

SimMechanics™

Getting Started Guide

R2012b

MATLAB®
& **SIMULINK®**

How to Contact MathWorks



www.mathworks.com
[comp.soft-sys.matlab](mailto:comp.soft-sys.matlab@mathworks.com)
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical Support



suggest@mathworks.com
bugs@mathworks.com
doc@mathworks.com
service@mathworks.com
info@mathworks.com

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
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™ Getting Started Guide

© COPYRIGHT 2002 - 2012 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

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

Revision History

September 2012 Online only

New for Version 4.1 (Release R2012b)

Getting Started

1 Introduction to SimMechanics Software

Product Description	1-2
Key Features	1-2
Required and Related Products	1-3
Required Products	1-3
Related Products	1-4
Modeling with First and Second Generation Technologies	1-5
SimMechanics Block Libraries	1-9
Body Elements	1-9
Constraints	1-9
Forces and Torques	1-9
Frames and Transforms	1-10
Joints	1-10
Utilities	1-10
Essential SimMechanics Blocks	1-11
World Frame	1-11
Mechanism Configuration	1-11
Solver Configuration	1-12
Multibody Modeling Workflow	1-13
Model Rigid Bodies	1-13
Assemble Multibody Model	1-17
Add Forces, Torques, and Motion Sensors	1-20
Simulate and Analyze model	1-24

Model an Aiming Mechanism	1-27
Modeling Steps	1-28
Model Files	1-29
Open Model	1-29
Identify Rigid Bodies and Joints	1-30
Model Rigid Body Subsystems	1-31
Connect Rigid Body Subsystems with Joint Blocks	1-35
Troubleshoot Assembly Error	1-36

Getting Started

Chapter 1, Introduction to
SimMechanics Software (p. 1-1)

Description of SimMechanics™
software

Introduction to SimMechanics Software

- “Product Description” on page 1-2
- “Required and Related Products” on page 1-3
- “Modeling with First and Second Generation Technologies” on page 1-5
- “SimMechanics Block Libraries” on page 1-9
- “Essential SimMechanics Blocks” on page 1-11
- “Multibody Modeling Workflow” on page 1-13
- “Model an Aiming Mechanism” on page 1-27

Product Description

Model and simulate multibody mechanical systems

SimMechanics provides a multibody simulation environment for 3D mechanical systems, such as robots, vehicle suspensions, construction equipment, and aircraft landing gear. You model the multibody system using blocks representing bodies, joints, constraints, and force elements, and then SimMechanics formulates and solves the equations of motion for the complete mechanical system. Models from CAD systems, including mass, inertia, joint, constraint, and 3D geometry, can be imported into SimMechanics. An automatically generated 3D animation lets you visualize the system dynamics.

You can parameterize your models using MATLAB® variables and expressions, and design control systems for your multibody system in Simulink®. You can add electrical, hydraulic, pneumatic, and other components to your mechanical model using Simscape™ and test them all in a single simulation environment. To deploy your models to other simulation environments, including hardware-in-the-loop (HIL) systems, SimMechanics supports C-code generation (with Simulink Coder™).

Key Features

- Blocks and modeling constructs for simulating and analyzing 3D mechanical systems in Simulink
- Rigid body definition using standard geometry and custom extrusions defined in MATLAB
- Automatic calculation of mass and inertia tensor
- Simulation modes for analyzing motion and calculating forces
- Visualization and animation of multibody system dynamics with 3D geometry
- SimMechanics Link utility, providing an interface to Pro/ENGINEER®, SolidWorks®, and Autodesk Inventor, and an API for interfacing with other CAD platforms
- Support for C-code generation (with Simulink Coder)

Required and Related Products

In this section...
“Required Products” on page 1-3
“Related Products” on page 1-4

Required Products

To use SimMechanics software, you must have installed current versions of the following products:

- MATLAB
- Simulink
- Simscape

SimMechanics Visualization Requirements

SimMechanics visualization requires Silicon Graphics OpenGL® graphics support on your system to display and animate mechanical systems.

You can improve your speed and graphics resolution by adding a graphics accelerator hardware card to your system. Animation of simulations is sensitive to central processor and graphics card speed and memory. Experiment with graphics hardware and system settings to find a reasonable compromise between quality and speed for your system.

Support for Recorded Animations

You can record simulation animations in Microsoft Audio Video Interleave® (AVI) format using SimMechanics visualization. Animations are recorded as compressed AVI files with the default Motion JPEG codec. This codec is available on most operating systems.

To play back AVI files, you need an AVI-compatible media application. MATLAB has an internal movie player compatible with AVI. You can also use an external AVI-compatible player.

Note SimMechanics visualization does not support uncompressed AVI recording.

Related Products

You can extend the capability of SimMechanics using other physical modeling products found in the Simscape family. Each physical modeling product gives you a set of block libraries with which you can model common components found in industry and academia: rigid bodies, gears, valves, solenoids, etc.

With the physical modeling products, you can model not only mechanical systems, but also electrical, hydraulic, and power systems. You can model each system separately, and then integrate the systems into a single multiphysics model where you can analyze combined system performance.

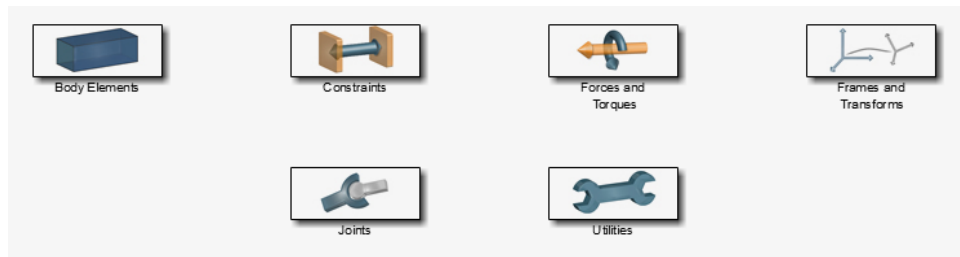
Physical Modeling Product Family

The physical modeling family includes five products:

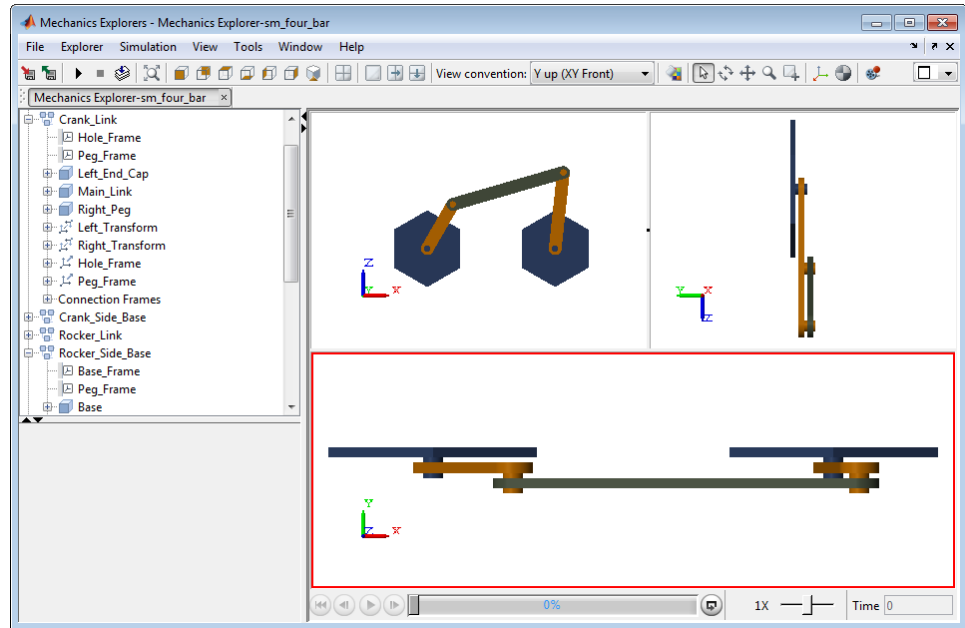
- SimDriveline™, for modeling and simulating drivetrain systems
- SimElectronics®, for modeling and simulating electronic systems
- SimHydraulics®, for modeling and simulating hydraulic systems
- SimMechanics, for modeling and simulating three-dimensional mechanical systems
- SimPowerSystems™, for modeling and simulating electrical power systems

Modeling with First and Second Generation Technologies

SimMechanics software contains two modeling technologies, first- and second-generation. Second-generation technology includes a new block library, computational engine, and visualization utility. First-generation technology includes libraries and computational engine in all releases prior to R2012a. The following figure shows the new second-generation library.

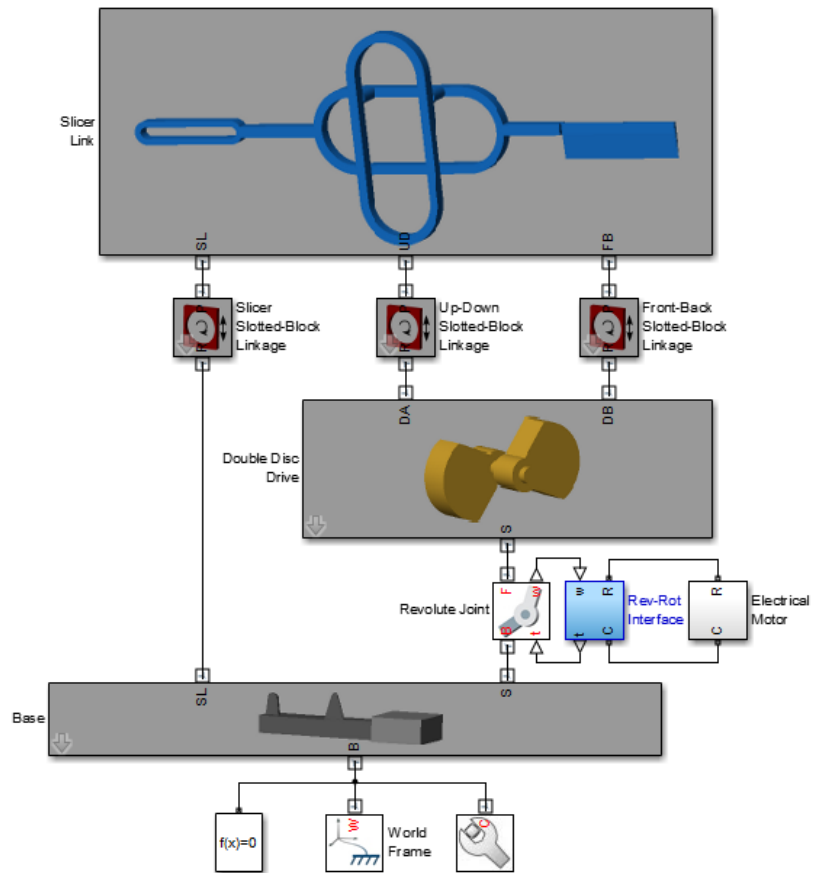


The new visualization utility is called Mechanics Explorer. The utility displays realistic 3-D geometries in static and dynamic modes. You can rotate, pan, and zoom the display to customize the view before or during simulation.

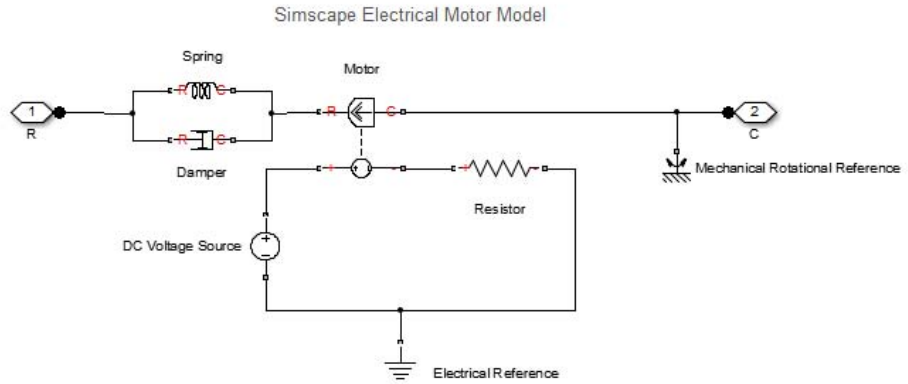


First-generation blocks remain functional. You can create and simulate new models built with first-generation blocks. Any first-generation models you already have also remain functional. You can modify existing first-generation models, removing or adding new first-generation blocks, editing block parameters, and setting model simulation preferences.

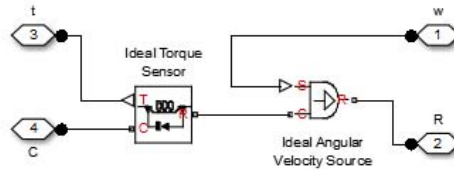
Second-generation blocks are built directly on the Simscape language. You can connect SimMechanics blocks directly to other Simscape blocks and build multiphysics models that include mechanical, electrical, and hydraulic domains. Model `sm_ssci_bread_slicer` provides an example of a multiphysics model. The model integrates mechanical and electrical domains, using Simscape Foundation and SimMechanics Second Generation blocks. The following figure shows the model block diagram.



In the model, subsystems Rev-Rot Interface and Electrical Motor are Simscape subsystems. The following figure shows the subsystem block diagrams.



Simscape Rev-Rot Interface Subsystem



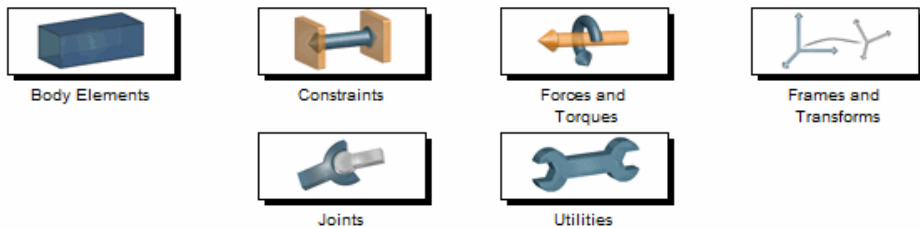
First- and second-generation blocks are based directly on the Simulink and Simscape platforms, respectively. First-generation blocks use dimensionless Simulink signals. Second-generation blocks use Simscape signals that contain physical units.

You can interconnect first- and second-generation blocks. The connection is indirect and requires a special interface block. Interface blocks include PS-Simulink Converter and Simulink-PS Converter blocks found in the Simscape Utilities library. The blocks convert between Simscape physical signals (PS) that second-generation blocks use and Simulink signals that first-generation blocks use.

SimMechanics Block Libraries

SimMechanics contains two block libraries: First Generation and Second Generation. This section is about the Second Generation block library.

The SimMechanics Second Generation block library contains six sublibraries. With the six sublibraries you can define every element in a mechanical model: bodies, joints and constraints, forces and torques, and more.



SimMechanics™ Product Library

Body Elements

The Body Elements library gives you the basic blocks to model a rigid body, including blocks for specifying geometry, inertia, and graphic properties. You can model rigid bodies with simple predefined, solid shapes and colors, or with custom geometry and inertia. You can also rigidly connect multiple solid shapes to form compound rigid bodies.

Constraints

The Constraints library provides blocks to restrict the relative motion between two body frames. You can constrain two body frames to maintain a specified distance or a specified angle.

Forces and Torques

The Forces and Torques library provides blocks to actuate and otherwise influence a mechanical model. You can apply different forces and torques to model the different interactions between components, or between a component and the external environment.

Frames and Transforms

The Frames and Transforms library provides blocks that allow you to define and measure spatial relationships between different frames in your model.

Joints

The Joints library provides blocks that connect bodies to one another in kinematic relationships representing different degrees of freedom between two body frames. These are time-varying relationships that restrict translation and rotation between the bodies to a specified set of axes.

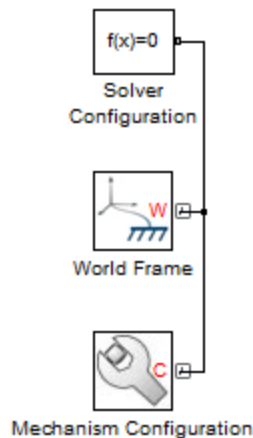
Utilities

The Utilities library contains blocks to configure solver and simulation parameters in the SimMechanics environment.

Essential SimMechanics Blocks

Most SimMechanics models use three essential blocks: World Frame, Mechanism Configuration, and Solver Configuration.

To begin building a SimMechanics model, drag and connect these three blocks in a new Simulink window. When you enter `smnew` at the MATLAB command line, SimMechanics automatically inserts and connects the blocks for you in a new Simulink window. The following figure shows the three essential blocks connected by a branched connection line.



World Frame

World Frame provides a unique inertial frame in a model, against which all other frames are referenced. Most SimMechanics models employ one World Frame block. For a non-inertial reference frame, you can use block Reference Frame instead. Both frame blocks belongs to block library **Frames and Transforms**.

Mechanism Configuration

Mechanism Configuration provides a gravity vector and a numerical linearization parameter. Each SimMechanics model can have multiple

Mechanism Configuration blocks, but only one is allowed per machine. This block belongs to block library **Utilities**.

Solver Configuration

Solver Configuration provides solver settings for numerical simulation of a kinematic model. Each SimMechanics model can have multiple Solver Configuration blocks, but only one per machine. This block belongs to Simscape block library **Utilities**.

Multibody Modeling Workflow

In this section...

“Model Rigid Bodies” on page 1-13

“Assemble Multibody Model” on page 1-17

“Add Forces, Torques, and Motion Sensors” on page 1-20

“Simulate and Analyze model” on page 1-24

The multibody modeling process contains a set of well-defined, sequential steps. Before you begin, define the intent of the model. Then, select a modeling approach and define the model requirements.

SimMechanics supports two modeling approaches:

- Manually build model — most common modeling approach. The following sections outline the manual approach.
- Automatically generate model from CAD assembly — for users wishing to simulate a CAD assembly in a physical modeling environment. Approach requires access to supported CAD software and assembly files. See “CAD Translation”.

Model Rigid Bodies

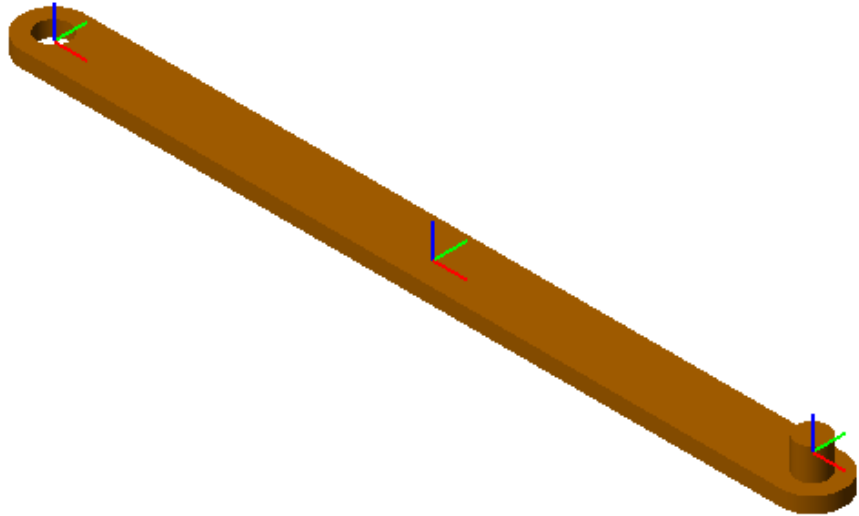
Rigid bodies are the building blocks of a multibody system. Before you can connect rigid bodies with joints and constraints, you must model the rigid bodies. You can create two types of rigid bodies:

- Simple — Use a single Solid block to model the rigid body. Simple rigid bodies are quicker to model. Use simple rigid bodies to quickly produce proof-of-concept models, before you create more detailed rigid body subsystems.

A simple cylinder is a simple rigid body.



- Compound — Rigidly connect multiple Solid blocks to model the rigid body. Compound rigid bodies require more time to model, but provide the most accurate representation of a rigid body that you can manually build.
A detailed binary link is a compound rigid body.



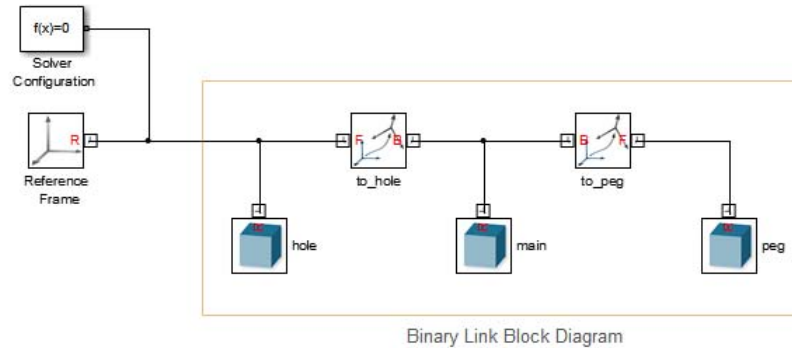
A complete rigid body subsystem contains the following:

- Frames — Construct with position and orientation in space.
- Physical properties — Geometry, inertia, and graphic parameters of a solid.

Add Frames

To add frames to a rigid body subsystem, you use the Rigid Transform block. The block rotates and/or translates a base frame to obtain a follower frame. A chain of Rigid Transform blocks represents a chain of frames. Connect the reference frame port of a Solid block to a frame port or line to give the frame a position and orientation in space. See “Frame Connection Rules”.

The binary link subsystem contains three Solid blocks rigidly connected with two Rigid Transform blocks. Each Rigid Transform block translates a local reference frame, located at the center of the link, by an amount equal to half the link length along the length axis. The two end frames serve as joint connection frames.



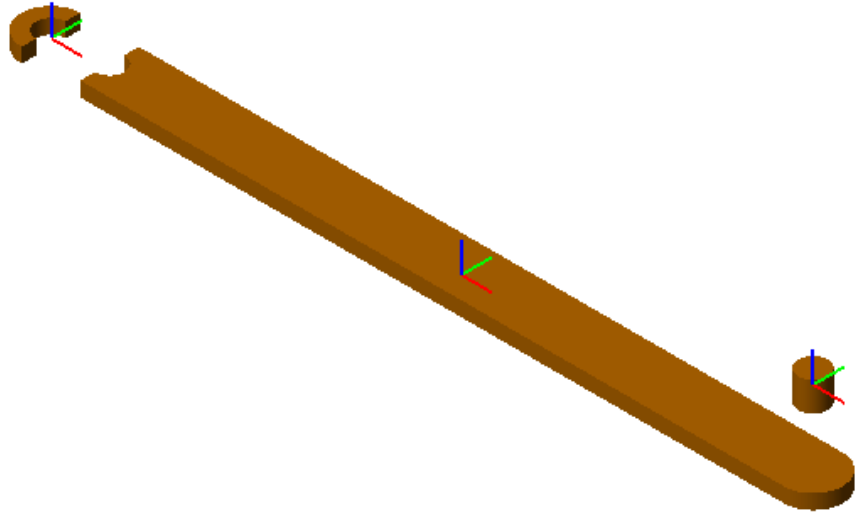
Add Geometry, Inertia, and Graphic Properties

To add geometry, inertia, and graphic properties to a rigid body, you use the Solid block. In the dialog box of the block, you can specify the numerical values for each parameter set:

- Geometry — 3-D shape of the rigid body. Required for assembly, visualization and simulation.
- Inertia — Mass or density and inertia tensor. Required for simulation.
- Graphic — Color of the rigid body. Required for visualization.

In the binary link subsystem, three Solid blocks provide the physical properties of three elementary pieces:

- Peg — Cylinder shape
- Main body — General Extrusion shape
- Hole Section — general Extrusion shape



In the Inertia menu of the Solid dialog box, you can specify the density if the link material — e.g. 2.70 g/cm^3 for Aluminum. SimMechanics provides the option to automatically calculate the moments and products of inertia from geometry. To provide a uniform density to the binary link, the three Solid blocks contain the same value in the **Density** field.

In the Graphic menu of the Solid dialog box, you can specify the color of the link material. In this example, the color is orange with RGB vector $[0.8 \ 0.45 \ 0]$. To provide a single color to the binary link, the three Solid blocks contain the same RGB vector in the **Color** parameter.

Assemble Multibody Model

Once you have created the rigid body subsystems, you can assemble the subsystems into a mechanism or machine. In a mechanism or machine, joints and constraints provide well-defined degrees of freedom that constrain rigid body motion. Two types of motion are possible:

- Rotation about one or more axes
- Translation along one or more axes

Connect Joints

To assemble rigid body subsystems into complete multibody systems, connect the with joints. SimMechanics provides a library with fourteen joint blocks. Each block provides a different number and/or type of mechanical degrees of freedom. Exactly two rigid body frames connect to each joint, with each frame belonging to a different rigid body.

Joints are composed of joint primitives — elementary joints that you can combine to create more complex joints. SimMechanics contains three joint primitives:

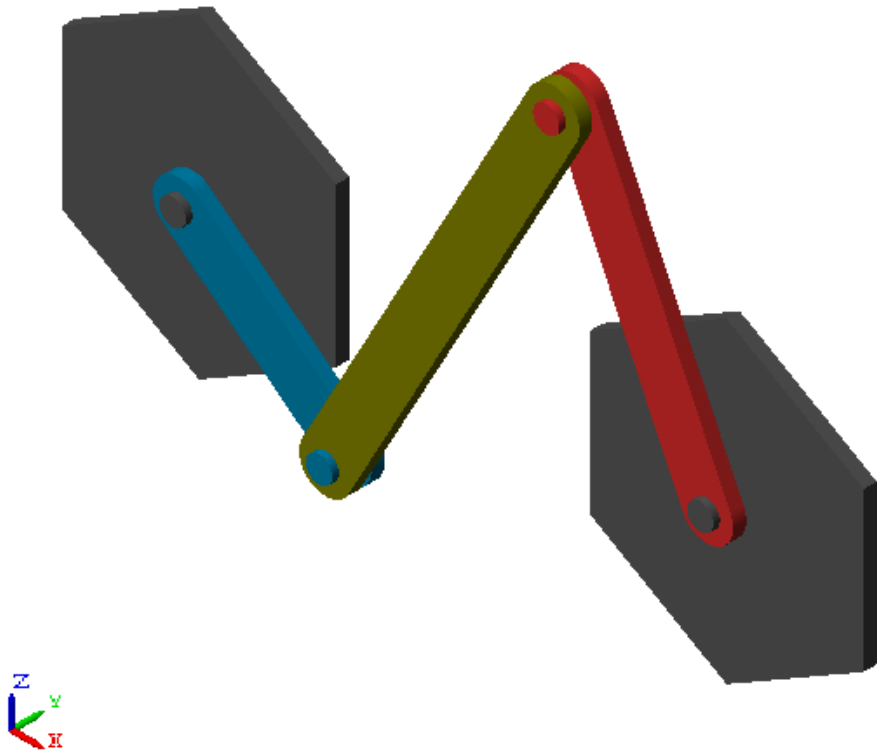
- Prismatic — one translational degree of freedom along an axis
- Revolute — one rotational degree of freedom about an axis
- Spherical — three concurrent degrees of freedom about three axes

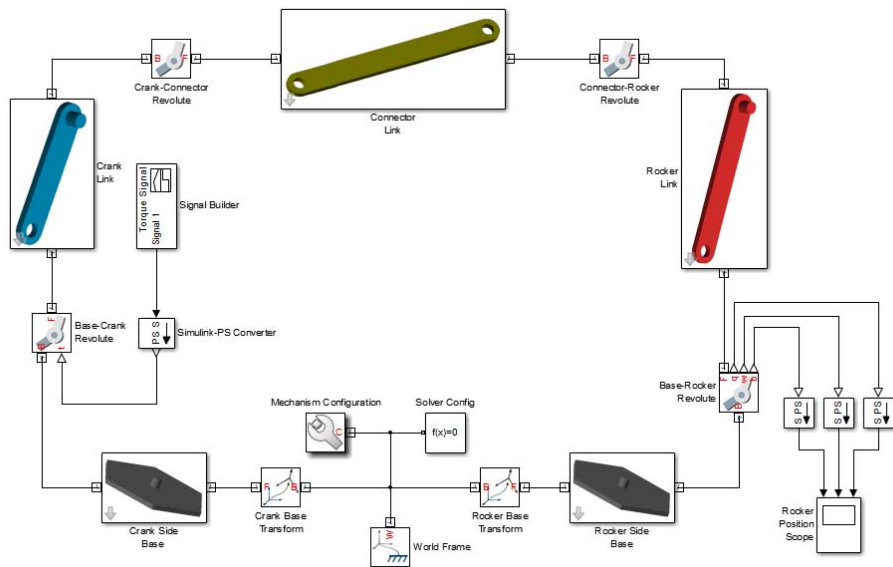
The following table identifies the joint primitives associated with each joint.

		Prismatic Primitives			
		0	1	2	3
Revolute Primitives	0	Weld Joint	Prismatic Joint	Rectangular Joint	Cartesian Joint
	1	Revolute Joint	Cylindrical Joint	Planar Joint	—
	2	Universal Joint	—	—	—
	3	Gimbal Joint	Bearing Joint	—	6-DOF Joint
Spherical Primitives	1	Spherical Joint	Telescoping Joint	—	Bushing Joint

The four-bar system contains four rigid bodies connected with four revolute joints. The block diagram for this model contains four rigid body subsystems

connected with four Revolute Joint blocks. The following figures show the Mechanics Explorer display of a four-bar model and the associated block diagram.





Specify Joint State Targets

To specify the initial configuration of a multibody assembly, use the **State Targets** menu of the joint dialog box. You can specify values for the initial joint position and velocity, and assign different priorities for the state targets. SimMechanics attempts to meet the state target. If the state targets make assembly impossible, SimMechanics tries to use an approximate value that allows assembly.

Add Forces, Torques, and Motion Sensors

You can add force and torques to, or between, rigid body frames. The following table summarizes the force/torque types that you can add to a model. The **Procedure** column identifies the steps required to add each force/torque type to a model.

Force/Torque Type	Description	Procedure
Joint actuation	Force or torque between base and follower frames along or about joint primitive axis	<ol style="list-style-type: none"> 1 In the Actuation menu of the joint dialog box, select the desired actuation mode. 2 Connect a physical signal line to each exposed physical signal port.
Joint internal mechanics	Spring and damper force or torque between base and follower frames along or about joint primitive axis	<ol style="list-style-type: none"> 1 In the Internal Mechanics menu of joint dialog box, enter the spring and damper parameters for the joint.
Rigid body actuation	Force or torque acting on a rigid body frame not connected to a joint	<ol style="list-style-type: none"> 1 Connect an External Force and Torque block to the rigid body frame you wish to actuate. 2 In the dialog box of the block, select the force and torque components you wish to specify and the frame in which you wish to resolve the components. 3 Connect a physical signal line to each

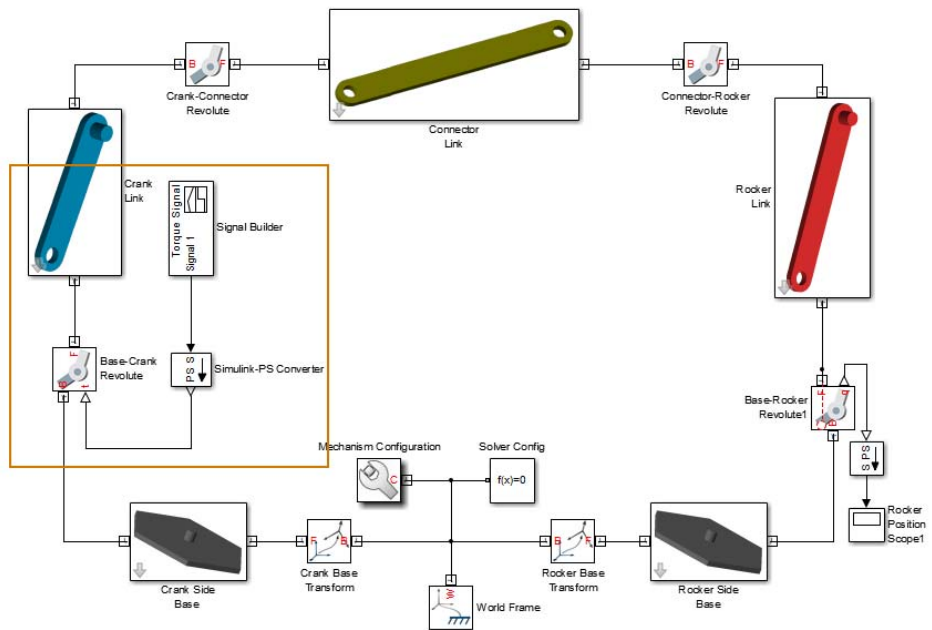
Force/Torque Type	Description	Procedure
		exposed physical signal port.
Rigid Body Internal Forces	Force acting between two rigid body frames, one frame per rigid body	<ol style="list-style-type: none"> 1 Connect the Inverse Square Law Force or Spring and Damper Force block between the two rigid body frames. 2 In the dialog box of the block, enter the force constants.

You can also add mechanical sensing between frames. Force sensing is limited Inverse Square Law Force or Spring and Damper Force blocks. The following table summarizes the types of motion and force sensing you can add to a model. The **Procedure** column identifies the steps required to add each sensing type to a model.

Sensing	Description	Procedure
Joint Motion	Sensing of position, velocity, and acceleration parameters of joint follower frame with respect to base frame	<ol style="list-style-type: none"> 1 In the Sensing menu of the joint dialog box, select the desired motion parameter. 2 Connect a physical signal to each exposed physical signal port.
Rigid Body Motion	Sensing of position, velocity, and acceleration parameters of one rigid body frame with	<ol style="list-style-type: none"> 1 Connect the Transform Sensor block to two rigid body frames.

Sensing	Description	Procedure
	respect to any other frame in a model	<p>2 In the dialog box of the Transform Sensor block, select the motion parameters you wish to sense.</p> <p>3 Connect a physical signal to each exposed physical signal port.</p>
Force Input	Sensing of force magnitude in Inverse Square Law Force and Spring and Damper Force blocks	<p>1 In the dialog box of the force block, select Sense Force.</p> <p>2 Connect physical signal to the exposed fm physical signal port.</p>

The following figure shows an application of joint actuation inputs. The Base-Crank joint of a four-bar model receives a torque input indirectly from the Simulink block Signal Builder. The Simulink-PS Converter block converts the Simulink signal into a physical signal compatible with SimMechanics blocks.

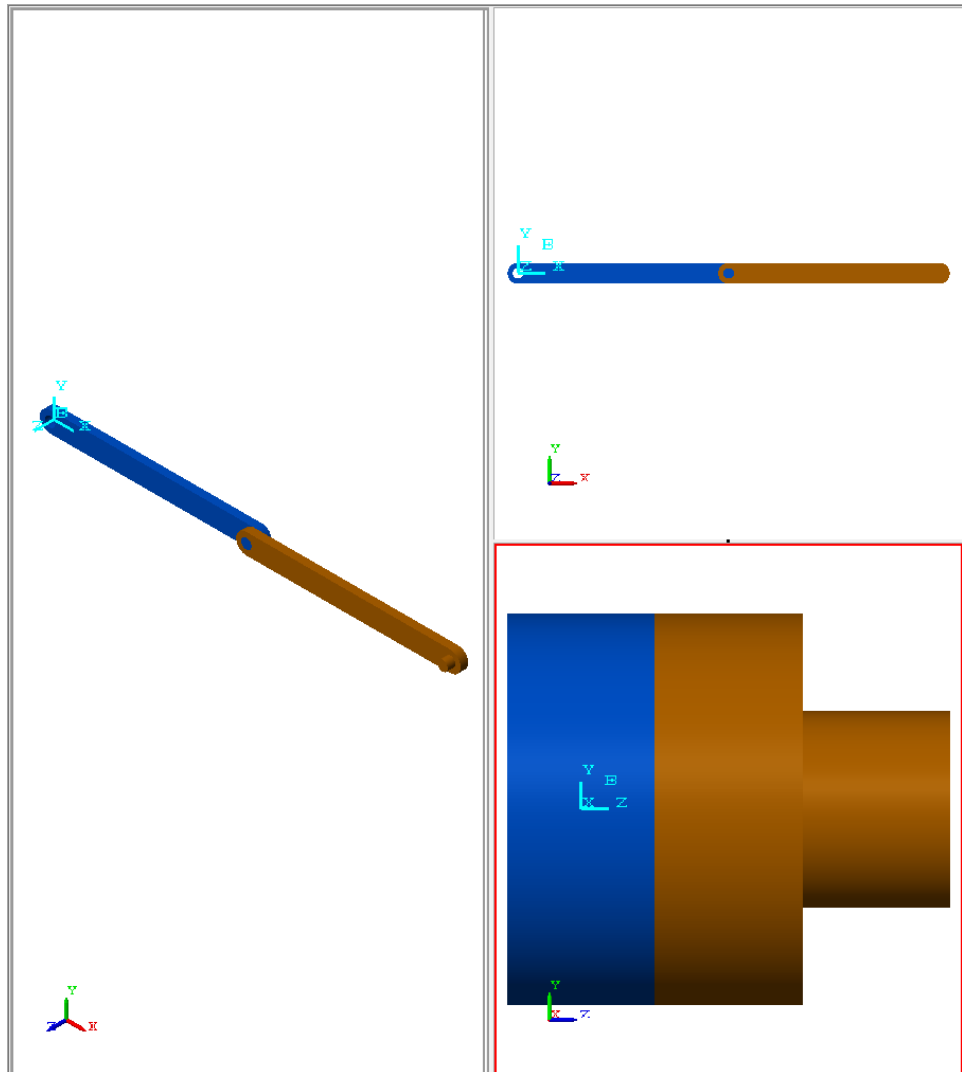


Simulate and Analyze model

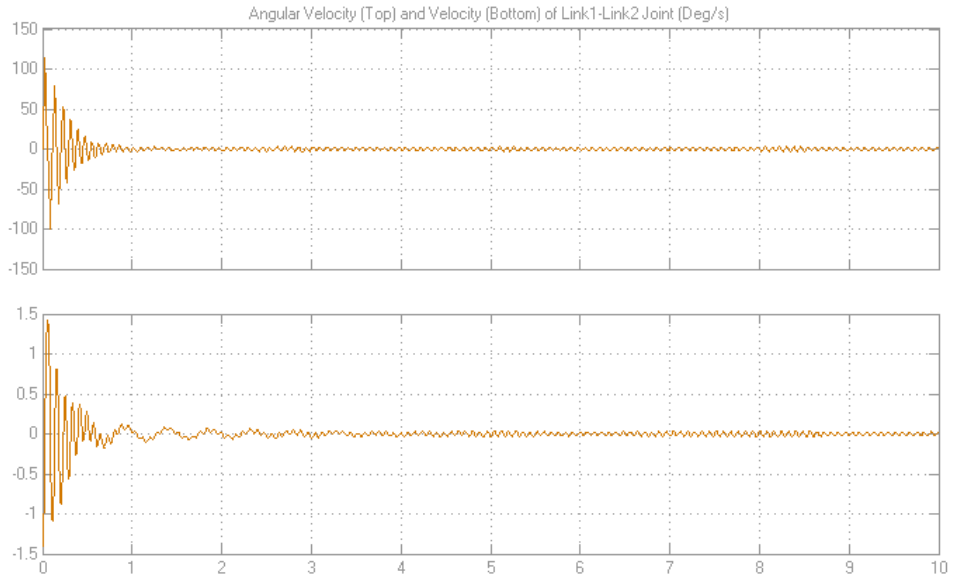
Run the simulation to examine the dynamic behavior of the model. Simulink provides a set of model simulation options that you can adjust for your application. Before running simulation:

- Check the choice of solver. SimMechanics models require a continuous solver. Variable-step solvers provide the best performance. Use `ode45` (Dormand-Prince) as a first try for most models.
- Check the solver step sizes and tolerances. Smaller values for both parameters produce more accurate results at the cost of increased simulation time.
- Check the simulation time span. To let a simulation run indefinitely, specify `inf` in the **Stop time** field.

The Mechanics Explorer utility provides 3-D visualization for SimMechanics models. By default, the utility opens the first time you update a model, and refreshes for all subsequent model updates. The following figure shows the Mechanics Explorer display of a double-pendulum model.



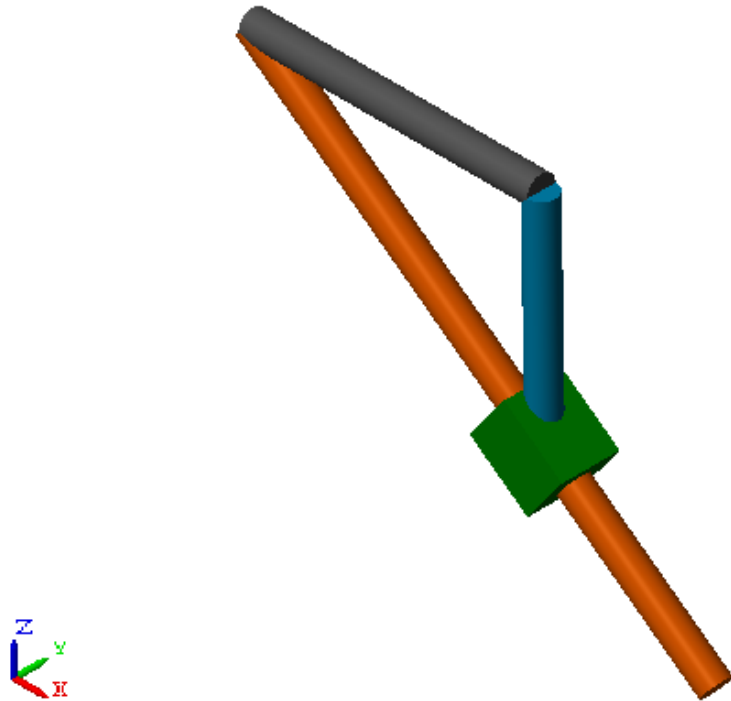
During simulation, you can open any scope blocks in the model to examine the dynamic behavior of the model. The following figure shows the angular position and velocity of the bottom frame of a damped double-pendulum model as functions of time.



Model an Aiming Mechanism

In this section...
“Modeling Steps” on page 1-28
“Model Files” on page 1-29
“Open Model” on page 1-29
“Identify Rigid Bodies and Joints” on page 1-30
“Model Rigid Body Subsystems” on page 1-31
“Connect Rigid Body Subsystems with Joint Blocks” on page 1-35
“Troubleshoot Assembly Error” on page 1-36

In this tutorial, you model a double-crank aiming mechanism. The mechanism contains four rigid bodies that connect with four joints. The tutorial begins with the identification of rigid bodies and joints. You model the rigid bodies and then connect the rigid bodies with joints. The modeling approach is iterative. In the first iteration, a simple shape represents each rigid body. When assembly and simulation are successful, a second iteration adds detail to each rigid body. The tutorial progression illustrates the recommended workflow for SimMechanics models.



Modeling Steps

The tutorial progresses in the following order:

- 1** “Identify Rigid Bodies and Joints” on page 1-30
- 2** “Model Rigid Body Subsystems” on page 1-31
- 3** “Connect Rigid Body Subsystems with Joint Blocks” on page 1-35
- 4** “Troubleshoot Assembly Error” on page 1-36

Model Files

The tutorial uses model `sm_dcrankaim_assembly_with_error`. The model is present in your SimMechanics installation. You can access model files in the following directory:

```
matlabroot\toolbox\physmod\sm\smdemos\double_crank_aiming
```

The model represents the complete aiming mechanism, with one intentional error. The tutorial uses the error to illustrate troubleshooting tools and techniques you can use in your own models. Simple shapes represent the four rigid bodies. For a rigid body example, you can open three rigid body models. The following table lists all models relevant to this tutorial. To open one of the models, at the MATLAB command line, enter the model name. For example:

```
sm_dcrankaim_approx_body_A
```

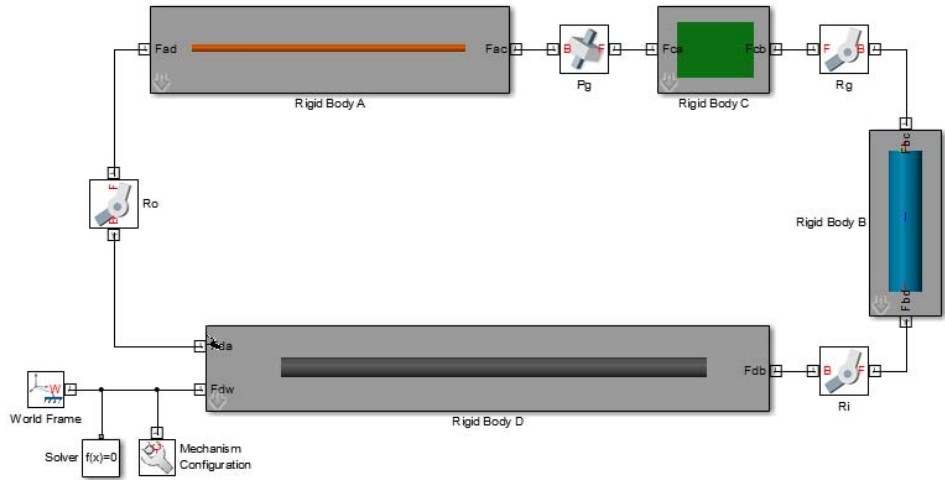
Model File	Description
<code>sm_dcrankaim_assembly_with_error</code>	Multibody model
<code>sm_dcrankaim_approx_body_A</code>	Rigid body model
<code>sm_dcrankaim_approx_body_B</code>	
<code>sm_dcrankaim_approx_body_C</code>	
<code>sm_dcrankaim_cplx_body_A</code>	

Open Model

Before you begin the tutorial, open model `sm_dcrankaim_assembly_with_error`. At the MATLAB command line, enter the model name:

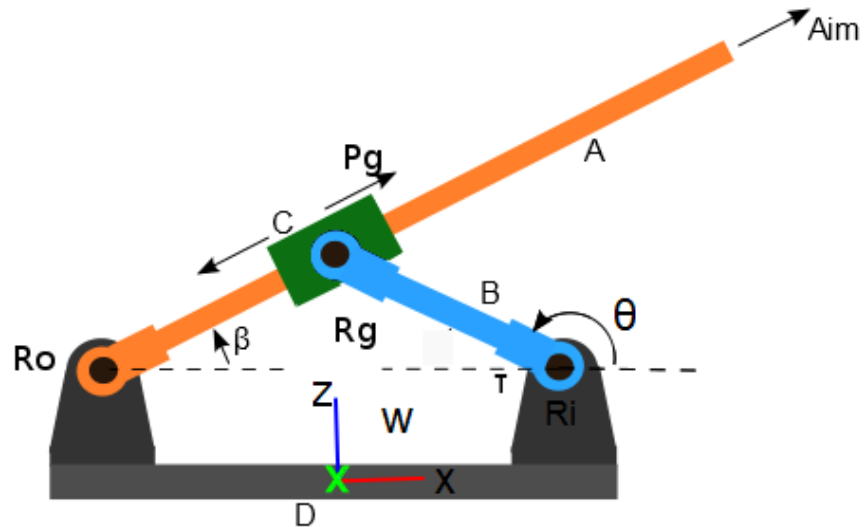
```
sm_dcrankaim_assembly_with_error
```

The model opens in a new Simulink Editor window.



Identify Rigid Bodies and Joints

The following figure shows a schematic of the double-crank aiming mechanism. Labels **A-D** identify the four rigid bodies. Labels **Ro**, **Rg**, **Ri**, and **Pg** identify the four joints. Each joint connects exactly two rigid bodies.



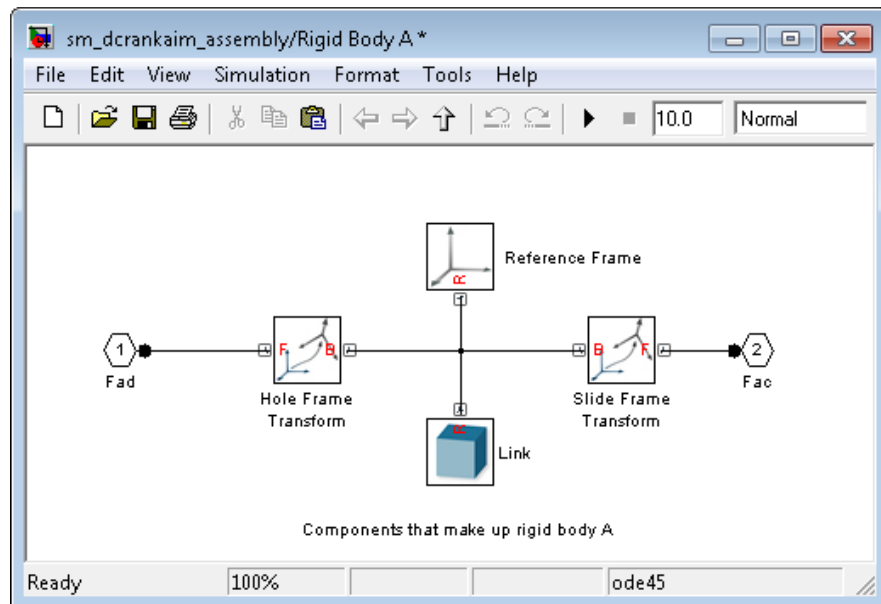
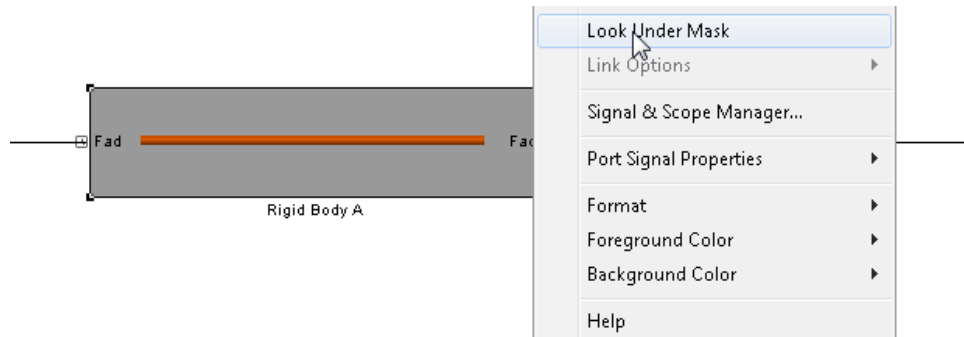
Model Rigid Body Subsystems

The model uses a separate subsystem for each rigid body. In the initial iteration, rigid bodies use a simple shape. You can specify shape in **Geometry** section of the Solid block dialog box.

- Rigid bodies **A**, **B**, and **D** use the Cylinder shape.
- Rigid body **C** uses the Brick shape.

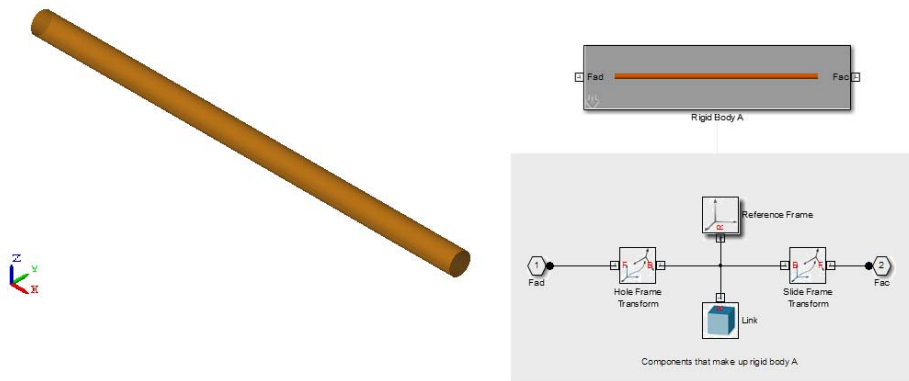
Each rigid body subsystem contains two frames that identify the position and orientation of each joint. During multibody assembly, you connect the frames directly to joint blocks. You can add frames to a rigid body subsystem with the Rigid Transform block.

To view the block diagram of a rigid body subsystem, click the subsystem block once and press **Ctrl+U**.



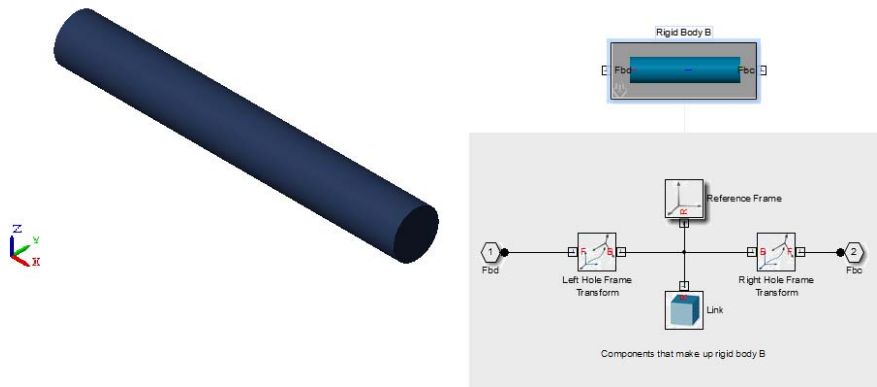
Rigid Body A

The example models Rigid Body A as a subsystem with two frames that identify joint location. The following figure shows the block diagram for Rigid Body A. The Solid block provides the Cylinder geometry, inertia, and color. The two Rigid Transform blocks provide the two joint frames relative to the reference frame. The Reference Frame block identifies a non-inertial reference frame.



Rigid Body B

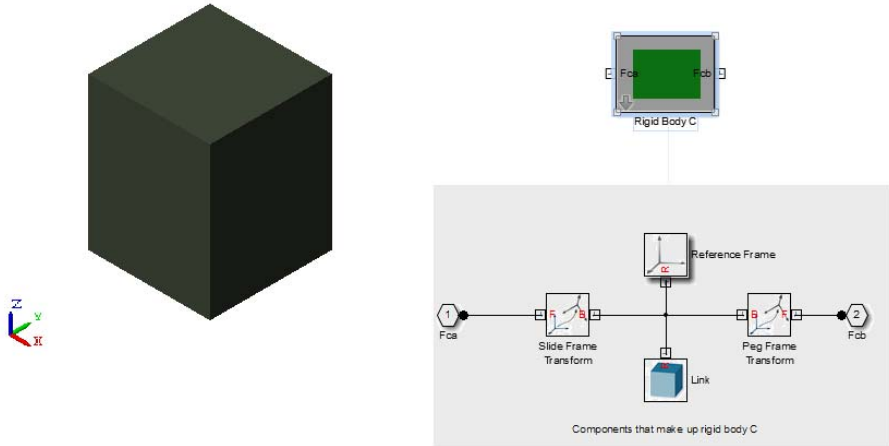
The example models Rigid Body B as a subsystem with two frames that identify joint location. The following figure shows the block diagram for Rigid Body B. Rigid Body B is identical to Rigid Body A, except for dimensions and color.



Rigid Body C

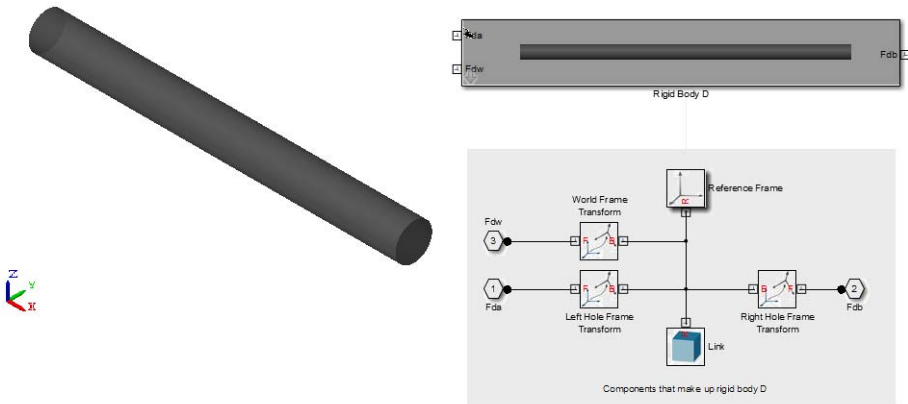
The example models Rigid Body C as a subsystem with two frames that identify joint location. The following figure shows the block diagram for Rigid Body C. The Solid block provides a Brick shape that distinguished Rigid Body

C from other rigid bodies. The block diagram of Rigid Body C is identical to Rigid Bodies A and B.



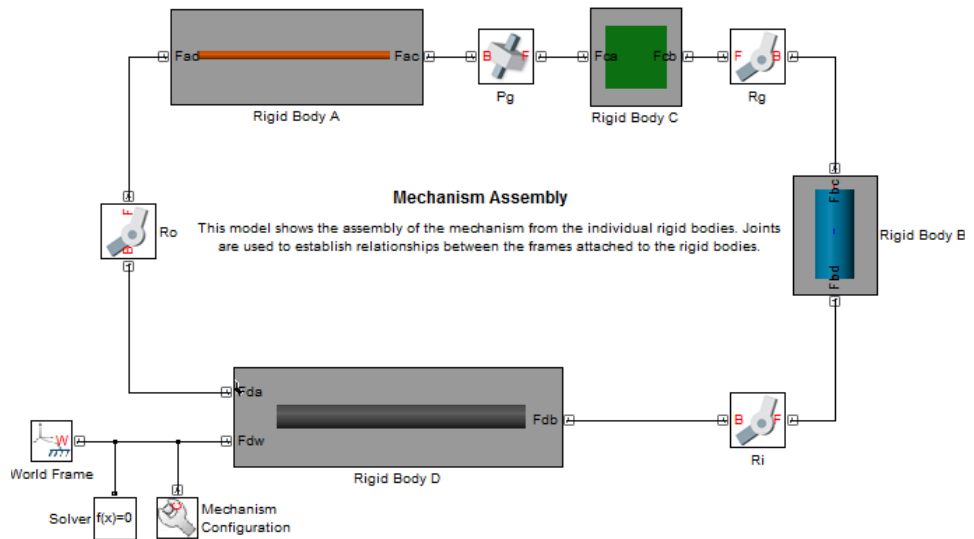
Rigid Body D

The example models Rigid Body D as a subsystem with two frames that identify joint location. The following figure shows the block diagram for Rigid Body D. Rigid Body D is identical to Rigid Body A, except for dimensions and color. An additional frame port connects Rigid Body D to the World frame.



Connect Rigid Body Subsystems with Joint Blocks

Once you have modeled the rigid bodies as separate subsystems, you can connect the subsystems with Joints and Constraints blocks. The double-crank aiming mechanism contains three revolute joints and one prismatic joint. Each joint connects two rigid bodies. The joint provides translational and/or rotational degrees of freedom (DOF) between two rigid bodies. The following figure illustrates the complete model:

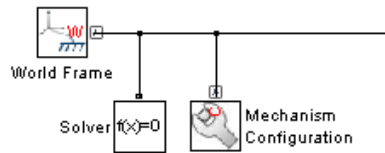


In the model:

- Prismatic joint **P_g** allows exactly one translational DOF between rigid bodies **A** and **C** — they can move along a single axis relative to each other.
- Revolute joint **R_g** allows exactly one rotational degree of freedom between rigid bodies **B** and **C** — they can rotate about one axis relative to each other.
- Revolute joint **R_i** allows exactly one rotational degree of freedom between rigid bodies **B** and **D** — they can rotate about one axis relative to each other.
- Revolute joint **R_o** allows exactly one rotational degree of freedom between rigid bodies **C** and **D** — they can rotate about one axis relative to each other.

In the joint block dialog box, you can specify initial state conditions. These include position and velocity parameters. You can also actuate the two connecting bodies at the joint with force or torque, and sense the relative motion parameters — position, velocity and acceleration. To open the dialog box, double-click the corresponding joint block.

A World Frame block grounds rigid body D to an inertial frame at absolute rest. Solver and Mechanism Configuration blocks provide model- and machine- wide simulation settings.



Troubleshoot Assembly Error

SimMechanics provides two diagnostic tools that you can use to guide the modeling process:

- Mechanics Explorer
- Model Report.

The Mechanics Explorer utility provides a 3-D display of your SimMechanics model. Use this utility often throughout the modeling process as a guide in rigid body modeling and multibody assembly. By default, Mechanics Explorer opens automatically when you update a SimMechanics model. In the Simulink Editor window, select **Simulation > Update Diagram** to update a model.

The Mechanics Explorer utility contains a Model Report tool which identifies the status of joints and constraints. Use the Model Report utility to determine if joints and constraints have successfully assembled. The report also provides information on the configuration of joints and constraints. In the Mechanics Explorer window, select **Tools > Model Report** to open the Model Report utility.

Discover Assembly Errors

Model `sm_dcrankaim_assembly_with_error` contains an assembly error. Open the model and troubleshoot the error.

- 1 At the MATLAB command line, enter `sm_dcrankaim_assembly_with_error`.
- 2 With the model window active, press **Ctrl+D** to update the model.

Note SimMechanics issues a position violation error. Close the error window, and continue with the troubleshooting procedure.

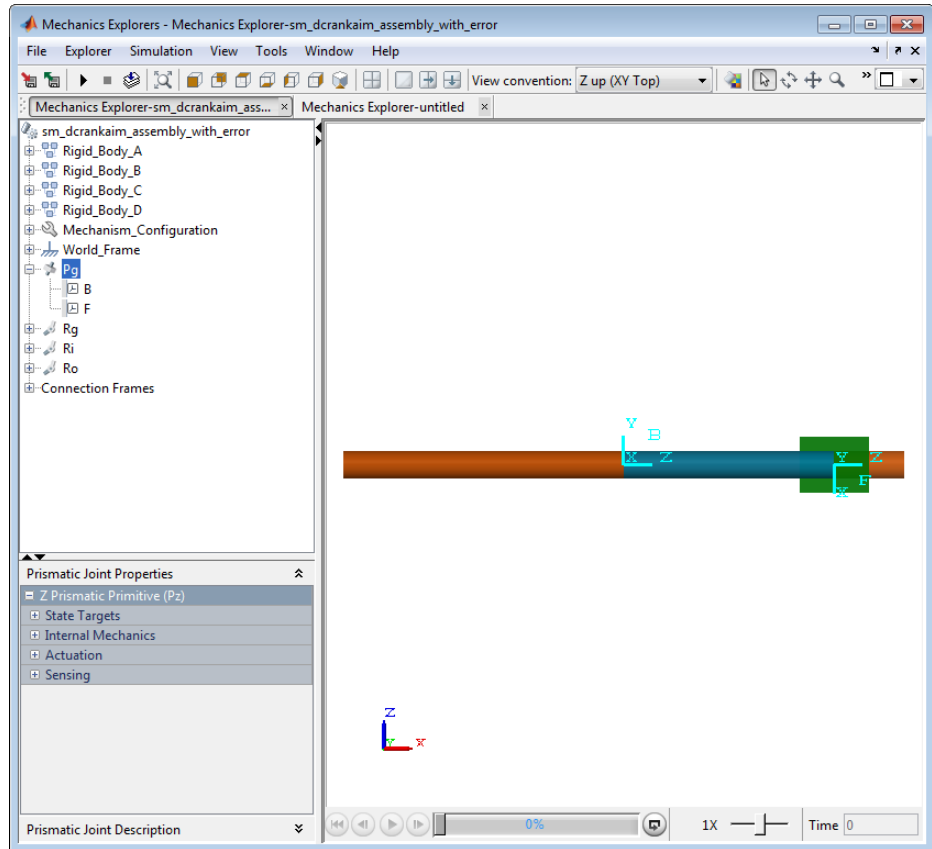
- 3 In the Mechanics Explorer window, select **Tools > Model Report**.
- 4 Model Report identifies the unassembled joint, Pz.

Identify Root Cause of Assembly Error

The position violation of joint Pg suggests that the joint base and follower frames are improperly aligned. You can confirm the frames are improperly aligned in Mechanics Explorer.

- 1 In the model navigation browser of Mechanics Explorer, click the name of joint Pg.
- 2 In the visualization pane of Mechanics Explorer, note the axes of base and follower frames point in different directions, causing the position violation that prevented model simulation.

Rotating either base or follower frame so that x- and y- axes point in the same directions will correct the position violation error.

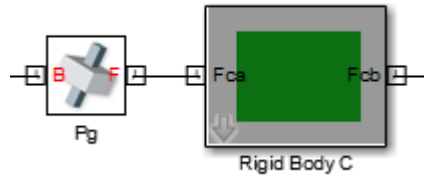


Correct Assembly Error

Joint Pg failed to assemble because the x- and y- axes of the joint base and follower frames point in different directions. To correct the position error, you must rotate either base or follower frame so that *all* axes of base and follower frames point in the same directions. In this example, you correct the orientation of the follower frame of joint Pg.

The follower frame of joint Pg is coincident with frame Fca of Rigid Body C. You can determine the rigid body frame that coincides with the follower frame of joint Pg in the block diagram for the aiming mechanism model. The

following figure shows that frame Fca of Rigid Body C connects directly to the follower frame of joint Pg, and is therefore coincident with that frame.

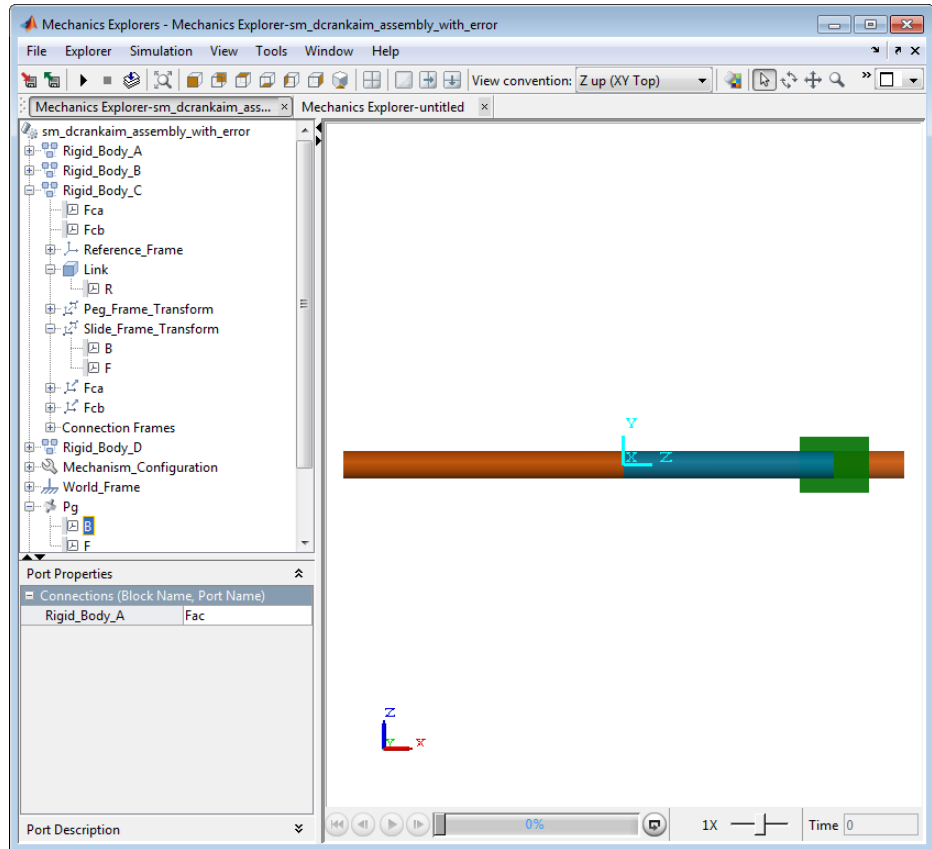


In the lower left corner of the subsystem block for Rigid Body C, click the down-arrow icon. The block diagram for Rigid Body C opens in a window. In the block diagram, identify the Rigid Transform block that connects to frame port Fca, *Slide Frame Transform*.

To correct the follower frame of joint Pg, follow these steps:

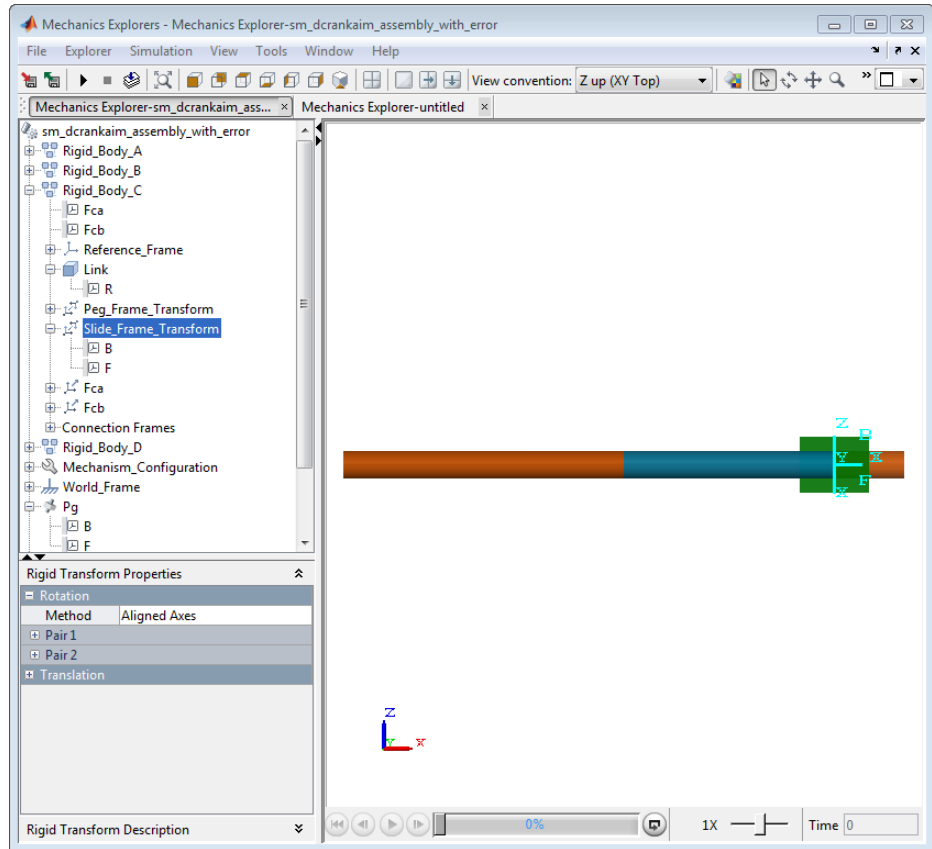
- 1** In the model navigation browser of Mechanics Explorer, select **Pg > B**.
- 2** In the visualization pane of Mechanics Explorer, note the axis orientation of the base frame.

Note This is the axis orientation you need to specify for the follower frame.

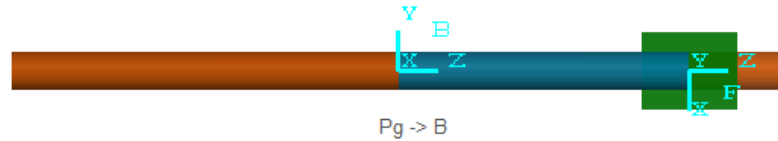
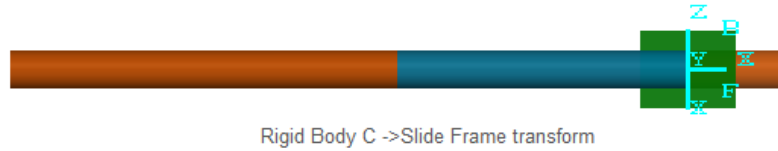


- 3 In the model navigation pane of Mechanics Explorer, select **Rigid Body C** > **Slide Frame Transform**
- 4 In the visualization pane of Mechanics Explorer, note the orientation of the base (B) and follower (F) frames.

Note The **Slide Frame Transform** block contains the definition of frame Fca, and hence of the follower frame of joint Pg.



The following figure shows the base and follower frames of the incorrectly specified frame **Fca**, and the base and follower frames of the original joint **Pg**. The objective is to align the axes of the **Slide Frame Transform** follower frame (**F**) with the axes of the joint **Pg** base frame (**B**).



- 5 In the block diagram for the aiming mechanism model, click the **Rigid Body C** subsystem block.
- 6 Press **Ctrl+U** to open the block diagram for Rigid Body C.
- 7 Double-click the Rigid Transform block **Slide Frame Transform**.
- 8 In **Rotation**, check that **Method** is Aligned Axes.
- 9 Expand **Pair 1** and **Pair 2**.
- 10 Change the incorrect **Pair 2** as follows:

Base	Follower
+Z	+Y

- 11 Click **OK**.
- 12 Press **Ctrl+D** to update the model.

Note Mechanics Explorer displays the updated model without error messages.

13 Press CTRL+T to simulate the model.