

Control System Toolbox™

Release Notes

How to Contact MathWorks



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

Web
Newsgroup
Technical Support



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

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



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

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

Control System Toolbox™ Release Notes

© COPYRIGHT 2002–2012 by The MathWorks, Inc.

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

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

Trademarks

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

Patents

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

Summary by Version	1
Version 9.3 (R2012a) Control System Toolbox Software	5
Version 9.2 (R2011b) Control System Toolbox Software	9
Version 9.1 (R2011a) Control System Toolbox Software	11
Version 9.0 (R2010b) Control System Toolbox Software	16
Version 8.5 (R2010a) Control System Toolbox Software	20
Version 8.4 (R2009b) Control System Toolbox Software	22
Version 8.3 (R2009a) Control System Toolbox Software	23
Version 8.2 (R2008b) Control System Toolbox Software	25
Version 8.1 (R2008a) Control System Toolbox Software	28
Version 8.0.1 (R2007b) Control System Toolbox Software	29
Version 8.0 (R2007a) Control System Toolbox Software	30

Version 7.1 (R2006b) Control System Toolbox	
Software	32
Version 7.0 (R2006a) Control System Toolbox	
Software	33
Version 6.2.1 (R14SP3) Control System Toolbox	
Software	37
Version 6.2 (R14SP2) Control System Toolbox	
Software	38
Version 6.1 (R14SP1) Control System Toolbox	
Software	40
Version 6.0 (R14) Control System Toolbox Software ...	41
Version 5.2.1 (R13SP2) Control System Toolbox	
Software	44
Version 5.2 (R13) Control System Toolbox Software ...	45
Compatibility Summary for Control System Toolbox	
Software	49

Summary by Version

This table provides quick access to what's new in each version. For clarification, see “Using Release Notes” on page 2.

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Latest Version V9.3 (R2012a)	Yes Details	Yes Summary	Bug Reports
V9.2 (R2011b)	Yes Details	Yes Summary	Bug Reports
V9.1 (R2011a)	Yes Details	Yes Summary	Bug Reports
V9.0 (R2010b)	Yes Details	Yes Summary	Bug Reports
V8.5 (R2010a)	Yes Details	Yes Summary	Bug Reports
V8.4 (R2009b)	Yes Details	No	Bug Reports
V8.3 (R2009a)	Yes Details	Yes Summary	Bug Reports
V8.2 (R2008b)	Yes Details	Yes Summary	Bug Reports
V8.1 (R2008a)	Yes Details	No	Bug Reports
V8.0.1 (R2007b)	Yes Details	No	Bug Reports
V8.0 (R2007a)	Yes Details	No	Bug Reports
V7.1 (R2006b)	Yes Details	No	Bug Reports

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
V7.0 (R2006a)	Yes Details	No	Bug Reports
V6.2.1 (R14SP3)	No	No	Bug Reports
V6.2 (R14SP2)	Yes Details	No	Fixed Bugs
V6.1 (R14SP1)	No	No	Fixed Bugs
V6.0 (R14)	Yes Details	Yes Summary	Fixed bugs
V5.2.1 (R13SP2)	No	No	No bug fixes
V5.2 (R13)	Yes Details	No	Fixed bugs

Using Release Notes

Use release notes when upgrading to a newer version to learn about:

- New features
- Changes
- Potential impact on your existing files and practices

Review the release notes for other MathWorks® products required for this product (for example, MATLAB® or Simulink®). Determine if enhancements, bugs, or compatibility considerations in other products impact you.

If you are upgrading from a software version other than the most recent one, review the current release notes and all interim versions. For example, when you upgrade from V1.0 to V1.2, review the release notes for V1.1 and V1.2.

What Is in the Release Notes

New Features and Changes

- New functionality
- Changes to existing functionality

Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility between versions, the **Compatibility Considerations** subsection explains the impact.

Compatibility issues reported after the product release appear under Bug Reports at the MathWorks Web site. Bug fixes can sometimes result in incompatibilities, so review the fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

MathWorks offers a user-searchable Bug Reports database so you can view Bug Reports. The development team updates this database at release time and as more information becomes available. Bug Reports include provisions for any known workarounds or file replacements. Information is available for bugs existing in or fixed in Release 14SP2 or later. Information is not available for all bugs in earlier releases.

Access Bug Reports using your MathWorks Account.

Documentation on the MathWorks Web Site

Related documentation is available on mathworks.com for the latest release and for previous releases:

- Latest product documentation

- Archived documentation

Version 9.3 (R2012a) Control System Toolbox Software

This table summarizes what's new in Version 9.3 (R2012a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports

New features introduced in this version:

- “Frequency Analysis Commands for Calculating Peak Gain and Finding Gain-Crossover Frequencies” on page 5
- “Specify Target Crossover Frequency as Input to pidtune” on page 6
- “Rescaled Impulse Response and Impulse-Invariant Time Domain Conversion” on page 6
- “First-Order Hold Method for d2c” on page 7
- “tzero Computes Invariant Zeros and Transmission Zeros” on page 7
- “Models Created With System Identification Toolbox Can Be Used Directly With Control System Toolbox Functions” on page 7
- “Functionality Being Removed or Changed” on page 8

Frequency Analysis Commands for Calculating Peak Gain and Finding Gain-Crossover Frequencies

Control System Toolbox™ software includes two new frequency analysis commands:

- `getPeakGain` — Peak gain of frequency response of a dynamic system model
- `getGainCrossover` — Frequencies at which system gain crosses a specified gain level

For more information, see the `getPeakGain` and `getGainCrossover` reference pages.

These functions use the SLICOT library of numerical algorithms. For more information about the SLICOT library, see <http://slicot.org>.

Specify Target Crossover Frequency as Input to `pidtune`

A new syntax for `pidtune` lets you specify a target crossover frequency directly as an input argument. For example, the following command designs a PI controller, `C`, for a plant model `sys`. The command also specifies a target value `wc` for the 0 dB gain crossover frequency of the open-loop response $L = \text{sys} * C$.

```
C = pidtune(sys, 'pi', wc);
```

Previously, you had to use `pidtuneOptions` to specify a target crossover frequency.

For more information, see the `pidtune` reference page.

Rescaled Impulse Response and Impulse-Invariant Time Domain Conversion

For discrete-time dynamic system models, the input signal applied by `impulse` is now a unit area pulse of length T_s and height $1/T_s$. T_s is the sampling time of the discrete-time system. Previously, `impulse` applied a pulse of length T_s and unit height.

Compatibility Considerations

Results of this change include:

- The amplitude of the impulse response calculated by `impulse` and `impzplot` is scaled by $1/T_s$ relative to previous versions.
- Discretization using the impulse-invariant ('`impzvar`') method of `c2d` returns a model that is scaled by T_s compared to previous releases. This scaling ensures a close match between the frequency responses of the continuous-time model and the impulse-invariant discretization as T_s

approaches zero (for strictly proper models). In previous releases, the frequency responses differed by a factor of T_s .

First-Order Hold Method for d2c

The `d2c` command now supports the first-order hold (FOH) method for converting a discrete-time dynamic system model to continuous time. The FOH method converts by performing linear interpolation of the inputs, assuming the control inputs are piecewise linear over the sampling period. For more information about using this method, see the `d2c` reference page and “Continuous-Discrete Conversion Methods”.

tzero Computes Invariant Zeros and Transmission Zeros

The `tzero` command computes the invariant zeros of SISO and MIMO dynamic system models. For minimal realizations, `tzero` computes transmission zeros. `tzero` also returns the normal rank of the transfer function of the system. For more information, see the `tzero` reference page.

Models Created With System Identification Toolbox Can Be Used Directly With Control System Toolbox Functions

Identified linear models that you create using System Identification Toolbox™ software can now be used directly with Control System Toolbox analysis and compensator design commands. In prior releases, doing so required conversion to Control System Toolbox LTI model types.

Identified linear models include `idfrd`, `idss`, `idproc`, `idtf`, `idgrey` and `idpoly` models.

Identified linear models can be used directly with:

- Any Control System Toolbox or Robust Control Toolbox™ functions that operate on dynamic systems, including:
 - Response plots — `nichols`, `margin`, and `rlocus`
 - Model simplification — `pade`, `balred` and `minreal`

- System interconnections — series, parallel, feedback and connect

For a complete list of these functions, enter:

```
methods('DynamicSystem')
```

- Analysis and design tools such as `ltiview`, `sisotool` and `pidtool`.
- The LTI System block in Simulink models.

Functionality Being Removed or Changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
<code>impulse(sys)</code> and <code>impzplot(sys)</code> , for discrete-time <code>sys</code>	Still works.	N/A	Amplitude of response is scaled by $1/T_s$ compared to previous versions. T_s is sampling time of <code>sys</code> .
<code>c2d(sys,Ts,'impz')</code>	Still works.	N/A	Resulting discretized model is scaled by T_s compared to previous releases.
<pre>[y,t] = impz(sys,Tfinal) [y,t] = step(sys,Tfinal) [y,t,x] = initial(sys,Tfinal)</pre>	For discrete-time <code>sys</code> with undefined sample time ($T_s=-1$), <code>Tfinal</code> is interpreted as the number of sampling periods to simulate.	N/A	Expect the number of simulation data points to be <code>Tfinal + 1</code> instead of <code>Tfinal</code> .

Version 9.2 (R2011b) Control System Toolbox Software

This table summarizes what’s new in Version 9.2 (R2011b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes Details below	Bug Reports

New features introduced in this version:

- “Formula-Based Specification of Summing Junctions and Vector Signal Naming for sumblk and connect” on page 9
- “Commands for Interacting with Control Design Blocks in Generalized LTI Models” on page 9
- “Functionality Being Removed or Changed” on page 10

Formula-Based Specification of Summing Junctions and Vector Signal Naming for sumblk and connect

You can now use formula strings to specify the behavior of summing junctions with sumblk. For example, to create a summing junction, S, that takes the difference between signals r and y to produce signal e, enter the following command:

```
S = sumblk('e = r-y');
```

Additionally, both sumblk and connect now support vector-based signal naming for interconnecting multi-input, multi-output (MIMO) models. For more information, see the sumblk and connect reference pages.

Commands for Interacting with Control Design Blocks in Generalized LTI Models

The following new commands allow you to examine and set the values of Control Design Blocks in Generalized LTI Models:

- `getValue` — Get nominal value of Generalized Model (replaces `getNominal`)

- `setValue` — Modify value of Control Design Block
- `getBlockValue` — Get nominal value of Control Design Block in Generalized Model
- `setBlockValue` — Set value of Control Design Block in Generalized Model
- `showBlockValue` — Display nominal values of Control Design Blocks in Generalized Model

For more information about these commands, see the reference pages for each command.

Functionality Being Removed or Changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
<code>delay2z</code>	Errors	<code>absorbDelay</code>	Replace <code>delay2z</code> with <code>absorbDelay</code> .
<code>getNominal</code>	Errors	<code>getValue</code>	Replace <code>getNominal</code> with <code>getValue</code> .
Scale and Info properties of <code>realp</code> parameter	Errors	None	None
<code>sumblk('a','b','c','+ -')</code>	Still works	<code>sumblk('a=b-c')</code>	Use new formula-based syntax for <code>sumblk</code> .

Version 9.1 (R2011a) Control System Toolbox Software

This table summarizes what's new in Version 9.1 (R2011a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports

New features introduced in this version:

- “New Model Objects for Representing Tunable Parameters and Systems with Tunable Components” on page 11
- “New Time and Frequency Units for Models and Response Plots” on page 12
- “Discrete-Time PID Controller Objects Have Stable Derivative Filter Pole” on page 13
- “New Variable q^{-1} for Expressing Discrete-Time Transfer Functions” on page 14

New Model Objects for Representing Tunable Parameters and Systems with Tunable Components

Control System Toolbox includes new model objects that you can use to represent systems with tunable components. You can use these models for parameter studies or controller synthesis using `hinfstruct` (requires Robust Control Toolbox). The new model types include:

- “Control Design Blocks”—Parametric components that are the building blocks for constructing tunable models of control systems. Control Design Blocks include:
 - `realp`—Tunable real parameter
 - `1tiblock.gain`—Tunable static gain block
 - `1tiblock.tf`—Fixed-order SISO transfer function with tunable coefficients

- `ltiblock.ss`—Fixed-order state-space model with tunable coefficients
- `ltiblock.pid`—One-degree-of-freedom PID controller with tunable coefficients
- “Generalized Matrices”—Matrices that include parametric (tunable) values. Generalized matrices are `genmat` models.
- “Generalized and Uncertain LTI Models”—Models representing systems that have both fixed and tunable coefficients. Generalized LTI models include:
 - `genss`—Generalized state-space model
 - `genfrd`—Generalized frequency response data model

These models arise from interconnections between numeric LTI models (such as `tf`, `ss`, or `frd`) and Control Design Blocks. You can also create `genss` models by using the `tf` or `ss` commands with one or more `realp` or `genmat` inputs.

This release also adds new functions for working with generalized models:

- `getNominal`—Nominal value of generalized model
- `replaceBlock`—Replace Control Design Blocks in generalized model
- `nblocks`—Number of blocks in generalized model
- `isParametric` — Determine if model has tunable blocks
- `getLFTModel`—Decompose generalized model

For more information about the new model types and about modeling systems that contain tunable coefficients, see the following in the *Control System Toolbox User's Guide*:

- “Types of Model Objects”
- “Models with Tunable Coefficients”

New Time and Frequency Units for Models and Response Plots

All linear model objects now have a `TimeUnit` property for specifying unit of the time variable, time delays in continuous-time models, and sampling

time in discrete-time models. The default time units is seconds. You can specify the time units, for example, as hours. See “Specify Model Time Units” for examples.

Frequency-response data (`frd` and `genfrd`) models also have a new `FrequencyUnit` property for specifying units of the frequency vector. The default frequency units is `rad/TimeUnit`, where `TimeUnit` is the system time units. You can specify the units, for example as KHz, independently of the system time units. See “Specify Frequency Units of Frequency-Response Data Model” for examples. If your code uses the `Units` property of frequency-response data models, it continues to work as before.

See the model reference pages for available time and frequency units options.

Changing the `TimeUnit` and `FrequencyUnit` properties changes the overall system behavior. If you want to simply change the time and frequency units without modifying system behavior, use `chgTimeUnit` and `chgFreqUnit`, respectively.

The time and frequency units of the model appear on the response plots by default. For multiple systems, the units of the first system are used. You can change the units of the time and frequency axes:

- Graphically, using the following editors:
 - Toolbox Preferences Editor
 - “LTI Viewer Preferences Editor”
 - “Graphical Tuning Window Preferences Editor”
 - Property Editor of individual plots
- Programmatically, by setting the following properties of plots:
 - `TimeUnits` for time-domain plots using `timeoptions`
 - `FreqUnits` for frequency-domain plots using, for example, `bodeoptions`

Discrete-Time PID Controller Objects Have Stable Derivative Filter Pole

New requirements for creating `pid` and `pidstd` controller objects ensure that the derivative filter pole is always stable.

- For a discrete-time pid controller with a derivative filter ($T_f \neq 0$) and Dformula set to 'ForwardEuler', the sampling time T_s must be less than $2 \cdot T_f$.
- For a discrete-time pidstd controller with a derivative filter ($N \neq \text{Inf}$) and Dformula set to 'ForwardEuler', the sampling time T_s must be less than $2 \cdot T_d / N$.
- The Trapezoidal value for DFormula is not available for a discrete-time pid or pidstd controller with no derivative filter ($T_f = 0$ or $N = \text{Inf}$).

Compatibility Considerations

On loading pid or pidstd controllers saved under previous versions, the software changes certain properties of controllers that do not have stable derivative filter poles.

- For a discrete-time pid controller with a derivative filter ($T_f \neq 0$), Dformula set to 'ForwardEuler', and sampling time $T_s \geq 2 \cdot T_f$, the derivative filter time is reset to $T_f = T_s$.
- For a discrete-time pidstd controller with a derivative filter ($N \neq \text{Inf}$), Dformula set to 'ForwardEuler', the sampling time $T_s \geq 2 \cdot T_d / N$, the derivative filter constant is reset to $N = T_d / T_s$.
- For a discrete-time pid or pidstd controller with no derivative filter and DFormula = 'Trapezoidal', the derivative filter integrator formula is reset to DFormula = 'ForwardEuler'.

The software issues a warning when it changes any of these values. If you receive such a warning, validate your controller to ensure that the new values achieve the desired performance.

New Variable q^{-1} for Expressing Discrete-Time Transfer Functions

You can now express discrete-time tf and zpk models in terms of the inverse shift operator q^{-1} . The variable q^{-1} is equivalent to z^{-1} .

Note This new definition is consistent with the System Identification Toolbox definition of q^{-1} .

Use the new variable by setting the `Variable` property of a `tf` or `zpk` model to q^{-1} . For example, entering:

```
H = tf([1 2 3],[5 6 7],0.1,'Variable','q^-1')
```

creates the following discrete-time transfer function:

Transfer function:

$$\frac{1 + 2 q^{-1} + 3 q^{-2}}{5 + 6 q^{-1} + 7 q^{-2}}$$

Sampling time (seconds): 0.1

When you set `Variable` to q^{-1} , `tf` interprets the numerator and denominator vectors as ascending powers of q^{-1} .

For more information, see the `tf` and `zpk` reference pages.

Version 9.0 (R2010b) Control System Toolbox Software

This table summarizes what's new in Version 9.0 (R2010b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports

New features introduced in this version:

- “New Commands and GUI for Modeling and Tuning PID Controllers” on page 16
- “Improved PID Tuning Options in SISO Design Tool” on page 18
- “Ability to Analyze a Controller Design for Multiple Models Simultaneously in SISO Design Tool” on page 18
- “Change in Output of repsys Command” on page 19

New Commands and GUI for Modeling and Tuning PID Controllers

This release introduces specialized tools for modeling and designing PID controllers.

PID Controller Design with the New PID Tuner GUI

The new PID Tuner GUI lets you interactively tune a PID controller for your required response characteristics. Using the GUI, you can adjust and analyze your controller's performance with response plots, such as reference tracking, load disturbance rejection, and controller effort, in both time and frequency domains.

The PID Tuner supports all types of SISO plant models, including:

- Continuous- or discrete-time plant models

- Stable, unstable, or integrating plant models
- Plant models that include I/O time delays or internal time delay

For more information about using PID Tuner, see:

- “Designing PID Controllers” in the *Control System Toolbox Getting Started Guide*
- The new demo Designing PID for Disturbance Rejection with PID Tuner

PID Controller Design with the New `pidtune` Command

The new `pidtune` command lets you tune PID controller gains at the command line.

`pidtune` automatically tunes the PID gains to balance performance (response time) and robustness (stability margins). You can specify your own response time and phase margin targets using the new `pidtuneOptions` command.

`pidtune` supports all types of SISO plant models, including:

- Continuous- or discrete-time plant models.
- Stable, unstable, or integrating plant models.
- Plant models that include I/O time delays or internal time delays.
- Arrays of plant models. If `sys` is an array, `pidtune` designs a separate controller for each plant in the array.

For additional information, see:

- The `pidtune` and `pidtuneOptions` reference pages
- The new Control System Toolbox demo Designing Cascade Control System with PI Controllers

Modeling PID Controllers in Parallel Form or Standard Form

The new LTI model objects `pid` and `pidstd` are specialized for modeling PID controllers.

With `pid` and `pidstd` you can model a PID controller directly with the PID parameters, expressed in parallel (`pid`) or standard (`pidstd`) form. The `pid` and `pidstd` commands can also convert to PID form any type of LTI object that represents a PID controller.

Previously, to model a PID controller, you had to derive the controller's equivalent transfer function (or other model), and could not directly store the PID parameters.

For additional information, see the `pid` and `pidstd` reference pages

Improved PID Tuning Options in SISO Design Tool

This release includes improvements to the PID Tuning options in the Automated Tuning pane of SISO Design Tool.

In addition to the Robust Response Time tuning algorithm, SISO Design Tool offers a collection of classical design formulas, including the following:

- Approximate M -Constrained Integral Gain Optimization (MIGO) Frequency Response
- Approximate MIGO Step Response
- Chien-Hrones-Reswick
- Skogestad Internal Model Control (IMC)
- Ziegler-Nichols Frequency Response
- Ziegler-Nichols Step Response

For information about using SISO Design Tool, see “SISO Design Tool” in the *Control System Toolbox User's Guide*. For specific information about the automatic PID Tuning options in SISO Design Tool, see “PID Tuning” in the *Control System Toolbox User's Guide*.

Ability to Analyze a Controller Design for Multiple Models Simultaneously in SISO Design Tool

You can now analyze a controller design for multiple models simultaneously using the SISO Design Tool. This feature helps you analyze whether the

controller satisfies design requirements on a system whose exact dynamics are not known and may vary.

System dynamics can vary because of parameter variations or different operating conditions. You represent variations in system dynamics of the plant (G), sensor (H), or both in a feedback structure using arrays of LTI models. Then, design a controller for a nominal model in the array and analyze that the controller satisfies the design requirements on the remaining models using the design and analysis plots. For more information, see:

- “Control Design Analysis of Multiple Models” in the Control System Toolbox documentation.
- Compensator Design for a Set of Plant Models demo.
- Reference Tracking of a DC Motor with Parameter Variations demo in Simulink Control Design™ software.

Change in Output of `repsys` Command

The output of the `repsys` command when called with a single dimension argument has changed.

In prior versions, the output of `repsys(sys,N)` was the same as that of `append(sys,...,sys)`.

Now, `repsys(sys,N)` returns the same result as `repsys(sys,[N N])`.

The results of other syntaxes for `repsys` have not changed.

See the `repsys` and `append` reference pages for more information.

Compatibility Consideration

Code that depends upon the previous result of `repsys(sys,N)` no longer returns that result. To obtain the previous result, replace `repsys(sys,N)` with `sys*eye(N)`.

Version 8.5 (R2010a) Control System Toolbox Software

This table summarizes what’s new in Version 8.5 (R2010a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports

New features introduced in this version:

- “Enhanced c2d Command to Approximate Fractional Time Delays in Tustin and Matched Discretization Methods” on page 20
- “New Commands for Specifying Options for Continuous-Discrete Conversions” on page 21
- “New FDEL Command to Remove Specified Data from Frequency Response Data (FRD) Models” on page 21

Enhanced c2d Command to Approximate Fractional Time Delays in Tustin and Matched Discretization Methods

The c2d command can now approximate fractional time delays when discretizing linear models with the tustin or matched methods. The new c2dOptions command lets you specify an optional Thiran all-pass filter. The Thiran filter approximates fractional delays for improved phase matching between continuous and discretized models. Previously, c2d rounded fractional time delays to the nearest multiple of the sampling time when using the tustin or matched methods. For more information, see the c2d and c2dOptions reference pages and in the *Control System Toolbox User Guide*.

New Commands for Specifying Options for Continuous-Discrete Conversions

New commands `c2dOptions`, `d2dOptions`, and `d2cOptions` make it easier to specify options for

- Discretization using `c2d`
- Resampling using `d2d`.
- Conversion from discrete to continuous time using `d2c`.

Compatibility Consideration

This release deprecates the `prewarp` method for `c2d`, `d2d`, and `d2c`. Instead, use `c2dOptions`, `d2dOptions`, or `d2cOptions` to specify the `tustin` method and a prewarp frequency. For more information, see in the *Control System Toolbox User Guide* and the `c2d`, `d2d`, and `d2c` reference pages.

New FDEL Command to Remove Specified Data from Frequency Response Data (FRD) Models

You can now remove selected data from `frd` models using the new `fdel` command. For example, use `fdel` to:

- Remove spurious or unneeded data from `frd` models you create from measured frequency response data.
- Remove data at intersecting frequencies from `frd` models before merging them into a single `frd` model with `fcat`, which can only merge `frd` models containing no common frequencies.

For more information, see `fdel` reference page.

Version 8.4 (R2009b) Control System Toolbox Software

This table summarizes what’s new in Version 8.4 (R2009b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports

New features introduced in this version:

- “Ability to Design Compensators for New Types of Plants” on page 22
- “New Automated PID Tuning Method” on page 22

Ability to Design Compensators for New Types of Plants

In the SISO Design Tool, you can now design compensators for plants models that:

- Contain time delays
Previously, you had to approximate delays before designing compensators.
- You specify as frequency-response data (FRD)

For more information on designing compensators using the SISO Design Tool, see “SISO Design Tool”.

New Automated PID Tuning Method

You can now tune compensators using a new automated PID tuning algorithm called Robust Response Time, which is available in the SISO Design Tool. 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 “Automated Tuning”.

Version 8.3 (R2009a) Control System Toolbox Software

This table summarizes what's new in Version 8.3 (R2009a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports

New change introduced in this version:

Variable q Now Defined as the Forward Shift Operator z

The variable q is now defined in the standard way as the forward shift operator z . Previously, q was defined as z^{-1} .

Note This new definition is consistent with the System Identification Toolbox definition of q .

Compatibility Consideration

If you use the q variable, you may receive different results than in previous releases when you:

- Create a transfer function
- Modify the num or den properties of an existing transfer function

The resulting transfer function differs from previous releases when both the

- Variable property is set to q
- num and den properties have different lengths

For example, the following code:

```
H = tf([1,2],[1 3 8],0.1,'Variable','q')
```

now returns the transfer function

$$\frac{q+2}{q^2+3q+8} \equiv \frac{z+2}{z^2+3z+8}$$

Previously, the code returned the transfer function

$$\frac{1+2q}{1+3q+8q^2} \equiv \frac{1+2z^{-1}}{1+3z^{-1}+8z^{-2}} \equiv \frac{z^2+2z}{z^2+3z+8}$$

The two transfer functions have different numerators.

Version 8.2 (R2008b) Control System Toolbox Software

This table summarizes what's new in Version 8.2 (R2008b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports

New features and changes introduced in this version are

- “New Design Tools for Linear-Quadratic-Gaussian (LQG) Servo Controllers with Integral Action” on page 25
- “New Upsampling Method for Rate Conversion in Discrete-Time Models” on page 26
- “New Scaling Tools to Enhance the Accuracy of Computations with State-Space Models” on page 26
- “New Command to Reorder the States of State-Space Models” on page 27
- “Enhanced Support for Customizing Response Plots” on page 27

New Design Tools for Linear-Quadratic-Gaussian (LQG) Servo Controllers with Integral Action

You can now design a Linear-Quadratic-Gaussian (LQG) servo controller for set-point tracking using the new `lqi` and `lqgtrack` commands. This compensator ensures that the system output tracks the reference command and rejects process disturbances and measurement noise.

For more information on forming LQG servo controllers, see “Linear-Quadratic-Gaussian (LQG) Design”, the `lqi` reference page, and the `lqgtrack` reference page.

Current Flag Moved from lqgreg to kalman

The 'current' flag was moved from the lqgreg function to the kalman function.

Compatibility Consideration. The following code:

```
kest = kalman(sys,Qn,Rn)
c = lqgreg(kest,k)
```

now returns the current regulator $u[n] = -K\hat{x}[n|n]$ instead of the delayed regulator $u[n] = -K\hat{x}[n|n-1]$.

To update your code to return the same results as in previous releases, use the following code with the added string 'delayed' in the kalman command:

```
kest = kalman(sys,Qn,Rn,'delayed')
c = lqgreg(kest,k)
```

For information on using these functions with the current flag in the kalman function, see the kalman and lqgreg reference pages.

New Upsampling Method for Rate Conversion in Discrete-Time Models

You can now upsample a discrete-time system to an integer multiple of the original sampling rate without any distortion in the time or frequency domain using the upsample command.

For more information on upsampling, see the upsample reference page and in the *Control System Toolbox User's Guide*.

New Scaling Tools to Enhance the Accuracy of Computations with State-Space Models

You can now scale state-space models to maximize accuracy over the frequency band of interest using the prescale command and associated GUI. Use this functionality when you cannot achieve good accuracy at all frequencies and some tradeoff is necessary. A warning alerts you when accuracy may be poor and using prescaling is recommended.

For more information on setting the frequency band for scaling state-space realizations, see “Scaling State-Space Models” and the `prescale` reference page.

New Command to Reorder the States of State-Space Models

You can now reorder the states of state-space models according to a specified permutation using the `xperm` command.

For more information on reordering states, see the `xperm` reference page.

Enhanced Support for Customizing Response Plots

You can now make the following changes to your Control System Toolbox response plots using the figure plotting tools:

- System name
- Line color
- Line style
- Line width
- Marker type

For more information on customizing the appearance of response plots using plot tools, see “Customizing Response Plots Using Plot Tools” in the *Control System Toolbox User's Guide*.

Version 8.1 (R2008a) Control System Toolbox Software

This table summarizes what's new in Version 8.1 (R2008a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports

New change introduced in this version:

Updated Error and Warning Message System

The Control System Toolbox 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.

Version 8.0.1 (R2007b) Control System Toolbox Software

This table summarizes what's new in Version 8.0.1 (R2007b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports

New change introduced in this version:

Updated and Expanded Demos

The Control System Toolbox demos have been reformatted and expanded to include more examples and content. Demos in the following categories now have new and improved content:

- Getting Started with LTI Models
- Discretization and Sampling Rate Conversions
- How to Get Accurate Results

To open the Control System Toolbox demos, type

```
demo toolbox control
```

at the MATLAB prompt.

Version 8.0 (R2007a) Control System Toolbox Software

This table summarizes what's new in Version 8.0 (R2007a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Bug Reports

New features and changes introduced in this version are

- “Analysis of Time Delay Systems Now Fully Supported” on page 30
- “New and Updated Automated Tuning Methods” on page 31
- “New Tustin and Prewarp Options for d2d Function” on page 31

Analysis of Time Delay Systems Now Fully Supported

Control System Toolbox software now lets you:

- Model, simulate, and analyze any interconnection of linear systems with delays, such as systems containing feedback loops with delays.
- Exactly analyze and simulate control systems with long delays. You can evaluate control strategies, such as Smith Predictor and PID control for first-order-plus-dead-time plants.
- Use new commands for modeling state-space models with delays including: `delayss`, `getDelayModel`, and `setDelayModel`.

For more information, see the section on “Models with Time Delays” in the Control System Toolbox documentation.

New and Updated Automated Tuning Methods

Control System Toolbox software now provides the following new and updated automated tuning methods:

- New Singular Frequency Based Tuning lets you design PID compensators for both stable and unstable plants.
- New H-infinity Loop Shaping lets you find compensators based on a desired open-loop bandwidth or loop shape. This feature requires Robust Control Toolbox software.
- Updated Internal Model Control (IMC) Tuning now supports unstable plants.

For more information, see the section on automated tuning in the Control System Toolbox documentation.

New Tustin and Prewarp Options for d2d Function

The `d2d` function now includes the following new options for the resampling method:

- `'tustin'`—Performs Bilinear (Tustin) approximation
- `'prewarp'`—Performs Tustin approximation with frequency prewarping

For more information, see the `d2d` reference pages.

Version 7.1 (R2006b) Control System Toolbox Software

This table summarizes what’s new in Version 7.1 (R2006b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	No bug fixes

New features and changes introduced in this version are

- “New Loop Configurations in the SISO Design Tool” on page 32
- “New Design Requirements” on page 32

New Loop Configurations in the SISO Design Tool

Two new loop configurations are available from the SISO Design Tool. See Modifying Block Diagram Structure for more information.

New Design Requirements

The LTI Viewer now supports step response and upper/lower time bound design requirements. See Adding Design Requirements to the LTI Viewer for more information.

Version 7.0 (R2006a) Control System Toolbox Software

This table summarizes what's new in Version 7.0 (R2006a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	No bug fixes

New features and changes introduced in this version are

- “SISO Design Tool” on page 33
- “LTI Viewer Enhancements” on page 34
- “LTI Objects” on page 34
- “Numerical Algorithms” on page 36

SISO Design Tool

The SISO Design Tool now provides one-click automated tuning using systematic algorithms such as Ziegler-Nichols PID tuning, IMC design, and LQG design. In addition, you can calculate low-order approximations of the IMC/LQG compensators to keep the control system complexity low.

Compensator Optimization Is Now Supported

If you have installed Simulink Response Optimization™ software, you can now optimize the compensator parameters inside the SISO Design Tool GUI. You can specify time- and frequency-domain requirements on SISO Design Tool plots such as `bode` and `step`, and use numerical optimization algorithms to automatically tune your compensator to meet your requirements. See the Simulink Response Optimization documentation for more details.

Improved Compensator Editor

The Compensator Editor used to edit the numerical values of poles and zeros has been upgraded to better handle common control components such as lead/lag and notch filters.

Multi-Loop Compensator Design Support

Many control systems involve multiple feedback loops, some of which are coupled and need joint tuning. The SISO Design Tool now lets you analyze and tune multi-loop configurations. You can focus on a specific loop by opening signals to remove the effects of other loops, gain insight into loop interactions, and jointly tune several SISO loops.

SISO Design Tool Fully Integrated with the Controls & Estimation Tools Manager

To improve workflow and better leverage other tools, such as Simulink Control Design software and Simulink Response Optimization software, the SISO Design Tool is now fully integrated with the Controls & Estimation Tools Manager (CETM). This provides a signal environment for the design and tuning of compensators.

When you open the SISO Design Tool, the CETM also opens with a SISO Design Task. Many SISO Design Tool features, such as importing models, changing loop configurations, etc., have been moved to the SISO Design Task in CETM. In addition, related tasks such as Simulink based Tuning and Compensator Optimization are seamlessly integrated with the SISO Design Task. See the *Control System Toolbox Getting Started Guide* for details on the new work flow.

LTI Viewer Enhancements

The LTI Viewer now lets you plot the response of a system to user-defined input signals (`lsim`) and initial conditions (`initial`). A new GUI lets you select input signals from a signal generator library, or import signal data from a variety of file formats.

LTI Objects

Descriptor and Improper State-Space Models Fully Supported

There is now full support for descriptor state-space models with a singular E matrix. This now lets you build state-space representations, such as PID, and manipulate improper models with the superior accuracy of state-space

computations. In previous versions, only descriptor models with a nonsingular E matrix were supported.

New Commands to Calculate Time Response Metrics

The new `stepinfo` and `lsiminfo` commands compute time-domain performance metrics, such as rise time, settling time, and overshoot. You can use these commands to write scripts that automatically verify or optimize such performance requirements. Previously, these metrics were available only from response plots.

Simplified System Interconnections Using I/O Channel Names

The commands `connect`, `feedback`, `series`, `parallel`, and `lft` now let you connect systems by matching names of I/O channels. A helper function, `sumblk`, has also been added to simplify the specification of summing junctions. Altogether this considerably simplifies the task of deriving models for complicated block diagrams. In previous releases, only index-based system connection was supported.

Changes in the Representation of I/O Delays in State-Space Models

The `ioDelay` property is deprecated from state-space models. Instead, these models have a new property called `InternalDelay` for logging all delays that cannot be pushed to the inputs or outputs. Driving this change is the switch to a representation of delays in terms of delayed differential equations rather than frequency response. See “Models with Time Delays” in the Control System Toolbox documentation for more details on internal delays, and `ss/getdelaymodel` for details on the new internal representation of state-space models with delays.

New Name Property for LTI Objects

This new property lets you attach a name (string) to a given LTI model. The specified name is reflected in response plots.

New Commands and Operations for LTI Objects

The new `exp` command simplifies the creations of continuous-time transfer functions with delays. For more information, type `help lti/exp` at the MATLAB prompt.

The `frd` object has the following new methods:

- `fcats` — Concatenates one or more FRD models along the frequency dimension (data merge).
- `fselect` — Selects frequency points or range in `frd` model.
- `fnorm` — Calculates pointwise peak gain of `frd` model.

The `.*` operation is supported for transfer functions and zero-pole-gain objects. This allows you to perform element-by-element multiplication of MIMO models.

Numerical Algorithms

There have been several major improvements in the Control System Toolbox numerical algorithms, many of which benefit the upgraded SISO Design Tool:

- New scaling algorithm that maximizes accuracy for badly scaled state-space models
- Performance improvement in time and frequency response computations through MEX-files
- More accurate computations of the zero-pole-gain and transfer function representations of a state-space model
- More accurate state-space representations of zero-pole-gain models
- Better handling of nonminimal modes in model reduction commands (`balred`, `balreal`)
- `canon` now computes a block modal form for A matrices that are not diagonalizable or are nearly defective
- Exact phase computation for zero-pole-gain models in `bode` and `nichols`
- Accurate handling of improper models using the descriptor state-space representation

Version 6.2.1 (R14SP3) Control System Toolbox Software

This table summarizes what's new in Version 6.2.1 (R14SP3):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
No	No	Bug Reports

Version 6.2 (R14SP2) Control System Toolbox Software

This table summarizes what's new in Version 6.2 (R14SP2):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Fixed Bugs

New features and changes introduced in this version are

- “Command-Line API for Customizing Plots” on page 38
- “Constraint Types for SISO Design” on page 38
- “Bode and Nichols Plots Have Additional Options” on page 39
- “Model-Approximation and Order-Reduction Commands” on page 39

Command-Line API for Customizing Plots

The Control System Toolbox software now provides a command-line API for customizing units, labels, limits, and other plot options. You can now change default plot options before generating a plot, or modify plot properties after creation.

For a detailed description of the commands, see the Control System Toolbox documentation.

Constraint Types for SISO Design

You can now create

- Single piecewise linear constraints for root-locus and Bode plots
- Gain/phase exclusion regions for Nichols plots

Design constraints are displayed as shaded regions.

Bode and Nichols Plots Have Additional Options

When editing Bode and Nichols plots, you can now

- Set the lower limit of the magnitude manually.
- Adjust the phase offsets by multiples of 360 degrees to facilitate comparing multiple responses.

Model-Approximation and Order-Reduction Commands

New commands have been added for model approximation and order reduction:

- `hsvd` computes and plots the Hankel singular values.
- `balred` computes low-order approximations using a numerically stable, balancing-free algorithm. You can perform multiple order reductions with a single command.

Version 6.1 (R14SP1) Control System Toolbox Software

This table summarizes what's new in Version 6.1 (R14SP1):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
No	No	Fixed Bugs

Version 6.0 (R14) Control System Toolbox Software

This table summarizes what’s new in Version 6.0 (R14):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary on page 23.	Fixed bugs Details below

New features and changes introduced in this version are

- “Linearization Tools Moved to Simulink® Control Design™ Software” on page 41
- “Enhanced Numerical Algorithms” on page 42
- “Support for Model Order Reduction Is Improved” on page 42
- “Modal Decompositions Commands Expanded” on page 42
- “Fixed Bugs” on page 42

Linearization Tools Moved to Simulink Control Design Software

The LTI Viewer interface with the Simulink linearization tools has been moved from the Control System Toolbox product to the Simulink Control Design product.

If you have Simulink Control Design software installed on your system, type `slview` to open the Linear Analysis Tool.

Compatibility Considerations

To perform linearization, you must install Simulink Control Design software. You can no longer perform this function using Control System Toolbox software.

Enhanced Numerical Algorithms

The numerical engine has been overhauled to leverage the state-of-the-art LAPACK and SLICOT libraries, resulting in faster and more accurate computations. This upgrade benefits the following areas:

- Computation of system zeros (zero, ss to tf or zpk conversions)
- Lyapunov and Riccati solvers (lyap, dlyap, care, dare)
- Model order reduction (balreal and modred)

This release also introduces several new foundation algorithms, including:

- Square-root solvers for stable Lyapunov equations (lyapchol and dlyapchol)
- Block diagonalization algorithm (bdschur)
- Commands for reordering eigenvalues in Schur decompositions (ordschur and ordqz). These commands are part of the MATLAB product.

Support for Model Order Reduction Is Improved

The support for model order reduction has been significantly improved:

- balreal now leverages the new square-root Lyapunov solvers for greater robustness, accuracy, and performance.
- Reduction of unstable or marginally stable systems is now supported.

Modal Decompositions Commands Expanded

New commands have been added for additive modal decompositions:

- stabsep separates the stable and unstable modes of a system.
- modsep performs modal decompositions into N arbitrary regions.

Fixed Bugs

The following bugs were fixed in this release:

- Issues with frequency-unit conversions in the LTI Viewer
- allmargin issues for some discrete models with poles near $z=1$

- Empty `sigma` plot for MIMO system with zero gain between some I/O pairs
- Issues with `rlocus` command applied to improper systems

Version 5.2.1 (R13SP2) Control System Toolbox Software

This table summarizes what's new in Version 5.2.1 (R13SP2):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
No	No	No fixed bugs

Version 5.2 (R13) Control System Toolbox Software

This table summarizes what's new in Version 5.2 (R13):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	No	Fixed bugs Details below

New features and changes introduced in this version are

- “Response Plots” on page 45
- “Importing in the LTI Viewer” on page 46
- “SISO Design Tool” on page 46
- “c2d Function Is Enhanced” on page 47
- “Fixed Bugs” on page 47

Response Plots

New features for response plots include

- Normalization of time plots for easy trend comparison.
- Display of driving input signal in linear simulation (lsim) plots.
- Standard zooming for all response plots.
- The **Full View** right-click menu.
- Ability to toggle magnitude and phase displays in Bode plots.
- In Bode plots, a modified phase-correction algorithm that produces consistent phase at $w=\text{Inf}$ and is robust to perturbations of the pole/zero structure at the origin.
- Displaying only positive frequencies in Nyquist plots.
- A new function, `iopzmap`, that plots the transfer poles and zeros for each I/O pair in a MIMO system.

Importing in the LTI Viewer

The following are new features and enhancement in the LTI Viewer.

Models with Unequal Number of I/Os Can Be Imported

The LTI Viewer no longer restricts importing models with unequal numbers of inputs and/or an unequal numbers of outputs. You can import any combination of LTI models into the viewer.

Import and Export Dialog Boxes Easier to Use

The import and export dialog boxes are now easier to use.

System Identification Toolbox Models Can Be Imported

The LTI Viewer now imports System Identification Toolbox SISO models. The noise model is automatically discarded and the `idmodel` is converted to ZPK format.

Zoom Property in the LTI Viewer Right-Click Menu Is Removed

The zoom property in the LTI Viewer right-click menu has been removed. You can access this functionality from the new toolbar.

SISO Design Tool

The following are new features in the SISO Design tool.

Additional Feedback Structures

The SISO Design tool supports two new feedback structures:

- Feedforward configuration
- Cascade configuration with filter \mathbf{F} in the minor loop

All the usual functionality (e.g., graphical tuning of \mathbf{F} and \mathbf{G}) is available in the new feedback structures.

Analysis Menu

The SISO Design tool contains a new **Analysis** menu that raises **Loop Responses** under **Tools** to the top level. New plots available from this menu are

- Input disturbance rejection
- Output disturbance rejection

Linked LTI Viewers Are Dynamically Updated

LTI Viewers that are linked to the SISO Design tool update response plots in real-time during mouse-driven compensator tuning. This is the default behavior. To deactivate this feature, clear the **Real-Time Update** check box in the linked LTI Viewer. In this case, plots update only when you release the mouse button.

System Identification Toolbox Models Can Be Imported

The SISO Design tool now imports System Identification Toolbox SISO models. The noise model is automatically discarded and the `idmodel` is converted to ZPK format.

Reworked Store and Retrieve Windows

A new window, the **Design Archive**, combines both store and retrieve functions within a single, more-intuitive GUI.

Nichols Plots

You can now trace the frequency by placing your mouse cursor over the Nichols contour.

c2d Function Is Enhanced

The `c2d` function has a new impulse-invariant discretization method that guarantees matching continuous and discrete responses for pulse inputs.

Fixed Bugs

The following bugs were fixed in this release:

allmargin

The `allmargin` function converts ZPK models to TF format only when there is an algebraic loop in your system. This change ensures that the SISO Design tool can correctly diagnose closed-loop stability (open loop is always in ZPK format).

dlinmod

The `dlinmod` function now correctly linearizes all multirate discrete or hybrid (continuous and discrete) Simulink models when using Jacobian-based linearization.

place

Formerly, bad edge cases in the orthogonalization loop caused a divide by zero in the `place` function. This is no longer the case.

Compatibility Summary for Control System Toolbox Software

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided with the description of the new feature or change.

Version (Release)	New Features and Changes with Version Compatibility Impact
Latest Version V9.3 (R2012a)	See the Compatibility Considerations subheading for new features and changes: <ul style="list-style-type: none"> “Functionality Being Removed or Changed” on page 8
V9.2 (R2011b)	See the Compatibility Considerations subheading for new features and changes: <ul style="list-style-type: none"> “Functionality Being Removed or Changed” on page 10
V9.1 (R2011a)	See the Compatibility Considerations subheading for this change: <ul style="list-style-type: none"> “Discrete-Time PID Controller Objects Have Stable Derivative Filter Pole” on page 13
V9.0 (R2010b)	See the Compatibility Considerations subheading for this change: <ul style="list-style-type: none"> “Change in Output of repsys Command” on page 19
V8.5 (R2010a)	See the Compatibility Considerations subheading for this change: <ul style="list-style-type: none"> “New Commands for Specifying Options for Continuous-Discrete Conversions” on page 21
V8.4 (R2009b)	None

Version (Release)	New Features and Changes with Version Compatibility Impact
V8.3 (R2009a)	See the Compatibility Considerations subheading for this new feature or change: <ul style="list-style-type: none"> • “Variable q Now Defined as the Forward Shift Operator z” on page 23
V8.2 (R2008b)	See the Compatibility Considerations subheading for this new feature or change: <ul style="list-style-type: none"> • “Current Flag Moved from lqgreg to kalman” on page 26
V8.1 (R2008a)	None
V8.0.1 (R2007b)	None
V8.0 (R2007a)	None
V7.1 (R2006b)	None
V7.0 (R2006a)	None
V6.2.1 (R14SP3)	None
V6.2 (R14SP2)	None
V6.1 (R14SP1)	None
V6.0 (R14)	See the Compatibility Considerations subheading for this new feature or change: <ul style="list-style-type: none"> • “Linearization Tools Moved to Simulink® Control Design™ Software” on page 41
V5.2.1 (R13SP2)	None
V5.2 (R13)	None