SimMechanics 2
CAD Translator Guide
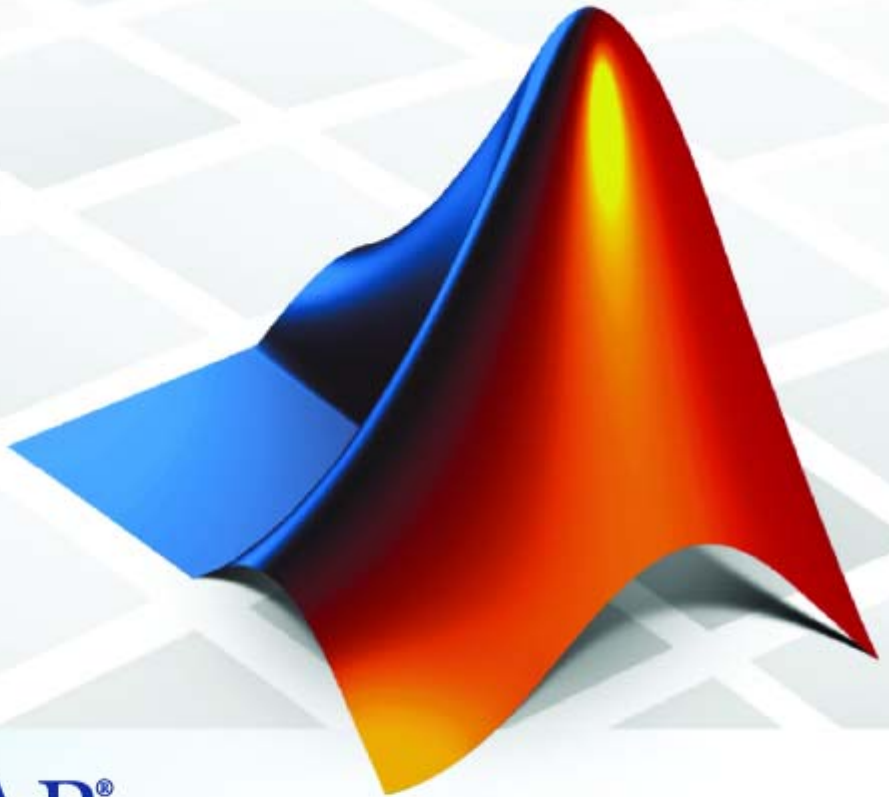
MATLAB®
&SIMULINK®

How to Contact The MathWorks

www.mathworks.com	Web
comp.soft-sys.matlab	Newsgroup
www.mathworks.com/contact_TS.html	Technical Support

suggest@mathworks.com	Product enhancement suggestions
bugs@mathworks.com	Bug reports
doc@mathworks.com	Documentation error reports
service@mathworks.com	Order status, license renewals, passcodes
info@mathworks.com	Sales, pricing, and general information

508-647-7000 (Phone)

508-647-7001 (Fax)

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

SimMechanics CAD Translator Guide

© COPYRIGHT 2003–2007 by The MathWorks, Inc.

Trademarks

MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, and xPC TargetBox are registered trademarks, and SimBiology, SimEvents, and SimHydraulics are trademarks of The MathWorks, Inc.

Other product or brand names are trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Revision History

Contents

Index

About This Guide

This guide is intended to help you use the CAD-to-SimMechanics translator with your computer-aided design (CAD) platform. From your CAD assemblies, the translator generates XML representations ready for use with SimMechanics. The translator is independent of SimMechanics, as well as MATLAB® and Simulink®.

Using This Guide

This guide helps you work with the CAD-to-SimMechanics translator from The MathWorks. The translator enables you to translate CAD assemblies from your CAD platform into a Physical Modeling XML file compatible with SimMechanics. You can then generate a SimMechanics block diagram model from this file to simulate the dynamics of the CAD assembly in the Simulink environment.

This guide covers both aspects of CAD translation, exporting the CAD assembly and generating the model.

- Exporting a CAD assembly into a Physical Modeling XML file requires only your CAD platform and the translator. It does not require MATLAB, Simulink, or SimMechanics.

- Generating a SimMechanics block diagram model requires MATLAB, Simulink, SimMechanics, and the Physical Modeling XML file that you generate from your assembly. It does not require the CAD platform from which you exported the assembly.

The CAD-to-SimMechanics translator is designed to work with your CAD platform. The translator and its associated help files are independent of MATLAB and any product based on MATLAB.

Note This guide refers to sections of the SimMechanics User's Guide. For complete documentation of SimMechanics, including CAD translation and related features, consult the user's guide.

Ways to Get Help Online

Online help is available for the CAD-to-SimMechanics translator and for SimMechanics.

Getting Translator Help

In addition to this guide, there are two sources of online help while you are using the translator.

During the Translator Installation

When you install the translator for your CAD platform, the installer displays a README file containing basic information about the translator, including how to uninstall it.

From the Translator Window

Once you start using the translator in your CAD platform, you can obtain online help in two ways:

- Select **Help** in the **SimMechanics** menu of your CAD platform's interface.

- Click the **?** or equivalent symbol (if available) in the translator window.

After either step, a Web browser opens to display a help page specific to your CAD platform.

Getting SimMechanics Help

There are a number of ways to get online help while you work with MATLAB, Simulink, and SimMechanics.

- *Help browser* — There are several ways to open the Help browser:

  - Select **Full Product Family Help** from the MATLAB **Help** menu.

  - Select **Help** from the MATLAB **View** menu.

  - Enter doc mech at the command line.

  Use the **Contents** pane on the left side of the Help browser to find a section or chapter. Use the **Search** and **Index** features to find specific words.

- *Block library browser* — Click **Help** on the SimMechanics library menu bar to open online Help on Simulink, blocks, shortcuts, S-functions, and demos.

- *Context-sensitive help* — To access the help for a block, right-click the block or click **Help** on the block's dialog box.

- *Command-line help* — Enter doc('block name') at the MATLAB command line to access the help for a block with the name block name. Spaces and capitalization in the block name are ignored.

  If the same block name appears in other blocksets, MATLAB returns an Overloaded methods warning in the command window to flag those instances.

- *SimMechanics block reference* — Expand the **SimMechanics** entry in the **Contents** pane of the Help browser, and select **Blocks — By Category** or **Blocks — Alphabetical List**.

- *Help desk* (*via the Web*) — Use a Web browser or the Help browser to connect to the MathWorks Web site at www.mathworks.com. Click **Support** and follow the **Documentation** link for remote access to the documentation.

Mechanical Units

SimMechanics and the CAD-to-SimMechanics translator accept any mixture of MKS (SI), cgs, and English units.

Quantity	MKS (SI)	cgs	English
Length	meter (m)	centimeter (cm)	inch (in) foot (ft)
Mass	kilogram (kg)	gram (g)	slug (slug)
Angle	radian (rad) degree (deg)	radian (rad) degree (deg)	radian (rad) degree (deg)
Inertia	kilogram-meter^2 (kg-m^2)	gram-centimeter^2 (g-cm^2)	slug-foot^2 (slug-ft^2)

Mechanical Conventions and Abbreviations

Right-Hand Rule Is Assumed

For rotational motion and vector cross products $a \times b$, the right-hand (RH) rule is always assumed.

Vector Multiplication

Scalar-vector products and matrix-vector multiplication are denoted by $a \cdot b$ and $M \cdot v$, respectively.

Common Abbreviations

These are the abbreviations of mechanical terms most commonly used in this guide.

Abbreviation	Meaning
CG	Center of gravity
CS	Coordinate system
DoF	Degree of freedom
RF	Reference frame

Modeling with Computer-Aided Design

Using SimMechanics with computer-aided design (CAD) extends your mechanical modeling and simulation capabilities, allowing you to create SimMechanics models from CAD assemblies.

Introducing CAD Translation

Computer-aided design (CAD) tools allow you to model machines geometrically as collections of parts, or *assemblies*. Simulink and SimMechanics use a block diagram approach to model control systems around mechanical devices and simulate their dynamics. The block diagram approach does not include full geometric information, nor do CAD assemblies typically incorporate controllers or allow you to perform dynamic simulations. With CAD translation, you can combine the power of CAD and SimMechanics.

The translator transforms geometric CAD assemblies into Simulink block diagram models. The intermediary between a CAD assembly and its SimMechanics model is an XML file in a Physical Modeling format. This section covers what you need to get started with CAD translation:

- "CAD Translation Software Requirements" on page 1-3

- "Overview of the CAD Translation Steps" on page 1-3

- "Installing the CAD-to-SimMechanics Translator" on page 1-5

CAD Translation Software Requirements

Before starting to use CAD with SimMechanics, you must first

1 Install your CAD platform or CAD software application.

2 Install the CAD-to-SimMechanics translator appropriate to your CAD platform, using the translator installer. See "Installing the CAD-to-SimMechanics Translator" on page 1-5.

Overview of the CAD Translation Steps

Using the CAD-to-SimMechanics translator with a CAD assembly requires two major steps, exporting the CAD assembly into XML and importing the XML to create a SimMechanics model.

Exporting an Assembly

You export the assembly from the CAD platform into a Physical Modeling XML file that you can later use with SimMechanics. This step requires the CAD platform and the platform-specific CAD-to-SimMechanics translator,

but not MATLAB. See "Exporting CAD Assemblies into Physical Modeling XML" on page 1-6.
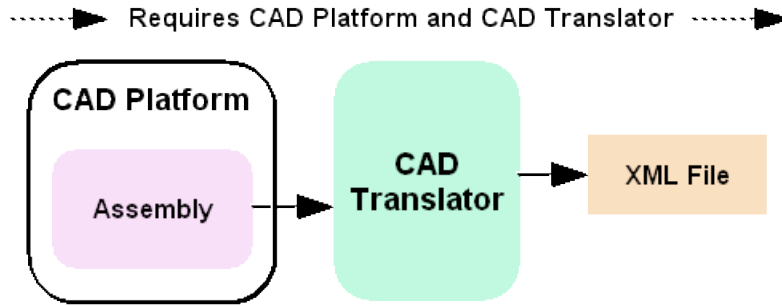
Exporting a CAD Assembly into a Physical Modeling XML File

Importing a Model

You then convert the Physical Modeling XML file into a SimMechanics model in Simulink. This step requires the XML file and SimMechanics, but not the CAD platform or the translator. See "Creating Models from Physical Modeling XML" on page 1-14.
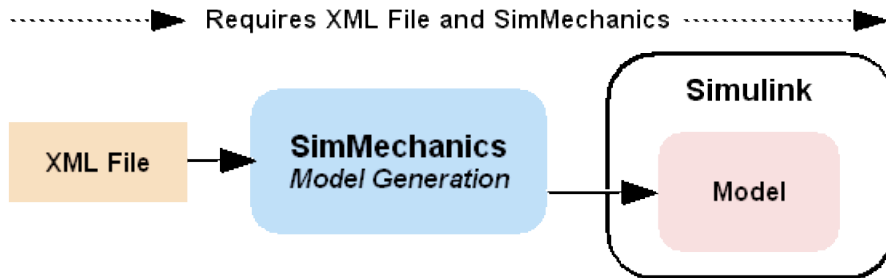
Importing a Physical Modeling XML File into SimMechanics

Installing the CAD-to-SimMechanics Translator

You do not need any MATLAB components to install the CAD-to-SimMechanics translator, but the target CAD platform must be installed.

Downloading the Translator

To obtain the translator, locate and download the self-extracting installer specific to your CAD platform and operating system from the SimMechanics product Web page, `www.mathworks.com/products/simmechanics/`.

Installing the Translator

Once you have obtained the translator download, run the installer and follow the instructions in the README file provided. Print this file for future reference.

Installing the Translator over a Network

You can install the translator with the installer, your CAD platform, or both, on your network. Use the correct paths to specify the locations. If you can, map these network locations as drives on your computer.

Linking the Translator to Your CAD Platform

To configure your CAD platform to work with the translator, you need to link the translator to the CAD platform so that it is enabled and available as you work with an assembly.

Consult the translator's README instructions file and your CAD platform's documentation for further information about linking.

Finding the Translator Help and Example Files

Once your translator is installed, the target installation directory includes subdirectories containing help files for your CAD platform and example files containing preconstructed assemblies to learn with.

Exporting CAD Assemblies into Physical Modeling XML

The CAD-to-SimMechanics translator converts an existing computer-aided design (CAD) assembly into a Physical Modeling XML file that is portable and independent of SimMechanics. From this XML file, you can generate a SimMechanics model, as discussed in "Creating Models from Physical Modeling XML" on page 1-14.

Note You do not need MATLAB or any MATLAB component to use the CAD-to-SimMechanics translator, but you do need to have the target CAD platform installed.

When you export from your CAD platform, you must export a complete assembly into XML, not just a part.

This section explains how to export CAD assemblies into the Physical Modeling XML format.

- "Building a CAD Assembly for SimMechanics" on page 1-6 explains the requirements for a CAD assembly to produce a valid SimMechanics model.

- "Translating CAD Assemblies into XML" on page 1-10 walks you through the steps to export the XML file representing the CAD assembly.

- "Troubleshooting Assembly Export Problems" on page 1-12 examines some of the problems you might encounter when you export a CAD assembly into a Physical Modeling XML representation.

- "Getting Help in the Translator Window" on page 1-12 shows you how to get online help while you are working with the translator.

Building a CAD Assembly for SimMechanics

The CAD-to-SimMechanics translator creates a Physical Modeling XML file that represents the assembly's parts as bodies and maps the constraints between the parts into joints. You need to specify enough information in your CAD assembly for SimMechanics to construct a dynamically meaningful model from the XML file.

CAD Assembly Component	Corresponding SimMechanics Blocks
Part	Body
Constraints *	Joints
Fundamental Root	Ground – Root Weld – Root Body
Subassembly	Subsystem
Subassembly Root: [[*trunk*]] – constraint(s) – *subassembly*	[[Root Body – Root Weld – Fixed Body]] – Joint – *subsystem* ....
Fixed Part (in a subassembly)	Root Body – Weld – Body

* Constraints on parts in a CAD assembly are sometimes called *mates*.

Roots and Root Bodies

Every CAD assembly has a single *fundamental root*, a fixed point that does not move. The positions and orientations of all parts refer directly or indirectly to this fundamental root. The translator converts the fundamental root into a unique Ground – Weld – Body combination in SimMechanics.

A *root body* is a zero-mass, zero-inertia body used in the generated SimMechanics model to represent a CAD root. A root body is always welded to ground, so that its zero mass and zero inertia do not affect the model's dynamics. A root body is necessary, in general, to represent a fixed anchor for part constraints in the original assembly. This body can carry multiple coordinate systems for this purpose, while the single Ground block in the generated model can carry only one.

Subassemblies and Hierarchies

You can isolate a collection of CAD components (parts and their constraints) into a *subassembly*. The translator converts subassemblies into subsystems in SimMechanics.

The main assembly is like the trunk of a tree, and its subassemblies are like the branches of the tree. Subassemblies can have subassemblies, and so on. This tree is the assembly's *hierarchy*. Each CAD subassembly has its own *subassembly root*. A *fixed part* of a CAD subassembly is a part that is welded to the subassembly root. It cannot move relative to the subassembly root.

See "Creating a CAD-Based Robot Arm Model" on page 1-40 for an example of subassembly hierarchy.

Improving Your Assembly with Subassemblies

Use subassemblies to organize your assembly hierarchically. This will simplify your subsequent SimMechanics model by grouping blocks into corresponding subsystems. Follow these guidelines to ensure that your CAD assembly translates into a functioning SimMechanics model:

- You must have at least one fixed part inside each subassembly. (See "Subassemblies and Hierarchies" on page 1-7 for more on fixed parts.)

- Put as many welded components as you can inside rigid subassemblies or combine welded components into a single equivalent part.

  If your assembly has a group of parts that do not move relative to one another, model them so that the translator treats this group as a single part, eliminating unnecessary Body and Joint blocks from your subsequent SimMechanics model.

- Avoid imposing constraints between a subassembly part and geometrical abstractions such as the *x-y*, *y-z*, and *x-z* planes of the subassembly. Instead, impose constraints between the part and a fixed part in the subassembly. Normally, this fixed part is the subassembly root and translates to the Fixed Body in the Root Body – Root Weld – Fixed Body sequence.

Mass Properties of Assembly Parts

The CAD assembly's parts need to have masses and inertia tensors. When you generate the SimMechanics model, this mass property information is used to specify the properties of the SimMechanics Body block corresponding to each assembly part.

Your CAD platform might compute masses and inertia values from the mass density and geometry of the assembly parts. Otherwise, you must specify the mass and inertia tensor with respect to the part's center of gravity. The translator computes the center of gravity of each part automatically.

See "Exporting a CAD Part" on page 1-23 for an example.

Constraint Geometries

The constraints in your CAD assembly restrict how the assembly's parts can move with respect to each other. Without any constraints, a pair of CAD parts can move with six unrestricted degrees of freedom (DoFs) relative to one another. Constraints between pairs of parts reduce the six to fewer DoFs. In SimMechanics, joints express DoFs between bodies because bodies by themselves carry no DoFs. Constraints and joints are complements of one another.

You must specify the constraint geometry in the CAD assembly consistently and in enough detail for SimMechanics to reconstruct the assembly's DoFs as joints. The relationship between constraints in CAD and joints in SimMechanics is not, in general, a simple mapping. Some SimMechanics joints have only one DoF, while others represent more than one DoF. The translator often combines multiple DoFs into one joint. Constraint specification details often depend on the specific CAD platform.

Each joint is connected to each of two bodies at a body coordinate system (CS). The constraint geometry determines the joints into which the translator transforms the constraints and controls the position and orientation of the body CSs. Each of these body CSs has an origin and axis triad fixed relative to its body. The translator creates body CSs on the bodies as necessary for connecting joints.

See "Creating a CAD-Based Robot Arm Model" on page 1-40 for an example of configuring constraints.

Avoiding Redundant Constraints

Keep constraints simple and few enough to avoid creating unnecessary joints in your SimMechanics model.

For example, consider three parts, P1, P2, and P3, in an assembly. Suppose P1 and P2 are constrained so that there is no movement possible between them. When you attach P3, you could put one constraint between P3 and P1 and the other between P3 and P2. This leads to a redundant joint in the SimMechanics model, making it harder to understand and troubleshoot than if you created only one constraint. In this example, it is better to create a constraint just between P3 and P2, since P2 cannot move with respect to P1 anyway.

Translating CAD Assemblies into XML

To translate the CAD assembly into a form that SimMechanics can use, you must check and configure the translator settings, then save the assembly through the translator into the Physical Modeling XML format.

Applying the Translator Settings

Open your CAD assembly.

1 Open the SimMechanics settings interface for your CAD platform.

In this interface, you set the tolerance configurations. See "Configuring Tolerances" on page 1-10.

2 At any time, you can

- Apply your settings.

- Cancel your settings. You lose whatever new settings you have entered.

Configuring Tolerances

In the **Settings** area, you can configure one or more of the translation tolerances. Geometrical and numerical differences smaller than the tolerances are treated as zero. The entries implicitly have the same units selected in your CAD platform settings.

- **Linear tolerance** specifies the smallest significant difference in length.

- **Angular tolerance** specifies the smallest significant difference in angle.

- **Relative roundoff** specifies the smallest significant numerical difference.

Creating the XML File

To complete translation of the assembly into a Physical Modeling XML file,

1 Apply your translator settings.

2 If you changed the assembly or any subassemblies, you need to rebuild the assembly and resave it in its native format before exporting it to XML.

3 Export the assembly into XML format. The default name is the same as that of the CAD assembly file. You can change the XML filename and directory at this point.

4 Click **OK**.

The assembly is saved in the new form as an XML file.

Note To use the exported XML file to automatically generate a SimMechanics model, you need to move or copy the file into a MATLAB working directory. See "Creating Models from Physical Modeling XML" on page 1-14.

Troubleshooting Assembly Export Problems

The CAD-to-SimMechanics translator can encounter difficulties when it attempts to represent your assembly in the XML file as SimMechanics bodies and joints.

Constraint Translation Errors

Constraint translation errors occur when you specify a constraint in your CAD assembly that is not supported for export. Constraints (or mates) supported for a specific CAD platform are listed in its help page (see "Getting Help in the Translator Window" on page 1-12).

If the translator fails to map one or more constraints into joints, it issues one or more error dialogs and logs the errors into a text file. The error dialog indicates the name of this error log file, which is located in the same directory as the exported Physical Modeling XML file. The XML file is generated regardless of constraint translation errors.

The XML file itself contains the same errors, each paired with the corresponding failed joint. The constraints that failed to translate properly are converted instead into welds. These errors reappear as MATLAB command line warnings if you generate a SimMechanics model from the XML file.

Subassembly Configuration Errors

Be sure that you configure your subassemblies' positions and orientations to be consistent with the main assembly's configuration *before* you export the assembly.

If subassemblies are not consistent with their main assembly, the resulting model will not be valid and will encounter simulation errors.

Getting Help in the Translator Window

The translator installation for your platform includes online help files that are independent of the MATLAB Help system.

Getting HTML Help

To obtain online translator help in your CAD platform, select the help option available in the SimMechanics interface for your CAD platform.

Your default Web browser opens and displays an HTML help file specific to your CAD platform.

Getting PDF Help

You can also access a translator guide in PDF format by looking in your CAD platform's CAD-to-SimMechanics translator installation directory.

Creating Models from Physical Modeling XML

A CAD assembly can contain enough part and constraint information that you can generate a SimMechanics model consisting of Body and Joint blocks representing the assembly. Once you have exported a CAD assembly into a Physical Modeling XML file, you can generate a SimMechanics block diagram model with this file. Although the generated model will run, you often need to manually simplify and complete it to faithfully represent the original dynamic system.

- "Generating Body-Joint CAD-Based Models" on page 1-14 shows how to import the XML file into SimMechanics using the command import_physmod, which constructs the model.

- "Common Features of CAD-Based Models" on page 1-16 explores the characteristics of an automatically generated CAD-based model.

- "Editing and Completing Generated Models" on page 1-16 explains when and how to edit your model and expand it with additional blocks, such as Constraints, Drivers, Actuators, and Sensors.

- "Troubleshooting CAD-Based Models" on page 1-19 points out major model-generation and simulation errors and problems that can arise from CAD-based models, and techniques for solving them.

Note This section assumes that you have SimMechanics installed locally or remotely, and that you have the XML file representing a CAD assembly in your current MATLAB directory.

To generate the SimMechanics model, you do not need the CAD platform from which the XML file was exported or the CAD-to-SimMechanics translator.

Generating Body-Joint CAD-Based Models

You generate the CAD-based SimMechanics model with the import_physmod command.

1 Move or copy the Physical Modeling XML file you want to use into your MATLAB working directory.

2 To start the model generation from an XML file called `cad_assembly.xml`, enter

```
import_physmod('cad_assembly.xml')
```

at the command line. A progress bar appears and is updated as the model is imported.
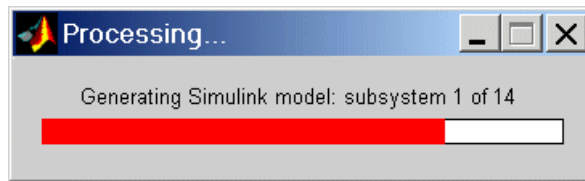
A Simulink model window opens. The model is populated by Bodies and Joints corresponding to the assembly parts and constraints saved in `cad_assembly.xml`. (The corresponding connections and hierarchy are explained in "Exporting CAD Assemblies into Physical Modeling XML" on page 1-6.) The name of the generated model is the name of the original assembly file, regardless of the name chosen for the XML file.

Changing the Appearance of a Generated Model

You can change the appearance of your generated model by using optional import settings. See the `import_physmod` command reference for more details.

For example, entering

```
import_physmod('cad_assembly.xml','FontSize',18)
```

at the command line generates a model with a block label font size of 18 pixels.

Using the Physical Modeling XML File Import Dialog

You can also select a Physical Modeling XML file and generate a model from it through the **Physical Modeling XML File Import** dialog. Open it by entering `import_physmod` at the command line with no arguments.

See the `import_physmod` command reference for complete information about the XML file import dialog.

Common Features of CAD-Based Models

Most of the properties of models that you generate from an XML file are the same as the default properties of all Simulink models. See the Simulink documentation for general information on working with Simulink models.

Models generated from a CAD-based XML file have certain common features:

- Exactly one Ground block and connected Machine Environment block
- Fundamental root, represented by Ground – Root Weld – Root Body
- Subassembly roots, represented by Root Body – Root Weld – Fixed Body
- Joints with degrees of freedom (DoFs) containing the correct joint primitives for the translational and rotational DoFs between any two bodies
- Joints without DoFs, represented by Welds

Joints and Subsystem Hierarchy

Joints that directly connect a subsystem's Body to the next higher level in the hierarchy appear in that next higher level, not within the subsystem. This means that the boundary between subassembly and main assembly in CAD is drawn differently from the boundary between subsystem and top-level system in SimMechanics. This difference does not change the degrees of freedom of the model. They are the same in a CAD assembly and in the SimMechanics model generated from the assembly.

Editing and Completing Generated Models

The import_physmod command generates a model containing only Machine Environment, a Ground, Bodies, and Joints. A complete SimMechanics model typically contains other blocks, including Constraints, Drivers, Sensors, and Actuators (from the mechlib block library), as well as blocks from Simulink, such as Scopes. Creating a complete SimMechanics model requires inserting and connecting these additional blocks in your generated model. Use the relevant sections of "Modeling Mechanical Systems" in the SimMechanics User's Guide to learn how to use these blocks.

> **Note** When you first generate a model from Physical Modeling XML, you
> might want to save this original CAD-based model before you create later
> versions by eliminating unnecessary blocks and adding new ones.

Deleting Unnecessary Blocks

The constraint-to-joint mapping creates whatever blocks are needed to
correctly isolate the degrees of freedom (DoFs). In some models, certain
generated blocks are not needed for simulation. You can delete these
unnecessary blocks to simplify your model without affecting its dynamics, as
long as you reconnect the remaining blocks properly.

In particular, you can simplify the fundamental root, Ground – Root Weld –
Root Body, in certain cases. (The same holds for subassembly roots. However,
the corresponding subsystem's Root Body takes the place of Ground, and the
subsystem's Fixed Body plays the role of Root Body.)

- If the Root Body is connected to the rest of the block diagram by a single
  Joint, you can delete the Root Weld – Root Body blocks and reconnect the
  Joint directly to Ground.

- If the Root Body is connected to the rest of the block diagram by multiple
  Joints, you cannot delete the Root Weld – Root Body sequence because a
  Ground can connect to only one Joint, and the generated model contains
  only one Ground. You have two alternatives in this case.

  - Determine where the Root Body coordinate systems (CSs) are. Delete
    the Root Weld – Root Body sequence. Then manually add more Grounds
    at the spatial points representing each of the deleted Root Body CS
    origins. Reconnect each Joint to its corresponding Ground at the correct
    spatial point.
  - Leave the Root Weld – Root Body sequence as it is.

See "Building a CAD Assembly for SimMechanics" on page 1-6.

Caution Not all Welds are redundant. Some are needed to connect a Body that would otherwise be isolated from the rest of the machine. Such Welds represent rigid connections between distinct assembly parts.

If you want to reduce or eliminate such Welds, return to your original assembly and reexport it with rigid subassemblies. See "Improving Your Assembly with Subassemblies" on page 1-8.

Constraining and Driving Degrees of Freedom

Constraints reduce the number of independent degrees of freedom in a machine by preventing certain movements. Drivers affect DoFs, not by eliminating them completely, but by forcing their motions to follow an external time-dependent signal. A constrained or driven DoF is not free to respond to forces and torques independently.

These sections in the SimMechanics User's Guide discuss how to add constraints and drivers to your model:

- "Modeling Constraints and Drivers"

- "Actuating a Driver"

Note You insert these constraints into the SimMechanics model as blocks, and they act in addition to the part constraints in the original CAD assembly.

Actuating Bodies and Joints with Motions and Forces

SimMechanics gives you the ability to actuate bodies and joints in your models with external forces and motions, and to set up internal forces between bodies. You can also vary the mass or inertia (or both) of a body in time. Consult these sections in the SimMechanics User's Guide for more details:

- "Actuating a Body"

- "Varying a Body's Mass and Inertia Tensor"

- "Actuating a Joint"

- "Modeling Force Elements"

Setting the Model's Initial Conditions

When you import a CAD assembly into SimMechanics, the XML file determines the initial geometric configuration of the model's bodies. By default, the initial velocities of the bodies are zero.

If you want to change the initial positions and/or velocities of the bodies in your model to be different from the CAD-determined configuration, you need to add initial condition actuators, as discussed in "Specifying Initial Positions and Velocities" in the SimMechanics User's Guide.

Sensing Forces and Motions

To detect the motions of bodies and joints and measure the forces acting on or through them, you can add sensors to your model. See "Modeling Sensors" in the SimMechanics User's Guide.

Satisfying General SimMechanics Requirements

You can find the general instructions for building valid SimMechanics models in these sections of the SimMechanics User's Guide.

- "Modeling Machines"
- "Checking Model Validity"

Troubleshooting CAD-Based Models

Problems might arise in your generated SimMechanics model if your CAD assembly was not constructed or exported properly.

General Guidelines

- See "Building a CAD Assembly for SimMechanics" on page 1-6 instructions on constructing a CAD assembly specifically for SimMechanics.
- See "Translating CAD Assemblies into XML" on page 1-10 for details about exporting a CAD assembly.

Troubleshooting Errors During Model Generation

Errors in the Physical Modeling XML file appear as warnings at the MATLAB command line during model generation.

These warnings arise from CAD constraint translation errors encountered when the XML file was originally exported. Such errors occur when the translator fails to map one or more CAD constraints, which restrict the degrees of freedom between parts, into their corresponding SimMechanics joints. The translator warns you at the export step if such errors occur.

The failed Joint appears in your generated model as a Weld. You can fix such an error in two ways:

- In the generated model, manually replace the Weld with the proper Joints.

- Return to the original CAD assembly, reconfigure the constraints, and export it again.

Consult "Constraint Translation Errors" on page 1-12 for more about export errors.

Troubleshooting Model Dynamics Errors

Certain problems with CAD-based models appear only when you run the model.

- You must "fix" at least one part in every CAD subassembly by mating it to the subassembly root. Otherwise, the massless root body is dynamically active and experiences infinite acceleration when forces or torques are applied to it. See these sections for more details:
  - "Roots and Root Bodies" on page 1-7
  - "Subassemblies and Hierarchies" on page 1-7
  - "Improving Your Assembly with Subassemblies" on page 1-8
- If you find constraints are violated while SimMechanics is running your model, try the following:
  - Examine your original CAD assembly for redundant constraints.

- Check and possibly increase the assembly tolerances at the original CAD export step.

- Check and possibly increase the translated model's assembly tolerances in the machine's Machine Environment block, in the **Parameters** pane.

- On the **Constraints** panel in the Machine Environment dialog, select the **Use robust singularity handling** check box.

- Never *decrease* assembly tolerances in a CAD-based SimMechanics model. Instead, decrease the assembly tolerances at the original CAD export step.

Troubleshooting SimMechanics and Simulink Problems

You might also encounter general Simulink or SimMechanics problems while running your model.

- For problems specific to SimMechanics, see "Troubleshooting Simulation Errors" in the SimMechanics User's Guide.

- For general Simulink problems, consult the Simulink documentation.

Overview of CAD Translation Examples

Note The following CAD assembly and translation examples are based on SolidWorks® and the SolidWorks-to-SimMechanics translator.

These examples can be recreated with other CAD platforms. The assembly, geometric, kinematic, and part details differ from platform to platform. In SolidWorks, constraints on CAD parts are called *mates*.

The following sections of this chapter illustrate the stages of converting CAD assemblies into SimMechanics models in Simulink. Each section presents an example or set of examples based on a specific machine type represented in CAD and demonstrates how to apply in specific situations the procedures outlined in the chapter's preceding sections. The SimMechanics models in each section are generated with default settings.

- "Exporting a CAD Part" on page 1-23 presents the simplest case, translating an assembly with a single part or body.

- "Designing and Exporting CAD Constraints" on page 1-27 examines in detail how you translate assemblies with constrained parts into bodies with degrees of freedom represented by SimMechanics Bodies and Joints.

- "Creating a CAD-Based Robot Arm Model" on page 1-40 illustrates the translation of an assembly representing a simple machine, including subassembly-subsystem hierarchy. This section also introduces post-translation additions to the model.

- "Modeling a Stewart Platform in CAD" on page 1-47 illustrates the translation of a moderately complex machine assembly with many degrees of freedom, subassembly-subsystem hierarchy, and visualization of its motion.

Exporting a CAD Part

In this example, you export an assembly with one part and no constraints. Locate the two example CAD files:

- The full assembly file, cup_assembly.*ASSEMBLYFILETYPE*

- The part, a cup, in a file called cup.*PARTFILETYPE*

Although it has only one part, you must export the full assembly into XML, not just the cup part.

- "Viewing the CAD Assembly" on page 1-23

- "Exporting the CAD Assembly" on page 1-25

- "Generating the SimMechanics Model" on page 1-25

Viewing the CAD Assembly

Open the cup assembly file in your CAD platform and check its geometry and mass properties.

CAD hierarchy   Base origin
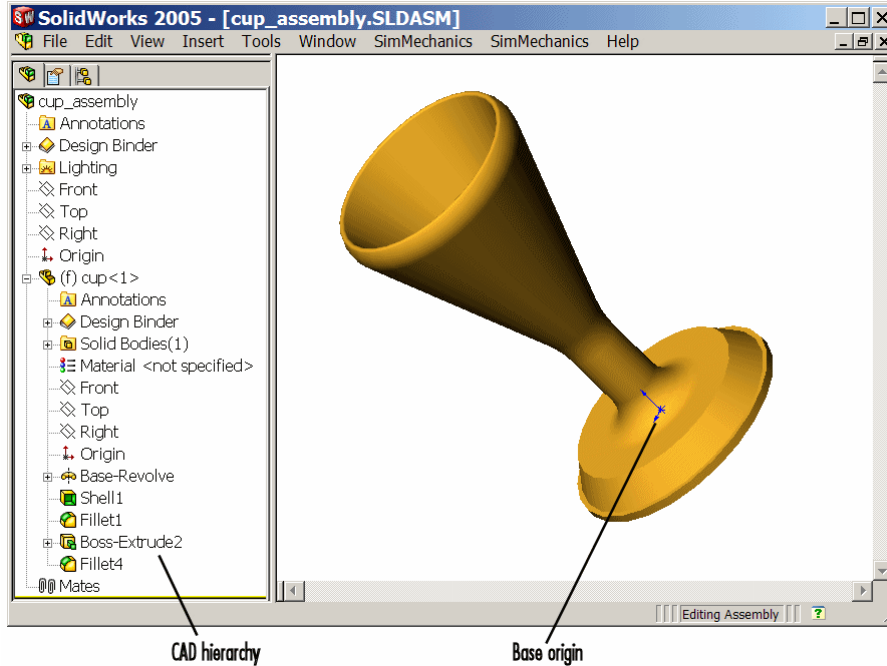
Cup Assembly in a CAD Platform

Property	Value
Volume	0.0001 cubic meters (m^3)
Surface area	0.0381 square meters (m^2)
Density	3.0 grams/cm^3 = 3000 kg/m^3
Mass	0.2906 kilograms (kg)
Principal moments of inertia at the center of gravity	$I_x = 0.00015$, $I_y = 0.00067$, $I_z = 0.00067$ kg-m^2

The inertia tensor is computed with the origin at the center of gravity and the coordinate axes aligned with base-origin axes, indicated in the figure Cup Assembly in a CAD Platform on page 1-24. The x-axis is the cup's axis of symmetry, and the y- and z-axes point across the cup.

Exporting the CAD Assembly

Now export the assembly into Physical Modeling XML format. The XML file
cup_assembly.xml appears in your working CAD directory.

See "Exporting CAD Assemblies into Physical Modeling XML" on page 1-6 for
a discussion of converting CAD assemblies into XML format.

Generating the SimMechanics Model

See "Creating Models from Physical Modeling XML" on page 1-14 for complete
instructions on how to generate models from XML files.

1 Move or copy the exported XML file into a MATLAB working directory to
generate a SimMechanics model from the file.

2 Generate the model from cup_assembly.xml with the import_physmod
command.
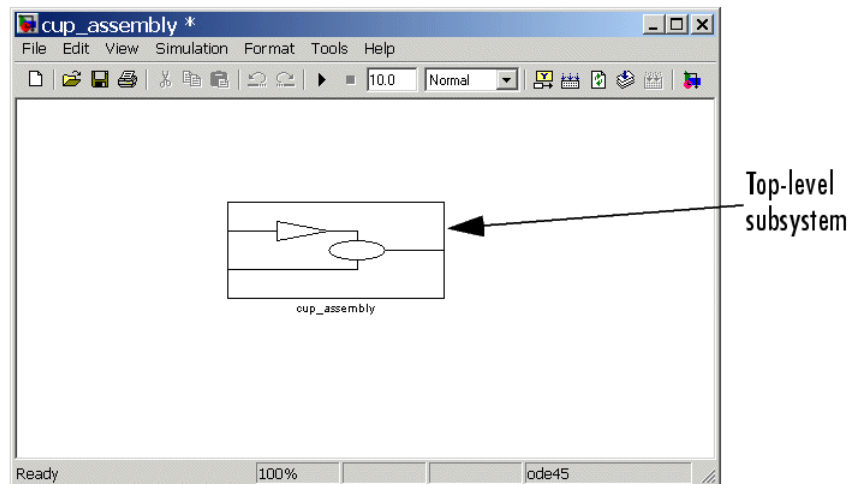
Once you generate the SimMechanics model, it has six blocks, a combination

   Machine Environment – Ground – Weld – Root Body – Weld – Cup

inside a subsystem, called cup_assembly, representing the entire assembly.

1-25

- The Root Body is the nondynamical zero-mass/zero-inertia body inserted between ground and the cup.

- The second joint is a Weld because the original CAD assembly has no degrees of freedom.

Deleting the Root Body and one of the Welds does not physically change the model, as long as you reconnect the remaining blocks.

Designing and Exporting CAD Constraints

In "Exporting a CAD Part" on page 1-23, you create and export an assembly composed of a single part. The CAD-to-SimMechanics translator converts the part into a body, with a mass, an inertia tensor, and body coordinate systems (CSs). Because there are no other parts in that CAD assembly, the SimMechanics body is welded to ground and has no degrees of freedom (DoFs). This lack of DoFs is not realistic for most assemblies.

This section presents a set of complete CAD assemblies with both parts and constraints. Each example assembly consists of two instances of the same part file, representing two identical cubes.

- "Locating the Example Assembly Files" on page 1-28 shows how to find the files needed for this section.

- "Common Steps for Generating the Two-Part Models" on page 1-28 presents the essential steps for generating these models.

- "Block Structure of the Two-Part Models" on page 1-29 discusses the common structure of all the generated models in this section.

In different assemblies, the two cubes are constrained with different constraint combinations to create different relative DoFs between the cubes. You can typically represent a set of DoFs with a large number of different constraint combinations. Each constraint combination here, in general, is *not* the unique way to create the corresponding set of DoFs.

- "Modeling a Six-DoF Joint" on page 1-30 assembles the two cubes with no constraints so that the cubes have the full six degrees of freedom relative to one another.

- "Modeling a Prismatic Joint" on page 1-31 shows how to constrain the two cubes in two different ways so as to produce the same result, a single prismatic (translational) DoF between them.

- "Modeling a Revolute Joint" on page 1-35 constrains the two cubes so as to allow only a single revolute (rotational) DoF between them.

- "Modeling an Inplane Joint" on page 1-36 constrains the two cubes so as to allow two prismatic (translational) DoFs between them.

- "Modeling a Spherical-Spherical Massless Connector" on page 1-37 constrains the cubes so as to allow relative spherical joint motion, with the two cubes separated by a constant nonzero distance.

Locating the Example Assembly Files

Locate the CAD assembly files used as examples for this section. The assemblies have the generic name `<assembly-name>.ASSEMBLYFILETYPE`. The cube part is contained in `magic_cube.PARTFILETYPE`.

Assembly Name	Assembly Configuration
sixDOF	Two cubes with no constraints
prismatic1	Two cubes with planar and cylindrical constraints
prismatic2	Two cubes with planar constraints
revolute	Two cubes with planar and cylindrical constraints
inplane	Two cubes with planar constraints
spherical_spherical_ massless_connector	Two cubes with a distance constraint

Common Steps for Generating the Two-Part Models

The steps for exporting a two-part assembly and generating SimMechanics models based on it are essentially the same for all the examples of this section.

Viewing and Exporting an Assembly

To see a two-part assembly and export it into Physical Modeling XML,

1 Open the assembly `<assembly-name>.ASSEMBLYFILETYPE`. The two parts are `magic_cube-1` and `magic_cube-2`.

In the CAD hierarchy, note any constraints imposed on the parts. These constraints define the relative DoFs between the parts.

2 Now translate this CAD assembly into Physical Modeling XML format. The XML file is saved in your current working CAD directory.

See "Exporting CAD Assemblies into Physical Modeling XML" on page 1-6 for a discussion of converting CAD assemblies into XML format.

Generating a Model

Now you can generate a model based on this assembly.

1 Move or copy the XML file to a working MATLAB directory. Then open MATLAB in that directory.

2 At the command line, enter `import_physmod('<assembly_name>')`. SimMechanics automatically generates a model, `<assembly_name>.mdl`, based on `<assembly_name>.xml`.

The entire assembly is translated into a subsystem, also called `<assembly_name>`, within the model.

3 Open the subsystem. The blocks are arranged in the common structure described in "Block Structure of the Two-Part Models" on page 1-29. A set of Joints represents the DoFs between the two cubes.

Block Structure of the Two-Part Models

All the models that you generate in this section from the example CAD assemblies have a common structure because each assembly has a fundamental root and two moving parts. Each model has eight blocks.

- *The assembly's fundamental root.* As in any generated CAD-based model, the four-block combination Machine Environment – RootGround – RootWeld – RootPart represents the assembly's fundamental root. The RootPart is a nonmoving, zero-mass/zero-inertia body.

  For more about roots, see "Building a CAD Assembly for SimMechanics" on page 1-6 and "Common Features of CAD-Based Models" on page 1-16.

- *The moving bodies.* The bodies representing the assembly's parts are `magic_cube-1` and `magic_cube-2`.

- *The joints.* In all the models, the first cube is connected by a Weld to RootPart and cannot move. The second cube is connected to RootPart by a Joint that represents the appropriate degrees of freedom (DoFs).

Depending on the DoFs in question in a particular assembly, the translator configures the Joint to represent different DoFs with combinations of prismatic, revolute, and spherical primitives. The second cube can move with respect to the first through the DoFs represented by the Joint.

Some of the blocks in the generated models are redundant. You can manually edit and simplify the models without changing their physical properties. For more about manual editing of generated models, see "Editing and Completing Generated Models" on page 1-16.

Restricting Degrees of Freedom with Constraints

CAD platforms normally assume that two parts with no constraints between them have the complete six relative DoFs possessed by any rigid body relative to another body. You restrict the DoFs between parts by connecting them with constraints in the CAD assembly. Constraints restrict relative body motion and reduce the number of relative DoFs between body pairs. There is always one assembly part welded to ground.

Modeling a Six-DoF Joint

The simplest assembly with two parts has no constraints between the parts. The parts can move with respect to one another with all six degrees of freedom (DoFs).

Exporting the Assembly

To see and export such an assembly,

1 Open the assembly `sixDOF.ASSEMBLYFILETYPE`.

  Note, in the CAD hierarchy, that the constraints (Mates) node has no entries. Therefore, relative to one another, the cubes are unconstrained in their motion and have six relative DoFs.

2 Translate this CAD assembly into `sixDOF.xml`.

Generating the Model

Now generate a model based on this assembly.

1 At the MATLAB command line, enter import_physmod('sixDOF').
   SimMechanics automatically generates a model, sixDOF.mdl.

2 Open the subsystem sixDOF. There are eight blocks.

   The Six-DoF Joint represents the six DoFs between the two cubes with one
   spherical and three prismatic primitives.
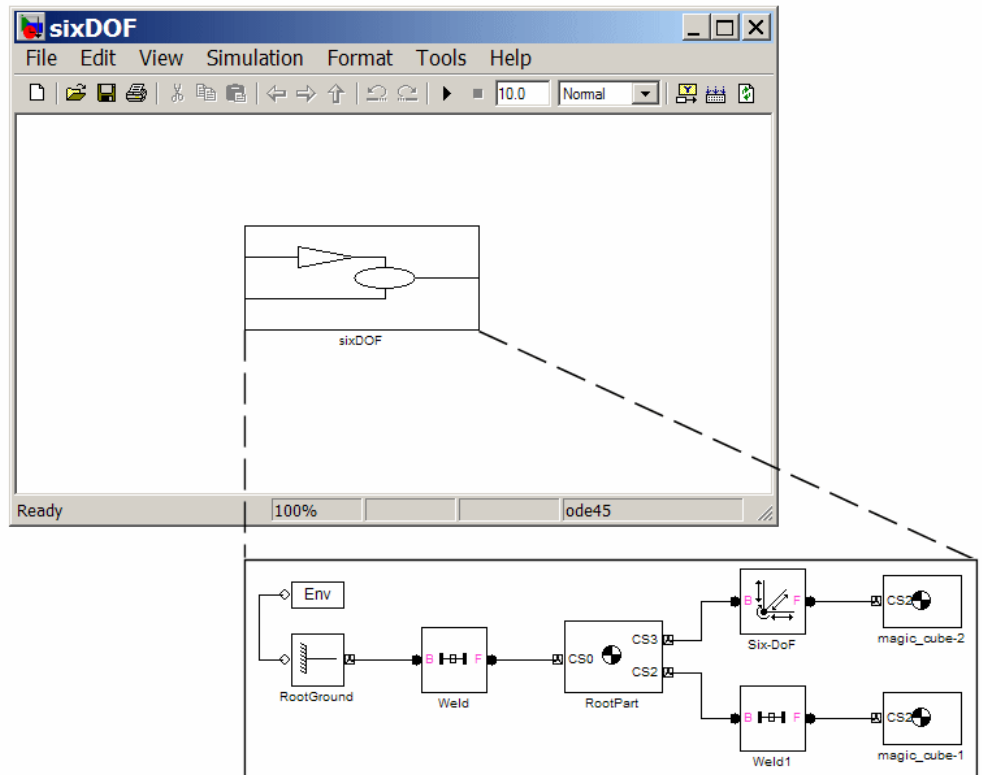
Modeling a Prismatic Joint

In the following two assemblies, the two cubes are constrained to have
only a single translational degree of freedom (DoF) between them. These
assemblies illustrate two ways to accomplish this; you can experiment with

other constraints to find more. In the translated SimMechanics models, this single DoF is a prismatic joint.

Prismatic as a Planar Constraint and a Cylindrical Constraint

To see the first way of constraining the DoFs to produce a prismatic joint,

1 Open the assembly file prismatic1.*ASSEMBLYFILETYPE* and examine the CAD hierarchy.

2 Locate and expand the constraints (Mates) node. There are two constraints on the two cubes.

- Highlight the first constraint, Concentric1. The constraint geometry is highlighted in the assembly. This constraint allows the two cubes only to slide along and rotate about the *z*-axis running through the center of the parallel and concentric upper holes of each cube.

- Highlight the second constraint, Coincident2. The constraint geometry is highlighted in the assembly. This constraint allows the two cubes to slide along the *y-z* plane, with the two sides marked "SimMechanics" sharing a common plane, representing two translational DoFs. It also allows the two cubes to rotate about the *x*-axis. The cubes are not allowed to rotate about any other axis, or to translate perpendicular to the *y-z* plane.

These two constraints mean that the two cubes can only slide along the *z*-axis common to the two upper concentric holes. The second constraint prevents rotation about this axis, leaving the whole assembly with only one translational DoF.
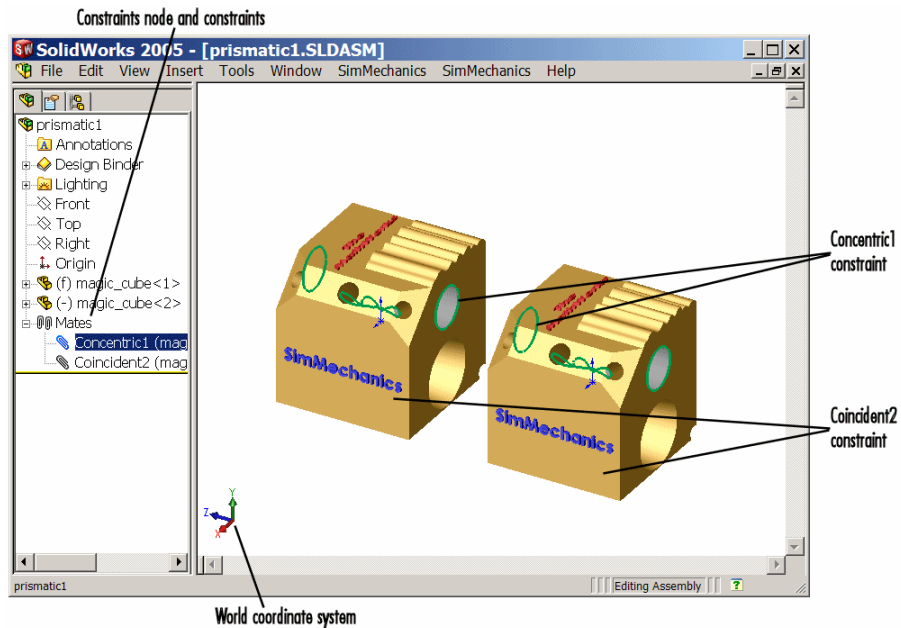
Planar and Cylindrical Constraints on Two Cubes

Prismatic as Two Orthogonal Planar Constraints

To see the second way of constraining the DoFs to produce a prismatic joint,

1 Open the assembly file prismatic2.*ASSEMBLYFILETYPE* and examine the CAD hierarchy.

2 Locate and expand the constraints (Mates) node. There are two constraints on the two cubes.

- Highlight the first constraint, Coincident2. The constraint geometry is highlighted in the assembly. This constraint allows the two cubes to slide along the *y-z* plane, with the two sides marked "SimMechanics" sharing a common plane, representing two translational DoFs. It also allows the

two cubes to rotate about the x-axis. The cubes are not allowed to rotate about any other axis, or to translate perpendicular to the y-z plane.

- Highlight the second constraint, Coincident3. The constraint geometry is highlighted in the assembly. This constraint allows the two cubes to slide along the x-z plane, with the two sides marked "The MathWorks" sharing a common plane, representing two translational DoFs. It also allows the two cubes to rotate about the y-axis. The cubes are not allowed to rotate about any other axis, or to translate perpendicular to the x-z plane.

These two constraints mean that the two cubes can only slide along the z-axis common to the two planes y-z and x-z, leaving the whole assembly with only one translational DoF.
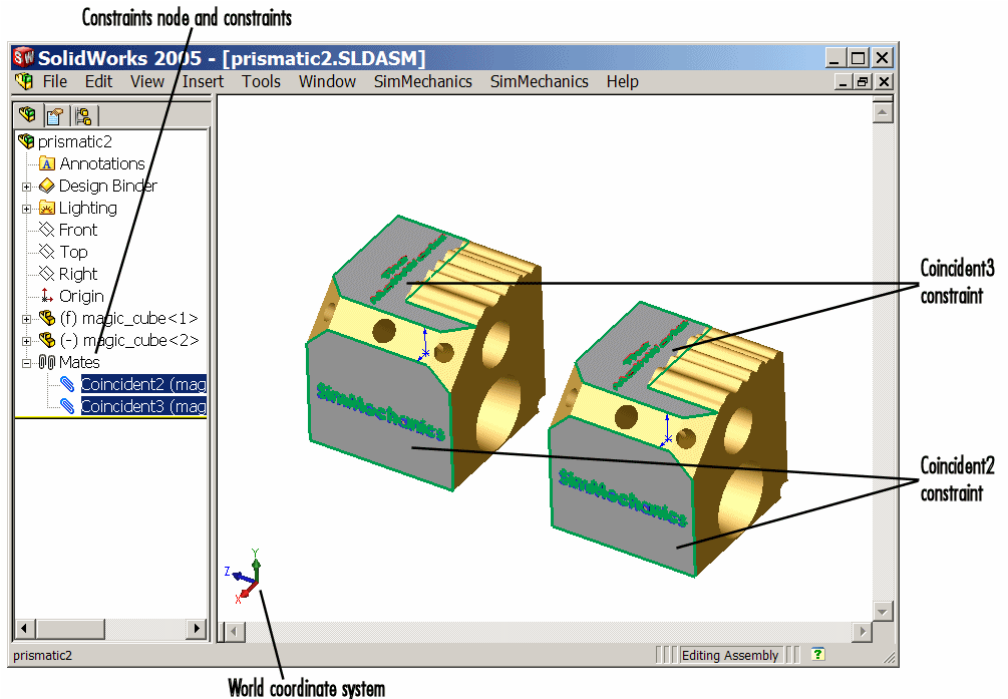
Two Planar Constraints on Two Cubes

Exporting the Assemblies and Generating SimMechanics Models

To create models from the assemblies,

1 Export the two assemblies into the XML files `prismatic1.xml` and `prismatic2.xml`.

2 Copy or move them to a MATLAB working directory. At the MATLAB command line, generate SimMechanics models using `import_physmod`.

In both models, the assemblies are translated into the `prismatic1` and `prismatic2` subsystems, respectively. Each subsystem has eight blocks. The Prismatic Joint represents the single translational DoF between the two cubes with one prismatic primitive along the *z*-axis.

Modeling a Revolute Joint

In the following assembly, the two cubes are constrained to have only a single rotational degree of freedom (DoF) between them. In the translated SimMechanics model, this single DoF is a revolute joint.

Viewing the Assembly

To see an assembly with one rotational DoF,

1 Open the assembly file `revolute.`*ASSEMBLYFILETYPE* and examine the CAD hierarchy.

2 Locate and expand the constraints (Mates) node. There are two constraints on the two cubes.

- Highlight the first constraint, Concentric1. The constraint geometry is highlighted in the assembly. This constraint allows the two cubes to slide along and rotate about the *z*-axis running through the center of the parallel and concentric upper holes of each cube.

- Highlight the second constraint, Coincident1. The constraint geometry is highlighted in the assembly. This constraint allows the two cubes to slide along the *x-y* plane, with the parallel sides sharing a common plane. It also allows the two cubes to rotate about the *z*-axis. The cubes are not

allowed to rotate about any other axis, or to translate perpendicular to the *x-y* plane.

These two constraints mean that the two cubes can only rotate about the *z*-axis orthogonal to the *x-y* plane, leaving the whole assembly with only one rotational DoF.

Exporting the Assembly and Generating the Model

Now generate a model based on this assembly.

1 Export the assembly as `revolute.xml`. Copy or move it to a MATLAB working directory.

2 At the MATLAB command line, generate a SimMechanics model using `import_physmod`.

The assembly is translated into an eight-block subsystem called `revolute`. The Revolute Joint represents the single rotational DoF between the two cubes with one revolute primitive about the *z*-axis.

Modeling an Inplane Joint

In the following assembly, the two cubes are constrained to have only two translational degrees of freedom (DoFs) between them. In the translated SimMechanics model, these two DoFs are two prismatic joints.

Viewing the Assembly

To see an assembly with two translational DoFs,

1 Open the assembly file inplane.*ASSEMBLYFILETYPE* and examine the CAD hierarchy.

2 Locate and expand the constraints (Mates) node. There are two constraints on the two cubes.

- Highlight the first constraint, Coincident2. The constraint geometry is highlighted in the assembly. This constraint allows the two cubes to slide along the *y-z* plane, with the two sides marked "SimMechanics" sharing a common plane. It also allows the two cubes to rotate about the

x-axis. The cubes are not allowed to rotate about any other axis, or to translate perpendicular to the *y-z* plane.

- Highlight the second constraint, Parallel1. The constraint geometry is highlighted in the assembly. This constraint allows the two cubes to slide parallel to the *x-z* plane, with the two sides marked "The MathWorks" parallel but not necessarily in the same plane. It also allows the two cubes to translate perpendicular to the *x-z* plane and to rotate about the *y*-axis. The cubes are not allowed to rotate about any other axis.

These two constraints mean that the two cubes can only slide in the *y-z* plane, leaving the whole assembly with only two translational DoFs.

Exporting the Assembly and Generating the Model

Now generate a model based on this assembly.

1 Export the assembly as `inplane.xml`. Copy or move it to a MATLAB working directory.

2 At the MATLAB command line, generate a SimMechanics model using `import_physmod`.

The assembly is translated into a subsystem, `inplane`, having eight blocks. The In-Plane Joint represents the two translational DoFs between the two cubes with two prismatic primitives, along the *y*-axis and the *z*-axis.

Modeling a Spherical-Spherical Massless Connector

In the following assembly, the two cubes are constrained to have six rotational degrees of freedom (DoFs) between them, represented by two spherical primitives. The spherical primitives pivot independently about two pivot points at a fixed relative distance. In the translated SimMechanics model, a spherical-spherical massless connector represents these six DoFs.

Viewing the Assembly

To see an assembly with three rotational DoFs separated from three other rotational DoFs,

1 Open the assembly file

spherical_spherical_massless_connector.*ASSEMBLYFILETYPE*

and examine the CAD hierarchy.

2 Locate and expand the constraints (Mates) node. There is one constraint on the two cubes.

Highlight this constraint, Distance1. The two spherical pivot points are highlighted as small red or green squares, one on each cube. These points are the endpoints of the rigid massless connector. The cubes can move such that the distance between these two points (the length of the massless connector) does not change. The constraint allows the two cubes to pivot independently about their connector endpoints.
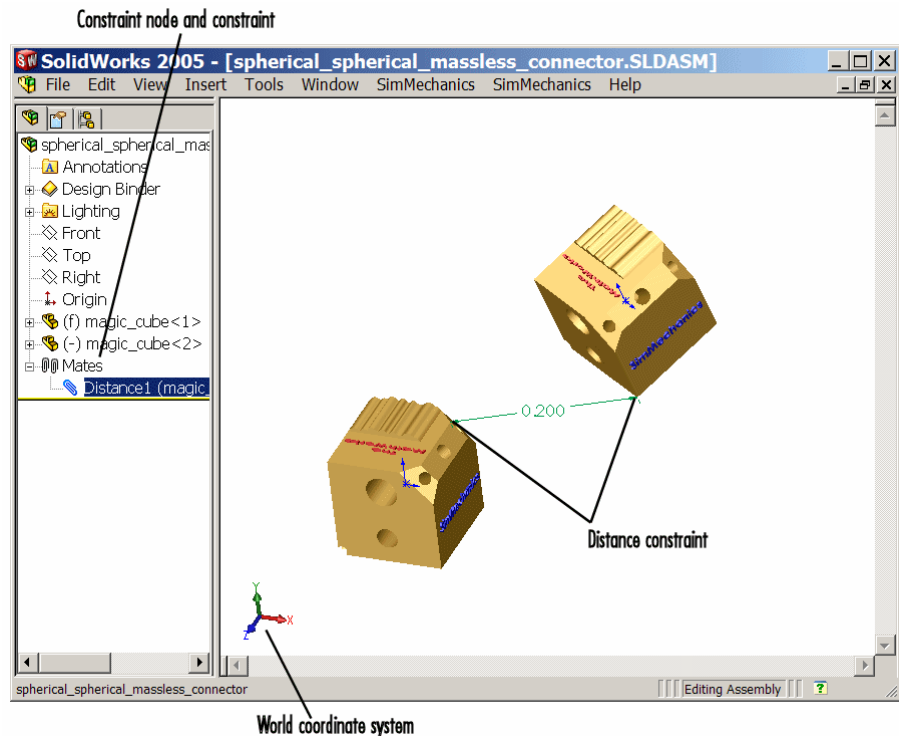
Distance Constraint on Two Cubes

Exporting the Assembly and Generating the Model

Now generate a model based on this assembly.

1 Export the assembly as

    spherical_spherical_massless_connector.xml

Copy or move it to a MATLAB working directory.

2 At the MATLAB command line, generate a SimMechanics model using
`import_physmod`.

The assembly is translated into an eight-block subsystem called
`spherical_spherical_massless_connector`, arranged in the common
structure described in "Block Structure of the Two-Part Models" on page 1-29.

The Spherical-Spherical massless connector Joint block represents the two
spherical primitives, each with three rotational DoFs, independently pivoting
at each end of the massless, rigid connector connecting the two cubes.

Creating a CAD-Based Robot Arm Model

The example of this section is based on a more complex CAD assembly, a robot arm. It includes multiple parts, multiple constraints, and a subassembly.

- "Viewing the Robot Arm Assembly" on page 1-41

- "Exporting the Robot Arm Assembly" on page 1-42

- "Generating and Completing the Robot Arm Model" on page 1-42

- "Simulating and Observing the Robot Arm Motion" on page 1-46

Locate the 11 CAD files for the robot arm are. They are

Filename	CAD Filetype
robot.*ASSEMBLYFILETYPE*	Assembly
grip.*ASSEMBLYFILETYPE*	Subassembly (flexible)
base.*PARTFILETYPE* forearm.*PARTFILETYPE* upperarm.*PARTFILETYPE* wrist.*PARTFILETYPE*	Parts (main assembly)
fingertips.*PARTFILETYPE* (twice) firstfingerlink.*PARTFILETYPE* firstfingerlinkL.*PARTFILETYPE* metacarpal.*PARTFILETYPE* secondfingerlink.*PARTFILETYPE* (twice)	Parts (subassembly)

Viewing the Robot Arm Assembly

Open the assembly file for the whole robot.
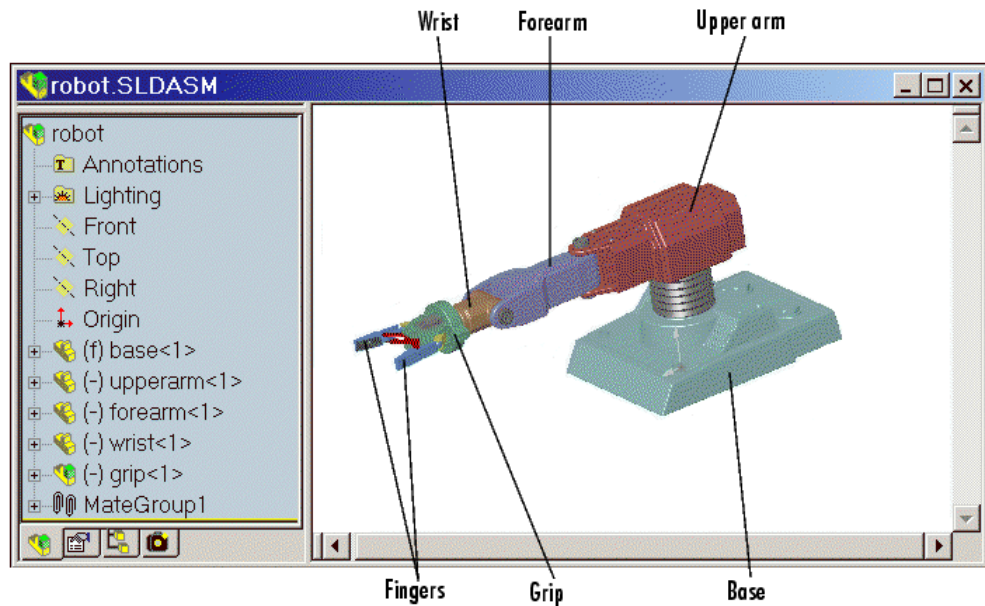
Robot Arm Assembly in a CAD Platform

Then examine the CAD hierarchy:

- Five of the part files are grouped into the subassembly grip. The subassembly uses two instances each of fingertips and secondfingerlink.

- The subassembly has its own group of 18 constraints, MateGroup1.

  Two constraints, Angle1 and Angle2, are not active. If they were, they would lock the grip fingers into the open position. Here, each grip finger can move separately.

- The other four part files are separate and grouped into the main assembly.

- The main assembly has its own MateGroup1, consisting of seven constraints.

The whole assembly has eight DoFs. The `grip` subassembly alone contains two, allowing each finger to open and close separately. The main assembly has six DoFs:

- The upper arm can move relative to the base by pitching, yawing, and rolling (three DoFs).

- The forearm can yaw relative to the upper arm (one DoF).

- The wrist can pitch relative to the forearm (one DoF).

- The grip can rotate about its symmetry axis (one DoF).

Exporting the Robot Arm Assembly

Apply any changes you want to the assembly configuration or settings. If you change the assembly or any subassemblies, you need to rebuild the assembly before exporting it to XML.

Now export the assembly into Physical Modeling XML. The XML file `robot.xml` appears in your working CAD directory.

Generating and Completing the Robot Arm Model

Now generate a model for a robot arm based on the file `robot.xml`. You can use this preconfigured demo file or export your own version of the XML file from the robot arm CAD assembly. In either case, copy or move the XML file to your MATLAB working directory.

See "Creating Models from Physical Modeling XML" on page 1-14 to learn more about how to generate models from XML files.

Generating the Initial Model

The preconfigured `robot.xml` file is located in the MATLAB directory

```
toolbox/physmod/mech/mechdemos/
```

1 Generate the model by entering

```
import_physmod('robot')
```

at the command line. The status bar appears and indicates the progress of model generation.

A model window, named **robot**, opens and is populated with blocks.

2 Save this initial body-joint model as robot, and note these properties:

- The whole robot arm assembly is contained in the subsystem `robot`.

- The top level of the assembly has 13 blocks and the `grip-1` subsystem.

- The `grip-1` subsystem has 18 blocks.

  The original robot arm assembly has eight DoFs, with two in the grip subassembly and six at the top level. These translate into eight DoFs in the SimMechanics model:

  - Six DoFs occur at the top level. These include the upper arm relative to the base, the forearm relative to the upper arm, the wrist relative to the forearm, and the grip relative to the wrist.

  - Two DoFs occur in the subsystem. These are the rotational DoFs of the two grip fingers.

    There are eight revolute primitives in the subsystem. They occur in two closed loops as two independent four-bar mechanisms. Each four-bar mechanism actually has only one independent DoF because each four-bar loop closes on itself. (See "Four Bar Mechanism" and "Counting Degrees of Freedom" in the SimMechanics User's Guide.)

Obtaining Simulink and Additional SimMechanics Blocks

To modify and extend the robot arm model, you need blocks from the SimMechanics and Simulink block libraries. Open these libraries by entering `mechlib` and `simulink`, respectively, at the command line.

You can also open the Simulink library from the MATLAB window menu or toolbar.

Editing the Bodies

Some of the bodies in the generated robot arm model are redundant. You can remove them without affecting the model's dynamics, as long as you properly reconnect the remaining blocks.

- At the top level, the SimMechanics_RootPart block is a zero-mass, zero-inertia Root Body. You can delete it, along with the connected Weld1 block, then reconnect the Root Ground to the base-1 block through the Weld block.

- In the grip-1 subsystem, you can delete the grip-1 (Root Body) block and the connected Weld block because they are unnecessary. You can also delete the associated body coordinate system on the metacarpal-1 Body block.

See the reference page for more details about the Body block.

Editing the Joints

The non-Weld Joint blocks, those that carry DoFs, are Revolute and Spherical Joints configured with the proper primitives to represent the original CAD assembly's DoFs.

- The first non-Weld Joint encountered as you move away from the Ground block is a Spherical, representing three DoFs.

- Each Revolute block contains a single revolute primitive, representing one rotational DoF.

Save this intermediate model as robot2.

Adding an Actuator and a Sensor

You can motion-actuate the wrist relative to the forearm.

1 Double-click the Revolute Joint that connects the forearm-1 and wrist-1 Body blocks. Change the **Number of sensor/actuator ports** to 2.

2 Click **OK**. Two new ports appear on the Joint.

3 From the SimMechanics Sensors & Actuators library, insert and attach a Joint Actuator and a Joint Sensor to these new ports.

4 Configure the Joint Actuator to accept motion signals. Be sure the angular units are deg (degrees).

5 From the Simulink library, insert a Sine Wave, a Mux, two Integrator blocks, and one Scope block. Connect them to the previous blocks as shown in the following figure. Rename the Scope block to Pitch Angle.

Consult the Simulink documentation for more about these Simulink blocks.
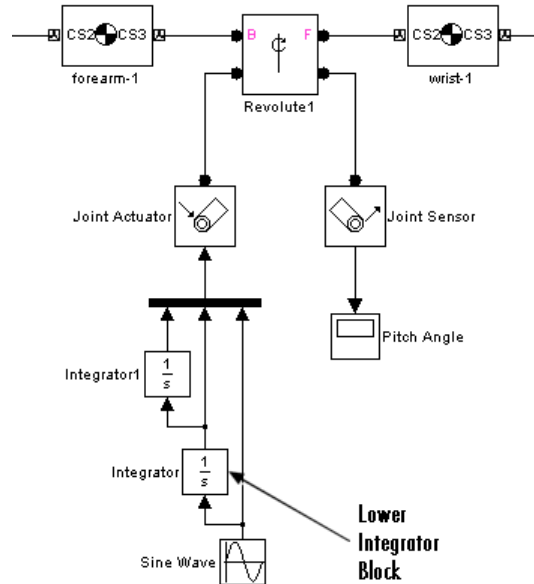
6 In the Sine Wave block, set the **Amplitude** to `60*pi*pi` and the **Frequency** to `60`. Leave all other defaults unchanged.

7 In the lower Integrator block, set **Initial condition** to `-60*pi`. Leave all other defaults unchanged.

Configuring Tolerances

The original robot arm CAD assembly requires looser tolerances than the SimMechanics defaults, and its motion can lead to singularities. To avoid simulation errors or slowdown, you need to reconfigure the assembly tolerances and constraint solver.

1 Open the Machine Environment block.

2 On the **Parameters** panel, reset the **Linear assembly tolerance** to 1e-2 m (meters) and the **Angular assembly tolerance** to 1e-1 rad (radians).

3 On the **Constraints** panel, select the **Use robust singularity handling** check box. Leave all other defaults. Click **OK**.

4 Resave your finished model as robot3.

Simulating and Observing the Robot Arm Motion

You can now run robot3 and examine its motion.

To use the motion sensor,

1 Double-click the Pitch Angle block to open a scope.

2 Click the **Start simulation** button. The scope plot displays a trace of the pitch angle motion.

To visualize the body motions,

1 From the **Simulation** menu, select **Configuration Parameters**, then the **SimMechanics** node.

2 Select **Display machines after updating diagram** and **Show animation during simulation**. Click **OK**.

3 Select **Update Diagram** from the **Edit** menu. The SimMechanics visualization window opens.

4 In the **SimMechanics** menu of the visualization window, select **Machine Display**, then **Ellipsoids**. The display now shows the robot arm's component bodies as ellipsoids.

5 Click the **Start** button. The simulation begins. Observe the robot arm motion in the SimMechanics window.

Modeling a Stewart Platform in CAD

This section introduces a complex computer-aided design (CAD) assembly that models the Stewart platform, a six-degree-of-freedom (DoF) mechanical system used for accurate positioning applications.

- "What the Stewart Platform Does" on page 1-47
- "Viewing the Stewart Platform Assembly" on page 1-48
- "Exporting the Stewart Platform Assembly" on page 1-49
- "Generating the Stewart Platform Model" on page 1-49
- "Visualizing the Stewart Platform Motion" on page 1-52

Note The Stewart platform assembly of this section is an advanced example of computer-aided design. You should work through the previous examples of this chapter before attempting to work with this assembly.

To learn more about the Stewart platform, see "Case Studies" in the SimMechanics User's Guide.

Locate the 45 CAD files for the Stewart platform. The master assembly file is

    stewart_platform.*ASSEMBLYFILETYPE*

What the Stewart Platform Does

The Stewart platform consists of two plates connected by six mobile and extensible legs. The lower or base plate is immobile. The upper or mobile plate has six degrees of freedom, three rotational and three translational. The platform is highly stable and easy to control.

The platform's six legs each have two parts, an upper and a lower leg, with a piston-like cylindrical DoF between each pair of parts. The legs are connected to the base plate and the top plate by universal joints at each end of each leg. (These universals are not just sets of abstract DoFs. Each also contains a spider-like body, while also having two DoFs.) The upper part of each leg can slide into and out of the lower leg, allowing each leg to be varied in length. The

position and orientation of the mobile platform (top plate) varies depending on the lengths to which the six legs are separately adjusted.

Once the top is connected to the legs, the entire Stewart platform assembly has 36 DoFs. Only six DoFs are *independent*, the same as the top plate would have if it were disconnected. You can think of these independent DoFs as the six adjustable leg lengths or as equivalent to the six DoFs of the mobile plate. . See "Counting Degrees of Freedom" in the "Modeling Mechanical Systems" chapter of the SimMechanics User's Guide.

Viewing the Stewart Platform Assembly

Open the master assembly file, stewart_platform.*ASSEMBLYFILETYPE*. Click the assembly and rotate it to view the top and bottom plates and the legs.
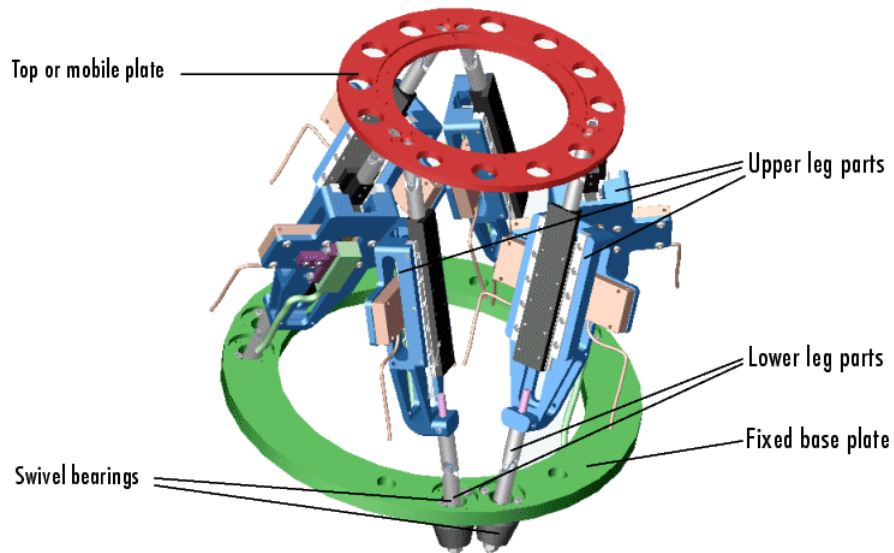
Stewart Platform CAD Assembly

The CAD hierarchy for the Stewart platform contains assemblies for the top and base plates, as well as assemblies for the six legs. All the constraints on the assembly parts are grouped into one group, containing 30 constraints. There are 448 component parts and 38 subassemblies, which you can open individually to examine the separate parts.

The base plate is about 24 centimeters (cm) in diameter, the top plate about 16.5 cm. When centered and oriented flat, the top plate is about 20 cm above the base. The assembly models the platform material as aluminum (about 2.7 grams per cubic cm).

Exporting the Stewart Platform Assembly

Apply any changes you want to the assembly configuration or settings. If you change the assembly or any subassemblies, you need to rebuild the assembly before exporting it to XML.

Now export the assembly into Physical Modeling XML. Because the assembly is so complex, the export process takes longer than it does for simpler assemblies. As the export proceeds, the translator highlights various parts and subassemblies. When the highlighting stops, the export is finished.

The exported model appears as the XML file `stewart_platform.xml` in your working CAD directory.

Generating the Stewart Platform Model

Now move or copy the `stewart_platform.xml` file into your working MATLAB directory.

At the MATLAB command line, enter

```
import_physmod('stewart_platform')
```

and wait for SimMechanics to complete the generation of the new Simulink model `stewart_platform`.

Inspecting the Generated Model and Counting Its DoFs

The entire Stewart platform model is contained in a subsystem, also called `stewart_platform`. This subsystem itself contains seven subsystems.
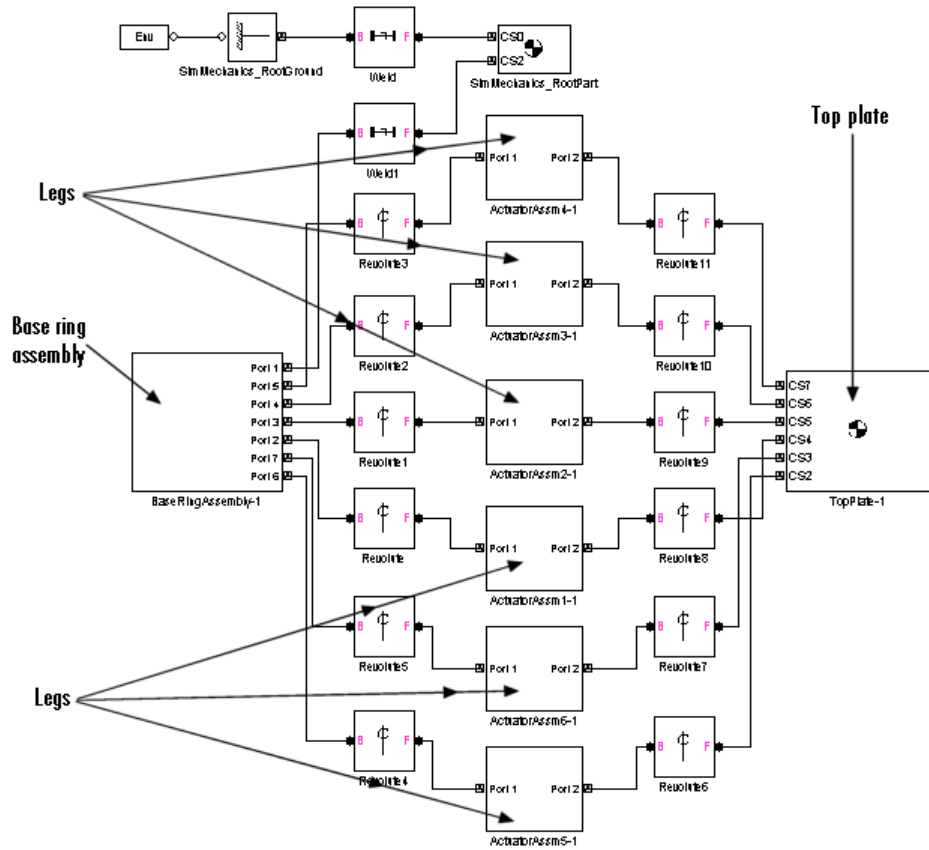
Stewart Platform Model: Base, Legs, and Top Plate

The subsystems correspond to the subassemblies of the original CAD assembly: the base plate and the six platform legs.

- The base plate subassembly BaseRingAssembly-1 contains six subassemblies, modeling a base swivel bearing for each leg.

- The six leg subassemblies, ActuatorAssm, model the upper and lower halves of each leg and represent part of their DoFs.

For each leg, there are six DoFs. Two pairs of revolutes associated with each leg represent the two universal joints connecting each leg to the top and base plates, respectively. Each of these universals has two DoFs.

- At the top level, there are two revolutes, one attached to either end of a leg subassembly, connecting each leg to the base and top plates, respectively.

- Within each leg subassembly, there are two other revolutes, each one connecting the leg to the top and base plates, respectively.

One of the revolutes inside the leg subassembly pairs with one of the revolutes outside the leg assembly to make up a two-DoF universal. These pairs occur twice on each leg, one connecting the leg to the top plate, the other connecting the leg to the base plate.

- Within each leg subassembly, there is one prismatic, representing the leg's freedom to expand or contract along its shaft.

- Within each swivel bearing subassembly, itself located within the base ring assembly, is another revolute representing each leg's freedom to rotate about its shaft.

Each leg has six DoFs. However, the constraints imposed by attaching each leg to fixed points on the base and top plates, respectively, reduce these to one independent DoF for each leg: the freedom to expand or contract along its shaft.

- The rotational DoFs associated with the universals at the attachment points are completely dependent on the leg's prismatic DoF.

- The rotational DoFs associated with the cylindricals in each leg are completely dependent on the universals at the top and bottom of each leg.

Deleting Unnecessary Bodies and Joints

The generated model contains a large number of redundant Root Weld and zero-mass Root Part blocks. You can delete these and not affect the model's dynamics, if you take care to reconnect the remaining bodies properly after deleting each Weld.

Adding Actuators and Sensors

If you want the motion of the platform to be controlled by something other than gravity, you need to add the appropriate Actuators to the model. To quantify the model's motion, you need to make precise measurements with Sensors. You can drive the actuators with external control signals to model an open-loop controller for the Stewart platform. If you introduce feedback from the sensors to the actuators, you can model a closed-loop controller.

You can find out more about using actuator and sensor blocks in "Modeling Mechanical Systems" in the SimMechanics User's Guide.

Visualizing the Stewart Platform Motion

Note Find more information about SimMechanics visualization in "Visualizing and Animating Machines" in the SimMechanics User's Guide.

Without any external forces acting, apart from gravity, the platform collapses under its own weight. You can verify this by running and visualizing your Stewart platform model.

1 From the **Simulation** menu, select **Configuration Parameters**. The **Configuration Parameters** dialog opens. Choose the **SimMechanics** node.

2 Select **Display machines after updating diagram** and **Show animation during simulation**. Click **Apply** or **OK**.

3 From the **Edit** menu, select **Update Diagram**. The SimMechanics visualization window opens with the SimMechanics controls. The window displays the Stewart platform in its initial position.

4 Start the simulation by clicking the **Start** button in the toolbar of either the visualization window or the model window.

The mobile plate falls under its own weight and reaches the base plate in about 0.2 seconds. Because there is nothing to stop the legs or the top plate, the platform continues to collapse: the mobile plate falls below the base plate, and the upper and lower parts of each leg come apart.
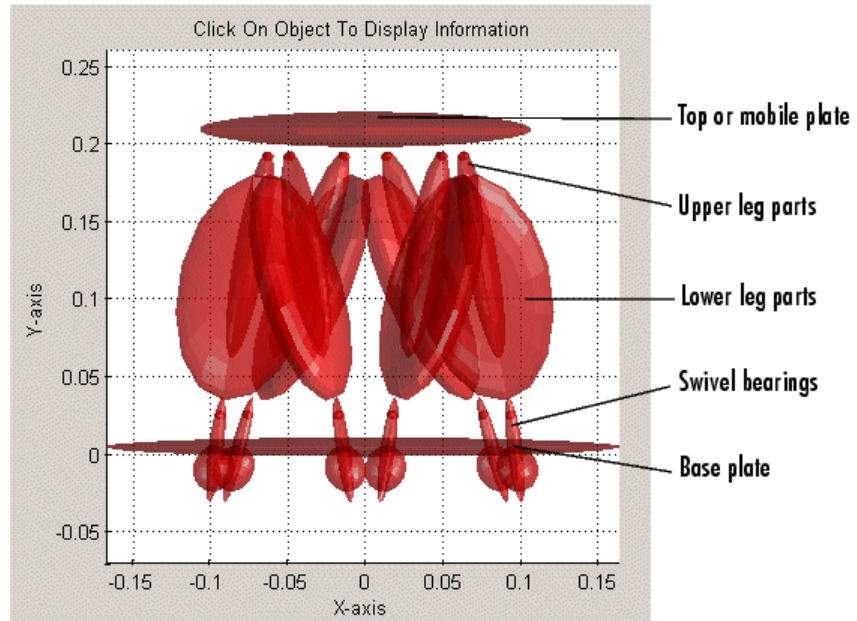
SimMechanics Visualization of the CAD-Based Stewart Platform

Index

A

assembly 1-6
    CAD 1-6
    See also computer-aided design

C

CAD. *See* computer-aided design
CAD-to-SimMechanics translator 1-3
computer-aided design (CAD)
    building assemblies 1-6
    editing models derived from 1-16
    exporting assemblies into XML 1-10
    generating CAD-based models 1-14
    getting translator help 1-12
    importing assemblies into
        SimMechanics 1-14
    installing CAD-to-SimMechanics
        translators 1-5
    models derived from 1-16
    troubleshooting assembly export errors 1-12
    troubleshooting models derived from 1-19

E

exporting CAD assemblies. *See* computer-aided
    design. *See* XML

I

importing CAD assemblies. *See* computer-aided
    design. *See* XML

S

Stewart platform
    computer-aided design 1-47

T

troubleshooting. *See* computer-aided design

X

XML (Extensible Markup Language) 1-3
    exporting from CAD 1-10
    importing into SimMechanics 1-14
    Physical Modeling format 1-3
    See also computer-aided design