## Simulink<sup>®</sup> Design Optimization™ Getting Started Guide

**R**2014**b** 

# MATLAB® & SIMULINK®



## How to Contact MathWorks

Latest news:	www.mathworks.com
Sales and services:	www.mathworks.com/sales_and_services
User community:	www.mathworks.com/matlabcentral
Technical support:	www.mathworks.com/support/contact_us
Phone:	508-647-7000

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

#### Simulink<sup>®</sup> Design Optimization<sup>™</sup> Getting Started Guide

#### © COPYRIGHT 1993–2014 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.



T

#### **Revision History**

March 2009	Online only
September 2009	Online only
March 2010	Online only
September 2010	Online only
April 2011	Online only
September 2011	Online only
March 2012	Online only
September 2012	Online only
March 2013	Online only
September 2013	Online only
March 2014	Online only
October 2014	Online only

New for Version 1 (Release 2009a) Revised for Version 1.1 (Release 2009b) Revised for Version 1.1.1 (Release 2010a) Revised for Version 1.2 (Release 2010b) Revised for Version 1.2.1 (Release 2011a) Revised for Version 2.0 (Release 2011b) Revised for Version 2.1 (Release 2012a) Revised for Version 2.2 (Release 2012b) Revised for Version 2.3 (Release 2013a) Revised for Version 2.4 (Release 2013b) Revised for Version 2.5 (Release 2014a) Revised for Version 2.6 (Release 2014b)

## Contents

#### **Product Overview**

Simulink Design Optimization Product Description Key Features	1-2 1-2
Optimization Support for Simulink Models Using Third-Party Applications	1-3
Speeding Up Design Optimization Using Parallel Computing	1-4
Required and Related Products	1-5

1

2

#### **Parameter Estimation**

#### Supported Data 2-2Prepare Data for Parameter Estimation ..... 2 - 3About This Tutorial ..... 2-3Start a Parameter Estimation Tool Session ..... 2-4Create an Experiment for Parameter Estimation ..... 2-6Import Data ..... 2-7Analyze Data 2-72-9 Extract Data for Estimation Replacing Outliers ..... 2-10 Filtering Data ..... 2 - 12Saving the Session ..... 2 - 14Estimate Parameters from Measured Data 2 - 16About This Tutorial ..... 2 - 16

Estimate Model Parameters Using Default Estimation	
Settings	2-19
Improve Estimation Results Using Parameter Bounds	2-29
Validate Estimated Model Parameters	2-32

## **Response Optimization**

S	Supported Design Requirements
Ι	Design Optimization to Meet Step Response Requirements
	(GUI)
	Model Structure
	Design Requirements
	Specify Step Response Requirements
	Specify Design Variables
	Optimize Model Response
	Save the Session
т	Design Antimization to Most Ston Peanonce Peaninements
1	Design Optimization to Meet Step Response Requirements (Code)
	Model Structure
	Design Requirements
	Specify Step Response Requirements
	Specify Design Variables
	Optimize Model Response
I	Design Optimization to Track Reference Signal (GUI)
	Model Structure
	Design Requirements
	Specify Reference Signal
	Specify Design Variables
	Optimize Model Response
I	Design Optimization Using Frequency-Domain Check Block
	(GUI)
	Model Structure
	Design Requirements
	Specify Design Requirements
	Specify Design Variables

Optimize Design	3-46
Time-Domain Model Verification	3-52

## **Optimization-Based Linear Control Design**

When to Use Optimization-Based Linear Control Design	4-2
Types of Time- and Frequency-Domain Design Requirements for Optimization-Based Control Design	4-3
$\mathbf{Quick} \ \mathbf{Start} - \mathbf{Optimization}$ -Based Linear Control Design .	4-4
Design Optimization-Based PID Controller for Linearized	
Simulink Model (GUI)	4-15
About This Tutorial	4-15
Configuring a Project for Optimization-Based Control Design Designing an Initial PID Controller to Meet Bode Magnitude	4-16
and Phase Margins Requirements Refining the Controller Design to Meet Controller Output	4-22
Bounds	4-42
Saving the Project	4-57

4

## **Product Overview**

- "Simulink Design Optimization Product Description" on page 1-2
- "Optimization Support for Simulink Models Using Third-Party Applications" on page 1-3
- "Speeding Up Design Optimization Using Parallel Computing" on page 1-4
- "Required and Related Products" on page 1-5

## Simulink Design Optimization Product Description

Estimate and optimize Simulink model parameters

Simulink Design Optimization provides interactive tools, functions, and Simulink blocks for estimating and tuning Simulink model parameters using numerical optimization. An interactive tool lets you automatically estimate model parameters such as friction and aerodynamic coefficients from test data to increase model accuracy. You can preprocess test data, select model parameters to estimate, start an optimization, and validate estimation results.

You can also automatically tune design parameters in a Simulink model to meet objectives such as improved system performance and minimized energy consumption. Using design optimization techniques, you can meet both time- and frequency-domain constraints such as overshoot and phase margin. You can also jointly optimize physical plant parameters and algorithmic or controller gains to maximize overall system performance.

## **Key Features**

- · Model parameter estimation from test data
- Simultaneous optimization of time- and frequency-domain responses of Simulink models (with Simulink Control Design<sup>™</sup>)
- Graphical specification of response requirements and visual monitoring of the optimization progress
- Optimization of parameters to meet requirements specified by Model Verification blocks
- · Custom constraints and cost functions for response optimization
- · Scripting interface for programmatic specification of design optimization problems
- · Robust design optimization, accounting for parameter variation or uncertainty

## Optimization Support for Simulink Models Using Third-Party Applications

You can use Simulink Design Optimization software to optimize Simulink models that invoke third-party simulation tools or contain legacy simulation code. To do so, use the S-function block in Simulink. When using the command-line functions, use MATLAB<sup>®</sup> MEX functions.

## References

Cherian, V., Shenoy, R., Stothert, A., Shriver, J. et al., "Model-Based Design of a SUV Anti-rollover Control System" SAE Technical Paper 2008-01-0579, 2008, doi:10.4271/2008-01-0579.

## More About

• "Introducing MEX-Files"

## Speeding Up Design Optimization Using Parallel Computing

When you have the Parallel Computing Toolbox<sup>™</sup> software, you can use parallel computing to speed up parameter estimation and response optimization. When you use parallel computing, the software distributes the independent simulations on multiple MATLAB sessions. Thus, the simulations run in parallel which reduces the optimization time.

Using parallel computing may reduce the optimization time in the following cases:

- The model contains a large number of parameters to optimize, and the Gradient descent or Nonlinear least squares method is selected.
- ThePattern search method is selected as the optimization method.
- The model contains a large number of uncertain parameters and uncertain parameter values.
- The model is complex and takes a long time to simulate.

## **Required and Related Products**

Simulink Design Optimization software requires MATLAB, Simulink, and Optimization Toolbox  $^{\rm TM}$  software.

The following table summarizes MathWorks<sup>®</sup> products that extend and complement the Simulink Design Optimization software. For current information about these and other MathWorks products, visit http://www.mathworks.com/products/.

Product	Description
Control System Toolbox	Enables you to design controllers for linear time-invariant (LTI) models using optimization methods.
Global Optimization Toolbox	Provides genetic algorithms, and direct search methods to estimate and optimize model parameters.
Neural Network Toolbox	Provides Simulink models of neural networks for optimization-based control design.
Parallel Computing Toolbox	Enables parallel computing on multicore processors and multiprocessor networks to speed up estimation and optimization.
Simulink Control Design	Lets you linearize Simulink models. Use Simulink Design Optimization software to design controllers for linearized models using optimization methods.
System Identification Toolbox	Lets you estimate linear and nonlinear models from measured data. Import the estimated model into Simulink software, and use Simulink Design Optimization software for optimization-based control design.

## **Parameter Estimation**

- "Supported Data" on page 2-2
- "Prepare Data for Parameter Estimation" on page 2-3
- "Estimate Parameters from Measured Data" on page 2-16

## Supported Data

From signal data, you can estimate model parameters and initial conditions of single or multiple input and output Simulink models.

Simulink Design Optimization software lets you estimate model parameters from the following types of data:

- *Time-domain* data Data with one or more input variables u(t) and one or more output variables y(t), sampled as a function of time. See "Import Data".
- Time-series data Data stored in time-series objects. See "Time-Series Data".

Simulink Design Optimization software estimates model parameters by comparing the measured signal data with simulation data generated from the Simulink model. Using optimization techniques, the software estimates the parameters and initial conditions of states to minimize a user-selected cost function. The cost function typically calculates a least-square error between the measured and simulated data. To learn more, see "Estimate Parameters from Measured Data" on page 2-16.

## More About

- "Time Series Objects"
- "Complex Data"

## **Prepare Data for Parameter Estimation**

In this section					
"About This Tutorial" on page 2-3					
"Start a Parameter Estimation Tool Session" on page 2-4					
"Create an Experiment for Parameter Estimation" on page 2-6					
"Import Data" on page 2-7					
"Analyze Data" on page 2-7					
"Extract Data for Estimation" on page 2-9					
"Replacing Outliers" on page 2-10					
"Filtering Data" on page 2-12					
"Saving the Session" on page 2-14					

## **About This Tutorial**

- "Objectives" on page 2-3
- "About the Sample Data" on page 2-4

#### **Objectives**

This tutorial explains how to import, analyze, and prepare measured input and output (I/ O) data for estimating parameters of a Simulink model.

**Note:** Simulink Design Optimization software estimates parameters from real, timedomain data only.

Perform the following tasks using the Parameter Estimation tool:

- Import data from the MATLAB workspace.
- Analyze data quality using a time plot.
- Select a subset of data for estimation.
- Replace outliers.
- Filter high-frequency noise.

#### About the Sample Data

Load spe\_engine\_throttle1.mat, which contains I/O data measured from an engine throttle system. The MAT-file includes the following variables:

- input1 Input data samples
- position1 Output data samples
- time1 Time vector

**Note:** The number of input and output data samples must be equal to the length of the corresponding time vector.

The engine throttle system controls the flow of air and fuel mixture to the engine cylinders. The throttle body contains a butterfly valve which opens when a driver presses the accelerator pedal. Opening this valve increases the amount of fuel mixture entering the cylinders, which increases the engine speed. A DC motor controls the opening angle of the butterfly valve in the throttle system. The input to the throttle system is the motor current (in amperes), and the output is the angular position of the butterfly valve (in degrees).

spe\_engine\_throttle1 contains the Simulink model of the engine throttle system.

### Start a Parameter Estimation Tool Session

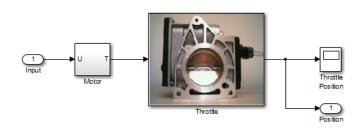
To perform parameter estimation, you must first start a Parameter Estimation tool session.

**1** Open the engine throttle system model by typing the following at the MATLAB prompt:

spe\_engine\_throttle1

This command opens the Simulink model, and loads the data into the MATLAB workspace.

#### Engine Throttle Model



2 In the Simulink model window, select Analysis > Parameter Estimation.

This action opens a new session, **Parameter Estimation - spe\_engine\_throttle1**, in the Parameter Estimation tool.

📣 Parameter Estimat	ion - spe_engi	ne_throttle1						
PARAMETER ESTI	MATION	VALID	ATION	VIEW				🖪 🗄 🖌 🖬 🖆 🕁 🖻 🔁 🗖 🔼
Copen Session ▼	Select Parameters	New Experiment	Select Experiments	Add Plot	Plot Model Response	Cost Function: Sum Squared Error 💌	Estimate	
FILE	PARAMETERS	EXPER	RIMENTS	PL	OTS	OPTIONS	ESTIMATE	
Data Browser		$\overline{\mathbf{v}}$						
<ul> <li>Parameters</li> </ul>								
<ul> <li>Experiments</li> </ul>								
<ul> <li>Results</li> </ul>								
▼ Preview								

Note: The Simulink model must remain open to perform parameter estimation tasks.

## **Create an Experiment for Parameter Estimation**

In the Parameter Estimation tool on the **Parameter Estimation** tab, click the **New Experiment** button. This will create an experiment with the name Exp in the **Experiments** list on the left pane. You can rename it by right-clicking and selecting **Rename** from the list. For example, call it NewData1.

### Import Data

This portion of the tutorial explains how to import measured I/O data into the experiment in the Parameter Estimation tool. Importing data assigns the data to the corresponding model input and output signals.

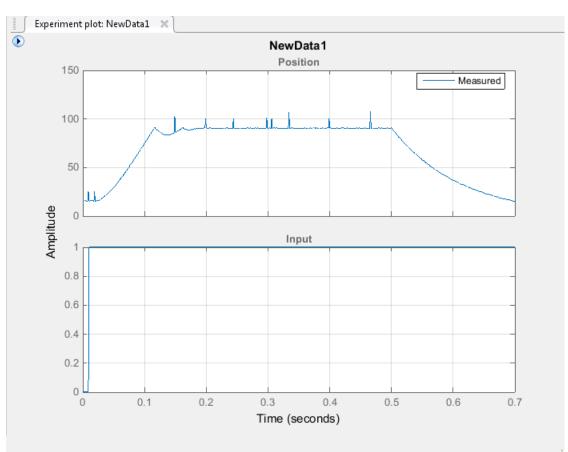
The model input and output signals are designated with the Inport Input and Outport **Position** blocks, respectively, as shown in the figure. To learn more about the blocks, see the "Inport" and "Outport" block reference pages in the Simulink documentation.

To import data into the experiment, right-click and select **Edit...** to launch the experiment editor. Import the output data by typing [time1,position1] in the dialog box in the **Outputs** panel. Import the input data by typing [time1,input1] in the dialog box in the **Inputs** panel.

Edit Experiment: NewData1	X
Outputs         Define measured output signals for this experiment.         spe engine throttle1/Throttle:1 (Position)         <1x1 Signal, 701 points>         Select Measured Output Signals	
Inputs         Optionally define input signals for this experiment.         spe engine throttle1/Input:1 (Input)         [time1,input1]         Image: Select Inputs	E

## Analyze Data

This portion of the tutorial explains how to analyze the output data quality by viewing the data characteristics on a time plot. Based on the analysis, you can decide whether to preprocess the data before estimating parameters. For example, if the data contains noise, you might want to filter the noise from the system dynamics before estimating parameters.



To create an experiment plot, click Add Plot on the Parameter Estimation tab and select the experiment name, for example, NewData1 under Experiment Plots.

The time plot shows the output data in response to a step input, as described in "About the Sample Data" on page 2-4. The plot shows a rapid decrease in the response after t = 0.5 s because the system is shut down. To focus parameter estimation on the time period when the system is active, select the data samples between t = 0 s and t = 0.5 s, as in "Extract Data for Estimation" on page 2-9.

The spikes in the data indicate *outliers*, defined as data values that deviate from the mean by more than three standard deviations. They might be caused by measurement errors or sensor problems. The response also contains noise. Before estimating model

parameters from this data, remove the outliers and filter the noise, as described in "Replacing Outliers" on page 2-10and "Filtering Data" on page 2-12.

You can also plot the experiment data by right-clicking the experiment, for example NewData1, and selecting Plot measured experiment data from the list.

### **Extract Data for Estimation**

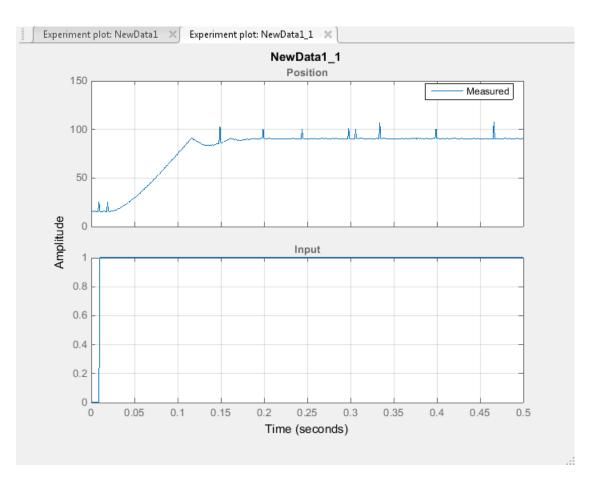
This portion of the tutorial explains how to select a subset of I/O data for estimation. As described in "Analyze Data" on page 2-7, the system is shut down at t = 0.5 s. To focus the estimation on the time period before t = 0.5 s, exclude the data samples beyond t = 0.5 s. This operation selects the data between t = 0 s and t = 0.5 s for estimation.

First, import the data into the experiment, as described in "Import Data" on page 2-7.

To select the portion of data between t = 0 s and t = 0.5 s:

- 1 Plot the measured data as described in "Analyze Data" on page 2-7, to have access to the **Experiment Plot** tab.
- 2 On the Experiment Plot tab, click Extract Data to launch the Extract Data tab.
- **3** Enter 0 in the **Start Time:** field and 0.5 in the **End Time:** field.
- 4 Click Save As to save data in a new experiment, for example, NewData1\_1.

The Parameter Estimation tool extracts the corresponding input data. To plot the new data, click on Add Plot on the Parameter Estimation tab. Select the experiment name, for example, NewData1\_1 in the Experiment Plots list to display the experiment plot of the data from t = 0 s to t = 0.5 s.



## **Replacing Outliers**

- "Why Replace Outliers" on page 2-10
- "How to Replace Outliers" on page 2-11

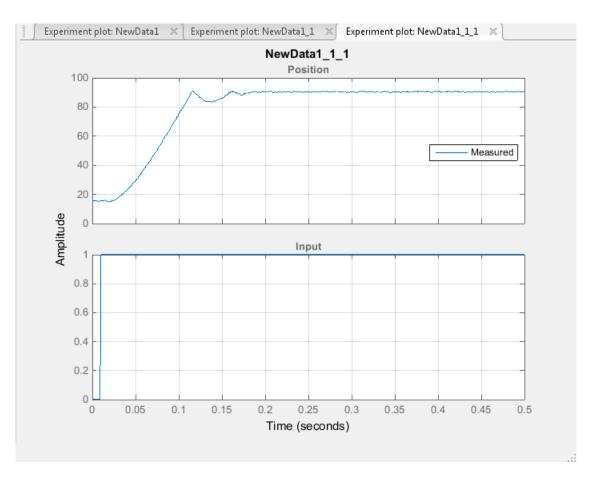
#### Why Replace Outliers

Outliers are data values that deviate from the mean by more than three standard deviations. When estimating parameters from data containing outliers, the results might not be accurate. Hence, you might choose to replace the outliers in the data before you estimate the parameters.

#### How to Replace Outliers

In the experiment plot of the data extracted as in "Extract Data for Estimation" on page 2-9, you can visually detect the data points that seem to be outliers. To replace these points:

- 1 In the **Experiment Plot** tab, click **Replace data** to launch the **Replace data** tab. The experiment plot shows the preview data, which is in light brown. On the preview, select the data point that you want to replace.
- 2 On the **Replace Data** tab, click **Replace data** and select the constant value. For example, replace the output signal data that correspond to time points 0.00899 and 0.0189 with 15, that corresponds to the time point 0.149 with 86, and the rest of the outlier data points with 90.
- 3 Click the arrow in the **Update** section and select **Save As: Create a new experiment from the modified data**. Parameter Estimation tool saves the modified data in the new experiment, for example, NewData1\_1\_1.
- 4 Click Add Plot on the Parameter Estimation tab and select the new experiment, for example, NewData1\_1\_1. This creates an experiment plot of the modified data. The spikes, that indicated outliers, no longer appear on the time plot.



## **Filtering Data**

This portion of the tutorial explains how to filter the noise and remove any periodic trends from the output data. First remove the outliers from the output data, as described in "Replacing Outliers" on page 2-10.

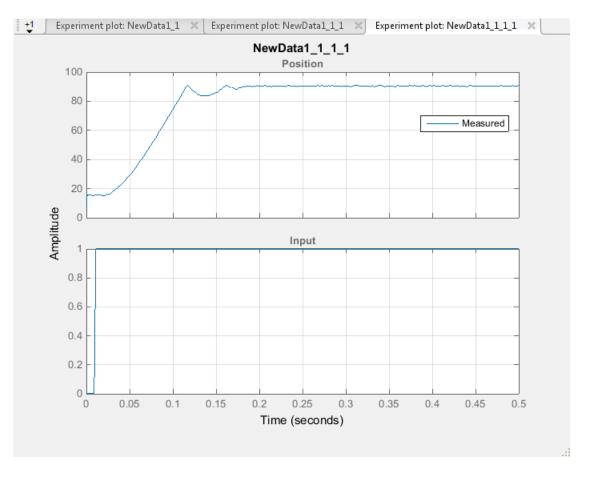
Click the experiment plot for the new experiment, for example, NewData1\_1\_1. On the **Experiment Plot** tab, click **Low-Pass Filter**.

1 On the Low-Pass Filter tab select Filter all signals.

- 2 Enter 0.4 in the Normalized cutoff frequency field.
- **3** Click **Options**. Enter 1 in the **Filter order** field and click **OK**.

Filter Options			×
Filter order:	1		
🔲 Zero-pha	ase sh	ift filter	
	ок	Cancel	Help

- 4 Click the arrow in the **Update** section and select **Save As: Create a new experiment from the modified data**. Parameter Estimation tool saves the modified data in the new experiment, for example, NewData1\_1\_1.
- 5 Click Add Plot on the Parameter Estimation tab and select the new experiment, NewData1\_1\_1\_1. This creates an experiment plot of the modified data. The noise is filtered and the output data appears smooth.



## Saving the Session

After you prepare the data, delete the data in the older experiments, for example, New Data1, New Data1\_1, New Data1\_1\_1. You can rename the last experiment, for example, NewData1\_1\_1\_1 as NewData1, and save the session.

To delete the experiments, right-click the experiment name in the **Experiments** pane, and select **Delete** from the list.

To save the session, click **Save Session** on the **Parameter Estimation** tab to select where to save the session. Specify a name for the session, for example,

spe\_engine\_throttle1\_sdosession.mat in the File name or Session field, and then click Save or OK. This saves your parameter estimation session as a MAT-file.

To learn how to estimate parameters from this data, see "Estimate Parameters from Measured Data" on page 2-16.

## **Estimate Parameters from Measured Data**

#### In this section ...

"About This Tutorial" on page 2-16

"Estimate Model Parameters Using Default Estimation Settings" on page 2-19

"Improve Estimation Results Using Parameter Bounds" on page 2-29

"Validate Estimated Model Parameters" on page 2-32

## **About This Tutorial**

- "Objectives" on page 2-16
- "About the Model" on page 2-16

#### **Objectives**

This tutorial shows how to estimate parameters of a single-input single-output (SISO) Simulink model from measured input and output (I/O) data.

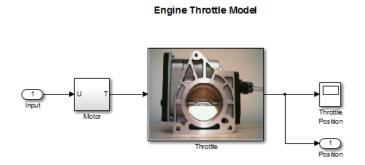
**Note:** Simulink Design Optimization software estimates parameters from real, timedomain data only.

You can perform the following tasks using the Parameter Estimation tool:

- · Load a saved session containing data
- · Estimate model parameters using default settings
- Validate the model, and refine the estimation results

#### About the Model

This tutorial uses the **spe\_engine\_throttle1** Simulink model, which represents an engine throttle system.

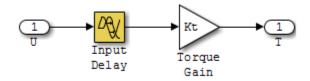


The throttle system controls the flow of air and fuel mixture to the engine cylinders. The throttle body contains a butterfly valve that opens when a driver presses the accelerator pedal. Opening this valve increases the amount of fuel mixture entering the cylinders, which increases the engine speed. A DC motor controls the opening angle of the butterfly valve in the throttle system. The models for these components are described in "Motor Subsystem" on page 2-17 and "Throttle Subsystem" on page 2-18.

The input to the throttle system is the motor current (in amperes), and the output is the angular position of the butterfly valve (in degrees).

#### **Motor Subsystem**

The Motor subsystem contains the DC motor model. To open the model, double-click the corresponding block.

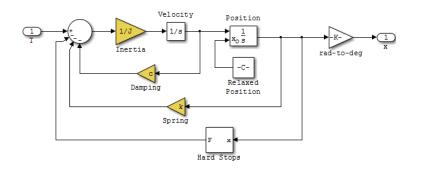


Components of the Motor subsystem	Description
Variables	U is the input current to the motor.
	T is the torque applied by the motor.
Parameters	$K_t$ is the torque gain of the motor, represented by Kt in the model.

Components of the Motor subsystem	Description
	$t_d$ is the input time delay of the motor, represented by input_delay in the model.
Equation	The torque applied by the motor is described in the following equation: $T(t) = K_t U(t - t_d)$ where t is time.
Input	U
Output	Т

#### **Throttle Subsystem**

The Throttle subsystem contains the butterfly valve model. To open the model, rightclick the corresponding block, and select Mask > Look Under Mask.



The Hard Stops block models the valve angular position limit of  $15^{\circ}$  to  $90^{\circ}$ .

The following table describes the variables, parameters, states, differential equations, inputs, and outputs of the .

Components of the Throttle subsystem	Description
Variables	T is the torque applied by the DC motor.
	$\theta$ is the angular position of the valve, represented by X in the model.

Components of the Throttle subsystem	Description				
	$T_{hardstop}$ is the torque applied by the hard stop.				
Parameters	J is the valve inertia.				
	<b>c</b> is the valve viscous friction.				
	k is the valve spring constant.				
States	$\theta$ is the angular position.				
	$\dot{\theta}$ is the angular velocity.				
Equations	The mathematical system for the butterfly valve is described in the following equation:				
	$J\ddot{\theta} + c\dot{\theta} + k\theta = T + T_{hardstop}$				
	where $15^{\circ} \le \theta \le 90^{\circ}$ , with initial conditions				
	$\theta_0 = 15^\circ$ , and $\dot{\theta}_0 = 0$ .				
	The torque applied by the Hard Stops block is described in the following equation:				
	$T_{hardstop} = \begin{cases} 0, & 15^{\circ} \le \theta \le 90^{\circ} \\ K(90^{\circ} - \theta), \theta > 90^{\circ} \\ K(15^{\circ} - \theta), \theta < 15^{\circ} \end{cases}$				
	$\left( \mathbf{A} (15 - 0), 0 < 15 \right)$				
	where $\boldsymbol{K}$ is the gain of the Hard Stops block.				
Input	Т				
Output	$\theta$				

## Estimate Model Parameters Using Default Estimation Settings

- "Overview of the Estimation Process" on page 2-20
- "Specify Estimation Data and Parameters" on page 2-20

#### **Overview of the Estimation Process**

Simulink Design Optimization software uses optimization techniques to estimate model parameters. In each optimization iteration, it simulates the model with the current parameter values. It computes and minimizes the error between the simulated and measured output. The estimation is complete when the optimization method finds a local minimum.

To start the estimation process, first open the engine throttle system Simulink model by typing the following at the MATLAB prompt:

spe\_engine\_throttle1

In the Simulink Editor, select Analysis > Parameter Estimation.

This action opens a new session with the name **Parameter Estimation - spe\_engine\_throttle1** in the Parameter Estimation tool.

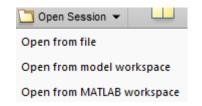
Note: The Simulink model must remain open to perform parameter estimation tasks.

#### **Specify Estimation Data and Parameters**

**1** Load or import the estimation data.

A Parameter Estimation - spe_engine_throttle1									
PARAMETER EST	IMATION	VALID	ATION	VIEW				à É S (	2 🗗 🕐 🔼
Copen Session ▼	Select Parameters	New Experiment	Select Experiments	Add Plot	Plot Model Response	Cost Function: Sum Squared Error 💌	Estimate		
FILE	PARAMETERS		RIMENTS	P	LOTS	OPTIONS	ESTIMATE		
Data Browser									
<ul> <li>Parameters</li> </ul>									
▼ Experiments									
NewData1									
Nucor Data									
▼ Results									
▼ Preview									
L									

**a** If you prepared data and saved your session as described in "Prepare Data for Parameter Estimation" on page 2-3, load the preconfigured session. On the **Parameter Estimation** tab, click the **Open Session** drop down list.



Select the correct option to browse to the location of your saved session, for example, Open from file. Then select the MAT-file.

**b** If you do not have a previously saved session, create a new experiment. on the **Parameter Estimation** tab, click **New Experiment**. In the **Experiments** list on the left pane. You can rename it by right-clicking and selecting **Rename** from the list. For example, call it NewData1.

To import data into the experiment, right-click and select **Edit...** to launch the experiment editor. Import the output data by typing in the dialog box in the **Outputs** panel, for example [time1,position1]. Import the input data by typing in the dialog box in the **Inputs** panel, for example [time1,input1].

Edit Experiment: NewData1	X
Outputs         Define measured output signals for this experiment.         spe engine throttle1/Throttle:1 (Position)         <1x1 Signal, 701 points>         Select Measured Output Signals	*
Inputs         Optionally define input signals for this experiment.         spe engine throttle1/Input:1 (Input)         [time1,input1]         The select Inputs	ш

2 Specify parameters for estimation. On the **Parameter Estimation** tab, click the **Select Parameters** button to open the **Edit:Estimated Parameters** dialog box. In the **Parameters Tuned for all Experiments** panel, click the **Select parameters** button to launch the **Select Model Variables** dialog box.

Select the parameters J, C, input\_delay, and k, and click OK.

•	Variable	Current value	Used By			
1	J	0.05	spe_engine_throttle1/Throttle			
	Kt	174.53	spe_engine_throttle1/Motor/TorqueGain			
	angle_init	15	spe_engine_throttle1/Throttle			
	angle_open	90	spe_engine_throttle1/Throttle			
1	c	40	spe_engine_throttle1/Throttle			
	input1	[0;0;0;0;0;0;0;0;0;0;0;				
1	input_delay	0.02	spe_engine_throttle1/Motor/Input Delay			
<b>V</b>			<u>spe_engine_throttle1/Throttle</u>			
	time1	[0;0.016638935108				
Specify e	expression indexing if n	ecessary (e.g., a(3) or s	5.X) 🖋 (	ок 💢 с	ancel	(?) H

The Edit:Estimated Parameters window now looks as follows.

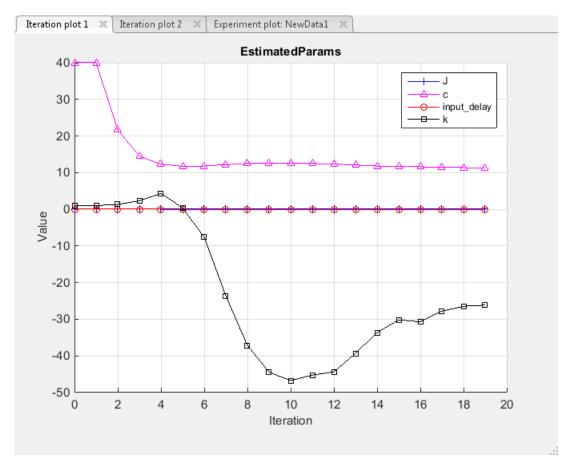
Edit: Estimated Parameters	×
Parameters Tuned for all Experiments	1
1	
▶ 0.05	🗶 🖾 Estimate
2	
▶ 40   ♥	🗶 🖾 Estimate
input delay	
▶ 0.02	🗶 🗹 Estimate
<u>k</u>	
▶ 1 · · · ·	🗙 🖾 Estimate
Select parameters	
Parameters and Initial States Tuned per Experiment	t
There are no experiments selected for estimation.	
Select Experiments	
End Coloci Experimento	
🔚 Update Model	؇ OK ( ?) Help

The tool selects the parameters you add for estimation by default. When estimating a large number of parameters, you can first select a subset of parameters to estimate.

**3** Specify an experiment for estimation. On the **Parameter Estimation** tab, click **Select Experiments**, and select the box under the **Estimation** column. Click **OK**.

Sele	ect Experime	ents						×
Se	lect experim	ents to includ	e for estimation	or valid	ation			
E	stimation	Validation	Experiment					
	<b>V</b>		NewData1					₽
								4
								]
						🖌 🔨	?	Help

- **4** To add progress plots, click **Add Plot** on the **Parameter Estimation** tab. Here you can choose the **Parameter Trajectory** and **Estimation Cost** iteration plots. You can also choose an experiment plot of measured and simulated data for NewData1.
- 5 Estimate the parameters using the default settings. On the Parameter Estimation tab, click Estimate to open the Parameter Trajectory plot and Estimation Progress Report window and estimate the parameters. The Parameter Trajectory plot shows the change in the parameter values at each iteration.



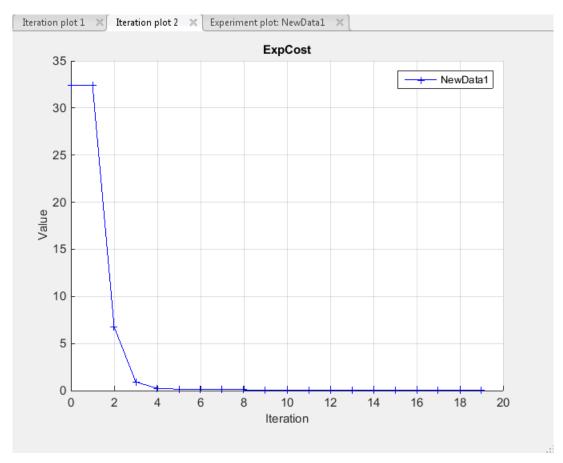
The **Estimation Progress Report** shows the iteration number, number of times the objective function is evaluated, and the value of the cost function at the end of each iteration. After the estimation converges, the **Estimation Progress Report** looks like this figure.

Estimati	on Progress	; Report
teration	F-count	NewData1
	· count	(Minimize)
0	9	32.4446
1	18	32.444
2	27	6.773
3	36	0.9226
4	45	0.229
5	54	0.150
6	63	0.127
7	72	0.099
8	81	0.085
9	90	0.077
10	99	0.070
11	108	0.064
12	117	0.059
13	126	0.055
14	135	0.053
15	144	0.052
16	153	0.050
17	162	0.049
18	171	0.048
19	180	0.047
timation	converged, '	Jun-2014 16:19:52
stimated (	parameter va experiment v	ipdated with estimated parameter values lues written to 'EstimatedParams' alues written to 'NewData1'
sumation	solver outpu	h
		Save Iteration Display Options Estimate

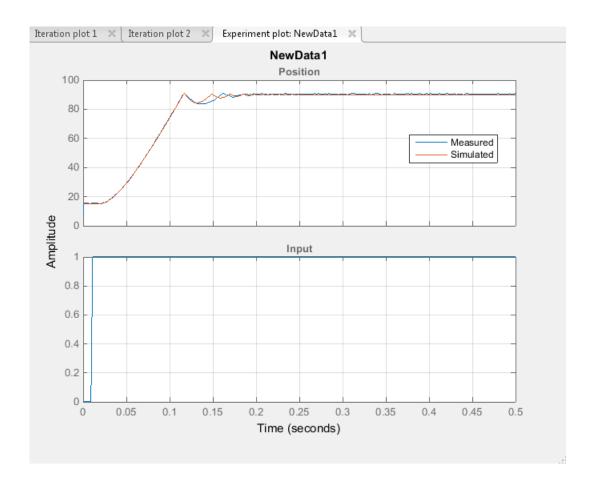
The estimated parameters are saved in the Parameter Estimation tool, in the **Results** section of the **Data Browser** pane, as EstimatedParams. Right-click EstimatedParams, and select **Open...** to view the results.

View Result : EstimatedParams
Estimation result(s):
J = 0.22708
c = 11.346
input_delay = 0.0094873
k = -26.119
Parameters estimated using experiments:
NewData1, cost = 0.047861
Solver output:
Cost: 0.047861
ExitFlag: 3
FCount: 181
Date: 16-Jun-2014 16:20
Solver termination message:
Local minimum possible.
Isqnonlin stopped because the final change in the sum of squares relative to
its initial value is less than the selected value of the function tolerance.
Stopping criteria details:
Optimization stopped because the relative sum of squares (r) is changing
by less than options.TolFun = 1.000000e-03.
Optimization Metric Options
relative change r = 6.65e-04 TolFun = 1e-03 (selected)
Use as initial guess 📓 Update Model 🛷 Of

6 Examine the estimated cost function graph. Cost function is the error between the simulated and measured output. During estimation, the default optimization method Nonlinear least squares, "lsqnonlin", minimizes the cost function by changing the parameter values. The following figure displays the change in the expected cost during iterations.



7 Examine the simulated response plot to see how well the simulated output matches the measured output. The experiment plot shows that the output simulated using the estimated parameters is close to the measured outputs.



#### Improve Estimation Results Using Parameter Bounds

You can improve the accuracy of estimation by specifying bounds on parameter values. This technique restricts the region in which the optimization method searches for a local minima.

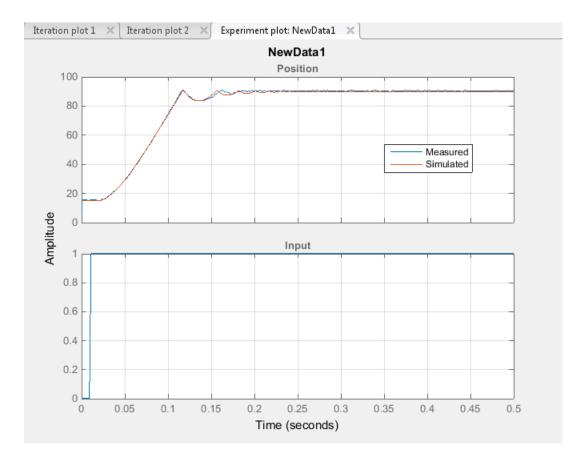
The engine throttle system has these characteristics:

- All parameter values are positive.
- Maximum time delay of the system, represented by input\_delay, is 0.1 s.

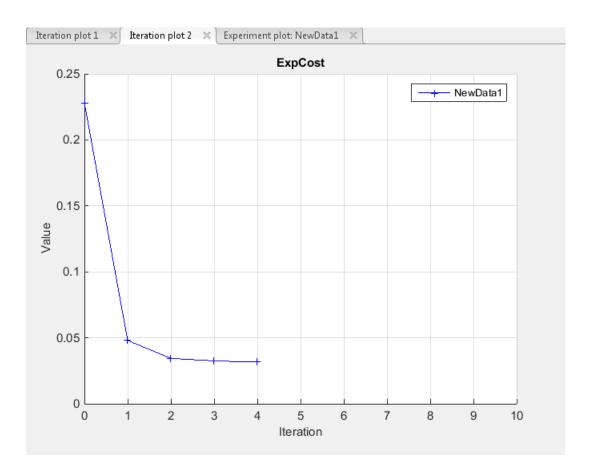
Therefore, specify 0 as the minimum value for all parameters, and 0.1 as the maximum value of input\_delay. In the Parameter Estimation tool, click the Select Parameters button to specify bounds on the parameter values. For each parameter, click the right arrow toggle to display the minimum, maximum, and scale fields. Specify the minimum value for each parameter by replacing - Inf with 0 in the Minimum field. Specify the maximum value for input\_delay by replacing +Inf with 0.1 in the corresponding Maximum field.

input dela	Υ			
▼ 0.02		•	₽	🗙 🖾 Estimate
Minimum:	0	•	₽	
Maximum:	0.1	•	₽	
Scale:	0.03125	•	₽	

After estimating the parameters, analyze the results using the experiment plot and the plot for expected cost.



The data simulated using the estimated parameter values agree better with the measured data than when the parameter limits were not specified.



#### Validate Estimated Model Parameters

After estimating model parameters, validate the model using another data set *(validation data)*. A good match between the simulated response and the validation data indicates that you have not overfitted the model.

To validate the estimated parameters using a validation data set:

1 Create a new experiment to use for validation. Name it ValidationData. Import the validation I/O data, input2 and position2, and the time vector, time2 in the ValidationData experiment. To do this, in the Parameter Estimation tool, in the Experiments pane, right-click ValidationData and select Edit... to open the experiment editor. Then, type [time2,position2] in the output dialog box and [time2,input2] in the input dialog box. For more information, see "Import Data".

Edit Experiment: ValidationData	×
Outputs	
Specify measured output signals for this experiment.	
/Throttle:1 (Position)	
[time2,position2]	🔻 🖶 🚽 💌
Select Measured Output Signals	

2 Select the experiment for validation. On the **Parameter Estimation** tab, click **Select Experiments**. By default, the ValidationData experiment is selected for estimation. Deselect the check box that corresponds to ValidationData for estimation and select the check box for validation.

Select Experi	iments		X
Select expe	riments to incl	ude for estimation or validation	
Estimatio	n Validation	Experiment	
<b>V</b>		NewData1	
	V	ValidationData	<u> </u>
			5
		🖋 ок	(?) Help

3 Select results to use. On the Validation tab, click Select Results to Validate.

PARAN	IETER ESTIMAT	ION	VALIDATION	EXPERIMENT PLOT		
New Experiment	Select Experiments	Select Res to Valida	sults 🔲 Plot Residu	red & Simulated Data als	Validate	
EXPER	RIMENTS	RESULT	S PLOT	OPTIONS	VALIDATE	

Deselect Use current parameter values and select EstimatedParams, and click OK. \$2-33\$

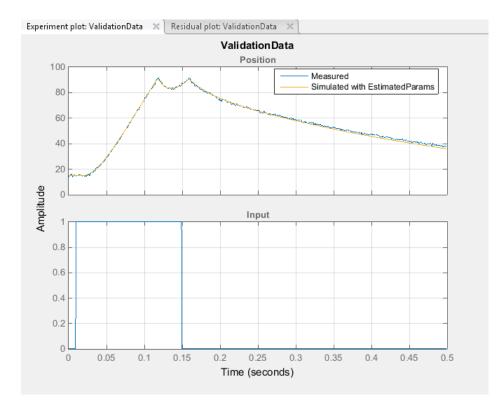
elect Re	sults ×
	Select estimation results to validate
	Results
	Use current parameter values
<b>V</b>	EstimatedParams
	Selected results are used to simulate the model and compare with measured expriment data.
	🖌 OK 🔇 Help

**4** Select the plots for measured and simulated data, and residuals on the **Validation** tab. You can assess how much the data simulated using the estimated parameters agrees with the measured data using these plots.

Plot Measured & Simulated Data
Plot Residuals

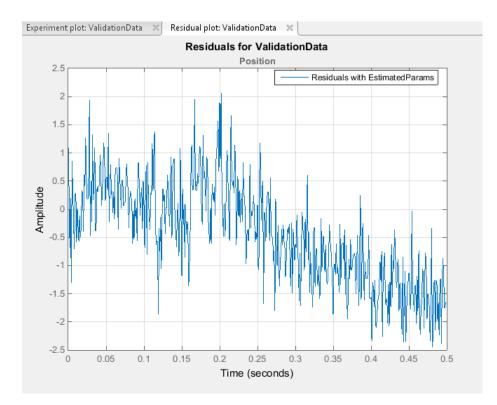
On the Validation tab, click Validate to start validation.

- **5** Examine the plots.
  - **a** Examine the experiment plot to see how well the simulated output matches the output data.



The simulated response as shown in light brown on the top experiment plot is overlaid on the measured out put data, and closely matches the measured validation data.

**b** Examine the residuals plot to compare the difference between the simulated response and measured data.



The difference between the simulated and measured data varies between 2 and -2.5. The residuals lie within 6% of the maximum output variation and do not display any systematic patterns. This indicates a good fit between the simulated output and measured data.

6 Save the session. On the Parameter Estimation tab, click Save Session.

Sele Save Session - Sele
Save to file
Save to model workspace
Save to MATLAB workspace

From the drop-down list select where to save the session. Specify the file name, and click **Save** or **OK** to save your parameter estimation session as a MAT-file.

# **Response Optimization**

- "Supported Design Requirements" on page 3-2
- "Design Optimization to Meet Step Response Requirements (GUI)" on page 3-4
- "Design Optimization to Meet Step Response Requirements (Code)" on page 3-18
- "Design Optimization to Track Reference Signal (GUI)" on page 3-25
- "Design Optimization Using Frequency-Domain Check Blocks (GUI)" on page 3-40
- "Time-Domain Model Verification" on page 3-52

## **Supported Design Requirements**

You can optimize response of Simulink models to meet time- and frequency-domain design requirements.

Simulink Design Optimization software optimizes model response by formulating the requirements into a constrained optimization problem. It then solves the problem using optimization methods.

• For time-domain requirements, the software simulates the model during optimization, compares the current response with the requirement and uses gradient methods to modify design variables (model parameters) to meet the objectives.

You can specify time-domain requirements either in blocks from the **Signal Constraints** library or without adding blocks to the model. You can also include requirements specified in Check Static Range, Check Static Lower Bound and Check Static Upper Bound blocks from the Simulink **Model Verification** library.

• For frequency-domain requirements, the software linearizes the portion of the model between specified linearization inputs and outputs, compares the linear system with the requirement and uses gradient methods to modify the design variables to meet the objectives.

If you have Simulink Control Design software, you can optimize the model to meet frequency-domain requirements, such as Bode magnitude and gain and phase margin bounds. You can specify the frequency-domain requirements without adding blocks to the model or by using the "Model Verification" blocks of the Simulink Control Design software library.

## **Related Examples**

"Design Optimization to Meet Step Response Requirements (GUI)" on page 3-4

"Design Optimization to Meet Step Response Requirements (Code)" on page 3-18

"Design Optimization to Track Reference Signal (GUI)" on page 3-25

"Design Optimization to Meet Frequency-Domain Requirements (GUI)"

"Design Optimization Using Frequency-Domain Check Blocks (GUI)" on page 3-40

"Design Optimization to Meet Time- and Frequency-Domain Requirements (GUI)"

# More About

"How the Optimization Algorithm Formulates Minimization Problems"

"Specify Time-Domain Design Requirements"

"Specify Frequency-Domain Design Requirements"

## Design Optimization to Meet Step Response Requirements (GUI)

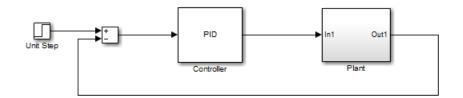
#### In this section ...

"Model Structure" on page 3-4 "Design Requirements" on page 3-5 "Specify Step Response Requirements" on page 3-5 "Specify Design Variables" on page 3-8 "Optimize Model Response" on page 3-12 "Save the Session" on page 3-15

This example shows how to optimize controller parameters to meet step response design requirements using the Design Optimization tool. You specify the design requirements in a Check Step Response Characteristics block.

#### **Model Structure**

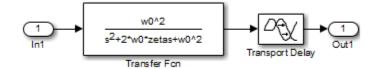
The model **sldo\_model1** includes the following blocks:



- **Controller block**, which is a PID controller. This block controls the output of the Plant subsystem.
- **Unit Step block** applies a step input and produces the model output that should meet step response requirements.

You can also use other types of inputs, such as ramp, to optimize the response to meet step response requirements generated by such inputs.

• **Plant subsystem** is a second-order system with delay. It contains "Transfer Function" and "Transport Delay" blocks.



#### **Design Requirements**

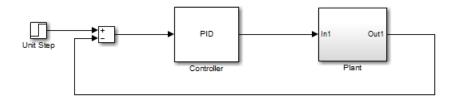
The plant output must meet the following step response requirements:

- Rise time less than 2.5 seconds
- Settling time less than 30 seconds
- Overshoot less than 5%

## Specify Step Response Requirements

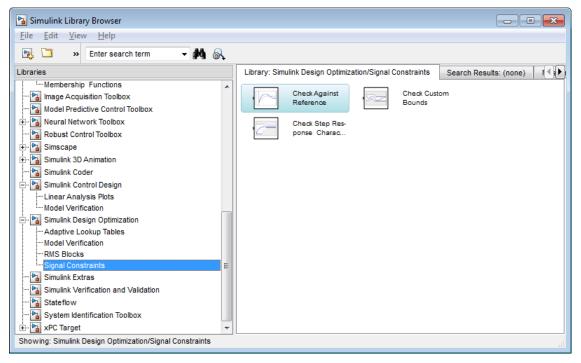
**1** Open Simulink model.

```
sys = 'sldo_model1';
open system(sys);
```

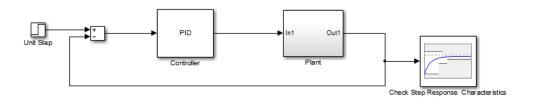


To learn more about the model, see "Model Structure" on page 3-4.

- 2 Add a Check Step Response Characteristics block to the model.
  - **a** In the Simulink model window, select **View > Library Browser**.
  - **b** In the **Libraries** pane, expand the **Simulink Design Optimization** node and select **Signal Constraints**.



- **c** Drag and drop the Check Step Response Characteristics block into the model window.
- **d** Connect the block to the output.



You must connect the block to the signal on which you want to specify design requirements.

**3** Double-click the Check Step Response Characteristics block to open the Sink Block Parameters: Check Step Response Characteristics dialog box.

Sink Block Parameters: Check Step Response Characteristics					
Check Step Response Characteristics					
Assert that the input signa	al satisfies bounds spec	ified by step respo	onse characteristics.		
Bounds Assertion					
☑ Include step response	e bound in assertion	_			
Step time (seconds):	0				
Initial value:	0	Final value:	1		
Rise time (seconds):	5	% Rise:	80		
Settling time (seconds):	7	% Settling:	1		
% Overshoot:	10	% Undershoot:	1		
Enable zero-crossing detection					
Show Plot Show plot on block open Response Optimization					
OK Cancel Help Apply					

- **4** Specify step response requirements:
  - In Rise time (seconds), enter 2.5
  - In Settling time (seconds), enter 30
  - In % Overshoot, enter 5

🔁 Sink Block Parameters: Check Step Response Characteristics							
Check Step Response Characteristics							
Assert that the input signa	Assert that the input signal satisfies bounds specified by step response characteristics.						
Bounds Assertion							
✓ Include step response	bound in assertion						
Step time (seconds):	0						
Initial value:	0	Final value:	1				
Rise time (seconds):	2.5	% Rise:	80				
Settling time (seconds):	30	% Settling:	1				
% Overshoot:	5	% Undershoot:	1				
Enable zero-crossing detection							
Show Plot Show plot on block open Response Optimization							
OK Cancel Help Apply							

#### Click OK.

Instead of specifying time-domain requirements in the Check blocks, you can specify them in the Design Optimization tool without adding blocks. For an example, see "Specify Reference Signal" on page 3-26.

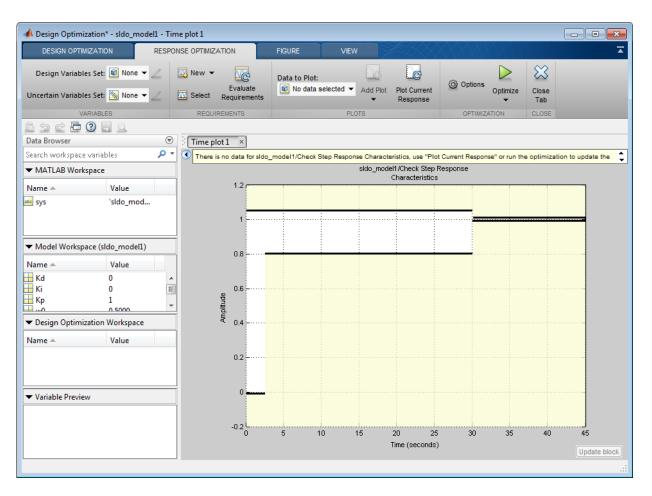
#### **Specify Design Variables**

Before you begin this task, specify the design requirements as described in "Specify Step Response Requirements" on page 3-5.

When you optimize the model response, the software modifies the design variable (parameter) values to meet the design requirements.

1 In the Simulink model window, select Analysis > Response Optimization.

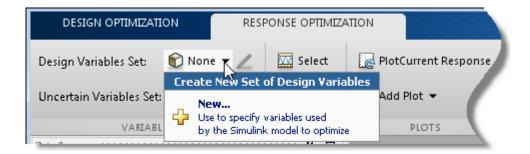
Alternatively, click **Response Optimization** in the Block Parameters dialog box.



A Design Optimization tool for the model opens.

The amplitude versus time plot graphically shows the step response requirements specified in the Check Step Response Characteristics block.

2 Select New in the Design Variables Set drop-down list.



A window opens where you specify design variables.

reate Design Variables set: DesignVars						
Variable Value Min Max	Scale	Variable	Current value	Used By		
		Kd	0	sldo model1/Cont		
		Ki	0	sldo model1/Cont.		
		Кр	1	sido model1/Control.		
		w0	0.5			
	lues	zeta	0.5	sido model1/Plant/Trai		
Variable Detail     Variable values     PO     proc model variable values       Variable Detail     Specify expression (e.q., s.x or a(3))       OK     Cancel     Her						

- **3** Click Kd, Ki and Kp to select them.
- 4

Click

to add the selected parameters to a design variables set.

Create Design Variables set: DesignVars						
<b>V</b>	Variable	Value	Minimum	Maximum	Scale	
1	Kd	0	-Inf	Inf	1	
-	Кі	0	-Inf	Inf	1	
<b>V</b>	Кр	1	-Inf	Inf	1	

The software displays the following parameter settings:

• Variable — Parameter name

- Value Current parameter value
- Minimum and Maximum Parameter bounds
- Scale Scaling factor for the parameter

The check-box indicates that the parameter is included in the design variable set. The default design variable set name is **DesignVars**.

**5** To limit the parameters to positive values, enter the minimum value of each parameter as **0** in the corresponding **Minimum** field and press **Enter**.

Create D	esign Variables set: DesignVars				
<b>V</b>	Variable	Value	Minimum	Maximum	Scale
V	Kd	0	0	Inf	1
<b>v</b>	Ki	0	0	Inf	1
<b>V</b>	Кр	1	0	Inf	1

Click **OK**. A new variable **DesignVars** appears in **Design Optimization Workspace** of the Design Optimization tool. You can click the variable to view its contents in the **Variable Preview** area.

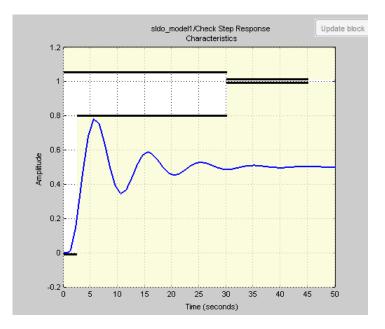
▼ Design Optimization Workspace				
Name 🔺	Value			
DesignVars	<3x1 para			
▼ Variable Previ	e\w			
DesignVars(1	,1) =			
Name:	'Kd'			
Value:	0			
Minimum:	0			
Maximum:	Inf			
Free:	1			
Scale:	1			
Info:	[1x1	struct]		

## **Optimize Model Response**

Before you begin this task, you must have already specified the design requirements and design variables as described in "Specify Step Response Requirements" on page 3-5, and "Specify Design Variables" on page 3-8, respectively.

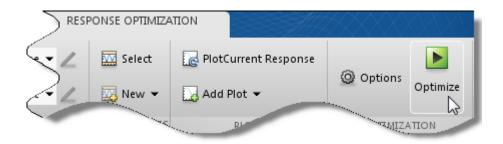
1

(Optional) View the current response of the model. Click



The plot shows that the model output does not meet the specified step response requirements.

2 Click Optimize.



An Optimization Progress window opens.

📣 Optimiz	ation Prog	jress	- • ×
Iteration	F-count	Check Step Response Characteristics (Upper) (<=0)	
			_
Optimizati	on started 0	8-Jan-2013 08:44:48	
		Save Iteration Display Options	Stop Optimization

**Tip** To view the model response and optimization progress windows simultaneously,

tile them using the plot layout area



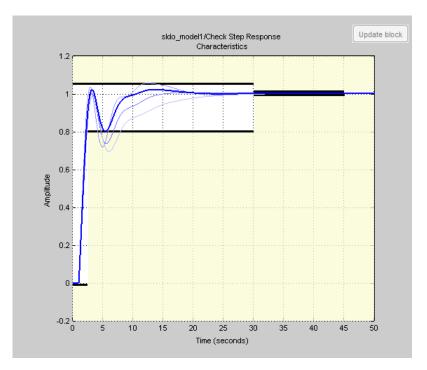
At each optimization iteration, the software simulates the model, and the default optimization solver Gradient descent (fmincon) modifies the design variables to reduce the distance between the simulated response and the design requirement line segments. For more information, see "Selecting Optimization Methods" and "How the Optimization Algorithm Formulates Minimization Problems".

After the optimization completes, the optimization progress window resembles the next figure.

Iteration	F-count	Check Step Response Characteristics (Upper) (<=0)	
0	5	(<=0)	431.9566
1			98.5409
2	21		33.6186
3	31		17.7437
4	39		3.1353
5	57		3.1368
6	76		3.1495
7	95		3.1556
8	107		1.7116
9	115		0.2930
10	123		0.0824
		8-Jan-2013 08:44:48	
· · ·		ed, 08-Jan-2013 08:45:23 lues written to 'DesignVars' in the Design Optimization workspace	

The message Optimization converged indicates that the optimization solver found a solution that meets the design requirements within the tolerances and parameter bounds. For more information about the outputs displayed in the optimization, see "Iterative Display" in the Optimization Toolbox documentation.

**3** Verify that the model output meets the step response requirements.



The plot displays the last five iterations. The final response using the optimized parameter values appears as the thick line.

The optimized response lies in the white region bounded by the design requirement line segments and thus meets the requirements.

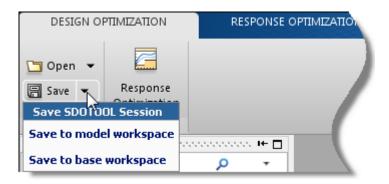
4 View the optimized parameter values. Click DesignVars in Design Optimization Workspace and view the updated values in the Variable Preview area.

The optimized values of the design variables are automatically updated in the Simulink model.

#### Save the Session

After you optimize the model response to meet design requirements, you can save the Design Optimization tool session which includes the optimized parameter values.

- 1 Click the **Design Optimization** tab of the Design Optimization tool.
- 2 In the Save drop-down list, select Save to model workspace.



A window opens where you specify the session name.

	×
Save to Simulink model workspace	
Existing Session Varia	bles
Session: sldo_model1_sdosessi	
	OK Cancel Help

**3** Specify a session name in the **Session** field.

Click OK.

**Tip** To open the saved session, click the **Open from model workspace** option in the **Open** drop-down list of the Design Optimization tool for the model.

# **Related Examples**

- "Design Optimization to Meet Step Response Requirements (Code)"
- "Design Optimization to Track Reference Signal (GUI)" on page 3-25  $\,$

## Design Optimization to Meet Step Response Requirements (Code)

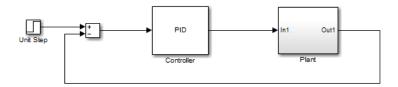
#### In this section...

"Model Structure" on page 3-18 "Design Requirements" on page 3-19 "Specify Step Response Requirements" on page 3-19 "Specify Design Variables" on page 3-20 "Optimize Model Response" on page 3-20

This example shows how to programmatically optimize controller parameters to meet step response requirements using sdo.optimize.

#### **Model Structure**

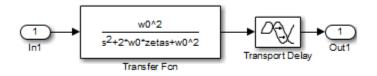
The model **sldo\_model1** includes the following blocks:



- **Controller block**, which is a PID controller. This block controls the output of the Plant subsystem.
- **Unit Step block** applies a step input and produces the model output that should meet step response requirements.

You can also use other types of inputs, such as ramp, to optimize the response to meet step response requirements generated by such inputs.

• **Plant subsystem** is a second-order system with delay. It contains "Transfer Function" and "Transport Delay" blocks.



#### **Design Requirements**

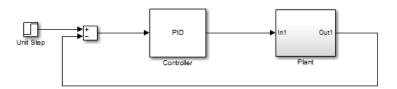
The plant output must meet the following step response requirements:

- Rise time less than 2.5 seconds
- Settling time less than 30 seconds
- Overshoot less than 5%

## Specify Step Response Requirements

**1** Open the Simulink model.

```
sys = 'sldo_model1';
open_system(sys);
```



2 Log the model output signal.

Design requirements require logged model signals. During optimization, the model is simulated using the current value of the model parameters and the logged signal is used to evaluate the design requirements.

```
PlantOutput = Simulink.SimulationData.SignalLoggingInfo;
PlantOutput.BlockPath = [sys '/Plant'];
PlantOutput.OutputPortIndex = 1;
PlantOutput.LoggingInfo.NameMode = 1;
PlantOutput.LoggingInfo.LoggingName = 'PlantOutput';
```

**3** Store the logging information.

```
simulator = sdo.SimulationTest(sys);
simulator.LoggingInfo.Signals = PlantOutput;
```

**simulator** is a **sdo.SimulationTest** object that you also use later to simulate the model.

**4** Specify step response requirements.

```
StepResp = sdo.requirements.StepResponseEnvelope;
StepResp.RiseTime = 2.5;
StepResp.SettlingTime = 30;
StepResp.PercentOvershoot = 5;
```

StepResp is a sdo.requirements.StepResponseEnvelope object.

#### **Specify Design Variables**

When you optimize the model response, the software modifies parameter (design variable) values to meet the design requirements.

**1** Select model parameters to optimize.

p = sdo.getParameterFromModel(sys,{'Kp','Ki','Kd'});

p is an array of 3 param.Continuous objects.

**2** To limit the parameters to positive values, set the minimum value of each parameter to **0**.

```
p(1).Minimum = 0;
p(2).Minimum = 0;
p(3).Minimum = 0;
```

#### **Optimize Model Response**

**1** Create a design function.

evalDesign = @(p) sldo\_model1\_design(p,simulator,StepResp);

evalDesign is an anonymous function that calls the cost function
sldo\_model1\_design. The cost function simulates the model and evaluates the
design requirements.

Tip Type edit sldo\_model1\_design to view this function.

- **2** (Optional) Evaluate the current response.
  - **a** Compute the model response using the current values of the design variables.

initDesign = evalDesign(p);

During simulation, the **Step Response** block throws assertion warnings at the MATLAB prompt which indicate that the requirements specified in the block are not satisfied.

**b** Examine the nonlinear inequality constraints.

#### initDesign.Cleq

ans = -0.2571 -0.4968 -0.5029 -1.0000 0.7060 0.5092 0.4964 201.4682

Some Cleq values are positive, beyond the specified tolerance, indicating that the response using the current parameter values violates the design requirements.

**3** Specify optimization options.

```
opt = sdo.OptimizeOptions;
opt.MethodOptions.Algorithm = 'active-set';
```

The software configures **opt** to use the default optimization method, fmincon. However, because the optimization problem specifies only constraints, you specify the Active Set algorithm for fmincon. The Active Set algorithm is good for solving feasibility (constraints only) optimization problems.

4 Optimize the response.

```
[pOpt,optInfo] = sdo.optimize(evalDesign,p,opt);
```

At each optimization iteration, the software simulates the model, and the default optimization solver fmincon modifies the design variables to meet the design requirements. For more information, see "Selecting Optimization Methods" and "How the Optimization Algorithm Formulates Minimization Problems".

After the optimization completes, the command window displays the following results:

				max	Step-size	First-order
Iter	F-count		f(x)	constraint		optimality
0	5		0	201.5		
1	12		0	167.5	0.687	0
2	19		0	0.8462	1.22	5.86
3	29		0	0.8169	0.133	0
4	37		0	0.7819	0.115	0
5	44		0	3.121	0.265	3.29
6	52		0	3.328	2.92	0
7	59		0	0.1413	1.26	2.81
8	66		0	0.02418	0.443	0.0029
9	73		0	8.125e-05	0.0563	1.03e-05
Local	minimum	found	that s	atisfies the	constraints.	

Optimization completed because the objective function is non-decreasing in feasible directions, to within the selected value of the function tolerance, and constraints are satisfied to within the selected value of the constraint tolerance.

The message Local minimum found that satisfies the constraints indicates that the optimization solver found a solution that meets the design requirements within specified tolerances. For more information about the outputs displayed during the optimization, see "Iterative Display" in the Optimization Toolbox documentation.

5 Examine the optimization termination information, contained in the optInfo output argument. This information helps you verify that the response meets the step response requirements.

For example, check the following fields:

• Cleq, which shows the optimized nonlinear inequality constraints.

#### optInfo.Cleq

ans = 0.0001 -0.0099 -0.0099 -1.0000 0.0004 -0.0100 -0.0101 -0.1997

All values satisfy  $Cleq \leq 0$ , within the optimization tolerances, which indicates that the step response requirements are satisfied.

• exitflag, which identifies why the optimization terminated.

The value is 1 which indicates that the solver found a solution which was less than the specified tolerances on the function value and constraint violations.

**6** View the optimized parameter values.

```
p0pt
```

```
pOpt(1,1) =
       Name: 'Kp'
      Value: 1.2364
   Minimum: 0
   Maximum: Inf
       Free: 1
      Scale: 1
       Info: [1x1 struct]
pOpt(2,1) =
       Name: 'Ki'
      Value: 0.3901
   Minimum: 0
   Maximum: Inf
       Free: 1
      Scale: 1
       Info: [1x1 struct]
pOpt(3,1) =
       Name: 'Kd'
      Value: 2.5937
   Minimum: 0
   Maximum: Inf
       Free: 1
      Scale: 1
       Info: [1x1 struct]
```

- 7 Simulate the model with the optimized values.
  - **a** Update optimized parameter values in the model.

```
sdo.setValueInModel(sys,pOpt);
```

**b** Simulate the model.

sim(sys);

## See Also

```
sdo.requirements.StepResponseEnvelope | param.Continuous |
sdo.getParameterFromModel | sdo.optimize | sdo.OptimizeOptions |
sdo.SimulationTest | Simulink.SimulationData.SignalLoggingInfo
```

## **Related Examples**

- "Design Optimization to Meet Step Response Requirements (GUI)" on page 3-4
- "Design Optimization to Track Reference Signal (GUI)" on page 3-25

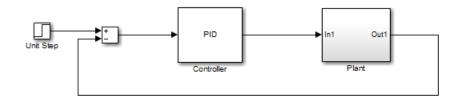
## **Design Optimization to Track Reference Signal (GUI)**

In this section	
"Model Structure" on page 3-25	
"Design Requirements" on page 3-26	
"Specify Reference Signal" on page 3-26	
"Specify Design Variables" on page 3-33	
"Optimize Model Response" on page 3-36	

This example shows how to optimize controller parameters to track a reference signal using the Design Optimization tool. You specify the reference signal without adding any Check blocks to the model.

## **Model Structure**

The model **sldo\_model1** includes the following blocks:



- **Controller block**, which is a PID controller. This block controls the output of the **Plant** subsystem.
- **Unit Step block** applies a step input and produces the model output that should meet step response requirements.

You can also use other types of inputs, such as ramp, to optimize the response to meet step response requirements generated by such inputs.

• **Plant subsystem** is a second-order system with delay. It contains "Transfer Function" and "Transport Delay" blocks.



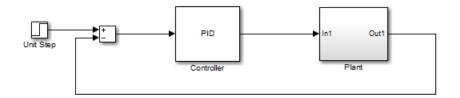
## **Design Requirements**

The model output must track a reference signal  $y = 1 - \exp(-0.1 \times t)$ , where *t* is time.

## **Specify Reference Signal**

**1** Open Simulink model.

```
sys = 'sldo_model1';
open_system(sys);
```



To learn more about the model, see "Model Structure" on page 3-25.

2 In the Simulink model window, select Analysis > Response Optimization.

A Design Optimization tool for the model opens.

DESIGN OPTI	MIZATION RES	PONSE OPTIMIZATION	VIEW		
	les Set: 😨 None ▾ ∠	New - Evalua Select Requirer	ate 🔽 No data selected 👻 Add Plot	Plot Current Response	
	ARIABLES	REQUIREMENTS	PLOTS	OPTIMIZATION	CLOSE
a 8 %.					00000
Search workspac		<del>ب</del> م			
<ul> <li>MATLAB World</li> </ul>	kspace				
Name 🔺	Value				
abc sys	'sldo_mod				
▼ Model Worksp	oace (sldo_model1)				
Name 🔺	Value				
Kd	0	<b>^</b>			
Ki Kp	0 1	E			
	0 5000	-			
<ul> <li>Design Optimi</li> </ul>	ization Workspace				
Name 🔺	Value				
▼ Variable Previe	2W				

- **3** Select the model signal which must track the reference signal.
  - **a** In the **New** drop-down list, select **Signal**.

- 🔤 N	lew 👻 🛄 Add Plot 👻
Nev	Requirement
	Signal Bound Use to specify a piecewise linear bound on a signal.
	Step Response Envelope Use to specify a step response envelope on a signal.
W	Signal Tracking Use to specify a tracking requirement on a signal.
ħ	Custom Requirement Use to create a custom requirement.
Nev	Requirement Using Model Blocks
<b>N</b>	Time Domain Check Blocks Use the Simulink Design Optimization library to add time domain check blocks to a Simulink model.
1	Frequency Domain Check Blocks Use the Simulink Control Design library to add frequency domain check blocks to a Simulink model.
Nev	/ Signal
-5	Signal Specify Simulink Signals to log so they can be used in design requirements.

A window opens where you select a model signal.

	×
Signal set: Sig	
No signals have currently been selected. Please go back to the model and click on a signal to select it.	Signal
	OK Cancel Help

 $\label{eq:bound} \boldsymbol{b} \quad \text{In the Simulink model window, click the output of the Plant block.}$ 

The window updates to display the selected signal.

	×
Signal set: Sig	
Currently selected signals	Signal
	Remove Signal
	OK Cancel Help

Select the signal and click to add it to the signal set.

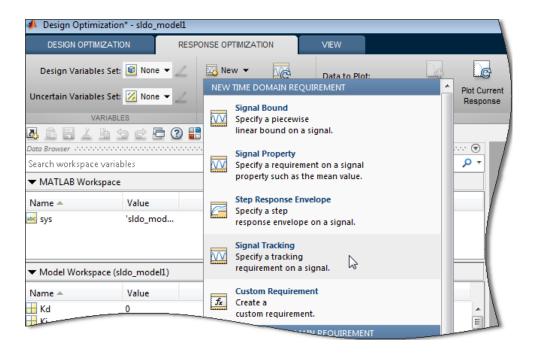
С

Signal set: Sig	×
Currently selected signals	Signal sldo model1/Plant:1
	Remove Signal
	OK Cancel Help

**d** In **Signal set**, enter **PlantOutput** as the selected signal name.

Click **OK**. A new variable **PlantOutput** appears in the **Design Optimization Workspace** of the Design Optimization tool.

- **4** Specify the reference signal that the model output must track.
  - a In the New drop-down list, select Signal Tracking.



A window opens where you specify the reference signal.

Create Requirement	x
	Signal Tracking
	Specify a tracking requirement on a signal.
Name: SignalTrac	
• Specify Refere	
Time vector.	[0;1;2;3;4;5;6;7;8;9;10]
Amplitude:	[0;0.632120558828558;0.864664716763387;0.950212931632136;0.98168436111:
	Update reference signal data
Tracking Method: ▼ Specify Signal	SSE 🔹
	Signal
PlantOut	put (sldo_model1/Plant:1) +
<table-cell> Create Plot</table-cell>	OK Cancel Help

- **b** In the **Name** edit box, enter ref\_sig.
- c In the Time vector edit box, enter linspace(0,50,200)
- d In the Amplitude edit box, enter 1-exp(-0.1\*linspace(0,50,200)).

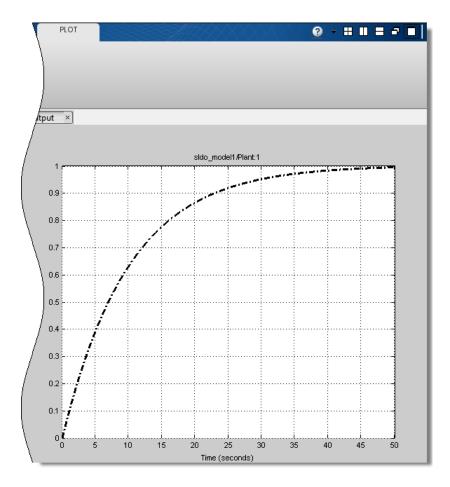
Create Requiremen	t	x
	Signal Tracking	
	Specify a tracking requirement on a signal.	
Name: ref_siq  Specify Refer	ence Signal	
Time vector:	linspace(0,50,200)	]
Amplitude:	1-exp(-0.1*linspace(0,50,200))	]
	Update reference signal data	]

The **Tracking Method** is **SSE**, which means that at each optimization iteration, the solver attempts to reduce the sum of squared errors between the simulated output and reference signal.

- e Click Update reference signal data.
- **f** Select the check-box corresponding to the signal you selected in the previous step in the **Specify Signal to Track Reference Signal** area.

Tracking Method: SSE 🔻	
<ul> <li>Specify Signal to Track Reference Signal</li> </ul>	
Signal	
PlantOutput (sldo_model1/Plant:1)	

Click **OK**. A new variable ref\_sig appears in the **Design Optimization Workspace** and the Design Optimization window updates to plot the reference signal.



## **Specify Design Variables**

Before you begin this task, specify the reference signal to track as described in "Specify Reference Signal" on page 3-26.

When you optimize the model response, the software modifies the design variable (model parameter) values to meet the design requirements.

In the **Response Optimization** tab:

1 Select New in the Design Variables Set drop-down list.



A window opens where you specify design variables.

reate Design Variables se		Min Max	Scale	Variable	Current value	Used By
				Kd	0	sldo model1/Cont
				Ki	0	sldo model1/Conti
				Кр	1	sido model1/Control
				w0	0.5	
	Undate more	lel variable va	luer	zeta	0.5	sldo model1/Plant/Tr
Variable Detail	opuse mee			Specify expression	ession (e.q., s.x or a(3))	Cancel Hei,

- 2 Click Kd, Ki and Kp to select them.
- 3

Click

to add the selected parameters to a design variables set.

Create D	esign Variables set: DesignVars				
<b>V</b>	Variable	Value	Minimum	Maximum	Scale
1	Kd	0	-Inf	Inf	1
1	Ki	0	-Inf	Inf	1
1	Кр	1	-Inf	Inf	1

The software displays the following parameter settings:

• Variable — Parameter name

- Value Current parameter value
- Minimum and Maximum Parameter bounds
- Scale Scaling factor for the parameter

The check-box indicates that the parameter is included in the design variable set. The default design variable set name is **DesignVars**.

**4** To limit the parameters to positive values, enter the minimum value of each parameter as **0** in the corresponding **Minimum** field and press **Enter**.

Create D	esign Variables set: DesignVars				
<b>V</b>	Variable	Value	Minimum	Maximum	Scale
V	Kd	0	0	Inf	1
<b>V</b>	Ki	0	0	Inf	1
<b>V</b>	Кр	1	0	Inf	1

Click **OK**. A new variable **DesignVars** appears in **Design Optimization Workspace** of the Design Optimization tool. You can click the variable to view its contents in the **Variable Preview** area.

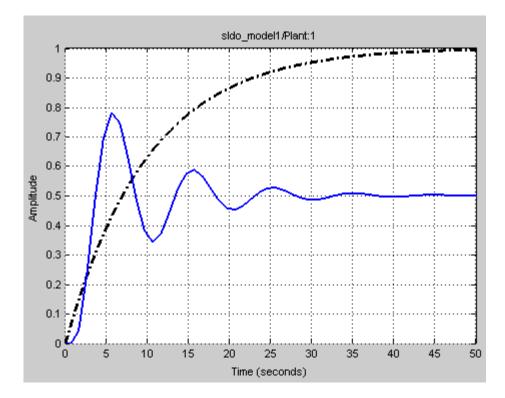
<ul> <li>Design Optimization Workspace</li> </ul>				
Name 🔺	Name 🔺			
DesignVars	<3x1 para			
▼ Variable Previ	ew			
DesignVars(1	,1) =			
Name:	'Kd'			
Value:	0			
Minimum:	0			
Maximum:	Inf			
Free:	1			
Scale:	1			
Info:	[1x1	struct]		

## **Optimize Model Response**

Before you begin this task, you must have specified the reference signal to track and design variables, as described in "Specify Reference Signal" on page 3-26 and "Specify Design Variables" on page 3-33, respectively.

1

(Optional) View the current model response. Click



The plot shows that the response does not track the reference signal.

2 Click Optimize.



An optimization progress window opens.

Tip	To view the model response and optimization progress windows simultaneously,
tile	them using the plot layout area 🖽 💷 🚍 🖃 🔲

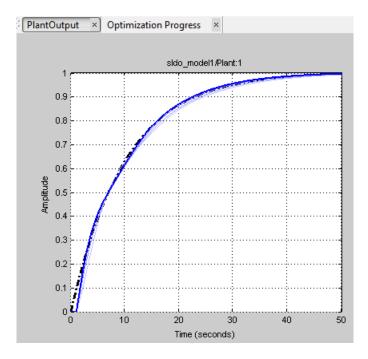
At each iteration, the optimization solver Gradient descent (fmincon) modifies the controller parameters to minimize the error between the simulated response and the reference signal. To learn more, see "How the Optimization Algorithm Formulates Minimization Problems".

After the optimization completes, the optimization progress window resembles the following figure.

Iteration	F-count	ref_sig				
		(min)				
0	5		30.19 🔺			
1	34		24.66			
2	43		21.39			
3	51		1.65 ≘			
4	62		1.45			
5	71		0.57			
6	82		0.51			
7	90		0.36			
8	98		0.31			
9	106		0.31			
10	114		0.31			
11	122		0.31 +			
<		III	•			
Optimization started 08-Jan-2013 09:48:55 Optimization converged, 08-Jan-2013 09:49:34 Optimized variable values written to 'DesignVars' in the Design Optimization workspace						

The message Optimization converged indicates that the optimization method found a solution that tracks the reference signal within the tolerances and parameter bounds. For more information about the outputs displayed in the optimization progress window, see "Iterative Display" in the Optimization Toolbox documentation.

**3** Verify that the response tracks the reference signal.



The optimized response closely tracks the reference signal.

4 View the optimized parameter values. Click DesignVars in Design Optimization Workspace and view the updated values in the Variable Preview area.

The optimized values of the design variables are automatically updated in the Simulink model.

## **Related Examples**

- "Design Optimization to Meet Step Response Requirements (GUI)" on page 3-4
- "Design Optimization to Meet Step Response Requirements (Code)" on page 3-18

# **Design Optimization Using Frequency-Domain Check Blocks (GUI)**

In this section
-----------------

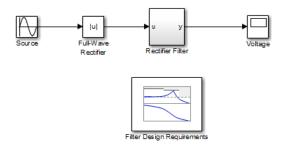
"Model Structure" on page 3-40 "Design Requirements" on page 3-41 "Specify Design Requirements" on page 3-41 "Specify Design Variables" on page 3-44 "Optimize Design" on page 3-46

This example shows how to optimize a design to meet frequency-domain requirements using the Design Optimization tool. Simulink Control Design software must be installed to optimize a design to meet frequency-domain design requirements.

In this example, you specify the design requirements in a Check Bode Characteristics block and optimize a rectifier filter parameters to meet gain and bandwidth requirements, and minimize a custom objective.

## **Model Structure**

The model sdorectifier includes the following blocks:



- Full-Wave Rectifier block An Abs block
- + Rectifier Filter subsystem RLC filter implemented using integrator and gain blocks
- **Filter Design Requirements block** Check Bode Characteristics block that specifies the gain and bandwidth design requirements

#### **Design Requirements**

The design optimization problem is multi-objective. The design must:

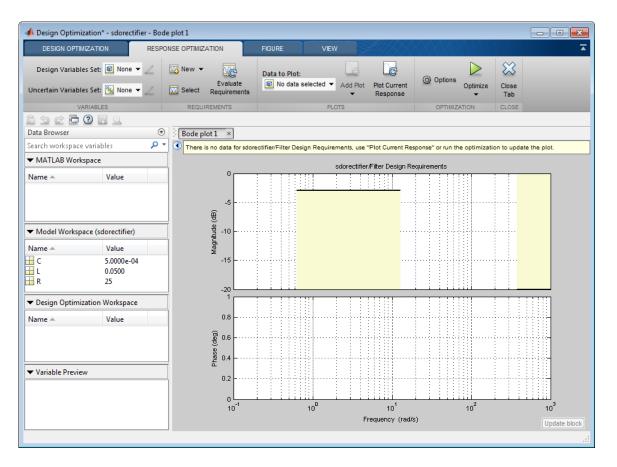
- Have a -3db bandwidth of at least 2Hz
- · Limit the gain across the frequency range 2Hz—60Hz to at most 0db
- Limit the gain above 60Hz to at most –20db
- Maximize the filter resistance R
- Minimize the filter inductance L

The requirements ensure that the rectifier filter combination has minimal high frequency content, responds quickly to voltage changes, and limits filter currents.

## **Specify Design Requirements**

**1** Open the Design Optimization tool for the model.

```
sdotool('sdorectifier')
```



The plot shows the gain and bandwidth requirements specified in the Filter **Design Requirements** block. See their values in the **Bounds** tab of the Block Parameters dialog box.

**2** Specify a custom objective to minimize the filter inductance and maximize the resistance.

The custom objective is specified in the sdorectifier\_cost function. The function accepts the design variables R and L, and returns the objective to be minimized.

Tip Type edit sdorectifier\_cost to view this function.

- 🚾 Select 🔂 PlotCurrent Response 0 🚾 New 👻 🛃 Add Plot 👻 New Requirement Signal Bound Use to specify a piecewise linear bound on a signal. Step Response Envelope Use to specify a step response envelope on a signal. Signal Tracking W Use to specify a tracking requirement on a signal. **Custom Requirement** 2 ÷. Use to create a custom requirement.
- a Select Custom Requirement in the New drop-down list.

A window opens where you specify the custom requirement.

	Custom Requirement				
Use to create a custom requirement. The optimizer will call the specified function handle during optimization passing a structure with fields containing the current design variable values and the selected simulation signal values.					
Requirement Name: Req					
Requirement function:	@myCustomRequirement Open				
Requirement type:	<= •				

- **b** Specify the following:
  - Enter MaxMinRL in Requirement Name
  - Enter @sdorectifier\_cost in **Requirement function**. The optimization solver calls the specified function handle.
  - Select min in Requirement type

	Custom Requirement
during	m requirement. The optimizer will call the specified function handle optimization passing a structure with fields containing sign variable values and the selected simulation signal values.
Requirement Name: Ma	×MinRL
Requirement function:	sdorectifier_cost
Requirement type:	min -
	Select Model Signals
	Signal
	Signal

Click **OK**. A new variable MaxMinRL appears in **Design Optimization Workspace** of the Design Optimization tool.

## **Specify Design Variables**

Before you begin this task, specify the design requirements as described in "Specify Design Requirements" on page 3-41.

When you optimize the model response, the software modifies the design variable (parameter) values to meet the design requirements.

1 Select New in the Design Variables Set drop-down list.



A window opens where you specify design variables.

				x
Create Design Variables set: De	esignVars			
Variable	Value Min Max Scale	Variable	Current value	Used By
		С	0.0005	sdorectifier/Rectifier
		L	0.05	sdorectifier/Rectifier
		R	25	sdorectifier/Rectifier
	Update model variable values			
Variable Detail		Specify expression	(e.g., s.x or a(3))	
			ОК	Cancel Help
				.:

 $\label{eq:click R, L and C to select them.} 2 \qquad \text{Click R, L and C to select them.}$ 

3

Click to add the selected parameters to a design variables set.

- **4** Specify the value range for each design variable:
  - R in the range 1–50 ohms
  - L in the range 1–500mH
  - **C** in the range  $1\mu$ F-1mF
  - Because the variables are different orders of magnitude, specify scaling factors in the corresponding **Scale** column of the variables:
    - R by 25
    - L by 0.05
    - C by 0.0005

Create Design Variables set: DesignVars

<b>V</b>	Variable	Value	Minimum	Maximum	Scale
<b>V</b>	С	0.0005	1e-6	1e-3	0.0005
<b>V</b>	L	0.05	1e-3	500e-3	0.5
<b>V</b>	R	25	1	50	25

Click **OK**. A new variable **DesignVars** appears in **Design Optimization Workspace** of the Design Optimization tool.

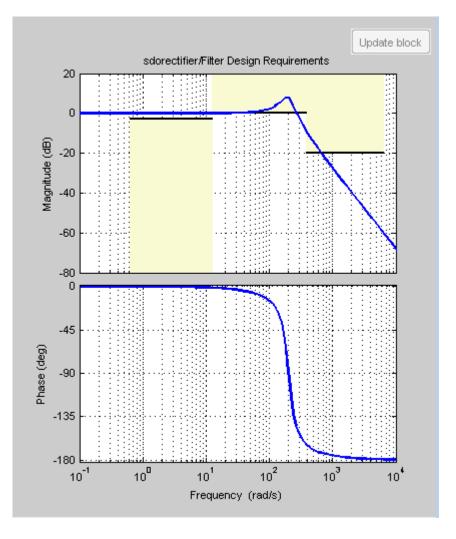
## **Optimize Design**

Before you begin this task, you must have already specified the design requirements and design variables, as described in "Specify Design Requirements" on page 3-41 and "Specify Design Variables" on page 3-44, respectively.

1

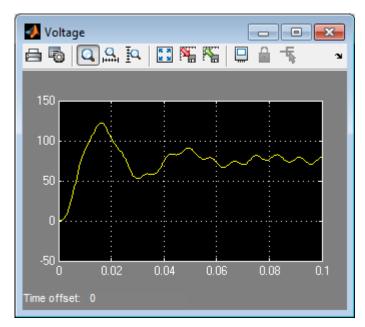
(Optional) View the current response of the model. Click

3-46

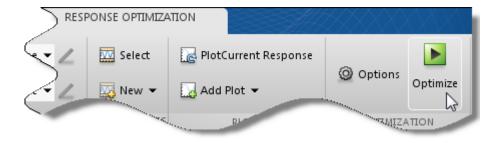


The plot shows that the model does not meet the specified gain and bandwidth requirements.

You also see that the filter voltage signal overshoots its steady-state value and contains significant harmonic content in the Voltage scope window.



2 Click Optimize.

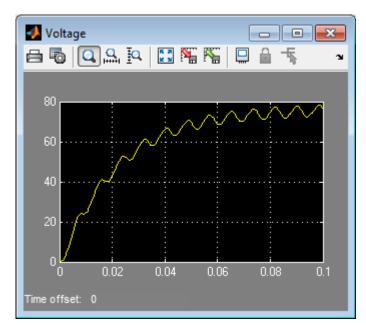


An optimization progress window opens.

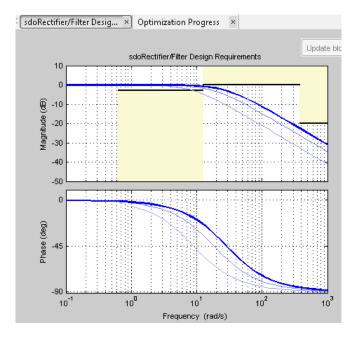
After the optimization completes, the optimization progress window resembles the next figure. After a few iterations, the optimization converges to satisfy the filter bandwidth requirements.

lteration	F-count	MaxMinRL (min)	Filter Design Requireme (<=0)	Filter Design Requireme (>=0)		
0	8	1,1000		1.0067		
1	16	0.3519	-0.8514	-0.8115		
2	23	0.0420	0.9941	1.0000		
3	31	0.0688	0.2360	0.9654		
4	39	0.0794	0.0451	0.9415		
5	46	0.0840	0.0056	0.9342		
6	53	0.0840	0.0056	0.9342		
Optimization started 08-Jan-2013 10:07:11 Optimization converged, 08-Jan-2013 10:07:48 Optimized variable values written to 'DesignVars' in the Design Optimization workspace						

The filter voltage signal in the Voltage scope window also settles to within its steady-state value in around 0.08 seconds without any overshoot and the harmonic content is significantly reduced from the initial design.



**3** Verify that the model meets the gain and bandwidth requirements.



The plot displays the last five iterations. The final response using the optimized parameter values appears as the thick line.

The optimized response lies in the white region bounded by the design requirement line segments and thus meets the requirements.

4 Click DesignVars in Design Optimization Workspace and view the updated values in the Variable Preview area.

The optimized values of the design variables are automatically updated in the Simulink model.

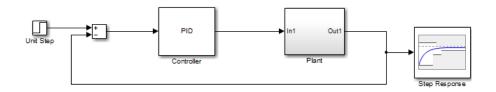
## **Time-Domain Model Verification**

This example shows how to perform time-domain model verification using Simulink Design Optimization **Model Verification** blocks. During time-domain verification, the software monitors a signal to check if it meets time-domain characteristics such as step response characteristics and upper and lower amplitudes, or tracks a reference signal.

You can also use blocks from Simulink and Simulink Control Design **Model Verification** libraries to design complex assertion logic for time- and frequency-domain verification, and signal monitoring. If you have Simulink Verification and Validation software, you can construct simulation tests for your model using the Verification Manager tool in the Signal Builder.

1 Open Simulink model.

```
sys = 'sldo_model1_stepblk';
open_system(sys);
```



The model includes a Step Response block which is a Check Step Response Characteristics block from the Simulink Design Optimization **Model Verification** library and has default step response bounds.

Sink Block Parameters: Check Step Response Characteristics							
Check Step Response Characteristics							
Assert that the input signal satisfies bounds specified by step response characteristics.							
Bounds Assertion							
✓ Include step response	bound in assertion	_					
Step time (seconds):	0						
Initial value:	0	Final value:	1				
Rise time (seconds):	5	% Rise:	80				
Settling time (seconds):	7	% Settling:	1				
% Overshoot:	10	% Undershoot:	1				
Enable zero-crossing detection							
Show Plot Show plot on block open Response Optimization							
OK Cancel Help Apply							

2 In the Simulink Editor, click **Simulation** > **Run**.

The block asserts multiple times during simulation because the signal to which the block is connected violates the specified bounds. Assertion warnings appear in the MATLAB command window.

You can optimize model parameters to satisfy the bounds and eliminate assertion warnings. See "Design Optimization to Meet Step Response Requirements (GUI)" on page 3-4.

# Optimization-Based Linear Control Design

- "When to Use Optimization-Based Linear Control Design" on page 4-2
- "Types of Time- and Frequency-Domain Design Requirements for Optimization-Based Control Design" on page 4-3
- "Quick Start Optimization-Based Linear Control Design" on page 4-4
- "Design Optimization-Based PID Controller for Linearized Simulink Model (GUI)" on page 4-15

## When to Use Optimization-Based Linear Control Design

When you have Control System Toolbox software installed, you can design and optimize control systems for LTI models by optimizing controller parameters in the SISO Design Tool. To use optimization methods for linear control design, also known as *optimization-based tuning*, you must already have an initial controller. You can then use optimization-based tuning to refine the controller design to meet additional design requirements. For more information on designing controllers, see the Control System Toolbox documentation.

**Note:** Optimization-based tuning only changes the value of the controller parameters and not the controller structure itself.

Optimization-based tuning provides flexibility in terms of specifying additional design requirements for the controller. When you have a large number of design requirements, you can first design an initial controller by selecting a subset of requirements and subsequently select additional requirements to refine the design.

Optimization-based tuning also provides flexibility in terms of selecting a subset of controller parameters to optimize, and specifying bounds on the controller parameters.

To design linear controllers for Simulink models using optimization-based tuning, you must first linearize the model using the Simulink Control Design software. For more information on linearizing Simulink models, see the Simulink Control Design documentation.

## Types of Time- and Frequency-Domain Design Requirements for Optimization-Based Control Design

When you design linear controllers for LTI or Simulink models using the Simulink Design Optimization software, you can specify both time- and frequency-domain requirements on the system response. You can specify design requirements on the following plots:

- Root Locus plot
- Open-Loop and Prefilter Bode plots
- Open-Loop Nichols plot
- Step/Impulse Response plots

For more information, see "Time- and Frequency-Domain Requirements in SISO Design Tool".

Simulink Design Optimization software uses the frequency-domain requirements to compute the frequency response of the system. It then uses optimization methods to reduce the distance between the current response and the requirements by modifying the controller parameters. The software does not change the controller structure when optimizing the controller parameters.

# Quick Start — Optimization-Based Linear Control Design

In this quick start, you get an overview of the typical tasks for optimization-based linear control design using the SISO Design Tool:

- 1 Open a SISO Design Tool session.
- 2 Configure a project for optimization-based control design.
- **3** Specify the controller parameters to design.
- **4** Specify the design requirements.
- **5** Design the controller.
- **6** Evaluate the controller design.

**Note:** The same workflow applies to optimization-based control design for LTI models created at the command line using Control System Toolbox software. To learn how to create LTI models, see "Linear (LTI) Models" in the Control System Toolbox documentation.

Prerequisites for optimization-based linear control design include:

• Simulink Compensator Design Task that contains a linearized version of the Simulink model and, optionally, any response plots you configure.

For more information on how to linearize a Simulink model for control design, see the Simulink Control Design documentation.

· Time- and frequency-domain design requirements

To design a controller using optimization methods:

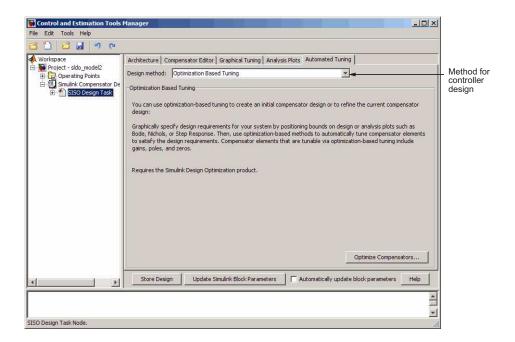
**1** Open a SISO Design Tool session by typing the following command at the MATLAB prompt:

```
sisotool('projectname.mat')
```

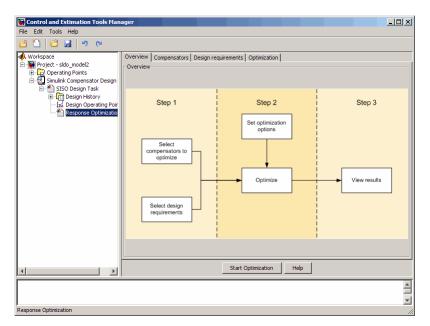
	Control and Estimation Tools I File Help	Manager		
SISO Design Task for SISO Design — Tool		Project Settings Title : Subject : Created by: Date modified : Simulink model : Description :	Project - sido_model2	
	Project node.			

The command also opens a SISO Design for SISO Design Task window by default and any response plots you configured when you linearized the Simulink model using Simulink Control Design software.

2 Configure a project for optimization-based control design by clicking **Optimize Compensators** in the Automated Tuning tab of the **SISO Design Task**.



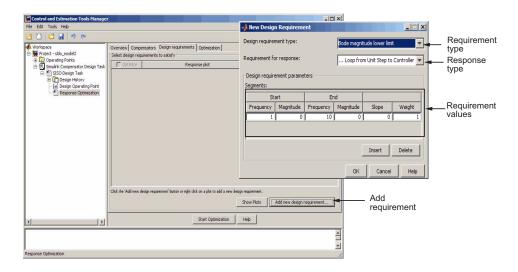
This action creates a new **Response Optimization** node in the Control and Estimation Tools Manager.



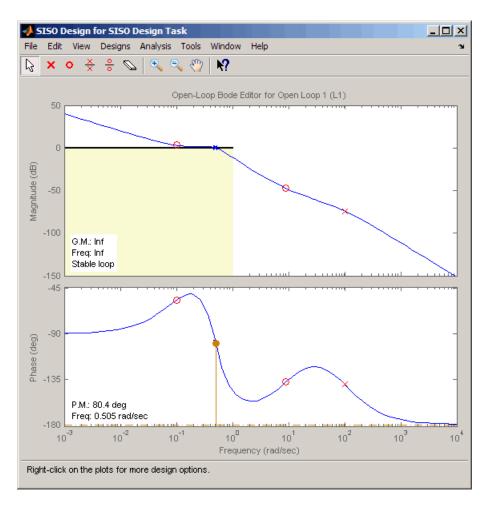
**3** Specify the controller parameters to design in the **Compensators** tab.

ontrol and Estimation	Tools Manager						
Edit Tools Help							
	(Pi						
/orkspace	Overview Con	npensators Design requirements	Optimization				
Project - sldo_model2		sator elements to optimize	[ openingdon ]	(			
Derating Points	C Optimize	Compensator elements	Value	Initial guess	Minimum	Maximum	Typical valu
E-M SISO Design	Г	sldo_model2/Controller					
🕀 🛄 Design Hi		Gain	0,1	0,1	-InF	Inf	0.1
🛛 🛃 Design O		Real Zero	-8.999	-8.999	-Inf	Inf	-8.999
		Real Zero	-0.10102	-0.10102	-Inf	Inf	-0.10102
						1.041.0	
		Real Pole	-100	-100	-Inf	Inf	-100
	I	Real Pole	-100			∏nf	-100

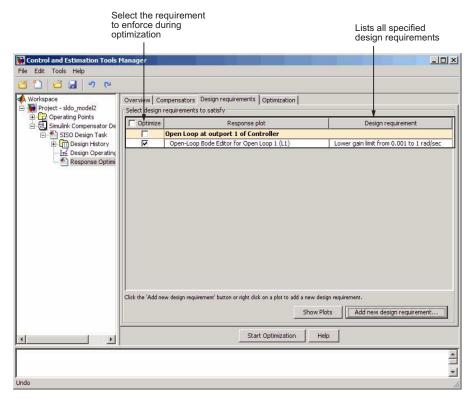
- **4** Specify the design requirements.
  - **a** In the **Design requirements** tab, click **Add new design requirement**. Specify the design requirements, for example Bode magnitude lower limit, in the New Design Requirement dialog box.



In the SISO Design window, the yellow region with the black line segment represents the design requirement on the response plot.



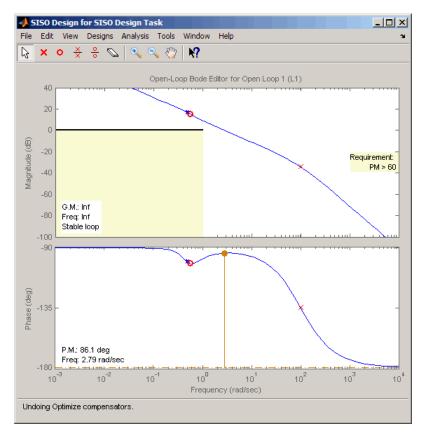
The **Design Requirements** tab also lists the design requirement.



- **b** Repeat step a to specify additional time- and frequency-domain requirements.
- **5** Start the optimization to design the controller by clicking **Start Optimization** in the **Optimization** tab.

Control and Estimation Too File Edit Tools Help	ols Manager			<u></u>				>	1
	í								-
Workspace	Overview Co		esign requiren	nents Optimiza	tion				_
Project - sldo_model2 Operating Points	Optimization p	rogress							
E E Simulink Compensator	Iteration	Eval-Count	Cost function	Constraint	Step size	Procedure	T	Optimization options	
SISO Design Task	0	7	0	57.12					
🕀 🛄 Design History	1	14	0	1.777	8.36	-		Display options	
- 🛃 Design Opera	2	21	0	1.697	12.1	Hessian m			
Response Opl	3	28	0	0	21.7				
								-	Optimization
									progress
							*		
	Constructing	optimization p	roblem						
		started 10-Nov							
		finished 10-No		1 r optimal solutio	n within the	manified			
	tolerances.	miniation. Pou	nu a leasible u	r opumai soluti	in wann ure :	specified			Optimization
									status
	0						*		
	1								
al				Start Optimi	zation	Help			
	J		7						
Redo									11

- **6** Evaluate the controller design.
  - **a** Examine the system's response in the response plot, for example the Bode plot, to see if it meets the requirements. The system's response must lie in the white region in order to meet the design requirement.



**b** Examine the controller parameter values in the **Compensator** tab.

Workspace	Overview C	ompensators Design requirements Op	timization				
Project - sldo_model2		ensator elements to optimize	· · · ·				
Operating Points     Simulink Compensator De	C Optimiz	Compensator elements	Value	Initial guess	Minimum	Maximum	Typical v.
E SISO Design Task		sldo_model2/Controller		1			
🕀 🛄 Design History	1	P	6.7627	1	-Inf	Inf	1
📈 🔂 Design Operating		1	3.4445	0.1	-Inf	Inf	1
🐴 Response Optimi	~	D	11.106	0.1	-Inf	Inf	1
		N	100	100	-Inf	Inf	15
	Right click on a	compensator name to change its representation	h		Usa	s Value as Int	tial Guess

Optimized controller parameter values

7 Write the controller parameter values into the Simulink model. To do so, clickUpdate Simulink Block Parameters in the SISO Design Task node.

**See Also:** "Design Optimization-Based PID Controller for Linearized Simulink Model (GUI)".

## Design Optimization-Based PID Controller for Linearized Simulink Model (GUI)

#### In this section...

"About This Tutorial" on page 4-15

"Configuring a Project for Optimization-Based Control Design" on page 4-16

"Designing an Initial PID Controller to Meet Bode Magnitude and Phase Margins Requirements" on page 4-22

"Refining the Controller Design to Meet Controller Output Bounds" on page 4-42

"Saving the Project" on page 4-57

### **About This Tutorial**

- "Objectives" on page 4-15
- "About the Model" on page 4-15
- "Design Requirements" on page 4-16

#### **Objectives**

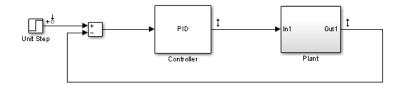
In this tutorial, you learn to use optimization methods to design a PID controller to meet frequency-domain design requirements on a system's response.

You accomplish the following tasks using the GUI:

- Specify frequency-domain Bode magnitude and phase margin requirements.
- Design an initial controller to meet the frequency-domain requirements.
- Refine the initial controller design to limit the controller's output signal.

#### About the Model

In this tutorial, you use the Simulink model named  $\verb"sldo_model2"$ , as shown in the next figure.



The model contains a **Controller** block, which is a PID Controller. This block controls the output of the **Plant** subsystem.

Using the Simulink Control Design software has linearized the Simulink model at the operating point specified in the model. The sldo\_model2.mat file contains a preconfigured SISO Design Tool session saved after linearizing the model. To learn more about linearizing Simulink models for control design, see "Control Design".

Double-click the Plant subsystem to open it. The plant is modeled as a second-order system with delay. It contains Transfer Function and Transport Delay blocks, as shown in the next figure.



To learn more about the blocks, see "Transfer Fcn" and "Transport Delay" block reference pages.

#### **Design Requirements**

The compensator you design in this tutorial must meet the following design requirements:

- Bode lower magnitude bound of 0 in the frequency range 1e-3 to 1 rad/sec
- Phase margin greater than 60 degrees
- Controller output bounds in the range [-250 550]

## Configuring a Project for Optimization-Based Control Design

To design a linear controller for a Simulink model, you must first configure a Control and Estimation Tools Manager project.

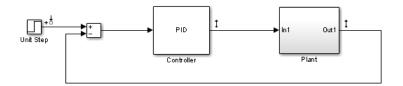
**1** Open a SISO Design Tool session for the linearized Simulink model by typing the following command at the MATLAB prompt:

```
sisotool('sldo_model2.mat')
```

**Note:** sldo\_model2.mat file contains a preconfigured SISO Design Tool session. This session was saved after Simulink Control Design software linearized the sldo\_model2 Simulink model.

This command opens the following windows:

• Simulink model

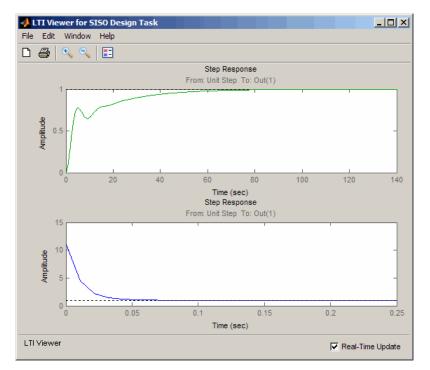


To learn more about the model, see "About the Model" on page 4-15.

• Control and Estimation Tools Manager GUI, which contains a SISO Design Task node under the Simulink Compensator Design Task node.

🙀 Control and Estimation Tools	Manager	
File Help		
🖆 🗋 🖨 🛃		
Workspace	Project Settings	
Project - sldo_model2     Operating Points	Title:	Project - sldo_model2
🗄 🗐 Simulink Compensator De	Subject:	
	Created by:	
	Date modified:	10-Nov-2008
	Simulink model:	sldo_model2
	Description:	
₹F		
		<u> </u>
		A
Project node.		

- LTI Viewer for SISO Design Task window, which contains the following plots:
  - Closed-loop step response of the system in the top plot
  - Output of the Controller block in the bottom plot



A blank SISO Design for SISO Design Task window

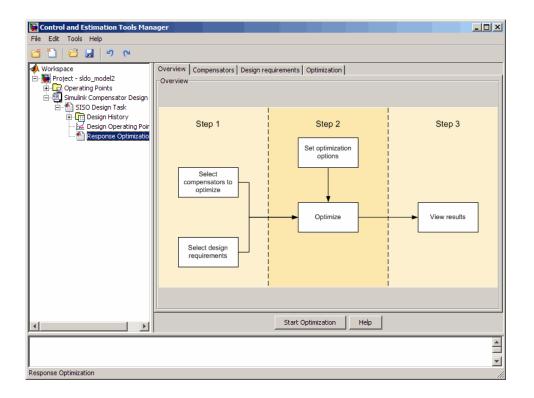
	150														
											Help				
8	×	0	×	÷	Ø	.   @	<b>)</b> (	₹	1	?					
	qht-c														

2 In the Control and Estimation Tools Manager GUI, select the **Automated Tuning** tab in the **SISO Design Task** node.

Control and Estimation Tools	Manager
File Edit Tools Help	
🗃 🛅 🖨 🖬 🤊 🍋	
Workspace     Project - sido_model2     Sido_model2     Sido_model2     Sido_model2     Project - Sido_model2     Sido_model2     Project - Sido_model2     Project - Sido_model2     Sido_model2     Project - Sido_model2     Project - Sido_model2     Sido_model2     Sido_model2     Sido_model2     Project - Sido_model2     Sido_model     Sido_model     Sido_model2     Sido_mod	Architecture       Compensator Editor       Graphical Tuning       Automated Tuning         Design method:       Optimization Based Tuning       Image: Compensator Based Tuning         Optimization Based Tuning       Image: Compensator Based Tuning       Image: Compensator Based Tuning         You can use optimization-based tuning to create an initial compensator design or to refine the current compensator design:       Graphically specify design requirements for your system by positioning bounds on design or analysis plots such as Bode, Nichols, or Step Response. Then, use optimization-based methods to automatically tune compensator elements to satisfy the design requirements. Compensator elements that are tunable via optimization-based tuning include gains, poles, and zeros.         Requires the Simulink Design Optimization product.       Optimize Compensators
	Store Design Update Simulink Block Parameters Automatically update block parameters Help
	×
SISO Design Task Node.	

### **3** Click **Optimize Compensators**.

This action creates a new **Response Optimization** node.



### Designing an Initial PID Controller to Meet Bode Magnitude and Phase Margins Requirements

- "Specifying the Controller Parameters" on page 4-22
- "Specifying Bode Magnitude and Phase Margin Design Requirements" on page 4-26
- "Designing the Controller" on page 4-36

#### **Specifying the Controller Parameters**

In this portion of the tutorial, you specify the controller parameters to design.

You must have already configured a project for control design, as described in "Configuring a Project for Optimization-Based Control Design" on page 4-16.

To specify the controller parameters to design:

1 In the Control and Estimation Tools Manager GUI, select the **Compensators** tab in the **Response Optimization** node.

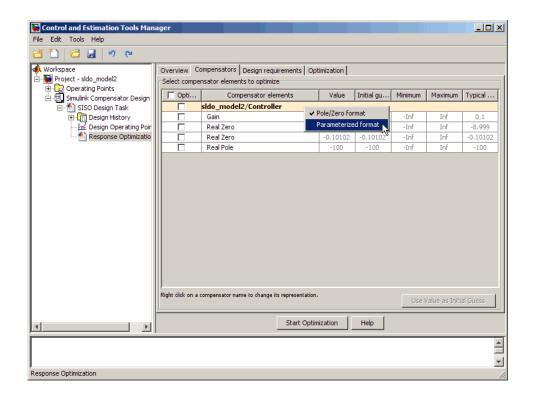
📲 Project - sldo_model2		mpensators Design requirements Isator elements to optimize	Optimization				
Operating Points     Simulink Compens	Optimize	Compensator elements	Value	Initial guess	Minimum	Maximum	Typical value
E SISO Design		sldo model2/Controller					
🕀 🔐 Design Hi		Gain	0.1	0.1	-Inf	Inf	0.1
🛃 Design O		Real Zero	-8.999	-8.999	-Inf	Inf	-8.999
🚹 Response		Real Zero	-0.10102	-0.10102	-Inf	Inf	-0.10102
		Real Pole	-100	-100	-Inf	Inf	-100
	Right click on a c	ompensator name to change its represent	ation.				
	Right click on a c		ation.	n ( Help		Use Value a	s Initial Guess

The controller parameters appear as poles and zeros in the **Compensator elements** column and represent the following:

- Gain Overall gain of the controller
- Real zeros Zeros resulting from the differentiator and integrator
- \* Real pole Pole resulting from the low-pass filter of the differentiator

**Tip** To view the structure of the **Controller** block, right-click the block and select **Mask** > **Look Under Mask**.

2 For convenience, change the PID controller parameters to Simulink block mask parameters format. To do so, right-click the **sldo\_model2/Controller** column, and select **Parameterized format**.



This action displays the controller parameters as Simulink block mask parameters P, I, and D, as shown in the next figure. To learn more about mask parameters, see "Mask Parameters" in the Simulink documentation.

≝ <u>`</u>   <b>C</b> ⊟   ") ("		1					
♦ Workspace ∃- 🙀 Project - sldo_model2	Overview Compensat	ors Design requirements Optimi	zation				
🛨 📴 Operating Points	Optimize	Compensator elements	Value	Initial guess	Minimum	Maximum	Typical value
🖃 🗐 Simulink Compensator Design Task		model2/Controller	value	Initial guess	Minimum	Maximum	Typical value
		model2/controller	1	1	-Inf	Inf	1
Design Firstory			0.1	0.1	-Inf	Inf	1
Response Optimization			0.1	0.1	-Inf	Inf	1
<u> </u>			100	100	-Inf	Inf	1
	Right click on a compensate	or name to change its representation.				Jse Value as I	nitial Guess
[]>	Right click on a compensat		Dptimization	Help		Jse Value as I	nitial Guess

The **Compensators** tab displays the following parameter settings:

- Value Current controller parameter value
- Initial Guess Initial controller parameter value
- Minimum and Maximum Controller parameter bounds
- **Typical Value** Scaling factor for the controller parameter

**Note:** Compensator elements or parameters cannot have uncertainty when used with frequency-domain based response optimization.

**3** In the **Optimize** column, select the check boxes for P, I, and D.

Control and Estimation Tools Manager File Edit Tools Help	r						
ජ 🛅 🖾 🖬 🔊 ര							
Norkspace	Overview Comp	oensators Design requirements Optim	ization				
Project - sldo_model2     Operating Points	Select compensa	tor elements to optimize					
E E Simulink Compensator Design Task	Coptimize	Compensator elements	Value	Initial guess	Minimum	Maximum	Typical value
🖃 🎦 SISO Design Task		sldo_model2/Controller					
Design History	<b>V</b>	P	1	1	-Inf	Inf	1
		I	0.1	0.1	-Inf	Inf	1
Response Optimization		D N	0.1	0.1	-Inf -Inf	Inf	1
		N	100	100	-101	Int	1
	Right click on a com	pensitor name to change its representation.				Inc Volum on T	ažist Guess
	Right click on a com	pensator name to change its representation.				Jse Value as I	nitial Guess
())	Right click on a com		Optimization	Help		Jse Value as I	nitial Guess
	Right click on a com		Optimization	Help		Use Value as I	nitial Guess

#### Specifying Bode Magnitude and Phase Margin Design Requirements

In this portion of the tutorial, you specify the Bode magnitude and phase margin requirements that the controller must satisfy.

Before you specify the design requirements, you can specify the controller parameters to design, as described in "Specifying the Controller Parameters" on page 4-22.

To specify the Bode design requirements:

**1** In the **Response Optimization** node of the Control and Estimation Tools Manager GUI, select the **Design requirements** tab.

Control and Estimation Tools Manage	r					
File Edit Tools Help						
🖆 🗋 🗳 📓 🔍 🍽						
Workspace	Overview Comp	ensators Design requiremen	ts Optimization			
Project - sldo_model2     Operating Points	Select design req	uirements to satisfy				
Simulink Compensator Design Task	C Optimize	Re	sponse plot		Design req	uirement
🗄 🐁 SISO Design Task						
Design History						
Response Optimization						
	Click the 'Add new o	design requirement' button or right	click on a plot to add a new	v design requirement		
				Show Plots	Add pew des	ign requirement
				510111005		
			Start Optimization	Help		
						<b>A</b>
						-
Response Optimization						

2 Click Add new design requirement.

This action opens the New Design Requirement dialog box.

📣 New Design Requirem	ent	
Design requirement type:	Step	response bounds
Requirement for response:	Close	ed Loop from Unit Step to Plant
Design requirement parame	ters –	
Initial value:	0	Final value: 1
Step time:	0	
Rise time:	5	% Rise: 80
Settling time:	10	% Settling: 1.0000
% Overshoot: 10.	0000	% Undershoot: 1
		OK Cancel Help

- **3** Specify the Bode magnitude lower limit requirement:
  - **a** In the New Design Requirement dialog box, select Bode magnitude lower limit from the **Design requirement type** drop-down list.

This action updates the New Design Requirement dialog box, as shown in the next figure.

📣 New Des	igr	n Requireme	ent					
Design requirement type:				Bode magnitude lower limit				
Requirement for response:				Loop from Unit Step to Controller 💌				
Design requ	Design requirement parameters							
Segments:								
	Sta	art	Er	nd				
Frequency	1	Magnitude	Frequency	Magnitude	Slope	Weight		
	1	0	10	0	0	1		
Insert Delete								
				ОК	Cancel	Help		

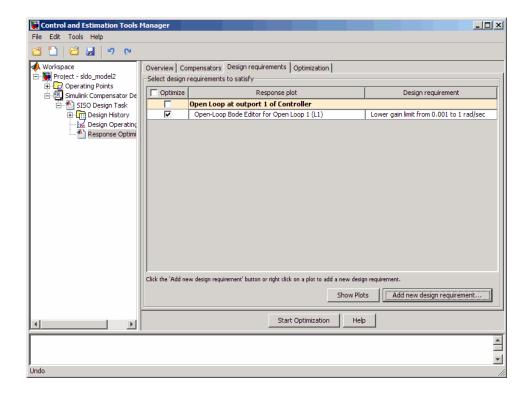
- **b** Select Open Loop at outport 1 of Controller from the **Requirement** for response drop-down list.
- c In the **Frequency** field of the **Start** column, enter 1e-3.
- **d** In the **Frequency** field of the **End** column, enter 1.

The New Design Requirement dialog box resembles the following figure.

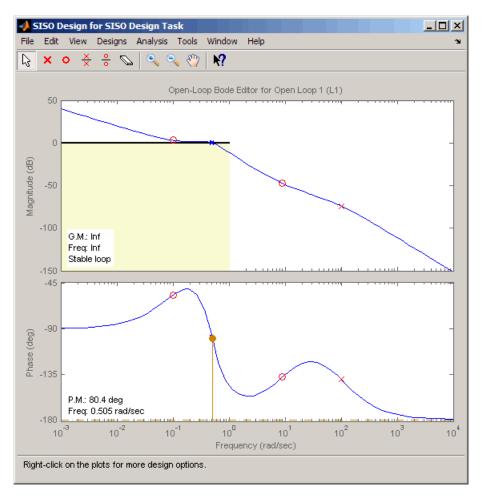
📣 New Desig	n Requireme	ent				
Design requirement type:			Bode magnitude lower limit			
Requirement for response:			en Loop at outport 1 of Controller			
Design requirement parameters Segments:						
St	Start					
Frequency	Magnitude	Frequency	Magnitude	Slope	Weight	
1.0000e-03	0	1	0	0	1	
Insert Delete						
			ОК	Cancel	Help	

e Click **OK** to close the New Design Requirement dialog box.

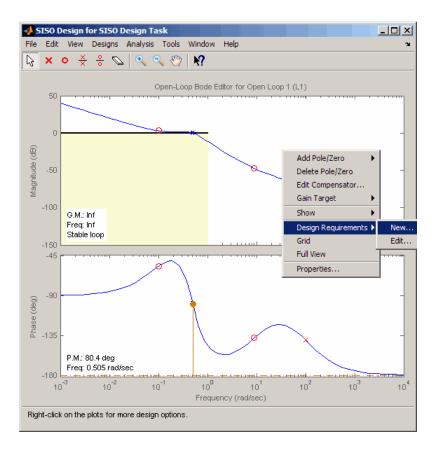
The Bode lower magnitude limit is added to the **Design requirements** tab, as shown in the next figure.



The SISO Design for SISO Design task window also updates to show the Bode plot with the design requirement displayed as the black line segment.



- **4** Add the phase margin requirement:
  - **a** In the SISO Design window, right-click the white area on the top plot, and select **Design requirement > New**.



This action opens the New Design Requirement dialog box, as shown in the next figure.

1	🗼 New Design Requirement							
De	Design requirement type: Upper gain limit							
Design requirement parameters Segments:								
Start End								
F	requency	Magnitude	Frequency	Magnitude	Slope	Weight		
	1	0	10	0	0	1		
Insert Delete								
				ОК	Cancel	Help		

**b** In the New Design Requirement dialog box, select Gain & phase margins from the **Design requirement type** drop-down list.

The New Design Requirement dialog box updates to display the **Gain Margin** > and **Phase Margin** > options, as shown in the next figure.

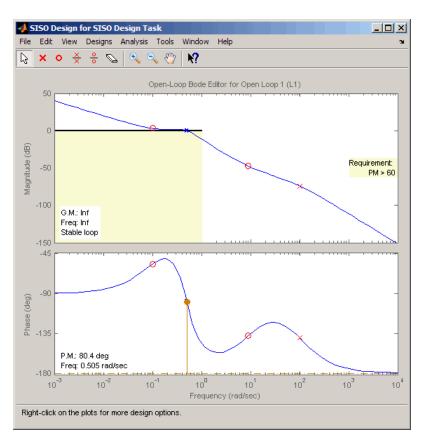
📣 New Design Requirement 📃 🔲 🗙						
Design requirement type:	& Phase margins 💌					
Design requirement parameters						
Gain margin >	20					
Phase margin >	30					
ок	Cancel Help					

c Select the **Phase margin** > check box, and enter 60 in the adjacent field. Then, click **OK** to close the dialog box.

📣 New Design Requirement 📃 🔲 🗙					
Design requirement ty	/pe:in & Phase margins 💌				
Design requirement p	arameters				
Gain margin >	20				
Phase margin >	60				
ОК	Cancel Help				

The **Design requirements** tab in the Control and Estimation Tools Manager GUI updates. It now displays the phase margin requirement, as shown in the next figure.

Control and Estimation Tools	lanager						
File Edit Tools Help							
🖆 🛅 😂 🛃 🔊 🍋							
📣 Workspace	Overview Compensators Design requirements Optimization						
Project - sldo_model2	Select design requirements to satisfy						
Erel Dig Operating Points	Coptimize Response plot	Design requirement					
🖃 🎦 SISO Design Task	Open Loop at outport 1 of Controller						
🕀 🛄 Design History	Open-Loop Bode Editor for Open Loop 1 (L1)	Lower gain limit from 0.001 to 1 rad/sec					
🖂 🔁 Design Operating	Open-Loop Bode Editor for Open Loop 1 (L1)	Requirement: PM > 60					
Response Optimi							
	J Click the 'Add new design requirement' button or right click on a plot to add a new des	ign requirement.					
	Show Pl	ots Add new design requirement					
	Start Optimization He	-In I					
		*					
		-					
1		<b>_</b>					



The SISO Design window also updates to display the phase margin requirement.

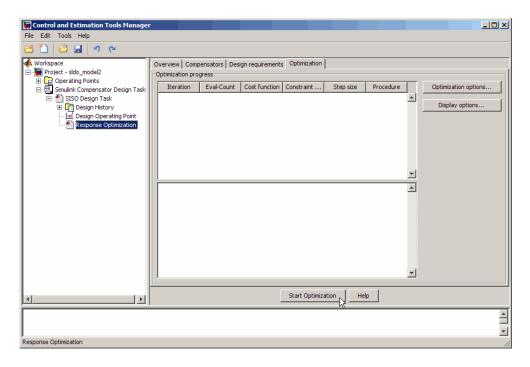
#### **Designing the Controller**

In this portion of the tutorial, you design the controller to meet the Bode magnitude and phase requirements.

You must have already specified the controller parameters to design, as described in "Specifying the Controller Parameters" on page 4-22, and the design requirements, as described in "Specifying Bode Magnitude and Phase Margin Design Requirements" on page 4-26.

To design the controller:

1 In the **Optimization** tab, click **Start Optimization**.



The **Optimization** tab updates as shown in the next figure.

Control and Estimation Tools Manager									
File Edit Tools Help									
C C C C C C C C C C C C C C C C C C C									
📣 Workspace	Overview Compensators Design requirements Optimization								
🖻 💽 Project - sldo_model2	Optimization progress								
Operating Points     Simulink Compensator De	Iteration	Eval-Count	Cost funct	Constrain	Step size	Procedure		Optimization options	
SISO Design Task	0	7	0	57.12					- 11
Design History								Display options,	
🔛 Design Operating Market Response Optimi									
Response Optimi									
							-		
	Constructing optimization problem								
	Optimization started 12-Nov-2008 11:44:04								
				Stop Optimi	Tation	Help			
				Stop Optimi		Theip			
									-
Personance Optimization									
Response Optimization									11.

- At every optimization iteration, the default optimization method Gradient descent reduces the distance between the current response and the magnitude requirement line segment by modifying the controller parameters. Simultaneously, the software also computes the phase margin and reduces the distance between the current response and the phase margin. To learn more about the optimization method, see "Selecting Optimization Methods".
- After the optimization completes, the **Optimization** tab displays the optimization iterations and status, as shown in the next figure.

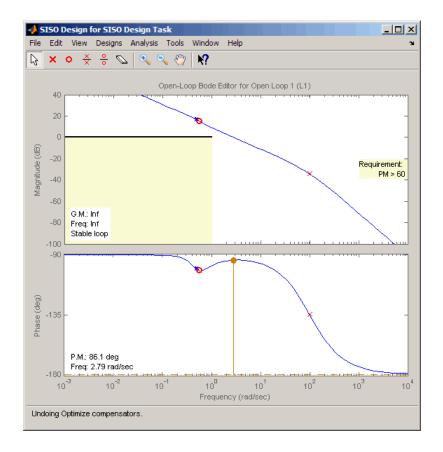
Gontrol and Estimation Too	ls Manager						
File Edit Tools Help							
🖆 🗋 🖆 🛃 🔊 (M							
Workspace	Overview Com		esign requirem	ents Optimiza	ation		
Operating Points	Optimization pro	ogress					 
Simulink Compensator	Iteration	Eval-Count	Cost function	Constraint	Step size	Procedure	Optimization options
🖃 🏝 SISO Design Task	0	7	0	57.12			
🕀 📆 Design History	1	14	0	1.777	8.36		Display options
Design Opera	2	21	0	1.697	12.1	Hessian m	
Response Opl	3	28	0	0	21.7		
	Optimization f	tarted 10-Nov inished 10-No	roblem 2008 14:01:11 v-2008 14:01:2 nd a feasible or	optimal solutio		-	
				Start Optimi	zation	Help	
							<u> </u>
							<b>_</b>
Redo							//

The message Successful termination indicates that the optimization method found a solution that meets the design requirements. For more information about the outputs displayed in the **Optimization progress** table, see "Iterative Display" in the Optimization Toolbox documentation.

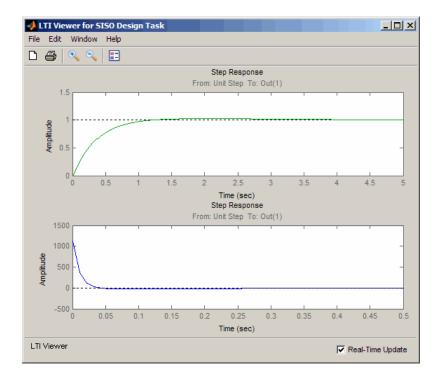
- **2** Examine the controller parameters and the system's response:
  - **a** View the optimized parameter values in the **Value** field of the **Compensator** tab.

Workspace	Overview Co	mpensators Design requirements Opti	mization				
Project - sldo_model2	Select compen	nsator elements to optimize					
E-E Simulink Compensator De	Optimize	Compensator elements	Value	Initial guess	Minimum	Maximum	Typical v
SISO Design Task		sldo_model2/Controller					
🗄 🛄 Design History		P	6.7627	1	-Inf	Inf	1
🛃 Design Operating		I	3.4445	0.1	-Inf	Inf	1
🔛 Response Optimi		D	11.106	0.1	-Inf	Inf	1
		N	100	100	-Inf	Inf	1
	Right click on a c	ompensator name to change its representation.			Use	e Value as Ini	tial Guess

- **b** Examine the system's response on the following plots:
  - The SISO Design window



- The top plot shows that the magnitude of the system, displayed as the blue curve, lies outside the yellow region. This plot indicates that the system has met the Bode magnitude requirement.
- The bottom plot displays the phase margin (P.M.) value of 86.1 degrees. This indicates that the system has met the phase margin design requirement of >60 degrees.
- The LTI Viewer



- The top plot shows that the closed-loop response of the system is stable. The system with the designed controller thus meets both the magnitude and phase margin requirements.
- The bottom plot in the LTI Viewer shows that the peak value of the controller's output is 1000, which is very large and can cause damage to the plant. To limit the controller output, you apply lower and upper bounds on the signal, as described in "Refining the Controller Design to Meet Controller Output Bounds" on page 4-42.

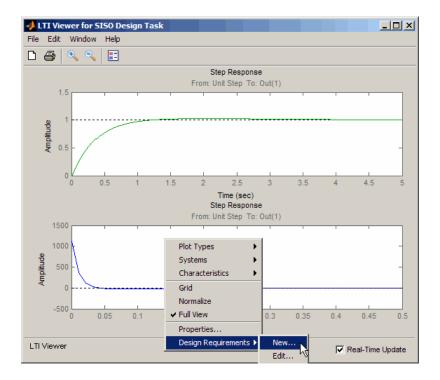
## Refining the Controller Design to Meet Controller Output Bounds

In this portion of the tutorial, you refine the controller to satisfy bounds on the controller's output.

You must have already designed an initial controller, as described in "Designing an Initial PID Controller to Meet Bode Magnitude and Phase Margins Requirements" on page 4-22.

To tune the compensator parameters to meet the bounds on the controller's signal:

- 1 Add the upper-bound on the controller's output:
  - **a** In the LTI Viewer, right-click the white area on the bottom plot, and select **Design requirement > New**.



This action opens the New Design Requirement dialog box.

📣 New Design	Requirem	ent	_ 🗆 🗙
Design requirem	nent type: Ste	ep response bour	nds 💌
-Design requirem	nent paramet	ers	
Initial value:	0	Final value:	1
Step time:	0		
Rise time:	5	% Rise:	80
Settling time:	10	% Settling:	1.0000
% Overshoot:	10.0000	% Undershoot:	1
	ОК	Cancel	Help

**b** In the New Design Requirement dialog box, select Upper time response bound from the **Design requirement type** drop-down list.

-	i New Desig	ın Requirem	ent			
	Design require	ement type: Up	oper time respo	onse bound		
Г	Design require	ement parame	ters			
	Segments:					
	Sta	art	En	ıd		
	Time	Amplitude	Time	Amplitude	Slope	Weight
	0	1	10	1	0	1
ľ						
					Insert	Delete
				ок	Cancel	Help

- c In the Time field of the End column, enter Inf.
- **d** In the **Amplitude** field of the **Start** column, enter **550**.
- e In the Amplitude field of the End column, enter 550.

The New Design Requirements dialog box resembles the following figure.

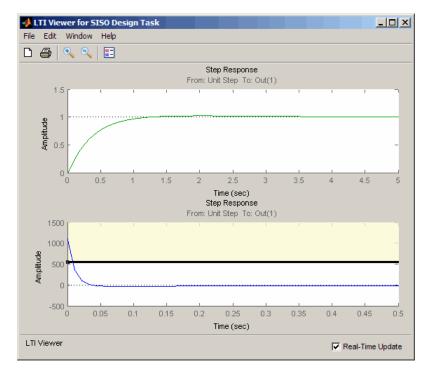
📣 New Desig	jn Requirem	ent			_ 🗆 🗙
Design require	ement type: Up	oper time resp	onse bound		•
Design require	ement parame	ters			
Segments:					
St	art	Er	nd		
Time	Amplitude	Time	Amplitude	Slope	Weight
0	550	Inf	550	0	1
1					
				Insert	Delete
			ОК	Cancel	Help

f Click OK to close the dialog box.

The **Design requirements** tab in the Control and Estimation Tools Manager GUI updates. It now displays the upper-bound requirement, as shown in the next figure.

Control and Estimation	Tools Manager		X
	0		
Workspace	<u>`</u>	pensators Design requirements Optimization	
🖻 💽 Project - sldo_model2	Select design re	quirements to satisfy	
Operating Points     Simulink Compens	Optimize	Response plot	Design requirement
SISO Design		Closed Loop from Unit Step to Controller	
🕀 📆 Design Hi	<b>V</b>	Step Response	Upper time response bound from 0 to 10 sec
🚽 📈 Design O		Open Loop at outport 1 of Controller	
Response	<b>V</b>	Open-Loop Bode Editor for Open Loop 1 (L1)	Lower gain limit from 0.001 to 1 rad/sec
		Open-Loop Bode Editor for Open Loop 1 (L1)	Requirement: PM > 60
		design requirement' button or right click on a plot to add a new design re	
	Click the Add new	design requirement button or right click on a plot to add a new design re-	quirement.
		Show Pl	Add new design requirement
•		Start Optimization Help	
J.			· · · · · · · · · · · · · · · · · · ·

The LTI Viewer also updates to show the design requirement, as shown in the next figure.



- **2** Add the lower-bound on the controller's output:
  - **a** Right-click the white area in the bottom plot in the LTI Viewer, and select **Design requirement > New**.

This action opens the New Design Requirement dialog box.

**b** Select Lower time response bound from the **Design requirement type** drop-down list.

The New Design Requirement dialog box updates to show the lower bound, as shown in the next figure.

📣 New Desig	ın Requirem	ent			
Design require	ement type:	wer time resp	onse bound		
-Design require	ement parame	ters			
Segments:					
St	art	En	ıd		
Time	Amplitude	Time	Amplitude	Slope	Weight
0	1	10	1	0	1
1					
				Insert	Delete
			ок	Cancel	Help

- c In the **Time** field of the **End** column, enter **Inf**.
- **d** In the **Amplitude** field of the **Start** column, enter -250.
- e In the Amplitude field of the End column, enter -250.

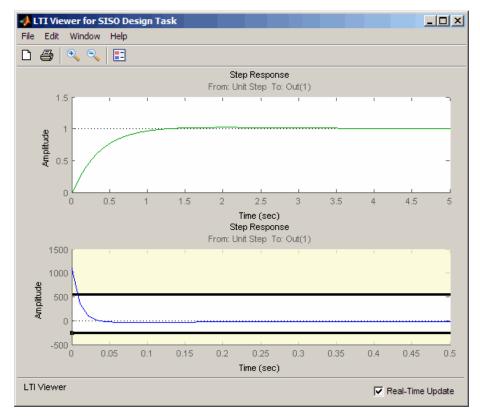
The New Design Requirement dialog box resembles the next figure. Click **OK** to close the dialog box.

Design require		ower time resp	onse bound		_
	ement parame	ters			
Segments:	art	Er	nd		
Time	Amplitude	Time	Amplitude	Slope	Weight
0	-250	Inf	-250	0	1
			ļ	Insert	Delete
			ОК	Cancel	Help

The **Design requirements** tab in the Control and Estimation Tools Manager GUI updates. It now displays the lower-limit requirement, as shown in the next figure.

File Edit Tools Help	Tools Manager		
	C		
Workspace	<u> </u>	pensators Design requirements Optimization	
Project - sldo_model2		quirements to satisfy	
Operating Points	Optimize	Response plot	Design requirement
🗄 🗐 Simulink Compens		Closed Loop from Unit Step to Controller	Design requirement
E Design Hi		Step Response	Upper time response bound from 0 to 10 sec
🚽 Design O	<b>N</b>	Step Response	Lower time response bound from 0 to 10 sec
🔤 Response		Open Loop at outport 1 of Controller	
		Open-Loop Bode Editor for Open Loop 1 (L1)	Lower gain limit from 0.001 to 1 rad/sec
	<b>V</b>	Open-Loop Bode Editor for Open Loop 1 (L1)	Requirement: PM > 60
	Click the 'Add new	design requirement' button or right click on a plot to add a new design rea	quirement.
		Show Pl	ots Add new design requirement
		Start Optimization Help	

The LTI Viewer also updates to show the lower-bound on the controller's output, as shown in the next figure.



- **3** Optimize the parameters to meet the design requirements on the controller output:
  - **a** In the **Compensators** tab, select the rows containing P, I, and D, and click **Use Value as Initial Guess**.

🙀 Control and Estimation Tools N	lanager						<u>_   X</u>
File Edit Tools Help							
🖆 🛅 🖨 🛃 🔊 🤊							
♦ Workspace ⊡-• ₩ Project - sldo_model2		mpensators Design requirements Of	otimization				
🕀 🔁 Operating Points	Optimize	Compensator elements	Value	Initial guess	Minimum	Maximum	Typical v
🗄 🗐 Simulink Compensator De		sldo_model2/Controller	, take	Thinking guess		- Iskingin	r ypicar trii
E. Design History		P	6.7627	1	-Inf	Inf	1
Design Operating		I	3.4445	0.1	-Inf	Inf	1
Response Optimi		D	11.106	0.1	-Inf	Inf	1
		N	100	100	-Inf	Inf	1
	Right click on a c	ompensator name to change its representation	<b>1.</b>		Use	e Value as Ini	tial Guess 1
		Start O	ptimization	Help			
							▲ ▼

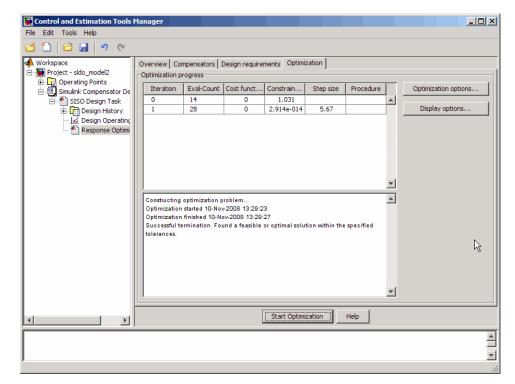
When you run the optimization again, the optimization method uses the updated parameter values as the starting point for refining the values.

Clicking **Use Value as Initial Guess** updates the values in the **Initial Guess** column, as shown in the next figure.

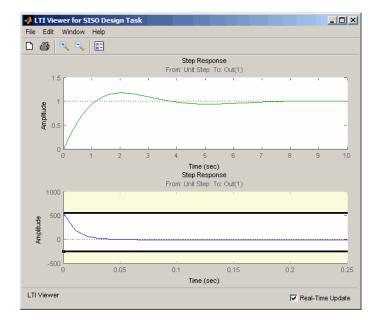
Workspace	Our main of the	Companyat	ors Design requirements Opti	ninetice 1				
Project - sldo_model2			ments to optimize	mization				
Operating Points	· · · ·			1	[n		[	1
🗄 🗐 Simulink Compensator De	C Optimiz		Compensator elements	Value	Initial guess	Minimum	Maximum	Typical v.
🖻 🎦 SISO Design Task		P	odel2/Controller	6.7627	6.7627	-Inf	Inf	
Design History     Design Operating	<u>र</u>	I		3,4445	3.4445	-Inf	Inf	1
Response Optimi	V V	D		11.106	11.106	-Inr -Inf	Inf	1
- Kesponse Optimi		N		100	100	-Inf	Inf	1
	Right click on a	a compensati	r name to change its representation.					
	Right click on a	a compensato	r name to change its representation.			Use	e Value as Init	tial Guess

## **b** In the **Optimization** tab, click **Start Optimization**.

- At every optimization iteration, the optimization method reduces the distance between the current response and the upper and lower bounds on the signal.
- After the optimization completes, the **Optimization** tab resembles the next figure.



- **4** Examine the parameter values refined controller and the system's response:
  - **a** Examine the following response plots:
    - The LTI viewer

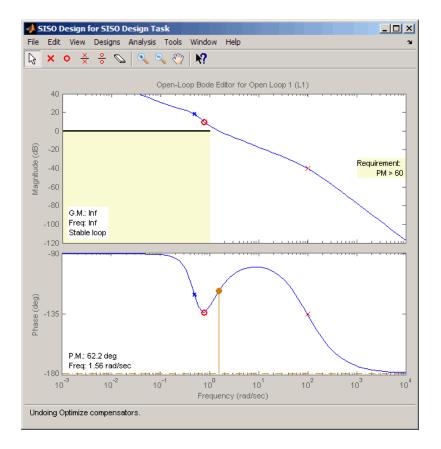


The bottom plot shows that the controller output lies between 550 and -250, and thus meets the design requirement on the controller's output.

**Note:** You must also check that the closed-loop response, shown in the top plot, remains stable after refining the controller design.

The SISO Design window

•



The plots show that after refining the design, the system continues to meet the magnitude and phase margin requirements specified in "Design Requirements" on page 4-16.

**b** View the optimized controller parameter values in the **Value** field in the **Compensators** tab.

		mpensators Design requirements Opti	imization				
Project - sldo_model2	Select comper	Compensator elements	Value	Initial guess	Minimum	Maximum	Typical v
🗄 🗐 Simulink Compensator De		sldo model2/Controller	Value	Thicki guess	Paratican	Maximum	Typicarv
E. C Design History		P	6.706	6.7627	-Inf	Inf	1
Design Operating		I	3.4445	3.4445	-Inf	Inf	1
Response Optimi		D	5.4329	11.106	-Inf	Inf	1
		N	100	100	-Inf	Inf	1
		1					
	Right click on a (	ompensator name to change its representation.			Use	: Value: as Ini	tial Guess

## 5 Select the SISO Design Task node, and click Update Simulink Block Parameters.

This action writes the optimized controller parameter values to the Controller block in the Simulink model.

## Saving the Project

To save a project with the optimized controller parameters:

1 In Control and Estimation Tools Manager GUI, select **File > Save**.

This action opens the Save Projects dialog box.

📣 Save	Projects			<u> </u>
Projects	:			
Project	- sldo_mod	el2		
				<b>T</b>
1				
	ОК	(	Cancel	Help

- **2** In the Save Projects dialog box, select **Project sldo\_model2**, and click **OK** to open the Save Projects window .
- 3 In the Save Projects window, enter sldo\_model2\_optimized.mat in the File name field, and click Save.

The action saves the project as a MAT-file.

**Tip** You can reload this project by typing **sisotool**('sldo\_model2\_optimized') at the MATLAB prompt.