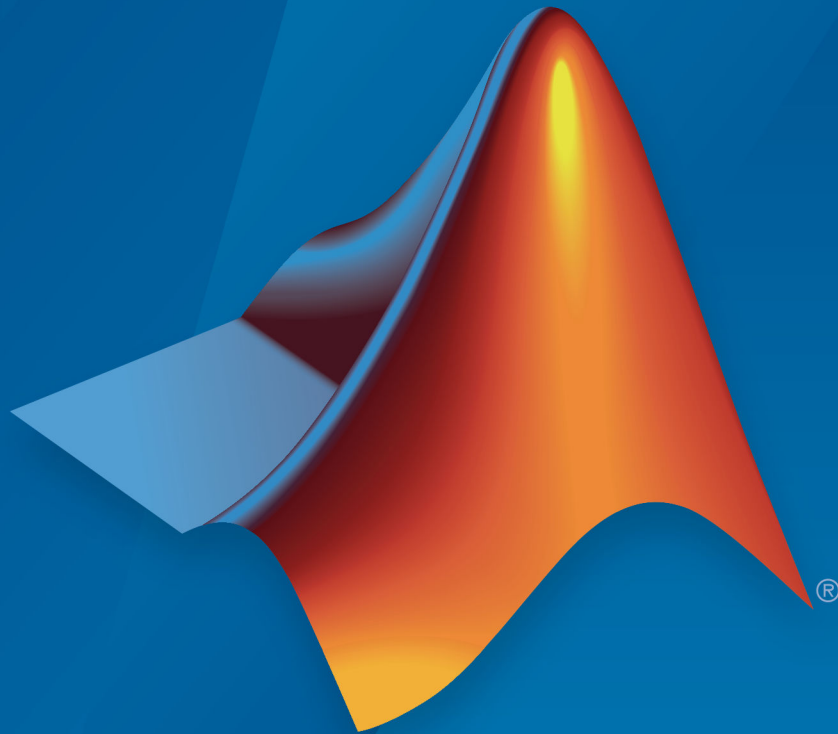


# Simscape™ Multibody™

## Getting Started Guide



MATLAB® & SIMULINK®

R2019b

 MathWorks®

# How to Contact MathWorks



Latest news: [www.mathworks.com](http://www.mathworks.com)  
Sales and services: [www.mathworks.com/sales\\_and\\_services](http://www.mathworks.com/sales_and_services)  
User community: [www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)  
Technical support: [www.mathworks.com/support/contact\\_us](http://www.mathworks.com/support/contact_us)



Phone: 508-647-7000



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

*Simscape™ Multibody™ Getting Started Guide*

© COPYRIGHT 2002–2019 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](http://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](http://www.mathworks.com/patents) for more information.

## Revision History

March 2012	Online only	New for Version 4.0 (Release R2012a)
September 2012	Online only	Revised for Version 4.1 (Release R2012b)
March 2013	Online only	Revised for Version 4.2 (Release R2013a)
September 2013	Online only	Revised for Version 4.3 (Release R2013b)
March 2014	Online only	Revised for Version 4.4 (Release R2014a)
October 2014	Online only	Revised for Version 4.5 (Release R2014b)
March 2015	Online only	Revised for Version 4.6 (Release R2015a)
September 2015	Online only	Revised for Version 4.7 (Release R2015b)
March 2016	Online only	Revised for Version 4.8 (Release R2016a) (Renamed from <i>SimMechanics™ Getting Started</i> )
September 2016	Online only	Revised for Version 4.9 (Release R2016b)
March 2017	Online only	Revised for Version 5.0 (Release R2017a)
September 2017	Online only	Revised for Version 5.1 (Release R2017b)
March 2018	Online only	Revised for Version 5.2 (Release R2018a)
September 2018	Online only	Revised for Version 6.0 (Release R2018b)
March 2019	Online only	Revised for Version 6.1 (Release R2019a)
September 2019	Online only	Revised for Version 7.0 (Release R2019b)



## 1

### Introduction to Simscape Multibody Software

<b>Simscape Multibody Product Description</b> .....	<b>1-2</b>
Key Features .....	<b>1-2</b>
<b>Multibody Model Anatomy</b> .....	<b>1-3</b>
Basic Model Components .....	<b>1-3</b>
Model Actuation .....	<b>1-4</b>
Dynamical Sensing .....	<b>1-6</b>
<b>Creating a Multibody Model</b> .....	<b>1-9</b>
Step 1: Study the System to Model .....	<b>1-9</b>
Step 2: Model the Bodies .....	<b>1-9</b>
Step 3: Assemble the Multibody System .....	<b>1-10</b>
Step 4: Specify and Sense Dynamic Variables .....	<b>1-10</b>
<b>Start a Model from a Template</b> .....	<b>1-11</b>
<b>Model a Simple Link</b> .....	<b>1-12</b>
Tutorial Overview .....	<b>1-12</b>
Build Model .....	<b>1-12</b>
Generate Subsystem .....	<b>1-14</b>
Visualize Model .....	<b>1-15</b>
Save Custom Block .....	<b>1-16</b>
<b>Model a Simple Pendulum</b> .....	<b>1-17</b>
Tutorial Overview .....	<b>1-17</b>
Build Model .....	<b>1-18</b>
Specify Gravity .....	<b>1-19</b>
Set Pendulum Starting Position .....	<b>1-19</b>
Configure Solver .....	<b>1-19</b>
Assemble Model .....	<b>1-19</b>
Simulate Model .....	<b>1-20</b>
Save Model .....	<b>1-20</b>

<b>Analyze a Simple Pendulum</b> .....	<b>1-21</b>
Tutorial Overview .....	<b>1-21</b>
Sense Pendulum Motion .....	<b>1-22</b>
Analyze Undamped Pendulum .....	<b>1-23</b>
Analyze Damped Pendulum .....	<b>1-26</b>
Analyze Damped and Driven Pendulum .....	<b>1-29</b>

# Introduction to Simscape Multibody Software

---

- “Simscape Multibody Product Description” on page 1-2
- “Multibody Model Anatomy” on page 1-3
- “Creating a Multibody Model” on page 1-9
- “Start a Model from a Template” on page 1-11
- “Model a Simple Link” on page 1-12
- “Model a Simple Pendulum” on page 1-17
- “Analyze a Simple Pendulum” on page 1-21

## Simscape Multibody Product Description

### Model and simulate multibody mechanical systems

Simscape Multibody provides a multibody simulation environment for 3D mechanical systems, such as robots, vehicle suspensions, construction equipment, and aircraft landing gear. You can model multibody systems using blocks representing bodies, joints, constraints, force elements, and sensors. Simscape Multibody formulates and solves the equations of motion for the complete mechanical system. You can import complete CAD assemblies, including all masses, inertias, joints, constraints, and 3D geometry, into your model. An automatically generated 3D animation lets you visualize the system dynamics.

Simscape Multibody helps you develop control systems and test system-level performance. You can parameterize your models using MATLAB® variables and expressions, and design control systems for your multibody system in Simulink®. You can integrate hydraulic, electrical, pneumatic, and other physical systems into your model using components from the Simscape family of products. To deploy your models to other simulation environments, including hardware-in-the-loop (HIL) systems, Simscape Multibody supports C-code generation.

### Key Features

- Single environment for simulating 3D mechanical systems together with multidomain physical systems and control algorithms in Simulink
- Part definition tools, including standard geometry, extrusions defined in MATLAB, and STEP files
- Part definition by reading native CAD files, including CATIA
- CAD assembly import from SolidWorks®, PTC Creo™ (Pro/ENGINEER®), Autodesk Inventor®, and Onshape®
- Mechanical joints and constraints covering standard and custom kinematic relationships
- 3D animation of multibody system dynamics
- Support for C-code generation (with Simulink Coder™)



## Multibody Model Anatomy

### In this section...

“Basic Model Components” on page 1-3

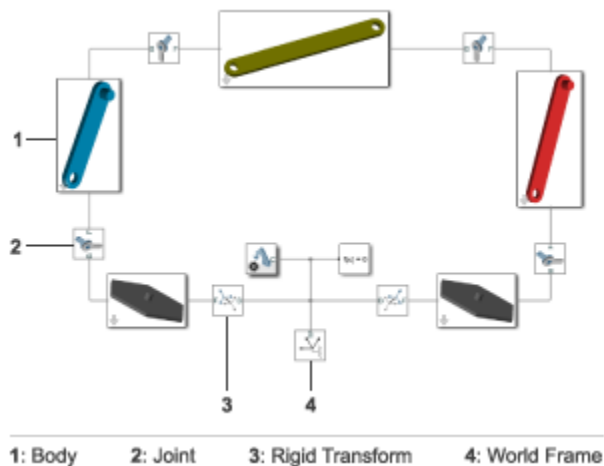
“Model Actuation” on page 1-4

“Dynamical Sensing” on page 1-6

At its core, a multibody system is a set of bodies linked through joints and bound by kinematic constraints such as gears. Forces and torques of various types enable you to actuate the various bodies, while sensors enable you to sense the resulting motion. In Simscape Multibody software, you use blocks to directly represent these components.

### Basic Model Components

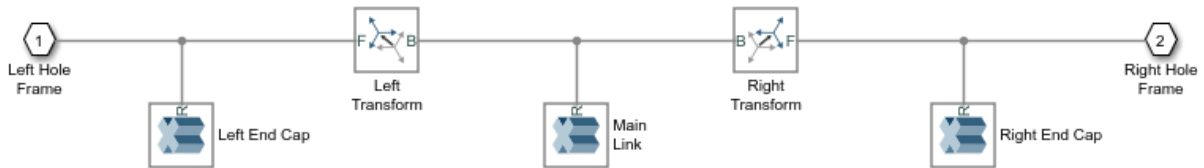
Consider the Simscape Multibody four-bar featured example. You can open this model at the MATLAB command prompt by entering `sm_four_bar`. The model consists of solid body subsystems interconnected through Joint blocks. The Joint blocks define the degrees of freedom available between bodies. The body subsystems define, through an underlying block diagram, the solid properties of the body.



### Basic Components of `sm_four_bar` Model

The body subsystems in turn consist of Extruded Solid blocks interconnected through Rigid Transform blocks. The Extruded Solid blocks provide the solid properties, including geometry, inertia, color, and frames—axis triads that encode all position and orientation information in a model. The Rigid Transform blocks rotate and translate the solids to ensure proper body assembly.

The figure shows the underlying block diagram for a binary link body subsystem. This diagram includes three Solid blocks, representing the central and distal portions of the binary link. Two Rigid Transform blocks translate the distal solids to their proper locations near the ends of the central solid.



## Body Subsystem Components

Other common blocks in a model include:

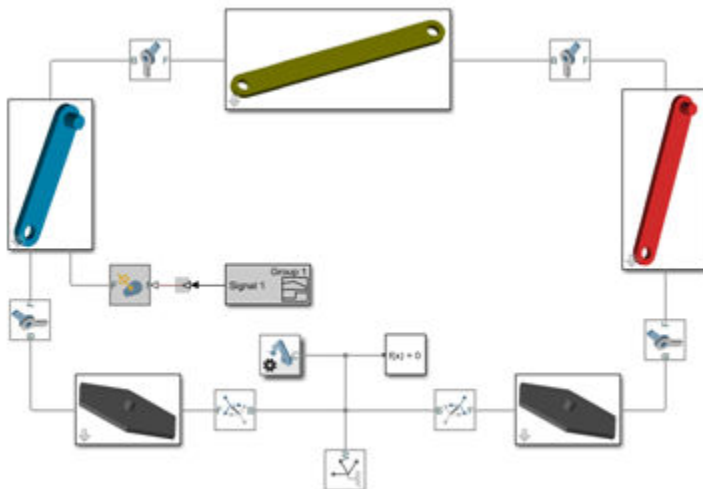
- World Frame — Adds an inertial reference frame to the model.
- Rigid Transform — Rotates and translates bodies and other frames relative to each other.
- Mechanism Configuration — Defines the gravitational constant in a model.
- Solver Configuration — Sets Simscape local solver options such as types, tolerances, and time steps. This is the only required block in a model.

## Model Actuation

You can actuate a model by applying a force or torque to a body or to a joint. To represent forces and torques acting on a body, Simscape Multibody provides a Forces and Torques library. Add a block from this library and connect it to the body frame(s) that you want to apply the force or torque to.

Block	Function
External Force and Torque	General force and/or torque originating outside of the multibody model
Internal Force	General force pair between two arbitrary frames
Spring and Damper Force	Spring-damper force pair between two arbitrary frames
Inverse Square Law Force	Force pair with inverse dependence on the square distance between two arbitrary frames (e.g., Coulomb electrostatic forces)
Gravitational Field	Gravitational pull of a point mass on all bodies as a function of their distances to the point mass itself

The figure shows a four-bar model with an External Force and Torque block for force and torque prescription at a crank link frame.

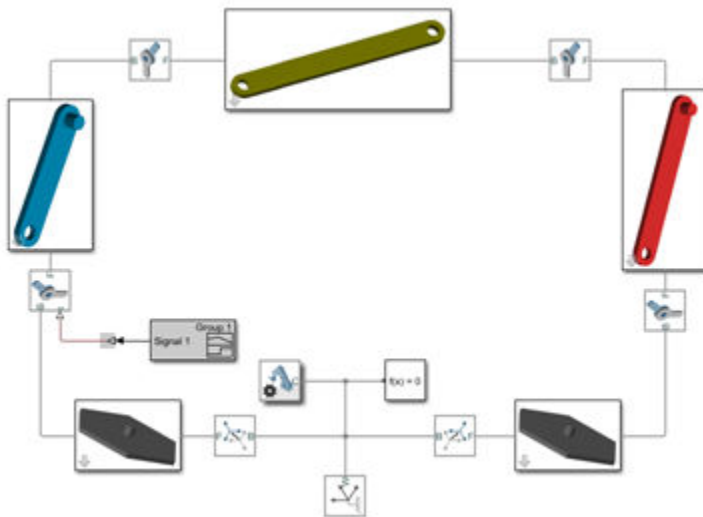


To specify the force or torque acting at a joint, Simscape Multibody provides a selection of actuation inputs directly in the joint blocks. Each joint primitive—the basic component of a joint block—provides a selection of actuation inputs specific to that primitive.

Joint actuation inputs can be of two types:

- Motion — Specify the time-varying trajectory of a given joint primitive.
- Force or torque — Specify the time-varying actuation force or torque acting at a given joint primitive.

The figure shows a four-bar model with an actuation torque acting on a revolute joint.



## Dynamical Sensing

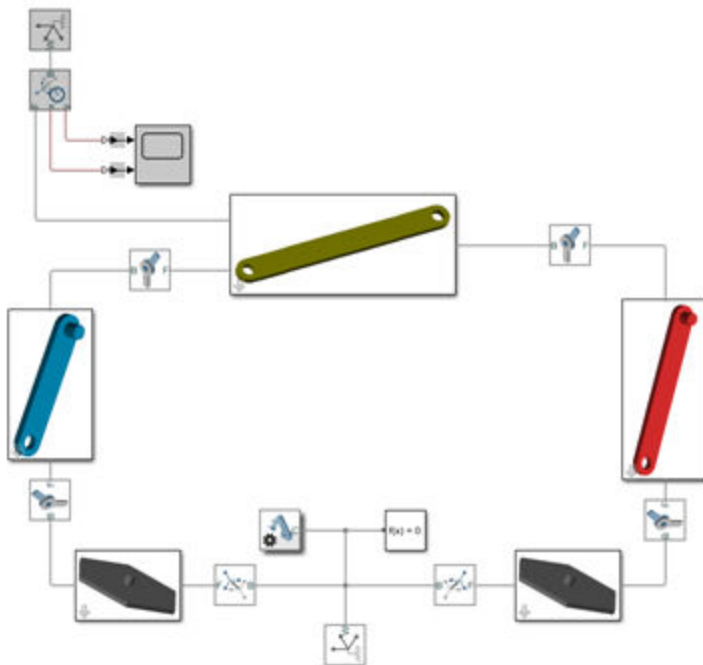
You can sense various dynamical variables between frame pairs, e.g., for analysis or control design. Sensing outputs can be of two types:

- Motion — Compute and output the relative position, velocity, or acceleration between two Simscape Multibody frames. You can sense motion between joint frames, by using the sensing capability of joint blocks, or between arbitrary frames, by using the Transform Sensor block.
- Force or torque — Compute and output the forces and torques acting between two Simscape Multibody frames. You can sense force and torque between the port frames of certain Forces and Torques blocks, such as the Inverse Square Law Force block, or between the port frames of a joint block.

Joint blocks enable you to sense different types of forces and torques between their respective port frames, including:

- Actuation force or torque acting at a given joint primitive.
- Constraint force and torque acting joint-wide to prevent motion normal to the joint degrees of freedom.
- Total force and torque, including constraint and joint primitive actuation contributions, acting joint-wide.

The figure shows a four-bar model with a Transform Sensor block for trajectory coordinate sensing between a coupler link frame and the world frame.



## See Also

### Related Examples

- “Model a Simple Link” on page 1-12
- “Model a Simple Pendulum” on page 1-17

- “Analyze a Simple Pendulum” on page 1-21

## Creating a Multibody Model

### In this section...

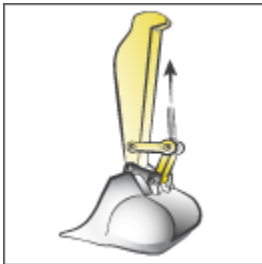
“Step 1: Study the System to Model” on page 1-9

“Step 2: Model the Bodies” on page 1-9

“Step 3: Assemble the Multibody System” on page 1-10

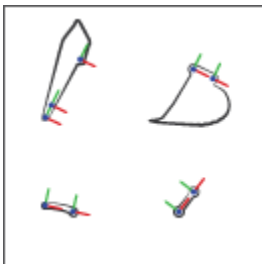
“Step 4: Specify and Sense Dynamic Variables” on page 1-10

### Step 1: Study the System to Model



Identify the relevant bodies, joints, and constraints to incorporate in your model. Consider the forces and torques acting between bodies or directly at joints.

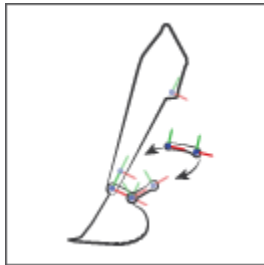
### Step 2: Model the Bodies



Specify the solid properties of the various bodies. Add frames to the bodies so that you can connect joints and constraints, apply forces and torques, and sense motion. See “Model a Simple Link” on page 1-12 for an example showing how to model a simple body

—one comprising a single solid. See “Try It: Create a Compound Geometry” for a more elaborate example showing how to model a compound body—one comprising multiple solids.

## Step 3: Assemble the Multibody System



Connect the bodies by pairs through joints. Constrain body motion through gears and other kinematic constraints. See “Model an Open-Loop Kinematic Chain” for an example showing how to connect bodies through joints. See “Model a Compound Gear Train” for an example showing how to constrain motion through gears.

## Step 4: Specify and Sense Dynamic Variables

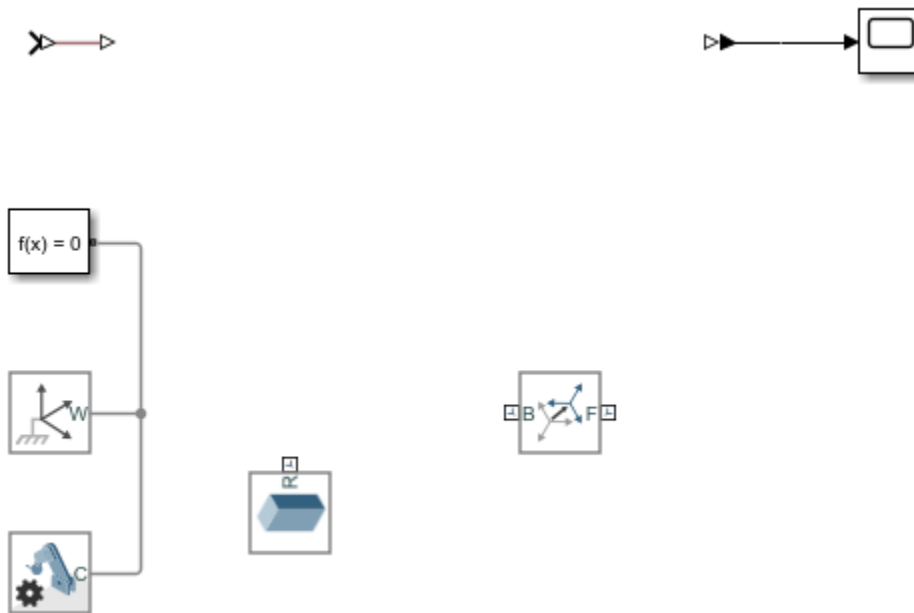


Model external loads, interactions between bodies, and joint actuation inputs. Sense forces, torques, and motion variables as functions of time. See “Specify Joint Actuation Torque” for an example showing how to specify a joint actuation torque. See “Sense Motion Using a Transform Sensor Block” for an example showing how to sense frame position using the Transform Sensor block.



## Start a Model from a Template

You can start a new Simscape Multibody model directly from the MATLAB command prompt by entering `smnew`. The `smnew` function opens a Simscape Multibody model template with commonly used blocks and variable-step automatic solver selection for best performance. The figure shows the model template.



You can also open the Simscape Multibody block library from the MATLAB command prompt by entering `sm_lib`. Browse the library to select blocks for your model. Alternatively, type the block name in the Simulink Model window and select a match from the Simulink search results.

See “Model a Simple Link” on page 1-12 to model a body in the Simscape Multibody model template. See “Model a Simple Pendulum” on page 1-17 to integrate the body in a multibody mechanism.

## Model a Simple Link

### In this section...

“Tutorial Overview” on page 1-12

“Build Model” on page 1-12

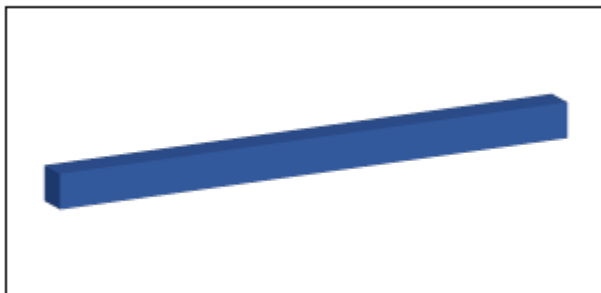
“Generate Subsystem” on page 1-14

“Visualize Model” on page 1-15

“Save Custom Block” on page 1-16

### Tutorial Overview

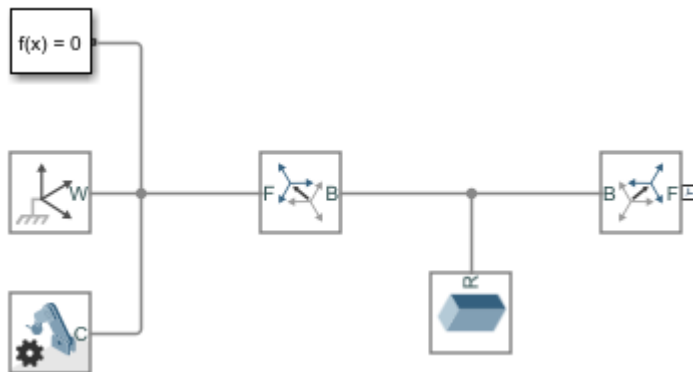
Mechanical links are common building blocks in linkages, mechanisms, and machines. The simple pendulum is an example with one link. In this tutorial, you model a simple link with two end frames that you can later connect to joints. Rigid Transform blocks provide the end frames, while a Solid block provides geometry, inertia, and color. For simplicity, the model assumes the link has a brick shape.



### Build Model

- 1 At the MATLAB command line, enter `smnew`. The Simscape Multibody block library and a model template with commonly used blocks open up.
- 2 Make a copy of the Rigid Transform block and paste it in the model. The Rigid Transform blocks enable you to create new frames to which you can connect joints during multibody assembly.

- 3 Delete the blocks Simulink-PS Converter, PS-Simulink Converter, and Scope. You do not need these blocks in this tutorial.
- 4 Connect the remaining blocks as shown in the figure. Ensure that the base frame ports (B) of the Rigid Transform blocks both face the Brick Solid block frame port. Since each Rigid Transform block applies a spatial transformation with respect to its base frame, switching port connections generally changes the spatial relationship between the two frames.



- 5 In the Brick Solid block dialog box, specify the following parameters. Later, you define the MATLAB variables shown using a Subsystem block that contains the Solid and Rigid Transform blocks. Among its advantages, this approach enables you to update variables used in multiple blocks from a single place—the Subsystem block dialog box.

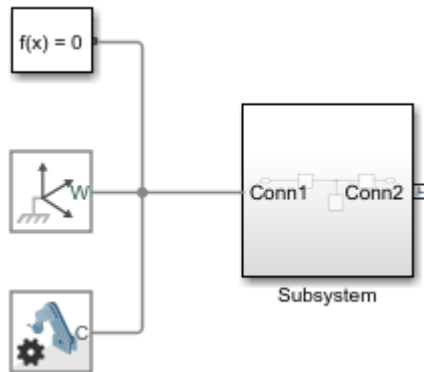
Parameter	Value	Units
<b>Geometry &gt; Dimensions</b>	[L W H]	Change to cm
<b>Inertia &gt; Density</b>	rho	Default units
<b>Graphic &gt; Visual Properties &gt; Color</b>	rgb	Not applicable


- 6 In the dialog boxes of the Rigid Transform and Rigid Transform1 blocks, specify the following parameters. These parameters encode the offset between the base and follower port frames of the Rigid Transform blocks, located at the link ends, with respect to the Solid reference port frame.

Parameter	Rigid Transform1	Rigid Transform	Units
Translation > Method	Standard Axis	Standard Axis	Not applicable
Translation > Axis	-X	+X	Not applicable
Translation > Offset	L/2	L/2	Change to cm

## Generate Subsystem

- 1 Select the Brick Solid block and the two Rigid Transform blocks.
- 2 Right-click the highlighted region and select **Create Subsystem from Selection**. Simulink adds a new Subsystem block containing the Solid and Rigid Transform blocks. At the end of the tutorial, this will be a custom block representing the simple link body.




- 3 Right-click the Subsystem block, and select **Mask > Create Mask**. A mask editor opens up, enabling you to specify the numerical values of the MATLAB variables you entered in the Solid and Rigid Transform block dialog boxes.
- 4 In the **Parameters & Dialog** tab of the Mask Editor window, add five edit fields  to the **Parameters** folder. You can find this folder in the **Dialog box** pane. In the edit fields, specify the following parameters and click **OK**. **Prompt** is the desired text for each parameter in the Subsystem block dialog box. **Name** is the MATLAB variable associated with each Subsystem block parameter.

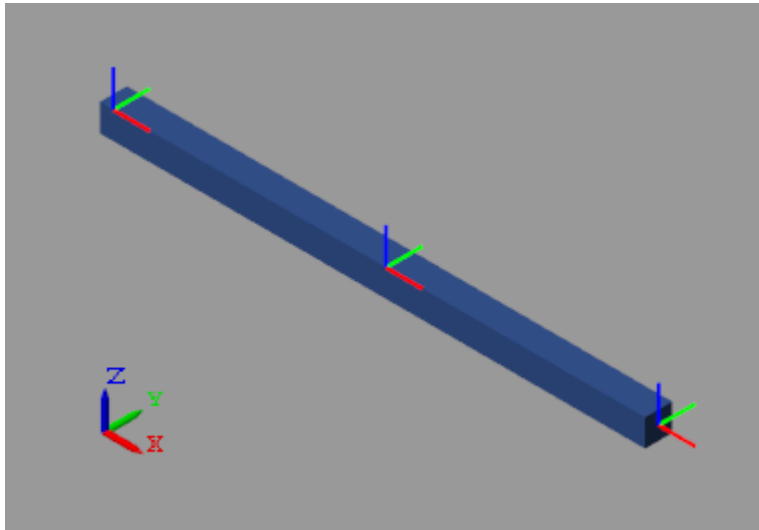
Prompt	Name
Length (cm)	L
Width (cm)	W
Thickness (cm)	H
Density (kg/m <sup>3</sup> )	rho
Color [R G B]	rgb

- 5 Double-click the Subsystem block dialog box and enter the following numerical values. These are the values of the MATLAB variables that you entered in the Solid and Rigid Transform block dialog boxes.

Parameter	Value
Length (cm)	20
Width (cm)	1
Thickness (cm)	1
Density (kg/m <sup>3</sup> )	2700
Color [R G B]	[0.25 0.40 0.70]

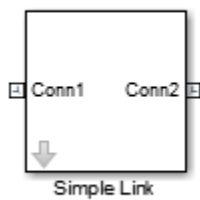
## Visualize Model

Update the block diagram. In the **Modeling** tab, click **Update Model**. Mechanics Explorer opens with a front view of the simple link model. In the Mechanics Explorer toolstrip, select the isometric view button  to obtain the 3-D view shown below. To view the frames present in the model—including those you created using the Rigid Transform blocks—select **View > Show Frames** in the Mechanics Explorer menu bar.



## Save Custom Block

Rename the Subsystem block Simple Link and save it in a custom block library. You reuse this block in tutorial “Model a Simple Pendulum” on page 1-17.



## Simple Link Custom Block

## Model a Simple Pendulum

### In this section...

“Tutorial Overview” on page 1-17

“Build Model” on page 1-18

“Specify Gravity” on page 1-19

“Set Pendulum Starting Position” on page 1-19

“Configure Solver” on page 1-19

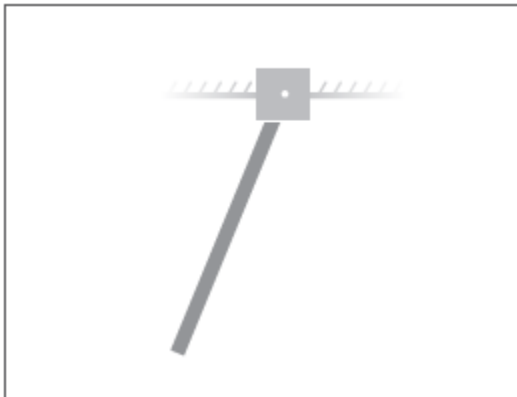
“Assemble Model” on page 1-19

“Simulate Model” on page 1-20

“Save Model” on page 1-20

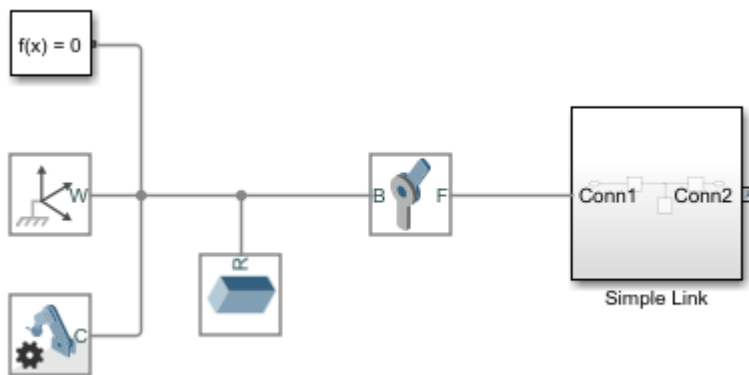
### Tutorial Overview

The pendulum is the simplest mechanical system you can model. This system contains two bodies, a link and a fixed pivot, connected by a revolute joint. In this tutorial, you model and simulate a pendulum using the custom link block you created in “Model a Simple Link” on page 1-12. A Revolute Joint block provides the rotational degree of freedom between the link and the world frame.



## Build Model

- 1 At the MATLAB command prompt, enter `smnew`. The Simscape Multibody block library and a model template with commonly used blocks open up.
- 2 Delete blocks Simulink-PS Converter, PS-Simulink Converter, Scope, and Rigid Transform. You do not need them in this tutorial.
- 3 Add to the model the Simple Link custom block created in “Model a Simple Link” on page 1-12.
- 4 Add a Revolute Joint block to the model. You can find this block in the **Simscape > Multibody > Joints** library. This block provides one rotational degree of freedom between its port frames.
- 5 Connect the blocks as shown in the figure. The port orientation of the Revolute Joint block becomes important when you specify joint state targets, prescribe joint actuation inputs, or sense joint dynamic variables. The Revolute Joint block interprets each quantity as that applied to the follower frame with respect to the base frame, so switching the port connections can affect model assembly and simulation.



- 6 In the Solid block dialog box, specify the following parameters. This block connects rigidly to the World frame and therefore has no effect on model dynamics. You can leave the inertia parameters in their default values.

Parameter	Value	Units
Geometry > Dimensions	[4 4 4]	Change to cm



Parameter	Value	Units
<b>Graphic &gt; Visual Properties &gt; Color</b>	[0.80 0.45 0]	Not applicable

## Specify Gravity

The Revolute Joint block uses the common Z axis of the base and follower frames as the joint rotation axis. To ensure the pendulum oscillates under the effect of gravity, change the gravity vector so it no longer aligns with the Z axis. To do this, in the Mechanism Configuration block dialog box, set the **Uniform Gravity > Gravity** parameter to [0 -9.81 0].

## Set Pendulum Starting Position

You can specify the desired joint angle using the **State Targets** menu in the Revolute Joint block dialog box. To do this, select **State Targets > Position** and enter the desired joint angle. For this tutorial, you can leave the angle in its default value, which corresponds to a horizontal pendulum starting position.

## Configure Solver

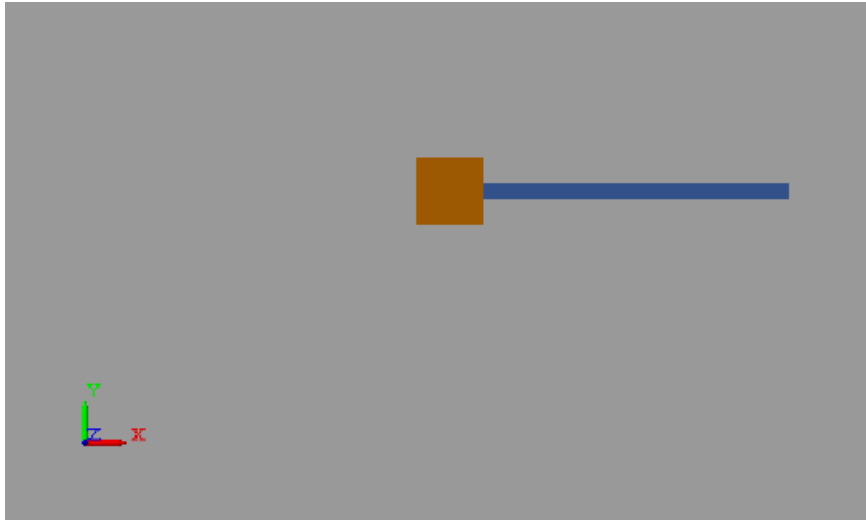
- 1 Open the Configuration Parameters. In the **Modeling** tab, click **Model Settings**.
- 2 In the **Solver** tab, set the **Solver** parameter to `ode15s (stiff/NDF)`. This solver is the recommended choice for physical models.
- 3 Set **Max step size** to `0.01` and click **OK**. The small step size increases the simulation accuracy and produces a smoother animation in Mechanics Explorer. Small step sizes can have a detrimental effect on simulation speed but, in such a simple model, a value of `0.01` provides a good balance between simulation speed and accuracy.

## Assemble Model

Update the block diagram. In the **Modeling** tab, click **Update Model**. Mechanics Explorer opens with a 3-D view of the model in its initial configuration.

In the Mechanics Explorer toolstrip, check that the **View convention** parameter is set to **Y up (XY Front)**. This view convention ensures that gravity is vertically aligned on your screen. Select a standard view button to refresh the Mechanics Explorer display. The

figure shows a front view of the model. Save the visualization settings by clicking the Save explorer configuration to model button .



## Simulate Model

Run the simulation. Mechanics Explorer plays a physics-based animation of the pendulum model.

## Save Model

Save the model in a convenient folder under the name `simple_pendulum`. You reuse this model in the tutorial “Analyze a Simple Pendulum” on page 1-21.

## Analyze a Simple Pendulum

### In this section...

“Tutorial Overview” on page 1-21

“Sense Pendulum Motion” on page 1-22

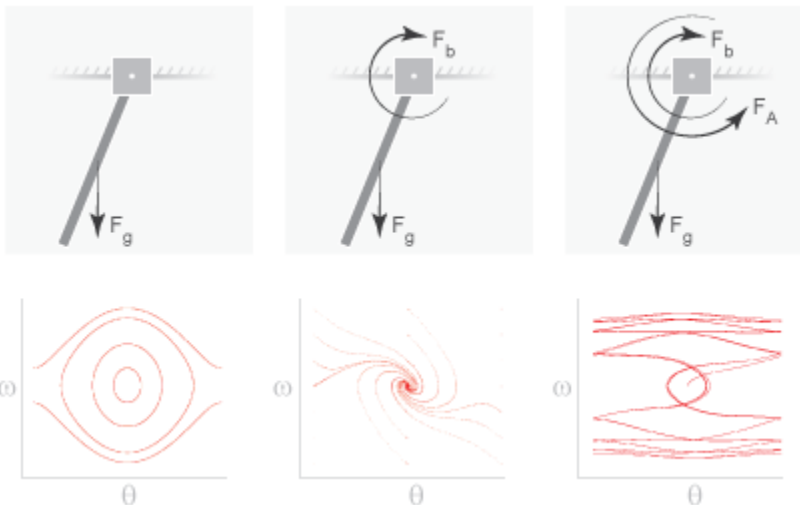
“Analyze Undamped Pendulum” on page 1-23

“Analyze Damped Pendulum” on page 1-26

“Analyze Damped and Driven Pendulum” on page 1-29

### Tutorial Overview

In this tutorial, you explore the various forces and torques that you can add to a model. Then, using blocks with motion sensing capability, you analyze the resulting dynamic response of the model. The end result is a set of time-domain and phase plots, one for each combination of forces and torques. You create these plots using MATLAB commands with Simscape Multibody motion outputs as arguments.



Your starting point is the simple pendulum model that you built in “Model a Simple Pendulum” on page 1-17. By adding forces and torques to this model, you incrementally

change the pendulum from undamped and free to damped and driven. The forces and torques that you apply include:

- Gravitational force ( $F_g$ ) — Global force, acting on every body in direct proportion to its mass, that you specify in terms of the acceleration vector  $g$ . You specify this vector using the Mechanism Configuration block.
- Joint damping ( $F_b$ ) — Internal torque, between the pendulum and the joint fixture, that you parameterize in terms of a linear damping coefficient. You specify this parameter using the Revolute Joint block that connects the pendulum to the joint fixture.
- Actuation torque ( $F_A$ ) — Driving torque, between the pendulum and the joint fixture, that you prescribe directly as a Simscape physical signal. You prescribe this signal using the Revolute Joint block that connects the pendulum to the joint fixture.

## Sense Pendulum Motion

- 1 Open the `simple_pendulum` model that you created in tutorial “Model a Simple Pendulum” on page 1-17.
- 2 In the Sensing menu of the Revolute Joint block dialog box, select the following variables:
  - **Position**
  - **Velocity**

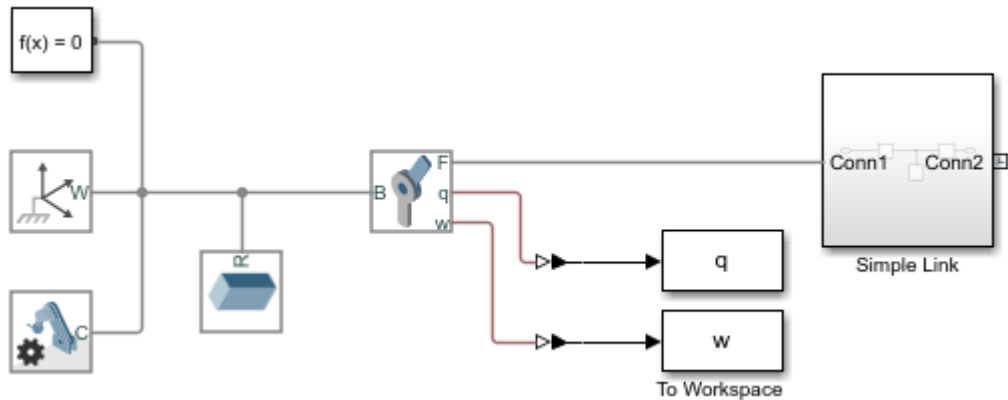
The block exposes two additional physical signal ports, labeled  $q$  and  $w$ , that output the angular position and velocity of the pendulum with respect to the world frame.

- 3 Add the following blocks to the model. You use them here to output the joint position and velocity to the MATLAB base workspace.

Library	Block	Quantity
<b>Simscape &gt; Utilities</b>	PS-Simulink Converter	2
<b>Simulink &gt; Sinks</b>	To Workspace	2

- 4 Change the **Variable name** parameters in the To Workspace block dialog boxes to  $q$  and  $w$ . These variables make it easy to identify the joint variables that the To Workspace blocks output during simulation—position, through the Revolute Joint block port  $q$ , and velocity, through the Revolute Joint block port  $w$ .
- 5 Connect the blocks as shown in the figure. Ensure that the To Workspace block with variable name  $q$  connects, through the PS-Simulink Converter block, to the Revolute

Joint block port  $q$ , and that the To Workspace block with variable name  $w$  connects to the Revolute Joint block port  $w$ .



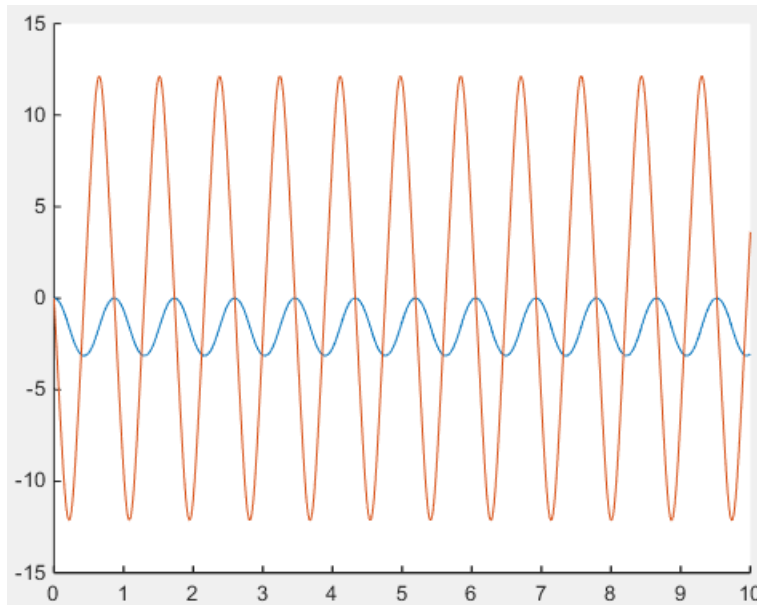
- 6 Save the model under a different name, e.g., `simple_pendulum_analysis`, in a convenient folder.

## Analyze Undamped Pendulum

- 1 Run the simulation. Mechanics Explorer opens with a 3-D animation of the simple pendulum model.
- 2 Plot the joint position and velocity with respect to time, e.g., by entering the following code at the MATLAB command prompt:

```
figure; % Open a new figure
hold on;
plot(q); % Plot the pendulum angle
plot(w); % Plot the pendulum angular velocity
```

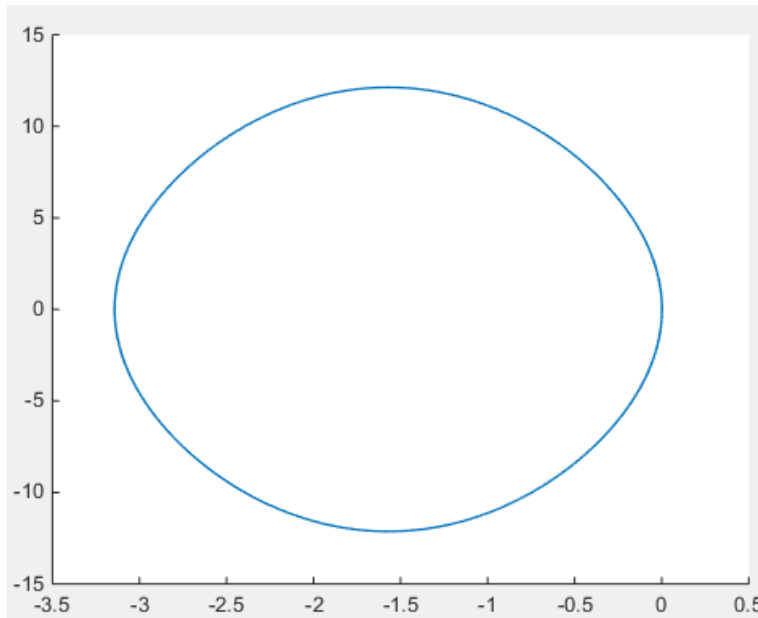
The figure shows the resulting plot.



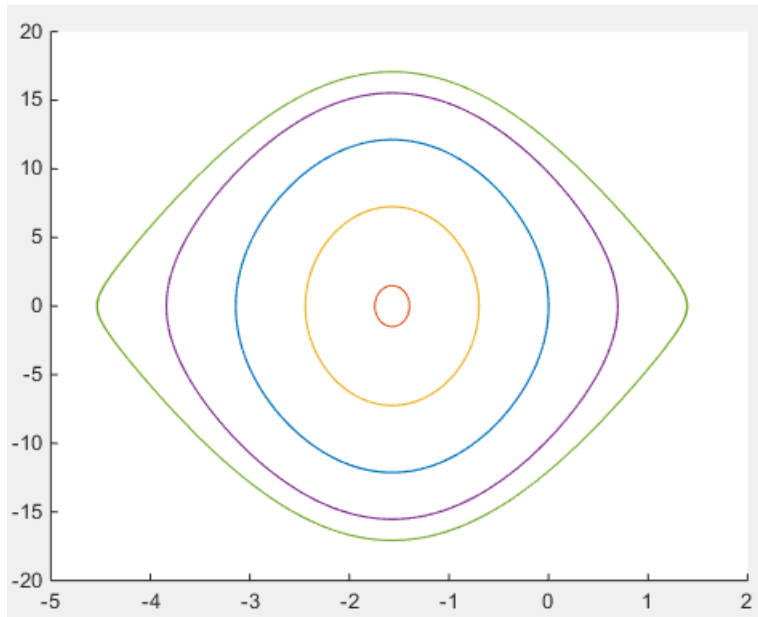
- 3** Plot the joint angular velocity with respect to the angular position, e.g., by entering the following code at the MATLAB command prompt.

```
figure;  
plot(q.data, w.data);
```

The result, shown in the figure, is the phase plot of the joint corresponding to a starting position of zero degrees with respect to the horizontal plane.



Try simulating the model using different starting angles. You can change the starting angle in the **State Targets > Position** menu of the Revolute Joint block dialog box. The figure shows a compound phase plot for starting angles of -80, -40, 0, 40, and 80 degrees.



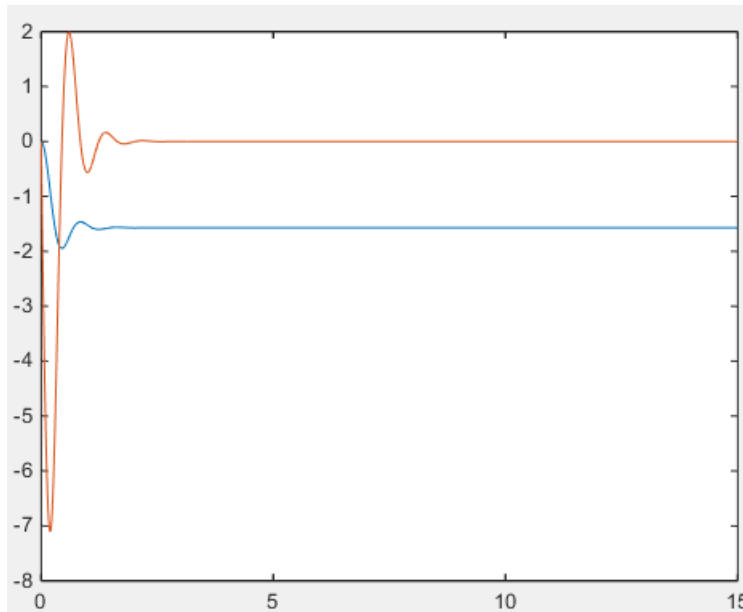
## Analyze Damped Pendulum

- 1 In the Revolute Joint block dialog box, set **Internal Mechanics > Damping** to  $8e-5$  (N\*m)/(deg/s). The damping coefficient causes energy dissipation during motion, resulting in a gradual decay of the pendulum oscillation amplitude.
- 2 Ensure that **State Targets > Position > Value** is set to 0 deg.
- 3 Run the simulation.
- 4 Plot the joint position and velocity with respect to time. To do this, at the MATLAB command prompt, you can enter this code:

```
figure;  
hold on;  
plot(q);  
plot(w);
```

The figure shows the resulting plot. Note that the pendulum oscillations decay with time due to damping. At larger damping values, the pendulum becomes overdamped, and the oscillations disappear altogether.

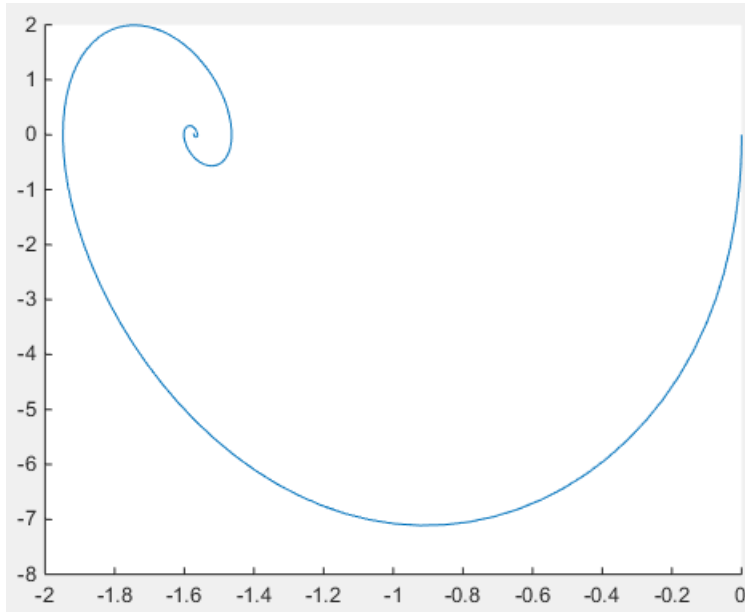




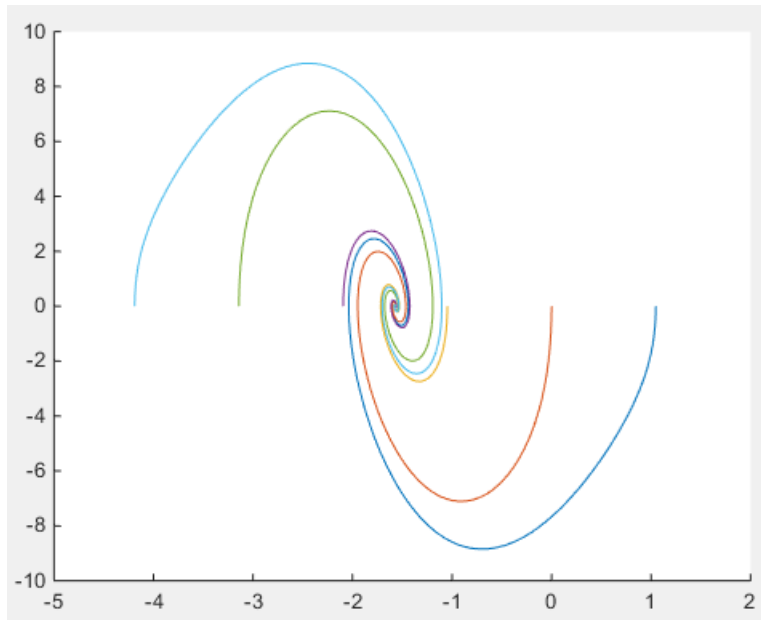
- 5 Plot the joint phase plot. To do this, at the MATLAB command prompt, you can enter this code:

```
figure;  
plot(q.data, w.data);
```

The figure shows the resulting plot.



Try simulating the model using different starting angles. You can change the starting angle in the **State Targets > Position** menu of the Revolute Joint block dialog box. The figure shows a compound phase plot for starting angles of -240, -180, -120, -60, 0, and 60 degrees.



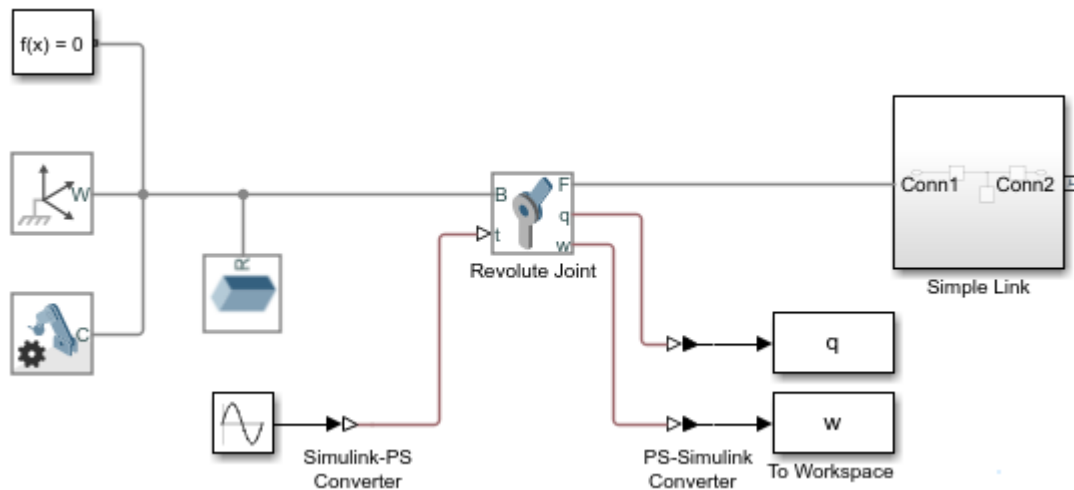
## Analyze Damped and Driven Pendulum

- 1 In the Revolute Joint block dialog box, set **Actuation** > **Torque** to Provided by Input. The block exposes a physical signal input port that you can use to prescribe the joint actuation torque.
- 2 Add these blocks to the model.

Library	Block
Simscape > Utilities	Simulink-PS Converter
Simulink > Sources	Sine Wave

The Sine Wave block provides a periodic torque input as a Simulink signal. The Simulink-PS Converter block converts the Simulink signal to a Simscape physical signal compatible with Simscape Multibody blocks.

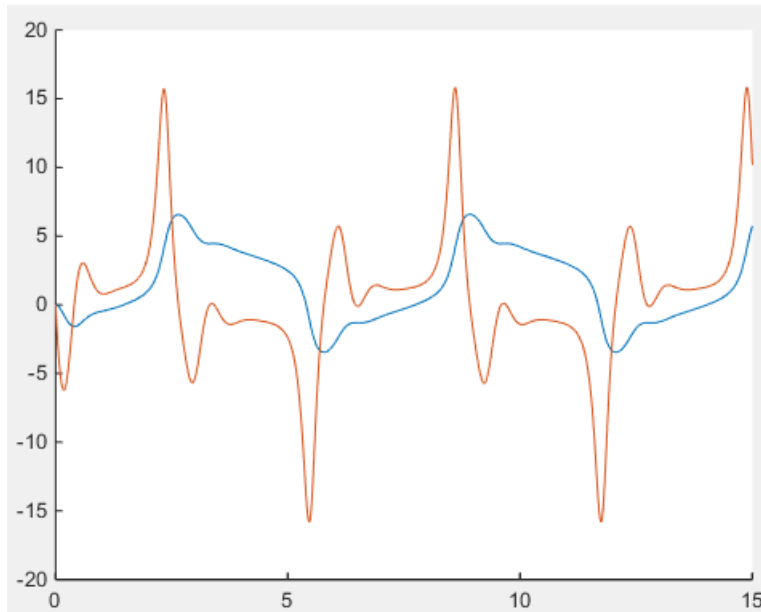
- 3 Connect the blocks as shown in the figure.



- 4 In the Sine Wave block dialog box, set **Amplitude** to  $0.06$ . This amplitude corresponds to an actuation torque oscillating between  $-0.06$  N and  $0.06$  N.
- 5 In the Revolute Joint block dialog box, ensure that **State Targets > Position > Value** is set to  $0$  deg.
- 6 Run the simulation.
- 7 Plot the joint position and velocity with respect to time. To do this, at the MATLAB command prompt, you can enter this code:

```
figure;  
hold on;  
plot(q);  
plot(w);
```

The figure shows the resulting plot.



- 8** Plot the joint phase plot. To do this, at the MATLAB command prompt, you can enter this code:

```
figure;  
plot(q.data, w.data);
```

The figure shows the resulting plot.

