), but currently not in the Strain Gage recovery.

6.6.4 Starting the analysis



Once you have set up the stress recovery options and performed the dynamics simulation, you can start the stress recovery for the selected time steps by clicking the *Recover Stress* button on the Solvers toolbar (or Solve menu). You may also run the stress recovery only on one, or a selection of links or on individual element groups, see [Section 6.2.4, "Link- and group-wise solving"](#).



NOTE: If element calculations fail during stress recovery analysis, the recovery continues on the other elements. All failed elements will appear grey in the contour plot after the animation is loaded.

6.7 Mode shape recovery analysis

6.7.1 Mode shape options

To animate or display detailed mechanism mode shapes, you must first set up and perform the dynamics analysis and then the mode shape recovery analysis.

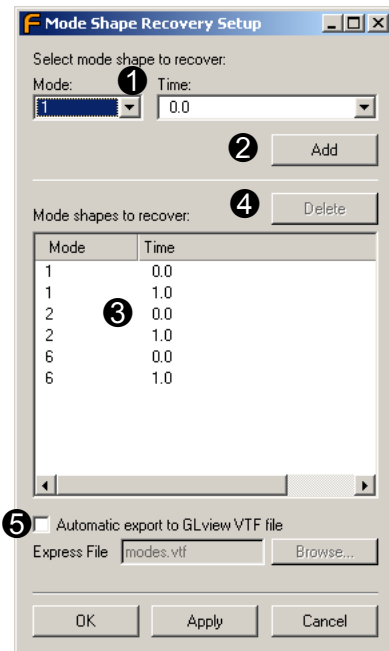


NOTE: Rigid body mode shapes can be animated without a mode shape recovery analysis (see [Section 7.3.2, "Animation properties"](#) for more information).



To set up the mode shape analysis, click the *Mode Shape Recovery Setup* button on the Solvers toolbar (or Solve menu). The Setup dialog is displayed as shown below. You can then select the modes you want to expand and add them to the list of selected modal results.

- ❶ *Mode and Time* – pull-down lists with mode numbers and times at which eigenmodes have been (or will be) calculated in the dynamics analysis.
- ❷ *Add* button – inserts selected mode shape for the selected time into the results list.
- ❸ *Mode shapes to recover* – list of selected mode shapes to expand during postprocessing (sorted by mode number).
- ❹ *Delete* button – removes the selected mode and time from the results list.
- ❺ You may enable a direct export of a GLview Express VTF-file for each recovered mode shape, for further viewing in the Ceetron GLview environment(www.ceetron.com). If a mode shape is to be recovered for more than one time step, that shape will be exported to VTF only for the first time step specified.



TIP: The *Mode and Time* pull-down lists both have an (All) entry in the bottom to facilitate easy selection of all entries in the list. This is useful if you want to recover all mode shapes and/or the selected mode shape for all time steps at which it has been computed.



NOTE: The expanded mode shapes for one link are stored in a single frs-file provided all modes are expanded for the same set of time steps. However, if only one mode is expanded for a different set of time steps than the other modes, one frs-file is created for each expanded mode. Therefore, if you are expanding a large amount of modes, it is advised to select the same time steps for all modes in order to limit the number of files in the results database, as this might impact the post-processing performance.

6.7.2 Starting the analysis



Once you have set up the mode shape recovery options and performed the dynamics simulation, you can post process the modes at the selected times by clicking the *Recover Mode Shapes* button on the Solvers toolbar (or Solve menu).

When Fedem has postprocessed all the selected modes, you can animate the mode shapes (see [Section 7.3, "Animations"](#) for more information).

6.8 Strain rosette analysis

The strain rosette analysis recovers the stresses and strains on the virtual strain rosettes defined in your model. See section [Section 4.11.3, "Strain rosettes"](#) on how to create and edit virtual strain rosettes. The output is similar to the output from real strain gages in addition to standard strain and stress quantities like Von Mises, principal stresses/strains, and angle of max/min. principals.

6.8.1 Strain rosette options



To specify parameters for the strain rosette analysis, click the Strain Rosette Recovery Setup button on the Solvers toolbar (or Solve menu).

- ① You can specify the *Start* and *Stop Time* for strain rosette recovery and the *Time Increment* at which to recover the rosettes. If the *Use all time steps* option is enabled, rosette recovery will be performed for all computed time steps between the specified start and stop time. The *Reset* button restores the default *Time Interval* values, which are equal to the start and stop times of the simulation as specified in the *Dynamics Solver Setup*, and using all time steps in between.

- ② You may enable a direct export of gage strains to DAC files with a specified sample rate.
- ③ You may enable a stress cycle count with a specified bin size. The results are written to the file `fedem_gage.res` when running the strain rosette recovery. The computed damage will also be reported to this file for all strain rosettes when this option is enabled.
- ④ Strain rosette definitions can be imported from a file by pressing this button. The strain rosettes defined will then be read into Fedem, and virtual strain rosettes will be created automatically based on the definitions. See [Section 6.8.4, "Strain rosette definition file format"](#) for the format of this file.

6.8.2 Starting the analysis



Once you have set up the strain rosette recovery options and performed the dynamics simulation, you can start the strain rosette recovery by clicking the *Recover Strain Rosettes* button on the Solvers toolbar (or in the Solve menu).

6.8.3 Result output

Results for all strain rosettes on a link are output to the binary `.frs` file named `<linkname>_#.frs`. The `.frs` file allows the post-processing of strain rosette recovery results through graphs.

In addition to the stress and strain results, you may also output the nodal deformations in the strain rosettes to the `.frs` file by specifying `-deformation` as an additional option to the Strain Rosette Recovery (see [Section 6.2.3, "Additional solver options"](#)).

If enabled, a file named `rosette<ID>_gage<n>.dac` is written for each strain gage, containing the strain of leg `<n>` in strain rosette `<ID>`. This file is output in the nCode DAC format.

It is also possible to output the strain rosette result directly to ASCII files. By specifying `-writeAsciiFiles` as an additional option to the Strain Rosette Recovery, you will get a file named `rosette<ID>.asc` for each strain rosette defined. A summary of contents for this ASCII file is given below.

All of the above mentioned files are created in separate directories for each link, which then are placed in the sub-directory `timehist_gage_rcy_####/` in the result file hierarchy (see [Section 8.3, "RDB directory structure"](#)).

ASCII output file format

The `rosette<ID>.asc` output file contains a heading and a data section. (See file cutout below.) The heading summarizes the strain rosette definition, the data section lists all measurements made.

```

# Strain rosette identifier:      1      Link :      1
# Global nodes      :      844      846      830      828
# Number of gages      :      2
# Angle between gages      :      90.000
# Position      :      2.336E-001 -7.206E-001  4.500E-002
# Position along Z-axis      :      0.000E+000
# X-direction unit vector      :      0.000E+000  1.000E+000  0.000E+000
# Y-direction unit vector      :      -1.000E+000  0.000E+000  0.000E+000
# Z-direction unit vector      :      0.000E+000  0.000E+000  1.000E+000
#
#      time      eps_x      eps_y      gamma_xy      eps_gage1      eps_gage2
0.000000E+000 -9.751E-018 -3.048E-018  2.205E-018 -9.751E-018 -3.048E-018
2.000000E-003  1.835E-006  4.353E-007 -2.367E-007  1.835E-006  4.353E-007
4.000000E-003  6.391E-006  1.490E-006  3.337E-007  6.391E-006  1.490E-006
6.000000E-003  1.041E-005  2.397E-006  1.660E-006  1.041E-005  2.397E-006
8.000000E-003  1.008E-005  2.320E-006  1.621E-006  1.008E-005  2.320E-006

```

Note that the cutout above is an edited version not showing the entire file contents. Detailed information on the full contents of the data section is listed below.

Data output	Description of output
Time	Time
ϵ_x	Strain XX in rosette coordinate system
ϵ_y	Strain YY in rosette coordinate system
γ_{xy}	Shear strain XY in rosette coordinate system
ϵ_1	First principle strain
ϵ_2	Second principle strain
$\gamma_{\max}$	Max shear strain
α_1	Angle from rosette X-axis to direction of principle strain
α_γ	Angle from rosette X-axis to direction of max shear
σ_x	Stress XX in rosette coordinate system
σ_y	Stress YY in rosette coordinate system
τ_{xy}	Shear stress XY in rosette coordinate system
von Mises	von Mises stress
σ_1	First principle stress
σ_2	Second principle stress

Data output	Description of output
$\tau_{\max}$	Max shear stress
$\varepsilon_{\text{gage1}}$	Strain in leg1
$\varepsilon_{\text{gage2}}$	Strain in leg2
$\varepsilon_{\text{gage3}}$	Strain in leg3

All angles are output in degrees to the ASCII file.

File buffering and flush frequency

The output to the frs-file may be controlled using the additional option *-flushinc*, which works in the same manner as with the Dynamics Solver (see [Section 6.5.2, "Result output control"](#)). However, the default value is here -1.0, i.e. the file is flushed to disk when the internal fixed-size file buffer is filled. Note that this option only affects the frs-file output. The DAC and ASCII file output is not affected.

6.8.4 Strain rosette definition file format

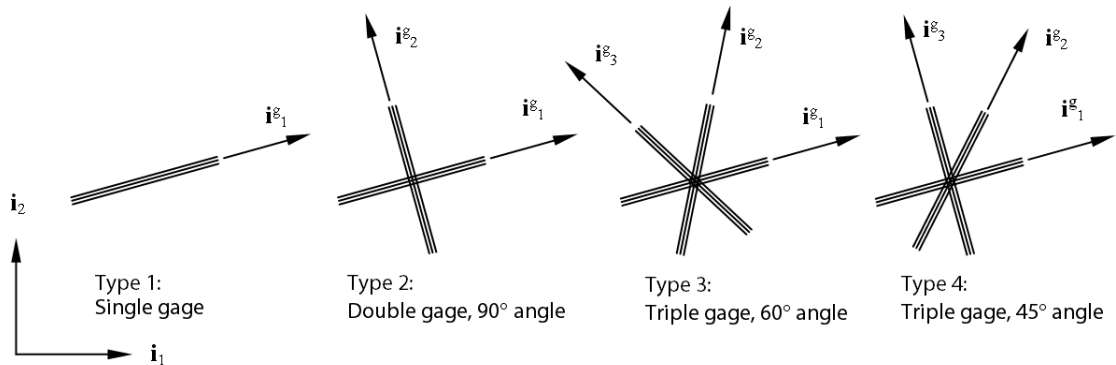
It is possible to define strain rosettes in an ASCII input file, as shown in the example below, and read them into fedem.

```
#id type link numNodes n1 n2 n3 n4 zPos X_x X_y X_z Z_x Z_y Z_z E-mod nu
10 2 1 4 844 846 830 828 0.0 0.0 1.0 0.0 0.0 0.0 1.0 2.06e+11 0.0
20 4 1 3 845 846 828 0.01 0.0 1.0 0.0 0.0 0.0 1.0 2.06e+11 0.0
end
```

Each line defines a strain rosette, where the following data must be given:

id: The id number is used for naming purposes of the result files. The above example defines two strain rosettes, with identifiers 10 and 20 respectively.

type: According to the figure below, rosette 10 is of type 2, i.e. a two gages with 90° degrees between the gages, whereas rosette 20 is of type 3, three gages with 60° degrees between the gages.



link: The super element (link) number that the present rosette is attached to.

numNodes n1 n2 n3 (n4): Number of nodes for the element followed by node numbers that define the element topology. Permissible values for numNodes are 3 and 4, giving CST triangle and bi-linear quad element respectively. The nodal numbers must be given in a circular fashion around the element, where both clockwise and counterclockwise sequences are permitted. The node numbers will possibly be reversed internally so that the element Z-axis points in the user defined direction.

zPos: defines the location of the rosette along the Z-axis. If the user wants the strains measured at the top of a plate, the number at the Z-pos column should be $h/2$ where h is the plate thickness. A value of $-h/2$ will give the strains at the bottom of the plate.

X_x X_y X_z: These three numbers define a vector, which is used to define the local X-direction for the rosette coordinate system. The vector is given in the link coordinate system. The vector does not need to be given as a unit vector, nor does the vector need to lie exactly in the element plane. The X-direction used for computations is obtained through a projection of the user-defined vector onto the element plane.

Z_x Z_y Z_z: These three numbers represent a vector, which is used for defining the positive Z direction of the rosette coordinate system. The vector defined by the user does not need to be exactly perpendicular to the element plane. As long as the vector has a component pointing out from the element plane, the

Z-direction for the rosette coordinate system will be correctly defined. The rosette Y-direction is implicitly defined through the X- and Z-directions.

E-mod nu: Young's modulus (E-mod) and Poisson's ratio (nu) are used for computing the stress state at the rosette location.

6.9 Strain coat analysis

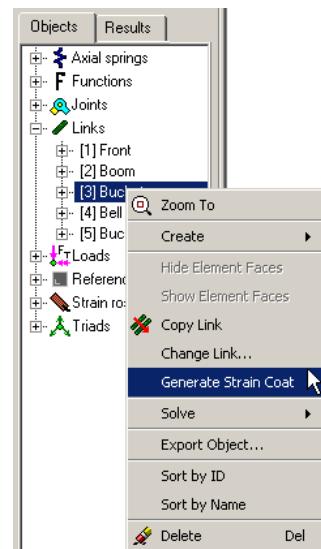
This analysis recovers the stresses and strains on all strain coat elements in the model, and calculates a summary of the recovered results as it processes. The output from the strain coat analysis is a result database file (.frs file) for each link containing the maximums of certain stress/strain quantities over the time interval considered. Optionally, you may also perform a rainflow and fatigue analysis based on the computed stress or strain histories during the strain coat recovery. The result files from the strain coat analysis are created in separate directories for each link below the *summary_rcy_####/* directory in the result file hierarchy (see [Section 8.3, "RDB directory structure"](#)).

6.9.1 Generating strain coat

Before you run the strain coat analysis, you must generate strain coat elements for the links or element groups in question. This is done by right-clicking the links or groups in the Model Manager Objects list and selecting Generate Strain Coat from the menu. Again, multi selection is possible.

One strain coat element is created for each non-interior face of the finite elements in the current selection. If a strain coat element already exist for a given face, a new strain coat element is not created. Therefore, repeating the Generate Strain Coat command for the same selection has no effect.

When creating strain coat elements by selecting an element group the new strain coat elements are automatically added to that group.



If a link is selected, the created strain coat elements are only added to the implicit element groups which the parent finite element belongs to, not to any explicit element groups that the parent element might be a member of. Refer to [Section 4.2, "Element groups"](#) to learn about implicit and explicit groups in Fedem.



NOTE: The strain coat elements will appear in the .ftl file in the link repository when the model is saved.

6.9.2 Strain coat analysis options



To specify parameters for the strain coat analysis, click the *Strain Coat Recovery Summary Setup* button on the Solvers toolbar (or Solve menu).

- 1 You can specify the *Start* and *Stop Time* for strain coat recovery summary and the *Time Increment* to be used. However, if the *Use all time steps* option is enabled, all computed time steps between the specified start and stop time will be considered. The *Reset* button restores the default *Time Interval* values which are equal to the start and stop times of the simulation as specified in the *Dynamics Solver Setup*, and to use all time steps in between.
- 2 You may limit the number of elements to be processed concurrently by adjusting this value. Especially for large links and long time series this might be necessary due to higher memory requirements.

- ③ You may set a gate value for the *Biaxiality* calculation. That is, the mean biaxiality will be computed only for elements whose max principal stress is larger than the specified threshold value.
- ④ You may toggle on/off rainflow and fatigue analysis¹. The remaining options in this window are sensitive only when this toggle is on.
- ⑤ When using the default fatigue solver, the Analysis type is always *Signed abs max stress (S-N)*. When using the nCode fatigue solver (see [Footnote 1.](#)), you may choose between *Signed abs max stress (S-N)* and *Signed abs max strain (E-N)* as the analysis type.
- ⑥ *Stress/Strain range threshold*: You may set a gate value for the *Peak valley extraction*. That is, only the stress/strain ranges with magnitude higher than this value will be included in the peak valley extraction.
- ⑦ You may set the *Histogram Range* values and number of bins to be used in each axis-direction for the rainflow analysis (only for nCode fatigue analysis, see [Footnote 1.](#)).

To perform rainflow and fatigue analysis during the strain coat recovery, you also need (unless nCode is used) to assign an S-N curve to base the damage calculation on for each element group you want to consider in the fatigue analysis. This is performed in the Property Editor panel for the element groups, see [Section 4.2.1, "Element group properties"](#).

6.9.3 Import of residual stresses from external analyses

Similarly as in the Stress Recovery (see [Section 6.6.3, "Import of residual stresses from external analyses"](#)) you may superimpose a constant residual stress state on the computed dynamics stress field, before calculating the strain coat summary quantities for S-N analysis. This feature will thus be enabled if you have specified an external results file with such residual stresses in the *"Advanced tab"* of the Property Editor panel for each link in question (see [Section 4.1.4, "Link properties"](#)).



CAUTION: The E-N analysis will not include the effects from the residual stresses as it is based on strain values.

1. When fatigue analysis is enabled, the default is to use an integrated fatigue solver based on pre-defined S-N curves. However, it is also possible to use nCode instead if you have that software installed and the necessary licenses available. To enable nCode usage, enter the string `#useNCode` in the *Strain Coat Recovery Summary* field of the *Additional Solver Options* dialog, see [Section 6.2.3, "Additional solver options"](#). See [Section 6.9.6, "Using nCode FE-Fatigue"](#) for more details on fatigue analysis with nCode.

6.9.4 Starting the analysis



Once you have set up the strain coat recovery summary options and performed the dynamics simulation, you can start the strain coat recovery by clicking the *Recover Strain Coat Summary* button on the Solvers toolbar (or Solve menu). The Strain Coat Recovery is then performed on each link in the model which currently contains strain coat elements. Links without strain coat elements are omitted automatically during the analysis.

6.9.5 Strain coat recovery on element groups or individual links

A strain coat recovery can also be performed on defined element groups or a selection of links. See [Section 6.2.4, "Link- and group-wise solving"](#) to learn how such a process is started. In this case, strain coat elements are created automatically for the selected link(s) or element group(s), before the recovery process is started. It is therefore no need to do this manually, as explained in [Section 6.9.1, "Generating strain coat"](#), in this case.

6.9.6 Using nCode FE-Fatigue

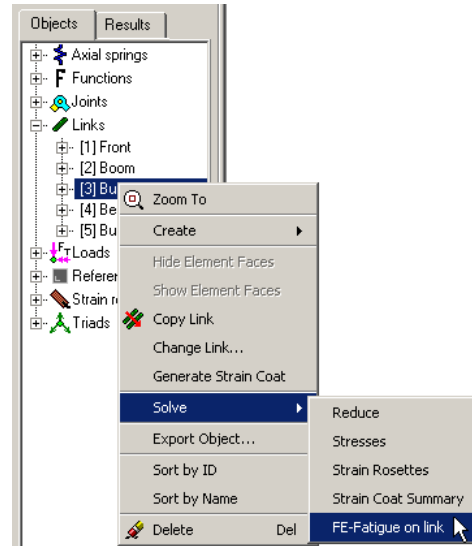
When *nCode* usage is enabled (see [Footnote 1.](#) above) an additional *.fpp* file is created for each link during the Strain Coat analysis. These files contains histograms over the maximum principal stress (for S-N analysis) or the maximum principal strain (for E-N analysis) for each strain coat element. In addition the following quantities are stored for each element:

- The element number, element type and element location
- The material group number
- Mean and standard deviation of the biaxiality ratio
- The biaxiality gate value (constant for each link)
- Most popular angle and angle spread

The FPP-file may be used in subsequent damage analysis in *nCode FE-Fatigue*. It is possible to launch FE-Fatigue from the Fedem User interface.

In similar manner to launching other solvers on specific links, FE-Fatigue is launched from the **Solve** menu in the right-click pop-up menu in the Model Managers Object list (see illustration to the right).

You may also launch FE-Fatigue on a selection of element groups. However, that has the same effect as launching FE-Fatigue on the link which the selected group(s) belong to.



NOTE: *nCode software needs to be installed on the computer to be able to run this command.*



NOTE: *The environment variable **npath** needs to be set to point to the nCode installation for the application to start. In addition, the path to the nSoft directory in the nCode installation must be in your **PATH**. On Windows, if nCode is already installed when you install Fedem, the installer takes care of this.*

6.9.7 Licensing needs

To be able to run the strain coat analysis with FPP-file creation, you will need both a license from Fedem Technology and nCode.

6.10 Duty cycle analysis

When you run a Duty Cycle analysis, you create a weighted sum of the damage from several load events. The results can be viewed with a *Time Summary Animation*.



NOTE: *Currently, Duty Cycle analysis can only be performed when the damage is computed using nCode. It cannot be combined with the internal fatigue solver.*

6.10.1 Getting started

To run a duty cycle analysis, you need several (almost) identical Fedem models. The links you wish to run Duty Cycle simulation on must be present in all models, and they need to have identical names and IDs in all events. Other than that, you are free to alter the models to your needs.



TIP: A Duty Cycle setup is an ideal case to use a link repository.

One of the models must be a designated Duty Cycle model, a model that is reserved for running Duty Cycle only. This model cannot have any other results.



NOTE: The chosen links need to be strain coated, otherwise you will not be able to view the results.

In all the events you must run Strain Coat Recovery simulation on the chosen links. In one model, referred to as "*Master*", you will have to create a *complete fpp file*, i.e. the fpp file must contain material data. This is achieved by running FE-Fatigue, but you may omit running the actual computation (answer "*No*" when asked if you wish to analyze job). If you omit running the analysis, it will be run for you in batch mode later.



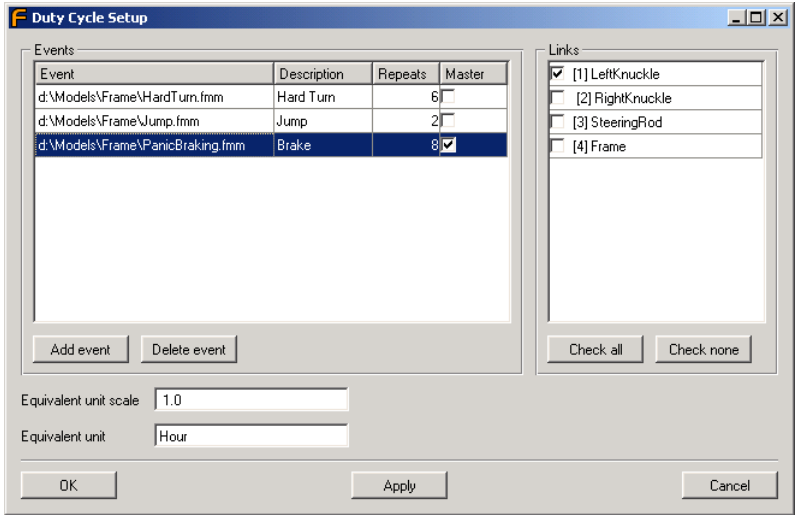
NOTE: Since FE-Fatigue is run in a duty cycle analysis the nCode software needs to be installed for this process to work.

6.10.2 Licensing needs

To be able to run the Duty Cycle analysis, you will need both a license from Fedem Technology and nCode.

6.10.3 Setting up a Duty Cycle analysis

In the Events table, use the Add button to add the events you want to use for your simulation. You may optionally provide a Description. Number of repeats gives the weight of the event. One of the events needs to be selected as *Master*. This event is the one where you have a complete fpp file, i.e. with material data included.



Select the links you want to use in the Links table. Note that the links you select must be present in all events, and they need to have the same ID in all events as well.

Next, set the Equivalent Unit and the Equivalent Unit scale. The Equivalent Unit and scale is the period which consists of the events with their repeats. E.g. in the above picture, 6 hard turns, 2 jumps and 8 breakings is equivalent to 1 hour of operation.

6.10.4 Running Duty Cycle

When running Duty Cycle, four nCode programs are run in succession on each link. First, *fpphcopy* is run to copy the fpp file header from the master model to the other fpp files. Secondly, *FE-Fatigue* is run in batch mode to create a summary for the link in each event. Thirdly, *fefcom* is run to sum up the damages from each event. Lastly, *fef2frs* is run to create a Fedem Results File. This procedure is repeated for all links.



NOTE: All Duty Cycle calculations are performed in a temporary sub-directory named `<TempDir> / fedem<####>`, where `<TempDir>` is the temporary directory defined as explained in [Section 6.2, "Controlling placement of temporary files"](#). However, if the name `<TempDir>` is longer than 120 characters, or contain spaces or (on UNIX) upper case letters, then `C: \Temp` (on Windows) or `/tmp` (on UNIX) is used instead. If the Duty Cycle calculation completes successfully the temporary sub-directory is deleted. However, if the calculation fails the directory is retained such that you may check for any hints to why the calculation failed in that directory.

6.10.5 Viewing Duty Cycle results

When Duty Cycle analysis has completed, you may create a *Time Summary animation* to view the results. The available results are *Damage*, *Log Damage*, *Life* and *Log Life*. See [Section 7.3.1, "Managing animations"](#) for details on how to set up animations.

6.11 Interaction during processing

6.11.1 Simultaneous viewing and processing

If you already have set up graphs and animations (see [Chapter 7, "Postprocessing Results"](#)), they can be viewed during the dynamics analysis. Your graphs and 3D views can be dynamically updated to show the results from the mechanism as they are computed.

Graphs

If graph views are displayed (see [Section 7.2.3, "Showing a graph"](#)) during the dynamics simulation, each curve is continuously updated with values from the solution, reflecting the progress of the dynamics analysis.



TIP: A graph showing the number of iterations for each time step gives a good indication of both the progress of the simulation, and how quickly each time step converges to the specified tolerance.

Animations

If a 3D animation is loaded (see ["Loading animations"](#) in [Section 7.3.1](#)) during the dynamics simulation, the 3D view of the model is continuously updated with the rigid body motion of the link, reflecting the simulation results. The model can also be examined (rotated, zoomed, and so on) during the animation.



TIP: The rigid body animation is an effective and intuitive way to observe the progress of the simulation. When the rigid body animation is displayed, a progress bar also appears in the upper-right corner of the Modeler window.



TIP: A rigid body animation may be started automatically when the dynamics solver starts by enabling a toggle in the Dynamics Solver Setup (see ["Output tab"](#) in [Section 6.5.1](#)).



TIP: Animations can be loaded and closed at any time during the simulation.



CAUTION: With large models and long simulations, the simultaneous visualization and simulation features may use a large amount of system resources.

6.11.2 Stop processing



Solution processes can be stopped at any time by pressing the *Stop All Solvers* button on the Solvers toolbar (or Solve menu).



NOTE: All results created before the stop button is pressed are stored and can be evaluated in the normal manner.

6.12 Deleting results



To delete result files from the dynamics simulation and recovery operations (stress recovery, modes recovery, strain rosette recovery and strain coat recovery), click the *Delete Results* button on the Solvers toolbar (or Result menu).



NOTE: Result files are not deleted until you save or close the model, and files from the link reduction are never deleted.



CAUTION: If you use the *Save As...* command to save a model to a different location, result files from previous solutions are not copied to the new location. Only result files and reduced links from the current analysis are copied. (Applicable only if the "Discard results" and "Discard reduced links" toggles are not set)

6.12.1 Deleting specific results

Results from specific recovery processes can be deleted. This is useful for reclaiming disk space or changing the 'result window' available to the post processing.



The stress recovery results can be deleted by clicking the *Delete Stress Recovery results* button on in the results menu.



The mode shape recovery results can be deleted by clicking the *Delete Mode shape recovery* button on in the results menu.



The strain rosette recovery results can be deleted by clicking the *Delete Strain Rosette results* button on in the results menu.



The strain coat recovery results can be deleted by clicking the *Delete Strain Coat Recovery Summary results* button on in the results menu.



The duty cycle recovery results can be deleted by clicking the *Delete Duty Cycle Recovery results* button on in the results menu.



NOTE: The results files are deleted immediately when performing these actions.

6.13 Automated curve export from multiple result database files

If you need to export curve data from several result database files into a single output file (e.g., if you want curves from one or more Strain Rosette analyses exported to a single RPC-file), that can be done by executing the Fedem Curve Export Utility module after the necessary solver tasks have been completed. This module can only be invoked as a separate command from a terminal window or command prompt.

Start the curve export by issuing the following command:

```
fedem_graphexp -frsFile (fnames) -modelFile (mname) \
  -curvePlotFile (cname) [<options>]
```

where *(fnames)* is a list of one or more frs-files on the form '*<file1, file2, ...>*', *(mname)* is the Fedem model file (*.fmm*) in which the curves to be exported are defined, and *(cname)* is the name of the output file where the curve data is exported. The module also have some other options to facilitate the export process (see [Appendix E.8, "Curve export options \(fedem_graphexp\)"](#) for a complete list of all options for this module).

6.14 Batch execution of solver processes

If you need to run several events for a given model, e.g., when preparing for a Duty Cycle analysis, etc., it is normally most efficient to edit the model file directly in an editor, or by means of scripting, and then run the solver processes in batch mode from a terminal window or command line prompt. To facilitate such batch executions, a set of command-line options is provided, that runs Fedem in a non-graphics and non-interactive mode (see [Appendix E.1, "Fedem UI Options \(fedem\)"](#) for the complete list of command-line options for the Fedem UI).

6.14.1 Batch solving through the User Interface

Start the Fedem UI from the command-line prompt with the following command:

```
fedem -f (mname) -solve (sname)
```

where *(mname)* is the Fedem model file and *(sname)* is one of the keywords *reducer*, *dynamics*, *stress*, *modes*, *straingage*,

straincoat, *dutycycle* or *all*. This will read the specified model file, and immediately launch the specified solver process(es), and then save the model file and the obtained results before the command exits. Thus, this has the same effect as (but is normally faster) the following manual operations:

- Start the Fedem UI interactively in the normal manner
- Open the desired model file
- Launch the desired solver process from the Solvers toolbar
- Exit Fedem with saving of model file

In the same way as when running from the Solver toolbar (or from the Solve menu), it is checked that all links are reduced and up to date, and that dynamics results exist, before a recovery process is started. And if needed, the necessary pre-requisite solver tasks are executed first. Thus, to simply run all recovery types on all links for a given model file, you can just use the option *-solve all* and all required solver tasks will be executed sequentially without user interaction.

6.14.2 Preparing for batch solving on remote computers

Sometimes it is desirable to execute solver processes directly on another computer than the Fedem UI is run on, and which have a separate file system. It is then necessary to create the needed RDB directory structure locally, populated with all the required solver input files, and then manually transfer it to the remote computer for batch execution (see [Section 8.3, "RDB directory structure"](#) for details on the RDB structure). When finished, the whole directory structure can then be copied back onto the local computer for further post-processing in the Fedem UI.



To create a complete RDB structure with input files for all the Fedem solver processes (except FE-Fatigue and Duty Cycle), click the *Prepare for batch execution* button on the Solvers toolbar or in the Solve menu.

The RDB structure required for a certain solver process can also be created by issuing the following command from a terminal window:

```
fedem -f (mname) -prepareBatch (sname)
```

where the meaning of *(mname)* and *(sname)* is as explained above in [Section 6.14, "Batch solving through the User Interface"](#) (except that the keyword *dutycycle* is not available here). Thus, the effect of this command is exactly the same as with the *-solve* option, except that the solver processes themselves are not executed. Specifying *(sname)=all*

with this command is equivalent to using the *Prepare for batch execution* button in the UI. When the command have exited you have the necessary RDB directory structure, which can be transferred to the remote computer for batch execution.

6.15 How to read error messages from the solvers

When a solver process fails to complete a simulation task, or any abnormality occur during the simulation, messages explaining the problem are written to the Output List during the process execution. These messages are prefixed either by "*Error:*", "*Warning:*" or "*Note:*" signifying something about their severity. This should be interpreted as follows:

Error: A problem has occurred that makes it impossible or undesirable to continue the simulation. The simulation is aborted in a controlled manner.

Warning: A problem has occurred that may affect the simulation results, although the simulation itself continues. However, one should be more critical to obtained results when warnings occur, and consider changing the model.

Note: The event causing this kind of message is usually of no significance for the results. The message informs the user that an action has been made to perform a specific task, etc.

When a simulation process aborts with *Error:-*messages, you will also see the message *See <filename.res> for further details*. That means that sometimes (but not always) there are further error messages on the *.res* file explaining the problem. Note, however, that these additional messages are often of a low-level character and harder to understand for the average user. The main rule is that messages printed to the Output List window should be sufficient to understand the scope of the problem.

A solver error will often create several messages on the *.res* file, all relating to the same problem/incident. This typically happens if an exception or error occurs deep inside a program module and from there prints an error message on what is wrong (basing the message on the knowledge available to that module.) As the program trace-back unfolds, more messages on the same problem may be printed, providing more information on the problem as it becomes available. The messages on this higher level can be more user friendly due to more available

information. Some messages on the lowest level will have meaning for the program developers only. It is, however, often necessary to have both the higher level and the lower level messages to get full insight into the problem.

So, to understand the problem quickly, it is therefore helpful to read these messages in reverse order, starting with the last message (the highest level) and ending with the first one (lowest level). Sometimes you may ignore the lowest level messages (those output to the *.res* file only) and still see what caused the problem.

Chapter 7 Postprocessing Results

This chapter introduces the options for postprocessing the results calculated by the various Fedem solvers. You will learn how to set up graphs, curves and animations.

Sections in this chapter address the following topics:

- [Postprocessing environment](#)
- [Graphs](#)
- [Animations](#)
- [Viewing animations](#)

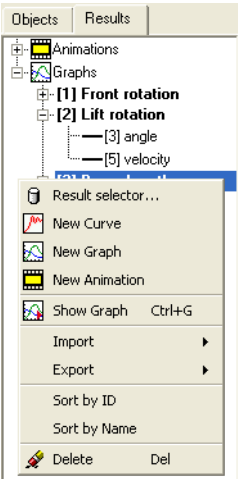
7.1 Postprocessing environment

In Fedem, postprocessing means evaluating the data collected from the mechanism analysis (see [Chapter 6, "Mechanism Analysis"](#)). The options for postprocessing are graphing and animating. You can perform these functions using the *Model Manager Results* list and the tools included on the *Result* menu. Results are displayed in the Workspace area.

Model Manager Results list

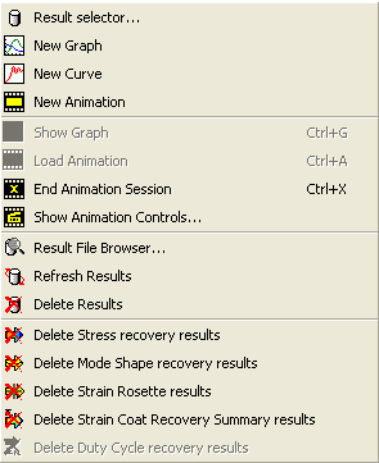
The *Model Manager Results* list (shown at right) displays the list of user-defined result view objects in your model. In addition to the commands available in the "[Result menu](#)" (see below), many shortcut commands can be used to manage results in the *Results* list. The shortcut menu is accessed by right-clicking in the *Model Manager* panel (see figure to the right) and displays commands relevant for the selected object.

Each curve in the *Model Manager Result* list has an icon next to the curve name, representing its legend. The curve color and symbol can be changed in the *Appearance* tab (See [Section 7.2.8, "Appearance"](#)).



Result menu

The Result menu (shown at right) contains commands for creating result views, and managing the result files in your model. Use of these commands is explained in the following sections.



Workspace area

The Workspace area is used to display each of the graphs and animations you create.

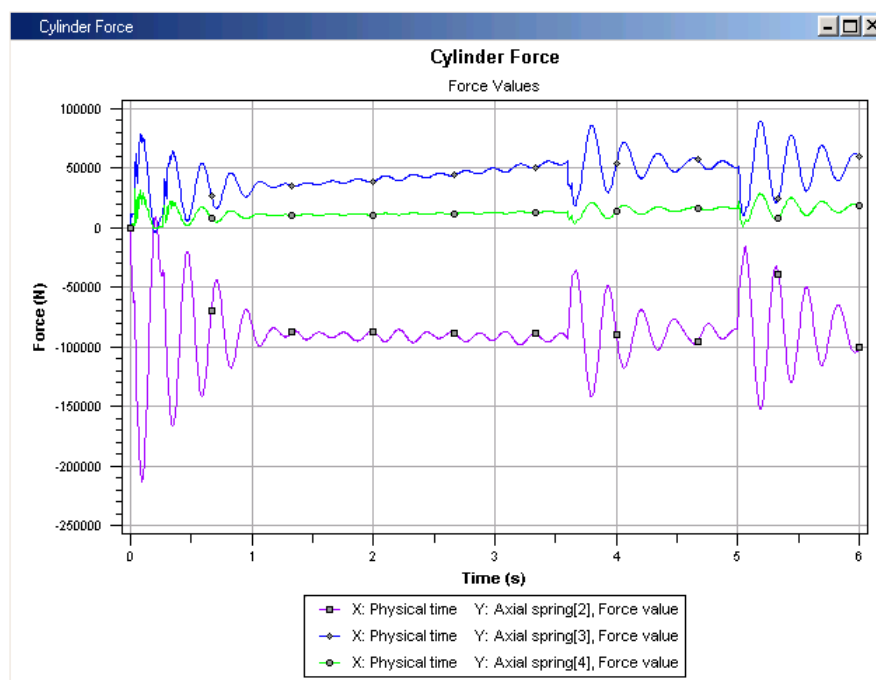
- Graphs can be displayed in individual windows that are labeled with the user-specified description of the graph; these windows are called *graph views*. (To specify a description and display a graph window, see [Section 7.2.4, "Graph properties"](#) and [Section 7.2.3, "Showing a graph"](#).)

- Animations can be displayed in the Modeler window (one at a time).
(To display an animation, see ["Loading animations"](#) in [Section 7.3.1.](#))

7.2 Graphs

To track the progress of any variable during the simulation, you can create two-dimensional graphs of the values. Each graph can contain several curves, enabling comparison of the simulation variables.

You can customize your graphs with titles, axis and data labels, and legends (as shown below).



Graphs can be set up before or after performing the dynamics simulation and other analyses. If you create a graph before performing the simulation, you can observe the values during the simulation, as they are constantly updated (see [Section 6.11, "Interaction during processing"](#)). Otherwise, you can view the entire set of graphed values after the simulation is complete.

Graphs plotting values vs. time will have a time indicator bar present when a time history animation is loaded. This bar will show the current animation time in the graph. This makes it easy to see the correlation between the motion and the graphed values.

7.2.1 Creating graphs and curves

You can create as many graphs as you like, either before or after solving the dynamics simulation. It is recommended that you provide descriptive names for your graphs, as the description is used in the Model Manager Results list to distinguish between graphs. The description is also used as the title of the graph view when the graph is shown in the Modeler window. (To specify a description, see [Section 7.2.4, "Graph properties"](#) below.)

Once you have created a graph, you will need to create curves (plotted data) for the graph and specify descriptions and properties for the curves.

There are two ways to create these items, either by using the *New Graph* and *New Curve* commands, or by dragging results from the *Result Selector* into the *Result* list of the *Model Manager*, as described below.

Graph



To create a graph, select *New Graph* from the Result menu, or right-click in the Model Manager Results list, select *New Graph*. An object titled "New Graph" is listed in the Model Manager Results list.

Curves

To create a curve, complete the following steps:

1. In the Model Manager Results list, select the graph to which you want to add the curve.
2. Select *New Curve* on the Result menu or right-click the graph in the Model Manager Results list, select *New Curve*.
3. Alternatively: If right-clicking an empty spot in the Model Manager Result list, selecting *New Curve*, a new graph with one curve will be made.



An object named "New Curve" is then listed under the current graph in the Model Manager Results list.

Creating curves and graphs by drag and drop

When you need to create several curves and graphs containing results from the results database, or want to inspect several result values without the need to create persisting graphs for all of them, it is convenient to use the drag and drop method to create and sort curves and graphs.

To create curves or graphs by drag and drop, right-click in the *Result* list of the *Model Manager* and select *Result selector...* A dialog will pop up (shown to the right).

Select the result you want to plot, and drag it from the *Result Selector* and onto the *Result* list of the *Model Manager*. If you drop it on an existing graph, it will be created as a curve in that graph, otherwise a new graph will be created with the selected results as a new curve.

If the graph receiving the new curve is visible, the new data is automatically loaded and displayed.

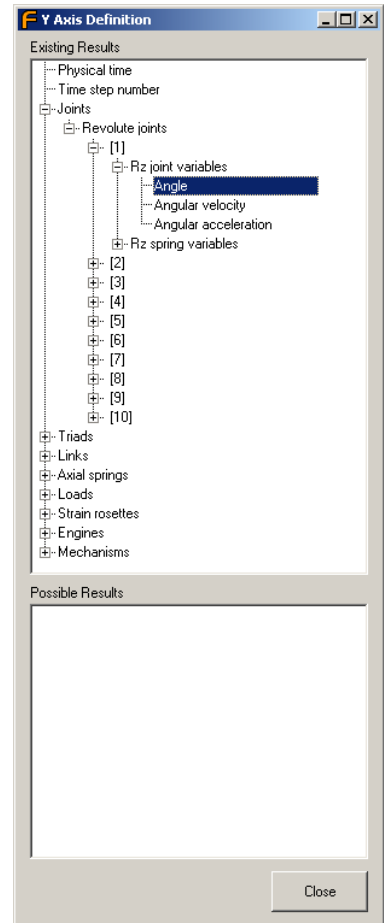
For more detailed information about the Result Selector, see "[Selecting RDB results](#)" in [Section 7.2.5](#).

Repeat a curve definition for every object in the model

In some cases it is useful to look at some particular result from all the objects of a certain type in the model. E.g. to find the triad with the highest forces in the model, or to look at the acceleration levels all over the model.



This can be achieved in a convenient way using the command labeled *Repeat curve for all objects*. This command is available in the right click menu of the *Result* list of the *Model Manager* when right clicking a curve. This command repeats the curve definition of the curve selected, and uses it on all the objects of the same type in the model. It also automatically assigns a color to the curves depending on the ID number



of the object. The colors assigned will be black-blue-cyan, where low ID numbers makes the curve get a blackish color, while higher numbers will turn the curve blue or cyan.

Normally it is only the object used for the y-axis value that is cycled, but if the curve being repeated uses the same object both for the x-, and y-axis they are cycled together.

7.2.2 Moving curves to a new graph

To move curves from one graph to another, simply select the curves and drag them to the new graph. This is done by pressing and holding the left mouse button while the mouse cursor is above the curves to move, then move the mouse to the target graph and release the mouse button.

7.2.3 Showing a graph



To display graphs, select one or more graphs or curves in the Model Manager Results list, right-click and select *Show Graph* on the shortcut menu (or select *Show Graph* on the Result menu). The selected graph views opens in the Workspace area displaying the graphs.

7.2.4 Graph properties

Now that you can create a graph and its curves, you need to specify properties for the graph. To display the properties in the Property Editor panel, select the graph in the Model Manager Results list. The properties for the graph are displayed in the Property Editor as shown below.

- ❶ *Description* – An optional user-specified description that is displayed as both the graph name in the Results list, and the title of the associated graph window in the Workspace area. It is recommended that you use a descriptive name or phrase to distinguish between similar graphs. This name is not included in the graph view itself.
- ❷ *Title* – You can specify a title for the graph that is displayed in the graph view.

- ③ *Subtitle* – You can provide a subtitle that is appended underneath the *Title* in the graph view.
- ④ *X-Axis Label* – You can provide a label for the abscissa that is displayed in the graph view.
- ⑤ *Y-Axis Label* – You can provide a label for the ordinate that is displayed in the graph view.
- ⑥ *Grid* – You can specify the type of grid (coarse, fine, or no grid) for the graph by selecting from the **Grid** pull-down list.
- ⑦ *Legend* – You can add a legend to the graph by enabling the **Legend** option. The legend names you specify for each curve are used in the legend (see [Section 7.2.5, "Curve properties"](#) below).
- ⑧ *Autoscale* – Enabling this option automatically scales the x- and y-axes to accommodate new curves as they are added to the graph (or when opening the graph view). Disabling Autoscale allows you to fully control the x- and y-axis range.
- ⑨ *Start time, Stop time, Use time interval* - You can enable the **Use time interval** toggle to specify a time interval for loading the graph. Only the RDB data that fall within the specified time interval are loaded when the graph is opened or exported. This only affects curves that are plotting simulation results. Imported curves, or curves plotting an internal function are not affected by these settings.

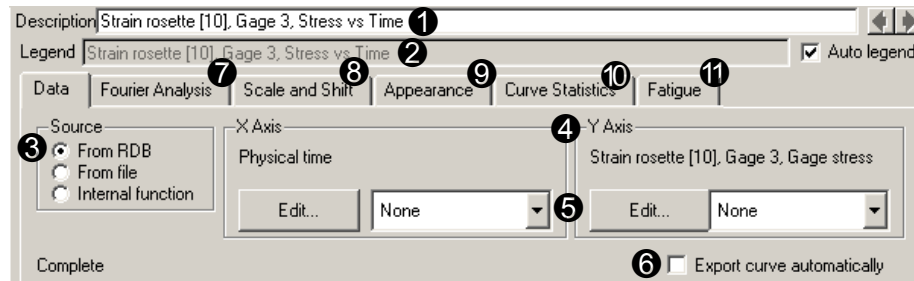


NOTE: When you make changes to graphs or curves in the Property Editor, the changes are updated dynamically in the graph view.

7.2.5 Curve properties

Having completed the specification of a graph, you want to set the properties of its associated curves. To display the curve properties in the Property Editor panel, select the curve in the Model Manager Results list. The curve information is organized under six tabs: Data, Fourier Analysis, Scale and Shift, Appearance, Curve Statistics and Fatigue. The Data tab is

used to define the x- and y-values to be plotted and is described below. The other five tabs are discussed in the subsequent sections.



- ① **Description** – this is used as the name for the curve in the Model Manager Results list. The description is updated automatically when changing the x- or y-axis definitions, unless it has been edited manually in the mean time. This name is not used in the graph view.

TIP: If you have manually edited the description field of a curve, you can at any time return to the auto-generated description by deleting it completely.

- ② **Legend** – if you choose not to supply a name for the curve, you can enable **Auto Legend** to use a combination of the x- and y-axis labels for the legend. When **Auto Legend** is used, the legend will be always be equal to the auto-generated description of the curve.

NOTE: The **Legend** option must be enabled in the Graph Property Editor to display the legend in the graph view (see [Section 7.2.4, "Graph properties"](#) above).

- ③ **From RDB, From file, Internal function** - the data plotted in a curve originates from either the results database (RDB), an external file (DAC, ASCII or RPCIII) or an internal Fedem function. The panel shown above is that of a "From RDB" curve. The panels of the "From file" and the "Internal function" curves are discussed below (see ["Creating curves from file"](#) and ["Creating curves from a function"](#)).

- ④ **X Axis / Y Axis** – to define the variables used for the x- and y-coordinates in a "From RDB" curve, click the **Edit** button. The Results Database Selector is displayed allowing you to pick the wanted results (see ["Selecting RDB results"](#) below for details). Alternatively one may also right-click a curve in the Model Manager Results list and then select *Edit X Axis...* or *Edit Y Axis...*

TIP: While editing the x-axis variable you can click **Apply** in the Results Database Selector, then click the **Edit** button for the y-axis, and start selecting for the y-axis.

- ⑤ **Result Operation** – the pull-down menus next to the *Edit* buttons list mathematical operations (such as extracting the x-component or

computing the length of a vector) related to the result quantities selected for the x- and y-axis. If the result quantity selected in Step 4 above is a *Position matrix* (of either a Triad or Link), the menu will also contain several angular quantities which can be derived from the position matrix. See ["Derived angular quantities from position matrices"](#) below for further details on these items.



NOTE: If you make a change to the Result Operation, the curve changes dynamically in the graph view.

- ⑥ *Export curve automatically* – toggles whether this curve shall be exported automatically by the solver to the file specified in Dynamics Solver Setup. See ["Output tab"](#) in [Section 6.5.1](#).
- ⑦ *Fourier Analysis* tab -allows you to perform a Fast Fourier Transformation of the curve data. See [Section 7.2.6, "Fourier analysis"](#).
- ⑧ *Scale and Shift* tab - allows you to apply scaling and shift to the curve data. See [Section 7.2.7, "Scale and Shift"](#).
- ⑨ *Appearance* tab - allows you to change appearance of individual curves. See [Section 7.2.8, "Appearance"](#).
- ⑩ *Curve Statistics* tab - allows you to extract statistical properties for individual curves. See [Section 7.2.9, "Curve Statistics"](#).
- ⑪ *Fatigue* tab - allows you to assess fatigue properties based on the curve data. See [Section 7.2.10, "Fatigue calculation from standard S-N curves"](#).



NOTE: The curve is not displayed in the graph view until you have fully defined the curve variables (indicated by the word "Complete" appearing at the bottom of the Property Editor panel). The word "Incomplete" appears at the same location until the curve is properly defined.



TIP: Once you have created and fully defined curves, you can select them directly in the graph view window, or clicking a curve in the **Legend**. When selected, curves are highlighted in red.



TIP: You can drag a curve from one graph and drop into another in the Model Manager Results List.

Selecting RDB results

To select the results to plot as x- and y-coordinates for a curve, complete the following steps:

1. In the Model Manager Results list, select the curve you want to edit. Its properties are displayed in the Property Editor.
2. In the Property Editor, click the *Edit* button for the X-Axis, or right-click on a curve in the Model Manager Results list and select *Edit X Axis...* or *Edit Y Axis...* The Results Database Selector window (shown at right) is displayed.
3. Select a mechanism element from the *Existing Results* list or the *Modeler* window.

NOTE: If the mechanism analysis have been performed, variables for the selected element are listed in the *Existing Results* list. If you have not yet performed an analysis, variables are listed in the *Possible Results* list.

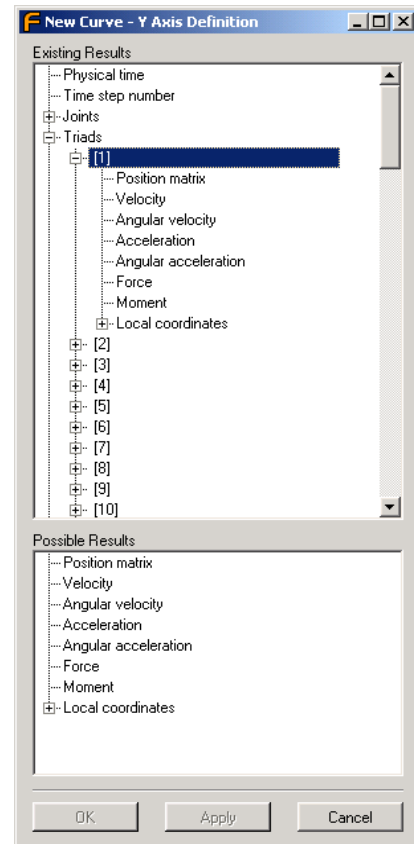
4. Select a variable to be used as the x-coordinates from either the *Existing Results* or *Possible Results* lists, and click *OK* to close the panel or *Apply* to continue variable selection.

CAUTION: Some of the variables listed in the *Possible Results* list may not be present in the results database. For example, a joint may or may not have a spring or damper attached at each DOF, but in the *Possible Results* list all possible springs and dampers are listed - one for each joint DOF. If such a nonexistent variable is selected, the associated curve does not appear in the graph view.

NOTE: Variables such as Physical Time and Time Step Number are not associated with a mechanism element, and are listed independently in both lists.

NOTE: Categories such as Revolute Joints or Z-Rotation Joint Variables cannot be selected as variables; only those items in the expanded lists (such as Angular Deflection) can be selected for use as variables.

5. Repeat steps 2 through 4 to select the Y-Axis variable.





TIP: To easily find out what result quantities, if any, that already have been plotted for a given mechanism object, just select the object and inspect the *Topology* panel (see [Section 2.5.4, "ID and Topology panel"](#)). The curves plotting quantities in the selected object are then listed under the *Plotted by:* heading.

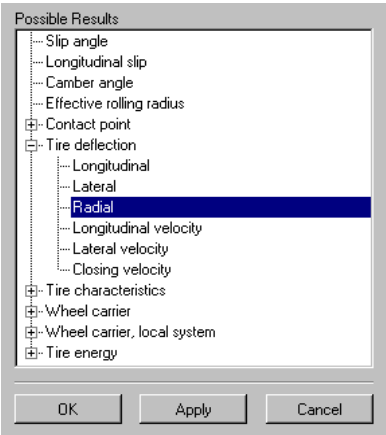
Derived angular quantities from position matrices

Totally 10 derived quantities may be plotted for a *Position matrix* result item (as shown to the right). The *Euler Angle* quantities (top three) are the indicated angles computed from an imagined incremental rotation from the global coordinate system axes to the orientation represented by the position matrix. The bottom three items are similar quantities computed from a Rodriguez parameterization of the incremental rotation. See the Fedem R5.0 Theory Guide, *Section 2.3, "Finite Rotation"* for the definition of these angular quantities.

- Euler Angle ZYXX
- Euler Angle ZYXY
- Euler Angle ZYXZ
- Position Length
- Position X
- Position Y
- Position Z
- Rotation Angle X
- Rotation Angle Y
- Rotation Angle Z

Results from the tire models

When you have one or more tires in the mechanism model, you may plot a variety of result quantities associated with each tire. A few of these quantities, the wheel carrier forces and tire energies, are computed by the Fedem Dynamics Solver based on the tire model outputs. All remaining tire quantities listed in the *Possible Results* list for a tire are direct output from the tire models.



However, not all of these quantities are available for all tire models supported in Fedem: Variables in the *Tire characteristics* group are available for MF-Tyre only. In the *Tire deflection* group, only the *Radial* variable is available for all three models. The remaining variables in this group are available for MF-Tyre and SWIFT only.

Plotting internal control variables

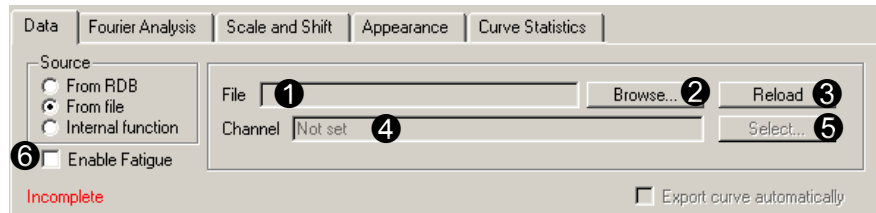
When you plot internal control variables, you will probably discover that some of the control lines don't have any results. The reason for this is that more than one control line share the same control variable, and the results appear on only one of these (usually the one with the lowest ID). This situation will occur when one element's output is used as input to more than one other element.



TIP: Have the control system editor open during curve result selection. If you select a control line in the RDB Selector, that line will also be highlighted in the control system editor. Vice versa, if you select a control line in the control system editor, that control line will be selected in the RDB Selector. That way you can easily see which control line you will have to plot to get the variable you want.

Creating curves from file

A curve can be created from an external file (DAC, ASCII or RPC) by selecting the *From file* option on the Property Editor panel's Data tab. The panel used to define such a curve is shown below.

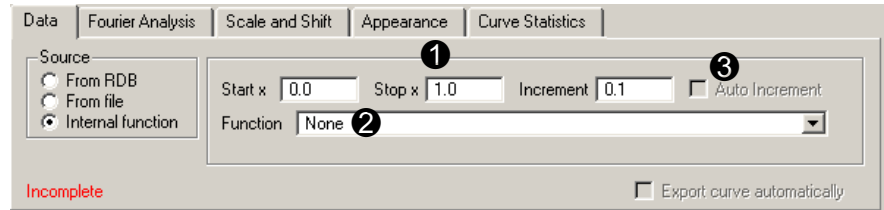


- ❶ *File field* - The selected file will be listed here.
- ❷ *Browse button* - Opens a file dialog.
- ❸ *Reload button* - If your data source file has changed, you can click this button to reload the curve into the viewer.
- ❹ *Channel field* - The name of the selected channel will appear here. (Only applicable for MTS RPCIII and multi-column ASCII files.)
- ❺ *Select button* - If you imported an MTS RPCIII or multi-column ASCII file, you will have to select which channel to extract data from. A dialog for doing so will appear when clicking this button.
- ❻ *Enable Fatigue* - You may chose to enable the *Fatigue* tab and run fatigue damage calculations on the loaded curve.

It is also possible to import multiple curves into an existing or new graph. See "[Importing Curves](#)" and "[Importing Graphs](#)" in [Section 7.2.13, "Importing Curves and Graphs"](#).

Creating curves from a function

A curve can be created from one of the functions defined in your model by selecting the *Internal function* option on the *Property Editor* panel's *Data* tab. The panel used to define such a curve is shown below.



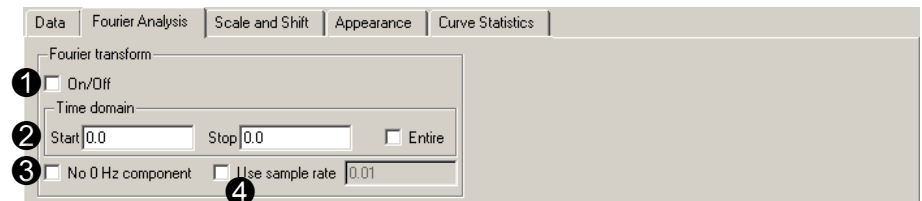
- ❶ *Start x, Stop x, Increment* - Sets the end points for the domain to plot, and the rate at which the function is sampled.
- ❷ *Function* - The function to be plotted is chosen from this pull-down menu. (The menu lists all functions currently in your Fedem model.)
- ❸ *Auto increment*- Most functions have the option to have the resolution set automatically. If used, Fedem will determine which points are sufficient to describe the function and plot those. In that case the specified increment is not used.



NOTE: If plotting a Poly-line function with *Auto Increment* set, the domain, *Start x* and *Stop x*, is not used either. In this case, the domain is automatically adjusted to fit the curve data.

7.2.6 Fourier analysis

Options for performing a Fast Fourier Transform (FFT) of the curve data are found on the *Fourier Analysis* tab of the *Property Editor* panel. The *Fourier Analysis* panel is shown below.



- ❶ *Fourier transform On/Off* - When toggled *On* the plotted curve is replaced by its discrete Fourier transform (the Fourier transform is a representation of the curve in the frequency domain). The value plotted is the magnitude of the transform. Note that the scale and shift parameters specified for the curve (see [Section 7.2.7, "Scale and](#)

Shift") are applied to the curve data *before* the transform is computed. You may not scale or shift the transformed curve.

- ② *Start, Stop, Entire* - What part of the curve to transform is specified using the Time Domain options. If *Entire* is toggled on, then data for the curve's entire domain is used. If *Entire* is toggled off, then a start and stop time may be set in the fields labeled *Start* and *Stop*.
- ③ *No 0 Hz component* - The arithmetic mean of the original curve data is reflected in the transform's value at 0 Hz (the transform's first point). To facilitate transform analysis you might want to "cancel out" the 0 Hz component by using this option.
- ④ *Use sample rate* - The sample rate used in the transform is by default equal to that of the curve data. However, if the curve has a non-constant sample rate the transform will fail. In such cases the wanted sample rate may be input using this field.



CAUTION: The Fourier transform needs to be recalculated each time points are added to the curve. Consequently, if curves are plotted while the Dynamics solver is run, then transforming these curves will increase the CPU load during solving.



TIP: A good reference on the theory of Fourier transforms is: W. Rudin, "Real and Complex analysis", McGraw-Hill, 1974.

7.2.7 Scale and Shift

Options to scale and shift the curve data are found on the Scale and Shift tab of the Property Editor panel. You may apply scaling and a shift on the curve data independently in the x- and y-directions. The Scale and Shift panel is shown below.

- ① *Scale* - Scale factor applied to the x- or y-axis values.
- ② *Shift values to zero out first value* - For the x-axis; shift the curve horizontally such that its first x-value becomes zero. For the y-axis; shift the curve vertically such that its first y-value becomes zero.
- ③ *Additional shift* - Additional horizontal/vertical shift (i.e., in addition to the zero-out operation, if applied) in the curve's x/y-values.



NOTE: Scaling the y-axis of a curve will affect the results of fatigue damage calculations (See [Section 7.2.10, "Fatigue calculation from standard S-N curves"](#)).

7.2.8

Appearance

The Curve appearance can be altered by selecting the Appearance tab in the curve's Property Editor panel. The associated panel is shown below.

- ❶ *Curve Type* – You can select *Lines*, *Dots*, or *Invisible* from the *Curve Type* pull-down list.
- ❷ *Curve Color* – Use the drop-down menu to select a different curve color. You may either select one of the pre-defined colors, or create a new by selecting *more...* at the bottom of the drop-down menu.
- ❸ *Symbol type*– You can select a symbol (cross, circle, or triangle, etc.) to display on the curve.
- ❹ *Symbol Size* – Use the spin box to control the size of the symbols.
- ❺ *Num. symbols* – Use the spin box to control the number of symbols shown on the curve, or select *All* to show a symbol on all points of the curve.



NOTE: The number of symbols entered is the maximum number of symbols shown on the curve. However, as symbols are only shown on existing data points, there may be fewer symbols shown than requested.

7.2.9

Curve Statistics

On the Curve Statistics tab you can display different statistical properties of a curve. The associated panel is shown below.

- ❶ *RMS* - The Root Mean Square value, found from

$$y_{rms} = \sqrt{\frac{\sum_{i=1}^n y_i^2}{n}}$$

- ❷ *Mean* - The Mean value, $\bar{y}$.

- ❸ *Std.dev.* - The Standard Deviation (biased), found from:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n}} \quad \text{where } \bar{y} \text{ is the mean}$$

- ❹ *Integral* - By the Trapezoid rule.

- ❺ *Max* - The overall maximum of the y-values in the curve data set.

- ❻ *Min* - The overall minimum of the y-values in the curve data set.

- ❼ *Calculate button* - Press this button to retrieve the statistical values.

- ❽ *Use scaled/shifted* - Toggling on this button will make the calculation take into account any scale and shift values defined in the Scale and Shift tab (see [Section 7.2.7, "Scale and Shift"](#)), and in effect do the calculations on the curve as it is shown in the graph view. If the button is toggled off, the unprocessed data will be used.

- ❾ *X Axis Domain* - Toggle the *Entire* button on to use all the data points on the curve, or specify a start and a stop value. If you specify an interval, two vertical lines at the start and stop values, will appear in the graph view when you click the *Calculate* button

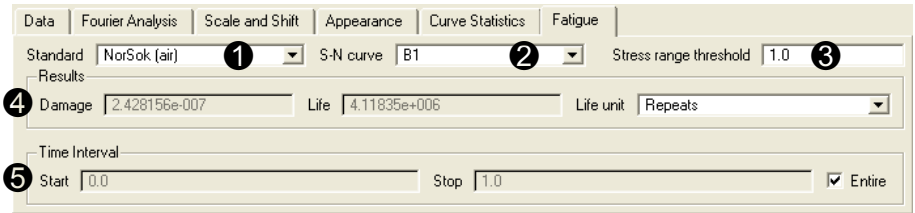
7.2.10 Fatigue calculation from standard S-N curves

Options to assess fatigue results based on plotted stress histories are found in the *Fatigue* tab of the *Property Editor* panel. This tab is visible only when fatigue calculation is applicable, which means that the curve must be plotting a stress quantity computed by the *Strain Gage Recovery* module, vs. *Physical Time*. The stress quantity can either be *Gage stress* from a *Strain rosette* or the *Signed Abs Max* value from the *Stress tensor* in a *Strain rosette*.

Fatigue calculation is also applicable for curves loaded from file when the *Enable Fatigue* toggle is enabled.

The Fatigue tab is shown below. Here, you may select different standard S-N curves to base the damage calculation on, specify a time interval for the damage calculation, and evaluate the equivalent life in days, hours or

repeats. For details on how the damage is calculated from a given time history response, see the Fedem R5.0 Theory Guide.



- 1 **Standard** - Select the fatigue standard to use for your fatigue calculations.
- 2 **S-N curve** - Use the drop-down to select a S-N curve from the selected standard.
TIP: The S-N curve standards listed in the Fatigue tab are defined in the file `sn_curves.fsn` located in the installation directory of Fedem. The syntax of the S-N curve definitions is description in the header of this file, and it is possible to add your own S-N curve definitions to that file.
- 3 **Stress range threshold** - Stress ranges with magnitude below this threshold are ignored in the stress cycle counting (rainflow analysis).
- 4 **Damage, Life, Life unit** - The Results frame displays the damage results for the plotted stress history. The life is displayed in days, hours or repeats, depending what you select from the *Life unit* drop-down.
- 5 **Start, Stop, Entire** - You can specify what part of the curve to use for fatigue calculations by setting the *Time Interval* options. If *Entire* is toggled on, data for the entire domain of the curve is used. If *Entire* is toggled off, the time interval is specified by the *Start* and *Stop* fields.

NOTE: If the y-axis of the curve is scaled, the scaling factor will be applied to the damage calculations too (See [Section 7.2.7, "Scale and Shift"](#)).

7.2.11 View control

In graph views, you can manipulate the display using the Zoom and Pan toolbar shown below. (For use of these commands, see [Section 2.7.3, "Zoom and Pan"](#).)



You can also use the Dynamic Pan  (F1) and Dynamic Zoom  (F2) commands, in the same way as they are used in the Modeler window (see [Section 2.7.1, "3D Navigation"](#)).



TIP: The following tips are useful for viewing graphs:

- If the quantity of points in your graph is very large, dynamic panning and zooming can take a long time; use the commands on the Zoom and Pan toolbar instead to speed up graphic performance.
- You can use the **Zoom Window** command to zoom in on a small area, enabling you to see the curves in that area with greater detail. This command is turned on whenever the Z key is pressed while viewing a graph.
- You can use the **Zoom Window With Autoscale** command to zoom in on an area. The contents inside the zoom rectangle will be scaled to fit inside the graph window. This command is turned on whenever the X key is pressed while viewing a graph.
- To adjust the graph axes so that the entire curves fit into the graph view, click the **Zoom All** button, or press **F5**.



7.2.12 Export of Curve Data

Curve and Graph objects in Fedem can be exported to files for further processing in external software. We distinguish between exporting graphs and exporting curves.

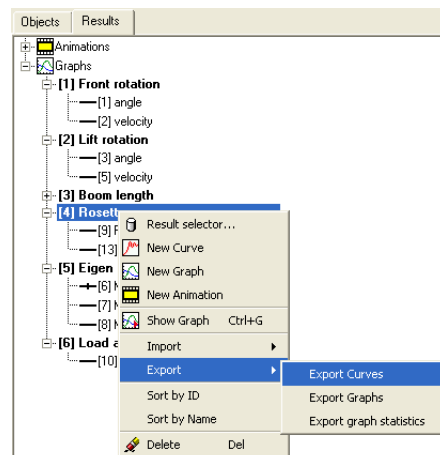
Curve export

When exporting one or more curves, each curve is written to a separate file. The file format can be either Single Column ASCII, nCode DAC or MTS RPCIII time history file.

To export curves, select the curve or curves you want to export in the Model Manager Results List, right-click and select Export -> Export Curves. A file dialog will pop up. If you have selected only one curve, you can select location and file name of the exported curve. If you have selected several curves, you must select a directory to export to. If you select a graph, all its curves will be exported.

When you select a directory to export to, the files will be given names automatically. The file name will be on the form:

$G_<OwnerGraphID>_C_<CurveID>_<CurveDescription>.<Format>$





TIP: Curves that plot result data from the Dynamics Solver may also be exported automatically when the solver has finished (see "Output tab" in Section 6.5.1).



NOTE: The exported data is equal to the results from the settings in the Fourier Analysis tab and the Scale and Shift tab. If you want to export unprocessed data, go to these tabs and set all the values back to default (see Section 7.2.6, "Fourier analysis" and Section 7.2.7, "Scale and Shift").

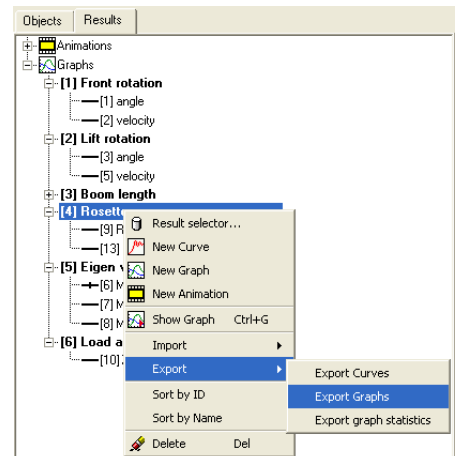


CAUTION: When exporting to nCode DAC or MTS RPCIII, Δx needs to be constant across the entire data set. Using either Physical Time with constant time step size, or Time Step Number will satisfy this requirement.

Graph export

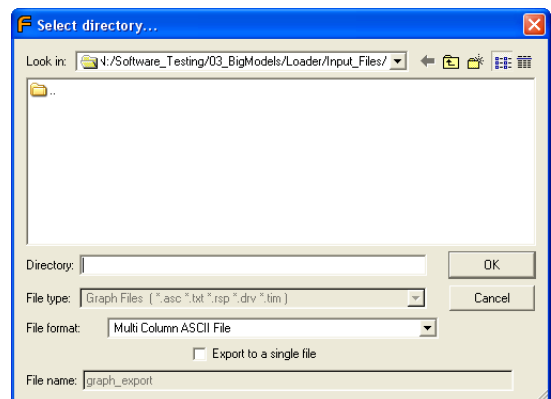
When exporting one or more graphs, each file exported will contain several curves. The file format is either Multi Column ASCII or MTS RPCIII time history file.

Select the graph or graphs you want to export in the Model Manager Results list, right-click and select Export -> Export Graphs. A file dialog will pop up, with slightly different appearance depending on what you have selected.



If you have selected one graph, or a collection of curves belonging to the same graph, the file dialog will let you select directory, file name and format for the exported graph file. If you selected only a subset of curves from a graph, only the selected curves will be written to the graph file.

If you have selected several graphs and/or curves belonging to different graphs, the file dialog will let you specify a directory to write the files to and the file format. It also gives you the option to write the entire selection to a single file. In that case you may also specify a file name.





CAUTION: When exporting graphs, all curves in the selection must have equal x-axis definitions, both in terms of number of data points and increment Δx . When exporting to Multi-Column ASCII, the x-axis values of the exported graph are thus set identical to those of the first curve in the selection. Subsequent curves having a lower resolution in their data sets will then be interpolated, where needed; if they have a higher resolution, some data points will be omitted in the exported graph. When exporting to MTS RPC III, the constant increment Δx for the graph is selected as the smallest increment between two data points among all the selected curves. Curves having a lower resolution than this increment will then be interpolated, where needed.



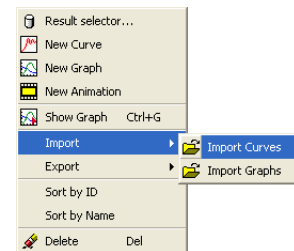
TIP: To export a single curve or graph, you may also use the **Export** and **Export Object...** item in the **File** menu, after selecting the desired curve or graph.

7.2.13 Importing Curves and Graphs

Curve data can be imported into Fedem both as single curves, and complete graphs.

Importing Curves

To start importing curves, right-click with your mouse on a graph, a curve or on an empty spot in the Model Manager Results list and select *Import* and *Import Curves*. In the file dialog that pops up, select one or more files you want to import. One curve is created from each file you have selected.



If you right-clicked on a curve or a graph, the curve data will be imported into that graph.

Importing Graphs

To start importing graphs, right-click with your mouse anywhere in the Model Manager Results List and select *Import* and *Import Graphs*. In the file dialog that pops up, select one or more graph files you want to import. One graph is created for each of the selected files, containing one curve for every channel in the file.

7.2.14 Exporting to picture files

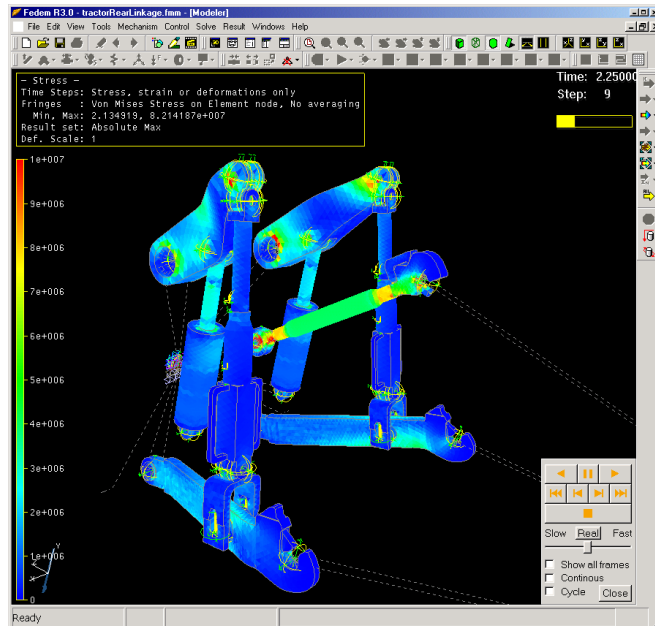
You may also want to export a graph view to a picture file. This is accomplished by first selecting the graph window you want to export and then selecting *Export* and *Export View* from the *File* menu. You may choose to output in either BMP, JPEG or PNG file format. See [Section 2.10.2, "Exporting"](#), for more about Fedem's export capabilities.

7.2.15 Printing graphs

When a graph window is active, you can send its contents directly to a printer for printing. Select the printer symbol on the tool-bar, or the *Print With Setup* command in the *File* menu. When selecting the latter, you will be able to select which printer to use, paper format and so on.

7.3 Animations

Fedem animations are used to visualize the motion or the structural results of your model in an intuitive and life-like fashion. Fedem is able to utilize link motion, link deformation and link color contours to visualize your results and help you understand them.



You can create several Animation objects and define different options for each animation. The created animations can then be loaded into the Modeler window one at a time

Animation objects can be set up before or after performing the dynamics simulation and other analyses. If you create a certain type of animation and load it into the modeler before performing the simulation, you can observe the mechanism motion during the simulation as it is constantly

updated (see [Section 6.11, "Interaction during processing"](#)). Otherwise, you can view the entire animation after the simulation is complete.

When an animation is loaded it can be controlled using the Play Panel (see [Section 7.4.1, "Play panel"](#)) and the Animation control window (see [Section 7.4.2, "Animation controls"](#)). The Play panel is displayed in the lower right corner of the Modeler window (as shown above) when loaded, the Animation control window can be activated when needed.

Graphs plotting values vs. time will also be animated in the sense that a time bar showing the time of the current animation frame is shown. This time bar is only present as long as the animation is actually showing a time step.



NOTE: You can set up animations at any time and load them; however, nothing will appear unless the appropriate results are present.

7.3.1 Managing animations

Creating animations

You can create as many animations as you like. It is recommended that you provide descriptive names for your animations (Stress, Strain, or Eigenmodes, and so on) as the description is used in the Model Manager Results list to distinguish between the animations.



To create an animation, select **Animation** on the Result menu or right-click in the Model Manager Results list, select **Create**, then **Animation** on the shortcut menu.

Loading animations

Once the animation is created, you can show it in the Modeler window by performing the following steps:



1. To open the Modeler window, click the **Show Modeler** button on the Windows toolbar (or Windows menu). The Modeler window opens in the Workspace area.



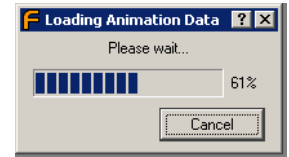
2. Select the animation in the Model Manager Results list.
3. Click the **Load Animation** button in the Property Editor panel, or right-click the animation in the Results list and select **Load Animation** on the shortcut menu.

The animation is loaded and both the *Contour Legend* and *Play Panel* are displayed in the Modeler window (see [Section 7.4.1, "Play panel"](#)).



TIP: Changes to animations cannot be updated instantaneously; however, after each change you can reload the animation to include the updates.

Loading an animation with contours and/or deformations for large FE models may take a considerable amount of time. However, it is possible to cancel the animation loading process at any time, by using the **Cancel** button of the progress dialog that appears while the animation is loading (shown to the right).



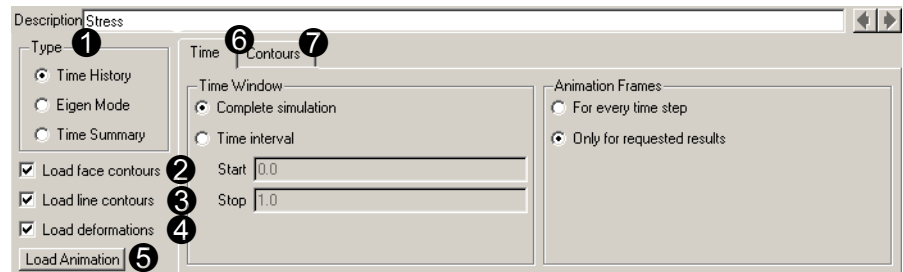
Closing animations



You can end an animation session at any time by clicking **Close** on the Play Panel or selecting **End Animation Session** from the Result menu.

7.3.2 Animation properties

To display the properties of an animation in the Property Editor panel, select the animation in the Model Manager Results list. The properties for the animation are displayed in the Property Editor as shown below.



- ① **Type** – The animation type determines the main type of results you want to animate. See ["Animation types"](#) below.
- ② **Load face contours** – If enabled, the values specified on the Contours tab are loaded and shown as color contours on the element faces of the mechanism assembly. See also ["Contours tab"](#) below.
- ③ **Load line contours** – This option controls whether or not contours are loaded and displayed on the FE-mesh lines. The values displayed will be the result selected on the Contours tab but always averaged on the nodes. See also ["Contours tab"](#) below.



NOTE: Element-to-Node averaging is not yet supported, and even if Load line contours is toggled, no contour colors will be shown on the mesh lines when the chosen Result class is Element.

- ④ *Load deformations* – If enabled, deformation results from the stress or mode shape recovery—depending on the animation type selected—are loaded, if such results are present. See [Section 6.6, "Stress recovery analysis"](#) and [Section 6.7, "Mode shape recovery analysis"](#).
- ⑤ *Load Animation* button – If you make changes to the animation properties, you can reload the animation at any time by clicking this button.
- ⑥ *Time* tab – Options to control what part of the time history to load. See ["Time tab"](#) below.
- ⑦ *Contours* tab – Options to control what result values to load and display as color contours. See ["Contours tab"](#) below.

Animation types

The following three animation types are available.

Time History– Time history animations are used to animate time history results. This can be the rigid body component of link motion calculated in the dynamics solver, or recovered stress, strain and deformational part of the link motion.

Eigen Mode– Eigenmode animations are used to visualize the shape of a system eigenmode at a specific point in time. To provide this visual interpretation, the mode shape is used to create an animation of the mechanism oscillating as if the eigenmode was excited.

The time displayed along with the progress bar shown during an eigenmode animation is the time elapsed during the oscillating motion. The legend text displays the point in time for existence of the animated eigenmode during the dynamics solution.

Unless a mode shape recovery has been run, you can only animate the rigid body component of the eigenmode for each link. Note that the partitioning of the total mode shape into a rigid body approximation plus a deformable component depends on the chosen computational coordinate system of the link (see [Section 4.1](#) in the Fedem R5.0 Theory Guide). In particular, if the center of rotation for a mode is far from the origin of the link coordinate system, the deformable component may be misleading. However, the sum of the deformable and rigid body components will always be correct.

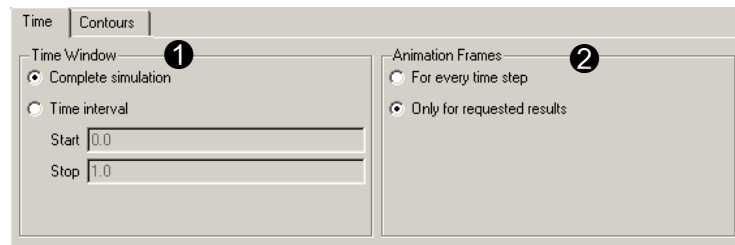
Time Summary – Time summary animations are used to show accumulated structural results within a specified time interval. These results are produced by running the Strain Coat Recovery Summary and,

if available, nCode FE-Fatigue or Duty Cycle. Examples of such results are the maximal principal stress reached within a time interval, or damage accumulated within the same interval.

Usually only one frame with accumulated results from the complete simulation time are produced and displayed. It is, however, also possible to show an animation of how the results are accumulated by running the Strain Coat Recovery Summary several times with different stop times.

Time tab

If you selected *Time History* as the animation type in the Property Editor, you can choose to show the entire simulation (default) or a specific time interval. To change the interval, you must specify the properties on the Time tab (shown below) in the Property Editor, before loading the animation. The Time tab is not available for *Eigen Mode* and *Time Summary* animations.



- ① **Time Window** – You can choose to display the entire duration of the simulation or a specific time interval. If you select **Time Interval**, enter values for the interval's **Start** and **Stop** time.
- ② **Animation Frames** – These options are useful if you specified different time steps for the dynamics simulation and the stress recovery calculation (see [Section 6.6.1, "Stress recovery options"](#)). Enabling **For every time step** loads deformation results and color contour values for all the time steps calculated within the time window specified.



NOTE: Enabling **For every time step** shows continuous motion; however, the stress color contours may flash on and off if the calculation intervals are different. If you select **Only for requested results**, then only intervals shown are those at which the stress is calculated; both stress contours and motion appear continuous.

Contours tab

If you selected *Load face contours* and/or *Load line contours* in the Property Editor, you can display color contours on the mechanism

assembly during the animation (for *Time History* and *Time Summary* animations only). You must then specify the values to be used for the color contours on the Contours tab (shown below) in the Property Editor panel, before loading the animation. The Contours tab is not available for *Eigen Mode* animation.

- ① **Contour Value** – These options are used to select the results that will be displayed as color contours.
 - *Result class* – this option allows selection of the FE entities for which you want to show color contours. You can select either nodes, elements (one single result for each element), or element nodes (one result for each node within an element). This selection controls the options for the *Result* and *Value* settings (see below).
 - *Result* – this option allows you to specify which type of result to display in the animation: stress, strain, deformation, etc. The options available depend on the *Result class* setting, and on the actual results currently available in the results database.
 - *Operation* – this option allows you to specify the scalar value to extract from the selected result, such as component selection, von Mises, principle values, and so on. The operations available depend on both the *Result class* and *Result* settings.



NOTE: There are two ways of animating some stress and strain measures. E.g. a von Mises stress animation may be set either by choosing "Result - Stress" and "Value - von Mises", or by choosing "Result - von Mises" directly. The reason for this is the option to recover the desired derived stress/strain measures directly (see [Section 6.6.1, "Stress recovery options"](#)). In the above example, if the entire stress tensor was recovered, the first animation definition would be correct. If only the von Mises stress measure was recovered, the second definition should be used. Choosing the wrong definition would leave an empty animation.

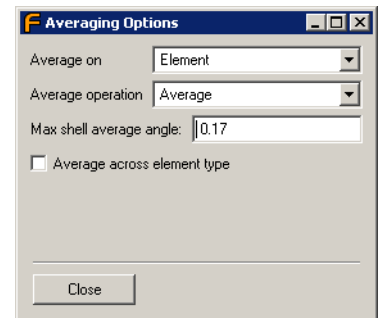
- ② **Result Set** – These options allow you to select one of the named result sets for the *Result class* (node, element, or element node) you selected. This setting is used to distinguish between multiple results for the specified FE entity, including results from different layers

within an element (for example, top and bottom of a shell element), or multiple deformation results on a node (for example, mode shape deformations for different eigenmodes).

- *By operation* – enabling this option consolidates the entire result set of the same type using the selected operation. The options include *Average*, *Maximum*, *Minimum*, *Max Difference*, and so on.
- *By name* – enabling this option allows you to load and display only the result sets with the specified name. Complete links without such a result set will remain unchanged. Parts of a link without the result set will be rendered gray.

- ③ *Averaging* – These options allow you to enable or disable the averaging options. If you select **Do Average**, press the **Opts...** button to define the properties in the Averaging Options dialog:

- *Average On* – Averaging on Node calculates a single value for each node in the FE model, based on the *Average Operation* and the values specified in the *Contour Value* settings. Averaging on Element calculates a single value for each element in the same way.
- *Average operation* – allows you to select the averaging operation (*Average*, *Max*, *Min*, etc.)



NOTE: *Element to Node and Node to Element averaging is not yet supported. This means that no contour colors will be shown when loading an animation with such settings.*

- *Max shell average angle* – you can specify an angle above which the averaging between shells stops. The angle is used as a tolerance when comparing the globalized coordinate systems of the elements in question.
- *Average across element type* – if enabled, Fedem averages the contour values across element type borders.

- ④ *Multiple Face Results* – This option enables you to specify the averaging behavior across elements that interface on a surface (such as stacking of elements with the same size and shape but different thicknesses).



TIP: Contour data is only loaded for visible elements. This means that you can select which of the multiple face results you want to show by hiding the appropriate elements.

Eigen Modes tab

If you selected *Eigen Mode* as the animation type in the Property Editor panel, you can animate the rigid body mode shapes calculated by Fedem while solving the dynamics. If the Mode Shape Recovery analysis is performed, it is also possible to include link deformations associated with the eigenmode in the animation. You may also animate link mode shapes computed during the link reduction if the *Expand mode shape* toggle in the *"Reduction Options tab"* was on (see [Section 4.1.4, "Link properties"](#)).

To animate mode shapes, you must specify the settings on the Eigen Modes tab (shown below) in the Property Editor panel, before loading the animation. This tab is not available for *Time History* and *Time Summary* animations.

- ❶ **Mode Selection** – These options allow you to select the mode type, time step or link, mode number, and a scale factor for the animation:
- Select between *System modes*, *Component modes of link* and *Free-free modes of reduced link* from the first pull-down list shown.
 - *Time* – If you selected *System modes*, this pull-down list shows the times at which an Eigenmode solution is specified in the Dynamics Solver Setup (see *"Eigenmode tab"* in [Section 6.5.1](#)).
 - *Link* – If you selected either *Component modes of link* or *Free-free modes of reduced link*, this pull-down lists all links in the model. It also have an entry (*All links*) to enable simultaneous link mode shape animations for all links. (See [Section 6.3.6, "Visualization of eigenmode shapes from the model reduction"](#) to learn more on link mode shape visualization.)
 - *Mode* – This pull-down list shows either the modes specified in the Dynamics Solver Setup (see *"Eigenmode tab"* in [Section 6.5.1](#)),

the component mode numbers (see "[Reduction Options tab](#)" in [Section 4.1.4](#)) or the free-free mode numbers for the reduced link, depending on the selected mode type in the first pull-down list.

- **Scale** – You can specify a scale factor to exaggerate the shapes during the animation.



NOTE: The default Scale setting (1.0) implies that the maximum amplitude of the shape is equal to the length scale used to model the mechanism. Therefore, if the length-span of the model is not 1.0, it is necessary to adjust the Scale option in order to obtain an appropriate deformation scale.

- ② **Frame Generation** – These options allow you to set the number of animation frames and the duration of the animation:
 - **Frames per cycle** – Entering a higher value increases the continuity of the animated motion.
 - **Length** – You can specify either a *Time*, *No. of cycles*, or *Until % damped* to limit the duration of the animation. These fields are available only for *System modes*. For *Component modes of link* and *Free-free modes of reduced link*, the duration of the animation will always be equal one full cycle.



NOTE: The *Until % Damped* option is relevant only if a damped eigenmode solution was performed (see "[Eigenmode tab](#)" in [Section 6.5.1](#)).



NOTE: When animating eigenmode shapes, the time that runs in the upper right corner of the modeler window is in the range $[0, n \cdot T]$ where T is the period ($= 1/\text{eigen frequency}$) and n is the number of cycles that are animated (usually $n=1$ for free vibrations and $n>1$ for damped vibrations).

7.3.3 Available animation results

The results available for animation depend on the solvers that have been run and their settings. For an indication of which results that could be available, use the Result File Browser (see [Section 8.2, "Result File Browser"](#)).



TIP: The heading of the .frs files (Fedem Result File) is readable and could give valuable information in some cases.

If an animation is loaded without the contour results in question being present, the legend text will display a question mark for the max and min values, and no contour colors will appear.

Most of the results that can be utilized in a contour animation are produced during stress recovery. All Element Node and Node results (selected in the Result Class pull-down) are currently produced during stress recovery. All Element results, however, are produced either directly

by the Strain Coat Recovery Summary, by FE-Fatigue, or by Duty Cycle analysis.

Element results produced during the Strain Coat Recovery Summary:

- Max principal stress
- Max principal strain
- Max shear stress
- Max shear strain
- Max von Mises stress
- Max von Mises strain
- Maximum stress range
- Maximum strain range
- Mean biaxiality
- Biaxiality standard deviation
- Most popular angle
- Angle Spread
- Damage
- Life (repeats)
- Life (equnits)

Element results that can be produced by FE-Fatigue and Duty Cycle:

- Damage
- Log Damage
- Life (repeats)
- Log Life (repeats)
- Life (equnits)
- Log Life (equnits)

Interpretation of shear strains in contour plots

The stress or strain state at a point in the model can be represented as a second-order tensor in a mathematical setting, i.e. in the 2D case we have

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix} \quad \text{and} \quad \varepsilon = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{yy} \end{bmatrix}$$

where equilibrium in angular momentum requires $\sigma_{xy} = \sigma_{yx}$. The strain components are defined as $\varepsilon_{ij} = (u_{i,j} + u_{j,i})/2$, where u is deformation and the indices i and j both run through x and y (x , y and z in 3D). The advantage of this representation is that computation of von Mises and principal quantities can be performed using the same piece of code for stress and strain thereby making the implementation more general and robust.

An alternative representation (more common to engineers) is by means of the stress and strain vectors, i.e. in 2D we have

$$\sigma = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} \quad \text{and} \quad \varepsilon = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yx} \\ \gamma_{xy} \end{bmatrix}$$

where $\gamma_{xy} = \varepsilon_{xy} + \varepsilon_{yx} = u_{x,y} + u_{y,x}$ and thus $\gamma_{xy} = 2\varepsilon_{xy}$.

The results from a stress recovery analysis that may be visualized in an animation are based on the tensorial strain representation, i.e. the shear strain components (including Max Shear) are equivalent to the ε_{xy} -term, and not γ_{xy} . However, the Max Shear strain quantity computed from the Strain Coat recovery analysis is currently based on the vector representation so that it will be twice as large as the equivalent quantity from the stress recovery analysis. It is important to be aware of this distinction when using that particular result quantity.

Stress and strain range quantities

The maximum stress- and strain range quantities computed in the Strain Coat Recovery are derived from the principal value history at each point. This computation is performed in a similar manner as for the *Most popular angle* and *Angle Spread* quantities. That is, the directions of the maximum (and minimum) principal values, in terms of the in-plane angle from the X-axis of the stress coordinate system, are used to divide the principal stress states into a discrete number of "bins" for each stress point. The principal stress values at a certain time is then assigned to one such bin depending on its angle, and the range for each such bin is computed as

the difference between the highest and the lowest value over the whole time history. The range value presented in the Time Summary animations is then selected from the bin having the largest computed range value.

7.3.4 Performance of animation loading

It is important to be aware that the performance of the animation loading is highly dependent upon the animation settings, especially when working with large FE-models (> 30 000 elements).

Memory usage is the most crucial point, and can be reduced to 1/3 when using the better options mentioned below.

The significant parameters are in order of decreasing importance:

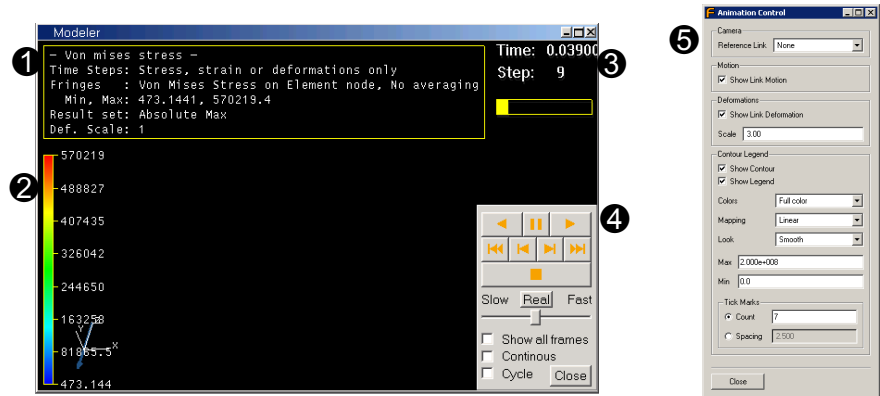
1. Whether result set selection is *By operation* or *By name*. (*By name* is better.)
2. Whether *Load face contours* and/or *Load line contours* are toggled. (*Only load face contours* is better.)
3. Whether averaging is turned on or off. (*No averaging* is better)

The number of elements visible when the animation is loaded is also important because result values will only be loaded for visible element faces. This means that you can load contour data for one group at a time if loading data for the complete link is too memory and time consuming.

7.4 Viewing animations

When an animation is loaded, you can use several tools to view and explore the data you have loaded. The modeler window will display some

additional features and the Animation Control window will be made available. These windows are shown in the figure below.

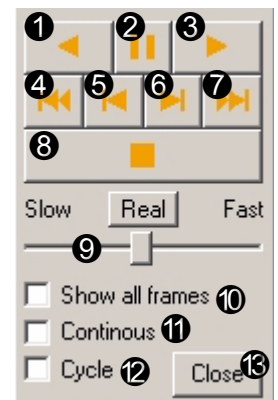


- ① *Legend text* – Displays information regarding the loaded animation
- ② *Legend bar* – Indicates the color assigned to each contour value.
- ③ *Time step information* – Displays the time and the time step number for the current frame along with a progress bar.
- ④ *Play Panel* – This panel is used to control how the animation is run.
- ⑤ *Animation controls* – This window features options to control several aspects of the animation including the contour legend domain.

7.4.1 Play panel

The Play Panel (shown below) is displayed in the lower-right corner of the Modeler window when an animation is loaded.

- ① *Reverse Play*
- ② *Pause Play*
- ③ *Forward Play*
- ④ *Beginning*
- ⑤ *Rewind by Frame*
- ⑥ *Forward by Frame*
- ⑦ *End*
- ⑧ *Stop Play*
- ⑨ *Speed slider* – you can adjust the speed of the animation by moving the slider to the right or



left. You can also click the *Real* button to return the speed to "real time."



NOTE: At faster speeds, Fedem may skip frames in order to maintain animation speed at the level you have specified.

- ⑩ *Show All Frames* – this option forces Fedem to show all the frames loaded, ignoring the *Speed* setting if necessary.
- ⑪ *Continuous* – this option allows you to play the animation repeatedly until you click the *Stop Play* button.
- ⑫ *Cycle* – enables playing the animation in *Forward Play* mode until it reaches the end, then plays it in *Reverse Play* mode.
- ⑬ *Close* button – closes the Play Panel and ends the animation session.



NOTE: The Play Panel does not appear if the animation consists of a single frame (e.g., Time Summary animations with results from a Strain Coat Recovery). To close such animations, you have to select *End Animation Session* from the Results menu, or type Ctrl+X.

7.4.2 Animation controls



Once an animation is loaded, you can open the Animation Control panel by selecting *Show Animation Controls...* on the *Tools* menu. The Animation Control panel is displayed as shown to the right. Changes made to settings in this panel are instantaneously applied to the animation.

- ① **Camera** – enables selection of reference link for the camera movement. See also "[Camera reference link](#)" below.
- ① **Motion** – enables display of rigid body motion of the mechanism.
- ② **Deformations** – enables display of link deformations scaled (exaggerated) by the *Scale* factor you specify.

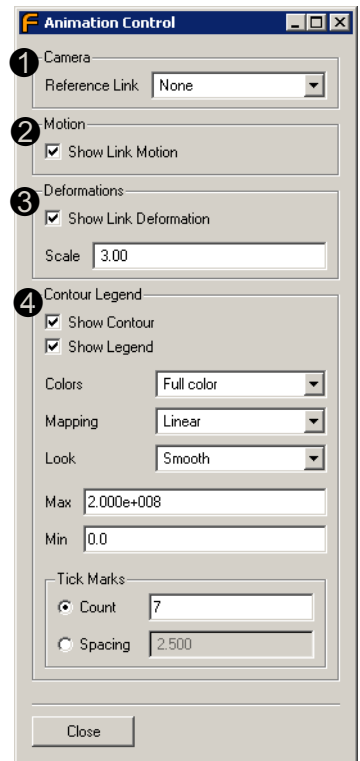


NOTE: This scale does not apply to the eigenmode deformations.

- ③ **Contour Legend** – these options enable you to customize the Contour Legend, and how the result numbers are converted to colors:
 - *Show Contour* – enables the display of contour contours on the mechanism during the animation.
 - *Show Legend* – enables display of the Contour Legend bar in the Modeler window.
 - *Colors* – the type of Color mapping used for the color contour, e.g. Full Color or Red Blue. See also "[Color mappings](#)" in [Section 7.4.3](#).
 - *Mapping* – *Linear* divides color increments linearly and *Log10* divides color increments on a logarithmic scale.
 - *Look* – either *Smooth* or *Discrete*.



NOTE: Discrete contours should only be selected when each face in the model has one single color, for example when showing single element results or results averaged on elements.



- *Max/Min* – you can set a maximum and minimum value to show on the Contour Legend. See also "[Contour value domain control](#)" in [Section 7.4.3](#).
- *Tick Marks* – you can select *Count* (number of ticks) or *Spacing* (difference between each tick mark) and specify a value to change the tick marks on the legend. See also "[Tick marks](#)" in [Section 7.4.3](#).

Camera reference link

When viewing an animation, it is possible to make the camera follow the motion of a link in the model. This is useful if some part of your model moves far during the simulation. To do this, selecting the link to follow in the "Reference Link" pull-down menu in the Animation Control window.

7.4.3 Contour legend control

Color mappings

The drop-down menu labeled Colors in the Animation Control dialog enables the selection of different color mappings for the contour values. The different color mappings map the contour values differently. This concerns values above, within and below the legend domain. They also show undefined results differently. There are currently four color mappings available:

Full Color – This is the default mapping, and is a common way to display structural results.

Values	Color
Above legend domain	Red
Within legend domain	Blue - Cyan - Green -Yellow - Orange - Red
Below legend domain	Blue
Undefined	Gray

Full Color B/W Limits – This is a mapping used to show what is within and outside the legend domain.

Values	Color
Above legend domain	White
Within legend domain	Blue - Cyan - Green -Yellow - Orange - Red

Values	Color
Below legend domain	Black
Undefined	Gray

Full Color Clipped Limits – This is a mapping that is useful when you want a quick overview of the few elements in a big complex model with results within a certain (narrow) value domain. It renders all elements outside the domain transparent, i.e. the elements within the domain will be the only ones visible.

Values	Color
Above legend domain	Invisible
Within legend domain	Blue - Cyan - Green -Yellow - Orange - Red
Below legend domain	Invisible
Undefined	Invisible

Red Blue – This is a mapping useful when you want the interpolation of colors within one element-face to be strictly consistent with the contour legend. Because of limitations in the 3D graphics hardware, interpolations of colors within one face will be linear. That makes the normal full color mapping inconsistent within one face if the different nodes in the element-face have values that map to colors separated by one or more colors in the legend domain.

Values	Color
Above legend domain	Red
Within legend domain	Blue - Red
Below legend domain	Blue
Undefined	Gray

Contour value domain control

The Max/Min fields in the Animation Control dialog are used to control the domain of the color legend. The defaults are *Max* = 0 and *Min* = 0. In this case Fedem uses the maximum and minimum values of the loaded contour results.

The Max/Min fields can also be used to “flip” the legend scale in order to show the least values as “red” or “worst”. To do this, simply enter the

worst value in the field labeled Max, and the best value in the field labeled Min, ignoring the relative size of the numbers.

When using a logarithmic mapping of the contour values, make sure that both the max and min values are above zero. If not, Fedem will not be able to calculate valid contour colors ($\log(x)$ where $x \leq 0$ does not exist).

Tick marks

There are two ways to control the tick marks on the color legend. Either by setting the number of tick marks wanted, or by setting the spacing between the tick marks.

When selecting Count, Fedem will distribute the given number of tick marks evenly in the value domain.

When selecting Spacing, Fedem will set tick marks with the given spacing between them, starting at the first complete multiple of the spacing value. When using a logarithmic mapping, the supplied value is multiplied with the decade in question.



TIP: When using logarithmic mapping, choose Spacing as tick mark distribution, and either 1, 2.5, 5 or 10 as spacing value.

Interpreting fatigue results

When nCode calculates fatigue results, some of the elements are rejected because they have near infinite life/zero damage. These elements are excluded from the max/min calculations done by Fedem when reading the results. Fedem assigns them a reference value that make them show up as little damage/ high life when using the ordinary Full Color legend mapping. If "Full Color B/W limits" are used, those elements will show up as Black (below legend domain) if plotting damage, or White (above legend domain) if plotting life. If using "Full Color Clipped Limits" all these elements will be removed from the display, leaving the interesting elements in the display.

The reference values Fedem uses are:

- Damage: $1e-20$
- Log Damage: -20
- All life plots (repeats/equits/Log): $1e20$

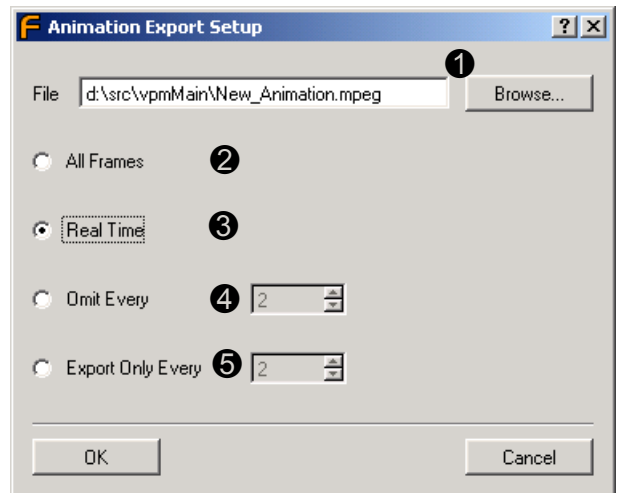
7.4.4 Exporting animations

The loaded animation can be exported to the mpeg-1, mpeg-2 and avi formats for viewing in an external viewer. Avi export is available on Windows only.

The animation you have loaded forms the basis for the exported animation. Through a simple setup dialog, you will be able to control the speed of the animation.

To start export, go to the *File* menu, select *Export...* and then *Export Animation*. The animation setup dialog will appear:

- ① *File*: Either type in the file name you want to export to, or click the "Browse" button to open a file dialog. If you type the file name manually, the given extension will decide which format the animation will be exported to. The default is to save the animation to model file root, with same name as in animation's description. Default file format is mpeg-1.



You can choose between the following operations:

- ② *All frames*. The animation will be exported "as is". The speed of the exported animation will depend solely on the simulation's time step size.
- ③ *Real time*. The animation will be exported so that one second movie time corresponds to one second of simulation time. Note that the animation's frame rate is 30 Hz, so this will be an approximation.
- ④ *Omit every nth frame*. Every nth frame will be omitted when exported.
- ⑤ *Export only every nth frame*. Most useful when you have solved using small time steps, and want finer control over what you export.

Hints and tricks

When exporting to mpeg, the mpeg-1 format produces smaller files than the mpeg-2 format, and the quality is approximately the same. For best result, use black background. It seems that the MS Media Player has problems playing the exported mpegs on some machines. If you discover this, you can try to play the animations with the Elecard mpeg player (<http://www.elecard.com>)

If you are working on Windows, we recommend exporting to avi. Avi export is faster and also produces better quality. When you export to avi, a dialog will pop up where you can set up which codec to use and settings for this codec. Be aware that some codecs may not work, this is dependent on your system configuration. The codecs that seem to have the best success rate is *MS Video 1*, *Cinepak by Radius*, and the *DivX codec*. If you don't have a DivX codec installed on your machine, you can get it from the DivX home page (<http://www.divx.com>).

Chapter 8 Managing Results

A Fedem simulation can generate large amounts of data. This chapter explains the concept of the Fedem *Results Data Base* (RDB), how to manage the data files and some information on how Fedem stores and handles the data.

Sections in this chapter address the following topics:

- [Model and Result file handling](#)
- [Result File Browser](#)
- [RDB directory structure](#)

8.1 Model and Result file handling

A Fedem model consists of the mechanism model file, the FE-model files and all the generated result data files. The generated results can be divided into three main groups:

- FE-model reduction data
- Dynamics response data
- FE-recovery data

Normally all these groups can be looked upon as one single model even though it is spread across several files and directories. Fedem keeps track of which files that are part of your saved model and which are not. Because the solvers write their data directly to disk while solving, Fedem also tracks what files are part of your modified and unsaved model, and can thus tell which files belong to the saved and the unsaved modified version of your model.

8.1.1 Discarding unsaved changes

When you open an existing model, the original result files present on disk are preserved, even if you delete results, change the model, run the solvers, etc., as long as you do not save your model. If you exit without saving, all the original data remains unchanged and the result database will be restored to the state of the last save.

The initial data is deleted or overwritten only when the model is saved.

Consequently, to be sure that the result files known by Fedem are those (and only those) present on disk, the model has to be saved first.



NOTE: *If you use the Result File Browser to delete result files, they will be physically removed from disc instantly. Their removal will not be delayed until the first save. This is done to actually free disk space as you delete the files, and not when you save your model.*

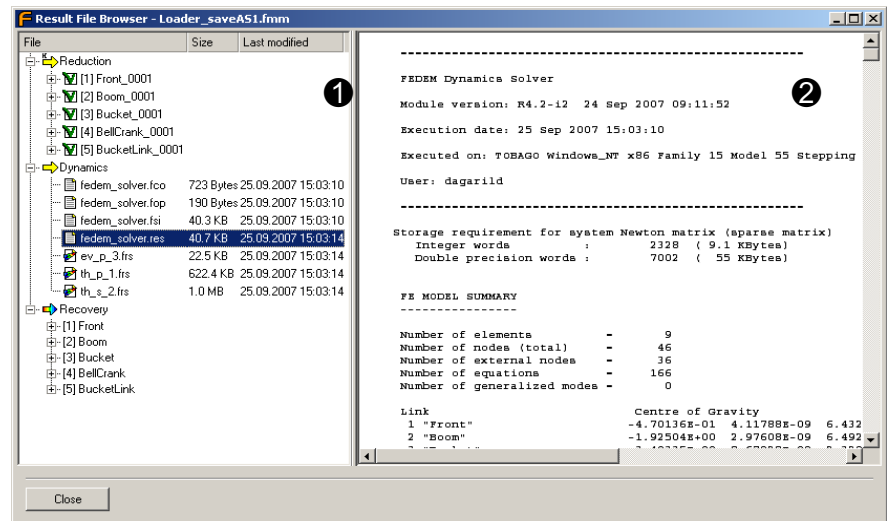
8.1.2 Saving a model

When the model is saved, the model file is updated with the current information from memory, including information on the contents and status of the results directory. In addition, all changed links are saved in the link repository (default: `link_DB/` directory) and the obsolete files in the results database are deleted.

8.2 Result File Browser

The Result File Browser can be used to view and manipulate files created by and used by the various Fedem modules. It offers features such as enable and disable result files, and delete individual results, or classes of results. It responds dynamically to any changes in the results data base, and is kept up-to-date at all times.

8.2.1 The Result File Browser dialog



- ❶ The *File list* - Lists all relevant files from the Reduction, Dynamics solver and Recovery processes.
- ❷ The *Info view* - Displays information about the selected file.

The File list

The list is ordered “chronologically”, with *Reduction* first, then *Dynamics* and *Recovery*. All directories will list the following file types, if present:

- - *frs* - Binary result files
- - *res* - Log file for the solver processes
- - *fco*, *fop*, *fao*, *fsi* - Input files to the solver processes
- - *fmX* - Reducer matrix files
- - *fsm* - Internal data structure files
- - *fpp*, *fef* - Fatigue result files

The files are listed with file name, size and time of last modification. Some of the files will also have an associated icon, corresponding to the icon that file type will have in Windows Explorer.

When you run a solver process, the result files from it will appear in the file list immediately after it is created, and it will be continuously updated up until the process finishes.

Reduction - This list shows the status and location of the results from the reduction process. The icon in front of each link indicates whether the link has a recognized set of reduced matrices or not. A green hatch indicates that the results are recognized as OK, a red cross indicates failure or that the link hasn't yet been reduced.

Dynamics - Shows the result files produced by the Dynamics Solver.

Recovery - Displays a list of all the links in your model, and for each link the result files for that link grouped by the recovery process they were created by.

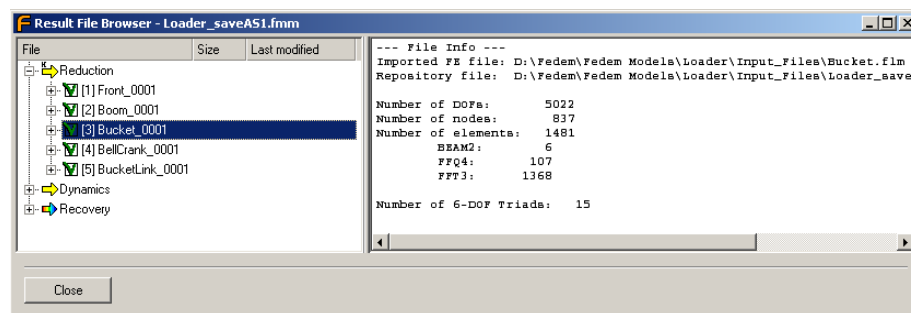


NOTE: If you are working on a slow machine and have a lot of results displayed in the file list, continuously updating the list may steal valuable CPU cycles from the solver process. Close the dialog to disable these updates.

The Info view

When selecting a file in the file list, the file can be viewed in the info view. The plain text files (*fco*, *fop*, *fao*, *fsi* and *res*) are displayed as-is, while selecting an *frs* file will show only the top of the header section.

Selecting a link under **Reduction** will show information about that link.



In addition to the full path to the imported FE data file and the internal repository file, you will here also get a summary of some size parameters of the FE model for the link. This includes the total number of degrees of freedom, the number of nodes and the number elements of each type. The number of triads attached to the link is also indicated.

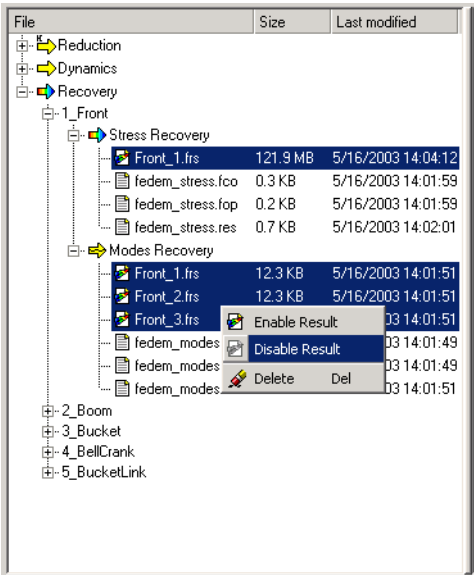
The size information of the FE model is available also before the link is reduced. It is therefore useful for assessing the computational cost of reducing the link. This size information is not shown for Generic Parts, even when a FE model is used for visualizing the link.

8.2.2 Result manipulation

The Result File Browser can be used to manipulate the results, both enable/disable results to decrease memory usage, and delete single or multiple result files to save disk space.

Disabling and Enabling results

To disable/enable results, select the files you want to enable/disable, right-click, and select either *Enable Results* or *Disable Results* from the menu. The icons of the files immediately changes to reflect the current result state.



What you actually do when you disable a result file, is to temporarily remove all the result information in the result file from memory, and the results from that file will be unavailable for post-processing (curve plotting and animation). As a result, Fedem consumes less memory. You may re-enable the disabled results any time you wish. The results from this file will then be available for post-processing again.



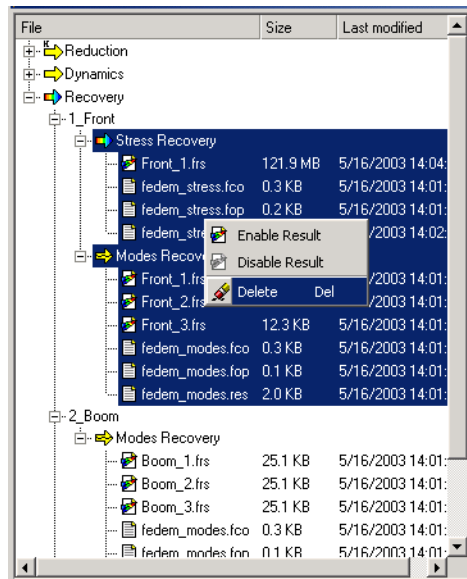
NOTE: Selecting a top level item in the file list will also automatically select any *frs* files located below that item in the list.



NOTE: Only results from the recovery processes are available for enabling/disabling.

Deleting results

Using the same approach as when "*Disabling and Enabling results*", you can also delete individual result files, classes of results, or results on selected links. Just select the results you want to delete, right-click, and select *Delete* from the menu.



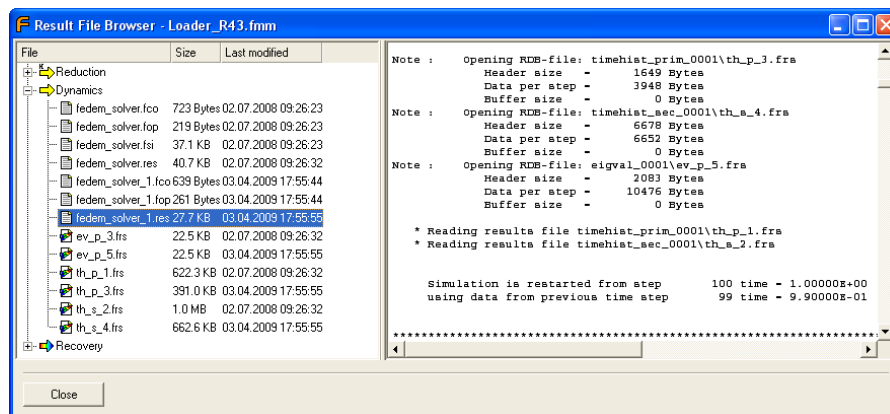
WARNING! Deleting the primary time history results (the file named *th_p_1.frs*) will also cause all other results to be deleted.



WARNING! When you delete results, they are physically removed from disc, and there is no way of getting them back at a later stage. An exception to this is when you delete the primary time history result file. In that particular case the results are not removed from disc until the next time you save your model.

8.2.3 Result files from restart simulations

When the Dynamics simulation has been restarted at least once (see *"Time tab"* in [Section 6.5.1](#)), the *Dynamics* part of *"The File list"* in the Result File Browser will contain solver option and result files for each individual run. The option files (extension `fop`, `fco` and `fao`) and log-files (`res`) from restart simulations will have a number appended to the base name, indicating the actual restart number (i.e. `fedem_solver_1.res`), whereas the binary result files (`frs`) will have numbers 3 or 4 and beyond, depending on whether eigenmode analysis was activated.



NOTE: There will always be only one `fedem_solver.fsi` file in the File list regardless of whether results have been performed or not, because the same file is used by all restart runs. This file only contains model data that is not allowed to change in a restart.

You may Enable/Disable and Delete individual files from restarts in a similar way as the files from the original run (see [Section 8.2.2, "Result manipulation"](#)), in order to control what results should be active for further post-processing and recovery runs. If multiple results are active for a given time set of time steps, only those that were produced latest will be used¹.

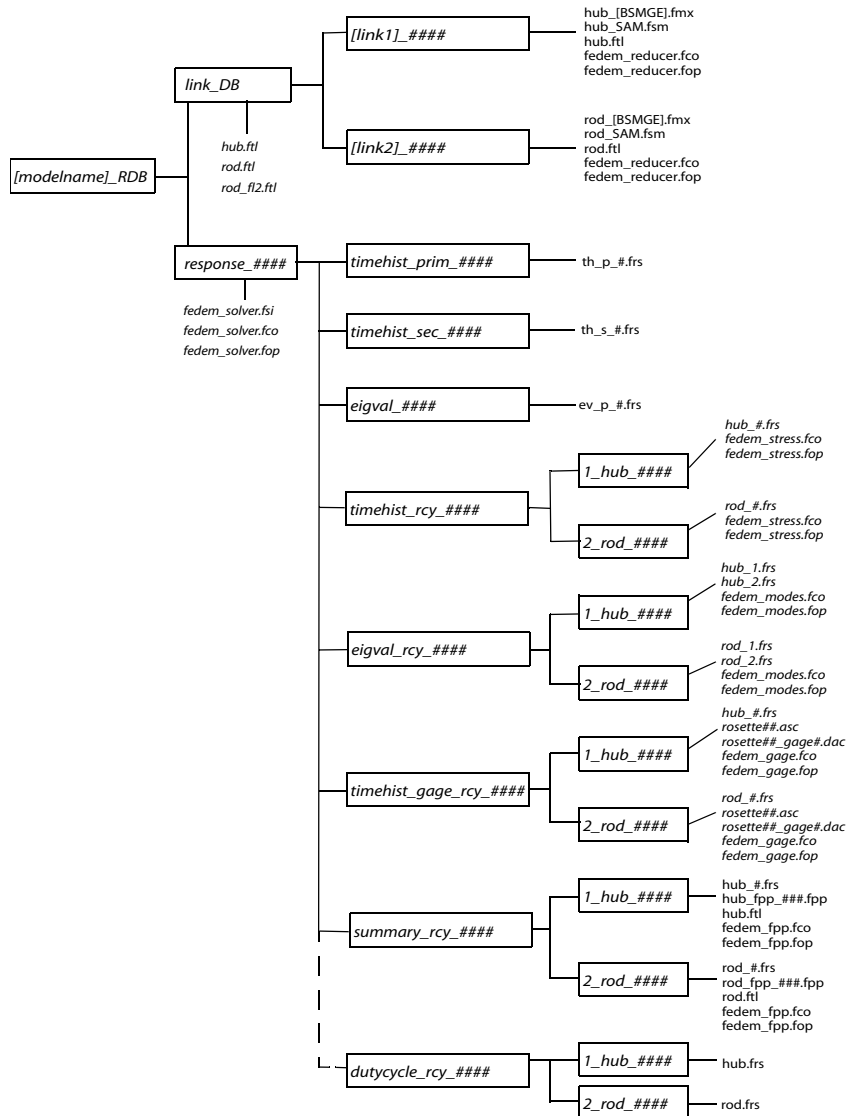


NOTE: If deleting a primary time history results file from a restart simulation (e.g. the file `th_p_3.frs` in the view above), all other simulation results are not automatically deleted as well, as happens when the primary result file of the original simulation, `th_p_1.frs`, is deleted).

1. The same is true when post-processing results from recovery simulations that have been rerun several times on the same dynamics simulation results.

8.3 RDB directory structure

The map below outlines the RDB directory structure created by Fedem. The hierarchy root is named `[modelname]_RDB`, where `[modelname]` is the name of the current model. For example, the model file `FrontSuspension.fmm` creates a results directory structure under the directory `FrontSuspension_RDB`.



8.3.1 Link database

The `link_DB` directory (either specified as default, link specific or model specific) contains Fedem `.ftl` files of all unique links imported into the Fedem model. All `.ftl` files in this directory are stored without external node information. This reduces the total storage requirements.

A set of new subdirectories is created under the `link_DB` directory to store information about already reduced links. This directory is named `[linkname]_####`, where `linkname` is the name of the actual link, and `####` represents a configuration number.

Option files for Fedem Reducer are also stored in the `[linkname]_####` directories. This enables reduced links to be moved between result databases.



NOTE: When link specific or model specific repository is used (see [Section 4.1.5, "Using link repositories"](#)), the file name conventions apply to the subdirectories of that directory, and not the `link_DB` directory.



CAUTION: If you use a file browser to remove unwanted files in the Fedem Results Database directory structure, do not remove the `.ftl` files directly under the `link_DB` directory. Other files will be automatically recreated when a new simulation starts.



CAUTION: If you use a file browser to move a Fedem Mechanism Model file (`.fmm`), you must also move the `link_DB` directory to ensure inclusion of the link definitions.

Model reduction file management

When Link Reduction is started, a new directory for each link is created and the files needed by `fedem_reducer` are written to this directory. These files include an `.ftl` file with external node specification, and the options files `.fco`, `.fop` and `.fao` (see [Appendix D.1, "File types"](#)).

After the reduction, all input files are retained for reference and a possible rerun of the process in batch mode at a later stage.

8.3.2 Response directory structure

The `response_####` directory is the entry-point to result files from dynamics simulation and recovery operations.

Option files for the dynamics solver are stored directly in the `response_####` directory. These files can be used to run the solver in batch mode.



CAUTION: These files are auto-generated from the main Fedem application. Manually changing the contents of these files and running the solver from the command line creates results that are inconsistent with the definitions in the model file.

All results files in the *response_####* structure are stored in the same format, but are placed in different directories, making separate result types easily distinguishable.

The recovery modules will store their results in a linkwise manner in subdirectories under their main result directory, named *[ID]_[linkname]_####*, where *ID* is the link identification number. All option files are also stored in these subdirectories.

The following is a list of the result directories and a description of their contents:

- *timehist_prim_####*: The primary time history result files are named *th_p_#.frs* and contain primary response results from the dynamics solver.
- *timehist_sec_####*: The secondary time history result files are named *th_s_#.frs* and contain all secondary response results from the dynamics solver.
- *eigval_####*: Calculated eigenvalues and associated eigen vectors from the dynamics simulation are stored in files named *ev_#.frs*.
- *timehist_rcy_####*: Results from the stress recovery process will be stored linkwise in subdirectories. The files will be named *[linkname]_#.frs*.
- *eigval_rcy_####*: Results from the eigenvalue recovery process will be stored linkwise in subdirectories under this directory, and will be named *[linkname]_#.frs*.
- *timehist_gage_####*: Results from strain rosette recovery are stored linkwise in subdirectories under this directory. The result files will be named *[linkname]_#.frs*. Fedem will also create ASCII and DAC files. See [Section 6.8, "Strain rosette analysis"](#).
- *summary_rcy_####*: Results from strain coat recovery are stored linkwise in subdirectories under this directory. The result files are named *[linkname]_#.frs*, and if you create a rainflow matrix file, it will be named *[linkname]_fpp_###.fpp*, where the running number is one higher than the corresponding frs file. If you run FE-Fatigue, the result file will have a similar syntax as the fpp file, and with the same running number.

- *dutycycle_rcy_####*: Results from Duty Cycle calculation is stored linkwise under this directory. One result file will be created, named *[linkname]_#.frs*.

Strain Coat Recovery Summary file management

When the Strain Coat Recovery process is started, a new directory called

response_####/summary_rcy_####/<ID>_<linkname>_####

is created, and the files needed by *fedem_fpp* are written to this directory. These files include an *.ftl* file with Strain Coat elements, and the option files *.fco*, *.fop* and *.fao*.

The Strain Coat elements are of a “non-structural” type and do not affect the link reduction results.



NOTE: The *.ftl* file written to this directory is not necessarily identical to the corresponding file in the link repository, depending on whether the model has been saved since the strain coat elements were created.

Appendix A Using the MF-Tyre Model



This chapter contains proprietary information of TNO. The contents of this chapter may not be disclosed to other parties, copied, nor duplicated for commercial use without the prior written permission of TNO.

The Magic-Formula (MF-Tyre) tire model is developed by TNO Automotive. MF-Tyre is the premier handling tire model available in Fedem.

This chapter includes the following sections:

- [About MF-Tyre](#)
- [MF-Tyre Version 5.2](#)
- [Tire-Road Interaction](#)
- [Axis Systems and Definitions](#)
- [The Magic Formula Tire Model \(MF-Tyre\)](#)
- [Steady-State: Magic Formula](#)
- [Some Practical Aspects](#)
- [Standard Tire Interface \(STI\)](#)
- [MF-Datasets and MF-Tool](#)
- [Definitions](#)
- [References](#)

A.1 About MF-Tyre

The MF-Tyre model uses a method known as the Magic Formula to calculate the steady-state behavior of a tire. The Magic Formula is actually a set of mathematical formula based on the physical background of the tire, road, and the tire-to-road contact.

The Magic Formula tyre model aims at an accurate description of the steady-state behavior of a tyre by providing a set of mathematical formulae, which are partly based on a physical background. The Magic Formula calculates the forces (F_x , F_y) and moments (M_x , M_y , M_z) acting on the tyre under pure and combined slip conditions, using longitudinal and lateral slip (k , a), wheel camber (g) and the vertical force (F_z) as input quantities. In addition to the Magic Formula description, a set of differential equations is defined, representing the transient behavior of the tyre with respect to handling at frequencies up to 8 Hz.

Further information can be found on the internet site:
www.delft-tyre.com.

A.2 MF-Tyre Version 5.2

Compared to MF-Tyre 5.1, following items have been changed or introduced:

- The scaling factors for the shifts have been defined such that conicity and plysteer effects can be easily switched off.
- Into the modelling of combined cornering and braking/traction E factors have been introduced, making the modelling more accurate.
- The rolling resistance torque has become a function of forward speed. The influence of the camber on the peak F_x has been introduced.

[Table A-1](#) lists the additional parameters.

Table A-1: New Parameters Introduced in MF-Tyre 5.2

Name:	Name used in tire property file:	Explanation:	Default Value:
$\lambda_{\gamma x}$	LGAX	Scale factor of camber for Fx	1
$\lambda_{\gamma y}$	LGAY	Scale factor of camber force stiffness	1
λ_{Vmx}	LVMX	Scale factor of Mx vertical shift	1
p_{Dx3}	PDX3	Variation of friction Mux with camber	0
r_{Ex1}	REX1	Curvature factor of combined Fx	0
r_{Ex2}	REX2	Curvature factor of combined Fx with load	0
r_{Hy2}	RHY2	Shift factor for combined Fy reduction with load	0
r_{Ey1}	REY1	Curvature factor of combined Fy	0
r_{Ey2}	REY2	Curvature factor of combined Fy with load	0
q_{sy3}	QSY3	Rolling resistance torque depending on speed	0
q_{sy4}	QSY4	Rolling resistance torque depending on speed ⁴	0

Furthermore, LONGVL should be defined and have a positive value. When the default values are used or omitted, the tire model is fully backward compatible with MF-Tyre version 5.1.



A.3 Tire-Road Interaction

The tire-road contact forces are mainly dependent of the tire mechanical properties (that is, stiffness and damping), the road condition (that is, the friction coefficient between tire and road, the road structure), and the motion of the tire relative to the road (that is, the amount and direction of slip).

The major control and disturbance forces on a vehicle arise from the contact of the tires with the road. The vertical loads transfer the weight of the vehicle to the road. Due to the compliance of the tires, a vehicle is

cushioned against disturbances by small road irregularities. The traction and braking forces arise from the longitudinal tire forces.

Lateral forces are required to control the direction of travel of the vehicle. The lateral behavior of tires is therefore dominant in vehicle handling. Proper description of the dynamic behavior of a vehicle requires an accurate model of the tire-road contact forces and moments generating properties under all of these different conditions.

Figure A-1. Tire Factors

Tyre factors			
	<i>(Quasi) steady state</i>		<i>Vibratory state</i>
in-plane	load carrying capacity braking/driving performance rolling resistance	radial deflection longitudinal slip and distortion	cushioning capacity dynamic coupling
out-of-plane	cornering performance lateral shift of F_z	lateral slip and distortion	phase shifts and destabilisation
primary effects	interactions between		
secondary effects	in- and out-of-plane behaviour		

Tire behavior results from a combination of several aspects. Factors may be distinguished which concern the primary tasks of the tire which involve (often important) secondary effects. In Figure A-1 these factors have been brought in matrix form. A distinction has been made between (quasi) steady-state and vibratory behavior and besides between in-plane and out-of-plane aspects. The primary task factors are shaded in green. The remaining secondary factors are not shaded.

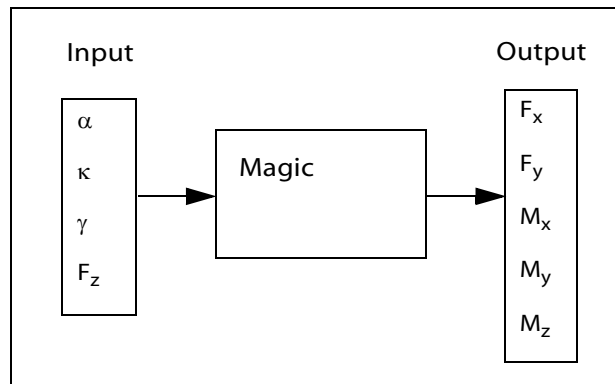
The requirements to transmit forces in the three perpendicular directions (F_x , F_y and F_z) and to cushion the vehicle against road irregularities involve secondary factors like radial, lateral and longitudinal distortions and slip.

Although considered as secondary factors, some of the quantities involved have to be treated as input variables into the system which generate the forces. Figure A-2 presents the input and output vectors. In this diagram the tire is assumed to be uniform and to move over a flat road surface. The input vector results from motions of the wheel relative to the road. It is advantageous to recognize the fact that, for small

deviations from the straight-ahead motion, in-plane and out-of-plane motions of the assumedly symmetric wheel-tire system are uncoupled.

The forces and moments are considered as output quantities of the tire model. They are assumed to act on a rigid disc with inertial properties equal to those of the undeflected tire. The forces may differ from the corresponding forces acting on the road due to the vibrations of the tire relative to the wheel rim. Braking and traction torques are considered as acting on the rotating disc.

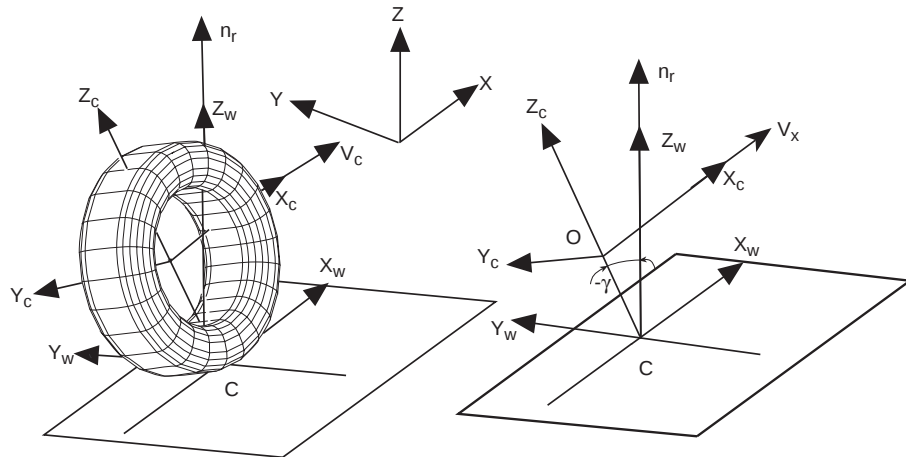
Figure A-2. Input and Output Variables of the Magic Formula Tire Model



A.4 Axis Systems and Definitions

A.4.1 W-Axis System

MF-Tyre conforms to the TYDEX STI conventions described in the TYDEX-Format [1] and the Standard Tire Interface [2]. Two TYDEX coordinate systems with ISO orientation are particularly important, the C- and W-axis systems as detailed in Figure A-3.

Figure A-3. TYDEX C- and W-Axis Systems Used in MF-Tyre, According to TYDEX

The C-axis system is fixed to the wheel carrier with the longitudinal x_c -axis parallel to the road and in the wheel plane (x_c - z_c -plane). The origin O of the C-axis system is the wheel center.

The origin of the W-axis system is the road contact-point (or 'point of intersection') C defined by the intersection of the wheel plane, the plane through the wheel spindle and the road tangent plane. The orientation of the W-axis system agrees to ISO. The forces and torques calculated by MF-MCTyre, which depend on the vertical wheel load F_z along the z_w -axis and the slip quantities, are projected in the W-axis system. The x_w - y_w -plane is the tangent plane of the road in the contact point C.

The camber angle is defined by the inclination angle between the wheel plane and the normal n_r to the road plane (x_w - y_w -plane).

A.4.2 Units

Next to the convention to the TYDEX W-axis system, all units of the parameters and variables used in MF-Tyre agree to the SI units. In [Table A-2](#) provides an overview of the most important parameters and variables, see also [Section A.10, "Definitions"](#).

Table A-2: SI Units Used in MF-Tyre

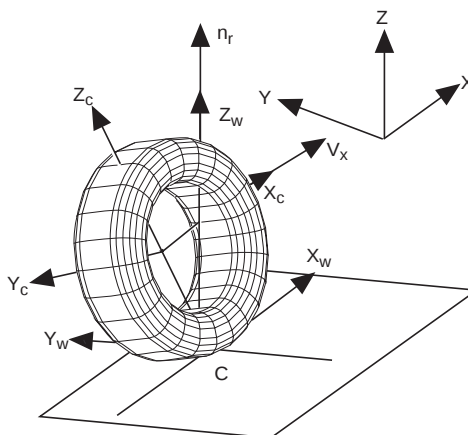
Variable Type:	Name:	Abbreviation:	Unit:
angle	slip angle	α	[rad]
	camber angle	γ	
force	longitudinal force	F_x	[N]
	lateral force	F_y	
	vertical load	F_z	
	overturning moment	M_x	[Nm]
	rolling resistance moment	M_y	
	self aligning moment	M_z	
speed	longitudinal speed	V_x	[m/s]
	lateral speed	V_y	
	longitudinal slip speed	V_{sx}	
	lateral slip speed	V_{sy}	
rotational speed	tire rolling speed	Ω	[rad/s]

A.4.3 The Contact-Point C and the Normal Load

The radius of curvature of the road profile is considered large as compared to the radius of the tire. The tire is assumed to have only a single contact point (C) with the road profile. Furthermore, for calculating the motion of the tire relative to the road, the road is approximated by its tangent plane at the point on the road below the wheel centre (see Figure A-4). The tangent plane is an accurate approximation of the road, as long as the road radius of curvature is not too small (that is, not smaller than 2 meters).



Figure A-4. Contact Point C (Intersection Between Normal-to-Road Tangent and Wheel Plane)



The normal load F_z of the tire is calculated with:

$$F_z = C_z \rho + K_z \cdot \dot{\rho} \quad (\text{A-1})$$

with ρ the tire deflection and $\dot{\rho}$ the deflection velocity of the tire.

Table A-3: Normal Load

Name:	Name Used in Tire Property File:	Explanation:
R_o	UNLOADED_RADIUS	Free tire radius
C_z	VERTICAL_STIFFNESS	Tire vertical stiffness
K_z	VERTICAL_DAMPING	Tire vertical damping

A.4.4 The Effective Tire Rolling Radius

The loaded tire radius R which is defined by the distance of the wheel centre to the centre of tire contact (see Figure A-5).

The effective rolling radius R_e (at free rolling of the tire) is defined by:

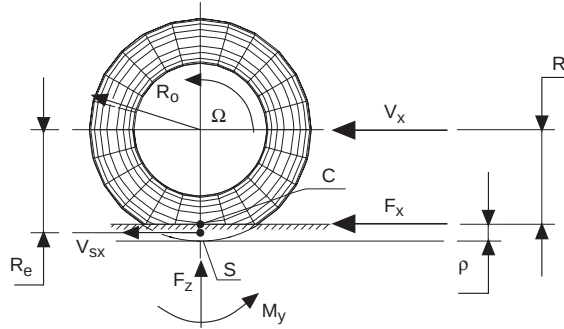
$$R_e = \frac{V_x}{\Omega} \quad (\text{A-2})$$

For radial tires the effective rolling radius decreases with increasing vertical load at low loads, but around its nominal load the influence of the vertical load is small, see Figure A-6.

When assuming a constant vertical tire stiffness C_z , the radial tire deflection ρ can be calculated with:

$$\rho = \frac{F_z}{C_z} \quad (\text{A-3})$$

Figure A-5. Effective Rolling Radius and Longitudinal Slip



For the estimation of the effective rolling radius R_e a Magic Formula approach is chosen. The equation of the effective rolling radius R_e reads:

$$R_e = R_0 - \rho_{F_{z0}} (D \arctan(B \rho^d) + F \rho^d) \quad (\text{A-4})$$

in which R_0 is the unloaded radius and the nominal tire deflection $\rho_{F_{z0}}$ is defined by:

$$\rho_{F_{z0}} = \frac{F_{z0}}{C_z} \quad (\text{A-5})$$

and the dimensionless radial tire deflection ρ^d can be calculated with:

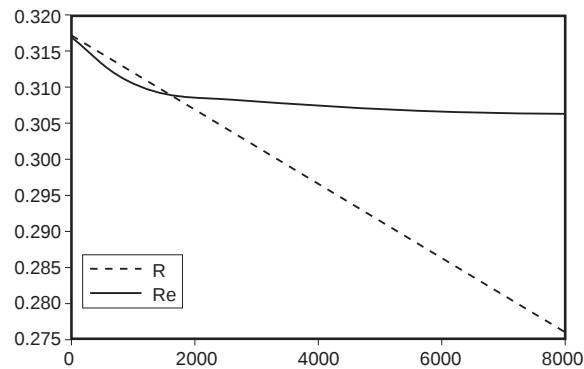
$$\rho^d = \frac{\rho}{\rho_{F_{z0}}} \quad (\text{A-6})$$

For a large range of tires, appropriate coefficient values are:

- 3,...,B,...,12
stretches the ordinate of the arctangent function, a large value of B means a high slope at $F_z=0$;
- 0.2,...,D,...0.4
defines the shift from the asymptote at high wheel loads;

— 0.03,...,F,...,0.25
defines the ratio between tire radial deformation r and effective tire deformation. Low values are obtained for extremely stiff tires.

Figure A-6. The Tire Effective Rolling Radius as a Function of the Vertical Load (B=8.4, D=0.27 and F=0.045)



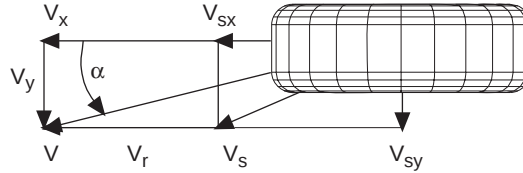
In Figure A-6 an example of the effective rolling radius is shown for a passenger car tire. The approximation of R_e is made with the proposed formula with: $B = 8.4$, $D = 0.27$ and $F = 0.045$.

Table A-4: Effective Rolling Radius Parameters

Name:	Name used in tire property file:	Explanation:
F_{z0}	FNOMIN	Nominal wheel load
B	BREFF	Low load stiffness eff. rolling radius
D	DREFF	Peak value of effective rolling radius
F	FREFF	High load stiffness effective rolling radius

A.4.5 Tire Slip Quantities

Figure A-7. Slip Quantities at Combined Cornering and Braking/Traction



The longitudinal slip speed is defined as:

$$V_{sx} = V_x - \Omega R_e \quad (\text{A-7})$$

and the lateral slip speed:

$$V_{sy} = V_y \quad (\text{A-8})$$

The practical slip quantities κ and α are defined as:

$$\kappa = -\frac{V_{sx}}{V_x} \quad (\text{A-9})$$

$$\tan \alpha = \frac{V_{sy}}{|V_x|} \quad (\text{A-10})$$

with V_{sx} and V_{sy} the components of the slip speed which may be defined as the velocity of point S in the W-axis system (see Figure A-7).

With Ω denoting the rotational speed of the tire, the linear rolling speed becomes:

$$V_r = R_e \Omega \quad (\text{A-11})$$

A.5 The Magic Formula Tire Model (MF-Tyre)

A.5.1 Introduction

For a given pneumatic tire and road condition, the tire forces due to slip follow a typical characteristic. The characteristics can be accurately approximated by a special mathematical function which is known as the

"Magic Formula". The parameters in the Magic Formula depend on the type of the tire and the road conditions. These parameters can be derived from experimental data obtained from tests. The tire is rolled over a road at various loads, orientations and motion conditions.

The Magic Formula tire model is mainly of an empirical nature and contains a set of mathematical formula, which are partly based on a physical background. The Magic Formula calculates the forces (F_x , F_y) and moments (M_x , M_y , M_z) acting on the tire at pure and combined slip conditions, using longitudinal and/or lateral slip (κ , α), wheel camber γ and the vertical force F_z as input quantities. The model takes into account plysteer and conicity. An extension has been provided that describes transient and oscillatory tire behavior for limited frequencies smaller than 8 Hz and wavelengths larger than the tire circumference.

A.5.2 History of the Magic Formula

Through the initiative of Volvo Car Corp. a cooperate effort was started in the mid-eighties with the Delft University of Technology to develop a tire model that accurately describes the tire's ability to have horizontal forces generated between road and tire.

The first Magic Formula version was presented in 1987 [3]. The basic idea of using the sine and arcsine functions was described for mainly pure slip conditions. Further 'prototype' formula were proposed for combined slip conditions.

In the second version [4], presented in 1989 the formula for combined cornering conditions, based on physical background, were improved and tire relaxations lengths were introduced in order to have a first order approach of the transient tire behavior. This model was improved on the description for combined slip calculations in 1993 [5].

Bayle e.o. [6] proposed to have a more empirical approach, reducing the complexity of the force calculations under combined slip conditions and yielding a considerably higher calculation speed. Their method improved the calculation speed during the calculation of the Magic Formula parameters and during simulation calculations.

The latest version [7] combines the advantage of the previous versions and has been modified for the following aspects:

The self aligning torque has been made dependent on the side force by a new approach using the pneumatic trail in pure and combined slip conditions;

- The forces under combined slip conditions are calculated according to the proposal of Bayle [6];
- Formulae describing overturning moment have been introduced;
- The transient tire behavior has been improved to enable zero speed;
- Loading variations to tire lift off situations;
- The parameters used in formulae are dimensionless improving manipulations with tire characteristics and parameter calculations ("fitting");
- Scaling factors are introduced for vehicle-tire optimization purposes.

A.5.3 Learning the Basics of the Magic Formula

The general form (sine version) of the formula reads:

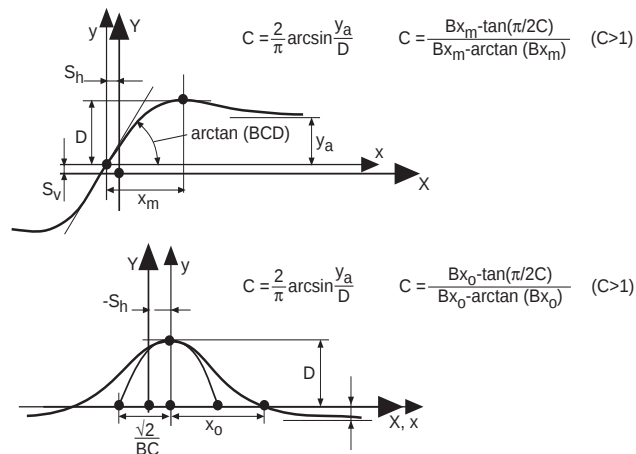
$$Y(x) = D \sin[C \arctan\{Bx - E(Bx - \arctan(Bx))\}] \quad (\text{A-12})$$

where $Y(x)$ is either, F_x or F_y .

The self aligning moment M_z is calculated by using the lateral force F_y and the pneumatic trail t , which is based on a cosine type of Magic Formula:

$$Y(x) = D \cos[C \arctan\{Bx - E(Bx - \arctan(Bx))\}] \quad (\text{A-13})$$

Figure A-8. Curves Produced by the Sine and Cosine Versions of the Magic Formula



When the formula is used to calculate the forces generated by the tire, the following variables should serve as input for the Magic Formula:

Input Variables

Longitudinal slip	κ	[-]
Slip angle	α	[rad]
Camber angle	γ	[rad]
Normal wheel load	F_z	[N]

In case the complete model including transient properties is used, the transient tire quantities are employed instead of the wheel slip quantities κ and α .

Output Variables (in contact point C)

Longitudinal force	F_x	[N]
Lateral force	F_y	[N]
Overturning couple	M_x	[Nm]
Rolling resistance torque	M_y	[Nm]
Aligning torque	M_z	[Nm]

Basic Tire Parameters

Nominal (rated) load	F_{z0}	[N]
Unloaded tire radius	R_0	[m]
Tire belt mass	m_{belt}	[kg]

Furthermore, the normalized vertical load increment df_z is defined:

$$df_z = \frac{F_z - F'_{z0}}{F'_{z0}} \quad [-] \tag{A-14}$$

with the possibly adapted nominal load (using the user scaling factor λ_{Fz0}):

$$F'_{z0} = F_{z0} \cdot \lambda_{Fz0} \tag{A-15}$$

Tire Model Parameters

In the subsequent sections, formulae are given with non-dimensional parameters a_{ijk} with the following values and connections:

Table A-5: Tire Model Parameters

Parameter:		Definition:
a =	p	Force at pure slip
	q	Moment at pure slip
	r	Force at combined slip
	s	Moment at combined slip
i =	B	Stiffness factor
	C	Shape factor
	D	Peak value
	E	Curvature factor
	K	Slip stiffness = BCD
	H	Horizontal shift
	V	Vertical shift
	s	Moment at combined slip
	t	Transient tire behavior
j =	x	Along the longitudinal axis
	y	Along the lateral axis
	z	About the vertical axis
k =	1, 2, ...	



A.5.4 User Scaling Factors

For the user convenience a set of scaling factors is available to examine the influence of changing a number of important overall parameters. The

default value of these factors is one. The following factors have been defined:

Table A-6: Scaling Coefficient, Pure Slip

Name:	Name used in tire property file:	Explanation:
I_{Fz0}	LFZO	Scale factor of nominal (rated) load
I_{Cx}	LCX	Scale factor of Fx shape factor
I_{mx}	LMUX	Scale factor of Fx peak friction coefficient
λ_{Ex}	LEX	Scale factor of Fx curvature factor
λ_{Kx}	LKX	Scale factor of Fx slip stiffness
λ_{Hx}	LHX	Scale factor of Fx horizontal shift
λ_{Vx}	LVX	Scale factor of Fx vertical shift
λ_{gx}	LGAX	Scale factor of camber for Fx
λ_{Cy}	LCY	Scale factor of Fy shape factor
λ_{my}	LMUY	Scale factor of Fy peak friction coefficient
λ_{Ey}	LEY	Scale factor of Fy curvature factor
λ_{Ky}	LKY	Scale factor of Fy cornering stiffness
λ_{Hy}	LHY	Scale factor of Fy horizontal shift
λ_{Vy}	LVY	Scale factor of Fy vertical shift
$\lambda_{\gamma y}$	LGAY	Scale factor of camber for Fy
λ_t	LTR	Scale factor of Peak of pneumatic trail
λ_{Mr}	LRES	Scale factor for offset of residual torque
$\lambda_{\gamma z}$	LGAZ	Scale factor of camber for Mz
λ_{Mx}	LMX	Scale factor of overturning couple
λ_{vMx}	LVMX	Scale factor of Mx vertical shift
λ_{My}	LMY	Scale factor of rolling resistance torque

Table A-7: Scaling Coefficients, Combined Slip

Name:	Name used in tire property file:	Explanation:
$\lambda_{x\alpha}$	LXAL	Scale factor of alpha influence on Fx
$\lambda_{\psi\kappa}$	LYKA	Scale factor of alpha influence on Fx
$\lambda_{\psi\gamma\kappa}$	LVYKA	Scale factor of kappa induced Fy
λ_s	LS	Scale factor of Moment arm of Fx

Table A-8: Scaling Coefficients, Transient Response

Name:	Name used in tire property file:	Explanation:
$\lambda_{\sigma\kappa}$	LSGKP	Scale factor of Relaxation length of Fx
$\lambda_{\sigma\alpha}$	LSGAL	Scale factor of Relaxation length of Fy
λ_{gyr}	LGYR	Scale factor of gyroscopic torque

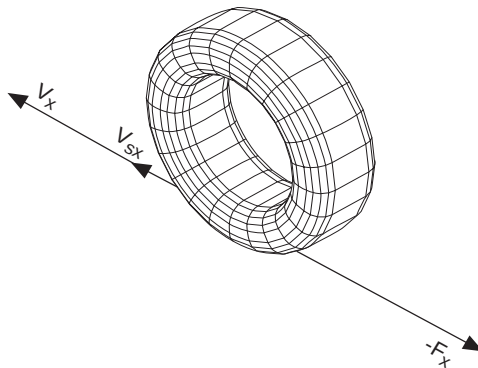


A.6 Steady-State: Magic Formula

A.6.1 Steady-State Pure Slip

Formula: Longitudinal Slip (Pure Slip)

Figure A-9. Longitudinal Slip Condition (Pure Braking/Traction)



$$F_x = F_{x0}(\kappa, F_z) \quad (\text{A-16})$$

$$F_{x0} = D_x \sin[C_x \arctan\{B_x \kappa_x - E_x(B_x \kappa_x - \arctan(B_x \kappa_x))\}] + S_{Vx} \quad (\text{A-17})$$

$$\kappa_x = \kappa + S_{Hx} \quad (\text{A-18})$$

$$\gamma_x = \gamma \cdot \lambda_{\gamma x} \quad (\text{A-19})$$

with coefficients:

$$C_x = p_{Cx1} \cdot \lambda_{Cx} \quad (\text{A-20})$$

$$D_x = \mu_x \cdot F_z \quad (\text{A-21})$$

$$\mu_x = (p_{Dx1} + p_{Dx2} df_z) \cdot (1 - p_{Dx3} \cdot \gamma_x^2) \lambda_{\mu x} \quad (\text{A-22})$$

$$E_x = (p_{Ex1} + p_{Ex2} df_z + p_{Ex3} df_z^2) \cdot \{1 - p_{Ex4} \operatorname{sgn}(\kappa_x)\} \cdot \lambda_{Ex} (\leq 1) \quad (\text{A-23})$$

$$K_x = F_z \cdot (p_{Kx1} + p_{Kx2} df_z) \cdot \exp(p_{Kx3} df_z) \cdot \lambda_{Kx} \quad (\text{A-24})$$

$$\left(K_x = B_x C_x D_x = \frac{\partial F_{x0}}{\partial \kappa_x} \text{ at } \kappa_x = 0 \right)$$

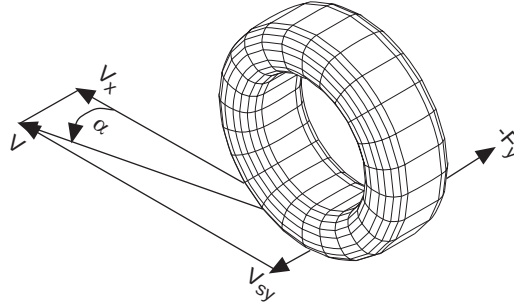
$$B_x = K_x / (C_x D_x) \quad (\text{A-25})$$

$$S_{Hx} = (p_{Hx1} + p_{Hx2} \cdot df_z) \lambda_{Hx} \quad (\text{A-26})$$

$$S_{Vx} = F_z \cdot (p_{Vx1} + p_{Vx2} df_z) \cdot \lambda_{Vx} \cdot \lambda_{\mu x} \quad (\text{A-27})$$

Table A-9: Longitudinal Coefficients, Pure Slip

Name:	Name used in tire property file:	Explanation:
p _{Cx1}	PCX1	Shape factor Cfx for longitudinal force
p _{Dx1}	PDX1	Longitudinal friction Mux at Fznom
p _{Dx2}	PDX2	Variation of friction Mux with load
p _{Dx3}	PDX3	Variation of friction Mux with camber
p _{Ex1}	PEX1	Longitudinal curvature Efx at Fznom
p _{Ex2}	PEX2	Variation of curvature Efx with load
p _{Ex3}	PEX3	Variation of curvature Efx with load squared
p _{Ex4}	PEX4	Factor in curvature Efx while driving
p _{Kx1}	PKX1	Longitudinal slip stiffness Kfx/Fz at Fznom
p _{Kx2}	PKX2	Variation of slip stiffness Kfx/Fz with load
p _{Kx3}	PKX3	Exponent in slip stiffness Kfx/Fz with load
p _{Hx1}	PHX1	Horizontal shift Shx at Fznom
p _{Hx2}	PHX2	Variation of shift Shx with load
p _{Vx1}	PVX1	Vertical shift Sv _x /Fz at Fznom

Formula: Lateral Slip (Pure Slip)**Figure A-10. Lateral Slip Condition Excluding Aligning Torque (Pure Cornering)**

$$F_y = F_{y0}(\alpha, \gamma, F_z) \quad (\text{A-28})$$

$$F_{y0} = D_y \sin[C_y \arctan\{B_y \alpha_y - E_y(B_y \alpha_y - \arctan(B_y \alpha_y))\}] + S_{Vy} \quad (\text{A-29})$$

$$\alpha_y = \alpha + S_{Hy} \quad (\text{A-30})$$

the scaled camber angle:

$$\gamma_y = \gamma \cdot \lambda_{\gamma y} \quad (\text{A-31})$$

with coefficients:

$$C_y = p_{Cy1} \cdot \lambda_{Cy} \quad (\text{A-32})$$

$$D_y = \mu_y \cdot F_z \quad (\text{A-33})$$

$$\mu_y = (p_{Dy1} + p_{Dy2} df_z) \cdot (1 - p_{Dy3} \gamma_y^2) \cdot \lambda_{\mu y} \quad (\text{A-34})$$

$$E_y = (p_{Ey1} + p_{Ey2} df_z) \cdot \{1 - (p_{Ey3} + p_{Ey4} \gamma_y) \operatorname{sgn}(\alpha_y)\} \cdot \lambda_{Ey} (\leq 1) \quad (\text{A-35})$$

$$K_y = p_{Ky1} F_{z0} \sin[2 \arctan\{F_z / (p_{ky2} F_{z0} \lambda_{F_{z0}})\}] \cdot (1 - p_{Ky3} |\gamma_y|) \cdot \lambda_{F_{z0}} \cdot \lambda_{Ky} \\ ((= B_y C_y D_y = \frac{\partial F_{y0}}{\partial \alpha_y} \text{ at } \alpha_y = 0)) \quad (\text{A-36})$$

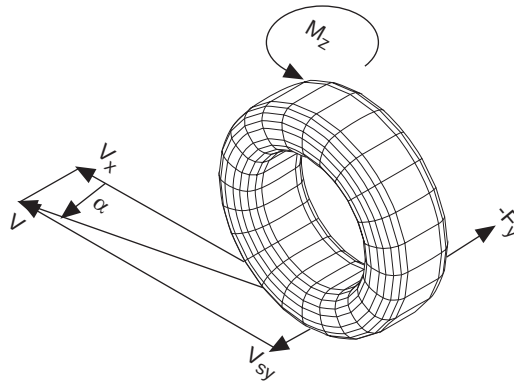
$$B_y = K_y / (C_y D_y) \quad (\text{A-37})$$

$$S_{Hy} = (P_{Hy1} + P_{Hy2} df_z) \cdot \lambda_{Hy} + P_{Hy3} \gamma_y \quad (\text{A-38})$$

$$S_{Vy} = F_z \cdot \{(p_{Vy1} + p_{Vy2} df_z) \cdot \lambda_{Vy} + (p_{Vy3} + p_{Vy4} \cdot df_z) \cdot \gamma_y\} \cdot \lambda_{\mu y} \quad (\text{A-39})$$

Table A-10: Lateral Coefficients, Pure Slip

Name:	Name used in tire property file:	Explanation:
p_{Cy1}	PCY1	Shape factor C_{fy} for lateral forces
p_{Dy1}	PDY1	Lateral friction μ_{uy}
p_{Dy2}	PDY2	Variation of friction μ_{uy} with load
p_{Dy3}	PDY3	Variation of friction μ_{uy} with squared camber
p_{Ey1}	PEY1	Lateral curvature E_{fy} at F_{znom}
p_{Ey2}	PEY2	Variation of curvature E_{fy} with load
p_{Ey3}	PEY3	Zero order camber dependency of curvature E_{fy}
p_{Ey4}	PEY4	Variation of curvature E_{fy} with camber
p_{Ky1}	PKY1	Maximum value of stiffness K_{fy}/F_{znom}
p_{Ky2}	PKY2	Load at which K_{fy} reaches maximum value
p_{Ky3}	PKY3	Variation of K_{fy}/F_{znom} with camber
p_{Hy1}	PHY1	Horizontal shift S_{hy} at F_{znom}
p_{Hy2}	PHY2	Variation of shift S_{hy} with load
p_{Hy3}	PHY3	Variation of shift S_{hy} with camber
p_{Vy1}	PVY1	Vertical shift in S_{vy}/F_z at F_{znom}
p_{Vy2}	PVY2	Variation of shift S_{vy}/F_z with load
p_{Vy3}	PVY3	Variation of shift S_{vy}/F_z with camber
p_{Vy4}	PVY4	Variation of shift S_{vy}/F_z with camber and load

Formula: Aligning Torque (Pure Slip)**Figure A-11. Lateral Slip Condition Including Aligning Torque (Pure Cornering)**

$$M'_z = M_{z0}(\alpha, \gamma, F_z) \quad (\text{A-40})$$

$$M_{z0} = -t \cdot F_{y0} + M_{zr} \quad (\text{A-41})$$

with the pneumatic trail:

$$t(\alpha_t) = D_t \cos[C_t \arctan\{B_t \alpha_t - E_t(B_t \alpha_t - \arctan(B_t \alpha_t))\}] \cos(\alpha) \quad (\text{A-42})$$

$$\alpha_t = \alpha + S_{Ht} \quad (\text{A-43})$$

the residual torque:

$$M_{zr}(\alpha_r) = D_r \cos[\arctan(B_r \alpha_r)] \cos(\alpha) \quad (\text{A-44})$$

$$\alpha_r = \alpha + S_{Hr} \quad (\text{A-45})$$

$$S_{Hf} = S_{Hy} + S_{Vy}/K_y \quad (\text{A-46})$$

the scaled camber angle:

$$\gamma_z = \gamma \cdot \lambda_{\gamma z} \quad (\text{A-47})$$

with coefficients:

$$B_t = (q_{Bz1} + q_{Bz2} df_z + q_{Bz3} df_z^2) \cdot (1 + q_{Bz4} \gamma_z + q_{Bz5} |\gamma_z|) \cdot \lambda_{Ky} / \lambda_{\mu y} \quad (\text{A-48})$$

$$C_t = q_{Cz1} \quad (\text{A-49})$$

$$D_t = F_z \cdot (q_{Dz1} + q_{Dz2} df_z) \cdot (1 + q_{Dz3} \gamma_z + q_{Dz4} \gamma_z^2) \cdot (R_0 / F_{z0}) \cdot \lambda_t \quad (\text{A-50})$$

$$E_t = (q_{Ez1} + q_{Ez2} df_z + q_{Ez3} df_z^2) \quad (\text{A-51})$$

$$\left\{ 1 + (q_{Ez4} + q_{Ez5} \gamma_z) \cdot \left(\frac{2}{\pi} \right) \cdot \arctan(B_t \cdot C_t \cdot \alpha_t) \right\} \leq 1$$

$$S_{Ht} = q_{Hz1} + q_{Hz2} df_z + (q_{Hz3} + q_{Hz4} \cdot df_z) \gamma_z \quad (\text{A-52})$$

$$B_r = q_{Bz9} \cdot \lambda_{Ky} / \lambda_{\mu y} + q_{Bz10} \cdot B_y \cdot C_y \quad (\text{A-53})$$

$$D_r = F_z \cdot ((q_{Dz6} + q_{Dz7} \cdot df_z) \cdot \lambda_r + (q_{Dz8} + q_{Dz9} \cdot df_z) \cdot \gamma_z) \cdot R_o \cdot \lambda_{\mu \gamma} \quad (\text{A-54})$$

An approximation for the aligning stiffness reads:

$$K_z = -t \cdot K_y \quad \left(\approx -\frac{\partial M_z}{\partial \alpha} \text{ at } \alpha \right) = 0) \quad (\text{A-55})$$

Table A-11: Aligning Coefficients, Pure Slip

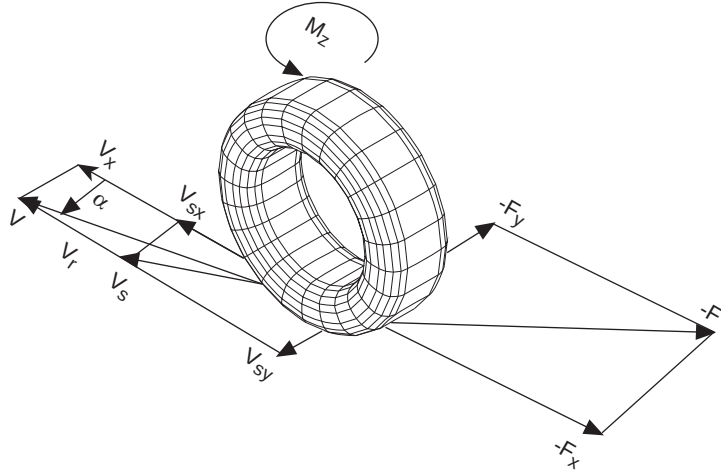
Name:	Name used in tire property file:	Explanation:
q _{Bz1}	QBZ1	Trail slope factor for trail Bpt at Fznom
q _{Bz2}	QBZ2	Variation of slope Bpt with load
q _{Bz3}	QBZ3	Variation of slope Bpt with load squared
q _{Bz4}	QBZ4	Variation of slope Bpt with camber
q _{Bz5}	QBZ5	Variation of slope Bpt with absolute camber
q _{Bz9}	QBZ9	Slope factor Br of residual torque Mzr
q _{Bz10}	QBZ10	Slope factor Br of residual torque Mzr
q _{Cz1}	QCZ1	Shape factor Cpt for pneumatic trail
q _{Dz1}	QDZ1	Peak trail Dpt'' = Dpt*(Fz/Fznom*R0)
q _{Dz2}	QDZ2	Variation of peak Dpt'' with load
q _{Dz3}	QDZ3	Variation of peak Dpt'' with camber
q _{Dz4}	QDZ4	Variation of peak Dpt'' with camber squared.

Table A-11: Aligning Coefficients, Pure Slip

Name:	Name used in tire property file:	Explanation:
q_{Dz6}	QDZ6	Peak residual torque $Dmr'' = Dmr / (Fz * R0)$
q_{Dz7}	QDZ7	Variation of peak factor Dmr'' with load
q_{Dz8}	QDZ8	Variation of peak factor Dmr'' with camber
q_{Dz9}	QDZ9	Variation of peak factor Dmr'' with camber and load
q_{Ez1}	QEZ1	Trail curvature Ept at $Fznom$
q_{Ez2}	QEZ2	Variation of curvature Ept with load
q_{Ez3}	QEZ3	Variation of curvature Ept with load squared
q_{Ez4}	QEZ4	Variation of curvature Ept with sign of $\alpha-t$
q_{Ez5}	QEZ5	Variation of Ept with camber and sign $\alpha-t$
q_{Hz1}	QHZ1	Trail horizontal shift Sht at $Fznom$
q_{Hz2}	QHZ2	Variation of shift Sht with load
q_{Hz3}	QHZ3	Variation of shift Sht with camber
q_{Hz4}	QHZ4	Variation of shift Sht with camber and load

A.6.2 Magic Formula Steady-State Combined Slip

Figure A-12. Combined Slip Condition (Combined Braking/Traction and Cornering)



Formula: Longitudinal Slip (Combined Slip)

$$F_x = F_{x0} \cdot G_{x\alpha}(\alpha, \kappa, F_z) \quad (\text{A-56})$$

with $G_{x\alpha}$ a weighting function.

We write:

$$F_x = D_{x\alpha} \cos[C_{x\alpha} \arctan\{B_{x\alpha}\alpha_s - E_{x\alpha}(B_{x\alpha}\alpha_s - \arctan(B_{x\alpha}\alpha_s))\}] \quad (\text{A-57})$$

$$\alpha_s = \alpha + S_{Hx\alpha} \quad (\text{A-58})$$

with coefficients:

$$B_{x\alpha} = r_{Bx1} \cos[\arctan\{r_{Bx2}\kappa\}] \cdot \lambda_{x\alpha} \quad (\text{A-59})$$

$$C_{x\alpha} = r_{Cx1} \quad (\text{A-60})$$

$$D_{x\alpha} = \frac{F_{xo}}{\cos[C_{x\alpha} \arctan\{B_{x\alpha}S_{Hx\alpha} - E_{x\alpha}(B_{x\alpha}S_{Hx\alpha} - \arctan(B_{x\alpha}S_{Hx\alpha}))\}]} \quad (\text{A-61})$$

$$E_{x\alpha} = r_{Ex1} + r_{Ex2}df_z \quad (\text{A-62})$$

$$S_{Hx\alpha} = r_{Hx1} \quad (\text{A-63})$$

The weighting function follows as:

$$G_{x\alpha} = \frac{\cos[C_{x\alpha} \arctan\{B_{x\alpha}\alpha_s - E_{x\alpha}(B_{x\alpha}\alpha_s - \arctan(B_{x\alpha}\alpha_s))\}]}{\cos[C_{x\alpha} \arctan[B_{x\alpha}S_{Hx\alpha} - E_{x\alpha}(B_{x\alpha}S_{Hx\alpha} - \arctan(B_{x\alpha}S_{Hx\alpha}))]]} \quad (\text{A-64})$$

Table A-12: Longitudinal Coefficients, Combined Slip

Name:	Name used in tire property file:	Explanation:
r_{Bx1}	RBX1	Slope factor for combined slip Fx reduction
r_{Bx2}	RBX2	Variation of slope Fx reduction with kappa
r_{Cx1}	RCX1	Shape factor for combined slip Fx reduction
r_{Ex1}	REX1	Curvature factor of combined Fx
r_{Ex2}	REX2	Curvature factor of combined Fx with load
r_{Hx1}	RHX1	Shift factor for combined slip Fx reduction

Formula: Lateral Slip (Combined Slip)

$$F_y = F_{y0} \cdot G_{y\kappa}(\alpha, \kappa, \gamma, F_z) + S_{Vy\kappa} \quad (\text{A-65})$$

with $G_{y\kappa}$ a weighting function and $S_{Vy\kappa}$ the 'k-induced' side force can be written:

$$F_y = D_{y\kappa} \cos[C_{y\kappa} \arctan\{B_{y\kappa}\kappa_s - E_{y\kappa}(B_{y\kappa}\kappa_s - \arctan(B_{y\kappa}\kappa_s))\}] + S_{Vy\kappa} \quad (\text{A-66})$$

$$\kappa_s = \kappa + S_{Hy\kappa} \quad (\text{A-67})$$

with coefficients:

$$B_{y\kappa} = r_{By1} \cos[\arctan\{r_{By2}(\alpha - r_{By3})\}] \cdot \lambda_{y\kappa} \quad (\text{A-68})$$

$$C_{y\kappa} = r_{Cy1} \quad (\text{A-69})$$

$$D_{y\kappa} = \frac{F_{y0}}{\cos[C_{y\kappa} \arctan\{B_{y\kappa}S_{Hy\kappa} - E_{y\kappa}(B_{y\kappa}S_{Hy\kappa} - \arctan(B_{y\kappa}S_{Hy\kappa}))\}]} \quad (\text{A-70})$$

$$E_{y\kappa} = r_{Ey1} + r_{Ey2}df_z \quad (\text{A-71})$$

$$S_{Hy\kappa} = r_{Hy1} + r_{Hy2}df_z \quad (\text{A-72})$$

$$S_{Vy\kappa} = D_{Vy\kappa} \sin[r_{Vy5} \arctan(r_{Vy6}\kappa)] \cdot \lambda_{Vy\kappa} \quad (\text{A-73})$$

$$D_{Vy\kappa} = \mu_y F_z \cdot (r_{Vy1} + r_{Vy2}df_z + r_{Vy3}\gamma) \cdot \cos[\arctan(r_{Vy4}\alpha)] \quad (\text{A-74})$$

The weighting function appears to read:

$$G_{y\kappa} = \frac{[\cos[C_{y\kappa} \arctan\{B_{y\kappa}\kappa_s - E_{y\kappa}(B_{y\kappa}\kappa_s - \arctan(B_{y\kappa}\kappa_s))\}]]}{(\cos[C_{y\kappa} \arctan\{B_{y\kappa}S_{Hy\kappa} - E_{y\kappa}(B_{y\kappa}S_{Hy\kappa} - \arctan(B_{y\kappa}S_{Hy\kappa}))\}])} \quad (\text{A-75})$$

Table A-13: Lateral Coefficients, Combined Slip

Name:	Name used in tire property file:	Explanation:
r_{By1}	RBV1	Slope factor for combined Fy reduction
r_{By2}	RBV2	Variation of slope Fy reduction with alpha
r_{By3}	RBV3	Shift term for alpha in slope Fy reduction
r_{Cy1}	RCY1	Shape factor for combined Fy reduction
r_{Ey1}	REY1	Curvature factor of combined Fy
r_{Ey2}	REY2	Curvature factor of combined Fy with load
r_{Hy1}	RHY1	Shift factor for combined Fy reduction
r_{Hy2}	RHY2	Shift factor for combined Fy reduction with load
r_{Vy1}	RVY1	Kappa induced side force Svyk/Muy*Fz at Fznom
r_{Vy2}	RVY2	Variation of Svyk/Muy*Fz with load
r_{Vy3}	RVY3	Variation of Svyk/Muy*Fz with camber
r_{Vy4}	RVY4	Variation of Svyk/Muy*Fz with alpha
r_{Vy5}	RVY5	Variation of Svyk/Muy*Fz with kappa
r_{Vy6}	RVY6	Variation of Svyk/Muy*Fz with atan (kappa)

Formula: Aligning Torque (Combined Slip)

$$M'_z = -t \cdot F'_y + M_{zr} + s \cdot F_x \quad (\text{A-76})$$

with:

$$t = t(\alpha_{t,eq}) \quad (\text{A-77})$$

$$= D_t \cos[C_t \arctan\{B_t \alpha_{t,eq} - E_t(B_t \alpha_{t,eq} - \arctan(B_t \alpha_{t,eq}))\}] \cos(\alpha)$$

$$F'_{y,\gamma=0} = F_y - S_{Vy\kappa} \quad (\text{A-78})$$

$$M_{zr} = M_{zr}(\alpha_{r,eq}) = D_r \cos[\arctan(B_r \alpha_{r,eq})] \cos(\alpha) \quad (\text{A-79})$$

$$s = \{s_{sz1} + s_{sz2}(F_y/F_{z0}) + (s_{sz3} + s_{sz4}df_z)\gamma\} \cdot R_0 \cdot \lambda_s \quad (\text{A-80})$$

with the arguments:

$$\alpha_{t,eq} = \arctan \sqrt{\tan^2 \alpha_t + \left(\frac{K_x}{K_y}\right)^2 \kappa^2} \cdot \text{sgn}(\alpha_t) \quad (\text{A-81})$$

$$\alpha_{r,eq} = \arctan \sqrt{\tan^2 \alpha_r + \left(\frac{K_x}{K_y}\right)^2 \kappa^2} \cdot \text{sgn}(\alpha_r) \quad (\text{A-82})$$

Table A-14: Aligning Torque, Combined Slip

Name:	Name used in tire property file:	Explanation:
s_{sz1}	SSZ1	Nominal value of s/R0 effect of Fx on Mz
s_{sz2}	SSZ2	Variation of distance s/R0 with Fy/Fznom
s_{sz3}	SSZ3	Variation of distance s/R0 with camber
s_{sz4}	SSZ4	Variation of distance s/R0 with load and camber

Formula: Overturning Moment

$$M_x = R_0 \cdot F_z \cdot \{q_{sx1} \cdot \lambda_{Vmx} + (-q_{sx2} \cdot \gamma + q_{sx3} \cdot F_y/F_{z0}) \cdot \lambda_{Mx}\} \quad (\text{A-83})$$

Table A-15: Overturning Coefficients

Name:	Name used in tire property file:	Explanation:
q _{sx1}	QSX1	Lateral force induced overturning couple
q _{sx2}	QSX2	Camber induced overturning couple
q _{sx3}	QSX3	Fy induced overturning couple

Formula: Rolling Resistance Torque

$$M_y = R_o \cdot F_z \cdot \{q_{Sy1} + q_{Sy2} F_x / F_{z0} + q_{Sy3} |V_x / V_{ref}| + q_{Sy4} (V_x / V_{ref})^4\} \tag{A-84}$$

If q_{sy1} and q_{sy2} are both zero, then the following is true (as in MF-Tyre 5.0):

$$M_y = R_0 (S_{Vx} + K_x \cdot S_{Hx}) \tag{A-85}$$

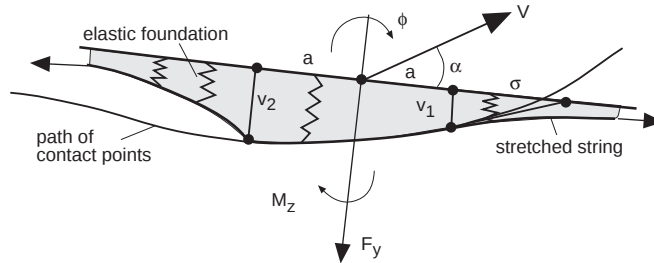
Table A-16: Rolling Coefficients

Name:	Name used in tire property file:	Explanation:
q _{sy1}	QSY1	Rolling resistance torque coefficient
q _{sy2}	QSY2	Rolling resistance torque depending on Fx
q _{sy3}	QSY3	Rolling resistance torque depending on speed
q _{sy4}	QSY4	Rolling resistance torque depending on speed^4
V _{ref}	LONGVL	Measurement speed

A

A.6.3 Transient Behavior

Figure A-13. Stretched String Model for Transient Tire Behavior



A.6.4 Transient Model Equations

The present version, using slip speeds instead of a and k , allows starting from stand-still. First-order lag of tire longitudinal and lateral deformations u and v are introduced through relaxation lengths s_k and s_α , see [Figures A-13](#):

$$\sigma_k \frac{du}{dt} + |V_x|u = -\sigma_k V_{sx} \quad (\text{A-86})$$

$$\sigma_\alpha \frac{dv}{dt} + |V_x|v = \sigma_\alpha V_{sy} \quad (\text{A-87})$$

These differential equations are based on the assumption that the contact points near the leading edge remain in the adhesion with the road surface (no sliding). The relaxation lengths (in this version not considered to decrease with increasing composite deformation slip) are functions of the vertical load and camber angle represented in a similar way as the slip stiffnesses K_x (Eq. A-12) and K_y (Eq. A-23).

$$\sigma_k = F_z \cdot (p_{Tx1} + p_{Tx2} df_z) \cdot \exp(-p_{Tx3} df_z) \cdot (R_0/F_{z0}) \cdot \lambda_{\sigma k} \quad (\text{A-88})$$

$$\sigma_\alpha = p_{Ty1} \sin[2 \arctan\{F_z/(p_{Ty2} F_{z0} \lambda_{Fz0})\}] \cdot (1 - p_{Ky3} |\gamma|) \cdot R_0 \lambda_{Fz0} \lambda_{\sigma \alpha} \quad (\text{A-89})$$

The practical tire deformation slip quantities are defined as:

$$\kappa' = \frac{u}{\sigma_k} \cdot \text{sign}(V_x) \quad (\text{A-90})$$

$$\tan \alpha' = \frac{v}{\sigma_\alpha} \quad (\text{A-91})$$

Equations (A-56), (A-65), (A-76), (A-83), and (A-84) are subsequently used with arguments k' and a' from Equations (A-90) and (A-91) instead of the longitudinal and lateral wheel slip quantities k and a (Equations (A-9) and (A-10)).

$$F_x = F_x(\alpha', \kappa', F_z) \tag{A-92}$$

$$F_y = F_y(\alpha', \kappa', \gamma, F_z) \tag{A-93}$$

$$M_z' = M_z'(\alpha', \kappa', \gamma, F_z) \tag{A-94}$$

A.6.5 The Gyroscopic Couple

This moment due to tire inertia acting about the vertical axis reads:

$$M_{z, gyr} = c_{gyr} m_{belt} V_{rl} \frac{dv}{dt} \cos[\arctan(B_r \alpha_{r, eq})] \tag{A-95}$$

with parameter (in addition to the basic tire parameter m_{belt}):

$$c_{gyr} = q_{Tzl} \cdot \lambda_{gyr} \tag{A-96}$$

and

$$\cos[\arctan(B_r \alpha_{r, eq})] = 1 \tag{A-97}$$

for pure cornering conditions.

The total aligning torque now becomes:

$$M_z = M_z' + M_{z, gyr} \tag{A-98}$$



Table A-17: Coefficients, Transient Response

Name:	Name used in tire property file:	Explanation:
pTx1	PTX1	Relaxation length SigKap0/Fz at Fznom
pTx2	PTX2	Variation of SigKap0/Fz with load
pTx3	PTX3	Variation of SigKap0/Fz with exponent of load
pTy1	PTY1	Peak value of relaxation length Sig_alpha

Table A-17: Coefficients, Transient Response

Name:	Name used in tire property file:	Explanation:
p_{Ty2}	PTY2	Shape factor for Sig_alpha
q_{Tz1}	QTZ1	Gyroscopic torque constant
M_{belt}	MBELT	Belt mass of the wheel

A.6.6 Switching from a Simple to a Complex Tire Model

MF-Tyre enables the user to switch from a simple tire model (for example only calculations for steady state pure cornering slip conditions) to tire model for transient combined slip situations. The parameter USE_MODE of the MF-Dataset determines the type of use of the tire model. In the [Table A-18](#) the possible options of USE_MODE are indicated. Note that the maximum valid USE_MODE depends on the tire test data used to determine the MF-Dataset parameters (that is, if only tire test data for pure cornering is fit-ted, the calculation of the contact forces under combined cornering and braking/traction slip is not possible unless the user adds the required additional parameters).

Table A-18: The Different USE_MODE Values of MF-Tyre

USE MODE:	State:	Slip conditions	MF-Tyre output (forces and torques)
0	spring	-	$0, 0, F_z, 0, 0, 0$
1	steady state	pure longitudinal	$F_x, 0, F_z, 0, M_y, 0$
2	steady state	pure lateral	$0, F_y, F_z, M_x, 0, M_z$
3	steady state	longitudinal and lateral (not combined)	$F_x, F_y, F_z, M_x, M_y, M_z$
4	steady state	combined slip forces	$F_x, F_y, F_z, M_x, M_y, M_z$
11	transient	pure longitudinal	$F_x, 0, F_z, 0, M_y, 0$
12	transient	pure lateral	$0, F_y, F_z, M_x, 0, M_z$

Table A-18: The Different USE_MODE Values of MF-Tyre

USE MODE:	State:	Slip conditions	MF-Tyre output (forces and torques)
13	transient	longitudinal and lateral (not combined)	$F_x, F_y, F_z, M_x, M_y, M_z$
14	transient	combined slip forces	$F_x, F_y, F_z, M_x, M_y, M_z$

A.7 Some Practical Aspects

A.7.1 Rolling Resistance Torque

For a free rolling wheel at a constant forward velocity without camber and slip angle a drag force (rolling resistance) is generated. Passenger car tires usually have a rolling resistance coefficient between 0.7-1.2%; for truck tires the rolling resistance force is usually around 0.5% to 0.7% of the vertical load. Note that the parameter q_{sy1} in equation (A-80) determines the rolling resistance factor. According to the ISO sign convention this drag force as well as the rolling resistance torque M_y have negative signs ($q_{sy1} > 0$).

In order to reach equilibrium between the force and the torque on the wheel, in general a small negative value for the longitudinal slip is obtained.

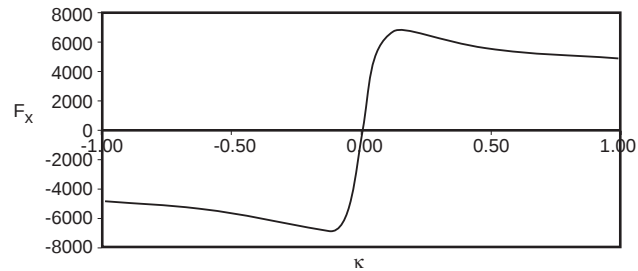
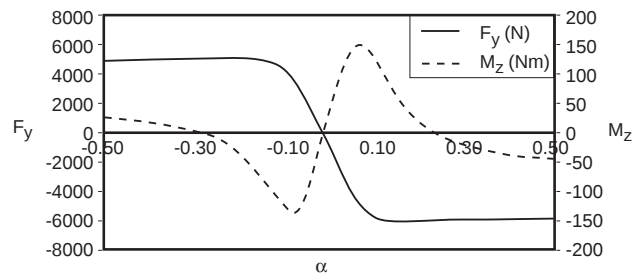
A.7.2 Typical Tire Characteristics

For pure slip conditions (either longitudinal or lateral) three typical graphs can be made:

- F_x as a function of the longitudinal slip κ ;
- F_y as a function of the slip angle α ;
- M_z as a function of the slip angle α .

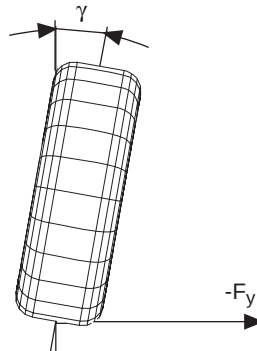
In [Figures A-14](#) and [A-15](#), examples of these characteristics valid for the W-axis system are shown.



Figure A-14. Longitudinal Force as a Function of Longitudinal Slip**Figure A-15. The Lateral Force and Self-Aligning Torque as a Function of the Slip Angle**

A.7.3 Effect of Camber Angle

According to the W-axis system, an increase of the camber angle causes a decrease of the lateral force, as shown in Figure A-16.

Figure A-16. Tire Camber Angle and the Positive Direction of the Lateral Force According to the W-Axis System (Rear View)

A.7.4 Tire Model Output at Extreme Input Values

At extreme large input values, like a vertical load more than 3 times the nominal tire load, a real physical tire might puncture or go to pieces. In the tire model measures have been taken to avoid calculation errors or a computer simulation break down. Depending on your simulation software the tire model warns the user when the input exceeds the validity range of the MF-Dataset.

The tire property files, generated by MF-Tool, contain maxima and minima values for the tire model input, defining the validity range of the MF-Dataset:

- F_{zmin} and F_{zmax} for the vertical load F_z
- α_{pmin} and α_{pmax} for the slip angle α
- γ_{min} and γ_{max} for the camber angle γ
- k_{pmin} and k_{pmax} for the longitudinal slip k .

In general the tire model fixes the **B, C, D, E** and shift factors when exceeding the upper mentioned limits at the corresponding limit. For vertical loads smaller than **F_{zmin}** the output of the tire model is equal to the output of the tire model for **F_{zmin}** proportionally scaled to zero output.

A.8 Standard Tire Interface (STI)

As a result of the First International Colloquium on Tire Models for Vehicle Dynamics Analysis on October 21-22, 1991, the international Tire Workshop working group was established (TYDEX).

The working group concentrated on tire measurements and tire models used for vehicle simulation purposes. For most vehicle dynamics studies people usually develop their own tire models. Since all car manufacturers and their tire suppliers have the same goal (that is development of tires to improve dynamic safety of the vehicle) standardization in tire behavior description should be aimed for.

In TYDEX two expert groups were defined with following goals:

- The first expert group (Tire Measurements - Tire Modelling) has as its main goal to specify an interface between tire measurements and tire models. The work shall include a description of the test

conditions. The interface could be described as a definition of a method or format to describe tire measurement data in such a way that it contains all necessary items to fit tire models to the underlying data. The format shall also allow for a description of the test conditions.

- The second expert group (Tire Modelling - Vehicle Modelling) has as its main goal to specify an interface between tire models and simulation tools. Intentionally, use of this interface will ensure that a wide range of simulation software can be linked to a wide range of tire software available.

Both expert groups consist of participants of vehicle industry (passenger cars and trucks), tire manufacturers, other suppliers and research laboratories. The large number of participants indicates that there is a need for this kind of 'standardization' work. DVR is strongly involved in TYDEX.

At the Second International Colloquium on Tire Models for Vehicle Dynamics Analysis on February 19 and 20, 1997 the final documents on both interfaces have been presented [9]. The TYDEX-Format [1] describes a standard format for the exchange of tire testing and modelling data; the second document describes the standard interface between tire model and vehicle model, called the Standard Tire Interface (STI) [2].

At the moment, a concept for the description of the Tire Modeling - Vehicle Modeling interface have been developed and will be tested within the different companies. This interface is named the Standard Tire Interface (STI) [2].

The Standard Tire Interface prescribes a subroutine call with a number of subroutine arguments to pass all relevant information from tire models to multi-body programs and vice versa. The subroutine represents a shell around tire software and is fixed to the axle hub which is modelled by the multi-body programs.

A.9 MF-Datasets and MF-Tool

The final objective of the user is to optimize vehicle behavior (including tire behavior) using the potential of simulation software. Because the tire properties determine to a great extent the vehicle behavior, a tire model without proper tire data will be useless in most cases. For full

optimization purposes the engineer requires the availability of datasets under a large range of conditions.

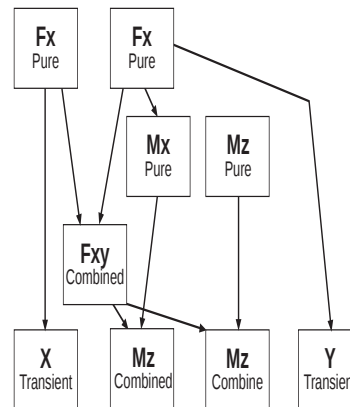
A.9.1 Tire Measurements

Tire characteristics can be well described by the Magic Formula tire model. The formulae are specified by a set of Magic Formula parameters that represent the characteristics in a compact form. The parameters depend on the type of the tire and the road conditions and can be obtained from outdoor and/or laboratory tests.

A.9.2 Calculation of Magic Formula Parameters

Calculation of parameters from the measurement data is performed with regression techniques (also known as parameter fitting ref [8]). In such a so called fitting procedure, the results from measurements under pure slip conditions have to be used first to determine the Magic Formula parameters for side force, self aligning torque and longitudinal force and in a second step the parameters for combined slip conditions, see Figure A-17. The pure cornering measurements must include the influence of camber. The parameters for transient cornering and braking are based on the steady state pure cornering and braking properties

Figure A-17. Fit process



The MF-Tool+ software of Mf-Tyre offers the engineer a user-friendly tool to determine the MF-Tyre parameters (MF-Datasets) out of any Force and Moment tyre test data. Next to software also MF-Datasets can be selected out of existing Libraries. See www.delft-tyre.com.

A.10 Definitions

General

Table A-19: General definitions

Term:	Definition:
Inertial coordinate system	Inertial space according to ISO
Road tangent plane	Plane with the normal unit vector n_r (tangent to the road) in C.
Wheel centre O	Centre of the wheel
C-axis system	Coordinate system mounted on the wheel carrier at the Wheel center orientation according ISO.
Wheel plane	The plane in the wheel centre that is formed by the wheel when considered a rigid disc with zero width.
Contact point C	Contact point between tyre and road, defined as the intersection of the wheel plane and the projection of the wheel axis onto the road plane.
W-axis system	Coordinate system at the tyre contact point C, orientation according ISO.

Tire Kinematics

Table A-20: Tire Kinematics Definitions

Abbreviation:	Definition:	Units:
R_0	Unloaded tire radius	[m]
R	Loaded tire radius	[m]
R_e	Effective tire radius	[m]
r_t	Tire cross section radius (half tyre width)	[m]
ρ	Radial tire deflection	[m]
ρ^d	Dimensionless radial tire deflection	[-]

Table A-20: Tire Kinematics Definitions

Abbreviation:	Definition:	Units:
ρ_{Fz0}	Radial tire deflection at nominal load	[m]
m_{belt}	Tire belt mass	[kg]
Ω	Rotational velocity of the wheel	[rads ⁻¹]
h_{α}	Distance wheel centre to road plane	[m]

Slip Quantities

Table A-21: Slip Quantities Definitions

Abbreviation:	Definition:	Units:
V	Vehicle speed	[ms ⁻¹]
V_{sx}	Slip speed in x-direction	[ms ⁻¹]
V_{sy}	Slip speed in y-direction	[ms ⁻¹]
V_s	Resulting slip speed	[ms ⁻¹]
V_x	Rolling speed in x-direction	[ms ⁻¹]
V_y	Lateral speed of tire contact center	[ms ⁻¹]
V_r	Linear speed of rolling	[ms ⁻¹]
κ	Longitudinal slip	[-]
α	Slip angle	[rad]
γ	Camber angle	[rad]



Forces and Moments

Table A-22: Force and Moment Definitions

Abbreviation:	Definition:	Units:
F_z	vertical wheel load	[N]
F_{z0}	nominal (rated) load	[N]
df_z	dimensionless vertical load	[-]
F_x	longitudinal force	[N]
F_y	lateral force	[N]
F_z	nominal load	[N]
M_x	overturning couple	[Nm]
M_y	braking/driving moment	[Nm]
M_z	aligning moment	[Nm]

A.11 References

- 1. H.-J. Unrau, J. Zamow**
TYDEX-Format, Description and Reference Manual, Release 1.1, Initiated by the International Tire Working Group, July 1995.
- 2. A. Riedel**
Standard Tire Interface, Release 1.2, Initiated by the Tire Workgroup, June 1995.
- 3. E. Bakker, L. Nyborg, H.B. Pacejka**
Tire Modelling for Use in Vehicle Dynamics Studies, SAE paper 870421, 1987.
- 4. E. Bakker, H.B. Pacejka, L. Lidner**
A New Tire Model with an Application in Vehicle Dynamics Studies, SAE paper 890087, 1989.
- 5. H.B. Pacejka, E. Bakker**
The Magic Formula Tire Model, Proceedings 1st International Colloquium on Tire Models for Vehicle Dynamics Analysis, Swets & Zeitlinger B.V., Amsterdam/Lisse, 1993.
- 6. P. Bayle, J.F. Forissier, S. Lafon**

A New Tire Model for Vehicle Dynamics Simulations Combined Cornering and Braking - Driving Manoeuvres, Michelin France, 1990.

7. H.B. Pacejka

The Tire as a Vehicle Component, XXVI FSITA Congress Prague, June 16-23, 1996.

8. J.J.M. van Oosten, E. Bakker

Determination of Magic Formula Tire Model Parameters, Proceedings 1st International Colloquium on Tire Models for Vehicle Dynamics Analysis, Swets & Zeitlinger B.V., Amsterdam/Lisse, 1993.

9. J.J.M. van Oosten, H.-J. Unrau, G. Riedel, E. Bakker

TYDEX Workshop: Standardization of Data Exchange in Tyre Testing and Tyre Modelling, Proceedings of the 2nd International Colloquiums Tyre Models for Vehicle Dynamics Analysis, Vehicle System Dynamics, Volume 27, Swets & Zeitlinger, Amsterdam/Lisse, 1996.

Appendix B Using the SWIFT-Tyre Model



This chapter contains proprietary information of TNO. The contents of this chapter may not be disclosed to other parties, copied, nor duplicated for commercial use without the prior written permission of TNO.

The SWIFT-Tyre model combines a Magic Formula slip force description with a rigid ring model and has been validated by experiments up to frequencies of 60-100 Hz. Typical applications of the SWIFT-Tyre model are: durability studies, shimmy analysis, chassis control system evaluation (that is, ABS, ESP) and cornering on uneven roads.

This chapter includes the following sections:

- [Introduction](#)
- [Notation](#)
- [Force Evaluation](#)
- [Tire Model Parameters](#)
- [Tire Property File Example](#)
- [Road Property File Example](#)

B.1 Introduction

The Magic Formula is a widely used and accepted method for modelling tire forces and moments under steady-state rolling conditions. At higher excitation frequencies (>1 -2 Hz) relaxation effects and belt dynamics become important for the forces transmitted by the tire to the wheel centre. SWIFT combines a Magic Formula slip force calculation with a rigid ring model, thus greatly extending the frequency range where the tire model is valid. The SWIFT-Tyre model was developed in a joint cooperation between the Delft University of Technology and TNO Automotive under the guidance of Dr. Pacejka. Reference documentation can be found in [Section B.1.4, "References"](#).

B.1.1 Dynamics

The SWIFT-Tyre model is a rigid ring model, in which the tire belt is assumed to behave like a rigid body. This means that the model is accurate in the frequency range where the bending modes of the tire belt can be neglected, which, depending on the tire properties is up to 50 – 60 Hz for lateral behavior and up to 100 Hz for vertical and longitudinal behavior. SWIFT has been validated using measurements of a rolling tire (7 to 40 m/s) containing frequencies up to 120 Hz. The model includes essential gyroscopic effects.

B.1.2 Slip Force Calculation

SWIFT uses the Magic Formula for calculation of slip forces, providing an accurate representation of measurement results which usually are available up to 15 degrees side slip, 100% brake slip and 5 degrees of camber angle for different vertical load. For efficiency reasons SWIFT uses a single point contact for slip calculation and therewith is fully compatible with MF-Tyre. Due to the introduction of a so-called phase leading network for the pneumatic trail, SWIFT is suitable for path curvature with a wavelength in the order of two times the contact length. For braking/traction applications, wavelengths as small as half the contact length are well described. The transient slip behavior is well described up to full sliding, due to modelling of decrease in relaxation length for increased slip levels.

B.1.3 Road Input

The dynamic model has been validated for load variations up to 100 Hz, and the slip model for wavelengths as small as two times the contact length. SWIFT uses a single point contact model, which generally can be

applied as long as the road curvature is about half of the tire curvature. To be able to cope with shorter obstacles a method of describing enveloping behavior is applied. It is assumed that a measured road profile can be evaluated as a series of step obstacles, for which the enveloping effect of the tire is described with so-called basic functions. This method has been validated for isolated obstacles up to 10% of the tire radius, and provides an accurate prediction of vertical load, longitudinal force and wheel rotation fluctuations. Also with measured road profiles good correlation has been found with vehicle measurement data.

B.1.4 References

- [1] Zegelaar, P.W.A., The Dynamic Response of Tyres to Brake Torque Variations and Road Unevennesses, PhD Thesis, Delft University of Technology, The Netherlands, 1998
- [2] Maurice, J.P., Short Wavelength and Dynamic Tyre Behavior under Lateral and Combined Slip Conditions, PhD Thesis, Delft University of Technology, The Netherlands, 1999

B.2 Notation

The equations in the model are expressed in dimensionless quantities as much as possible. This is achieved by the introduction of various reference values that are described in this section.

The reference speed V_0 is the speed at which the contact slip characteristics are measured. This reference speed is used with the nominal tire radius (R_0) for calculation of the reference wheel rotational velocity (Ω_0) in accordance with Equation (B-1):

$$\Omega_0 = \frac{V_0}{R_0} \quad (\text{B-1})$$

The reference stiffness for translation C_{t0} , is derived from the nominal tire load (F_{z0}) and nominal tire radius (R_0) in accordance with Equation (B-2), and the reference stiffness for rotation C_{r0} is given in Equation (B-3):

$$C_{t0} = \frac{F_{z0}}{R_0} \quad (\text{B-2})$$

$$C_{r0} = F_{z0} R_0 \quad (\text{B-3})$$

Dimensionless damping is defined in Equation (B-4), in which m represents the inertia, c represents the stiffness and k the damping as is common for simple mass-spring systems. Additionally reference values for damping of translations (k_{t0}) and rotations (k_{r0}) are defined in Equation (B-5) and (B-6) respectively:

$$\kappa = \frac{k}{2\sqrt{mc}} \quad (\text{B-4})$$

$$k_{t0} = \sqrt{\frac{m_0 F_{z0}}{R_0}} \quad (\text{B-5})$$

$$k_{r0} = \sqrt{m_0 F_{z0} R_0^3} \quad (\text{B-6})$$

The reference moment of inertia (I_0) is calculated using Equation (B-7), in which m_0 is the mass of the tire:

$$I_0 = m_0 \cdot R_0^2 \quad (\text{B-7})$$

The normalized variables that occur in the equations are denoted by an overbar.

Table B-1: Forces and Moments

Symbol:	Description:	Units:	Normalized with:
F_{bx}	Longitudinal belt force	[N]	-
F_{by}	Lateral belt force	[N]	-
F_{bz}	Vertical belt force	[N]	-
F_{grv}	Gravity force	[N]	-
F_{rx}	Longitudinal residual force	[N]	-
F_{ry}	Lateral residual force	[N]	-
$\overline{F}_x$	Normalized longitudinal force	-	F_{z0}
$\overline{F}_y$	Normalized lateral force	-	F_{z0}
$\overline{F}_z$	Normalized vertical force	-	F_{z0}
F_z	Vertical axle load	[N]	-
F_{z0}	Nominal tire load	[N]	-

Table B-1: Forces and Moments

Symbol:	Description:	Units:	Normalized with:
M_{bx}	Camber belt torque	[Nm]	-
M_{by}	Wind-up belt torque	[Nm]	-
M_{bz}	Yaw belt torque	[Nm]	-
M_{rz}	Yaw residual torque	[Nm]	-

Table B-2: Radii

Symbol:	Description:	Units:	Normalized with:
D_{r0}	Speed radius increase	-	-
R_l	Loaded tire radius	[m]	-
R_0	Nominal tire radius	[m]	-
R_e	Effective rolling radius	[m]	-
R_W	Free tire radius	[m]	-

Table B-3: Inertia

Symbol:	Description:	Units:	Normalized with:
m_0	Tire mass	[kg]	-
$\overline{m}_b$	Normalized belt mass	-	m_0
I_0	Reference moment of inertia	[kgm ²]	-

B

Table B-4: General Coefficients

Symbol:	Description:	Units:	Normalized with:
c_{grv}	Gravity constant	-	-
$q_{bv\alpha}$	Correction coefficient radial belt stiffness	-	-

Table B-4: General Coefficients

Symbol:	Description:	Units:	Normalized with:
$q_{bv\theta}$	Correction coefficient tangential belt stiffness	-	-
q_{Fcx}	Brake force stiffness scaling coefficient	-	-
q_{Fcy}	Side force stiffness scaling coefficient	-	-
$q_{Fz1,2}$	Vertical force coefficients	-	-
$q_{kc1,2}$	Coefficients for tread element damping characteristics	-	-
q_{re0}	Tire radius scaling coefficient	-	-
Q_v	Speed and load correction coefficient	-	-
q_{v1}	Tire growth coefficient	-	-
q_{v2}	Vertical force speed coefficient	-	-

Table B-5: Displacements and Deflections

Symbol:	Description:	Units:	Normalized with:
ρ_{bx}	Longitudinal belt displacement	[m]	-
$\bar{\rho}_{bx}$	Normalized longitudinal belt displacement	-	R_0
$\rho_{b\gamma}$	Camber belt displacement	[rad]	-
ρ_{bz}	Vertical belt displacement	[m]	-
$\bar{\rho}_{bz}$	Normalized vertical belt displacement	-	R_0
$\bar{\rho}_{by}$	Normalized lateral belt displacement	-	R_0
$\rho_{b\theta}$	Wind-up belt displacement	[rad]	-
$\rho_{b\psi}$	Yaw belt displacement	[rad]	-
ρ^d	Dimensionless radial deflection	-	ρ_{Fz0}
ρ_{Fz0}	Nominal tire deflection	[m]	-
$\bar{\rho}_z$	Normalized vertical tire deflection	-	R_0

Table B-5: Displacements and Deflections

Symbol:	Description:	Units:	Normalized with:
$\bar{\rho}_{rx}$	Normalized longitudinal residual deflection	-	R_0
$\bar{\rho}_{ry}$	Normalized lateral residual deflection	-	R_0
$\rho_{r\psi}$	Yaw residual deflection	[rad]	-
W	Wheel rotational speed	[rad/s]	
Ω	Normalized wheel rotation speed	-	W
Ω_0	Nominal wheel rotation speed	[rad/s]	-
$V_{c,sx}$	Longitudinal contact point velocity	[m/s]	-
V_0	Nominal wheel speed	[m/s]	-
V_x	Wheel speed	[m/s]	-

Table B-6: Stiffness and Damping

Symbol:	Description:	Units:	Normalized with:
c_{bx}	Translation belt stiffness	[N/m]	-
$\bar{c}_{bx}$	Normalized in-plane translation belt stiffness	-	C_{t0}
$\bar{c}_{bx0}$	Normalized nominal in-plane translation belt stiffness	-	C_{t0}
c_{by}	Lateral belt stiffness	[N/m]	-
$\bar{c}_{by}$	Normalized out-of-plane translation belt stiffness	-	C_{t0}
$c_{b\gamma}$	Out-of-plane rotation belt stiffness	[N/m]	-
$\bar{c}_{b\gamma}$	Normalized out-of-plane rotation belt stiffness	-	C_{r0}
$c_{b\theta}$	Wind-up belt stiffness	[N/m]	-
$\bar{c}_{b\theta}$	Normalized in-plane rotation belt stiffness	-	C_{r0}

B.3 Force Evaluation

B.3.1 Rigid Ring Model

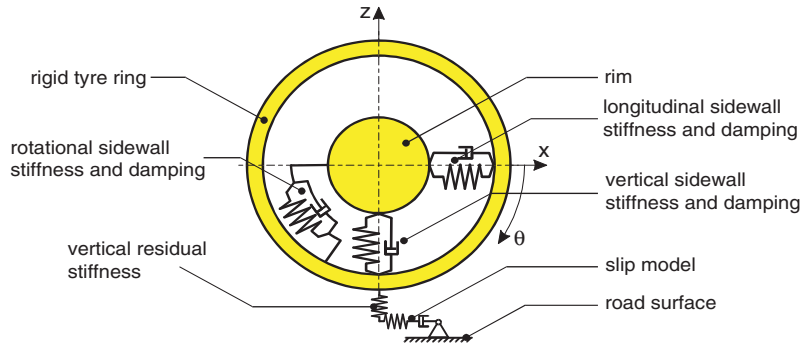
The tire belt is modelled as a rigid body with mass and moments of inertia, that is suspended with spring-damper systems to the rim. The stiffness of the springs is calculated from the frequencies of the so-called rigid body modes. The gravitational force is along the global Z-axis, and is defined in accordance with Equation (B-8):

$$F_{grv} = c_{grv} \bar{m}_b m_0 \quad (B-8)$$

B.3.2 In-Plane Characteristics

Figure B-1 shows a side view of the rigid-ring representation of the tire.

Figure B-1. Side View of the Rigid-Ring Representation of the Tire



For the in-plane behavior, the stiffness of the springs is dependent on the in-plane belt displacements (r_{bx} , r_{bz}) and rotating speed (W). The influence of speed and load is implemented by using a correction coefficient Q_v . The correction coefficient is defined by Equation (B-9):

$$Q_v = |\bar{\Omega}| \cdot \sqrt{\bar{\rho}_{bx}^2 + \bar{\rho}_{bz}^2} \quad (B-9)$$

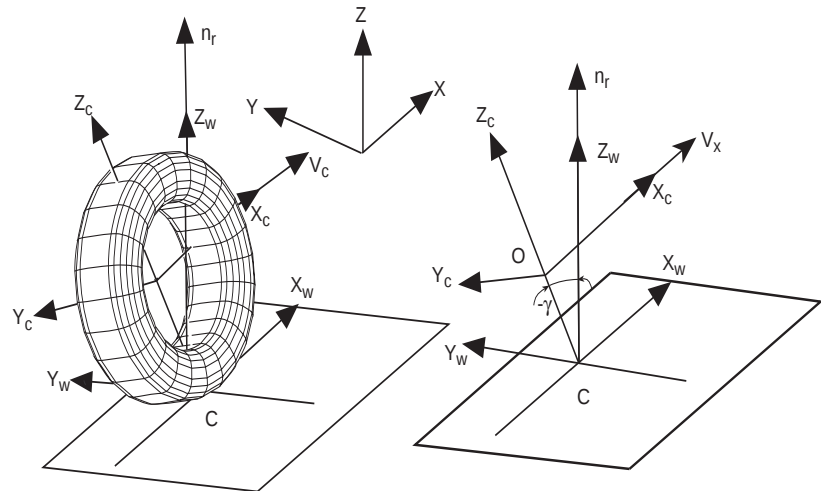
The nominal tire belt stiffness for the in-plane motions is corrected for deflection and speed in accordance with Equations (B-10) and (B-11):

$$\bar{c}_{bx} = \bar{c}_{bx0} (1 - q_{bvx} \sqrt{Q_v}) \quad (B-10)$$

$$\bar{c}_{b\theta} = \bar{c}_{b\theta 0}(1 - q_{bV\theta}\sqrt{Q_v}) \quad (\text{B-11})$$

This results in a dependency of the belt stiffness on speed and load typically as displayed in Figure B-2.

Figure B-2. Belt Stiffness for Various Speeds



The in-plane forces and torque that are transmitted by the tire belt to the rim are given in Equations (B-12), (B-13) and (B-14).

Longitudinal belt force:

$$F_{bx} = \bar{c}_{bx} \bar{\rho}_{bx} F_{z0} + 2\kappa_{bx} \dot{\bar{\rho}}_{bx} k_{t0} \quad (\text{B-12})$$

Vertical belt force:

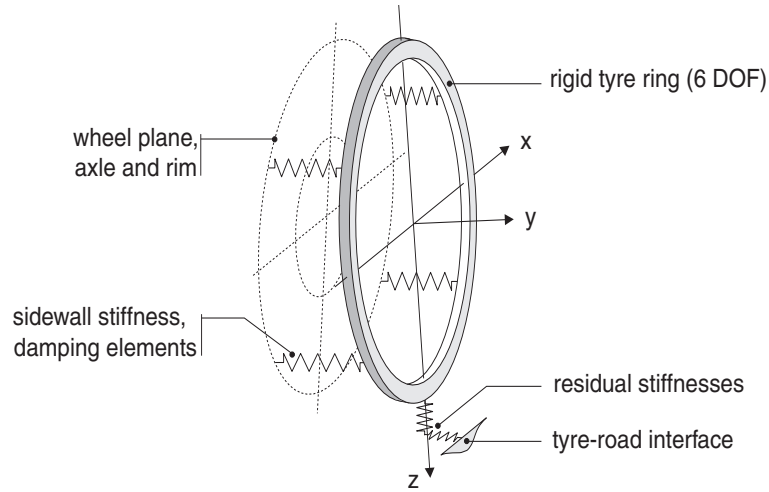
$$F_{bz} = \bar{c}_{bx} \bar{\rho}_{bz} F_{z0} + 2\kappa_{bx} \dot{\bar{\rho}}_{bz} k_0 \quad (\text{B-13})$$

Belt wind-up torque:

$$M_{by} = \bar{c}_{b\theta} \rho_{b\theta} C_{r0} + 2\kappa_{b\theta} \dot{\rho}_{b\theta} k_{r0} \quad (\text{B-14})$$

B.3.3 Out-of-Plane Characteristics

Figure B-3 shows the out-of-plane deflection of the tire, and related specifications.

Figure B-3. Out-of-Plane Deflection of the Tyre

The out-of-plane stiffness is not dependent on speed and/or load. The lateral force acting on the wheel carrier is given in Equation (B-15):

$$F_{by} = \bar{c}_{by} \bar{\rho}_{by} F_{z0} + 2\kappa_{by} \dot{\rho}_{by} k_{to} \quad (\text{B-15})$$

Belt camber torque is calculated using Equation (B-16), and the yaw torque is calculated using Equation (B-17):

$$M_{bx} = \bar{c}_{b\gamma} \bar{\rho}_{b\gamma} C_{r0} + 2\kappa_{b\gamma} \dot{\rho}_{b\gamma} k_{r0} \quad (\text{B-16})$$

$$M_{bz} = \bar{c}_{b\psi} \bar{\rho}_{b\psi} C_{r0} + 2\kappa_{b\psi} \dot{\rho}_{b\psi} k_{r0} \quad (\text{B-17})$$

B.3.4 Vertical Force Characteristics

The overall vertical tire force is the tire force that results from a steady state vertical deflection of the tire. As the vertical tire belt stiffness is modelled with the Rigid Ring model, a residual spring is introduced in order to achieve the overall vertical force characteristic of the tire (see Figure B-5). Changes in vertical force due to slip forces are incorporated in the vertical force calculation.

The vertical tire force is a function of deflection and speed. The tire deflection is used for the overall vertical force and is the difference between the free tire radius R_Ω and the loaded tire radius R_l . The free tire

radius is a function of speed as the tire grows with speed as given in Equation (B-18):

$$\Delta r_0 = q_{V1} \bar{\Omega}^2 \quad (\text{B-18})$$

The free tire radius is calculated using Equation (B-19), and is displayed as function of the wheel speed in Figure B-4:

$$R_\Omega = (q_{re0} + \Delta r_0) R_0 \quad (\text{B-19})$$

The normalized vertical tire deflection is defined by Equation (B-20):

$$\bar{\rho}_z = (R_\Omega - R_l) / R_0 \quad (\text{B-20})$$

The overall vertical tire force is a function of the tire deflection, wheel rotation velocity and slip forces as given in Equation (B-21):

$$F_z = (1 + q_{V2} |\bar{\Omega}| - (q_{FCx} \bar{F}_x)^2 - (q_{FCy} \bar{F}_y)^2) (q_{Fz2} \bar{\rho}_z^2 + q_{Fz1} \bar{\rho}_z) F_{z0} \quad (\text{B-21})$$

Figure B-4. Free Tire Radius as Function of Wheel Speed

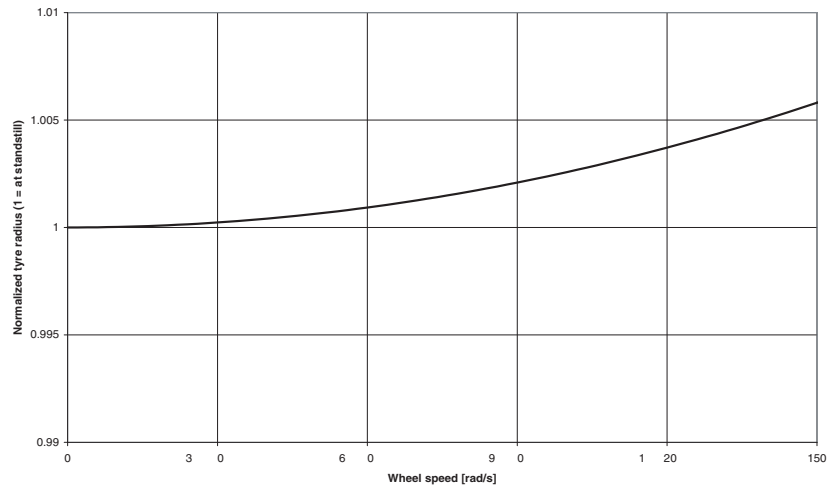
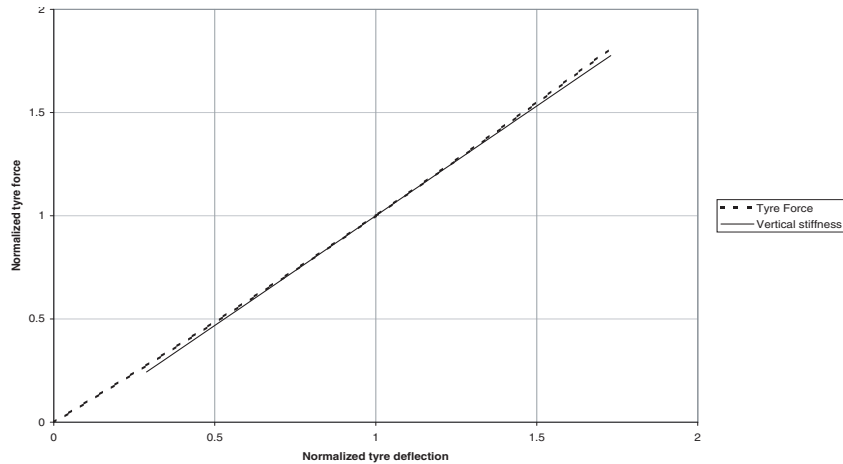
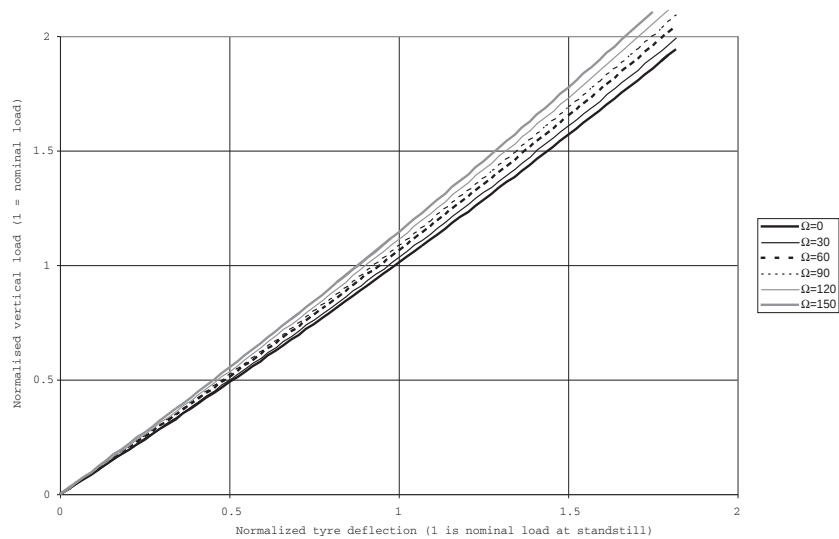
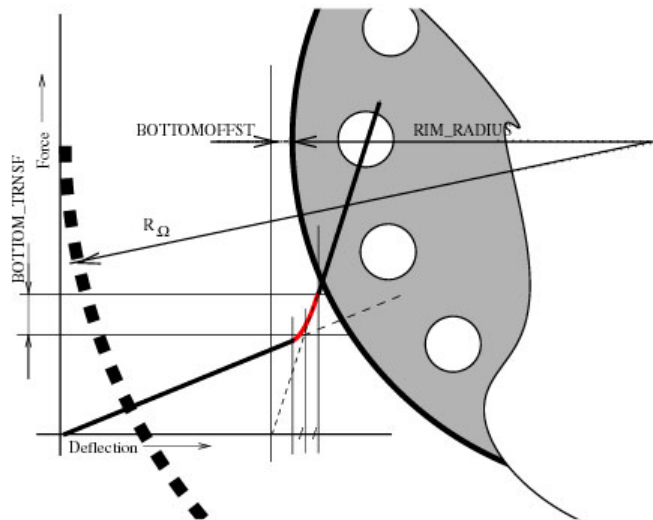


Figure B-5. Vertical Tire Force and Stiffness

As shown in Figure B-5, the vertical tire force as function of the deflection is a parabola. The vertical stiffness value in the tire property file C_z is the stiffness at the nominal tire load. The relation between C_z , q_{Fz1} and q_{Fz2} is given by equation (B-22):

$$C_z = \frac{F_{z0}}{R_0} \sqrt{q_{Fz1}^2 + 4q_{Fz2}} \quad (\text{B-22})$$

The vertical force characteristic for different wheel speeds is shown in Figure B-6.

Figure B-6. Wheel Load as Function of Tire Deflection at Different Wheel Speeds**Figure B-7. SWIFT with Bottoming Characteristics**

SWIFT incorporates bottoming effects for load-case studies as shown in Figure B-7. Bottoming occurs when the deflection of the tire results in

contact of the tire tread band with the wheel rim, a radius that generally will be somewhat larger than the rim radius (RIM_RADIUS). The assumption is made that the bottoming characteristics are independent from the normal vertical spring curve. Three parameters are required to define the bottoming characteristics:

BOTTOM_STIFF: Defines the linear vertical stiffness of the tyre-wheel assembly when the tyre is bottoming. As a first estimate a value of ten times the vertical stiffness may be appropriate.

BOTTOM_OFFST: Defines (in combination with RIM_RADIUS) the maximum radius where bottoming can start to occur, see Figure B-7. The actual point where the vertical force starts to increase is the point of intersection between the normal vertical spring and bottoming spring curve.

BOTTOM_TRNSF: Defines the size of the transition range where the normal spring curve is smoothly changed into the bottoming spring curve, see Figure B-7. The unit of this parameter is force.

B.3.5 The Effective Tire Rolling Radius

The effective tire-rolling radius R_e is estimated using a Magic Formula approach. Equation (B-23) holds the formula for the effective tire-rolling radius:

$$R_e = R_\Omega - \rho_{Fz0} (D \arctan((B \rho^d) + F \rho^d)) \quad (B-23)$$

The nominal tire deflection $rFz0$ is defined by Equation (B-24) (C_z = radial tire stiffness), and the dimensionless radial deflection is calculated using Equation (B-25):

$$\rho_{Fz0} = \frac{F_{z0}}{C_z} \quad (B-24)$$

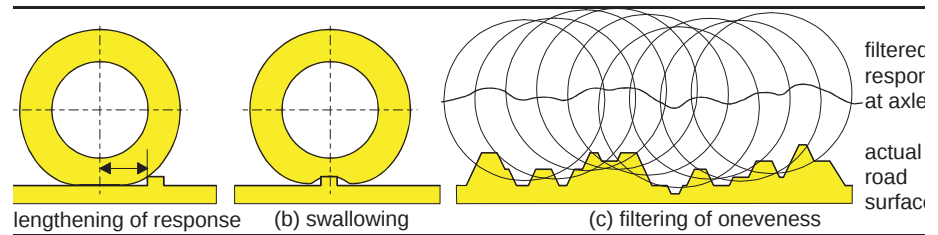
$$\rho^d = \frac{\rho_z}{\rho_{Fz0}} \quad (B-25)$$

B.3.6 Effective Road Input

The standard single point contact model is valid for vertical road input for wavelengths larger than the contact length ($>0.2m$). For short wavelength obstacles, the enveloping behavior of the tire needs to be described more accurately. The enveloping properties of the tire are described in SWIFT by so-called basic functions. This method is

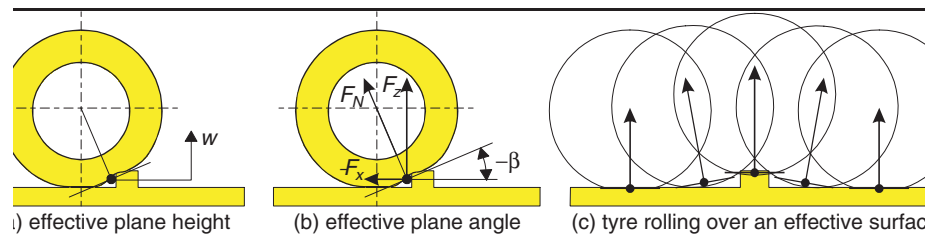
incorporated for steps in road height, and stochastic road input is treated as a sequence of steps. The phenomena that occur when a tire is rolling on an uneven road are illustrated in Figure B-8.

Figure B-8. Tire Enveloping Behavior

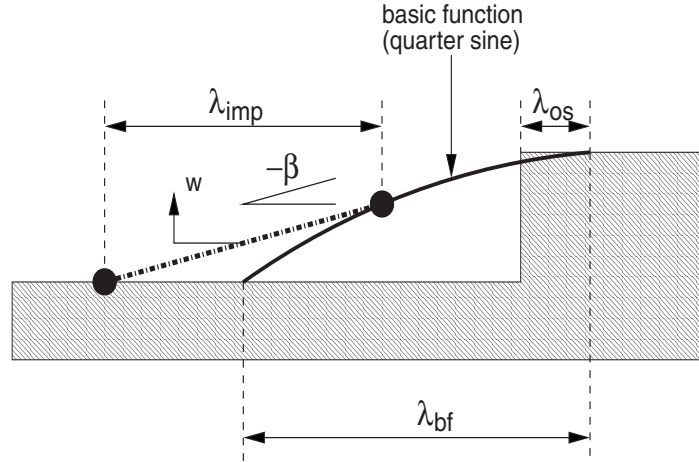


The basic function concept uses a quarter sine function to describe the response of the tire to a step obstacle. The length of the basic function corresponds to the lengthening effect, and the swallowing effect is taken care of by using a two-point contact. Additionally, the rolling radius variations that result from rolling over an obstacle is modelled. A cleat obstacle is converted into an effective height and plane angle as shown in Figure B-9.

Figure B-9. Effective Inputs



The effective inputs (plane height w and plane angle β) are calculated using the basic function (or curve) is demonstrated in Figure B-10. The basic function relates to the response of a rigid wheel, and the parameters depend on the tire radius R_0 and obstacle height h only.

Figure B-10. Example for Effective Inputs with Basic Curve

The height of the sine wave is equal to the step height and the width is approximately equal to the width of the rigid wheel response. The effective road surface is obtained by 'travelling' over the basic curve with a two-point tire-road interface. The distance between the two points (shift) is slightly smaller than the contact length of the tire. The effective plane height is obtained from the average height at the edges of the contact patch. The effective plane angle is the slope of the line through the two-point tire-road interface with respect to the horizontal.

The length (or width) of the basic function can be approximated by a Rigid Wheel response (λ_{RIGID}). The length of the basic function (λ_{bf}) is calculated in accordance with Equation (B-26):

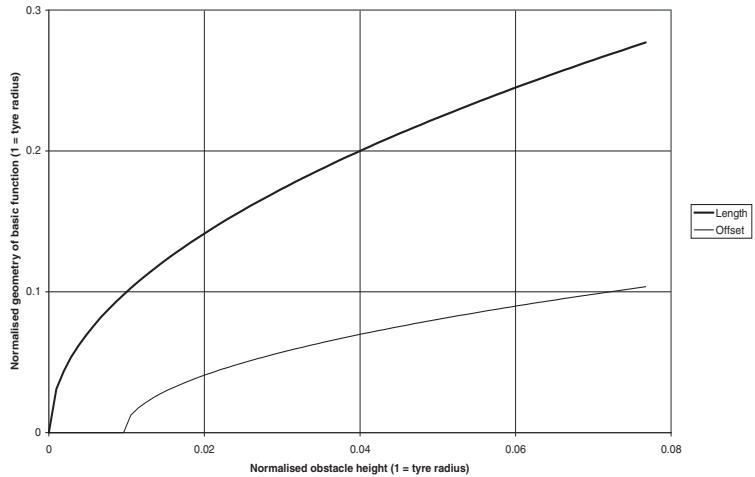
$$\lambda_{bf} = q_{\lambda bf} \lambda_{RIGID} \quad (B-26)$$

The offset λ_{os} of the basic functions occurs when a threshold value for the height is exceeded. In general, the dependency on the height resembles the Rigid Wheel response. This is incorporated in the functions of Equation (B-27):

$$\lambda_{os} = 0 \quad |h| \leq q_{\lambda os 1} R_0 \quad (B-27)$$

$$\lambda_{os} = q_{\lambda os 2} \lambda_{RIGID} (|h| - q_{\lambda os 1} R_0) \quad |h| > q_{\lambda os 1} R_0 \quad (B-28)$$

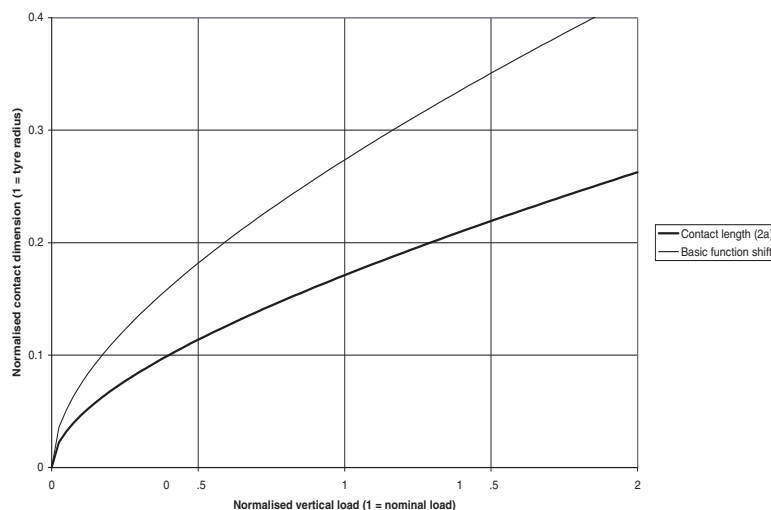
Both the length of the basic function and the offset of the basic function are displayed as function of the obstacle height in Figure B-11.

Figure B-11. Basic Function Length and Offset as Function of Obstacle Height

The next step in determining the effective input is made by taking two points and run over the basic function (see Figure B-12). The distance between the two points is indicated as the shift of the basic function. The distance between the points (λ_{imp}) depends on the contact length as is displayed in Figure 4.28 of [1], and the shift is calculated using Equation (B-29):

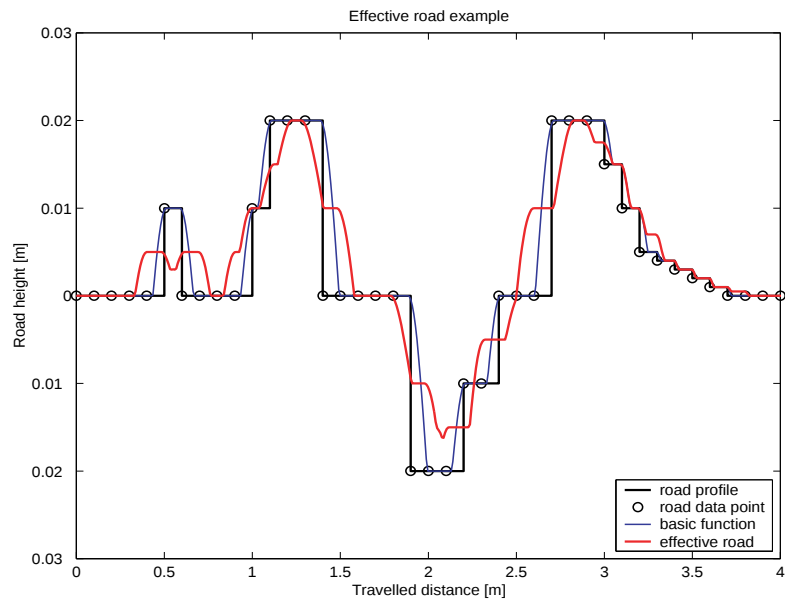
$$\lambda_{imp} = 2(q_{\lambda_{imp}1} \cdot a + q_{\lambda_{imp}2} \cdot a^2) \quad (B-29)$$

Both the contact length and the shift of the basic function are displayed in Figure B-12.

Figure B-12. Contact Length and Shift of the Basic Function as Function of Load

This method of effective inputs can not only be used for discrete obstacles, but also for measured road data having a random character, Figure B-13 gives an illustration. In this example the road height is specified every 0.1 meter. Using the stepwise changes in road height the basic functions can be calculated and using the two-point-tire-road interface model finally the effective road height is obtained.

The SWIFT-Tyre model samples the road using a fixed interval. This value is specified by *ROAD_INCREMENT* in the *[MODEL]* section of the tire property file, as seen on page B-29. Typically this value is in the range of 0.1–0.2 meter or larger; values below 0.01 meter are ignored. If the road data has a fixed sample interval, then the most accurate results will be obtained when *ROAD_INCREMENT* is set equal to the sample interval of the road data. In the example of Figure B-13 the value of *ROAD_INCREMENT* is set to 0.1 meter, the actual road data used in this example can be found in [Section B.6, "Road Property File Example"](#).

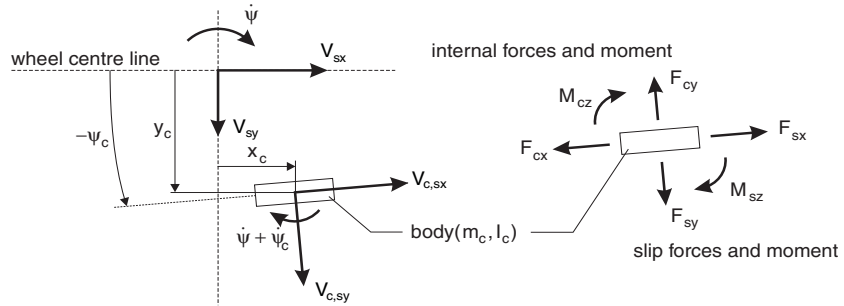
Figure B-13. Filtering of Road Data Using Basic Functions

B.3.7 Contact Model

Residual Stiffness

The contact patch is modelled as a body with mass and inertia, and has three degrees of freedom: longitudinal, lateral and yaw motion, as depicted in Figure B-14. The contact patch is connected to the rigid ring body of the tire belt with residual spring-damper systems. The slip forces are applied to the contact patch, and the transient of slip forces is modelled following the relaxation length concept, with an elaborate model for the aligning moment calculation.

B

Figure B-14. Contact Patch Model

The forces and torque that are transmitted through residual springs from the contact patch to the Rigid Ring are given by Equations (B-30) to (B-32):

$$F_{rx} = \bar{c}_{rx} \bar{\rho}_{rx} F_{z0} + 2\kappa_{rx} \dot{\bar{\rho}}_{rx} k_{t0} \quad (\text{B-30})$$

$$F_{ry} = \bar{c}_{ry} \bar{\rho}_{ry} F_{z0} + 2\kappa_{ry} \dot{\bar{\rho}}_{ry} k_{t0} \quad (\text{B-31})$$

$$M_{rz} = \bar{c}_{r\psi} \rho_{r\psi} C_{r0} + 2\kappa_{r\psi} \dot{\rho}_{r\psi} k_{r0} \quad (\text{B-32})$$

Transient Slip Behavior

At change of slip, it takes a certain distance to build up the forces in the contact area. This transient behavior is incorporated in the model, and is referred to as relaxation length. The contact length is the main determining factor for the transient properties in the contact patch and it is a function of vertical load in accordance with Equation (B-33):

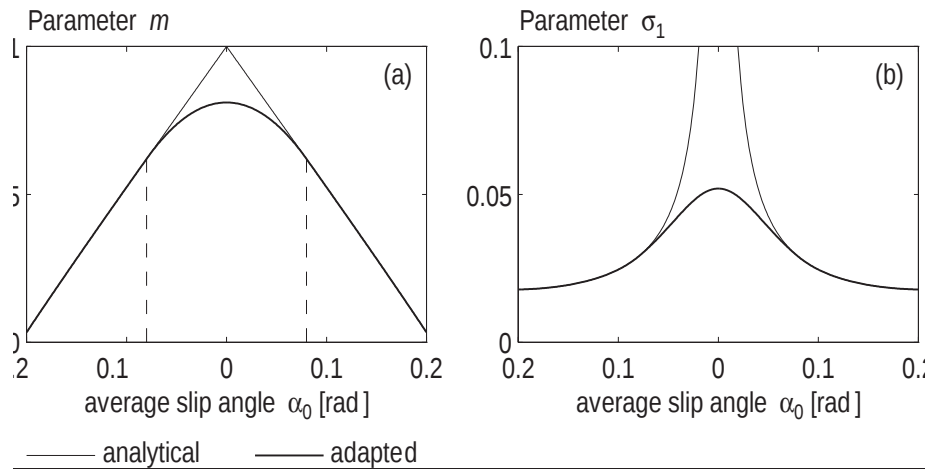
$$a = \left(q_{a2} \sqrt{\bar{F}_z^2} + q_{a1} \sqrt{\bar{F}_z} \right) R_0 \quad (\text{B-33})$$

The resulting function for the contact length is displayed in Figure B-12 on page B-18. The relaxation length in the contact area (σ_c) is a function of the adhesion level (m) in accordance with Equation (B-34), which is similar to Equation 3.28 in reference [2]:

$$\sigma_c = m \cdot a \quad (\text{B-34})$$

The value of m is also used in the so-called Phase leading network that is applied in the aligning moment calculation (see reference [2]). In order to prevent numeric instability around zero slip (when m approaches 1), the value is modified within a small band of slip as displayed in Figure B-15.

Figure B-15. Modification of Adhesion Coefficient in Calculation



The band of modification is determined by Equation (B-35), where α_{sl} is the slip value where full sliding is assumed:

$$\alpha_{min} = q_{amin} \alpha_{sl} \quad (B-35)$$

In addition to the common relaxation length system, the longitudinal relaxation length system is extended to Equation (B-36) to increase damping at lower speeds. The damping parameters are applied in the model through Equation (B-37). In both equations, the forward speed (V_x) is used instead of the rolling speed (V_{cr}) to increase the robustness of the model:

$$(\sigma_c + \frac{k_{cx}}{c_{cx}} |V_x|) \zeta_{cx} + |V_x| \zeta_{cx} = -V_{c, sx} \frac{k_{cx}}{c_{cx}} \dot{V}_{c, sx} \quad (B-36)$$

$$\frac{k_{cx}}{c_{cx}} = \frac{q_{kc1}}{1 + q_{kc2} |V_x|} \quad (B-37)$$

B.3.8 Switching from Simple to Complex Tire Model

The SWIFT software incorporates a switch for tire complexity selection: the parameter *USE_MODE* in the [MODEL] section of the tire property file.

Instead of full Rigid Ring tire dynamics, the switch can be set for transient behavior only, or steady state behavior. Under these conditions the SWIFT model behaves just as MF-Tyre. The optional settings are given in Table B-7.

Table B-7: Various Options for the Value of USE_MODE

	Steady state:	Transient:	SWIFT:
Vertical only	0	0	20
Longitudinal only	1	11	21
Lateral only	2	12	22
Long. and lat. uncombined	3	13	23
Combined slip	4	14	24

In case of transient behavior, the belt stiffness is taken into account as well as the contact length for the calculation of the tire relaxation length. The longitudinal relaxation length is calculated from the SWIFT stiffness using Equation (B-38), which incorporates a scaling factor:

$$\sigma_{\kappa} = \left(\alpha + K_x \left(\frac{1}{c_{bx}} + \frac{1}{c_{rx}} + \frac{R_e^2}{C_{b\theta}} \right) \right) \lambda_{\sigma\kappa} \quad (\text{B-38})$$

Similarly, the lateral relaxation length is calculated from the SWIFT stiffness using Equation (B-39) (note that K_y is a negative quantity), including a scaling factor:

$$\sigma_{\alpha} = \left(\alpha - K_y \left(\frac{1}{c_{by}} + \frac{1}{c_{ry}} + \frac{R_e^2}{C_{b\gamma}} \right) \right) \lambda_{\sigma\alpha} \quad (\text{B-39})$$

B.4 Tire Model Parameters

The tire parameters that are defined in the tire property file are related to the model structure in Figure B-16. For each of the parameters, a reference to the equations in this manual is given. The full context of the parameters can be understood by looking up the appropriate equations in this guide. The SWIFT-specific parameters are listed below. MF-Tyre indicates that the parameter or group of parameters is also used for

MF-Tyre. For more information on those parameters see [Appendix A, "Using the MF-Tyre Model"](#).

Figure B-16. Tire Parameters in Model Structure

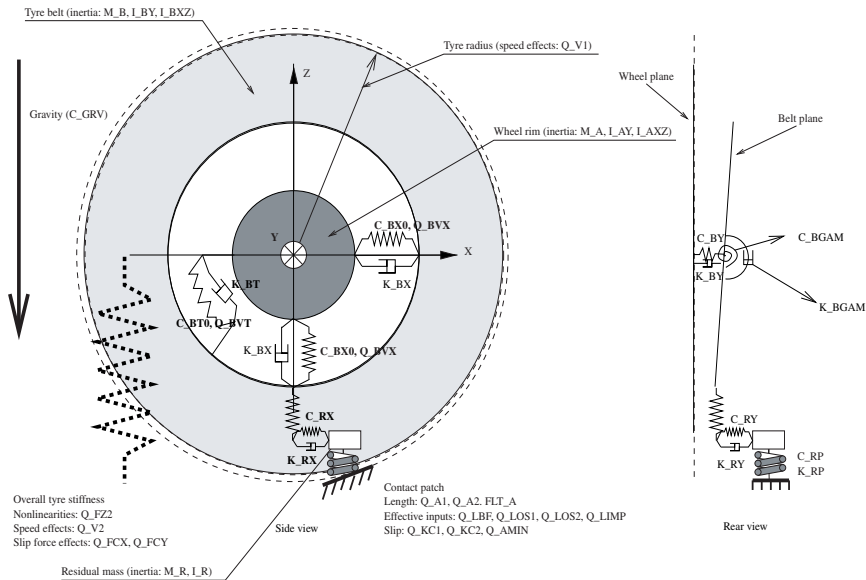


Table B-8: Definition of Parameters in Tire Property File

Tire parameter:	User manual notation:	Notes Tire characteristic (if applicable):
[MODEL]		
LONGVL	V_0	See Equation (B-1), derived from Test Trailer measurement conditions
ROAD_INCREMENT	n.a	Sample interval road data
ROAD_DIRECTION	n.a	>0 driving in positive x-direction <0 driving in negative x-direction
[DIMENSION]		
UNLOADED_RADIUS	R_0	

Table B-8: Definition of Parameters in Tire Property File (Continued)

Tire parameter:	User manual notation:	Notes Tire characteristic (if applicable):
[SHAPE]		MF-Tyre
[INERTIA]		
MASS	m_0	Tire mass
I_AY	n.a	Value to be added to wheel rim (multiply with $I_0=m_0R_0^2$)
I_AXZ	n.a	Value to be added to wheel rim (multiply with $I_0=m_0R_0^2$)
I_BY	n.a	Tire belt moment of inertia about Y-axis / Belt wind-up frequency – gyroscopic effects
I_BXZ	n.a	Tire belt moment of inertia about X and Z-axis / Belt camber and yaw frequency – gyroscopic effects
I_R	n.a.	Residual mass moment of inertia about Z-axis, tuned value for optimal performance
M_A	n.a.	Value to be added to wheel rim (multiply with m_0)
M_B	m_b	See Table B-3 on page B-5 / Translation belt frequencies
M_R	n.a.	Residual mass, tuned value for optimal performance
C_GRV	C_{grv}	See Equation (B-8)
[CONTACT_PATCH]		
Q_A2	q_{a2}	See Equation (B-33) / Relaxation length & enveloping behavior
Q_A1	q_{a1}	See Equation (B-33) / Relaxation length & enveloping behavior
Q_LBF	q_{lbf}	See Equation (B-26) / Enveloping behavior
Q_LOS1	q_{los1}	See Equation (B-27) / Enveloping behavior

Table B-8: Definition of Parameters in Tire Property File (Continued)

Tire parameter:	User manual notation:	Notes Tire characteristic (if applicable):
Q_LOS2	q_{los2}	See Equation (B-27) Enveloping behavior
Q_LIMP1	q_{limp1}	See Equation(B-29) Enveloping behavior
Q_LIMP2	q_{limp2}	See Equation(B-30) Enveloping behavior
Q_KC1	q_{kc1}	See Equation (B-37), tuned value for optimal performance / Tire damping for low speed
Q_KC2	q_{kc2}	See Equation (B-37), tuned value for optimal performance /Tire damping for low speed
Q_AMIN	q_{amin}	See Equation (B-35), tuned value for optimal performance / Aligning moment for short wavelength around slip=0
FLT_A	n.a.	Contact length filter, tuned value for optimal performance / Contact length for load variations
[VERTICAL]		
VERTICAL_STIFFNESS	C_z	Flat Planksee Equation (B-22) / Vertical stiffness + vertical belt frequency
VERTICAL_DAMPING		MF-Tyre
BREFF		MF-Tyre
DREFF		MF-Tyre
FREFF		MF-Tyre
FNOMIN	F_{z0}	see Equation (B-2), defined value for measurement programme /Tire load rating
Q_RE0	q_{re0}	see Equation (B-19) / Free tire and effective tire radius
Q_V1	q_{v1}	see Equation (B-18) /Tire growth due to speed
Q_V2	q_{v2}	see Equation (B-21) / Speed effect on vertical stiffness

Table B-8: Definition of Parameters in Tire Property File (Continued)

Tire parameter:	User manual notation:	Notes Tire characteristic (if applicable):
Q_FZ2	q_{Fz2}	See Equation (B-21) / Progressiveness of vertical load for deflection
Q_FCX	q_{FCx}	See Equation (B-21) / Decrease in vertical stiffness due to brake slip force
Q_FCY	q_{FCy}	See Equation (B-21) / Decrease in vertical stiffness due to side slip force
[LONG_SLIP_RANGE]		MF-Tyre
[SLIP_ANGLE_RANGE]		MF-Tyre
[INCLINATION_ANGLE_RANGE]		MF-Tyre
[VERTICAL_FORCE_RANGE]		MF-Tyre
[SCALING_COEFFICIENTS]		MF-Tyre
[LONGITUDINAL_COEFFICIENTS]		MF-Tyre
[OVERTURNING_COEFFICIENTS]		MF-Tyre
[LATERAL_COEFFICIENTS]		MF-Tyre
[ROLLING_COEFFICIENTS]		MF-Tyre
[ALIGNING_COEFFICIENTS]		MF-Tyre
[STRUCTURAL]		
C_BX0	c_{bx0}	See Equation (B-10) / In-plane translation belt frequency – longitudinal relaxation length
C_RX	c_{rx}	See Equation (B-30), tuned value for optimal performance /Longitudinal relaxation length
C_BT0	c_{bq0}	see Equation (B-11) / Belt wind-up rotation frequency – longitudinal relaxation length
C_BY	c_{by}	See Equation (B-15) /Out-of-plane translation belt frequency – lateral relaxation length
C_RY	c_{ry}	See Equation (B-31) /Out-of-plane translation belt frequency – lateral relaxation length

Table B-8: Definition of Parameters in Tire Property File (Continued)

Tire parameter:	User manual notation:	Notes Tire characteristic (if applicable):
C_BGAM	c_{bg}	See Equation (B-16) /Belt Camber / yaw rotation frequency – lateral relaxation length
C_RP	c_{ry}	See Equation (B-32), tuned value for optimal performance / Aligning moment at short wavelength
K_BX	k_{bx}	See Equation (B-12) /damping for in-plane translation belt frequency
K_RX	k_{rx}	See Equation (B-30), tuned value for optimal performance
K_BT	k_{bq}	See Equation (B-14) /damping for wind-up belt frequency
K_BY	k_{by}	See Equation (B-15) /damping for lateral belt frequency
K_RY	k_{ry}	See Equation (B-31)
K_BGAM	k_{bg}	See Equation (B-16) /Belt Camber / yaw rotation frequency – lateral relaxation length
K_RP	k_{rp}	See Equation (B-32), tuned value for optimal performance
Q_BVX	$q_{bv x}$	See Equation (B-10) /Change of in-plane belt translation frequency with speed
Q_BVT	$q_{bv q}$	See Equation (B-11) /Change of wind-up belt rotation frequency with speed

B.5 Tire Property File Example

```

FILE_TYPE                = 'tir'
FILE_VERSION              = 3.0
FILE_FORMAT               = 'ASCII'
! : TIRE_VERSION :       SWIFT-Tyre 1.0
! : COMMENT :           New File Format v3.0
! : COMMENT :           Tire                205/60 R15
! : COMMENT :           Manufacturer        DELFT-TYRE
! : COMMENT :           Nom. section with  (m) 0.205
! : COMMENT :           Nom. aspect ratio  (-) 60
! : COMMENT :           Infl. pressure    (Pa) 220000
! : COMMENT :           Rim radius        (m) 0.19
! : COMMENT :           Measurement ID     DELFT-TYRE
! : COMMENT :           Test speed        (m/s) 16.667
! : COMMENT :           Road surface      Asphalt
! : COMMENT :           Road condition    Dry
! : FILE_FORMAT :       ASCII
! : USER :             MF-Tool
! : Generated by :      TNO
! : Copyright TNO, Tue Aug 07 16:33:34 2001
!
! USE_MODE specifies the type of calculation performed:
!   0: Fz only, no Magic Formula evaluation
!   1: Fx,My only
!   2: Fy,Mx,Mz only
!   3: Fx,Fy,Mx,My,Mz uncombined force/moment calculation
!   4: Fx,Fy,Mx,My,Mz combined force/moment calculation
! +10: including relaxation behaviour
! +20: including rigid ring dynamics
! *-1: mirroring of tyre characteristics
!
! example: USE_MODE = -12 implies:
!   -calculation of Fy,Mx,Mz only
!   -including relaxation effects
!   -mirrored tyre characteristics
!
$-----units
[UNITS]
LENGTH                = 'meter'
FORCE                  = 'newton'
ANGLE                  = 'radians'
MASS                    = 'kg'
TIME                   = 'second'

```

```

$-----model
[MODEL]
PROPERTY_FILE_FORMAT      = 'SWIFT-TYRE'
TYPE                      = 'CAR'
FIT'TYP                   = 21
USE_MODE                   = 24
MFSAFE1                    = -5280
MFSAFE2                    = 0
MFSAFE3                    = 150
LONGVL                     = 16.667
VXLOW                      = 1
ROAD_INCREMENT             = 0.1
ROAD_DIRECTION             = 1
$-----dimensions
[DIMENSION]
UNLOADED_RADIUS            = 0.3135
WIDTH                      = 0.205
ASPECT_RATIO               = 0.6
RIM_RADIUS                 = 0.19
RIM_WIDTH                  = 0
$-----shape
[SHAPE]
{radial width}
1.0    0.0
1.0    0.4
1.0    0.9
0.9    1.0
$-----
--inertia
[INERTIA]
MASS                        = 9.3
I_AY                       = 0.109406207
I_AXZ                      = 0.0711140344
I_BY                       = 0.695823475
I_BXZ                      = 0.356664234
I_R                        = 0.0547031034
M_A                        = 0.23655914
M_B                        = 0.76344086
M_R                        = 0.107526882
C_GRV                      = -9.81
$-----contact_patch
[CONTACT_PATCH]
Q_A2                       = 0.0353429027
Q_A1                       = 0.135228475
Q_LBF                      = 1
Q_LOS1                     = 0.01

```

```

Q_LOS2                = 0.4
Q_LIMP1                = 0.8
Q_LIMP2                = 0.0
Q_KC1                  = 0.106328549
Q_KC2                  = 6.6668
Q_AMIN                 = 0.3
FLT_A                  = 2000
$-----vertical
[VERTICAL]
VERTICAL_STIFFNESS     = 196261
VERTICAL_DAMPING       = 50
BREFF                  = 9
DREFF                  = 0.23
FREFF                  = 0.01
FNOMIN                 = 4000
Q_RE0                  = 0.997448166
Q_V1                   = 7.15073791e-005
Q_V2                   = 2.4892
Q_FZ2                  = 14.3468
Q_FCX                  = 0
Q_FCY                  = 0
BOTTOM_OFFST           = 0.01
BOTTOM_TRNSF           = 1000
BOTTOM_STIFF           = 2E+6
$-----long_slip_range
[LONG_SLIP_RANGE]
KPUMIN                 = -1.5
KPUMAX                 = 1.5
$-----slip_angle_range
[SLIP_ANGLE_RANGE]
ALPMIN                 = -1.5708
ALPMAX                 = 1.5708
$-----inclination_slip_range
[INCLINATION_ANGLE_RANGE]
CAMMIN                 = -0.2619
CAMMAX                 = 0.2619
$-----vertical_force_range
[VERTICAL_FORCE_RANGE]
FZMIN                  = 0
FZMAX                  = 9000
$-----scaling
[SCALING_COEFFICIENTS]
LFZO                   = 1
LCX                    = 1
LMUX                   = 1
LEX                    = 1

```

```
LKX = 1
LHX = 0
LVX = 0
LGAX = 1
LCY = 1
LMUY = 1
LEY = 1
LKY = 1
LHY = 0
LVY = 0
LGAY = 1
LTR = 1
LRES = 0
LGAZ = 1
LXAL = 1
LYKA = 1
LVYKA = 1
LS = 1
LSGKP = 1
LSGAL = 1
LGYR = 1
LMX = 1
LVMX = 1
LMY = 1
$-----longitudinal
[LONGITUDINAL_COEFFICIENTS]
PCX1 = 1.6846
PDX1 = 1.2096
PDX2 = -0.03705
PDX3 = 0
PEX1 = 0.34446
PEX2 = 0.095439
PEX3 = -0.020488
PEX4 = 0
PKX1 = 21.512
PKX2 = -0.16314
PKX3 = 0.24502
PHX1 = -0.0016331
PHX2 = 0.001517
PVX1 = 0
PVX2 = 0
RBX1 = 12.35
RBX2 = -10.767
RCX1 = 1.0918
REX1 = 0
REX2 = 0
```



```

RHX1                = 0.0066313
PTX1                = 1
PTX2                = 0
PTX3                = 0
$-----overturning
[OVERTURNING_COEFFICIENTS]
QSX1                = 0
QSX2                = 0
QSX3                = 0
$-----lateral
[LATERAL_COEFFICIENTS]
PCY1                = 1.1931
PDY1                = -0.99006
PDY2                = 0.14522
PDY3                = -11.231
PEY1                = -1.0026
PEY2                = -0.53683
PEY3                = -0.083107
PEY4                = -4.7866
PKY1                = -14.946
PKY2                = 2.1297
PKY3                = -0.028283
PHY1                = 0.0033518
PHY2                = -0.00053863
PHY3                = 0.07452
PVY1                = 0.044552
PVY2                = -0.023557
PVY3                = -0.53156
PVY4                = 0.03923
RBY1                = 6.461
RBY2                = 4.1957
RBY3                = -0.015164
RCY1                = 1.0812
REY1                = 0
REY2                = 0
RHY1                = 0.0086257
RHY2                = 0
RVY1                = 0.053266
RVY2                = -0.073458
RVY3                = 0.51728
RVY4                = 35.444
RVY5                = 1.9
RVY6                = -10.715
PTY1                = 1
PTY2                = 1
$-----rolling resistance

```

```

[ROLLING_COEFFICIENTS]
QSY1           = 0.01
QSY2           = 0
QSY3           = 0
QSY4           = 0
$-----aligning
[ALIGNING_COEFFICIENTS]
QBZ1           = 8.9644
QBZ2           = -1.1064
QBZ3           = -0.8422
QBZ4           = 0
QBZ5           = -0.22733
QBZ9           = 18.465
QBZ10          = 0
QCZ1           = 1.1805
QDZ1           = 0.099556
QDZ2           = -0.00074773
QDZ3           = 0.0065197
QDZ4           = 13.053
QDZ6           = -0.0079448
QDZ7           = 0.00019609
QDZ8           = -0.29569
QDZ9           = -0.0089855
QEZ1           = -1.6085
QEZ2           = -0.3592
QEZ3           = 0
QEZ4           = 0.17433
QEZ5           = -0.8957
QHZ1           = 0.0067668
QHZ2           = -0.0018847
QHZ3           = 0.14697
QHZ4           = 0.0042775
SSZ1           = 0.043285
SSZ2           = 0.0013747
SSZ3           = 0.73146
SSZ4           = -0.23758
QTZ1           = 0.05
MBELT          = 9.3
$-----structural
[STRUCTURAL]
C_BX0          = 121.3872
C_RX           = 391.875
C_BT0          = 61.9617225
C_BY           = 40.049625
C_RY           = 62.7
C_BGAM         = 20.3349282

```

C_RP	= 55.8213716
K_BX	= 0.113761382
K_RX	= 0.45504553
K_BT	= 0.0398641872
K_BY	= 0.141974205
K_RY	= 0.45504553
K_BGAM	= 0.0185199476
K_RP	= 0.416698821
Q_BVX	= 3.9567458
Q_BVT	= 3.9567458

B.6 Road Property File Example

In the road property file the road height is specified as a function of traveled distance. In a road data file the left and right track data may be specified; the appropriate track data is selected depending on the role of the tire in the model.

SWIFT uses a zero-order sample and hold when evaluating the road profile, as shown in Figure B-13. Changes in the height of the road profile are interpreted as steps. For maximum accuracy it is important that the sample points coincide with the data provided by the user, otherwise interpolated data will be used. So you should use road data with a fixed sample interval and specify this value for *ROAD_INCREMENT* in the *[MODEL]* section of the tire property file. Typically, the road sample interval should be in the range of 0.1-0.2 meters or larger. For the road data given below, the value of *ROAD_INCREMENT* should be set to 0.1 meter.

```
FILE_TYPE = 'rdf'
FILE_VERSION = 5.00
FILE_FORMAT = 'ASCII'
(COMMENTS)
{comment_string}
'polyline style road description'
$-----UNITS
[UNITS]
MASS = 'kg'
LENGTH = 'meter'
TIME = 'sec'
ANGLE = 'degree'
FORCE = 'newton'
```

```
$-----MODEL
[MODEL]
METHOD           = '2D'
ROAD_TYPE        = 'poly_line'
$-----PARAMETERS
[PARAMETERS]
OFFSET           = 0
ROTATION_ANGLE_XY_PLANE = 0
MU              = 1
$
(XZ_DATA)
-10000          0      0
      0          0      0
      0.1        0      0
      0.2        0      0
      0.3        0      0
      0.4        0      0
      0.5        0.01  0.01
      0.6        0      0
      0.7        0      0
      0.8        0      0
      0.9        0      0
      1          0.01  0.01
      1.1        0.02  0.02
      1.2        0.02  0.02
      1.3        0.02  0.02
      1.4        0      0
      1.5        0      0
      1.6        0      0
      1.7        0      0
      1.8        0      0
      1.9        -0.02 -0.02
      2          -0.02 -0.02
      2.1        -0.02 -0.02
      2.2        -0.01 -0.01
      2.3        -0.01 -0.01
      2.4        0      0
      2.5        0      0
      2.6        0      0
      2.7        0.02  0.02
      2.8        0.02  0.02
      2.9        0.02  0.02
      3          0.015 0.015
      3.1        0.01  0.01
      3.2        0.005 0.005
      3.3        0.004 0.004
```



3.4	0.003	0.003
3.5	0.002	0.002
3.6	0.001	0.001
3.7	0	0
3.8	0	0
3.9	0	0
4.0	0	0
10000	0	0

Appendix C FE Link Interface

Finite Element (FE) models are generated in external CAE systems and stored in separate files that Fedem refers to as *FE model files* or *link files*. Fedem stores link files in the Fedem Technology Link format (*.ftl*), and has import filters for the Nastran Bulk Data Format (*.nas* or *.bdf*) and the older Fedem Link Model format (*.flm*). This appendix describes these files and their formats.

Sections in this appendix address the following topics:

- [Fedem Technology Link format](#)
- [Nastran Bulk Data File format](#)

C.1 Fedem Technology Link format

In Release 2.5, Fedem Technology introduced the new *.ftl* file format for the definition of FE links. This format is designed to be flexible, powerful, and extensible for adding new entries in the model. The file is defined in ASCII format and can be easily edited using a text editor.

C.1.1 Syntax

An *.ftl* file contains a set of identifiers (nodes, elements, control data, and attributes) and parameters that are expressed with the same overall syntax:

```
identifier{id value1 ... valueN {reference id text}}
```

Name	Description
<i>identifier*</i>	Specifies field type (e.g., element type, attribute type).
<i>id*</i>	Unique ID for the field (relative to the other fields with the same identifier).
<i>value1 ... valueN</i>	Primary values for the object (can be text, integers, or decimal digits).
<i>references</i>	Additional data or other fields can be referred to using this field.
<i>reference and id</i>	Field reference (reference specified in combination with a valid ID).
<i>text</i>	Can be used as additional information for a field reference or as an optional tag (e.g., a group name)

* Denotes required text.

The following are examples:

```
TET4{4 22 34 12 32 {PMAT 1}}
```

A constant strain tetrahedron element with ID 4, referring to nodes 22, 34, 12 and 32. The element uses an attribute of type PMAT (material property) with ID 1.

```
PMAT{1 2.10e+11 8.00e+10 2.90e-01 7.82e+03}
```

A material property entry with ID 1 and four decimal numbers describing the different parameters in the material.



NOTE: All text between a comment symbol ("#") and the end of the line is ignored.

C.1.2 Nodes

Nodes are expressed by the following syntax:

NODE{id state x y z}

Parameter	Value Type	Description
<i>id</i>	Integer	External node identifier
<i>state</i>	Integer	Internal/external state flag = 0 (an internal node condensed out in the reduction) = 1 (an external node retained in the reduction)
<i>x y z</i>	Real	Global nodal coordinates

C.1.3 Structural elements

Elements are expressed in several ways depending on the element type. Each element type is expressed by one of the statements in the table below.

Element statements
<i>BEAM2{id n1 n2 {PMAT pid}{PBEAMSECTION gid} [{PORIENT oid}][{PBEAMECCENT eid}][{PBEAMPIN bpid}][{PEFFLENGTH lid}][{PNSM nid}][{VDETAIL vid}]}</i>
<i>FFT3{id n1 n2 n3 {PMAT pid}{PTHICK gid} [{PNSM nid}][{VDETAIL vid}]}</i>
<i>FFQ4{id n1 n2 n3 n4 {PMAT pid}{PTHICK gid} [{PNSM nid}][{VDETAIL vid}]}</i>
<i>TET4{id n1 n2 n3 n4 {PMAT pid} [{VDETAIL vid}]}</i>
<i>TET10{id n1 n2 ... n10 {PMAT pid} [{VDETAIL vid}]}</i>
<i>WEDG6{id n1 n2 ... n6 {PMAT pid} [{VDETAIL vid}]}</i>
<i>WEDG15{id n1 n2 ... n15 {PMAT pid} [{VDETAIL vid}]}</i>
<i>HEX8{id n1 n2 ... n8 {PMAT pid} [{VDETAIL vid}]}</i>
<i>HEX20{id n1 n2 ... n20 {PMAT pid} [{VDETAIL vid}]}</i>



Element statements

```
BUSH{id n1 n2 [{PBUSHCOEFF bcid}][{PBUSHECCENT beid}]
[{PORIENT oid}][{PCOORDSYS csid}][{VDETAIL vid}]}
```

```
SPRING{id n1 n2 {PSPRING sid} [{VDETAIL vid}]}
```

```
RSPRING{id n1 n2 {PSPRING sid} [{VDETAIL vid}]}
```

```
CMASS{id n1 [{PMASS mid}][{VDETAIL vid}]}
```

```
RBAR{id n1 n2 {PRBAR rid} [{VDETAIL vid}]}
```

```
RGD{id n1 n2 ... nn [{PRGD rid}][{VDETAIL vid}]}
```

```
WAVGM{id n1 n2 ... nn {PWAvgm wid} [{VDETAIL vid}]}
```



NOTE: The identifiers correspond to the element types defined in the Fedem R5.0 Theory Guide, Appendix A, "Finite Element library".



NOTE: The square brackets ([]) denote optional parameters. The vertical bar (|) means that either one of the two parameters on each side of it may be specified, but not both.



NOTE: The elements BUSH and CMASS may exist without any associated properties (PBUSHCOEFF and PMASS, respectively) in the .ftl file. Such elements are created automatically by Fedem during modelling, e.g. when a mechanism joint is attached to a slave FE node in a Link (see [Section 3.6, "Attaching and detaching elements"](#)). When the Fedem Link Reducer encounters such property-less elements, some stiffness/mass properties are computed automatically for these elements based on the assembled stiffness/mass matrix of the whole Link, such that the element can be regarded as nearly mass-less and rigid. Refer to the Fedem R5.0 Theory Guide, Appendix A, "Finite Element library" for details on this computation.

Parameters for element statements are given in the table below.

Parameter	Description
<i>id</i>	External element identifier
<i>ni</i>	Reference to the <i>i</i> th node connected to this element
<i>bcid</i>	Reference to a stiffness coefficient field for bushing elements
<i>beid</i>	Reference to an eccentricity field for bushing elements
<i>bpid</i>	Reference to a pin flag field for beam elements
<i>csid</i>	Reference to a local coordinate system field
<i>eid</i>	Reference to an eccentricity field for beam elements

Parameter	Description
<i>gid</i>	Reference to a geometric property field for beam and shell elements
<i>lid</i>	Reference to an effective length field for beam elements
<i>mid</i>	Reference to a mass property field for concentrated mass elements
<i>nid</i>	Reference to a non-structural mass field for beam and shell elements
<i>oid</i>	Reference to an orientation field for this element
<i>pid</i>	Reference to a material field for this element
<i>rid</i>	Reference to a component numbers field for rigid elements
<i>sid</i>	Reference to a stiffness matrix field for spring elements. For SPRING elements it refers to a PSPRING entry with <code>type=1</code> , whereas for RSPRING elements it refers to a PSPRING entry with <code>type=2</code> .
<i>vid</i>	Reference to a visibility status field for this element
<i>wid</i>	Reference to a weight- and component numbers field for weighted averaged motion elements

C.1.4 Properties

The various properties that are used in the structural element expressions have the following syntax:

PMAT{*pid e g v ρ*}

Parameter	Value Type	Description
<i>pid</i>	Integer	Material property identifier
<i>e</i>	Real	Young's modulus
<i>g</i>	Real	Shear modulus (used by beam elements only)
<i>v</i>	Real	Poisson's ratio
<i>ρ</i>	Real	Material density



PBEAMSECTION{gid a iyy izz ix x ky kz cy cz}

Parameter	Value Type	Description
<i>gid</i>	Integer	Geometric property identifier
<i>a</i>	Real	Cross-sectional area
<i>iyy izz</i>	Real	Moments of inertia about the local y- and z-axes of the beam
<i>ix x</i>	Real	Torsional stiffness parameter
<i>ky kz</i>	Real	Transverse shear reduction factors
<i>cx cz</i>	Real	Local y- and z-coordinates of the shear center of the beam

PBEAMECCENT{eid ex1 ey1 ez1 ex2 ey2 ez2}

Parameter	Value Type	Description
<i>eid</i>	Integer	Eccentricity vectors identifier
<i>ex1 ey1 ez1</i>	Real	Eccentricity vector at local node 1 of the beam
<i>ex2 ey2 ez2</i>	Real	Eccentricity vector at local node 2 of the beam

PEFFLENGTH{lid leff}

Parameter	Value Type	Description
<i>lid</i>	Integer	Effective length identifier
<i>leff</i>	Real	Effective beam length

PBEAMPIN{bpid pa pb}

Parameter	Value Type	Description
<i>bpid</i>	Integer	Beam pin flag identifier
<i>pa</i>	Integer	Local DOFs in end 1 that are released
<i>pb</i>	Integer	Local DOFs in end 2 that are released

PNSM{nid rho flag}

Parameter	Value Type	Description
<i>nid</i>	Integer	Non-structural mass identifier
<i>rho</i>	Real	The non-structural mass per unit length if <i>flag</i> =0, and per unit area if <i>flag</i> =1
<i>flag</i>	Integer	Flag indicating if this entry is used by a beam or shell

PTHICK{gid t}

Parameter	Value Type	Description
<i>gid</i>	Integer	Geometric property identifier
<i>t</i>	Real	Shell thickness

PBUSHCOEFF{bcid k1 k2 k3 k4 k5 k6}

Parameter	Value Type	Description
<i>bcid</i>	Integer	Bushing coefficients identifier
<i>ki</i>	Real	Stiffness coefficients in local directions of the bushing element



PBUSHECCENT{beid ex ey ez}

Parameter	Value Type	Description
<i>beid</i>	Integer	Eccentricity vector identifier
<i>ex ey ez</i>	Real	Offset vector from local node 1 to the bushing element location

PORIENT{oid ox oy oz}

Parameter	Value Type	Description
<i>oid</i>	Integer	Orientation vector identifier
<i>ox oy oz</i>	Real	Local z-axis of the element

PCOORDSYS{csid ox oy oz zx zy zz px py pz}

Parameter	Value Type	Description
<i>csid</i>	Integer	Local coordinate system identifier
<i>ox oy oz</i>	Real	Origin of the local coordinate system
<i>zx zy zz</i>	Real	Local Z-axis in the coordinate system
<i>px py pz</i>	Real	Point in the local XZ-plane

PSPRING{sid k11 k21 k22 k31 k32 k33 k41 k42 k43 k44 k51 k52 k53 k54 k55 k61 k62 k63 k64 k65 k66 type}

Parameter	Value Type	Description
<i>sid</i>	Integer	Spring stiffness matrix identifier
<i>kij</i>	Real	Component (<i>i,j</i>) of the symmetric 6x6 spring stiffness matrix
<i>type</i>	Integer	Spring type flag =1 : Translatory spring =2 : Rotational spring

PMASS{mid I11 I21 I22 I31 I32 I33 I41 I42 I43 I44 I51 I52 I53 I54 I55 I61 I62 I63 I64 I65 I66}

Parameter	Value Type	Description
<i>mid</i>	Integer	Mass property identifier
<i>Iij</i>	Real	Component (<i>i,j</i>) of the symmetric point-mass matrix

PRBAR{rid cn1 cn2 cm1 cm2}

Parameter	Value Type	Description
<i>rid</i>	Integer	Rigid bar property identifier
<i>cn1 cn2</i>	Integer	Component numbers of independent DOFs in the link coordinate system for the element at end 1 and 2 respectively
<i>cm1 cm2</i>	Integer	Component numbers of dependent DOFs in the link coordinate system assigned by the element at end 1 and 2 respectively

PRGD{rid cm}

Parameter	Value Type	Description
<i>rid</i>	Integer	Rigid body property identifier
<i>cm</i>	Integer	Component numbers of the dependent DOFs in the link coordinate system at all slave nodes of the rigid element



```
PWAVGM{wid rc x1 ... x6 w11 ... w1n w21 ... w2n ...}
```

Parameter	Value Type	Description
<i>wid</i>	Integer	Weighted averaged motion property identifier
<i>rc</i>	Integer	Component numbers of the dependent DOFs at the slave node of the weighted average motion element.
<i>xi</i>	Integer	Row index into the weighting matrix for local DOF <i>i</i> at the slave node
<i>wij</i>	Real	Weighting factor at master node <i>j</i> for the element, for all slave DOFs whose row index equals <i>i</i>

```
VDETAIL{vid visible}
```

Parameter	Value Type	Description
<i>vid</i>	Integer	Visibility status identifier
<i>visible</i>	Bool	Visibility flag (<i>visible</i> = 0 means invisible element faces, <i>visible</i> = 1 means visible element faces)

C.1.5 Loads

Both concentrated point loads on nodes and distributed surface loads on shell and solid elements are supported. The FTL-syntax is as follows:

```
CFORCE{sid fx fy fz n1 ... n<n>}
```

```
CMOMENT{sid mx my mz n1 ... n<n>}
```

```
SURFLOAD{sid p1 ... p<n> e1 ... e<n> [{PORIENT oid}]}
```

```
FACELOAD{sid p1 ... p<n> e1 f1 ... e<n> f<n>
[{PORIENT oid}]}
```

Parameter	Value Type	Description
<i>sid</i>	Integer	Load set identifier
<i>fx fy fz</i> <i>mx my mz</i>	Real	Global force and torque components

Parameter	Value Type	Description
<i>p<i></i>	Real	Surface force intensity in local node <i>i</i>
<i>n<i></i>	Integer	Node IDs of load target points
<i>e<i></i>	Integer	Element ID
<i>f<i></i>	Integer	Local face number

C.1.6 Strain Coat Elements

Two strain coat element types are supported, where the type names reflect the number of element nodes. The FTL-syntax is as follows:

STRCT3{id n1 n2 n3 {PSTRC pid1} ... {PSTRC pid<n>} {FE eid}}

STRCQ4{id n1 n2 n3 n4 {PSTRC pid1} ... {PSTRC pid<n>} {FE eid}}

Parameter	Description
<i>id</i>	External element identifier
<i>n<i></i>	Reference to the <i>i</i> th node connected to this element
<i>pid<i></i>	Reference to the property field of the <i>i</i> 'th calculation point for this element
<i>eid</i>	Reference to the underlying structural FE element

C.1.7 Strain Coat Properties

The property fields that are referred to by the strain coat element fields have the following syntax:

PSTRC{pid name {PMAT mid} [{PTHICKREF tid}] [{PHEIGHT hid}]}

Parameter	Value Type	Description
<i>pid</i>	Integer	Strain coat property identifier
<i>name</i>	String	Result set name to be displayed in the animation UI (one of "Top", "Bottom" or "Basic")
<i>mid</i>	Integer	Reference to a material property field



Parameter	Value Type	Description
<i>tid</i>	Integer	Reference to a thickness-relative z-position field
<i>hid</i>	Integer	Reference to an absolute z-position field

PTHICKREF{*tid fact {PTHICK gid}}*}

Parameter	Value Type	Description
<i>tid</i>	Integer	z-position identifier
<i>fact</i>	Real	Location of the calculation point in the thickness direction of a shell as a fraction of the referenced shell thickness, i.e. the z-position is $z = fact * t$, where t is the thickness referenced through the parameter <i>gid</i> .
<i>gid</i>	Integer	Reference to a thickness field

PHEIGHT{*hid h*}

Parameter	Value Type	Description
<i>hid</i>	Integer	z
<i>h</i>	Real	Absolute location of the calculation point in the thickness direction of a shell, i.e. the z-position is $z = h$

C.1.8 Other identifiers

The following identifiers are used to define element groups:

GROUP{*id e1 e2 ... en {NAME name}}*}

Parameter	Value Type	Description
<i>id</i>	Integer	Group identifier
<i>ei</i>	Integer	Element ID of the i^{th} element in this group
<i>name</i>	String	Name tag of this group

C.2 Nastran Bulk Data File format

Fedem supports a wide range of Nastran bulk data entries (*see table below*). For most element types, implicit conversion to a known Fedem element type is performed. Refer to Nastran's Bulk Data File documentation for details about properties and syntax for each entry, and the Fedem R5.0 Theory Guide, *Appendix A, "Finite Element Library"* for details about Fedem's library of elements.

FE models in *Nastran bulk data file* format can be directly imported as links into Fedem using the Import Link command (*see Section 4.1.1, "Creating links by file import"*).



NOTE: Each FE model to be imported into Fedem must be stored in a separate bulk data file with the Nastran Bulk Data File extension (.nas or .bdf).

Nastran Bulk Data Conversion	Comments
BAROR	
BEAMOR	
CBAR	
CBEAM	Same as CBAR
CBUSH	
CELAS1	
CELAS2	Same as CELAS1
CHEXA	Supports both 8 and 20 nodes
CONM1	
CONM2	Same as CONM1
CONROD	
CORD1C	
CORD1R	
CORD1S	
CORD2C	
CORD2R	
CORD2S	
CPENTA	Supports both 6 and 15 nodes
CQUAD4	
CQUAD8	Converted to CQUAD4
CROD	Same as CONROD
CTETRA	Supports both 4 and 10 nodes
CTRIA3	
CTRIA6	Converted to CTRIA3

Nastran Bulk Data Conversion	Comments
CWELD	
FORCE	
GRDSET	
GRID	
INCLUDE	
MAT1	See below
MOMENT	
PBAR	
PBARL	See below
PBEAM	
PBEAML	See below
PBUSH	
PELAS	
PLOAD2	
PLOAD4	
PROD	
PSHELL	
PSOLID	
PWELD	
RBAR	
RBE2	
RBE3	
ASET	Automatic definition of external nodes
ASET1	Same as ASET



NOTE: For the PBARL/BPEAML entries, Fedem currently supports the following cross section types (see the MSC/Nastran Reference guide for details): "ROD", "TUBE", "BAR", "BOX", "I" and "T". Any other cross section types have to be manually replaced by equivalent PBAR/PBEAM entries.



NOTE: The shear modulus in the MAT1 bulk entry is only used by beam elements. If the value on the Nastran file is zero for a MAT1 entry that is used by a beam element, the shear modulus is automatically recomputed from the Young's modulus and Poisson's ratio through the formula $G = E/(2+2*\nu)$. However, if $G=0$ is desired for a beam element, that is still possible by editing the ftl-file created in the link_DB directory when the model is saved.



NOTE: If the Poisson's ratio in the MAT1 bulk entry is not given or is outside the valid range $[0, 0.5]$, but the shear modulus is given, the Poisson's ratio will be derived from the Young's modulus and the shear modulus, through the expression $\nu = E/2G - 1$, if that yields a value within the valid range.

Appendix D File Types and Usage

This appendix describes the file types used by Fedem and explains which program module use which files.

Sections in this appendix address the following topics:

- [File types](#)
- [File usage for each program module](#)

D.1 File types

Fedem uses three main types of files: input, intermediate, and results files. Fedem files have an ASCII text or Binary (platform-independent) format, a common tag syntax, and use intuitive naming conventions.

D.1.1 Input files

Fedem uses several types of input files to import FE models, existing mechanism models, and simulation data as listed in the following table.

Ext.	File type	Data	Format
<i>.fmm</i>	Fedem Mechanism Model file	Mechanism Model description	ASCII
<i>.ftl</i>	Fedem Technology Link file	FE model data	ASCII
<i>.ftc</i>	Fedem Technology Cad file	Cad geometry data	ASCII
<i>.nas</i> <i>.bdf</i>	Nastran Bulk Data File	FE model data	ASCII
<i>.flm</i>	Fedem Link Model (used by previous Fedem versions)	FE model data	ASCII
<i>.anf</i> <i>.cdb</i>	Ansys input file	FE model data	ASCII
<i>.inp</i>	ABAQUS input file	FE model data	ASCII
<i>.dac</i>	nCode DAC file	Time history data	Binary
<i>.asc</i> <i>.txt</i>	ASCII text file	Time history data (xy-paired coordinates)	ASCII
<i>.rsp</i> <i>.drv</i> <i>.tim</i>	MTS RPC file	Time history data	Binary
<i>.tpf</i>	TNO Tire file	Tire model description	ASCII
<i>.tir</i>	COSIN Tire file	Tire model description	ASCII
<i>.rdf</i>	Road property file	Road description	ASCII

D.1.2 Intermediate files

Fedem uses the following intermediate files to store analysis options and simulation data from the analyses.

Ext.	File type	Format
<i>.fsi</i>	Fedem Solver Input file	ASCII
<i>.fco</i>	Fedem Calculation Options file*	ASCII
<i>.fop</i>	Fedem Output Options file†	ASCII
<i>.fao</i>	Fedem Additional Options file	ASCII
<i>.fsm</i>	Fedem SAM data file	Binary
<i>.fmx</i>	Fedem Matrix file	Binary

* Changes to files of this type affect the calculated results.

† Changes to files of this type do not affect the calculated results.

D.1.3 Results files

The following table lists the results file types.

Ext.	Filetype	Format
<i>.frs</i>	Fedem results database file	Binary
<i>.res</i>	Fedem results output file	ASCII
<i>.fpp</i>	nCode rainflow matrix file	Binary
<i>.fef</i>	nCode nodal/elemental damage results	ASCII

D.1.4 Other files

The following table lists all other file types used by Fedem.

Ext.	Filetype	Format
<i>.fcd</i>	Fedem unit conversion file	ASCII
<i>.fsn</i>	Fedem S-N curves file	ASCII

D.2 File usage for each program module

The table below is an overview of Fedem file usage.

Fedem application	Uses these file types	Creates these file types
User Interface	FE link files (.ftl, .nas, .bdf, .flm) CAD geometry files (.wrl, .ftc) Time history data files (.asc, .txt, .dac, .rsp, .drv, .tim) – optional Tire description files (.tpf, .tir) – optional Unit conversion file (.fcd) – optional Fedem results files (.frs) S-N curve definition file (.fsn)	Model file (.fmm) FE link files (.ftl) CAD geometry files (.ftc) Solver input files (.fsi) Solver control files (.fco, .fop, .fao)
Fedem Reducer (link reduction)	FE link file (.ftl) Solver control files (.fco, .fop, .fao)	Matrix files (stiffness, mass, gravitation, loads, component modes, recovery) (.fmx) Data structure file (.fsm) Result files (.frs) ASCII results (.res)
Fedem Solver (dynamics solver)	Solver input file (.fsi) Solver control files (.fco, .fop, .fao) Matrix files (stiffness, mass, gravitation, loads) (.fmx) Time history data files (.asc, .txt, .dac, .rsp, .drv, .tim) – optional Tire description files (.tpf, .tir) – optional Road description files (.rdf) – optional CAD geometry files (.ftc) – optional Solver result files (.frs) – optional, in case of restart	Fedem results files (time history response, frequency response) (.frs) ASCII results (.res)
Fedem Stress (stress recovery)	Solver input file (.fsi) Solver control files (.fco, .fop, .fao) Matrix files (recovery, component modes) (.fmx) Data structure file (.fsm) FE link file (.ftl) Solver result files (.frs)	Result files (.frs) ASCII results (.res)

Fedem application	Uses these file types	Creates these file types
Fedem Modes (eigenmode recovery)	Solver input file (. <i>fsi</i>) Solver control files (. <i>fco</i> , . <i>fop</i> , . <i>fao</i>) Matrix files (recovery, component modes) (. <i>fm</i> x) Data structure file (. <i>fsm</i>) FE link file (. <i>ftl</i>) Solver result files (. <i>frs</i>)	Result files (. <i>frs</i>) ASCII results (. <i>res</i>)
Fedem Gage (strain rosette recovery)	Solver input files (. <i>fsi</i>) Solver control files (. <i>fco</i> , . <i>fop</i> , . <i>fao</i>) Rosette input files (. <i>ros</i> , . <i>dat</i>) Matrix files (recovery, component modes) (. <i>fm</i> x) Data structure file (. <i>fsm</i>) FE link file (. <i>ftl</i>) Solver results files (. <i>frs</i>)	Result files (. <i>frs</i>) ASCII results (. <i>res</i> , . <i>asc</i>) DAC time history files (. <i>dac</i>)
Fedem FPP (strain coat recovery)	Solver input file (. <i>fsi</i>) Solver control files (. <i>fco</i> , . <i>fop</i> , . <i>fao</i>) Matrix files (recovery, component modes) (. <i>fm</i> x) Data structure file (. <i>fsm</i>) FE link file (. <i>ftl</i>) Solver result files (. <i>frs</i>) S-N curve definition file (. <i>f</i> sn) – optional	Result files (. <i>frs</i>) ASCII results (. <i>res</i>) nCode rainflow matrix file (. <i>fpp</i>)
Duty Cycle	nCode rainflow matrix file (. <i>fpp</i>)	Result files (. <i>frs</i>)
Fedem GraphExp (curve export utility)	Solver control files (. <i>fco</i> , . <i>fop</i> , . <i>fao</i>) Model file (. <i>fmm</i>) Results files (. <i>frs</i>)	RPC time history data file (. <i>rsp</i>) DAC time history files (. <i>dac</i>) ASCII time history data files (. <i>asc</i>)

Appendix E Command line options

Each of the Fedem programs may be run manually from a console window or using the *Run...* option from the Start menu in Windows. To facilitate such batch-execution of the programs, the complete list of command-line options for each solver module is given in this appendix. Any of these options may also be specified in the Additional Solver Options dialog (see [Section 6.2.3, "Additional solver options"](#)).

The command-line options may contain both upper case and lower case letters. However, the interpretation of the options is case insensitive. For options accepting a numerical (or text string) value, a '=' character may optionally be added between the option and its value. Thus, all the following option specifications are equivalent:

```
-myOption 1.0  
-myoption 1.0  
-MYOPTION 1.0  
-myOption=1.0
```

If you mis-spell or specify a non-existing option, the option is ignored. A warning for each unrecognized option is printed to the console window, or in the *"Output List"* if executed through the user interface.

Sections in this appendix address the following topics:

- [Fedem UI Options \(fedem\)](#)
- [Link reducer options \(fedem_reducer\)](#)
- [Dynamics solver options \(fedem_solver\)](#)
- [Stress recovery options \(fedem_stress\)](#)
- [Mode shape recovery options \(fedem_modes\)](#)
- [Strain rosette recovery options \(fedem_gage\)](#)
- [Strain coat recovery options \(fedem_fpp\)](#)
- [Curve export options \(fedem_graphexp\)](#)

E.1 Fedem UI Options (fedem)

Command-line option	Description	Default value
<code>-checkRDB\interval</code>	Time [ms] between each RDB check/update during solve	500
<code>-connectToSw</code>	Connect to, or create a SolidWorks session	- (false)
<code>-console</code>	Enable console window	- (false)
<code>-debug</code>	Run in debug mode	- (false)
<code>-f</code>	Model file to open	untitled.fmm
<code>-help</code>	Display this help and exit	- (false)
<code>-licenseinfo</code>	Print out license information at startup	- (false)
<code>-logFile</code>	Write all Output List contents to log-file. Log-file name: <modelFilePrefix>.log	+ (true)
<code>-noAddOn</code>	Do not use licenses for add-on modules	- (false)
<code>-noFEData</code>	Load model file without FE-Models and FE-Results info. Use together with <code>-f</code>	- (false)
<code>-plotElements</code>	Enable plotting of element results	- (false)
<code>-plotNodes</code>	Enable plotting of nodal results	- (false)
<code>-prepareBatch</code>	Prepare for batch execution. This option can have the following arguments: all = all solvers reducer = reduction of all links dynamics = dynamics solver stress = stress recovery for in links modes = mode shape recovery in all links straingage = strain gage recovery in all links straincoat = strain coat recovery in all links	
<code>-purgeOnSave</code>	Purge inactive mechanism objects on Save	- (false)
<code>-solve</code>	Start given solver(s) in batch mode. This option can have the following arguments: all = all solvers reducer = reduction of all links dynamics = dynamics solver stress = stress recovery for in links modes = mode shape recovery in all links straingage = strain gage recovery in all links straincoat = strain coat recovery in all links duty cycle = duty cycle in all specified links	
<code>-version</code>	Display program version and exit	- (false)

E.2 Link reducer options (fedem_reducer)

Command-line option	Description	Default value
<code>-autoMassScale</code>	Scale factor for auto-added masses	<i>1e-009</i>
<code>-autoStiffMethod</code>	Method for automatic stiffness computations in auto-added springs = 1: $k = \text{Min}(\text{diag}(K)) * 0.1 / \text{tolFactorize}$ = 2: $k = \text{Mean}(\text{diag}(K)) * \text{autoStiffScale}$ = 3: $k = \text{Max}(\text{diag}(K)) * \text{autoStiffScale}$	3
<code>-autoStiffScale</code>	Scale factor for auto-added springs	<i>100</i>
<code>-Bmatfile</code>	Name of B-matrix file	
<code>-Bmatprecision</code>	Storage precision of the B-matrix on disk = 1: Single precision = 2: Double precision	2
<code>-Bramsize</code>	In-core size (MB) of displacement recovery matrix ≤ 0: Store full matrix in core	-1
<code>-bufsize_rigid</code>	Buffer size (in DP-words) per rigid element ≤ 0: Use conservative estimate computed internally	0
<code>-cachesize</code>	Cache size (KB) to be used by the SPR solver, or core memory (MB) reserved for numerical data for the GSF solver. Applies to the stiffness matrix only when lumped mass is used. = 0: Let all numerical data be in core for the GSF solver	0
<code>-consolemsg</code>	Output error messages to console	- (<i>false</i>)
<code>-cwd</code>	Change working directory	
<code>-datacheck</code>	Do data check only (exiting after data input)	- (<i>false</i>)
<code>-debug</code>	Debug print switch	0
<code>-eigenshift</code>	Shift value for eigenvalue analysis (target frequency for generalized DOFs)	0
<code>-eigfile</code>	Name of eigenvector file	
<code>-extNodes</code>	List of external nodes to use in the reduction (in addition to the nodes specified in the link file)	
<code>-factorMass</code>	Factorize mass matrix in the eigensolver	- (<i>false</i>)
<code>-fao</code>	Read additional options from this file	
<code>-fco</code>	Read calculation options from this file	

Command-line option	Description	Default value
<code>-fop</code>	Read output options from this file	
<code>-frsfile</code>	Name of results database file for mode shapes	
<code>-ftlout</code>	Name of link output file in FTL-format	
<code>-gravfile</code>	Name of gravity force vector file	
<code>-help</code>	Print out this help text	- (false)
<code>-licenseinfo</code>	Print out license information at startup	- (false)
<code>-licensepath</code>	License file directory	
<code>-linkId</code>	Link base-ID number	1
<code>-linkfile</code>	Name of link input file (must be specified)	
<code>-loadfile</code>	Name of load vector file	
<code>-lumpedmass</code>	Use lumped element mass matrices	- (false)
<code>-massfile</code>	Name of mass matrix file	
<code>-neval</code>	Number of eigenvalues/eigenvectors to compute	0
<code>-nevred</code>	Number of eigenvalues to compute for reduced system	12
<code>-ngen</code>	Number of generalized modes	0
<code>-printArray</code>	Additional debug print switch for certain arrays	0
<code>-rdbinc</code>	Increment number for the results database files	1
<code>-resfile</code>	Name of result output file	
<code>-samfile</code>	Name of SAM data file	
<code>-singularityHandler</code>	Option on how to treat singular matrices = 0: Abort on all occurring singularities = 1: Suppress true zero pivots, abort on reduced-to-zero pivots > 1: Suppress all occurring singularities of any kind	1
<code>-skylinesolver</code>	Use the skyline equation solver	- (false)
<code>-sprMass</code>	Use the SPR equation solver for the mass matrix when the GSF equation solver is used for the stiffness matrix	+ (true)
<code>-sprSolver</code>	Use the SPR equation solver	- (false)
<code>-stifffile</code>	Name of stiffness matrix file	
<code>-terminal</code>	File unit number for terminal output	6



Command-line option	Description	Default value
<i>-tolEigval</i>	Max acceptable relative error in eigen-values	<i>1e-008</i>
<i>-tolFactorize</i>	Equation solver singularity criterion (smaller values are less restrictive). The lowest value allowed is 1e-20	<i>1e-012</i>
<i>-tolWAVGM</i>	Geometric tolerance for WAVGM elements	<i>0.0001</i>
<i>-version</i>	Print out program version	<i>- (false)</i>

E.3 Dynamics solver options (fedem_solver)

Command-line option	Description	Default value
<code>-addBC_eigsolver</code>	Use additional BCs on eigensolver	- (false)
<code>-allAccVars</code>	Output all acceleration variables	- (false)
<code>-allAlgorVars</code>	Output all algorithm variables	- (false)
<code>-allCGVars</code>	Output all centre of gravity variables	- (false)
<code>-allControlVars</code>	Output all control variables	- (false)
<code>-allDampCoeff</code>	Output all damper coefficients	- (false)
<code>-allDamperVars</code>	Output all damper variables	- (false)
<code>-allDefVars</code>	Output all deflection variables	- (false)
<code>-allEnergyVars</code>	Output all energy quantities	- (false)
<code>-allEngineVars</code>	Output all engine values	- (false)
<code>-allForceVars</code>	Output all force variables	- (false)
<code>-allFrictionVars</code>	Output all friction variables	- (false)
<code>-allGenDOFVars</code>	Output all generalized DOF variables	- (false)
<code>-allHDDVars</code>	Output all hydrodynamics variables	- (false)
<code>-allJointVars</code>	Output all joint variables	- (false)
<code>-allLengthVars</code>	Output all length variables	- (false)
<code>-allLoadVars</code>	Output all external load variables	- (false)
<code>-allRestartVars</code>	Output all variables needed for restart	+ (true)
<code>-allSecondaryVars</code>	Output all secondary variables	+ (true)
<code>-allSpringVars</code>	Output all spring variables	- (false)
<code>-allStiffVars</code>	Output all spring stiffnesses	- (false)
<code>-allSupelVars</code>	Output all superelement variables	- (false)
<code>-allSystemVars</code>	Output all system variables	- (false)
<code>-allTireVars</code>	Output all tire variables	- (false)
<code>-allTriadVars</code>	Output all triad variables	- (false)
<code>-allVelVars</code>	Output all velocity variables	- (false)
<code>-alphaNewmark</code>	Numerical damping coefficient	0.1
<code>-autoTimeStep</code>	Time stepping procedure = 0: Fixed time step size = 1: Automatically computed time step size	0
<code>-centripForceCorr</code>	Use centripetal force correction	- (false)

Command-line option	Description	Default value
<code>-consolemsg</code>	Output error messages to console	- (false)
<code>-ctrlAccuracy</code>	Accuracy parameter for control iterations	0.5
<code>-ctrlTolAbs</code>	Absolute tolerance in control iterations	0.002
<code>-ctrlTolRel</code>	Relative tolerance in control iterations	0.002
<code>-ctrlfile</code>	Name of control system database file	<code>ctrl.frs</code>
<code>-curveFile</code>	Name of curve definition file	<code>response.bak.fmm</code>
<code>-curvePlotFile</code>	Name of curve export output file	
<code>-curvePlotPrec</code>	Output precision for exported curve data files = 0 : half precision (int*2) = 1 : single precision (real*4) = 2 : double precision (real*8)	0
<code>-curvePlotType</code>	Format of curve export output file = 0 : ASCII (separate file for each curve) = 1 : DAC, Windows (separate file for each curve) = 2 : DAC, UNIX (separate file for each curve) = 3 : RPC, Windows (all curves in one file) = 4 : RPC, UNIX (all curves in one file) = 5 : ASCII (all curves in one file)	0
<code>-cutbackFactor</code>	Time step reduction factor in cut-back	1
<code>-cutbackSing</code>	Try cut-back when detecting singularities	- (false)
<code>-cutbackSteps</code>	Number of cut-back steps	0
<code>-cwd</code>	Change working directory	
<code>-damped</code>	Solve the damped eigenproblem using LAPACK	- (false)
<code>-datacheck</code>	Do data check only (exiting after data input)	- (false)
<code>-debug</code>	Debug print switch	0
<code>-delayBuffer</code>	Initial buffer size for delay elements	1000
<code>-densesolver</code>	Use LAPACK dense matrix equation solver	- (false)
<code>-double1</code>	Save primary variables in double precision	+ (true)
<code>-double2</code>	Save secondary variables in double precision	- (false)
<code>-effModalMass</code>	Compute the effective mass for each mode	- (false)
<code>-eigenshift</code>	Shift value for vibration eigenvalue analysis (negative value captures zero frequency modes)	0

Command-line option	Description	Default value
<code>-eiginc</code>	Time between each eigenvalue analysis	0
<code>-factorMass_eigen\ solver</code>	Factor mass matrix in eigensolver	- (false)
<code>-fao</code>	Read additional options from this file	
<code>-fco</code>	Read calculation options from this file	
<code>-flushinc</code>	Time between each database file flush < 0.0: Do not flush results database (let the OS decide) = 0.0: Flush at each time step, no external buffers > 0.0: Flush at specified time interval, use external buffers	0
<code>-fop</code>	Read output options from this file	
<code>-frs1file</code>	Name of primary response database file	<i>th_p.frs</i>
<code>-frs2file</code>	Name of secondary response database file	<i>th_s.frs</i>
<code>-fsi2file</code>	Name of additional solver input file	
<code>-fsifile</code>	Name of solver input file	<i>fedem_solver.fsi</i>
<code>-GSFsolver</code>	Use the GSF sparse equation solver	- (false)
<code>-help</code>	Print out this help text	- (false)
<code>-ignoreIC</code>	Ignore initial conditions from the fsi-file	- (false)
<code>-initEquilibrium</code>	Initial static equilibrium iterations	- (false)
<code>-JacobiPerturb</code>	Relative perturbation for computation of numerical Jacobian in control iterations	1e-005
<code>-lancz1</code>	Use the LANCZ1 eigensolver	- (false)
<code>-licenseinfo</code>	Print out license information at startup	- (false)
<code>-licensepath</code>	License file directory	
<code>-limInitEquilStep</code>	Initial equilibrium step size limit	1
<code>-lineSearch</code>	Use line search in the nonlinear iterations	- (false)
<code>-maxInc</code>	Maximum time increment	0.05
<code>-maxSeqNoUpdate</code>	Max number of sequential iterations without system matrix update	100
<code>-maxit</code>	Maximum number of iterations	15
<code>-minInc</code>	Minimum time increment	0.001
<code>-minit</code>	Minimum number of iterations	1
<code>-modesfile</code>	Name of primary modes database file	<i>ev_p.frs</i>
<code>-monitorIter</code>	Number of iterations to monitor before maxit	2

Command-line option	Description	Default value
<i>-monitorWorst</i>	Number of DOFs to monitor on poor convergence	6
<i>-noStiffDamp\Filtering</i>	Turn off rigid body filtering of stiffness-proportional damping	- (<i>false</i>)
<i>-nosolveropt</i>	Switch off equation system reordering	- (<i>false</i>)
<i>-numEigModes</i>	Number of eigenmodes to calculate	0
<i>-num_damp_energy_skip</i>	Number of steps without calculation of energy from stiffness proportional damping	1
<i>-numit</i>	Fixed number of iterations	0
<i>-nupdat</i>	Number of iterations with system matrix update	0
<i>-printinc</i>	Time between each print to result output file	0
<i>-quasiStatic</i>	Do a quasi-static simulation to this time	0
<i>-rdbinc</i>	Increment number for the results database files	1
<i>-resfile</i>	Name of result output file	<i>fedem_solver.res</i>
<i>-restartfile</i>	Response database file(s) to restart from	
<i>-restarttime</i>	Physical time for restart < 0: No restart, but regular simulation	-1
<i>-rpcFile</i>	Get number of repeats, averages, and points per frame and group, from this RPC-file	
<i>-saveinc2</i>	Time between each save of secondary variables	0
<i>-saveinc3</i>	Time between each save for external recovery	0
<i>-saveinc4</i>	Time between each save of control system data	0
<i>-savestart</i>	Time for first save to response database	0
<i>-scaleToKG</i>	Scaling factor to SI mass unit [kg]	1
<i>-scaleToM</i>	Scaling factor to SI length unit [m]	1
<i>-scaleToS</i>	Scaling factor to SI time unit [s]	1
<i>-skylinesolver</i>	Use skyline solver	- (<i>false</i>)
<i>-stopOnDivergence</i>	Number of warnings on possible divergence before the dynamics simulation is aborted (0 = no limit)	0

Command-line option	Description	Default value
<code>-stressStiffDyn</code>	Use geometric stiffness for dynamics	<code>- (false)</code>
<code>-stressStiffEig</code>	Use geometric stiffness for eigenvalue analysis	<code>- (false)</code>
<code>-stressStiffEqu</code>	Use geometric stiffness for statics	<code>- (false)</code>
<code>-stressStiff\UpdateSkip</code>	Number of iterations without updating stress stiffening (always updated in predictor step)	<code>0</code>
<code>-targetFrequency\Rigid</code>	Target frequency for auto-stiffness calculation	<code>10000</code>
<code>-terminal</code>	File unit number for terminal output	<code>6</code>
<code>-timeEnd</code>	Stop time	<code>0</code>
<code>-timeInc</code>	Initial time increment	<code>0</code>
<code>-timeStart</code>	Start time	<code>0</code>
<code>-tolAccGen</code>	Max generalized acceleration tolerance	<code>0 *</code>
<code>-tolAccNorm</code>	Acceleration vector convergence tolerance	<code>0 *</code>
<code>-tolAccRot</code>	Max angular acceleration tolerance	<code>0 *</code>
<code>-tolAccTra</code>	Max acceleration tolerance	<code>0 *</code>
<code>-tolDispGen</code>	Max generalized DOF tolerance	<code>0 *</code>
<code>-tolDispNorm</code>	Displacement vector convergence tolerance	<code>0 *</code>
<code>-tolDistRot</code>	Max rotation tolerance	<code>0 *</code>
<code>-tolDispTra</code>	Max displacement tolerance	<code>0 *</code>
<code>-tolEigval</code>	Max acceptable relative error in eigenvalues	<code>1e-008</code>
<code>-tolEigvector</code>	Orthogonality limit for the eigenvectors	<code>1e-008</code>
<code>-tolEnerMax</code>	Max energy in a single DOF tolerance	<code>0 *</code>
<code>-tolEnerSum</code>	Energy norm convergence tolerance	<code>0 *</code>
<code>-tolFactCtrl</code>	Singularity criterion for the internal control system solver (smaller value less restrictive)	<code>1e-012</code>
<code>-tolFactDyn</code>	Linear solver singularity criterion for dynamics solver (smaller value less restrictive)	<code>1e-012</code>
<code>-tolFactorize</code>	Linear solver singularity criterion for initial equilibrium and eigenvalue analysis (smaller value less restrictive)	<code>1e-009</code>

Command-line option	Description	Default value
<i>-tolInitEquil</i>	Convergence tolerance for initial equilibrium iterations	<i>0.001</i>
<i>-tolResGen</i>	Max residual generalized DOF force tolerance	<i>0 *</i>
<i>-tolResNorm</i>	Residual force vector convergence tolerance	<i>0 *</i>
<i>-tolResRot</i>	Max residual torque tolerance	<i>0 *</i>
<i>-tolResTra</i>	Max residual force tolerance	<i>0 *</i>
<i>-tolUpdateFactor</i>	Convergence criterion for continuing matrix updates	<i>0 *</i>
<i>-tolVelGen</i>	Max generalized velocity tolerance	<i>0 *</i>
<i>-tolVelNorm</i>	Velocity vector convergence tolerance	<i>0 *</i>
<i>-tolVelRot</i>	Max angular velocity tolerance	<i>0 *</i>
<i>-tolVelTra</i>	Max velocity tolerance	<i>0 *</i>
<i>-version</i>	Print out program version	<i>- (false)</i>
<i>-VTFfile</i>	<i>Name of VTF output file</i>	

* For all the convergence tolerance options, its value is interpreted as follows:
= 0 : This tolerance is ignored
> 0 : This tolerance is in a set of tests where all must be satisfied
< 0 : This tolerance is in a set of tests where only one must be satisfied
(using the absolute value as the actual tolerance value)

E.4 Stress recovery options (fedem_stress)

Command-line option	Description	Default value
<code>-Bmatfile</code>	Name of B-matrix file	
<code>-Bramsize</code>	In-core size (MB) of displacement recovery matrix < 0: Use the same as in the reducer = 0: Store full matrix in core	-1
<code>-consolemsg</code>	Output error messages to console	- (false)
<code>-cwd</code>	Change working directory	
<code>-datacheck</code>	Do data check only (exiting after data input)	- (false)
<code>-debug</code>	Debug print switch	0
<code>-deformation</code>	Save deformations to results database	- (false)
<code>-double</code>	Save all results in double precision	- (false)
<code>-dumpDefNas</code>	Save deformations to Nastran bulk data files	- (false)
<code>-eigfile</code>	Name of eigenvector file	
<code>-fao</code>	Read additional options from this file	
<code>-fco</code>	Read calculation options from this file	
<code>-fop</code>	Read output options from this file	
<code>-frsfile</code>	Name of solver results database file	
<code>-fsifile</code>	Name of solver input file	<i>fedem_solver.fsi</i>
<code>-group</code>	List of element groups to do calculations for	
<code>-help</code>	Print out this help text	- (false)
<code>-licenseinfo</code>	Print out license information at startup	- (false)
<code>-licensepath</code>	License file directory	
<code>-linkId</code>	Link base-ID number	0
<code>-linkfile</code>	Name of link input file	
<code>-maxPStrain</code>	Save max principal strain to results database	- (false)
<code>-maxPStress</code>	Save max principal stress to results database	- (false)
<code>-maxSStrain</code>	Save max shear strain to results database	- (false)
<code>-maxSSstress</code>	Save max shear stress to results database	- (false)
<code>-minPStrain</code>	Save min principal strain to results database	- (false)

Command-line option	Description	Default value
<code>-minPStress</code>	Save min principal stress to results database	- (<i>false</i>)
<code>-nodalForces</code>	Compute and print nodal forces	- (<i>false</i>)
<code>-rdbfile</code>	Name of stress results database file	
<code>-rdbinc</code>	Increment number for the results database file	1
<code>-resfile</code>	Name of result output file	
<code>-resStressFile</code>	Name of residual stress input file	
<code>-resStressSet</code>	Name of residual stress set	
<code>-samfile</code>	Name of SAM data file	
<code>-SR</code>	Save stress resultants to results database	- (<i>false</i>)
<code>-statm</code>	Start time	0
<code>-stotm</code>	Stop time	1
<code>-strain</code>	Save strain tensors to results database	- (<i>false</i>)
<code>-stress</code>	Save stress tensors to results database	- (<i>false</i>)
<code>-terminal</code>	File unit number for terminal output	6
<code>-tinc</code>	Time increment (= 0.0: process all time steps)	0.1
<code>-version</code>	Print out program version	- (<i>false</i>)
<code>-vmStrain</code>	Save von Mises strain to results database	- (<i>false</i>)
<code>-vmStress</code>	Save von Mises stress to results database	- (<i>false</i>)
<code>-VTFavgelm</code>	Write averaged element results to VTF-file	+ (<i>true</i>)
<code>-VTFdscale</code>	Deformation scaling factor for VTF output	1
<code>-VTFfile</code>	Name of VTF output file	
<code>-VTFinit</code>	Write initial state to VTF-file	- (<i>false</i>)
<code>-VTFoffset</code>	TF result block id offset	0
<code>-VTFparts</code>	Number of parts in VTF-file	0
<code>-write_nodes</code>	Save deformations as nodal data	+ (<i>true</i>)
<code>-write_vector</code>	Save deformations as vector data	- (<i>false</i>)

E.5 Mode shape recovery options (fedem_modes)

Command-line option	Description	Default value
<i>-Bmatfile</i>	Name of B-matrix file	
<i>-Bramsize</i>	In-core size (MB) of displacement recovery matrix < 0: Use the same as in the reducer = 0: Store full matrix in core	-1
<i>-consolemsg</i>	Output error messages to console	- (false)
<i>-cwd</i>	Change working directory	
<i>-damped</i>	Complex modes are calculated	- (false)
<i>-datacheck</i>	Do data check only (exiting after data input)	- (false)
<i>-debug</i>	Debug print switch	0
<i>-double</i>	Save all results in double precision	- (false)
<i>-eigfile</i>	Name of eigenvector file	
<i>-energy_density</i>	Save scaled strain energy density	- (false)
<i>-fao</i>	Read additional options from this file	
<i>-fco</i>	Read calculation options from this file	
<i>-fop</i>	Read output options from this file	
<i>-frsfile</i>	Name of solver results database file	
<i>-fsifile</i>	Name of solver input file	<i>fedem_solver.fsi</i>
<i>-help</i>	Print out this help text	- (false)
<i>-licenseinfo</i>	Print out license information at startup	- (false)
<i>-licensepath</i>	-licensepath	
<i>-linkId</i>	Link base-ID number	0
<i>-linkfile</i>	Name of link input file	
<i>-rdbfile</i>	Name of modes results database file	
<i>-rdbinc</i>	Increment number for the results database file	1
<i>-recover_modes</i>	List of mode numbers to expand	
<i>-resfile</i>	Name of result output file	
<i>-samfile</i>	Name of SAM data file	
<i>-terminal</i>	File unit number for terminal output	6
<i>-version</i>	Print out program version	- (false)

Command-line option	Description	Default value
-VTFdscale	Deformation scaling factor for VTF output	1
-VTFexpress	Write express VTF-files (one file per mode)	- (false)
-VTFfile	Name of VTF output file	
-VTFoffset	VTF result block id offset	0
-VTFparts	Number of parts in VTF-file	0
-write_nodes	Save results as nodal data	- (false)
-write_vector	Save results as vector data	+ (true)

E.6 Strain rosette recovery options (fedem_gage)

Command-line option	Description	Default value
<i>-binSize</i>	Bin size for stress cycle counting [MPa]	10
<i>-Bmatfile</i>	Name of B-matrix file	
<i>-Bramsize</i>	In-core size (MB) of displacement recovery matrix < 0: Use the same as in the reducer = 0: Store full matrix in core	-1
<i>-consolemsg</i>	Output error messages to console	- (false)
<i>-cwd</i>	Change working directory	
<i>-dac_sampleinc</i>	Sampling increment for dac output files	0.001
<i>-datacheck</i>	Do data check only (exiting after data input)	- (false)
<i>-debug</i>	Debug print switch	0
<i>-deformation</i>	Save nodal deformations to results database	- (false)
<i>-eigfile</i>	Name of eigenvector file	
<i>-fao</i>	Read additional options from this file	
<i>-fatigue</i>	Perform damage calculation on the gage stresses	0
<i>-fco</i>	Read calculation options from this file	
<i>-flushinc</i>	Time between each database file flush < 0.0: Do not flush results database (let the OS decide) = 0.0: Flush at each time step, no external buffers > 0.0: Flush at specified time interval, use external buffers	-1
<i>-fop</i>	Read output options from this file	
<i>-frsfile</i>	Name of solver results database file	
<i>-fsifile</i>	Name of solver input file	<i>fedem_solver.fsi</i>
<i>-gate</i>	Stress gate value for the damage calculation [MPa]	25
<i>-help</i>	Print out this help text	- (false)
<i>-licenseinfo</i>	Print out license information at startup	- (false)
<i>-licensepath</i>	License file directoryd:\src\vpmmain\vpmmBase\vpmmStress\lm	
<i>-linkId</i>	Link base-ID number	0
<i>-linkfile</i>	Name of link input file	

Command-line option	Description	Default value
<code>-littleEndian</code>	Use Little Endian formatting of DAC files	- (<i>false</i>) [*]
<code>-loga1</code>	Parameter log(a1) of the S-N curve	15.117
<code>-loga2</code>	Parameter log(a2) of the S-N curve	17.146
<code>-m1</code>	Parameter m1 of the S-N curve	4
<code>-nullify_start_\rosettestrains</code>	Set start strains to zero for the rosettes	- (<i>false</i>)
<code>-rdbfile</code>	Name of strain gage results database file	
<code>-rdbinc</code>	Increment number for the results database file	1
<code>-resfile</code>	Name of result output file	
<code>-rosfile</code>	Name of strain rosette input file	
<code>-samfile</code>	Name of SAM data file	
<code>-statm</code>	Start time	0
<code>-stotm</code>	Stop time	1
<code>-terminal</code>	File unit number for terminal output	6
<code>-tinc</code>	Time increment (= 0.0: process all time steps)	0
<code>-version</code>	Print out program version	- (<i>false</i>)
<code>-writeAsciiFiles</code>	Write rosette results to ASCII files	- (<i>false</i>)

* This is the default value on UNIX platforms. On Windows, the default is + (*true*). Thus, the default formatting of the DAC files will be suitable for the platform the recovery is run on.

E.7 Strain coat recovery options (fedem_fpp)

Command-line option	Description	Default value
<code>-angleBins</code>	Number of bins in search for most popular angle	541
<code>-biAxialGate</code>	Gate value for the biaxiality calculation	10
<code>-blockSize</code>	Max number of elements processed together	2000
<code>-Bmatfile</code>	Name of B-matrix file	
<code>-Bramsize</code>	In-core size (MB) of displacement recovery matrix < 0: Use the same as in the reducer = 0: Store full matrix in core	-1
<code>-BufSizeInc</code>	Buffer increment size	20
<code>-consolemsg</code>	Output error messages to console	- (false)
<code>-cwd</code>	Change working directory	
<code>-datacheck</code>	Do data check only (exiting after data input)	- (false)
<code>-debug</code>	Debug print switch	0
<code>-double</code>	Save results in double precision	- (false)
<code>-eigfile</code>	Name of eigenvector file	
<code>-fao</code>	Read additional options from this file	
<code>-fco</code>	Read calculation options from this file	
<code>-fop</code>	Read output options from this file	
<code>-fppfile</code>	Name of fpp output file	
<code>-frsfile</code>	Name of solver results database file	
<code>-fsifile</code>	Name of solver input file	<i>fedem_solver.fsi</i>
<code>-group</code>	List of element groups to do calculations for	
<code>-help</code>	Print out this help text	- (false)
<code>-HistDataType</code>	Histogram data type = 0: None = 1: Signed abs max stress = 2: Signed abs max strain	0
<code>-HistXBins</code>	Histogram number of X-bins	64
<code>-HistXMax</code>	Histogram max X-value	100
<code>-HistXMin</code>	Histogram min X-value	-100
<code>-HistYBins</code>	Histogram number of Y-bins	64

Command-line option	Description	Default value
<code>-HistYMax</code>	Histogram max Y-value	100
<code>-HistYMin</code>	Histogram min Y-value	-100
<code>-licenseinfo</code>	Print out license information at startup	- (false)
<code>-licensepath</code>	License file directory	
<code>-linkId</code>	Link base-ID number	0
<code>-linkfile</code>	Name of link input file	
<code>-PVXGate</code>	Gate value for the Peak Valley extraction (MPa or microns depending on Hist-DataType)	10
<code>-rdbfile</code>	Name of strain coat results database file	
<code>-rdbinc</code>	Increment number for the results database file	1
<code>-resfile</code>	Name of result output file	
<code>-resStressFile</code>	Name of residual stress input file	
<code>-resStressSet</code>	Name of residual stress set	
<code>-samfile</code>	Name of SAM data file	
<code>-SNfile</code>	Name of SN-curve definition file	
<code>-statm</code>	Start time	0
<code>-stotm</code>	Stop time	1
<code>-stressToMPaScale</code>	Stress conversion factor to MPa	1e-06
<code>-surcface</code>	Surface selection option = 0: All element surfaces = 1: Bottom shell surfaces only = 2: Middle shell surfaces only = 3: Top shell surfaces only	0
<code>-terminal</code>	File unit number for terminal output	6
<code>-tinc</code>	Time increment (= 0.0: process all time steps)	0
<code>-version</code>	Print out program version	- (false)

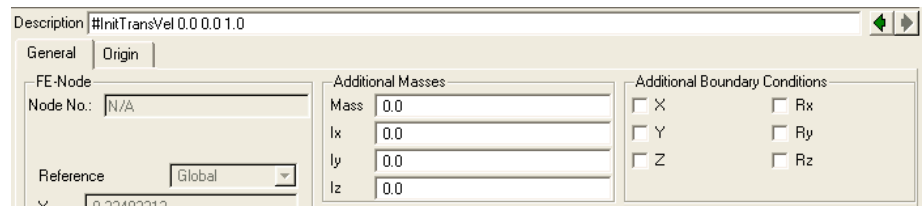
E.8 Curve export options (fedem_graphexp)

Command-line option	Description	Default value
<i>-curvePlotFile</i>	Name of curve export output file	<i>response.rsp</i>
<i>-curvePlotPrec</i>	Output precision for curve data files = 0 : half precision (int*2) = 1 : single precision (real*4) = 2 : double precision (real*8)	0*
<i>-curvePlotType</i>	Format of curve export output file = 0 : ASCII (separate file for each curve) = 1 : DAC, Windows (separate file for each curve) = 2 : DAC, UNIX (separate file for each curve) = 3 : RPC, Windows (all curves in one file) = 4 : RPC, UNIX (all curves in one file) = 5 : ASCII (all curves in one file)	3
<i>-cwd</i>	Change working directory	
<i>-fao</i>	Read additional options from this file	
<i>-fco</i>	Read calculation options from this file	
<i>-fop</i>	Read output options from this file	
<i>-frsFile</i>	List of results database files	
<i>-help</i>	Display this help and exit	- (false)
<i>-licenseinfo</i>	Print out license information at startup	- (false)
<i>-licensepath</i>	License file directory	
<i>-modelFile</i>	Name of model file with curve definitions	
<i>-rpcFile</i>	Get number of repeats, averages, and points per frame and group, from this RPC-file	
<i>-version</i>	Display program version and exit	- (false)

* The *-curvePlotPrec* option is effective for multi-column ASCII and RPC files only. The default value 0 (half precision) is applicable to RPC files only. For ASCII files, the default value is 1 (single precision). DAC files and single-column ASCII files are always written in single precision. This footnote also applies to the *-curvePlotPrec* option of the Dynamics Solver.

Appendix F Beta feature documentation

Some new features in Fedem are still in a state of Beta testing. These features are typically available through commands or options entered in the description field of an entity's property window. This means to enter the special character "#" followed by a keyword and possibly some values into the description field along with the user description, like in the example shown below. See also [Section 2.5.5, "Property Editor"](#).



Due to the Beta nature of the features, they should be used with some care. The present way of accessing these features through the options or commands in the description field is subject to change, or no longer being supported, in future releases. However, most of the features listed here will be supported in a more "permanent" fashion in future releases.

Sections in this appendix address the following topics:

- [Joints](#)
- [Links](#)
- [Springs](#)
- [Frictions](#)
- [Tires](#)
- [Roads](#)
- [Additional masses](#)
- [Sensors](#)
- [Generic database objects](#)
- [Initial conditions for dynamics analysis](#)
- [Boundary conditions for dynamics analysis](#)
- [Prescribed Triad motion](#)

F.1 Joints

F.1.1 Universal Joint

The universal joint is obtained from the Ball Joint when the command

```
#UniversalJoint
```

is entered in the description field of the selected Ball Joint. The initial orientation of the joint cross is assumed to be the Y and Z-axis of the Ball Joint master triad. The Z-axis of the cross is connected to the master link whereas the Y-axis is connected to the slave link. The user must thus reorient the master triad appropriately for the desired joint configuration. Attention must also be paid to the selection of master versus slave link.



NOTE: The universal joint has only two independent joint DOFs (the Y- and Z-rotations). Specifying a spring stiffness and/or damping for the third DOF (X-rotation) for this joint is therefore meaningless, although that is possible in the property window. Any such properties specified on this DOF will be silently ignored. The same is true for initial conditions.

F.1.2 Constant Velocity Joint

A Constant Velocity Joint is obtained from a Ball Joint by entering the command

```
#CVJoint (#RZ <float>) (#RY <float>)
```

in the description field of the selected Ball Joint.

The X-axis of the master triad should be oriented along the rotation axis of the master shaft (link). The direction of the slave shaft rotation axis, the X-axis, is determined by one, or both, of the optional additional commands

```
#RZ <float> #RY <float>
```

#RZ gives direction of slave X-axis relative to master X-axis by a rotation about the Z-axis of the master triad. #RY obtains the X-axis from the master X-axis by a rotation about the Y-axis of the master triad. Care should be exercised when applying both rotations to obtain the X-axis of the slave shaft since the angles are Euler-Z-Y rotations. It is recommended that the master triad is oriented such that one only needs to use #RZ or #RY to define the X-axis of the slave link.



NOTE: The constant velocity joint has only two independent joint DOFs (the Y- and Z-rotations). Specifying a spring stiffness and/or damping for the third DOF (X-rotation) for this joint is therefore meaningless, although that is possible in the property window. Any spring/damper property specified on this DOF will be silently ignored. The same is true for initial conditions.

F.1.3 Rigid Joint

The DOFs of the Rigid Joint can be released individually by entering one or more of the commands

```
#FreeX #FreeY #FreeZ #FreeRX #FreeRY #FreeRZ
```

in the description field of the selected Rigid Joint. The rotational parameterization is rotation axis components. All DOFs are referred to master triad coordinate system.

F.1.4 Axial Joint

An Axial Joint is obtained from the Free Joint when the command

```
#Axial
```

is entered in the description field of the selected Free Joint. This is a single-DOF joint that works like an Axial Spring and/or Damper, except that the length between the two triads here is used as a joint DOF. The **Tx** tab of the joint property panel (see [Section 4.4.2, "Joint properties"](#)) is used to control the behavior of the length DOF. All the other DOF tabs of the property panel are not used when `#Axial` is specified.

The advantage of using an Axial Joint opposed to a combined Axial Spring and Damper is that you now are able to prescribe the length between the two triads directly, without the need for stiff springs and associated damping, which might render the model numerically unstable. It also reduces the total number of DOFs in the mechanism compared with an Axial Spring/Damper (reduced by 5 for each Axial Joint). More details on the formulation of the Axial Joint may be found in the Fedem R5.0 Theory Guide, [Section 6.2.7, "Axial Joint"](#).

F.1.5 Free Joint

An alternative Free Joint formulation is now available through the specification of the following description field command

```
#GlobalSpring
```

The default master-slave joint formulation is then not used (which for the Free Joint only is a transformation of the free DOFs from the slave Triad to the joint variables, and no constraining). With the alternative formulation the added spring (and damper) properties are applied directly in the global coordinate directions between the two triads instead, and no change in free variables. There are no eccentricity contributions either when the force between the two triads is not attacking along their common axis. Consequently, this free joint is somewhat equivalent to a BUSH element on the system level (see the Fedem R5.0 Theory Guide, *Section E.10, "BUSH"*).

F.1.6 Prismatic Joint and Cylindric Joint

Both Prismatic and Cylindric Joint can take the command

#Extended

in the description field. Entering this command allows the follower to travel beyond the first and last triad of the track. When the follower is past the beginning of the track, the first two triads receive the contact force in a statically consistent manner. When the follower is past the last triad the last two triads receive the contact force.

The default rotational parameterization of the Cylindric Joint is Euler-Z-Y-X rotations, but this can be changed to components of the rotation axis vector by entering the command

#RotAxisParam

in the description field of the selected Cylindric Joint. This command can also be applied to the Cam Joint when the alternative master-slave based formulation is used (see [Section F.1.7, "Cam Joint"](#) below).

F.1.7 Cam Joint

The default cam formulation uses non-linear springs to model contact forces between the follower (slave triad), and the master triads along the cam curve. This formulation has independent DOFs at the follower and all master triads, and is able to handle large separation between cam and follower.

An alternative formulation, in which the motion of the follower is expressed in a curvilinear coordinate system surrounding the cam surface and the slave triad is treated more like a true slave of the master triads, is obtained by entering the following command in the description field:

```
#MasterSlaveCam
```

With this formulation one can also enter one, or both, of the commands:

```
#FixX #FixY
```

The joint DOFs in these directions are then eliminated (along with the joint springs and dampers), and the follower is restricted to follow the cam in that direction. Using these options can give a numerically more stable formulation when no separation between the cam and follower is expected (or possible). The master-slave cam formulation is only able to handle small separation (relative to the cam-segment radius) between follower and cam surface. A cam surface with corners can also cause problems with this alternative formulation.

F.2 Links

F.2.1 Geometric stiffness

The **Geometric stiffness contribution** toggle in the *"Integration tab"* of the Dynamics Solver Setup dialog (see [Section 6.5.1, "Dynamics Solver Setup"](#)) applies normally to all flexible links in the model during the dynamics simulation. It is possible to override this setting for a specific link using the following description field commands:

```
#DynStressStiffening    - Enables geometric stiffness for this link
```

```
#NoDynStressStiffening  - Disables geometric stiffness for this link
```

The commands affect the dynamics simulation only. They have no effect during the initial equilibrium and eigenmode analyses. They have no effect for generic parts with automatic stiffness calculation toggled on.

F.2.2 Component modes

The number of **Component modes** that is entered in the Property Editor panel for a selected link, defines how many component mode shapes that should be computed during the reduction of that link, see *"Reduction Options tab"* in [Section 4.1.4, "Link properties"](#). Thus, whenever you change this number, the link might need to be reduced again. The default is to use all the computed component mode shapes as degrees of freedom for the link during the dynamics simulation. However, it is possible to use

only a subset of the computed modes, by specifying the following description field commands for the link:

`#InclModes <m1> <m2> ...` - Use only these component modes

`#ExclModes <m1> <m2> ...` - Use all but these component modes

The structural damping parameters entered in the Property Editor panel for a selected link, see "*Part tab*" in [Section 4.1.4, "Link properties"](#), are by default applied to the whole stiffness- and mass matrix of the link. However, it is possible to specify individual Rayleigh damping factors for each of the component modes through the following description field commands:

`#Alpha1 <c1> <c2> ...` - Mass-proportional damping factors for the component modes

`#Alpha2 <c1> <c2> ...` - Stiffness-proportional damping factors for the component modes

Thus, `<c1>` is the damping factor applied to the first component mode used, `<c2>` is the factor for the second mode, etc. If you specify fewer such individual damping factors than the number of component modes being used, the last value entered is used for all the remaining modes.

F.3 Springs

Spring stiffness for both axial and joint springs can be adjusted using scale functions. Different adjustment for positive and negative deflection of a spring is often desirable when adjusting the stiffness of, for instance, hydraulic cylinders. To achieve this, use the description field commands:

`#PosStiffScaleEngine <id>`

`#NegStiffScaleEngine <id>`

The `<id>` is the function's base ID. These commands set the scale function only for the respective deflection state. If used, they will override the scale function selected through the Spring property panel, if any.



NOTE: The base ID of a mechanism object is normally not visible in the Fedem UI. To see the base ID, you have to launch Fedem in debug mode (using command-line option `-debug`). Then the base ID appears in curly braces `{}` in the Object Browser.



TIP: You can access the description field of a joint spring by double-clicking the desired spring entry in the Topology view of the actual joint.

F.4 Frictions

An alternative friction formulation is available by specifying the following command in the description field of a friction object:

```
#Kstick <k>
```

where <k> is the stiffness that is used to enforce that the friction DOF has no movement when it is in a stick condition. This friction formulation is based on the use of a spring with a varying yield criterion (see [Section 4.7.6, "Advanced spring characteristics"](#)). That is, the Max Yield Force is taken as the maximum occurring friction force before the friction DOF is slipping (see the Fedem R5.0 Theory Guide, [Section 6.5.4 "Total friction"](#)).

F.5 Tires

The following commands can be entered in the description field of a Tire to alter the characteristics of the tire models (MF-Tyre and SWIFT only):

- #SteadyState - use the steady state formulation instead of the dynamic one.
- #MirrorTChar - mirror the tire characteristics about the local XY-plane.
- #ISWTCH <n> - directly specify the value of the ISWTCH parameter in the DTYRE call (see STI documentation).

Radial stiffness and damping coefficients of the tire may be specified through:

```
#radialStiff <k>
```

```
#radialDamp <c>
```

These parameters will be added to the system Newton matrix of the mechanism. They do not affect the tire models internally.

F.6 Roads

The following commands can be entered in the description field of a Road to apply a time-dependent translation of the road surface (applicable for the MF-Tyre and SWIFT tire models only):

```
#RoadXengId <id>
```

```
#RoadYengId <id>
```

```
#RoadZengId <id>
```

where <id> is the base ID of the Function defining the time-dependent road translation in the indicated global coordinate direction.



NOTE: The base ID of a mechanism object is normally not visible in the Fedem UI. To see the base ID, you have to launch Fedem in debug mode (using command-line option `-debug`). Then the base ID appears in curly braces {} in the Object Browser.

F.7 Additional masses

The additional masses are scaled using a Function when the command

```
#MassScaleEngine <id>
```

is entered in the description field for the selected Triad. The <id> is the function's base ID. The additional rotational inertias for the selected triad are also scaled using the same function.



NOTE: The base ID of a mechanism object is normally not visible in the Fedem UI. To see the base ID, you have to launch Fedem in debug mode (using command-line option `-debug`). Then the base ID appears in curly braces {} in the Object Browser.

The additional masses may be applied in a specified local triad direction only, by specifying one of the following commands in the Triad description field (the commands only affects the translational mass):

```
#MassDir <cx> <cy> <cz>
```

```
#MassX
```

```
#MassY
```

```
#MassZ
```

where $\langle cx \rangle$, $\langle cy \rangle$ and $\langle cz \rangle$ defines the local triad direction the additional mass should be applied in. `#MassX` is equivalent to `#MassDir 1 0 0`, `#MassY` is equivalent to `#MassDir 0 1 0`, and `#MassZ` is equivalent to `#MassDir 0 0 1`.

An additional mass is regarded as a *virtual added mass* when

`#AddedMass <mx> <my> <mz>`

is entered in the description field of the Triad description field. The additional mass then contributes to the mass matrix and inertia forces only, and not to the gravitational force vector. The added mass terms then equal $\{ \langle M \rangle * \langle mx \rangle, \langle M \rangle * \langle my \rangle, \langle M \rangle * \langle mz \rangle \}$ in the local triad directions, where $\langle M \rangle$ is the value specified in the *Mass* field.

F.8 Sensors

Sensors on triads can measure rotations in terms of Rodriguez rotations by entering the command

`#Rodrig`

in the description field for the selected Sensor. The angular quantities measured by this sensor are then the component of rotation vector, as defined in the Fedem R5.0 Theory Guide, *Section 2.3.3 "Rodriguez parameterization"*. The default is to measure Euler-Z-Y-X angles.

F.9 Generic database objects

A generic database object (not to be confused with *Generic Part* links) can be used to access new and undocumented features in the dynamics solver. A generic database object is created by selecting **Generic DB Object** from the **Mechanism** menu. The object is manipulated (edited and deleted) in the same way as for the other Fedem objects.



CAUTION: The contents of the *Type* and *Definition* fields of the *Generic object property* panel is used as direct input to the dynamics solver, and may thereby change the computed response. This feature should only be used with guidance from Fedem support personnel.

F.10 Prescribed Triad motion

A predefined prescribed motion may be imposed directly on Triad degrees of freedom. A generic database object (see [Appendix F.9, "Generic database objects"](#)) may be used to define such motions in terms of a Function and/or a constant value. The Type and Definition fields of the Generic object property panel then need to contain the following:

Type *MOTION*

Definition *triadId = <triadId>, IDof = <Idof>, type = 'deflection',
d0 = <d0>, d1 = 1.0, motionEngineId = <functionId>*

where <triadId> and <functionId> are the base IDs of, respectively, the Triad at which the motions should be applied, and the Function defining the time-variation of the motion. Furthermore, <Idof> is the local DOF number (range 1 to 6) of the Triad, and <d0> is a time-independent constant motion value. If the prescribed motion should not have any time-variation at all, it is sufficient to specify a non-zero <d0>, and then leave out the *d1 = 1.0* and *motionEngineId = <functionId>* fields.



NOTE: The base ID of a mechanism object is normally not visible in the Fedem UI. To see the base ID, you have to launch Fedem in debug mode (using command-line option *-debug*). Then the base ID appears in curly braces {} in the Object Browser.

F.11 Initial conditions for dynamics analysis

It is possible to specify different initial velocities for all Triad- and Joint degrees of freedom either by editing the model file (*.mm*), or by entering description field commands. These settings will override the global initial velocity defined in the Model Preferences dialog, for the affected objects. Of course, this should be used with care such that the velocity state specified this way is consistent throughout the model. If that is not the case, fictitious transients will occur in the first time steps of the analysis.

When using initial conditions, it is recommended to specify initial velocities for the joint DOFs also. The initial velocities at the slave triad of the joint will then be computed from the governing constraint equations of the joint. However, it is also possible to instead specify the initial conditions in the slave triad and let Fedem derive the corresponding joint velocities by inverting the constraints.

F.11.1 Initial velocities in the model file

The initial velocity in a triad is specified by entering the following line within the appropriate TRIAD record:

$$GL_VEL = u_x \ u_y \ u_z \ \omega_x \ \omega_y \ \omega_z;$$

where u_x , u_y , u_z , ω_x , ω_y and ω_z are the initial velocity components in the global DOF directions of the triad. For a joint, initial velocities in the joint degrees of freedom may be specified through

$$X_TRANS_JVAR_INIT_VEL = u_x;$$

$$Y_TRANS_JVAR_INIT_VEL = u_y;$$

$$Z_TRANS_JVAR_INIT_VEL = u_z;$$

$$X_ROT_JVAR_INIT_VEL = \omega_x;$$

$$Y_ROT_JVAR_INIT_VEL = \omega_y;$$

$$Z_ROT_JVAR_INIT_VEL = \omega_z;$$

where u_x , u_y , u_z , ω_x , ω_y and ω_z apply in the joint DOF directions. Of course, you should only enter initial conditions for existing DOFs in the joint, e.g. for a Revolute joint you can only specify $Z_ROT_JVAR_INIT_VEL$.

F.11.2 Initial velocities using description field commands

As an alternative to editing the model file, the following set of description field commands are available for detailed specification of initial conditions:

On Link objects:

```
#InitTransVel <ux> <uy> <uz>
```

On Triad objects:

```
#InitTransVel <ux> <uy> <uz>
```

```
#InitRotVel <omegax> <omegay> <omegaz>
```

On Joint objects:

```
#InitTXvel <ux>
```

```
#InitTYvel <uy>
```

```
#InitTZvel <uz>
```

```
#InitRXvel <omegax>
#InitRYvel <omegay>
#InitRZvel <omegaz>
```

An `#InitTransVel` specified on a link object will be applied to all triads attached to that link (including the automatically generated center of gravity triad for generic parts). This is overridden by a `#InitTransVel` command on a triad object though. The joint object commands will implicitly override the initial conditions defined on the slave triad of the joint (if any).



NOTE: If initial conditions have been defined both through traditional model file entries and description field commands, the description field commands will be used.

F.11.3 Turning initial conditions off

Sometimes during the creation of a complex model, it is useful to test the dynamics when all initial conditions are off (equal to zero). To facilitate this without having to manually remove all the defined initial conditions in the model file, the solver option `-ignoreIC` may be used instead (in the Additional Solver Options dialog, **Dynamics Solver** field, see [Section 6.2.3, "Additional solver options"](#)). Then all defined initial conditions will be ignored (assumed zero). This option also works when running the solver in batch mode, even when the `fedem_solver.fsi` file is loaded with initial triad/joint velocities.

F.12 Boundary conditions for dynamics analysis

The additional boundary conditions that are defined in the triad property panel (see [Section 4.3, "Triad properties"](#)), are normally applied only during the initial static equilibrium analysis, or also during the system eigenmode analysis. However, by specifying the following command in the triad description field:

```
#DynBC
```

Then the specified boundary conditions for that triad are also applied during the dynamics simulation (and also in the eigenvalue analysis). This is equivalent to attaching a constraining joint to that triad, which also is attached to ground. However, using `#DynBC` is a more direct (and slightly more efficient) way of applying such boundary conditions.

Index

A

- absolute integration tolerances, [6-20](#)
- active view, [2-23](#)
 - enlarge, [2-24](#)
 - reduce, [2-24](#)
 - scaling, [2-23](#)
- active window
 - controlling, [2-8](#)
- ADAMS/Car
 - using with ADAMS/Tire, [B-3](#)
- adder block, [5-4](#)
- additional boundary conditions for triads, [6-23](#)
- additional solver options, [6-6](#), [E-1](#)
- Align CS, [3-10](#)
- Align rotations, [3-10](#)
- Altair® HyperMesh®, [1-4](#)
- amplifier block, [5-4](#)
- amplifiers, [5-4](#)
- analysis options
 - management of, [6-5](#)
- animation
 - available results, [7-29](#)
 - control, [7-32](#)
 - loading performance, [7-32](#)
 - properties, [7-23](#)
- Animation Control, [7-36](#)
- Animation Control panel, [7-35](#)
- animation frames, [7-25](#), [7-29](#)
 - load fringes and deformations, [7-25](#)
 - stress recovery options, [7-25](#)
- animation speed, [7-34](#)
- animations, [7-3](#)
 - averaging options, [7-27](#)
 - close, [7-34](#)
 - cycling, [7-34](#)
 - deformation results, [7-24](#)
 - eigenmode, [7-28](#)

- eigenmodes tab, [7-24](#)
- fringes tab, [7-24](#)
- loading, [7-24](#)
- Modeler window, [7-21](#)
- Play Panel, [7-21](#)
- playing repeatedly, [7-34](#)
- showing continuous motion, [7-25](#)
- specifying averaging behavior, [7-27](#)
- speed, [7-33](#)
- time tab, [7-24](#)

anti-aliasing, [2-26](#)

ASCII format, [2-35](#), [D-2](#)

attached elements

- color, [3-21](#)
- See also* general appearance

autoscale, [7-7](#)

averaging options, [7-27](#)

axial dampers, [4-42](#)

axis rotation, [2-21](#)

B

- Back view, [2-22](#)
- Backward Euler, [6-2](#)
- ball joint, [4-28](#)
- ball movement, [3-10](#)
- batch-execution of solvers, [E-1](#)
- bdf, [2-4](#)
- .bdf format, [1-4](#), [C-1](#)
- binary format, [D-2](#)
- binary input blocks
 - adder, [5-4](#)
 - comparator, [5-4](#)
 - multiplier, [5-4](#)
- binary-input control elements, [5-4](#)
- .bmp format, [2-35](#)
- Bottom view, [2-22](#)
- Bulk Data File format, [1-4](#)
 - exporting to, [1-4](#)

See also *.bdf*

C

- CAE systems, [C-1](#)
- cam joint, [4-33](#)
 - creating, [4-33](#)
 - follower, [4-33](#)
- cam surface, [4-33](#)
- Cam thickness, [4-36](#)
- cam triads, [4-33](#)
- Centre of Gravity, [4-3](#)
- centripetal moment correction, [6-21](#)
- co-located items, [2-18](#)
- Color mapping, [7-35](#)
- colors for attached/unattached elements, [3-21](#)
- command sensitivity, [2-7](#)
- commands
 - accessing, [2-8](#)
 - delete all stickers, [3-7](#)
 - detach, [3-21](#)
 - general appearance, [2-24](#)
 - item appearance, [2-27](#)
 - modeling, [2-8](#)
 - multiple selection, [2-18](#)
 - observing with Output List, [2-16](#)
 - pan (F1), [2-20](#)
 - performing, [2-17](#)
 - print view, [2-35](#)
 - rotate (F3), [2-21](#)
 - select dynamic center (F4), [2-21](#)
 - 3D viewing, [2-22](#)
 - zoom (F2), [2-20](#)
- comparator block, [5-4](#)
- complex conjugate pole block, [5-7](#)
- Component Mode Synthesis, [6-2](#), [6-11](#)
- component modes, [6-11](#)
 - calculating, [6-11](#)
- connections
 - stiff translating, [4-31](#)
 - triad, [3-2](#), [4-18](#)
- contact, [4-36](#)
- Contact model for SWIFT-Tyre, [B-19](#)
- Contact-point C and normal load
 - for MF-Tyre, [A-7](#)
- control blocks (control elements), [5-4](#)
 - about, [5-4](#)
 - controlling output, [5-6](#)
 - creating and manipulating, [2-15](#)
- Control Creation toolbar, [2-8](#), [5-2](#)
- Control Editor, [1-3](#), [2-15](#), [5-2](#)
 - deleting blocks or connections, [5-10](#)
 - opening, [2-15](#), [5-2](#)
- Control menu, [5-2](#)
- control module
 - defining, [5-7](#)
- control modules
 - building, [5-2](#)
- control system
 - about, [5-1](#)
 - amplifiers supported, [5-4](#)
 - connection with mechanism, [5-3](#)
 - creating, [5-2](#)
 - diagram of, [2-15](#)
 - external, [5-11](#)
 - modeling, [1-3](#), [5-1](#)
- control system blocks
 - adding lines, [5-9](#)
 - defining the module, [5-9](#)
 - editing properties, [5-8](#)
 - inserting, [5-8](#)
 - moving, [5-8](#)
 - removing breaking points, [5-9](#)
 - rotating, [5-10](#)
- control toolbars, [5-2](#)
- control tools, [2-8](#)
- Control Tools toolbar, [5-3](#), [5-8](#)
- control/servo systems, [1-3](#)
- Coordinate system
 - ISO, described for tires, [A-5](#)
- coupling effects, [1-3](#)
- Ctrl key
 - using, [3-21](#)
- curves
 - exporting, [2-35](#) — [2-36](#)

modifying appearance, 7-15
cut-back, 6-18
cylindric joint, 4-32
cylindrical motion, 3-11

D

.*dac* format, 2-35
dampers
 properties, 4-41
dead-zone block, 5-6
 lower and upper limits, 5-6
 output, 5-6
Definitions
 listed for MF-Tyre, A-38
degrees of freedom (DOFs), 4-52
delay block, 5-5
delete
 all stickers, 3-7
 selecting multiple items, 2-18
deleting
 stickers and recreating, 3-7
derivator block, 5-5
description field commands, F-1
deselecting items in the Model Manager, 2-9
detach, 3-21
DOFs, 4-20, 4-52
 See also degrees of freedom
duty cycle analysis, 6-44
 license, 6-45
dynamic
 functions, 2-19 — 2-20
 pan (F1) command, 2-20
 rotate (F3) command, 2-21
 select dynamic center (F4)
 command, 2-21
 viewing, 2-19
 zoom (F2) command, 2-20
dynamic zooming and rotation, 2-21
dynamics analysis
 about, 6-2
 controlling parameters, 6-18

performing, 6-2
 unbalanced forces, 6-3
dynamics simulation, 8-9
 before beginning, 6-2
 deleting result files, 6-48
 result files, 8-9
 starting, 6-30
Dynamics Solver
 option files, 8-9
 setup, 6-18, 6-30

E

eigenmode
 specify shift factor, 6-23
Eigenmode animations, 7-24
eigenmode solutions, 1-2
 calculating, 6-23
 computed by Fedem, 6-3
eigenmodes
 calculating, 6-3
 damped, 6-23
 options, 7-28
 specify number to be computed, 6-23
element
 expressions syntax, C-5
 properties, 1-5
 topologies, 1-5
Element group properties, 4-17
element groups
 creation, 4-5, 4-16
 other identifiers, C-12
element statements
 parameters, C-4
equilibrium analysis
 conditions, 6-25
error messages
 intepretation, 6-51
example control system, 5-2
exporting
 3D modeler view, 2-36
 animations, 2-35, 2-37
 curves, 2-35

graphs, 2-35, 2-37, 7-20
links, 2-35
objects, 2-35
External Control System, 1-4
external control system, 5-11
 Simulink, 5-11
external nodes, 1-4

F

F1 (dynamic pan), 2-20
F2 (dynamic zoom), 2-20
F3 (dynamic rotate), 2-21
F4 (select dynamic center), 2-21
fatigue analysis, 6-43
Fedem, 2-7
 about, 1-2, 2-1
 binary input blocks, 5-4
 building techniques, 3-2
 commands, 2-1
 control systems, 5-2
 available control elements, 5-4
 controllers, 5-6
 definition of, 1-2
 directory structures, 2-3, 8-8
 editing environment, 5-2
 file formats, D-2
 file types, 2-3
 input, D-2
 intermediate, D-2
 results, D-2
 secondary, D-2
 file use overview, D-4
 files and directory structures, D-1
 graphical representation, 5-2
 main window, 2-6
 major user tasks, 3-3
 mechanism, 1-4
 analysis, 6-1
 elements, 3-1
 model
 definition of, 1-4
 modeling environment, 3-2

module execution, 1-5
reducer
 options, E-2
results database
 removing unwanted files, 8-9
starting, 2-4
strain rosette analysis, 1-7
technical support, 2-6
user interface, 2-1, 2-6
Fedem Link Model (.*flm*) format, C-1
Fedem Mechanism Model (.*fmm*)
format, 2-3
Fedem Technology Link (.*ftl*)
format, C-1 — C-2
 nodes, C-3
FE-Fatigue, 6-43
 license, 6-44
file types
 input files, D-2
 intermediate files, D-3
 other files, D-3
 results files, D-3
Finite Element (FE)
 dynamics in elastic mechanisms, 1-2
 material data, 1-5
 method, 1-5
 models, 1-4 — 1-5
 creating, 1-5
 generation of, C-1
 importing, 1-4
 nonlinear, 1-4
 storage, 2-3, C-1
 See also links
 nodes, 1-4
first-order transfer function block, 5-7
five-DOF joint, 4-38
flip element direction, 5-10
.flm format, 2-4, C-1
.fmm format, 2-3
follower triads
 constraining, 4-33
Force evaluation in SWIFT-Tyre, B-8
force vector, 4-47

- force vector orientation, 4-47
- forces, 4-46
- format
 - .flm, 2-4
 - .fmm, 2-3
 - .ftl, 2-3
- free joint, 4-29
- free movement, 3-10
- friction, 4-22, 4-28 — 4-29, 4-31, 4-35
 - behavior, 4-38
 - computing force, 4-38
 - creating, 4-39
 - editing, 4-39
 - parameters, 4-39
 - properties, 4-39
 - with appropriate joint, 4-39
- fringe legend, 7-35
 - customizing, 7-35
- fringe value
 - selecting results, 7-26
- From and To options, 4-47
- front view, 2-22
- .ftl files, 4-20
 - examples, C-2
 - identifiers, C-2
- .ftl format, 2-3, 2-35, C-2
- Full Color, 7-36
- Full Color B/W Limits, 7-36
- Full Color Clipped Limits, 7-37
- function
 - creating, 4-51
- function keys, 2-20
- functions
 - properties, 4-51

G

- general appearance, 2-24
- general transfer functions, 5-7
- geometric stiffness contribution (link stiffening), 6-21, 6-23 — 6-24
- geometrically nonlinear FE model
 - treatment of, 1-4

- graphic performance
 - increasing speed, 2-22, 2-26
 - modeling tools, 1-3
- graphics card settings, 2-26
- graphs
 - about, 7-3
 - abscissa label, 7-4
 - adding legend, 7-4
 - controlling x- and y-axis range, 7-7
 - creating, 7-4
 - creating curves, 7-4
 - displaying, 7-4
 - displaying curve properties, 7-7
 - dynamic updating, 6-47
 - dynamic viewing, 7-17
 - enabling Autoscale option, 7-4
 - exporting, 2-35
 - naming, 7-4
 - naming curves, 7-8
 - opening window, 2-15
 - ordinate label, 7-4
 - Possible Results list, 7-10, 7-14
 - print view, 2-35
 - printing, 7-21
 - result operation, 7-9
 - selecting results, 7-10
 - specifying properties, 7-4
 - variables, 7-10
 - viewing tips, 7-18
 - viewing values, 7-3
 - views, 2-8, 2-15, 6-47, 7-2
 - displaying legend, 7-8
 - manipulating, 7-17
 - updating curves, 6-47
 - X-Axis/Y-Axis, 7-8
- Gravitation, 3-24
- grid and snap, 2-15
 - manipulating, 5-8
- ground
 - selecting, 2-19
- group-wise solving, 6-9
- Guide bar, 2-17

H

hardware requirements, [2-2](#)
History
 of MF-Tyre, [A-11](#)
HyperMesh®, [1-4](#)
hysteresis (backlash) block, [5-6](#)
 deadband, [5-6](#)

I

ID and Topology panel, [2-7](#), [2-10](#)
 sensors, [4-65](#)
ID Number, [2-11](#)
I-DEAS®, [1-4](#)
identification numbers, [2-9](#)
importing links, [2-8](#)
initial equilibrium, [6-24](#)
 analysis, [4-19](#), [6-24](#)
Initial translational velocity, [3-24](#)
input and output blocks, [5-3](#)
input file types, [D-2](#)
input force orientation, [4-48](#)
input signal
 amplifying, [5-4](#)
 calculating and outputting power, [5-4](#)
integrator and limited derivator
 blocks, [5-5](#)
integrator block, [5-5](#)
Interactive Odometer, [3-5](#), [3-8](#)
 about, [3-5](#)
 specifying a discrete point, [3-12](#)
 using with Smart Move command, [3-5](#)
intermediate file types, [D-3](#)
isometric button, [2-21](#)
item type, [2-11](#)
iteration step size, [6-25](#)
Iteration step size limit, [6-24](#)

J

joint
 variables, [4-21](#)
joint pairs, [4-37](#)

Joint Variables, [4-20](#)
joint variables, [4-22](#)
joints
 about, [4-20](#)
 adding friction, [4-22](#)
 adding motion constraints, [4-24](#)
 attaching, [3-16](#)
 attaching multiple, [3-17](#)
 attaching to links, [4-20](#)
 attachment restrictions, [3-16](#)
 ball, [4-27](#)
 cam, [4-33](#)
 cylindric, [4-32](#)
 detaching, [3-21](#)
 free, [4-27](#)
 friction, [4-38](#)
 master and slave, [3-16](#)
 pair constraints
 rack and pinion, [4-38](#)
 point-to-path, [4-30](#)
 point-to-point, [4-27](#)
 prismatic, [4-30](#)
 revolute, [4-27](#)
 rigid, [4-27](#)
 summary table, [4-25](#)
 triad connections, [4-20](#)

K

Kinematics
 described for MF-Tyre, [A-36](#)

L

left view, [2-22](#)
License, [2-39](#)
limiter block, [5-6](#)
line view, [2-22](#)
lines
 definition, [5-10](#)
lines (item appearance), [2-28](#)
Link coordinate system, [4-3](#)
Link Database, [4-14](#)
link files

- directory, [2-3](#), [8-1](#)
- link reduction files, [6-48](#)
- link repository, [4-14](#)
- Link_DB directory, [2-3](#)
 - subdirectories, [8-9](#)
- link_DB directory, [8-9](#)
- links, [1-4](#), [3-8](#)
 - about, [4-2](#)
 - applied forces, [4-46](#)
 - as masters and slaves, [4-20](#)
 - changing appearance, [2-27](#)
 - duplicating, [4-6](#)
 - exporting, [2-35](#)
 - importing, [2-8](#), [4-3](#)
 - importing several at once, [4-4](#)
 - proportional damping, [4-8](#)
- link-wise solving, [6-9](#)
- load fringes, [7-23](#)
- loads, [4-46](#)
 - load target point, [4-47](#)
- loads and torques
 - control of magnitude, [4-46](#)
- logical-switch block, [5-5](#)
- lumped mass matrix, [6-12](#)

M

- Magic Formula
 - about MF-Tyre model, [A-11](#)
- mass and stiffness matrices, [6-2](#)
- master/slave triad, [4-20](#)
 - attaching manually, [4-20](#)
- MATLAB®/Simulink®, [1-3](#)
- MATRIXx®, [1-3](#)
- mechanism, [1-4](#)
 - 3D view of, [2-14](#)
 - accessing a response variable, [5-3](#)
 - building, [3-4](#)
 - components
 - flexibility of, [1-3](#)
 - element connections, [1-4](#)
 - introducing motion, [4-46](#)
 - modeling process, [3-1](#)

- mechanism analyses, [2-8](#)
 - speeding up, [6-2](#)
- Mechanism Creation toolbar, [3-4](#)
- mechanism elements, [3-1](#)
 - attaching to a link, [3-16](#)
 - constrained by cylindrical joint, [3-11](#)
 - constrained by prismatic joints, [3-11](#)
 - creating, [2-8](#), [3-8](#)
 - customizing, [4-1](#)
 - defining properties, [4-1](#)
 - deleting from model, [3-22](#)
 - detaching, [3-21](#)
 - fully constrained, [3-11](#)
 - limiting display of, [2-18](#)
 - links, [3-8](#)
 - measuring movement and variables, [4-64](#)
 - moving, [3-10](#)
 - viewing and editing properties, [2-12](#)
- mechanism entities
 - display manipulation, [2-25](#)
- mechanism mode shapes
 - animating and displaying, [6-33](#)
- mechanism model
 - building, [3-1](#), [3-3](#)
- mechanism symbols, [2-25](#)
 - manipulating appearance, [2-24](#)
- Mechanism Tools toolbar, [3-4](#)
- menus and toolbars, [2-7](#)
- MF-Datasets
 - described for MF-Tyre, [A-36](#)
- MF-Tool, described, [A-36](#)
- MF-Tyre
 - about Magic Formula, [A-11](#)
 - about tire model, [A-2](#)
 - axis systems in, [A-5](#)
 - contact-point C and normal load, [A-7](#)
 - described, [A-2](#)
 - formula for gyroscopic couple, [A-31](#)
 - history, [A-11](#)
 - learning basics, [A-13](#)
 - lists of definitions, [A-38](#)
 - references, [A-40](#)

- rolling resistance formula, [A-33](#)
- slip formulas in, [A-18](#)
- steady-state formulas, [A-18](#)
- switching from simple to complex models, [A-32](#)
- tire kinematics, [A-36](#)
- tire relaxation length, [A-30](#)
- tire rolling radius, [A-8](#)
- tire slip quantities, [A-11](#)
- tire-road interaction, [A-3](#)
- transient behavior formulae, [A-30](#)
- units, [A-6](#)
- modal analysis, [6-3](#)
- mode and time lists, [6-34](#)
- mode shape analysis, [6-4](#)
 - setting up, [6-33](#)
 - specifying parameters, [6-4](#)
- mode shape recovery, [6-34](#)
 - eigenmodes animations, [7-28](#)
 - options, [6-34](#)
- mode shapes
 - animating, [6-33](#) — [6-34](#)
 - expanding, [6-4](#)
- model
 - applying motion constraints, [3-1](#)
 - reduction process, [1-4](#)
 - storing contents, [2-3](#)
 - visualizing, [2-19](#)
- model display
 - color, [2-27](#)
 - complexity level, [2-27](#)
 - fog, [2-26](#)
 - transparency, [2-27](#)
- model file
 - creating new, [2-5](#)
 - directory, [2-3](#)
- Model Manager
 - selecting items, [2-9](#)
- Model Manager Objects list, [3-7](#)
- Model Manager panel, [2-7](#), [2-9](#)
- Model Manager Results list, [7-2](#)
- Model Manager tabs, [3-2](#)
- model objects
 - editing properties, [2-7](#)
 - managing, [2-7](#), [2-9](#)
- model reduction, [6-2](#), [6-11](#)
 - initiating manually, [6-11](#)
- Modeler window, [2-14](#), [3-3](#)
 - Global Directions, [2-14](#)
 - Interactive Odometer, [3-8](#)
 - locating specific points, [3-5](#)
 - manipulating the view, [2-8](#)
 - opening, [3-3](#)
 - Reference Plane, [2-14](#)
- modeling
 - aids
 - stickers, [3-6](#)
 - commands, [2-8](#)
 - control systems, [1-3](#)
 - objects
 - triads, [4-18](#)
 - programs, [1-4](#)
 - tools, [1-3](#), [3-4](#)
- Modeling tolerance, [3-24](#)
- modeling tools, [3-3](#)
- models, [2-26](#)
- models and results
 - constructing and viewing, [2-7](#)
- modes
 - postprocessing, [6-34](#)
- modified Newton-Raphson iteration, [6-20](#)
- motion
 - constraints, [2-8](#), [3-10](#)
- mouse
 - using, [2-17](#)
- Mouse button, [4-47](#)
- movability, [3-4](#), [3-6](#)
- movability, [3-2](#)
- move curves, [7-6](#)
- MSC/Patran®, [1-4](#)
- MSC.Nastran®
 - Bulk Data File (.*bdf*) format, [1-4](#)
- multibody systems
 - applications, [1-3](#)
- multiple face results, [7-27](#)

multiplier block, [5-4](#)

N

.nas format, [2-4](#), [C-1](#)

Nastran Bulk Data (*.nas* or *.bdf*)
format, [C-1](#)

navigating, [2-13](#)

nCode DAC (*.dac*) format, [2-35](#)

Negative pivots, [6-14](#)

Newmark time-integration, [6-2](#)

Newton integration algorithm, [6-18](#)

Newton-Raphson equilibrium
iterations, [6-2](#)

nodal

attributes, [1-5](#)

coordinates, [1-5](#)

nodes

syntax for, [C-3](#)

nonlinear equations

solution of, [6-20](#)

Normal load

and contact-point C for MF-Tyre, [A-7](#)

Notation for SWIFT-Tyre, [B-3](#)

O

object movability

determining, [3-10](#)

object selection, [2-17](#)

confirming, [2-19](#)

ground, [2-19](#)

restrictions, [2-18](#)

objects

related, [2-11](#)

selection history, [2-18](#)

Objects list

functions, [4-51](#)

Objects tab, [2-10](#)

option files

storage, [8-9](#)

Origin property, [3-10](#), [3-13](#)

output block

definition, [5-3](#)

Output List

attachment process, [3-16](#)

window, [2-16](#)

opening, [2-16](#)

P

pan (F1), [2-20](#)

pan down, [2-24](#)

pan left, [2-24](#)

pan right, [2-24](#)

pan up, [2-24](#)

panels

hiding, [2-14](#)

parallel projection, [2-23](#)

Parameters

for SWIFT-Tyre, [B-22](#)

perform, [1-5](#)

perspective view, [2-23](#)

PI, PD, and PID controllers, [5-6](#)

PID control block, [5-9](#)

pin joints, [4-26](#)

Play Panel, [7-33](#)

point rotation, [2-21](#)

point-force vectors, [4-46](#)

point-to-path joints, [4-30](#), [4-32](#)

point-to-point joints, [4-27](#)

polygons (item appearance), [2-27](#)

postprocessing

animating options, [7-2](#)

capabilities, [1-2](#)

definition, [7-2](#)

graphing options, [7-2](#)

modes, [6-34](#)

power block, [5-4](#)

Print View button, [2-35](#)

prismatic and cylindric joints, [4-31](#)

improving load distribution, [4-31](#)

prismatic joints, [4-31](#)

adding friction, [4-31](#)

Pro/ENGINEER®, [1-4](#)

processing results

deleting, [6-48](#)

- storing, [6-48](#)
- program modules, [1-5](#)
 - strain rosette analysis, [1-7](#)
- properties, [2-12](#)
- Property Editor panel, [2-7](#), [2-12](#)
- Property menus, [2-13](#)

R

- rack-and-pinion, [4-38](#)
 - transmission ratio, [4-38](#)
- Radial contact, [4-36](#)
- rainflow analysis, [6-42](#)
- real-pole block, [5-7](#)
- recovery operations
 - deleting result files, [6-48](#)
- Red Blue, [7-37](#)
- reducer
 - options, [E-2](#)
- Reference, [2-27](#)
- Reference Plane, [3-4](#)
 - changing appearance, [2-27](#), [3-4](#)
- References for MF-Tyre, [A-40](#)
- relations, [2-11](#)
- relative sensor, [4-65](#)
- remove breakpoint, [5-10](#)
- resolution settings, [2-2](#), [2-34](#)
- result directories, [8-10](#)
- result set
 - by name, [7-26](#)
 - by operation, [7-26](#)
- results file types, [D-3](#)
- results files, [8-9](#)
- Results list
 - managing, [7-2](#)
 - shortcut menus, [7-2](#)
- Results tab, [2-10](#)
- revolute joint, [4-27](#)
- .rgb* format, [2-35](#)
- Right view, [2-22](#)
- rigid body animation, [6-33](#) — [6-34](#), [6-47](#)
- rigid joint, [4-28](#)
- rigidity symbol, [3-11](#)

- Road, [4-49](#), [4-53](#)
- Road input for SWIFT-Tyre, [B-14](#)
- Road property files
 - example of for SWIFT-Tyre, [B-34](#)
- Rolling resistance
 - torque, formula for MF-Tyre, [A-33](#)
- rotate (F3), [2-21](#)
- rotating
 - about a point, [2-21](#)
 - about an axis, [2-21](#)
 - selecting new point, [2-21](#)
- rubber bushings, [4-26](#)
- Run... option, [E-1](#)
- Runge-Kutta method (Lobatto IIC), [6-2](#)

S

- sample-and-hold block, [5-5](#)
- Save As... command, [6-48](#)
- screw ratio, [4-32](#)
- second-order accuracy
 - achieving, [6-2](#)
- second-order transfer function
 - block, [5-7](#)
- Select, [2-21](#)
- select, [2-13](#)
- select dynamic center (F4), [2-21](#)
- selecting items in the Model Manager, [2-9](#)
- selection, [2-18](#)
- selection filter, [2-18](#)
- selection history, [2-18](#)
- sensors, [4-64](#)
 - managing, [4-65](#)
 - processing data, [4-64](#)
 - simple, [4-65](#)
- Show All Frames forcing option, [7-34](#)
- Simplified visualization, [4-3](#)
- simulating nonlinear behavior, [5-5](#)
- simulation
 - graphed results, [7-3](#)
 - managing results, [7-2](#)
 - progress indication, [6-47](#)

- simulation results
 - dependent on, [1-5](#)
- Simulink, [5-11](#)
- singularities, [6-12](#)
- Singularity criterion, [6-14](#)
- slave triad (follower), [4-33](#)
- Slip
 - formulas for MF-Tyre, [A-18](#)
- Smart Move, [3-5](#), [3-8](#), [3-10](#)
 - from-point and to-point, [3-11](#)
 - motion constraints, [3-6](#)
 - performing, [3-11](#), [3-13](#)
 - selecting multiple items, [2-18](#)
 - stickers, [3-6](#)
 - types of motion allowed, [3-10](#)
 - using, [3-6](#)
- S-N curve, [4-17](#)
- snap, [2-17](#)
- solid view, [2-22](#)
- solution processes
 - stopping, [6-48](#)
- solvers
 - command-line options, [E-1](#)
 - running in batch mode, [8-9](#)
 - setting up parameters, [6-5](#)
- Solvers toolbar
 - about, [6-5](#)
 - accessing commands, [6-5](#)
- speed slider, [7-33](#)
- spring inter-connectivity, [4-25](#)
- springs
 - assigning values, [4-21](#)
 - changing stress-free length during simulation, [4-40](#)
 - properties, [4-40](#)
- Standard tire interface (STI), defined, [A-35](#)
- static equilibrium analysis
 - about, [6-2](#)
 - setting up, [6-3](#)
- Status bar, [2-7](#)
- STI (standard tire interface), defined, [A-35](#)
- stickers, [3-6](#), [3-8](#)
 - applying manually, [3-6](#)
 - creating manually, [3-7](#)
 - deleting, [3-7](#)
- storage, [2-3](#)
 - eigenvalues from dynamics simulation, [8-10](#)
- Fedem Link Model (*.flm*) format, [2-3](#)
- Fedem Mechanism Model (*.fmm*) format, [2-3](#)
- Fedem Technology Link (*.ftl*) format, [2-3](#)
- link files, [2-3](#)
- MSC.Nastran Bulk Data File (*.bdf* or *.nas*) format, [2-3](#)
- primary time history result files, [8-10](#)
- results from eigenvalue recovery, [8-10](#)
- results from stress recovery, [8-10](#)
- simulation results, [2-3](#)
- strain coat analysis, [6-40](#)
 - options, [6-41](#)
 - recovery, [6-43](#)
- strain coat elements, [6-40](#)
- strain rosette analysis, [1-7](#), [6-35](#)
 - options, [6-35](#)
 - result files, [6-36](#)
- strain rosette definition, [6-38](#)
- stress analysis, [1-2](#)
 - improving performance, [6-31](#)
 - specifying parameters, [6-4](#), [6-31](#)
- Stress concentration factor, [4-17](#)
- stress recovery
 - analysis, [6-4](#)
 - starting, [6-33](#)
- superelements, [1-4](#), [6-2](#)
- SWIFT-Tyre
 - about, [B-2](#)
 - contact model, [B-19](#)
 - example road property file, [B-34](#)
 - example tire property file, [B-28](#)
 - force evaluation, [B-8](#)
 - notation for, [B-3](#)
 - parameters, [B-22](#)

- road input, [B-14](#)
- switching from simple to complex models, [B-21](#)
- tire rolling radius, [B-14](#)
- symbols
 - line-smoothing, [2-26](#)
- system requirements, [2-1](#) — [2-2](#)
 - See also* hardware requirements
- system resources, [6-47](#)

T

- target point, [4-47](#)
- 3D
 - animation, [6-47](#)
 - modeling
 - stickers, [3-10](#)
 - point marker, [3-5](#)
 - viewing, [2-14](#)
 - dynamic updating, [6-47](#)
- 3D View Control toolbar (View menu), [2-22](#)
- 3D viewing, [2-25](#)
 - commands, [2-22](#)
- time history
 - changing interval, [7-25](#)
- Time History Animations, [7-24](#)
- time integration
 - optimizing numerical performance, [6-20](#)
- time step iterations, [6-47](#)
- Time summary animations, [7-24](#)
- time window, [7-25](#)
- time-dependent control blocks
 - delay, [5-5](#)
 - sample-and hold, [5-5](#)
- tire data file, [4-49](#)
- Tire model, [4-49](#), [4-53](#)
- Tire property files
 - example for SWIFT-Tyre, [B-28](#)
- Tire relaxation length, described for MF-Tyre, [A-30](#)
- Tire rolling radius

- in MF-Tyre, [A-8](#)
- in SWIFT-Tyre, [B-14](#)
- Tire slip quantities in MF-Tyre, [A-11](#)
- Tire-road interaction
 - in MF-Tyre, [A-3](#)
- toolbars, [2-8](#)
 - 3D View Control, [2-7](#)
 - Control Creation, [2-8](#), [5-2](#)
 - Control Tools, [2-8](#), [5-3](#)
 - managing, [2-8](#)
 - mechanism tools, [2-7](#)
 - Solvers, [2-7](#)
 - standard, [2-7](#)
 - Windows, [2-7](#)
 - Zoom and Pan, [2-7](#)
- top view, [2-22](#)
- Topology List, [2-11](#)
- torques, [4-46](#)
- tpf-file, [4-49](#)
- translational motion
 - restraining, [3-6](#)
- triads, [3-2](#), [4-2](#), [4-18](#)
 - about, [4-18](#)
 - adding mass and mass inertias, [4-19](#)
 - additional boundary conditions, [4-19](#)
 - attachment restrictions, [3-16](#)
 - color representations, [4-18](#)
 - connections, [3-2](#), [4-18](#)
 - constrained DOFs, [4-20](#)
 - coordinate system, [4-19](#)
 - editing properties, [4-19](#)
 - FE node, [4-19](#)
 - follower, [4-33](#)
 - in joints, [3-16](#), [4-18](#)
 - master and slave, [4-20](#)
 - purposes, [4-19](#)
 - restraining movement, [4-19](#)
 - symbols, [4-18](#)
- triangularization, [6-12](#)

U

- unattached elements

- color, [3-21](#)
 - See also* general appearance
- undo option, [3-7](#)
- Units, [3-24](#)
 - in MF-Tyre, [A-6](#)
- user interface, [2-1](#)

V

- variables
 - functions of, [5-3](#)
- view
 - achieving maximum zoom, [2-20](#)
 - axis rotation, [2-21](#)
 - bottom, [2-21](#)
 - flat colors, [2-22](#)
 - front, [2-21](#)
 - isometric, [2-21](#)
 - magnifying, [2-21](#)
 - manipulating, [2-20](#)
 - parallel, [2-23](#)
 - perspective, [2-23](#)
 - point rotation, [2-21](#)
 - rotate, [2-21](#)
 - rotating, [2-8](#)
 - selecting dynamic center, [2-21](#)
 - show top faces, [2-22](#)
 - side, [2-21](#)

- solid/shaded objects, [2-22](#)
 - top, [2-21](#)
- View button, [4-47](#)
- viewer options, [2-26](#)
- viewing capabilities, [2-14](#)
- views
 - animations, [7-2](#)
 - graphs, [6-47](#), [7-2](#)

W

- Workspace, [2-7](#), [2-13](#)
 - displaying results, [7-2](#)
 - graph views, [2-16](#)
 - managing windows in, [2-13](#)
 - printing active graph view, [2-35](#)

Z

- Z-offset, [4-49](#)
- zoom (F2), [2-20](#)
- zoom all, [2-23](#)
- Zoom and Pan toolbar (View menu), [2-23](#), [7-17](#)
- zoom in, [2-24](#)
- zoom out, [2-24](#)
- zoom window, [2-23](#)

