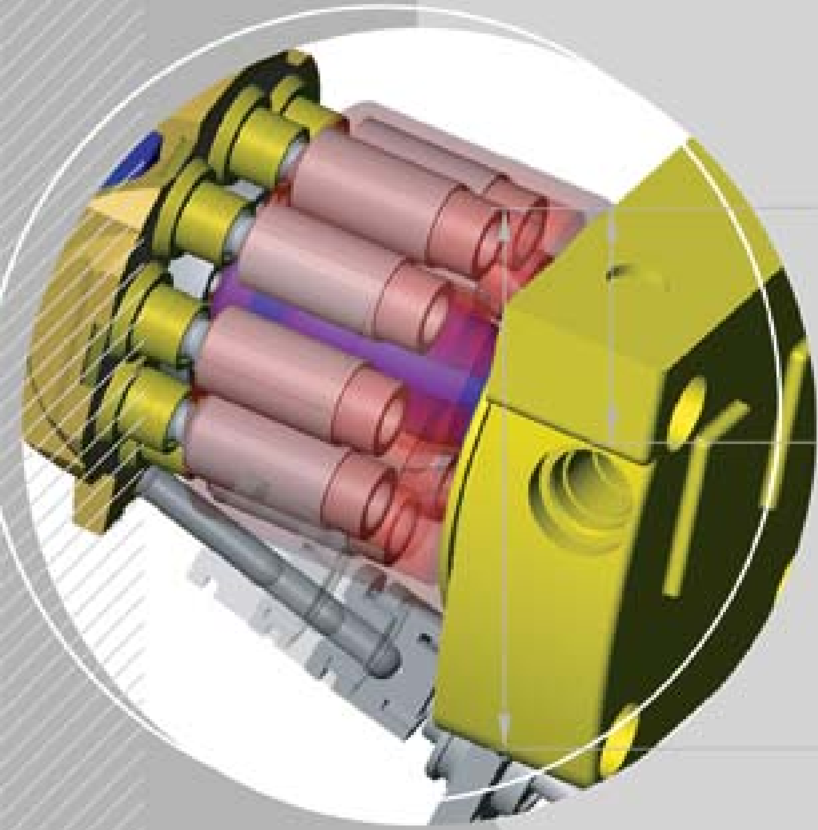
2004

COSMOS™

DESIGN ANALYSIS MADE SIMPLE

User's Guide



SolidWorks

COSMOSMotion User's Guide

COPYRIGHT NOTICE

REVISION HISTORY

First Printing December 2001
Second Printing September 2002
Third Printing August 2003

TRADEMARKS

Table of Contents

Table of Contents i

1 **COSMOSMotion 1**

Why are Mechanisms Important? 1
Benefits of Using COSMOSMotion 2
Product Structure 2
User Interface 3
Steps in Defining and Simulating a Mechanism 6

2 **Creating Mechanisms 11**

Modeling Procedure 11
Automatically Create Parts and Joints 12
Motion Parts 14
Rigidly Attached Parts 16
Constraints 17
Revolute Joint 19
Translational Joint 20
Cylindrical Joint 21
Spherical Joint 22
Universal Joint 23
Screw Joint 24
Planar Joint 25
Fixed Joint 26
Joint Friction 27
Understanding Joint Primitives 35
Inline JPrim 36
Inplane JPrim 37
Orientation JPrim 38
Parallel Axes JPrim 39
Perpendicular JPrim 40
Understanding Motions 41
Motion Expression 42
Creating Joints, Joint Primitives, and Motions 48
Understanding Contact Constraints 55
Creating Contact Constraints 60
Creating 3D Contacts 69
Joint Couplers 73
Motion on Parts 75
Rigid Bodies 78
Forces 81
Creating Applied Forces 87
Creating Bushings 93
Creating Springs and Dampers 95
Creating Impact Forces 103

Gravity 107
Manipulating Mechanism Entities....................108

3 **Materials 109**

Adding Materials 110
Editing Materials 114

4 **Mechanism Solution.................... 117**

Simulation Panel 117
Simulating 125
Simulation Troubleshooting 125

5 **Reviewing Your Results.................... 127**

Animating The Mechanism 127
Exporting an AVI movie 131
Exporting Animations to VRML 133
Exporting Results to Excel 134
Exporting Results to a Text File 138
Interference Detection....................139
Exporting to FEA....................141
Creating Trace Paths 144
Creating Linear Displacements 145
Creating Angular Displacements....................146
Creating Velocity Vectors....................147
Creating Acceleration Vectors....................148
Creating Reaction Forces and Moments....................149
Exporting Result Object Values 149

6 **XY Plotting.................... 151**

Plot Defaults....................151
Creating Plots 162
Adding Values to Plots....................164
Other XY Plot Capabilities 164
Plot Persistence....................165

7 **IntelliMotion Builder 167**

IntelliMotion Builder Units Page 167
IntelliMotion Builder Gravity Page....................169
IntelliMotion Builder Part Page....................170
IntelliMotion Builder Joint Page....................174
IntelliMotion Builder Springs Page 175
IntelliMotion Builder Motion Page....................176
IntelliMotion Builder Simulation Page....................177
IntelliMotion Builder Interference Page 179
IntelliMotion Builder VRML Page 180

8 **Interfacing to MSC.ADAMS.................... 181**

MSC.ADAMS Dataset File 181
Exporting Your Model to MSC.ADAMS....................181

9 **IntelliMotion Browser 183**

Activating the Browser 184
Detailed Browser Documentation 184

10 **MSC.ADAMS Functions 185**

Function Expression Basics 185
ABS 192
ACCM 193
ACCX 194
ACCY 195
ACCZ 196
ACOS 197
AINT 198
ANINT 199
ASIN 200
ATAN 201
ATAN2 202
AX 203
AY 204
AZ 205
BISTOP 206
CHEBY 208
COS 210
COSH 211
DIM 212
DM 213
DTOR 214
DX 215
DY 216
DZ 217
EXP 218
EXP 218
FM 219
FORCOS 220
FORSIN 222
FX 224
FY 225
FZ 226
IF 227
IMPACT 228
LOG 230
LOG10 231
MAX 232
MIN 233
MOD 234
MOTION 235
PHI 236
PI 237
PITCH 238

POLY 239
PSI 241
ROLL 242
RTOD 243
SHF 244
SIGN 245
SIN 246
SINH 247
SQRT 248
STEP 249
STEP5 251
TAN 252
TANH 253
THETA 254
TIME 255
TM 256
TX 257
TY 258
TZ 259
VM 260
VR 261
VX 262
VY 263
VZ 264
WDTM 265
WDTX 266
WDTY 267
WDTZ 268
WM 269
WX 270
WY 271
WZ 272
YAW 273

Index 275

1

COSMOSMotion

COSMOSMotion is design software for mechanical system simulation. Embedded in the SolidWorks interface, it enables engineers to model 3D mechanical systems as “virtual prototypes”.

This chapter provides an overview of the following topics:

❑ Benefits of Using COSMOSMotion

❑ Installing COSMOSMotion

❑ User Interface

❑ Defining and simulating a Mechanism

❑ MSC.ADAMS Terms

Why are Mechanisms Important?

Many of the products that we use contain moving assemblies of components (mechanisms). Mechanisms play a crucial role in the performance of such products. Examples of how mechanisms enable and improve mechanical products are provided in the table below:

General Machinery	• Cable and pulley systems that increase load capacities • Material handling systems that increase production rates
Electro-Mechanical	• Tape-loading mechanisms for VCR’s that reduce jamming • Paper-handling mechanisms that increase the throughput of photocopiers
Automotive	• Suspension designs that improve handling and reduce tire wear • Window drop mechanisms that operate smoothly
Aerospace	• Wing flaps and other control surfaces that require less power to Control • Landing gear that stow tightly within the fuselage
Off-Highway	• Backhoe linkages that dig more quickly • Loaders that are stable when working on the side of a hill
Entertainment	• Amusement park rides that safely recreate exciting motions

Benefits of Using COSMOSMotion

COSMOSMotion enables you to:

□ Have confidence that your assembly will perform as expected without parts colliding while the assembly moves.

□ Increase the efficiency of your mechanical design process by providing mechanical system simulation capability within the familiar SolidWorks environment. Defining the motion of the mechanism, simulating it, and animating the results can be performed without learning a new interface.

□ Remain within a single engineering model, eliminating the need to transfer geometry and other data from application to application.

□ Eliminate the expense caused by design changes late in the manufacturing process. COSMOSMotion speeds the design process by reducing costly design change iterations. It enables you to design and simulate moving assemblies so that you can find and correct design mistakes before building physical prototypes. It also calculates loads that can be used to define load cases for structural analysis.

Product Structure

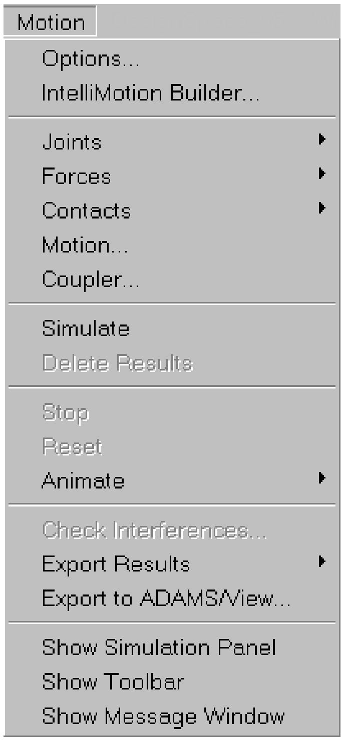
This manual is intended to describe the operation of both the COSMOSMotion product.

User Interface

COSMOSMotion fits neatly into the SolidWorks interface. This section describes the changes to the SolidWorks user interface.

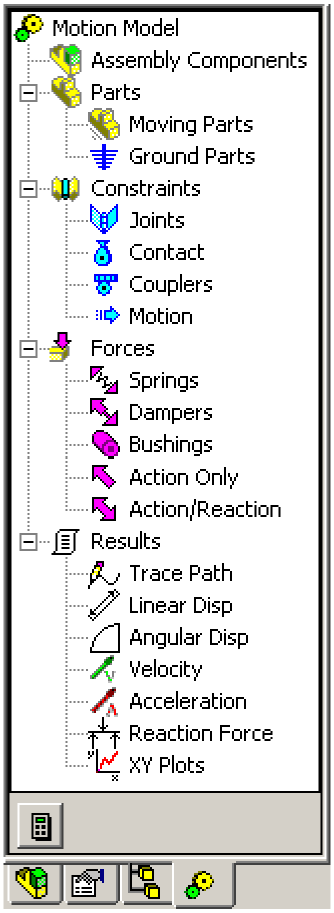
Motion Menu

COSMOSMotion adds a new menu to the SolidWorks main menu for assembly documents. The Motion menu contains all of the tools needed to build, simulate, and animate a SolidWorks assembly.



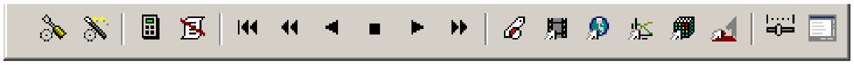
IntelliMotion Browser

When COSMOSMotion is active, a new feature tree, called the IntelliMotion Browser, appears on the left side of the screen. This browser provides you with a graphical, hierarchical view of your motion model and allows you to access all of COSMOSMotion's functionality through a combination of drag and drop and right mouse button activated pop-up menus.



Motion Toolbar

The Motion Toolbar provides single button click access to simulation and most post processing commands in COSMOSMotion.



Options – Display options dialog window

IntelliMotion Builder – Activate the IntelliMotion Builder dialog.

Simulate Model – Run the simulation

Delete Results – Delete the simulation results
Reset to start position – Reset model to original starting position

Fast backward replay - Replay Motion in reverse direction at fast increment

Backward replay - Replay Motion in reverse direction

Stop replay – Stop Motion replay

Forward replay – Replay Motion frame by frame

Fast Forward replay – Replay motion at fast increment

Find Interference – Check bodies for interference over range of simulation

Export to AVI file – Export animation to an AVI movie format

Export to VRML – Export animation to VRML file

Export to FEA – Export motion based loads for stress analysis

Export to MSC.ADAMS – Export mechanism and geometry to MSC.ADAMS products

Show Simulation Panel – Display Simulation panel for replaying motion

Show Message Window - Display messages from simulation

For further information, refer to the specific areas in the manual on how to use these features.

Steps in Defining and Simulating a Mechanism

To create a mechanism, you first indicate to COSMOSMotion which of the components in your assembly participate in the motion model. You do this by dragging and dropping the components in the IntelliMotion Browser. Any assembly mates that exist between the components are automatically converted to COSMOSMotion joints. You can then add other motion specific elements to your motion model resulting in a completely defined mechanism. You then submit the mechanism to the embedded MSC.ADAMS simulation engine, so it can determine how the mechanism will perform and behave. You can view the results of the simulation as an animation showing the motion of your mechanism or as numeric output. The steps in defining and analyzing a mechanism are explained below.

1. Review your product concept -

   Identify the components of interest, how they are connected, and what drives the movement of the components. Determine which characteristics of the product you want to understand by running a system-level simulation.

2. Indicate which components from your assembly will participate in the motion model. -

   Using drag and drop techniques in the IntelliMotion Browser, indicate which assembly components are moving parts, which are ground parts, and which components may be rigidly attached to other components. COSMOSMotion will create motion joints from an assembly mates that exist between components in the motion model. Additionally, you may manually add motion joints when a suitable assembly mate does not exist.

3. Apply motion to the constraints in your mechanism -

   You can attach motion inputs to a joint’s free degrees of freedom. A motion can input either rotational or translational motion as a function of time. For example, the function, TIME * 360d, defines a motion driver that rotates one body one complete revolution (360 degrees) with respect to another body per unit of time.

4. Add applied loads (optional, COSMOSMotion only) -

   Applied loads are external forces and torques that act on your mechanism.

5. Run a simulation of the mechanism -

   With a click of a button, you invoke the embedded simulation engine, the MSC.ADAMS/Solver that solves the equations of motion for your mechanism.

The solver calculates the displacement, velocity, acceleration, and reaction forces acting on each moving part in the mechanism.

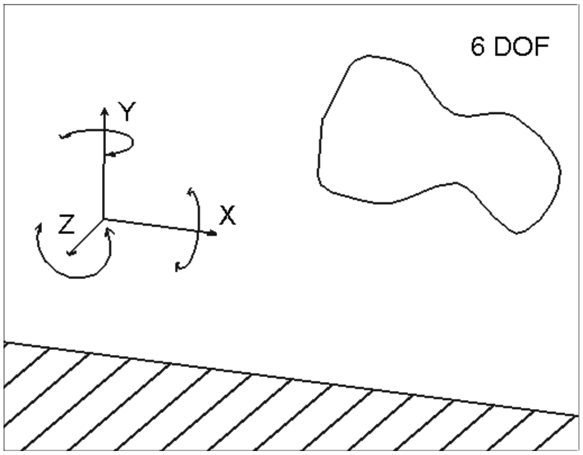
6 Review the simulation results -

You can view an animation of the simulation. Animations help you understand the behavior of your mechanism and help you communicate that information to others.

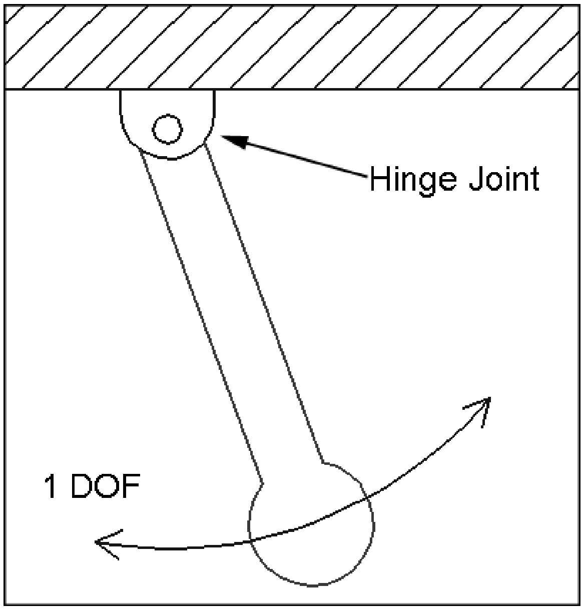
You can also view the numeric output from the simulation to understand various characteristics of your mechanism. For example, COSMOSMotion reports the loads for each joint and motion. Joint loads can be used to set up load cases for the structural analysis of any component in your mechanism.

About Degrees of Freedom

A rigid body free in space has six degrees of freedom: three translational and three rotational. It can move along its X, Y, and Z axes and rotate about its X, Y, and Z axes. When you add a constraint, such as a hinge joint, between two rigid bodies, you remove degrees of freedom between the bodies, causing them to remain positioned with respect to one another regardless of any motion or force in the mechanism. The constraints in COSMOSMotion remove various numbers and combinations of degrees of freedom.



For example, a hinge joint removes all three translational degrees of freedom and two of the rotational degrees of freedom between two rigid bodies. If each rigid body had a point on the joint that was on the center line of the hinge pin, then the two points would always remain coincident. They would only rotate with respect to one another about one axis: the center line of the hinge. The hinge joint is a single degree-of-freedom joint because it allows a single rotation between the rigid bodies.



A cylindrical joint, on the other hand, is a two degree-of-freedom joint. You can create a cylindrical joint from the hinge joint by adding a translational degree of freedom along the axis of rotation of the hinge joint. There are also three degree-of-freedom joints, such as ball joints, which constrain all translations but allow rotations about all three axes.

When you submit your mechanism to the MSC.ADAMS/Solver for simulation, the solver calculates the number of degrees of freedom in your mechanism as it determines the algebraic equations of motion to be solved in your mechanism.

When a mechanism has a closed loop, such as in a four bar linkage, there may be redundant constraints. There are three redundant constraints in a four bar linkage when all of the joints are defined as hinge joints. This is because each side of the loop (starting from ground) constrains the connecting rod to stay in the plane of the mechanism.

The MSC.ADAMS/Solver attempts to resolve the redundant constraints automatically, and can do so easily for a four-bar linkage. For more complex closed loop linkages, it is suggested that the connecting part that closes the loop be attached with a ball joint on one end and a universal joint on the other end.

2

Creating Mechanisms

With COSMOSMotion, you can use your assemblies to define working prototypes of your product concept. Joints and forces can be quickly and easily added to your solid model. This chapter describes MSC.ADAMS entities and how you create them with COSMOSMotion.

This chapter covers the following topics:

- ❑ Mechanism Modeling Procedure
- ❑ Motion Parts
- ❑ Constraints
- ❑ Motion Drivers
- ❑ Rigid Bodies
- ❑ Forces
- ❑ Manipulating mechanism entities

Modeling Procedure

The second step in simulating your mechanical system involves defining the assembly and creating the mechanism. See “Steps in Defining and Simulating a Mechanism” on page 5 for the procedure on how to simulate your mechanical system. To create a mechanism, use the following procedure. The mechanism entities discussed in the procedure below are covered in more detail later in the chapter.

To create a mechanism:

1 Indicate which components from your SolidWorks assembly will participate in the motion model by using drag and drop in the IntelliMotion Browser.

2 Define any additional joints in your mechanical system by selecting the appropriate joint from the Joint menu, opening the Insert Joint dialog box, and then selecting the SolidWorks components that you wish to constrain.

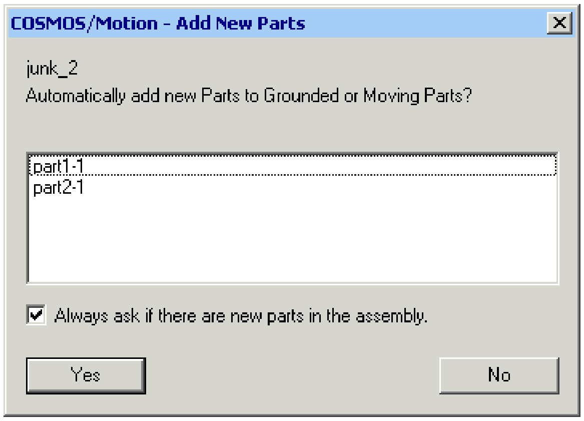
3 Define motion drivers to drive joints in your mechanical system. Not all joints will have motions applied to them.

4 Define any gravitational forces, springs, dampers, or other loads acting on your mechanism. (optional step)

Automatically Create Parts and Joints

An optional way to create your mechanism is to let COSMOSMotion do most of the work for you as you build your SolidWorks assembly model.

The first time you do a rebuild or add a SolidWorks assembly constraint (which calls a rebuild) after a inserting a new component to your SolidWorks assembly model, the following dialog box will appear:



COSMOS/Motion - Add New Parts

junk_2

Automatically add new Parts to Grounded or Moving Parts?

part1-1

part2-1

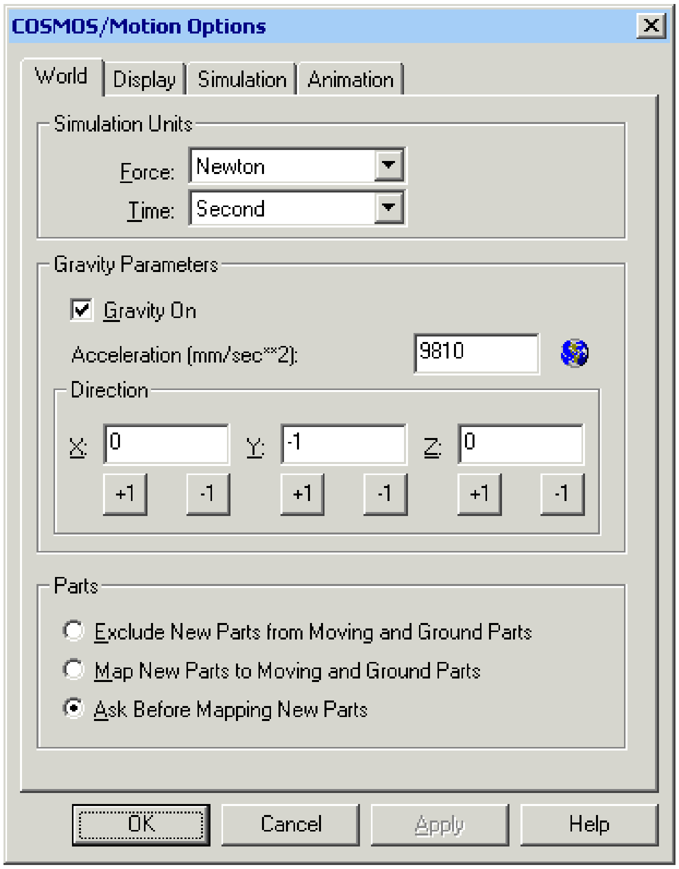
☑ Always ask if there are new parts in the assembly.

If you click **Yes** the new part(s) will be added to the motion model according to the setting of its Fix/Float attribute. If the attribute is set to float, the part will be added as a moving part, if the attribute is set to fixed, the part will be added as a ground part.

Clicking **No** will not add this part to the COSMOSMotion model.

If the checkbox labeled **Always ask if there are new parts in the assembly** is checked, then this option will be presented each time a new component is detected after a rebuild. Clearing this checkbox will disable this feature.

This feature can be re-enabled using the **Motion Options** menu selection, which displays the following dialog:



COSMOS/Motion Options

Simulation Units

Force: Newton

Time: Second

Gravity Parameters

☑ Gravity On

Acceleration (mm/sec**2): 9810

Direction

X: 0 Y: -1 Z: 0

Parts

○ Exclude New Parts from Moving and Ground Parts

○ Map New Parts to Moving and Ground Parts

● Ask Before Mapping New Parts

The options to control the automatic mapping of parts are listed in the box labeled **Parts**. The options are:

Exclude new parts from Moving and Ground parts – Automatic mapping is turned off and you will have to manually designate which parts are moving and ground when you enter the COSMOSMotion environment.

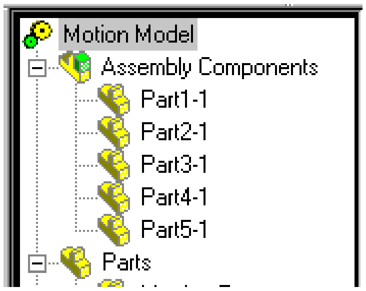
Map new parts to Moving and Ground parts – Every new part added to the SolidWorks assembly is automatically mapped to a COSMOSMotion part. The choice of moving or ground part is determined by the setting of the part's fixed/float attribute in SolidWorks.

Ask before mapping parts - The dialog box described above will be displayed each time the first constraint is added to a new part in SolidWorks.

Motion Parts

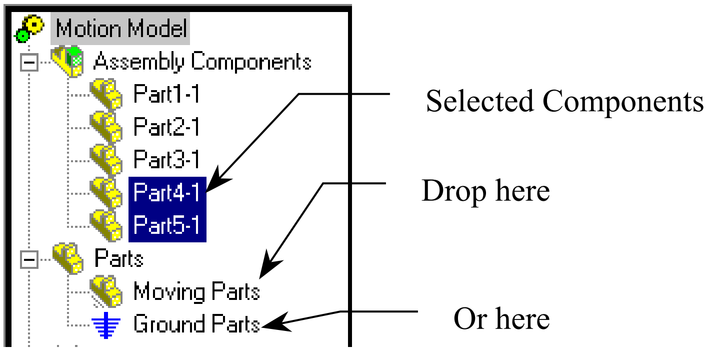
The first step in creating a mechanism is to indicate which components from your SolidWorks assembly model will participate in the motion model. This is accomplished by using either drag-and-drop, or right mouse button activated pop-up menus from the IntelliMotion Browser.

You activate COSMOSMotion by clicking on the in the SolidWorks feature manager. When COSMOSMotion is active, the SolidWorks feature manager displays the IntelliMotion Browser. In an assembly that does not yet have any motion parts defined, all of the SolidWorks assembly components will be listed under the **Assembly Components** branch in the Browser like is shown below:



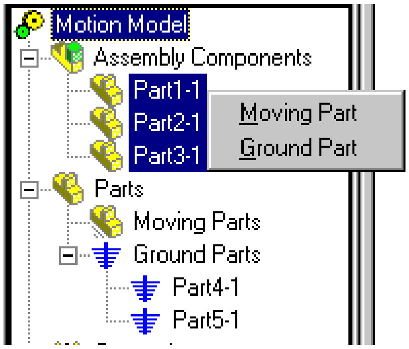
Any component that is listed under the **Assembly Components** branch of the Browser does not participate in the motion model. To add a component to the motion model, select one or more of the components listed under the **Assembly Components** branch. You can either select a single component by clicking on it, select multiple components by holding the CTRL key down and picking each component, by selecting one components, and the holding the SHIFT key down and selecting another component. All of the components between the first and second selected components will be selected. You can also "drag select" by depressing the left mouse button and moving the mouse so that the selection rectangle intersects the components. Any components with the selection rectangle will be selected.

Once you have selected one or more components, you can drag them, by holding the left mouse button down, and moving the mouse. Drag the selected components until the mouse cursor is over either the **Moving Parts** or the **Ground Parts** branch of the Browser and then drop them on that branch by releasing the mouse button.



If you drop the components on the **Moving Parts** branch, the components are added to the motion model as motion parts that can move. If you drop the components on the **Ground Parts** branch, the components are added to the motion model as motion parts that are grounded, that is they cannot move.

Another method to add components to the motion model is via the right mouse button activated pop-up menus. To use this method, select one or more components listed under the **Assembly Components** branch and then click the right mouse button. The following pop-up menu will be displayed:



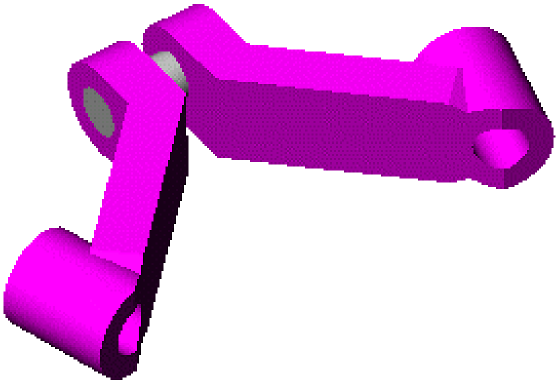
Selecting **Moving Parts** from the pop-up menu will add the components to the motion model as moving parts. Selecting **Ground Parts** will add the components to the motion model as a ground part.

Any time a component is added to the motion model, COSMOSMotion looks at all of the SolidWorks assembly mates that are attached to that component. If it finds that there is an assembly mate between the newly added component and another component that is already participating in the motion model, it will generate a motion joint that maps to the assembly mate. This allows you to take a fully constrained

SolidWorks assembly model and quickly build a simulation ready motion model just by indicating which components from the assembly participate in the motion model.

Rigidly Attached Parts

There often exists in motion simulation the situation where parts that exist as separate, independently constrained components in the SolidWorks assembly model, are really part of a single moving object in the motion model. Consider the model below:

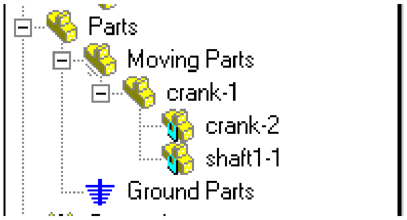


The parts in this assembly are **crank-1**, **crank-2**, and **shaft1-1**. These parts are each separate components in the SolidWorks assembly model. For the purposes of the motion simulation, these three parts act together as a single moving object. To model this in COSMOSMotion, the following procedure can be used:

1 Select component **crank-1** and make it a moving part.

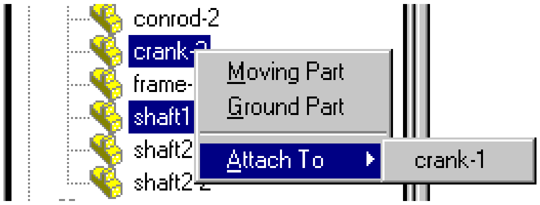
2 Select the components **crank-2** and **shaft1-1**, drag them and drop them on the **crank-1** part.

This has the effect of creating a rigid connection between **crank-1** and **crank-2**, and **crank-1** and **shaft1-1**. The method the IntelliMotion Browser uses to represent these relationships is shown below:



The Browser clearly shows that **crank-2** and **shaft1-1** are subordinate to **crank-1**. During the motion simulation, the mass properties from **crank-2** and **shaft1-1** will be added to the mass properties of **crank-1** and all three components will move as a single moving object.

The pop-up menu can also be used to accomplish the same thing. Once **crank-1** has been added to the motion model as a moving part, you can select components **crank-2** and **shaft1-1**, click the right mouse button and see the following menu:



Selecting **crank-1** from this menu will rigidly attach the two selected components to **crank-1**.

Constraints

Constraints specify how rigid bodies are attached and how they move relative to each other. Constraints in COSMOSMotion are idealized in that they are infinitely rigid, do not have mass, and do not have any clearances or “slop”. There are three types of constraints in COSMOSMotion:

❑ Joints used to constrain the relative motion of a pair of rigid bodies by physically connecting them.

❑ Joint primitives used to enforce standard geometric constraints.

❑ Cam constraints used to simulate contact between a point and a curve or between two curves.

Constraints and applied loads are associative with the geometry that is used to define them. In other words, if a joint origin is defined by an endpoint of an edge and if that endpoint is moved because of a modification made to the geometry, then the joint will move with that endpoint.

Joint and Joint Primitives are generated automatically from SolidWorks assembly mates. However, any joint or joint primitive may be added manually to the motion model using the techniques described below.

Note: The associativity between SolidWorks and COSMOSMotion is unidirectional. Any changes to the COSMOSMotion constraints or applied loads will not be transferred to the SolidWorks geometry.

Understanding Joints

A joint is used to constrain the relative motion of a pair of rigid bodies by physically connecting them.

Note: A rigid body acts and moves as a single unit. SolidWorks components automatically become rigid bodies in COSMOSMotion.

The following table shows the joints supported by COSMOSMotion with the translational and rotational degrees of freedom that they constrain.

Joint	Translational DOF	Rotational DOF	Total DOF Constrained
Revolute	3	2	5
Translational	2	3	5
Cylindrical	2	2	4
Spherical	3	0	3
Universal	3	1	4
Screw	.5	.5	1
Planar	1	2	3
Fixed	3	3	6

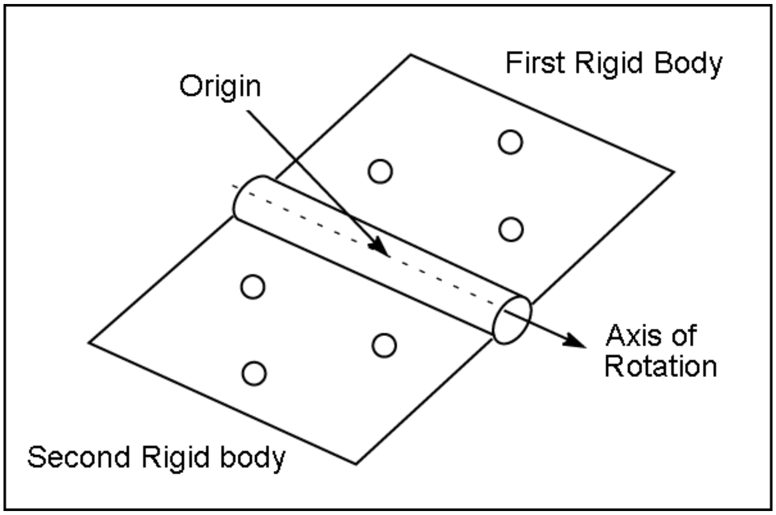
Note: Since a screw joint constrains one degree of freedom by relating a translation to a rotation, we divide the one degree of freedom between translation and rotation.

The following sections describe each joint in detail.

Revolute Joint

The revolute joint allows the rotation of one rigid body with respect to another rigid body about a common axis. The revolute joint origin can be located anywhere along the axis about which the joint's rigid bodies can rotate with respect to each other.

Orientation of the revolute joint defines the direction of the axis about which the joint's rigid bodies can rotate with respect to each other. The rotational axis of the hinge joint is parallel to the orientation vector and passes through the origin.



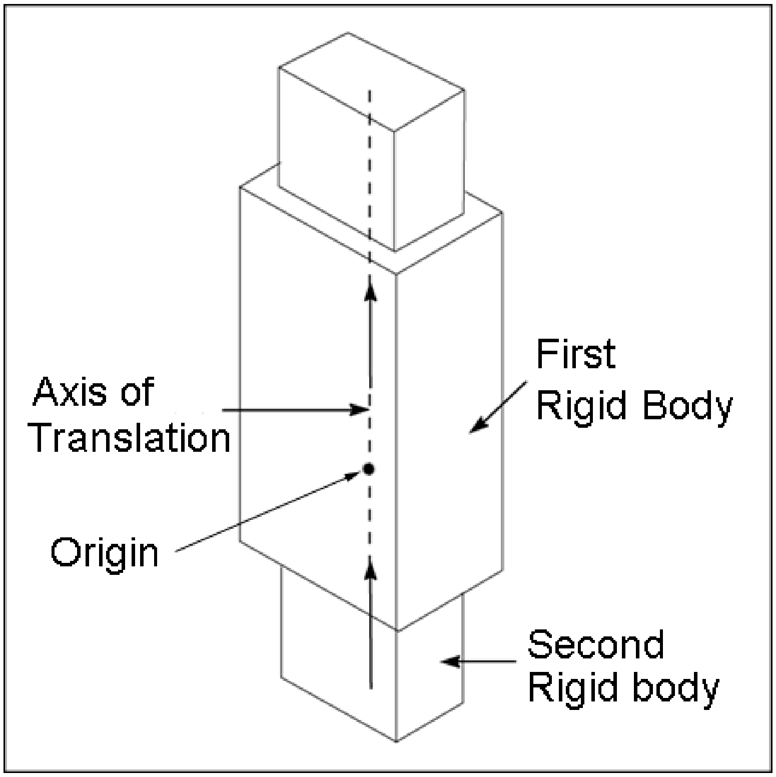
Translational Joint

The translational joint allows one rigid body to translate along a vector with respect to a second rigid body. The rigid bodies may only translate, not rotate, with respect to each other.

The location of the origin of a translational joint with respect to its rigid bodies does not affect the motion of the joint but does affect the reaction loads on the joint. The joint origin location determines where the joint icon is located.

Orientation of the translational joint determines the direction of the axis along which the rigid bodies can slide with respect to each other (axis of translation).

The direction of the motion of the translational joint is parallel to the orientation vector and passes through the origin.



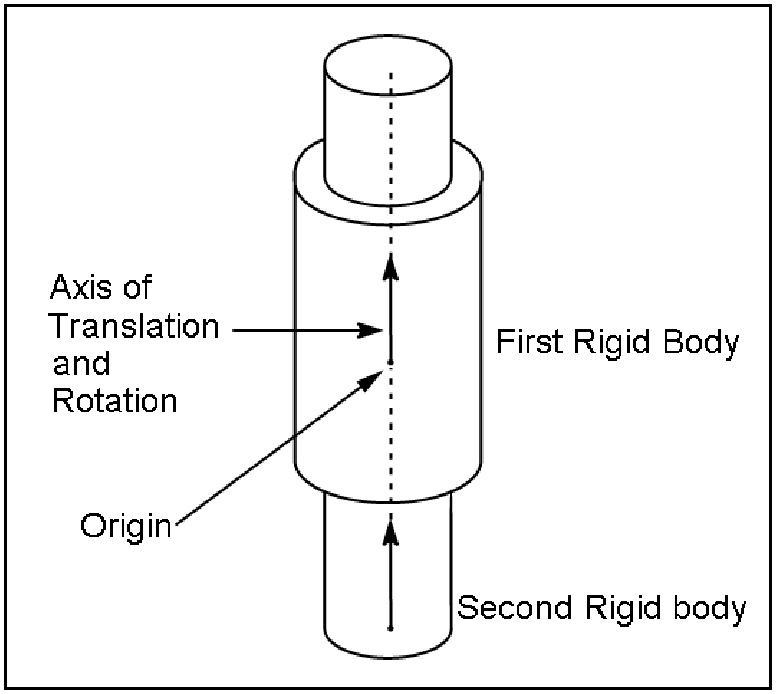
Cylindrical Joint

The cylindrical joint allows both relative rotation as well as relative translation of one rigid body with respect to another rigid body. The cylindrical joint origin can be located anywhere along the axis about which the rigid bodies can rotate or slide with respect to each other.

Orientation of the cylindrical joint defines the direction of the axis about which the rigid bodies can rotate or slide along with respect to each other.

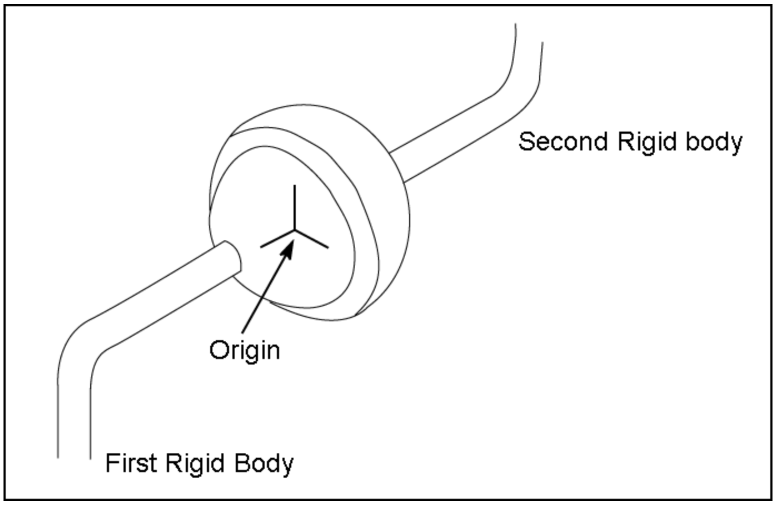
The rotational/translational axis of the cylindrical joint is parallel to the orientation vector and passes through the origin.

A real world example of a cylindrical joint is a hydraulic cylinder.



Spherical Joint

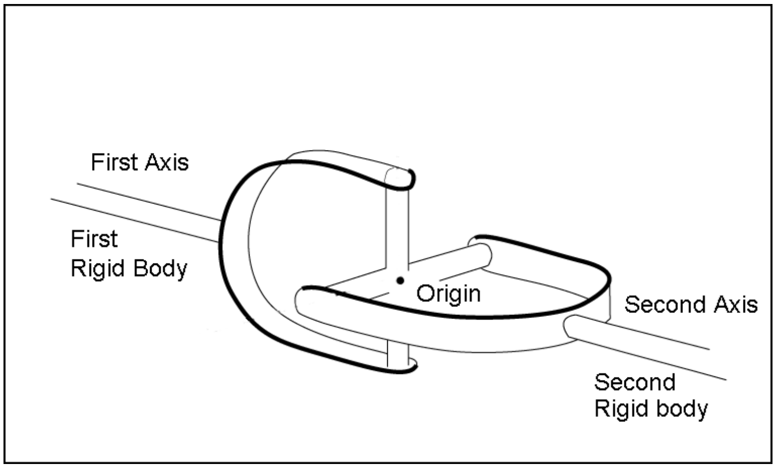
The spherical joint allows free rotation about a common point of one rigid body with respect to another rigid body. The origin location of the spherical joint determines the point about which the joint's rigid bodies can pivot freely with respect to each other.



Universal Joint

The universal joint allows the rotation of one rigid body to be transferred to the rotation of another rigid body. This joint is particularly useful to transfer rotational motion around corners, or to transfer rotational motion between two connected shafts that are permitted to bend at the connection point (such as the drive shaft on an automobile).

The origin location of the universal joint represents the connection point of the two rigid bodies. The two shaft axes identify the center lines of the two rigid bodies connected by the universal joint. Note that COSMOSMotion use rotational axes parallel to the rotational axes you identify but passing through the origin of the universal joint.

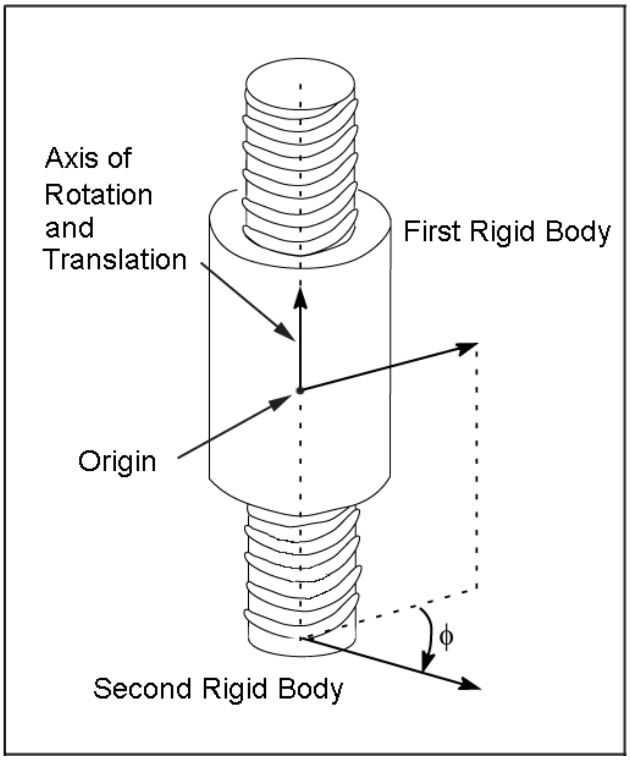


Screw Joint

The screw joint removes one degree of freedom. It constrains one rigid body to rotate as it translates with respect to another rigid body.

When defining a screw joint, you can define the pitch. The pitch is the amount of translational displacement of the two rigid bodies for each full rotation of the first rigid body. The displacement of the first rigid body relative to the second rigid body is a function of the rotation of the first rigid body about the axis of rotation. For every full rotation, the displacement of the first rigid body along the translation axis with respect to the second rigid body is equal to the value of the pitch.

Very often, the screw joint is used with a cylindrical joint. The cylindrical joint removes two translational and two rotational degrees of freedom. The screw joint removes one more degree of freedom by constraining the translational motion to be proportional to the rotational motion.

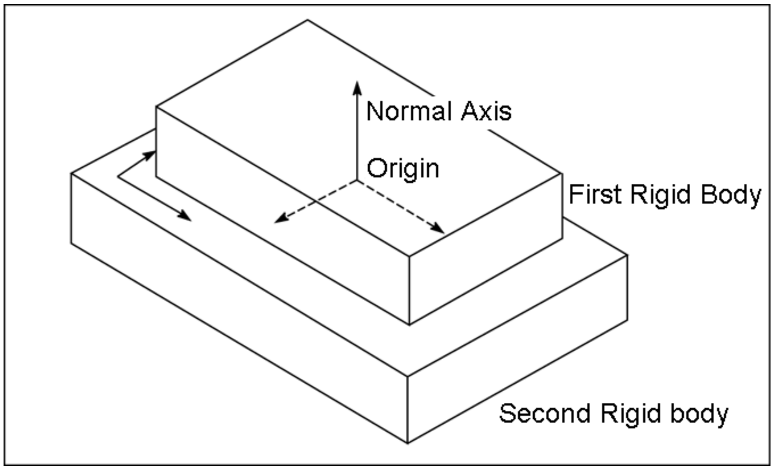


Planar Joint

The planar joint allows a plane on one rigid body to slide and rotate in the plane of another rigid body. The origin location of the planar joint determines a point in space through which the joint's plane of motion passes.

The orientation vector of the planar joint is perpendicular to the joint's plane of motion.

The rotational axis of the planar joint, which is normal to the joint's plane of motion, is parallel to the orientation vector.

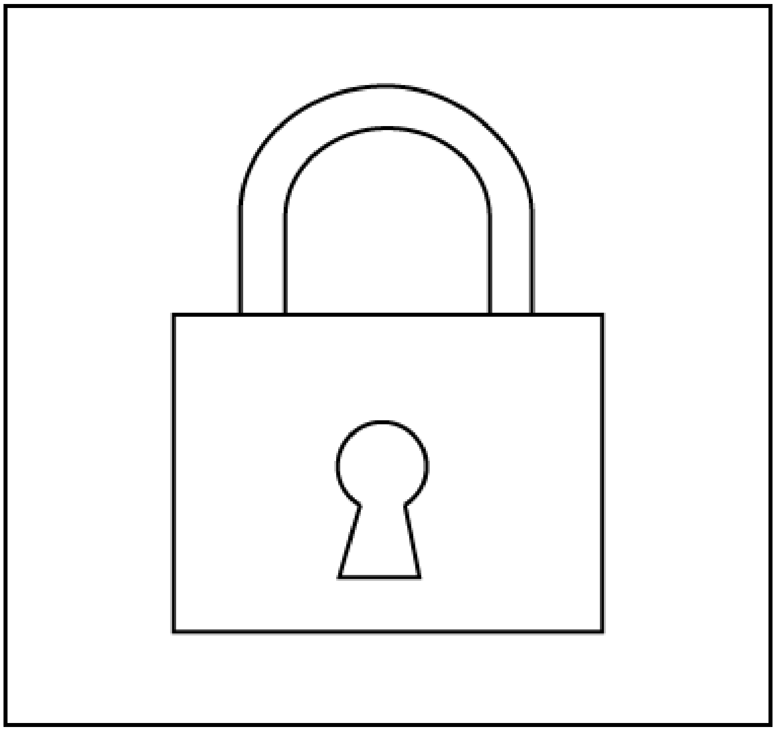


Fixed Joint

The fixed joint locks two rigid bodies together so they may not move with respect to each other.

For a fixed joint, the origin location and orientation of the joint does not effect the outcome of the simulation. We recommend that you place the joint origin at a location where the graphic icon is easily visible.

A real world example of a fixed joint is a weld that holds two parts together.

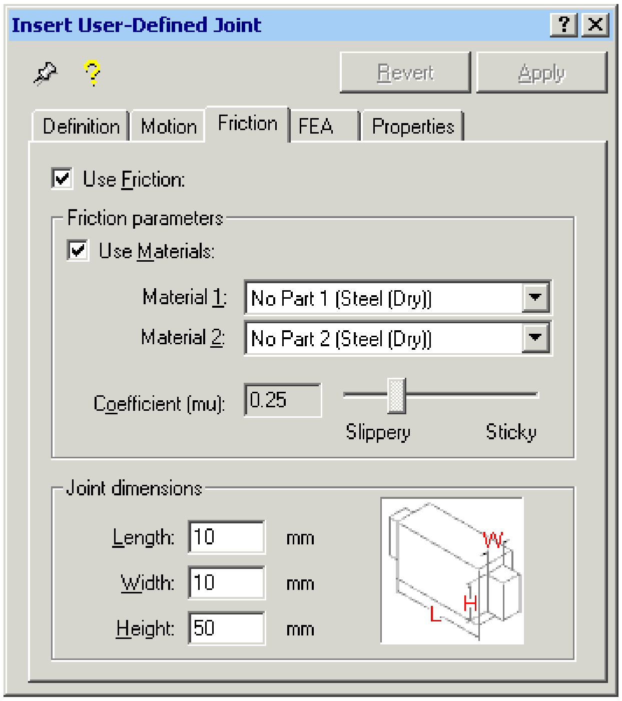


Joint Friction

Revolute, Cylindrical, Translational, Spherical, Universal, and Planar joints all support the application of friction. When friction effects are enabled for these joint types, a force is induced that opposes the motion of the joint and is a function of the reaction forces acting on the joint.

The COSMOSMotion joint friction model uses a combination of dimensional information assigned to a joint and a coefficient of friction that may be entered directly, or that may be obtained automatically from the materials database.

Friction effects are enabled by selecting the Friction tab on the joint properties dialog and then by setting the **Use Friction** checkbox. When this option is checked, the friction parameters can be specified.



☑ Use Friction:

Friction parameters

☑ Use Materials:

Material 1: No Part 1 (Steel (Dry))

Material 2: No Part 2 (Steel (Dry))

Coefficient (mu): 0.25

Joint dimensions

Length: 10 mm

Width: 10 mm

Height: 50 mm

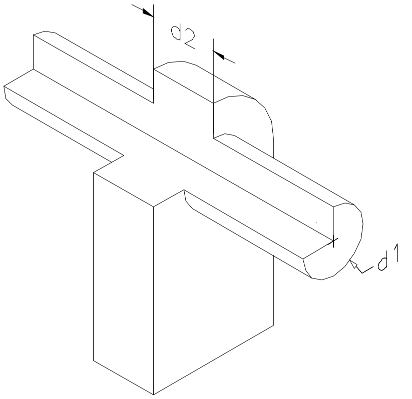
The friction coefficient can be obtained from the materials database by checking the **Use Materials** checkbox, and then by selecting materials from the **Material 1** and **Material 2** combo boxes. The **Coefficient** value is automatically obtained from the materials database.

The coefficient can be entered directly by clearing the **Use Materials** checkbox which enables the edit box labeled **Coefficient (mu)** and also the slider next to the edit box. The coefficient can be entered directly, or the slider can be used to select a higher or lower coefficient.

The values in the **Joint dimensions** area are used to specify the geometric portion of the friction model. The values required differ by join type and are described in the following section.

Revolute Joint Friction Model

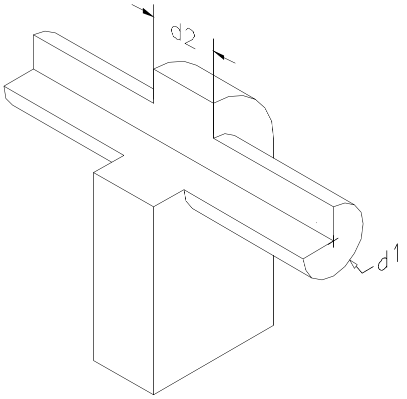
For the purpose of calculating friction effects, a revolute joint is modeled as a snug fit pin rotating in a hole.



Dimension 1 is the radius of the pin, and Dimension 2 is the length of the pin that is in contact with the hole.

Cylindrical Joint Friction Model

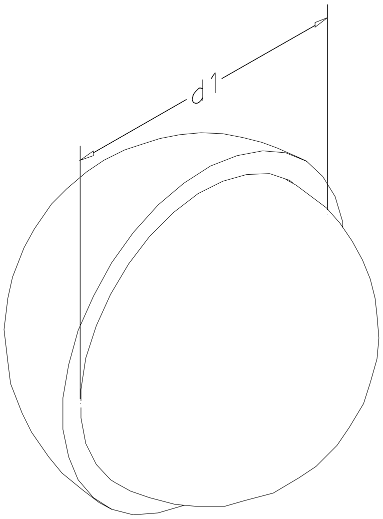
For the purpose of calculating friction effects, a cylindrical joint is modeled as a snug fit pin rotating and sliding in a hole.



Dimension 1 is the radius of the pin, and Dimension 2 is the length of the pin that is in contact with the hole.

Spherical Joint Friction Model

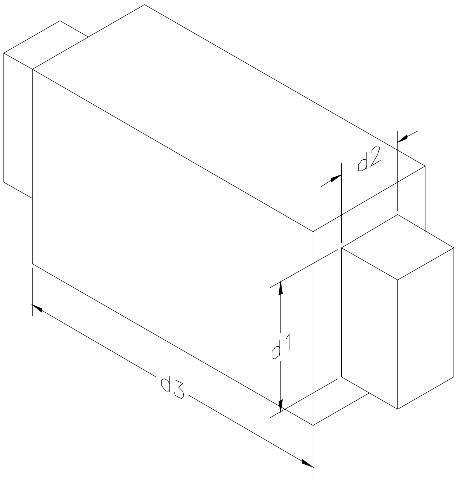
For the purpose of calculating friction effects, a spherical joint is modeled as a ball rotating in a socket. Some portion of the ball's surface area is in contact with the socket.



Dimension 1 is the diameter of the ball.

Translational Joint Friction Model

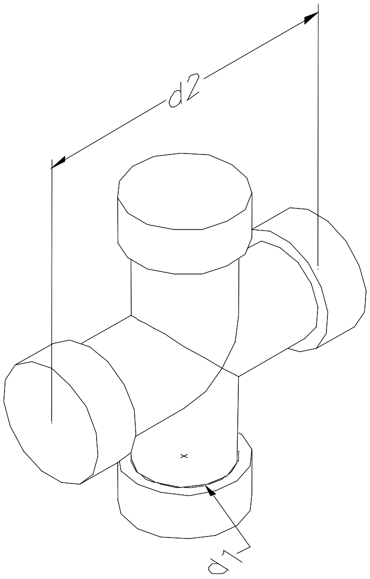
For the purpose of calculating friction effects, a translational joint is modeled as a rectangular bar sliding in a rectangular sleeve.



Dimension 1 is the height of the rectangular bar.
Dimension 2 is the width of the rectangular bar.
Dimension 3 is the length of the bar that is in contact with the sleeve.

Universal Joint Friction Model

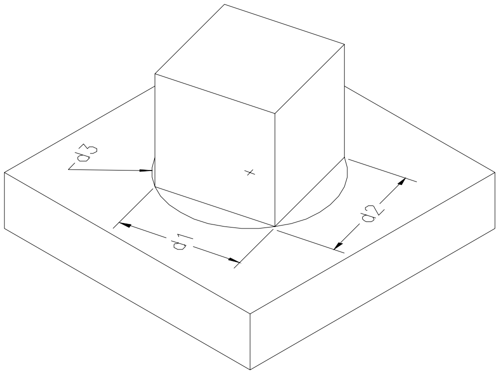
For the purpose of calculating friction effects, a universal joint is modeled as a cylindrical cross piece rotating in a set of end caps.



Dimension 1 is the radius of the bearing end cap.
Dimension 2 is the height of the cross pieces.

Planar Joint Friction Model

For the purpose of calculating friction effects, a planar joint is modeled as one block sliding and rotating across the surface of another block.



Dimension 1 is the length of the sliding block.
Dimension 2 is the width of the sliding block.
Dimension 3 is the radius of a circle, centered at the center of the block face that circumscribes the face of the sliding block that is in contact with the other block.

Friction Results

When friction effects are enabled for a particular joint, an additional reaction force and/or moment, depending on the joint, type will be calculated and displayed. The table below summarizes this information:

Joint Type	Reaction Force	Reaction Moment
Revolute	No	Yes
Cylindrical	Yes	Yes
Spherical	No	Yes
Universal	No	Yes
Translational	Yes	No
Planar	Yes	Yes

Understanding Joint Primitives

Joint primitives (Jprims) constrain one, two, or three degrees of translational and/or rotational freedom. Jprims do not usually have a physical analogue and are predominately useful in enforcing standard geometric constraints.

Jprims can be combined to define a complex constraint. In fact, they can be used to create any of the recognizable joints (except the screw joint).

The following table shows the joint primitives (Jprims) supported by COSMOSMotion with the translational and rotational degrees of freedom that they constrain.

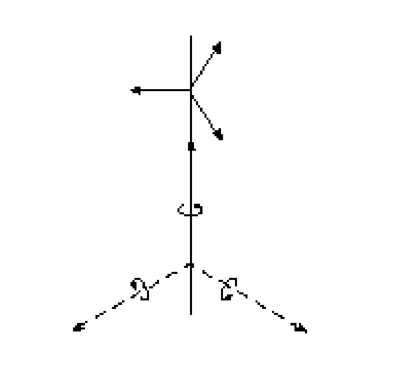
Joint Primitive	Translational DOF	Rotational DOF	Total DOF Constrained
Inline	2	0	2
Inplane	1	0	1
Orientation	0	3	3
Parallel Axis	0	2	2
Perpendicular	0	1	1

The following sections describe the joint primitives in detail.

Inline JPrim

The inline Jprim is a four-degree-of-freedom primitive that allows one translational and three rotational motions of one part with respect to another.

For an inline primitive, COSMOSMotion impose two translational constraints, which confine the translational motion between the parts to the orientation axis. The origin defines the location of the axis.

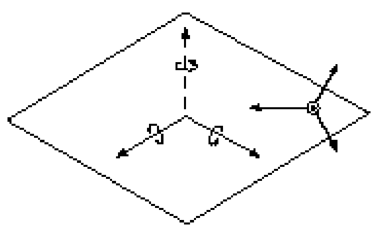


Inline Primitive

Inplane JPrim

The Inplane Jprim is a five-degree-of-freedom primitive that allows both translational and rotational motion of one part with respect to another.

For an Inplane primitive, COSMOSMotion imposes one translational constraint, which confines the translational motion between the parts to the orientation plane.

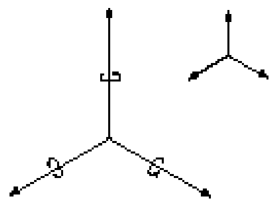


Inplane Primitive

Orientation JPrim

Indicates a three-degree-of-freedom primitive that allows only translational motion of one part with respect to another.

For an orientation primitive, COSMOSMotion imposes three rotational constraints to prevent the two parts from rotating with respect to each other.

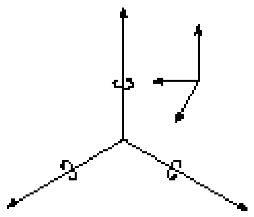


Orientation Primitive

Parallel Axes JPrim

The parallel axes constraint is a four-degree-of-freedom primitive that allows both translational and rotational motion of one part with respect to another.

For a parallel axes primitive, the orientation vector specifies the axes on the two components that are kept parallel. The orientation vector runs through the origin. The primitive permits relative rotation about the orientation vector and permits all relative displacements.

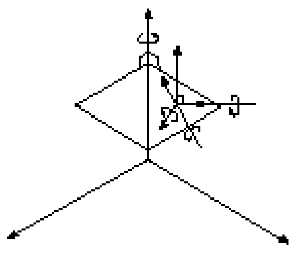


Parallel Axes Pimitive

Perpendicular JPrim

The perpendicular constraint is a five-degree-of-freedom primitive that allows both translational and rotational motion of one part with respect to another.

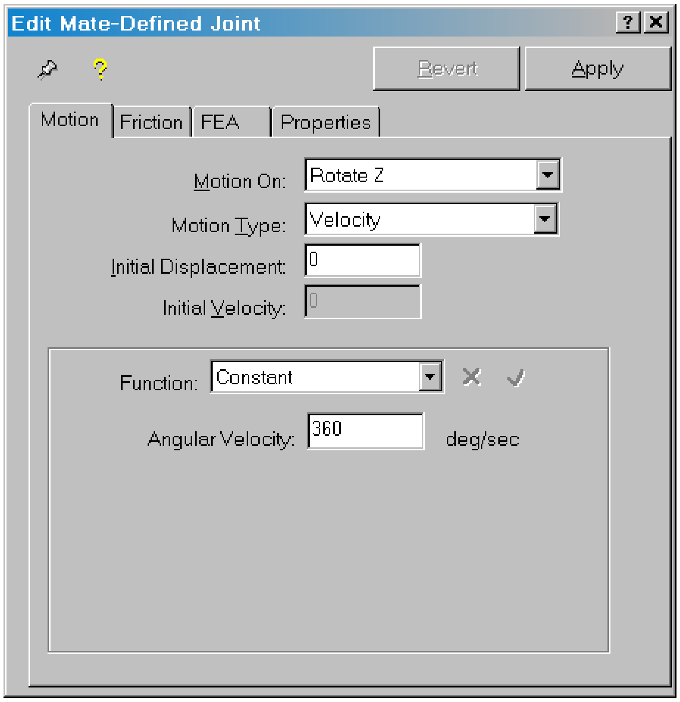
For a perpendicular primitive, COSMOSMotion imposes a single rotational constraint on the components so that the component axes remain perpendicular. This allows relative rotations about either z-axis, but does not allow any relative rotation in the direction perpendicular to both z-axes.



Perpendicular Primitive

Understanding Motions

You add motion drivers to joints to define the movement of the joint over time. A motion driver is a feature of a joint. The motion is defined on the Motion panel in the Joint dialog box.



Motion On: Rotate Z

Motion Type: Velocity

Initial Displacement: 0

Initial Velocity: 0

Function: Constant

Angular Velocity: 360 deg/sec

A motion dictates the displacement, velocity, or acceleration of a joint as a function of time. For example, the following function expression defines a constant velocity motion driver that can be used to rotate the first rigid body one complete revolution (360 degrees) per unit time with respect to the second rigid body:

360d * TIME

Note: See the Motion Expression section below for more details on motion function expressions.

The motion driver supplies the force required to make the joint satisfy the defined motion. This force is available as an output at each motion driver in COSMOSMotion. This output can be used to properly size a motor or actuator.

Degrees of Freedom

In the DOF field on the Motion panel, you can select the degree of freedom to which the motion is applied. The degrees of freedom can be either rotational or translational. For example, hinge joints have one rotational degree of freedom. You may apply only one rotational motion to a hinge joint. Planar joints have two translational and one rotational degree of freedom. Therefore, you can apply two translational motions and one rotational motion to a planar joint. You can apply a motion to any degree of freedom on a joint.

Motion Type

A motion driver can define either the joint displacement, velocity, or acceleration. By default, the motion driver type is set to "free", meaning that the joint is free to move as driven by the rest of the mechanism. You can set the motion type in the Insert Joint dialog box on the Motion panel. The simplest motions to define are the constant displacement, constant velocity, or constant acceleration motions:

- ❑ **Constant Displacement -** Placing a constant value in the Motion Expression field when the motion type is set to Displacement creates a constant displacement motion. A constant displacement motion holds the joint in a fixed position. The rigid bodies that the joint connects do not move relative to each other during simulation. The effect is very much like using a fixed joint to constrain the two rigid bodies. The advantage of constraining two parts with a constant displacement motion is that the motion can be adjusted to different positions. Constraining two rigid bodies with a fixed joint, however, is computationally more efficient.

- ❑ **Constant Velocity -** Placing a constant value in the Motion Expression field when the motion type is set to Velocity creates a constant velocity motion, which moves the joint with the required force to produce a constant velocity.

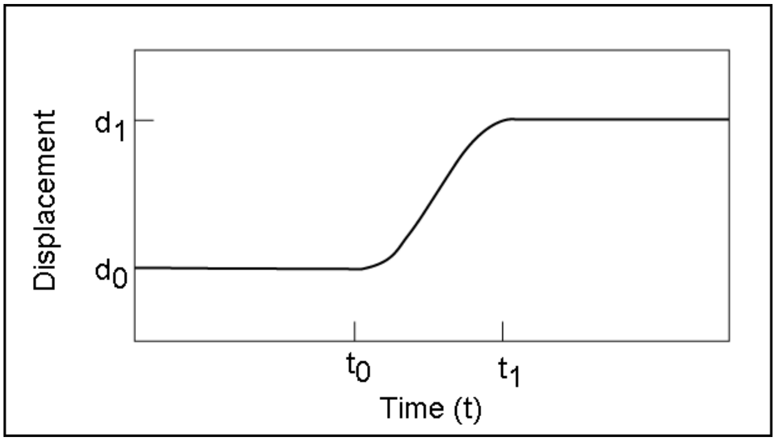
- ❑ **Constant Acceleration -** Placing a constant value in the Motion Expression field when the motion type is set to Acceleration creates a constant acceleration motion. This motion moves the joint with the required force to produce a constant acceleration.

Motion Expression

A motion function specifies the exact displacement, velocity, or acceleration applied to a joint as a function of time. There are four pre-defined functions supported.

Additionally a general expression capability is available that allows any MSC.ADAMS function to be specified. When the **Motion Type** is set to something other than **Free**, a combo box labeled **Function** is displayed the allows the choice of function. The four predefined function are described below.

- ❑ **Constant Function -** The Constant Function creates a function expression that moves the joint displacement, velocity, or acceleration in a constant manner. One value needs to be specified.
- ❑ **Step Function** – A Step Function creates a function expression that moves the joint displacement, velocity, or acceleration between two values with a smooth motion. Before and after the transition, the displacement, velocity, or acceleration of the joint is constant.



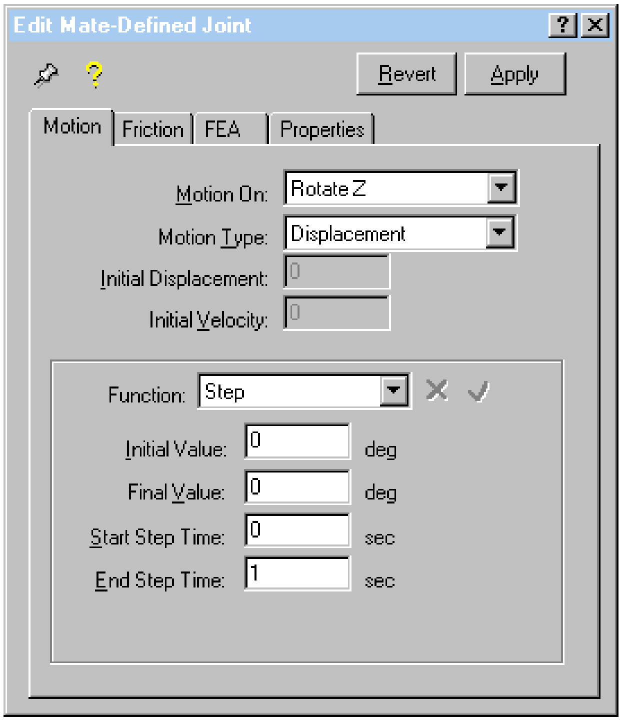
When you select Step from the Function Type list, additional fields are displayed that allows you to enter the values for the starting and ending times of the transition and the initial and final displacement, velocity, or acceleration values. You should enter a real value for each of these terms. The input items are as follows:

Initial Value: The value of the function before the step.

Final Value: The value of the function after the step.

Start Step Time: The time at which the step begins.

End Step Time: The time at which the step ends.



Motion On: Rotate Z

Motion Type: Displacement

Initial Displacement: 0

Initial Velocity: 0

Function: Step

Initial Value: 0 deg

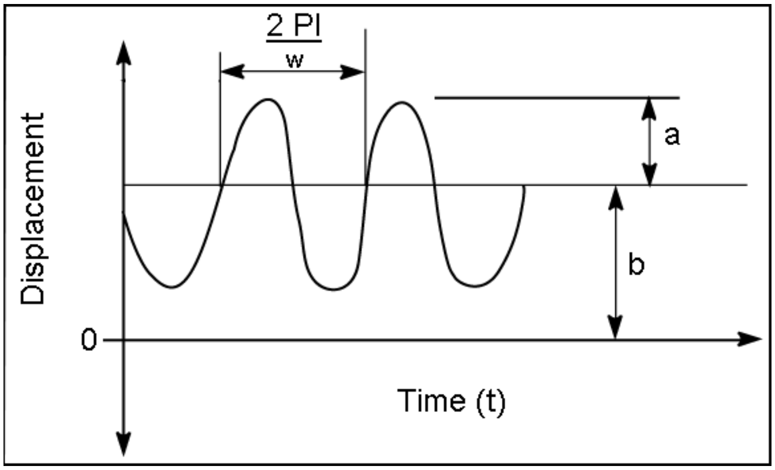
Final Value: 0 deg

Start Step Time: 0 sec

End Step Time: 1 sec

Note: For more details on the Step function, see Chapter 6..

- ❑ **Harmonic Function -** The harmonic function creates a sinusoidal displacement, velocity, or acceleration. When you select the Harmonic function type additional fields are displayed that allow you to enter the values of the parameters for a simple harmonic function. Each of these prompt you for a real value.



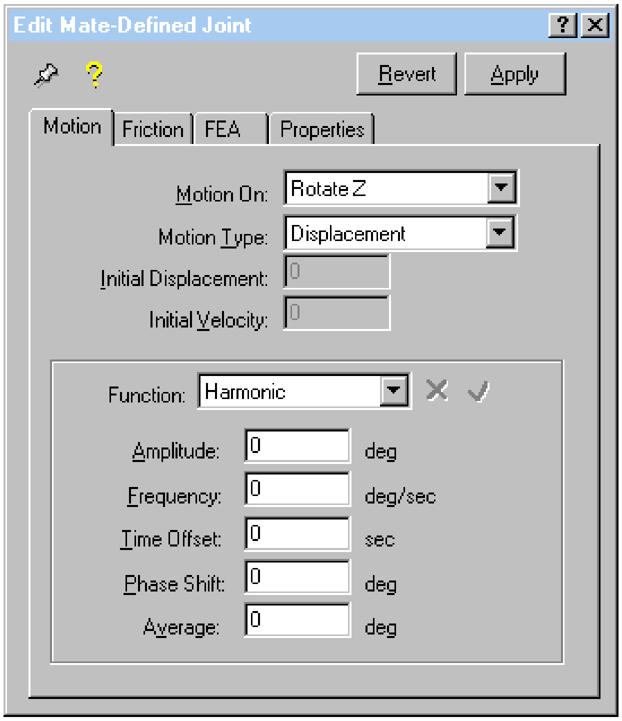
The input items are as follows:

Amplitude: Enter largest displacement from average value.

Frequency: Enter speed of oscillation.

Time Phase Shift: Enter starting time offset.

Average: Enter the value about which the motion oscillates.



Motion On: Rotate Z

Motion Type: Displacement

Initial Displacement: 0

Initial Velocity: 0

Function: Harmonic

Amplitude: 0 deg

Frequency: 0 deg/sec

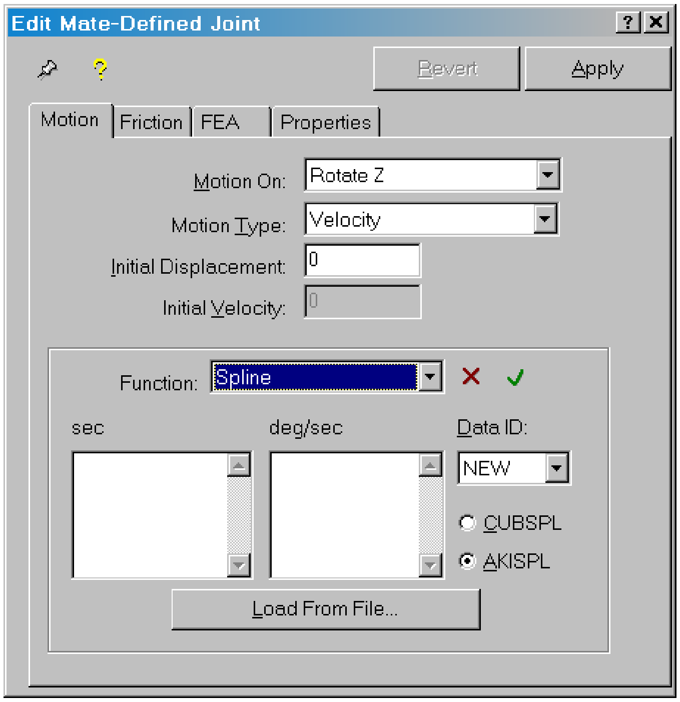
Time Offset: 0 sec

Phase Shift: 0 deg

Average: 0 deg

Note: For more details on the Harmonic Function, see Chapter 6.

- **Spline -** The Spline function allows the joint displacement, velocity, or acceleration to be defined by entering a table of data. One column in the table is the time, and the other column is the displacement, velocity, or acceleration of the joint at that time.



Motion On: Rotate Z
Motion Type: Velocity
Initial Displacement: 0
Initial Velocity: 0
Function: Spline
Data ID: NEW
○ CUBSPL
● AKISPL

The MSC.ADAMS/Solver fits a smooth curve to the data point values by interpolating the data points. The function expression for the motion then references this curve to get the actual position of the joint during the simulation. You must define at least four data points.

The AKISPL and CUBSPL toggles control which kind of curve is used to fit your data points, Akima or Cubic. A cubic curve will work well even if your data points are not evenly spaced. An Akima curve is very fast, but may not work as well if your points are not evenly spaced.

The Load From File button displays the standard Windows File Open dialog and allows the selection of a file that contains data points. The file can be free format, white space or commas can separate the columns of data. The only real restriction is that each line in the file contains one and only one set of data points.

You can create more than one set of data points and reference them between motions. You can control which set of points is being applied to the motion by using the Data ID list. The first time you create a curve the Data ID list will be set to “NEW”. If you create more than one set or modify existing sets, you can select from existing sets of data points or create a new one.

Other Function Expressions

You can also enter the text for other MSC.ADAMS function expressions in the Motion Expression field. The function expression of a motion driver must be a function of time. If you make your motion expression a function of displacements, forces, or any other variables in the system, the MSC.ADAMS/Solver issues an error message and stops execution.

If you enter an incorrect expression, COSMOSMotion will issue an error message indicating that your expression is incorrect. You can not close the Insert Joint dialog box until you correct the expression.

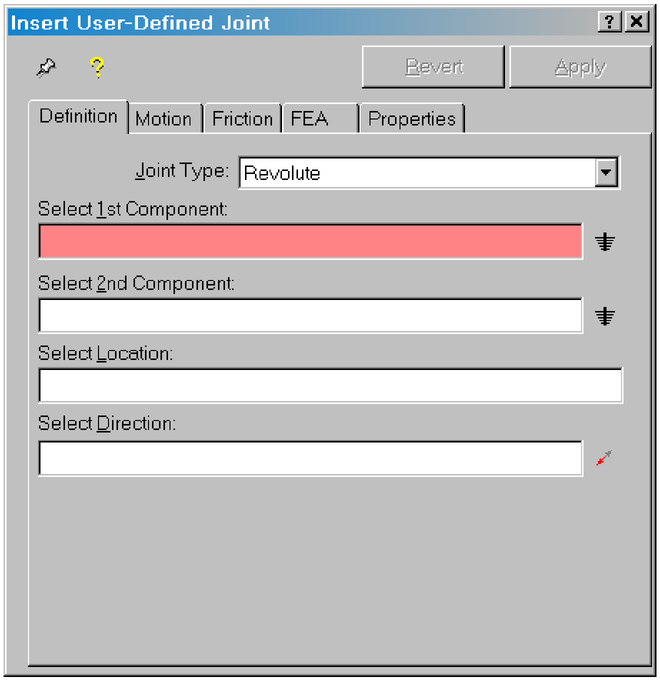
See Chapter 6 for a list of functions that are supported in COSMOSMotion.

Creating Joints, Joint Primitives, and Motions

Joints, joint primitives and motion can be created by a number of methods:

- From the Motion menu by selecting **Motion, Joint, <desired joint type>**
- From the Browser by clicking on the **Constraints** branch and selecting **Add Constraint, <desired joint type>** from the pop-up menu.
- From the Browser by clicking on the **Joints** branch and selecting the type of joint to add from the pop-up menu.
- From the Browser by selecting one or two moving or ground parts and choosing **Add Constraint, <desired joint type>** from the pop-up menu.

No matter which method is used, the following dialog box is displayed:



Joint Type: Revolute

The **Joint Type** combo box allows the type of joint to be added to be changed.

The **Select 1st Component** and **Select 2nd Component** boxes allow the components to which the joint will be attached to be selected. You can choose the component by either selecting it on screen, or by selecting it from the IntelliMotion Browser.

You can attach one end of a joint to global ground by clicking the ⏚ button.

You can specify the location of the joint by first clicking in the **Select Location** box, and then by choosing an edge or vertex in your SolidWorks model. The joint will be attached to the geometry at this point.

You can choose the direction of the joint axes, if the joint supports an axis, by first clicking in the **Select Direction** box, and choosing an edge, planar face, circular edge, or cylinder to define the joint orientation. If the joint is to have a motion attached, pay attention to the direction of the axis defined. The positive direction of the motion will use the right hand rule with respect to the direction of the joint axis. You can check the **Flip** checkbox to reverse the direction of the joint axis.

To locate and orient joints, you can take full advantage of the existing SolidWorks geometry. When you are selecting the components to which the joint is connected, you can select geometry features that will automatically define the origin and orientation of the joint. The first geometry feature selected may define both the origin and orientation. If it does not, the second geometry feature selected may define either or both the origin and the orientation. If geometry features are not selected or if the origin or the orientation is not defined by the selection of the components, then you must separately define the origin and/or the orientation. The table below lists which geometry features automatically define the origin and orientation.

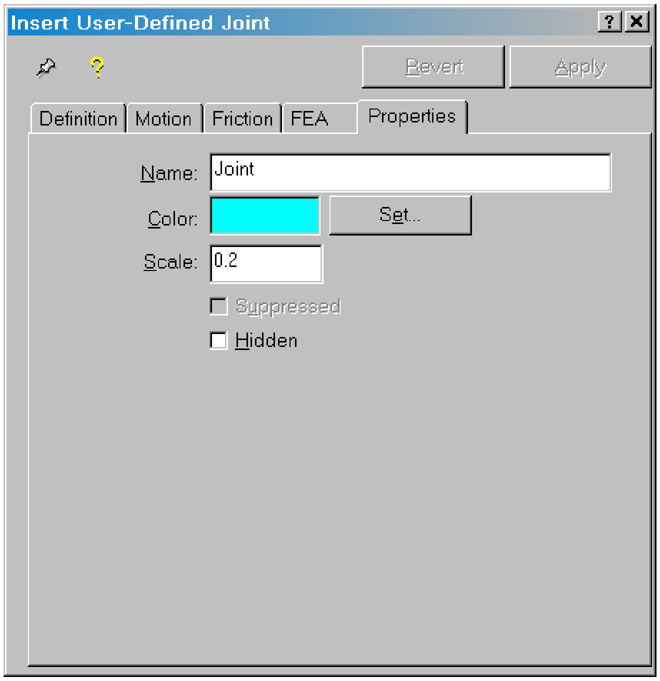
Geometry Feature	Joint Origin at:	Orientation/direction:
Linear Edge	Midpoint of edge	Vector parallel to edge
Vertex	Point	<Is not set>
Planar face	<Is not set>	Normal to face
Circular edge	Center of circle	Normal to face
Cylinder	<Is not set>	Centerline of cylinder

When selecting the components or geometry when defining joints or forces, you can easily replace the geometry or components after you select them. You can replace them as you create the joint or force or when you edit them.

Once all of the information for the joint has been defined, you can click the **Apply** button to actually create the joint. Once **Apply** has been clicked, the joint dialog will be closed.

If you would like to keep the joint dialog visible, in order to add more than one joint, click the button. You can then specify a joint's parameters, click **Apply** to create the joint, and repeat the process for any other joints you want to add.

You can specify a joint's properties by clicking on the **Properties** tab. This will change the dialog so it looks like:



Name: Joint

Color:

Scale: 0.2

☐ Suppressed

☐ Hidden

The name of the joint can be specified by enter a new name in the **Name** box.

The color of the joint can be changed by clicking the **Set** button and selecting a new color from the Windows Color dialog box.

The size of the joint can be changed by entering a new number that represents the size of the joint in the **Scale** box.

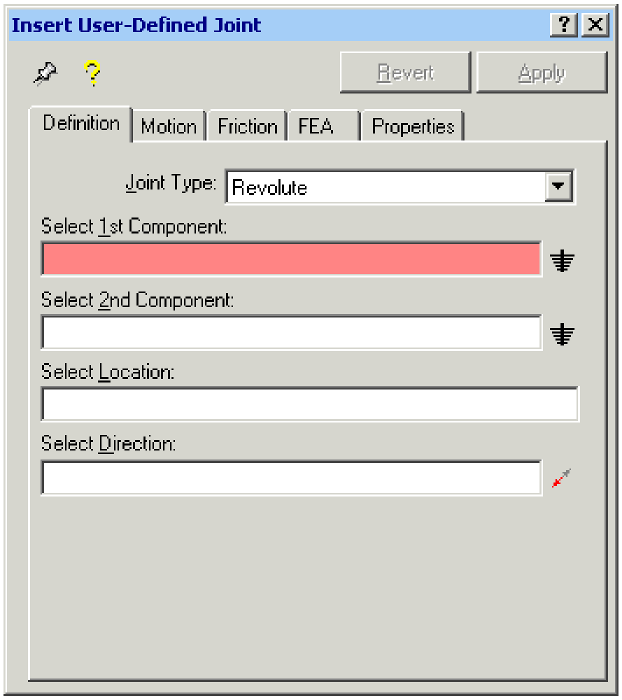
The **Suppressed** checkbox controls whether the joint is included in the simulation or not. If the box is checked, the joint will not be included in the simulation, although it will still exist in the motion model.

The **Hidden** checkbox determines whether the joint is visible or not. If the box is checked, the joint is not visible.

To create a joint or joint primitive:

1 Initiate the joint creation process by any of the methods discussed previously. The joint dialog box will appear.

2 The **Definition** panel should be visible when the Insert Joint dialog box appears. If it is not, click the **Definition** tab.



Joint Type: Revolute

3 Select the two rigid bodies or components to be connected by the joint. If you selected the components in the Browser before opening the Insert Joint dialog box, the component names should appear in the Components field.

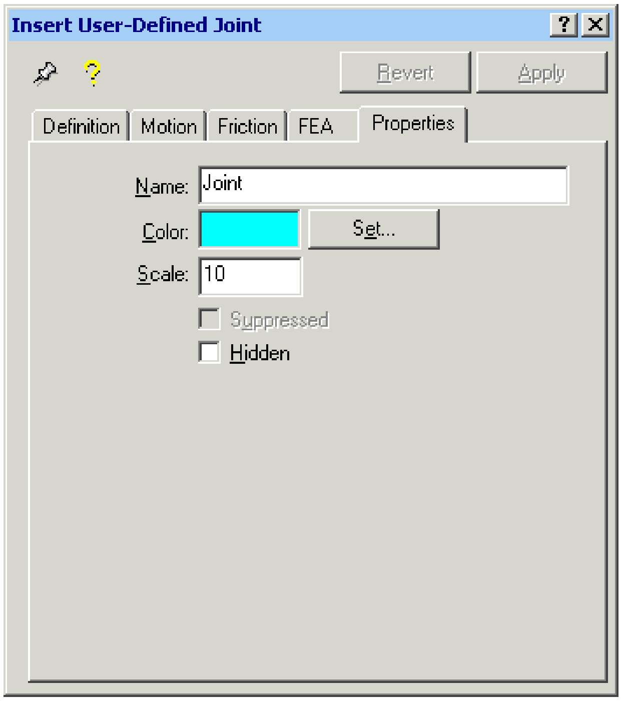
4 If the correct origin has not already been defined, select an edge, vertex, or circular edge in your existing model geometry to define the origin of the joint. You may have automatically defined the origin if you selected a geometry feature on one of the components.

5 Select an edge, planar face, circular edge, or cylinder to define the joint orientation. If you are creating a universal joint, you must define orientations for the two shaft axes. Hold down the Control key while selecting the second axis.

You can change the direction of the joint axis by toggling the **Flip** toggle, found at the bottom of the Orientation panel, on or off.

Once the origin and orientation have been defined, the joint icon will appear on the screen at the joint origin with the defined orientation.

6. If you are creating a screw joint, you can also define the pitch of the joint by entering it in the box at the bottom of the **Definitions** page.
7. Optional step: modify default constraint name by clicking on the **Properties** tab to display the **Properties** panel and entering a new name in the **Joint Name** field.



Name: Joint

Color:

Scale: 10

□ Suppressed

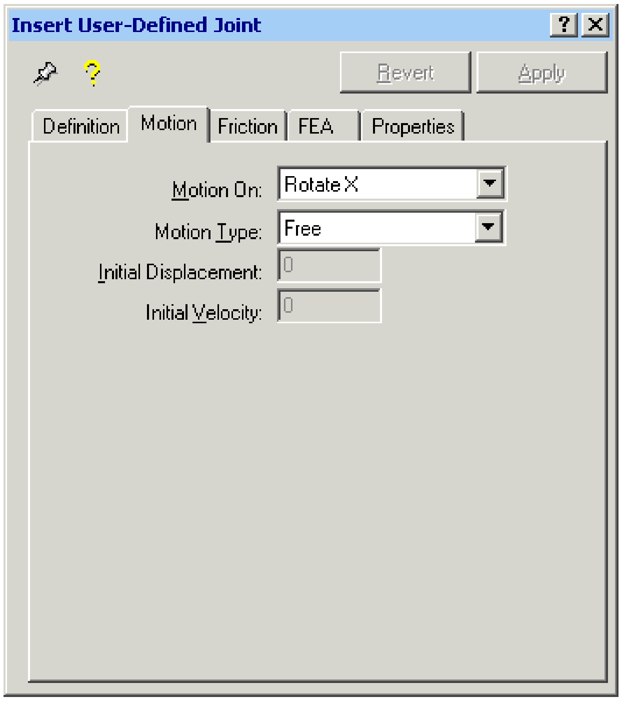
□ Hidden

8. If you would like to apply a motion driver to an open degree of freedom on the joint, select the Motion tab and continue to the next set of instructions. If you do not need to apply a motion driver to this constraint, click **Apply** to complete the joint creation process and leave the Insert Joint dialog box open. You can also click **OK** to complete the joint creation process and close the Insert Joint dialog box. Clicking **Cancel** closes the Insert Joint dialog box without creating the joint.

The final step in creating a joint or a joint primitive is to apply a motion or motions to the degrees of freedom of the joint. This is an optional step since you will not need to apply motions to all joints.

To define a motion:

1 Select the **DOF** (degree of freedom) to which you would like to apply the motions.



Insert User-Defined Joint

Motion On: Rotate X

Motion Type: Free

Initial Displacement: 0

Initial Velocity: 0

2 Select the type of motion to apply: **Free**, **Displacement**, **Velocity**, or **Acceleration**.

The motion type is “free” by default, meaning that the joint is free to move as driven by joints, motions, and forces in the system.

3 Enter an expression in the **Motion Expression** field OR click **Template Editor** to open the function template.

With the Template Editor you can create step or harmonic functions or you can enter data points. Select the desired function from the **Function Type** list. After you have filled in all of the fields for the function, click **OK** to complete the function and close the Template Editor dialog box.

The Template Editor inserts the function expression at the cursor in the Motion Expression field. If you have an expression in the Motion Expression field that is not highlighted, the Template Editor adds the new function to the existing expression. An expression that is highlighted in the Motion Expression field when you open the Template Editor will be replaced by the expression that you create in the Template Editor.

In the Template Editor, degrees are assumed for rotational motion. In the Motion Expression field radians are assumed unless you enter a “d” behind the number. For example, to specify a velocity of 60 degrees per second, you should enter “60d” in the **Motion Expression** field and set the **Type** to Velocity. If you enter “60” in the Motion Expression field, you will set the velocity to 60 radians per second.

For more details on the motion expression, see the Motion Expression section above.

4 Click **Apply**

Understanding Contact Constraints

COSMOSMotion supports the following Contact constraints:

❑ Point-curve - restricts a point on one rigid body to lie on a curve on a second rigid body

❑ Curve-curve - constrains one curve to remain in contact with a second curve. Curve-curve also supports an **intermittent** option that allows the two curves to separate and rejoin. This allows 2D contact to be modeled.

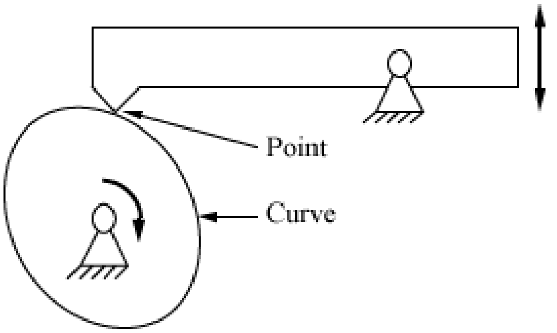
❑ 3D Contact – Simulates two bodies colliding by detecting when they come in contact with each other and calculating and applying the forces that result from that collision. While not technically a constraint, 3D contact is grouped together with the other contact functionality in COSMOSMotion.

The following table shows the Contact constraints in COSMOSMotion with the degrees of freedom that they constrain. One way of thinking of this is that a point-curve adds a curve parameter, which when selected, specifies an x-y-z location, resulting in a net constraining of 2 degrees of freedom.

Joint	Translational DOF	Rotational DOF	Total DOF Constrained
Point-curve	2	0	2
Curve-curve	2	0	2
Curve-curve Intermittent	0	0	0
3D	0	0	0

Point-Curve Contact

A point-curve constraint restricts a fixed point defined on one rigid body to lie on a curve defined on a second rigid body. This is useful for modeling devices such as a pin in a slot or a simple cam follower mechanism where a lever arm is articulated by the profile of a revolving cam.

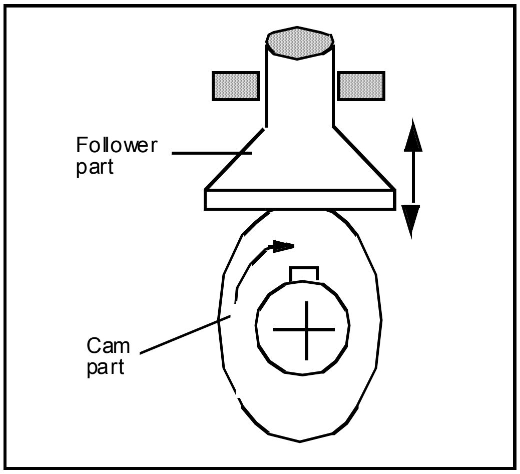


When modeling a pin-in-slot mechanism, the point-curve constraint keeps the center of the pin in the center of the slot, while allowing it to move freely along the slot and rotate in the slot. Note that the point-curve does not stop the pin at the end of the slot. If you need to restrict the travel of the pin, use an applied load to represent the impact of the pin with the end of the slot.

Note: The point does not have to be contacting the curve when you define a point-curve constraint. When you simulate the mechanism, the assembly must be constrained such that the point and curve are allowed to come in to contact or the simulation will fail.

Curve-Curve Contact

A curve-curve Contact constraint restricts a curve defined on the first rigid body to remain in contact with a second curve defined on a second rigid body. Note that the exact point of contact along the curves is not specified and will likely move during the course of the simulation as one curve rolls or slips along the other curve.



The curve-curve constraint is useful for modeling cams where the point of contact between two rigid bodies changes during the motion of the mechanism. An example is a valve lifter where the cam is lifting a plate-like object. The point of contact between the plate (flat curve) and the cam (oval curve) change depending on the position and shape of the cam. Edges selected representing curves for the curve-curve constraint must have all points lying in the same plane. The resulting planar curve may exist in any location and orientation in your SolidWorks assembly with the following restriction.

The two curves (each defined by selected edges/datum curves) used in defining the curve-curve constraint must lie in the same plane. It is possible to initially define the curves such that they are not in the same plane, but COSMOSMotion will move the rigid bodies during simulation to ensure that the two curves are constrained to the same plane of motion with respect to each other.

Note: The curves do not have to be in contacting when you define a curve-curve constraint. When you simulate the mechanism, the assembly must be `constrained such that the curves are allowed to come in to contact or the simulation will fail.

Curve-Curve Contact with Intermittent Contact

If the Curve-Curve constraint is designated as **intermittent**, the curve may separate and rejoin as the dynamics of the system dictate. An intermittent curve-curve constraint supports friction and multiple points of contact. When a curve-curve constraint is designated as intermittent, there are additional parameters that effect that contact characteristics of the curve that must be specified.

3D Contact

3D contact allows the simulation of general interactions between parts as they collide and rebound. It takes into account the impact and friction portions of the collision and calculates all the locations where the two bodies come in contact.

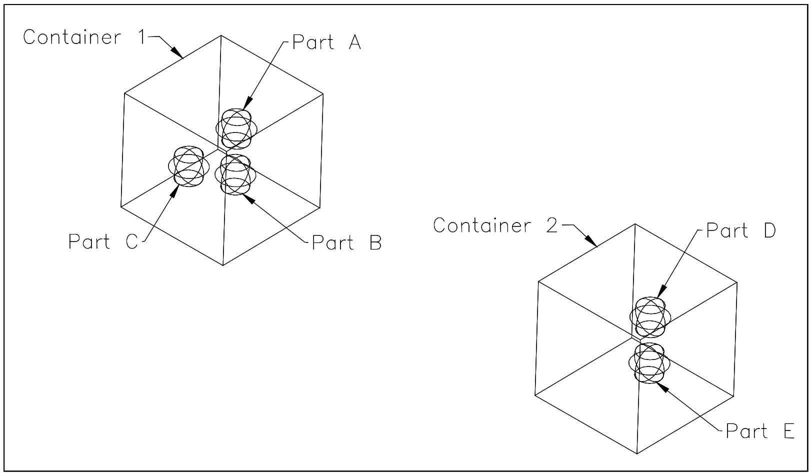
3D contact is more straightforward to use than 2D (Intermittent Curve/Curve) contact, but requires more computational resources than 2D contact, and in many cases will be slower to simulate.

Contact Containers

In contrast to other COSMOS/ Motion entities which are attached to one or two parts, a 3D contact entity is comprised of one or two *contact containers*. A contact container contains one or more motion parts. In the case where a contact entity is defined by two contact containers, 3D contact will be enabled between every part in the first container, and every part in the second container. In the case where a contact entity is defined by just one contact container, 3D contact will be enabled between every part in the container and every other part in the container.

When you define a contact entity you enter parts in 1 or 2 selection sets. These selection sets generate the containers. If you only add parts in the first selection set, then only one container is generated. If you enter parts in two selections sets, then two container groups are generated.

Consider the example of the two containers shown below:



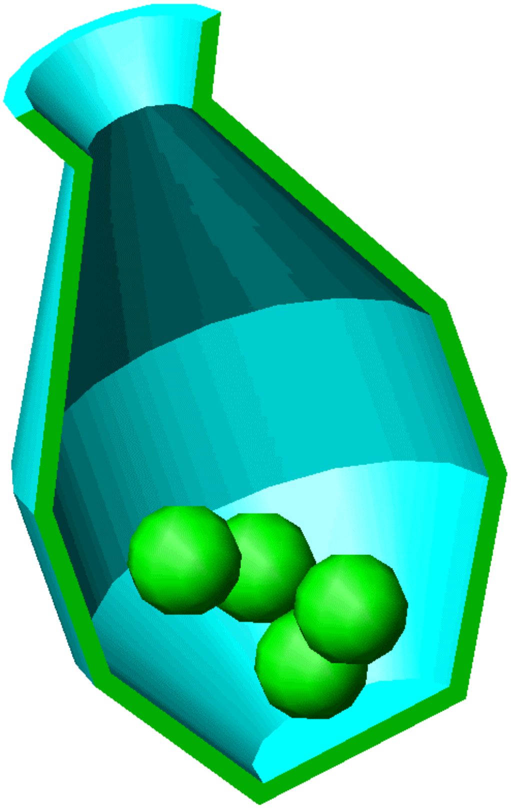
If a 3D contact object is attached to Container1 and Container2, contacts will be detected between the following pairs of parts:

Part A and Part D
Part A and Part E
Part B and Part D
Part B and Part E
Part C and Part D
Part C and Part E

If a 3D contact object is attached only to Container1, contacts will be detected between the following pairs of parts:

Part A and Part B
Part A and Part C
Part B and Part C

Consider another example of some balls bouncing in a vase:



This problem can be modeled using two 3D contact entities. One would be attached to two contact containers. The first contact container would contain the four balls, and the second contact container would contain the vase. This would enable contact between each of the four balls and the vase. The second 3D contact object would be attached to a single container which would only contain the four balls. This would enable contact between each ball and all the other balls.

Creating Contact Constraints

Point/Curve and Curve/Curve constraints can be created by a number of methods:

- From the Motion menu by selecting **Motion, Contact, Point-Curve** or **Motion, Contact, Curve-Curve**

- From the Browser by clicking on the **Constraints** branch and selecting **Add Contact Point/Curve Contact** or **Add Contact Curve/Curve Contact**.
- From the Browser by clicking on the **Contacts** branch and selecting either **Add Point/Curve Contact** or **Add Curve/Curve Contact** from the pop-up menu.
- From the Browser by selecting one or two moving or ground parts and choosing **Add Constraint, Point/Curve Contact** or **Add Constraint Curve/Curve Contact** from the pop-up menu.

Creating Point-Curve Constraints

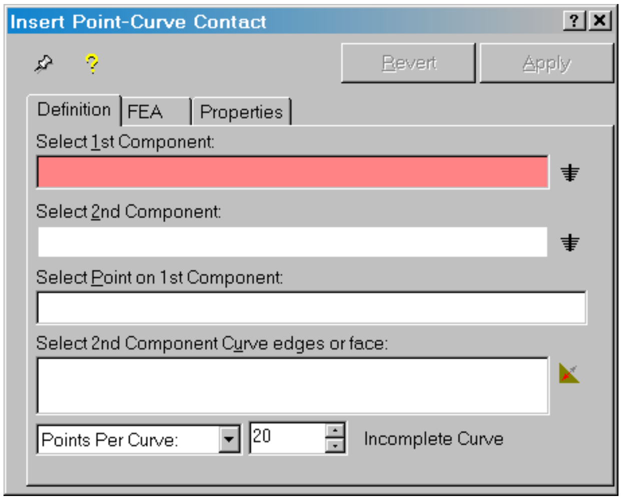
For the point-curve constraint, you must define a point on one rigid body and a curve on another rigid body.

To Create a Point-Curve Constraint:

1 Initiate the Point/Curve creation process by any of the methods listed above.

The Insert Point-Curve dialog box appears.

2 The **Definition** panel should be visible when the Insert Point-curve dialog box appears. If it is not, click the **Definition** tab.



3 Select the first component where the point will reside.

4 Select the

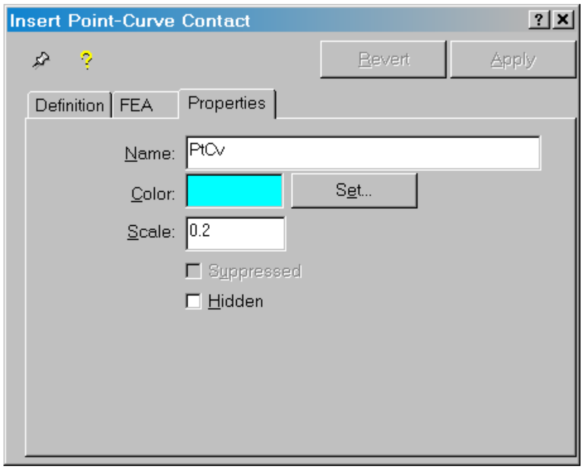
5 Select a vertex, circular edge, or linear edge to define the point on the first rigid body that will contact the curve.

6 The combo box located in the lower left corner of the dialog box determines how the selected curves are approximated by COSMOSMotion. Choosing **Points per Curve** requires that the number of points generated on *each* selected curve by specified. Selecting Tolerance requires that a curve tolerance be specified. The curve used by COSMOSMotion will deviate no more than the specified tolerance from the original curve.

7 Select one or more connected curves on the second rigid body to define the curve with which the point will be in contact.

As you select curves, the curve names appear in the Entity/Component field. In the lower right corner of the dialog box, the curve status is displayed. The curve status is Incomplete until you select a curve. After you select a curve, the status changes to Open or Closed depending on the curve you select. If you select multiple curves and select a curve that is not contacting any of the previously selected curves, the status becomes Incomplete. If you select multiple curves that define a closed curve, the status becomes Closed.

8 Optional step: modify the default name of the constraint. Select the **Properties** tab to display the **Properties** panel. Enter a new name in the **Contact Name** field.



Name: PtCv

Color:

Scale: 0.2

☐ Suppressed

☐ Hidden

9 Click **Apply** to create the point-curve constraint and close the dialog box.

Creating Curve-Curve Constraints

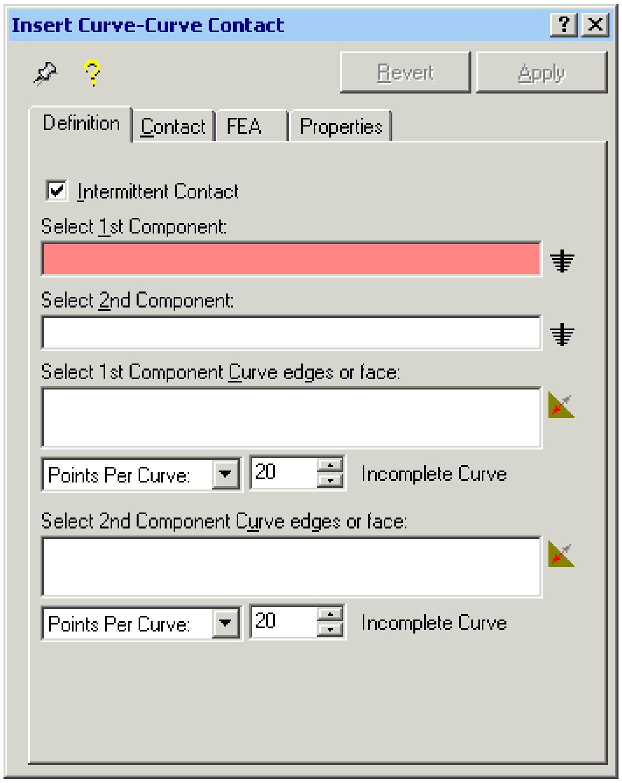
For the curve-curve constraint, you must define two curves on two separate parts.

To Create a Curve-Curve Constraint:

1 Initiate the creation process by any of the methods discussed previously.

The Insert Curve-Curve dialog box appears.

2 If necessary, click the left **Definitions** tab to display the **Definitions** panel.


Insert Curve-Curve Contact
?
Revert
Apply
Definition
Contact
FEA
Properties
Select 1st Component:
Select 2nd Component:
Select 1st Component Curve edges or face:
Select 2nd Component Curve edges or face:

☑ Intermittent Contact

Points Per Curve: 20 Incomplete Curve

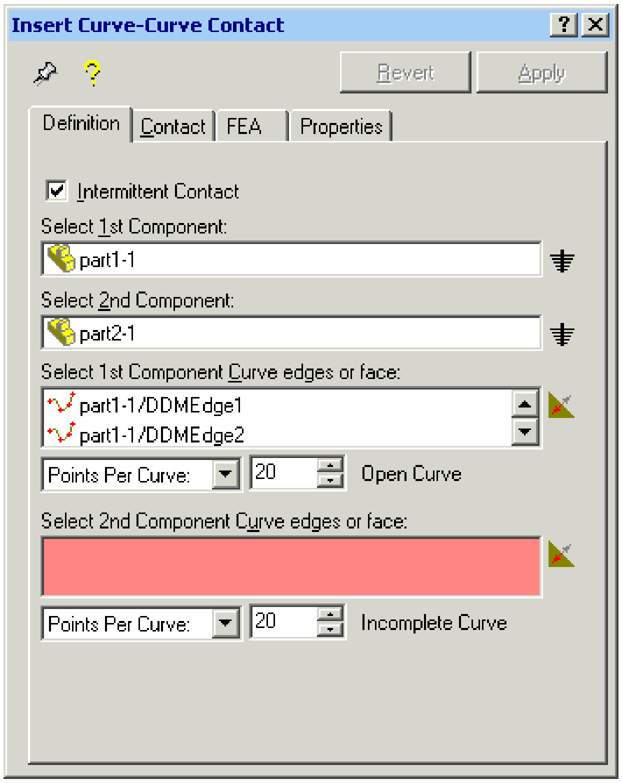
Points Per Curve: 20 Incomplete Curve

3 Select the two components that will be constrained.

4 Select one or more connected curves on the first rigid body to define the curve. These curves will appear in the first large box.

As you select curves, the curve names appear in the Entity/Component field. In the lower right corner of the dialog box, the curve status is displayed. The curve status is Incomplete is until you select a curve. The curve status changes to Open after you select a curve. If you select multiple curves and select a curve that is not contacting any of the previously selected curves, the status becomes Incomplete. Once you define a closed curve, the status becomes Closed.

5 When finished with the first curve, click in the second large box .



☑ Intermittent Contact

Select 1st Component:
part1-1

Select 2nd Component:
part2-1

Select 1st Component Curve edges or face:
part1-1/DDMEdge1
part1-1/DDMEdge2

Points Per Curve: 20 Open Curve

Select 2nd Component Curve edges or face:

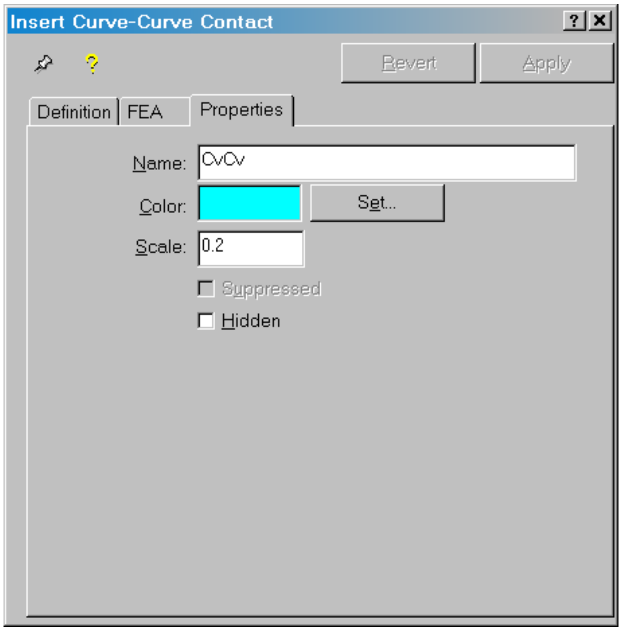
Points Per Curve: 20 Incomplete Curve

6 The combo box below each list of selected curves determines how the selected curves are approximated by COSMOSMotion. Choosing **Points per Curve** requires that the number of points generated on *each* selected curve by specified. Selecting Tolerance requires that a curve tolerance be specified. The curve used by COSMOSMotion will deviate no more than the specified tolerance from the original curve.

7 Select one or more connected curves on the second rigid body to define the second curve.

8 As you select curves, the curve names appear in the Entity/Component field.

9 Optional step: modify the default name of the constraint. Select the **Properties** tab to display the **Properties** panel. Enter a new name in the **Contact Name** field.



Name: CvCv

Color:

Scale: 0.2

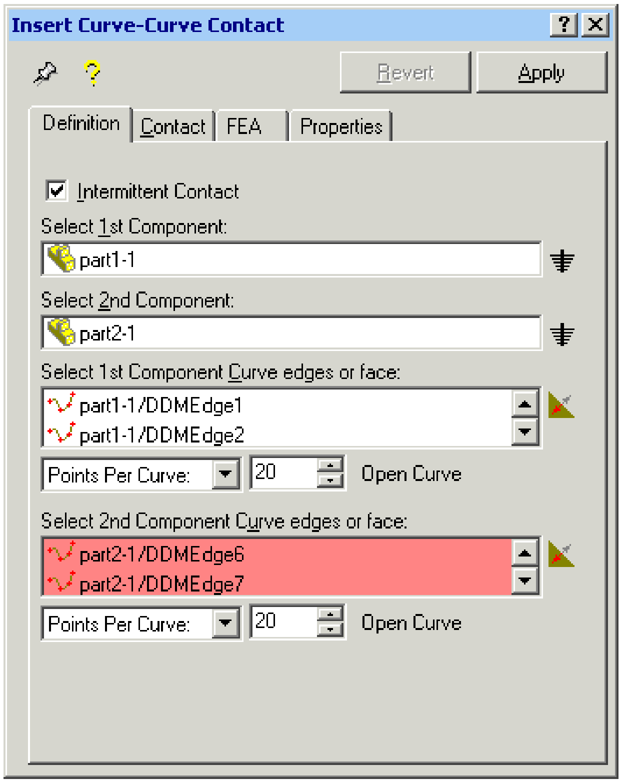
□ Suppressed

□ Hidden

10 Click **Apply** to create the curve-curve constraint and close the dialog box.

To Create an Intermittent Curve-Curve Constraint

1. Follow the instructions in the previous section to create the Curve-Curve constraint.
2. On the **Definitions** panel, check the box labeled **Intermittent Contact**.



☑ Intermittent Contact

Select 1st Component: part1-1

Select 2nd Component: part2-1

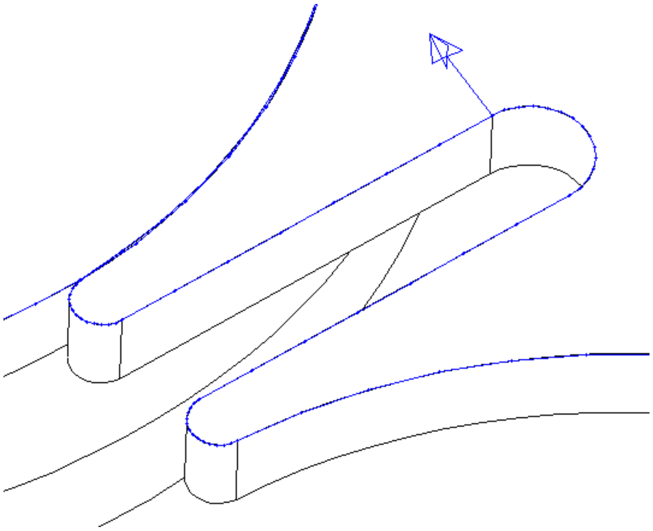
Select 1st Component Curve edges or face:
part1-1/DDMEdge1
part1-1/DDMEdge2

Points Per Curve: 20 Open Curve

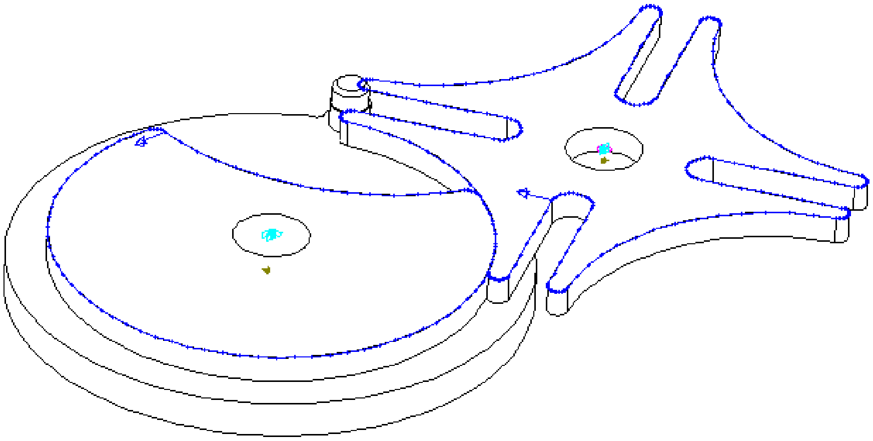
Select 2nd Component Curve edges or face:
part2-1/DDMEdge6
part2-1/DDMEdge7

Points Per Curve: 20 Open Curve

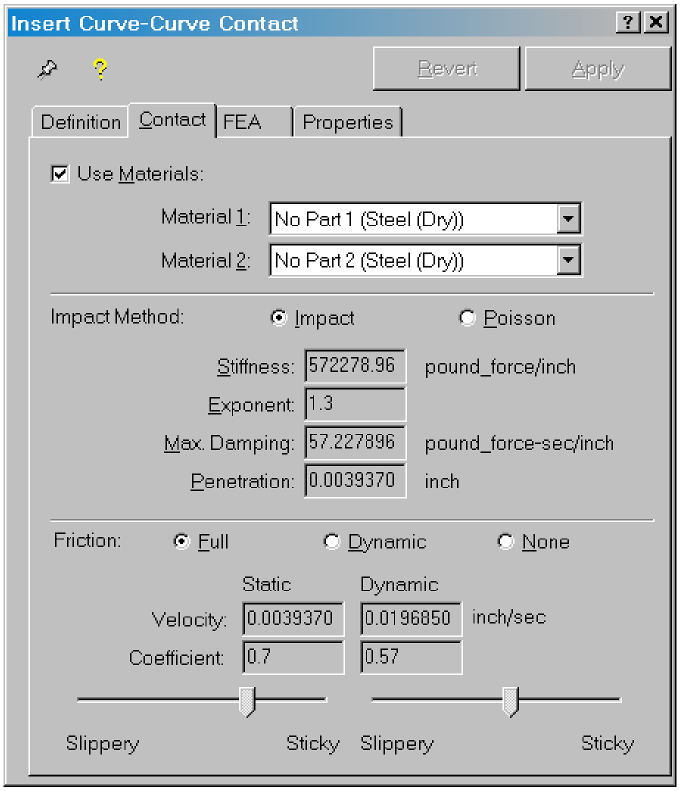
3 On the graphics window, once curves are defined for both parts, you will see an arrow appear on each curve. These arrows are used to denote the direction pointing into the solid.



The ◩ buttons next to each curve selection on the dialog allows you to toggle the direction each arrow points.



4 Click the **Contact** tab to display the contact panel.



☑ Use Materials:

Material 1: No Part 1 (Steel (Dry))

Material 2: No Part 2 (Steel (Dry))

Impact Method: ● Impact ○ Poisson

Stiffness: 572278.96 pound_force/inch

Exponent: 1.3

Max. Damping: 57.227896 pound_force-sec/inch

Penetration: 0.0039370 inch

Friction: ● Full ○ Dynamic ○ None

	Static	Dynamic	
Velocity:	0.0039370	0.0196850	inch/sec
Coefficient:	0.7	0.57	

5 This page allows the friction and impact properties for the contact to be established.

If the **Use Materials** box is checked, the two combo boxes labeled **Material 1** and **Material 2** are enabled. Selecting materials from these boxes will extract impact and friction parameters from the materials database.

If the **Use Materials** box is not checked then impact and friction parameters can be entered directly. The impact parameters provide two options, **Poisson** which allows two numbers to be entered that relates the incoming and outgoing velocities of the two contacting parts, or the traditional impact parameters that allow the stiffness, damping, exponent, and penetration to be specified. See Page 103 for more information on impact parameters.

Friction can be established by selecting the **Full** option which uses both static and dynamic friction, the **Dynamic** option which only uses dynamic friction, or the **None** option which turns friction off.

If Full friction is selected, both the static and dynamic friction values are used. If the **Use Materials** option is selected, the values will be obtained from the materials database. If this option is not selected, these values can be entered directly.

These parameters are:

The **Static Velocity** is the absolute value of the velocity at which the static coefficient of friction starts to transition to the dynamic coefficient of friction. At velocities below this value, the static coefficient of friction is active.

The **Dynamic Velocity** is the absolute value of the velocity above which the dynamic coefficient of friction is active. For velocities between **Static Velocity** and **Dynamic Velocity**, a step function is used to transition between the static coefficient of friction and the dynamic coefficient of friction.

The **Static Coefficient** value is the static coefficient of friction and the **Dynamic Coefficient** is the dynamic coefficient of friction. These values must be between 0 and 1. The sliders can also be used to change the coefficient value.

Creating 3D Contacts

3D Contacts can be created by a number of methods:

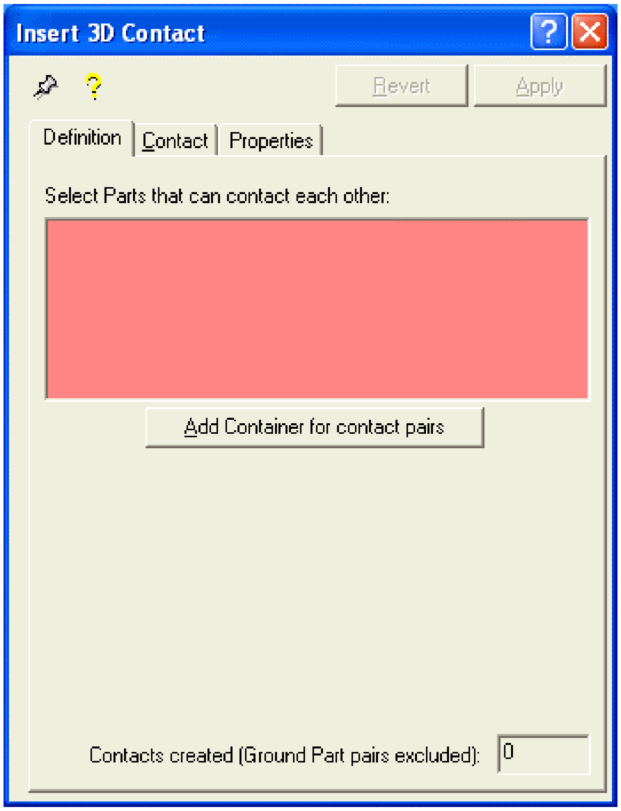
- From the Motion menu by selecting **Motion, Contact, 3D Contact…**
- From the Browser by clicking on the **Constraints** branch and selecting **Add Contact 3D Contact**.
- From the Browser by clicking on the **Contacts** branch and selecting **Add 3D Contact**.
- From the Browser by selecting one or two moving or ground parts and choosing **Add 3D Contact** from the pop-up menu.

To Create 3D Contacts:

1 Initiate the 3D contact creation process by any of the methods listed above.

The Insert 3D Contact dialog appears:

2 If necessary, click the left **Definitions** tab to display the **Definitions** panel.

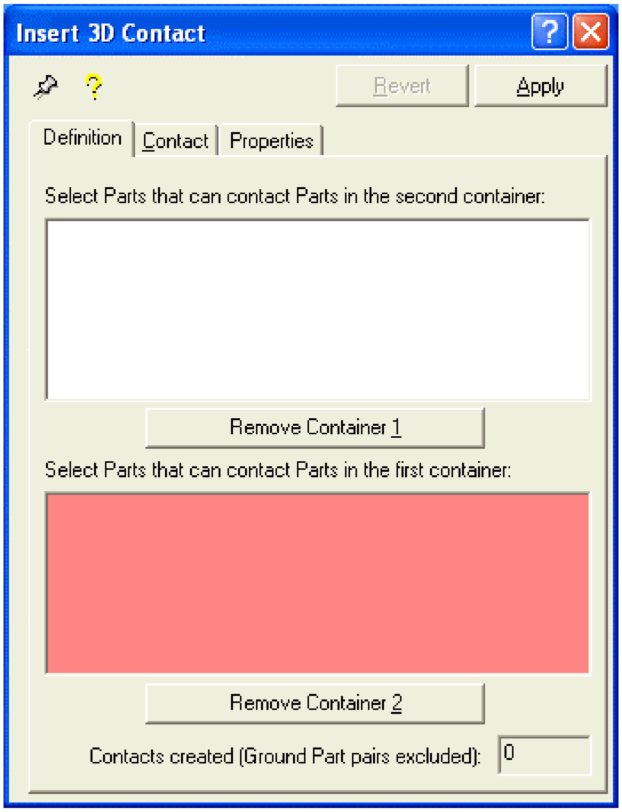


Select Parts that can contact each other:

Contacts created (Ground Part pairs excluded): 0

3 Select the components that will be added to the first contact container. If a second container is to be defined, click the **Add Container for contact pairs** button. See page 58 for more information on contact containers.

4 Add components to the second container.



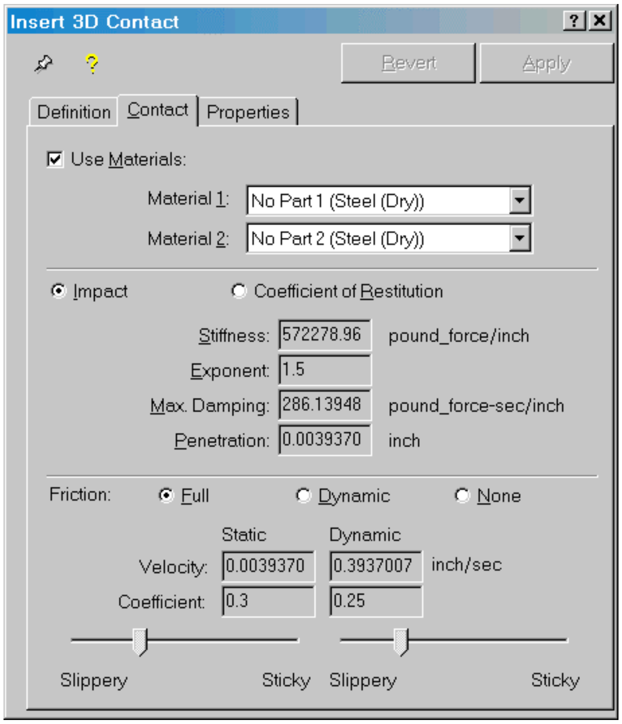
Select Parts that can contact Parts in the second container:

Select Parts that can contact Parts in the first container:

Contacts created (Ground Part pairs excluded): 0

Either of the containers can be removed by clicking the **Remove Container** button.

5 Click the **Contact** tab to display the contact panel.



☑ Use Materials:

Material 1: No Part 1 (Steel (Dry))

Material 2: No Part 2 (Steel (Dry))

● Impact ○ Coefficient of Restitution

Stiffness: 572278.96 pound_force/inch

Exponent: 1.5

Max. Damping: 286.13948 pound_force-sec/inch

Penetration: 0.0039370 inch

Friction: ● Full ○ Dynamic ○ None

	Static	Dynamic	
Velocity:	0.0039370	0.3937007	inch/sec
Coefficient:	0.3	0.25	

6 This page allows the friction and impact properties for the contact to be established. Please see the previous section on Intermittent contact on page 68 for a complete explanation.

Tips on Creating Contact Constraints

The following are some suggestions for creating Contact constraints.

❑ **Specify a large number of curve points.**

Be sure to specify a sufficiently large number of points to achieve an acceptable fit. The MSC.ADAMS/Solver attempts to represent the curves without exceeding the number of points defined in the Points Per Edge Curve setting. Assuming this setting is 3, then the MSC.ADAMS/Solver tries to fit the spline to each edge using only 3 points per edge. This will likely result in a relatively poor fit because of the number of points per edge is relatively low.

❑ **Use closed curves whenever possible.**

It is generally easier to select a closed curve if possible. Open curves represent modeling difficulties when the point on the follower rigid body approaches one of the end points of the open curve. Closed curves have no beginning and ending points, and, therefore, avoid this numerically difficult situation.

❑ **Avoid defining the initial configuration of a point-curve constraint with the point near to one of the end points of the curve.**

❑ **Avoid highly nonlinear edges.**

❑ **Avoid curve-curve constraints that have more than one contact point.**

The MSC.ADAMS/Solver may have difficulties finding the correct solution if you have a curve-curve constraint with more than one contact point. One way to ensure that contact points are unique is to specify curve shapes that are convex. Note that for a convex curve, any line segment connecting two arbitrary points on the curve lies in the domain of the curve (it does not intersect the curve). The same is not true for non-convex curves.

Joint Couplers

Joint couplers allow the motion of a revolute, cylindrical, or translational joint to be "coupled" to the motion of another revolute, cylindrical or translational joints. The two coupled joints may be of the same type, or the may be different types, for example a revolute joint may be coupled to a translational joint. The coupled motion may also be of the same type, or may be a different type. For example, the rotary motion of a revolute joint may be coupled to the rotary motion of a cylindrical joint, or the translational motion of a translational joint may be coupled to the rotary motion of a cylindrical joint.

A coupler takes to value, one for the motion of each joint, and from these derives a ratio that relates the motion of one joint to the motion of another joint. For example, if a revolute joint is being coupled with a translational joint, 90 degrees could be specified for the revolute joint, and 100 mm for the translational joint. This means that for every 90 degrees the revolute joint rotates, the translational joint will be moved 100 mm, or 1 degree of rotation results in 0.9 mm of translation (90/100).

A coupler does not differentiate between a "driver" joint and a "driven" joint. Either joint in the couple may at one point in time be "driving" the other joint. A potential for locking does exist if both joints in a couple are driven by motion generators. The couple may conflict with the motion generator.

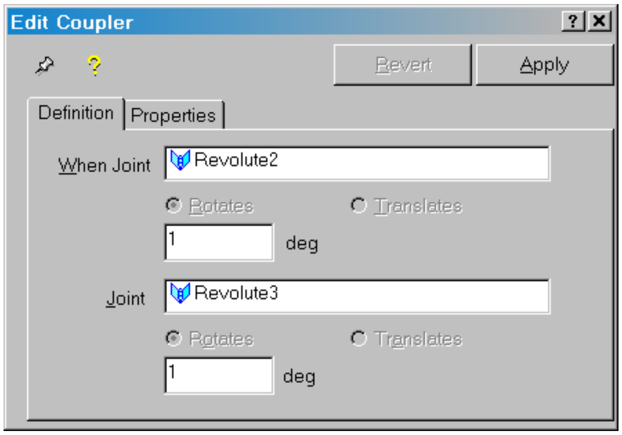
A single joint may be coupled to more than one joint. Again, the potential for locking does exist if the multiple couples conflict with each other.

A coupler removes one additional degree of freedom from the Motion Model.

A joint coupler may be created by one of the following methods:

- Choosing **Motion, Couplers** from the Menu.
- Selecting the **Couplers** branch in the Browser and choosing **Add Coupler** from the pop-up menu.
- Selecting one or two revolute, cylindrical, or translational joints in the Browser and choosing **Add Coupler** from the pop-up menu.

Regardless of the method used, the following dialog box will be displayed:



When Joint: Revolute2

● Rotates ○ Translates

1 deg

Joint: Revolute3

● Rotates ○ Translates

1 deg

When the first box is highlighted you can select a revolute, translational, or cylindrical joint from the Browser and its name will be entered in the box. If the selected joint is a revolute joint, the **Rotates** option button will be enabled. If the joint is a translational joint, the **Translates** option button will be enabled. If the joint is a cylindrical joint, then both option buttons will be enabled and you will have to choose which degree of freedom of the joint to couple.

When the second box is highlighted, you can selected a revolute, translational, or cylindrical joint from the Browser. The same rules apply in enabling the second set of option buttons.

In the two boxes that take numbers, you should specify the ratio between the two coupled joints. For example to couple two joints with a ratio of 2:1, enter 2 in the first box and 1 in the second box. To reverse the direction of one joint, enter a minus sign in front of the number.

Click **Apply** to create the coupler when all of the parameters have been specified.

Motion on Parts

A Motion on a Part consists of a point on a part with six (6) degrees of freedom. Each of the degrees of freedom can be driven with a motion generator. Each motion generator defined removes one degree of freedom.

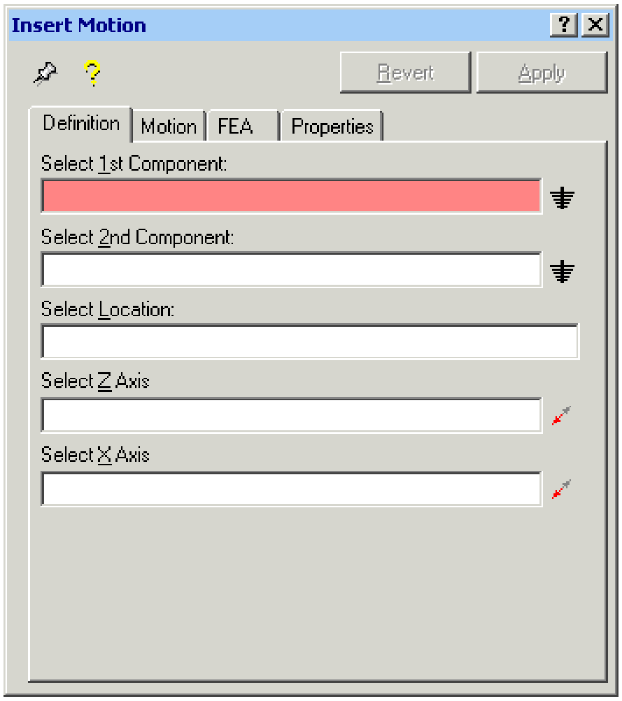
A Motion on Part constraint can be created by a number of methods:

- From the Motion menu by selecting **Motion, Motion…**
- From the Browser by clicking on the **Motions Constraints** branch and selecting **Add Motion on Part** from the pop-up menu.
- From the Browser by clicking on a part node and selecting **Add Motion, Motion on Part** from the pop-up menu..

To Create a Motion on Part Constraint:

1 Initiate the Motion on Part creation process by any of the methods listed above.

The Motion on Part dialog box appears.



2 Select the part where the motion will be applied. The part name should appear in the **Select 1st Component** box.

3 Select the part that the motion will be relative to. The part name should appear in the **Select 2nd Component** box.

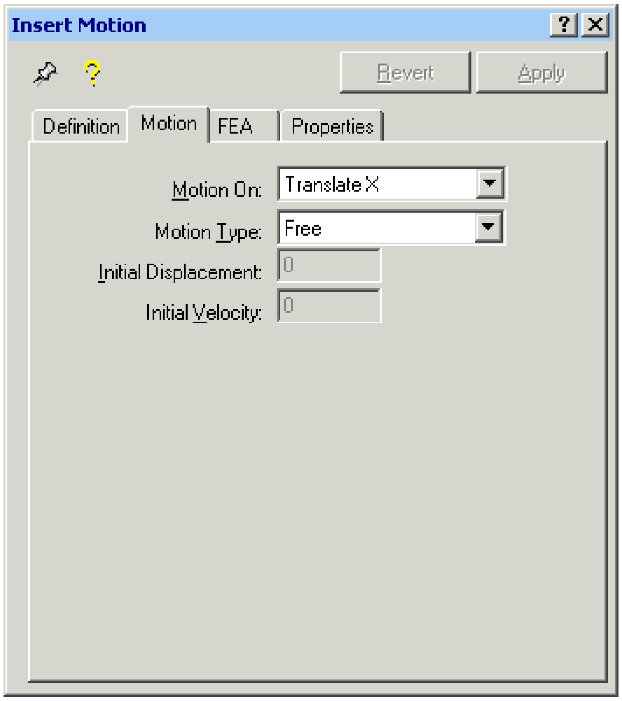
4 Select the location where the motion will act.

5 Select a direction that corresponds to the Z-Axis of the motion. At the point where the motion acts, a local coordinate system is also defined to designate the X, Y, and Z directions of motion. This sections determines the Z direction of the motion.

6 Select another direction, that is not collinear with the Z direction. This is designated as the X direction of motion. Because the coordinate system defined at the point of motion has to be orthogonal, the specified Z axis is crossed with this axis to obtain the Y-Axis, and then the Y-Axis is crossed with the Z axis to

obtain the real X-Axis. The only way to insure that the specified direction is really the X-Axis is to make it orthogonal with the specified Z-Axis.

7 Click the **Motion** tab to specify the motions generators for each direction, which displays the following page:



Motion On: Translate X

Motion Type: Free

Initial Displacement: 0

Initial Velocity: 0

Each direction may optionally have a motion generator applied. Select the direction from the **Motion On** combo box, and then specify the motion parameters. These parameters are the same as described on Page 42.

Rigid Bodies

In COSMOSMotion, a rigid body can be a single SolidWorks part or a sub-assembly. The term rigid body means that the individual components that make up the rigid body are rigidly attached to each other: during animation, they move together as they move with respect to other rigid bodies.

Ground

The ground rigid body is unique. It is by definition at absolute rest. All mechanisms have a ground rigid body. It serves as the reference frame for the remaining rigid bodies. All non-grounded rigid bodies in your mechanism move with respect to ground.

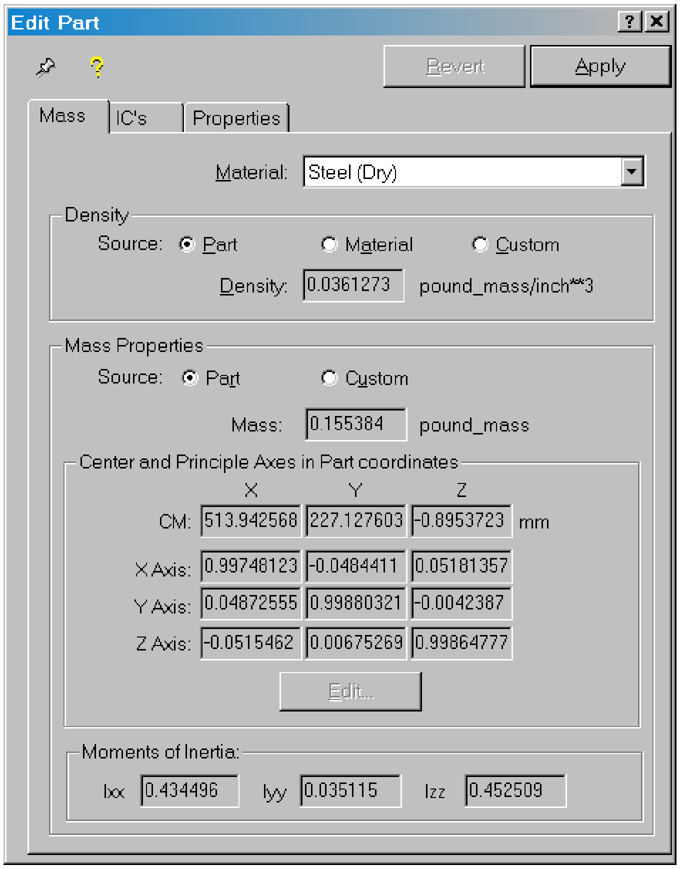
Mass properties

The mass properties by default are directly defined by the solids in your SolidWorks assembly.

Mass properties are important in COSMOSMotion because inertial loads effect reaction loads in joints that could be used as inputs for structural analysis.

Part Properties

Part properties, including mass and inertia, initial velocities, and display attributes of the mass properties symbol can be changed by selecting the part in the COSMOSMotion Feature Manager, and choosing the **Properties** selection from the pop-up menu. The following dialog will be displayed:



Material: Steel (Dry)

Density

Source: ● Part ○ Material ○ Custom

Density: 0.0361273 pound_mass/inch**3

Mass Properties

Source: ● Part ○ Custom

Mass: 0.155384 pound_mass

Center and Principle Axes in Part coordinates

	X	Y	Z	
CM:	513.942568	227.127603	-0.8953723	mm
X Axis:	0.99748123	-0.0484411	0.05181357	
Y Axis:	0.04872555	0.99880321	-0.0042387	
Z Axis:	-0.0515462	0.00675269	0.99864777	

Moments of Inertia:

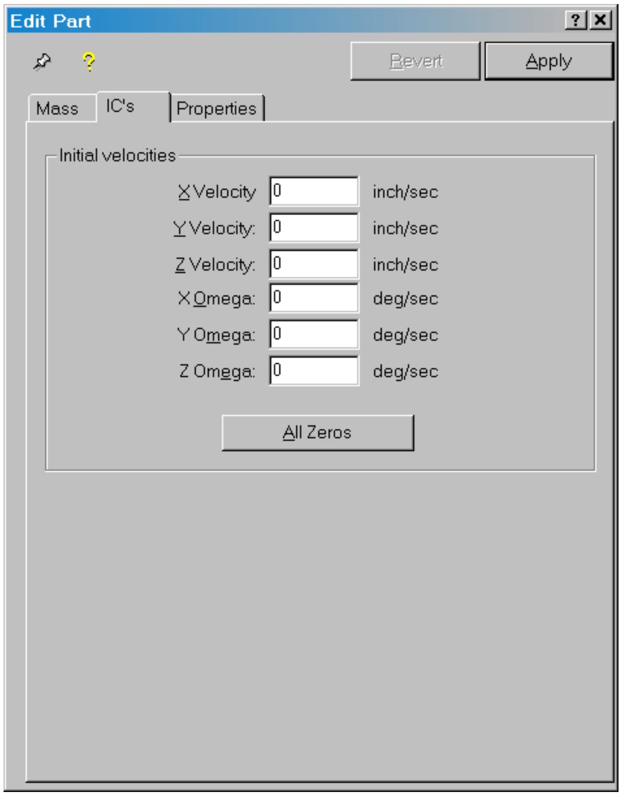
Ixx 0.434496 Iyy 0.035115 Izz 0.452509

The mass page allows the part mass properties to be modified. By default these properties come directly from SolidWorks.

The **Density** options determine what density is applied to the part volume calculated by SolidWorks to determine the actual part mass and inertia properties. The choices are **Part** where the density comes from the properties established by SolidWorks, **Material** where the density comes from the materials database and the selected COSMOSMotion material assigned to the part, or **Custom** which requires that the density value be specified in the edit box labeled **Density**.

The **Mass Properties** options allow the actual part mass properties to be set or changed. The **Source** option determines where the properties are derived from. Selecting **Part** indicates that the SolidWorks calculated values are used. Selecting **Custom** enables the remaining fields in the dialog which allow the Mass, Center of Mass, and Inertia properties to be specified. Clicking the **Edit** button displays another dialog that allows the Center of Mass and Axial Inertia properties to be changed.

The **IC** page allows the part initial velocities to be established.

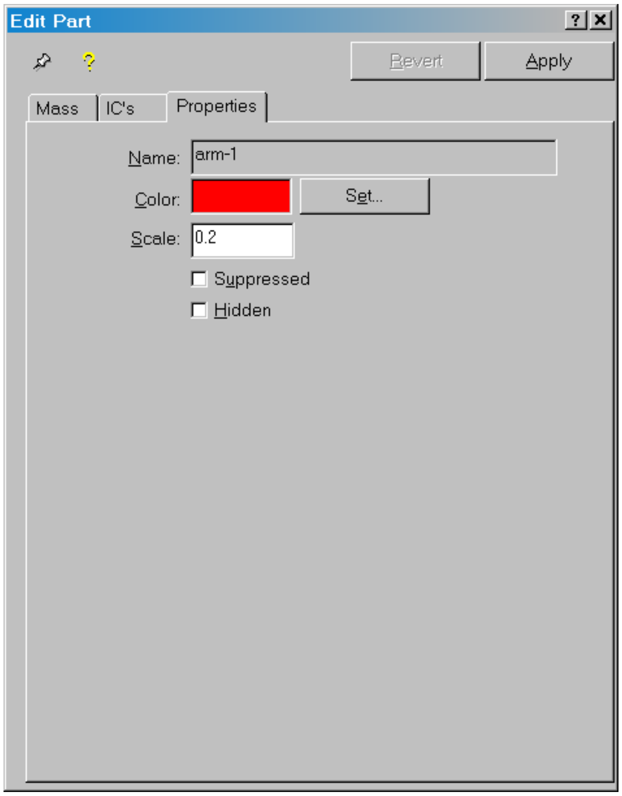


Initial velocities

X Velocity	0	inch/sec
Y Velocity:	0	inch/sec
Z Velocity:	0	inch/sec
X Omega:	0	deg/sec
Y Omega:	0	deg/sec
Z Omega:	0	deg/sec

These values are the velocities that part has at the very start of the simulation.

The **Properties** page allows the properties of the part to be set.



Name: arm-1

Color: 🟥

Scale: 0.2

□ Suppressed

□ Hidden

The mass property symbol attached to the part can have its color and scale set, and can be hidden. The part itself can be suppressed, which means it is not included in the simulation.

Forces

You can use forces to model spring and damping elements, actuation and control forces, and many other part interactions. This section introduces forces and explains how to create them.

Understanding Forces

Forces define loads and compliances on parts. Forces do not absolutely prohibit or prescribe motion. Therefore, they do not add or remove degrees of freedom from your model. Forces may resist motion, such as springs or dampers, or they may induce motion.

COSMOSMotion provides the following type of forces:

❑ **Applied Forces -** Applied forces are forces that define loads and compliances on parts so that they move in certain ways. Applied forces are very general, but you must supply your own description of the force behavior by specifying a constant force value or function expression. The applied forces in COSMOSMotion are the applied force, applied torque, action/reaction force, and action/reaction torque.

❑ **Flexible Connectors -** Flexible connectors resist motion and are simpler and easier to use than applied forces because you only supply constant coefficients for the forces. The flexible connectors include translational springs, torsion springs, translational dampers, and torsion dampers.

❑ **Gravity -** For more information on gravity and setting gravity, see the Gravity section below.

For every force that you create in COSMOSMotion, you specify the following information:

❑ Whether the force is translational or rotational.

❑ Which part or parts is the force applied.

❑ At what point or points is the force applied.

❑ Magnitude and direction of the force.

The following sections explain in more detail how to define the magnitude and direction of forces.

Force Magnitude

A force function specifies the force magnitude throughout your simulation. You can define force magnitudes in COSMOSMotion in the following ways:

❑ Enter values used to define stiffness and damping coefficients (springs and dampers only). In this case, COSMOSMotion automatically makes the force magnitude proportional to the distance or velocity between two points. The coefficients represent the proportionality constants.

❑ Enter a function expression using one of the pre-defined functions.

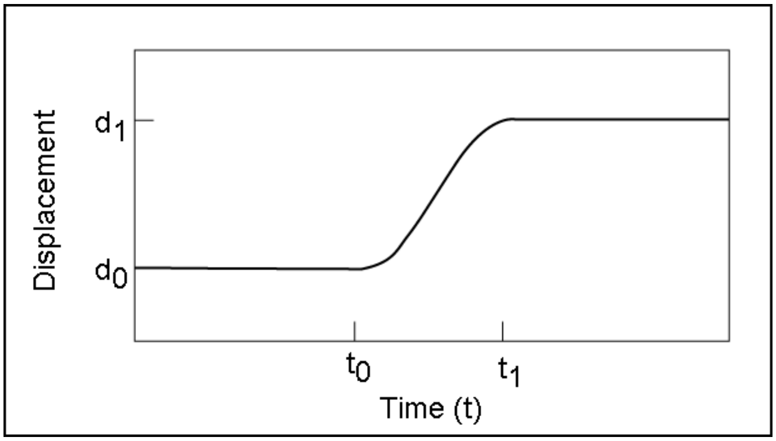
❑ Enter a function expression directly into the Function Expression field using the library of built-in MSC.ADAMS functions. See Chapter 6 for a list of supported functions.

The following sections describe the Template Editor and other function expressions.

Pre-Defined Functions

The following pre-defined functions are available to drive forces:

❑ **Step Function -** The Step Function creates a function expression that defines a force or torque that makes a smooth transition from one value to another. Before and after the transition, the force or torque value is constant.



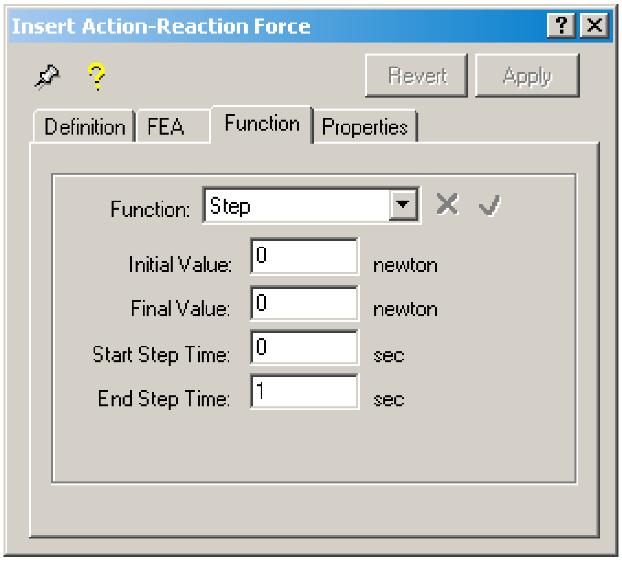
When you select Step from the Function Type list, additional fields are displayed that allow you to enter the values for the starting and ending times of the transition and the initial and final force or torque values. You should enter a real value for each of these terms. The input items are as follows:

Initial Value: The value of the function before step.

Final Value: The value of the function after step.

Start Step Time: The time at which step begins.

End Step Time: The time at which step ends.



Function: Step

Initial Value: 0 newton

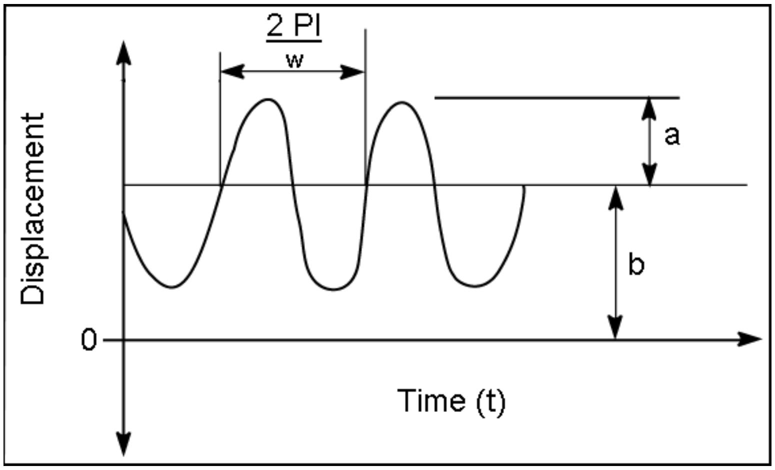
Final Value: 0 newton

Start Step Time: 0 sec

End Step Time: 1 sec

For more details on the Step function, see Chapter 6.

❑ **Harmonic Function -** The harmonic function creates a harmonically varying or sinusoidal force.



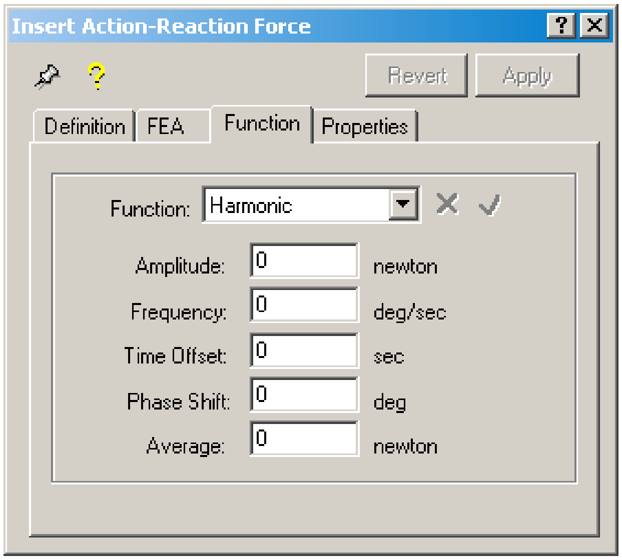
When you select the function type, additional fields are displayed that allow you to enter the values of the parameters for a simple harmonic function. Each of these prompt you for a real value. The input items are as follows:

Amplitude: Enter largest displacement from average value

Angular frequency: Enter speed of oscillation

Time Phase Shift: Enter starting time offset

Average Value: Enter the value about which the motion oscillates.



Function: Harmonic

Amplitude: 0 newton

Frequency: 0 deg/sec

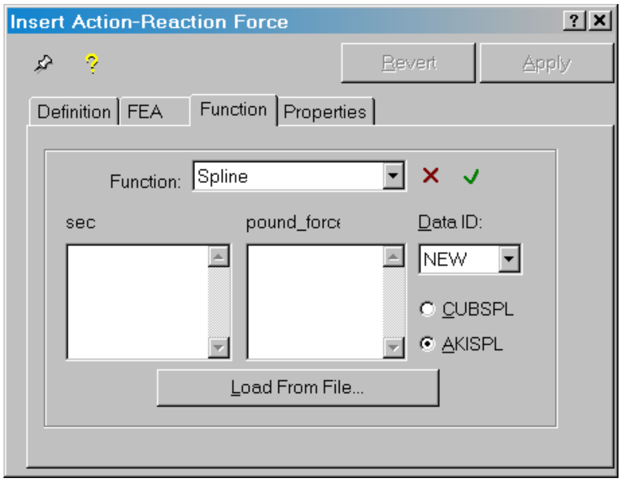
Time Offset: 0 sec

Phase Shift: 0 deg

Average: 0 newton

For more details on the Harmonic function, see Chapter 6.

❑ **Spline -** The Spline function allows the force value to be defined by entering a table of data. One column in the table is the time, and the other column is the value of the force at that time.



Function: Spline

sec

pound_forc

Data ID: NEW

○ CUBSPL

● AKISPL

The MSC.ADAMS/Solver fits a smooth curve to the data point values by interpolating the data points. The function expression for the motion then references this curve to get the actual position of the joint during the simulation. You must define at least four data points.

The Load From File button displays the standard Windows File Open dialog and allows the selection of a file that contains data points. The file can be free format, white space or commas can separate the columns of data. The only real restriction is that each line in the file contains one and only one set of data points.

The AKISPL and CUBSPL toggles control which kind of curve is used to fit your data points, Akima or Cubic. A cubic curve will work well even if your data points are not evenly spaced. An Akima curve is very fast, but may not work as well if your points are not evenly spaced.

You can create more than one set of data points and reference them between motions. You can control which set of points is being applied to the motion by using the Data ID list. The first time you create a curve the Data ID list will be set to “NEW”. If you create more than one set or modify existing sets, you can select from existing sets of data points or create a new one.

Function Expressions

In addition to using the pre-defined functions, you can also enter the text for an MSC.ADAMS function expression by selecting Expression from the **Function** combo-box. Placing a constant value in the Force Expression field creates a constant force. See Chapter 6 for more information on COSMOSMotion functions. The functions you enter can be dependent on system values besides time.

Force Direction

You can define force directions in COSMOSMotion in one of two ways:

- ❑ Along an axis defined by an edge or a plane.
- ❑ Along the line-of-sight between two points.

The way you define the directions of your forces depends on which forces you are creating. If you are creating an Applied Force, the force direction remains fixed with respect to some part in your model, either a moving part or the ground part. In this case, you can define the force direction using one vector defined by an edge or a plane.

If you are creating an Action/Reaction force, then the direction along which you want the force applied is defined by the line between two points in your model and is constantly changing throughout the simulation. In this case, you can define the force direction with as the line between the two points.

Creating Applied Forces

Applied forces are forces that define loads and compliances on parts so they move in certain ways. The applied forces supported in COSMOSMotion are Action-Only Force, Action-Only Moment, Action/Reaction Force, and Action/Reaction Moment. An Action-Only Force or Moment is applied at a point on a single rigid body, and no reaction forces are calculated. The Action/Reaction Force and Moment are applied between two points. The force or moment is applied to the first rigid body, and an equal but opposite reaction force or moment is applied to the second rigid body.

To create an Action-Only Force or Moment:

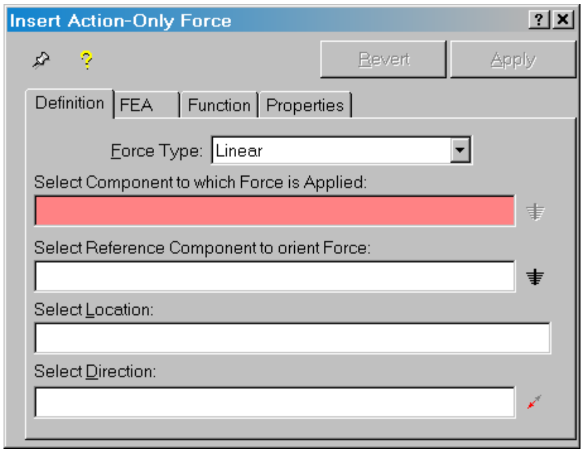
1 Initiate the process by any of the following methods:

- Select **Motion, Forces, Action-Only Force...** or **Motion, Force, Action-Only Moment** from the Menu.
- Select the **Forces** branch in the Browser and choose either **Add Action-Only Force**, or **Add Action-Only Moment** from the pop-up menu.
  The Insert Applied Force dialog box appears. You can easily change the

force type from a force to a torque and vice versa by using the **Force Type** list.

- Select the **Action-Only** branch in the Browser and choose either **Add Action-Only Force** or **Add Action-Only Moment** from the pop-up menu.
- Select one moving part in the Browser ad choose either **Add Force Action-Only Force** or **Add Force Action-Only Moment** from the pop-up menu.

2 No matter which method is chosen, the following dialog will be displayed:



Force Type: Linear

Select Component to which Force is Applied:

Select Reference Component to orient Force:

Select Location:

Select Direction:

3 The **Definition** panel should be visible when the dialog box opens. If not, click the **Definition** tab.

4 Select the component to which the force or torque is going to be applied. The component name appears in the Component Field.

5 Select a second component that will serve as a reference frame for the force. Click on the ⏚ button if you want the reference frame to be the global ground.

6 Click in the **Select Location** box if it is not already highlighted.

7 Select a vertex, circular edge, or linear edge to define the point at which the force or torque will be applied.

8 Click in the **Select Direction** box if it is not already highlighted.

9 Select an edge or a face to define the direction in which the force is acting or to define the axis about which the torque is acting.

The entity and component name appear in the Entity/Component field.

Click the ↙ button to switch the force or torque to the opposite direction.

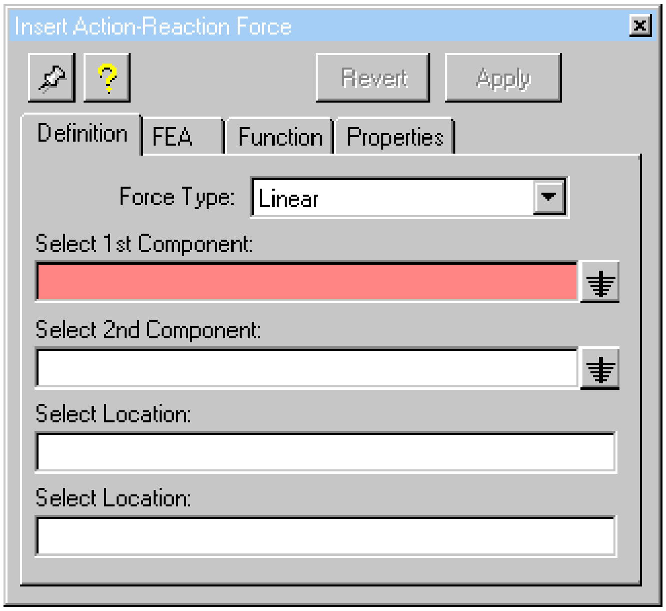
10. Click the **Function** tab to display the **Function** panel.
11. Select the Function type from the combo box and fill in the displayed fields. You can create Step or Harmonic functions, or you can enter data points.

    For more information on the functions see the Force Function section above.
12. Click **Apply** to create the force or torque and close the Insert Applied Force dialog box.

To create an Action/Reaction Force:

1. Initiate the process by one of the following methods:
    - Choose **Motion, Forces, Action/Reaction Force** from the Menu.
    - Select the **Forces** branch in the Browser and choose **Add Action/Reaction Force** from the pop-up menu.
    - Select the **Action/Reaction** branch in the Browser and choose **Add Action/Reaction Force** from the pop-up menu.
    - Select one or two moving or ground parts in the Browser and choose **Add Force Action/Reaction Force** from the pop-up menu.

2 No matter which method is used, the Action/Reaction Force dialog box is displayed:



Force Type: Linear

Select 1st Component:

Select 2nd Component:

Select Location:

Select Location:

3 Select the components to which the force is going to be applied.

The component names appear in the Component Field.

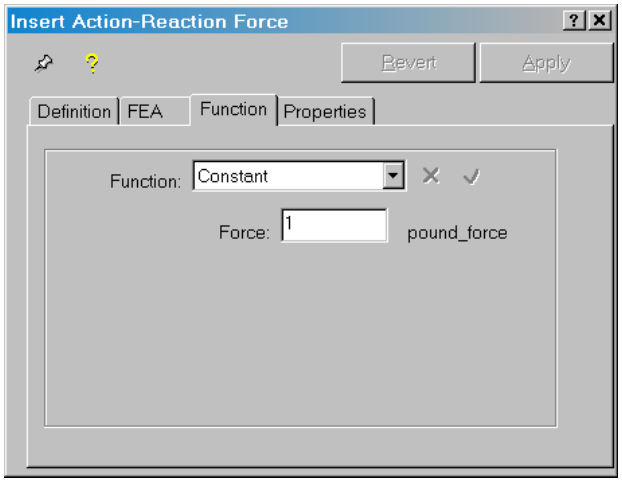
4 Click in the first **Select Location** box if it is not already highlighted.

5 Select a vertex, circular edge, or linear edge on each component to define the points at which the force is applied.

The entities and component names appear in the Entity/Component field.

Note: Select the points in the same order as you selected the components. In other words, verify that the first point that you select belongs to the first component selected and the second point belongs to the second component selected. If you select the points in the opposite order as the components, your mechanism may not behave as expected.

6 Click the **Function** tab.



Function: Constant

Force: 1 pound_force

7 Select the Function type from the combo box and fill in the displayed fields. You can create Step or Harmonic functions, or you can enter data points.

With the Template Editor, you can create Step or Harmonic functions, or you can enter data points. Select the desired function from the **Function Type** list. After you have filled in all of the fields for the function, click **OK** to complete the function and close the Template Editor dialog box.

The Template Editor inserts the function expression at the cursor in the Motion Expression field. If you have an expression in the Motion Expression field that is not highlighted, the Template Editor adds the new function to the existing expression. An expression that is highlighted in the Motion Expression field when you open the Template Editor will be replaced by the expression that you create in the Template Editor.

For more information on the functions see the Force Function section above.

8 Click **Apply** to create the force and close the Insert Applied Force dialog box.

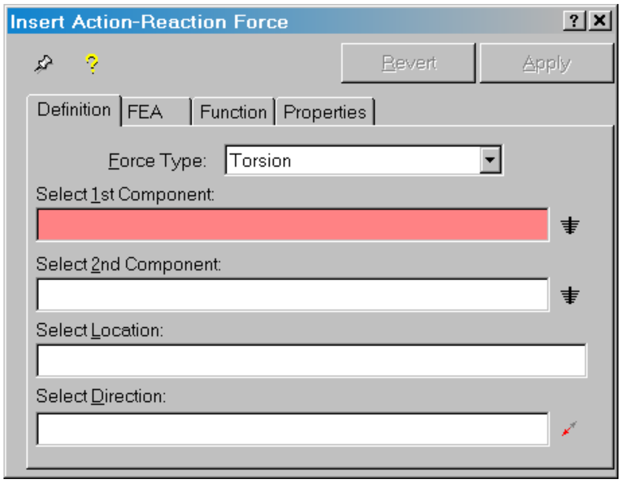
To create an Action/Reaction Moment:

1 Initiate the process by one of the following methods:

- Choose **Motion, Forces, Action/Reaction Moment** from the Menu.
- Select the **Forces** branch in the Browser and choose **Add Action/Reaction Moment** from the pop-up menu.

- Select the **Action/Reaction** branch in the Browser and choose **Add Action/Reaction Moment** from the pop-up menu.
- Select one or two moving or ground parts in the Browser and choose **Add Force Action/Reaction Moment** from the pop-up menu.

2 No matter which method is used, the Action/Reaction Moment dialog box is displayed:



Force Type: Torsion

3 Select the components to which the moment is going to be applied.

The component names appear in the Component Field.

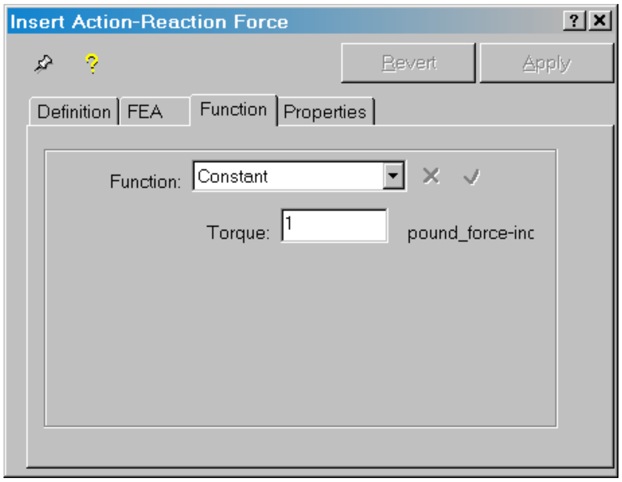
4 Click in the **Select Location** box if it is not already highlighted.

5 Select a vertex, circular edge, or linear edge to define the points at which the moment is applied.

6 Click in the **Select Direction** box if not already highlighted.

7 Select an edge or a face to define the axis about which the moment is applied.

8 Click the **Function** tab.



Function: Constant

Torque: 1 pound_force-inc

9 Select the Function type from the combo box and fill in the displayed fields. You can create Step or Harmonic functions, or you can enter data points.

For more information on the functions see the Force Function section above.

10 Click **Apply** to create the moment and close the Insert Applied Force dialog box.

Creating Bushings

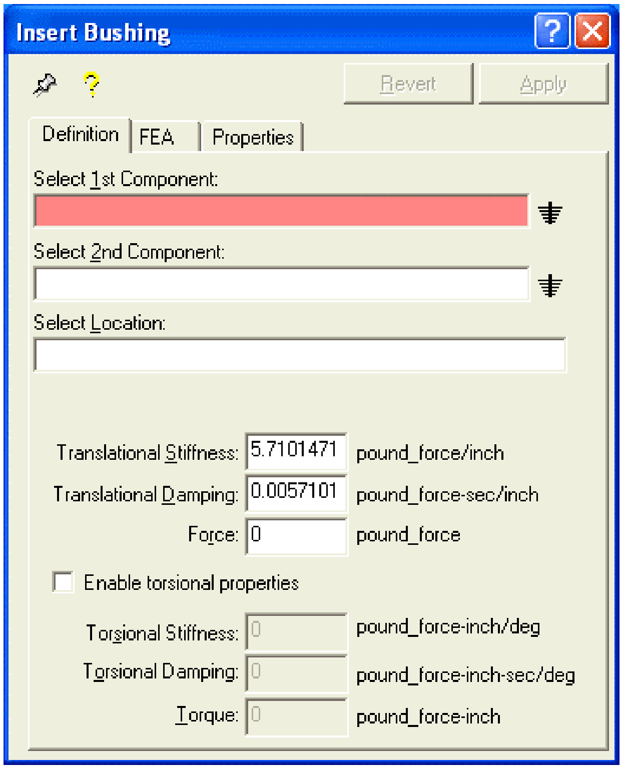
A bushing is an object that defines a force acting in all three directions. Optionally a torsional force can also be defined. This allows constraints such as joints to be replaced by a force based object.

To create a Bushing:

1 Initiate the process by one of the following methods:

- Select **Motion, Forces, Bushing...**
- Select the **Forces** branch in the Browser and choose **Add Bushing** from the pop-up menu.
- Select the **Springs** branch in the Browser and choose **Add Bushing** from the pop-up menu.
- Select one or two moving or ground parts, and choose **Forces Add Bushing** from the pop-up menu.

2 The Bushing dialog box will be displayed. The **Definitions** panel should be visible when the dialog box opens. If not, click the **Definitions** tab.



Select 1st Component:

Select 2nd Component:

Select Location:

Translational Stiffness: 5.7101471 pound_force/inch

Translational Damping: 0.0057101 pound_force-sec/inch

Force: 0 pound_force

☐ Enable torsional properties

Torsional Stiffness: 0 pound_force-inch/deg

Torsional Damping: 0 pound_force-inch-sec/deg

Torque: 0 pound_force-inch

3 Select the components to which the bushing is going to be attached.

4 Click in the **Select Location** box if not already highlighted.

5 Select a vertex, circular edge, or linear edge on the first component you selected to define the point at which the Bushing is attached.

The entity and component name appears in the Entity/Component field.

6 Modify the default Translational stiffness, damping and pre-load by clicking in the appropriate field and entering a new value.

7 Optionally modify the default Torsional stiffness, damping and pre-load by clicking in the appropriate field and entering a new value.

8 Click **Apply** to create the bushing and close the dialog box.

Creating Springs and Dampers

Translational springs and dampers represent forces acting between two parts over a distance and along a particular direction. You specify the location of the spring or damper on two parts. COSMOSMotion calculates the spring and damping forces based on the distance between the locations on the two parts.

A torsion spring or damper is a rotational spring or damper applied between two components.

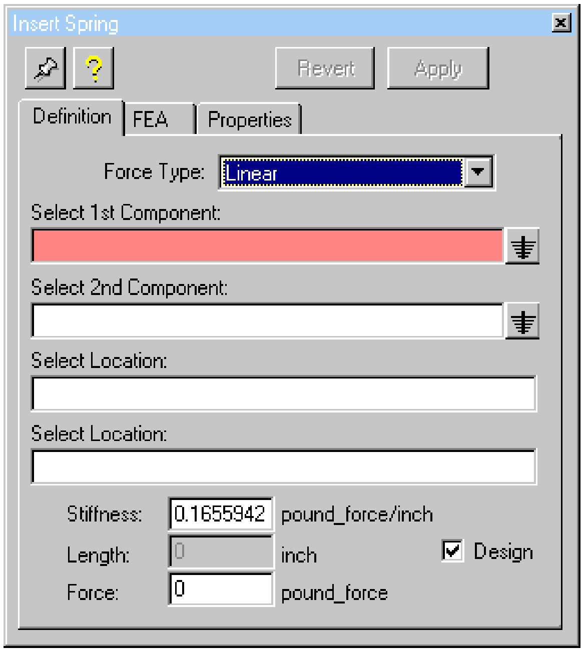
For both translational and rotational springs and dampers, COSMOSMotion applies the action force to the first part you select, called the action body, and applies an equal and opposite reaction force along the line of sight of the second part you select, called the reaction body.

To create a Spring:

9 Initiate the process by one of the following methods:

- Select **Motion, Forces, Spring...**
- Select the **Forces** branch in the Browser and choose **Add Linear Spring** from the pop-up menu.
- Select the **Springs** branch in the Browser and choose **Add Linear Spring** from the pop-up menu.
- Select one or two moving or ground parts, and choose **Forces Add Linear Spring** from the pop-up menu.

10 The Spring dialog box will be displayed. The **Definitions** panel should be visible when the dialog box opens. If not, click the **Definitions** tab.



Force Type: Linear

Stiffness: 0.1655942 pound_force/inch

Length: 0 inch ☑ Design

Force: 0 pound_force

11 Select the components to which the spring is going to be attached.

12 Click in the first **Select Location** box if not already highlighted.

13 Select a vertex, circular edge, or linear edge on the first component you selected to define the point at which the spring is attached.

The entity and component name appears in the Entity/Component field.

14 Click in the second **Select Location** box if not already highlighted and then select a vertex, circular edge, or linear edge on the second component you selected to define the other point at which the spring is attached.

Note: Verify that your first point is on the first component and the second point is on the second component. If you select the points out of order with the components, your spring will not behave as expected.

15 Modify the default spring stiffness by clicking in the Stiffness field and entering a new value.

16 By toggling the **Design** toggle off, you can define the free length of the spring. If you leave the Design toggle on, the free length of the spring is automatically defined to be the distance between the two points that you selected in the Points panel.

Note: The free length of the spring is the distance between those points in the initial position of the assembly.

17 You can define a preload on the spring by entering a value in the Force field.

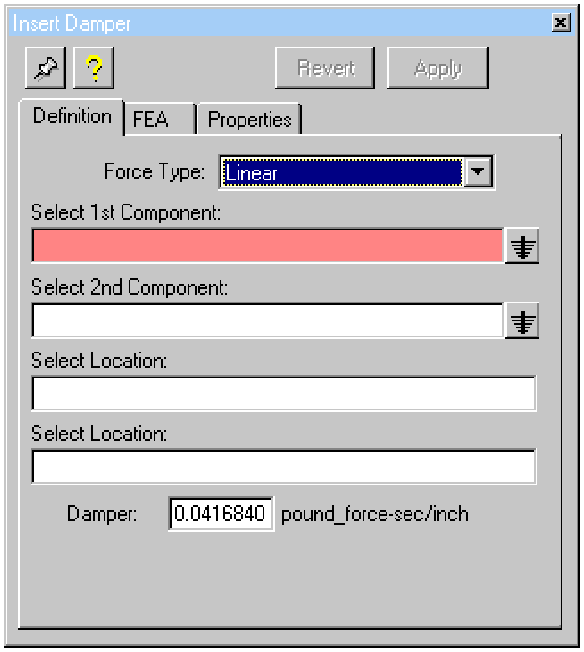
18 Click **Apply** to create the spring and close the Insert Spring dialog box.

To create a Damper:

1 Initiate the process by one of the following methods:

- Select **Motion, Forces, Damper…** from the Menu.
- Select the **Forces** branch in the Browser and choose **Add Linear Damper** from the pop-up menu.
- Select the **Damper** branch in the Browser and choose **Add Linear Damper** from the pop-up menu.
- Select one or two moving or ground parts, and choose **Forces Add Linear Damper** from the pop-up menu.

2 The Damper Dialog box will be displayed. The **Definition** panel should be visible when the dialog box opens. If not, click the **Definition** tab.



Force Type: Linear

Damper: 0.0416840 pound_force-sec/inch

3 Select the components to which the damper is going to be attached.

4 Click in the first **Select Location** box if not already highlighted.

5 Select a vertex, circular edge, or linear edge on the first component you selected to define the point at which the damper is attached.

6 Click in the second **Select Location** box if not already highlighted and then select a vertex, circular edge, or linear edge on the second component you selected to define the other point at which the damper is attached.

Note: Verify that your first point is on the first component and the second point is on the second component. If you select the points out of order with the components, your damper will not behave as expected.

7 Modify the default damping coefficient by clicking in the **Damping** field and entering a new value.

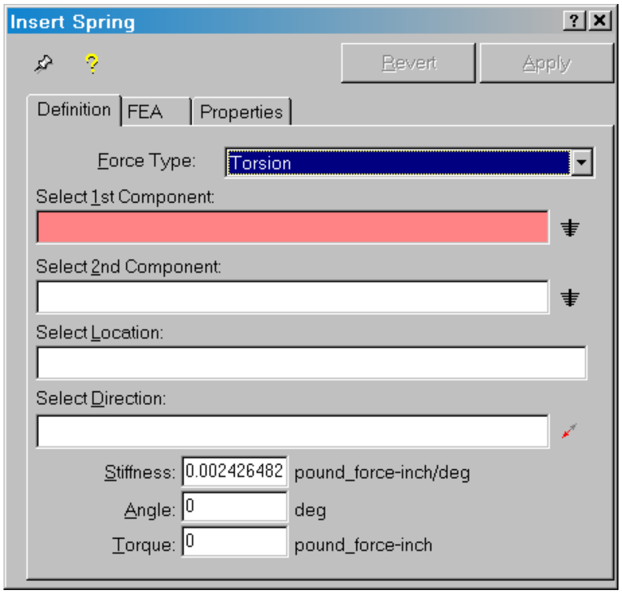
8 Click **Apply** to create the damper and close the Insert Damper dialog box.

To create a Torsion Spring:

1 Initiate the process by any of the following methods:

- Select **Motion, Forces, Torsion Spring...**
- Select the **Forces** branch in the Browser and choose **Add Torsion Spring** from the pop-up menu.
- Select the **Springs** branch in the Browser and choose **Add Torsion Spring** from the pop-up menu.
- Select one or two moving or ground parts, and choose **Forces Add Torsion Spring** from the pop-up menu.

2 The Spring dialog box will be displayed. The **Definition** panel should be visible when the dialog box opens. If not, click the **Definition** tab.



Force Type: Torsion

Select 1st Component:

Select 2nd Component:

Select Location:

Select Direction:

Stiffness: 0.002426482 pound_force-inch/deg

Angle: 0 deg

Torque: 0 pound_force-inch

3 Select the components to which the spring is going to be attached.

4 Click in the **Select Location** box if not already highlighted.

5 Select a vertex, circular edge, or linear edge to define the point at which the spring is attached.
The entity and component name appears in the Entity/Component field.

6 Click in the **Select Direction** box if not already highlighted.

7 Select an edge or a face to define the axis about which the spring is acting. The entity and component name appear in the Entity/Component field. Click the ↙ button to switch the spring to the opposite direction.

8 Modify the default spring stiffness by clicking in the **Stiffness** field and entering a new value.

9 You can define the preload angle of the spring by entering a value in the **Angle** field. By default, the angle is set to zero, the angle in the design position.

10 You can define a preload on the spring by entering a value in the **Torque** field.

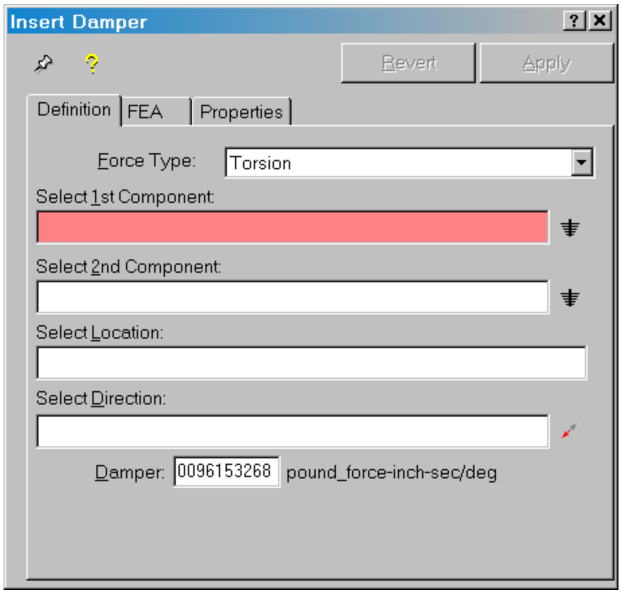
11 Click **Apply** to create the spring and close the Insert Spring Torque dialog box.

To create a Torsion Damper:

1 Initiate the process by one of the following methods:

- Select **Motion, Forces, Torsion Damper...**
- Select the **Forces** branch in the Browser and choose **Add Torsion Damper** from the pop-up menu.
- Select the **Springs** branch in the Browser and choose **Add Torsion Damper** from the pop-up menu.
- Select one or two moving or ground parts, and choose **Forces Add Torsion Damper** from the pop-up menu.

2 The Damper Dialog box will be displayed. The **Definition** panel should be visible when the dialog box opens. If not, click the **Definition** tab.



Force Type: Torsion

Damper: 0096153268 pound_force-inch-sec/deg

3 Select the components to which the damper is going to be attached.

4 Click in the **Select Location** box if not already highlighted.

5 Select a vertex, circular edge, or linear edge to define the point at which the spring is attached.

6 Click in the **Select Direction** box if not already highlighted.

7 Select an edge or a face to define the axis about which the damper is acting.

The entity and component name appear in the Entity/Component field. Click the ↙ button to switch the damper to the opposite direction.

8 Modify the default damping coefficient by clicking in the **Stiffness** field and entering a new value.

9 You can define the preload angle of the spring by entering a value in the **Angle** field. By default, the angle is set to zero, the angle in the design position.

10 Click **Apply** to create the damper and close the Insert Torsion Damper dialog box.

Creating Impact Forces

Impact forces are used to simulate the collision between two parts. As two parts approach within a specified distance, the impact force becomes active, and a force specified by the impact parameters will be applied to both of the colliding parts.

An impact force is controlled by the following five parameters:

Distance The distance from the contact point on the first link, to the force action point on the first part (the point that was graphically identified). This can be thought of as a sphere of radius <distance> centered at the action point of the force. During the simulation, when the distance between the impact point on the second part is equal to this parameter, then impact will occur.

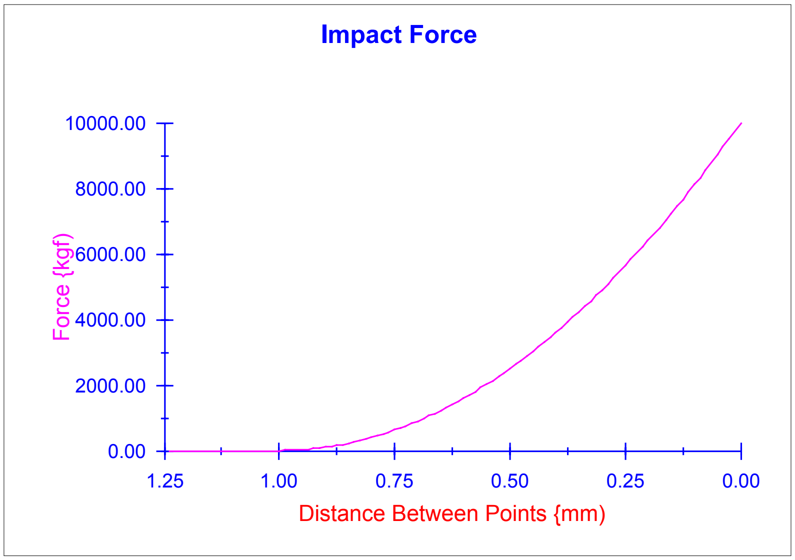
Stiffness This is the stiffness of the boundary interaction between the two parts. It should be set to a number that approximates the material stiffness of the two parts in collision.

Exponent The exponent of the force deformation characteristic. COSMOSMotion models the collision using an exponential force function.

Max Damping The maximum damping coefficient of the boundary interaction.

Penetration The boundary penetration distance where full damping will occur.

When the distance between the points is reached, and the impact force becomes active, it produces a force that is a function of the relative velocity of the two parts to which the force is attached. The graph below shows an example:

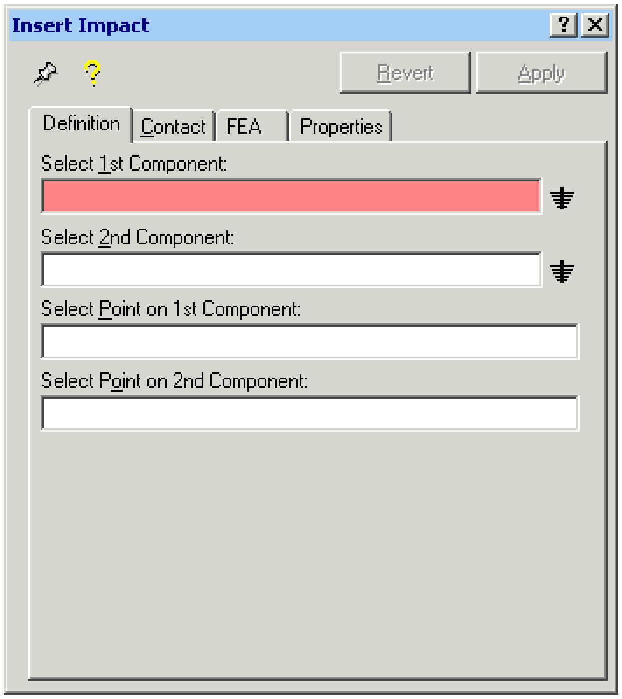


To create an Impact Force:

1 Initiate the process by one of the following methods:

- Select **Motion, Forces, Impact...**
- Select the **Forces** branch in the Browser and choose **Add Impact Force** from the pop-up menu.
- Select the **Springs** branch in the Browser and choose **Add Impact Force** from the pop-up menu.
- Select one or two moving or ground parts, and choose **Forces Add Impact Force** from the pop-up menu.

2 The Impact Force Dialog box will be displayed. The **Definition** panel should be visible when the dialog box opens. If not, click the **Definition** tab.



Insert Impact

Select 1st Component:

Select 2nd Component:

Select Point on 1st Component:

Select Point on 2nd Component:

3 Select the components to which the force is going to be applied.

The component names appear in the Component Field.

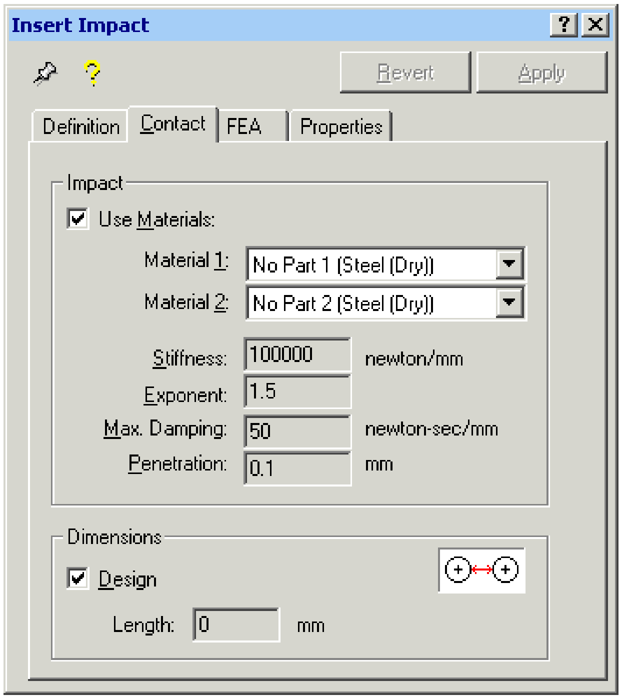
4 Click in the first **Select Location** box if it is not already highlighted.

5 Select a vertex, circular edge, or linear edge on each component to define the points at which the force is applied.

The entities and component names appear in the Entity/Component field.

Note: Select the points in the same order as you selected the components. In other words, verify that the first point that you select belongs to the first component selected and the second point belongs to the second component selected. If you select the points in the opposite order as the components, your mechanism may not behave as expected.

6 Click on the Contact tab to set the impact values.



Impact

☑ Use Materials:

Material 1: No Part 1 (Steel (Dry))

Material 2: No Part 2 (Steel (Dry))

Stiffness: 100000 newton/mm

Exponent: 1.5

Max. Damping: 50 newton-sec/mm

Penetration: 0.1 mm

Dimensions

☑ Design

Length: 0 mm

If the **Use Materials** box is checked, the two combo boxes labeled **Part 1 Material** and **Part 2 Material** are enabled. Selecting materials from these boxes will extract impact parameters from the materials database.

If the **Use Materials** box is not checked then impact parameters can be entered directly.

7 By toggling the **Design** toggle off, you can define the impact distance of the force. If you leave the Design toggle on, the impact distance is automatically defined to be the distance between the two points that you selected in the Points panel.

8 Click **Apply** to create the impact force and close the dialog box.

Gravity

Gravity is applied to your entire model uniformly. Gravity is defined by a value and a direction. By default, gravity is 9.81 meters per second squared (or the equivalent in other units) in the negative Y direction in the design coordinate system.

To Modify the Gravity Setting:

1. Use **Motion, Options** from Menu, or click the **Motion Model** branch in the Browser and choose **System Defaults** from the pop-up menu. Either method will open the COSMOSMotion Options dialog box. You can also open the Setting dialog box by clicking on the Options button on the Simulation tool bar, next to the Simulate button and directly above the DOF counter.
2. If the **World** panel isn't visible, click the **World** tab to display it.
3. You can modify the gravity settings by selecting one of the options in the **Acceleration** list. You can turn the gravity force off by selecting Off, you can set it to be the standard gravity value for earth by selecting Earth Standard, or you can set the acceleration to Custom.
4. If you set the Acceleration setting to Custom, you can enter your own gravity values in the X, Y, and Z fields below the Acceleration setting.
5. Click **OK** to modify gravity and close the COSMOSMotion Options dialog box.

Manipulating Mechanism Entities

You can easily modify or delete mechanism entities in COSMOSMotion.

Editing Mechanism Elements

You can edit any COSMOSMotion object. The parameters you can edit are usually the same as those you define when the object is created.

To Edit a COSMOSMotion element:

1. Select the element in the IntelliMotion Browser.
2. Choose **Properties** from the pop-up menu.

   The Edit panel appears.
3. Edit the attribute that you want to edit and click **Apply** to complete the edit and close the dialog box.
4. To replace a component, location or direction field for a motion entity, select the field (so that it is highlighted) and then press the “Delete” key on the keyboard. This will clear the field. Now pick the object to replace the once just cleared.

Deleting Mechanism Elements

The IntelliMotion Browser is used to delete Motion object from your mechanism.

To delete a mechanism element:

1. Select the element in the IntelliMotion Browser.
2. Choose **Delete** from the pop-up menu. Alternatively, you can also press the “Delete” key on the keyboard to delete the highlighted entities in the browser.

3

Materials

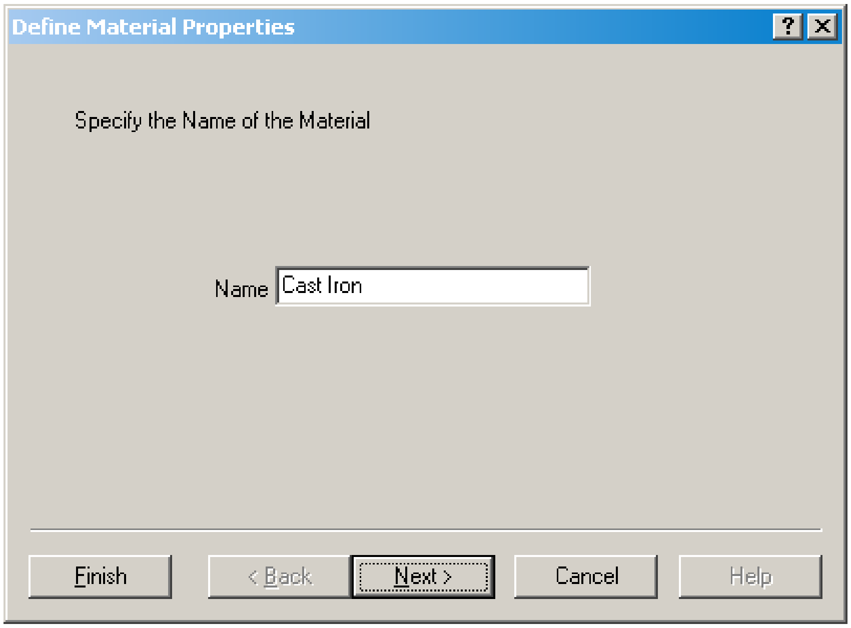
COSMOSMotion supports a materials database capability. The database is primarily intended to define interaction properties between material types, but it also contains some basic material property information. The interaction properties are used by impact forces, intermittent contact, and joint friction.

Adding Materials

The database supplied with COSMOSMotion contains property information for some common materials. It is also possible to add new material information to the database.

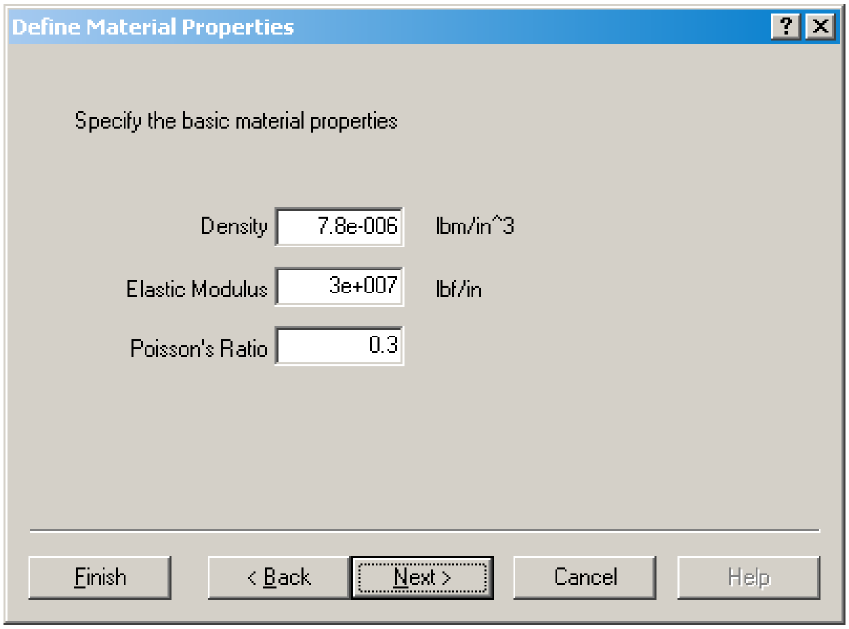
A new material can be added to the database by clicking on the **Motion Model** root node in the Browser and choosing **Materials->Add** from the pop-up menu.

The Add Materials wizard will be started. The first page in the wizard is:



Name: Cast Iron

The name of the new material being defined should be entered. If a name is entered that already exists in the database, an informational message will be issued. When a valid name is entered, the **Next>** button should be clicked which will display the following page:



Define Material Properties

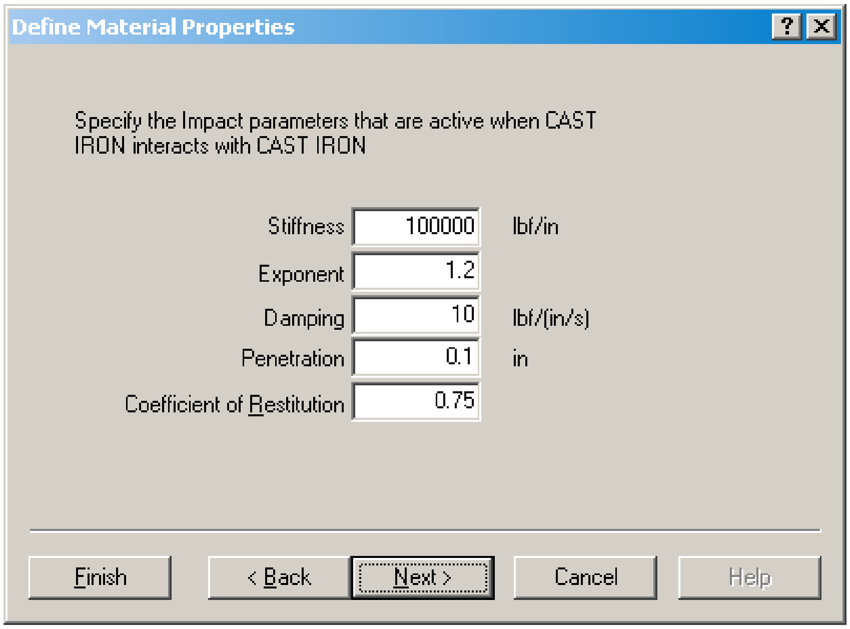
Specify the basic material properties

Density 7.8e-006 lbm/in^3

Elastic Modulus 3e+007 lbf/in

Poisson's Ratio 0.3

This page requires that some basic material property information be specified. The density, Elastic Modulus, and Poisson's Ratio should all be specified in the appropriate edit box. Clicking the **Next>** button will display the following page:

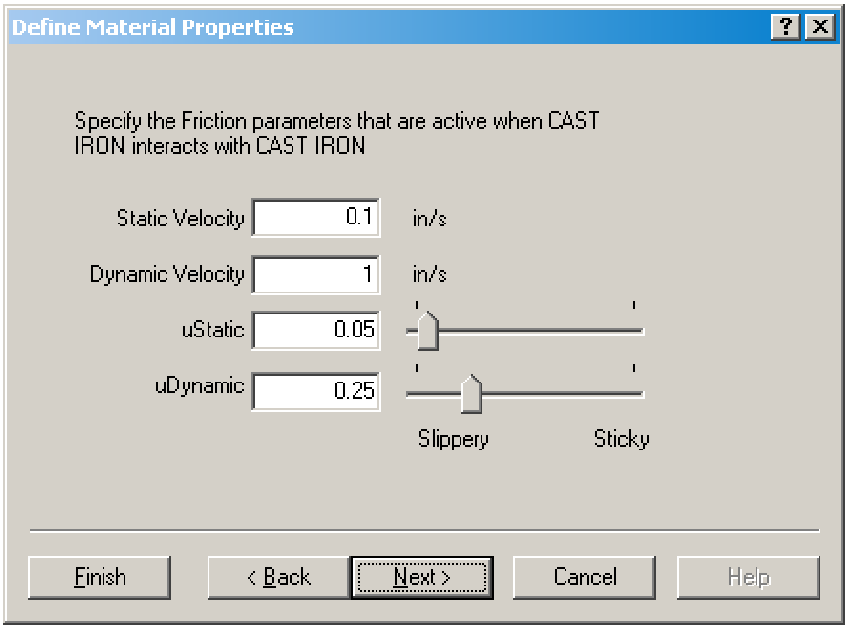

Define Material Properties

Specify the Impact parameters that are active when CAST IRON interacts with CAST IRON

Stiffness	100000	lbf/in
Exponent	1.2	
Damping	10	lbf/(in/s)
Penetration	0.1	in
Coefficient of Restitution	0.75	

This page starts a sequence of two pages that repeats for every material in the database. This page requests the value of the impact properties that are in effect when the new material interacts with another material already defined in the database. The names of the materials for which the interaction parameters are being requested are displayed at the top of the dialog box. Please see the section about Impact Forces on Page 103 for more information on the specific parameters. If no interaction properties are desired for the two listed materials, the **Next>** button can be clicked, which will move to the next material in the database.

If values are specified, clicking the **Next>** button will display the following page:



Define Material Properties

Specify the Friction parameters that are active when CAST IRON interacts with CAST IRON

Static Velocity 0.1 in/s

Dynamic Velocity 1 in/s

uStatic 0.05

uDynamic 0.25

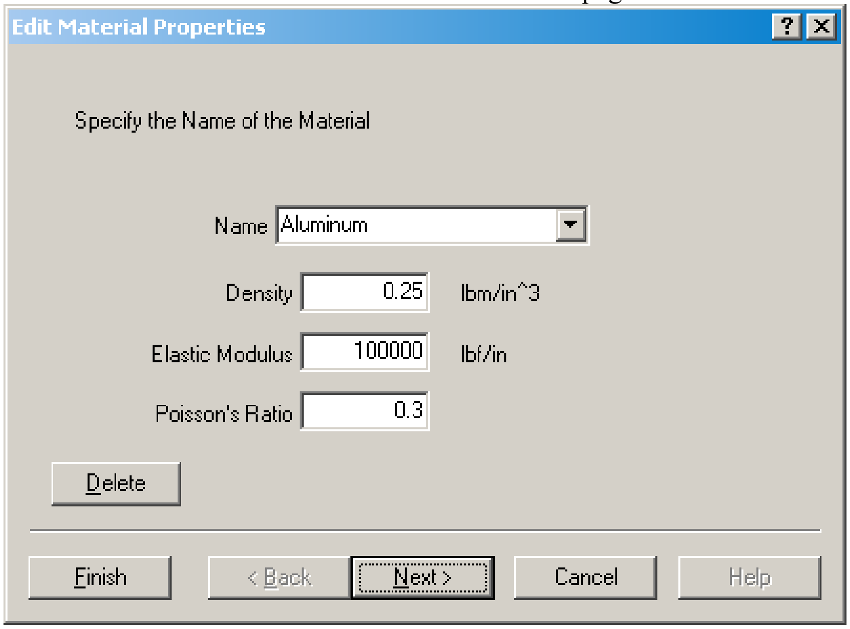
This page requests the friction parameters that are in effect when the two listed materials interact. See the section about Intermittent Contact on Page 66 for more information on the parameters.

These last two pages will repeat for each material in the database. The definition of interaction properties can be terminated by clicking the **Finish** button which will add the new material and any defined interaction properties to the database. If the **Cancel** button is clicked, the new material will not be added to the database.

Editing Materials

Existing material properties can be modified by clicking on the **Motion Model** root node in the Browser and choosing **Materials->Edit** from the pop-up menu.

The Edit Materials wizard will be started. The first page in the wizard is:



Edit Material Properties

Specify the Name of the Material

Name: Aluminum

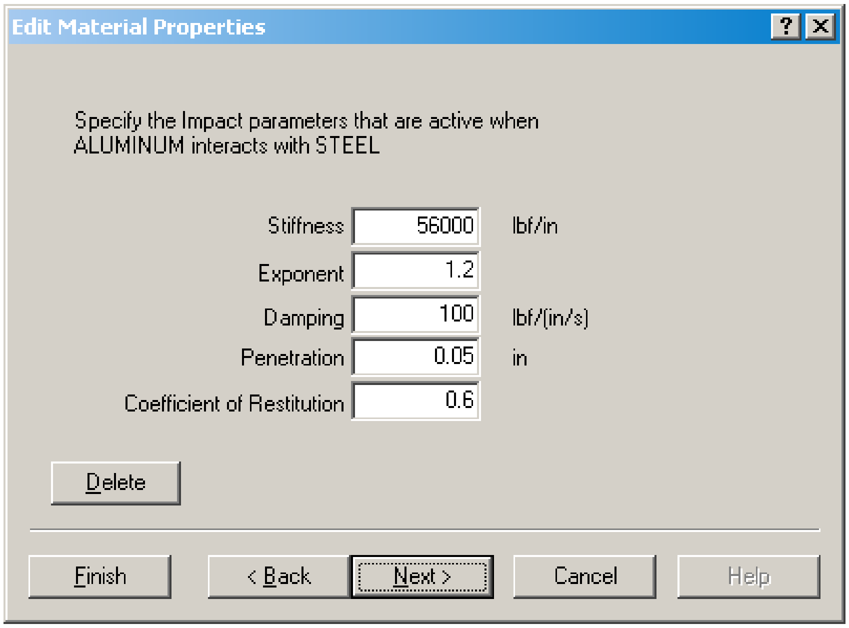
Density: 0.25 lbm/in^3

Elastic Modulus: 100000 lbf/in

Poisson's Ratio: 0.3

The name of the material to be edit should be selected from the combo box labeled "Name". If any of the basic properties need to be changed, new values can be specified from this page. Clicking the **Delete** button will remove the material, and any interaction properties that are defined for it, from the database.

To edit any defined interaction properties, click the **Next>** button which displays the following page:



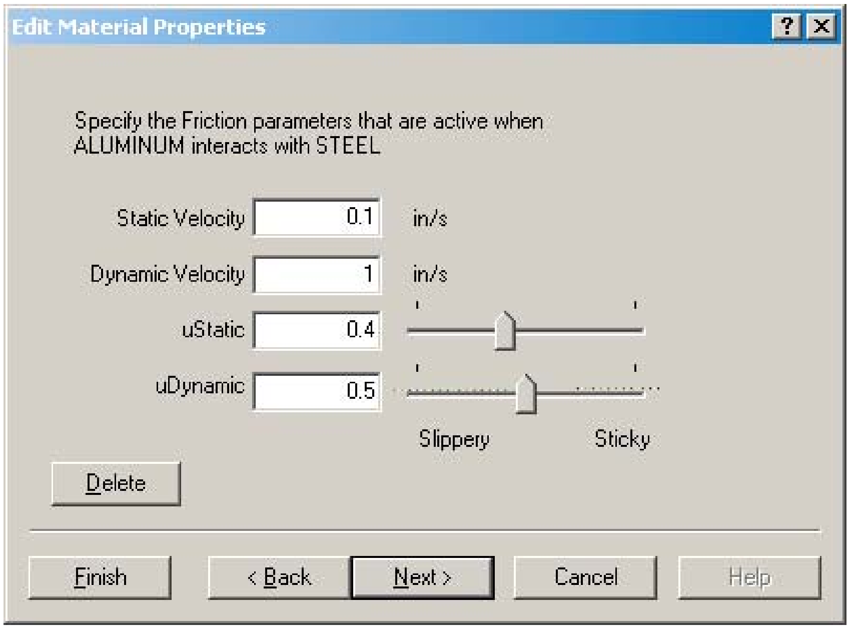
Edit Material Properties

Specify the Impact parameters that are active when
ALUMINUM interacts with STEEL

Stiffness	56000	lbf/in
Exponent	1.2	
Damping	100	lbf/(in/s)
Penetration	0.05	in
Coefficient of Restitution	0.6	

This page allows the interaction impact properties between the two materials listed at the top of the page to be modified.

When the new values have been specified, click **Next>** to display the following page:



Edit Material Properties

Specify the Friction parameters that are active when ALUMINUM interacts with STEEL

Static Velocity 0.1 in/s

Dynamic Velocity 1 in/s

uStatic 0.4

uDynamic 0.5

This page allows interaction friction properties between the two materials listed at the top of the page to be modified.

When the new values have been specified, click **Next>** to proceed to the next material in the database.

4

Mechanism Solution

Once you have created your mechanism by defining constraints, motions, and optionally forces, the next step is to simulate it so that you can observe the system’s behavior.

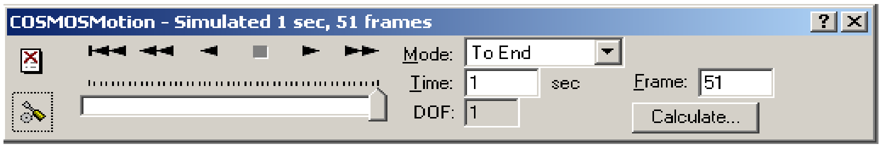
This chapter covers the following topics on simulating your mechanism:

- ❑ Simulation Panel
- ❑ Simulation Settings
- ❑ Running a simulation
- ❑ Simulation troubleshooting

Simulation Panel

One way to control the simulation of your mechanism is via the Simulation panel. This section describes the controls and information that you can use and retrieve from the toolbar.

Note: You might find it helpful to leave the Simulation panel open as you create your mechanism so that you can monitor the degrees of freedom in your system.


COSMOSMotion - Simulated 1 sec, 51 frames

Mode: To End

Time: 1 sec

DOF: 1

Frame: 51

Status

The status message line in the title of the Simulation panel indicates the status of your mechanism. It displays one of the following messages:

- ❑ No Active Mechanism - You do not have an open mechanism file.
- ❑ No Simulation - You have an active mechanism, but have not simulated it.
- ❑ Simulating... - Displayed when a simulation is running. Indicates the current simulation time and frame number.
- ❑ Simulated... - Displayed after a simulation has run and indicates the total simulation time and total number of frames [Example: Simulated 1 sec, 50 frames].
- ❑ Aborted - Indicates that the simulation ended abnormally before the end of the simulation time specified was reached.

DOF Counter

The automatic DOF counter keeps a running total of the degrees of freedom in your system. This count refers to the actual degrees of freedom in the system after the removal of redundant constraints unless the “Estimated” label appears after the counter. When the “Estimated” label is displayed, the DOF counter shows an approximated Gruebler count of the degrees of freedom, which does not compensate for redundant constraints. Pressing the Calculate button will employ the MSC.ADAMS Solver to calculate the actual degrees of freedom taking into account redundant constraints. A detailed Gruebler count and a list of redundancies will appear in the message window.

Calculate

The “Calculate…” button forces an update of the DOF counter and displays a detailed Gruebler count with a list of redundancies in the message window.

Button Controls and Scroll Bar

The arrow buttons on the Simulation panel are the animation playback controls. These buttons are, from left to right, Mechanism Reset, Fast Reverse, Reverse Play, Stop, Forward Play, and Fast Forward. The Mechanism Reset button can be used to reset the mechanism to its original configuration after a simulation or animation. You can also reset the mechanism by clicking **Motion, Reset**. Once the simulation has been started, the Stop button can be used to end the simulation. The other buttons can be used to animate the mechanism only after a simulation is run.

You can drag the scroll bar to move the mechanism to any frame in the simulation.

The animation playback controls are discussed in more detail in the following chapter “Reviewing Your Results”.

Mode

The Mode option list controls the animation mode. You can set the mode to:

- To End - The animation plays one time either forwards or backwards and stops.
- Loop - The animation plays to the end, jumps back to the beginning of the animation, and plays through again.
- Retrace - The animation plays forward to the end and then plays in reverse back to the beginning.

The Mode control is discussed in more detail in the following chapter.

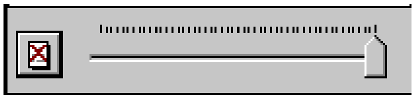
Time and Step

The Time and Step fields on the Simulation panel show the simulation time and frame that are currently displayed. While a simulation is running, these fields show the time and step that are currently simulating. If you would like to display the mechanism at a certain time or frame during the simulation, you can enter a value into the Time or Step fields and press Enter. The mechanism display will update to show the mechanism at that time or frame.

Simulating from the Browser

At the lower left corner of the IntelliMotion Browser, is the simulation button ▦. Clicking this button will start the simulation.

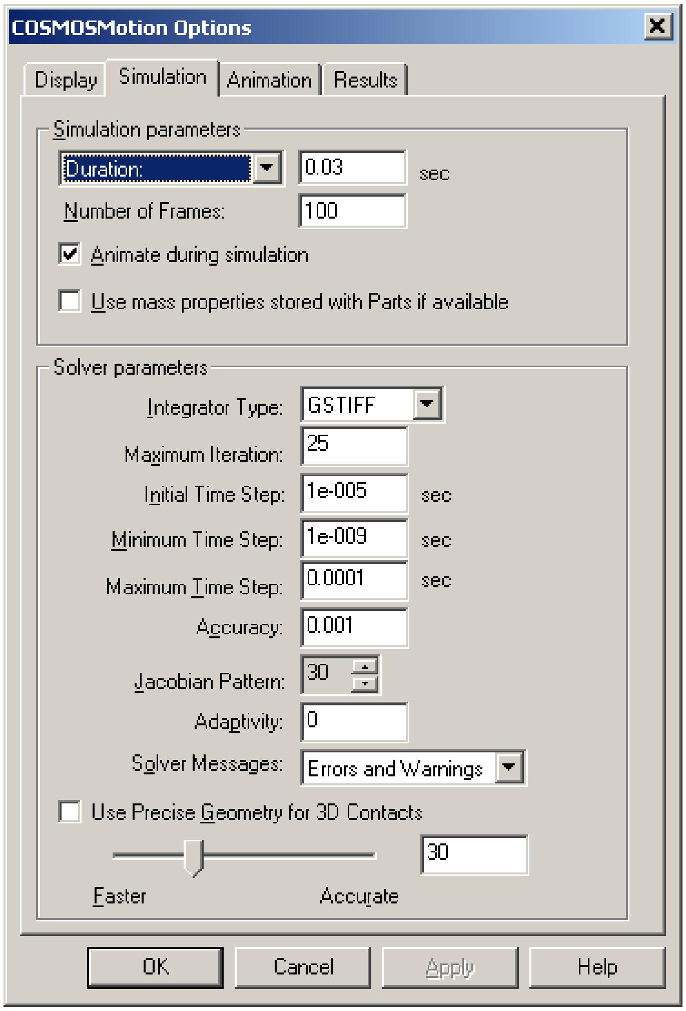
When simulation results already exist, the IntelliMotion Browser changes this button and adds a slider control:



The simulation button changes and now become the **Delete Results** button. Clicking the button will remove any simulation results and allows modifications and additions to be made the motion model.

Simulation Settings

Before you simulate your model, you can define several simulation settings. The simulation settings can be modified in the Options dialog box.



Simulation parameters

Duration: 0.03 sec

Number of Frames: 100

☑ Animate during simulation

☐ Use mass properties stored with Parts if available

Solver parameters

Integrator Type: GSTIFF

Maximum Iteration: 25

Initial Time Step: 1e-005 sec

Minimum Time Step: 1e-009 sec

Maximum Time Step: 0.0001 sec

Accuracy: 0.001

Jacobian Pattern: 30

Adaptivity: 0

Solver Messages: Errors and Warnings

☐ Use Precise Geometry for 3D Contacts

Faster — Accurate: 30

Number of Frames

This required parameter defines how many output steps the MSC.ADAMS/Solver creates. You can find this setting in the MSC.ADAMS Options dialog box; and it is, by default, set to 50 frames. You may use this parameter to control the "granularity" of your simulation: more frames give you a smoother animation while fewer frames may produce a coarse, jerky animation, depending on the speed of your mechanism.

The number of output steps per second that you specify should increase as the speed of your mechanism increases. If you don't request an adequate number of output steps, you might be missing some events in the behavior of your mechanism, and you might produce a jerky animation. On the other hand, if your mechanism moves very slowly, you may want to specify fewer output steps per second. With a slow moving mechanism, a large number of output steps per second may be unnecessary because events are happening more slowly.

Note: The number in the Number of Frames field is the total number of output steps for your simulation, not the number of output steps per second.

Note: Do not confuse the speed at which the mechanism moves with the animation rate! A mechanism's speed is based on how fast its components are moving or changing direction. The animation rate is the number of frames per second that are displayed and is dependent on your computer hardware.

Duration or Time Increment

This is a required parameter that specifies the length of time for a simulation or the time increment for each output step. You can find this setting in the Simulation Options dialog box; and it is, by default, set to 1 second. Note that the beginning time for a simulation is always time=0. A logical ending time is the time required for the mechanism to complete one cycle. However, you may use this parameter to focus on a particular ending time for your simulation. For example, you may have a robot arm that extends and retracts. If you are only examining the extension of the arm, you could set the simulation duration to the point where the arm is fully extended.

Animate during simulation

You can find the Animate during simulation button on the Simulate panel in the MSC.ADAMS Options dialog box. If you turn on the Animate during simulation setting, the mechanism animates as the simulation runs. If this setting is off, the simulation runs and the mechanism remains in its assembled position. Animating during a simulation provides instant feedback to you as your simulation runs.

Use mass properties stored with parts

If this option is checked, COSMOSMotion will use the last mass properties values it maintains for the parts during the simulation. If this box is cleared, COSMOSMotion will recalculate all part mass properties prior to running the simulation. Use this option if you have a large model that has not had any geometric modifications made recently. It will significantly increase performance. A good strategy is to leave this box cleared for the first simulation, and then check it for subsequent simulation as long as no geometric modifications have been made to any of the parts in the motion model that may have changed their mass properties.

Solver Parameters

The values listed in the **Solver Parameters** group are used to control the operation of the MSC.ADAMS/Solver. These parameters are explained below:

Integrator Type

MSC.ADAMS/Solver supports two integrator types. GSTIFF is the default and is faster and, in certain circumstances, slightly less accurate than the WSTIFF integrator.

Maximum Iterations

The “Maximum Iterations” parameter specifies the maximum number of times the numeric integrator iterates in the search for a solution for a given time step. If the Solver exceeds this limit, a convergence failure is recorded. The Solver will then quarter the last time step taken and integrate again. If four consecutive failures are recorded, the Solver stops with information on the probable cause. The default is 4 and should usually not be changed.

Initial Time Step

The “Initial Time Step” is useful to get the Solver started more quickly. It is the first step size used by the variable step integrator. If it results in a convergence error to be greater than one, a very conservative first step size is estimated by the solver. Resetting “Initial Time Step” to a less conservative value could get the solver started faster in subsequent runs of the model or slight variation of it.

Maximum Time Step

The “Maximum Time Step” should be large and is usually of little concern unless the solver is missing some short lived events like impacts. Then “Maximum Time Step” should be set to the same order as the short lived events. Integration time can be much longer because the Solver cannot move forward as fast.

Minimum Time Step

The "Minimum Time Step" should be small and is usually of little concern unless the model has a locked position which the Solver is trying to locate to a very high precision needlessly. Setting the parameter to a larger value will cause the solver to stop sooner once a locked position is found.

Accuracy

The "Accuracy" parameter specifies the criteria that is used to determine when the solution for the time step has converged. Larger value will cause the solver to run faster, but produce more "sloppy" results. Smaller values cause the solver to take longer, but result in more accurate results. The default value chosen is a good compromise for most problems.

Jacobian Pattern

The "Jacobian Pattern" parameter controls how often the simulation engine's jacobian matrix is updated. The value is in percent. At 100%, the jacobian is updated at every integration time step. At 50% the jacobian is updated every other integration time step. This value impacts models that contain intermittent curve/curve contact. The higher the value the better from an accuracy standpoint, but the higher the value, the more impact in the overall simulation time.

Adaptivity

The "Adaptivity" parameter is also used primarily for intermittent curve/curve problems. If problems occur where contacts are not detected, this value can be progressively decreased. A good starting point is 10e-6, with reductions of 10e-1.

Solver Messages

The "Solver Messages" option determines the level or message that are displayed during a simulation. The user can choose from:

- Errors and Warnings – Messages only for errors or warnings displayed.
- Errors only – Messages only for errors displayed.
- All – All messages displayed.

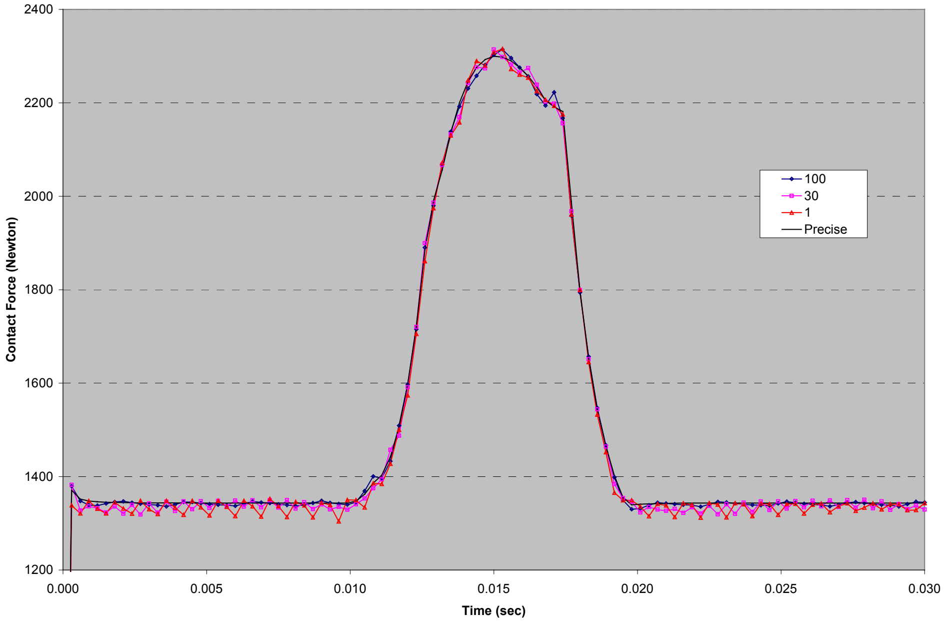
Use Precise Geometry for 3D Contact

If this option is checked, COSMOSMotion will use precise surface geometry to detect contact yielding more accurate solutions but also resulting in significantly longer solution times.

If this option is not selected the user can specify a scaling factor to adjust the facet tolerance used in the solver. The scale factor can be varied from 1 to 100. This factor is used to adjust several parameters, but the main parameter is the curvature tolerance for the facetted surfaces to the real geometry. A value of one give the least accurate representation and a value of 100 gives the best representation available.

Note that a value of 100 does not indicate that the geometry is precise, just that the facet settings are at a maximum accuracy.

The best way to use the facet tolerance is to minimize error in resulting force/acceleration results. For example, using 3-D contact with the valvecam tutorial provided with the software generates the following contact force results between the cam and the rocker:



Simulating

This section describes how to perform a simulation.

To perform a simulation:

1 Click the Browser Simulate button, or if you would like to use the simulation panel, open it by **Motion, Toolbar** from the menu.

The Simulation panel appears. If you have not already simulated your mechanism, the title bar should read “No Simulation.”

2 If you would like to modify the **Number of Frames** or the **Duration/Time Increment** settings, click the **Options** button on the Simulation panel, directly above the DOF counter.

The MSC.ADAMS Options dialog box appears.

3 Click the **Simulate** tab.

4 You can modify the **Number of Frames**, **Duration/Time Increment**, or **Animate during simulation** settings.

5 Click **OK** to complete the settings modification and close the MSC.ADAMS Options dialog box.

6 Click the **Simulate** button on the Simulation panel to begin the simulation.

If you toggled off the **Animate during simulation** setting, the scroll bar on the toolbar advances as the simulation advances. Also the Time and Step fields show the current simulation time and frame. If the **Animate during simulation** toggle is on, the mechanism animates as the simulation advances.

7 After the simulation is complete, you can click the Forward Play, Reverse Play, Fast Forward, Fast Rewind, or Reset buttons to animate or reset the mechanism.

Simulation Troubleshooting

The following are frequently asked simulation questions and answers.

1 **Q**: The reaction forces in the joints that are calculated are incorrect by quite a bit, over a factor of ten. What might I be doing wrong?

A: First check that you have properly set the units of your mechanism. Also check that you have properly set the density of each of your components. To check the density of a component, select **Tools, Options** and then select the **Material Properties** tab while viewing the component (not the assembly). Incorrect settings of units or density can affect the reaction forces significantly.

2 **Q**: The standard (built-in) animation runs slowly on my PC. Is there a way to get fast, shaded animation?

A: Yes. You can save mechanism animation to an ".AVI" file. See "Exporting an .Ave movie" on page 131 for more information.

3 **Q**: How do I get a plot of my output data?

A: You can export your output data to Excel or to a text file. See Exporting Results to Excel on page 134 or Exporting Results to a Text File on page 138 for more information.

5

Reviewing Your Results

After you have simulated your mechanism, you can review the results in several ways. You can play animations or review the numerical data and plots in Excel. Animations can also be exported to AVI movies and VRML format. This chapter describes the following:

- ❑ Animating in SolidWorks
- ❑ Exporting animations to AVI movies
- ❑ Exporting animations to VRML
- ❑ Exporting results to Excel
- ❑ Exporting results to a text file
- ❑ Interference Detection
- ❑ Exporting Force results to CosmosWorks

Note: All these actions can also be accessed using the motion toolbar icons.

Animating The Mechanism

Animation of a software prototype of a mechanism can help you predict the mechanism's behavior and refine its design prior to manufacturing. You can animate your mechanism during or immediately following a simulation.

One frame of animation is created for each output step.

Playback Controls ⏮ ⏪ ◀ ■ ▶ ⏩

The animation playback controls are located on the Simulation panel, and are also explained in the previous chapter.

These buttons are, from left to right:

❑ Mechanism Reset - Resets the mechanism to its original configuration after a simulation or animation.

❑ Fast Rewind - Plays the mechanism in the reverse direction but skips frames so that the animation is faster than regular play. You can control the number of skipped frames in the MSC.ADAMS Options dialog box.

❑ Reverse Play - Plays every frame of the animation in the reverse direction.

❑ Stop - Stops a simulation or animation.

❑ Forward Play - Plays every frame of the animation in the forward direction.

❑ Fast Forward - Plays the animation in the forward direction, skipping frames so that the animation is faster than Forward Play. You can control the number of skipped frames in the MSC.ADAMS Options dialog box.

As mentioned above, the Mechanism Reset button resets the mechanism to its original configuration. You can also reset the mechanism by clicking **Motion, Reset**. Once a simulation has been started, the Stop button can be used to end the simulation. The Stop button also stops an animation. The other buttons can be used to animate the mechanism only after a simulation or portion of a simulation is run.

You can drag the scroll bar to move the mechanism to any frame in the simulation.

Mode

The Mode option list controls the animation mode. You can set the mode to:

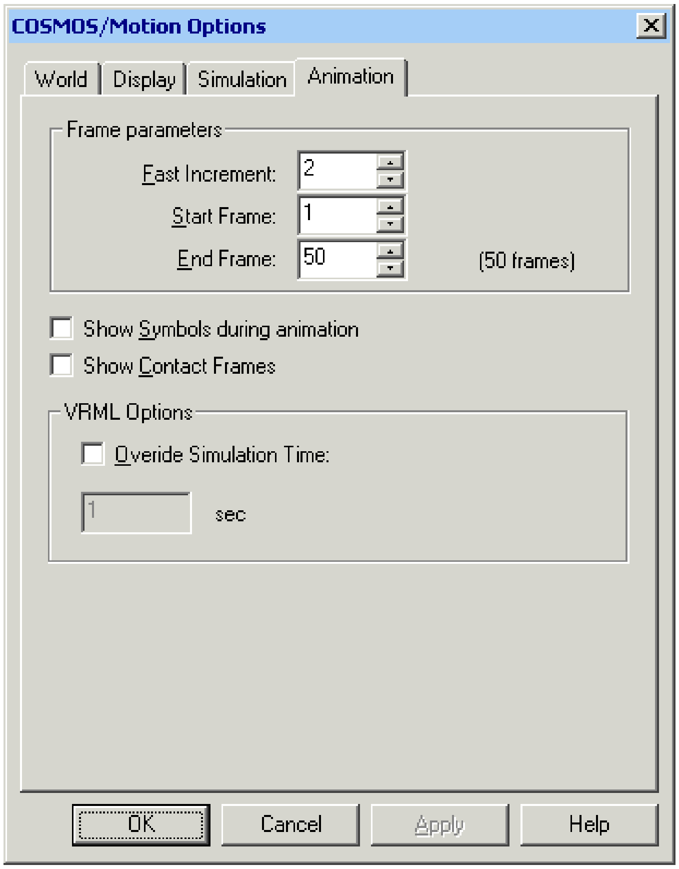
❑ To End - The animation plays one time either forwards or backwards and stops.

❑ Loop - The animation plays to the end, jumps back to the beginning of the animation, and plays through again. The animation plays until you press the Stop button.

❑ Retrace - The animation plays forward to the end and then plays in reverse back to the beginning. The animation plays until you press the Stop button.

Animation Settings

The animation settings are located on the Animation panel in the MSC.ADAMS Options dialog box . These settings control some aspects of the playback controls.



COSMOS/Motion Options

Frame parameters

Fast Increment: 2

Start Frame: 1

End Frame: 50 (50 frames)

□ Show Symbols during animation

□ Show Contact Frames

VRML Options

□ Overide Simulation Time:

1 sec

Fast Increment

The Fast Increment setting specifies the number of frames to skip between each animation frame for Fast Forward and Fast Rewind. This allows you to speed up the viewing of motions that take a large number of simulation steps to analyze. For example, if you wish to animate every other frame, you can enter Ò2Ó in the Fast Increment field. You can enter “5” in the field to display only every fifth frame.

Start Frame, End Frame

You can control how much of the animation is played by using the Start Frame and End Frame fields. The Start Frame and End Frame settings allow you to specify the frame numbers at which to start and stop the animation. This is particularly useful if you wish to view only a subset of a long animation.

The first frame always shows the mechanism in its assembled state, and the second frame is the first frame of the animation. The last frame is therefore numbered 1 greater than the number of steps in the simulation. For example, if you perform a simulation with 10 output steps, the frames are numbered from 1 to 11.

Show Symbols during animation

When this option is checked, COSMOSMotion object symbols are shown during animation. When this option is clear, the symbols are not displayed.

Show Contact Frames

When this option is checked, COSMOSMotion will show any additional frames associated with contact elements. The MSC.ADAMS/Solver will output frames at points when contact forces are significant in addition to the requested number of output frames. When this option is clear, these additional frames will be skipped.

Animating a Simulation

In order to animate a mechanism, you must first simulate it. Nearly all of the simulation and animation controls are located on the Simulation panel. The other simulation and animation settings can be found on the MSC.ADAMS Options dialog box.

To animate a mechanism:

1. If the Simulation panel is not open, click Motion, Toolbar.
2. Simulate the mechanism by clicking the calculator icon on the upper left corner of the toolbar.

   See Chapter 5, “Simulating a Mechanism” for more information on simulating.
3. After the simulation has finished, press any of the arrow buttons to animate the mechanism forward, backward, fast forward, or fast rewind. Press the stop button to stop an animation or a simulation.

4 If you would like to change the animation settings, open the Options dialog box by clicking the options icon on the Toolbar, which is directly above the Degree of Freedom Counter.

5 Click the **Animation** tab.

6 Modify any of the animation settings and click **OK** to close the dialog box.

7 You can continue to animate your mechanism by pressing the arrow buttons.

Exporting an AVI movie

If the animations run slowly on your PC, you can create faster animations by converting them into AVI movies.

To create an AVI movie from your animation:

1 If you have not already simulated the model, do so now.

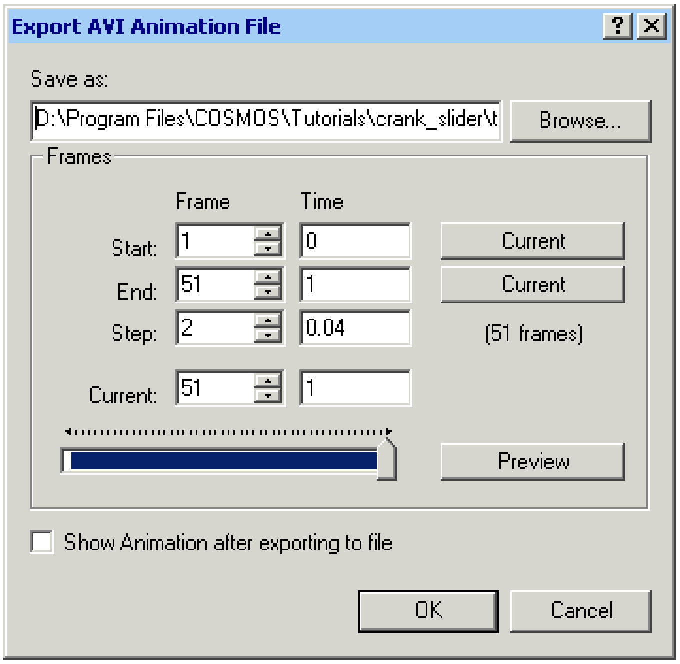
For more information on simulating the mechanism, see Chapter 4, “Simulating Your Mechanism”.

2 If your SolidWorks assembly window is covering most of the screen, you may want to make it smaller. The size of the AVI movie file is proportional to the size of your assembly window. If you try to save an AVI file of an assembly window that covers most of the screen and has many output frames, you may run out of disk space.

Note: Resize the assembly window before you begin creating the AVI file. You must not resize the assembly window while the AVI file is being created!

Note: If the Simulation panel is overlapping your assembly window, it will be included in the AVI movie. If you wish to avoid this, move the toolbar so that it is not covering any part of the assembly window.

3 Click **Motion, Export Results, To AVI Movie...** or select the **Motion Model** in the COSMOSMotion feature manager and select **Export AVI Movie** from the pop-up menu. You can also use the export AVI toolbar button 🎞**.**The following dialog will be displayed:



Export AVI Animation File

Save as: D:\Program Files\COSMOS\Tutorials\crank_slider\t

Frames

	Frame	Time
Start:	1	0
End:	51	1
Step:	2	0.04
Current:	51	1

(51 frames)

☐ Show Animation after exporting to file

4 Select the start and end frames and the frame increment, or the start and end times and the time increment. The buttons labeled **Current** can be used to set either the start or end frame to the current frame.

The **Preview** button will display the frames that will be included in the animation. The **Show Animation after exporting to file** will replay the animation after it has been generated.

5 Click **OK**.

The mechanism animates through its range of motion as the AVI movie is created.

Exporting Animations to VRML

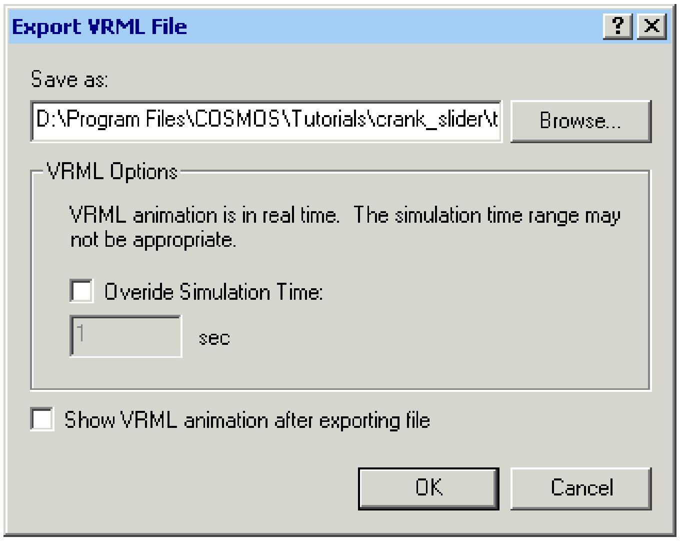
If you would like to place your animation on your website or your intranet, you can easily export your animations in VRML format. You can download a plug-in for your Web browser called Cosmo Player to view VRML files. This plug-in can be downloaded from: http://cosmo.sgi.com/.

Before exporting the animation, you may need to modify the animation duration for the VRML movie. The default animation duration is the simulation time. The Cosmo Player will attempt to squeeze the animation into the simulation time and may skip frames as a result. You can avoid this situation by modifying the Override Simulation Time setting in the MSC.ADAMS Options dialog box.

To Export an Animation to VRML:

1. Click **Motion, Options.**
2. Click the Animation tab.
3. On the Animation panel, toggle on the Override Simulation Time setting.
4. Set the Animation Time so that the animation will run smoothly without skipping frames. For example, for an animation with 50 frames, you should probably set the Animation Time to approximately 10-15 seconds.
5. Click OK.

6 Click **Motion, Export Results, To VRML...** or select the **Motion Model** in the COSMOSMotion feature manager and select **Export VRML** from the pop-up menu. You can also use the toolbar button. The following dialog will be displayed:



Export VRML File

Save as:

D:\Program Files\COSMOS\Tutorials\crank_slider\t

VRML Options

VRML animation is in real time. The simulation time range may not be appropriate.

□ Overide Simulation Time:

1 sec

□ Show VRML animation after exporting file

7 Enter a VRML file name, or click **Browse** to specify it using the standard Windows Files Save As dialog. Click the **Override Simulation Time** option to enter a new value for the total replay time for the VRML file. Click the **Show VRML animation after exporting to file** if you want to replay the VRML animation after generating it. You must have an appropriate VRML player installed on your system to use this option.;

8 Click **OK**.

Exporting Results to Excel

You can easily export your simulation results to Excel. You can export rigid body displacement, velocity, and acceleration results along with force results for motions and joints. In Excel, you can view the data in graphs or tables.

Note: You must have at least Excel 97 installed.

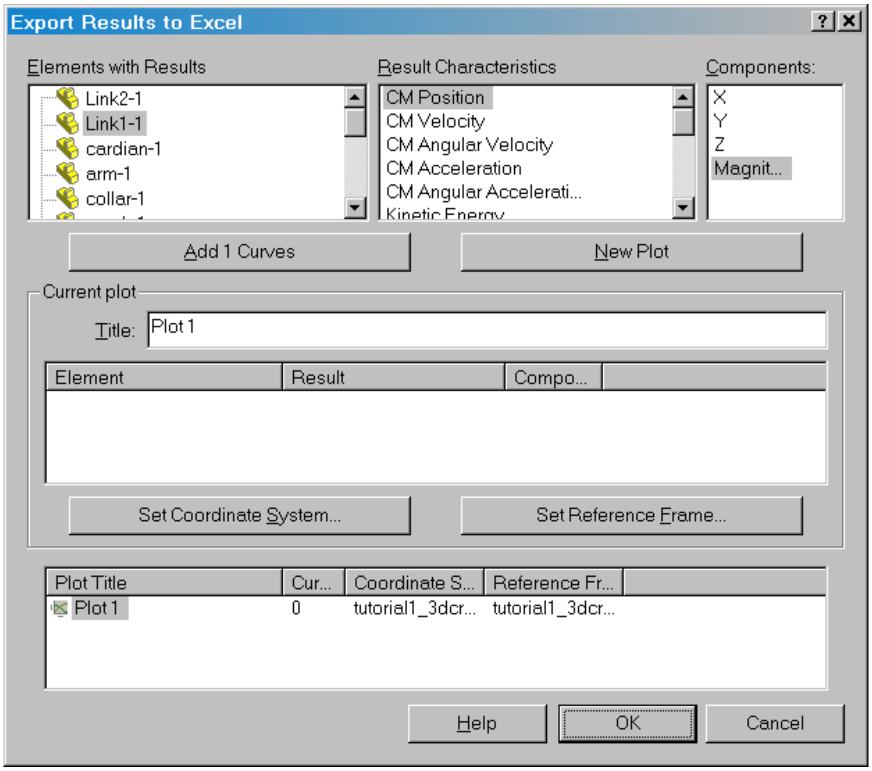
To Export Results to Excel:

1 Click **Motion, Export Results, To Spreadsheet...** or select the **Motion Model** in the COSMOSMotion feature manager and select **Export to Spreadsheet** from

the pop-up menu. You can also use the toolbar button.

Note: You must simulate your mechanism before you can export your results.

2 The following dialog box is display. It allows you to select the type of data you would like to export, and configure how that data is displayed.



Export Results to Excel

Title: Plot 1

Element	Result	Compo...

Plot Title	Cur...	Coordinate S...	Reference Fr...
Plot 1	0	tutorial1_3dcr...	tutorial1_3dcr...

You must make a selection in each of the boxes across the top of the dialog. Choose the element whose results you want in the **Elements with Results** box. Then choose the type of result to export in the **Result Characteristic** box, and finally choose the result component in the **Components** box.

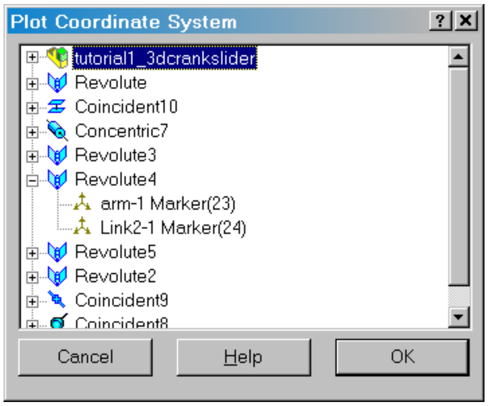
At this point you can click the **Add 1 Curve** button to add this curve to the queue of curves to be plotted. You may repeat this process for all the result types you desire to export. Any curves added in this manner will be plotted on the same plot in Excel.

If you click the **New Plot** button, this will create a new XY plot sheet in Excel. Any new curves added will be added to the new plot.

You can change the title of the plot by entering it in the **Title** box.

You may select an existing queued plot and add more curves to it by clicking on the plot name in the last box on the dialog.

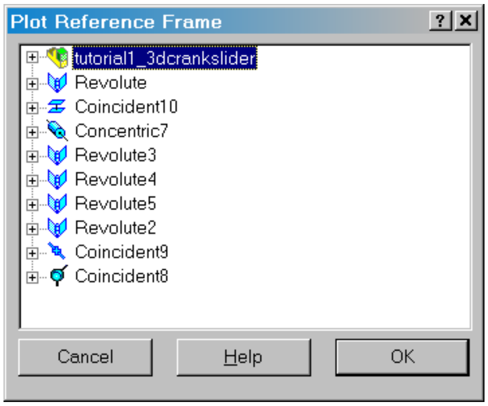
You can change the coordinate system the plot values are output in by clicking the **Set Coordinate System** button. This will display the following dialog:



This dialog contains every object that is defined in the Motion model and every coordinate system implied by that object. Click on the coordinate system you want to use as the output coordinate system.

By default the global coordinate system is the output coordinate system. The coordinate system controls the orientation of the x, y, and z components in the plot. This option allows you to output results so they are aligned with any part or motion entity in the system. For example, if you were plotting the translational velocity of Part A at the CM, where Part A is a pivoting arm in the XY-plane. Using the default global coordinate system to plot the x-axis velocity component would result in a sinusoidal plot (as would the y-axis component). If you selected Part A as the coordinate system, and lets say that on Part A, the x-axis is tangential to the direction of rotation. Then the x-axis plot would be a constant value equal to the velocity at the CM and the y-axis component would be zero.

You can change the output reference coordinate system by clicking the **Set Reference Frame** button. This will display the following dialog:



This dialog contains every object that is defined in the Motion model and every coordinate system implied by that object. Click on the coordinate system you want to use as the reference coordinate system.

By default the global coordinate system is the reference coordinate system. The reference frame determines the relative system the result is measured with respect to. For example, if you selected to plot the velocity of partA, and you selected partB as the reference frame, you would be measuring the relative velocity of partA with respect to partB.

You can delete a queued curve or plot by selecting it in the appropriate box and pressing the DELETE key.

3 When all the curves and plots have been specified, Click **OK** to actually create the plots in Excel.

COSMOSMotion automatically creates an Excel worksheet with graphs and tables for each of the elements that you selected.

Exporting Results to a Text File

You can export your simulation results to a text file in case you would like to review the results in another format. For example, if you do not have Excel 97, you can still export you results and read them into your favorite spreadsheet tool.

To Export Results to Text File:

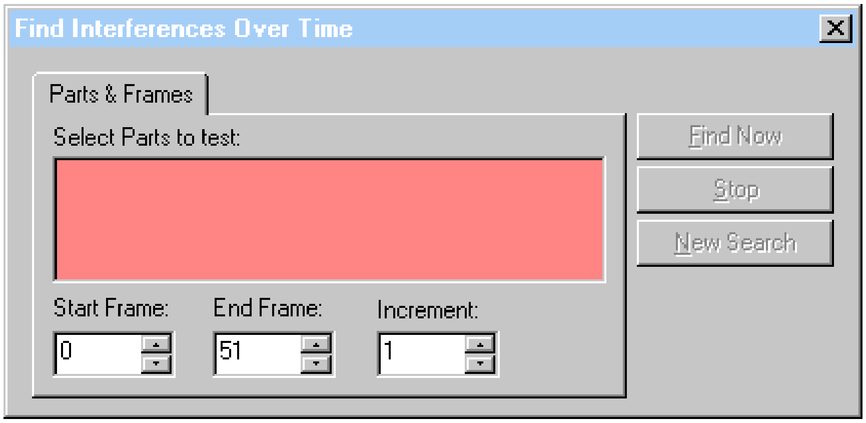
1 Click **Motion, Export Results, To File...**

2 The dialog and output options are the same as for exporting to a spreadsheet described previously.

Interference Detection

COSMOSMotion allows you to check for interference in your design as the parts move. You can check any of the components in your SolidWorks assembly model, regardless of whether a component participates in the motion model.

To initiate the interference check, choose either **Motion, Check Interference** from the Menu, or select the **Motion Model** branch in the Browser and choose **Interference Check**. You can also use the toolbar button. No matter what method you choose to start the process, the following dialog box is displayed:



Start Frame: 0

End Frame: 51

Increment: 1

You can select the parts to include in the interference check by selecting in the box labeled **Select Parts to test** and then picking two or more components from the graphics screen.

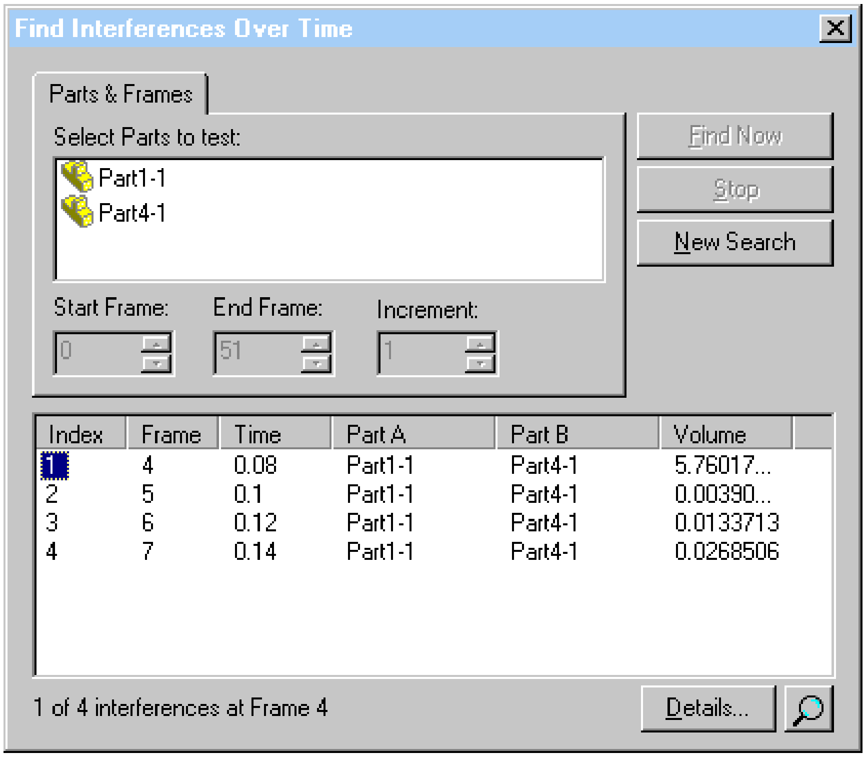
The box labeled **Start Frame** allows the specified motion frame to be used as the starting position for the interference check.

The box labeled **End Frame** allows the specified motion frame to be used as the final position for the interference check.

The box labeled **Increment** controls the increment applied to the motion frames as the interference check proceeds. Setting it to 2 will check every other frame, to 3 every third frame, and so on.

Once two or more components have been selected, the **Find Now** button is enabled. Clicking this button will start the interference check.

While the check is running, the dialog changes to:



Select Parts to test:

- Part1-1
- Part4-1

Start Frame: 0

End Frame: 51

Increment: 1

Index	Frame	Time	Part A	Part B	Volume
1	4	0.08	Part1-1	Part4-1	5.76017...
2	5	0.1	Part1-1	Part4-1	0.00390...
3	6	0.12	Part1-1	Part4-1	0.0133713
4	7	0.14	Part1-1	Part4-1	0.0268506

1 of 4 interferences at Frame 4

The list at the bottom of the dialog shows any interference conditions that have been detected. It lists the frame, the simulation time, the two parts that caused the interference, and the volume of the interference detected. Clicking on the **index** number of a particular interference enables the **Details** button and the magnifying glass button. Clicking the **Details** button displays more information about the interference.

When an index number is selected, the interfering volume will also be drawn on the graphics screen. Clicking the 🔍 will zoom the model so that the interference can be more clearly seen.

While the interference check is running, the **Stop** button is enabled. Clicking this will immediately stop the check.

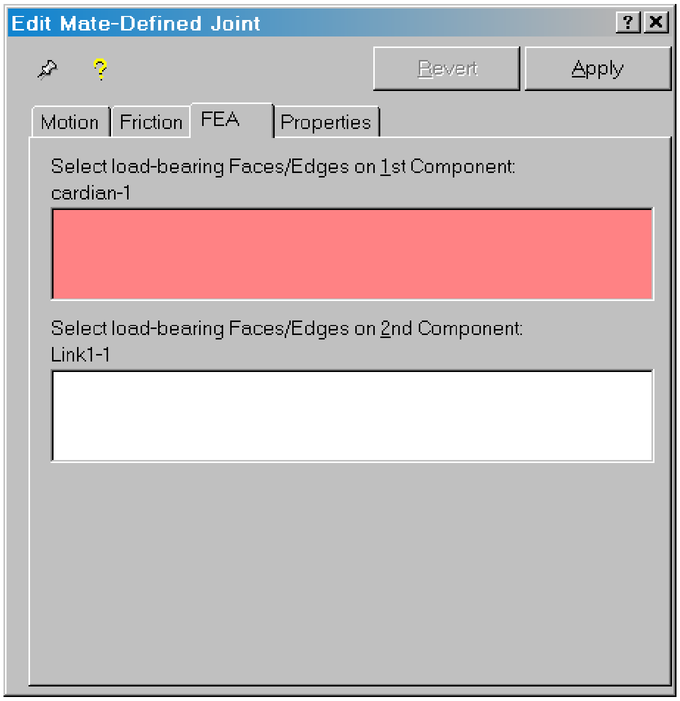
The **New Search** button clears all of the selected components and allows you to select new components to check.

Exporting to FEA

COSMOSMotion allows you to export reaction force results to COSMOS/Works. Using this capability you can use COSMOSMotion to predict the forces that your component will encounter as they move, and use this data in COSMOS/Works to predict the component stresses and deflections.

COSMOS/Works requires that any forces applied to its Finite Element Model are distributed over SolidWorks geometry. A force at a general 3D location in space is not supported. COSMOSMotion provide capabilities to allow the definition of **Bearing Surfaces** for entities that produce reaction forces (like joints) and for entities that are in themselves forces (like Action/Reaction Forces).

All COSMOSMotion entity property dialogs contains a tab labeled **FEA**. Clicking this tab display the following dialog box:



Select load-bearing Faces/Edges on 1st Component:
cardian-1

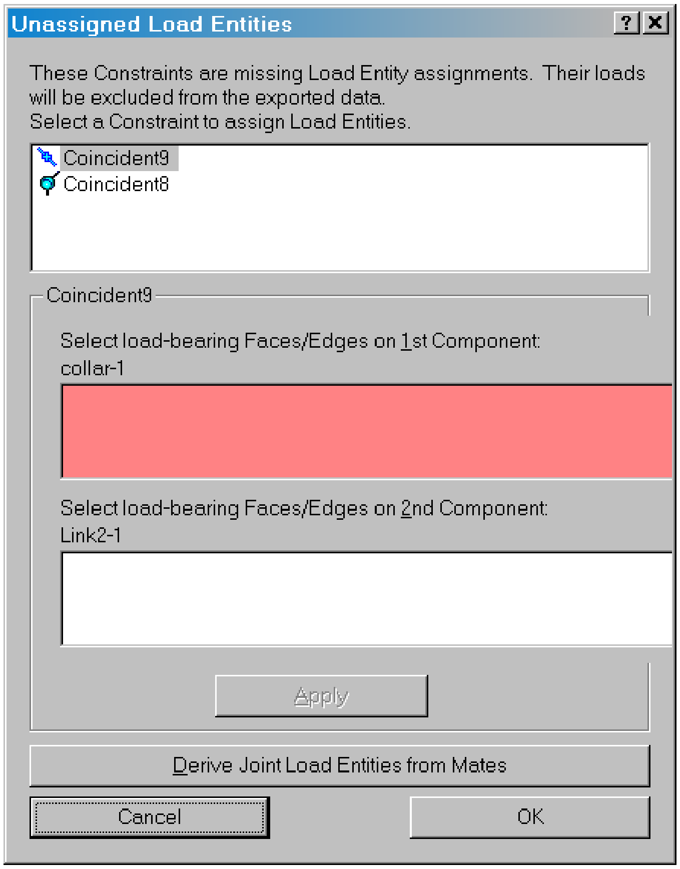
Select load-bearing Faces/Edges on 2nd Component:
Link1-1

In each of these boxes, you can select SolidWorks faces or edges on the part where the COSMOSMotion entity is attached. These faces and/or edges become the bearing surface for the forces calculated by COSMOSMotion. When the force data is transferred to CosmosWorks, it will distribute the forces and moments onto the selected faces and edges.

You can choose to define the bearing surface at the time you create a COSMOSMotion entity, or at the time you export the results to COSMOS/Works. The bearing surface only needs to be defined once, it will be used for any subsequent transfers to COSMOS/Works after its initial definition.

To initiate the export to CosmosWorks, first place the model at the position you want to analyze it in. At this time, only a single frame of data can be exported to these applications at one time.

Select the **Motion, Export Results, To COSMOS/Works** from the Menu or select the **Motion Model** in the COSMOSMotion feature manager and select **Export COSMOS/Works** from the pop-up menu. You can also use the toolbar button. If, at this point, entities exist where bearing surfaces have not been defined, the following dialog will be displayed:



These Constraints are missing Load Entity assignments. Their loads will be excluded from the exported data.
Select a Constraint to assign Load Entities.

Coincident9
Coincident8

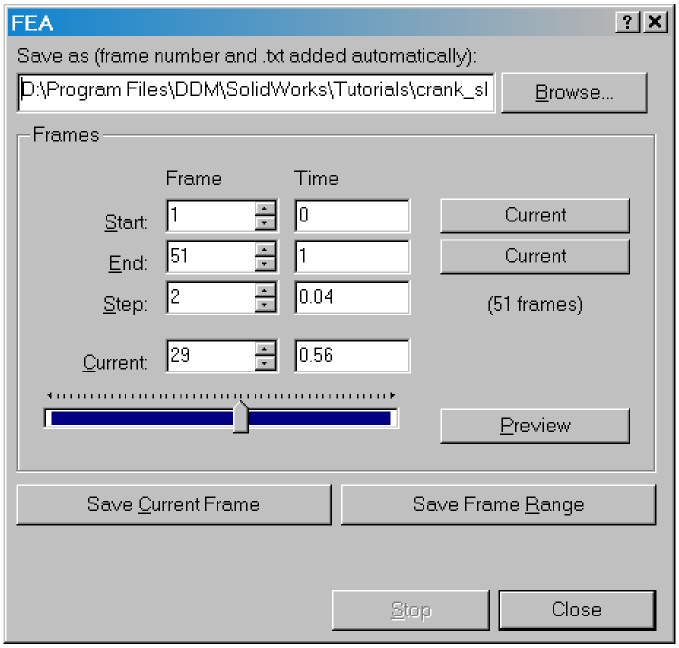
Coincident9

Select load-bearing Faces/Edges on 1st Component:
collar-1

Select load-bearing Faces/Edges on 2nd Component:
Link2-1

If the entities listed are joints or joint primitives that were generated from SolidWorks mates, then clicking the **Derive Joint Load Entities from Mates** will use the geometry that was originally used to create the mate as the bearing surface for these entities. If the entities listed here are not joints that were created from mates, then you can select the entity, and use the fields below to specify the bearing surfaces. Click **Apply** when finished with each entity.

Once all the bearing surfaces are defined, click **OK**, which will display the following dialog:



FEA

Save as (frame number and .txt added automatically): D:\Program Files\DDM\SolidWorks\Tutorials\crank_sl

Frames

	Frame	Time
Start:	1	0
End:	51	1
Step:	2	0.04
Current:	29	0.56

(51 frames)

You can enter the output file name in the box provided or click **Browse** to use the standard Windows File Save As dialog box. The frame number will be appended to the file name specified.

You can output the current frame by clicking **Save Current Frame** or a range of frames by clicking **Save Frame Range**. The range is specified using the **Start**, **End**, and **Step** parameters.

When the frames have been output, click **Close** to close the dialog.

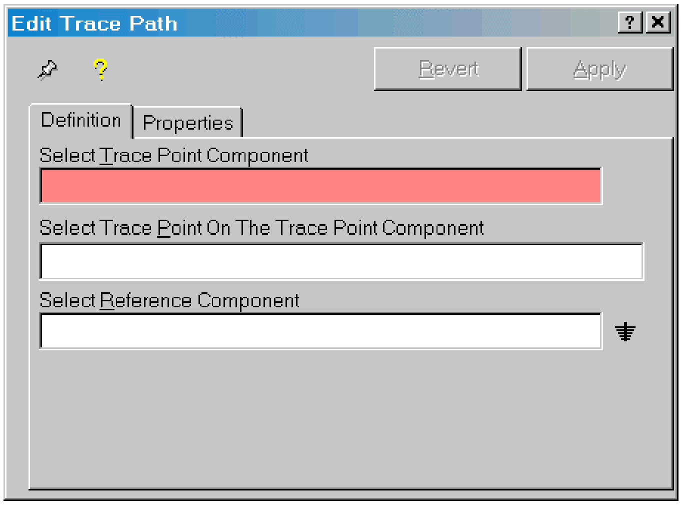
Please see the CosmosWorks documentation for information about importing the loads..

Creating Trace Paths

COSMOSMotion allows you to graphically display the path that any point on any moving part follows. This is called a **Trace Path**.

A Trace Path can be created by either using the menu **Motion->Result Object->Trace Path** or by clicking on a moving part node in the Browser, and selecting **Add Result Object->Trace Path** from the pop-up menu, or by clicking on either the **Results** or **Trace Path** nodes in the Browser and choosing **Create Trace Path** from the pop-up menu.

Regardless of the method used, the following dialog box will be displayed:


Edit Trace Path
?
Revert
Apply
Definition
Properties
Select Trace Point Component
Select Trace Point On The Trace Point Component
Select Reference Component

You can select the part that will be used to generate the trace curve by selecting in the box labeled **Select Trace Point Component** and then picking one moving part from the graphics screen.

You can specify the point that the trace path will follow by selecting in the box labeled **Select Trace Point on the Trace Point Component** and by selecting a point from the graphics screen.

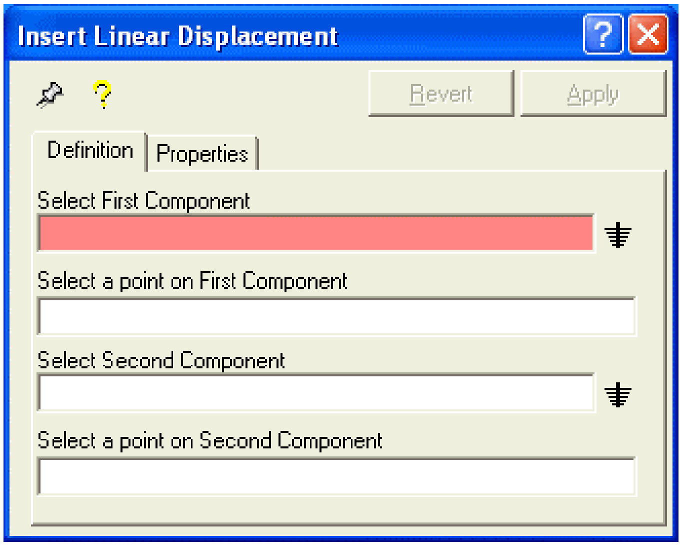
Optionally, you can select a reference component that defines a reference frame for the trace path by selecting in the box labeled **Select Reference Component** and then by selecting any motion part from the graphics screen. The default reference frame is the global reference frame defined by the ground part.

Creating Linear Displacements

COSMOSMotion allows you to graphically display the distance between any two points on any two parts. This is called a **Linear Displacement**.

A Linear Displacement can be created by either using the menu **Motion->Result Object->Linear Displacement** or by clicking on a moving part node in the Browser, and selecting **Add Result Object->Linear Displacement** from the pop-up menu, or by clicking on either the **Results** or **Linear Displacement** nodes in the Browser and choosing **Create Linear Displacement** from the pop-up menu.

Regardless of the method used, the following dialog box will be displayed:


Insert Linear Displacement
Revert
Apply
Definition
Properties
Select First Component
Select a point on First Component
Select Second Component
Select a point on Second Component

You can select the first part that will be used to measure the distance by selecting in the box labeled **Select First Component** and then picking one moving part from the graphics screen.

You can specify the point on the first part where the distance will be measured by selecting in the box labeled **Select a point on the First Component** and by selecting a point from the graphics screen.

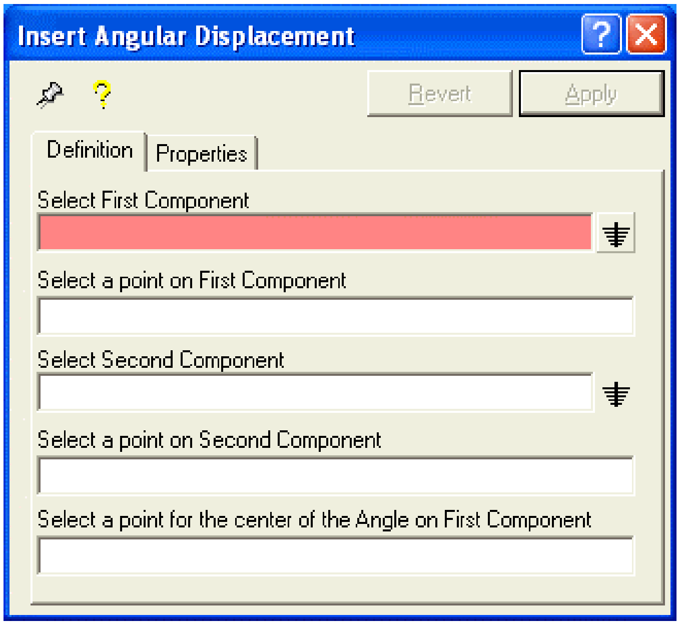
Repeat this process for the second component and the second point on that component.

Creating Angular Displacements

COSMOSMotion allows you to graphically display the angle between any three points on any two parts. This is called an **Angular Displacement**.

An Angular Displacement can be created by either using the menu **Motion->Result Object->Angular Displacement** or by clicking on a moving part node in the Browser, and selecting **Add Result Object->Angular Displacement** from the pop-up menu, or by clicking on either the **Results** or **Angular Displacement** nodes in the Browser and choosing **Create Angular Displacement** from the pop-up menu.

Regardless of the method used, the following dialog box will be displayed:


Insert Angular Displacement
Revert
Apply
Definition
Properties
Select First Component
Select a point on First Component
Select Second Component
Select a point on Second Component
Select a point for the center of the Angle on First Component

You can select the first part that will be used to measure the angle by selecting in the box labeled **Select First Component** and then picking one moving part from the graphics screen.

You can specify the point on the first part where the angle will be measured by selecting in the box labeled **Select a point on the First Component** and by selecting a point from the graphics screen.

Repeat this process for the second component and the second point on that component.

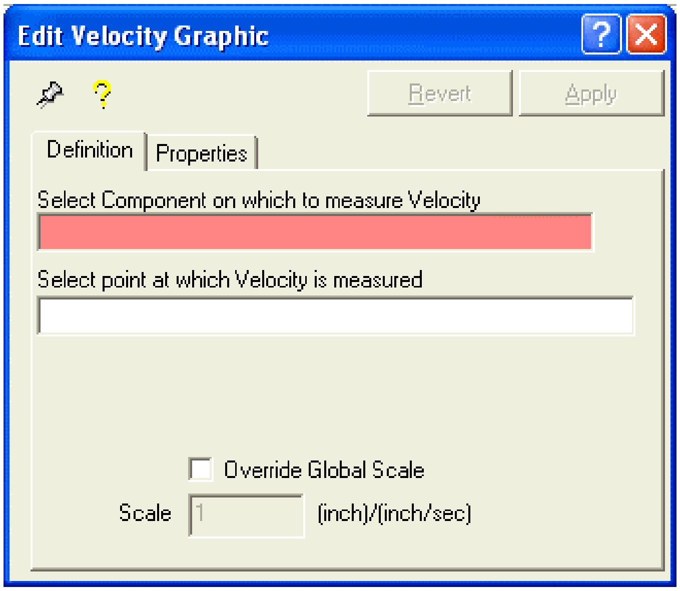
Finally pick a point on the first component that will be the center of the angle.

Creating Velocity Vectors

COSMOSMotion allows you to graphically display the velocity at any point on any moving part. This is called a **Velocity Vector**.

A Velocity Vector can be created by either using the menu **Motion->Result Object->Velocity** or by clicking on a moving part node in the Browser, and selecting **Add Result Object->Velocity** from the pop-up menu, or by clicking on either the **Results** or **Velocity** nodes in the Browser and choosing **Create Velocity** from the pop-up menu.

Regardless of the method used, the following dialog box will be displayed:



Select Component on which to measure Velocity

Select point at which Velocity is measured

☐ Override Global Scale

Scale 1 (inch)/(inch/sec)

You can select the part that contains the point that the velocity will be coimputed at by selecting in the box labeled **Select Component at which to measure Velocity** and then picking one moving part from the graphics screen.

You can specify the point where the velocity will be computed by selecting in the box labeled **Select a point at which the Velocity is measured** and by selecting a point from the graphics screen.

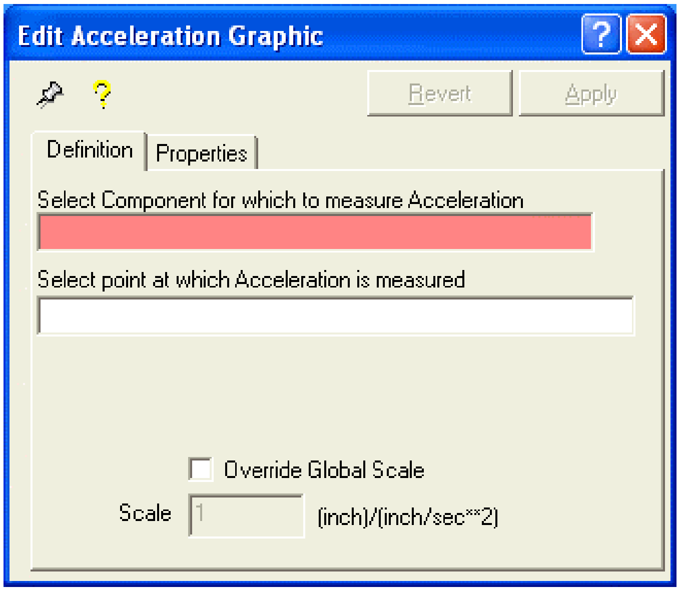
You can override the global scale for velocity vectors by enabling the checkbox labeled **Override Global Scale** and then by entering a new scale value in the **Scale** box,

Creating Acceleration Vectors

COSMOSMotion allows you to graphically display the acceleration at any point on any moving part. This is called an **Acceleration Vector**.

An Acceleration Vector can be created by either using the menu **Motion->Result Object->Acceleration** or by clicking on a moving part node in the Browser, and selecting **Add Result Object-> Acceleration** from the pop-up menu, or by clicking on either the **Results** or **Acceleration** nodes in the Browser and choosing **Create Acceleration** from the pop-up menu.

Regardless of the method used, the following dialog box will be displayed:


Edit Acceleration Graphic
Revert
Apply
Definition
Properties
Select Component for which to measure Acceleration
Select point at which Acceleration is measured

□ Override Global Scale

Scale 1 (inch)/(inch/sec**2)

You can select the part that contains the point that the velocity will be coimputed at by selecting in the box labeled **Select Component at which to measure Acceleration** and then picking one moving part from the graphics screen.

You can specify the point where the velocity will be computed by selecting in the box labeled **Select a point at which the Acceleration is measured** and by selecting a point from the graphics screen.

You can override the global scale for acceleration vectors by enabling the checkbox labeled **Override Global Scale** and then by entering a new scale value in the **Scale** box,

Creating Reaction Forces and Moments

COSMOSMotion allows you to graphically display the reaction forces and moments that act on any part in the motion simulation.

Reaction forces and moments can be created by clicking on a moving or ground part node in the Browser, and selecting **Add Result Object-> Reaction Forces** from the pop-up menu.

Exporting Result Object Values

Each of the result objects (Trace Path, Linear and Angular Displacements, Velocity and Acceleration Vectors, and Reaction Forces and Moments) can have their values exported to a comma separated value (CSV) formatted file which can then be imported to a spreadsheet program like Microsoft Excel.

To export the values, click on the object(s) whose values are to be exported, and choose **Export CSV** from the pop-up menu. The standard Windows File Save As dialog box will be displayed requesting the name of the CSV file to be created.

6

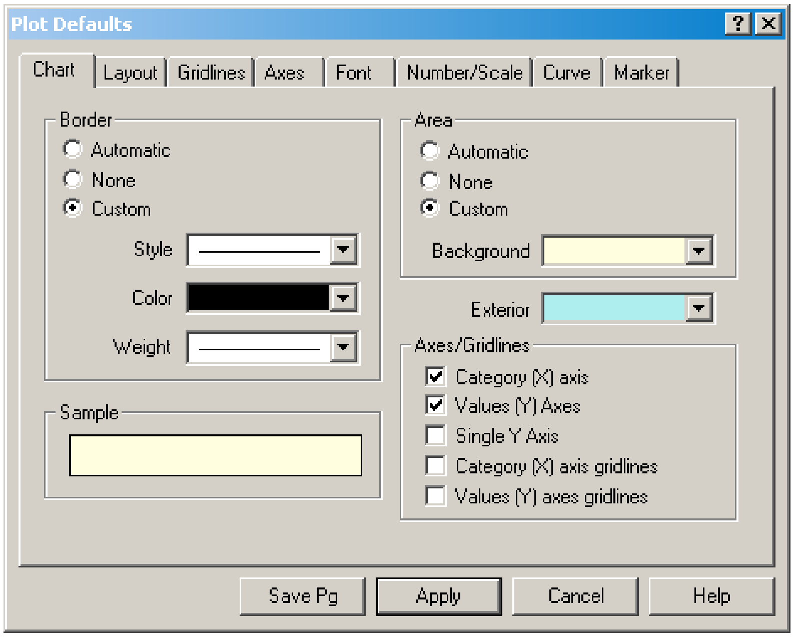
XY Plotting

XY Plots are a convenient means of displaying results quantities on a graph.. XY Plots in the Results Viewer allow multiple quantities to be displayed on one graph. XY Plots can also be visible in animations. Any number of XY Plots can be created, and multiple values can be displayed on a single XY Plot.

Plot Defaults

Plot defaults can be set by selecting either the **Motion Model** node or the **XY Plots** node and in the Browser and selecting **Plot Defaults** from the pop-up menu.

Regardless of the method used, the following dialog box will be displayed:



Plot Defaults

Chart | Layout | Gridlines | Axes | Font | Number/Scale | Curve | Marker

Border
○ Automatic
○ None
● Custom
Style: ————
Color: ■
Weight: ————

Sample

Area
○ Automatic
○ None
● Custom
Background

Exterior

Axes/Gridlines
☑ Category (X) axis
☑ Values (Y) Axes
☐ Single Y Axis
☐ Category (X) axis gridlines
☐ Values (Y) axes gridlines

Chart Tab

The first page in the Plot Default dialog controls properties that apply to the complete XY Plot.

The “Border” option is used to control the type of border drawn around the plotting area of the XY Plot. “Automatic” will create a border one pixel wide drawn in a solid black line. “None” will not create any border, and “Custom” allows the selection of the color, line weight, and line style of the border via the three combo boxes labeled “Style”, “Color” and “Weight”.

The “Area” option is used to control the background color of the plotting area, that is the area contained within the plot’s X and Y axes. “Automatic” will set the background color to white. “None” will use the same color as the “Exterior” selection. “Custom” will allow selection of the background color via the combo box labeled “Background”.

The combo box labeled “Exterior” is used to set the background color of the plot area that is outside of the plot’s X and Y axes.

The options listed under “Axes/Gridline” control the display of the plot’s axes and gridlines.

If the “Category (X) Axis” option is checked, the plot will display an X axis.

If the “Values (Y) Axis” option is checked, the plot will display one or more Y axes.

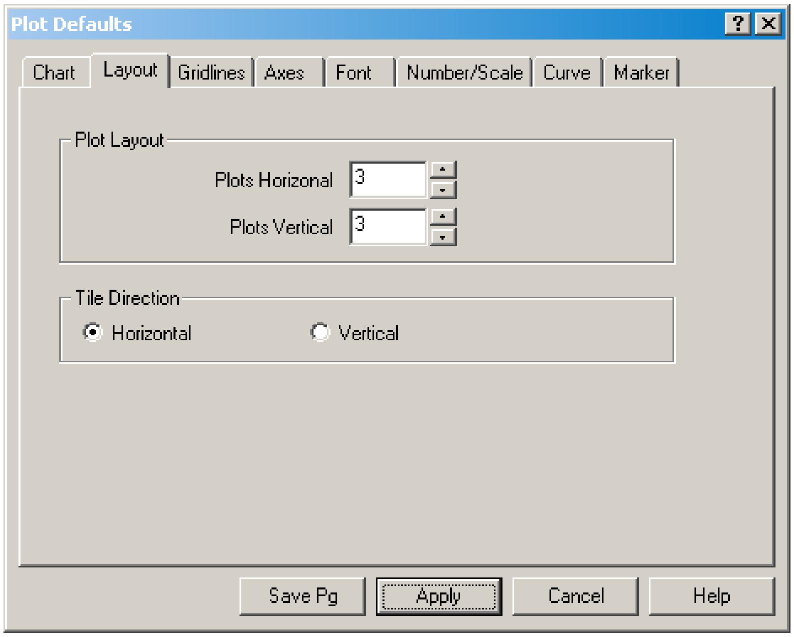
If the “Single Y Axis” option is checked, plots that display more than one set of values, will only have a single Y axis displayed. If the option is left unchecked, each new value added to the plot will create a new Y axis.

If the “Category (X) axis gridlines” option is checked, the plot will display vertical gridlines along the plot’s X axis.

If the “Value (Y) axis gridlines” option is checked, the plot will display horizontal gridlines along the plot’s Y axis.

Layout Tab

Clicking the “Layout” tab on the “Plot Defaults” dialog will display the following page:



Plot Layout

Plots Horizonal 3

Plots Vertical 3

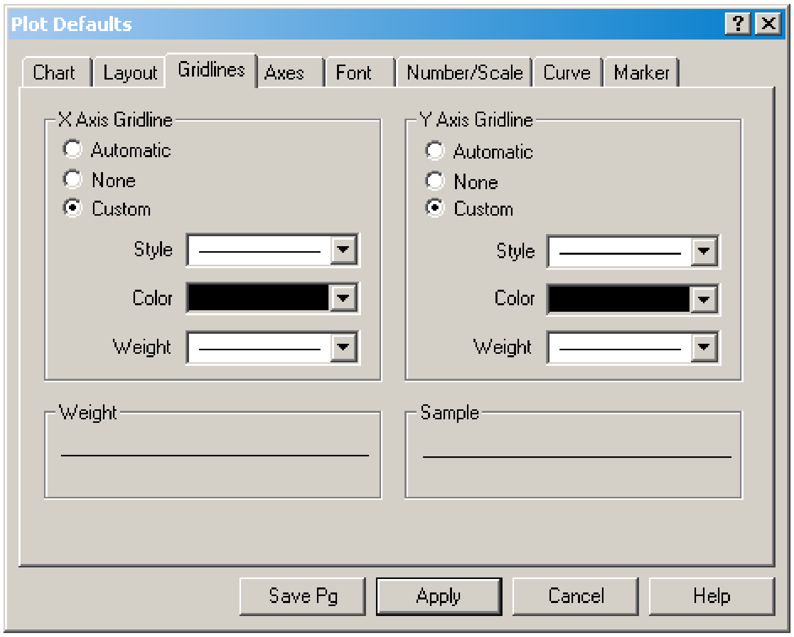
Tile Direction

● Horizontal ○ Vertical

These options control the number of plots that are tiled across the SolidWorks window, and the direction that new plots are tiled. The “Plots Horizontal” and “Plots Vertical” options control how many plots are created horizontally and vertically within the SolidWorks window. The first plot is created at the lower right hand corner of the SolidWorks frame window. Subsequent plots are placed either above or to the left of the preceding plot depending on the “Tile Direction” setting. Choosing the “Horizontal” tiling direction caused plots to be placed from right to left along the bottom of the frame window. When the “Plots Horizontal” number of plots have been created, a new row of plots will be created above the first row. If the “Vertical” tiling direction is active, subsequent plots are added above the previous plot. When “Plots Vertical” number of plots have been created a new column of plots will be created.

Gridlines Tab

Clicking the “Gridlines” tab on the “Plot Defaults” dialog will display the following page:



X Axis Gridline

○ Automatic
○ None
● Custom

Style
Color
Weight

Y Axis Gridline

○ Automatic
○ None
● Custom

Style
Color
Weight

Weight

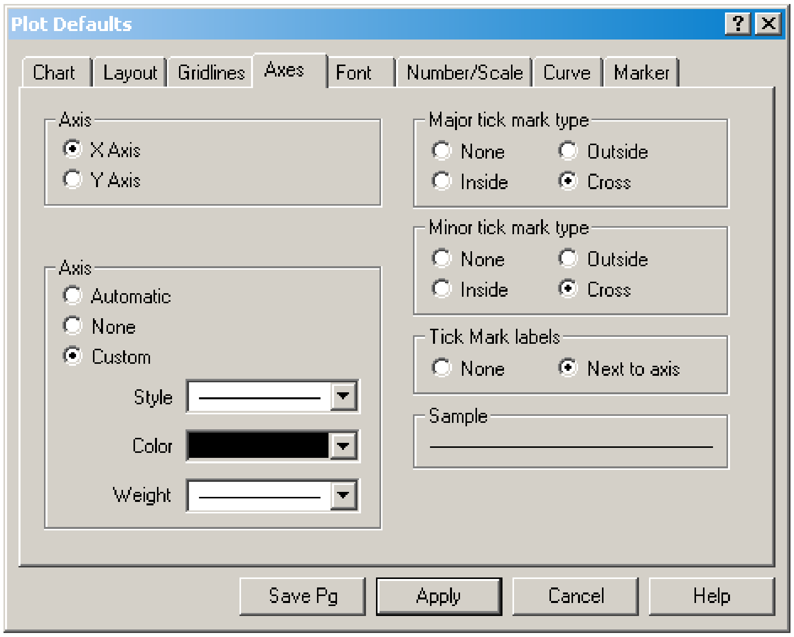
Sample

This page contains options for displaying gridlines along the X and Y axes of the plot. The options listed under “X Axis Gridline” apply to the vertical gridlines that emanate from the plot’s X axis and those listed under “Y Axis Gridline” apply to the horizontal gridlines that emanate from the plot’s Y axis.

“Automatic” will display a gridline that is drawn in a solid black color with a pen width of 1. “None” will cause no gridlines to be drawn. “Custom allows the gridline style, color, and line weight to be specified via the combo boxes labeled “Style”, “Color” and “Weight”

Axes Tab

Clicking the “Axes” tab on the “Plot Defaults” dialog will display the following page:



Axis
● X Axis
○ Y Axis

Axis
○ Automatic
○ None
● Custom
Style ────────── ▾
Color ■ ▾
Weight ────────── ▾

Major tick mark type
○ None ○ Outside
○ Inside ● Cross

Minor tick mark type
○ None ○ Outside
○ Inside ● Cross

Tick Mark labels
○ None ● Next to axis

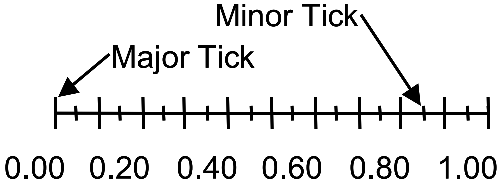
Sample

This page allows options that apply to the plot’s axes to be set.

The “X Axis” and “Y Axis” options determine which axis properties will be set. If “Y Axis” is selected the edit box labeled “Y Axis Number” is displayed. A plot may contain more than one Y axis, and the number displayed in this box determines for which Y axis the properties are being set.

The options listed under the lower “Axis” option control how the specified axis is to be displayed. “Automatic” will draw the axis in a solid black line with a pen width of 1. “None” will cause no axis to be displayed on the plot. “Custom” allows the axis Style, Color, and line weight to be specified via the combo boxes labeled “Style”, “Color” and “Weight”.

The options listed under “Major tick mark type” control how major tick marks are displayed. Major tick mars are the increments along an axis that have numbers displayed beneath or next to the axis.



The options listed under “Minor tick mark type” apply to the minor tick marks.

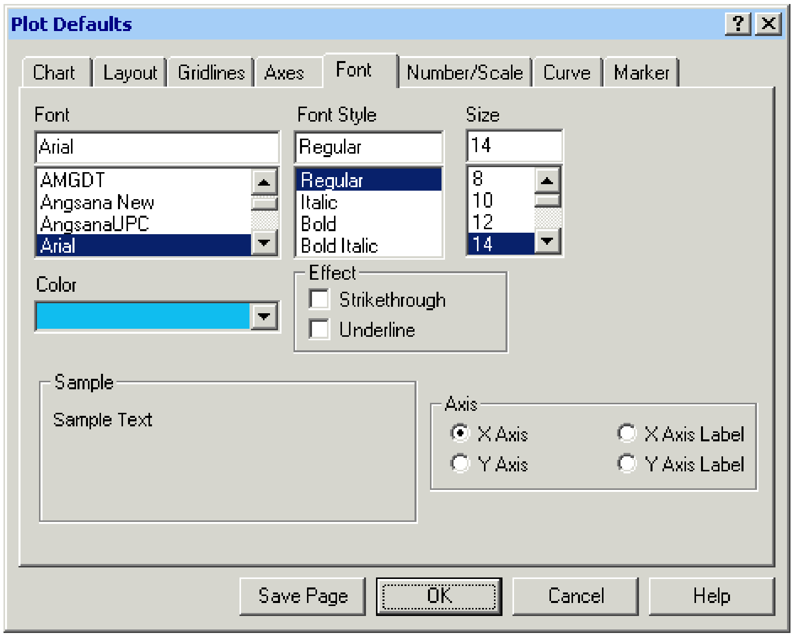
“None” will cause no tick mark to be displayed. “Inside” causes the tick mark to start on the axis and to be drawn towards the inside of the plot. “Outside” causes the tick mark to start on the axis and to be drawn away from the plot. “Cross” causes the tick mark to start below the axis, and end above the axis.

The options under “Tick mark labels” control whether text is drawn below or next to the tick mark. “None” results in no text being drawn, and “Next to axis” results in the text being displayed.

If one of the Y axes properties are being defined, the “Show moving tick” option is displayed. Checking this box causes an additional tick to be displayed on the Y axis that corresponds to the current value of the quantity being displayed on the plot at the current position of the model. As the model is stepped through different frames, the tick will move up and down the axis.

Font Tab

Clicking the “Font” tab on the “Plot Defaults” dialog will display the following page:



Font: Arial

Font Style: Regular

Size: 14

Color:

Effect

□ Strikethrough

□ Underline

Axis

● X Axis ○ X Axis Label

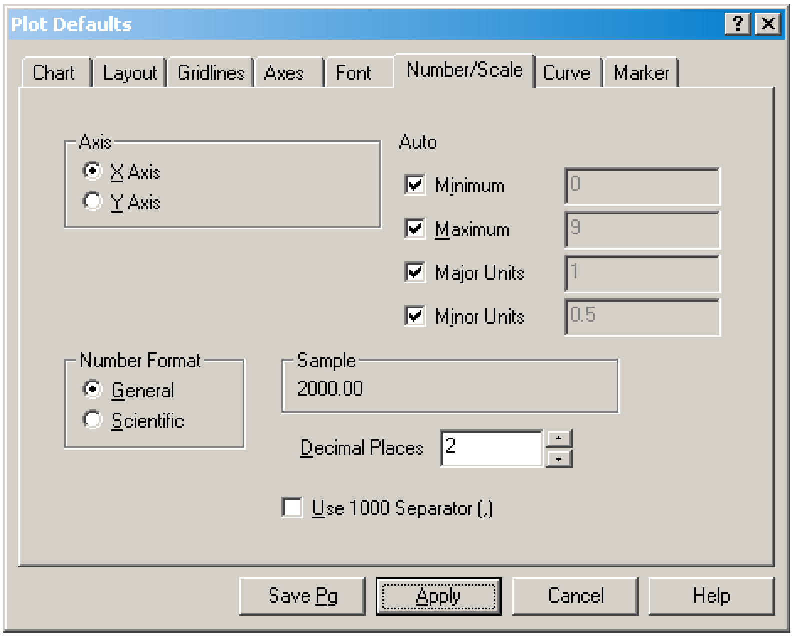
○ Y Axis ○ Y Axis Label

Four different types of fonts may be specified on a plot. These are selected in the options box labeled “Axis” at the lower right hand corner. The four font types that may be specified are the X Axis tick text (“X Axis”), the Y Axis tick text (“Y Axis”), the X Axis label (“X Axis Label”) and the Y Axis label (“Y Axis Label”). If “Y Axis” or “Y Axis Label” is selected, the edit box labeled “Y Axis Number”' is displayed. This can be used to specify which Y Axis or Y Axis Label properties are being set.

The remaining options are standard Windows font selection options for Font type, effects, size, and color.

Number/Scale Tab

Clicking the “Number/Scale” tab on the “Plot Defaults” dialog will display the following page:



Axis

● X Axis

○ Y Axis

Auto

☑ Minimum 0

☑ Maximum 9

☑ Major Units 1

☑ Minor Units 0.5

Number Format

● General

○ Scientific

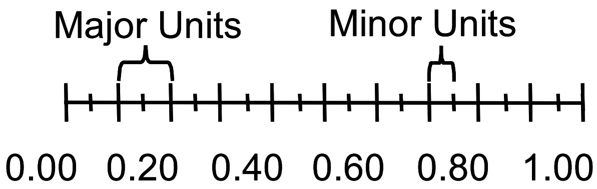
Sample

2000.00

Decimal Places 2

□ Use 1000 Separator (,)

This page allows the scale and number format for each axis to be specified. As in previous pages, the “Axis” option allows the selection of the axis for which the scale and number format will be specified.



For each axis a minimum and maximum value can be set which is the range of values displayed along the axis. If the checkbox under the “Auto” column is set, the minimum and maximum values will be obtained automatically from the data.

The “Major Units” and “Minor Units” values are the distance along the axis between major and minor tick marks. Since tick mark text is usually placed next to the major tick marks, the “Major Units” value is also the separation between the tick mark text.

If the checkbox in the “Auto” column is set, these values will be obtained automatically from the data.

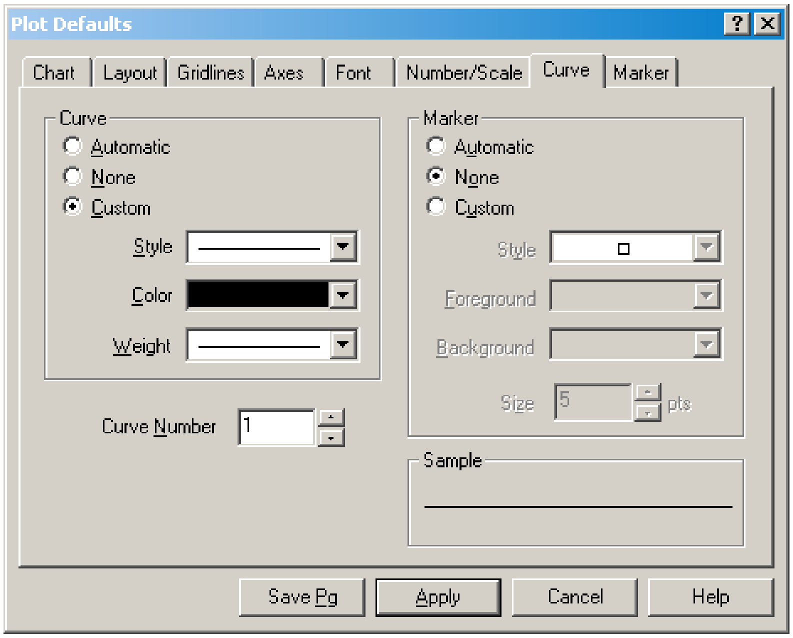
The box labeled “Number Format” can be used to select the format for displaying numbers. “General” will result in numbers displayed in normal, or non-scientific notation, and “Scientific” will use scientific notation to display the number. Check the “sample” box on the dialog to see exactly how the number will be formatted.

The “Decimal Places” edit box accepts the number of digits to be displayed to the right of the decimal point in either general or scientific notation.

The check box labeled “Use 1000 separator (,)”, if checked, will use the specified separator between every three groups of digits to left of the decimal point.

Curve Tab

Clicking the “Curve” tab on the “Plot Defaults” dialog will display the following page:



Curve
○ Automatic
○ None
● Custom
Style
Color
Weight

Curve Number 1

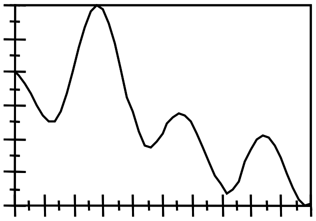
Marker
○ Automatic
● None
○ Custom
Style □
Foreground
Background
Size 5 pts

Sample

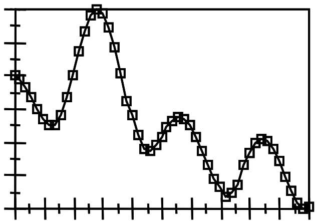
Curve in this case refers to the actual curve displayed on the plot. Each set of values is displayed on the plot in the form of a curve. This dialog allows curve properties to be set. Since more than one curve can be displayed on a plot, the edit box labeled “Curve Number” is used to specify which curve properties are being set.

The options listed under “Curve” allow the line style of the plotted curve to be specified. “Automatic” results in a solid black curve draw with a line weight of 1. “None” will not display any curve. “Custom” allows the curve’s style, color, and line weight to be specified via the combo boxes labeled “Style”, “Color”, and “Weight”.

The option listed under “Marker” controls the display of symbols drawn at each data point on the curve.



Curve without Markers displayed

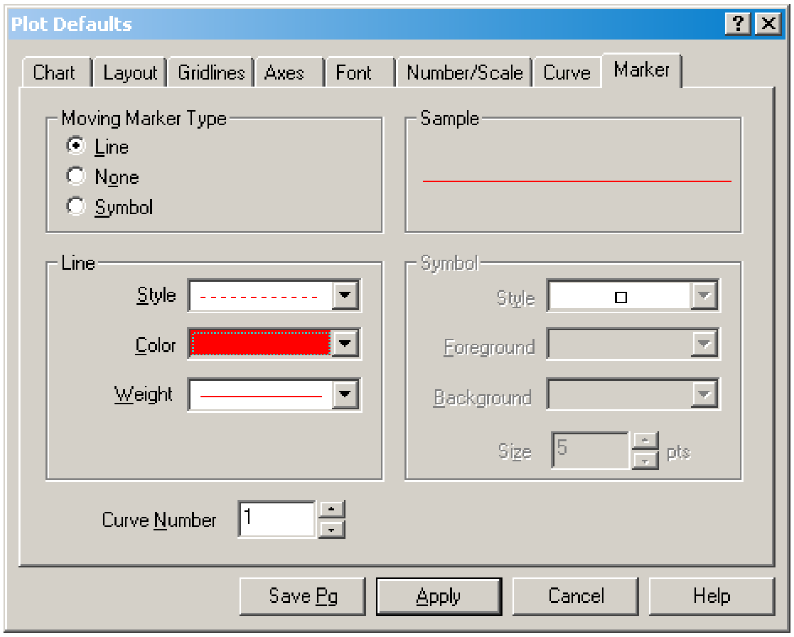


Curve with Markers displayed

“Automatic” will display a default symbol for each curve. The size of the symbol can be specified using the “Size” edit box. “None” will not display any curve markers. “Custom” allows the marker style, foreground color, background color, and size to be specified by the combo boxes labeled “Style”, “Foreground”, and “Background”, and by the edit box labeled “Size”.

Marker Tab

Clicking the “Marker” tab on the “Plot Defaults” dialog will display the following page:



Moving Marker Type
● Line
○ None
○ Symbol

Line
Style
Color
Weight

Symbol
Style □
Foreground
Background
Size 5 pts

Curve Number 1

Marker in this case refers to a marker that moves along a curve, showing the current value of a quantity. The current value changes as the motion model moves from frame to frame.

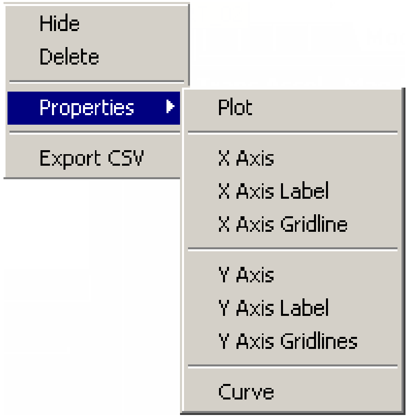
The options under “Moving Marker Type” control the type of moving marker for the specified curve. “Line” creates a vertical scan line that starts at the X axis and extends to the top of the plot. As the model moves between frames, the line sweeps across the plot. The line style, color, and weight are specified via the combo boxes under the “Line” options. “None” turns off any moving marker for the specified curve.

“Symbol” specifies that the moving marker is a symbol instead of a vertical line. The options listed under “Symbol” are enabled allowing the symbol style, foreground color, background color, and size to be specified.

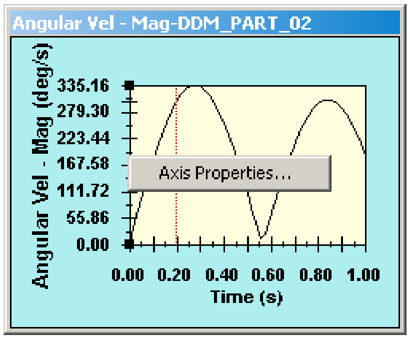
Properties of Existing Plots

All of the properties that were described in the previous section can also be set for plots that already exist. There are two primary ways of accomplishing this; 1)

Clicking on the icon of the plot to be changed in the browser and selecting the appropriate property pick from the pop-up menu,



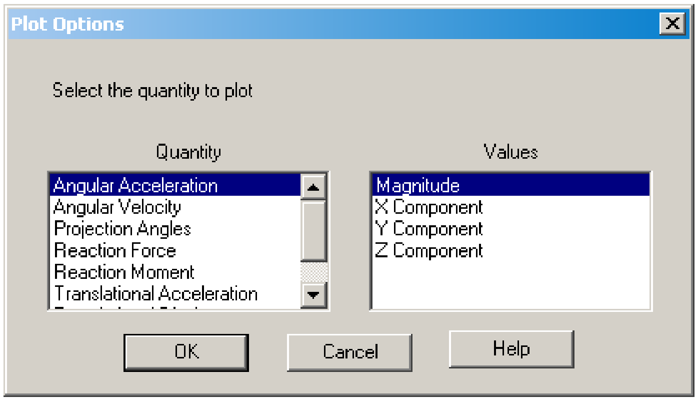
and 2) clicking on an object like an axis or a curve in the plot itself, and selecting from the pop-up menu.



Creating Plots

XY Plots can be created by clicking on an object like a part or a joint and selecting **Plot** from the pop-up menu., or by dragging an object in the Browser and dropping it on the XY Plots node.

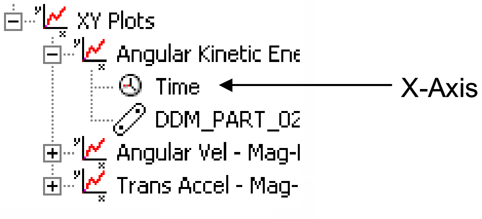
If the drag and drop method is used, a dialog box will be displayed listing all of the quantities for the dragged object that can be plotted.



Plot Options

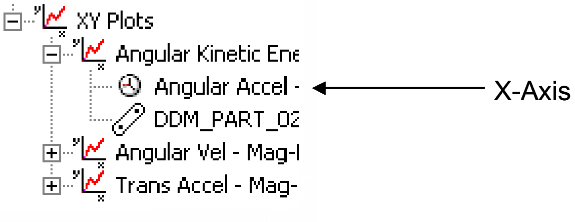
Select the quantity to plot

When a plot is created from a Browser pop-up menu or via drag and drop from the Browser, the X Axis of the plot is set to simulation time.. A plot's X Axis can be changed to show either the simulation frame, or another set of object values. The Plot's X Axis is show in the browser as the first node under the plot's node:



To change the X Axis to frame, or to set it back to time if it is set to another value, click on the X Axis node, display the pop-up menu, and choose either "Versus Time" or "Versus Frame".

To use an object's values as the X Axis, drag the object in the Browser and drop it on the plot's X Axis node. The Plot Quantity Selection dialog box is displayed and the value to be used as the X Axis can be selected.

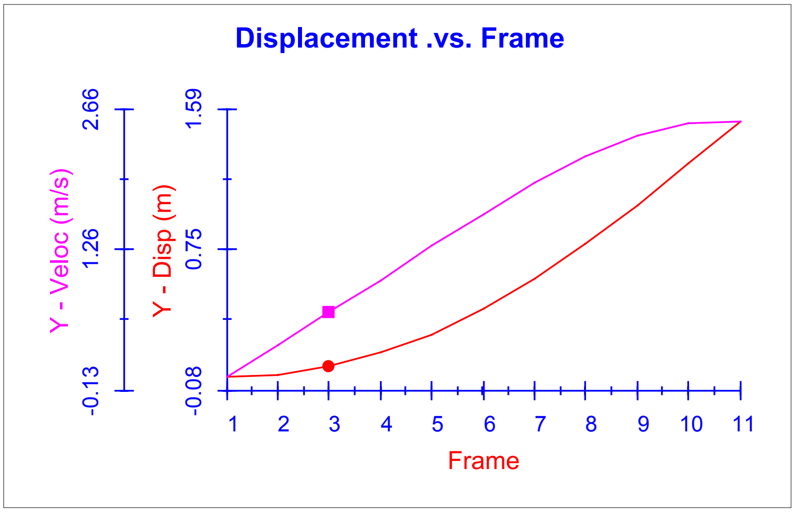


Adding Values to Plots

XY Plots can support up to ten quantities simultaneously displayed on the same plot, subject to available space on the screen. Multiple quantity plots all use the same X axis value, only additional Y axis values can be added. Values can be added to an existing plot by dragging and dropping in the browser.

To add an object to a plot, drag an object which contains the value to be added and drop it on the node that represents the XY plot. The Plot Quantity Selection dialog box will be displayed and the quantity to be added can be selected. A new Y axis will created and a new curve added to the plot.

An example of a multiple quantity XY Plot is shown below:

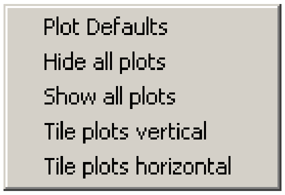


Other XY Plot Capabilities

An XY Plot may be deleted either by selecting one or more plot nodes in the Browser and selecting Delete from the pop-up menu, or by right clicking in the plot window and selecting Delete from the pop-up menu.

Clicking on an XY plot curve, will move the current frame to the X axis value of the curve where the click occurred. This feature is only active when the X axis displays either Frame or Time.

The pop-up menu on the Plots root node in the Browser offers the following options:



“Plot Defaults” displays the plot default dialog box.

“Hide all plots” will cause all displayed plots to be hidden.

“Show all plots’ will cause all plots to be displayed.

“Tile plots vertical” will retile and resize all plots in a vertical configuration. It will also set the tile mode of any future plots added to vertical.

“Tile plots horizontal” will retile and resize all plots in a horizontal configuration. It will also set the tile mode of any future plots added to horizontal.

The plot nodes also have options for showing and hiding individual plots. Please refer to the Browser on-line documentation for more information.

Plot Persistence

Plots are persistent across both simulation runs and database opens and saves. If one or more plots exist and results are deleted, the plots are all hidden. When a new simulation is run and results are present, all of the existing plots will be regenerated with the new simulation data. Those that were previously visible will become visible again.

Plots are “owned” by the objects that they take their values from. If all of the objects that share ownership of a plot are deleted, then the plot itself is automatically deleted.

When in the modeling mode, plots are still visible as icons in the Browser, but they cannot be modified in any way.

7

IntelliMotion Builder

The IntelliMotion Builder is a combination of a tabbed dialog box and a wizard that leads you step-by-step through the process of changing an Assembly Model into a Motion Model, and performing the Motion simulation.

Any Assembly component that is included in the motion simulation will automatically be converted in a motion part complete with mass properties. Any assembly constraints between components included in the motion simulation will automatically be converted into a motion joint. For assembly models that are completely constrained, all that remains to be done after the IntelliMotion Builder has finished its work is to add motion specified entities like motion generators and springs. The IntelliMotion Builder is the quickest and most efficient way to perform motion simulation on a SolidWorks Assembly model.

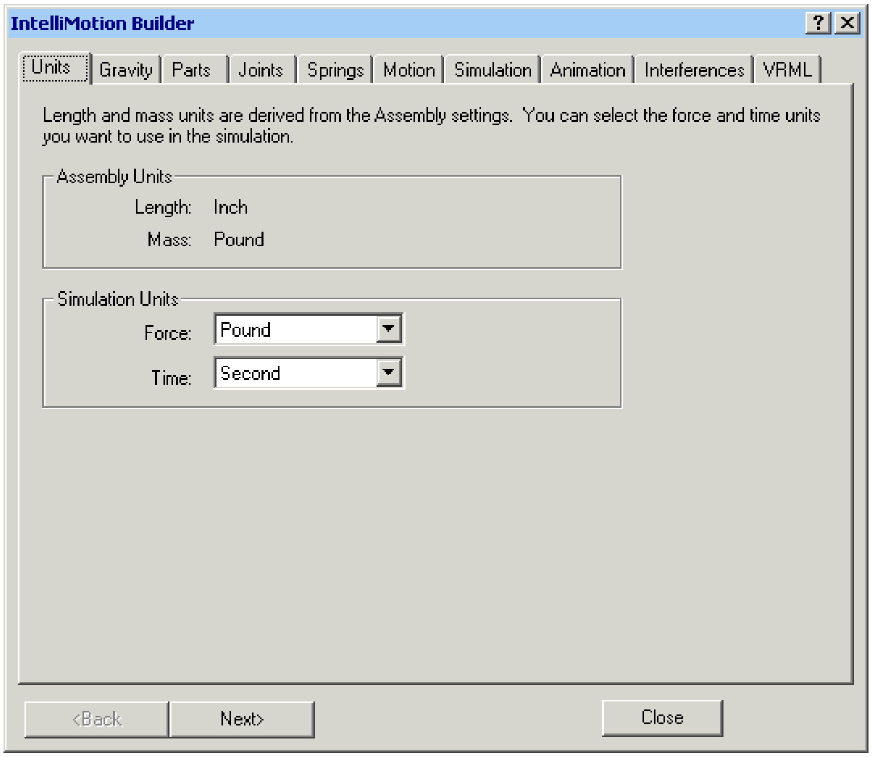
The IntelliMotion Builder is started from the Browser by clicking the **Motion Model** node and choosing **IntelliMotion Builder** from the pop-up menu or by selecting **IntelliMotion Builder** from the **Motion** pull down menu.

In the lower left corner of each page in the IntelliMotion Builder are two buttons labeled **<BACK** and **NEXT>**. These buttons can be used to move sequentially forward and backwards through the Motion process. The IntelliMotion Builder will not allow you to proceed to the next step until all of the information required for the current step has been specified.

IntelliMotion Builder Units Page

The first step in the IntelliMotion Builder process is to specify the units that will be used for the Motion Simulation. The information specified on this page is the same that can be specified using the **MOTION OPTIONS** command. Units can be changed at any point in the IntelliMotion Builder process. Any values entered prior to changing to another system of units will be automatically converted and presented in the new units system.

The Units Page looks like:



Length and mass units are derived from the Assembly settings. You can select the force and time units you want to use in the simulation.

Assembly Units

Length: Inch

Mass: Pound

Simulation Units

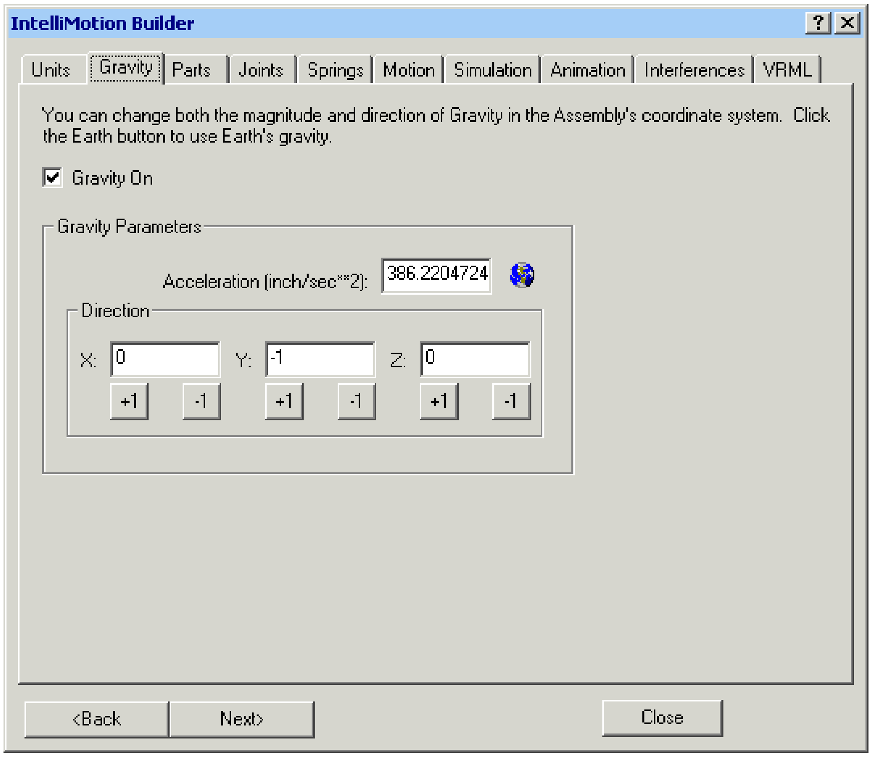
Force: Pound

Time: Second

IntelliMotion Builder Gravity Page

The second step in the IntelliMotion Builder process is to specify the direction and magnitude of Gravity. The information specified on this page is the same that can be specified using the **MOTION OPTIONS**. Gravity can be changed at any point in the IntelliMotion Builder process prior to performing a simulation.

The Gravity Page looks like:



You can change both the magnitude and direction of Gravity in the Assembly's coordinate system. Click the Earth button to use Earth's gravity.

☑ Gravity On

Gravity Parameters

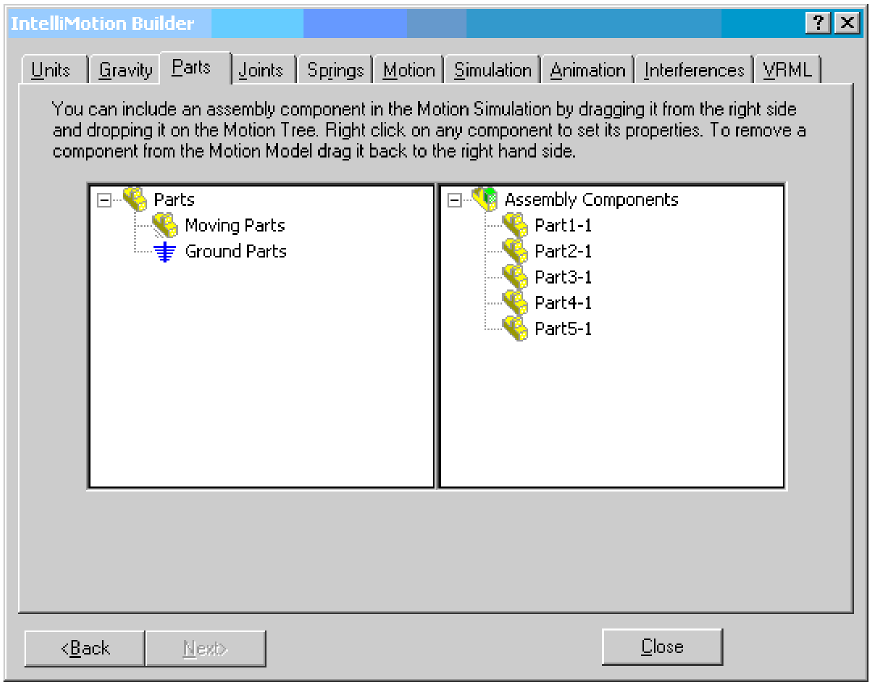
Acceleration (inch/sec**2): 386.2204724

Direction

X: 0 Y: -1 Z: 0

IntelliMotion Builder Part Page

The Parts page allows you to specify what Assembly components will participate in the Motion Simulation as both moving and non-moving (Ground) parts. The page is shown below:



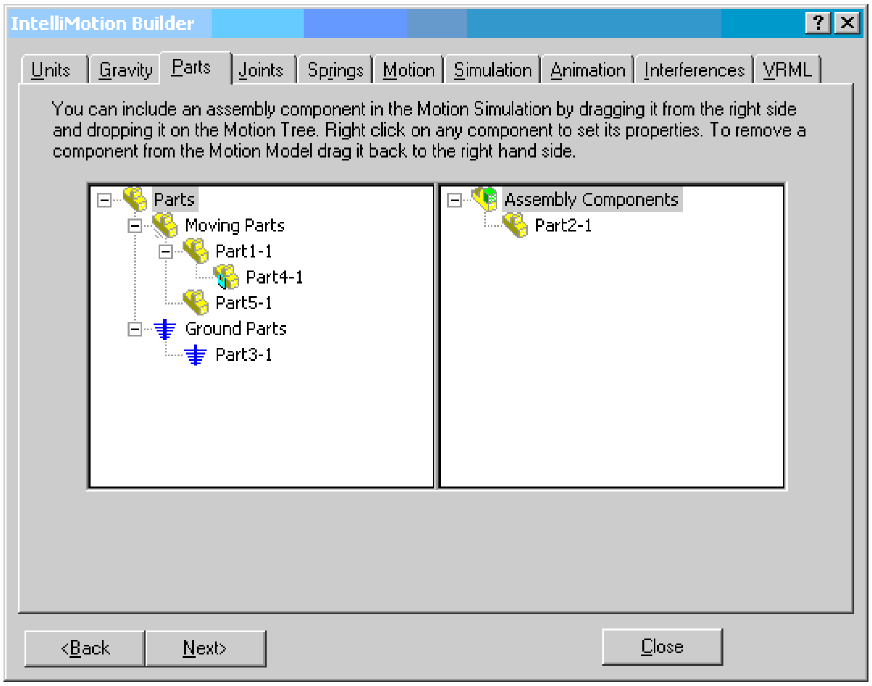
You can include an assembly component in the Motion Simulation by dragging it from the right side and dropping it on the Motion Tree. Right click on any component to set its properties. To remove a component from the Motion Model drag it back to the right hand side.

The page consists of two panes, each containing a tree. The tree on the right hand side displays of all the defined components in the assembly., and the tree on the left hand side contains two branches, one for Moving Parts, and one for non-moving or Ground Parts.

To include a component in the Motion model, simply drag the component from the right hand pane, and drop it on the appropriate branch in the left hand pane. As soon as a component is included in the Motion model, mass property symbols are added to the component in the drawing. If a component is dropped that shares an assembly constraint with another component already included in the motion model, the joints are automatically created between the two components based on the assembly constraints.

Rigidly Attaching Components

The parts page also allows a component to be dragged from the right side and dropped onto an component already included in the motion model. This action builds a rigid connection between the two components. It generates a COSMOSMotion fixed joint, and ignores any assembly constraints the two components may share. This feature is useful when there is a part that has other parts attached to it. A common example is fasteners such as nuts and bolts. It may be desired to have the fasteners move together with the part(s) they fasten. Since there is no relative motion between the part and its fasteners, they can be considered for the purpose of motion simulation as one single part, whose mass properties are the composite of all the individual mass properties. The picture below shows what an attached part looks like in the IntelliMotion Builder:



You can include an assembly component in the Motion Simulation by dragging it from the right side and dropping it on the Motion Tree. Right click on any component to set its properties. To remove a component from the Motion Model drag it back to the right hand side.

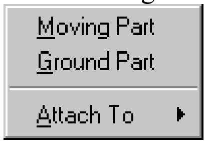
- Parts
  - Moving Parts
    - Part1-1
      - Part4-1
    - Part5-1
  - Ground Parts
    - Part3-1

- Assembly Components
  - Part2-1

In this case, the component PART4-1 has been rigidly attached to the moving component PART1-1. PART4-1 will move just like PART1-1. PART5-1 has been defined as a Moving Part, and PART3-1 as a non-moving or Ground Part.

Assembly Component Pop-Up Menu

The two panes on the part page also support right mouse button clicks. Highlighting a component in the right hand list and then clicking the right mouse button pops-up the following menu:



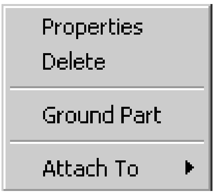
Selecting MOVING PART transfers the selected component to the left hand tree as a Moving Part.

Selecting GROUND PART transfers the selected component to the left hand tree as a Ground Part.

Selecting ATTACH TO displays another pop-up menu that lists all of the component that are part of the Motion Model. Clicking on one of them transfers the selected component to the left hand tree and rigidly attaches it to the component selected from the pop-up menu.

Motion Parts Pop-Up Menu

Highlighting a component in the left hand tree, in the Moving Parts branch, and then clicking the right mouse button pops-up the following menu:



Selecting PROPERTIES displays the Part Properties Dialog that allows part properties to be defined.

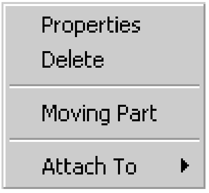
Selecting DELETE removes the component from the Motion Tree, and places it back in the right hand pane.

Selecting GROUND PART moves the component, and all attached components, from the Moving Parts branch to the Ground Parts branch.

Selecting ATTACH TO displays another pop-up menu that lists all of the component that are part of the Motion Model. Clicking on one of them transfers the selected component to the left hand tree and rigidly attaches it to the component listed on the pop-up menu.

Ground Parts Pop-Up Menu

Highlighting a component in the left hand tree, in the Ground Parts branch, and then clicking the right mouse button pops-up the following menu:



Selecting PROPERTIES displays the Ground Properties Dialog that allows part properties to be defined.

Selecting DELETE removes the component from the Motion Tree, and places it back in the right hand pane.

Selecting MOVING PART moves the component, and all attached components, from the Ground Parts branch to the Moving Parts branch.

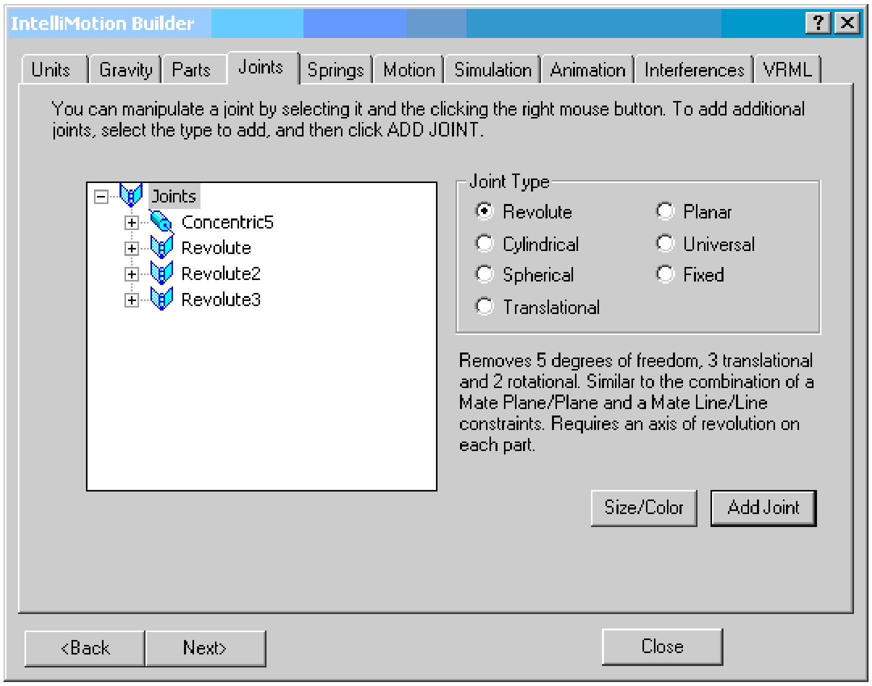
Selecting ATTACH TO displays another pop-up menu that lists all of the component that are part of the Motion Model. Clicking on one of them transfers the selected component to the left hand tree and rigidly attaches it to the component listed on the pop-up menu.

Dragging within the Left Pane

Parts may also be dragged and dropped from within the left hand pane. Dragging a component from the GROUND PART branch and dropping it on the MOVING PARTS branch changes it from a grounded part to a moving part. Dragging a component from the left hand pane to the right hand pane removes the component from the motion model.

IntelliMotion Builder Joint Page

The Joints page allows modification of joints that were automatically created from assembly constraints and additional joints to be added to the Motion model. This page contains a single tree control that shows all of the joints in the motion model.



You can manipulate a joint by selecting it and the clicking the right mouse button. To add additional joints, select the type to add, and then click ADD JOINT.

Joint Type

● Revolute ○ Planar
○ Cylindrical ○ Universal
○ Spherical ○ Fixed
○ Translational

Removes 5 degrees of freedom, 3 translational and 2 rotational. Similar to the combination of a Mate Plane/Plane and a Mate Line/Line constraints. Requires an axis of revolution on each part.

Pop-up Menus

Expanding one of the joint branches displays the components that the joint is attached to. Right clicking on a joint displays the following pop-up menu:

The PROPERTIES selection will display the Joint properties dialog to allow modification of the joint's properties.

The DELETE selection will delete the joint.

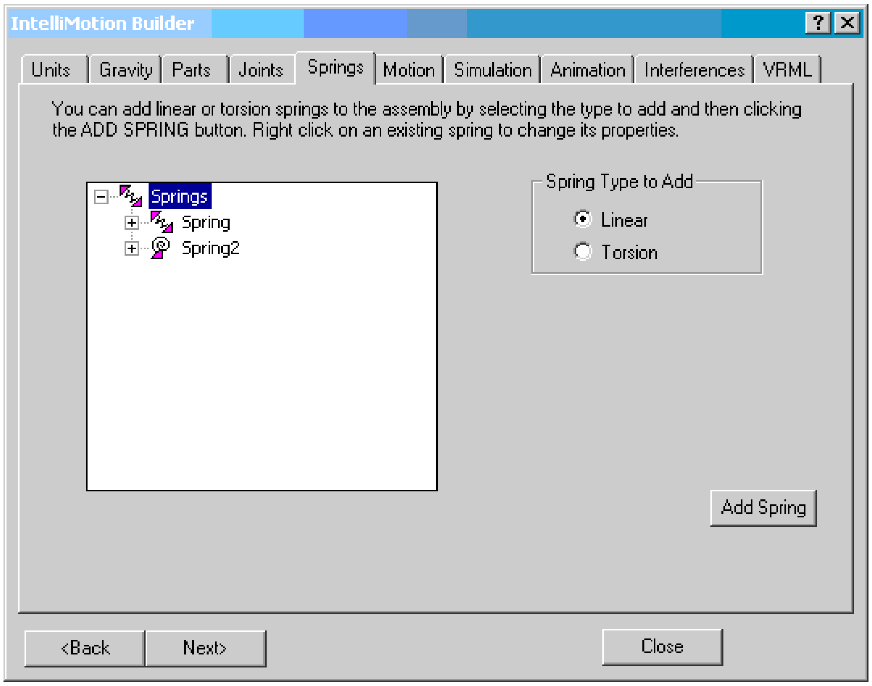
Manually Adding Joints

The right side of the joints page contains controls that allows joints to be added manually to the motion model. Clicking a particular joint type, and then clicking ADD JOINT will hide the IntelliMotion Builder and display the add joints dialog box.

Manually added joints can have the same operations performed on them as automatically created joints.

IntelliMotion Builder Springs Page

The Springs page allows linear and torsion springs to be added to the COSMOSMotion model. This page contains a single tree control with a single branch for springs that were added manually.



You can add linear or torsion springs to the assembly by selecting the type to add and then clicking the ADD SPRING button. Right click on an existing spring to change its properties.

Spring Type to Add

● Linear
○ Torsion

Expanding one of the spring branches displays the components that the spring is attached to. Right clicking on a spring displays the following pop-up menu:

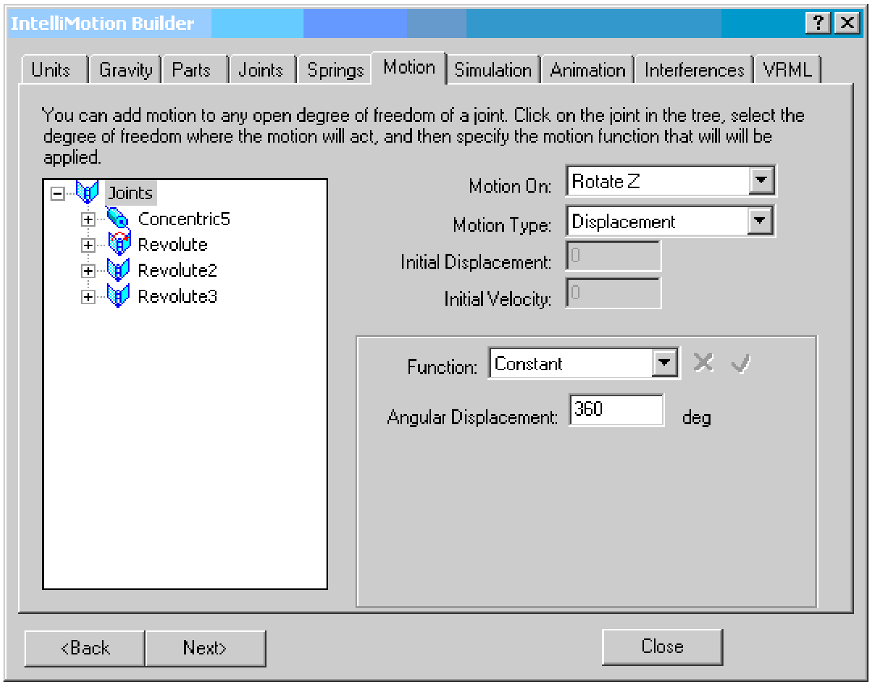
The PROPERTIES selection will display the Spring properties dialog to allow modification of the spring's properties.

The DELETE selection will delete the spring.

The right side of the page contains controls that allow springs to be added to the motion model. Clicking a particular spring type, and then clicking ADD SPRING will hide the IntelliMotion Builder and display the Add Springs dialog box.

IntelliMotion Builder Motion Page

The motion generators page allows motion generators to be added to the COSMOSMotion model. This page resembles the joints page in that the tree control on the left of the page contains a list of all the defined joints in the model. Selecting a joint in the left pane will display Motion Generator information on the right side of the dialog:



You can add motion to any open degree of freedom of a joint. Click on the joint in the tree, select the degree of freedom where the motion will act, and then specify the motion function that will will be applied.

Motion On: Rotate Z

Motion Type: Displacement

Initial Displacement: 0

Initial Velocity: 0

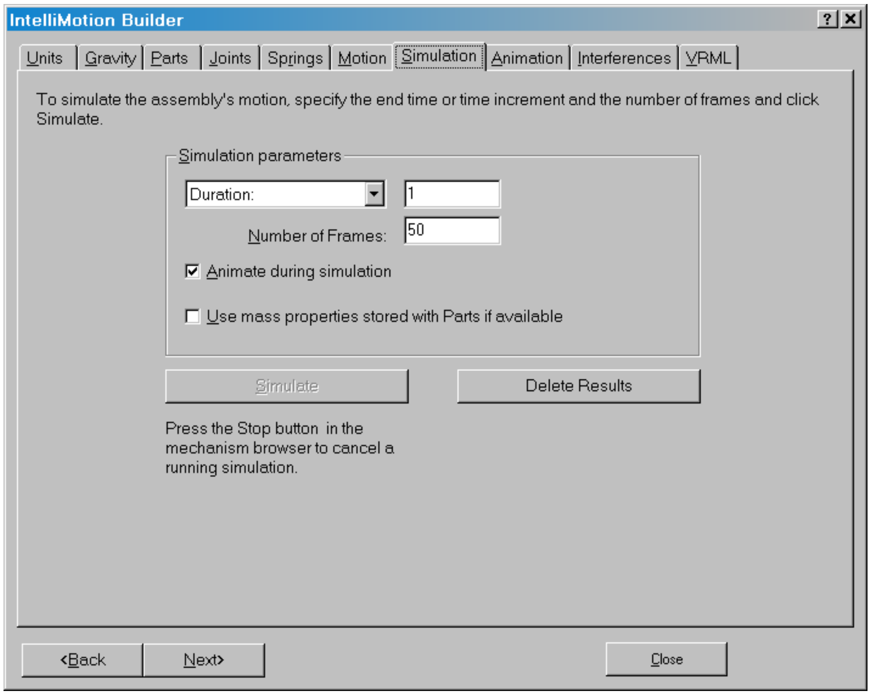
Function: Constant

Angular Displacement: 360 deg

The right side of the page contains controls that allow motion generators to be added to as an attribute of a joint. The command syntax used to create each motion generator type is the same as that for motion generators added from the menu.

IntelliMotion Builder Simulation Page

The Simulation page allows the simulation to be run.



To simulate the assembly's motion, specify the end time or time increment and the number of frames and click Simulate.

Simulation parameters

Duration: 1

Number of Frames: 50

☑ Animate during simulation

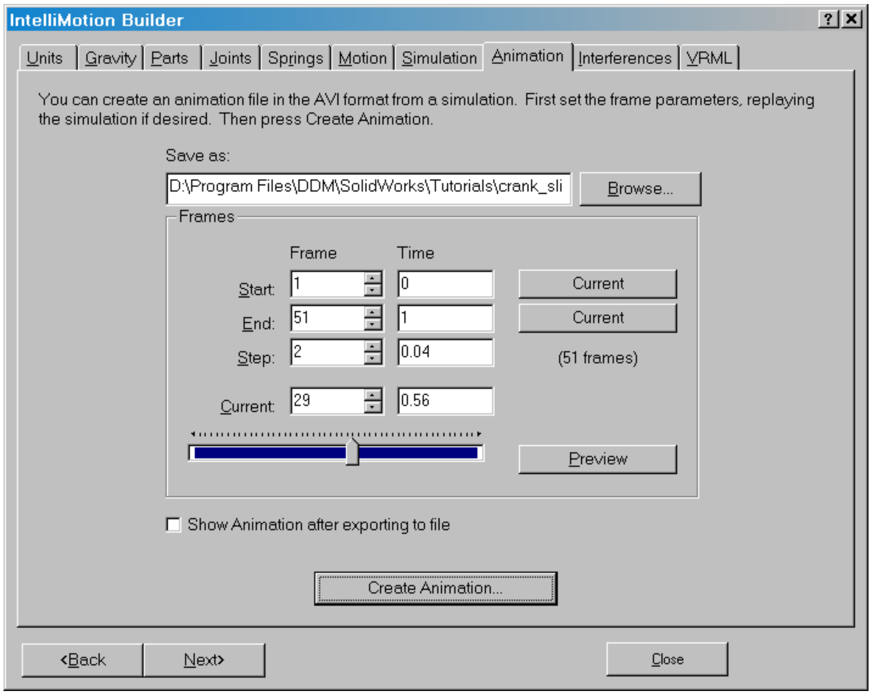
□ Use mass properties stored with Parts if available

Press the Stop button in the mechanism browser to cancel a running simulation.

Click the Simulate button to run the simulation.

IntelliMotion Builder Animation Page

The Animation page allows animation to be generated and replayed:



You can create an animation file in the AVI format from a simulation. First set the frame parameters, replaying the simulation if desired. Then press Create Animation.

Save as: D:\Program Files\DDM\SolidWorks\Tutorials\crank_sli

	Frame	Time
Start:	1	0
End:	51	1
Step:	2	0.04
Current:	29	0.56

(51 frames)

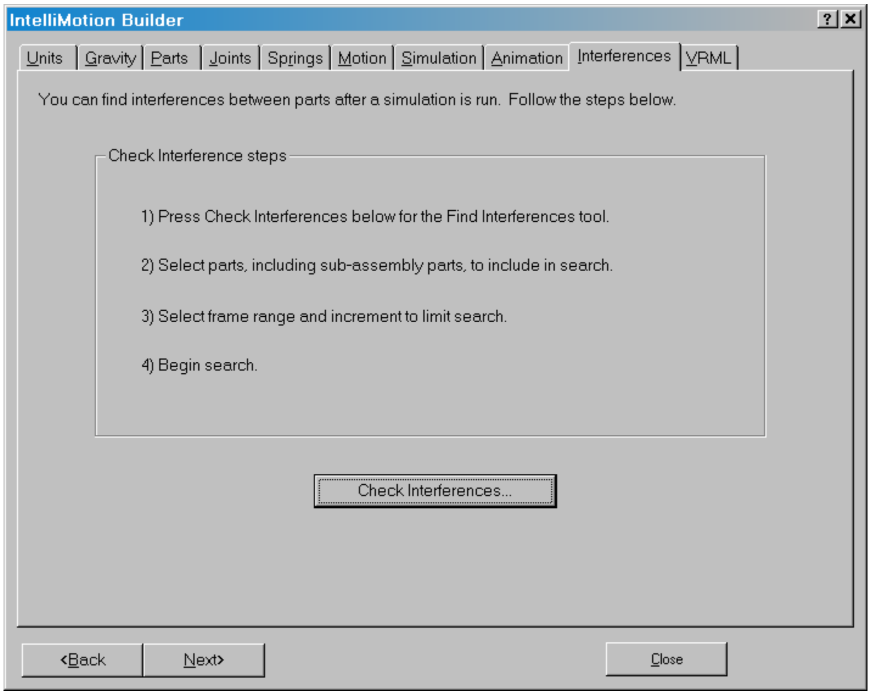
□ Show Animation after exporting to file

The first three edit boxes, those labeled “Start”, “End”, and “Step” control how many frames will be present in the animation.

When all the parameters have been specified, the button labeled “Create Animation” is used to initiate the animation generation process. The name of the Avi file to create will be prompted for. Each frame that will be in the animation is displayed, and copied and compressed into the animation file.

IntelliMotion Builder Interference Page

The Interference page allows interference conditions to be checked and identified:



You can find interferences between parts after a simulation is run. Follow the steps below.

Check Interference steps

1) Press Check Interferences below for the Find Interferences tool.

2) Select parts, including sub-assembly parts, to include in search.

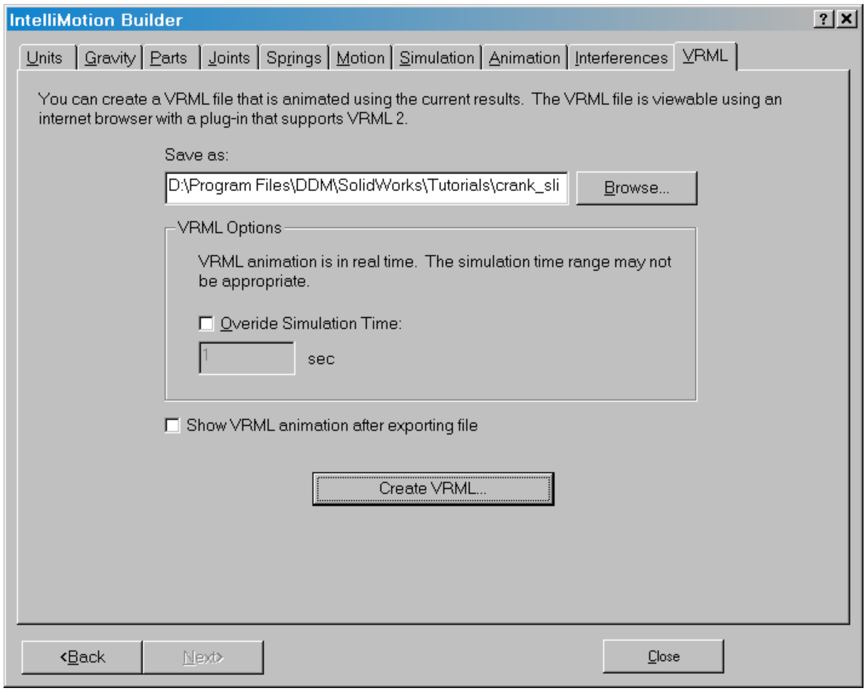
3) Select frame range and increment to limit search.

4) Begin search.

Click the **Check Interferences** button to display the Check Interference dialog. See page 139 for more information.

IntelliMotion Builder VRML Page

The VRML page allows VRML files of the assembly's motion to be created:



You can create a VRML file that is animated using the current results. The VRML file is viewable using an internet browser with a plug-in that supports VRML 2.

Save as: D:\Program Files\DDM\SolidWorks\Tutorials\crank_sli

VRML Options

VRML animation is in real time. The simulation time range may not be appropriate.

□ Overide Simulation Time: 1 sec

□ Show VRML animation after exporting file

Click the "Create VRML" button to generate a VRML file. The name of the file will be prompted for. See page 133 for more information on the options.

8

Interfacing to MSC.ADAMS

COSMOSMotion can export your mechanism to the full MSC.ADAMS Simulation product for complex simulations involving flexible bodies, control systems, user subroutines, etc.

MSC.ADAMS Dataset File

COSMOSMotion allows your mechanism model to be exported in a format suitable for use by the MSC.ADAMS Full Simulation product. It can export an MSC.ADAMS dataset file along with geometry for the parts included in the mechanism.

The information in the MSC.ADAMS dataset file is based upon the current configuration of your assembly (the position of your parts) at the time the mechanism is stored. If you remove any parts or subassemblies, any mechanism entities that were attached to those parts or subassemblies will be deleted.

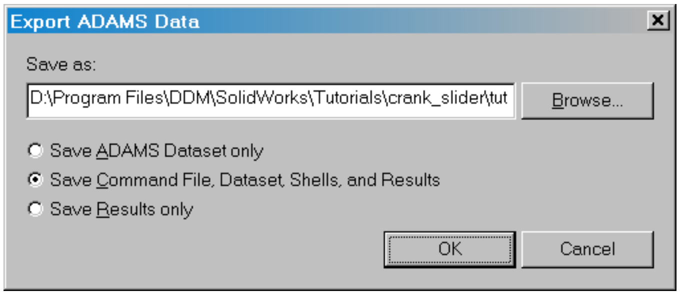
Exporting Your Model to MSC.ADAMS

By exporting your model to MSC.ADAMS, you gain the ability to perform complex dynamic simulations including flexible parts, tires, control systems, etc. You can export your model to MSC.ADAMS/View, where you can access the MSC.ADAMS/Solver, make modifications, animate, and plot results.

To export your mechanism to MSC.ADAMS/View:

1 Export the MSC.ADAMS dataset file by choosing either **Motion, File, Export Mechanism** from the Menu, or by selecting the **Motion Model** branch in the Browser and choosing **Export Simulation** from the pop-up menu. You can also select the toolbar button**.**

2 The following dialog will be displayed:



Export ADAMS Data

Save as: D:\Program Files\DDM\SolidWorks\Tutorials\crank_slider\tut

○ Save ADAMS Dataset only

● Save Command File, Dataset, Shells, and Results

○ Save Results only

Enter the base name of the MSC.ADAMS files to be created, or click the **Browse** button to use the standard Windows File Save As dialog to specify the file name.

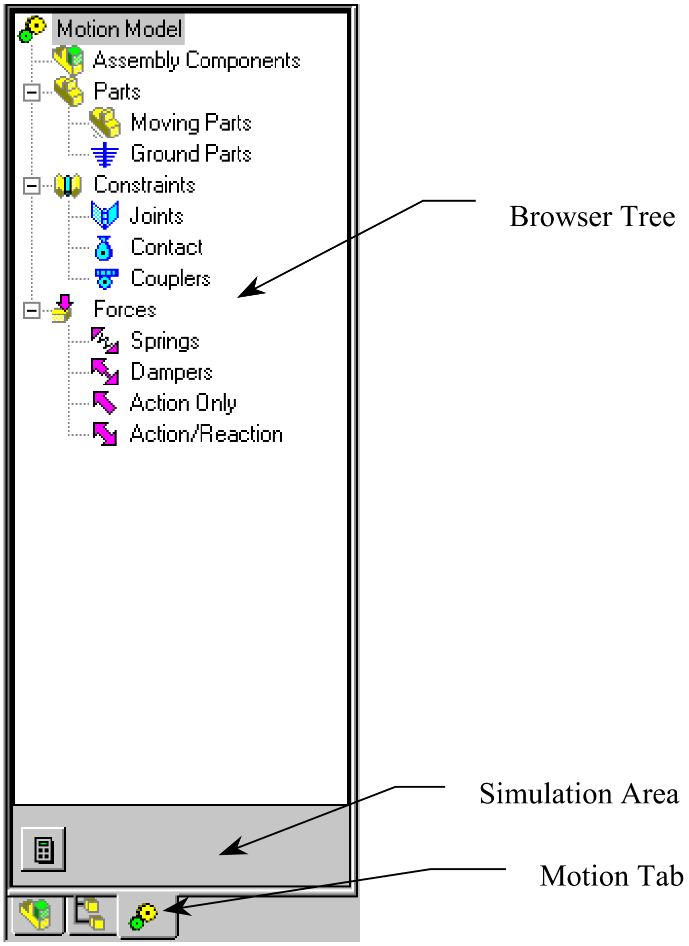
3 Choose one of the options. **Save MSC.ADAMS Dataset only,** will output an MSC.ADAMS ADM file. **Save Command File, Dataset, Geometry, and Results**, will output a complete model suitable to be imported to MSC.ADAMS/View. The geometry in your model is output as Parasolid bodies, the motion data is output as an ADM file, any results that exist are output as a RES file, and an MSC.ADAMS/View CMD file is created to automatically import and build the model in MSC.ADAMS/View.

4 Click **OK** to start the process.

9

IntelliMotion Browser

This chapter explains in detail the operation and options available in the IntelliMotion Browser.



The IntelliMotion Browser contains two separate areas:

1. The Browser Tree is the hierarchical view of the motion model. All drag and drop operations, and right mouse button activated pop-up menus occur in this area.
2. The Simulation Area contains controls that, depending on the mode of the motion model, either starts the simulation or remove simulation results.

Activating the Browser

The IntelliMotion Browser is activated by clicking on the tab in the SolidWorks Feature Manager.

Detailed Browser Documentation

Due to the extensive amount of information on the Browser and it's operation, this section of the documentation is only available in the on-line help. Please see the on-line help topic **IntelliMotion Browser** for more information.

10

MSC.ADAMS Functions

This appendix explains MSC.ADAMS function expressions and describes of all of the MSC.ADAMS functions supported in COSMOSMotion. You can enter these functions in the expression fields on the motion or force Property panels. You can use these function expressions to define a wide variety of non-standard phenomena in your model. See Creating Motions or Creating Forces for detailed information on how to create the function expressions to define a force or a motion.

Function Expression Basics

A function expression is a single C-like or FORTRAN-like expression that evaluates a single value. This value may be a function of system variables such as displacements between markers, velocities, accelerations, and current simulation time.

Syntax and Notation

A function expression may be composed of any valid combination of simple constants, operators, parameters, and available standard MSC.ADAMS/Solver functions.

Simple Constants

Function expressions may include integers and real numbers only. Complex numbers are not supported at this time. Any legal number is accepted by the function expression. The definition of a legal integer and a legal real number is quite machine dependent, and may therefore change from system to system. Typically in MSC.ADAMS/Solver, an integer is limited to having an absolute value less than $2^{31\ -}$ ^{1}. A real number in MSC.ADAMS/Solver is typically limited to having an absolute value less than 10^{26}.

Operators

MSC.ADAMS/Solver allows the standard FORTRAN-77 set of operators. The operators and their sequence of evaluation are shown in the table below.

Symbol	Operation	Precedence
**	Exponentiation	1
/	Division	2
*	Multiplication	3

A function expression can contain any number of blank spaces. They are often used to improve the readability of the function expression. The following restrictions exist on the usage of blanks:

- ❑ You cannot put a blank space in the middle of a number.
- ❑ MSC.ADAMS/Solver does not accept a blank space between a function and its left bracket.

Limitations

MSC.ADAMS/Solver imposes some limitations in the definition of functions.

Nesting of Functions

Functions, sub-expressions or operations may be nested only ten levels deep. For example: FUNCTION = f1 (f2 (f3 (f4))), where f1, f2, f3, f4 are function sub-expressions, is a syntactically legal function definition. However, FUNCTION = f1 (f2 (... (f11) ...) is not a legal definition, since sub-expressions have been nested eleven levels deep.

Limitations on Number of Symbols in a Function Expression

A single function expression must not use more than one thousand symbols. This includes operators, brackets, functions, number, etc.

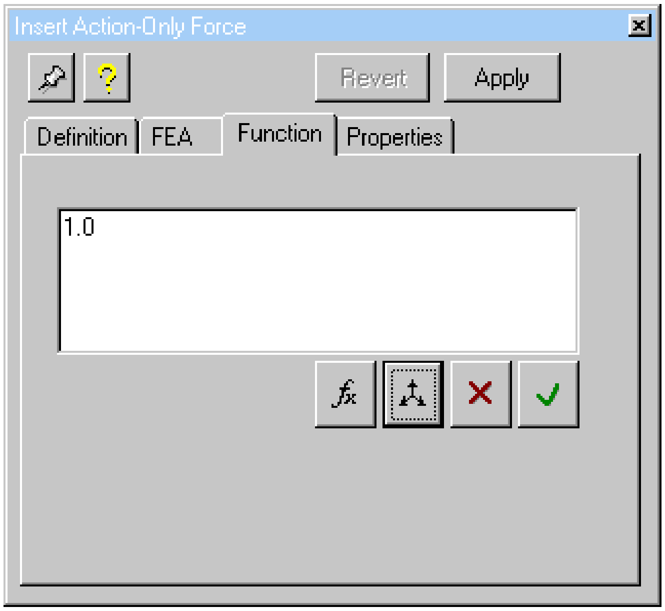
Limitations on Number of Elements to Which a Function Can Refer

A function is limited to being dependent on a maximum of 25 standard MSC.ADAMS/Solver elements of each type. Therefore, each expression may be a function of a maximum of 25 PARTS, 25 FORCES, etc.

Markers

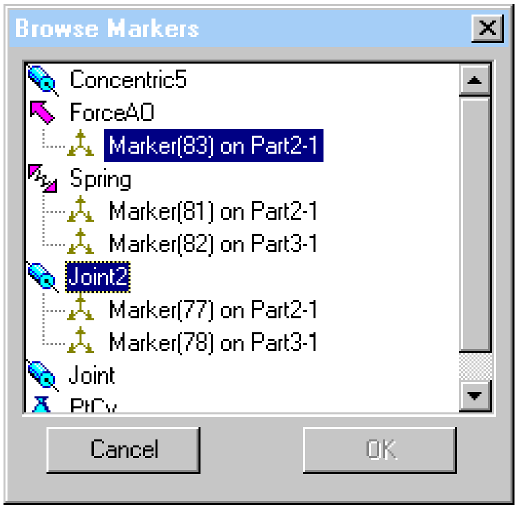
Many of the function supported by MSC.ADAMS require the use of marker numbers. Markers are the means by which motion entities are attached to parts, internal to MSC.ADAMS.

Any of the entities that support function, such as joints, forces, and moments, has a **Function** tab that display a dialog box similar to the one below:



1.0

Clicking on the button will display the follow dialog:



This dialog can be used to browse all of the motion entities in the current model and be used to access the marker numbers of any entity in the motion model. If you are using a function such as AX that uses at least one marker number as an argument, you can enter the **AX(** portion of the function in the function edit box, click the button and select a marker. The marker number will be automatically added to the function statement.

Supported Functions

Below, the MSC.ADAMS functions expressions are grouped according to their function. Detailed information on each function expression is found on the page indicated after its name below.

FORTRAN 77 Functions

Function Name	Page No.	Summary Definition
ABS	192	Absolute value of (a)
ACOS	197	Arc cosine of (a)
AINT	198	Nearest integer whose magnitude is not larger than (a)
ANINT	199	Nearest whole number to (a)
ASIN	200	Arc sine of (a)
ATAN	201	Arc tangent of (a)
ATAN2	202	Arc tangent of (a1, a2)
COS	210	Cosine of (a)
COSH	211	Hyperbolic cosine of (a)

DIM	212	Positive difference of a1 and a2
EXP	218	e raised to the power of (a)
LOG	229	Natural logarithm of (a)
LOG10	231	Log to base 10 of (a)
MAX	232	Maximum of a1 and a2
MIN	233	Minimum of a1 and a2
MOD	234	Remainder when a1 is divided by a2
SIGN	245	Transfer sign of a2 to magnitude of a1
SIN	246	Sine of (a)
SINH	247	Hyperbolic sine of (a)
SQRT	248	Square root of a1
TAN	252	Tangent of (a)
TANH	253	Hyperbolic tangent of (a)

Simulation Constants and Variables

Function Name	Page No.	Summary Definition
DTOR	213	Degrees to radians conversion factor
PI	237	Ratio of circumference to diameter of a circle
RTOD	243	Radians to degrees conversion factor
TIME	255	Current simulation time

Arithmetic IF

Function Name	Page No.	Summary Definition
IF	227	Defines a function expression

General Functions

Function Name	Page No.	Summary Definition
CHEBY	207	Evaluates a Chebyshev polynomial
FORCOS	220	Evaluates a Fourier Cosine series
FORSIN	222	Evaluates a Fourier Sine series
POLY	239	Evaluates a standard polynomial at a user specified value x
SHF	244	Evaluates a simple harmonic function
STEP	249	Approximates the Heaviside step function with a cubic polynomial
STEP5	251	Approximates the Heaviside step function with a quintic polynomial

Result Dependant Functions

Displacement	Page No.	Summary Definition
AX	203	Rotational displacement about x-axis
AY	204	Rotational displacement about y-axis
AZ	205	Rotational displacement about z-axis
DM	213	Magnitude of linear displacement
DX	215	Translational displacement in x-axis
DY	216	Translational displacement in y-axis
DZ	217	Translational displacement in z-axis
PHI	236	Third Euler rotation angle
PITCH	238	Second angle of yaw-pitch-roll rotation
PSI	241	First Euler rotation angle
ROLL	242	Third angle of yaw-pitch-roll rotation
THETA	254	Second Euler rotation angle
YAW	273	First angle of yaw-pitch-roll rotation
Velocity	**Page No.**	**Summary Definition**
VM	260	Magnitude of translational velocity
VR	261	Relative velocity
VX	262	Translational velocity in x-axis
VY	263	Translational velocity in y-axis
VZ	264	Translational velocity in z-axis
WM	269	Magnitude of angular velocity
WX	270	Angular velocity about x-axis
WY	271	Angular velocity about y-axis
WZ	272	Angular velocity about z-axis
Acceleration	**Page No.**	**Summary Definition**
ACCM	193	Magnitude of translational acceleration
ACCX	194	Translational acceleration in x-axis dir.
ACCY	195	Translational acceleration in y-axis dir.
ACCZ	196	Translational acceleration in z-axis dir.
WDTM	265	Magnitude of angular acceleration
WDTX	266	Angular acceleration about the x-axis
WDTY	267	Angular acceleration about the y-axis
WDTZ	268	Angular acceleration about the z-axis

Force	Page No.	Summary Definition
FM	219	Magnitude of Linear Force
FX	224	Linear force in x-axis direction
FY	225	Linear force in y-axis direction
FZ	226	Linear force in z-axis direction
TM	256	Magnitude of Torque
TX	257	Torque about x-axis
TY	258	Torque about y-axis
TZ	259	Torque about z-axis

ABS

Definition

The ABS function returns the absolute value of the expression a.

Format

ABS(a)

Arguments

a Any valid function expression.

Examples

ABS(-10*time+15*time**2)

This function returns the absolute instantaneous value of the expression (-10*time+15*time**2).

ABS(DX(21,11)*VX(21,11)+DY(21,11)*VY(21,11)

+ DZ(21,11)*VZ(21,11))

This function returns the absolute value of the dot product of the

1) displacement vector of Marker 21 with respect to Marker 11, and
2) velocity vector of Marker 21 with respect to Marker 11.

ACCM

Definition

The ACCM function calculates the magnitude of the second time derivative of the displacement vector of marker i from marker j. The time derivatives are taken in the reference frame of marker l. Markers j and l default to the global coordinate system and the global reference frame if they are not specified.

Format

ACCM(i[,j][,l])

Arguments

i	The marker whose acceleration is being measured.
j	The marker with respect to which the acceleration is being measured. Set j = 0, while still specifying l, if you want j default to the global coordinate system.
l	The reference frame in which the second time derivative of the displacement vector is taken. Set l = 0 if you want the time derivatives to be calculated in the ground reference frame.

Examples

ACCM(21,11)

This statement obtains the magnitude of translational acceleration of Marker 21 with respect to Marker 11. Since the l marker is not specified, the derivatives are taken in the inertial reference frame.

ACCM(21,11,32)

This statement obtains the magnitude of the translational acceleration of Marker 21 with respect to Marker 11. Vector time derivatives taken in reference to Marker 32.

ACCX

Definition

The ACCX function returns the x-component of the difference between the acceleration vector of marker i and the acceleration vector of marker j as computed in the coordinate system of marker k. All vector time derivatives are taken in the reference frame of marker l. Marker j defaults to the global coordinate system if it is not specified. Similarly, markers k and l default to the global coordinate system and the global reference frame if they are not specified.

Format

ACCX(i[,j][,k][,l])

Arguments

i The marker whose acceleration is being measured.

j The marker with respect to which the acceleration is being measured. Set j = 0, while still specifying l, if you want j to default to the global coordinate system.

k The marker in whose coordinate system the acceleration vector is being expressed. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

l The reference frame in which the second time derivative of the displacement vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

ACCX(21,0,31)

This statement obtains the x-component of the acceleration vector of Marker 21 with respect to the global origin. The vector is expressed in the coordinate system of Marker 31. All time derivatives are taken in the inertial reference frame since the l marker is not specified.

ACCY

Definition

The ACCY function returns the y-component of the difference between the acceleration vector of marker i and the acceleration vector of marker j as computed in the coordinate system of marker k. All vector time derivatives are taken in the reference frame of marker l. Marker j defaults to the global coordinate system if it is not specified. Similarly, markers k and l default to the global coordinate system and the global reference frame if they are not specified.

Format

ACCY(i[,j][,k][,l])

Arguments

i The marker whose acceleration is being measured.

j The marker with respect to which the acceleration is being measured. Set j = 0, while still specifying l, if you want j to default to the global coordinate system.

k The marker in whose coordinate system the acceleration vector is being expressed. Set k = 0 if you want the results to be calculated along the y-axis of the global coordinate system.

l The reference frame in which the second time derivative of the displacement vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

ACCY(21,11)

This statement obtains y-component of the acceleration of Marker 21 with respect to Marker 11. The coordinate system for calculating the component, and for referencing the frame that takes time derivatives, defaults to ground.

ACCZ

Definition

The ACCZ function returns the z-component of the difference between the acceleration vector of marker i and the acceleration vector of marker j as computed in the coordinate system of marker k. All vector time derivatives are taken in the reference frame of marker l. Marker j defaults to the global coordinate system if it is not specified. Similarly, markers k and l default to the global coordinate system and the global reference frame if they are not specified.

Format

ACCZ(i [,j][,k][,l])

Arguments

i The marker whose acceleration is being measured.

j The marker with respect to which the acceleration is being measured. Set j = 0, while still specifying l, if you want j default to the global coordinate system.

k The marker in whose coordinate system the acceleration vector is being expressed. Set k = 0 if you want the results to be calculated along the z-axis of the global coordinate system.

l The reference frame in which the first time derivative of the displacement vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Example

ACCZ(21,0,31)

This statement obtains the z-component of the acceleration vector of Marker 21 with respect to the global origin. The vector is expressed in the coordinate system of Marker 31. All time derivatives are taken in the inertial reference frame since the l marker is not specified.

ACOS

Definition

The `ACOS` function calculates the arc cosine of an expression `a`. Expression `a` must evaluate to a value whose absolute value is less than or equal to 1. The value returned by `ACOS` lies in the range [0,p] (i.e., 0 `ACOS(a)` p).

Format

ACOS(a)

Arguments

a Any valid function expression.

Examples

ACOS((DX(21,11)*DX(31,41)+DY(21,11)*DY(31,41)

+ DZ(21,11)*DZ(31,41))/(DM(21,11)*DM(31,41))

This function calculates the angle between the line from Markers 11 to 21 and the line from Markers 41 to 31. The result is in radians.

AINT

Definition

The `AINT` function returns the nearest integer whose magnitude is not larger than the integer value of the expression `a`.

`AINT(a)`= 0 if `ABS(a)`< 1

`AINT(a)=int(a)` if `ABS(a)` 1

The value of the mathematical function `int` of a variable `x` is equal to `x` if `x` is an integer. If `x` is not an integer, then `int(x)` is equal to the nearest integer to `x` whose magnitude is not greater than the magnitude of `x`. Thus `int`(-7.0) = -7, `int`(-4.8) = -4, and `int`(4.8) = 4.

Format

AINT(a)

Arguments

a Any valid function expression.

Cautions

Note that `AINT` is not a differentiable function. Be careful when using this function in an expression that defines a force or motion input to the system.

Examples

AINT(-6.5)= 0

AINT(4.6)= 4

ANINT

Definition

The ANINT function calculates the nearest integer whose magnitude is not larger than the real value of the expression a.

ANINT(a) = int(a + .5) if a 0

ANINT(a) = int(a - .5) if a < 0

The value of the mathematical function int of a variable x is equal to x if x is an integer. If x is not an integer, then int(x) is equal to the nearest integer to x whose magnitude is not greater than the magnitude of x. Thus int(-7.0) = -7, int(-4.8) = -4, and int(4.8) = 4.

Format

ANINT(a)

Arguments

a Any valid function expression.

Examples

ANINT(-4.6)= -5

ANINT(4.6)= 5

ASIN

Definition

The `ASIN` function computes the arc sine of an expression `a`. `ASIN` is defined only when the absolute value of `a` is less than or equal to 1. The range of `ASIN` is [-p/2, p/2] (i.e., -p/2 `ASIN(a)` p/2).

Format

ASIN(a)

Arguments

a Any valid function expression.

Examples

ASIN(DX(21,11)/DM(21,11))

This function calculates the value of the expression `DX(21,11)/DM(21,11)` and then applies the `ASIN` function to the result. The result is in radians.

ATAN

Definition

The ATAN function returns the arc tangent of an expression a. The range of ATAN is [-p/2, p/2] (i.e., -p/2 < ATAN(a)< p/2).

Format

ATAN(a)

Arguments

a Any valid function expression.

Examples

```
SFORCE/1, ROTATIONAL, I=21, J=11
, FUNCTION=-10*ATAN(DY(42,33)/DX(42,33))
```

This SFORCE statement defines a torque acting between Markers 21 and 11. The value of the torque is the spring constant (10) multiplied by the angle between the line joining Markers 33 and 42 and the global x-axis.

ATAN2

Definition

The `ATAN2` function expression returns the arc tangent of the expression `a1/a2.` `a1` and `a2` themselves may be expressions.

-p < `ATAN2(a1,a2)` < p

`ATAN2(a1,a2)` > 0 if `a1` > 0

`ATAN2(a1,a2)` = 0 if `a1` = 0, `a2` > 0

`ATAN2(a1,a2)` = p if `a1` = 0, `a2` < 0

`ATAN2(a1,a2)` < 0 if `a1` < 0

`ABS(ATAN2(a1,a2))`= p/2 if `a2` = 0

`ATAN2(a1,a2)` undefined if `a1` = 0, and `a2` = 0

Format

ATAN2(a1,a2)

Arguments

a1	Any valid function expression.
a2	Any valid function expression.

Examples

VARIABLE/1,

, FUNCTION=ATAN2(DY(21,31,41), DX(21,31,41))

This `VARIABLE` statement defines an algebraically-determined, user-defined state variable in MSC.ADAMS/Solver. Its value is specified by the expression containing `ATAN2`.

AX

Definition

The AX function returns the rotational displacement of marker i about the x-axis of marker j. Marker j defaults to the global coordinate system if it is not specified. This value is computed as follows: assume that rotations about the other two axes (y-, z-axes) of marker j are zero. Then AX is the angle between the two y-axes (or the two z-axes). AX is measured in a counter- clockwise sense from the y-axis of the J marker to the y-axis of the I marker.

Format

AX(i[,j])

Arguments

i	The marker whose rotations are being sought.
j	The marker with respect to which the rotations are being measured.

Examples

-20*AX(43,32)

The value of the function is -20 times the angle between the y axes of Markers 43 and 32. The angle is measured in a counterclockwise sense from the y-axis of Marker 32 to the y-axis of Marker 43.

AY

Definition

The AY function returns the rotational displacement of marker i about the y-axis of marker j. Marker j defaults to the global coordinate system if it is not specified. This value is computed as follows: assume that rotations about the other two axes (x-, z-axes) of marker j are zero. Then AY is the angle between the two x-axes (or the two z-axes). AY is measured in a counter- clockwise sense from the x-axis of the J marker to the x-axis of the I marker.

Format

AY(i[,j])

Arguments

i	The marker whose rotations are being sought.
j	The marker with respect to which the rotations are being measured.

Examples

-4*(AY(46,57)**2)

The value of the function is -4 times the square of the angle between the x axes of Markers 46 and 57. The angle is measured in a counterclockwise sense from the x-axis of Marker 57 to the x-axis of Marker 46.

AZ

Definition

The AZ function returns the rotational displacement of marker i about the z-axis of marker j. Marker j defaults to the global coordinate system if it is not specified. This value is computed as follows: assume that rotations about the other two axes (x-, y-axes) of marker j are zero. Then AZ is the angle between the two x-axes (or the two y-axes). AZ is measured in a counter- clockwise sense from the x-axis of the J marker to the x-axis of the I marker.

Format

AZ(i[,j])

Arguments

i The marker whose rotations are being sought.

j The marker with respect to which the rotations are being measured.

Examples

-30*(AZ(21,32)-25D)

BISTOP

Definition

The `BISTOP` function models a gap element.

Format

`BISTOP (x, x&,` $x_{1,}$ $x_{2,}$ `k, e, c`max, `d)`

Arguments

x A real variable that specifies the distance variable you want to use to compute the force. For example, if you want to use the `x` displacement of Marker 0201 with respect to 0301, then `x` is `DX(0201,0301,0301)`.

x& A real variable that communicates the time derivative of `x` to `BISTOP`. For example, if `x` is `DX(0201,0301,301)`then **x&** is `VX(0201,0301,0301,0301)`.

x1 A real variable that specifies the lower bound of `x`. If x is less than `x1` MSC.ADAMS/Solver calculates a positive value for the force. The value of `x1` must be less than the value of `x2`.

x2 A real variable that specifies the upper bound of `x`. If `x` is greater than `x2` MSC.ADAMS/Solver calculates a negative value for the force. The value of `x2` must be greater than the value of `x1`.

k A non-negative real variable that specifies the stiffness of the boundary surface interaction.

e A positive real variable that specifies the exponent of the force deformation characteristic. For a stiffening spring characteristic, `e` > 1.0. For a softening spring characteristic, 0 < `e` < 1.0.

cmax A non-negative real variable that specifies the maximum damping coefficient.

d A positive real variable that specifies the boundary penetration at which MSC.ADAMS/Solver applies full damping.

Extended Definition

The `BISTOP` function models a gap element. The gap element consists of a slot which defines the domain of motion of a Part I located in the slot. As long as Part I is located in the slot and has no interference with the ends of the slot, it is free to move without forces acting on it. When Part I tries to move beyond the physical definition of the slot, impact forces representing contact are created by the `BISTOP` function. The created force tends to move Part I back into the slot.

The `BISTOP` force has two components: A stiffness component dependent on the penetration of Part I into the restricting Part J and a damping or viscous component that may be used to model energy loss.

To prevent a discontinuity in the damping force at zero penetration, the damping coefficient is defined as a cubic step function of the penetration. Thus at zero penetration, the damping coefficient is always zero. The damping coefficient achieves a maximum, `c`max, at a user-defined penetration, `d`. Even though the points of contact between the floating part and the restricting part may change as the system moves, MSC.ADAMS/Solver always exerts the force between the `I` and the `J` markers.

CHEBY

Definition

The CHEBY function evaluates a Chebyshev polynomial at a user specified value x.

Format

CHEBY (x, x_0, a_0, a_1,..., a_{30})

Arguments

x A real variable that specifies the independent variable. For example, if the independent variable in the function is time, x is the system variable TIME.

x_0 A real variable that specifies a shift in the Chebyshev polynomial.

a_0,a_1,...,a_{30} The real variables that define as many as thirty-one coefficients for the Chebyshev polynomial.

Extended Definition

The CHEBY function evaluates a Chebyshev polynomial at a user specified value x. x_0, a_0, a_1, ..., a_{30} are parameters used to define the constants for the Chebyshev polynomial. The Chebyshev polynomial is defined as:

C(x) = å $a_j T_j$ (x-x0)

where the functions T_j are recursively defined as:

T_j(x-x0) = 2 * (x-x0) * T_{j-1}(x-x0) - T_{j-2}(x-x0)

where T0 (x-x_0) = 1, and T_1 (x-x0) = x-x0.

The index "j" has a range from zero to "n", where "n" is the number of terms in the series.

Note that:

T2 $(x-x_0) = 2 * (x- x_0)^2 - 1$

T3 $(x-x_0) = 4 * (x- x_0)^3 - 3 * (x- x_0)$

Examples

MOTION/1, JOINT=21, TRANSLATION,

,FUNCTION = IF (TIME-2:CHEBY(TIME, 1, 1, 0, -1), 0, 0)

This MOTION statement defines a motion using a quadratic Chebyshev polynomial and the system variable TIME. The arithmetic IF ensures that the function remains zero after 2 time units. When time is less than 2 time units, MSC.ADAMS/Solver evaluates a Chebyshev polynomial to determine the motion. The polynomial defined in the above example is:

Cheby = 1 + 0 * (time-1) - 1 * [2 (time-1)2 - 1]
      = 2*time2 - 4*time

COS

Definition

The `COS` function returns the cosine of an expression `a`.

Format

COS(a)

Arguments

a Any valid function expression.

Examples

MOTION/2060, JOINT=2060, ROTATION,

, FUNCTION=20D+COS(2*PI*TIME)

This `MOTION` statement defines a rotational motion acting on `JOINT/2060`. The rotational degrees of freedom is specified to be the function `20D+COS(2*PI*TIME)`.

COSH

Definition

The COSH function computes the hyperbolic cosine of an expression a.

COSH(a) = $(e^{a} + e^{-a}) / 2.0$

Format

COSH(a)

Arguments

a Any valid function expression.

Examples

```
REQUEST/36,
, F2=COSH(DZ(21,11,11))
```

This REQUEST statement returns the hyperbolic cosine of the z-component of the displacement of Marker 21 with respect to Marker 11 in its second column. The result is computed in the coordinate system of Marker 11.

DIM

Definition

The DIM function calculates the positive difference of the instantaneous values of two expressions a1 and a2.

DIM(a1,a2) = 0 if a1 a2

DIM(a1,a2) =a1-a2 if a1 > a2

Format

DIM(a1,a2)

Arguments

a1 Any valid function expression.

a2 Any valid function expression.

Cautions

❑ DIM is a discontinuous function and must be used with caution.

DM

Definition

The DM function returns the magnitude of the translational displacement vector from marker j to marker i. Marker j defaults to the global coordinate system if it is not specified. DM is the distance between markers i and j and by definition is always non-negative.

Format

DM(i[,j])

Arguments

i The marker whose origin is being measured.

j The marker whose origin is the reference point for the displacement calculation.

Examples

-30*(DM(21,11)-25)**1.72

DTOR

Definition

The DTOR function returns the degrees to radians conversion factor (PI/180).

Format

DTOR

Examples

VARIABLE/1, FUNCTION=30*DTOR*TIME

This VARIABLE statement specifies a user-defined, algebraically-determined state variable whose value is 30 degrees*time. The state variable is stored in radians.

DX

Definition

Returns the x-component of the translational displacement vector from marker j to marker i as expressed in marker k coordinate system. Marker j defaults to the global coordinate system if it is not specified. Similarly, marker k defaults to ground if it is not specified.

Format

DX(i[,j][,k])

Arguments

i The marker whose origin is being measured.

j The marker whose origin is the reference point for the displacement calculation.

k The marker in whose coordinates the x-component of the displacement vector is being calculated. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

Examples

DX(21,11,32)**2

This function is the square of the x-displacement of Marker 21 with respect to Marker 11 as computed in the coordinate system of Marker 32.

DY

Definition

The `DY` function returns the y-component of the translational displacement vector from marker `j` to marker `i` as expressed in the marker `k` coordinate system. Marker `j` defaults to the global coordinate system if it is not specified. Similarly, marker `k` defaults to the global coordinate system if it is not specified.

Format

DY(i[,j][,k])

Arguments

i The marker whose origin is being measured.

j The marker whose origin is the reference point for the displacement calculation. Set `j=0` if you want `j` to default to the global coordinate system while still specifying `l`.

k The marker in whose coordinates the y-component of the displacement vector is being calculated. Set `k` = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

Examples

DY(21,11,32)**2

This function is the square of the y-displacement of Marker 21 with respect to Marker 11 as computed in the coordinate system of Marker 32.

DZ

Definition

The DZ function returns the z-component of the translational displacement vector from marker j to marker i as expressed in the marker k coordinate system. Marker j defaults to the global coordinate system if it is not specified. Similarly marker k defaults to the global coordinate system if it is not specified.

Format

DZ(i[,j][,k])

Arguments

i The marker whose origin is being measured.

j The marker whose origin is the reference point for the displacement calculation. Set j=0 if you want j to default to the global coordinate system while still specifying l.

k The marker in whose coordinates the z-component of the displacement vector is being calculated. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

Examples

DZ(21,11,32)**2

This function is the square of the z-displacement of Marker 21 with respect to Marker 11 as computed in the coordinate system of Marker 32.

EXP

Definition

The EXP function computes the value $e^{a,}$ where a is any expression.

Format

EXP(a)

Arguments

a Any valid function expression.

Examples

REQUEST/10, F2=EXP(WDTX(21,11,11,31))

In its second column, REQUEST/10 stores the value $e^{WDTX(21,11,11,31)}$

FM

Definition

The FM function returns the magnitude of the **net** translational force acting at marker i due to all applied forces and constraints acting between markers i and j. To calculate the magnitude of the net translational force at marker i due to action-only forces acting at i, set j = 0 or do not specify the j marker.

Format

FM(i [, j])

Arguments

i An integer that specifies the identifier of the marker at which the magnitude of the net force is to be calculated.

j A second integer that specifies a marker identifier. All forces acting between the [i,j] pair are included in the calculation. Set j = 0 or do not specify it if you are interested in action-only forces.

Examples

VARIABLE/1, FUNCTION=FM(23)

A user-defined state variable is specified by this VARI ABLE statement. Its value is the magnitude of the sum of all action-only forces acting at Marker 23.

VARIABLE/1, FUNCTION=FM(23,11)

A user-defined state variable is specified by this VARI ABLE statement. Its value is the magnitude of the sum of all the forces at Marker 23 and acting between Markers 23 and 11.

FORCOS

Definition

The `FORCOS` function evaluates a Fourier Cosine series at a user specified value `x`. x_0 $a_0,a_1,...,a_{30}$ are parameters used to define the constants for the Fourier Cosine series.

Format

FORCOS (x, x_0, w, a_0, a_1,..., a_{30})

Arguments

x A real variable that specifies the independent variable. For example, if the independent variable in the function is time, `x` is the system variable `TIME`.

x_0 A real variable that specifies a shift in the Fourier Cosine series.

w A real variable that specifies the fundamental frequency of the series. MSC.ADAMS/Solver assumes w is in radians per unit of the independent variable unless you use a `D` after the value.

a_0,a_1,...,a_{30} The real variables that define as many as thirty-one coefficients for the Fourier Cosine series.

Extended Definition

The Fourier Cosine series is defined:

$F(x) = a_0 + * T_j (x-x0)$

where the functions T_j are defined as:

T_j (x-x0) = cos{j * w * (x-x0)}

The index j has a range from 1 to n, where n is the number of terms in the series.

Examples

MOTION/1, JOINT=21, TRANSLATION,

, FUNCTION=FORCOS(TIME, 0, 360D, 1, 2, 3, 4)

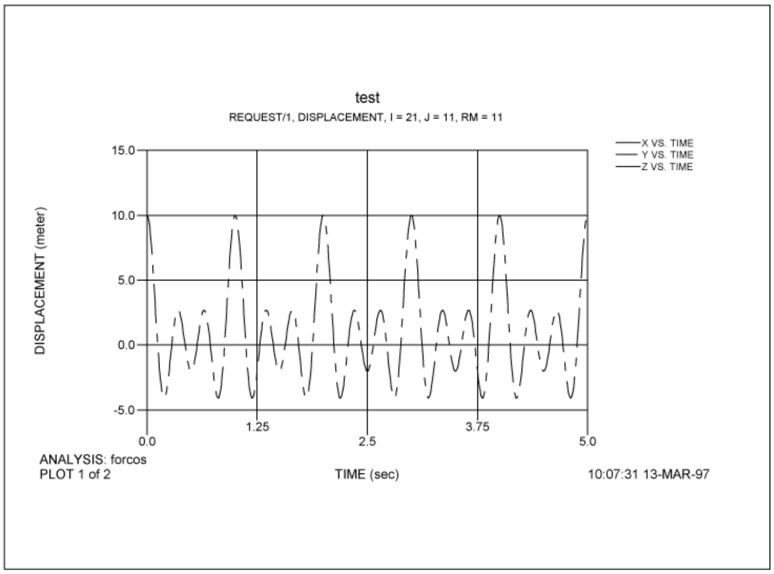
This MOTION statement defines a harmonic motion as a function of time. The motion has no shift, has a fundamental frequency of 1 cycle (360D) per time unit, has a constant value of 1.0. The function defined is:

FORCOS = 1+2*COS(360D*TIME)

+3*COS(2*360D*TIME)

+4*COS(3*360D*TIME)

The figure below shows the curve.



FORSIN

Definition

The `FORSIN` function evaluates a Fourier Sine series at a user specified value `x`. x_0,a_0,a_1,...,a_{30} are parameters used to define the constants for the Fourier Sine series.

Format

FORSIN (x, x_0, w, a_0, a_1,..., a_{30})

Arguments

x A real variable that specifies the independent variable. For example, if the independent variable in the function is time, `x` is the system variable `TIME`.

x_0 A real variable that specifies a shift in the Fourier Sine series.

w A real variable that specifies the fundamental frequency of the series. MSC.ADAMS/Solver assumes w is in radians per unit of the independent variable unless you use a `D` after the value.

a_0 A real variable that defines the constant bias term for the function.

a_1,...,a_{30} The real variables that define as many as thirty-one coefficients for the Fourier Sine series.

Extended Definition

The Fourier Sine series is defined:

$$F(x) = a_0 + * Tj\ (x\text{-}x0)$$

where the functions T_j are defined as:

T_j (x-x0) = sin{j * w * (x-x0)}

The index j has a range from 1 to n, where n is the number of terms in the series

Examples

MOTION/1, JOINT=21, TRANSLATION,

, FUNCTION=FORSIN(TIME,-0.25, PI, 0, 1, 2, 3)

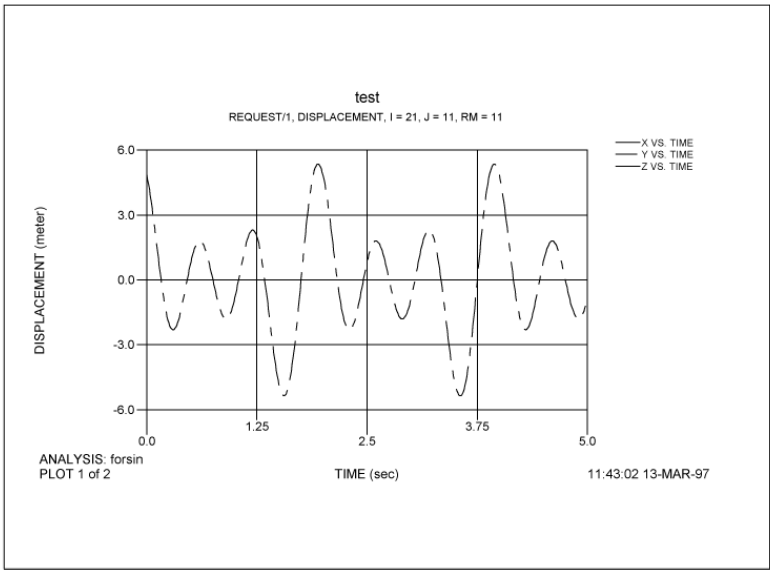
This `MOTION` statement defines a harmonic motion as a function of time. The motion has a -0.25 second shift, has a fundamental frequency of 0.5 cycle (p radians or 180 degrees) per time unit, has no constant value. The function defined is:

FORSIN = 0+SIN(p*(TIME+0.25))

+2*SIN(2p*(TIME+0.25))

+3*SIN(3p*(TIME+0.25))

The figure below shows the curve.



FX

Definition

The FX function returns the x-component of the net translational force acting at marker i, as computed in the coordinate system of marker k. All force elements acting between markers i and j are included in the calculation of the force, unless the force is an action-only type force. To return the x-component of the action-only forces acting at marker i, you should omit specification of markers j and k or specify them to be zero.

Format

FX(i[,j][,k])

Arguments

i An integer that specifies the identifier of the marker at which the magnitude of the net force is to be calculated.

j A second integer that specifies a marker identifier. All forces acting between the [i,j] pair are to be included in the calculation. Set j = 0 or do not specify it if you are interested in action-only forces.

k An integer that specifies the identifier of a marker (coordinate system) in which the x-component of the force is to be returned. k defaults to the global coordinate system when it is not specified.

Examples

FX(23,0,432)

This extracts the x-component of the sum of all the action-only forces acting at Marker 23. The x-axis of Marker 432 specifies the direction along which the net translation force is computed.

FY

Definition

The FY function returns the y-component of the net translational force acting at marker i, as computed in the coordinate system of marker k. All force elements acting between markers i and j are included in the calculation of the force, unless the force is an action-only type force. To return the y-component of the action-only forces acting at marker i, you should omit specification of markers j and k or specify them to be zero.

Format

FY(i[,j][,k])

Arguments

i An integer that specifies the identifier of the marker at which the magnitude of the net force is to be calculated.

j A second integer that specifies a marker identifier. All forces acting between the [i,j] pair are to be included in the calculation. Set j = 0 or do not specify it if you are interested in action-only forces.

k An integer that specifies the identifier of a marker (coordinate system) in which the y-component of the force is to be returned. k defaults to the global coordinate system when it is not specified.

Examples

-0.3*FY(677,866)

This statement calculates a value -0.3 times the y-component of the net force along

the global y-axis, acting at Marker 677. All forces acting between Markers 677 and 866 are included in this calculation.

FZ

Definition

The FZ function returns the z-component of the net translational force acting at marker i, as computed in the coordinate system of marker k. All force elements acting between markers i and j are included in the calculation of the force, unless the force is an action-only type force. To return the z-component of the action-only forces acting at marker i, you should omit specification of markers j and k or specify them to be zero.

Format

FZ(i[,j][,k])

Arguments

i An integer that specifies the identifier of the marker at which the magnitude of the net force is to be calculated.

j A second integer that specifies a marker identifier. All forces acting between the [i,j] pair are to be included in the calculation. Set j = 0 or do not specify it if you are interested in action-only forces.

k An integer that specifies the identifier of a marker (coordinate system) in which the z-component of the force is to be returned. k defaults to the global coordinate system when it is not specified.

Examples

STEP(FZ(34), -1.5, -200, 1.5, 200)

This equation users the value of the force in a step function. The independent variable for the STEP function is FZ(34), the net force acting along the global z-axis at Marker 34. Since no j marker is specified, only action-only forces acting at Marker 34 are included in the calculation of FZ().

IF

Definition

The arithmetic `IF` function allows you to conditionally define a function expression.

Format

`IF` (expression 1: expression 2, expression 3, expression 4)

MSC.ADAMS/Solver evaluates expression 1.

- ❑ If the value of expression 1 is less than zero, the arithmetic `IF` is evaluated using expression 2.
- ❑ If the value of expression 1 is zero, the arithmetic `IF` is evaluated using expression 3.
- ❑ If the value of expression 1 is greater than zero, the arithmetic `IF` is evaluated using expression 4.

Cautions

- ❑ When using an arithmetic `IF` function make sure that the resulting function is continuous. If the function is discontinuous, MSC.ADAMS/Solver may fail to find a solution when it encounters the discontinuity.

Examples

SFORCE/1, I=20, J=31,

, FUNCTION=-6*IF(VR(10,31): 0, 0, VR(10,31)**3)

This function is interpreted as follows:

- ❑ If the radial velocity between Markers 10 and 31 is less than or equal to zero, the value of `SFORCE/1` is zero.
- ❑ If the radial velocity between Markers 10 and 31 is greater than zero, the value of `SFORCE/1` is `-6*VR(10,31)**3`.

IMPACT

Definition

The `IMPACT` function models collisions.

Format

IMPACT (x, $\dot{x}$, x1, k, e, cmax, d)

Arguments

x A real variable that specifies the distance variable you want to use to compute the force. For example, if you want to use the `x` displacement of Marker 0201 with respect to Marker 0301, then `x` is `DX(0201,0301,0301)`.

$\dot{x}$ A real variable that communicates the time derivative of `x` to `IMPACT`. For example, if `x` is `DX(0201,0301)` then $\dot{x}$ is `VX(0201,0301)`.

x1 A positive real variable that specifies the free length of `x`. If `x` is less than `x1` MSC.ADAMS/Solver calculates a positive value for the force. Otherwise, the force value is zero.

k A non-negative real variable that specifies the stiffness of boundary surface interaction.

e A positive real variable that specifies the exponent of the force deformation characteristic. For a stiffening spring characteristic, `e` > 1.0. For a softening spring characteristic, 0 < `e` < 1.0.

cmax A non-negative real variable that specifies the maximum damping coefficient.

d A positive real variable that specifies the boundary penetration at which MSC.ADAMS/Solver applies full damping.

Extended Definition

The `IMPACT` function evaluates a function that turns on when the distance between the `I` and the `J` markers falls below a nominal free length, denoted as $x_{1,}$ (that is, when two parts collide). As long as the distance between the `I` and `J` markers is greater than `x1`, the force is zero. An example of a system you can model with the

IMPACT function is a ball falling towards the ground. Figure 2 shows the free length value x1 at which the IMPACT force turns on.

The force has two components, a spring or stiffness component and a damping or viscous component. The stiffness coefficient, k, is a function of the penetration of the I marker within the free length distance from the J marker. The stiffness component opposes the penetration. The damping component of the force is a function of the speed of penetration. The damping opposes the direction of relative motion. To prevent a discontinuity in the damping force at contact, the damping coefficient is, by definition, a cubic step function of the penetration. Thus, at zero penetration, the damping coefficient is always zero. The damping coefficient achieves a maximum, cmax, at a user-defined penetration, d.

Examples

IMPACT(DZ(11,21,21), , VZ(11,21,21,21),1.0, 100, 1.5, 25, 0.1)

This statement defines an impact when a ball penetrates another object such as a table. The function is a single-component force at Marker 11 and along the z-axis of Marker 21. DZ(11,21,21) defines the instantaneous displacement of Marker 11 with respect to Marker 21 along the z-axis of Marker 21. VZ(11,21,21,21) defines the instantaneous velocity. The free length is 1, i.e., the radius of the ball is 1 unit. The stiffness is 100, and the exponent of deformation is 1.5, the maximum damping coefficient is 25. The penetration at which MSC.ADAMS/Solver applies full damping is 0.1.

Note: IMPACT is automatically available in COSMOSMotion using the IMPACT force entity.

LOG

Definition

The LOG function returns the natural logarithm an expression a. If e^x=a then LOG(a)= x. The LOG function is defined only for positive values of a (i.e., a > 0). It is undefined for all other values.

Format

LOG(a)

Arguments

a Any valid function expression.

Examples

```
DIFF/1, IC=3.6
, FUNCTION=LOG(1+VM(21,31,31))
```

This DIFF statement defines a user specified differential equation. The initial condition for the associated state is 3.6 and its derivative is the natural logarithm of the expression 1+VM(21,31,31).

LOG10

Definition

The `LOG10` function returns log to base 10 of an expression `a`. If 10x=`a`, then `LOG10(a)=` x. The `LOG10` function is defined only for positive values of a (i.e., `a` > 0). It is undefined for all other values.

Format

LOG10(a)

Arguments

a Any valid function expression.

Examples

DIFF/1, IC=3.6

, FUNCTION=LOG10(1+VM(21,31,31))

This `DIFF` statement defines a user specified differential equation. The initial condition for the associated state is 3.6 and its derivative is the natural logarithm of the expression `1+VM(21,31,31)`.

MAX

Definition

The MAX function returns the maximum of two expressions a1 and a2.

MAX(a1,a2) = a1 if a1 a2

MAX(a1,a2) = a2 if a2 > a1

Format

MAX(a1,a2)

Arguments

a1 Any valid function expression.

a2 Any valid function expression.

Cautions

- ❑ The MAX function is generally discontinuous. Use this function expression with care when you are specifying force or motion input.

Examples

SFORCE/1, ROTATIONAL, ACTIONONLY, I=21, J=11

, FUNCTION=MAX(0,-3*(AZ(21,11)-25D))

This SFORCE statement defines an action-only torque that acts at Marker 21 and acts along the z-axis of Marker 11. The torque is defined to always be non-negative through the use of the MAX function. Thus, the torque can only act to increase the AZ rotation of Marker 21 with respect to Marker 11.

MIN

Definition

The `MIN` function returns the minimum of two expressions `a1` and `a2`.

`MIN(a1,a2)` = `a1` if `a1 a2`

`MIN(a1,a2)` = `a2` if `a2` < `a1`

Format

MIN(a1,a2)

Arguments

a1 Any valid function expression.

a2 Any valid function expression.

Cautions

- ❑ The `MIN` function is generally discontinuous. Use this function expression with care when you are specifying force or motion input.

Examples

SFORCE/1, ROTATIONAL, ACTIONONLY, I=21, J=11

, FUNCTION=MIN(0,-3*(AZ(21,11)-25D))

This `SFORCE` statement defines an action-only torque that acts at Marker 21 and acts along the z-axis of Marker 11. The torque is defined to always be negative through the use of the `MIN` function. Thus, the torque can only act to decrease the `AZ` rotation of Marker 21 with respect to Marker 11.

MOD

Definition

The `MOD` function returns the remainder when `a1` is divided by `a2.`

`MOD(a1,a2)` = `a1` - `int(a1/a2)` * `a2`

Format

MOD(a1,a2)

Arguments

a1 Any valid function expression.

a2 Any valid function expression.

Cautions

- ❑ The `MOD` function is generally discontinuous. Use this function expression with care when you are specifying force or motion input.

Examples

MOD(45,16)=13

MOTION

Definition

The MOTION function returns component comp of a force due to MOTION/*id* as calculated in the coordinate system of marker rm. If jflag is set to zero, the value returned is the force/torque acting on the I marker of the MOTION. If jflag is set to one, the value returned is that acting on the J marker. rm may be specified as zero if the results are desired in the global coordinate system.

Format

MOTION (id, jflag, comp, rm)

Arguments

id An integer specifying the identification number of the motion.

jflag An integer flag specifying the motion connectivity marker at which the forces and torques are computed.
0 = forces and moments at the I marker
1 = forces and moments at the J marker

comp An integer value that specifies the component of the motion force to be returned.
1 = Magnitude of the force
2 = x component of the force
3 = y component of the force
4 = z component of the force
5 = Magnitude of the torque
6 = x component of the torque
7 = y component of the torque
8 = z component of the torque

rm The coordinate system in which the results are expressed. To return the results in the global coordinate system, set rm = 0.

Examples

WZ(21,11)*MOTION(1,0,8,11)

This statement calculates the instantaneous power of MOTION/1 about the z-axis during a simulation (Power = force * velocity).

PHI

Definition

The PHI function calculates the third angle (in radians) of a body-2 [313] Euler rotation sequence between markers i and j. Marker j defaults to the global coordinate system if it is not specified.

Format

PHI(i[,j])

Arguments

i	The marker whose rotations are being sought.
j	The marker with respect to which the rotations are being measured.

Examples

PHI(222,434)

This function returns the phi angle between Markers 222 and 434 in its second column.

PI

Definition

PI returns the ratio of the circumference of a circle to its diameter (3.14159...).

Format

PI

PITCH

Definition

Calculates the second angle of a Body-3 [3 -2 1] yaw-pitch-roll rotation sequence between marker `i` and `j`. Marker `j` defaults to the global coordinate system if it is not specified. Note that pitch is an Euler angle.

Format

PITCH(i[,j])

Arguments

i	The marker whose rotations are being sought.
j	The marker with respect to which the rotations are being measured.

Examples

PITCH(143)

This function returns the pitch angle (in radians) of Marker 143 relative to the global coordinate system.

POLY

Definition

The `POLY` function evaluates a standard polynomial at a user-specified value `x`.

Format

POLY (x, x_0, a_0, a_1,..., a_{30})

Arguments

x A real variable that specifies the independent variable. For example, if the independent variable in the function is time, `x` is the system variable `TIME`.

x_0 A real variable that specifies a shift in the polynomial.

a_0,a_1,...,a_{30} The real variables that define as many as thirty-one coefficients for the polynomial series.

Extended Definition

The `POLY` function evaluates a standard polynomial at a user-specified value `x`. x_0,a_0,a_1,...,a_{30} are parameters used to define the constants for the polynomial. The standard polynomial is defined as:

$$P(x) = a_j (x-x0)^j$$

$$= a_0 + a_1 * (x-x0) + a_2 * (x-x0)^2 + ... + a_n * (x-x0)^n$$

The index j has a range from zero to n, where n is the number of terms in the series.

Examples

MOTION/1, JOINT=21, TRANSLATION

, FUNCTION=POLY(TIME, 0, 0, 0, 1)

This `MOTION` statement uses a quadratic polynomial function with respect to the system variable `TIME` to define a motion. That expanded function is:

`Poly` = time2

MOTION/1, JOINT=21, TRANSLATION

, FUNCTION = IF(TIME-5: 0, 0, POLY(TIME, 5, 0, 10))

This MOTION statement uses an arithmetic IF in its function expression to switch between functions.

The expanded function is:

if (time - 5) then Poly=0

else Poly=10*time-5

SFORCE/3, I=10, J=20, TRANSLATION

, FUNCTION=-POLY(DM(10, 20), 10, 0, 25, 0, 0.75)

This SFORCE statement defines a force with a nonlinear force deformation characteristic. This relationship is:

Poly=-25*[DM(10,20)-10]+ 0.75*[DM(10,20)-10]3,

where, DM(10,20) represents the magnitude of the displacement of Marker 10 with respect to Marker 20. The free length of the spring is 10 units.

PSI

Definition

The `PSI` function expression calculates the first angle (in radians) of a body-2 [313] Euler rotation sequence between markers `i` and `j`. Marker `j` defaults to the global coordinate system if it is not specified.

Format

PSI(i[,j])

Arguments

i The marker whose rotations are being sought.

j The marker with respect to which the rotations are being measured.

Examples

PSI(222,434)

This function returns the psi angle between Markers 222 and 434 in its second column.

ROLL

Definition

The ROLL function calculates the third angle of a Body-3 [3 -2 1] yaw-pitch-roll rotation sequence between markers i and j. Marker j defaults to the global coordinate system if it is not specified. Note that roll is an Euler angle.

Format

ROLL(i[,j])

Arguments

i	The marker whose rotations are being sought.
j	The marker with respect to which the rotations are being measured.

Examples

ROLL(143)

This function returns the roll angle (in radians) of Marker 143 relative to the global coordinate system.

RTOD

Definition

RTOD returns the radians to degrees conversion factor.

Format

RTOD

Examples

REQUEST/1, F2=ROLL(23,14)*RTOD

This REQUEST statement outputs the roll angle between Markers 23 and 14 in degrees.

SHF

Definition

The SHF function evaluates a simple harmonic function. The following equation defines SHF.

SHF = a*sin(ω*(x-x_0)-phi)+b

Format

SHF (x, x_0, a, w, phi, b)

Arguments

x - The independent variable in the function. It may be any value of interest to you so that you can compute the value using the function expression. For example, if the independent variable in the function is time, x is the system variable TIME.

x_0 The offset in the independent variable x.

a The amplitude of the harmonic function.

w The frequency of the harmonic function. MSC.ADAMS/Solver assumes w is in radians per unit of the independent variable. Conversion to degrees per unit of the independent is achieved by appending a D after the number specifying w.

phi A phase shift in the harmonic function. MSC.ADAMS/Solver assumes phi is in radians unless you use a D after the value.

b The average value of displacement of the harmonic function.

Examples

SHF(TIME, 25D, PI, 360D, 0, 5)

This statement uses SHF to define the harmonic function:

SHF = 5+PI*sin(360D*(time-25D))

The motion has a shift of 25 degrees, has an amplitude of PI, has a frequency of 1 cycle (360D) per time unit, has zero phase shift, and has an average value of displacement of 5 units.

SIGN

Definition

The SIGN function transfers the sign of a2 to the magnitude of a1.

SIGN(a1,a2) = ABS(a1) if a2 0

SIGN(a1,a2) = -ABS(a1) if a2 < 0

Format

SIGN (a1,a2)

Arguments

a1 Any valid function expression.

a2 Any valid function expression.

Cautions

- ❑ The SIGN function is discontinuous. Use this function with care to avoid creating expressions that are discontinuous.

Examples

VARIABLE/23, FUNCTION=10*SIGN(DM(24,33),VZ(24,33))

When VZ(24,33) ^{3} 0, the value of VARIABLE/23 is -10*DM(24,33). When VZ(24,33) < 0, the value of VARIABLE/23 is 10*DM(24,33).

SIN

Definition

The SIN function returns the sine of an expression a.

Format

SIN(a)

Arguments

a Any valid function expression.

Examples

SFORCE/1, TRANSLATION, I=23, J=34

, FUNCTION=SIN(10*TIME)

A translational force is defined to be acting along the line joining Markers 23 and 34. The force at Marker 23 is defined to be SIN(10*TIME).

SINH

Definition

The SINH function return the hyperbolic sine of an expression a.

SINH(a) = $(e^{a}-e^{-a})$ / 2.0

Format

SINH(a)

Arguments

a Any valid function expression.

Examples

VARIABLE/1, FUNCTION=SINH(DX(21,32))

The value of VARIABLE/1 is the hyperbolic sine of the x-component of the displacement between Markers 21 and 32, as calculated in the global coordinate system.

SQRT

Definition

The `SQRT` function returns the square root of an expression `a`. The square root function is defined only for non-negative values of the argument `a`.

Format

SQRT(a)

Arguments

a	Any valid function expression.

STEP

Definition

The STEP function approximates the Heaviside step function with a cubic polynomial. It has continuous first derivatives. Its second derivatives are discontinuous at $x=x_0$ and $x=x_1$.

Format

STEP (x, x_0, h_0, x_1, h_1)

Arguments

x The independent variable.

$\mathbf{x}_0$ A real variable that specifies the x value at which the STEP function begins.

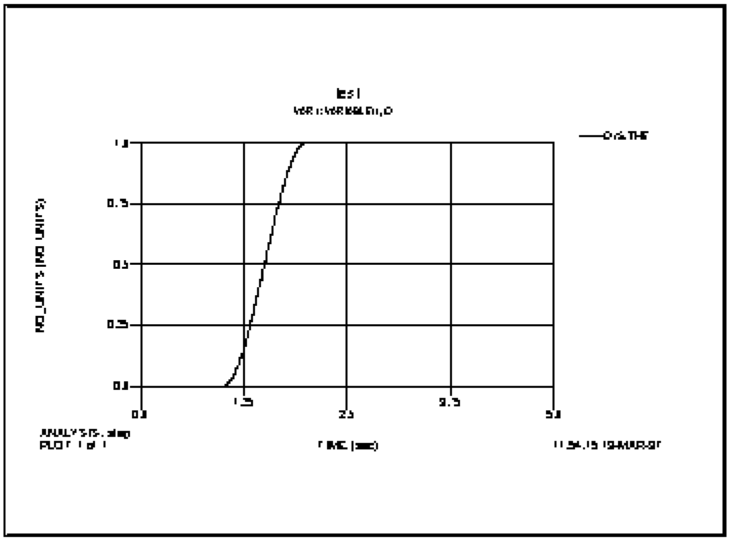
$\mathbf{x}_1$ A real variable that specifies the x value at which the STEP function ends.

$\mathbf{h}_0$ The initial value of the step.

$\mathbf{h}_1$ The final value of the step.

Extended Definition

The STEP function approximates the Heaviside step function with a cubic polynomial. The figure below illustrates the STEP function.



Examples

MOTION/1, JOINT=21, TRANSLATION

, FUNCTION=STEP(TIME, 1, 0, 2, 1)

This `MOTION` statement defines a smooth step function from time 1 to time 2 with a displacement from 0 to 1.

STEP5

Definition

The STEP5 function provides approximation to the Heaviside step function with a quintic polynomial. It has continuous first and second derivatives. Its third derivative is discontinuous at $x=x_0$ and $x=x_1$.

Format

STEP5 (x, x_0, h_0, x_1, h_1)

Arguments

x The independent variable.

x_0 A real variable that specifies the x value at which the STEP5 function begins.

x_1 A real variable that specifies the x value at which the STEP5 function ends.

h_0 The initial value of the step.

h_1 The final value of the step.

Extended Definition

The equation defining the STEP5 function is:

$a = h_1 - h_0$

$D = (x - x_0)/(x_1 - x_0)$

$$\mathrm{STEP5} = \begin{Bmatrix} h_0 & : & x \le x_0 \\ h_0 + a\Delta^3 * [10 - 15\Delta + 6\Delta^2] & : & x_0 < x < x_1 \\ h_1 & : & x \ge x_1 \end{Bmatrix}$$

Examples

STEP5(TIME, 1, 0, 2, 1)

This statement defines a smooth step function from time 1 to time 2 with a displacement from 0 to 1.

TAN

Definition

The TAN function returns the tangent of an expression a.

Format

TAN(a)

Arguments

a - Any valid function expression.

TANH

Definition

The TANH function returns the hyperbolic tangent of an expression a.

TANH(a)= $(e^{a} - e^{-a}) / (e^{a} + e^{-a})$

Format

TANH(a)

Arguments

a - Any valid function expression.

Examples

MOTION/1, JOINT=21, TRANSLATION

, FUNCTION=0.5(1+TANH(5*(TIME-1.5)))

This MOTION statement defines a smooth step function that transitions from a value of 0 to 1. The value of the step function at time=1.5 is 0.5.

THETA

Definition

The THETA function calculates the second angle (in radians) of a body-2 [313] Euler rotation sequence between markers i and j. Marker j defaults to the global coordinate system if it is not specified.

Format

THETA(i[,j])

Arguments

i	The marker whose rotations are being sought.
j	The marker with respect to which the rotations are being measured.

Examples

THETA(222,434)

This function returns the psi angle between Markers 222 and 434.

TIME

Definition

The TIME variable returns the current simulation time.

Format

TIME

Examples

SFORCE/1, TRANSLATION, I=21, J=32

, FUNCTION=10*SIN(2*PI*TIME)

This SFORCE statement defines a translational force acting between the origins of Marker 21 and Marker 11. The value of the force is 10*SIN*(2*PI*TIME).

TM

Definition

The TM function returns the magnitude of the **net** torque acting at marker i due to all applied torques and constraints acting between markers i and j. To calculate the magnitude of the net torque at marker i due to action-only torque acting at i, set j = 0.

Format

TM(i[,j])

Arguments

i An integer that specifies the identifier of the marker at which the magnitude of the net torque is to be calculated.

j A second integer that specifies a marker identifier. All torques acting between the [i,j] pair are to be included in the calculation. Set j = 0 or do not specify it if you are interested in action-only torques.

Examples

```
JOINT/1, REVOLUTE, I=32, J=44
MOTION/1, ROTATION, JOINT=1, FUNCTION=10*TIME
VARIABLE/31, FUNCTION=TM(32,44)
```

In this example a revolute joint and motion (through the joint) act between Marker 32 and 44. A new algebraically determined state variable VARIABLE/31 is defined to have the magnitude of the net torque between these two markers as its value.

TX

Definition

The TX function returns the x-component of the net torque acting at marker i, as computed in the coordinate system of marker k. All force elements acting between markers i and j are included in the calculation of the torque, unless the force element is an action-only type force. To return the x-component of the action-only torques acting at marker i, you should omit specification of marker j or specify it as zero.

Format

TX(i[,j][,k])

Arguments

i An integer that specifies the identifier of the marker at which the magnitude of the net torque is to be calculated.

j A second integer that specifies a marker identifier. All torques acting between the [i,j] pair are to be included in the calculation. Set j = 0 or do not specify it if you are interested in action-only torques.

k An integer that specifies the identifier of a marker (coordinate system) in which the x-component of the torque is to be returned.

Examples

```
JOINT/1, REVOLUTE, I=32, J=44
MOTION/1, ROTATION, JOINT=1, FUNCTION=10*TIME
VARIABLE/31, FUNCTION=TX(32,44)
```

In this example a revolute joint and motion (through the joint) act between Marker 32 and 44. A new algebraically determined state variable VARIABLE/31 is defined to have the magnitude of the net torque between these two markers as its value.

TY

Definition

The TY function returns the y-component of the net torque acting at marker i as computed in the coordinate system of marker k. All force elements acting between markers i and j are included in the calculation of the torque, unless the force element is an action-only type force. To return the y-component of the action-only torques acting at marker i, you should omit specification of marker j or specify it as zero.

Format

TY(i[,j][,k])

Arguments

i An integer that specifies the identifier of the marker at which the magnitude of the net torque is to be calculated.

j A second integer that specifies a marker identifier. All torques acting between the [i,j] pair are to be included in the calculation. Set j = 0 or do not specify it if you are interested in action-only torques.

k An integer that specifies the identifier of a marker (coordinate system) in which the y-component of the torque is to be returned.

Examples

```
JOINT/1, REVOLUTE, I=32, J=44
MOTION/1, ROTATION, JOINT=1, FUNCTION=10*TIME
VARIABLE/31, FUNCTION=TY(32,44)
```

In this example a revolute joint and motion (through the joint) act between Marker 32 and 44. A new algebraically determined state variable VARIABLE/31 is defined to have the magnitude of the net torque between these two markers as its value.

TZ

Definition

The TZ function returns the z-component of the net torque acting at marker i as computed in the coordinate system of marker k. All force elements acting between markers i and j are included in the calculation of the torque, unless the force element is an action-only type force. To return the z-component of the action-only torques acting at marker i, you should omit specification of marker j or specify it as zero.

Format

TZ(i[,j][,k])

Arguments

i An integer that specifies the identifier of the marker at which the magnitude of the net torque is to be calculated.

j A second integer that specifies a marker identifier. All torques acting between the [i,j] pair are to be included in the calculation. Set j = 0 or do not specify it if you are interested in action-only torques.

k An integer that specifies the identifier of a marker (coordinate system) in which the z-component of the torque is to be returned.

Examples

```
JOINT/1, REVOLUTE, I=32, J=44
MOTION/1, ROTATION, JOINT=1, FUNCTION=10*TIME
VARIABLE/31, FUNCTION=TZ(32,44)
```

In this example a revolute joint and motion (through the joint) act between Marker 32 and 44. A new algebraically determined state variable VARI ABLE/ 31 is defined to have the magnitude of the net torque between these two markers as its value.

VM

Definition

The VM function calculates the magnitude of the first time derivative of the displacement vector of marker i with respect to marker j. The vector time derivative is taken in the reference frame of marker l. Markers i and j default to the global coordinate system if they are not specified.

Format

VM(i[,j][,l])

Arguments

i The marker whose velocity is being measured. Set i= 0 if you want it to default to the global coordinate system.

j The marker with respect to which the displacement is being measured. Set j = 0 if you want j to default to the global coordinate system while still specifying l.

l The reference frame in which the time derivative of the displacement vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

VM(23,0,32)

This function returns the magnitude of the velocity of the origin of Marker 23 with respect to ground. The time derivative for the velocity computation is taken in the reference frame of Marker 32.

VM(21,32,43)

This function returns the magnitude of the velocity vector between Markers 21 and 32, as seen by an observer at Marker 43.

VR

Definition

The `VR` function calculates the radial (relative) velocity of marker `i` with respect to marker `j`. The vector time derivative is taken in the reference frame of marker `l`. Markers `j` and `l` default to the global coordinate system if they is not specified.

Format

VR(i[,j][,l])

Arguments

i The marker whose velocity is being measured.

j The marker with respect to which the displacement is being measured. Set `j` = 0 if you want `j` to default to the global coordinate system while still specifying `l`.

l The reference frame in which the second time derivative of the displacement vector is taken.

Examples

-10*VR(467,764)

This function implements a translational damper with a damping constant of 10.

VX

Definition

The VX function returns the x-component of the difference between the velocity vector of marker i and the velocity vector of marker j as computed in the coordinate system of marker k. All vector time derivatives are taken in the reference frame of marker l. Marker j defaults to the global coordinate system if it is not specified. Similarly, markers i and j default to the global coordinate system if they are not specified.

Format

VX(i[,j][,k][,l])

Arguments

i The marker whose velocity is being measured.

j The marker with respect to which the displacement is being measured. Set j = 0 if you want j to default to the global coordinate system while still specifying l.

k The marker in whose coordinate system the velocity vector is being expressed. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

l The reference frame in which the first time derivative of the displacement vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

-20*VX(236,168,168,168)

This function defines a damper acting between Markers 236 and 168. The damping force components are proportional to the components of the velocity between Markers 236 and 168 as seen and measured by an observer at Marker 168.

VY

Definition

The VY function returns the y-component of the difference between the velocity vector of marker i and the velocity vector of marker j as computed in the coordinate system of marker k. All vector time derivatives are taken in the reference frame of marker l. Marker j defaults to the global coordinate system if it is not specified. Similarly, markers i and j default to the global coordinate system if they are not specified.

Format

VY(i[,j][,k][,l])

Arguments

i The marker whose velocity is being measured.

j The marker with respect to which the displacement is being measured. Set j = 0 if you want j to default to global coordinate system while still specifying l.

k The marker in whose coordinate system the velocity vector is being expressed. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

l The reference frame in which the first time derivative of the displacement vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

-15*VY(236,168,168,168)

Definition

The VZ function returns the z-component of the difference between the velocity vector of marker i and the velocity vector of marker j as computed in the coordinate system of marker k. All vector time derivatives are taken in the reference frame of marker l. Marker j defaults to the global coordinate system if it is not specified. Similarly, markers i and j default to the global coordinate system if they are not specified.

Format

VZ(i[,j][,k][,l])

Arguments

i The marker whose velocity is being measured.

j The marker with respect to which the displacement is being measured. Set j = 0 if you want j to default to global coordinate system while still specifying l.

k The marker in whose coordinate system the velocity vector is being expressed. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

l The reference frame in which the first time derivative of the displacement vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

-20*VZ(236,168,168,168)

WDTM

Definition

The WDTM function returns the magnitude of the difference between the angular acceleration vector of marker i in the reference frame of marker l and the angular acceleration of marker j in the reference frame of marker l.

Format

WDTM(i[,j][,l])

Arguments

i The marker whose acceleration is being measured.

j The marker with respect to which the acceleration is being measured. Set j = 0, while still specifying l, if you want j default to the global coordinate system.

l The reference frame in which the first time derivative of the angular velocity vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

WDTM(1236,2169)

This function obtains the angular acceleration magnitude of Marker 1236 with respect to Marker 2169, as seen in the global coordinate system and measured in the ground reference frame.

WDTX

Definition

The WDTX function returns the x-component of the difference between the angular acceleration vector of marker i in the reference frame of marker l and the angular acceleration vector of marker j in the reference frame of marker l, as computed in the coordinate system of marker k. Marker j defaults to global coordinate system if it is not specified. Similarly, marker k and l default to global coordinate system if they are not specified.

Format

WDTX(i[,j][,k][,l])

Arguments

i The marker whose acceleration is being measured.

j The marker with respect to which the acceleration is being measured. Set j = 0, while still specifying l, if you want j to default to the global coordinate system.

k The marker in whose coordinate system the acceleration vector is being expressed. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

l The reference frame in which the first time derivative of the angular acceleration vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

F2=WDTX(1236,2169,2169,2169)

This function obtains the x-component of angular acceleration on Marker 1236 with respect to Marker 2169, as seen in the global coordinate system of Marker 2169 and measured in the reference frame containing Marker 2169.

WDTY

Definition

The WDTY function returns the y-component of the difference between the angular acceleration vector of marker i in the reference frame of marker l and the angular acceleration vector of marker j in the reference frame of marker l, as computed in the coordinate system of marker k. Marker j defaults to the global coordinate system if it is not specified. Similarly, marker k and l default to the global coordinate system if they are not specified.

Format

WDTY(i[,j][,k][,l])

Arguments

i The marker whose acceleration is being measured.

j The marker with respect to which the acceleration is being measured. Set j = 0, while still specifying l, if you want j to default to the global coordinate system.

k The marker in whose coordinate system the acceleration vector is being expressed. Set k = 0 if you want the results to be calculated along the y-axis of the global coordinate system.

l The reference frame in which the first time derivative of the angular acceleration vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

WDTY(1236,2169,2169,2169)

This function obtains the y-component of angular acceleration on Marker 1236 with respect to Marker 2169, as seen in the global coordinate system of Marker 2169 and measured in the reference frame containing Marker 2169.

WDTZ

Definition

The WDTZ function returns the z-component of the difference between the angular acceleration vector of marker i in the reference frame of marker l and the angular acceleration vector of marker j in the reference frame of marker l, as computed in the coordinate system of marker k. Marker j defaults to the global coordinate system if it is not specified. Similarly, marker k and l default to the global coordinate if they are not specified.

Format

WDTZ(i[,j][,k][,l])

Arguments

i The marker whose acceleration is being measured.

j The marker with respect to which the acceleration is being measured. Set j = 0, while still specifying l, if you want j to default to the global coordinate system.

k The marker in whose coordinate system the acceleration vector is being expressed. Set k = 0 if you want the results to be calculated along the z-axis of the global coordinate system.

l The reference frame in which the first time derivative of the angular acceleration vector is taken. Set l = 0 if you want the time derivatives to be taken in the ground reference frame.

Examples

WDTZ(1236,2169,2169,2169)

This function obtains the z-component of angular acceleration on Marker 1236 with respect to Marker 2169, as seen in the global coordinate system of Marker 2169 and measured in the reference frame containing Marker 2169.

WM

Definition

The `WM` function returns the magnitude of the angular velocity vector of marker `i` with respect to marker `j`. Marker `j` defaults to the global coordinate system if it is not specified.

Format

WM(i[,j])

Arguments

i The marker whose velocity is being measured.

j The marker with respect to which the displacement is being measured. Set `j` = 0 if you want `j` to default to the global coordinate system while still specifying `l`.

Examples

WM(1236,2169)

This function returns the magnitude of the angular velocity vector of Marker 1236 and Marker 2169.

WX

Definition

The WX function returns the x-component of the difference between the angular velocity vector of marker i in ground and the angular velocity vector of marker j in ground, and expressed in the coordinate system of marker k. Marker j defaults to the global coordinate system if it is not specified. Similarly, marker k defaults to the global coordinate system if it is not specified.

Format

WX(i[,j][,k])

Arguments

i The marker whose velocity is being measured.

j The marker with respect to which the displacement is being measured. Set j = 0 if you want j to default to the global coordinate system while still specifying l.

k The marker in whose coordinate system the velocity vector is being expressed. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

Examples

WX(1236,2169,2169)

This function returns the x-component of the angular velocity Markers 1236 and 2169 as measured in the coordinate system of Marker 2169.

WY

Definition

The WY function returns the y-component of the difference between the angular velocity vector of marker i in ground and the angular velocity vector of marker j in ground, and expressed in the coordinate system of marker k. Marker j defaults to the global coordinate system if it is not specified. Similarly, marker k defaults to the global coordinate system if it is not specified.

Format

WY(i[,j][,k])

Arguments

i The marker whose velocity is being measured.

j The marker with respect to which the displacement is being measured. Set j = 0 if you want j to default to the global coordinate system while still specifying l.

k The marker in whose coordinate system the velocity vector is being expressed. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

Examples

WY(1236,2169,2169)

This function returns the y-component of the angular velocity Markers 1236 and 2169 as measured in the coordinate system of Marker 2169.

WZ

Definition

The WZ function returns the z-component of the difference between the angular velocity vector of marker i in ground and the angular velocity vector of marker j in ground, and expressed in the coordinate system of marker k. Marker j defaults to the global coordinate system if it is not specified. Similarly, marker k defaults to the global coordinate system if it is not specified.

Format

WZ(i[,j][,k])

Arguments

i The marker whose velocity is being measured.

j The marker with respect to which the displacement is being measured. Set j = 0 if you want j to default to the global coordinate system while still specifying l.

k The marker in whose coordinate system the velocity vector is being expressed. Set k = 0 if you want the results to be calculated along the x-axis of the global coordinate system.

Examples

WZ(1236,2169,2169)

This function returns the z-component of the angular velocity Markers 1236 and 2169 as measured in the coordinate system of Marker 2169.

YAW

Definition

The `YAW` function calculates the first angle of a Body-3 [3 -2 1] yaw-pitch-roll rotation sequence between markers `i` and `j`. Marker `j` defaults to the global coordinate system if it is not specified. Note that yaw is an Euler angle.

Format

YAW(i[,j])

Arguments

i The marker whose rotations are being sought.

j The marker with respect to which the rotations are being measured.

Examples

YAW(21,11)

Th function returns the yaw angles between Markers 21 and 11.

Index

3

3D Contact 59
3D Contacts
- Creating 70

A

ABS function 196
Absolute value function expression 196
Acceleration Vectors 153
ACOS function 201
Action/Reaction Force
- Components 91, 93
- Creating 90, 92
- Deleting 109
- Editing 109
- Impact 104
- Magnitude 92, 94
- Points 91, 93, 106
- Properties 92, 94

Action/Reaction Torque
- Deleting 109
- Editing 109

Activating the Browser 188
ADAMS
- Exporting To 185
- Functions 187, 189
- Markers 191

ADAMS Functions
- ACCM 197
- ACCX 198
- ACCY 199
- ACCZ 200
- AX 207
- AY 208
- AZ 209
- BISTOP 210
- DM 217
- DX 219
- DY 220
- DZ 221
- FM 223
- FX 228
- FY 229
- FZ 230
- IMPACT 232
- MOTION 239
- PHI 240
- PITCH 242
- PSI 245
- ROLL 246
- THETA 258
- TM 260
- TX 261
- TY 262
- TZ 263
- VM 264
- VR 265
- VX 266
- VY 267
- VZ 268
- WDTM 269
- WDTX 270
- WDTY 271
- WDTZ 272
- WM 273
- WX 274
- WY 275
- WZ 276
- YAW 277

Angular Displacements 150
Animation 131
- End Step 134
- Exporting to AVI 135
- Exporting to VRML 137
- Fast Increment 133
- Frame and step selection 182
- Mode 132
- Settings 133
- Start Step 134
- Step Increment 134

Animation controls 131
Applied Force
- Application Point 89
- Component 89
- Creating 88
- Deleting 109
- Direction 88, 90

Editing 109
Properties 90
Template Editor 90, 92, 94
Applied Torque
Deleting 109
Editing 109
Arc tangent, calculating 205, 206
ASIN function 204
ATAN function... 205
ATAN2 function... 206
Automatically creating joints from mates......... 14
AVI File 135

B

Bearing surfaces for FEA load distribution 145
Browser
Activating 188

C

Cam Constraints 56
Curve/Curve... 58, 59
Creating.. 64
Curves 65
Properties 66
Tips 73
Point/Curve 57
Creating.. 62
Point 62
Properties 63
Tips 73
Chebyshev polynomial, evaluating................. 212
Circle:calculating circumference.................... 241
Circumference, calculating............................. 241
Contacts
2D 58
Creating.. 61
3D 59
Contact Containers 59
Creating.. 70
Facet Tolerance 127
Facetted Geometry 127
Precise Geometry 127
Converting:degrees to radians 218
Converting:radians to degrees 247
COS function.. 214
COSH function 215
Cosine, calculating................................. 201, 214
COSMOS/Motion
benefits of... 2
3
CosmosWorks
Bearing Surfaces 145
Exporting To 145
Couplers... 74, 76
Coupling joint motion................................ 74, 76
Creating a VRML Animation File 137
Creating an AVI Movie 135
Creating XY Plots... 166
Cubic polynomial, using with function........... 253
Curve/Curve Constraint.............................. 58, 59
Creating.. 64
Curves 65
Properties 66
Tips 73

D

Dampers.. 96
Components 99
Creating.. 99
Deleting.. 109
Editing.. 109
Points 99
Data Points Function... 87
Degree conversion factor................................ 218
Degrees of freedom... 7
Deleting
XY Plots in Result Viewer......................... 169
Deleting Elements... 109
DOF Counter 120
DTOR function 218

E

e value, calculating 222
Editing Elements... 109
Excel 97.. 139
Exporting Results 154
Exporting Results to a Text File 142
Exporting Results to Excel 97......................... 139
Exporting to ADAMS..................................... 185
149
Expressions
calculating maximum of two...................... 236

calculating minimum of two.......................237
calculating remainder 238

F

Feature Tree 4
Force
Applied Force...83
Applied Torque 83
Deleting...109
Editing...109
Gravity 83, 108
Magnitude 83
Force Magnitude
Template Editor..83
Fourier cosine series, evaluating.....................224
Fourier sine series, evaluating.........................226
Friction
Cylindrical Joint 30
Enabling Effects 28
Joint Results 35
Joints 28
Planar Joint...34
Revolute Joint...29
Spherical Joint 31
Translational Joint 32
Universal Joint 33
Functions used in Expressions 187, 189

G

Gravity 108

H

Harmonic Function 85
Heaviside step function, calculating 253, 255
Hyperbolic cosine, calculating........................215
Hyperbolic sine, calculating............................251
Hyperbolic tangent, calculating 257

I

Impact Forces..104
3
Interference Detection....................................143

J

Joint Couplers 74, 76
Joint Primitives 36
Components...53
Inline 37
Inplane 38
Orientation...39
Parallel Axes..40
Pependicular 41
Properties...54
Joints
Automatically create from mates 14
Ball 22
Components...53
Cylindrical 8, 21
Cylindrical, Friction 30
Definition 17
Degrees of freedom 18
Deleting 109
Editing 109
Fixed..27
Hinge 7, 19
Planar...26
Planar, Friction 34
Properties...54
Results, Friction..35
Revolute, Friction 29
Screw 24
Slider 20
Spherical, Friction 31
Translational, Friction 32
Universal 23
Universal, Friction..33

L

Linear Displacements.....................................149
Log to base 10, calculating.............................235
Logarithm, calculating natural 234

M

Magnitude, calculating nearest integer....202, 203
Markers 191
Materials 111
Adding 112
Editing 116

Menu 3
Model Builder
  Command....... 171
  Gravity Page 173
  Ground Parts Menu....... 182
  Joints Menu....... 178
  Joints Page 178
  Manually Adding Joints....... 179
  Motion Page....... 180
  Parts Page 174
  Rigidly Attaching Component 175
  Simulation Page....... 181
  Springs Menu....... 179
  Springs Page 181
  Units Page....... 172
Modeling Procedure 11, 12
Modling power transmission elements 74, 76
Motion Drivers 42
  Defining....... 55
  Deleting 109
  Editing 109
  Function Expression 55
  Other Functions 49
  Template Editor 43
  Template Editor 44
  Template Editor 55
  Type....... 43
Motion Parts
  Creating 14
  Drag and Drop within the Browser 14
  Rigidly Attaching Parts....... 16

O

Output Steps 123

P

Playback
  Mode....... 132
Point/Curve Constraint....... 57
  Creating 62
  Point....... 62
  Properties....... 63
  Tips....... 73
Polynomial, evaluating standard....... 243
Positive difference, calculating values....... 216

Q

Quintic polynomial, using with function 255

R

Radian conversion factor 247
Reaction Forces 154
  3
Results
  Acceleration Vectors....... 153
  Angular Displacements 150
  Exporting 154
  Exporting to a Text File 142
  Exporting to Excel 97....... 139
    149
  Interference Detection....... 143
  Linear Displacements....... 149
  Reaction Forces....... 154
  Trace Paths....... 148
  Velocity Vectors 152
Rigid Body
  definition....... 18
  ground 79
  mass properties....... 79
Rotational Drivers
  Deleting....... 109
  Editing....... 109
  Template Editor....... 43, 44
Running a Simulation 128

S

  3
Sign, transferring to magnitude....... 249
Simple harmonic function, evaluating 248
Simulation
  Animate during Simulation 124
  Animating 134
  Error Criteria....... 126
  Initial Time Step....... 125
  Integrator....... 125
  Maximum Iterations....... 125
  Maximum Time Step....... 125
  Minimum Time Step 126
  Output Steps....... 123
  Precise Geometry for 3D Contact 127

Running 128
Settings 123, 125
Solver Messages 126
Status 120
Time 124
Troubleshooting 129
Simulation Panel
Animation Controls 131
Buttons 120
DOF Counter 120
Mode 121
Step 121
Time 121
Simulation Panel 119
Simulation:returning current time 259
Sine, calculating 250
Springs 96
Components 95, 97, 101
Creating 95, 97, 101, 103, 105
Deleting 109
Editing 109
Points 95, 97
Springs and Dampers 96
Square root, calculating 252
Step Function 84

T

Tangent, calculating 256
Template Editor
Data Points Function 47, 87
Expression 43
Force Magnitude 83
Harmonic Function 46, 85
Motion Driver 43, 44
Step Function 44, 84
Text File 142
Torque Magnitude
Template Editor 83
Torsion Dampers
Components 103, 106
Creating 103, 105
Deleting 109
Editing 109
Location 103
Orientation 104
Torsion Springs
Components 101
Creating 101
Deleting 109
Editing 109
Location 101
Orientation 102
Trace Paths 148
Translational Driver
Deleting 109
Editing 109
Template Editor 43, 44

V

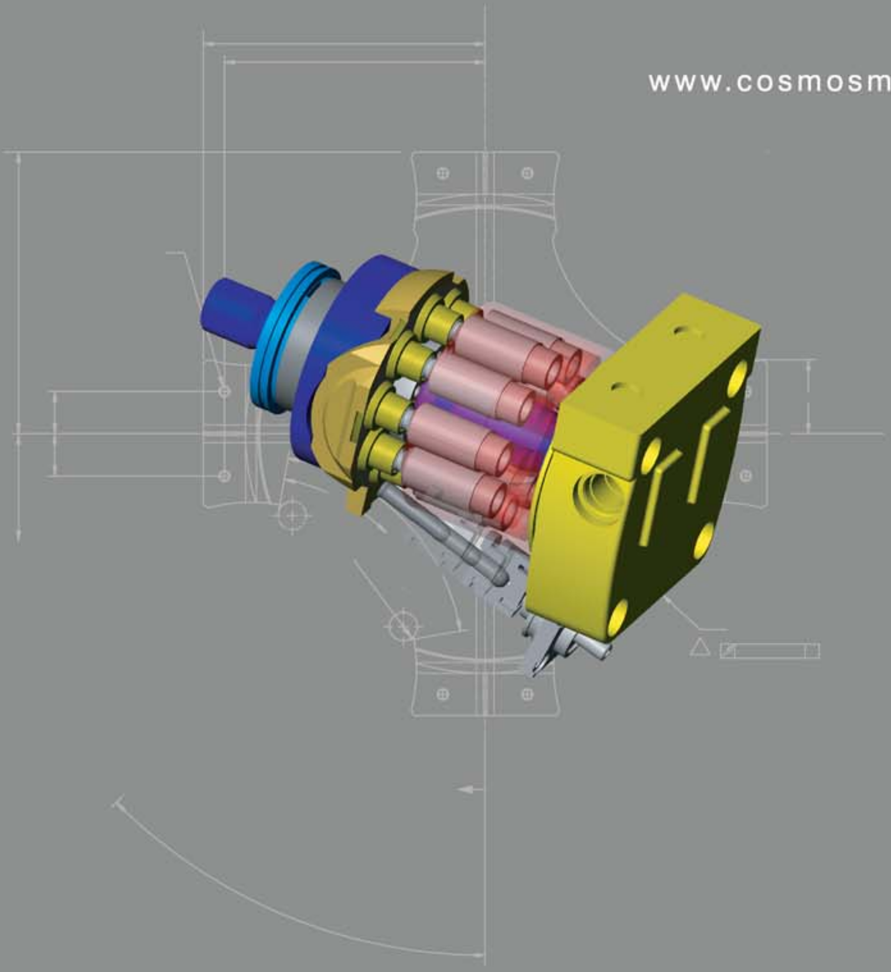
Velocity Vectors 152
VRML File 137

X

XY Plots, Result Viewer
Axes Properties 159
Chart Properties 156
Creating 166
Deleting 169
Double clicking on 169
Fonts 161
Gridlines 158
Moving Marker Properties 165
Numbers/Scale 162
Plot Curve Properties 163
Plot Defaults 155
Plot Layout 157
Plot Persistence 170
X Axis options 167

www.cosmosm.com

www.cosmosm.com



SolidWorks

COSMOS Analysis Products
Tel: +1-800-469-7287
Fax: +1-310-207-7805
Email: info@cosmosm.com
www.cosmosm.com