Simulink® Control Design™ Release Notes

How to Contact MathWorks

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

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

508-647-7000 (Phone)

508-647-7001 (Fax)

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

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

Simulink® Control Design™ Release Notes

© COPYRIGHT 2004–2013 by The MathWorks, Inc.

Trademarks

Patents

Contents

R2012a

R2011b

R2011a

R2010b

R2010a

R14SP3

R14SP2

R2013b

Version: 3.8

New Features: Yes

Bug Fixes: Yes

Enhanced `linearize` command, providing faster batch linearization for model parameter variations

`linearize` now allows you to efficiently batch linearize a model for variations in model parameter values. Use this feature to study the effects of varying a model parameter value. For example, you can vary plant parameter values and analyze the controller robustness to plant model uncertainty. The software uses only one model compilation, making batch linearization faster, especially for models that are expensive to compile repeatedly.

For more information, see "Batch Linearize Model for Parameter Value Variations Using linearize" and "Specify Parameter Samples".

`slLinearizer` interface, providing faster batch linearization for multiple I/O sets

Use the `slLinearizer` interface to efficiently batch linearize a model. The interface includes linearization commands that you use to extract any open-loop or closed-loop transfer function for varying operating points and parameter values. For instance, you can extract linearizations for the plant model, overall closed-loop transfer function, and plant disturbance rejection, at multiple operating points, for varying plant/controller parameter values.

`slLinearizer` extends the functionality of `linearize` and uses only one model compilation to compute multiple open-loop and closed-loop transfer functions for varying operating points and parameter values. This efficiency is especially advantageous for models that are expensive to compile repeatedly.

For examples that illustrate how to use `slLinearizer`, see:

- "Vary Parameter Values and Obtain Multiple Transfer Functions Using slLinearizer"

- "Vary Operating Points and Obtain Multiple Transfer Functions Using slLinearizer".

`linearizeOptions` and `findopOptions` for specifying options for linearization and operating point search

Compatibility Considerations: Yes

The new options commands `linearizeOptions` and `findopOptions` replace the command `linoptions`. This change simplifies the use of options for linearization and operating point search by separating the available options for each task. See the reference pages for these options commands for more information about how to use them.

Compatibility Considerations

The `linoptions` command will be removed in a future release, and issues a warning as of this release. If you have scripts or functions that use `linoptions`, consider updating them to instead use `linearizeOptions` for linearization and `findopOptions` for operating point search.

Highlight linear analysis points in Linear Analysis Plot Blocks and Model Verification Blocks

The Linear Analysis Plot Blocks and Model Verification Blocks now include a button that highlights signals in your Simulink® model which you have selected as linearization inputs or outputs. This highlighting makes it easier to identify linear analysis signals in your model when you are working in the block dialog box.

In the Linearizations tab of the block, select a signal from the **Linearization inputs/outputs** table. Then click the highlight button:
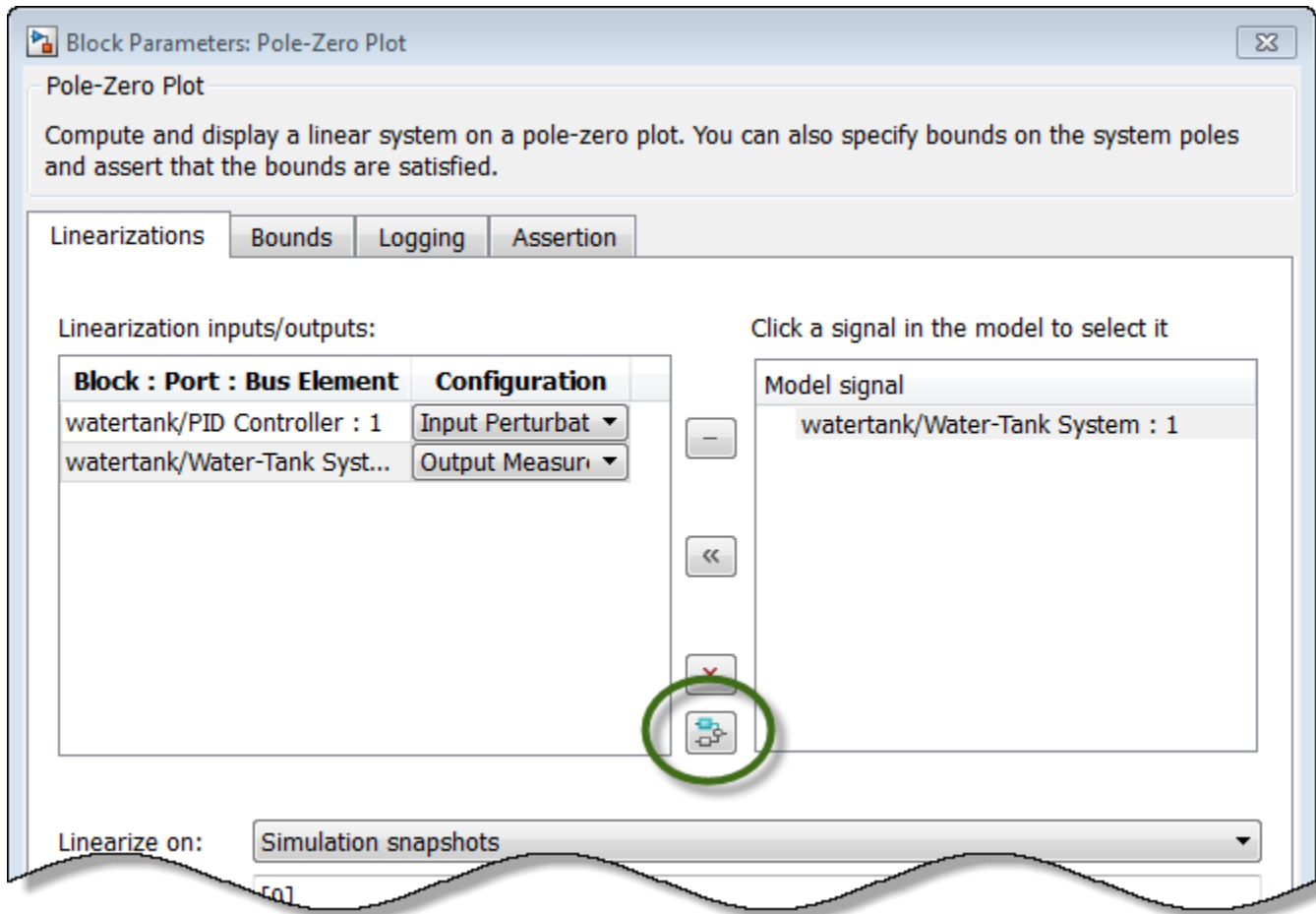
The block that originates the selected signal is highlighted in the Simulink model.

`writeBlockValue` **command can update Simulink model with tuned parameter values from generalized LTI model**

The `writeBlockValue` method of `slTunable` has new syntaxes that allow you to update a Simulink model with tunable parameter values from a generalized LTI model.

`writeBlockValue(ST,M)` updates the current values of the tunable parameters in an `slTunable` interface, ST, to match their values in a generalized model, M. The command also updates the corresponding parameters in the Simulink model associated with ST. `writeBlockValue(ST,M,BlockID)` updates only the parameters in the specified blocks.

These syntaxes are useful when you use Robust Control Toolbox™ tuning commands such as `systune`, `looptune`, or `hinfstruct` to tune system responses that you extract from the `slTunable` interface. For example, suppose you extract a particular closed-loop response from an `slTunable` interface using `getIOTransfer`. You then use `hinfstruct` to tune this closed-loop response. To validate the tuned parameters in the full non-linear Simulink model, use `writeBlockValue` to write the tuned parameter values back into the model.

For more information, see the `slTunable.writeBlockValue` reference page.

Format of `BlockData` structure identical for snapshot and operating point linearization
Compatibility Considerations: Yes

When you specify the linearization of a block as a function, the input argument to that function is the structure `BlockData`, whose fields are given in the `linearize` reference page. The field `BlockData.Inputs` is now a data structure array, regardless of whether you are linearizing the model at a snapshot time or at a specified operating point. The data structure `BlockData.Inputs` has the following fields:

- `BlockName` — Contains the name of the block whose output connects to the input of the block whose linearization you specifying. For example, if you

are specifying the linearization of a block called `Dynamics`, and the second input of `Dynamics` is driven by a signal from a block called `Torque`, then `BlockData.Inputs(2).BlockName` is the full block path name of `Torque`.

- `PortIndex` — Identifies which output port of `BlockName` corresponds to the input of the block whose linearization you are specifying. For example, if the third output from `Torque` drives the second input of `Dynamics`, then `BlockData.Inputs(2).PortIndex = 3`.

- `Values` — The value of the signal specified by `BlockName` and `PortIndex`. If this signal is a vector-valued signal, `Values` is a vector of corresponding dimension.

Previously, `BlockData.Inputs` had this format only for snapshot linearization. For linearization at a specified operating point, `BlockData.Inputs` was a numeric array of input values.

Compatibility Considerations

If you have scripts or functions that assume `BlockData.Inputs` is a numeric array, update your code to reflect the new structure of `BlockData.Inputs`.

Linear Analysis Blocks and Model Verification Blocks save operating points with computed linear systems

The data logging capability of the Linear Analysis Blocks or Model Verification Blocks now includes an option to save the operating points corresponding to the computed linear systems. When you check **Save data to workspace** and **Save operating points for each linearization** in the Logging tab of the block dialog box, a field named `operatingPoints` is added to the data structure containing the logged data. This field stores the operating point corresponding to each logged linear system in the data structure.

R2013a

Version: 3.7

New Features: Yes

Bug Fixes: Yes

Transient behavior slider added to PID Tuner for increased control over reference tracking and disturbance rejection performance

The PID Tuner now has a **Transient behavior** slider for emphasizing either reference tracking or disturbance rejection. When you open the PID Tuner, the tool starts in the `Time domain` design mode, displaying a step plot of the reference tracking response. The new **Transient behavior** slider is beneath the **Response time** slider.
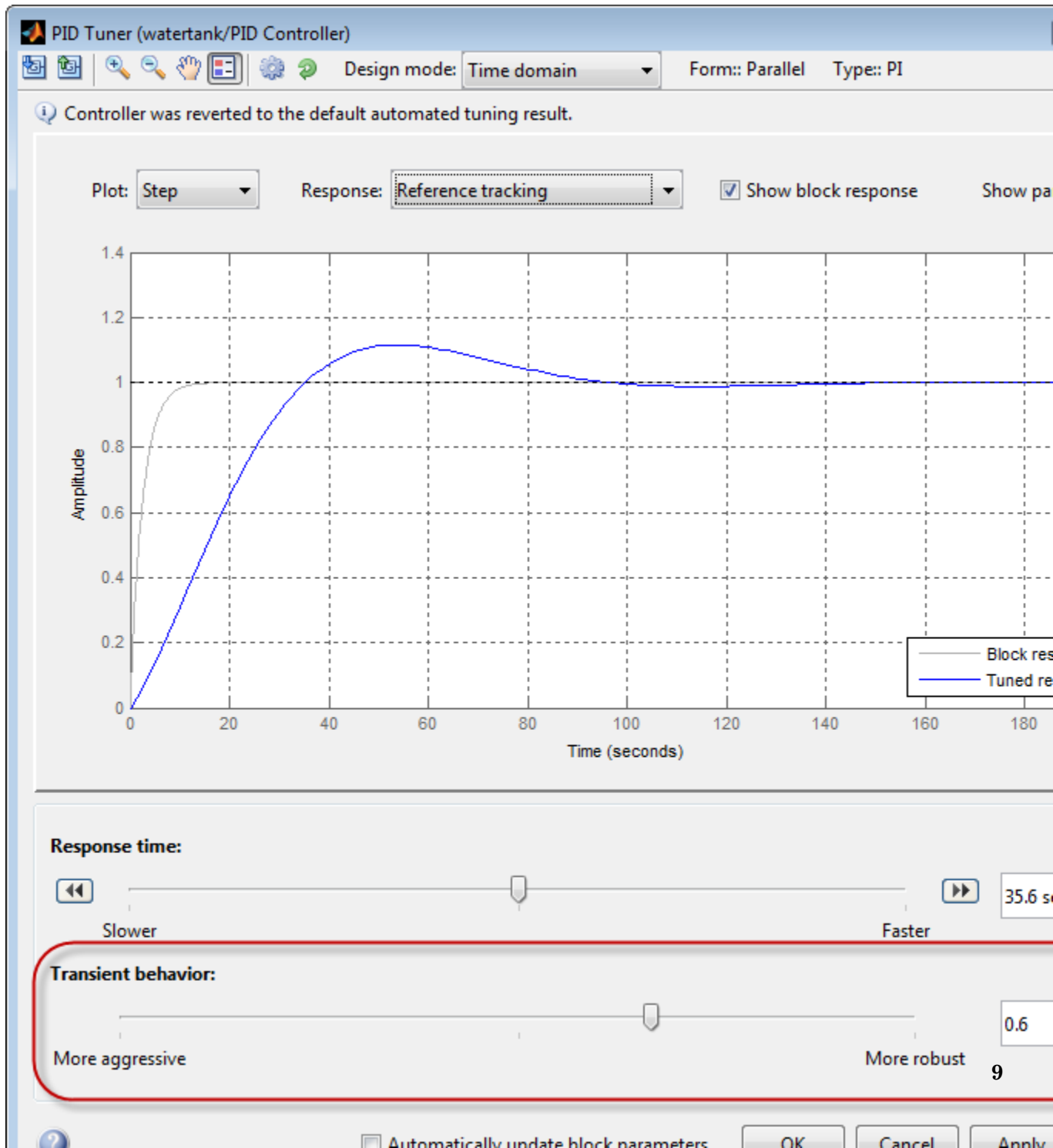
You can use the **Transient behavior** slider when:

- The tuned system's disturbance rejection response is too sluggish for your requirements. In this case, try moving the **Transient behavior** slider to the left to make the controller more aggressive at disturbance rejection.

- The tuned system's reference tracking response has too much overshoot for your requirements. In this case, try moving the **Transient behavior** slider to the right to increase controller robustness and reduce overshoot.

In `Frequency domain` design mode, the PID Tuner has **Bandwidth** and **Phase margin** sliders. These sliders are the frequency-domain equivalents of the **Response time** and **Transient behavior** sliders, respectively.

For an example illustrating the balance between reference tracking and disturbance rejection, see Tune PID Controller to Balance Tracking and Disturbance Rejection Performance.

Linear analysis points redesigned to clarify I/O types and loop openings

Linear analysis points (linearization inputs, outputs and loop-opening locations), that you use to specify the portion of the Simulink model to linearize, have been redesigned. The redesign clarifies what the I/O point means and helps you select the right type of point to compute the desired response.

The analysis points that you select in the **Linear Analysis Points** submenu or the Linear Analysis Tool have been renamed. The points also have updated icons that correspond to markers that appear on your model to indicate the linear analysis point type. The corresponding strings for the `type` argument in the `linio` command have also been renamed.
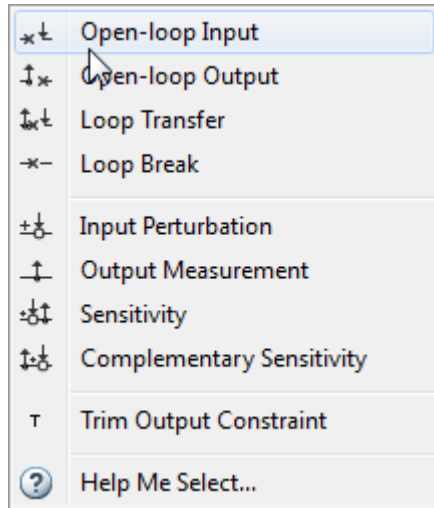
You can also specify inputs or outputs as open loop while specifying the I/O type. For example:

- In the **Linear Analysis Points** submenu or the Linear Analysis Tool, select **Open-loop Output**.

- At the command line, type `io(2)=linio('magball/Magnetic Ball Plant',1,'openoutput')`.

Thus, when performing tasks such as plant linearization or open-loop linearization, you do not need to specify the opening separately.

The following table maps the renamed analysis points to the point types in previous releases:

Command Line		Linear Analysis Points Submenu or Linear Analysis Tool	
Previous Releases	**R2013a**	**Previous Releases**	**R2013a**
`type = 'in'` and `openloop = 'on'`	`type = 'openinput'`	**Input Point** and **Open Loop**	**Open-loop Input**
`type = 'in'` and `openloop = 'off'`	`type = 'input'`	**Input Point**	**Input Perturbation**
`type = 'out'` and `openloop = 'on'`	`type = 'openoutput'`	**Output Point** and **Open Loop**	**Open-loop Output**
`type = 'out'` and `openloop = 'off'`	`type = 'output'`	**Output Point**	**Output Measurement**
`type = 'inout'` and `openloop = 'off'`	`type = 'sensitivity'`	**Input-Output**	**Sensitivity**
`type = 'outin'` and `openloop = 'on'`	`type = 'looptransfer'`	**Output-Input** and **Open Loop**	**Loop Transfer**
`type = 'outin'` and `openloop = 'off'`	`type = 'compsensitivity'`	**Output-Input**	**Complementary Sensitivity**
`type = 'none'` and `openloop = 'on'`	`type = 'loopbreak'`	**Open Loop**	**Loop Break**

Use the context-sensitive help, or see the `linio` reference page to assist you in selecting an analysis point. For examples on how to specify linear analysis points graphically, see Specify Portion of Model to Linearize in Simulink Model and Specify Portion of Model to Linearize in Linear Analysis Tool.

Linear Analysis Blocks and Model Verification Blocks save data in single object with Simulink model logging output
Compatibility Considerations: Yes

The data logging option in the Linear Analysis Blocks or Model Verification Blocks saves linear systems computed by the blocks. As of R2013a, when you configure your Simulink model to save simulation output in a single object, the logging output of these blocks is saved as a field in the simulation output object. Previously, these blocks saved logging output as a separate variable in the MATLAB® workspace, regardless of the model configuration settings.

For more information about data logging in Simulink, see Export Simulation Data and the `Simulink.SimulationOutput` reference page.

Compatibility Considerations

If you have a Simulink model which you have configured to save simulation output in a single object, and you use data logging option in a Linear Analysis Block or Model Verification Block, the resulting logging output is no longer a separate variable in the MATLAB workspace. If you use the logging output in any scripts, update those scripts to use the `Simulink.SimulationOutput` object instead of a separate variable.

R2012b

Version: 3.6

New Features: Yes

Bug Fixes: Yes

MATLAB code generation from Linear Analysis Tool for batch estimation of model frequency responses

You can now generate MATLAB code for frequency response estimation from the Linear Analysis Tool. You can generate either a MATLAB script or a MATLAB function. Generated MATLAB scripts are useful when you want to programmatically reproduce a result you obtained interactively. A generated MATLAB function allows you to perform multiple estimations with systematic variations in estimation parameters such as operating point (batch estimation).

For more information, see Generate MATLAB Code for Repeated or Batch Frequency Response Estimation.

Operating point calculation (trimming) from multiple specifications with only one model compilation

The findop command now can find operating points for multiple operating point specifications with a single model compilation. This feature allows you to find multiple trimmed operating points without the overhead of compiling the model for each trimming computation. For an example, see Batch Compute Operating Points with Single Model Compilation.

For more information about operating point calculation and operating point specifications, see the findop and operspec reference pages.

Export and import operating point specifications in Linear Analysis Tool

When you modify an operating point specification in the Linear Analysis Tool, you can now export the specification to the MATLAB workspace. Exported specifications are saved as operating point specifications objects (see operspec). Exporting specifications can be useful when you expect to perform multiple trimming operations using the same or a very similar set of specifications. Additionally, you can export interactively-edited operating point specifications when you want to use the findop command to perform multiple trimming operations with a single compilation of the model (see

"Operating point calculation (trimming) from multiple specifications with only one model compilation" on page 16).

You can also import saved operating point specifications to the Linear Analysis Tool and use them to interactively compute trimmed operating points. Importing a specification can be useful when you want to trim a model to a specification that is similar to one you previously saved. In that case, you can import the specification to the Linear Analysis Tool and interactively change it. You can then export the modified specification, or compute a trimmed operating from it.

For more information, see Import and Export Specifications For Operating Point Search.

For more information about operating point specifications, see the `operspec` and `findop` reference pages.

MATLAB script or function generation from Linear Analysis Tool for repeated or batch linearization

When you generate MATLAB code for linearization from the Linear Analysis Tool, you now have a choice of generating a script that uses the current linearization parameters, or a function that takes parameter values as input. Previously, you could only generate a MATLAB function with no input parameters.

Generated MATLAB scripts are useful when you want to programmatically reproduce a result you obtained interactively. A generated MATLAB function allows you to perform multiple linearizations with systematic variations in linearization parameters such as operating point (batch linearization).

For more information, see Generate MATLAB Code for Repeated or Batch Linearization.

Print plots to MATLAB figure in Linear Analysis Tool

You can now export a plot from the Linear Analysis Tool to a MATLAB figure window.

For more information, see Print Plot to MATLAB Figure in Linear Analysis Tool.

Commands for setting and querying rate conversion methods in tunable blocks

New commands allow you to query and specify the rate conversion method that the `slTunable` interface uses for converting the sampling time of the parametrization of tunable blocks in a Simulink model.

- `slTunable.getBlockRateConversion` — Query rate conversion method of tunable block
- `slTunable.setBlockRateConversion` — Set rate conversion method of tunable block

These commands are useful, for example, when you use the `slTunable` interface to tune a mixed-rate Simulink model, such as a model having a discrete-time controller and a continuous-time plant. The `slTunable` interface automatically converts the sampling times of tunable blocks where necessary. Using `slTunable.setBlockRateConversion`, you can control the conversion method that the `slTunable` interface uses.

For more information, see:

- Tuning of a Digital Motion Control System
- `slTunable.getBlockRateConversion` and `slTunable.setBlockRateConversion` reference pages

"Ignore saturation when linearizing" checked by default in PID Controller and PID Controller (2DOF) blocks

The default linearization behavior of the PID Controller and PID Controller (2DOF) blocks now forces linearization commands to ignore block output limits. Ignoring output limits allows you to linearize a model around an operating point even if that operating point causes the PID Controller block to exceed the output limits.

To cause linearization commands not to ignore block output limits, clear the **Ignore saturation when linearizing** checkbox in the block dialog box, PID Advanced tab.

showBlockValue **renamed to** showTunable
Compatibility Considerations: Yes

The command `showBlockValue` is now called `showTunable`. Use `showTunable` to obtain the current value of block parametrizations from an `slTunable` interface.

Compatibility Considerations

Replace instances of `showBlockValue` in your code with `showTunable`.

R2012a

Version: 3.5

New Features: Yes

Bug Fixes: Yes

Create Linearization Input/Output Sets in the Linear Analysis Tool

You can now interactively create I/O sets for linearization or frequency response estimation in the Linear Analysis Tool, without adding linearization points to your model. Previously, you had to modify your model to interactively create I/O sets.

For more information, see Create Linearization I/O Sets In Linear Analysis Tool.

For more information about using the Linear Analysis Tool, see Linearize at Model Operating Point and Estimating Frequency Response.

Specify Feedback Sign for getLoopTransfer Without Specifying Loop Openings

An additional syntax for `slTunable.getLoopTransfer` allows you to specify the feedback sign without having to provide an `openings` argument. Use the syntax

```
L = getLoopTransfer(ST,location,sign)
```

to calculate the point-to-point open-loop transfer function at `location` of the Simulink model described by the `slTunable` interface `ST`. The software uses the sign convention specified by `sign` (+1 or 1) to compute `L`.

Previously, to specify a feedback sign, you had to use the syntax

```
L = getLoopTransfer(ST,location,openings,sign)
```

This syntax required you to specify an `openings` argument to specify `sign`. The new syntax allows you to specify `sign` while using the default loop openings.

For more information, see the `slTunable` and `slTunable.getLoopTransfer` reference pages.

R2011b

Version: 3.4

New Features: Yes

Bug Fixes: Yes

Redesigned Graphical Tool for Improved Linear Analysis Workflows

The new interactive Linear Analysis Tool streamlines workflows for linear analysis tasks such as finding steady-state operating points (trimming) and linearization.

To access the Linear Analysis Tool:

1 Open a Simulink model.

2 In the Simulink model, select **Tools > Control Design > Linear Analysis**.

For more information about using the Linear Analysis Tool for operating-point and linearization workflows, see:

• Steady-State Operating Points

• Linearization

Interactive Frequency Response Estimation and Validation of Linearization Results

The new interactive Linear Analysis Tool provides a graphical interface for frequency response estimation.

To access the Linear Analysis Tool:

1 Open a Simulink model.

2 In the Simulink model, select **Tools > Control Design > Linear Analysis** to launch the Linear Analysis tool.

3 Click the Frequency Response Estimation tab to begin an estimation task.

For more information about using the Linear Analysis Tool for frequency response estimation, see Frequency Response Estimation.

Optimization of Model Parameters to Meet Design Requirements Specified by Model Verification Blocks

If you have Simulink Design Optimization™ software, you can optimize the Simulink model to meet frequency-domain requirements specified in Model Verification blocks. For example, you can optimize the linear system to meet Bode magnitude or gain and phase margin requirements. For more information, see Design Optimization To Meet Frequency-Domain Requirements (GUI).

You can also include time-domain requirements such as step response characteristics for optimization. For more information, see Design Optimization to Meet Time- and Frequency-Domain Requirements.

Automatic Tuning of PID Controller Blocks in a Referenced Model

You can now use the PID Tuner to tune a PID Controller block in a model that is referenced in one or more open models. When you launch the PID Tuner from a block within a model reference, the software prompts you to select which of the open models is the top-level model for analysis and tuning. Previously, the PID Tuner could only use the model containing the PID Controller block as the top-level model.

For more information, see Tuning a PID Controller Within a Model Reference.

Control System Tuning for Simulink Models with looptune or hinfstruct Using slTunable Interface

If you have Robust Control Toolbox software, you can use tuning commands, such as `slTunable.looptune` and `hinfstruct`, to tune control systems modeled in Simulink. The `slTunable` object provides an interface between your Simulink model and these commands.

Use `slTunable` to specify information about your control structure and parametrization. `slTunable` also automates tasks such as linearizing the Simulink model, parametrizing the tunable blocks of your system, and applying tuned parameter values to the model. After you create and configure

an `slTunable` object for your control architecture, you can tune the control system using `slTunable.looptune` or `hinfstruct`.

For more information, see Tuning Fixed Control Architectures in the Robust Control Toolbox documentation.

Change in Default Number of Samples in frest.Chirp
Compatibility Considerations: Yes

The default number of samples (`NumSamples`) in a `frest.Chirp` input signal for frequency response estimation is now given by the formula:

$$\frac{4\pi}{Ts * \min(FreqRange)}.$$

`Ts` is the sampling time of the chirp signal. `FreqRange` is the vector of signal frequencies of the chirp signal.

This formula returns twice the value returned in previous releases.

For example, if you create a default chirp input signal with the command

```
input = frest.Chirp
```

the value of `input.NumSamples` is 10000, instead of the previous value of 5000.

For more information, see the `frest.Chirp` reference page.

Compatibility Considerations

If you have scripts that rely on the default `NumSamples` formula, modify your scripts to account for the new value.

R2011a

Version: 3.3

New Features: Yes

Bug Fixes: Yes

Ability to Select Individual Bus Elements as Linearization Input and Output Points

Instead of selecting an entire bus, you can now select individual elements of the bus signal as linearization inputs or outputs (I/Os). By selecting individual bus elements, you can:

- Obtain the linearization only for the channels of interest.

- Specify multiple I/Os, possibly as different types, in the same bus.

Select individual bus elements as I/Os when you want to:

- Linearize a Simulink model using the GUI or programmatically.

- Visualize time- and frequency-domain response during simulation using the Linear Analysis Plots blocks.

- Verify frequency-domain characteristics using the Model Verification blocks.

For more information, see Select Individual Bus Elements as Linearization Points and the linio reference page.

Enhanced LINLFT Command Optionally Returns Linearization of Excluded Blocks

A new optional output argument to linlft returns the linearization of blocks you specify for exclusion from the linearized model.

For more information, see the linlft reference page.

Access to Current Linearization of a Simulink Block for Specifying Custom Linearization

When you use a specification function to specify a custom linearization of a Simulink block, you can now access the current linearization in your function. The current linearization is stored in the new BlockLinearization field of

the `BlockData` structure. The software automatically creates the `BlockData` structure and passes it to your block linearization specification function.

For more information and an example showing how to use the `BlockLinearization` field of `BlockData`, see Augmenting the Linearization of a Block.

Enhanced PID Controller Blocks Display Compensator Formula in Block Dialog Box

The PID Controller and PID Controller (2DOF) blocks now display the current compensator formula in the block dialog box. This display reflects the current settings for controller type, controller form, and time domain.

R2010b

Version: 3.2

New Features: Yes

Bug Fixes: Yes

New Blocks for Plotting and Verifying Linear System Characteristics of Simulink Models

This version of Simulink Control Design™ software provides new blocks for:

- "Plotting Linear System Characteristics of Simulink Models" on page 32.

- "Verifying Linear System Characteristics of Simulink Models" on page 32.

Plotting Linear System Characteristics of Simulink Models

Simulink Control Design software provides six new blocks in the **Linear Analysis Plots** library. Use these blocks to plot the time- and frequency-domain characteristics of a linear system computed from a nonlinear Simulink model. The linear system is computed and plotted during simulation at:

- Simulation snapshot times. The default snapshot time is 0.

- Trigger-based simulation events.

You can also use these blocks to specify bounds on the linear system characteristics, and view the bounds on the plot.

Because these blocks are same as the Model Verification blocks except for the default settings of the bound parameters, you can optionally use the blocks to verify that the bounds are satisfied during simulation.

For more information, see:

- Visualize Bode Response of Simulink Model During Simulation

- Linear Analysis Plots block reference pages

- Plotting Linear System Characteristics of a Chemical Reactor demo

Verifying Linear System Characteristics of Simulink Models

Simulink Control Design software provides six new blocks in the **Model Verification** library. Use these blocks to verify that the time- and frequency-domain characteristics of a linear system, computed from a nonlinear Simulink model, satisfy specified bounds during simulation. For

example, you can verify whether the linearized behavior of your model satisfies upper and lower magnitude bounds on a Bode plot or gain and phase margins.

You can perform the verification during simulation at:

- Simulation snapshot times. The default snapshot time is 0.
- Trigger-based simulation events

Because these blocks are same as the Linear Analysis Plots blocks except for the default settings of the bound parameters, you can use the blocks to view the bounds on time- and frequency-domain plots.

You can also use these blocks with the Simulink Model Verification library blocks to design complex logic for model verification.

If you have Simulink Verification and Validation™ software, you can construct simulation tests for your model using the Verification Manager.

For more information, see:

- Model Verification in the Simulink Control Design documentation.
- Model Verification block reference pages
- Verifying Frequency-Domain Characteristics of an Aircraft demo

New Tools for Identifying Time-Varying Source Blocks for Frequency Response Estimation

When you are performing frequency response estimation, time-varying source blocks in the signal path can skew the results. Previously, to obtain accurate estimation, you manually identified source blocks and inserted open-loop points in your model to disable them.

The new `frest.findSources` command automatically detects time-varying source blocks in the signal path of the output linearization points in your Simulink model. Additionally, a new option to the `frestimate` command, `BlocksToHoldConstant`, allows you to disable these blocks during frequency response estimation.

For more information, see the `frest.findSources` and `frestimateOptions` reference pages.

There is also a new Model Advisor check for detecting time-varying source blocks in your model. For more information about using the Model Advisor, see Consulting the Model Advisor in the *Simulink User's Guide*.

Tuning Tools Update Workspace Variables That Define Parameters of Tuned Blocks

When you use Simulink Control Design tools to tune a block with parameters defined as workspace variables, the software now updates the values of the workspace variables. This update occurs when you write the compensator design to your Simulink model. The block remains parametrized by the variables. The software can update variable parameters that represent numerical values or `Simulink.Parameter` objects, whether they reside in the base workspace or the model workspace.

Previously, tuning a block parametrized by workspace variables overwrote the block parameters with numerical values. Thus, such tuning did not change the value of the workspace variables.

Enhanced PID Tuner Including New Response Plots

This release introduces several enhancements to the PID Tuner, including:

- New response plot options let you analyze system responses in either time domain (step plot) or frequency domain (Bode plot). Available responses are reference tracking, input and output disturbance rejection, controller effort, open loop, and plant.

- New options when the plant linearizes to zero at the operating point defined in the Simulink model.

  When the plant linearizes to zero on launch of the PID Tuner, the PID Tuner provides an option to linearize at a different operating point. The PID Tuner also lets you import an LTI model of your plant, such as a model obtained by frequency response estimation. Previously, the PID Tuner did not launch when the plant linearized to zero.

For an example illustrating these options, see the new Simulink Control Design demo, Designing PID Controller in Simulink with Estimated Frequency Response.

For more information about using the PID Tuner, see Automatic PID Tuning in the *Simulink Control Design User's Guide*.

New Demo Illustrating Control Design for a Plant That Has Parameter Variations

The Reference Tracking of a DC Motor with Parameter Variations demo illustrates compensator design for a plant that has parameter variations. The demo shows the following techniques:

- Performing batch linearization to obtain multiple linear models from a single Simulink model

- Using the new SISO Design Tool ability to analyze a control design for multiple models simultaneously

R2010a

Version: 3.1

New Features: Yes

Bug Fixes: Yes

New Parallel Computing Support For Frequency Response Estimation

If you have the Parallel Computing Toolbox™ software installed, you can use parallel computing to speed up frequency response estimation.

For more information, see Speeding Up Estimation Using Parallel Computing in the Simulink Control Design documentation.

New Commands Support Recomputing Frequency Response Estimation Results at Specific Frequencies

If you use the sinestream input signal for estimation, you can now recompute your frequency response estimation for only those specific frequencies that do not reach steady state. Previously, you had to redo the entire estimation if a few frequencies needed recomputing.

For more information, see Time Response Not at Steady State.

New frest.simcompare Output Argument Returns Simulation Output Data From Linear System

This release provides enhanced `frest.simCompare` functionality, which allows the return of simulated output data. If your linear model uses state-space representation, you can also return the state vector.

For more information, see the `frest.simCompare` reference page.

New Options in Simulink Results Viewer GUI for Viewing Frequency Response Estimation Results

The Simulation Results Viewer GUI, which you display using `frest.simView`, now provides additional options for analyzing your frequency response estimation:

- You can now enable phase unwrapping in the frequency response plot. To unwrap phase, right-click the Bode plot, and select **Unwrap Phase**.

- You can also import frequency response estimation results into an open Simulation Results Viewer, which replaces the existing results. Previously, you had to open a new Viewer each time you viewed estimation results. To import results, select **File > Import** in the Simulation Results Viewer.

New Option for Labeling Bus Signal I/O Names in the SISO Design Task

You can now configure the SISO Design Task to automatically label model I/Os using bus signal names. Model I/O labels describe available loops in the Graphical Tuning tab and label response plots in the Analysis Plots tab of the SISO Design Task GUI. Previously, labels for bus signals were based on the block and port path, or the Simulink signal name.

To use this option, select **Tools > Options** in the SISO Design Task GUI, and enable **Use bus signal names to label model I/O**.

Existing Simulink Blocks Now Have Analytic Jacobians

The following Simulink blocks now have Analytic Jacobians, which are optimized for memory consumption in large-scale models:

- Switch
- Multiport Switch
- Math Function (Transpose and Hermitian Transpose)

For more information on each block, see the block reference pages.

Change in Format of Time Series in frestimate Output
Compatibility Considerations: Yes

When you use the `simout` output of the `frestimate` command, you obtain a cell array of `Simulink.Timeseries` objects. Previously, the time dimension was always the first dimension of each `Simulink.Timeseries` object.

As of R2010a, the time dimension of each `Simulink.Timeseries` in `simout` is:

- The first dimension, if the time series is 2-D

- The last dimension, if the time series is 3-D or higher

Compatibility Considerations

If you have scripts that run `frestimate` on a model having an output point on a vector or matrix signal and perform operations on the resulting 3-D or higher-dimensional `Simulink.Timeseries` object in `simout`, modify your scripts to reflect the new placement of the time dimension in the time series.

For additional information, see:

- `frestimate` reference page
- `Simulink.Timeseries` reference page

R2009b

Version: 3.0

New Features: Yes

Bug Fixes: Yes

New GUI for Tuning New PID Controller Blocks

You can automatically tune the new Simulink PID Controller blocks in the PID Tuner GUI R2009b introduces. You launch the PID Tuner directly from the PID block dialogs boxes. These blocks are available in the Continuous and Discrete Simulink libraries.

For more information on tuning PID Controller using the PID Tuner, see Automatic PID Tuning.

New Automated PID Tuning Algorithm

You can now tune compensators using a new automated PID tuning algorithm called Robust Response Time, which is available in the SISO Design Task. You specify the open-loop bandwidth and phase margin, and the software computes PID parameters to robustly stabilize your system.

For information on tuning compensators using automated tuning methods, see Completing the Design.

Ability to Compute Frequency Response of Simulink Models

You can now compute the frequency response of a Simulink model using simulation. You use new commands to easily create input signals and simultaneously simulate and estimate the frequency response without changing your model. You can also use this capability to validate the accuracy of exact linearizations.

Ability to Specify the Linearization of Simulink Blocks

You can now specify the linearization of Simulink blocks, subsystems, and model references without having to replace any block in your model. You can specify the linearizations as LTI models or Robust Control Toolbox uncertain models.

Ability to Design Compensators for Plant Models With Time Delays

You can now design compensators for plants with exact time delay representations. Previously, you had to specify a Padé approximation before designing compensators.

For more information on designing compensators for plants with time delays, see Designing Compensators for Plants with Time Delays.

New Commands to More Efficiently Compute Multiple Linearizations

You can now more efficiently compute multiple linearizations for large models when only a few blocks or model references change per linearization. You linearize the fixed portion of the model once using `linlft` and linearize the varying portion multiple times. Then, you combine the results using `linlftfold` to obtain linearizations equal to those you would receive if you linearized the entire model multiple times.

For more information, see the `linlft` and `linlftfold` reference pages.

Ability to Set Default Plot Type for Linear Analysis Results from GUI

You can now set the default plot type for viewing linear analysis results computed in the Control and Estimation Tools Manager. This setting applies to all future Control and Estimation Tools Manager sessions.

To set this option in the Simulink Control Design Preferences, select **File > Preferences** in the Control and Estimation Tools Manager.

R2009a

Version: 2.5

New Features: Yes

Bug Fixes: Yes

Ability to Generate MATLAB Code from the GUI for Creating Operating Points and Linearizing Models

You can now generate MATLAB code reflecting the configuration in the GUI when creating operating points and linearizing models.

Ability to Tune Additional Blocks

You can now tune the following blocks:

- Blocks that you discretized using the Simulink Model Discretizer
- Blocks in the Simulink Extras library that specify initial states or outputs

For information about how to tune these blocks, see Selecting Blocks to Tune in the Simulink Control Design documentation.

New Option for Labeling Bus Signal I/O Names in Linearization Results

You can now compute linear models that show the bus signal names for linearization I/O points located on buses. You can select this option using the following:

- Linearization Options GUI
- linoptions command

R2008b

Version: 2.4

New Features: Yes

Bug Fixes: Yes

New Upsampling Option for Rate Conversion When Linearizing Simulink Models

Version 2.4 includes an upsampling rate conversion method for linearization. This method upsamples discrete-time LTI systems at any sampling rate that is an integer-value-times faster than the sampling rate of the original system. You can select the upsampling rate conversion method in the following ways:

- Using the linearization options GUI
- From the command line using `linoptions`

For more information on the upsampling rate conversion method, see the `linoptions` reference page.

Ability to Specify State Order of Linearized Models from the Command Line

You can now specify the order of the states in your linearized model directly from the command line using the `linearize` command. Previously, you could only specify state order using the GUI.

For more information on specifying state order from the command line, see the `linearize` reference page.

Ability to Filter the Linearization Inspector to Show Blocks in the Linearization Path

You can now filter the list of blocks in the Linearization Inspector to show only the blocks in the linearization path. This filtering makes it easier for you to find blocks in the linearization path that you want to inspect.

Ability to Disable the Calculation of Linearization Diagnostics and Inspector Data in the GUI

You can now disable the calculation of the linearization inspector and diagnostics information when you linearize using the GUI. This capability allow you to choose when you want to calculate diagnostic information.

R2008a

Version: 2.3

New Features: Yes

Bug Fixes: Yes

New Diagnostic Messages Help You Troubleshoot Linearization Results

You can now view diagnostic messages for your linearized model that help you diagnose and troubleshoot linearization results. These messages identify blocks in your model that encounter the following block issues during linearization:

- Blocks that have been marked as not supported for linearization
- Blocks with linearization configuration warning messages
- Blocks without pre-programmed exact Jacobian that linearize using numerical approximation

Ability to Find Operating Points for Simscape Models

You can now find operating points for models that include Simscape™ and SimHydraulics® blocks using the Simulink Control Design `findop` command.

For more information on finding operating point for Simscape Models, see the Simscape documentation.

Updated Error and Warning Message System

The Simulink Control Design error and warning IDs and messages have been updated. If you use error and warning IDs in your code, you must update your code to reflect the new IDs.

R2007b

Version: 2.2

New Features: Yes

Bug Fixes: Yes

Ability to Linearize Models with Model-Reference Blocks by Any Linearization Method

You can now perform any type of linearization for models containing Model blocks that reference other Simulink models.

Previously, you could only perform numerical perturbation linearization for models containing Model blocks. Now, you can also perform block-by-block linearization for such models when you set the simulation mode of the Model blocks to Normal.

Ability to Design Compensators for Models Containing Model-Reference Blocks

You can now design compensators for tunable blocks inside external models referenced by your model. Your model references such external models by using Model blocks. You can also update the tunable blocks in the external model with the new compensator designs.

To view and select tunable blocks for compensator design from the referenced model, set the simulation mode of the Model block to Normal.

For more information about the types of blocks that you can tune, see Selecting Blocks to Tune in the Simulink Control Design documentation.

Ability to Generate Linearized Models with Exact Time-Delay Representations

You can now use the Simulink Control Design software to compute linearized models with exact time-delay representations. Time delays in the original nonlinear model can result from any of the following blocks:

- Transport Delay
- Variable Time Delay
- Variable Transport Delay
- Unit Delay

- Integer Delay

Previously, you could only achieve approximate linearizations of models with continuous time delays using a Padé approximation.

Ability to Linearize Periodic Function-Call Subsystems

You can now linearize periodic function-call subsystems with a constant sample time as discrete subsystems. You must set the sample time of the function-call trigger block to equal the sample time of the function-call generator block.

For more information on function-call subsystems, see Function-Call Subsystems in the Simulink documentation.

R2007a

Version: 2.1

New Features: Yes

Bug Fixes: Yes

Ability to Linearize Using an Operating Point Specified Directly in a Model

You can use the Simulink Control Design software to linearize around any operating point that you specify directly in a Simulink model. This capability allows you to make changes to a model and then perform a linearization around the newly specified operating point with the click of a single button. Previously, this same task required two steps: creating a new operating point and then linearizing around this operating point.

There are two ways to linearize around an operating point specified directly in the model:

- From the Control and Estimation Tools Manager
- Using `linearize`

Ability to Capture Linearization Snapshots in GUI

You can use the Control and Estimation Tools Manager to linearize at snapshots of your model operating point at the following simulation points:

- Specified simulation times, such as when the simulation reaches a steady state solution
- Events during a specified simulation interval

As in prior releases, you can use `linearize` to perform linearization at snapshots in your model operating point, as described in `linearize` in the reference pages.

Ability to Perform Control Design at Snapshots in GUI

You can use the Control and Estimation Tools Manager to perform control design at snapshots of your model operating point at the following simulation points:

- Specified simulation times, such as when the simulation reaches a steady state solution

- Events during a specified simulation interval

Ability to Retrieve Stored Compensator Designs

When you design a compensator using the SISO design tool, you can store the current design and then continue making changes to this design. A new button called **Retrieve Design** lets you retrieve the stored design at any time by undoing the design changes you made since you last stored the design. For more information on retrieving stored compensator designs, see Storing and Retrieving Designs in the Simulink Control Design documentation.

R2006b

Version: 2.0.1

New Features: No

Bug Fixes: Yes

R2006a

Version: 2.0

New Features: Yes

Bug Fixes: Yes

Compensator Design in Simulink Is Now Supported

This release provides new tools to streamline the workflow for designing Single-Input Single-Output (SISO) control loops directly in Simulink. In previous releases, designing a compensator was a multistep process that involved several tools.

The new tools support any linearizable control architecture, such as single loops, multiple loops, and cascaded loops. With the new tools, you simply select the blocks you want to tune. Then, the Simulink Control Design software automatically analyzes your model to identify the relevant control loops and opens a preconfigured session of the SISO Design Tool (in the Control System Toolbox™ software). For more information, see "Designing Compensators" in the Simulink Control Design documentation.

Supported tunable SISO Simulink blocks include Gain, Transfer Function, Zero-Pole-Gain, State-Space, and PID blocks.

In the SISO Design Tool you can

- Graphically tune multiple SISO loops in a single GUI.

- Gain visual insight into loop interactions and coupling effects.

- Focus the analysis on a specific loop in a multiloop design by removing the effect of other feedback loops.

- Generate first-cut compensator designs using systematic design algorithms, such as Ziegler-Nichols PID tuning, IMC design, or LQG design.

- Optimize linear responses to meet time and frequency-based design constraints (requires the Simulink Response Optimization™ software).

- Directly tune Simulink block parameters, such as PID gains, zero-pole-gain representations, and masked blocks.

- Tune continuous or discrete control loops.

- Examine the closed-loop response of any portion of a model.

After a design is completed, you can write the tuned parameter values back to your model for verification with the full nonlinear system.

R14SP3

Version: 1.3

New Features: Yes

Bug Fixes: Yes

Control and Estimation Tools Manager Enhanced

You can copy and edit operating points within the Control and Estimation Tools Manager. .

You can initialize a model for simulation using operating points from within the Control and Estimation Tools Manager.

Support for Operating Point Search and Linearization of Models with Model Reference Blocks

You can linearize and compute operating points for models that reference other models using the Model block. Linearization of model reference models must use the numerical perturbation linearization algorithm. This algorithm accepts state and input perturbation values in the form of an operating point object.

R14SP2

Version: 1.2

New Features: Yes

Bug Fixes: Yes

Context-Sensitive Help Added

Access context-sensitive help for the Linearization, Operating Point Search, and Linearization State Ordering Options window of the Control and Estimation Tools Manager. To access help on a field within the options window, right-click the option's label and select **What's this?** from the context menu. Help for the option will appear in this window.

View Linearizations in the Control and Estimation Tools Manager

You can view state space, transfer function, and zero-pole gain representations of linearized models within the Control and Estimation Tools Manager without exporting to the workspace. These linearized models appear in the linearization summary pane of the Control and Estimation Tools Manager.

Discretization Methods Added

You can select from three different discretization methods for linearization of multirate and hybrid models.

List of Blocks with Preprogrammed Analytic Jacobians Added

You can view a complete list of blocks indicating which blocks have preprogrammed analytic Jacobians for use with the block-by-block analytic linearization algorithm.

Block Name Readability Improved

You can use either truncated block names or full block names in the state space matrices of a linearized model, and within the LTI Viewer, to improve readability.