# Simscape™ Multibody™ Release Notes

MATLAB®&SIMULINK®

MathWorks®

How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

Technical support: www.mathworks.com/support/contact_us

Phone: 508-647-7000

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

Contents

R2022a

R2021b

R2021a

R2020b

R2020a

R2019a

R2018b

R2016b

R2016a

R2015b

R2015a

R2014b

R2014a

R2012a

R2011b

Bug Fixes

R2011a

No New Features or Changes

R2010b

No New Features or Changes

R2008a

R2007b

R2007a

R2023a

Version: 7.7

New Features

Bug Fixes

Compatibility Considerations

Flexible Body Model Builder: Generate reduced order model data

Starting in R2023a, you can use the **Flexible Body Model Builder** app to generate reduced order model (ROM) data for flexible bodies. The Reduced Order Flexible Solid block can use the ROM data to model a flexible body.

Reduced Order Flexible Solid Block Enhancements: Display deformed shape of flexible body

To show the reconstructed deformed shape of the flexible body, set the **Type** parameter to **Deformed Geometry**. The block uses the specified triangulation representation and the recovery data to reconstruct the elastic deformations of the vertices on the triangulation. For more information, see Reduced Order Flexible Solid.

Simulink Interface for Siemens MF-Tyre/MF-Swift Tire Model: Use the tire model

You can use the Simulink® Interface for Siemens® MF-Tyre®/MF-Swift Tire Model with Simscape Multibody and other MathWorks® products, such as Simulink. To use the tire model, download and install the Simulink Interface for Siemens MF-Tyre/MF-Swift Tire Model add-on. The add-on works only on Windows®.

Magic Formula Tire Force and Torque Block Enhancements: Use Grid Surface block, variable friction scaling coefficients, and contact frame sensing

You can use the Magic Formula Tire Force and Torque block to:

- Connect to the Grid Surface block that models the base geometry for the tire.
- Model the friction variation for a tire by using the **LMUX** and **LMUY** parameters. To enable the **LMUX** and **LMUY** ports, under the **Scaling Coefficients** section, set **LMUX** and **LMUY** to `Provided by Input`.
- Output the location and orientation of the contact frame with respect to the reference frame of the base geometry or with respect to the follower frame of the block. For more details, see the **Contact Frame** section of the Magic Formula Tire Force and Torque.

Transform Sensor Block Enhancement: Use rotation sequence velocity sensing

You can compute the derivatives of rotation sequence angles for a rotation. Under **Rotation**, select **Rotation Sequence**. For more information, see the **seqd** port of the Transform Sensor block.

simscape.multibody.Rotation class Enhancement: Compute rotation sequence angles using the sequenceAngles method

You can compute a set of rotation sequence angles from a `simscape.multibody.Rotation` object by using the `sequenceAngles` method.

Code Generation Improvements: Generates smaller files for models with flexible bodies

In R2023a, generating code for a Simscape Multibody model that includes flexible bodies generates smaller files.

For example, on a Windows 10, Intel® Xeon® W-2133 CPU @ 3.60GHz test system, if you generate C code for a model that has a flexible plate with a hole, the generated `ModelName_deriv.c` file is 2.22 MB. In R2022b, the generated `ModelName_deriv.c` file is 9.76 MB.

Featured Examples

- "Crane Boom Arm Test Bench" — Learn how to parameterize, test, and compare design cases for a counterbalance valve in a crane boom arm test bench.
- "Using Variable Mode Joints - Drop and Catch Mechanism" — Learn how to model a rotating arm that raises a ball, drops the ball, and catches the ball in one cycle of rotation.

Functionality being removed or changed

KinematicsSolver class no longer supports saving or loading objects
Warns

In R2023a, Simscape Multibody no longer supports saving or loading a `KinematicsSolver` object. If you save a `KinematicsSolver` object to a MAT-file by using the `save` function, MATLAB® issues a warning. The saved `KinematicsSolver` object corresponds to an empty model with no kinematic variables.

When loading a `KinematicsSolver` object from a MAT-file, MATLAB issues a warning and the loaded object corresponds to an empty model with no kinematic variables. All properties of the loaded object have default values except the `ModelName` property which is the string `DefaultEmpty`.

Instead of saving a `KinematicsSolver` object, you can save the commands that create the object to a MATLAB script. To recreate the `KinematicsSolver` object, you can run the saved script.

R2022b

Version: 7.6

New Features

Bug Fixes

Compatibility Considerations

Rigid Transform Block and Blocks with Spherical Primitive: Specify rotation with the quaternion method

Starting in R2022b, you can use the quaternion method to specify rotation in the Rigid Transform block or the initial position target of the spherical primitive for a block such as the 6-DOF Joint, Spherical Joint, or Telescoping Joint block.

The quaternion vector is a 1-by-4 or 4-by-1 vector

$Q = (S\ V)$,

where:

- $S = \cos\left(\dfrac{\theta}{2}\right)$

- $V = [U_x\, U_y\, U_z]\sin\left(\dfrac{\theta}{2}\right)$

- θ is the angle of rotation
- $[U_x, U_y, U_z]$ is the unit vector of the rotational axis.

For any given rotation, there are two quaternions that are negatives of each other, but represent the same rotation. For example, the quaternions [1 0 0 0] and [-1 0 0 0] represent the identity rotation.

Flexible Beam Enhancement: Customize stiffness and inertia properties

Starting in R2022b, you can specify custom stiffness and inertia properties for the flexible beam blocks in the Beams sublibrary. You can model beams with geometric variations or material variations, such as beams with fillets or beams made of orthotropic materials.

To specify custom stiffness and inertia properties for a beam block, under the **Stiffness and Inertia** section, set the **Type** parameter to `Custom`. The stiffness values are specified with respect to a frame that locates at the bending centroid, and the mass values are specified with respect to a frame that locates at the center of mass. The axes of the frames are oriented parallel to the respective axes of the reference frame for the beam.

Flexible Bodies Enhancement: Specify coordinate reduction for flexible bodies

To improve the performance of the modeling of flexible bodies, the blocks in the Flexible Bodies library, such as the Reduced Order Flexible Solid, General Flexible Plate, and General Flexible Beam blocks, support the flexible body coordinate reduction.

To use the method, under **Fidelity** section, set the **Reduction** parameter to `Modally Reduced`. Then specify the **Number of Retained Modes** parameter to your interested mode number. You can set **Number of Retained Modes** to 0 to convert the flexible body to a rigid body.

Transform Sensor Block Enhancement: Sense Euler angles between base and follower frames

Starting in R2022b, you can use the Transform Sensor block to sense Euler angles between the base and follower frames. The rotation angles are about an intermediate frame that is initially coincident with the base frame and changes its orientation after each elemental rotation.

To sense Euler angles, under the **Rotation** section, select the **Rotation Sequence** parameter.

Reduced Order Flexible Solid Block Enhancements: Specify orientations for interface frames

Starting in R2022b, you can specify the orientations for the interface frames in the Reduced Order Flexible Solid block.

To set the orientations of the interface frames, under the **Interface Frames** section, specify the **Orientations** parameter with an N-by-4 matrix of unit quaternions, where N is the number of interface frames.

The **Geometry** section has been removed. To specify **File Name** and **Unit Type** parameters, under the **Graphic** section, specify the **Type** parameter as `Partitioned Geometry`. The **File Name** parameter cannot be empty.

Featured Examples

- Contact Modeling - Ball on Grid Surface — Show a ball and dumbbell rolling down on an L-shaped surface modeled by a Grid Surface block.

Functionality being removed or changed

Behavior change for reading STEP files

Starting in R2022b, the technology used to read STEP files was upgraded. In models created in or before R2019a that contain instances of the Solid block, the frames created from geometric features may be changed or missing. To overcome this issue, manually verify the frames of the File Solid blocks. This change does not affect the models created in R2019b or later.

R2022a

Version: 7.5

New Features

Bug Fixes

Compatibility Considerations

MATLAB Classes for Building and Analyzing a Multibody Model: Programmatically model a multibody system

You can use the new MATLAB classes for Simscape Multibody to model a multibody system. The classes model the various entities of a multibody system. All the classes are in the `simscape.multibody` package.

To programmatically build a multibody model, you first create an object that represents an empty multibody model. Next, create individual components, such as rigid bodies or solids, then add these components to the multibody model. Finally, connect these components using joints or rigid transforms to form the desired structure.

To construct an empty multibody model, use an object of the `simscape.multibody.Multibody` class. To model an individual body, such as a rigid body or solid, use an object of the `simscape.multibody.RigidBody` or `simscape.multibody.Solid` class. To create a rigid transform, use an object of the `simscape.multibody.RigidTransform` class. To model a joint, use an object of a subclass of the `simscape.multibody.Joint` class, such as the `simscape.multibody.RevoluteJoint` or `simscape.multibody.PrismaticJoint` class. To connect the individual components together, use the `connect` or `connectVia` method of the `Multibody` object.

To connect a component to other components, the object of that component must have at least one connector. By default, the objects of the `simscape.multibody.Joint` and `simscape.multibody.RigidTransform` classes have two connectors, the objects of the `simscape.multibody.Solid` class have one connector, and the objects of the `simscape.multibody.Multibody` and `simscape.multibody.RigidBody` classes have zero connectors. To add a connector to the `Multibody` or `RigidBody` object, you can use the `addConnector` method of the corresponding object.

To perform analyses on the multibody system, such as computing the state or visualizing the structure of the multibody system, you need to compile the multibody model using the `compile` method of the `simscape.multibody.Multibody` object. The compilation process evaluates the parts of the model and checks for semantic correctness. If the compilation is successful, the `compile` method returns a `simscape.multibody.CompiledMultibody` object.

You can specify state targets for the joint primitives of the multibody model by using a `simscape.op.OperatingPoint` object. To compute the state of the multibody model, use the `computeState` method of the `CompiledMultibody` object with the specified `OperatingPoint` object. You can get the position and velocity of any joint and the 3-D rigid transformation between two arbitrary frames of the compiled multibody model in the computed state. Moreover, you can use the `visualize` method of the `CompiledMultibody` object to visualize the multibody system with 3-D graphics for the computed state.

Magic Formula Tire Force and Torque Block Enhancements: Extend modeling and sensing capabilities

In this release, the Magic Formula Tire Force and Torque block has three new capabilities:

• Turn slip mode. To enable the turn slip mode, set the **Slip Mode** parameter to `Combined + Turn`.

• New ports that output more intermediate variables. The block has several new output ports that output data related to the pneumatic trail, longitudinal slip, saturated longitudinal slip, slip angle,

saturated slip angle, turn slip, relative longitudinal velocity, relative lateral velocity, longitudinal slip velocity, lateral slip velocity, yaw velocity, camber angle, saturated camber angle, camber velocity, spin velocity, free radius, loaded radius, effective rolling radius, longitudinal and lateral friction coefficients.

- Moving geometry. The geometry of the contact surface no longer has to rigidly connect to the world frame. You can use the Magic Formula Tire Force and Torque block when modeling four-post test-rigs, rolling roads, and other models that have moving contact geometries.

Spatial Contact Force Block Enhancements: More inputs and outputs

In R2022a, the Spatial Contact Force block has new input ports that accept external physical signals for normal and frictional forces and output ports that output sensing data. You can use these new ports to implement custom force laws. The input of the normal force is a scalar that specifies the applied normal force. The block clips any negative values to zero. The input of the frictional force is a 2-by-1 vector that specifies the x and y components of the applied frictional force. The new output ports include:

- **Contact Signal**: Signal that shows the contact status of the base and follower geometries. The geometries are in contact if the value of the signal is 1 or separate if the value of the signal is 0.
- **Relative Normal Velocity**: z-component of the relative velocity between the contact points of the base and follower geometries resolved in the contact frame.
- **Relative Tangential Velocity**: x and y components of the relative velocity between the contact points of the base and follower geometries resolved in the contact frame.
- **Base Rotation**: Rotation matrix that maps vectors in the contact frame to vectors in the reference frame of the base geometry.
- **Base Translation**: Coordinates of the origin of the contact frame resolved in the reference frame of the base geometry.
- **Follower Rotation**: Rotation matrix that maps vectors in the contact frame to vectors in the reference frame of the follower geometry.
- **Follower Translation**: Coordinates of the origin of the contact frame resolved in the reference frame of the follower geometry.

Disk Block: Model 2-D disk geometry

Use the Disk block to model a 2-D disk in the xy-plane. The reference frame is coincident with the origin of the disk. You can model the contact between the disk and many other geometries including an infinite plane, brick, cylinder, ellipsoid, sphere, or convex hull.

To specify the size of the disk, use the **Radius** parameter. The **G** port is a geometry port that represents the disk defined by the block. The **R** port is associated with the reference frame of the generated disk. You can use the reference frame to position and orient the disk.

The Disk block is in the **Simscape > Multibody > Curves and Surfaces** library in the Simulink Library Browser.

Featured Examples

- Creating a Four Bar Multibody Mechanism in MATLAB Example — Show how to construct a four bar mechanism in MATLAB.

- Creating a Multibody with different joints in MATLAB Example — Show how to build a multibody containing various joints in MATLAB.
- Creating a Mobile Robot using a MATLAB App Example — Show how a multibody system can be built using an interactive MATLAB app.
- Creating a Robotic Gripper Multibody in MATLAB Example — Show how to construct a robotic gripper multibody in MATLAB.
- Creating a Simple Pendulum in MATLAB Example — Show how to construct a simple pendulum in MATLAB.
- How to Build a Multibody System in MATLAB Example — Show key concepts and recommended steps for building a multibody system in MATLAB.
- Modeling and Control of a Mars Rover Example — Model a mars rover performing a sample retrieval task. The rover follows a specified path on a rigid terrain surface, stops at the target location, and uses a manipulator to pick and store a sample from the surface.

Functionality being removed or changed

Frame variables of KinematicsSolver class no longer support certain roles in kinematics problem
Errors

In R2022a, the frame variables of the `KinematicsSolver` cannot be specified as initial guesses, the `LinearVelocity` and `AngularVelocity` types of frame variables cannot be specified as targets, and the *x* and *y* components of the variable must be specified simultaneously when using the `Rotation` type frame variable as a target.

If your model uses the frame variables as initial guesses, the model returns a warning and does not load these initial guesses. If your model uses the `LinearVelocity` and `AngularVelocity` types of frame variables as targets, the model returns a warning and does not load these targets. When using the `Rotation` type frame variable as a target, if your model does not specify the *x* and *y* components of the variable simultaneously, the model returns a warning and does not load these targets. To overcome these limitations, you can manually add a 6-DOF Joint block between the base and follower frames, then use the joint variable instead of the frame variable for initial guesses or targets.

R2021b

Version: 7.4

New Features

Bug Fixes

General Flexible Plate Block: Model flexible plates

You can use the General Flexible Plate block to model thin, flat structures that can have small and linear deformations. These deformations include stretch, bending, and shear effects. The block follows the shear deformation Mindlin plate theory and uses the finite element method for its solution.

To specify the geometry of a plate, use the **Midsurface** and **Thickness** parameters. The mid-surface of the plate is in the xy-plane and the thickness is symmetrically along the z-axis. The plate is assumed to be made of a homogeneous, isotropic, and linearly elastic material. The block supports two damping methods to control the performance of the modeling. To add custom frames on the plate, specify parameters in the **Frames** section.

The General Flexible Plate block is located in the **Simscape > Multibody > Body Elements > Flexible Bodies > Plates and Shells** library in the Simulink Library Browser.

Property Inspector: Set the parameters of blocks that do not use the visualization pane

In R2021b, all Simscape Multibody blocks that do not use the visualization pane open the parameters in the Property Inspector instead of the Block Parameters dialog box. To view and set the parameter for a block, double-click the block to open the Property Inspector, click the **Settings** tab, and modify the values of the parameters.

By default, changing a parameter immediately applies the new parameter values. To remove a series of changes, use the Undo button and Redo button in the upper-right corner of the Simulink Editor canvas. To manually apply parameter changes, clear the **Auto Apply** check box in the upper-right corner of the Property Inspector. Clearing the **Auto Apply** check box enables **Reset** and **Apply** buttons in the lower-right corner of the Property Inspector, which you can use to reset and apply parameter changes.

To view the description of a block, click **Description** tab in the Property Inspector. To view the documentation for a block, click the Question Mark button in the upper-right corner of the Property Inspector.

Magic Formula Tire Force and Torque Block: Model Magic Formula tires

The Magic Formula Tire Force and Torque block is an implementation of the Pacejka Magic Formula tire model. The tire is assumed to be a disk that lies in the xy-plane of the follower frame of the block. The block applies force and torque to the follower frame in accordance with the Magic Formula equations and follows the ISO sign convention.

To specify the properties of the tire model, use the **Tire Parameters** parameter. The input of this parameter needs to be a structure array that is returned from the `simscape.multibody.tirread` function. To specify which side of the vehicle the tire is mounted to, use the **Tire Side** parameter. The **F** port of the block is a frame port that represents the tire, and the **B** port is a geometry port that represents the surface that the tire contacts. The block only supports infinite plane as the geometry.

The Magic Formula Tire Force and Torque block is located in the **Simscape > Multibody > Forces and Torques** library in the Simulink Library Browser.

simscape.multibody.tirread Function: Read tire properties files

Use the `simscape.multibody.tirread` function to read the parameters in a tire properties file and return the parameters in a scalar structure array. The returned structure array can be used to specify the **Tire Parameters** parameter of the Magic Formula Tire Force and Torque block. The input argument of the function is a tire properties file name, specified as a character vector or scalar string. The file name must end with `.tir` or `.TIR`.

Grid Surface Block: Model grid surfaces for contact

The Grid Surface block lets you model a gridded height field surface. The grid surface can be used to model contacts with point and point cloud geometries.

You can define the surface by using a rectangular grid in the xy-plane and corresponding z values associated with the heights of grid points.

The **G** port of the block is a geometry port that represents the surface defined by the block. The **R** port is associated with the reference frame of the generated surface. You can use the reference frame to position and orient the surface.

The Grid Surface block is located in the **Simscape > Multibody > Curves and Surfaces** library in the Simulink Library Browser.

Featured Examples

- Package Delivery Quadcopter Example — Model a quadcopter that lifts off from the launchpad, flies along a desired trajectory, and releases a package at the drop-off location. The example uses the joint disengagement capability and Spatial Contact Force block to model the package release and the contact between the package and the ground surface.
- Vehicle Dynamics - Car with Heave and Roll Example — Model the vehicle dynamics using a vehicle model that has 14 degrees of freedom. The tires are modeled by the Magic Formula Tire Force and Torque block, and you can plot the forces and torques at the contact patch from the simulation results.

R2021a

Version: 7.3

New Features

Bug Fixes

Point Cloud Block: Create a point cloud for sustained and distributed contact modeling

The Point Cloud block in the Curves and Surfaces library lets you approximate a geometry, such as a solid or a convex hull, as a point cloud. A point cloud can improve the performance of the Spatial Contact Force block when modeling sustained and distributed contact problems.

The Point Cloud block has one reference frame port and one geometry port. To use a Point Cloud block for contact modeling, connect its geometry port to the base or follower geometry port of a Spatial Contact Force block. The Spatial Contact Force block treats a point cloud as an aggregation of N-point geometries and applies contact forces to each point independently. Each contact force is based on the penetration and velocity of the individual point of the cloud. Note that the Spatial Contact Force block doesn't support sensing when connected to a Point Cloud block.

Enter an N-by-3 matrix for the **Coordinates Matrix** parameter of the Point Cloud block to specify a point cloud. Each row of the matrix specifies the Cartesian coordinates of a point with respect to the reference frame of the Point Cloud block. An error occurs if the matrix has any repeated rows.

Constant Velocity Joint Block Enhancement: Use rotation sequence parameterization

The Constant Velocity Joint block now supports two different internal state parameterizations: `Rotation Sequence (faster simulations)` and `Quaternion (allows zero bend angle)`. Set **Internal State** to `Rotation Sequence (faster simulations)` to use the *Z-Y-Z* rotation sequence parameterization. This option generally leads to faster simulations and is recommended for most models. However, the *Z-Y-Z* rotation sequence parameterization has a kinematic singularity when the bend angle is zero. If a zero bend angle is required, set **Internal State** to `Quaternion (allows zero bend angle)`. For more information, see Constant Velocity Joint.

Multibody Interface Blocks: Connect Simscape Multibody joints to Simscape networks

Use the Rotational Multibody Interface and Translational Multibody Interface blocks to connect Simscape Multibody joints to Simscape networks in the mechanical rotational and translational domains. The interface blocks allow you to easily model multidomain systems in which Simscape networks with mechanical, fluid, and electrical components drive 3-D mechanical systems modeled in Simscape Multibody.

To connect Simscape Multibody joints to the Simscape networks, follow the steps in How to Use Interface Blocks. If the behavior of a Simscape block depends on position or angle, follow the steps in How to Pass Position Information.

Point on Surface Constraint Block: Confine a point to an infinite plane

The Point on Surface Constraint block in the Constraints library lets you confine a point to a 2-D surface. The point moves on the surface and has five degrees of freedom (three rotational and two translational) with respect to the reference frame of the 2-D surface.

The block has two ports: **F** and **B**. Port **F** is a frame port whose origin defines the point. Port **B** is a geometry port and should be connected to the geometry of a 2-D surface. Currently, the Point on Surface Constraint block supports only the Infinite Plane block.

You can sense the constraint forces and torques between the point and 2-D surface. The forces and torques are resolved in the frame that has been set in the **Resolution Frame** parameter. You can set **Resolution Frame** to `Base` or `Follower`. `Base` and `Follower` correspond to the reference frame of the connected surface geometry and follower frame, respectively.

Point on Curve Constraint Block Enhancement: Sense the constraint torques

Under **Constraint Force/Torque Sensing**, you can now select the **Torque Vector** parameter to sense the constraint torques between the point and curve. You can set **Resolution Frame** to `Base` or `Follower`. The `Base` and `Follower` correspond to the reference frame of the connected curve geometry and follower frame respectively.

Mechanics Explorer Enhancement: Use accelerator mode during simulations

Mechanics Explorer now supports accelerator mode. For information about the accelerator mode, see How Acceleration Modes Work. To use accelerator mode, in the Simulink Editor, on the **Simulation** tab, choose `Accelerator` from the drop-down list.

Featured Examples

New examples in this release include:

- Forklift Example — Model a forklift that uses hydraulic and pulley mechanisms to lift.
- Perform Forward and Inverse Kinematics on a Five-Bar Robot Example — Model a five-bar planar parallel robot. The `KinematicsSolver` class is used to compute forward and inverse kinematics for the motion planning and control of the robot.

R2020b

Version: 7.2

New Features

Bug Fixes

Flexible Beams Enhancement: Create custom frames

All the flexible beam blocks in the Beams library support custom frame creation. You can specify the origin and orientation of each frame based on:

- the geometric features such as points, lines, and surfaces;
- the reference frame origin and/or axes of the undeformed geometry;

Joint Enhancement: Change joint mode from disengaged to normal during simulation

You can change the mode of all the joints from disengaged to normal by changing the input signal of the **mode** port from -1 to 0. To enable the **mode** port, in the block dialog box, under the **Mode Configuration** section, select **Provided by Input**. The joint mode can be changed between normal and disengaged many times during a simulation.

Infinite Plane Block: Create an infinite plane for contact

The Infinite Plane block in the Curves and Surfaces library lets you create a 2-D infinite plane. The infinite plane can be used to model contacts with other types of geometries, such as points, brick solids, and convex hulls.

Point Block: Create a point for contact

The Point block in the Curves and Surfaces library lets you create a 0-D point geometry. The point geometry can be used to model contacts with other types of geometries, such as brick solids, planes, and convex hulls.

Spatial Contact Force Block Enhancement: Use revolved solids for contact

The Spatial Contact Force block supports contacts between revolved solids and other types of geometries, such as points, planes, brick solids, and convex hulls.

KinematicsSolver Enhancement: Specify maximum number of solver iterations

The `KinematicsSolver` class has a new property—`MaxIterations`—that specifies the maximum number of solver iterations.

Featured Examples

New examples in this release include:

- Cartesian 3D Printer example — Model a Cartesian 3D printing system.
- Pick and Place Robot Using Forward and Inverse Kinematics example — Model a pick-and-place delta robot. The `KinematicsSolver` class is used to compute forward and inverse kinematics for the motion planning and control of the robot.

- Ratchet Lifter example — Model a ratchet lifter system that demonstrates how to use contact proxies to simulate contacts that involve complex geometries.

R2020a

Version: 7.1

New Features

Bug Fixes

Compatibility Considerations

Flexible Solid Enhancements: Use proportional and uniform modal damping models with flexible body blocks

Use proportional damping and uniform modal damping models to specify the damping properties for the Reduced Order Flexible Solid block and all the flexible beam blocks in the Beams sublibrary.

The proportional damping model is an extension of the previous stiffness-proportional damping model. To specify the damping matrix, you can now specify the mass matrix coefficient **alpha** in addition to the stiffness matrix coefficient **beta**. With the uniform modal damping model, you can apply a single damping ratio to all the deformation modes of a flexible body. To select these damping models, change the value of the **Type** parameter in the **Damping** section of the dialog box for a flexible body block.

General Flexible Beam Enhancements: Model flexible beams that have arbitrary cross-sections with holes

You can now use the General Flexible Beam block to model flexible beams that have arbitrary cross-sections with holes. Input a cell array for the **Cross-section** parameter in the **Geometry** section of the General Flexible Beam dialog box. The coordinates in the first cell specify the outer boundary of the beam cross-section. The coordinates in the other cells specify the shapes and locations of holes in the beam cross-section.

New blocks for flexible beams with standard cross-section

Six blocks have been added to simplify the modeling process for flexible beams with standard cross-sections: Flexible Angle Beam, Flexible Channel Beam, Flexible Cylindrical Beam, Flexible I Beam, Flexible Rectangular Beam, and Flexible T Beam. Additionally, the **Hollow** option is available for the rectangular and cylindrical beam blocks. These standard cross-section flexible beam blocks work similarly to the General Flexible Beam block. To use these blocks, in the Simulink Library Browser, click **Simscape** > **Multibody** > **Body Elements** > **Flexible Bodies** > **Beams**.

KinematicsSolver Enhancements: Perform velocity-based kinematic analyses using the KinematicsSolver

You can now use the `KinematicsSolver` to perform kinematic analyses that depend on velocities as well as positions. For example, you can specify the angular or linear velocity targets of the end-effector frame of a robotic arm and then output the velocities of any joints and frames on the arm.

The velocity-based kinematic analysis procedure is similar to the position-based analysis procedure. Velocity-based relationships are expressed using kinematic variables, such as joint and frame variables. The velocity-based joint variables are generated automatically during the construction of a KinematicsSolver object. To view them, use the `jointVelocityVariables` function. To create velocity-based frame variables, set the `type` of `addFrameVariables` to `AngularVelocity` or `LinearVelocity`.

Compatibility Considerations

The `jointVariables` function is not recommended starting in R2020a and the use of this function causes a warning. It will be removed in a future release. To list the position-based and velocity-based

joint variables, use the `jointPositionVariables` and `jointVelocityVariables` functions instead.

Kinematics Solver Viewer: Use the Kinematics Solver Viewer for kinematic analyses

The new Simscape Multibody Kinematics Solver Viewer makes kinematic analyses more intuitive. Use the viewer to observe the mechanism that corresponds to the last successfully solved kinematic analysis. Use the `viewSolution` and `closeViewer` functions of the `KinematicsSolver` class to open and close the viewer.

Spatial Contact Force Enhancements: Apply contacts to the file and extruded solid blocks and simulate the contacts with zero-crossings functionality

You can now use the Spatial Contact Force block to model the contacts between true geometry and convex hull geometry. The Spatial Contact Force block supports all the solid blocks in the Body Elements library except the Revolved Solid block.

The Spatial Contact Force block now supports zero-crossings functionality. The zero-crossings function indicates contacts between a geometry pair by detecting the discontinuity of a simulation. Moreover, this function searches for the precise crossing times that show the start and end of each contact. To enable this option, in the Spatial Contact Force dialog box, under the **Zero-Crossings** section, select the **Detect Contact Start and End**.

Body Element Enhancements: Use File Solid and Extruded Solid blocks to export convex hull geometries

Use the File Solid and Extruded Solid blocks to export convex hull geometric representations, which are the approximations of their real geometries. In the block dialog boxes, under the **Geometry** section, select the **Convex Hull**.

Note that the Extruded Solid block can also export the real geometry of an extrusion. In the Extruded Solid block dialog box, under the **Geometry** section, select the **Extrusion Type > Regular**, and then select the **Export > Entire Geometry**.

Simscape Multibody First-Generation End of Support: Simscape Multibody no longer supports the use of first-generation functions and blocks

Compatibility Considerations

All the first-generation functions and blocks are not supported since this release. Use the corresponding second-generation functions and blocks instead.

Featured Examples

New examples in this release include:

- Elevator — Model a cable-driven elevator system.
- Cable Driven Space Manipulator — Model a cable-driven space manipulator.
- Train a Humanoid Walker— Model a humanoid walking robot.

R2019b

Version: 7.0

New Features

Bug Fixes

Compatibility Considerations

Reduced-Order Flexible Solid Block: Model deformation in bodies of diverse geometries

The new Reduced Order Flexible Solid block, in the Body Elements library, lets you model flexible bodies of arbitrary geometry, such as the swept wing of a plane or the streamlined body of a car. This block expands on the General Flexible Beam block, which was introduced in R2018b for slender extrusions.

The new block uses the mass, stiffness, and damping matrices of a solid to capture small elastic deformations. The matrices derive from a reduced-order model, the sort obtained by applying the Craig-Bampton method to a finite-element model. They are typically obtained using finite-element analysis software.

There is no limit on the number of frames in a flexible solid. There can be one or several—one each per interface node of the reduced model. The node coordinates determine the locations of the solid frames, specified in the block with respect to a common frame—one aligned with the reference frame of the part's finite-element mesh.

The damping matrix can be specified explicitly if it is available, computed from a simple proportional model if not, or ignored altogether. Visualization is simple, with the solid partitioned into sections free to deflect relative to each other. The undeformed solid geometry comes from a CAD file in any of the formats supported in the File Solid block.

Solid Block Split: Model solids of different shapes using blocks specific to shape type

The Solid block of previous releases has split into simpler blocks specific to a geometry type each. You can search for them as Brick Solid, Cylindrical Solid, Spherical Solid, Ellipsoidal Solid, Extruded Solid, and Revolved Solid. File Solid, introduced in R2018b, already covers shapes imported from CAD files.

The new blocks look and work much like the original Solid block. If the original block is familiar to you, then the new blocks will be too. There are geometry, inertia, and color parameters to specify. There is an interactive frame creator for solids that need custom frames based on geometry features. There is a visualization pane, to check that geometry, color, and frames are correct before saving any changes to such attributes.

The block split reflects in the Simscape Multibody template—opened by typing `smnew` at the MATLAB command prompt—and in models from older versions. The Solid block of the template is now a Brick Solid block. The Solid blocks of older models are now the various simpler blocks—whichever best captures the solid geometry type.

The Brick Solid, Cylindrical Solid, and Spherical Solid blocks each have a geometry port. The port is optional and by default hidden. Expose it by expanding the **Geometry > Export** node in the block dialog box, then checking the **Entire Geometry** option.

The port serves to export the surface or bounding box of a solid for use in contact modeling. Two ports of this sort feature on the Spatial Contact Force block now in the Forces and Torques library. Connect them solid to force to solid to enforce contact during simulation.

Compatibility Considerations

The original Solid block no longer exists. Use the new solid blocks instead. Older models should open with the new blocks in place, but references to the **Shape** parameter of the original block—a parameter that has no counterpart in the new blocks—may cause simulation to fail. Such references are possible in MATLAB scripts with `get_param` and `set_param` functions given `GeometryShape` for its `ParameterName` argument. Update or remove such references, if they exist, to avoid errors in your model.

Run-Time Parameter Updates: Set the parameters of General Flexible Beam blocks at run time

The General Flexible Beam block now supports run-time parameters. These are parameters that you can set before or between simulations without forcing the model to recompile. Use them to more quickly start simulation when iterating parameter values.

You must explicitly set a parameter as run-time in order to use it as such. This requires that you select from a drop-down list the option of Run-time. The drop-down list is always in line with the block parameter, adjacent to the units box or, in the case of unitless parameters, to the value box.

If you do not see the drop-down list, check that your Simscape preferences are set to show run-time parameter settings. Not all numerical parameters can be set at run time. If you've enabled run-time parameters but the drop-down list is inactive or missing, then you cannot set that parameter at run time.

Block parameters are by default `Compile-time`—even those that you can later change to `Run-time`. Changes made to `Compile-time` parameters reflect in the model only after recompile. This is the legacy behavior to which you may have grown accustomed when working with General Flexible Beam blocks.

Joint Disengagement: Release the locked axes of joints for unrestricted motion of the coupled bodies

Release the kinematic constraints of weld and other joints for unrestricted motion between the originally coupled bodies—to capture the dropping of a load from a backhoe bucket, for example, the snapping of a shear bolt in a flight control linkage, or the unlocking of a coupler between train cars.

A **mode** port new to the joint blocks controls the disengagement. The port takes a physical signal as its input. At `0`, the joint is active and its constraints all in effect. The bodies move only as permitted by joint type—pivoting about a revolute hinge or a spherical fulcrum or sliding down a prismatic track, for example. At a signal of `-1`, the joint disengages and the bodies are released, their motions to be constrained perhaps by other components in the model but no longer by the broken joint.

The disengagement is permanent. The **mode** signal controls when disengagement occurs, but once it does there is no reversing it.

The port is optional and it is by default hidden. To expose it, in the **Mode Configuration** area of the joint block dialog box, set the **Mode** parameter to **Provided by Input**. You can also leave the **Mode** parameter in its `Normal` setting for a joint that never breaks, or change it to `Disengaged` for a joint that begins simulation broken.

A disengaged joint impacts more than the mobility of bodies. Actuation inputs are ignored in all broken joints. Joint limits and internal springs and dampers are too. Force or torque and motion sensing outputs are set to zero.

Contact Forces: Model contact between bodies forced together during simulation

Model the striking of bodies moving on converging paths—a box falling on a conveyor, a foot landing on a step, or a gripper tightening on a target.

A new Spatial Contact Force block monitors the gap between two solid surfaces or their bounding boxes. If a surface or box impinges on the other, a repulsive force pair emerges to stop penetration and keep the solids apart.

The surface geometries come from Solid blocks. These now have a geometry port each—the same port type featured on the new force block. Connect them from solid to force to solid to pass the surface geometries to the force block for computation. The port is by default hidden in the Solid blocks, so you must first expose it. To do this, in the block dialog box, select the **Geometry > Export + Entire Geometry** check box.

The force has two components, one tangent to the surfaces at the point of contact, the other perpendicular to them. If contact is between two edges or two vertices or between one edge and one vertex, the tangent and perpendicular directions are estimated from the surfaces that terminate there.

The tangent force is a smooth friction force. The perpendicular force is the normal force, that resistance to penetration most strongly associated with contact. It is modeled as a spring damper, though one kept dormant until surfaces collide. A similar force model is used in joint limits. The stiffer the spring, the harder the contact. The more viscous the damper, the sharper the energy loss on contact, and the faster the decay in contact oscillations if any arise.

KinematicsSolver Objects: Save and Load KinematicsSolver objects in MAT files

You can now save and load `KinematicsSolver` objects in MAT files—for example, to finish analysis begun in earlier MATLAB sessions or to reuse the object in multibody control design. MAT files with `KinematicsSolver` objects contain all the data for those objects, including their frames and joint, frame, target, and initial guess variables.

Video Recorder Enhancement: Specify which tile of a split visualization pane to record

When recording animations from split visualization panes—those with two or more tiles—you can now specify which of the tiles to record. To this end, the smwritevideo function gets a new name-value pair argument, `tile`. Set the value to the number of the tile to record—`1`, say. The tiles are numbered first

top to bottom and then left to right. In a split screen with four identical tiles, the top left tile is numbered 1, the bottom left tile is 2, the top right tile is 3, and the bottom right tile is 4.

In older versions, the active tile, or that enveloped in red highlight, is automatically set as the recorded tile in `smwritevideo`. Note that this remains the behavior of the Video Recorder UI (**Tools > Video Recorder** in Mechanics Explorer) where the active tile is always the recorded tile. Click the desired tile to make it the active tile if recording with Video Creator.

Inertia Sensor Block: Measure inertial properties of a group of rigidly connected body elements or an entire mechanism

The new Inertia Sensor block lets you measure the instantaneous inertial properties of a collection of body elements with respect to a chosen measurement frame.

The block computes the mass, center of mass, and inertia matrix, among other properties, for a target at frame port **S**. The target can be a group of rigidly connected body elements or an entire mechanism. Set the **Sensor Extent** block parameter to `Body Group` for rigidly connected body elements, or set it to `Mechanism` for the whole of the Simscape Multibody network connected to port **S**.

In a model, a body group is a collection of solids that may be rigid or flexible, but whose connections are always rigid. Those connections can be through Rigid Transform blocks or direct frame connection lines.

The measurements come as physical signals at optional ports, and these are by default hidden until you enable them. Do so by checking the box for each parameter you want to sense—**Center of Mass** for port **com**, say, or **Principal Inertia Matrix** for port **Ip**. You can use the signals as inputs to Simscape blocks or pass them through PS-Simulink Converter blocks to feed them to Simulink blocks like Scope, PID Controller, or To Workspace.

All measurements, except for mass and principal inertia matrix, are dependent on the choice of measurement frame. Set the **Measurement Frame** block parameter to `Attached` to use the frame at port **S**, to `World` to resolve it in the world frame of the model, or to `Custom` to resolve it in the frame you choose to connect to frame port **M**—an optional port that shows when you select `Custom`.

A principal inertia frame marker is shown by default during simulation. Follow it to track the motion of the target. Use the **Graphic > Type** block parameter to replace the frame marker with the equivalent inertia ellipsoid, a special ellipsoid with radii on the principal axes of the target and scaled in proportion to the principal moments of inertia. You can also disable the visualization marker altogether.

The figure shows the principal inertia frame (black) of the planets of the solar system and their equivalent inertia ellipsoid (gray).
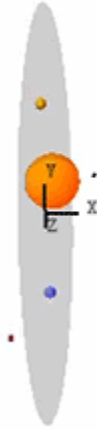
smimport Joint Limit Support: Import joint limits of URDF and RigidBodyTree models

You can now translate the joint position limits of URDF and `rigidBodyTree` models into their equivalent settings in Simscape Multibody joint blocks.

The `smimport` function now extracts the position limits of prismatic and revolute joints (where available) and reproduces them in the joint blocks that represent them. The block options **Specify Lower Limit** and **Specify Upper Limit** are enabled, where necessary, to specify the **Bound** parameters extracted from the source model.

Note that joint limits in CAD models are still discarded during import. You must reproduce them manually if you want to capture them in your simulation results. You can do so by enabling lower and upper limits in joint blocks and setting their bounds to the values in the original models.

Compatibility Considerations

In previous releases, the `smimport` function discarded joint limits. If you have a legacy model imported from a `rigidBodyTree` object or URDF file, reimporting it in R2019b may produce different simulation results.

Pulley Initial Conditions: Specify the minimum initial wrap angle of a cord around a pulley

The Pulley block now has an **Initial Wrap Angle > Lower Bound** parameter. Use it to specify the minimum wrap angle of a pulley at the start of simulation. The wrap angle is the angle of the arc drawn between the points at which the cord meets the pulley.

R2019a

Version: 6.1

New Features

Bug Fixes

Motion Limits in Joint Blocks: Limit joint travel to specified bounds using contact forces

Use limits to curb travel at a joint. Joint primitives now have two limits each—one for the bottom of the motion range and one for the top. You can enable both limits, just one, or, for unrestricted motion in either direction, none.

A joint primitive, if the term seems foreign to you, is the basic constituent of a joint. Think of it as the simplest joint of some kind—prismatic for translation along an axis, revolute for rotation about an axis, or spherical for rotation about a point. All joints are compounds of such primitives.

The limits are each modeled as a spring-damper. The spring constant sets the stiffness of the stop. The larger the constant, the harder the stop, and the faster the bounce—or contact oscillations, if any arise. A damping coefficient adds viscous losses that gradually lessen the oscillations or, if the coefficient is large enough, keep them from forming altogether.
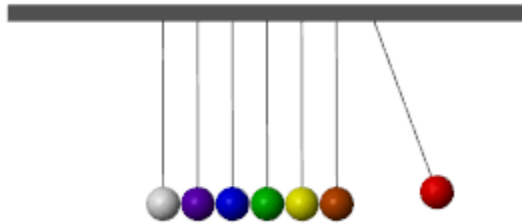
Newton's Cradle with Contact Modeled with Joint Stops

Unlike other spring-dampers, a limit is active only on contact. Between the lower and upper limits, the spring-damper force is zero, and, pending other constraints in the model, the joint primitive has freedom of movement. It is only in a small region past the location of a limit, below the lower bound or above the upper bound, that a spring-damper exerts its influence. There, the contact force rapidly increases until motion is either reversed or stopped.

The location of a limit is defined in terms of joint frames. Consider it the offset from base to follower frame, as measured in the base frame, at which contact begins. That offset is a distance along an axis in prismatic primitives, an angle about an axis in revolute primitives, and an angle between two axes in spherical primitives.
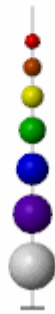
Galilean Canon with Contact Modeled with Joint Stops

Because of its sudden onset, contact has the potential to slow down simulation. The harder the stop, the faster the deceleration, and the smaller the time step required of the solver. Consider softening your joint limits if necessary to speed up simulation.

KinematicsSolver class: Run inverse, forward, and mixed kinematic analysis

Create a `KinematicsSolver` object and use the methods of its class to run a kinematic analysis on a multibody model. The analysis can be the standard forward or inverse kinematics or it can be another type. It can run from a MATLAB script independent of a model or from a MATLAB Function block incorporated in a simulation—for example, to generate set points for joint controllers tasked with moving an end effector along a certain path.
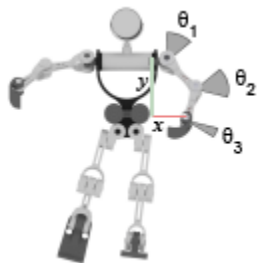
The type of the analysis depends solely on the variables in the object and on their roles in the analysis. All variables derive from joints (**J** in the figure) or arbitrary frame pairs (**F**). Joint and frame variables can serve as targets (**T**) to aim for during analysis, as guesses (**G**) to bias the solver toward

a particular solution when several exist, and as outputs (**O**) to be calculated from the knowns of the problem.
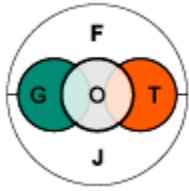
Variables of a KinematicsSolver object

The task of setting up a kinematic analysis is largely one of adding variables to the `KinematicsSolver` object and assigning them to suitable roles. The object functions include those needed for this task. You can remove variables a few at a time when they are no longer necessary or all at once to start over with a new kinematic problem. A `solve` method runs the analysis when ready. A `generateCode` function creates a C version of the solver to run the analysis outside of MATLAB or, alternatively, to more rapidly run it from within a MATLAB Function block.

Run-Time Parameter Updates: Set belt-cable, constraint, gear, and variable-mass block parameters as run-time

Belt-cable, constraint, gear, and variable-mass blocks now support run-time parameters. These are numerical parameters whose values you can set at run time—before or between simulations—without forcing the model to recompile. Use them to more quickly start simulation when iterating parameter values.

You must explicitly set a parameter as run-time in order to use it as such. This requires that you select from a drop-down list the option of `Run-time`. The drop-down list is always in line with the block parameter, adjacent to the units box or, in the case of unitless parameters, to the value box.

If you do not see the drop-down list, check that your Simscape preferences are set to show run-time parameter settings. Not all numerical parameters can be set at run time. If the drop-down list is inactive—grayed out and unresponsive to mouse clicks—then you can set that parameter at compile time only.

All parameters are by default `Compile-time`—even those that you can later change to `Run-time`. Changes made to `Compile-time` parameters reflect in the model only after recompile. This is the legacy behavior to which you may have grown accustomed when working with belt-cable, gear, and constraint blocks.

smimport function update: Import RigidBodyTree objects from Robotics System Toolbox models

Use the `smimport` function to recreate a Robotics System Toolbox model in the Simscape Multibody environment. The function now accepts `RigidBodyTree` objects—each a representation of a

multibody assembly—as source for import. Consider the objects an alternative to the multibody description files commonly named in the `smimport` function to import CAD, URDF, and other models.

A similar function, named `importrobot` and native to the Robotics System Toolbox environment, allows the import to proceed in reverse—from Simscape Multibody to Robotics System Toolbox software.

`RigidBodyTree` objects, if the class is new to you, can have `RigidBody` and `Joint` child objects. `RigidBody` objects translate into Simulink subsystems. These comprise Solid, Inertia, and Rigid Transform blocks. Solid blocks set geometry and color, Inertia blocks set mass properties, and Rigid Transform blocks set (among other things) the placement of joints.

`Joint` objects translate into joint blocks—normally Revolute Joint, Prismatic Joint, or Weld Joint. Home position, a property of `Joint` objects, translates into position targets in joint blocks. Position limits are ignored during import. You can, however, set these in Joint blocks after the import is complete. Joint position limits are new for R2019a.
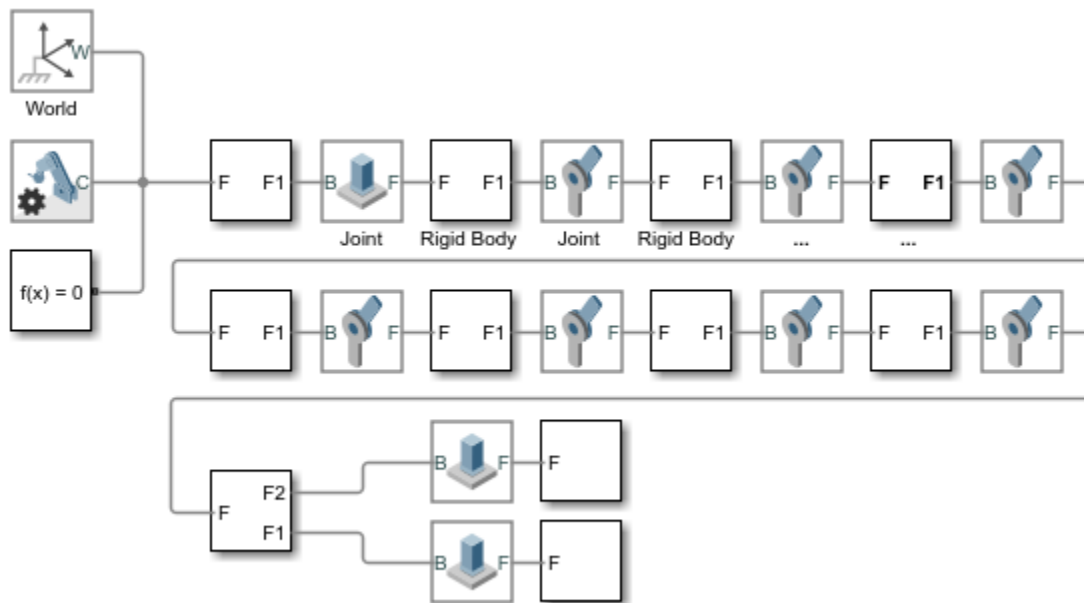
Block diagram for iiwa14 robot imported from rigidBodyTree object

Consider the LBR iiwa serial manipulator (a system by KUKA Robotics). You can find a URDF model for this system in your Robotics System Toolbox installation. Import the URDF model into the Robotics System Toolbox environment:

```
iiwaRST = importrobot('iiwa14.urdf');
```

`iiwaRST` is a `RigidBodyTree` object with `RigidBody` and `Joint` child objects. Import the `RigidBodyTree` object into the Simscape Multibody environment:

```
iiwaSM = smimport(iiwaRST);
```

`iiwaSM` is a Simscape Multibody block diagram with Simulink subsystems for rigid bodies and Joint blocks for their degrees of freedom. Update the Simscape Multibody block diagram and visualize the robot in Mechanics Explorer.
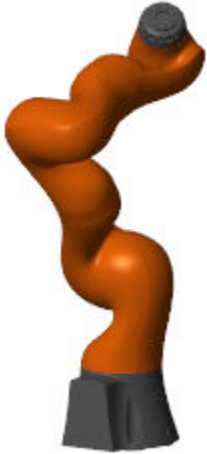
Note that you can import the URDF model into Simscape Multibody directly:

```
smimport('iiwa14.urdf')
```

R2018b

Version: 6.0

New Features

Bug Fixes

General Flexible Beam Block: Model flexible beams with general cross sections

Capture the elastic deformation of a slender body under a dynamic load. The new General Flexible Beam block (**Body Elements > Flexible Bodies > Beams**) uses material properties such as Young's modulus of elasticity and Poisson's ratio to determine the deflections of its connection frames.

Depending on its loading and boundary conditions, the beam can stretch, bend, and twist. The deformations are based on the classical theories of Euler-Bernoulli and Saint Venant. They are assumed to be small in magnitude and to conform to the simplifying assumptions of the theories. Dynamic deformations—such as oscillations—are damped using a linear model in which the damping matrix is proportional to the stiffness matrix by a constant factor.

The beam is modeled as a general extrusion. As in the Solid block, the cross section of the extrusion is obtained from a coordinate matrix specified in the xy plane. The cross section can have any shape—that of a tee, channel, or angle beam, for example. Holes, as specified in the Solid block, should be avoided. For accurate results, the beam should be slender, and therefore much greater in length than in either height or width.

For an example based on the new block, at the MATLAB command prompt, enter `smdoc_flexible_cantilever_channel`. The example uses an asymmetrical cross section for the beam and simulates its various deformations under a point load, an applied moment, or both. The degree to which the beam bends and twists varies with the point of application of the force. The figure shows the visualization results for the default model configuration.
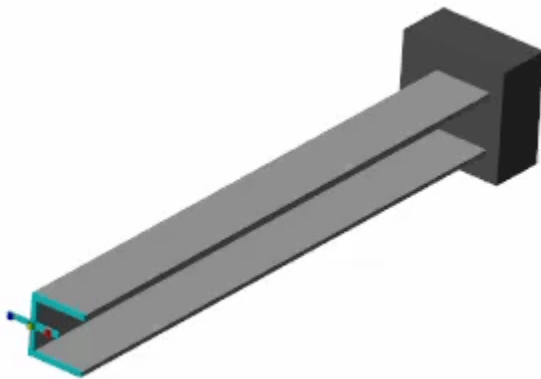
File Solid Block: Define parts by directly referencing CAD files

Import a solid element from a variety of CAD files. The new File Solid block parses the data in a selected file to generate the solid geometry and—given either mass or density—set its mass properties. As with the Solid block, you can manipulate color and interactively create frames for connection in a model. A visualization pane shows the imported solid and any changes made to it in the block.

Among the supported formats are those native to CATIA (V4, V5, and V6), Creo, Inventor, Unigraphics NX, Solid Edge, SolidWorks, and Parasolid (all CAD applications common in industry and academia).

These include CATPART, PRT, IPT, SLDPRT, and X_T (and its binary version, X_B). Other valid formats, not associated with a specific application but common in 3-D modeling, include SAT (often referred to as ACIS), JT, STL, and STEP. The figure shows an example of a CAD part imported into the block.
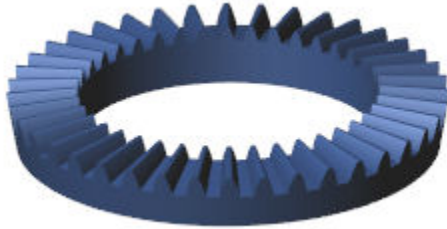
Nonplanar Belt and Cable Mechanisms: Model spools and pulleys with non-zero fleet angle

A cord can now enter the rotation plane of a drum at an angle. The out-of-plane angle, denoted *qfa* in the figure, is named here the *fleet angle*. A spool, which meets the cord at one point, has one fleet angle. A pulley, which meets the cord at two points, has two. These can differ in value. The span of cord wrapped on a drum is always in the rotation plane of that drum.

The Fleet Angle

This easing of the constraint on cord alignment allows for more authentic drum behavior. Because of the width of a spool, a true cord generally winds and unwinds at an angle. (The term ``fleet angle'' commonly refers to the maximum value of that angle.) Likewise, the geometry of a pulley block, where pulleys often sit on separate planes, demands that the cord meet each pulley at a slight angle.

Even what is normally a planar assembly can momentarily be forced out of strict alignment—by a sudden motion, for example, like that of the jib of a tower crane.
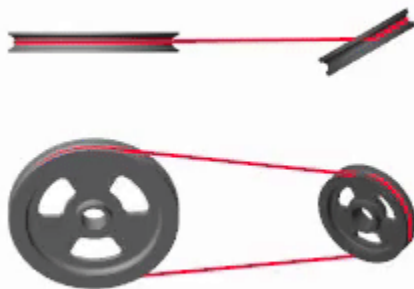
Fleet Angle in Pulley Simulation (Exaggerated for Clarity)

For cases in which the fleet angle must be zero, the Belt-Cable Properties block provides a means to enforce a planar alignment constraint. To do this, set the **Drum Belt-Cable Alignment** block parameter to `Monitored Planar`. The cord must then meet each drum in its plane of rotation—with a zero fleet angle—or the simulation stops with an error. The default setting is `Unrestricted`.

Note that the planar constraint applies to individual drums only. The assembly as a whole can still be nonplanar. Because a cord can simultaneously lie on separate (intersecting) rotation planes, it is possible to satisfy the planar constraint individually for each drum of a nonplanar assembly. See, for instance, the Pulley Mechanism Right Angle Drive featured example. (To open the example from the MATLAB command line, enter `sm_cable_drive_right_angle`.)

Spool and Pulley Kinematic Sensing: Measure wrap angles, pulley angles, and fleet angles

Measure the various angles that a cord makes on contact with a drum (either spool or pulley). The Belt-Cable Spool and Pulley blocks now provide the option to sense the drum and fleet angles at each point of contact. In addition to these angles, the Pulley block can output the wrap angle of the cord. These angles are defined as follows:

*   Drum angle — Angle, measured in the center plane of the drum (xy), spanning from the x-axis of the drum's reference frame to the drum's radius at the point of contact.

    A spool has one point of contact (associated with belt-cable port **A**) and therefore one drum angle. A pulley has two points of contact (associated with belt-cable ports **A** and **B**) and therefore two drum angles.

    The figure shows the drum angles (qpa and qpb on a pulley). The local reference frame indicates the x-axis (horizontal) and the y-axis (vertical) of the pulley.
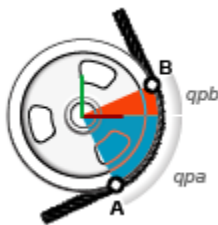
*   Fleet angle — Angle from the center plane of the drum (xy) to the cord at a point of contact. The spool has one point of contact and therefore one fleet angle (for port **A**. The pulley has two points of contact and therefore two fleet angles (one each for ports **A** and **B**).

    Note that to measure a nonzero fleet angle, the drums must connect at an angle, lie on different planes, or both. To allow for such configurations, the **Drum Belt-Cable Alignment** parameter of the Belt-Cable Properties block must be in its default setting, `Unrestricted`.

    The figure shows the fleet angle (qfa) at contact point **A** of a pulley. The local reference frame indicates the x-axis (horizontal) and the z-axis (vertical) of the pulley.

- Wrap angle — Angle, measured in the center plane of the drum (*xy*), between the points of contact of a pulley. The points of contact are associated with belt-cable ports **A** and **B**. This measurement is specific to the Pulley block.

  The figure shows the wrap angle between the contact points (**A** and **B** of a pulley). The local reference frame indicates the *x*-axis (horizontal) and the *y*-axis (vertical) of the pulley.
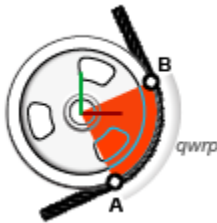
To sense any of the angles, open the block dialog box and, under **Sensing**, select the check box next to its name. The block exposes a new physical signal port through which to output the selected angle. You can use the signal as input in other Simscape blocks or convert it to a format compatible with Simulink blocks—for example, for plotting with the Scope block. (You must use the PS-Simulink Converter block for the signal conversion.)

Model Readability Enhancements: Interpret block diagram more quickly with sharper icons and styled connection lines

Blocks have new icons and connection lines have new styles. The icons are redesigned for clarity and are now in the modern SVG format. They can be scaled without loss of image quality. The connection lines are now gray in color and come in solid, dash, and dot-dash styles. The styles distinguish the domains to which the lines belong—frame, belt-cable, or geometry, respectively. The figure shows an example with frame connection lines on the left and belt-cable connection lines on the right.
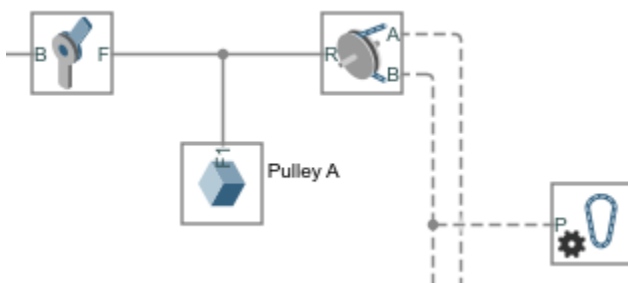
Solid Block Parameter Change: Command-line DensityBased parameter is now BasedOnType

The command-line name of the **Inertia > Based On** parameter of the Solid block, as well as the values that it can take, have changed. Previously referenced in code as `DensityBased` with `On` and `Off` as its two options, the parameter now goes by `BasedOnType` with `Mass` and `Density` as its new possible values.

The command-line name allows the block parameter to be referenced in code—for example, through the `set_param` and `get_param` functions that are sometimes used to specify and retrieve the values of block parameters.

The name change introduces a compatibility issue in MATLAB code reliant on `DensityBased` as an argument in such functions. If an error appears due to a "parameter named 'DensityBased' undefined," replace the references to the old command-line name with `BasedOnType`. If the error is in a `set_param` function, update also the value specified for the parameter—for example, from `On` to `Density`.

R2018a

Version: 5.2

New Features

Bug Fixes

Belts and Cables Library: Model systems with belts, cables, spools, and pulleys

Couple the motions of pulleys and spools using blocks from the new Belts and Cables library. Connect the pulleys in series to model an open system (such as a block and tackle) or in a loop to model a closed system (such as a belt drive). Use the block connections to set the direction of wrap, either clockwise or counterclockwise about the local rotation axis (by definition z of the local reference frame).
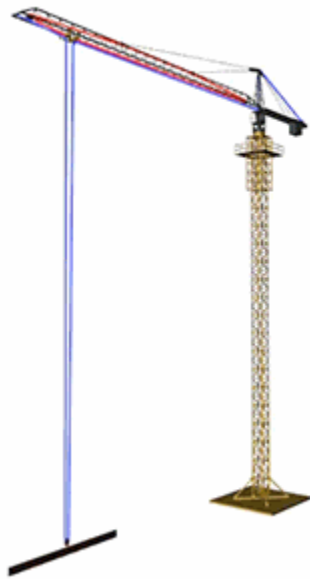
Tower Crane with Hoist and Trolley Pulley Systems

The Belts and Cables library provides four new blocks:

- Pulley — Model a wheel wrapped in a cord, a simple mechanism used frequently in the transmission of power and motion, often at an angle.
- Belt-Cable End — Identify the tip of a cord with which to anchor, drive, or load a pulley system.
- Belt-Cable Properties — Set the attributes of the cord of a pulley system, among them its visualization mode and color.
- Belt-Cable Spool — Model a cylindrical drum with which to wind and unwind the cord of a pulley system.

A new port type provides the means by which to run a cord through a component. The cord is generic and it can equally represent a (flat) belt or a (round) cable. The new ports are named *belt-cable* ports. Their rules of connection are provided by the *belt-cable* domain. Like the frame and geometry domains, the belt-cable domains allows only the connection of like ports.

The cord wraps counterclockwise about the local z-axis from port **A** to infinity in the case of a spool and from port **A** to port **B** in the case of a pulley. To reverse the wrap direction of a spool, you must flip the z-axis of the frame to which the spool is connected. To reverse the wrap direction of a pulley, you can do the same or, more conveniently, switch the connections of the belt-cable ports (**A** and **B**).

The belt-cable domain imposes a special requirement on belt-cable networks: each must connect to exactly one Belt-Cable Properties block. This block provides the necessary properties of the pulley cord. Some, like the visualization type and color, you specify explicitly as block parameters. Others, like the natural length of the cord, the block computes internally upon diagram update or during simulation.

The length calculation is based on the kinematic constraints present in a model. These are generally the product of rigid transforms, joints, and specialized constraints such as those characteristic of gears. To guide the calculation, and to better position an end of a cord, you must use the position state targets of an appropriate joint block (often one such as 6-DOF Joint placed between the pulley and some fixed frame).

For an example of a pulley model based on the belt-cable domain, at the MATLAB command prompt enter `smdoc_beltdrive_crossed`. The model comprises two pulleys connected by a cord in a crossed configuration. For an alternate example showing the belt drive in an open configuration, enter `smdoc_beltdrive_open` instead. The configurations reflect the ways in which the belt-cable ports connect—**A** to **A** and **B** to **B** in the crossed configuration, **A** to **B** and **B** to **A** in the open configuration.

Inertia Visualization: View the placement of inertia elements using graphic markers

Control the visualization of a pure inertia element in a model. You can now select a graphic marker to visualize the element associated with an Inertia block. Choices of marker encompass those already provided in the Solid block—`Sphere`, `Cube`, and `Frame`—and an additional `Inertia Icon` option included. Inertia visualization can be disabled in the block if necessary. Previously, pure inertia elements were identified by means of inertia icons only and their visualization was always on.

Inertia Icon, Sphere, Cube, and Frame Markers

URDF Import: Map <inertial> elements into Inertia blocks

The `smimport` function now maps the `<inertial>` elements of a URDF model into Inertia blocks. Previously, Solid blocks were used. The updated mapping allows for an imported model that more closely reflects in its structure the URDF model from which it derives. The Inertia blocks each provide the mass properties obtained from an <inertial> element—its mass, center of mass, moments of inertia, and products of inertia. For more information about URDF import, see URDF Import.

R2017b

Version: 5.1

New Features

Bug Fixes

Variable Solid Blocks: Model bricks, cylinders, and spheres with variable mass and size

Model a solid of variable mass and size using the new blocks provided in the **Body Elements > Variable Mass** library:

- Variable Brick Solid
- Variable Cylindrical Solid
- Variable Spherical Solid

Solid mass and size are each allowed to vary during simulation. Variable quantities can be specified explicitly through physical signals or automatically computed from other parameters during simulation. You can, for example, specify a variable mass and from it compute solid dimensions. You can also specify a variable solid dimension and from it compute the solid mass and other solid dimensions.

The visualization of variable solids is dynamic. Variable dimensions are shown as such during simulation. The solid reference frame is fixed relative to the variable solid geometry. The frame origin coincides with the center of mass in variable spheres. It lies on one of the sides normal to the z-axis in variable bricks and cylinders. Custom frames and in-block visualization, both features of the closely related Solid block, are not supported.

CAD Import Architecture Options: Simplify the structure of imported CAD models

Configure the smimport function to import a CAD assembly model with simplified topology. Model simplification is available through the `ModelSimplification` name-value pair argument. You can set this argument to:

- `bringJointsToTop` — Group rigidly connected parts into subsystems. Promote joints between rigid groups to the top level of the model hierarchy.
- `groupRigidBodies` — Group rigidly connected parts into subsystems. Leave joints between rigid groups in their original places in the model hierarchy.
- `None` — Import the model as is. Do not group rigidly connected parts. Leave joints between rigid groups in their original places in the model hierarchy.

Solid Block Inertia Calculations: Display automatically calculated values for inertia

The Solid block now displays the calculated values of the solid mass properties—mass, center of mass, moments of inertia, and products of inertia. Set the **Inertia > Type** parameter to `Calculate from Geometry` in order to expose a new node named **Derived Values**. Click the **Update** button under this node to calculate the solid mass properties and display their values in the field provided below the button.

Derived Values Display

Autodesk Inventor CAD Joint Translation: Import joints defined in CAD assembly using Simscape Multibody Link

Import Autodesk Inventor CAD assembly models containing joint definitions. All joint types are supported: `Rigid`, `Rotational`, `Slider`, `Cylindrical`, `Planar`, and `Ball`. Joint limits are ignored. Previously, only assembly constraints were supported. You can import a model as before, by first exporting it as a multibody description XML file—for example, using the Simscape Multibody Link CAD plugin—and then importing it using the `smimport` function.

New Featured Example

A new featured example (Modeling Self-Locking Worm and Gear Constraints - Worm Jack) shows how to model friction between a worm and a gear. The example features a worm jack, a device commonly used to lift and position loads. The worm jack is modeled in two configurations—one with self-locking characteristics and the other without. The self-locking worm jack remains in place upon removal of the driving torque by the action of friction.

R2017a

Version: 5.0

New Features

Bug Fixes

Run-Time Parameters: Speed up simulation tasks and modify component parameter values without regenerating C code

Simscape Multibody software now supports run-time parameters. You can configure most numerical block parameters as `Compile-time` (default) or `Run-time` using a drop-down list that appears beside configurable parameters. The figure shows the run-time drop-down list in a Solid block dialog box. All parameters are by default `Compile-time`.
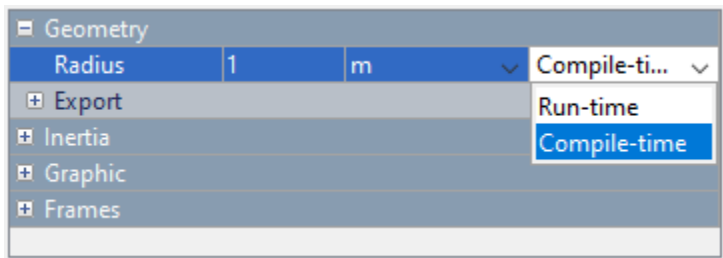
`Compile-time` parameters update in value only when you recompile the model. Use them when performing tasks that rely on inlined parameters such as model optimization. `Run-time` parameters update in value without the need for extra compilations. Use them when tuning parameter values in Fast Restart mode or when simulating models that rely at least in part on generated C code.

To set a parameter as `Run-time` from the block dialog box, you must configure your Simscape preferences. Open the MATLAB Preferences window, select the **Simscape** node, and check the **Show run-time parameter settings** check box. Parameters without a run-time option or with a run-time option that is inactive (i.e., "grayed out") cannot be configured. The **Length** parameter in the figure is an example.
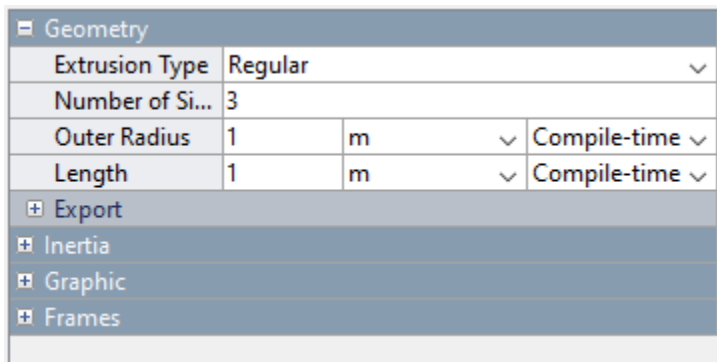
For more information about Simscape run-time parameters, see About Simscape Run-Time Parameters (Simscape).

Onshape CAD Import: Convert an Onshape CAD assembly into a Simscape Multibody model

A new function named `smexportonshape` enables you to export an Onshape® CAD assembly model in a format suitable for import into the Simscape Multibody environment.

The `smexportonshape` function generates a multibody description file compliant with the XML schema and a set of STEP geometry files with the geometries of the various bodies. You import the files using the `smimport` function, passing the name of the XML file as a function argument.

For an example showing how to use the new function to export an Onshape model of a humanoid robot, see Import an Onshape Humanoid Model.

URDF Import: Convert a URDF model into a Simscape Multibody model

Import a URDF multibody model into the Simscape Multibody environment using an updated version of the `smimport` function. The function distinguishes between XML and URDF multibody description files via the optional file extension. In the absence of a file extension, the function assumes that the multibody description file conforms to the Simscape Multibody XML schema.

Try importing a URDF model of a humanoid robot as an example. The URDF file and the supporting part geometry files are included in your Simscape Multibody installation. At the MATLAB command prompt, enter the command:

```
smimport('sm_humanoid.urdf');
```

Simscape Multibody software recreates the URDF model as a block diagram that you simulate and build on. To see an example of the imported model with simple motion inputs, at the MATLAB command prompt, enter `sm_import_humanoid_urdf`.

Worm and Gear Constraint Block: Model the kinematics of a worm in mesh with a gear

Model the kinematic constraint between a worm and a gear using the new Worm and Gear Constraint block. The constraint couples the worm and gear so that they rotate at a fixed speed about offset perpendicular axes. The new block imposes assembly requirements that you must satisfy in order to simulate the model. See Worm and Gear for a simple assembly example.
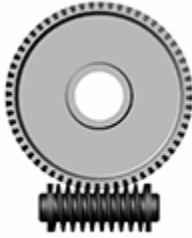
Worm and Gear Assembly

General Variable Mass Block: Model a point or distributed mass with time-varying inertial properties

Model the effects of a shifting center of mass and of a time-varying mass and inertia tensor using the new General Variable Mass block. You can specify all three variables as functions of time using physical signal ports. The ports are exposed when the corresponding block parameters—Mass, Center of Mass, and Inertia Tensor—are set to Time-Varying in the block dialog box.

The variable mass can be a point mass or a distributed mass. The Inertia Tensor parameter is active only for a distributed mass. To model a distributed mass, set the **Inertia > Type** parameter to Custom. Inertia ellipsoids provide a means to visualize the time-varying inertial properties.

Use the block in models where the variable mass contents of a system matter, as in a tank truck carrying a sloshing load of fluid.
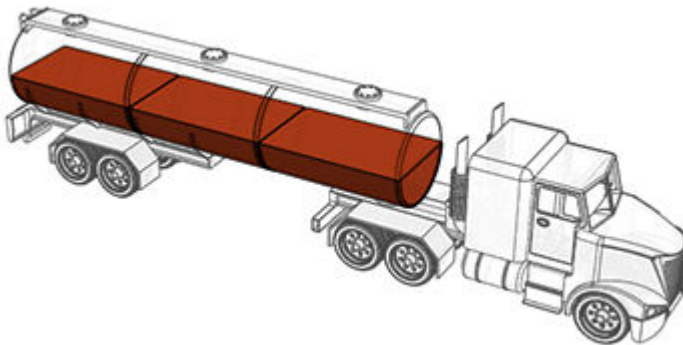
Tank Truck with Variable Fluid Load

New Featured Examples

The examples library includes two new featured examples. The Independent Suspension System Templates example shows how to model various types of automotive suspensions using Simscape Multibody blocks, including MacPherson, double wishbone, and pushrod types. The Using the Worm

and Gear Constraint Block — Solar Tracker example shows how to model a real application using the new Worm and Gear Constraint block.

R2016b

Version: 4.9

New Features

Bug Fixes

Compatibility Considerations

Tracking Mode for Dynamic Cameras: Track objects during animation playback

A new `Tracking` mode enables you to define dynamic cameras that follow specific components during simulation. You can fix the position of a dynamic camera to any frame origin in a model. You can aim the camera at any other frame origin in a model or align it with a specific frame axis.

To create or modify a dynamic camera, use the Camera Manager tool in Mechanics Explorer. You can create multiple dynamic cameras in keyframe and tracking modes and select them using the context-sensitive (right-click) menu in the Mechanics Explorer visualization pane.

The Configuring Dynamic Cameras - Vehicle Slalom featured example showcases three dynamic cameras defined in `Tracking` mode and compares them with a fourth dynamic camera defined in `Keyframes` mode.

For an overview of global and dynamic cameras, see Visualization Cameras. To create a dynamic camera and define its motion in `Keyframes` or `Tracking` mode, see Create a Dynamic Camera.

smwritevideo Function: Generate videos from the MATLAB command line

Programmatically configure and create videos of your simulation results from the MATLAB command line using a new function named `smwritevideo`. Use the optional `Name,Value` pair arguments to manipulate the video playback speed, frame rate, frame size, and file format.

The function supports several common file formats, including uncompressed AVI and MPEG-4 (MP4). Change the video playback speed to better capture the model dynamics for very fast systems, such as rapidly oscillating beams, or very slow systems, such as the solar system.

Video Creator Tool in Mechanics Explorer: Configure video properties such as encoding type, playback speed, frame rate, and frame size

Interactively configure and create videos of your simulation results using the new Video Creator tool in Mechanics Explorer. Modify the Video Creator parameters to obtain the desired video properties, including the video encoding type, playback speed, frame rate, and frame size. Video Creator is the UI equivalent of the new `smwritevideo` function.

Animation Resampling for Video Creation: Produce constant-speed videos using any solver settings

Video playback speed is now independent of Simulink solver type, step size, and data output options. Create videos with the desired playback speed and frame rate without having to adjust your model configuration parameters. Previously , the video frame rate was fixed at 30 frames per second and the playback speed was constrained to your solver and data export options.

Roll Tool in Mechanics Explorer: Rotate the animation view about the line of sight

Rotate the visualization viewpoint in the plane of your screen about the camera's line of sight. The new tool appears alongside the Rotate, Pan, and Zoom tools in the visualization toolstrips of the Solid

block dialog box and Mechanics Explorer. The Roll tool differs from the Rotate tool in that its rotations are strictly 2D.

Physical Unit Update: Hertz definition changed from rev/s to 1/s

The definition of hertz (`Hz`) has changed from `Hz` = `rev/s` to match the standard SI convention `Hz` = `1/s`. The updated definition is the same adopted in the Simscape Foundation library from R2013a onward. This change affects the kilo, mega, and giga SI multiples of hertz (`kHz`, `MHz`, `GHz`).

To discourage the use of hertz, this unit has been replaced with `rev/s` in the block units dropdown lists. You can still manually enter `Hz`, `kHz`, `MHz`, and `GHz` in the block units fields, but from R2016b onward these map into `1/s`, `1/ms`, `1/µs`, and `1/ns`.

For consistency with the old definition, in older models, Simscape Multibody software automatically replaces `Hz`, `kHz`, `MHz`, and `GHz` with `rev/s`, `rev/ms`, `rev/µs`, and `rev/ns`. The table shows the old and new mappings between `Hz` and `rev/s` multiples.

Unit	Old Definition	New Definition
Hz	rev/s	1/s
kHz	rev/ms	1/ms
MHz	rev/µs	1/µs
GHz	rev/ns	1/ns

New Featured Examples

The examples library includes two new featured examples. The Configuring Dynamic Cameras - Vehicle Slalom example showcases the tracking mode of dynamic cameras. The Hydraulic Interface - Dump Trailer with Hydraulic Cylinder example showcases a cross-domain interface between multibody and hydraulic networks.

R2016a

Version: 4.8

New Features

Bug Fixes

SimMechanics renamed to Simscape Multibody

SimMechanics™ software is now named Simscape Multibody. The product library is located in the **Simscape** node of the Simulink library browser under the name **Multibody**. You can open the product library at the MATLAB command prompt using the `sm_lib` command.

The product library structure has been flattened. Sublibraries previously under **SimMechanics Second Generation** now appear directly under **Multibody**. The complete SimMechanics First Generation library is now a **Multibody** sublibrary named **First Generation (1G)**.

Dynamic camera in Mechanics Explorer: Specify camera location and orientation during simulation using keyframes

Mechanics Explorer adds dynamic cameras with gradually shifting viewpoints during playback. A Camera manager enables you to create cameras and specify camera pose through keyframes—each a viewpoint attached to a playback time. During playback, Mechanics Explorer interpolates between the various keyframes to obtain smooth camera trajectories.

You can create multiple cameras in a model, each with its own set of keyframes. During playback, you can switch between previously created cameras by using the visualization pane context-sensitive (right-click) menu. If you split the visualization pane into tiles, you can assign different cameras to different tiles. For more information, see:

- Global and Dynamic Cameras
- Create a Dynamic Visualization Camera

Perspective view in Mechanics Explorer: Enhance three-dimensional quality of animated simulation results

Dynamic cameras capture your model animation using a perspective view. This view enhances the perception of depth by making bodies appear smaller with increasing distance from the dynamic camera. The perspective view is active only in dynamic cameras. The global camera continues to use an orthographic view. The figures compare the perspective and orthographic views of dynamic and global cameras.
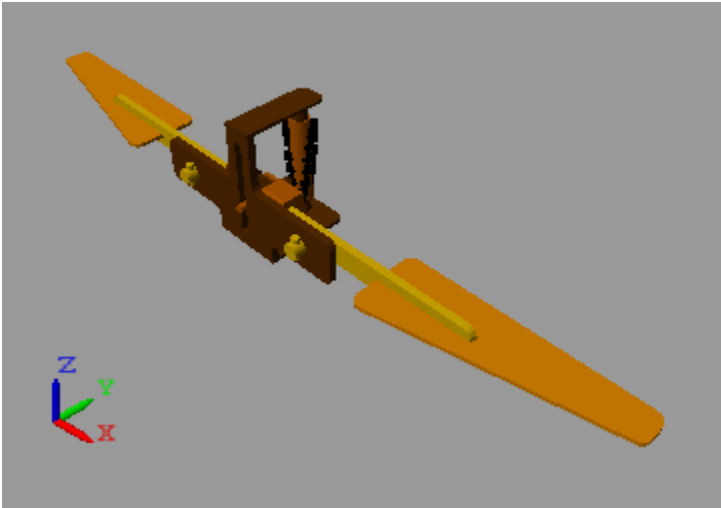
Perspective View of Dynamic Cameras
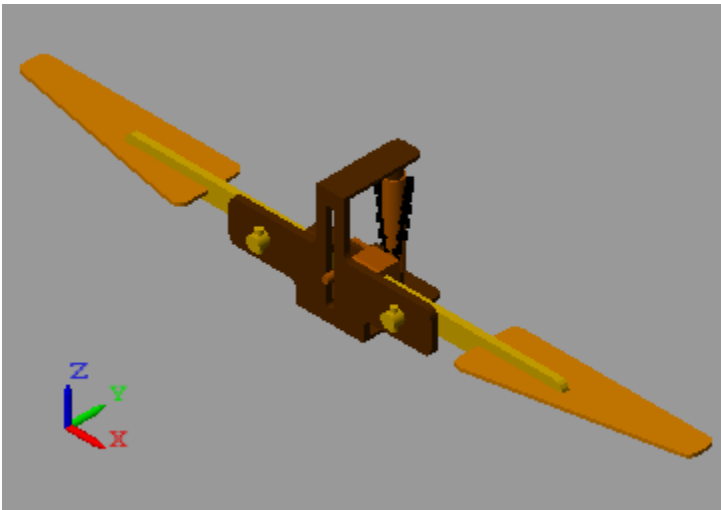
Orthographic View of Global Camera

Parameterized models from CAD import: Create parameterized model with option to update parameter values only

Imported models are now parameterized in terms of MATLAB variables defined in a separate data file. The `smimport` function generates the new data file for each model imported and links the two through the model workspace. The data file contains the numerical values of all block parameters, including those parameters left in their default settings.

The `smimport` function provides a new import mode for regenerating data files associated with previously imported models. The new mode provides a limited form of CAD update that updates block parameters but not the block diagram. You specify the new mode by setting the optional `ImportMode` Name,Value pair argument to `dataFile`.

For more information, see:

- CAD Translation
- Model Data File
- Data File Update

CAD joint definitions in Simscape Multibody XML schema: Express CAD joints directly in XML files

The `smimport` function now supports CAD joint definitions in XML import files. To this end, the Simscape Multibody XML schema provides a new Joints parameter section. The Simscape Multibody Link CAD plug-in does not support the new joint definitions, but you can create XML import files from the XML schema directly.

R2015b

Version: 4.7

New Features

Bug Fixes

Point On Curve Constraint block

Constrain the position of a frame to a specified curve, for example, to model a roller coaster or cam mechanism. You specify a curve using the new Spline block, which you connect to geometry port B. The Point On Curve Constraint block appears in the Constraints library.

The block constrains the attached frame by applying the force needed to keep it on the specified curve. You can sense the constraint force as a three-dimensional vector that the block outputs as a physical signal.

For a tutorial showing how to use the Spline and Point On Curve Constraint blocks, see Model Cam Mechanism.

Spline block for specifying curved paths

Specify a 2-D or 3-D curve along which to constrain the position of a frame. Connect the Spline block to geometry port B of a Point On Curve Constraint block in order to constrain a frame to the specified curve. The Spline block appears in the new Curves and Surfaces library.

Spline curves can be open or closed and are based on cubic interpolation between the curve points you specify. You can visualize the spline curve in the Spline block UI, for example, to ensure the curve shape is correct. You can also visualize the spline curve in Mechanics Explorer on model update and during simulation.

For a tutorial showing how to use the Spline and Point On Curve Constraint blocks, see Model Cam Mechanism.

Frame creation via Solid block UI

Interactively add frames to a solid using the frame creation interface in the Solid block dialog box. Use the Frames menu to add or edit any number of frames. You can define the frame origin and orientation based on the solid reference frame axes, the solid center of mass and principal axes, or geometric features such as points, lines, and surfaces.

You can use a geometric feature by selecting the **Based on Geometric Feature** radio button, selecting the desired geometric feature in the visualization pane, and then clicking the **Use Selected Feature** button. To keep your new frame, save the frame definition and select **OK** or **Apply**.

For a tutorial showing how to use the frame creation interface, see Interactively Create Solid Frames.

Mechanics Explorer link to block diagram

Go to the block corresponding to a node on the tree view pane in Mechanics Explorer. You can select a solid in the visualization pane, right-click the highlighted node in the tree view pane, and then select **Go to Block**. You can also select other node types directly from the tree view pane, such as those corresponding to subsystems, joints and constraints, forces and torques, and other blocks. For more information, see Go to Block from Mechanics Explorer.

Rotation Sequence option in Rigid Transform block

Specify a 3-D rotation as a sequence of three elementary rotations. Such rotation sequences are commonly known as Euler and Tait-Bryan sequences. The rotation sequence parameterization is

available in the Rigid Transform block. It is also available in joint blocks with spherical joint primitives, including Spherical Joint, 6-DOF Joint and Telescoping Joint.

Featured example

A new featured example showcases the Point-on-Curve Constraint and Spline blocks:

• sm_cam_flapping_wing

R2015a

Version: 4.6

New Features

Bug Fixes

Visibility setting to show or hide solids, bodies, and multibody subsystems in Mechanics Explorer

Switch the visibility of a solid, body, or multibody subsystem on or off in Mechanics Explorer. A new context-sensitive menu in the tree-view pane enables you to:

- Show the selected component
- Hide the selected component
- Show only the selected component
- Show all the components in the model

To switch the visibility setting, in the tree-view pane of Mechanics Explorer, right-click the name of a solid, body, or multibody subsystem. Then, select the desired option from the context-sensitive menu. For more information, see Filtering Model Visualization.

Lead Screw Joint block

The Joints library adds a Lead Screw Joint block. Use this block to convert rotational motion to translational motion and, conversely, translational motion to rotational motion. Lead screw applications include the linear actuators typically used in aircraft flap systems, CNC machines, and 3-D printers. The block is based on a new joint primitive named Lead Screw Primitive (LS).

Constant Velocity Joint block

The joints library adds a Constant Velocity Joint block. Use this block to model a joint with two rotational degrees of freedom—the bend and azimuth angles—and a kinematic constraint that ensures the base and follower port frames spin at constant velocity relative to each other regardless of their relative angle.

Constant velocity joint applications include automotive driveline systems with power transmission occurring between shafts connected at an angle. The block is based on a new joint primitive named Constant Velocity Primitive (CV).

R2014b

Version: 4.5

New Features

Bug Fixes

Constraint force and torque sensing in Constraint blocks

Compute and output the forces and torques that enable a Constraint block to enforce its kinematic constraint. The blocks output one or more physical signals with the variables that you select. These variables can include the constraint force and torque vectors and their signed magnitudes. The blocks resolve the vector quantities in the resolution frame that you specify.

The new blocks with constraint force and torque sensing are Angle Constraint and Distance Constraint. These blocks expand the list of blocks with constraint force and torque sensing. You can also sense constraint forces and torques through Joint blocks.

Solid 3-D visualization in Solid block dialog box

Visualize the solid you are modeling directly in the Solid block dialog box. The updated block includes a collapsible visualization pane that shows the current state of your 3-D solid. You can use this pane to check the geometry and color of individual solids before updating or simulating the model.

The visualization pane includes a toolstrip for controlling the solid view. The toolstrip, which provides many of the Mechanics Explorer toolstrip buttons, enables you to rotate, pan, and zoom a solid, to select a standard view point, and to visualize the solid reference frame.

Rotation matrix parameterization in Rigid Transform and Joint blocks

Specify a frame rotation using a 3-D rotation matrix. The new parameterization is available in two block types, Rigid Transform and joint blocks with spherical joint primitives. Use this parameterization in the Rigid Transform block to specify a fixed rotation or in a joint block to specify the position state target of a spherical joint primitive. The joint blocks with the new parameterization are:

- 6-DOF Joint
- Spherical Joint
- Telescoping Joint

Featured example

A new featured example showcases the constraint force and torque sensing capability of Joint blocks:

- Sensing Composite Forces and Torques in Joints — Potter's Wheel

R2014a

Version: 4.4

New Features

Bug Fixes

STEP file import

Load solid geometry directly from a STEP file. This file type enables SimMechanics to automatically compute inertia from solid geometry, an advantage over STL files. You can generate the STEP file that you want to load externally, e.g., using a CAD platform.

Total and constraint force and torque calculation in joint blocks

Sense the total and constraint forces and torques acting at whole joints. Joint block dialog boxes provide a new expandable parameter list, **Composite Force/Torque Sensing**, in which you can select the whole-joint variables to sense. Select a variable to calculate its time-dependent values during simulation and deliver them through a physical signal output port. Variables that you can sense include:

- Total Force/Torque — Net sum of forces and torques acting at the joint. These variables include contributions from actuation, internal mechanics, and constraint forces and torques.
- Constraint Force/Torque — Forces and torques acting orthogonally to the joint degrees of freedom. In a Revolute Joint block, which provides a single rotational degree of freedom about the Z axis, the constraint forces act along all the Cartesian axes (X, Y, and Z), while the constraint torques act about the X and Y axes.

Time-varying gravity option in Mechanism Configuration block

Specify the gravitational acceleration vector as a function of time, using the Mechanism Configuration block. To support this task, the **Uniform Gravity** parameter in the block dialog box now provides a `Time-Varying` option. Select that option to expose a physical signal port to which you can connect a time-varying gravity physical signal. The gravitational acceleration vector is strictly uniform in space. It does not vary with position.

Gravitational Field block

Model the gravitational pull of a point mass on every rigid body in a mechanism. The Gravitational Field block exerts a force that varies with mass and distance according to Newton's law of universal gravitation. Applications include large-scale orbital systems in which gravity changes due to varying distances play an important role—e.g., a solar system model for spacecraft trajectory planning. The block is in the Forces and Torques library.

Animation at arbitrary speed in Mechanics Explorer

Change the base playback speed of a SimMechanics animation by an arbitrary factor. This speed corresponds to the 1x animation slider setting in Mechanics Explorer. It equals the number of seconds in simulation time that play in one second of animation playback time.

You can specify the base playback speed as a numerical input in Mechanics Explorer. This parameter appears in **Tools** > **Animation Settings**. Change its value to better visualize motion at extreme time scales.

The new parameter is relevant, for example, in a solar system model. At the default value of the base playback speed, planet motion is exceedingly slow. Due to the time scale of such a model, moving the playback speed slider to its maximum value of 256× does not suffice. Planet motion remains nearly imperceptible.

To visualize one Earth revolution in minutes or seconds, you must increase the playback speed by a very large factor. This factor might be seven orders of magnitude larger than the default value—larger than the playback speed slider allows. You specify such a large playback speed using the new base playback speed parameter.

Consider a model of the solar system. You can play animation at a rate of one Earth revolution per second. To play animation at this rate, you specify a base playback speed of

$$365\frac{\text{days}}{\text{second}} \cdot 24\frac{\text{hours}}{\text{day}} \cdot 60\frac{\text{minutes}}{\text{hour}} \cdot 60\frac{\text{seconds}}{\text{minute}} = 3.15e7.$$

You can then use the playback speed slider to fine-tune the playback speed. The slider multiples the base playback speed by factors of two between a minimum of 1/256 and a maximum of 256.

R2013b

Version: 4.3

New Features

Bug Fixes

Joint blocks that now permit prescribed motion defined as an input signal

Joint blocks now accept motion variables—position, velocity, acceleration—as actuation inputs. Use these inputs to specify the time-dependent path of a joint directly.

Actuation force and actuation torque calculation for prescribed motion

Joint blocks can now compute and output actuation force and torque values. Use these outputs to analyze actuation force and torque requirements at a joint, for example, due to prescribed motion.

Additional CAD constraints translated during model import

The import function for second-generation models, `smimport`, supports three additional CAD constraints: parallel, perpendicular, and distance. Imported models represent parallel and perpendicular constraints using Angle Constraint blocks, and distance constraints using Distance Constraint blocks.

Bevel Gear Constraint block

The Gears and Couplings library contains one new block:

- Bevel Gear Constraint — Gear constraint between two frames spinning about arbitrarily oriented axes.

Featured examples

Two new featured examples are available:

- Computing Actuator Torques Using Inverse Dynamics
- 3-Roll Robotic Wrist Mechanism

R2013a

Version: 4.2

New Features

Bug Fixes

Common Gear block

SimMechanics provides a new Common Gear Constraint block. You can find this block in the new Gears, Couplings and Drives library.

Block	Description
Common Gear	Gear constraint that you can use to transmit angular motion between two rigid body frames rotating about parallel axes

Rack and Pinion block

SimMechanics provides a new Rack and Pinion Constraint block. You can find this block in the new Gears, Couplings and Drives library.

Block	Description
Rack and Pinion	Gear constraint that you can use to convert the angular motion of a pinion frame into linear motion of a rack frame and vice-versa

Pin Slot Joint block

SimMechanics provides a new Pin Slot Joint block. You can find this block in the Joints library.

Block	Description
Pin Slot Joint	Joint with two degrees of freedom, one translational and one rotational, with mutually orthogonal motion axes

Internal Force block for modeling action and reaction forces

SimMechanics provides a new Internal Force block. You can find this block in the Forces and Torques library.

Block	Description
Internal Force	General force acting between two rigid body frames along the line connecting their origins

Published XML Schema for model import, enabling users to import mechanical models from external applications such as CAD systems

SimMechanics now provides an XML schema that you can use to generate SimMechanics Import XML files. Use the schema in conjunction with the API of an external application, such as a CAD platform, to generate a valid SimMechanics Import XML file for a multibody model. Then, import this file to automatically generate an equivalent model in SimMechanics Second Generation format.

Higher Gravity Precision

The gravity vector in the Mechanics Configuration block now has a precision of six significant digits (`-9.80665 m/s^2`). Older models with a Mechanism Configuration block retain the older precision level of three significant figures. To apply the higher precision level in these models, you can manually update the gravity vector.

SimMechanics Generation Comparison

The Gears library adds to SimMechanics Second Generation new complex constraints previously found only in SimMechanics First Generation. For a full comparison between the two SimMechanics generations, see Modeling with SimMechanics First and Second Generations.

Featured examples

Four new featured examples are available:

- Double-Wishbone Suspension
- Backhoe
- Cardan Gear
- Windshield Wiper

R2012b

Version: 4.1

New Features

Bug Fixes

Compatibility Considerations

Second Generation technology now supporting CAD import from SolidWorks, Inventor, and Pro/ENGINEER

SimMechanics introduces CAD import in SimMechanics second-generation format. Import the SimMechanics Import XML file that represents a CAD assembly, and automatically generate the corresponding SimMechanics second-generation model.

Second Generation technology now supporting C-code generation with Simulink Coder

SimMechanics Second Generation now supports code generation with Simulink Coder™. Generate C code from a SimMechanics model, and use the code for applications like hardware-in-the-loop (HIL) testing and rapid prototyping.

SimMechanics Second Generation does not support run-time parameters. Model parameters are fixed during code generation, and are not available for edit outside SimMechanics. To update model parameters, you must edit them in SimMechanics and then regenerate the C code for your model.

Solid of revolution geometry in Solid block

The Solid block adds a `Revolution` geometry. The geometry represents solids that have a constant cross-section about an axis of revolution. Examples include cones, domes, and toroids. Revolutions can be full or partial.

Changes to inertia specification

The form of the **Products of Inertia** input vector has changed in Solid and Inertia blocks. The new input vector has the form [I_{yz}, I_{zx}, I_{xy}]. The vector elements correspond to the inertia matrix off-diagonal elements:

$$\begin{pmatrix} & I_{xy} & I_{zx} \\ I_{xy} & & I_{yz} \\ I_{zx} & I_{yz} & \end{pmatrix}$$

Previously, the **Products of Inertia** parameter required the negative values [-I_{yz}, -I_{zx}, -I_{xy}].

Four new examples

SimMechanics introduces four new examples:

- Radial Piston Engine
- Electrically Operated Bread Slicer
- CAD Import of Stewart Platform
- CAD Import of Four-Bar Mechanism

R2012a

Version: 4.0

New Features

Bug Fixes

Compatibility Considerations

SimMechanics Version 4.0 Software and Second Generation Technology

SimMechanics Version 4.0 introduces Second Generation technology to model, simulate, and analyze two- and three-dimensional multi-body mechanical systems. Second Generation technology provides a more intuitive and powerful block diagram language that lets you easily create reusable parts and libraries. It also adds advanced three-dimensional visualization and animation support.

Version 4.0 provides full support for both First Generation and Second Generation functionality, including block sets and tools. With Version 4.0, you have the option to create and simulate a mechanical model using First or Second Generation technology. Version 4.0 software gives you the two full block libraries, labeled First Generation and Second Generation.

Note The term *generation* — e.g. "Second *Generation*" — refers to the SimMechanics technology used. A SimMechanics *version* may contain one or both *generations*.

SimMechanics First and Second Generation Technologies

SimMechanics Second Generation technology is a comprehensive advancement of SimMechanics First Generation technology. To ease the transition from First Generation to Second Generation technology, and to support existing First Generation users, SimMechanics Version 4.0 fully supports *both* First Generation and Second Generation features.

This table summarizes First Generation and Second Generation technologies in their native SimMechanics versions.

Technology	Description	Block Library
SimMechanics First Generation	Encompasses the complete SimMechanics behavior through Version 3.2.3 (R2011b). First Generation features are designated by the First Generation library in both the Simulink Library Browser and the Help Browser windows.	To access the block library, at the command line, type `mechlib`.
SimMechanics Second Generation	Encompasses the new SimMechanics behavior in Version 4.0 (R2012a). Second Generation features are designated by the Second Generation library in both the Simulink Library Browser and the Help Browser windows.	To access the block library, at the command line, type `sm_lib`.

Compatibility Considerations

First- and second-generation blocks are based on different languages. To interconnect the two block types, you must use Simulink-PS Converter and PS-Simulink Converter blocks. You can find the converter blocks in the Simscape Utilities library.

Improvements in SimMechanics Second Generation Technology

The following functionality has been added or improved for SimMechanics Second Generation:

- Multibody modeling paradigm centered on frames, including:

    - Ports and nodes map that directly to frames
    - Frames that are strictly locally-defined

- Mechanics Explorer visualization utility

    - Visualize 3–D models
    - Explore model using tree browser and visualization panes

- Enhanced animation support

    - Control animation speed
    - Replay animation without rerunning simulation

- Inertia calculation and joint state targets

    - Automatically calculate inertia from geometry
    - Specify joint state target values and priority settings

- Improved integration with Simscape.

    - Includes support for physical signals and units.

Common Functionality in SimMechanics First and Second Generation Technologies

The following functionality is present in both SimMechanics First and Second Generations:

- Rigid body specification
- Joint specification
- Constraint specification
- Force and torque inputs
- Linearization of mechanical models
- Trimming mode of analysis
- Motion outputs

Functionality Found Only in SimMechanics First Generation Technology

The following functionality is present only in SimMechanics First Generation:

- Code generation support

    - Model reference accelerator mode is absent in the SimMechanics Second Generation sub-library.

- CAD import support

- Motion input in Joint Blocks
- Force outputs in **Joints** and **Constraints** blocks
- Missing **Constraints** blocks:

    - *point-curve*, *gear*, *velocity*, and *projection* constraints are not included in the SimMechanics Second Generation sub-library.
- Missing Force Outputs

    - Applied and reaction force outputs for joints are not available.
- Missing Joints

    - *Screw* and *massless connector* joint blocks are not included in the SimMechanics Second Generation sub-library.
- Variable gravity and mass distribution
- Visualization features

    - Model reference normal mode is not included in SimMechanics Second Generation technology.

R2011b

Version: 3.2.3

Bug Fixes

R2011a

Version: 3.2.2

No New Features or Changes

R2010b

Version: 3.2.1

No New Features or Changes

R2010a

Version: 3.2

New Features

Optional Simplification of Subsystem Hierarchy During Model Import

When you generate SimMechanics models with the `mech_import` command, you can now choose certain automatic simplifications of the imported model's hierarchy:

- Bringing all joints in the generated model to the highest level being imported, either the whole model or a subsystem.
- Grouping all welded bodies at each level of the model hierarchy into new rigid subsystems at the same level.

Without these options, the generated model's hierarchy follows the hierarchy specified in the imported XML file, derived from its original CAD assembly.

"What's This?" Context-Sensitive Help Available for Import Physical Modeling XML Dialog Box

This release introduces "What's This?" context-sensitive help for parameters that appear in the Import Physical Modeling XML dialog opened by entering `mech_import` with no arguments at the command line. This feature provides quick access to a detailed description of the parameters, saving you the time it would take to find the information in the Help browser.

To use the "What's This?" help, do the following:

1 Place your cursor over the label of a parameter.

2 Right-click. A **What's This?** context menu appears.

For example, the following figure shows the **What's This?** context menu appearing after a right-click on the **Start time** parameter in the **Solver** pane.

3 Click **What's This?** A context-sensitive help window appears showing a description of the parameter.

Connection Port Block Consolidated to Simscape Library and Documentation

The Connection Port block has been removed from the block libraries of all add-on products dependent on Simscape software. When using these dependent add-on products, look for the Connection Port block in the Simscape Utilities library.

The Connection Port block help for these add-on products now directs you to the Simscape block reference.

New Demos

Two new demos have been added.

Model Hierarchy Simplification During Import

You can learn more about how to simplify imported model hierarchy by reviewing the Mechanical Import with Model Simplification featured example.

This demo uses two new Physical Modeling XML files, `SimplificationExample1.xml` and `SimplificationExample2.xml`. Models generated from these XML files also appear as large graphics examples in Nondefault Hierarchy and Organization of Generated Models.

Reference Coordinate Systems in Imported Physical Modeling XML

With the SimMechanics Link utility, you can export reference coordinate systems that you insert into CAD assemblies. (This feature is not supported on all CAD platforms.) From the exported Physical Modeling XML, you can generate a SimMechanics model with extra Body coordinate systems corresponding to the assembly reference coordinate systems and not associated with CAD constraints and automatically generated Joints.

Using the new `GearAssembly.xml` file, the User-Added Coordinate Systems in an Imported Gear Assembly featured example shows how you can use reference coordinate systems to manually add a Constraint block to a model after import.

R2009b

Version: 3.1.1

No New Features or Changes

R2009a

Version: 3.1

New Features

Changes to External Virtual Reality Visualization

The new Simulink 3D Animation software replaces the Virtual Reality Toolbox™ product for visualizing virtual scenes. The documentation section on building and using a virtual reality client to visualize SimMechanics models has been updated to reflect this change.

Pro/ENGINEER CAD Translation Case Study

The CAD translation chapter now contains a case study demonstrating assembly export and re-export, along with model import and update. The study uses SimMechanics and SimMechanics Link software, together with Pro/ENGINEER™, and models a double pendulum, subsequently modified to a triple pendulum. The study illustrates how you can update an existing generated CAD-based model with successive changes to the original CAD assembly.

R2008b

Version: 3.0

New Features

Compatibility Considerations

New SimMechanics Visualization and Animation

The visualization has been upgraded with new features and a new interface. It now supports

- Customizable body geometries with STL (stereolithographic) files
- Customizable body colors
- Better dynamic view control
- Better animation speed control

Consult the new *SimMechanics Visualization and Import Guide* for complete information.

For more about creating STL files, see "BodyBuilder for Creating STL Body Geometry Files" on page 30-3.

Compatibility Considerations

The old SimMechanics visualization allowed you to save visualization settings to a MAT-file associated with a SimMechanics model.

When you open such a model in the new version, the model will attempt to load the MAT-file, as before, and map the old visualization options to the new options as closely as possible. This is a one-time conversion. The MAT-file has to be on the MATLAB path so it can be found when you open the model.

To save these converted visualization settings:

1 You must open the visualization window and save these settings to the model.
2 Then you must save the model itself.

Use the visualization controls to save the settings to the model. See the *SimMechanics Visualization and Import Guide*.

Caution To save visualization settings from an old model in the new visualization, you must follow these steps, in this order. Otherwise, you might lose the old settings permanently.

Compressed Animation Recording from New Visualization on Windows Vista and 64-Bit Windows Operating Systems

To compress a recorded animation of a simulation, you need the Indeo 5 codec. This codec might not be available on the 32-bit Windows Vista® operating system and the 64-bit Windows XP and Windows Vista operating systems. If you require compression and the codec is not found, the animation AVI file is still saved, but without compression. A warning appears.

For further details, consult Technical Solution 1-1HTNHW at the MathWorks web site.

New Visualization Window Status Bar Not Available on Intel Mac Operating System

The status bar of the new visualization window, below the machine display, does not appear at all on the Intel Mac operating system. On other operating systems, the status bar shows simulation time, as well as Body block and coordinate system names.

New Visualization Not Supported on Solaris 64 UNIX Operating System

The new visualization and animation interface is not available on the Solaris 64 UNIX operating system. SimMechanics software automatically reverts to the old visualization interface.

New SimMechanics Link Utility

The new SimMechanics Link utility provides a bridge from third-party mechanical design and modeling applications to the system and control design features of MATLAB and Simulink. It connects computer-aided design (CAD) platforms to mechanical modeling with SimMechanics software by enabling export of CAD assemblies into the Physical Modeling XML format that allows for automatic generation of SimMechanics models.

Upgraded Mechanical Import and Automatic Model Generation

The new `mech_import` command works with externally defined Physical Modeling XML files to automatically generate SimMechanics models. It supports a new SimMechanics feature, updating models generated previously.

Compatibility Considerations

Physical Modeling XML files generated by the old CAD-to-SimMechanics translators remain compatible with `mech_import`. Using this command, you can still import old XML files to generate SimMechanics models. Because these models have no associated exported STL body geometry files, they will visualize using the standard (noncustom) body shapes.

The updating feature of `mech_import` works only with Physical Modeling XML files generated by exporting from the SimMechanics Link utility. This feature does not work with Physical Modeling XML files generated from the old CAD-to-SimMechanics translators.

You should replace all existing instances of the `import_physmod` command with the new `mech_import` command. See "Command Being Removed" on page 30-4.

Caution Do not attempt to use the `import_physmod` command in older SimMechanics versions with Physical Modeling XML files generated by the new SimMechanics Link exporter.

BodyBuilder for Creating STL Body Geometry Files

In addition to generating STL body geometry files automatically from a CAD assembly, you can also create STL files by hand or by using a body geometry editor. BodyBuilder is an application based on MATLAB that provides a graphical means to create STL files for use in SimMechanics body visualization. You can obtain BodyBuilder from MATLAB Central.

Command Being Removed

Command Name	What Happens When You Use Command?	Use This Instead	Compatibility Considerations
`import_physmod`	Still runs	`mech_import`	Replace all existing instances of `import_physmod` with `mech_import`.

SimMechanics Documentation Reorganized

The SimMechanics documentation is now organized into four books.

Book	Includes Material from Old Chapters
SimMechanics Getting Started Guide	1, 2, and 3
SimMechanics User's Guide	4, 5, 8, and 9
SimMechanics Visualization and Import Guide	6 and 7
SimMechanics Reference	10, 11, 12, and 13

R2008a

Version: 2.7.1

New Features

"What's This?" Context-Sensitive Help Available for Simulink Configuration Parameters Dialog

R2008a introduces "What's This?" context-sensitive help for parameters that appear in the Simulink Configuration Parameters dialog. This feature provides quick access to a detailed description of the parameters, saving you the time it would take to find the information in the Help browser.

To use the "What's This?" help, do the following:

1 Place your cursor over the label of a parameter.

2 Right-click. A **What's This?** context menu appears.

For example, the following figure shows the **What's This?** context menu appearing after a right-click on the **Start time** parameter in the **Solver** pane.

3 Click **What's This?** A context-sensitive help window appears showing a description of the parameter.

R2007b

Version: 2.7

New Features

Interfacing with One-Dimensional Simscape Domains

The Prismatic-Translational Interface and Revolute-Rotational Interface blocks of the new Interface Elements library allow you to connect a SimMechanics Prismatic or Revolute joint primitive to a Physical Networks line connected to Simscape blocks. The Prismatic-Translational Interface and Revolute-Rotational Interface blocks transfer mechanical velocities and forces/torques as Physical Networks cross- and through-variables, respectively, without energy loss, along or around axes that you can define.

Several new demos illustrate the use of the Interface Elements blocks. See New Demos.

SimMechanics State Names Now Available to Simulink and Real-Time Workshop

You can access the names of the mechanical states of your SimMechanics models through the `mech_stateVectorMgr` command. The names of these states and the SimMechanics blocks that define them are now available to Simulink and appear in the outputs of model simulations. They also appear in simulations based on code generated with Real-Time Workshop.

See the Simulink for further details about states.

Controlling Redundant Constraint Analysis with New Tolerance Setting

You can now adjust the sensitivity of the SimMechanics redundant constraint analysis in the **Constraints** tab of the Machine Environment block, which allows you to choose between automatic constraint redundancy analysis or specifying a constraint redundancy tolerance explicitly.

Certain mechanical configurations are sensitive to the constraint redundancy tolerance and can spuriously lose or gain degrees of freedom if this tolerance is adjusted incorrectly.

New Demos

Nine new demos have been added.

Interface Blocks

These demos illustrate how to use the new Interface Elements blocks. See Interfacing with One-Dimensional Simscape Domains.

- mech_interface_crate_transfer
- mech_interface_dspring_damper
- mech_interface_hyd_cylinder
- mech_interface_hyd_slidercrank
- mech_interface_rot_spr_damper
- mech_interface_trans_spr_damper

Car Modeling

These new demos provide examples of sophisticated automotive modeling.

- `mech_car_handling`
- `mech_car_kinematics`

These demos shipped in the Version 2.6 (R2007a) SimMechanics product.

Code Generation Documentation Consolidated to Simscape User's Guide

Documentation of code generation features common to all Physical Modeling add-on products based on Simscape software has been consolidated to the *Simscape User's Guide*. The *SimMechanics User's Guide* continues to document uniquely SimMechanics features related to code generation.

R2007a

Version: 2.6

New Features

Compatibility Considerations

SimMechanics Software Now Requires Simscape Product

SimMechanics software now depends on and requires Simscape software, the foundation for Physical Modeling products. Simscape software includes common Physical Modeling utilities and block libraries.

Sharing Models Using Simscape Editing Modes

SimMechanics software now features a selection of two Simscape editing modes that allow full or restricted editing of models.

- The Restricted mode requires SimMechanics product to be installed, but does not require a license. It allows you to change a limited set of model parameters, but not the blocks or connections, in a SimMechanics model.
- The Full mode requires both Simscape and SimMechanics products to be installed. It allows you to change anything in a SimMechanics model.

Block Library Links Must Be Resolved

All SimMechanics blocks in your models must now have resolved block library links. You can neither disable nor break these library links. This is a global Simscape requirement. Consult the Simscape documentation for further details.

Compatibility Considerations

If you have an existing SimMechanics model with disabled or broken links from SimMechanics blocks to the SimMechanics block library, you must restore all the broken block library links for your model to be valid.

If you have disabled or broken the SimMechanics library link for blocks that you have customized and want to keep these modified blocks in your model, you must move these modified blocks to your own custom library or libraries, then copy the block instances that you need to your model.

You must still restore the block link to its parent library, whether that parent is the SimMechanics block library or your own.

Two Blocks Now with Tunable Parameters

These SimMechanics parameters are now tunable from their respective block dialogs:

- The **Gravity vector** field of the Machine Environment block.
- All three parameter fields of the Body Spring & Damper block.

Demos Expanded

Five new demos have been added.

Angular Motion Equivalence

The `mech_gimbal_transform` demo shows the equivalence of a body's angular motion as measured by Joint Sensors and as measured by Body Sensors.

Joint Limits

Two demos illustrate how to model limits or clearance on joint motion:

* `mech_jointlimit_prismatic`
* `mech_jointlimit_revolute`

The first illustrates limits on translational motion; the second, limits on rotational motion.

Stochastic Oscillator and Controller

Two demos simulate a mixture of deterministic and random forces with a model of a micromechanical damped linear oscillator:

* `mech_brownian_osc`
* `mech_brownian_osc_control`

The first model has no controller; the second, a control force in addition to the damped spring and white noise forces.

Modeling Constraints with SimMechanics Software

A new MATLAB digest article explains how to model mechanical constraints with SimMechanics blocks. The article uses a set of four existing SimMechanics demos as illustrations.

R2006b

Version: 2.5

New Features

Compatibility Considerations

Computer-Aided Design Translator for Pro/ENGINEER Available via the Web

A new computer-aided design (CAD) Pro/ENGINEER-to-SimMechanics translator is available. You install and use this translator with Pro/ENGINEER. It is independent of MATLAB.

The translator converts a Pro/ENGINEER CAD machine assembly into an XML file in the Physical Modeling format. The XML file represents the assembly in a form that you can use to generate a SimMechanics block diagram model dynamically equivalent to the original CAD assembly. To generate models, use the `import_physmod` command.

Installing the Pro/ENGINEER-to-SimMechanics Translator

Obtain and use the installer executable by locating and downloading its archive at `www.mathworks.com/products/simmechanics/`. A README page is provided that describes installation and configuration details.

Translator Documentation

Help files accompany the Pro/ENGINEER-to-SimMechanics translator. They include two HTML pages and a PDF book. These files are independent of the MATLAB help system.

Translator Examples

Some examples of Pro/ENGINEER CAD assemblies are included with the Pro/ENGINEER-to-SimMechanics translator.

One of the examples is a robot arm assembly. The corresponding XML file in Physical Modeling format, `robot.xml`, is located in the `toolbox/physmod/mech/mechdemos/` folder, relative to your MATLAB root.

Compatibility Considerations

Versions of Pro/ENGINEER compatible with the Pro/ENGINEER-to-SimMechanics translator are listed at the MathWorks Web site. The translator is available only for the Microsoft Windows (32-bit) operating system.

SolidWorks-to-SimMechanics Translator Now Available Only via the Web

The existing SolidWorks®-to-SimMechanics translator for the SolidWorks CAD platform continues to be available via the Web, at `www.mathworks.com/products/simmechanics/`. However, it no longer ships with the SimMechanics product.

Compatibility Considerations

Versions of SolidWorks compatible with the SolidWorks-to-SimMechanics translator are listed at the MathWorks Web site.

The SolidWorks CAD platform and the SolidWorks-to-SimMechanics translator are available only for the Microsoft Windows (32-bit) operating system.

New Demo

This version of SimMechanics software includes one new demo model, `mech_brownian_osc`, a stochastic linear harmonic oscillator based on micro- and biomechanics.

Modeling Flexible Bodies with SimMechanics Software

A new MATLAB Digest article and related technical paper explain how to simulate flexible bodies with SimMechanics models and third-party finite element analysis applications. A set of models accompanies the technical paper in a compressed zip archive available from MATLAB Central.

R2006a

Version: 2.4

New Features

New Block Dialogs and the Simulink Model Explorer

Most of the SimMechanics block dialogs have been upgraded and are now compatible with the Simulink Model Explorer. See the Simulink documentation for more about Model Explorer.

Block Dialogs Not Rendered in Model Explorer

Model Explorer cannot render the dialogs of these blocks:

- Shared Environment
- Point-Curve Constraint

Joint Spring & Damper Adds Extra Simulink State When Connected to a Revolute

Each Joint Spring & Damper block connected to a revolute primitive now adds a Simulink state to your model. These states are in addition to other normal Simulink states, such as those associated with Integrator and Transfer Fcn blocks. See the Simulink documentation for more about Simulink model states.

This new feature does not change the mechanical states of your model, those associated with SimMechanics joint blocks. See the mech_stateVectorMgr command reference for more about mechanical states.

Improved SolidWorks CAD Translation

The SolidWorks-to-SimMechanics translator and Physical Modeling XML import command have been improved. See Importing Mechanical Models and the `import_physmod` command reference in the SimMechanics User's Guide.

- Mates in a CAD assembly (restrictions on the free motion of CAD parts) are now translated into Joint blocks that reflect a particular combination of joint primitives, such as Prismatic, Revolute, Spherical, Bushing, etc. Previously, all degrees of freedom in a CAD assembly were translated into appropriately configured Custom Joint blocks. Rigid connections between parts continue to be translated into Welds.
- Blocks in SimMechanics models generated from Physical Modeling XML files are now, by default, labeled with shorter names. You can change the naming convention in the `import_physmod` command options.

R14SP3

Version: 2.3

New Features

New Shared Environment Block

The new Shared Environment block allows you to link two separate SimMechanics block diagrams with a nonphysical connection. The block enforces the same machine environment settings on both machines but adds no mechanical components (bodies or degrees of freedom) to either. Two machines so connected require exactly one Machine Environment block, rather than two.

See the Shared Environment block reference for further details.

Updated Body Spring & Damper Block

The Body Spring & Damper block no longer contains any degrees of freedom.

Previous versions of the Body Spring & Damper block contained six degrees of freedom (DoFs) represented by a Joint. This Joint had to be taken into account when counting DoFs and loops in your model.

The new Body Spring & Damper block instead uses a Shared Environment block to connect the two sides and contains no DoFs. See their respective block references for more information.

Normalizing Rotations

The Body dialog accepts various rotational forms (Euler angles, rotation matrix, quaternion). The Body block now interprets these rotational forms in a slightly different way, so that the standard normalization conditions on rotation matrices and quaternions are consistently enforced. As a result, you might now see slightly different simulation results from earlier versions.

New State Vector Commands

Four new SimMechanics commands allow you to manipulate the mechanical states of your models. See the command references for more.

- `mech_get_states`
- `mech_runtime_states`
- `mech_set_states`
- `mech_transfer_states`

New import_physmod Dialog

The `import_physmod` command generates SimMechanics models from Physical Modeling XML files. In addition to importing Physical Modeling XML through the command line, you can now set up the command through a dialog.

See the `import_physmod` command reference for further information.

Demos Expanded

Two new demos have been added:

- `mech_flexible_four_bar` models a flexible four bar machine.

- `mech_pole_vault` models a rigid pole vaulter, with a discrete event.

The second demo has an associated MAT-file.

Documentation Enhancements

The *SimMechanics User's Guide* has been enhanced.

- The review on representing motion has been revised.
- The analyzing motion has been revised.
- The computer-aided design (CAD) documentation (also available as a separate book) has been significantly revised. See the CAD chapter.
- A new chapter of case studies, based on the Stewart platform, has been added. The studies apply advanced methods and tasks to this system. SimMechanics software includes a related set of demo model and library files.

  - Degrees of freedom, states, closed topology, and constraints
  - Trimming and linearization
  - Designing controllers for the platform
  - Generating code versions of the model
  - Implementing code on dedicated hardware (hardware in the loop)

R14SP2

Version: 2.2.2

New Features

Demos Expanded

Two new demos have been added:

- mech_sticky_box
- mech_sticky_screw

Each demo has an associated MAT-file.

These demos illustrate static friction (stiction) implemented with the Joint Stiction Actuator.

Documentation Enhancements

The online SimMechanics Help has been enhanced with links to four new Web-based AVI animation files (one of them new in this release) that record SimMechanics simulations.

You can access these links most easily by using the Examples index in the Help browser, under **Prerecorded Animations**.