Simulink® Design Optimization™

Getting Started Guide

R2012b

MATLAB® & SIMULINK®

MathWorks®

How to Contact MathWorks

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`www.mathworks.com/contact_TS.html`	Technical Support
`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

508-647-7000 (Phone)

508-647-7001 (Fax)

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

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

Simulink® Design Optimization™ Getting Started Guide

Revision History

March 2009	Online only	New for Version 1 (Release 2009a)
September 2009	Online only	Revised for Version 1.1 (Release 2009b)
March 2010	Online only	Revised for Version 1.1.1 (Release 2010a)
September 2010	Online only	Revised for Version 1.2 (Release 2010b)
April 2011	Online only	Revised for Version 1.2.1 (Release 2011a)
September 2011	Online only	Revised for Version 2.0 (Release 2011b)
March 2012	Online only	Revised for Version 2.1 (Release 2012a)
September 2012	Online only	Revised for Version 2.2 (Release 2012b)

Contents

1 Product Overview

2 Parameter Estimation

3 Response Optimization

4 Optimization-Based Linear Control Design

A Examples

Index

Product Overview

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™)
- 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

What You Can Accomplish Using This Product

Simulink Design Optimization software supports parameter optimization for Simulink models that invoke third-party simulation tools or contain legacy simulation code (using S-functions).

The software uses methods from the Optimization Toolbox™ and Global Optimization Toolbox products for design optimization. Using the software with the Parallel Computing Toolbox™ software can speed up numerical computations during estimation and optimization.

The software provides library blocks that let you accomplish the following goals:

- Model time-varying systems as lookup tables using “Adaptive Lookup Tables”
- Optimize model parameters by graphically specifying time-domain design requirements and reference signal using Signal Constraint blocks
- Compute and optimize continuous and discrete root mean square values of signals using CRMS and DRMS blocks with Signal Constraint blocks

When the Control System Toolbox™ software is installed, you can refine controller parameters of linear time-invariant (LTI) models and Simulink models, linearized using the Simulink Control Design software, in the SISO Design Tool.

You can work either in the Graphical User Interface (GUI) or at the command line. New users should start by using the GUI to become familiar with the product. The following operations are available only using the GUI:

- Interactive data preprocessing for parameter estimation
- Model validation using residual plot for validating estimation
- Optimization-based controller design

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 S-function block in Simulink. When using the command-line functions, use MATLAB® 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.

Speeding Up Design Optimization Using Parallel Computing

When you have the Parallel Computing Toolbox 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.
- The`Pattern 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.

Learn More

The Simulink Design Optimization documentation provides information to use this product. Although this product employs optimization methods for parameter estimation and optimization, using the product does not require you to have a strong background in optimization theory. You may find it helpful to:

- Consult the Optimization Toolbox documentation to learn more about optimization theory, and minimizing an objective function.
- Learn how Simulink Design Optimization algorithms formulate the optimization problems. See "How the Optimization Algorithm Formulates Minimization Problems" in the User's Guide.

Required and Related Products

Simulink Design Optimization software requires MATLAB, Simulink, and Optimization Toolbox software.

The following table summarizes MathWorks® 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.

Documentation

Accessing Documentation

The Simulink Design Optimization documentation contains the following components:

- Getting Started Guide — Provides information for mapping your problem to the capabilities of the Simulink Design Optimization software. Step-by-step tutorials walk you through the most common tasks for estimating parameters, optimizing parameters, and designing controllers using optimization methods.
- Reference — Describes commands and blocks for design optimization.
- Release Notes — Describes important changes in the current product version and compatibility considerations.

If you are new to using this product, the Getting Started Guide helps you begin using this product quickly. You can follow the steps in the tutorials to perform design optimization using the graphical user interface (GUI) or the MATLAB Command Window.

You can also search or browse the documentation for information about specific design optimization tasks.

2

Parameter Estimation

Supported Data

You can estimate model parameters and initial conditions of single or multiple input and output Simulink models from *transient data*. You measure *transient data* when the system is not in steady-state to capture the system dynamics expected under normal operating conditions. For example, the response of a system to step or impulse inputs is transient data.

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 (GUI)".
- *Time-series* data — Data stored in time-series objects. For more information, see "Time Series Objects" in the MATLAB documentation. See "Importing Time-Series Data into the GUI".

Using complex data for parameter estimation is not directly supported. See "Importing Complex Data into the GUI".

Simulink Design Optimization software estimates model parameters by comparing the transient 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 Using the GUI" on page 2-47.

Quick Start — Estimating Model Parameters

In this quick start, you get an overview of the typical tasks for estimating model parameters using the Control and Estimation Tools Manager GUI:

1. Start a parameter estimation task.
2. Import estimation and validation data sets.
3. Select parameters to estimate.
4. Estimate the parameters from the estimation data.
5. Validate the estimated parameters using the validation data set.

Prerequisites for parameter estimation include:

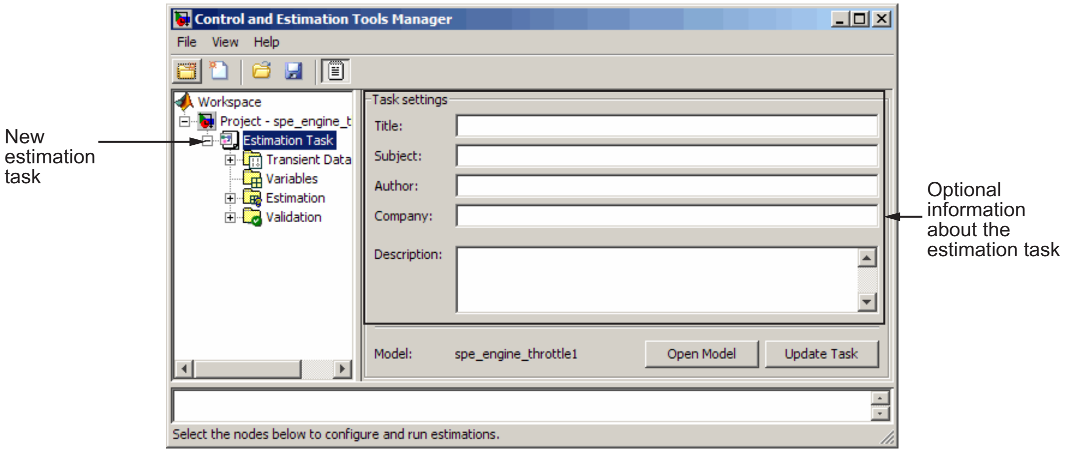
- Simulink model that contains inport or outport blocks, or signal logging

  For more information, see "Model Requirements for Importing Data" in the *Simulink Design Optimization User's Guide*.

- Transient data in the MATLAB workspace

To estimate model parameters:

1. Start a parameter estimation task by selecting **Analysis > Parameter Estimation** in the Simulink model window.

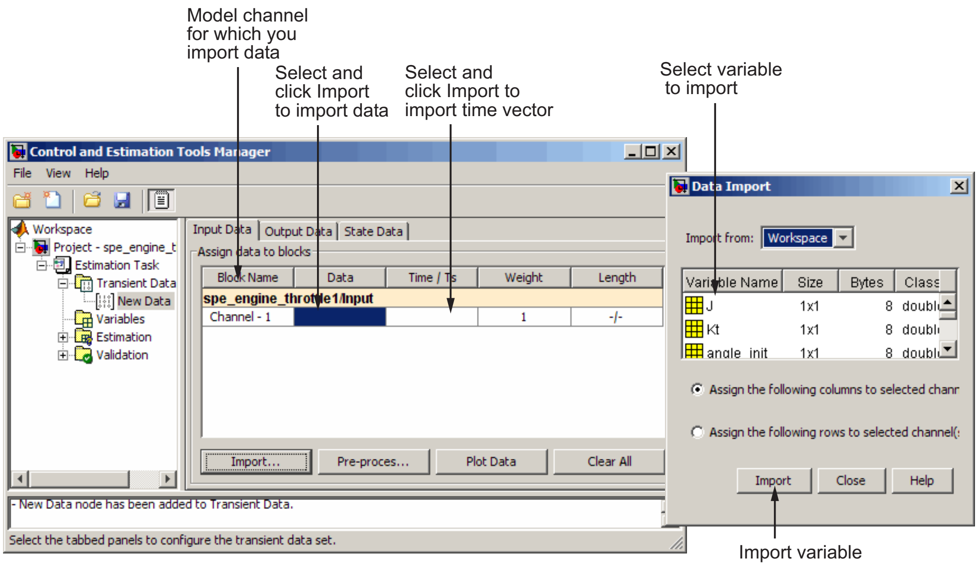

New estimation task
Optional information about the estimation task
Control and Estimation Tools Manager
File View Help
Workspace
Project - spe_engine_t
Estimation Task
Transient Data
Variables
Estimation
Validation
Task settings
Title:
Subject:
Author:
Company:
Description:
Model: spe_engine_throttle1
Open Model
Update Task
Select the nodes below to configure and run estimations.

2 Import the input and output data for estimating and validating model parameters.

a Select the **Transient Data** node, and click **New**.

b Select the **New Data** node.

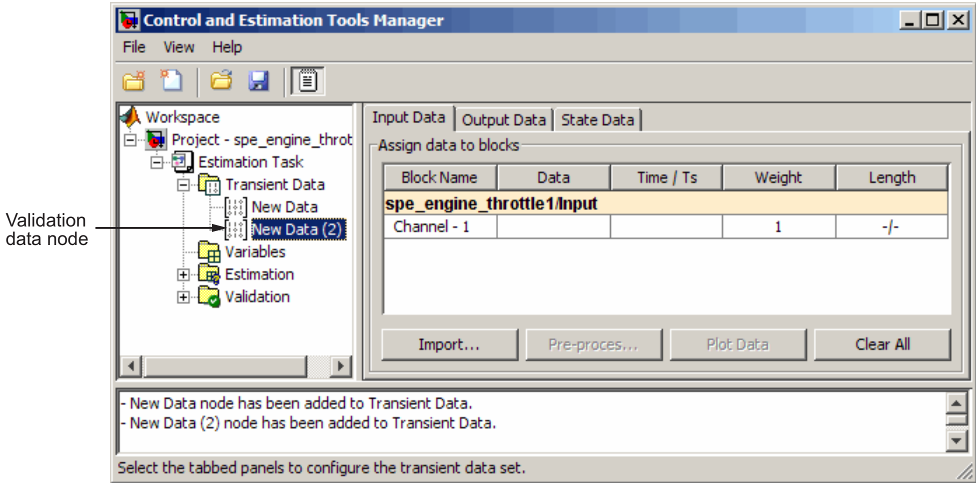
c In the **Input Data** tab, select the **Data** cell corresponding to the model channel, and click **Import**. Select the variable to import in the Data Import dialog box.



d Select the **Time /Ts** cell, and click **Import**. Select the time vector to import in the Data Import dialog box.

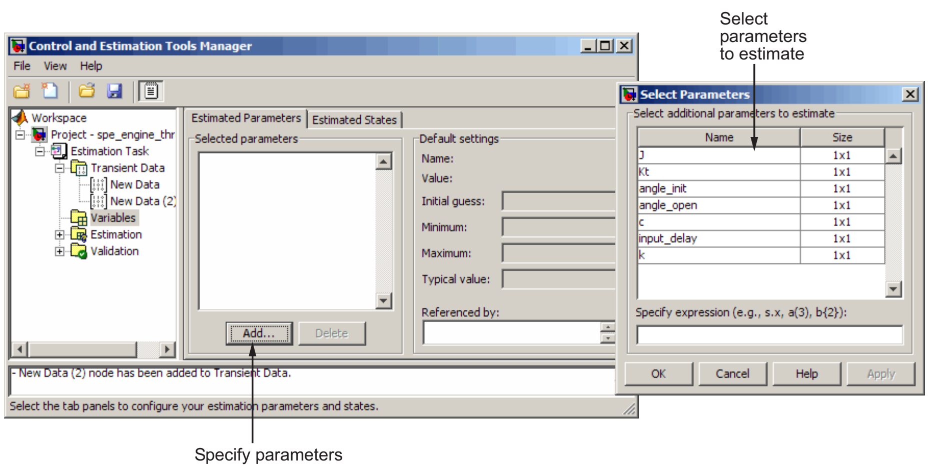
e In the **Output Data** tab, repeat steps c–d to import the output data and time vector.

f Repeat steps a–d to import the validation data set.



For more information, see “Import Data (GUI)” in the *Simulink Design Optimization User’s Guide*.

3 Specify parameters to estimate by selecting the **Variables** node, and clicking **Add**. Select the parameters in the Select Parameters dialog box.



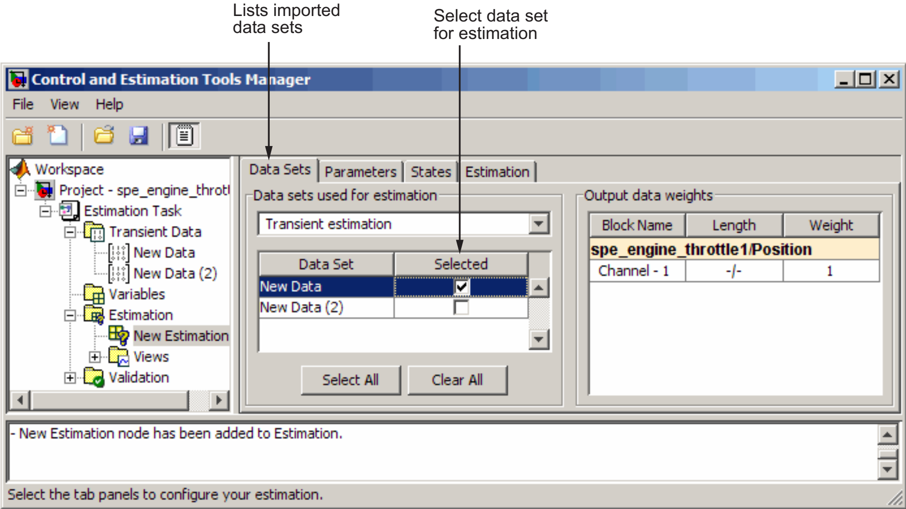
For more information, see “Specify Parameters to Estimate (GUI)” in the *Simulink Design Optimization User’s Guide*.

4 Estimate the parameters.

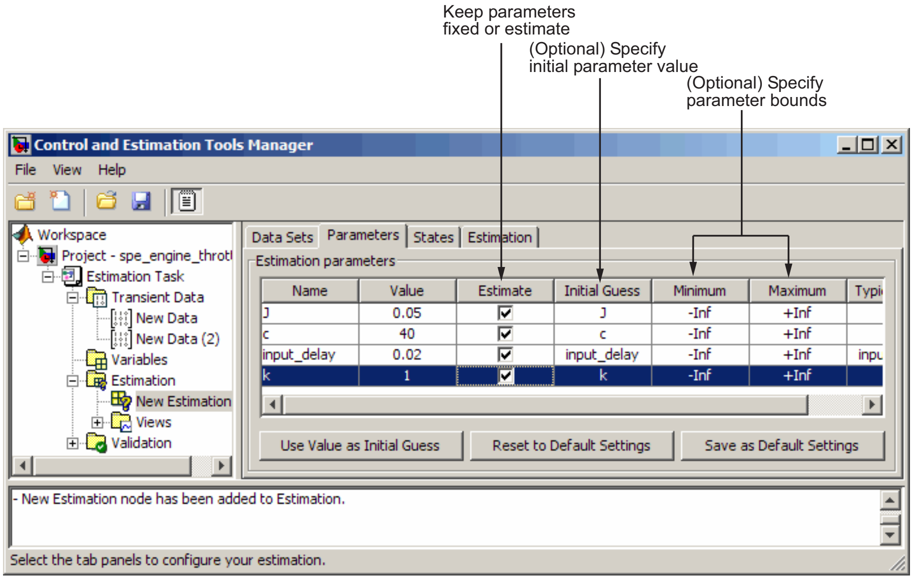
a Select the **Estimation** node, and click **New**.

b Select the **New Estimation** node.

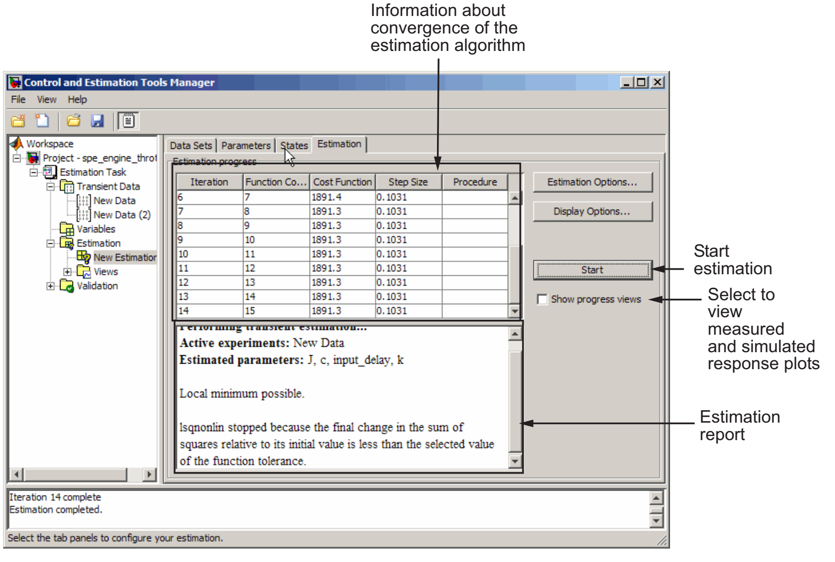
c In the **Data Sets** tab, select the estimation data set.



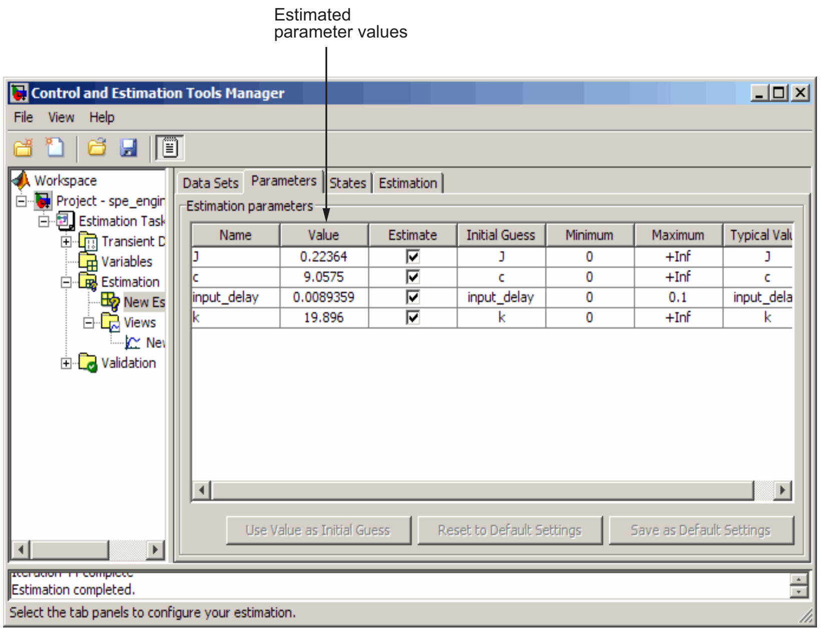
d In the **Parameters** tab, select the parameters to estimate.



e In the **Estimation** tab, begin estimation by clicking **Start**.



f In the **Parameters** tab, examine the estimated parameter values. The Simulink model also gets updated with the estimated parameter values.



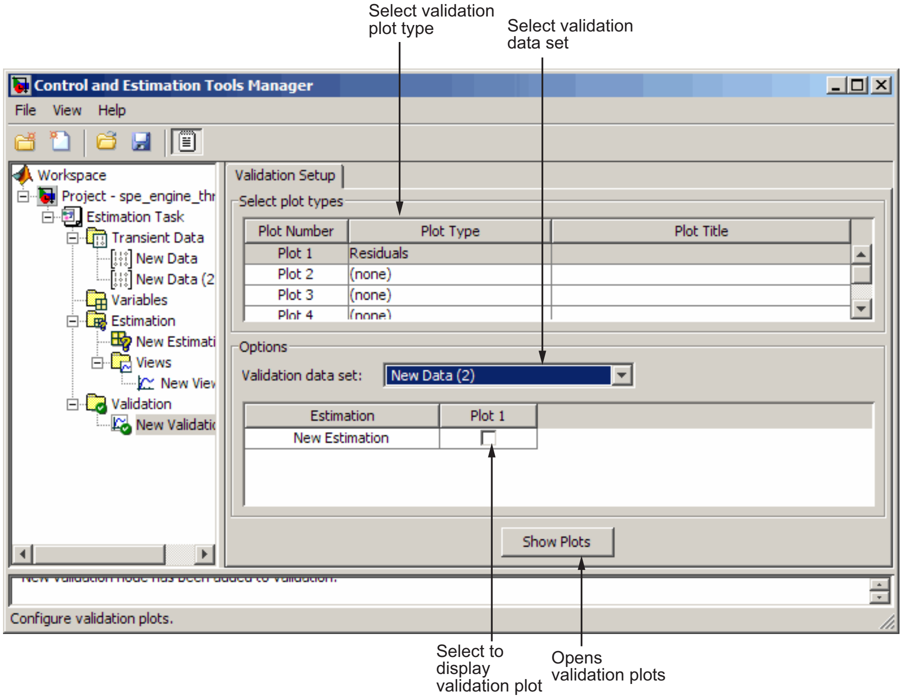
For more information, see "Estimate Parameters (GUI)" in the *Simulink Design Optimization User's Guide*.

5 Validate the estimated parameters.

a Select the **Validation** node, and click **New**.

b Select the **New Validation** node.

c Configure the validation plots and the validation data set.



For more information, see "Validate Parameters (GUI)" in the *Simulink Design Optimization User's Guide*.

See Also: "Estimate Parameters from Measured Data Using the GUI" on page 2-47

Prepare Data for Parameter Estimation Using the GUI

In this section...

About This Tutorial

Objectives

In this tutorial, you learn 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, time-domain data only.

You learn to perform the following tasks using the GUI:

- Import data from the MATLAB workspace.
- Analyze data quality using a time plot.
- Select a subset of data for estimation.

- Remove outliers.
- Filter high-frequency noise.
- Compute missing data using interpolation.

About the Sample Data

In this tutorial, you use `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.mdl` contains the Simulink model of the engine throttle system.

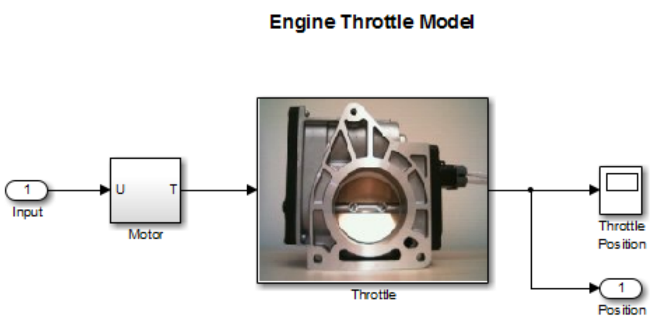
Configuring a Project for Parameter Estimation

To perform parameter estimation, you must first configure a Control and Estimation Tools Manager project.

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

```
spe_engine_throttle1
```

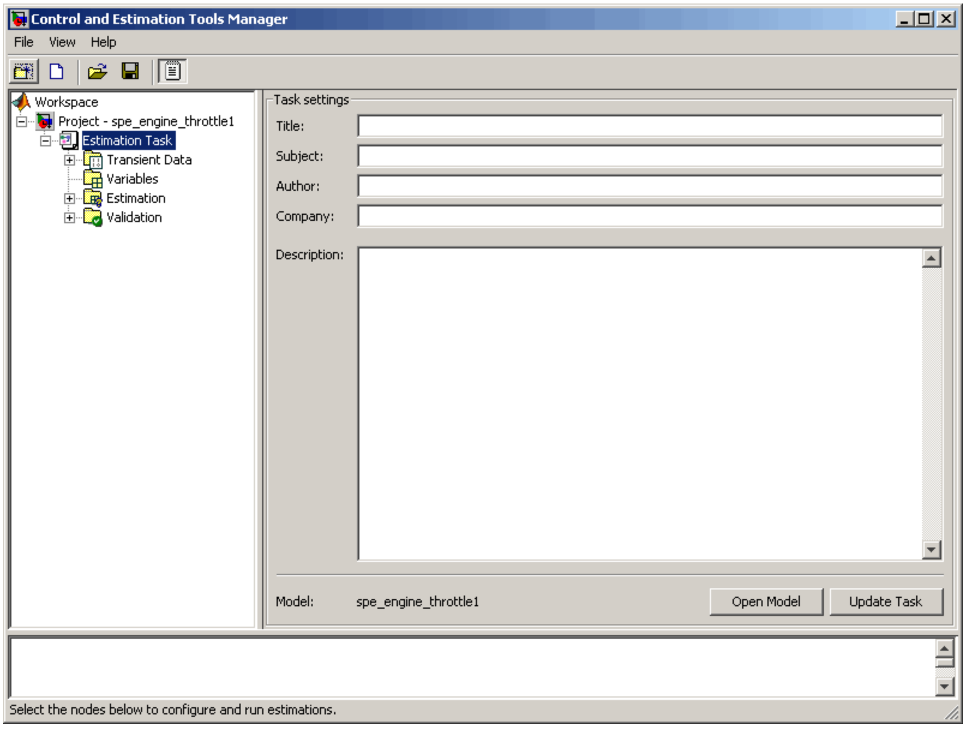
This command opens the Simulink model, and loads the data into the MATLAB workspace.



Simulink® Model of Engine Throttle System

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

This action opens a new project named **Project - spe_engine_throttle1** in the Control and Estimation Tools Manager GUI. This project contains the **Estimation Task**, as shown in the next figure.



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

Importing Data into the GUI

- "Importing Input Data and Time Vector" on page 2-15

Importing Input Data and Time Vector

In this portion of the tutorial, you import measured I/O data into the Control and Estimation Tools Manager GUI. Importing data assigns the data to the corresponding model input and output signals.

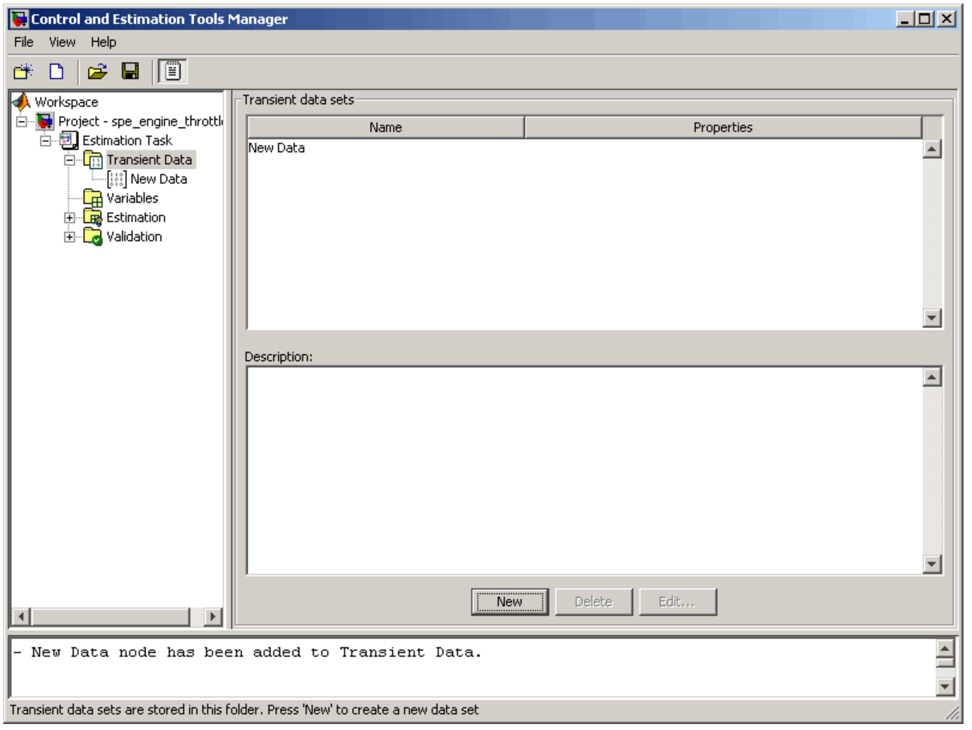
The model input and output signals are designated with the Inport `Input` and Outport `Position` blocks respectively, as shown in the figure Simulink® Model of Engine Throttle System on page 2-13. These blocks let you import I/O data into the GUI. To learn more about the blocks, see the Inport and Outport block reference pages in the Simulink documentation.

You must have already configured the parameter estimation project, as described in “Configuring a Project for Parameter Estimation” on page 2-12.

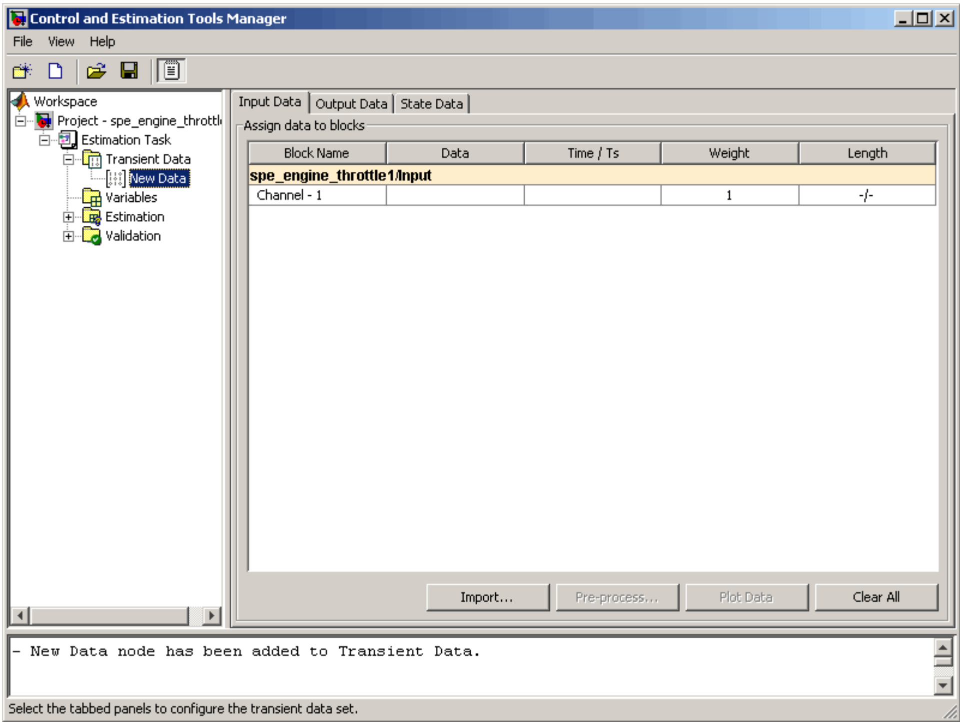
To import input data and time vector into the Control and Estimation Tools Manager GUI:

1 In the Control and Estimation Tools Manger GUI, select **Transient Data** under the **Estimation Task** node, and click **New**.

This action creates a **New Data** node under the **Transient Data** node.



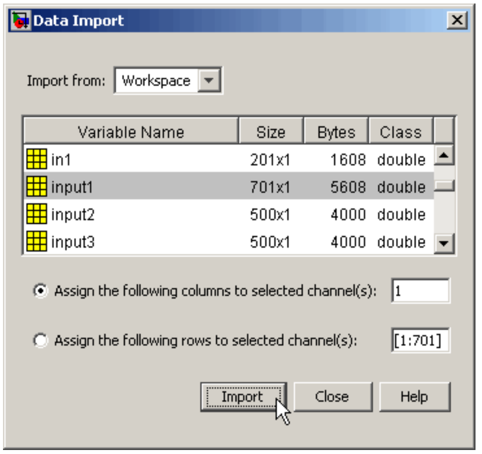
2 Select the **New Data** node.



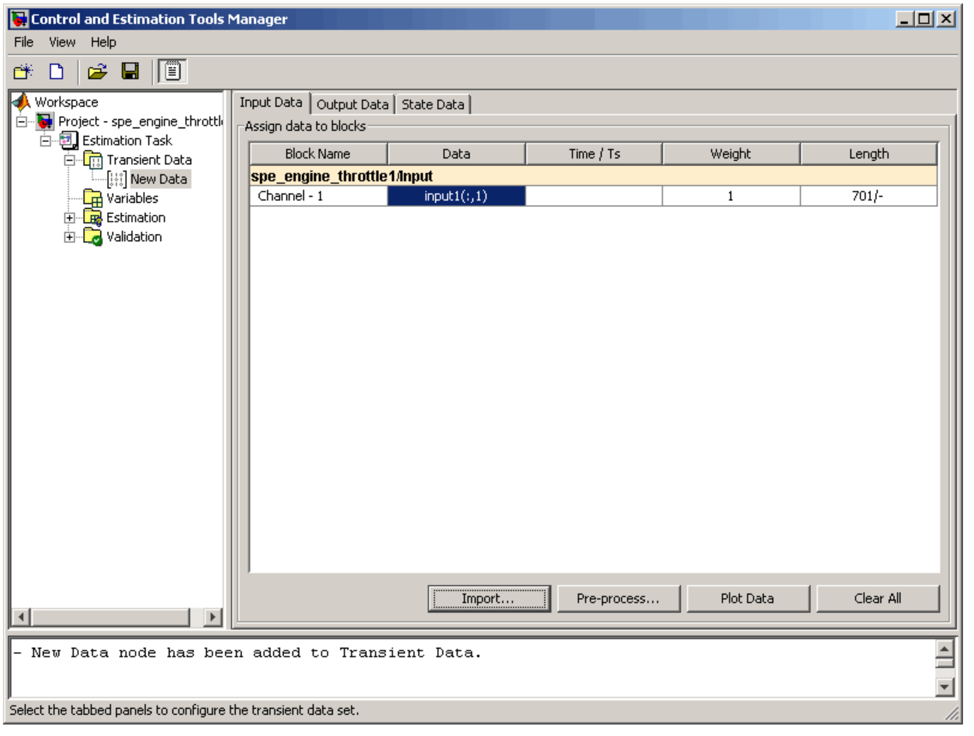
3 In the **Input Data** tab, select the **Data** cell for **Channel - 1**, and click **Import**.

This action opens the Data Import dialog box.

4 In the Data Import dialog box, select `input1`, and click **Import**.



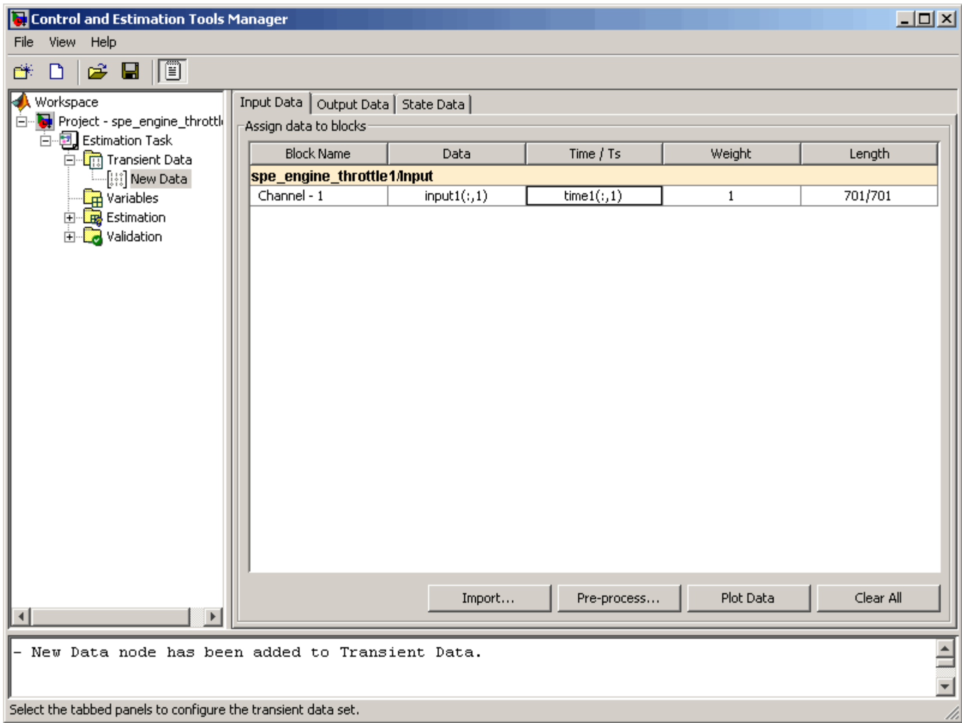
This action assigns the input data `input1` to the model input signal **spe_engine_throttle1/Input**.



5 Select the **Time / Ts** cell for **Channel - 1**.

6 In the Data Import dialog box, select `time1`, and click **Import**.

This action assigns the time vector to the model input signal **spe_engine_throttle1/Input**.

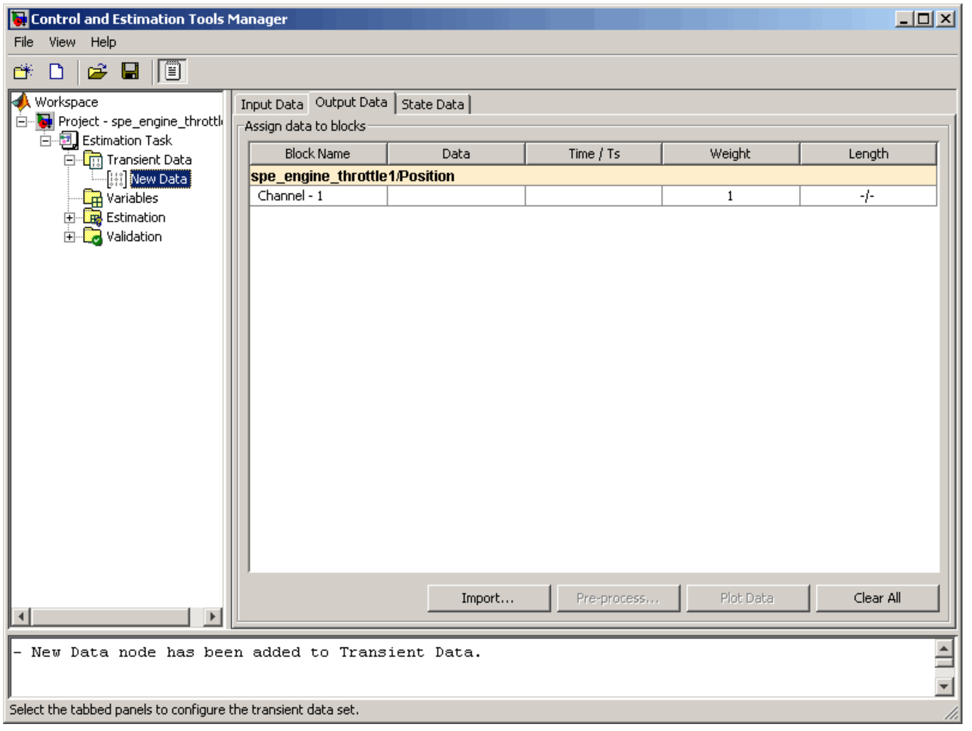


7 In the Data Import dialog box, click **Close**.

Importing Output Data and Time Vector

To import output data and time vector into the Control and Estimation Tools Manager GUI:

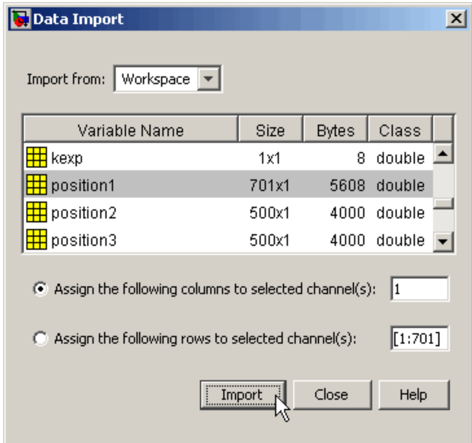
1 In the Control and Estimation Tools Manger GUI, select the **Output Data** tab of the **New Data** node.



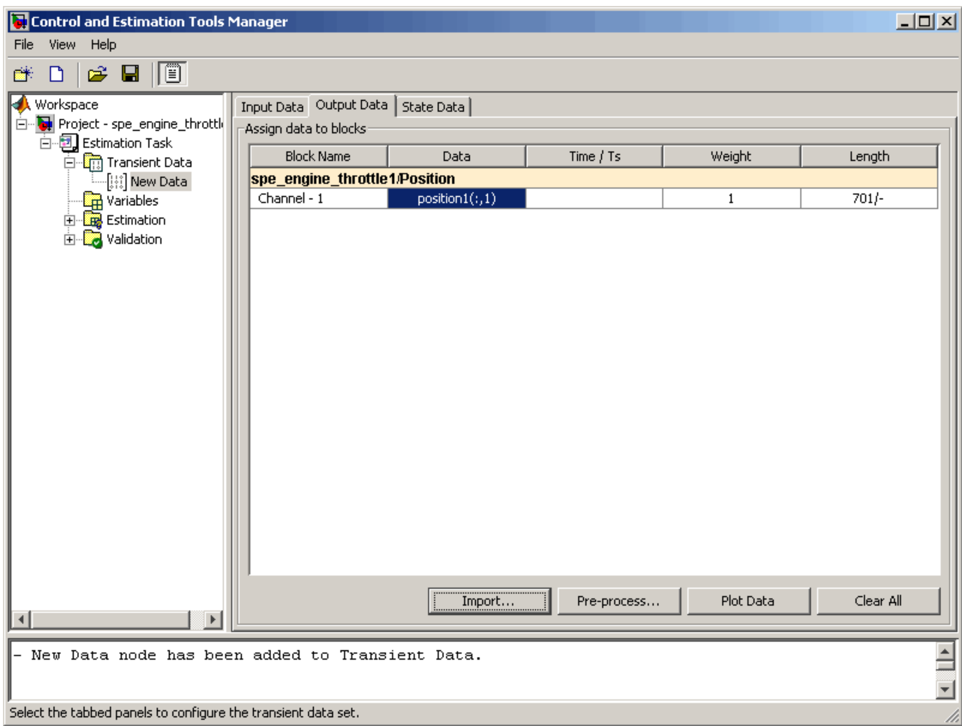
2 Select the **Data** cell for **Channel - 1**, and click **Import**.

This action opens the Data Import dialog box.

3 In the Data Import dialog box, select `position1`, and click **Import**.



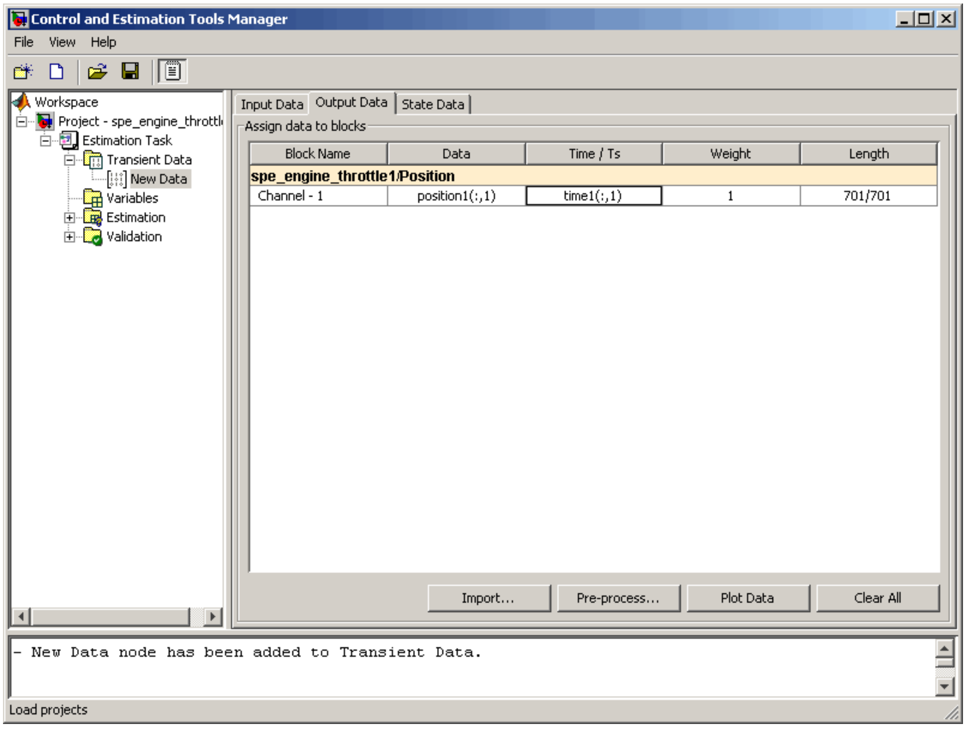
This action assigns the output data `position1` to the model output signal **spe_engine_throttle1/Position**.



4 Select the **Time / Ts** cell for **Channel - 1**.

5 In the Data Import dialog box, select `time1`, and click **Import**.

This action assigns the time vector to the model output signal **spe_engine_throttle1/Position**.



6 In Data Import dialog box, click **Close**.

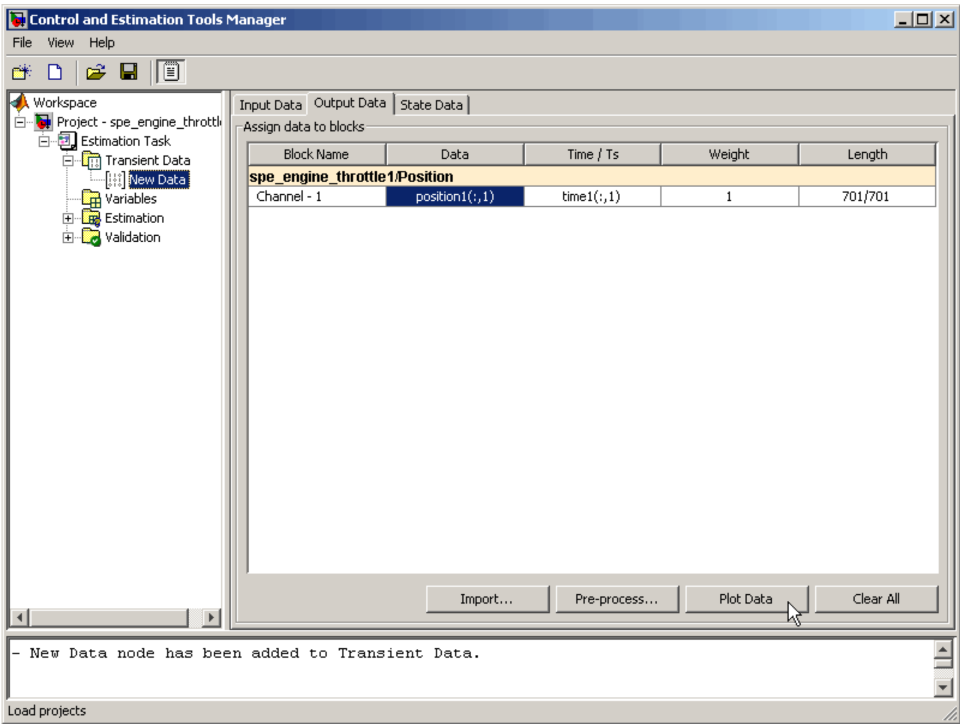
You have now imported the I/O data into the Control and Estimation Tools Manager GUI, and assigned the data to the corresponding model signals.

Analyzing Data

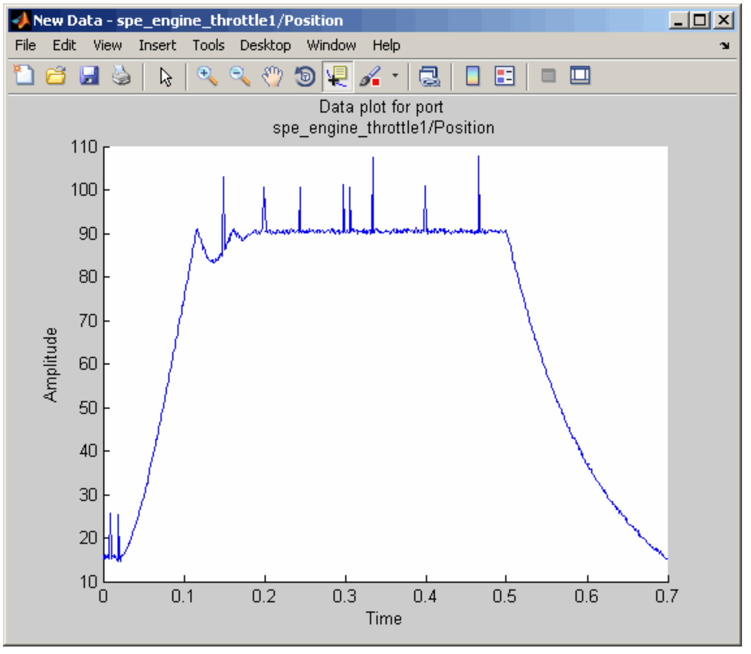
In this portion of the tutorial, you analyze the output data quality by viewing the data characteristics on a time plot. Based on the analysis, you decide whether to preprocess the data before estimating parameters. For example, if the data contains noise, you might want to filter the noise from the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

You must have already imported the data into the Control and Estimation Tools Manager GUI, as described in "Importing Data into the GUI" on page 2-14. If you have not imported the data, click here.

To plot the output data on a time plot, select the `position1(:,1)` cell in the **Output Data** tab, and click **Plot Data**.



This action plots the measured output data `position1(:,1)`, as shown in the next figure.



The time plot shows the output data in response to a step input, as described in "About the Sample Data" on page 2-12. 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, you select the data samples between t = 0 s and t = 0.5 s, as described in "Selecting Data for Estimation" on page 2-25 section of this tutorial.

The spikes in the data indicate *outliers*, defined as data values that deviate from the mean by more than three standard deviations. They may be caused by measurement errors or sensor problems. The response also contains noise. Before estimating model parameters from this data, you remove the outliers and filter the noise, as described in "Removing Outliers" on page 2-33, and "Filtering Data" on page 2-37 sections of this tutorial.

Tip You can also plot the input data on a time plot by selecting the `input1(:,1)` cell in the **Input Data** tab, and clicking **Plot Data**.

Selecting Data for Estimation

Selecting Output Data

In this portion of the tutorial, you select a subset of I/O data for estimation. As described in “Analyzing Data” on page 2-23, the system is shut down at t = 0.5 s. To focus the estimation on the time period before t = 0.5 s, you 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.

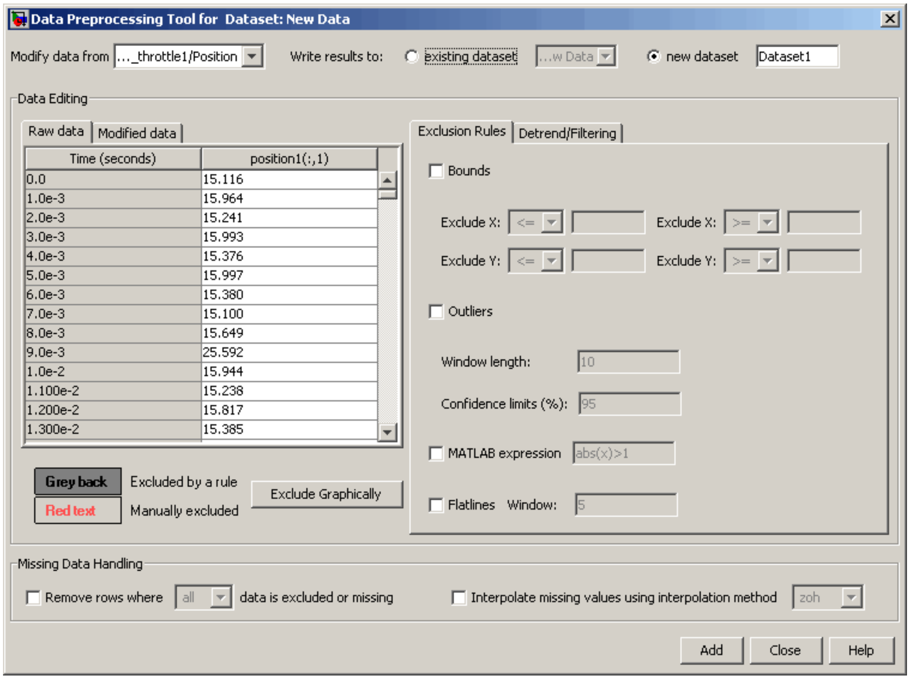
You must have already imported the data into the Control and Estimation Tools Manager GUI, as described in “Importing Data into the GUI” on page 2-14. If you have not imported the data, click here.

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

1 In the Control and Estimation Tools Manager window, select the **New Data** node under the **Transient Data** node.

2 Select the `position1(:,1)` cell in the **Output Data** tab, and click **Pre-process**.

This action opens the Data Preprocessing Tool GUI.

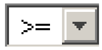


The **Data Editing** area of the Data Preprocessing Tool GUI shows the output data and time vector in the **position1(:,1)** and **Time (seconds)** columns, respectively. The Data Preprocessing Tool GUI lets you perform the following types of preprocessing operations:

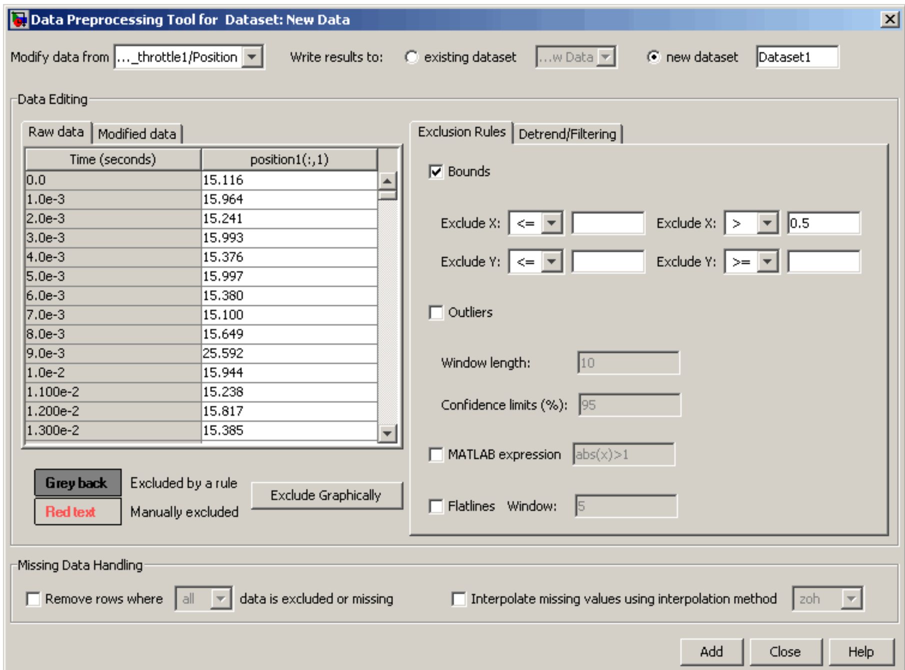
- Excluding data
- Detrending and filtering data
- Handling missing data

3 To exclude the output data beyond t = 0.5 s:

a In the **Exclusion Rules** tab, select the **Bounds** check box.

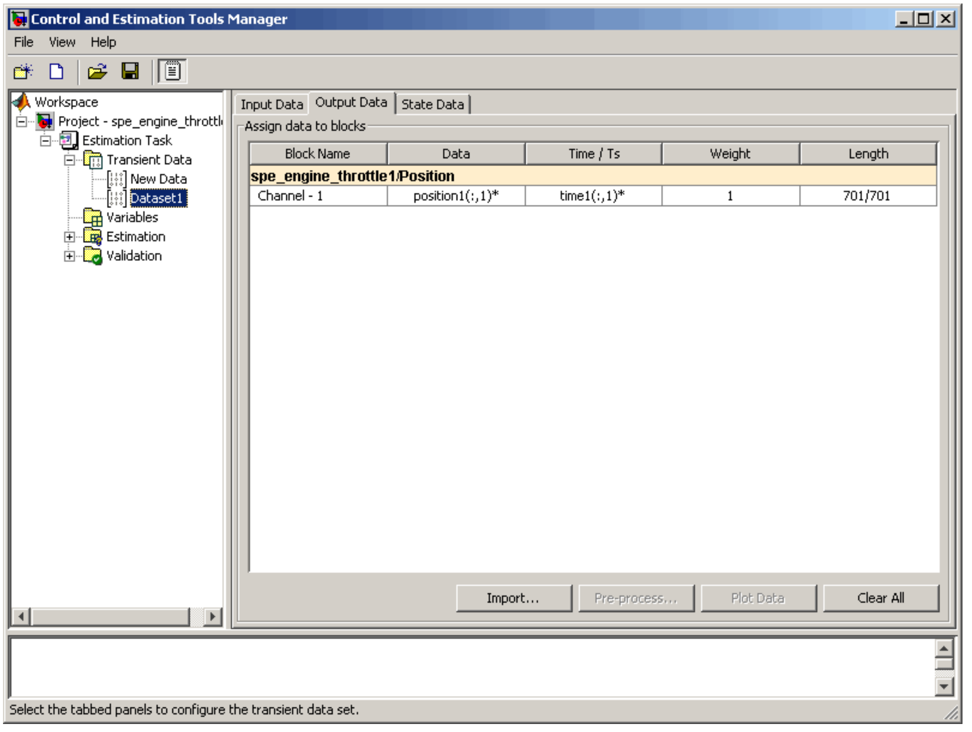
b In the **Exclude X**  field, where **X** corresponds to the time vector, select > from the drop-down list. Enter `0.5` in the adjacent field to specify the upper limit of the data to select for estimation.

The Data Preprocessing Tool GUI resembles the next figure.

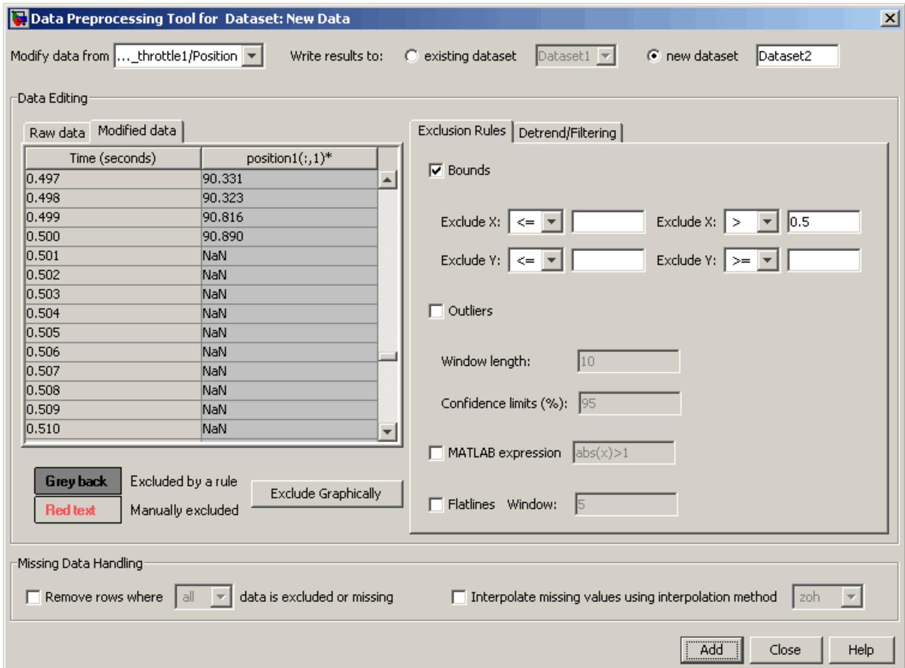


c Click **Add**.

This action adds a new **Dataset1** node under the **Transient Data** node in the Control and Estimation Tools Manager GUI. The **Dataset1** node contains the modified output data `position1(:,1)*` in the **Output Data** tab.



This operation also replaces the output data samples beyond t = 0.5 s in `position1(:,1)*` with `NaN`s. You can view the `NaN`s by selecting the **Modified data** tab in the Data Preprocessing Tool GUI, as shown in the next figure.



4 To remove the `NaN`s:

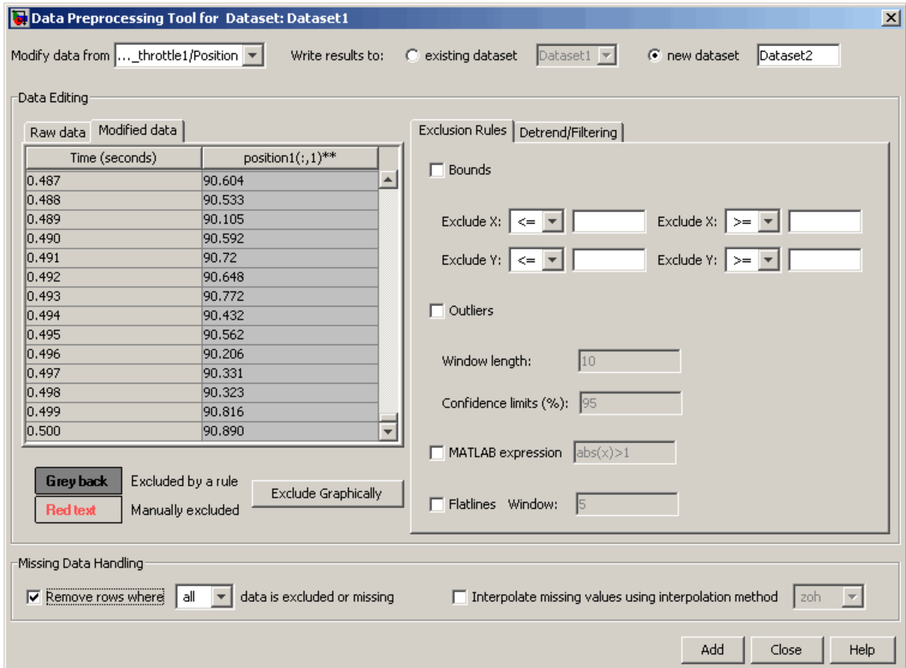
a In the Control and Estimation Tools Manager GUI, select the **Output Data** tab of the **Dataset1** node.

b Select the `position1(:,1)*` cell, and click **Pre-process**.

This action updates the Data Preprocessing Tool GUI with the selected data `position1(:,1)*`.

c In the **Missing Data Handling** area, select the **Remove rows where** 

data is excluded or missing check box.



Tip You can view the results of this operation in the **Modified data** tab.

---

d In the **Write results to** area, select the **existing dataset** 

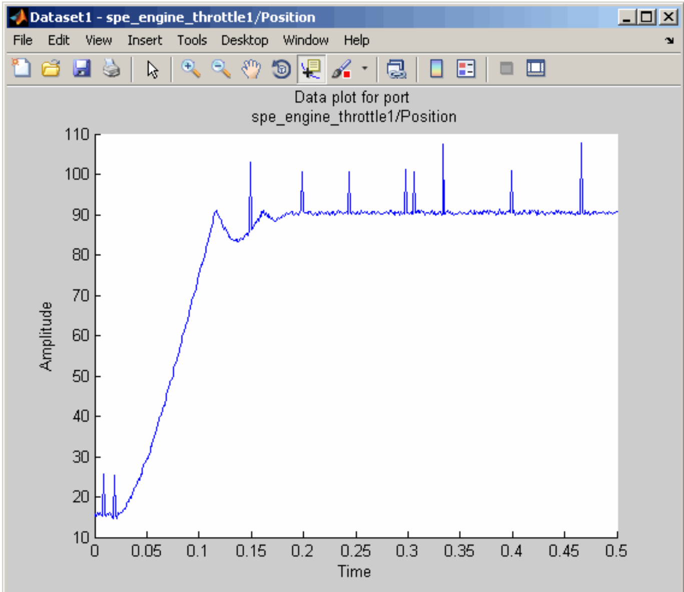
option.

e Click **Add**.

The Update table data dialog box appears. Click **Yes** to overwrite the `position1(:,1)*` data set with the modified data.

5 To plot the data, select the `position1(:,1)*` cell in the **Output Data** tab of the **Dataset1** node, and click **Plot Data**.

The selected output data from t = 0 s to t = 0.5 s is shown in the next figure.



Selecting Input Data

After you select the output data between t = 0 s and t = 0.5 s, as described in "Selecting Output Data" on page 2-25, you must also select the corresponding input data samples. This operation makes the number of I/O data samples equal.

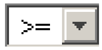
1 In the Control and Estimation Tools Manger GUI, select the **New Data** node under the **Transient Data** node.

2 In the **Input Data** tab, select the `input1(:,1)` cell, and click **Pre-process**.

This action updates the Data Preprocessing Tool GUI with the selected data `input1(:,1)`.

3 To exclude the data beyond t = 0.5 s:

a In the **Exclusion Rules** tab, select the **Bounds** check box.

b In the **Exclude X** 

field, where **X** corresponds to the time vector, select > from the drop-down list. Enter `0.5` in the adjacent field to specify the upper limit of the input data to select for estimation.

c In the **Write results to** area, verify that the **existing dataset** 

option remains selected.

d Click **Add**.

This action adds the modified data `input1(:,1)*` to the **Input Data** tab of the **Dataset1** node. This operation also replaces the input data samples beyond t = 0.5 s in `input1(:,1)*` with `NaN`s.

4 To remove the `NaN`s:

a In the Control and Estimation Tools Manager GUI, select the `input1(:,1)*` cell in the **Input Data** tab of the **Dataset1** node, and click **Pre-process**.

This action updates the Data Preprocessing Tool GUI with the selected data `input1(:,1)*`.

b In the **Missing Data Handling** area, select the **Remove rows where** 

data is excluded or missing check box.

c In the **Write results to** area, verify that the **existing dataset** 

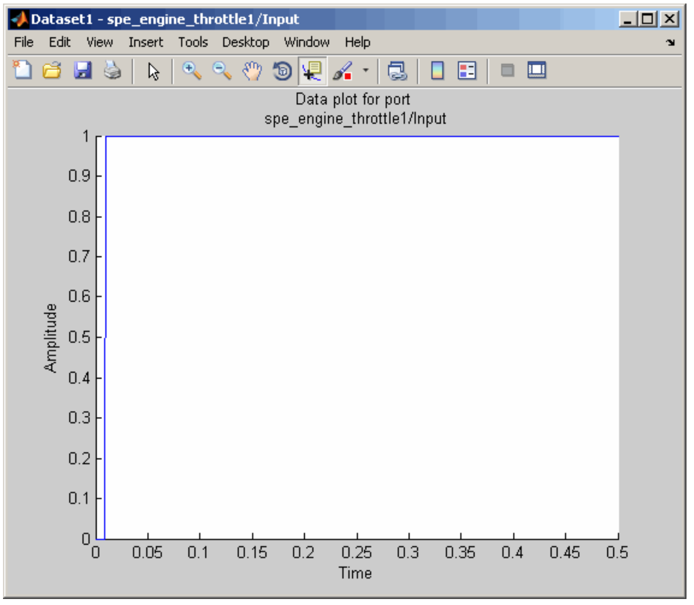
option remains selected.

d Click **Add**.

The Update table data dialog box appears. Click **Yes** to overwrite the `input1(:,1)*` data set with the modified data.

5 To plot the data, select the `input1(:,1)*` cell in the **Input Data** tab of the **Dataset1** node, and click **Plot Data**.

The selected input data from t = 0 s to t = 0.5 s is shown in the next figure.



Removing Outliers

- "Why Remove Outliers" on page 2-33
- "How to Remove Outliers" on page 2-34

Why Remove 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 may not be accurate.

Removing outliers replaces the data samples containing outliers with `NaN`s, which represent missing data. You interpolate the missing data values in a subsequent operation, as described in "Interpolating Missing Data" on page 2-42.

How to Remove Outliers

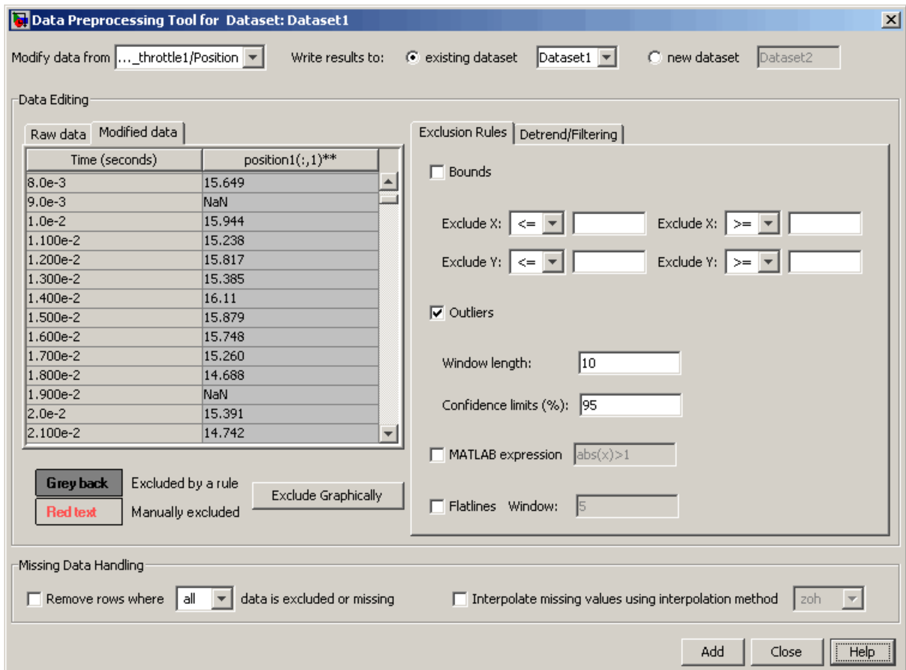
In this portion of the tutorial, you remove outliers from the output data. You must have already selected a subset of the data, as described in "Selecting Data for Estimation" on page 2-25. If you have not done this preparation, click here.

1 In the Control and Estimation Tools Manger, select the `position1(:,1)*` cell in the **Output Data** tab of the **Dataset1** node, and click **Pre-process**.

   This action updates the Data Preprocessing Tool GUI with the selected data `position1(:,1)*`.

2 In the **Exclusion Rules** tab, select the **Outliers** check box.

By default, the **Window length** and **Confidence limits** fields are set to 10 and 95 respectively. The **Window length** field specifies the number of successive data samples the software uses to compute the mean and standard deviation. The **Confidence limits** field specifies the threshold number for identifying outliers. In this example, the mean and standard deviation of 10 successive data samples are computed, and data values that exceed 95% of standard deviation are identified as outliers.



Note The data samples containing outliers are replaced with NaNs.

---

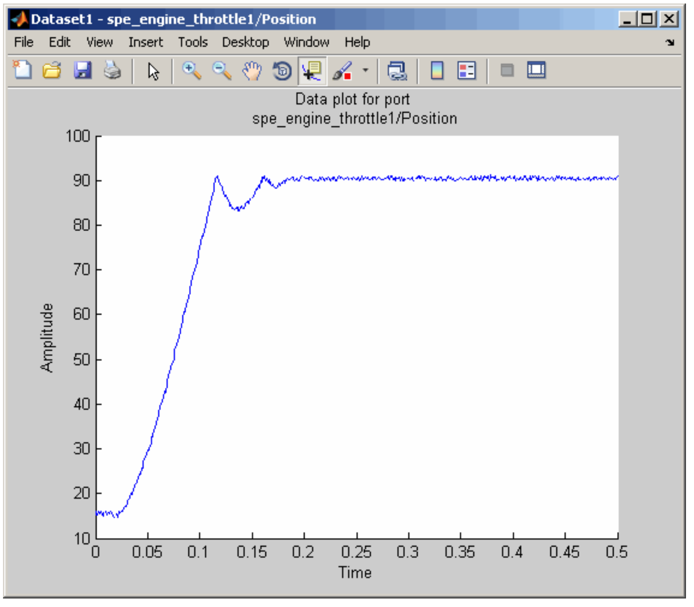
3 In the **Write results to** area, verify that the **existing dataset**  option remains selected.

4 Click **Add**.

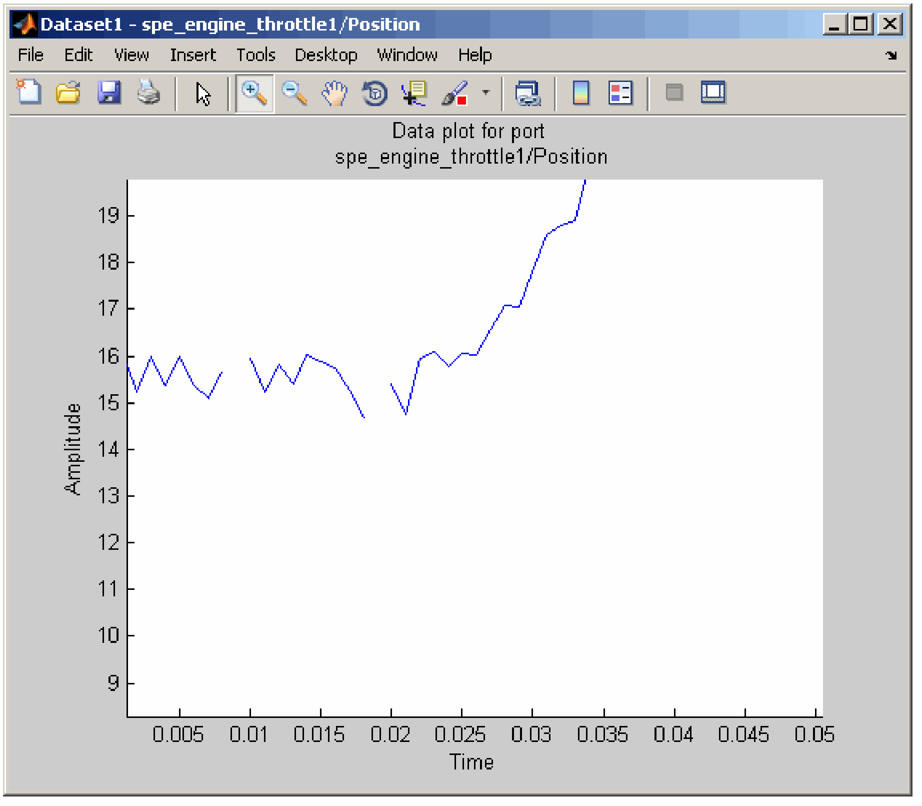
The Update table data dialog box appears. Click **Yes** to overwrite the `position1(:,1)*` data set with the modified data.

5 To plot the data, select the `position1(:,1)*` cell in the **Output Data** tab of the **Dataset1** node, and click **Plot Data**.

The spikes, which indicate outliers, no longer appear on the time plot, as shown in the next figure.



The missing data samples, represented by `NaN`s, appear as gaps on the time plot. To see an example, zoom in to the bottom-left corner of the plot. As shown in the next figure, the data values corresponding to t = 0.009 s and t = 0.019 s are missing.



Filtering Data

- "Filtering Output Data" on page 2-38
- "Filtering Input Data" on page 2-41

Filtering Output Data

In this portion of the tutorial, you filter the noise, and remove any periodic trends from the output data. To avoid relative phase shift between the I/O data, you must also apply the same filter to the input data.

You must have already removed outliers from the output data, as described in "Removing Outliers" on page 2-33. If you have not done this preparation, click here.

1 In the Control and Estimation Tools Manager window, select the `position1(:,1)*` cell in the **Output Data** tab of the **Dataset1** node, and click **Pre-process**.

   This action updates the Data Preprocessing Tool GUI with the selected data `position1(:,1)*`.

2 In the **Detrend/Filtering** tab:

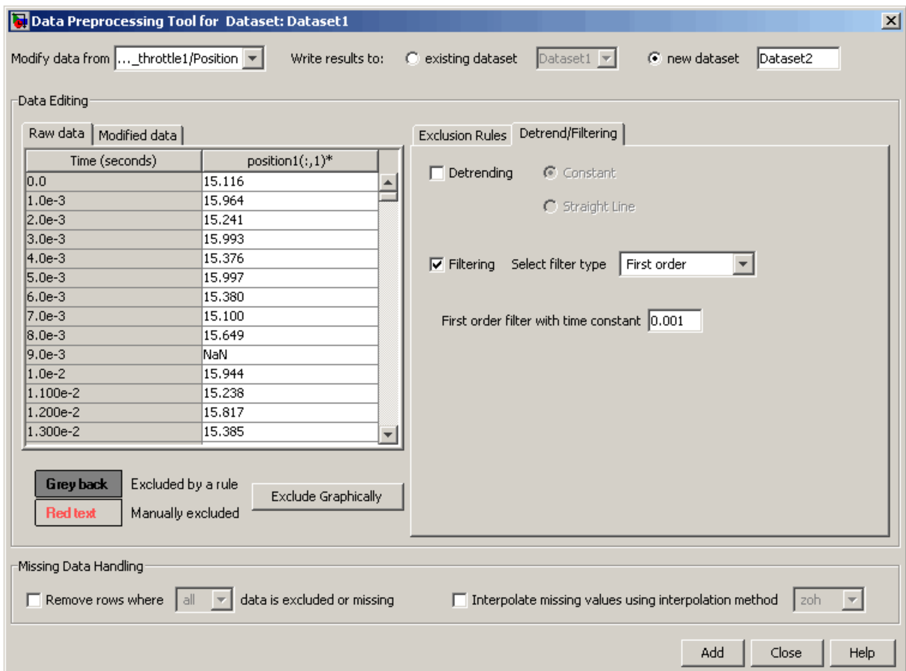
a Select the **Filtering** check box.

By default, `First order` is selected as the filter type. To learn more about the filters, see “Filtering Data”.

b Specify `0.001` in the **First order filter with time constant** field.

This field specifies the time constant for the first-order filter.

Tip For calculating the time constant, you can visually inspect the time plot to determine the frequency components.



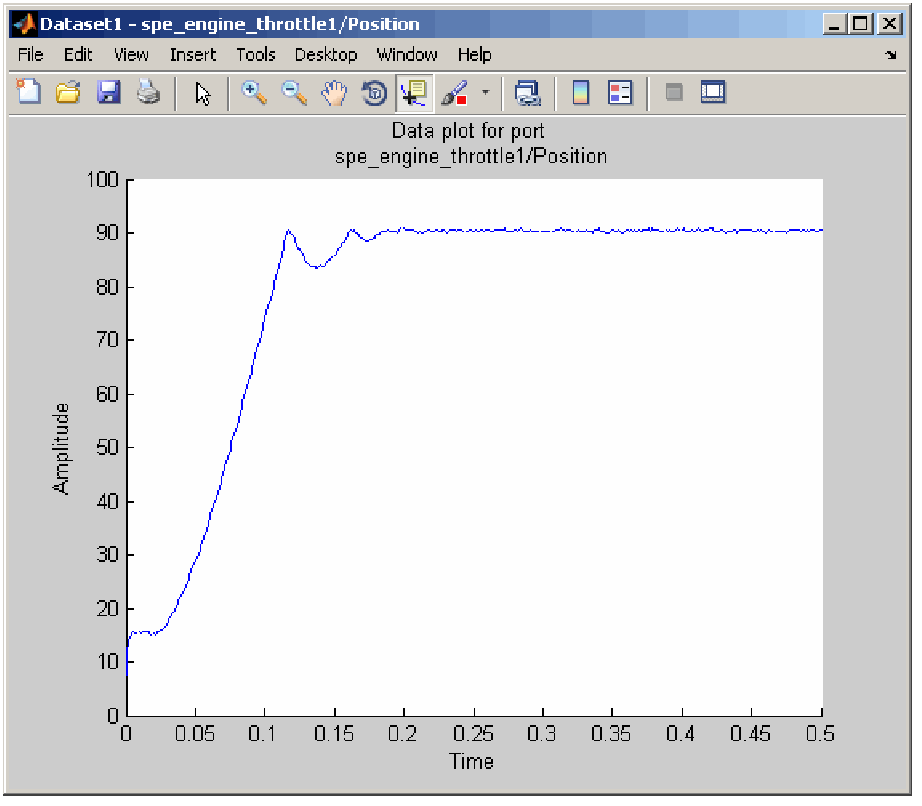
3 In the **Write results to** area, verify that the **existing dataset**  option remains selected.

4 Click **Add**.

The Update table data dialog box appears. Click **Yes** to overwrite the `position1(:,1)*` data set with the modified data.

5 To plot the data, select the `position1(:,1)*` cell in the **Output Data** tab of the **Dataset1** node, and click **Plot Data**.

The noise is filtered and the output data appears smooth, as shown in the next figure.



Filtering Input Data

After you filter the output data, as described in “Filtering Output Data” on page 2-38, you must also filter the input data with the same filter.

1 In the Control and Estimation Tools Manager window, select the `input1(:,1)*` cell in the **Input Data** tab of the **Dataset1** node, and click **Pre-process**.

   This action updates the Data Preprocessing Tool GUI with the selected data `input1(:,1)*`.

2 In the **Detrend/Filtering** tab:

   a Select the **Filtering** check box.

      By default, `First order` is selected as the filter type.

   b Specify `0.001` in the **First order filter with time constant** field.

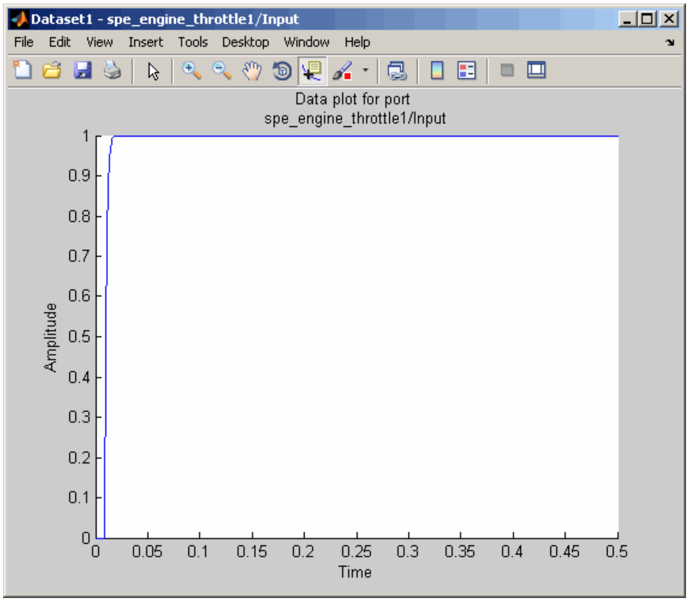
3 In the **Write results to** area, verify that the **existing dataset** 

option remains selected.

4 Click **Add**.

   The Update table data dialog box appears. Click **Yes** to overwrite the `input1(:,1)*` data set with the modified data.

5 To plot the data, select the `input1(:,1)*` cell in the **Input Data** tab of the **Dataset1** node, and click **Plot Data**.



Interpolating Missing Data

In this portion of the tutorial, you interpolate the output data to compute the missing values created when removing outliers, as described in "Removing Outliers" on page 2-33. The interpolation operation uses the known data values to compute the missing data values.

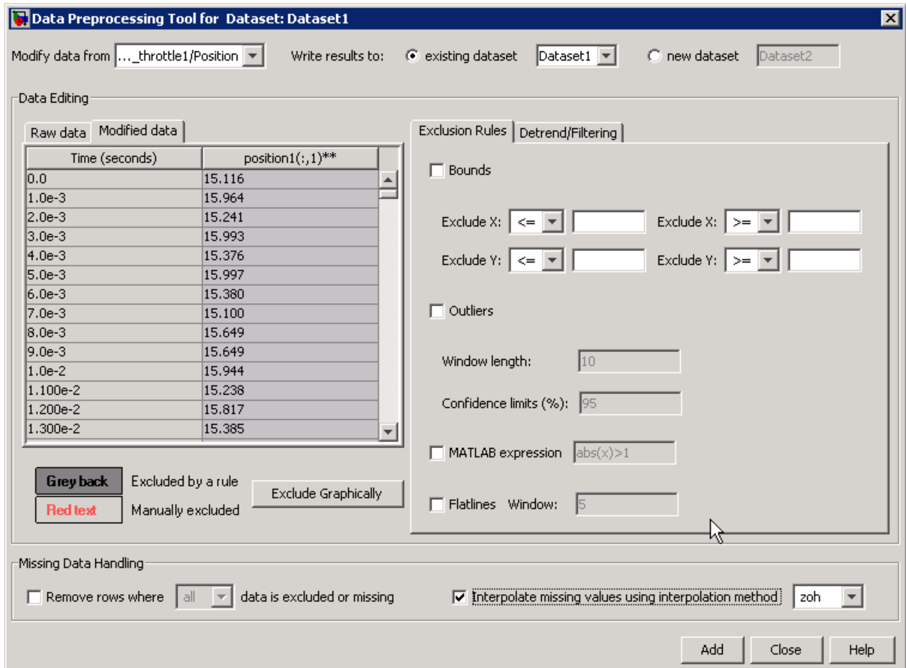
You must have already filtered the noise, as described in "Filtering Data" on page 2-37. If you have not already done this preparation, click here.

1 In the Control and Estimation Tools Manager GUI, select the `position1(:,1)*` cell in the **Output Data** tab of the **Dataset1** node, and click **Pre-process**.

   This action updates the Data Preprocessing Tool GUI with the selected data `position1(:,1)*`.

2 In the **Missing Data Handling** area, select the **Interpolate missing values using interpolation method** check box.

By default, `zoh` is selected as the interpolation method. This method fills the missing data sample with the data value immediately preceding it.



Tip You can view the interpolated data samples in the **Modified data** tab.

---

3 In the **Write results to** area, verify that the **existing dataset** 

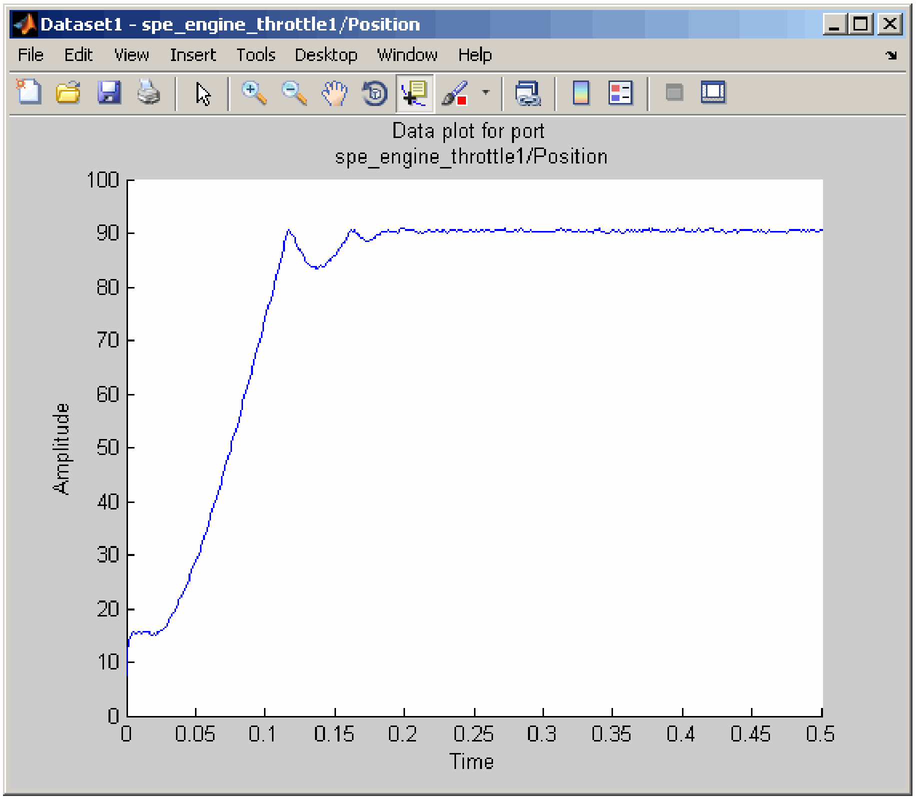
option remains selected.

4 Click **Add**.

The Update table data dialog box appears. Click **Yes** to overwrite the `position1(:,1)*` data set with the modified data.

5 To plot the data, select the `position1(:,1)*` cell in the **Output Data** tab of the **Dataset1** node, and click **Plot Data**.

The new estimation data, prepared by removing outliers, filtering noise, and interpolating missing data, is shown the next figure.



Saving the Project

After you prepare the data, you can delete the data in the **New Data** node, rename the prepared data, and save the session. To skip the data preparation steps, click here.

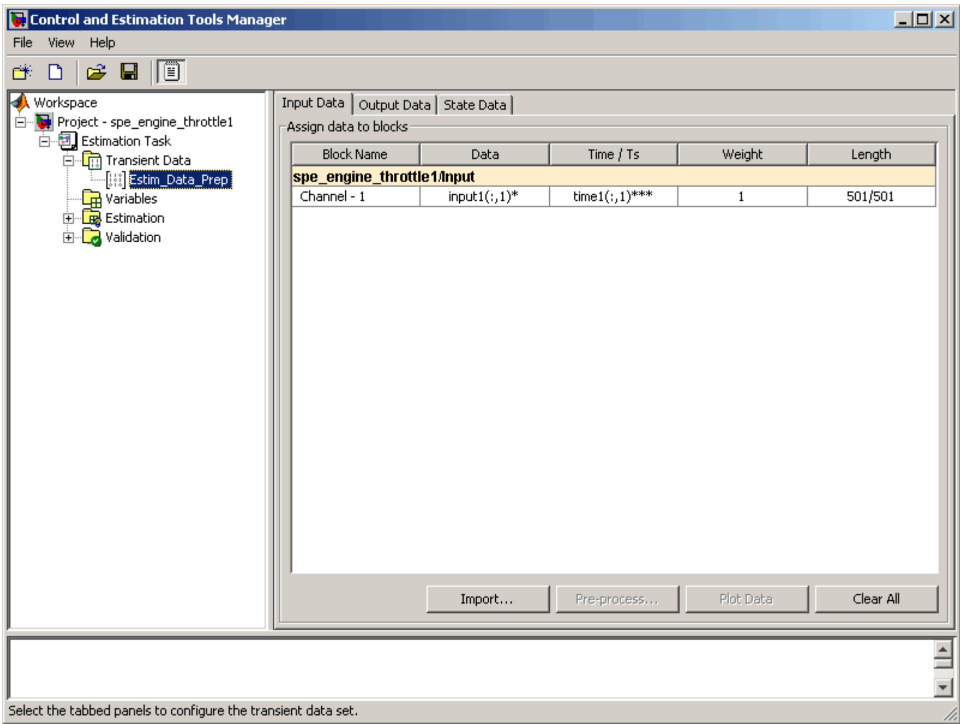
1 In the Control and Estimation Tools Manger GUI, select the **New Data** node under the **Transient Data** node.

2 Right-click the **New Data** node, and select **Delete**.

3 Select the **Dataset1** node under the **Transient Data** node.

4 Right-click the **Dataset1** node, and select **Rename**. Specify `Estim_Data_Prep` as the name of the new estimation data set.

The Control and Estimation Tools Manager GUI resembles the next figure.

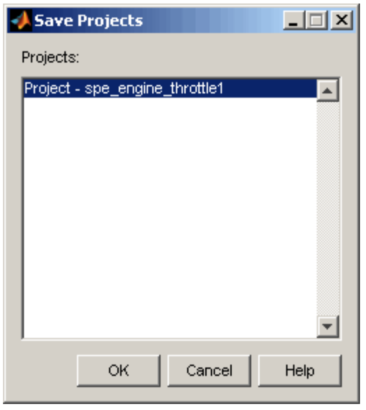


5 Save the Control and Estimation Tools Manager project:

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

This action opens the Save Projects dialog box.

b In the Save Projects dialog box, click **OK**.



c In the Save Projects window, specify `spe_engine_throttle1p.mat` in the **File name** field, and click **Save**.

The action saves the project as a MAT-file.

To learn how to estimate parameters from this data, see "Estimate Parameters from Measured Data Using the GUI" on page 2-47.

Estimate Parameters from Measured Data Using the GUI

In this section...

About This Tutorial

Objectives

In this tutorial, you learn how to estimate parameters of a single-input single-output (SISO) Simulink model from measured input and output (I/O) data.

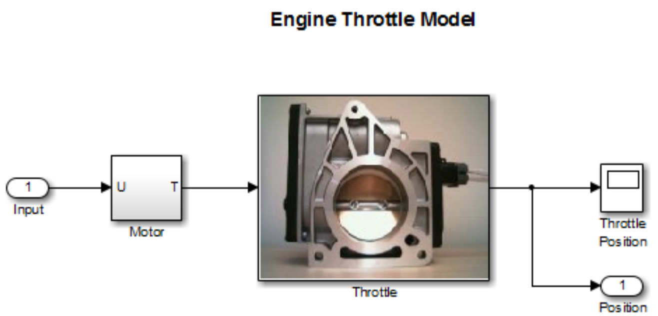
Note Simulink Design Optimization software estimates parameters from real, time-domain data only.

You learn to perform the following tasks using the GUI:

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

About the Model

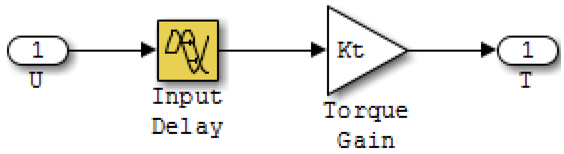
In this tutorial, you use the `spe_engine_throttle1` Simulink model. This model represents an engine throttle system, as shown in the next figure.



The 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 models for these components are described in “Motor Subsystem” on page 2-49, and “Throttle Subsystem” on page 2-49.

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).

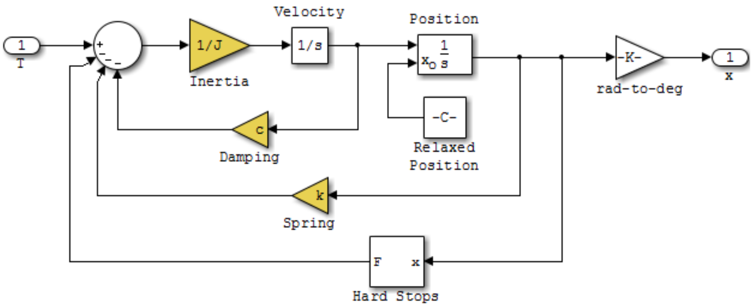
Motor Subsystem. The `Motor` subsystem contains the DC motor model. To open the model, double-click the corresponding block.



The following table describes the variables, parameters, equation, input, and output of the `Motor` subsystem.

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. 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	T

Throttle Subsystem. The `Throttle` subsystem contains the butterfly valve model. To open the model, right-click the corresponding block, and select **Mask > Look Under Mask**.



The `Hard Stops` block models the valve angular position limit of 15° to 90°.

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

Variables	T is the torque applied by the DC motor. θ is the angular position of the valve, represented by `x` in the model. $T_{hardstop}$ is the torque applied by the hard stop.
Parameters	`J` is the inertia. `c` is the viscous friction. `k` is the spring constant.
States	θ 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: $T_{hardstop} = \begin{cases} 0, & 15^\circ \le \theta \le 90^\circ \\ K(90^\circ - \theta), & \theta > 90^\circ \\ K(15^\circ - \theta), & \theta < 15^\circ \end{cases}$ where `K` is the gain of the `Hard Stops` block.
Input	T
Output	θ

Estimating Model Parameters Using Default Estimation Settings

- "Overview of the Estimation Process" on page 2-52
- "Specifying Parameters and Estimation Data" on page 2-53
- "Validating Model Parameters" on page 2-57

Overview of the Estimation Process

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

You perform the following tasks to estimate the model parameters:

- “Specifying Parameters and Estimation Data” on page 2-53
- “Validating Model Parameters” on page 2-57

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

Specifying Parameters and Estimation Data

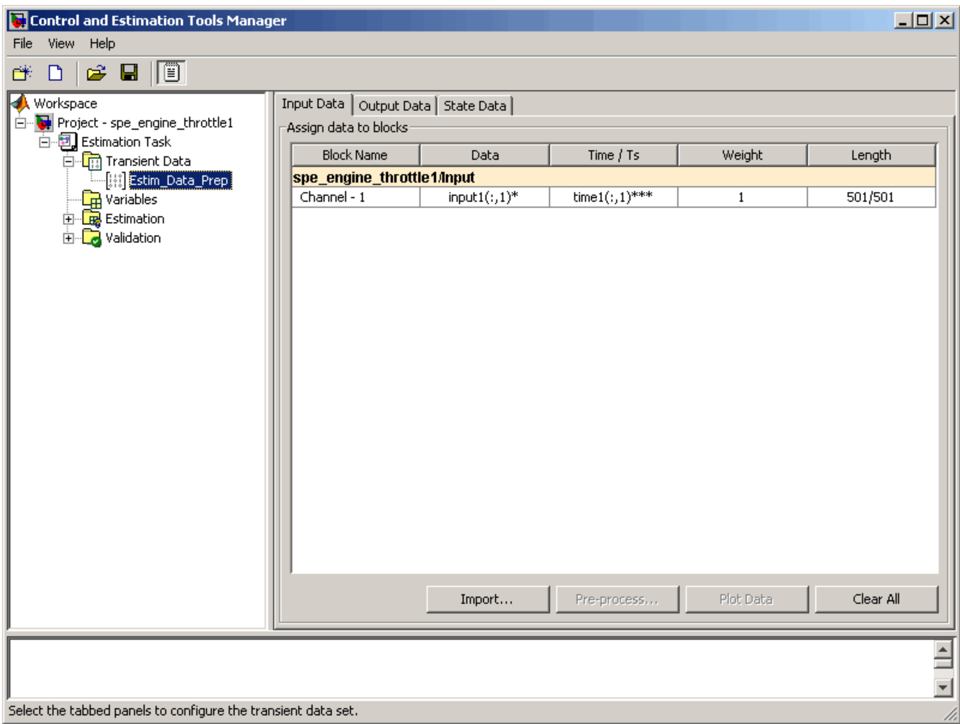
To specify parameters and estimation data:

1. Load the preconfigured project `spe_engine_throttle1p.mat`.

   This MAT-file contains the estimation data. To learn more about the data and how to prepare it, see “Prepare Data for Parameter Estimation Using the GUI” on page 2-11.

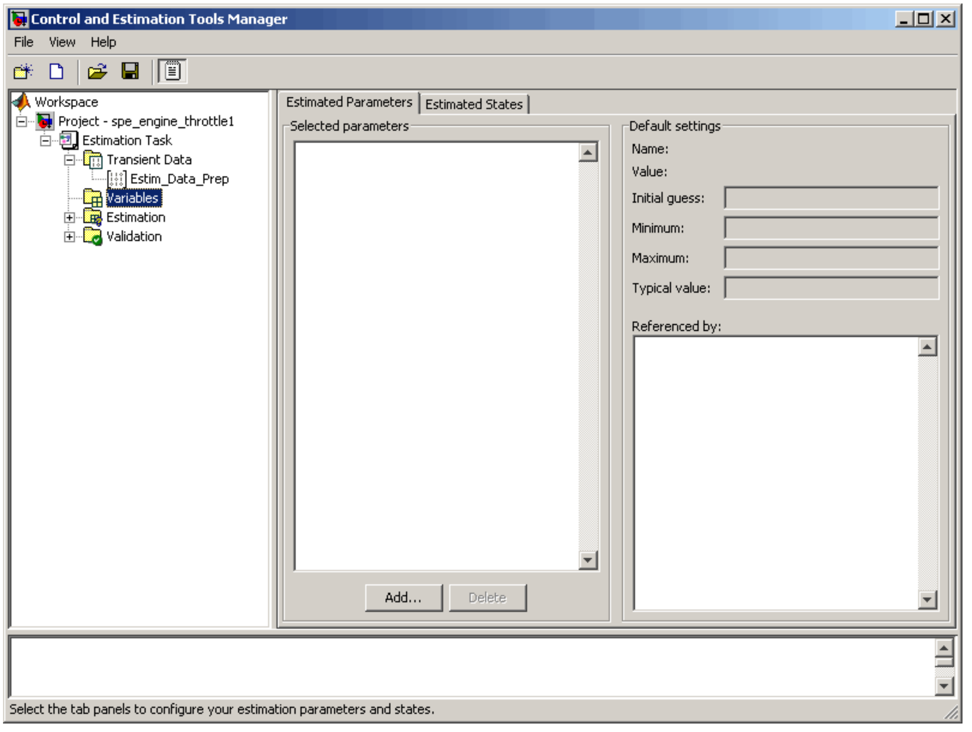
   To load the preconfigured project:

   The Control and Estimation Tools Manager GUI opens as shown in the next figure.



2 Specify parameters for estimation.

a Select the **Variables** node under the **Estimation Task** node.

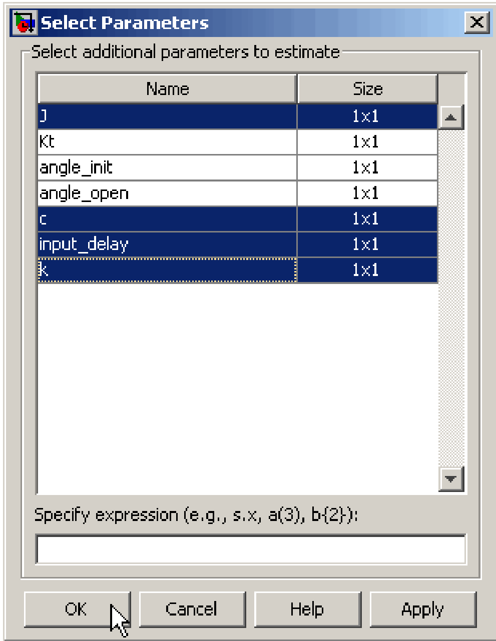


b Click **Add**.

This action opens the Select Parameters dialog box, which shows the model parameters for the Simulink model.

c Select the parameters `J`, `c`, `input_delay`, and `k` by pressing the **Ctrl** key while clicking each name, and then click **OK**.

This action adds the selected parameters to the **Estimated Parameters** tab.



Tip When estimating a large number of parameters, you can select a subset of parameters to estimate. For more information, see the example Inverted Pendulum Parameter Estimation.

---

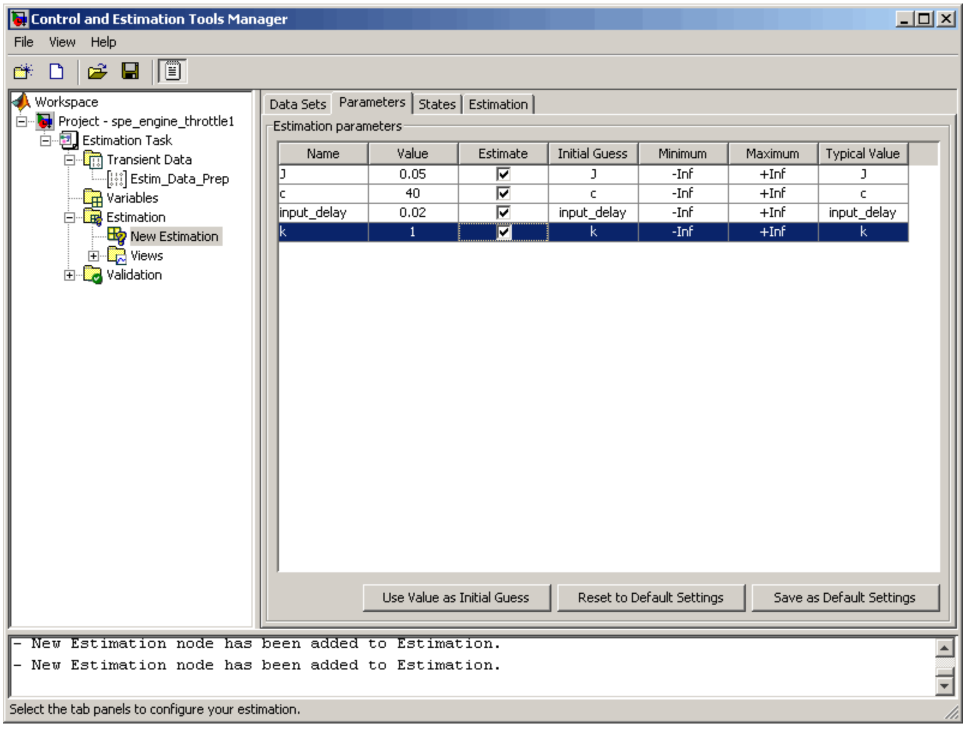
d Select the **Estimation** node, and click **New**.

This action adds a **New Estimation** node under the **Estimation** node.

e Select the **New Estimation** node.

f In the **Parameters** tab, select the **Estimate** check box for all the parameters.

This action specifies the selected parameters for estimation.



The **Parameters** tab, as shown in the previous figure, displays the following information for each parameter:

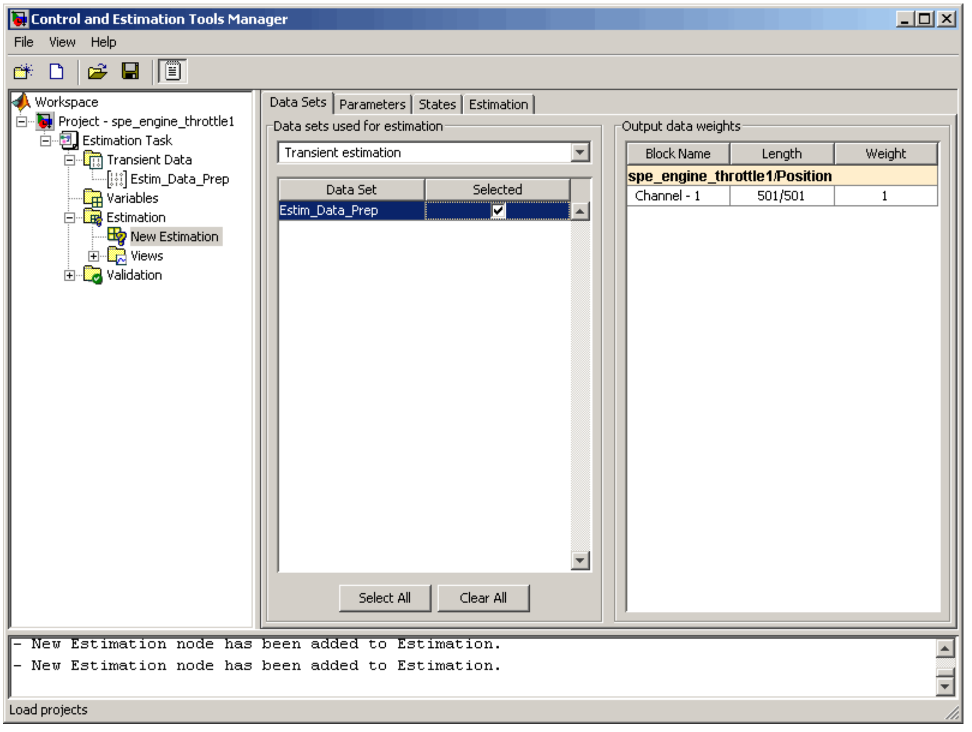
- **Value**: Current parameter value. By default, it is equal to the value specified in the Simulink model.

  During estimation, the optimization method might change a parameter value to minimize the error between the measured and simulated output.

- **Initial Guess**: Initial parameter value. By default, it is equal to the value in the **Value** field and is used at the start of estimation.

- **Minimum** and **Maximum**: Bounds on the parameter value. By default, they are set to `-Inf`, and `+Inf`, respectively. This means that the optimization method searches for a solution in the range `[-Inf, +Inf]`.
- **Typical Value**: Order of magnitude of the parameter value. By default, it is equal to the value in the **Initial Guess** field.

3 Specify data for estimation by checking the **Selected** check box for **Estim_Data_Prep** in the **Data Sets** tab.



Validating Model Parameters

After you specify parameters and data for estimation, as described in "Specifying Parameters and Estimation Data" on page 2-53, estimate the parameters and analyze the results to determine if the model needs to be refined.

1 Create plots to view the estimation results.

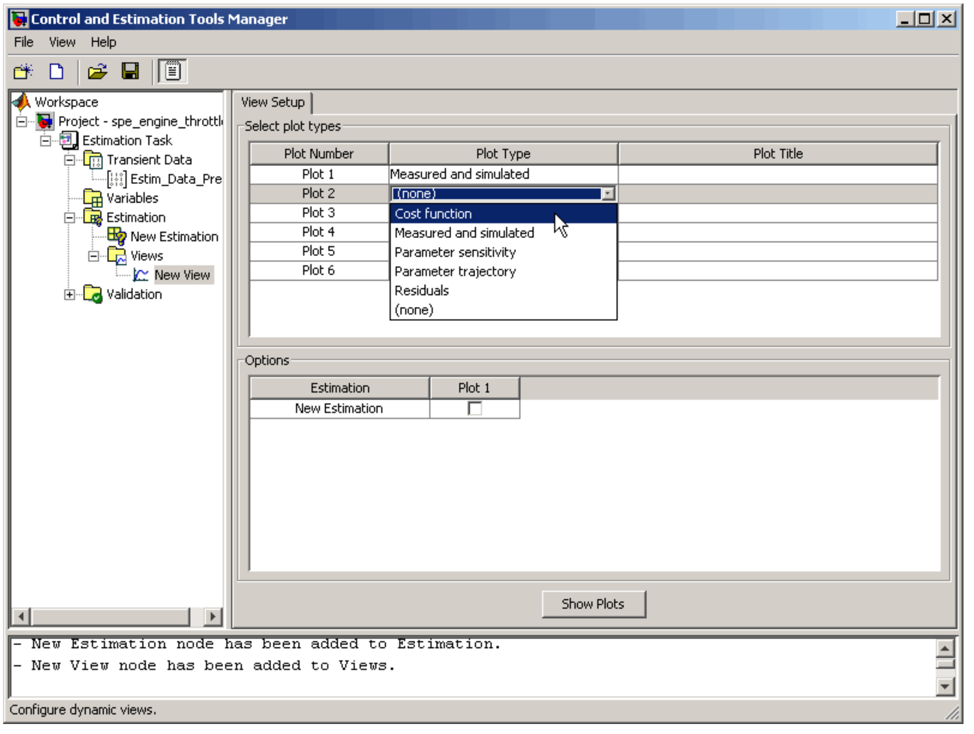
a Select the **Views** node under the **Estimation** node, and click **New**.

This action creates a **New View** node under the **Views** node.

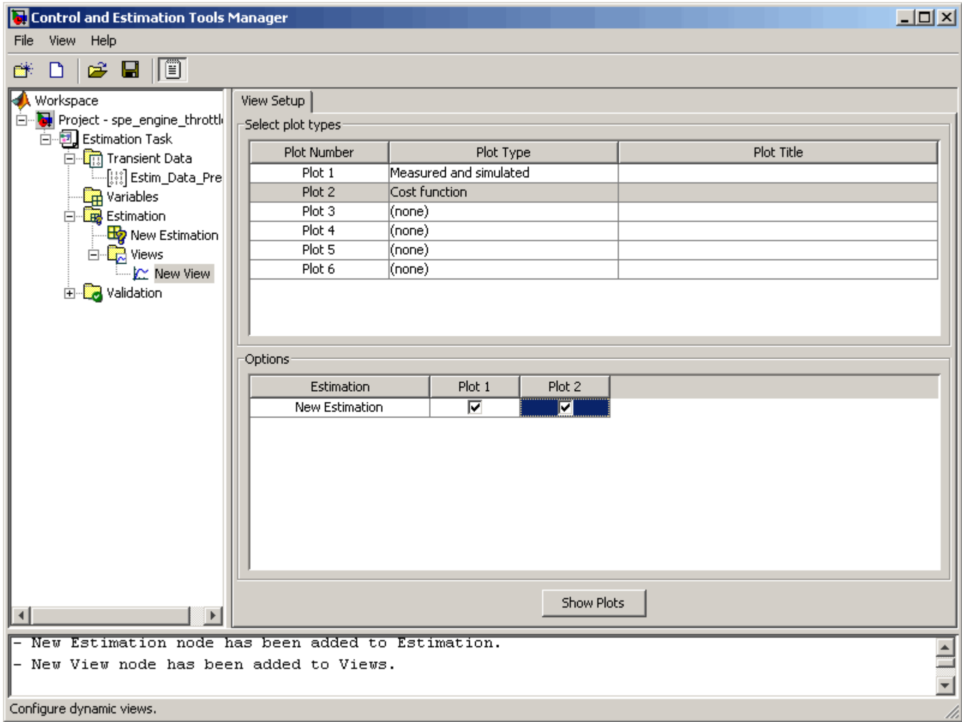
b Select the **New View** node.

c In the **View Setup** tab, select the following plots from the drop-down list in the **Plot Type** column:

- For **Plot 1**, select `Measured and simulated`.
- For **Plot 2**, select `Cost function`.



d In the **Options** area, select the **Plot 1** and **Plot 2** check boxes, and click **Show Plots**.

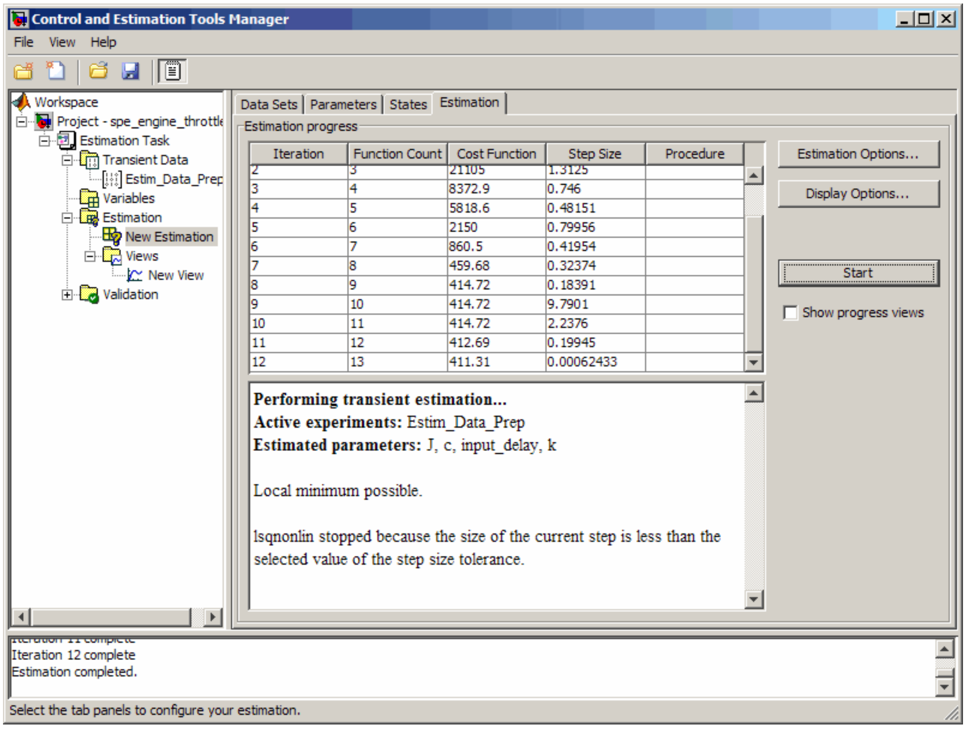


Clicking **Show Plots** opens the following figures:

- `New View - Plot 1 (Measured and simulated)`: Displays the measured output data `Estim_Prep_Data`. During estimation, this plot updates to display the simulated response at each iteration.
- `New View - Plot 2 (Cost function)`: Displays the error between the measured and simulated output, computed at each iteration. By default, the error is computed using *sum- of- squared- errors*, which is a least-squares method.

2 Estimate the parameters by selecting the **Estimation** tab of the **New Estimation** node, and clicking **Start**.

The **Estimation** tab updates at each iteration, and provides information about the estimation progress. When the estimation completes, the **Estimation** tab resembles the next figure.



Note The results of the optimization may differ slightly because of different numerical precision across platforms.

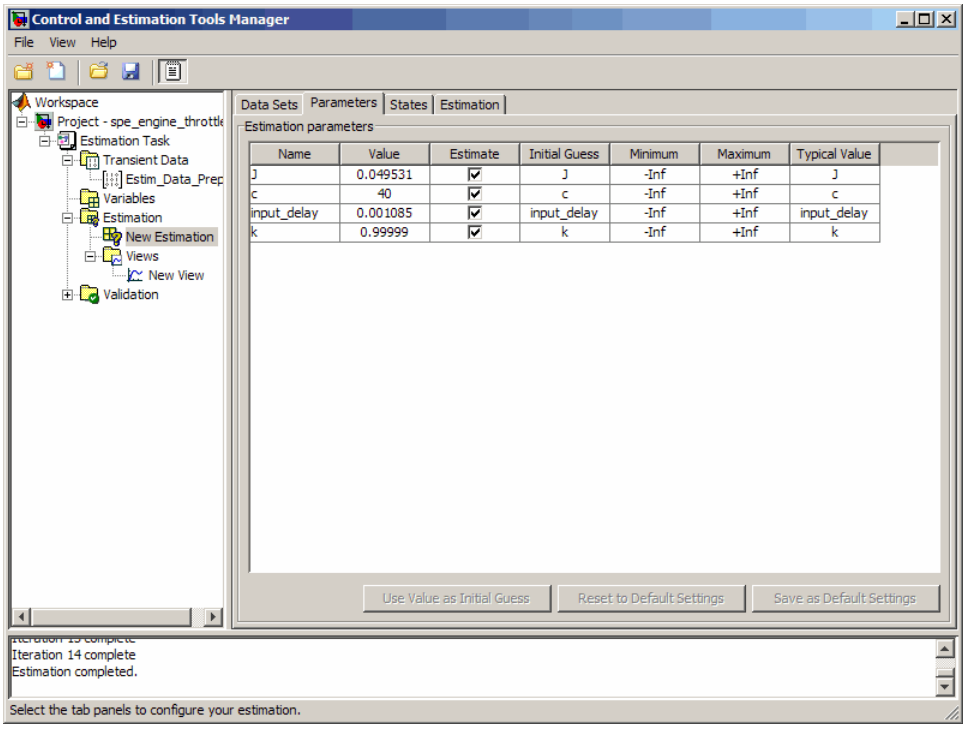
The table displays the following information for each iteration:

- **Cost Function**: 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. As shown in the previous figure, the cost function decreases by only 27% from 2.69e5 to 1.95e5.

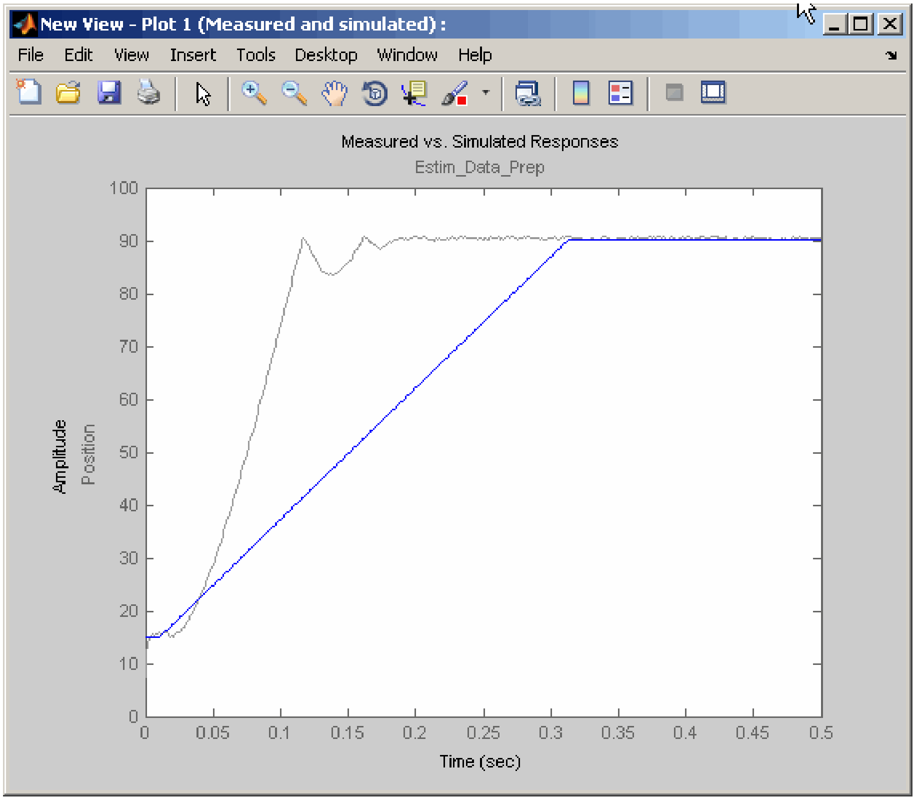
- **Step Size**: Displacement of the optimization method in the current search direction when minimizing the cost function. A final value close to zero indicates that the method has found a local minimum. As shown in the previous figure, the final step size in the last iteration is 5.96e-7.

3 View the estimated parameter values in the **Value** column of the **Parameters** tab.



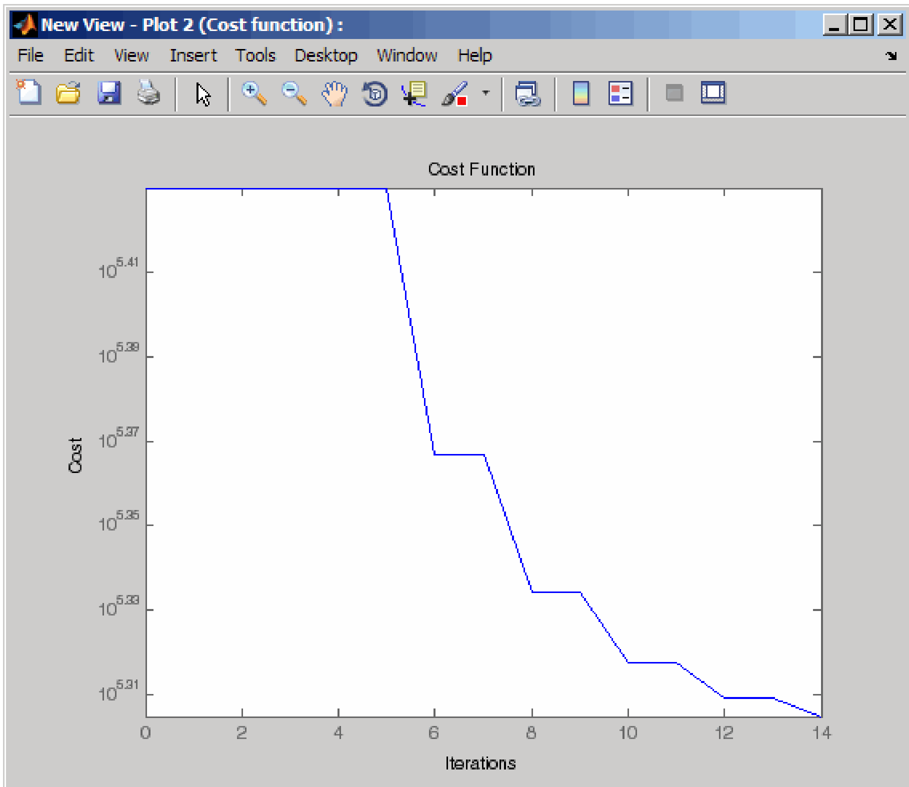
4 Examine the simulated response plot to see how well the simulated output matches the measured output.

The simulated response with the estimated parameters is shown in blue on the `New View - Plot 1 (Measured and simulated)` plot, and significantly differs from the measured data. This difference indicates that the estimated parameters are not accurate.



5 Examine the cost function plot to see how the cost function changes during the estimation.

The cost function values shown in step 2 are plotted on the `New View - Plot 2(Cost function)` plot. The cost function decreases by only 27% from its initial value, and in conjunction with the measured and simulated output plot, indicates that the estimated parameters are not accurate.



To improve the estimation results, you apply bounds on the parameter values, as described in "Improving Estimation Results Using Parameter Bounds" on page 2-64.

Improving Estimation Results Using Parameter Bounds

Strategy for Improving the Estimation Results

Analyzing the `Measured and Simulated` and `Cost function` plots, as described in "Validating Model Parameters" on page 2-57, you see that the model parameters estimated using the default estimation settings are not accurate. You must run additional estimations to improve the accuracy of the model. There are several techniques to improve the accuracy of the estimated parameters. For more information, see "How to Specify Estimation Options in the GUI".

In this portion of the tutorial, you improve the results by specifying bounds on parameter values. This technique restricts the region in which the optimization method searches for a local minima.

Based on physical insight, you know the following characteristics of the engine throttle system:

- All the parameter values are positive.
- Maximum time delay of the system, represented by `input_delay`, is 0.1 s.

Therefore, you specify 0 as the minimum value for all parameters, and 0.1 as the maximum value of `input_delay`.

After you estimate the parameters, analyze the results using the `Measured and Simulated` and `Cost function` plots.

How to Specify Parameter Bounds

You must have already estimated the parameters using default settings, as described in "Estimating Model Parameters Using Default Estimation Settings" on page 2-51.

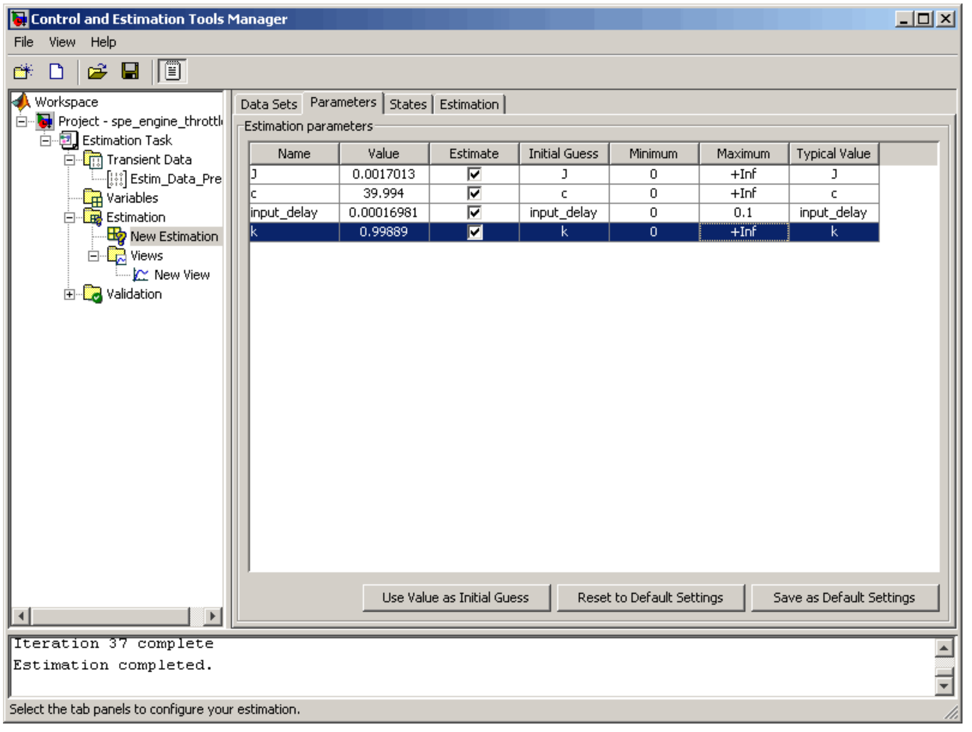
To improve the estimation results by specifying parameter bounds:

1 Select the **Parameters** tab of the **New Estimation** node.

2 Specify the minimum value for each parameter by double-clicking the corresponding **Minimum** cell, and replacing `-Inf` with `0`.

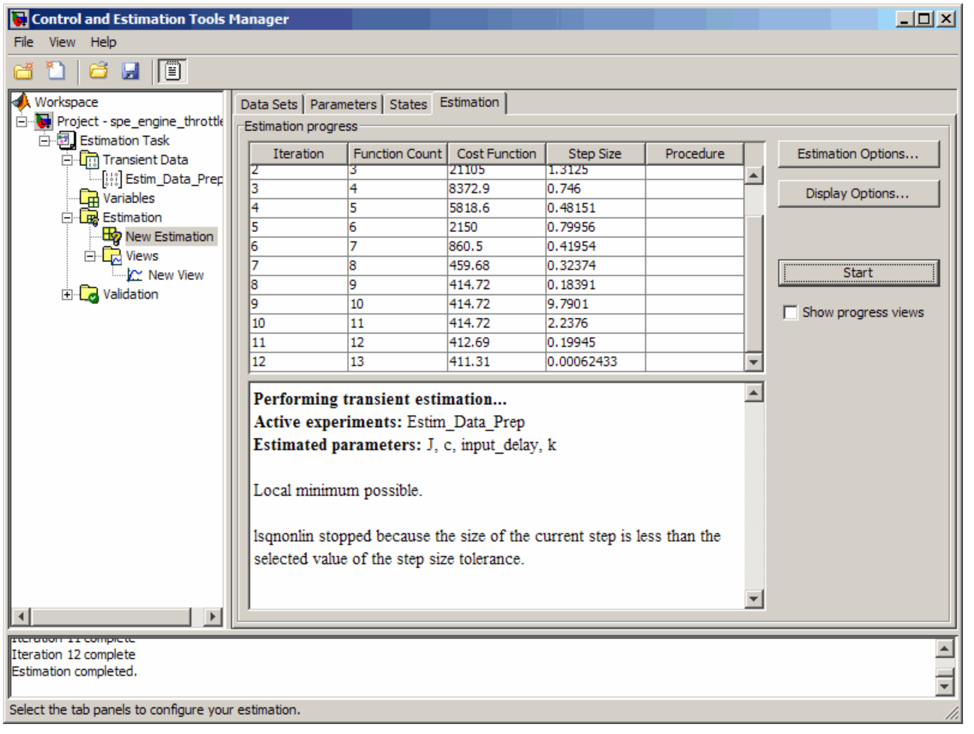
3 Specify the maximum value for `input_delay` by double-clicking the corresponding **Maximum** cell and replacing `+Inf` with `0.1`.

The **Parameters** tab resembles the next figure.



4 Select the **Estimation** tab, and click **Start**.

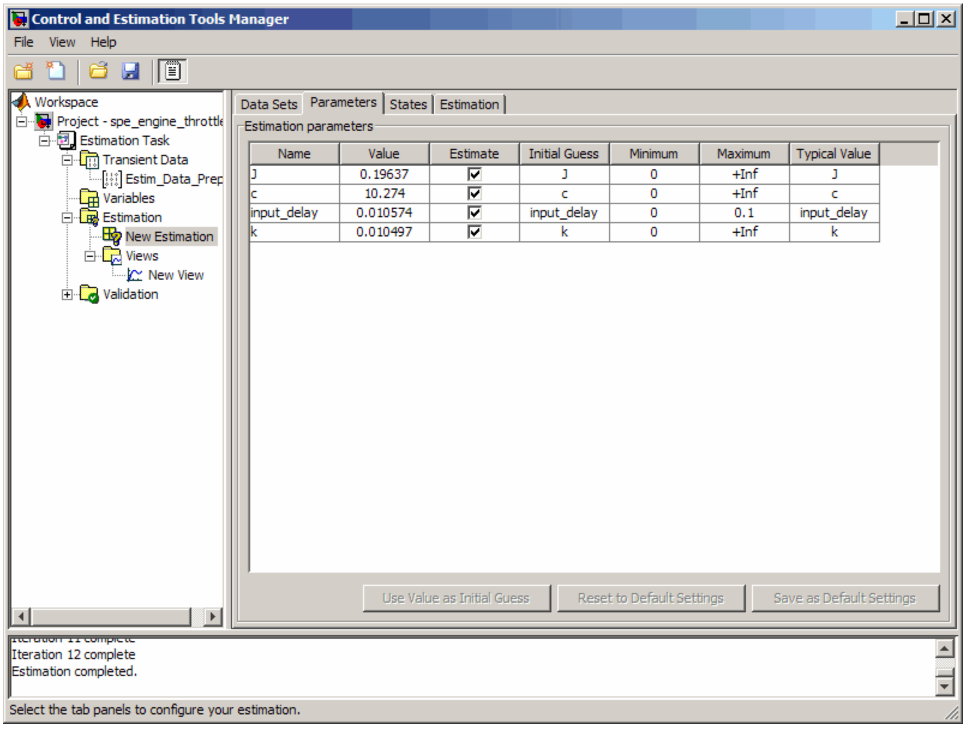
The **Estimation** tab updates at each iteration, and provides information about the estimation progress. When the estimation completes, the **Estimation** tab resembles the next figure.



Note The results of the optimization may differ slightly because of different numerical precision across platforms.

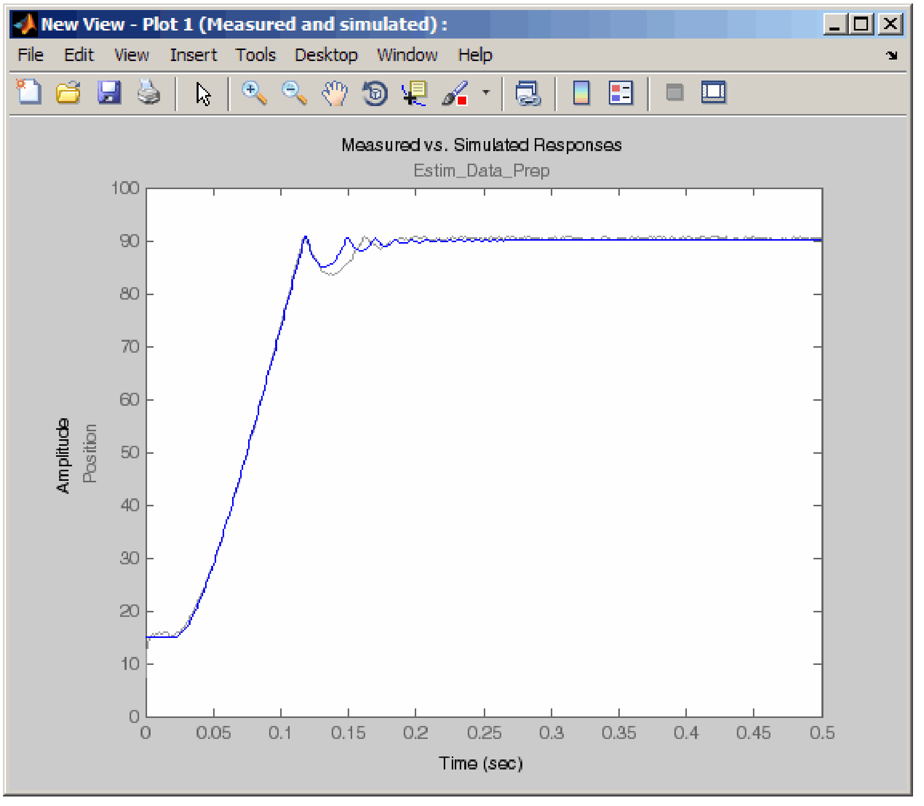
The cost function decreases by 99% from 1.95e5 to 239.94. The final step size 8.75e-7 is close to 0, which indicates that the optimization method has found a local minimum.

5 View the estimated parameter values in the **Value** column of the **Parameters** tab.



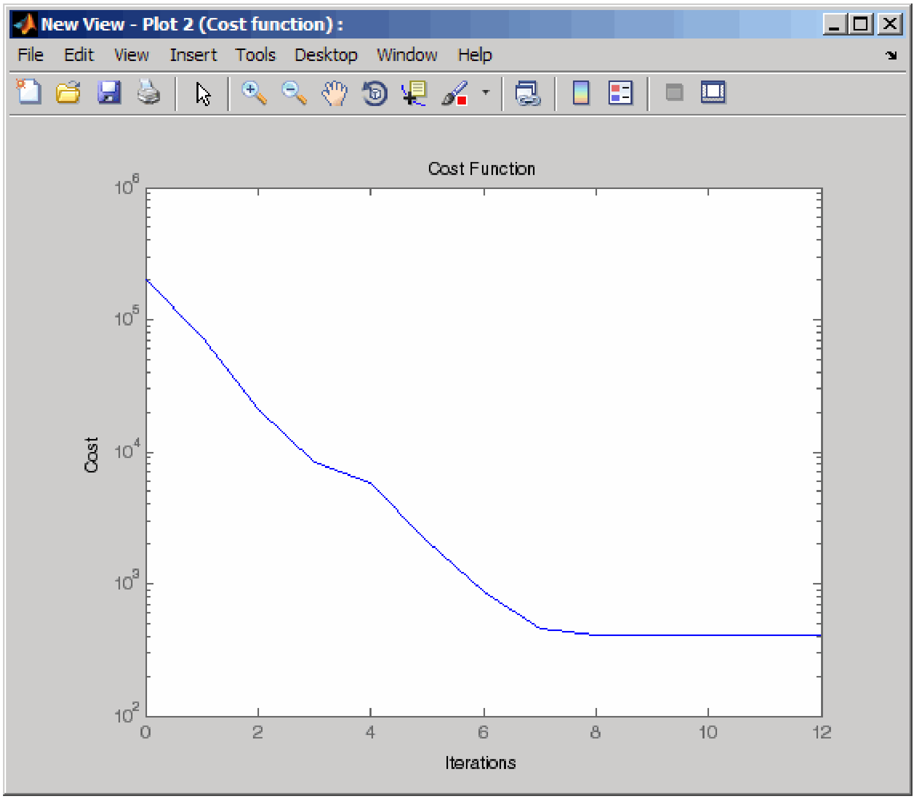
6 Examine the simulated response plot to see how well the simulated output matches the measured output.

The simulated response plot, shown in blue on the `New View - Plot 1 (Measured and simulated)` plot, is overlaid on the measured output data. The simulated output closely matches the measured data.



7 Examine the cost function plot to see how the cost function changes during the estimation.

The cost function values shown in step 4 are plotted on the `New View - Plot 2(Cost function)` plot. The cost function also decreases by 99% from its initial value, and in conjunction with the measured and simulated response plot indicates a good fit of the simulated response with the measured data.



Validating Estimated Model Parameters

After you estimate the model parameters, as described in "Improving Estimation Results Using Parameter Bounds" on page 2-64, 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:

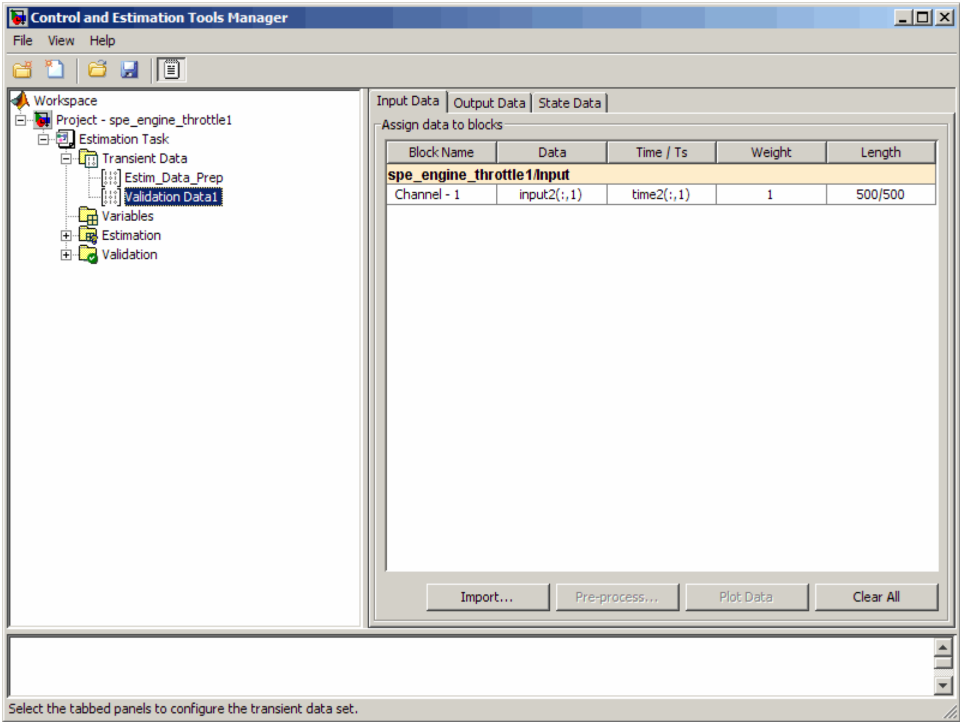
1 Load the preconfigured project `spe_engine_throttle1_importValData1.mat`.

   This MAT-file contains a validation data set already imported into the GUI, and the estimated parameters as described in "Improving Estimation Results Using Parameter Bounds" on page 2-64.

   Tip To learn how to import data and time vector into the Control and Estimation Tools Manager GUI, see "Importing Data into the GUI" on page 2-14.

   To load the preconfigured project:

   The Control and Estimation Tools Manager GUI now has a node that contains the validation data, as shown in the next figure.



The validation data contains the input data, output data and time vector in the MATLAB variables `input2`, `position2` and `time2` respectively.

2 Plot the measured and simulated response, and residuals.

a Select the **Validation** node, and click **New**.

This action creates a **New Validation** node.

b Select the **New Validation** node.

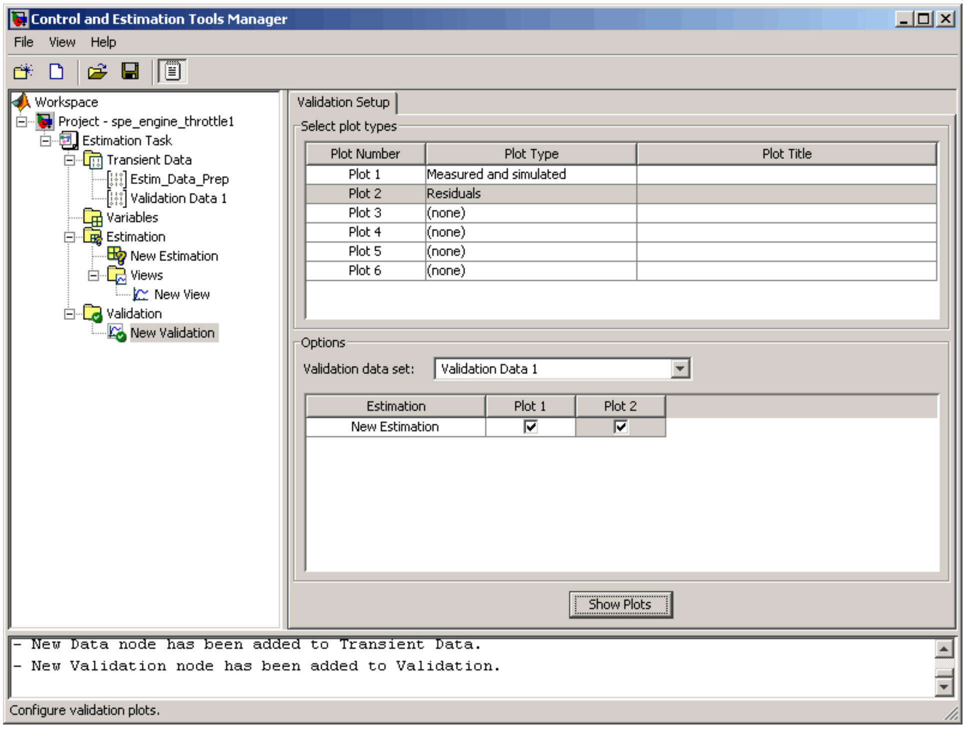
c In the **Validation Setup** tab, select the following plots in the **Plot Type** column.

- For **Plot 1**, select `Measured and simulated`.
- For **Plot 2**, select `Residuals`.

d In the **Options** area, select the **Plot 1**, and **Plot 2** check boxes.

e Select `Validation Data1` from the **Validation data set** drop-down list.

The **Validation Setup** tab resembles the next figure.



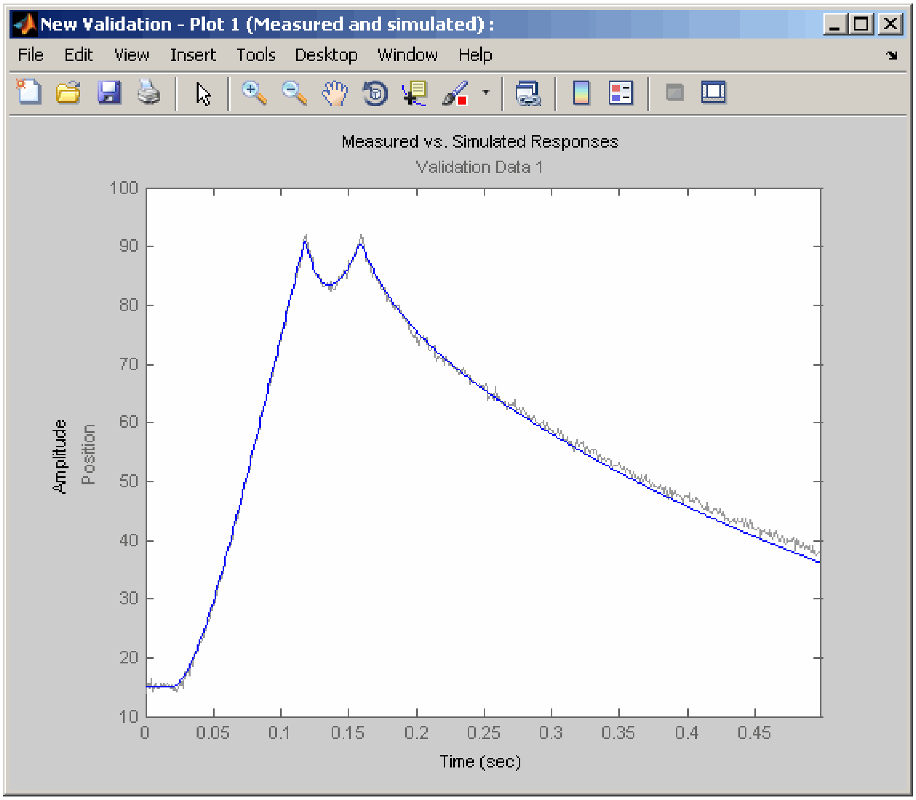
f Click **Show Plots**.

This action opens the following figures:

- `New Validation - Plot 1 (Measured and simulated)`: Displays the validation data and simulated response.
- `New Validation - Plot 2(Residuals)`: Displays the difference between the measured data and simulated response.

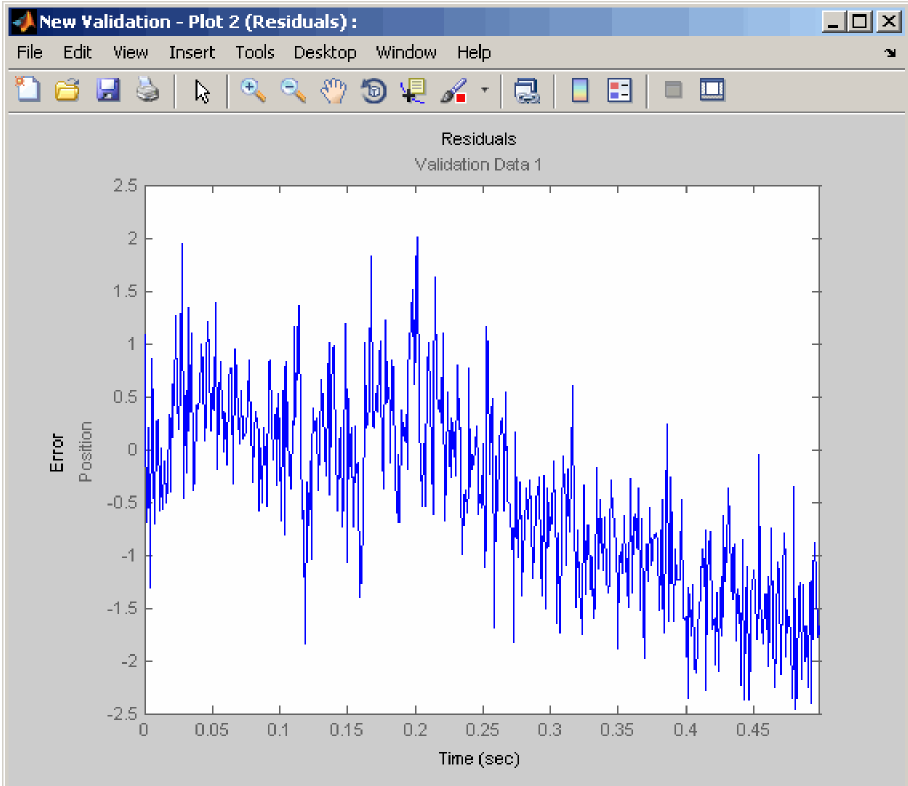
3 Examine the simulated response plot to see how well the simulated output matches the validation data.

The simulated response, as shown in blue on the `New Validation - Plot 1 (Measured and simulated)` plot, is overlaid on the measured output data, and closely matches the measured data `Validation Data 1`.



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

The difference between the simulated and measured data, as shown in the `New Validation - Plot 2(Residuals)` plot, 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.



5 Save the Control and Estimation Tools Manager project.

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

This action opens the Save Projects dialog box.

b In the Save Projects dialog box, click **OK**.

c In the Save Projects window, specify the name of the project in the **File name** field, and click **Save**.

This action saves the project as a MAT-file.

Tip You can load a project by selecting **File > Load** in the Control and Estimation Tools Manager GUI. To view the estimated parameter values, select the **Parameters** tab of **New Estimation** node.

Response Optimization

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

“Optimize Controller Parameters to Meet Step Response Requirements (GUI)” on page 3-4

“Optimize Controller Parameters to Meet Step Response Requirements (Code)” on page 3-18

“Optimize Controller Parameters to Track Reference Signal (GUI)” on page 3-26

“Design Optimization to Meet Frequency-Domain Requirements”

“Design Optimization to Meet Frequency-Domain Requirements (GUI, with Check Block)” on page 3-41

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

More About

“How the Optimization Algorithm Formulates Minimization Problems”

“Specifying Time Domain Design Requirements (GUI)”

“Specifying Frequency Domain Design Requirements (GUI)”

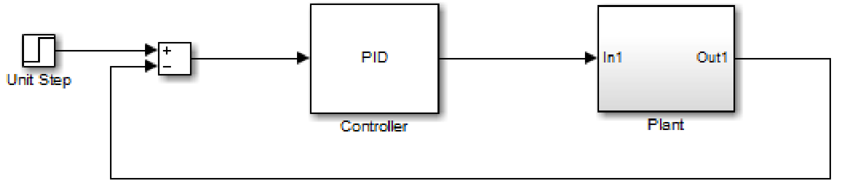
Optimize Controller Parameters to Meet Step Response Requirements (GUI)

In this section...

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.

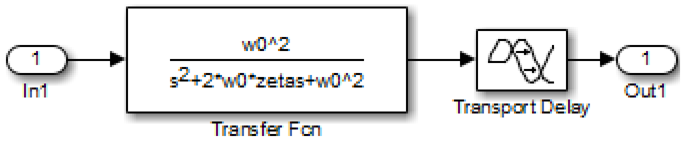
Model Structure

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

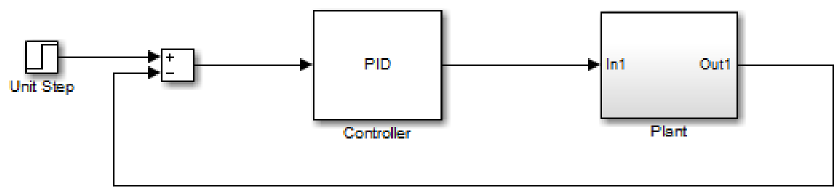
The plant output must meet the following step response requirements:

- 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);
```

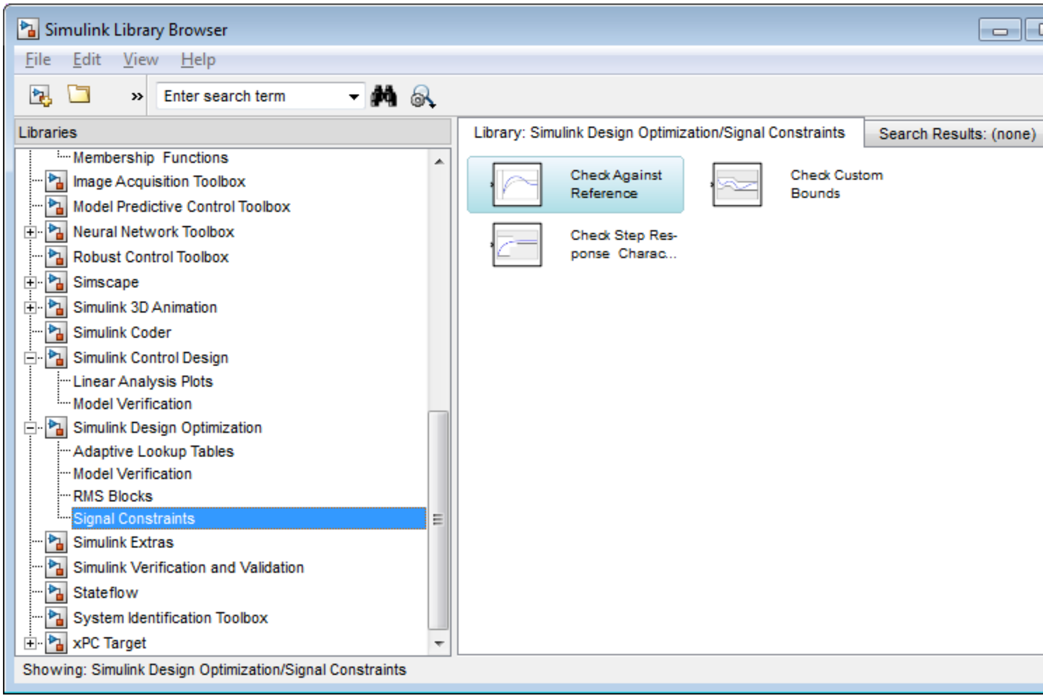


To learn more about the model, see “Model Structure” on page 3-4.

2 Add a Check Step Response Characteristics block to the model.

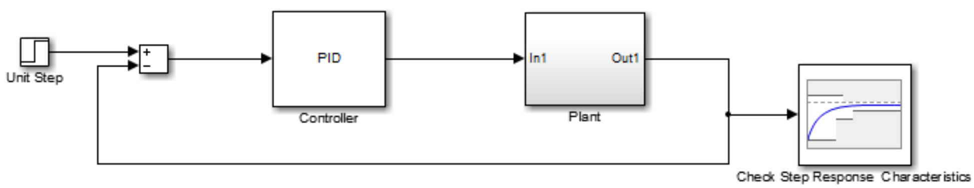
a In the Simulink model window, select **View > Library Browser**.

b In the **Libraries** pane, expand the **Simulink Design Optimization** node and select **Signal Constraints**.



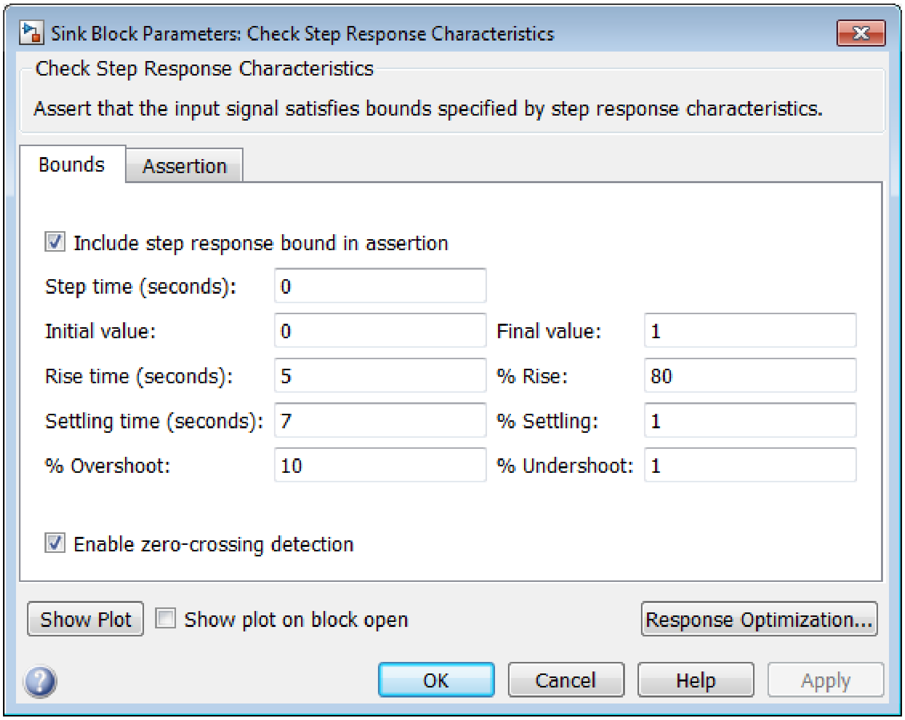
c Drag and drop the Check Step Response Characteristics block into the model window.

d Connect the block to the output.



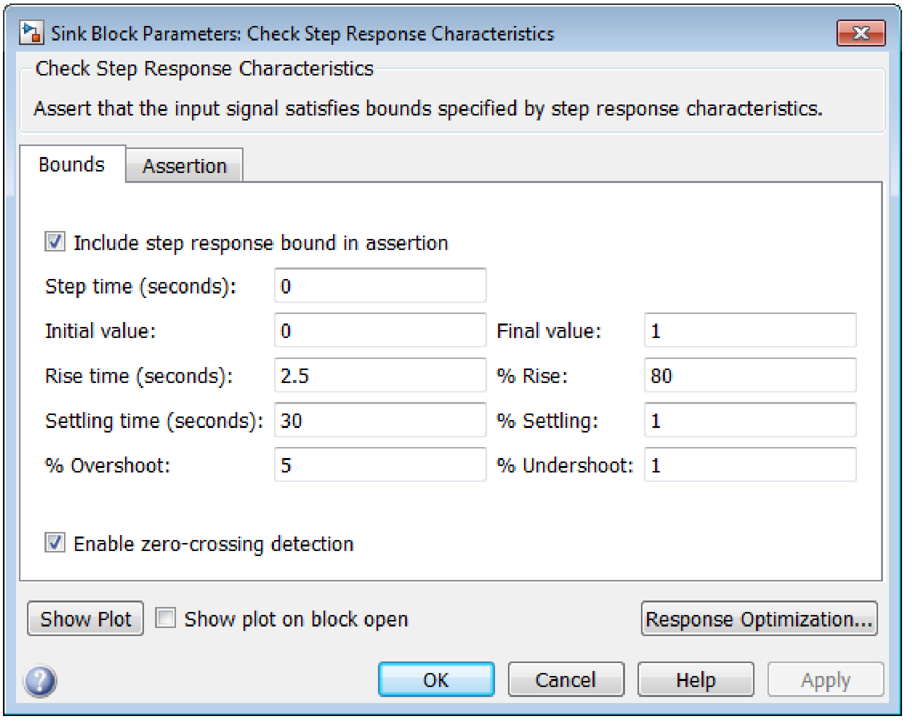
You must connect the block to the signal on which you want to specify design requirements.

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



4 Specify step response requirements:

- In **Rise time (seconds)**, enter `2.5`
- In **Settling time (seconds)**, enter `30`
- In **% Overshoot**, enter `5`



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 3-27.

Specify Design Variables

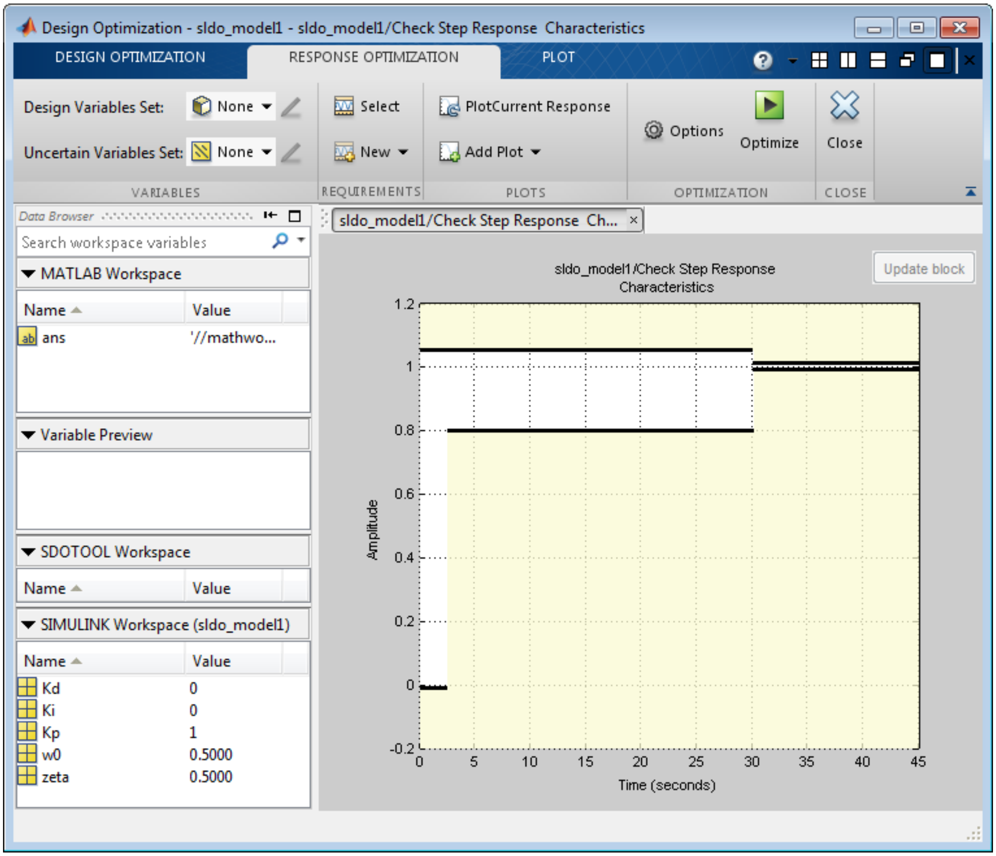
Before you begin this task, specify the design requirements as described in “Specify Step Response Requirements” on page 3-5.

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 **Tools > Response Optimization**.

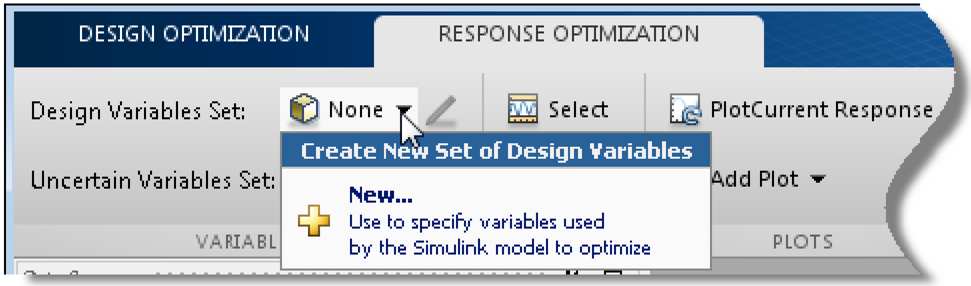
Alternatively, click **Response Optimization** in the Block Parameters dialog box.

A Design Optimization tool for the model opens.

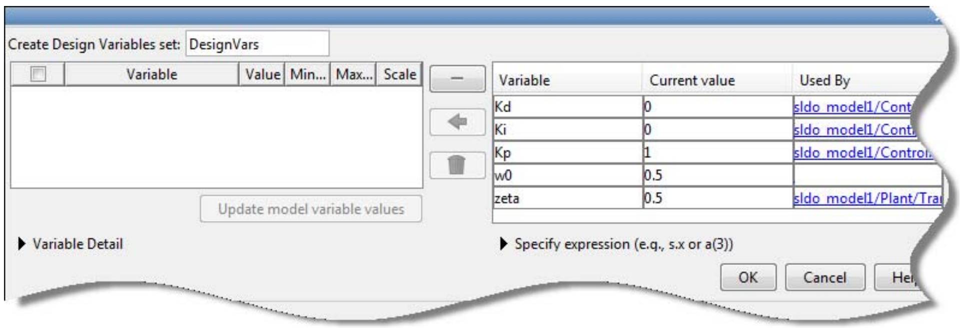


The amplitude versus time plot graphically shows the step response requirements specified in the Check Step Response Characteristics block.

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

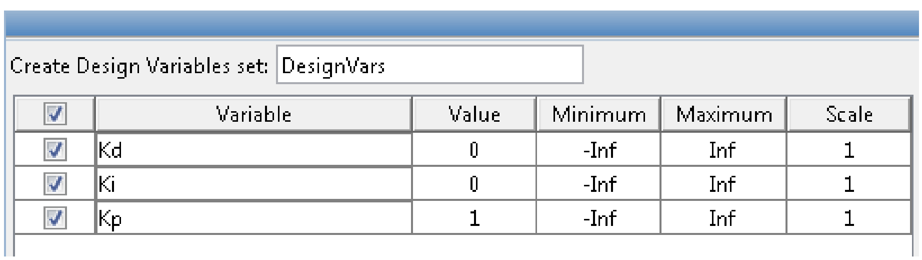


A window opens where you specify design variables.



3 Click `Kd`, `Ki` and `Kp` to select them.

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



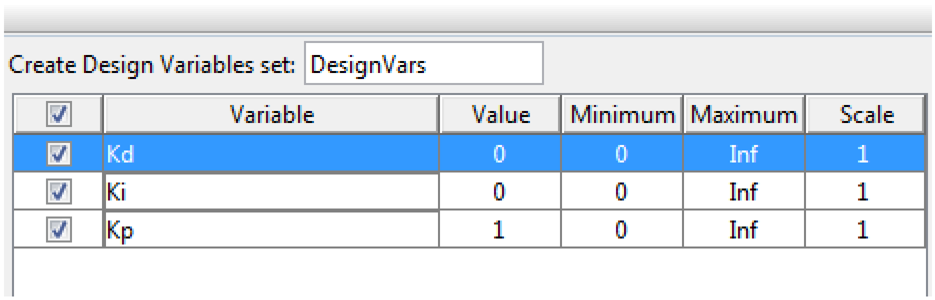
The software displays the following parameter settings:

- **Variable** — Parameter name

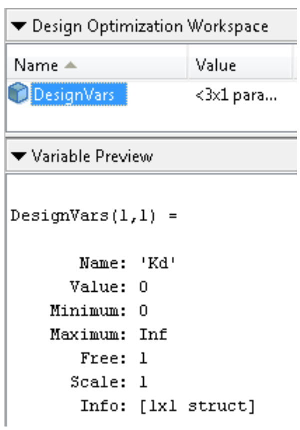
- **Value** — Current parameter value
- **Minimum** and **Maximum** — Parameter bounds
- **Scale** — Scaling factor for the parameter

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**.



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.

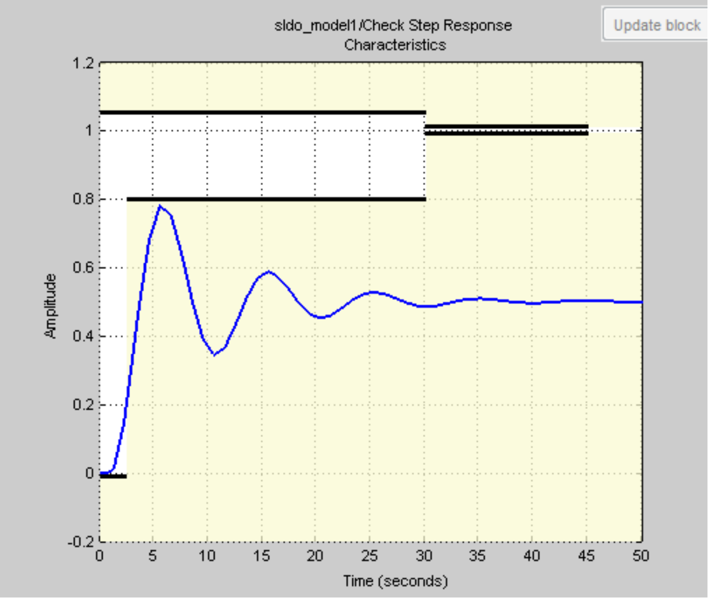


Optimize Model Response

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.

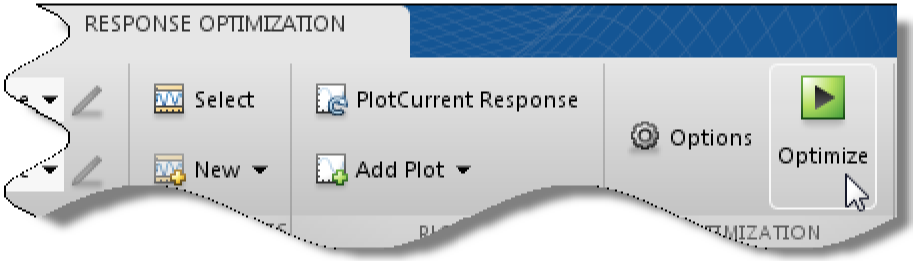
1 (Optional) View the current response of the model. Click

.

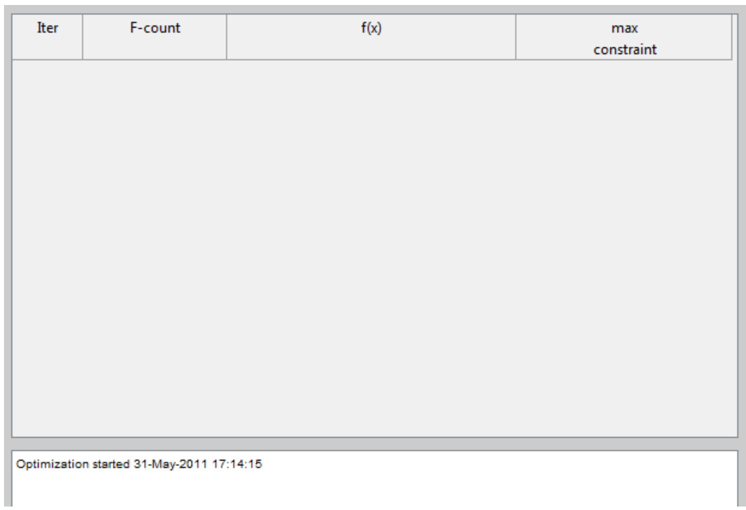


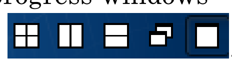
The plot shows that the model output does not meet the specified step response requirements.

2 Click **Optimize**.



An optimization progress window opens.



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.

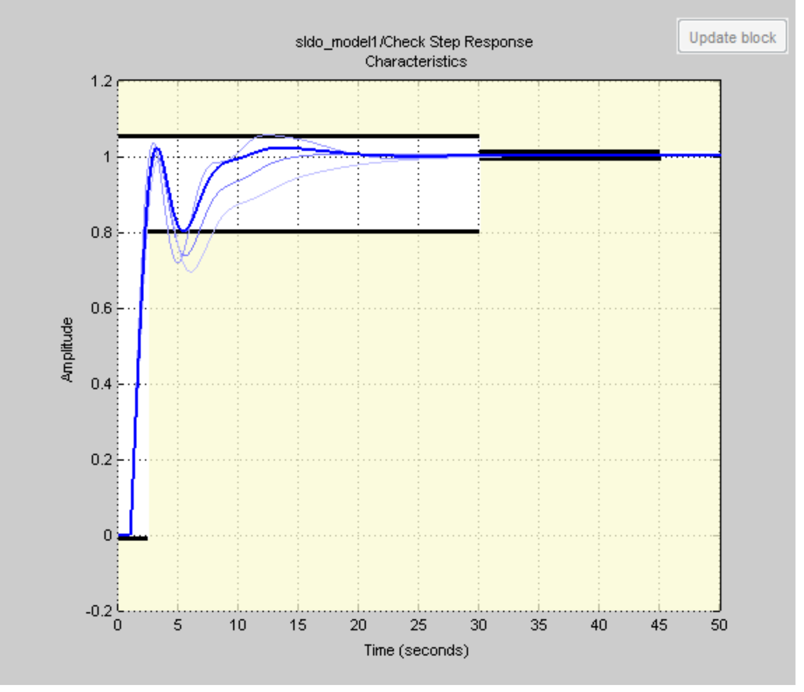
Iter	F-count	f(x)	max constraint
0	5	0	431.9566
1	12	0	264.3424
2	19	0	6.3727
3	26	0	1.0278
4	33	0	0.1333
5	40	0	0.1049
6	48	0	0.0805
7	55	0	0.0030
8	62	0	-1.1153e-06

Optimization started 31-May-2011 16:55:07
Optimization converged, 31-May-2011 17:03:36

Optimized variable values written to 'DesignVars' in SDOTOOL workspace
'sldo_model1' updated with optimized values

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.

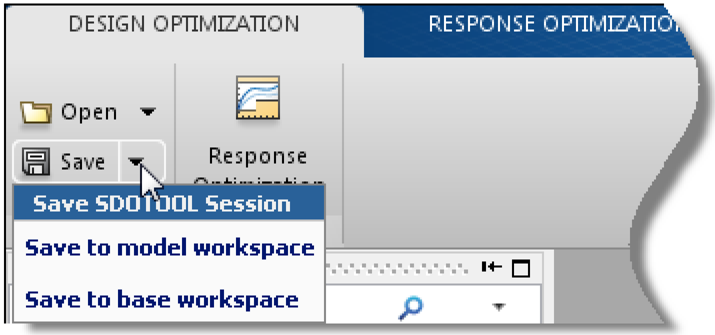
The optimized values of the design variables are automatically updated in the Simulink model.

Save the Session

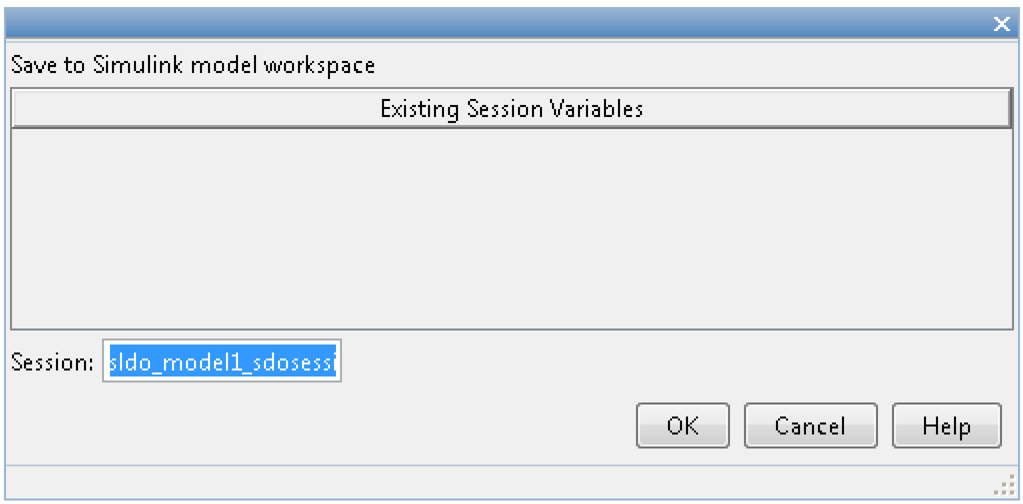
After you optimize the model response to meet design requirements, you can save the Design Optimization tool session which includes the optimized parameter values.

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.



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

- "Optimize Controller Parameters to Meet Step Response Requirements (Code)" on page 3-18
- "Optimize Controller Parameters to Track Reference Signal (GUI)" on page 3-26

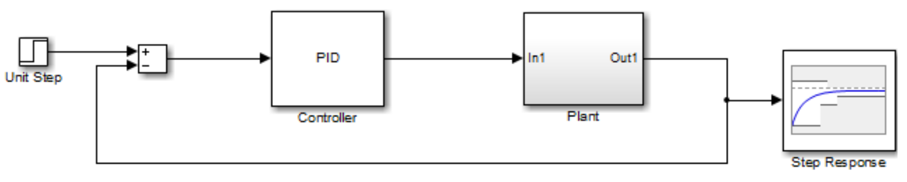
Optimize Controller Parameters to Meet Step Response Requirements (Code)

In this section...

This example shows how to optimize controller parameters to meet step response requirements using `sdo.optimize`. You programmatically modify the default step response requirements specified in a Check Step Response Characteristics block.

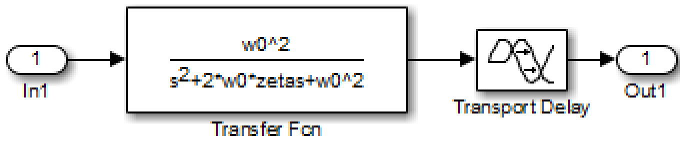
Model Structure

The model `sldo_model1_stepblk` 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.



- The Step Response block is a Check Step Response Characteristics block from the Simulink Design Optimization Signal Constraints library with default step response requirements.

Design Requirements

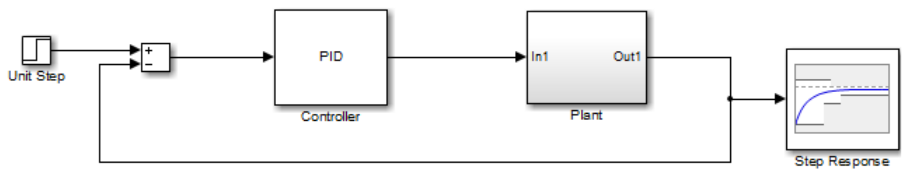
The plant output must meet the following step response requirements:

- 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_stepblk';
open_system(sys);
```



To learn more about the model, see “Model Structure” on page 3-18.

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               = 'sldo_model1_stepblk/Plant';
PlantOutput.OutputPortIndex         = 1;
PlantOutput.LoggingInfo.NameMode    = 1;
PlantOutput.LoggingInfo.LoggingName = 'PlantOutput';
```

3 Store the logging information.

```
simulator = sdo.SimulationTest('sldo_model1_stepblk');
simulator.LoggingInfo.Signals = PlantOutput;
```

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

4 Get the default requirements specified in Step Response block.

```
allBlkReq = getbounds('sldo_model1_stepblk/Step Response');
StepResp  = allBlkReq{1};
```

`StepResp` is a `sdo.requirements.StepResponseEnvelope` object.

Note You do not need Check blocks in the Simulink model to specify design requirements. You can specify requirements directly using requirement objects such as `sdo.requirements.BodeMagnitude`.

---

5 Specify step response requirements.

```
StepResp.RiseTime = 2.5;
StepResp.SettlingTime = 30;
StepResp.PercentOvershoot = 5;
```

Changing the values does not update them in the block.

Specify Design Variables

Before you begin this task, specify the design requirements as described in “Specify Step Response Requirements” on page 3-19.

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('sldo_model1_stepblk',{'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

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 3-19 and "Specify Design Variables" on page 3-20, respectively.

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 `type 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 =

   -0.2578
   -0.4968
   -0.5028
   -1.0000
    0.7297
    0.5095
    0.4964
  202.3590
```

Some Cleq values are positive which indicates that the response using the current parameter values does not satisfy the design requirements.

3 Specify Simulink model to optimize.

```
opt = sdo.OptimizeOptions;
opt.OptimizedModel = 'sldo_model1_stepblk';
```

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:

```
Iter F-count        f(x)    constraint    Steplength    derivative    optimality    Procedure
    0      5           0        202.4
```

```
     1     14          0       200.7       0.341           0           0   infeasible
     2     21          0         105        2.24           0        7.51
     3     29          0       117.2        1.57           0       0.521
     4     36          0          79        3.34           0           0   infeasible
     5     43          0       27.55       0.632           0       0.152
     6     50          0       50.85        1.06           0        1.14
     7     57          0       11.58        3.52           0         9.5
     8     64          0       3.124        2.48           0        13.9
     9     71          0      0.3022        1.57           0        2.91
    10     78          0      0.1289       0.715           0        7.12
    11     85          0     0.02747       0.489           0       0.199
    12     92          0    0.005082       0.184           0      0.0389
    13     99          0   0.0004325      0.0513           0      0.0092  Hessian modified
Local minimum found that satisfies 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 the parameter bounds. For more information about the outputs displayed during the optimization, see "Iterative Display" in the Optimization Toolbox documentation.

5 Examine the optimization termination information. This information helps you verify that the response meets the step response requirements.

```
optInfo
```

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 are negative 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.

```
pOpt
```

```
pOpt(1,1) =

       Name: 'Kp'
      Value: 1.2425
    Minimum: 0
    Maximum: Inf
       Free: 1
      Scale: 1
       Info: [1x1 struct]


pOpt(2,1) =

       Name: 'Ki'
      Value: 0.4037
    Minimum: 0
    Maximum: Inf
       Free: 1
      Scale: 1
       Info: [1x1 struct]


pOpt(3,1) =
```

```
       Name: 'Kd'
      Value: 2.6710
    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('sldo_model1_stepblk',pOpt);
   ```

   b Simulate the model.

   ```
   sim('sldo_model1_stepblk')
   ```

   Although the requirements are satisfied, the model throws assertion warnings at the MATLAB prompt because the signal does not lies within the assertion tolerance which are tighter than the optimization tolerances. Try the following to eliminate the warnings:

   - Continue to search for solutions that are typically located further inside the constraint region.

     ```
     opt.StopIfFeasible = 'off'
     ```

   - Increase the optimization solver tolerance.

     ```
     opt.MethodOptions.TolX = 1.5e-3
     ```

Related Examples

- "Optimize Controller Parameters to Meet Step Response Requirements (GUI)" on page 3-4
- "Optimize Controller Parameters to Track Reference Signal (GUI)" on page 3-26

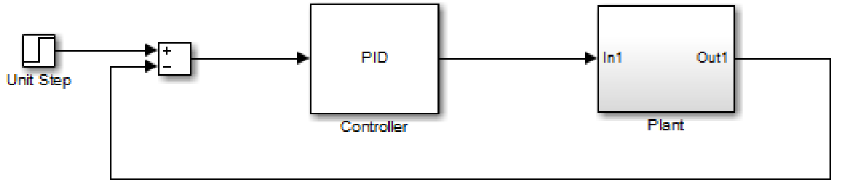
Optimize Controller Parameters to Track Reference Signal (GUI)

In this section...

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.

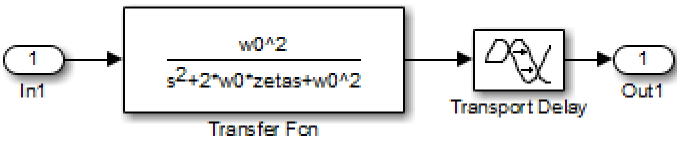
Model Structure

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.



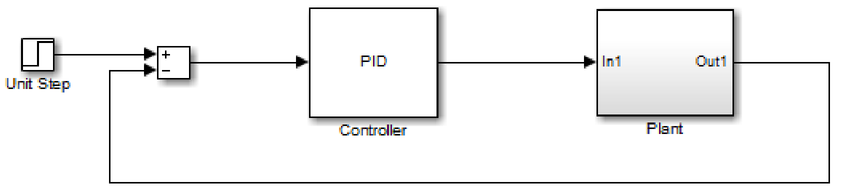
Design Requirements

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.

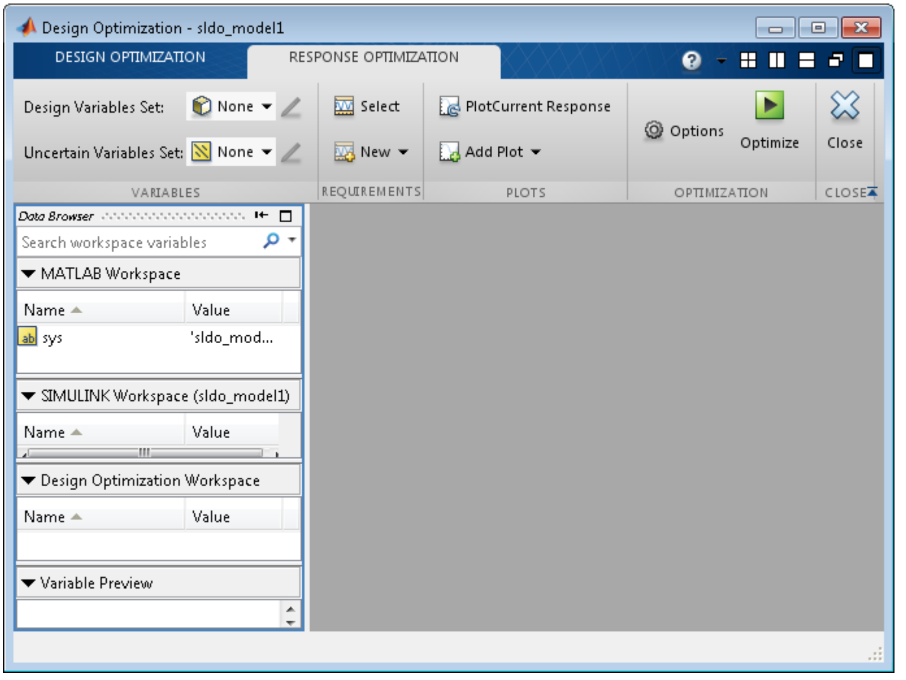
```
sys = 'sldo_model1';
open_system(sys);
```



To learn more about the model, see "Model Structure" on page 3-26.

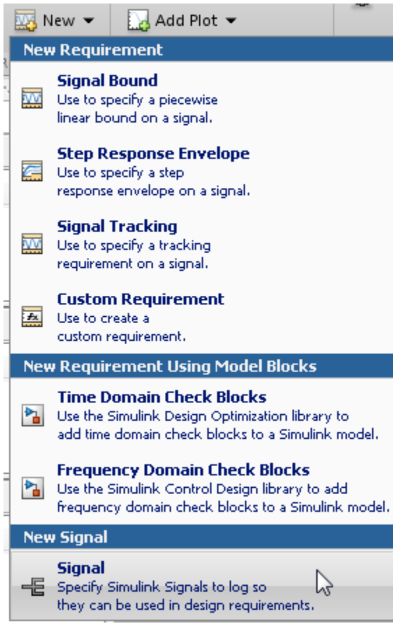
2 In the Simulink model window, select **Tools > Response Optimization**.

A Design Optimization tool for the model opens.

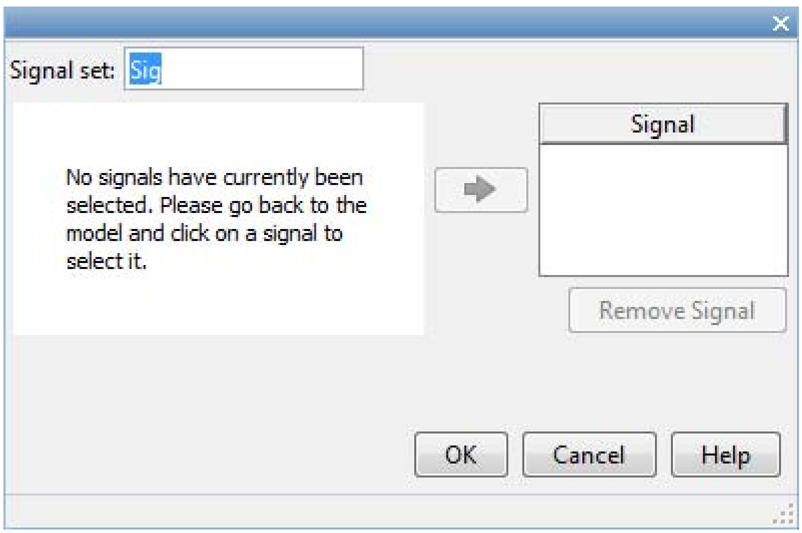


3 Select the model signal which must track the reference signal.

a In the **New** drop-down list, select **Signal**.

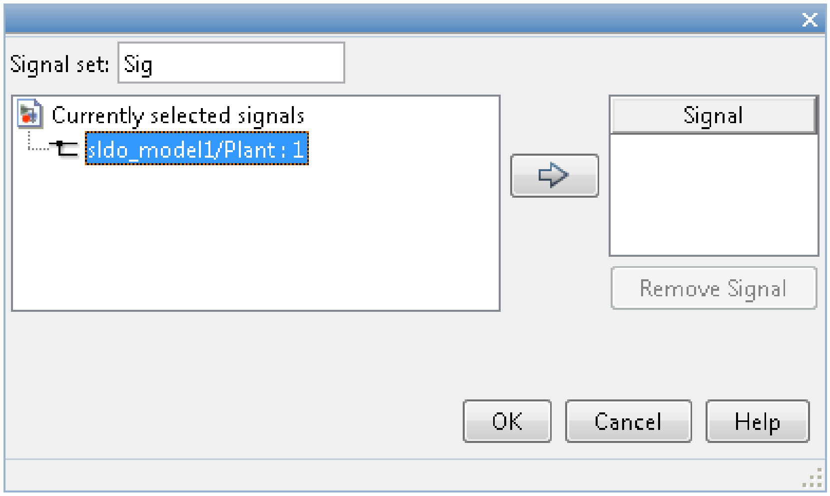


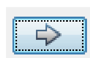
A window opens where you select a model signal.

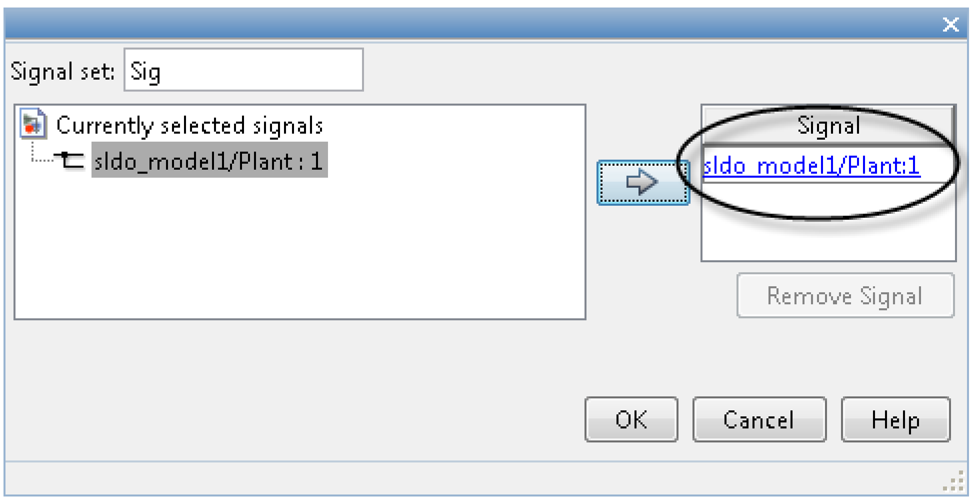


b In the Simulink model window, click the output of the `Plant` block.

The window updates to display the selected signal.



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

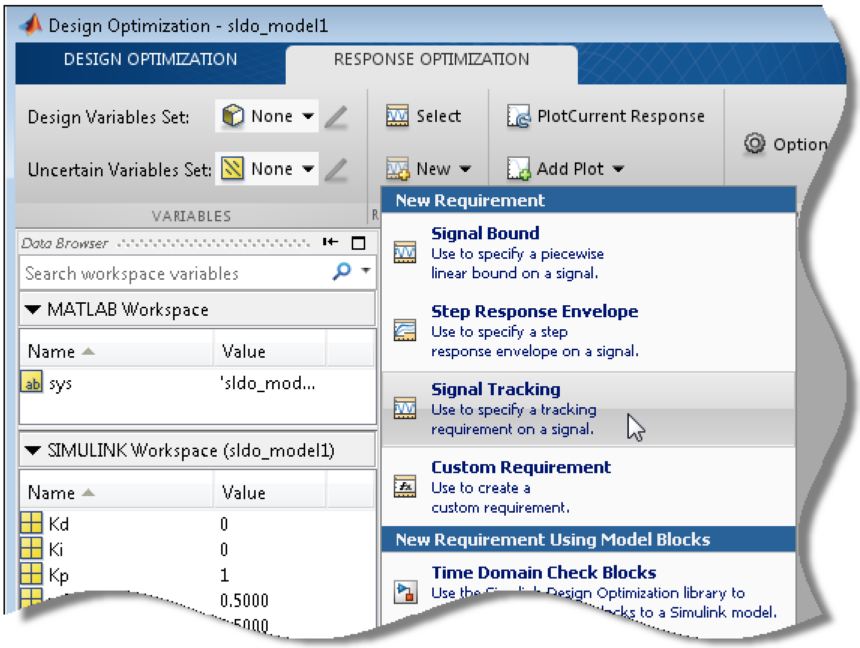


d In **Signal set**, enter `PlantOutput` as the selected signal name.

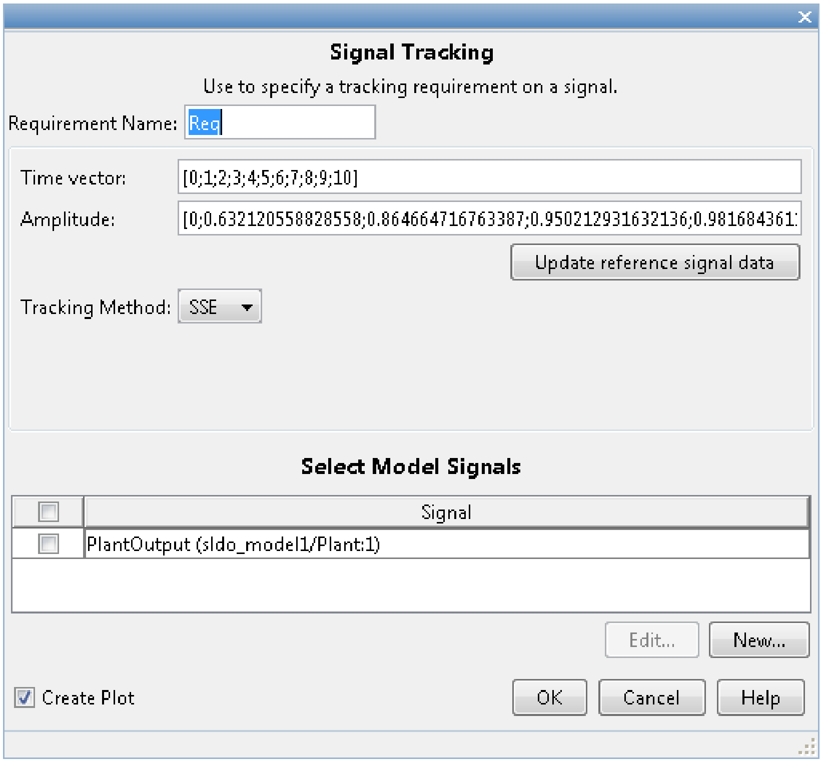
Click **OK**. A new variable `PlantOutput` appears in the **Design Optimization Workspace** of the Design Optimization tool.

4 Specify the reference signal that the model output must track.

a In the **New** drop-down list, select **Signal Tracking**.



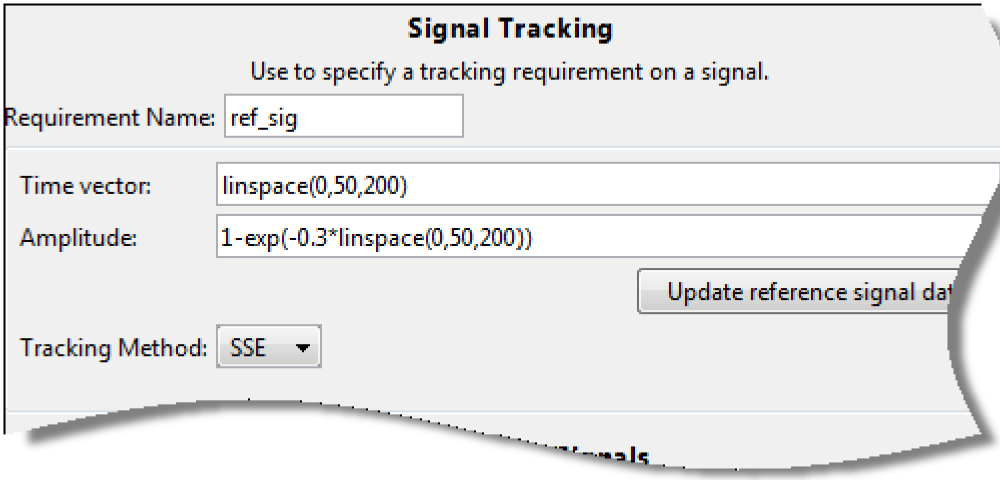
A window opens where you specify the reference signal.



b In **Requirement Name**, enter `ref_sig`.

c In **Time vector**, enter `linspace(0,50,200)`

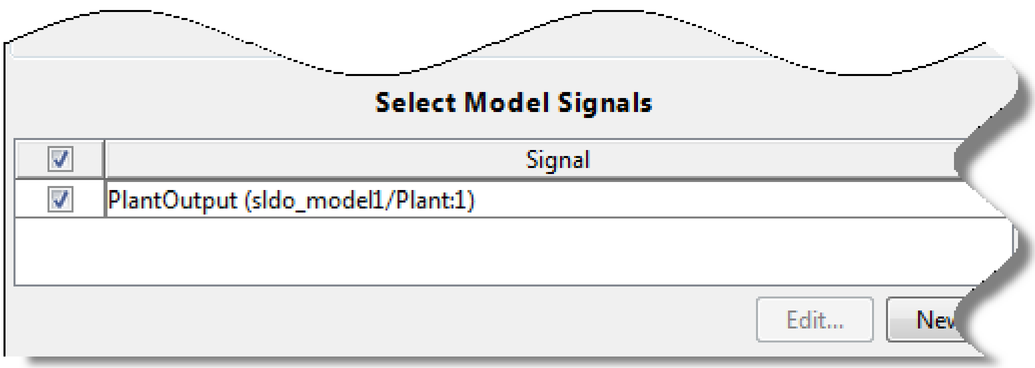
d In **Amplitude**, enter `1-exp(-0.1*linspace(0,50,200))`.



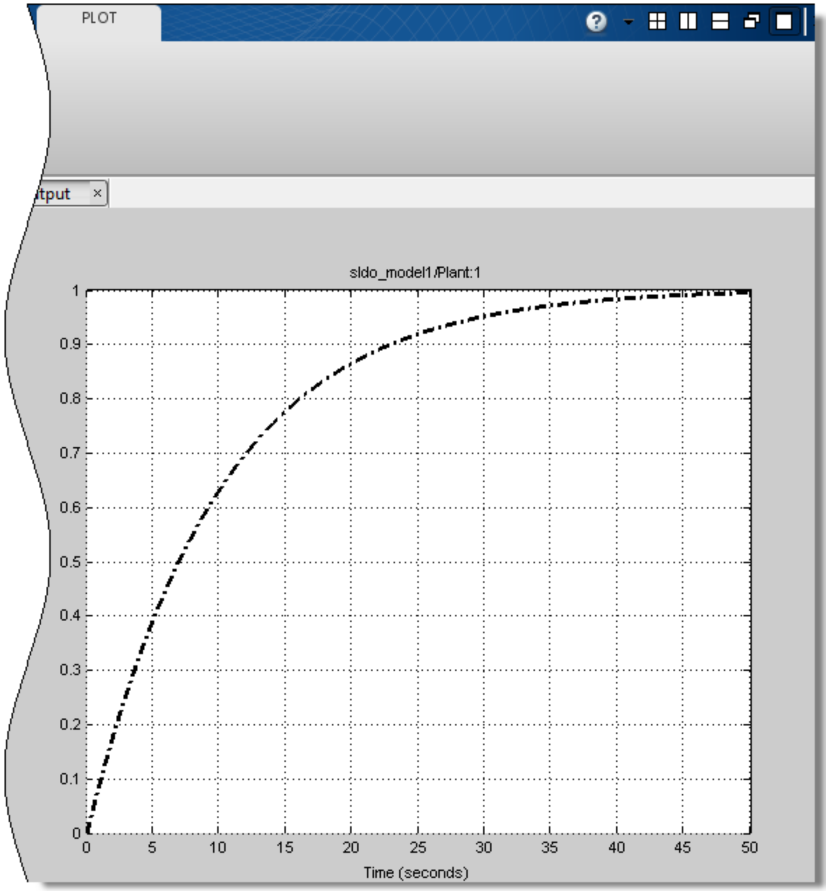
The **Tracking Method** is `SSE` which means that at each optimization iteration, the solver attempts to reduce the sum of squared errors between the simulated output and reference signal.

e Click **Update reference signal data**.

f Select the check-box corresponding to the signal you selected in the previous step in the **Select Model Signals** area.



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



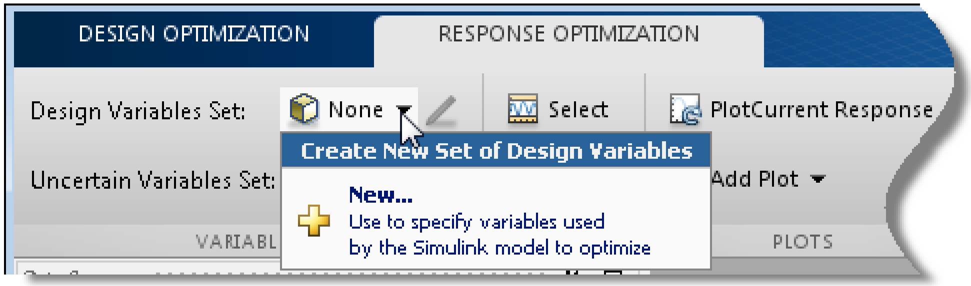
Specify Design Variables

Before you begin this task, specify the reference signal to track as described in "Specify Reference Signal" on page 3-27.

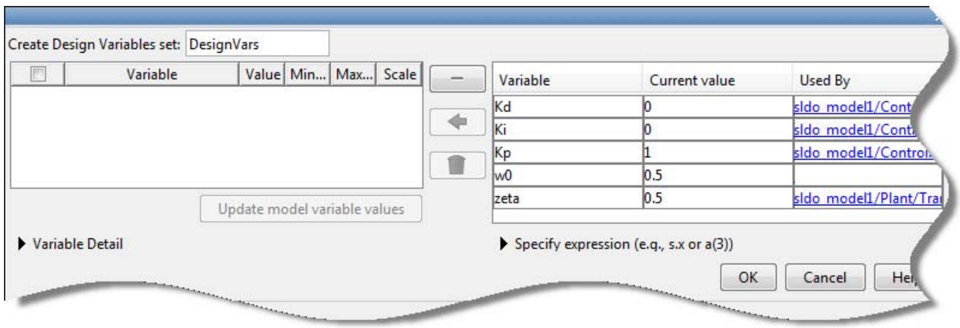
When you optimize the model response, the software modifies the design variable (model parameter) values to meet the design requirements.

In the **Response Optimization** tab:

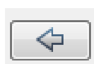
1 Select **New** in the **Design Variables Set** drop-down list.

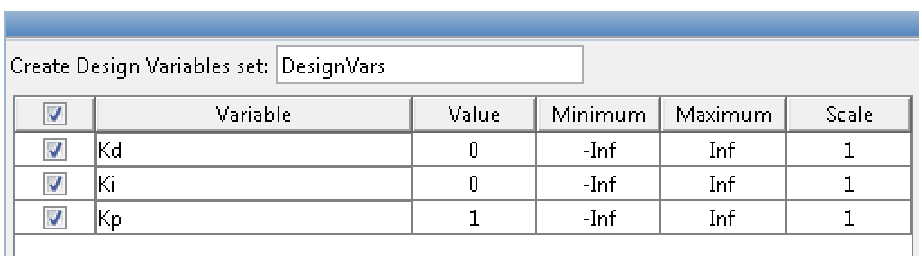


A window opens where you specify design variables.



2 Click `Kd`, `Ki` and `Kp` to select them.

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



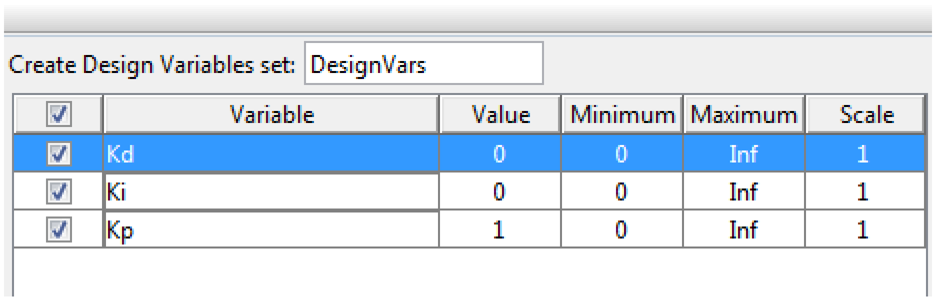
The software displays the following parameter settings:

- **Variable** — Parameter name

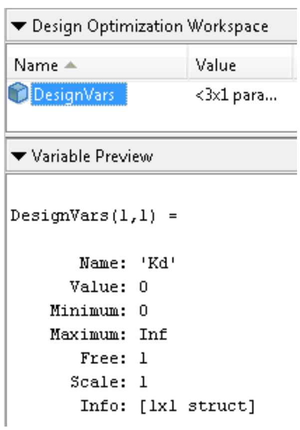
- **Value** — Current parameter value
- **Minimum** and **Maximum** — Parameter bounds
- **Scale** — Scaling factor for the parameter

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**.



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.

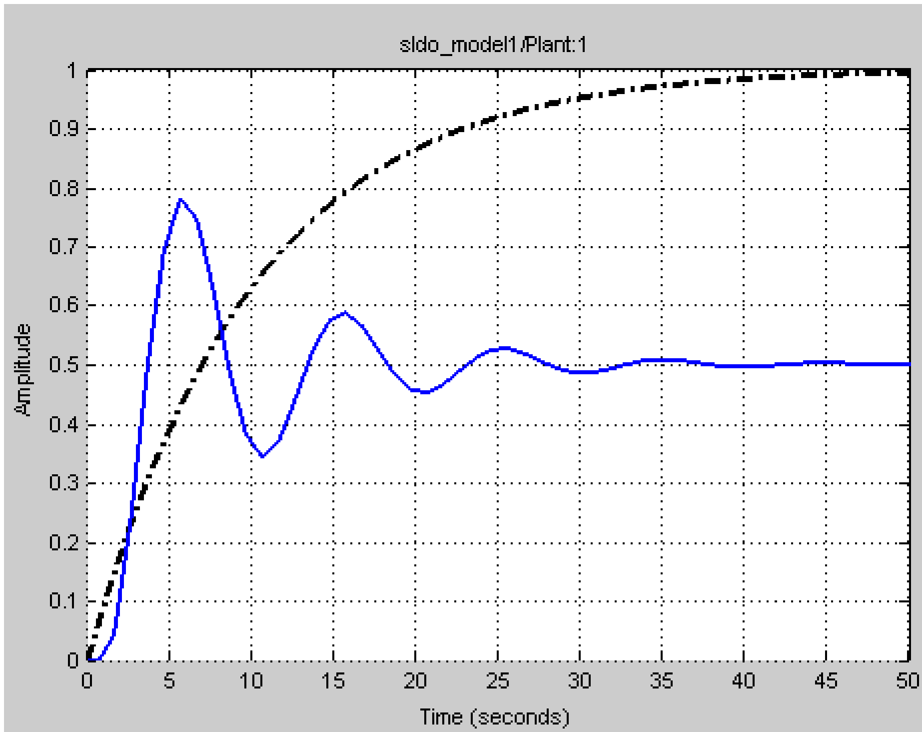


Optimize Model Response

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 3-27 and “Specify Design Variables” on page 3-34, respectively.

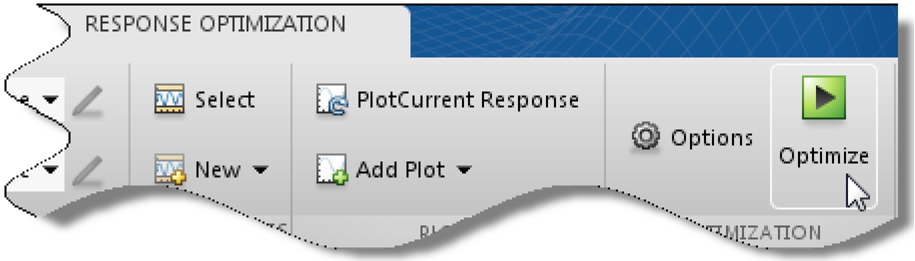
1 (Optional) View the current model response. Click 

.

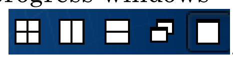


The plot shows that the response does not track the reference signal.

2 Click **Optimize**.

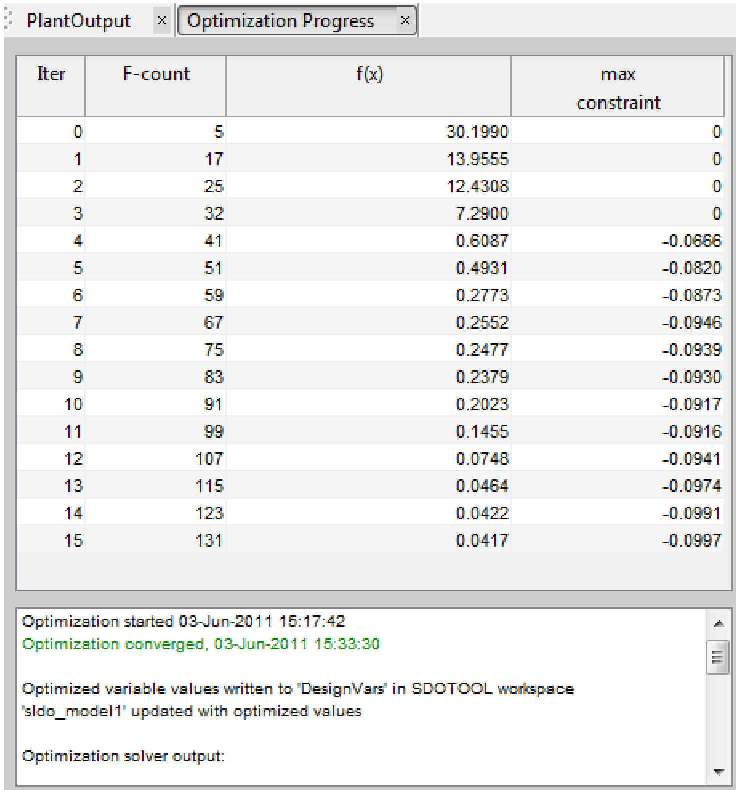


An optimization progress window opens.

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.



Iter	F-count	f(x)	max constraint
0	5	30.1990	0
1	17	13.9555	0
2	25	12.4308	0
3	32	7.2900	0
4	41	0.6087	-0.0666
5	51	0.4931	-0.0820
6	59	0.2773	-0.0873
7	67	0.2552	-0.0946
8	75	0.2477	-0.0939
9	83	0.2379	-0.0930
10	91	0.2023	-0.0917
11	99	0.1455	-0.0916
12	107	0.0748	-0.0941
13	115	0.0464	-0.0974
14	123	0.0422	-0.0991
15	131	0.0417	-0.0997

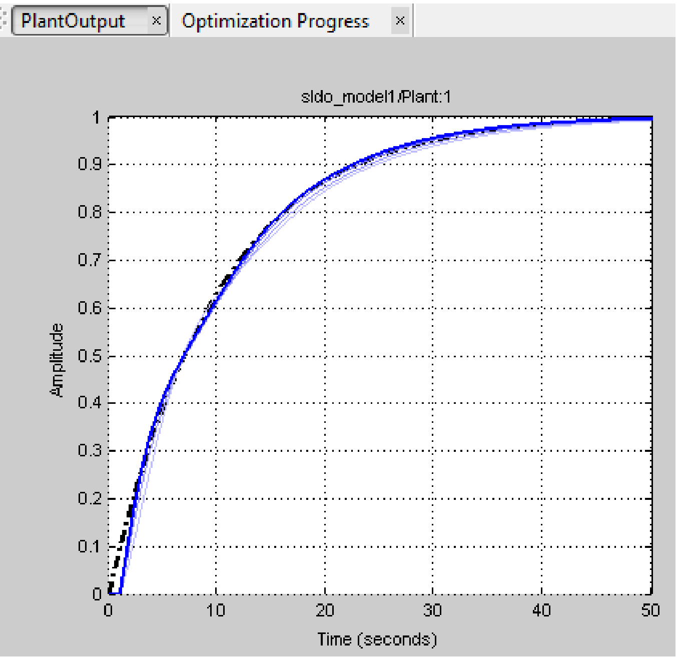
Optimization started 03-Jun-2011 15:17:42
Optimization converged, 03-Jun-2011 15:33:30

Optimized variable values written to 'DesignVars' in SDOTOOL workspace
'sldo_model1' updated with optimized values

Optimization solver output:

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

- "Optimize Controller Parameters to Meet Step Response Requirements (GUI)" on page 3-4
- "Optimize Controller Parameters to Meet Step Response Requirements (Code)" on page 3-18

Design Optimization to Meet Frequency-Domain Requirements (GUI, with Check Block)

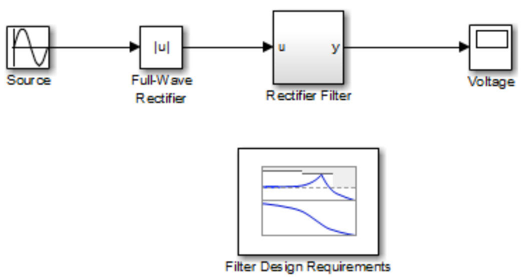
In this section...

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:

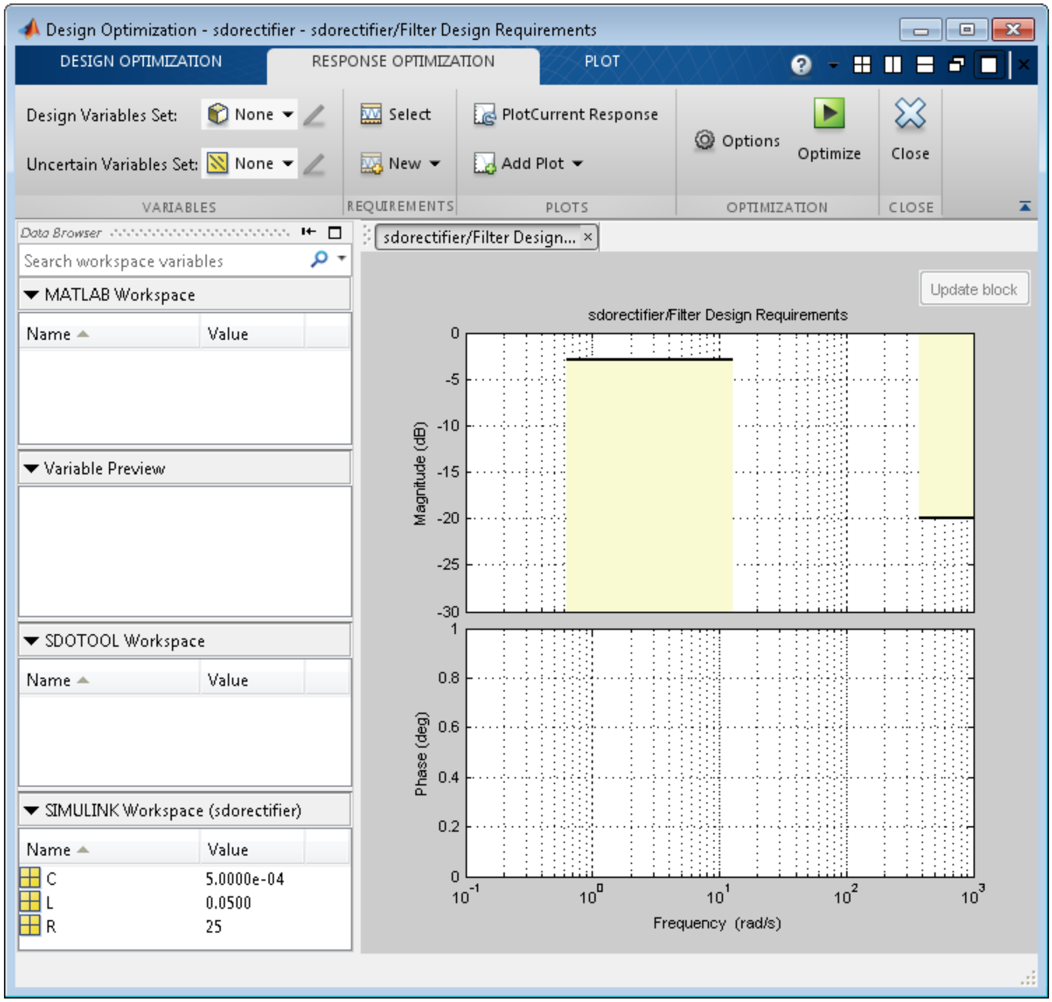
- Have a –3db bandwidth of at least 2Hz
- Limit the gain across the frequency range 2Hz—60Hz to at most 0db
- Limit the gain above 60Hz to at most –20db
- Maximize the filter resistance R
- Minimize the filter inductance L

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.

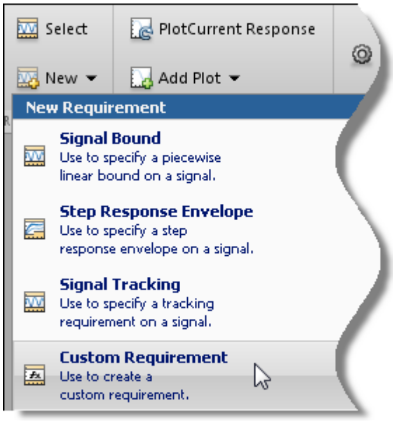
2 Specify a custom objective to minimize the filter inductance and maximize the resistance.

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.

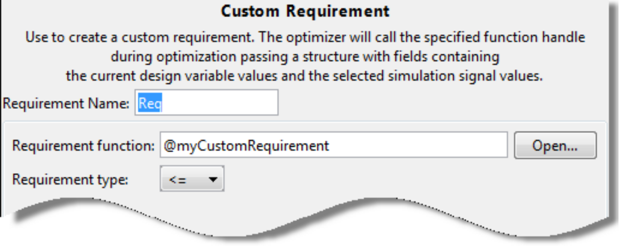
Tip Type `edit sdorectifier_cost` to view this function.

---

a Select **Custom Requirement** in the **New** drop-down list.



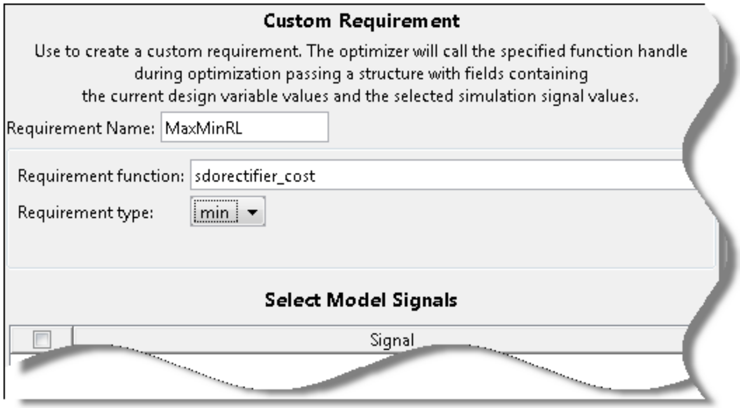
A window opens where you specify the custom requirement.



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**



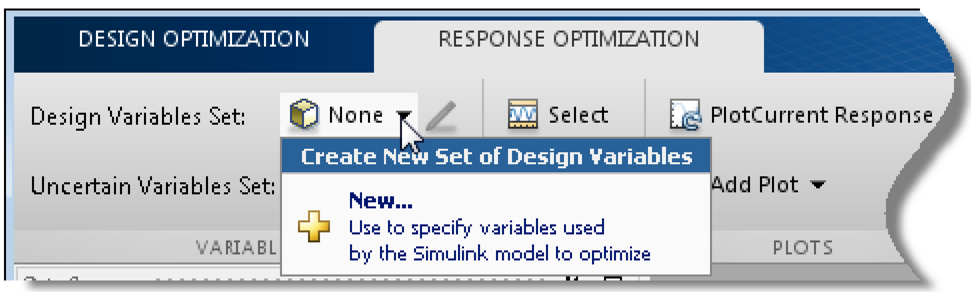
Click **OK**. A new variable `MaxMinRL` appears in **Design Optimization Workspace** of the Design Optimization tool.

Specify Design Variables

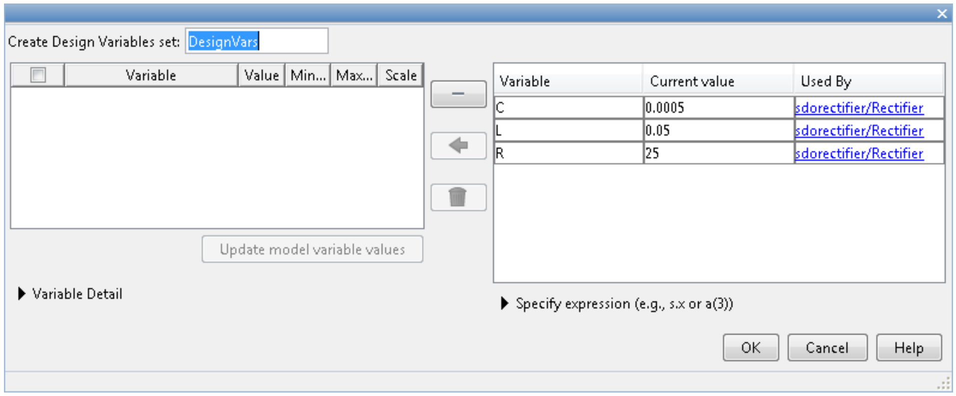
Before you begin this task, specify the design requirements as described in “Specify Design Requirements” on page 3-42.

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.

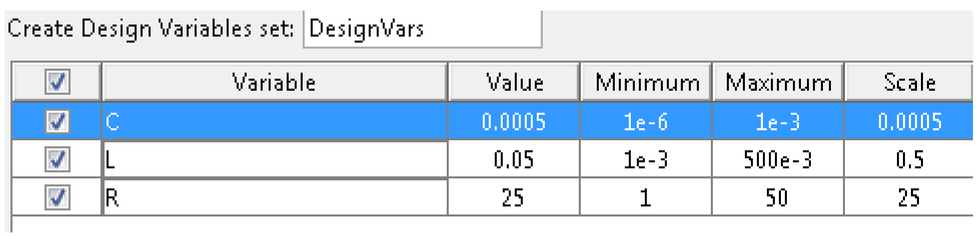


2 Click `R`, `L` and `C` to select them.

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 1µF–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



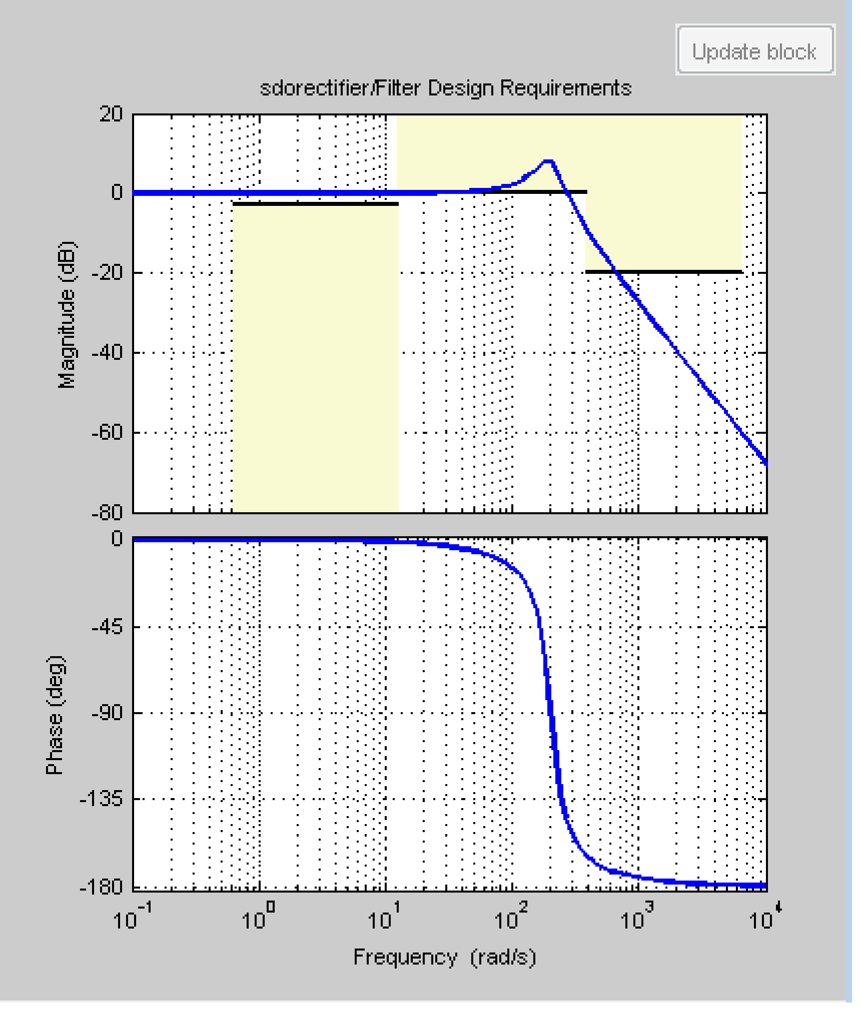
Click **OK**. A new variable `DesignVars` appears in **Design Optimization Workspace** of the Design Optimization tool.

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 3-42 and "Specify Design Variables" on page 3-45, respectively.

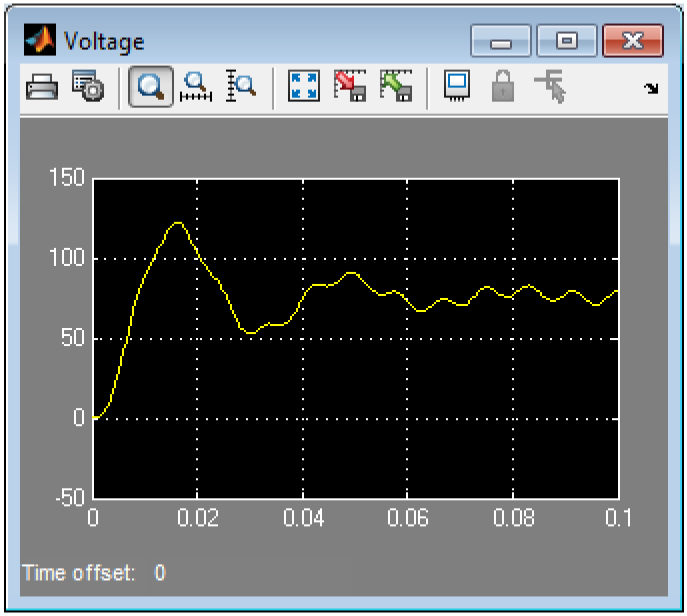
1 (Optional) View the current response of the model. Click 

.

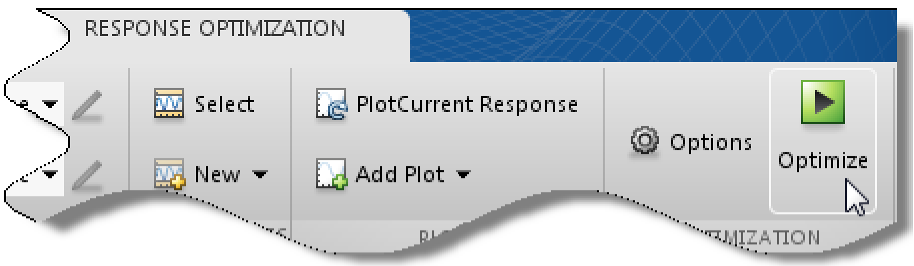


The plot shows that the model does not meet the specified gain and bandwidth requirements.

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.

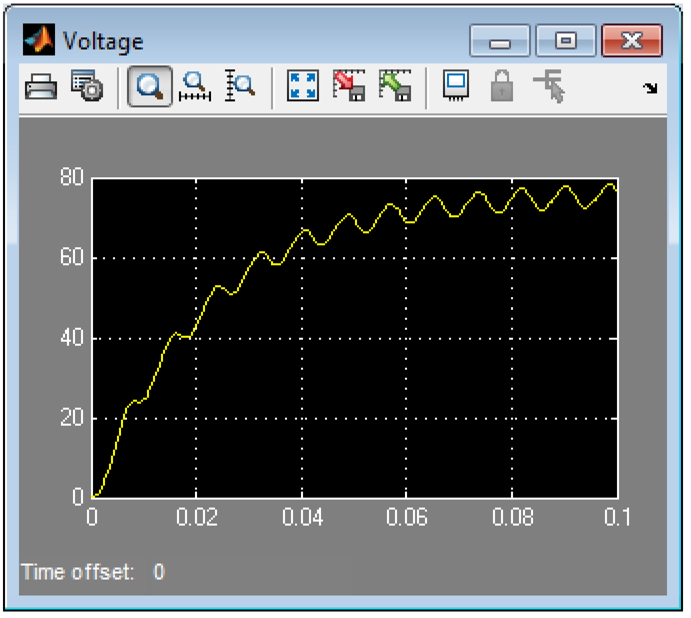
After the optimization completes, the optimization progress window resembles the next figure. After a few iterations, the optimization converges to satisfy the filter bandwidth requirements.

Iter	F-count	f(x)	max constraint
0	8	2	8.1361
1	16	2.2829	0.2409
2	21	1.1305	0
3	29	0.7484	1.2431e-13
4	34	0.7484	0

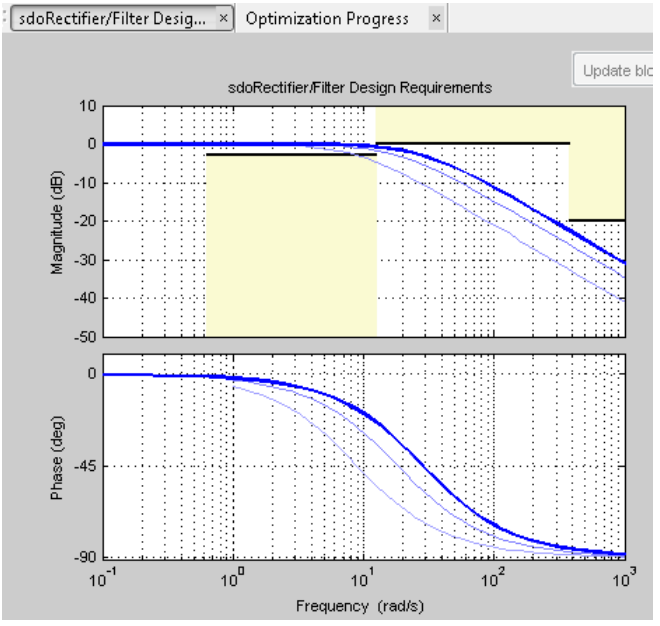
```
Optimization started 13-Jun-2011 14:08:41
Optimization converged, 13-Jun-2011 14:10:27

Optimized variable values written to 'DesignVars' in SDOTOOL 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.

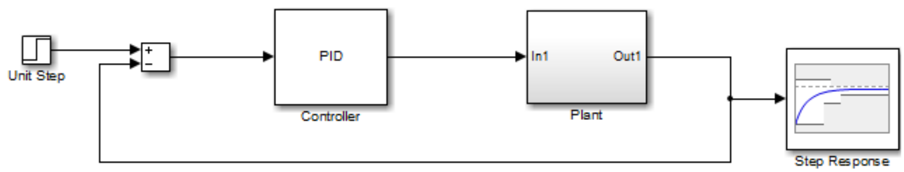
Time-Domain Model Verification

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.

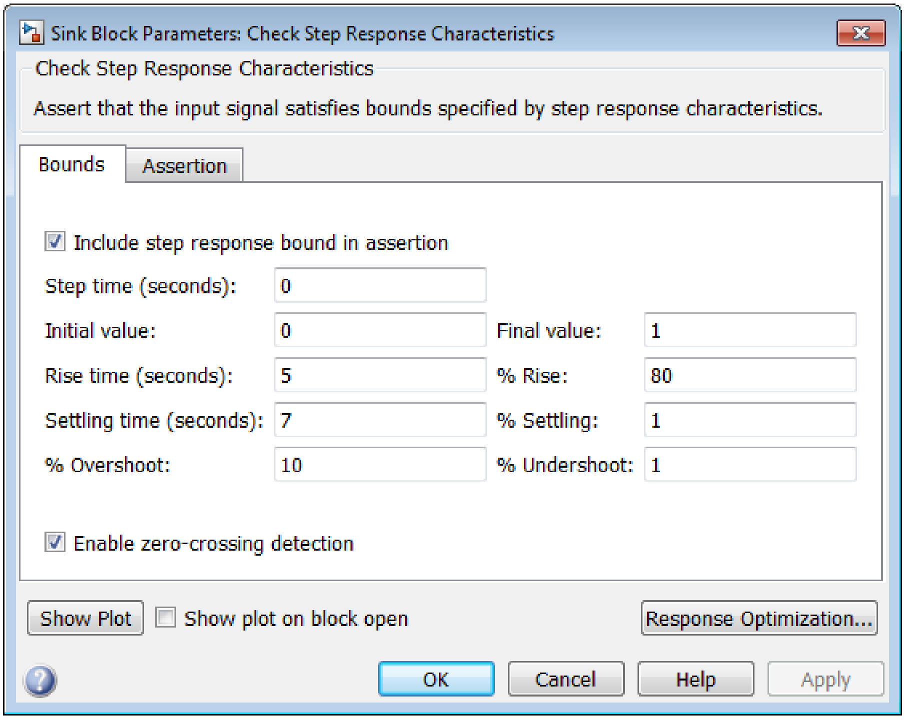
You can also use blocks from Simulink and Simulink Control Design **Model Verification** libraries to design complex assertion logic for time- and frequency-domain verification, and signal monitoring. If you have Simulink Verification and Validation™ software, you can construct simulation tests for your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



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 "Optimize Controller Parameters to Meet Step Response Requirements (GUI)" on page 3-4.

4

Optimization-Based Linear Control Design

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:

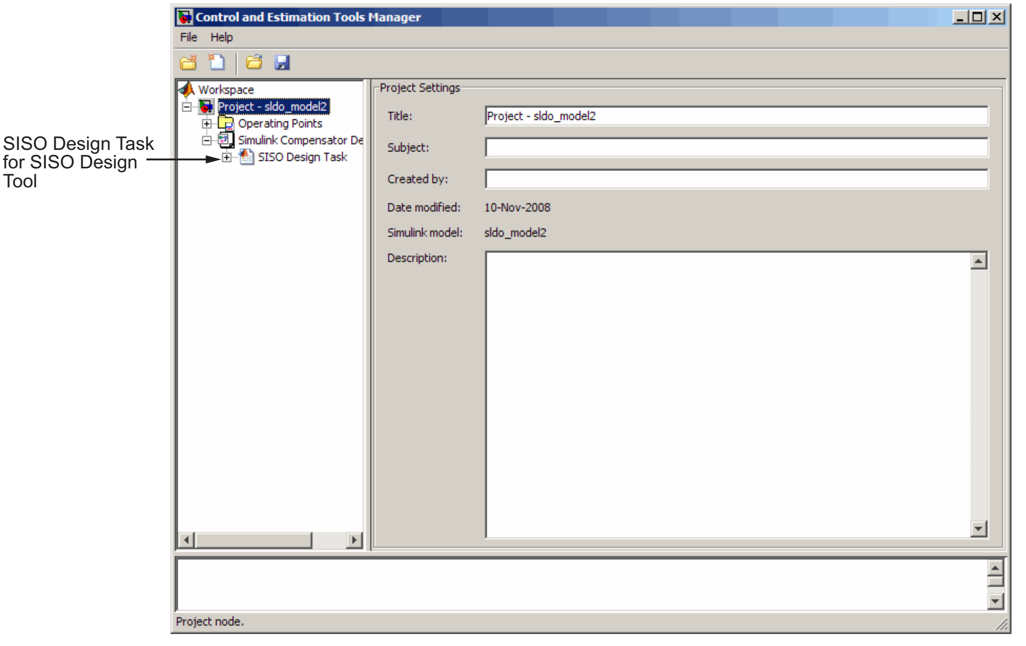
- Simulink Compensator Design Task that contains a linearized version of the Simulink model and, optionally, any response plots you configure.

  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:

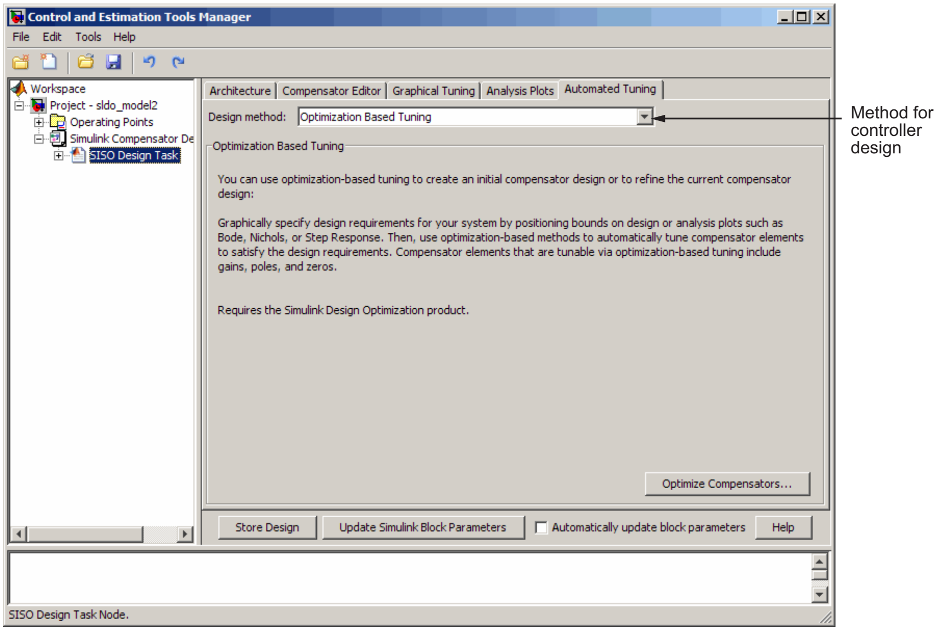
1 Open a SISO Design Tool session by typing the following command at the MATLAB prompt:

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

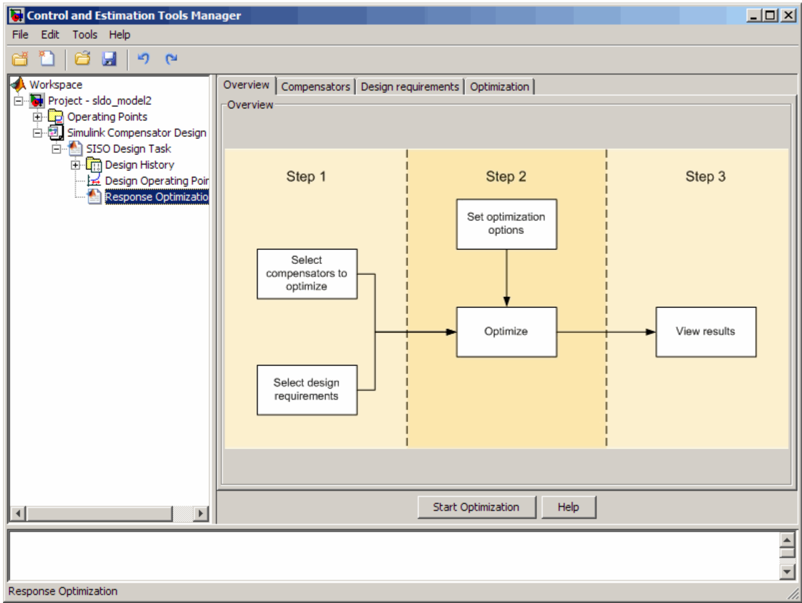


The command also opens a SISO Design for SISO Design Task window by default and any response plots you configured when you linearized the Simulink model using Simulink Control Design software.

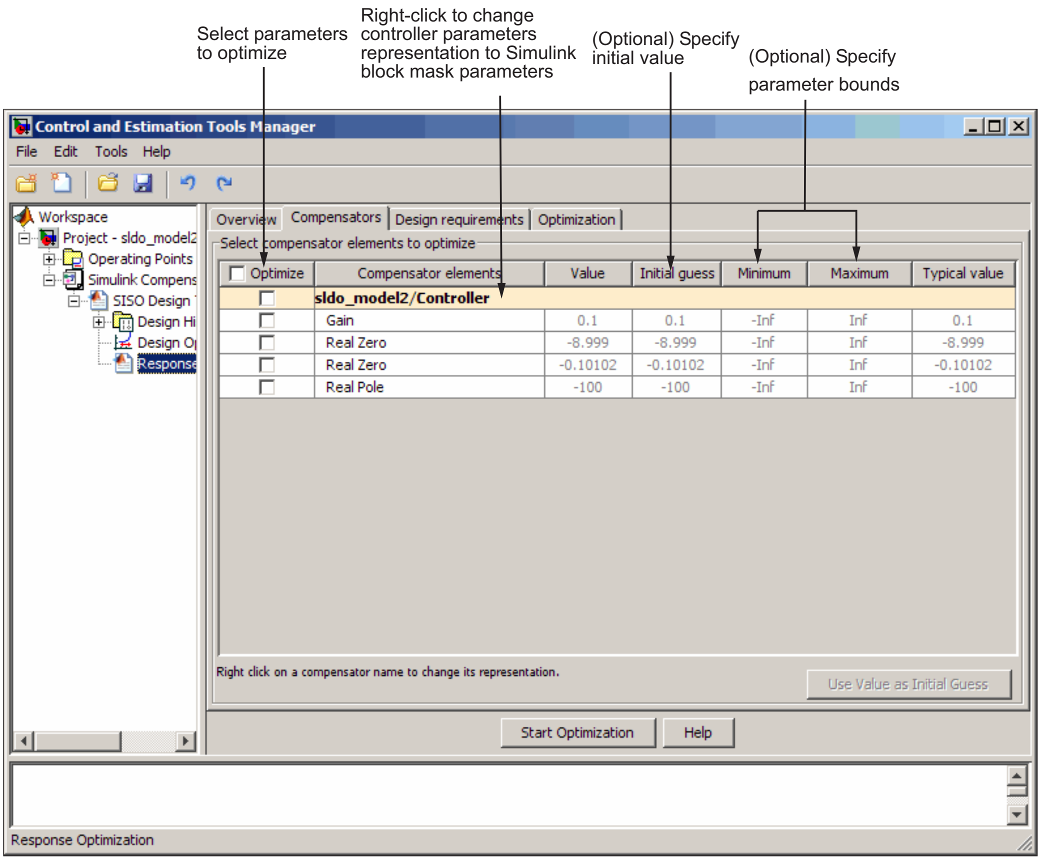
2 Configure a project for optimization-based control design by clicking **Optimize Compensators** in the Automated Tuning tab of the **SISO Design Task**.



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

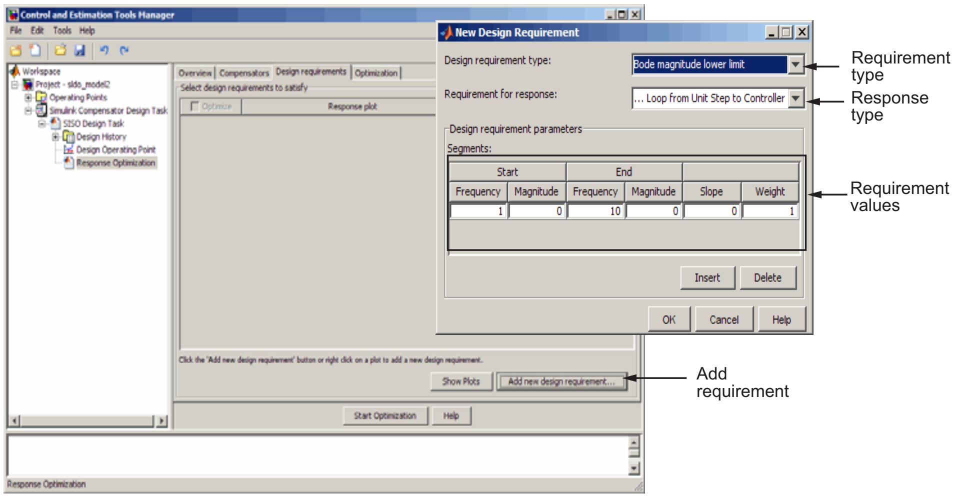


3 Specify the controller parameters to design in the **Compensators** tab.

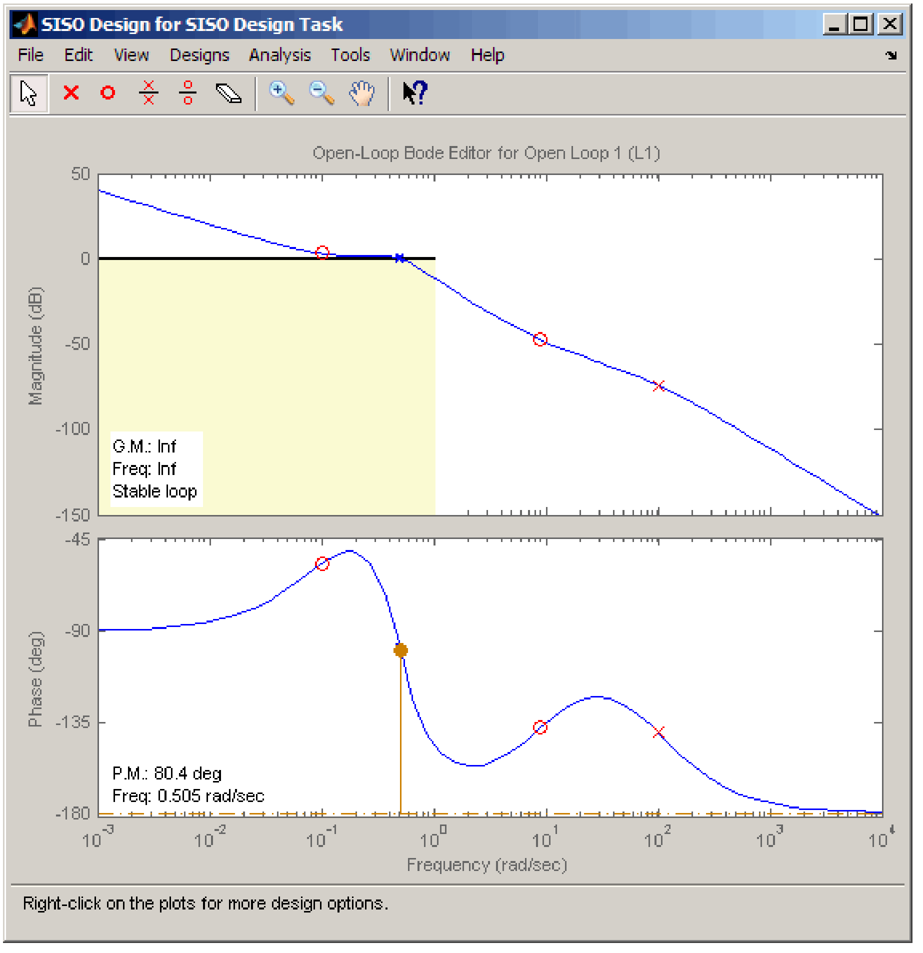


4 Specify the design requirements.

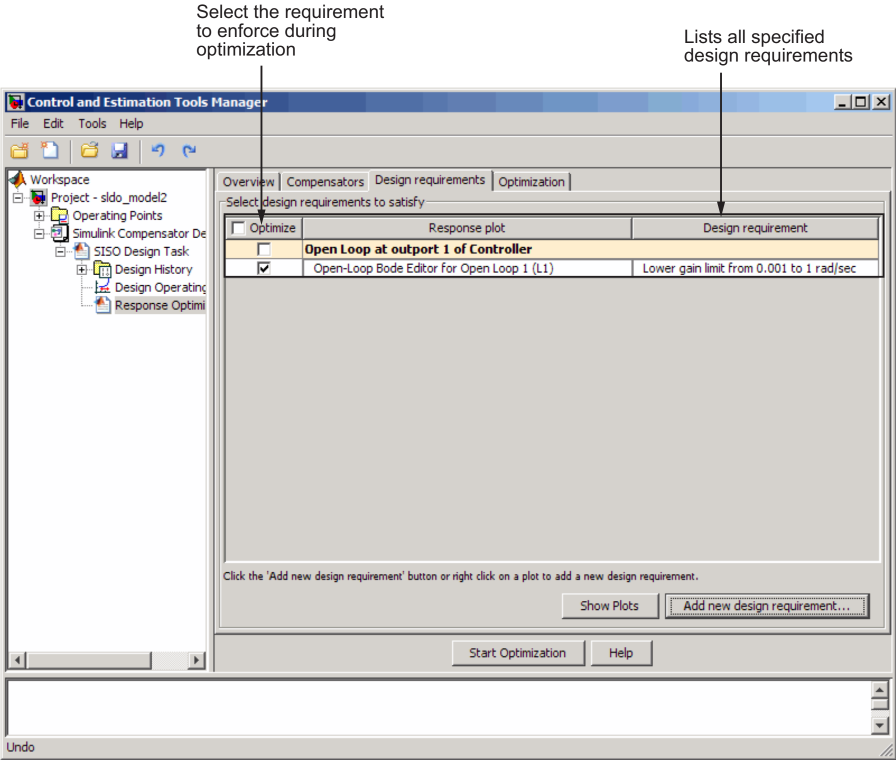
a In the **Design requirements** tab, click **Add new design requirement**. Specify the design requirements, for example Bode magnitude lower limit, in the New Design Requirement dialog box.



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

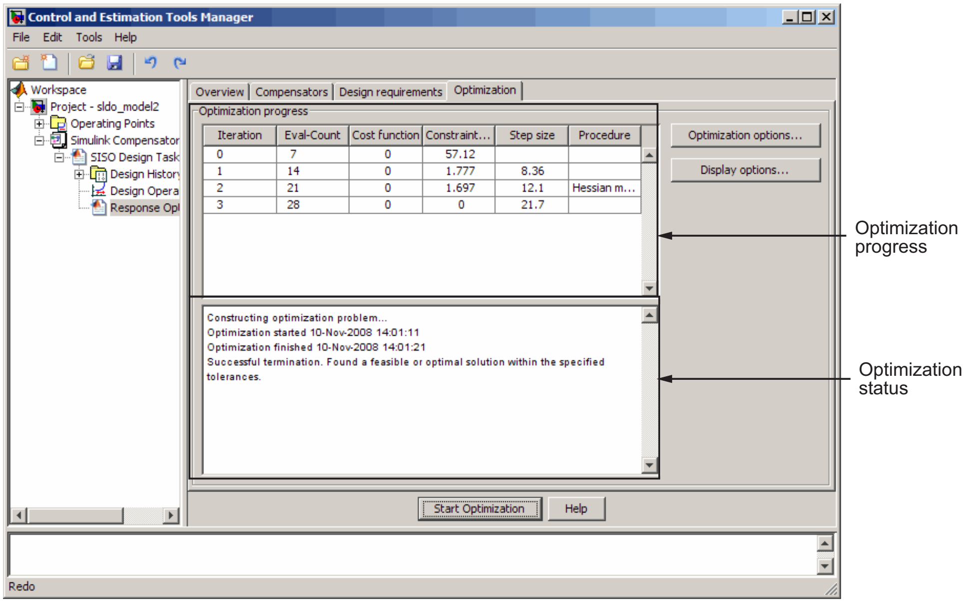


The **Design Requirements** tab also lists the design requirement.



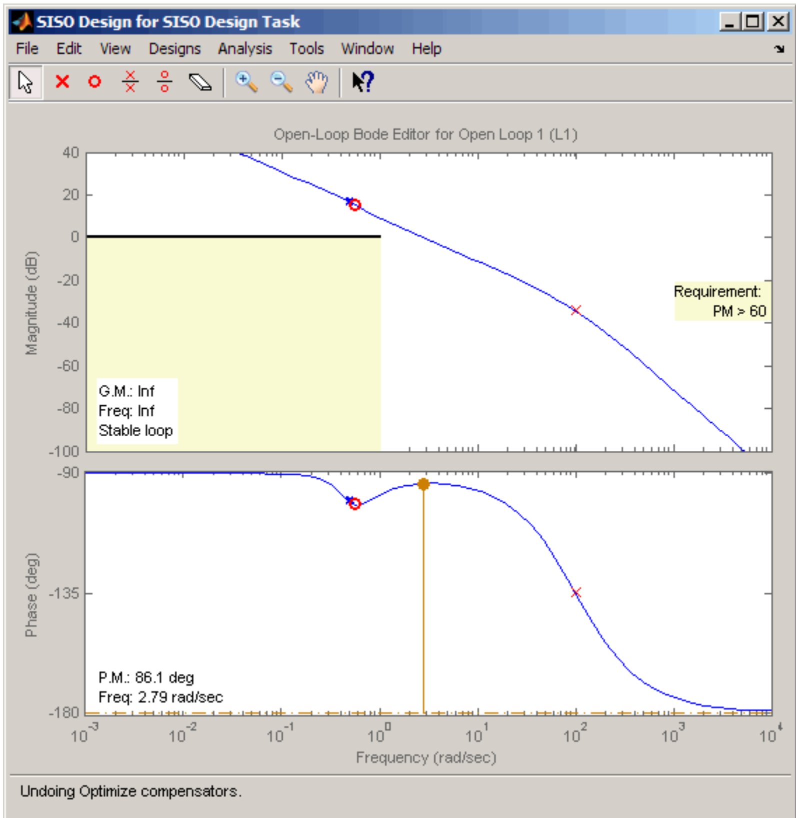
b Repeat step a to specify additional time- and frequency-domain requirements.

5 Start the optimization to design the controller by clicking **Start Optimization** in the **Optimization** tab.

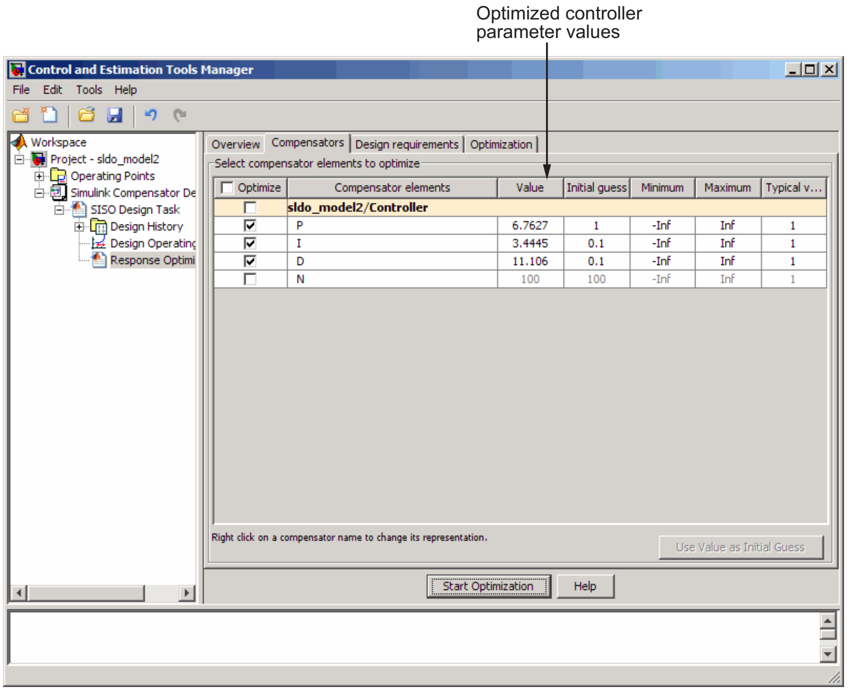


6 Evaluate the controller design.

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.



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

See Also: "Design Optimization-Based PID Controller for Linearized Simulink Model" on page 4-15.

Design Optimization-Based PID Controller for Linearized Simulink Model

In this section...

About This Tutorial

Objectives

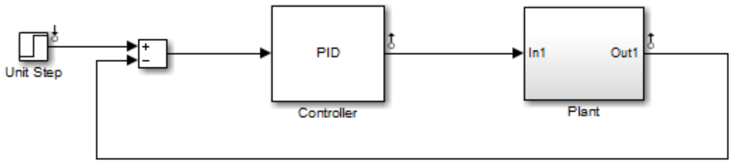
In this tutorial, you learn to use optimization methods to design a PID controller to meet frequency-domain design requirements on a system's response.

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

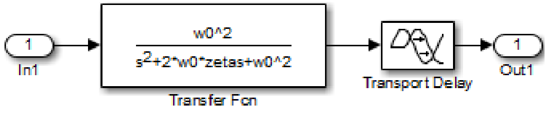
In this tutorial, you use the Simulink model named `sldo_model2`, as shown in the next figure.



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.

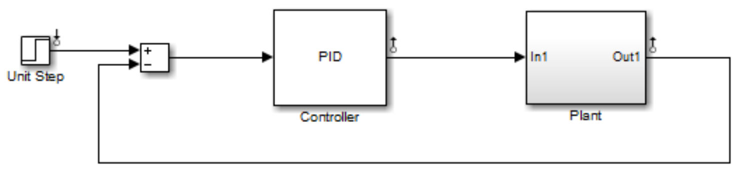
1 Open a SISO Design Tool session for the linearized Simulink model by typing the following command at the MATLAB prompt:

```
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.

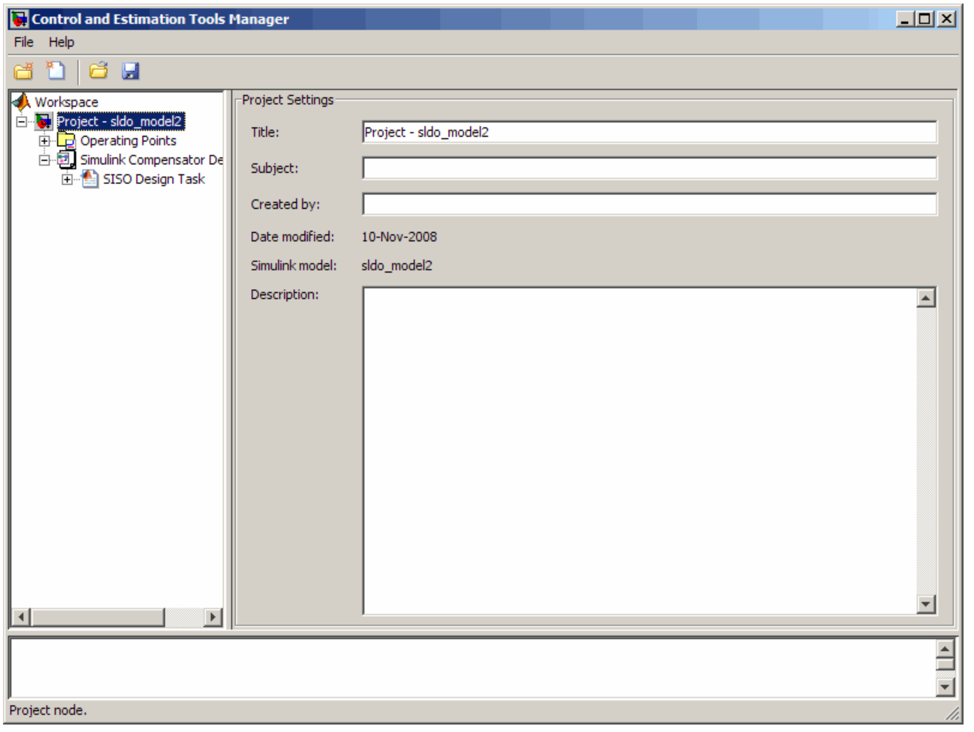
This command opens the following windows:

- Simulink model

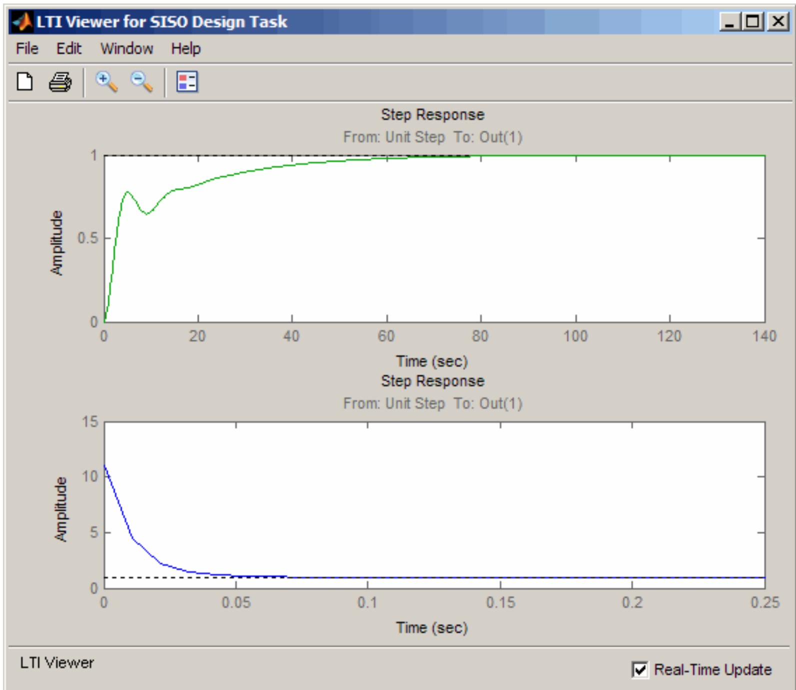


To learn more about the model, see “About the Model” on page 4-15.

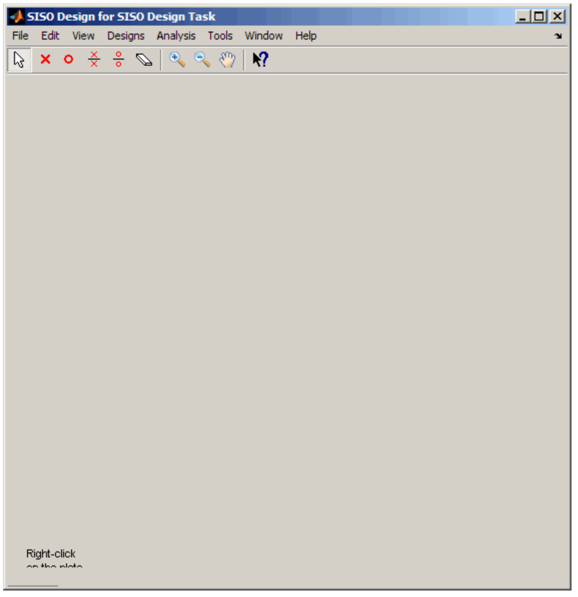
- Control and Estimation Tools Manager GUI, which contains a **SISO Design Task** node under the **Simulink Compensator Design Task** node.



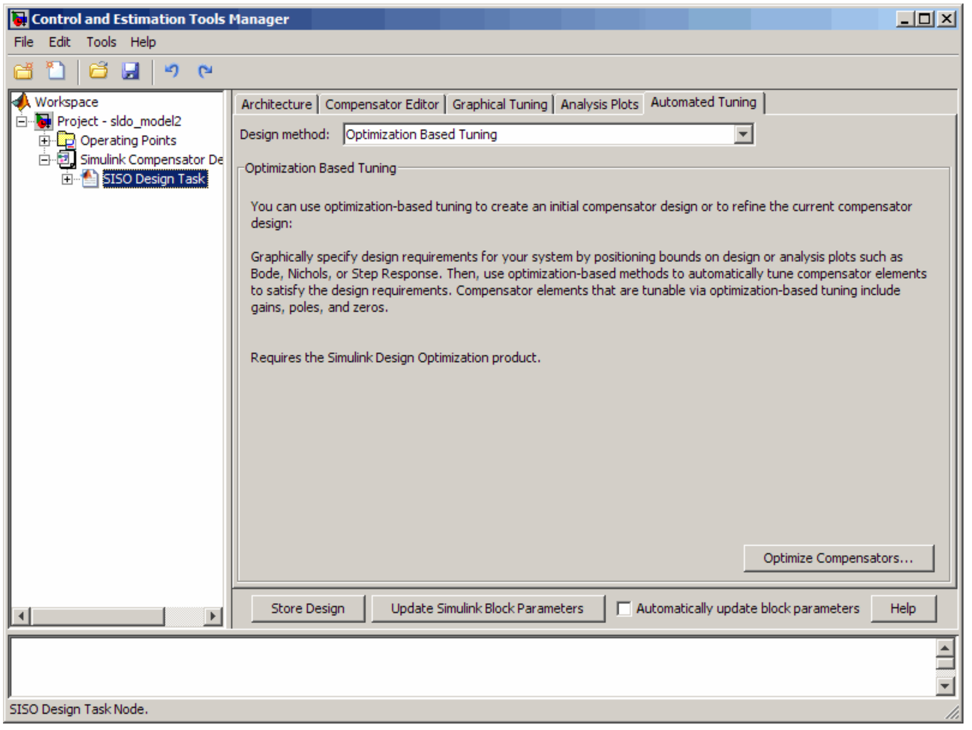
- LTI Viewer for SISO Design Task window, which contains the following plots:
  - Closed-loop step response of the system in the top plot
  - Output of the `Controller` block in the bottom plot



- A blank SISO Design for SISO Design Task window

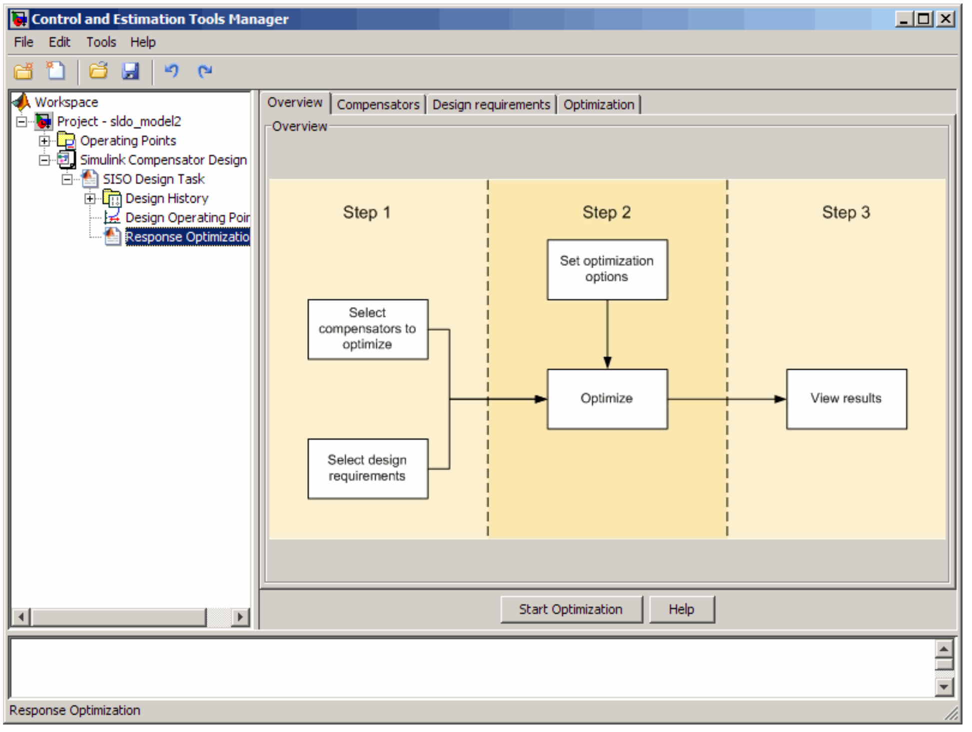


2 In the Control and Estimation Tools Manager GUI, select the **Automated Tuning** tab in the **SISO Design Task** node.



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 4-27
- "Designing the Controller" on page 4-37

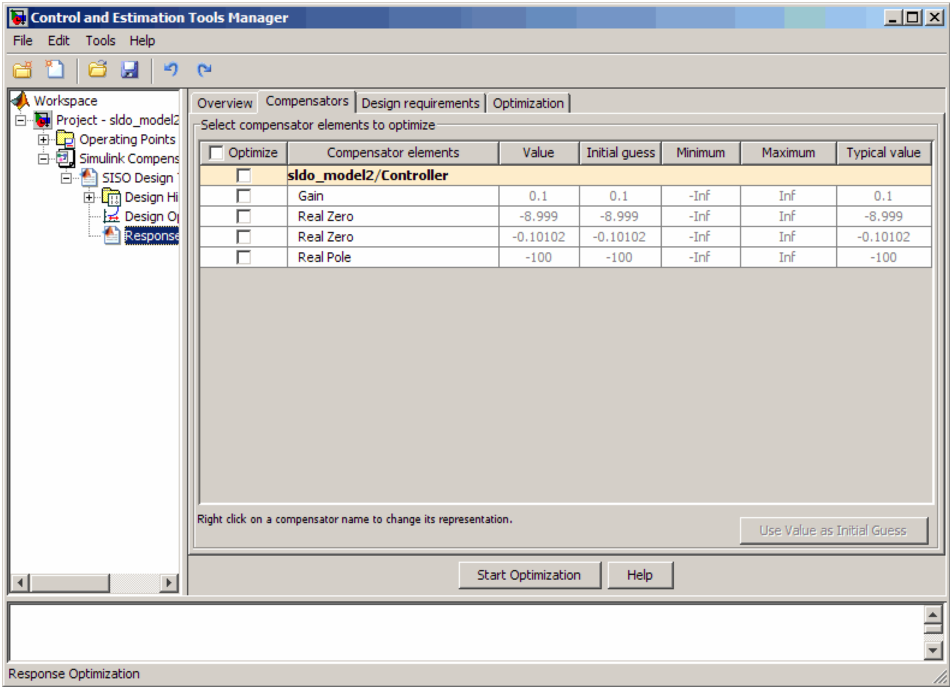
Specifying the Controller Parameters

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 4-17.

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.



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

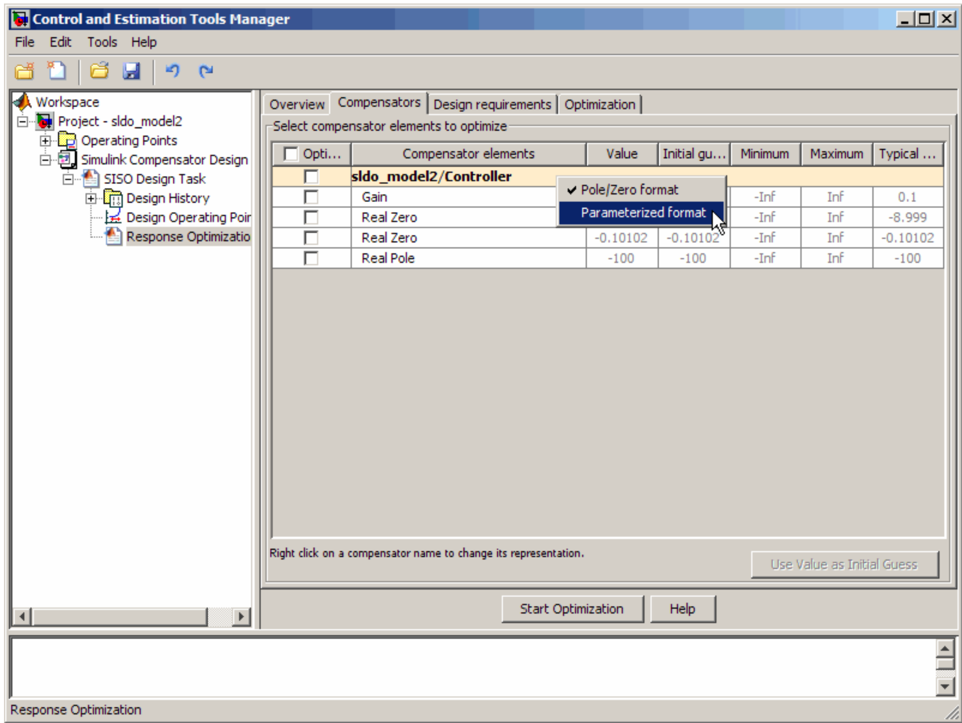
- Gain — Overall gain of the controller
- Real zeros — Zeros resulting from the differentiator and integrator
- Real pole — Pole resulting from the low-pass filter of the differentiator

---

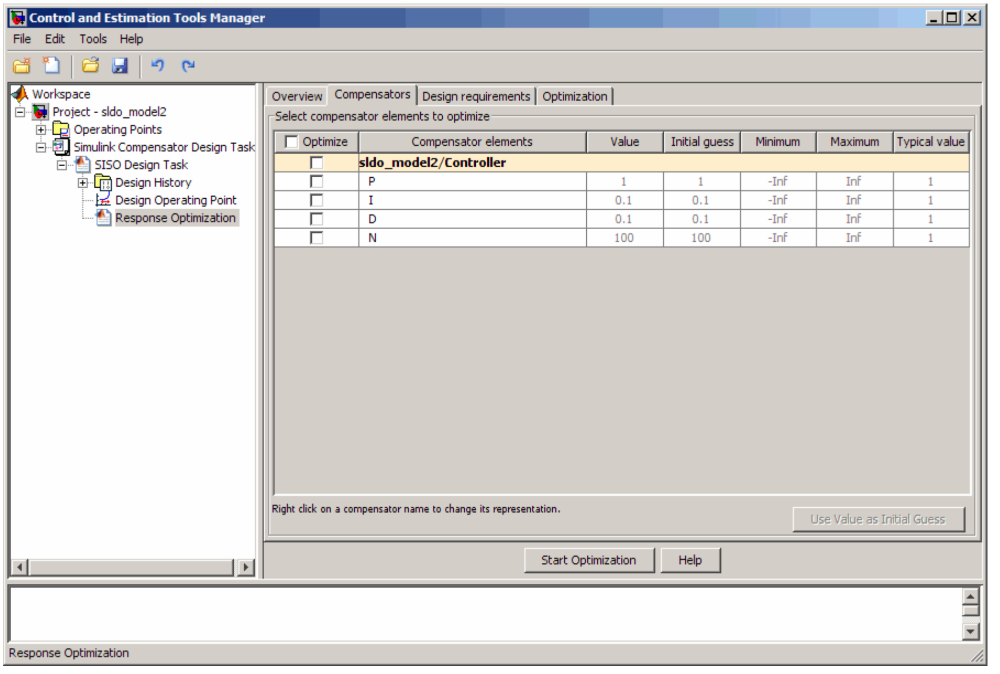
Tip To view the structure of the `Controller` block, right-click the block and select **Mask > Look Under Mask**.

---

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.



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

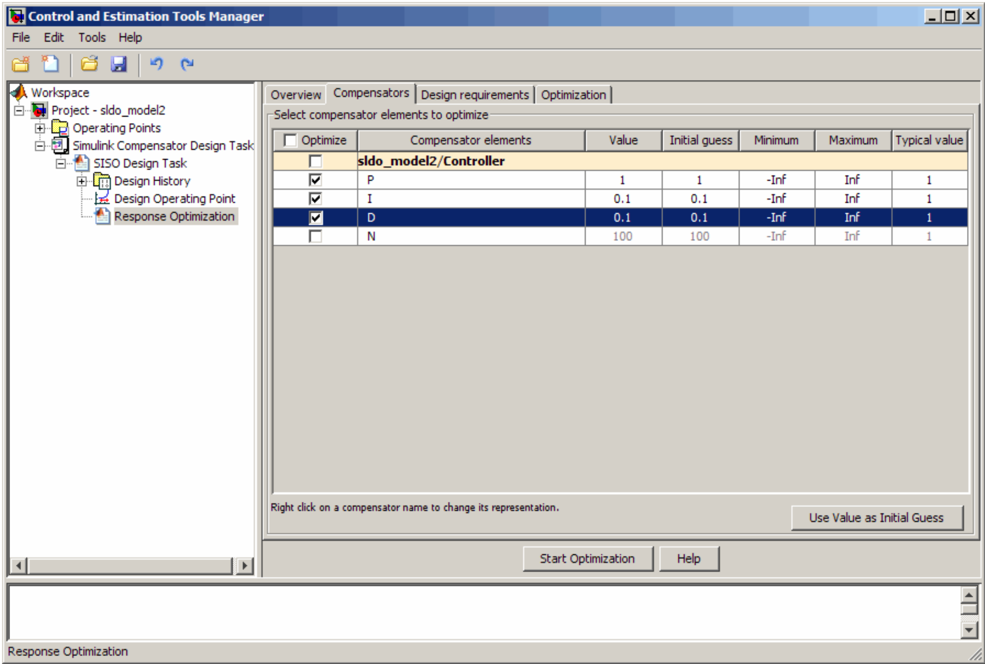
- **Value** — Current controller parameter value
- **Initial Guess** — Initial controller parameter value
- **Minimum** and **Maximum** — Controller parameter bounds
- **Typical Value** — Scaling factor for the controller parameter

---

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`.



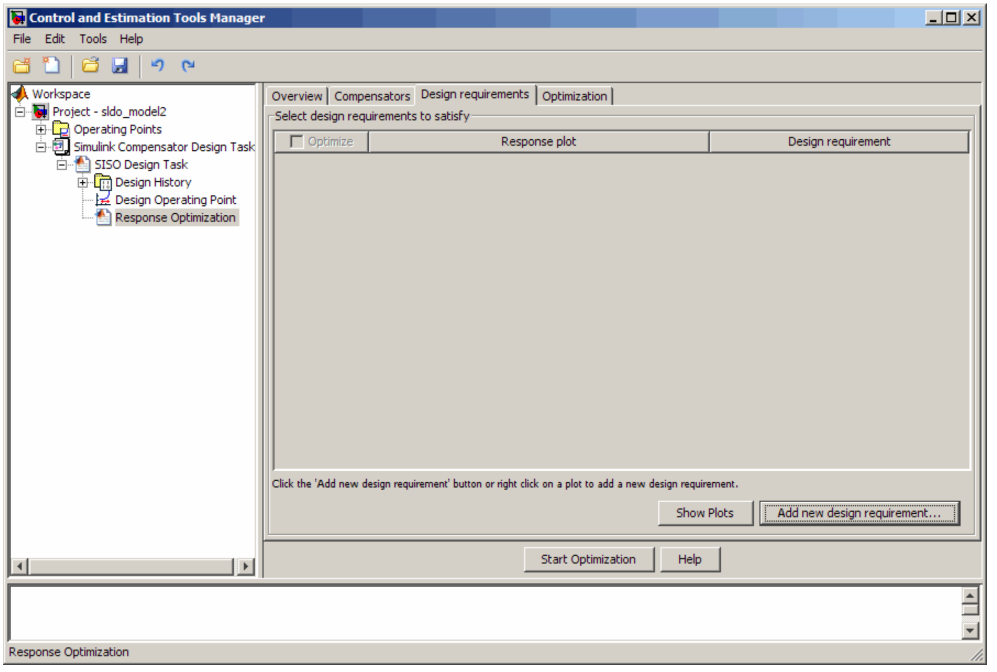
Specifying Bode Magnitude and Phase Margin Design Requirements

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.

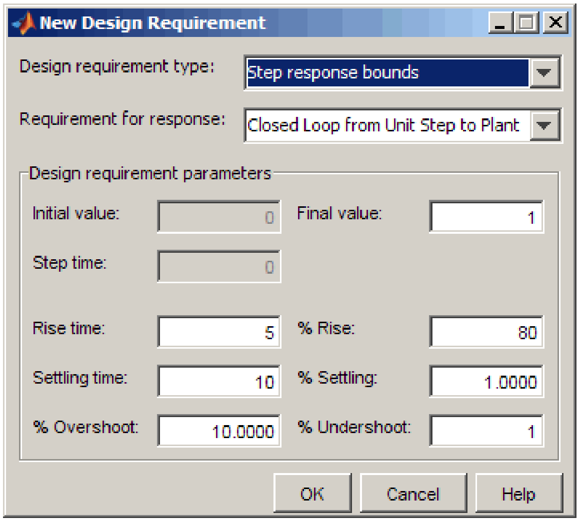
To specify the Bode design requirements:

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



2 Click **Add new design requirement**.

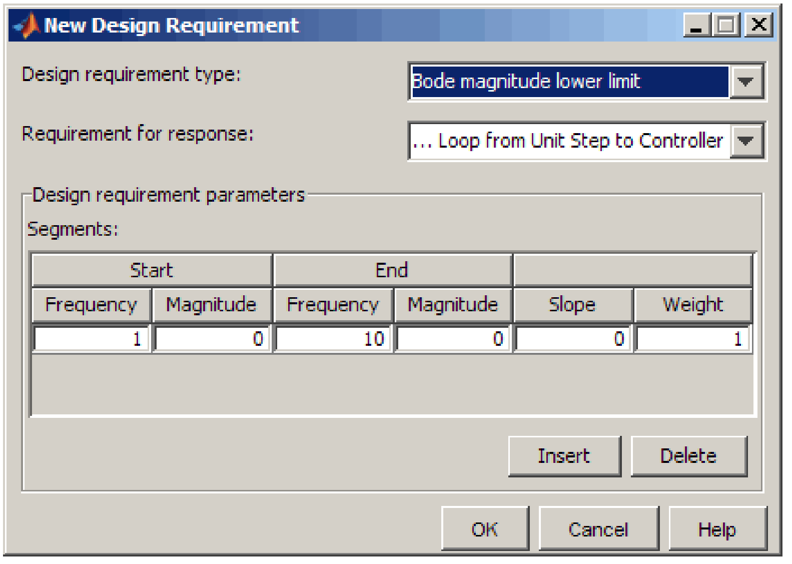
This action opens the New Design Requirement dialog box.



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.

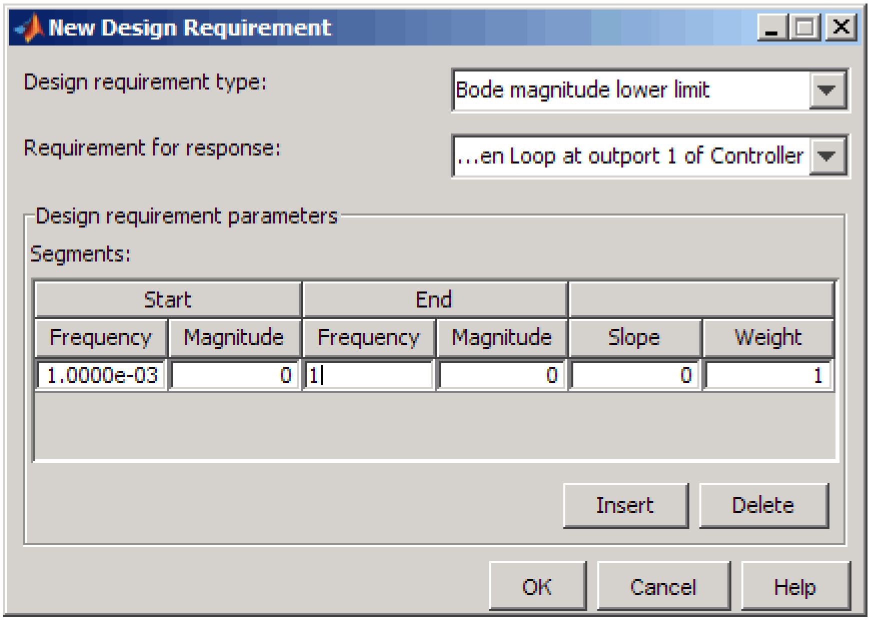


b Select `Open Loop at outport 1 of Controller` from the **Requirement for response** drop-down list.

c In the **Frequency** field of the **Start** column, enter `1e-3`.

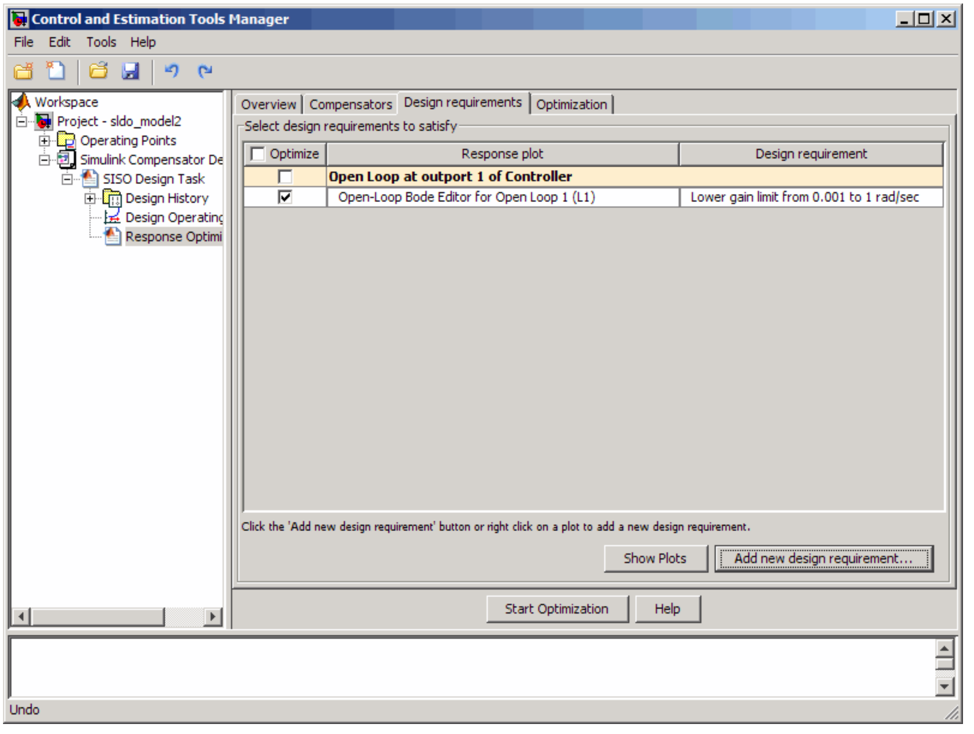
d In the **Frequency** field of the **End** column, enter 1.

The New Design Requirement dialog box resembles the following figure.

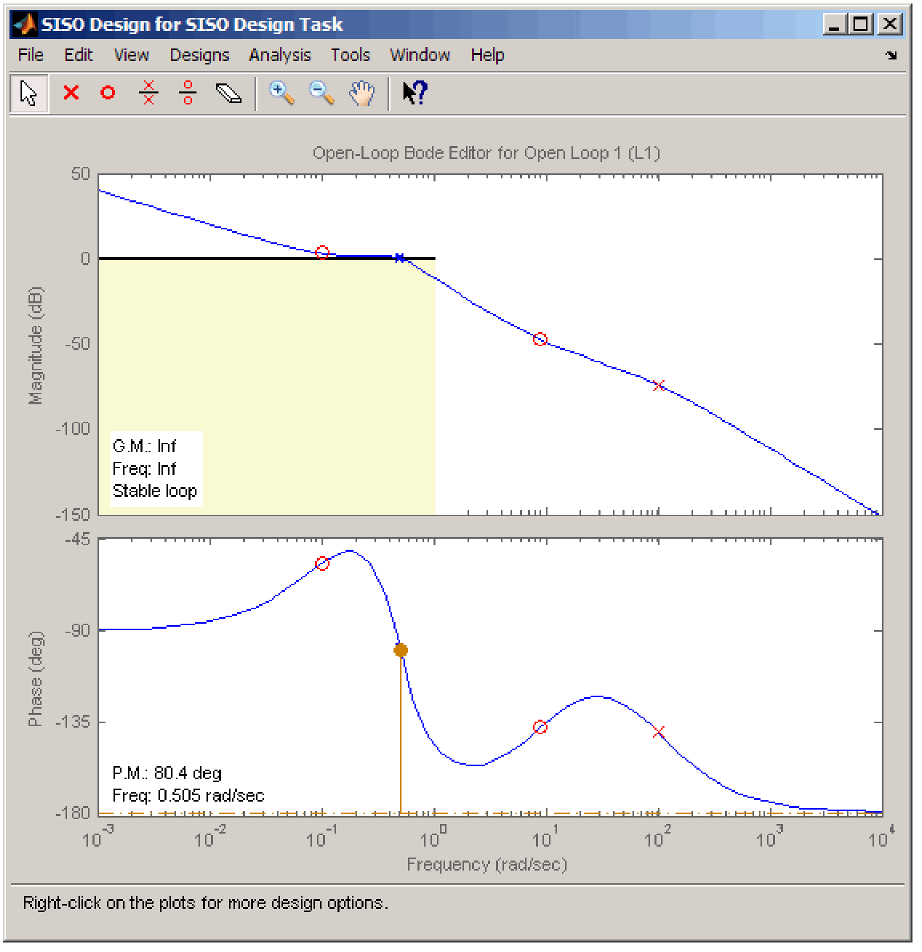


e Click **OK** to close the New Design Requirement dialog box.

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

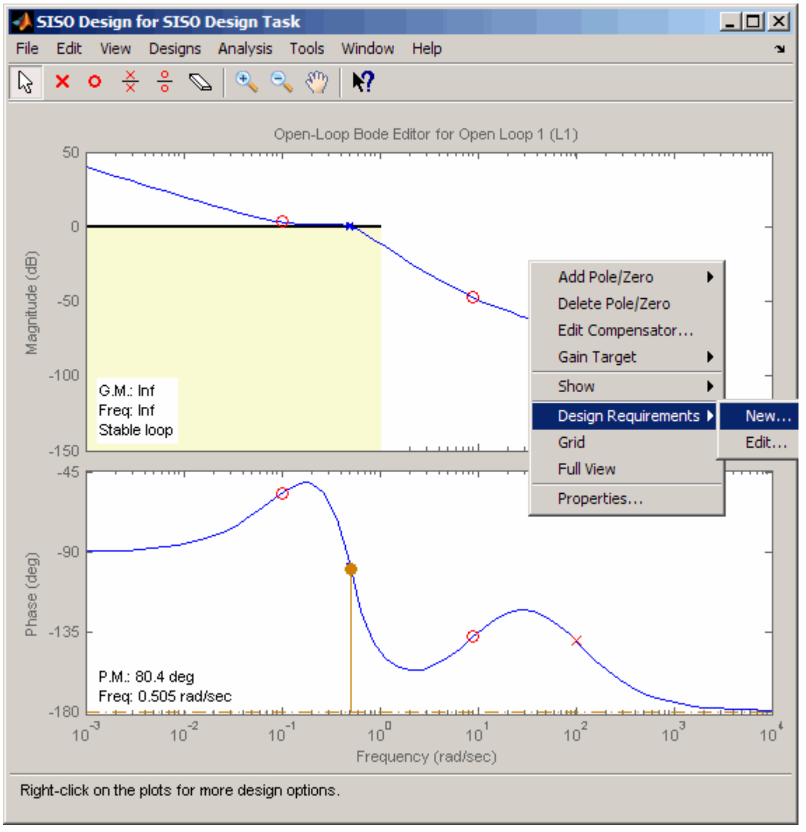


The SISO Design for SISO Design task window also updates to show the Bode plot with the design requirement displayed as the black line segment.

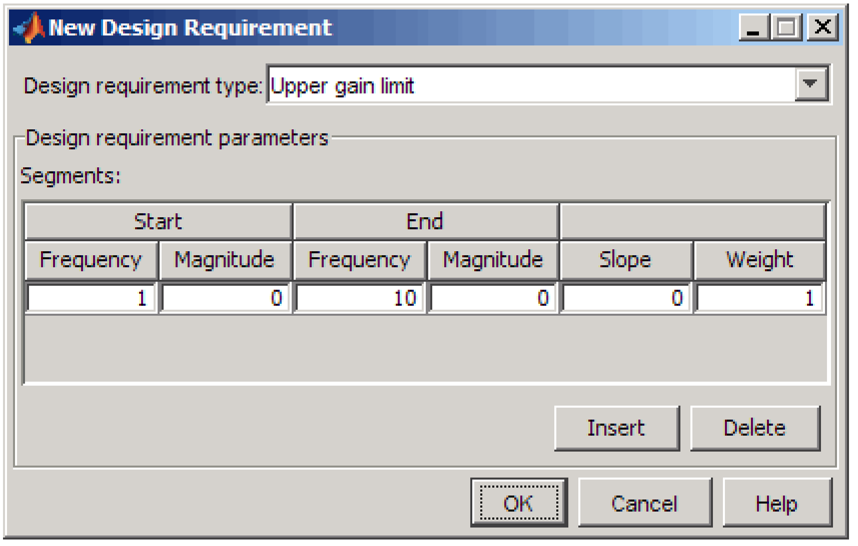


4 Add the phase margin requirement:

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

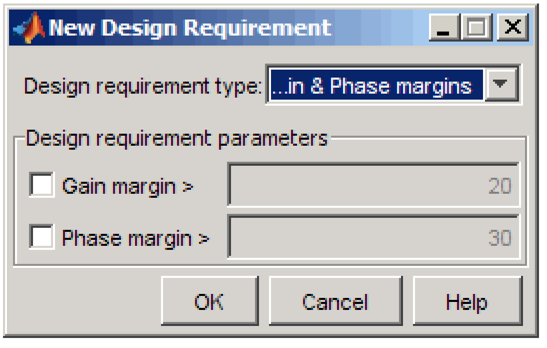


This action opens the New Design Requirement dialog box, as shown in the next figure.

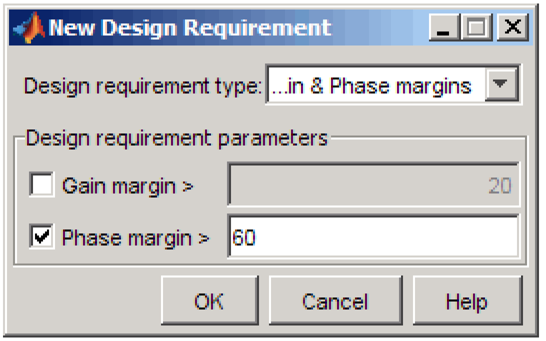


b In the New Design Requirement dialog box, select `Gain & phase margins` from the **Design requirement type** drop-down list.

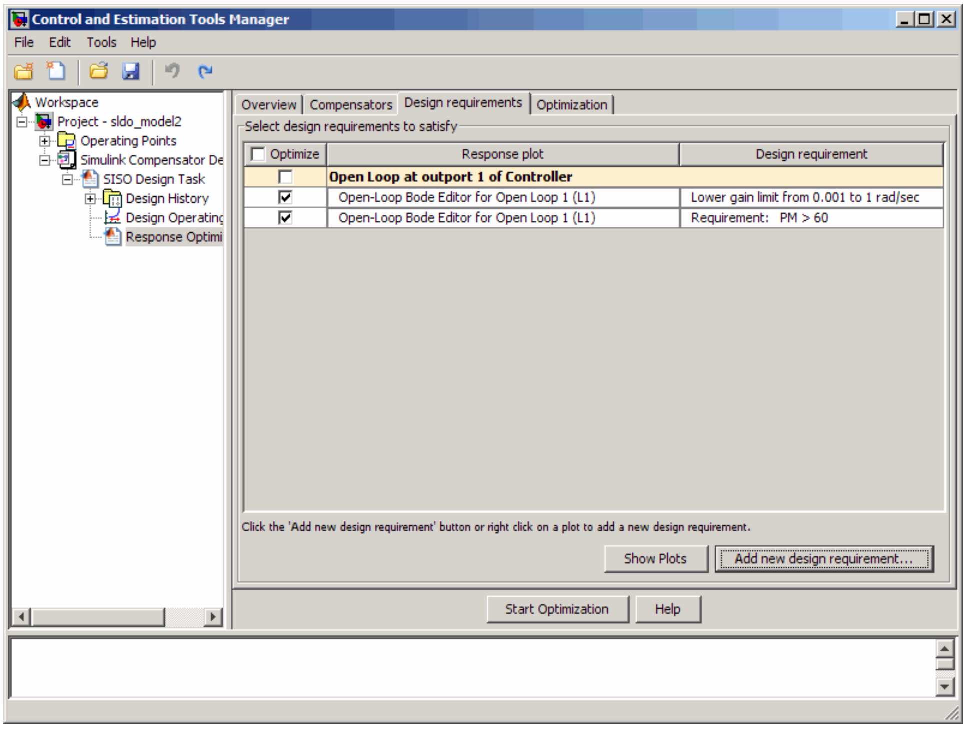
The New Design Requirement dialog box updates to display the **Gain Margin >** and **Phase Margin >** options, as shown in the next figure.



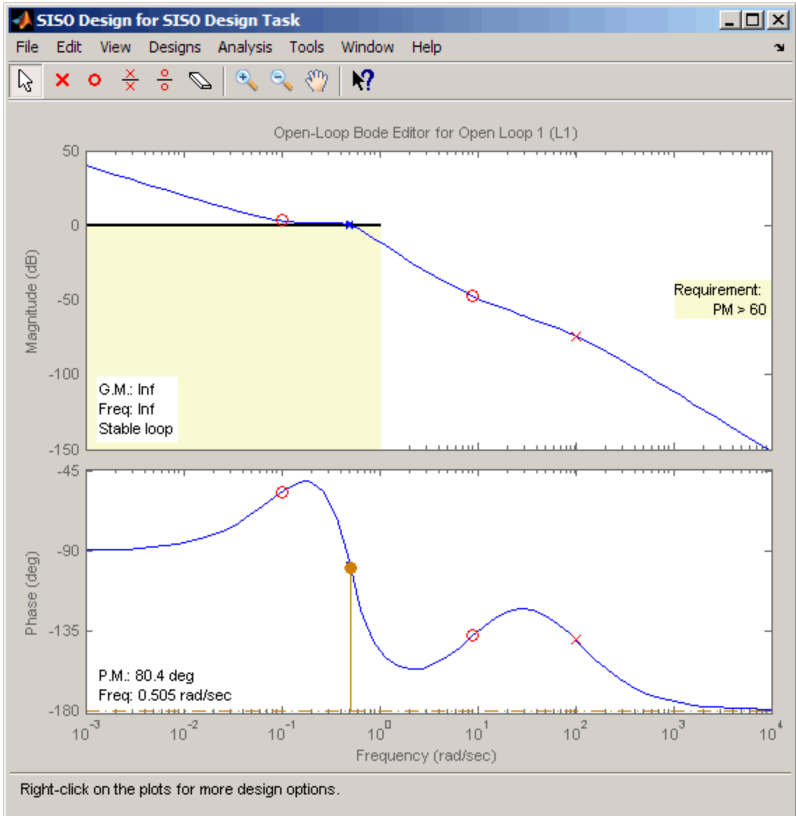
c Select the **Phase margin >** check box, and enter `60` in the adjacent field. Then, click **OK** to close the dialog box.



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.



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



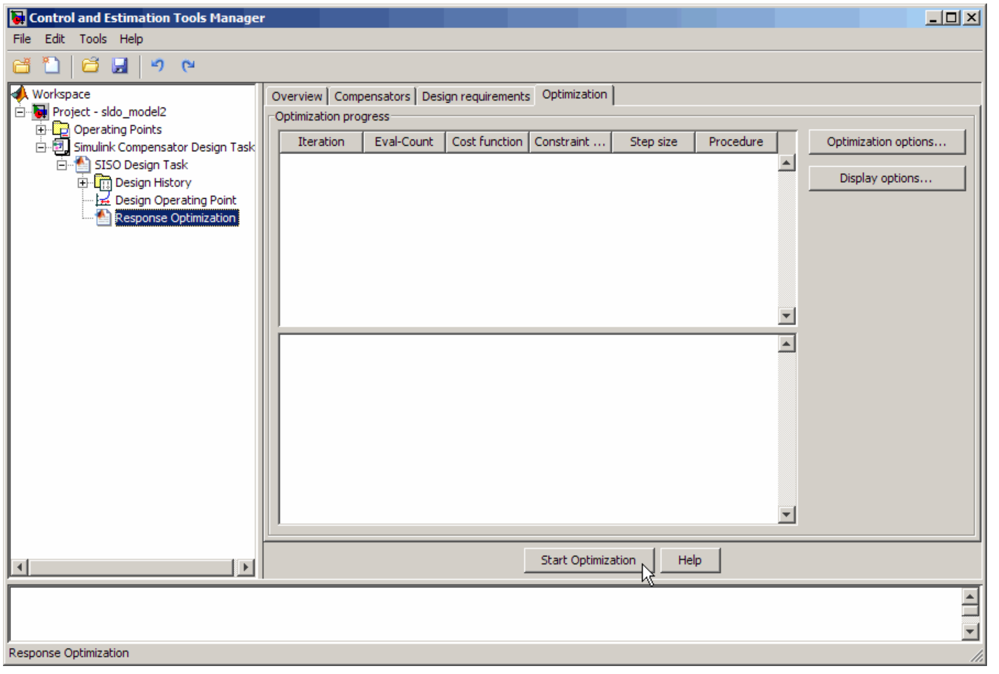
Designing the Controller

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

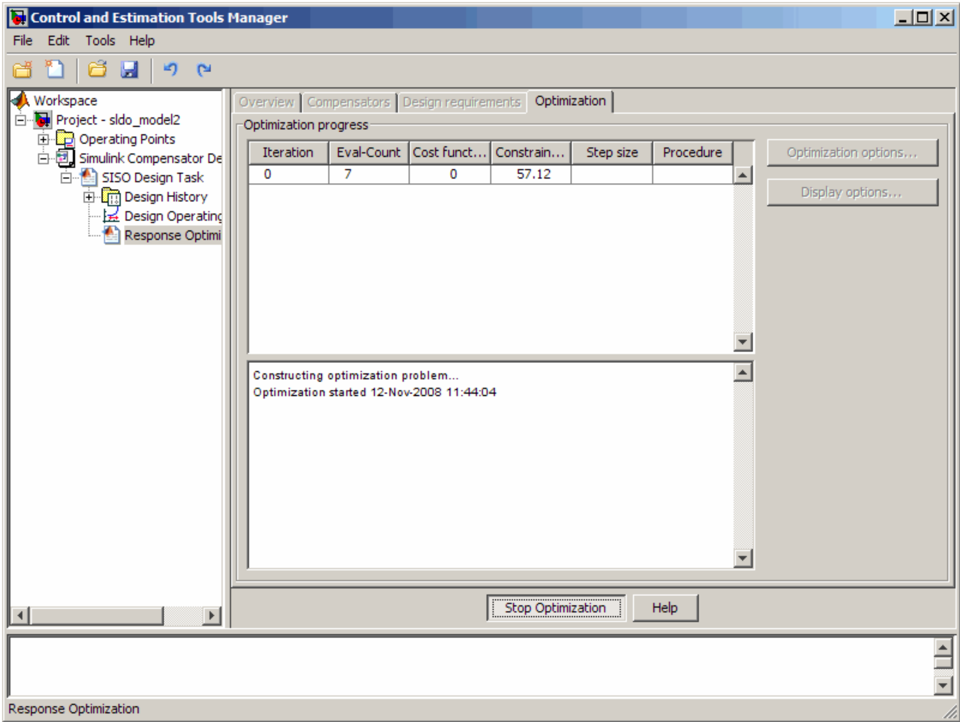
You must have already specified the controller parameters to design, as described in "Specifying the Controller Parameters" on page 4-22, and the design requirements, as described in "Specifying Bode Magnitude and Phase Margin Design Requirements" on page 4-27.

To design the controller:

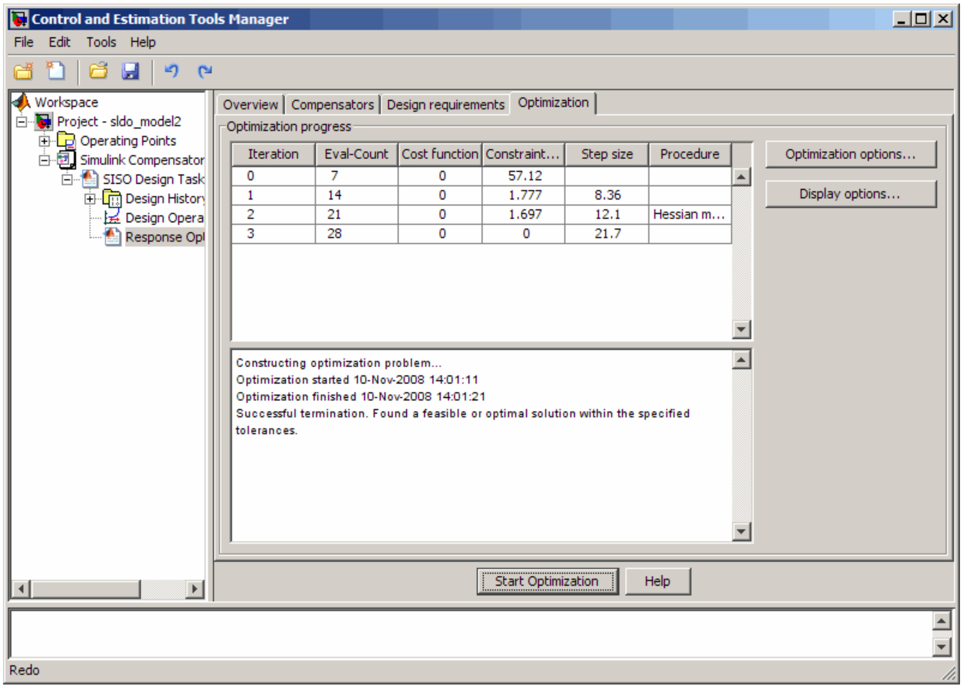
1 In the **Optimization** tab, click **Start Optimization**.



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



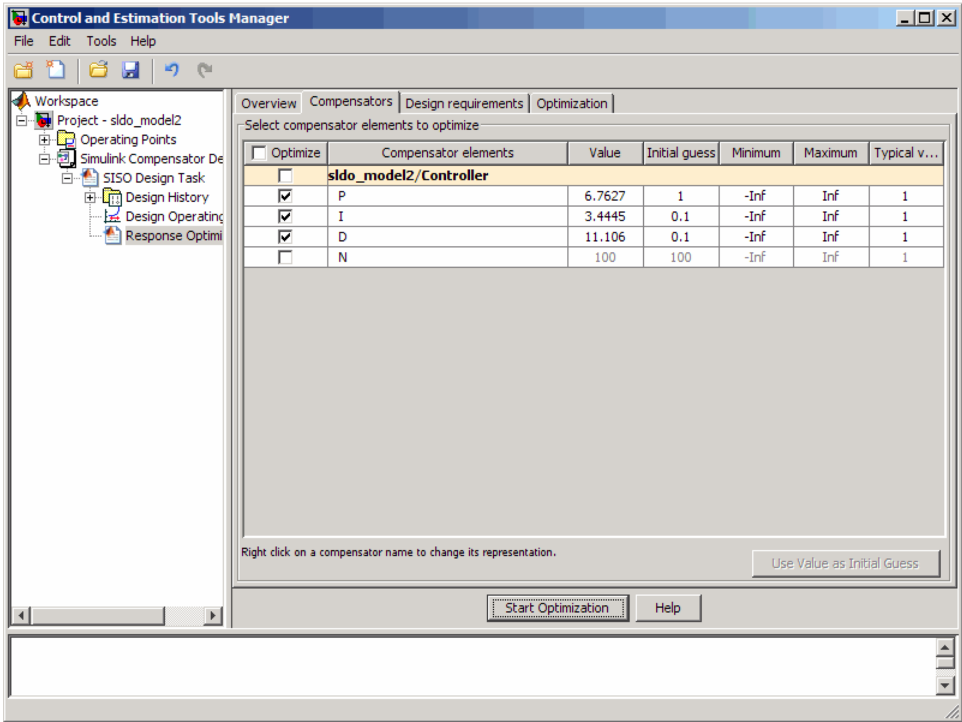
- At every optimization iteration, the default optimization method `Gradient descent` reduces the distance between the current response and the magnitude requirement line segment by modifying the controller parameters. Simultaneously, the software also computes the phase margin and reduces the distance 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.



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.

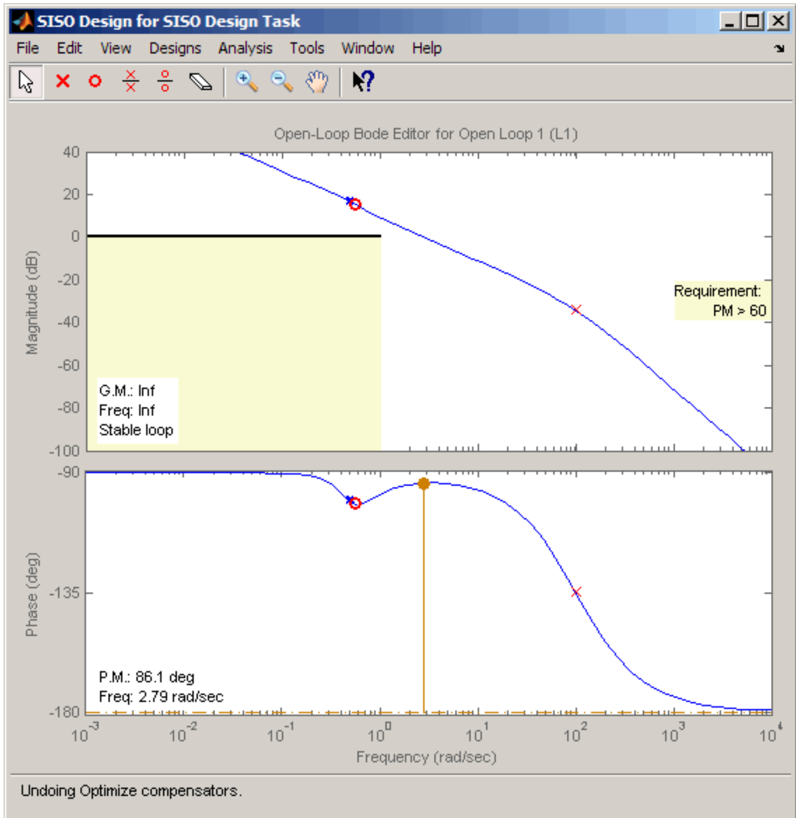
2 Examine the controller parameters and the system's response:

a View the optimized parameter values in the **Value** field of the **Compensator** tab.



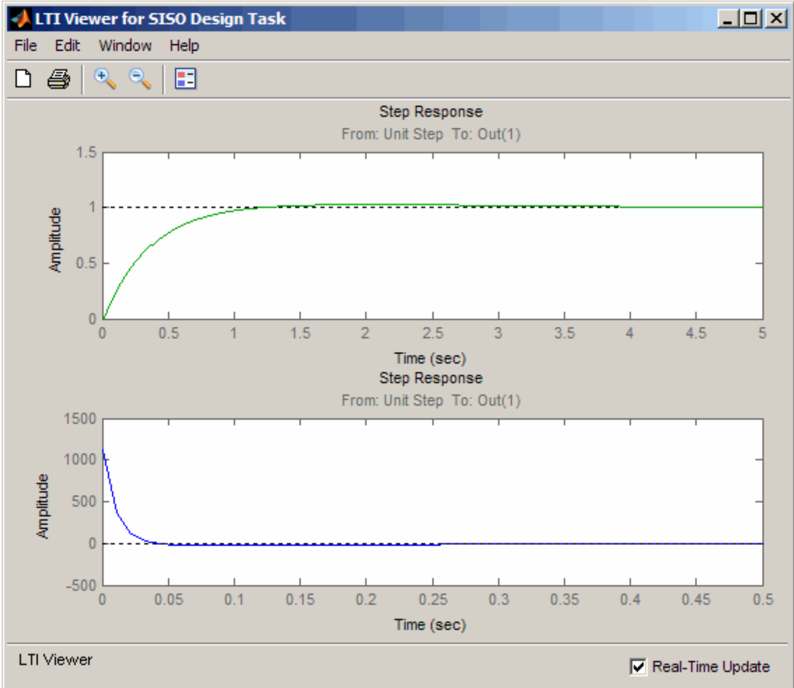
b Examine the system's response on the following plots:

- 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 4-43.

Refining the Controller Design to Meet Controller Output Bounds

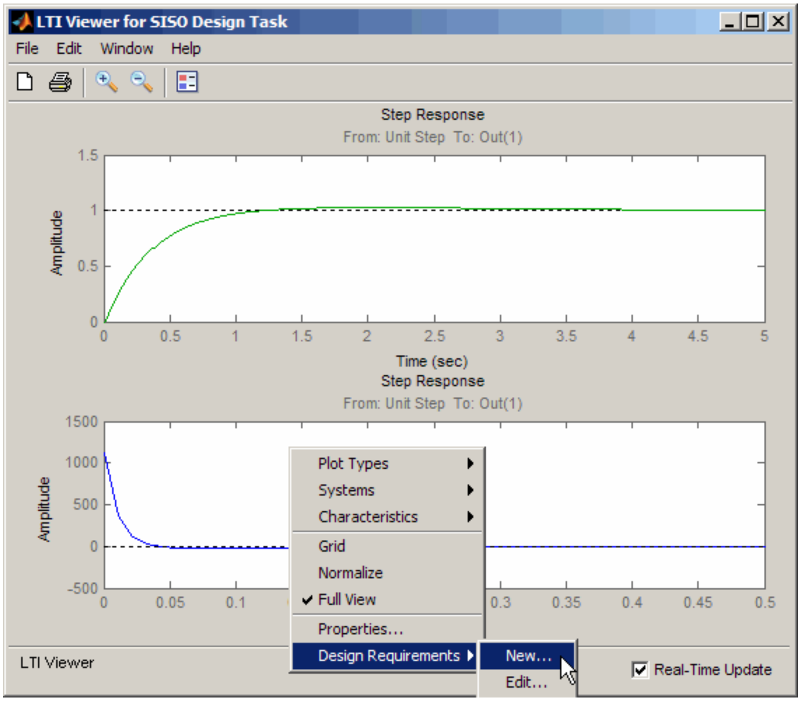
In this portion of the tutorial, you refine the controller to satisfy bounds on the controller's output.

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.

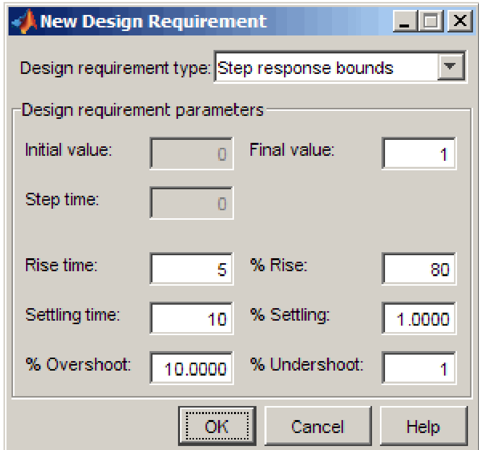
To tune the compensator parameters to meet the bounds on the controller's signal:

1 Add the upper-bound on the controller's output:

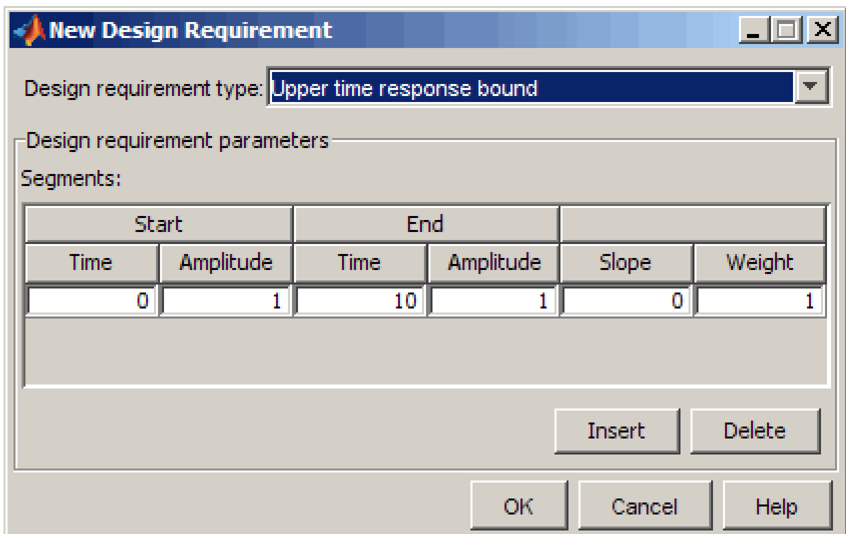
a In the LTI Viewer, right-click the white area on the bottom plot, and select **Design requirement > New**.



This action opens the New Design Requirement dialog box.



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

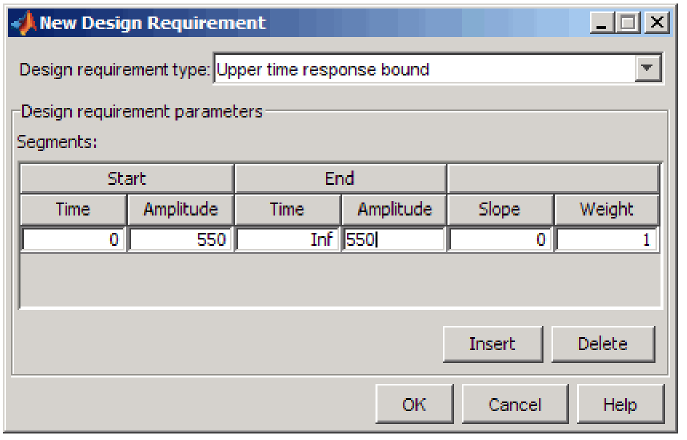


c In the **Time** field of the **End** column, enter `Inf`.

d In the **Amplitude** field of the **Start** column, enter `550`.

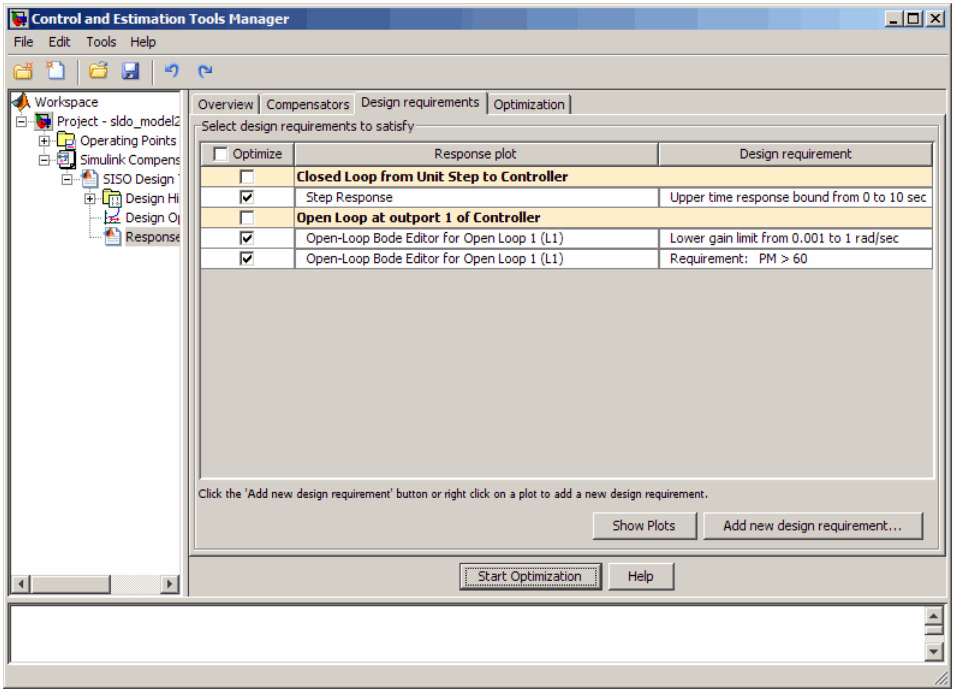
e In the **Amplitude** field of the **End** column, enter 550.

The New Design Requirements dialog box resembles the following figure.

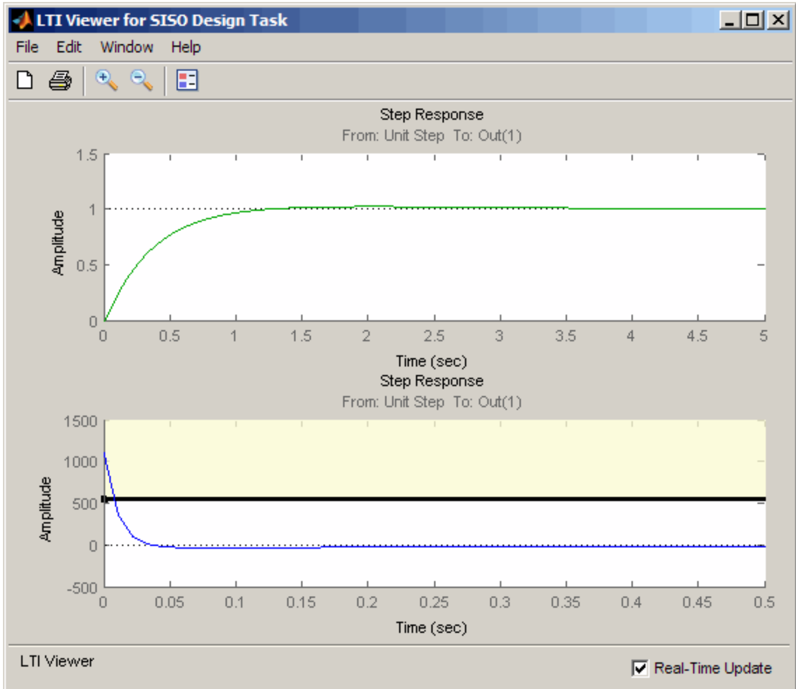


f Click **OK** to close the dialog box.

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.



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



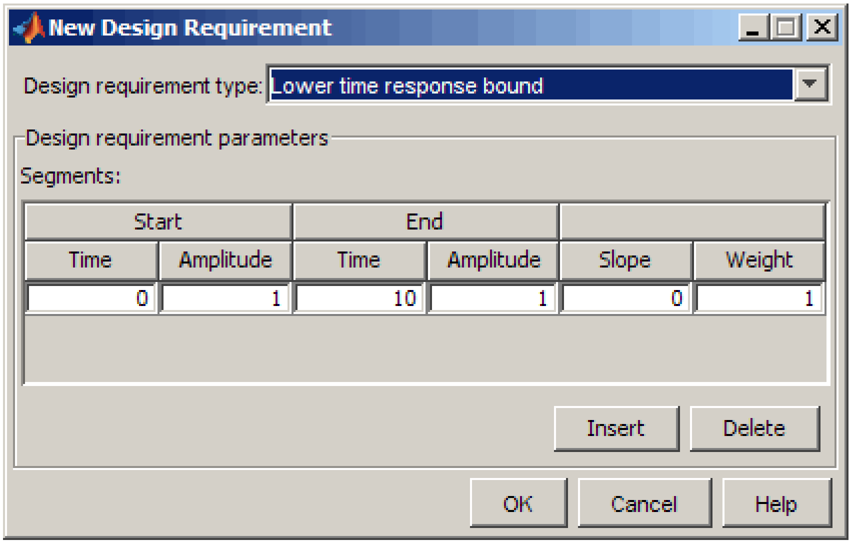
2 Add the lower-bound on the controller's output:

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.

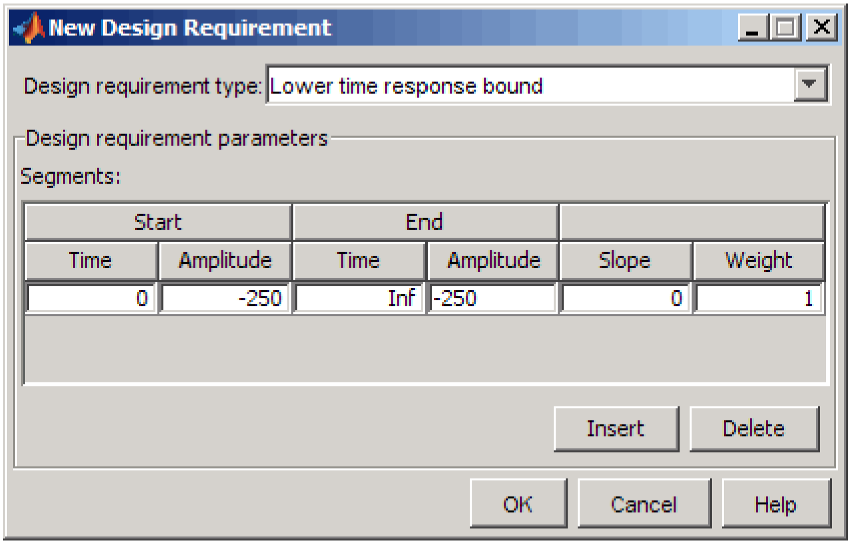


c In the **Time** field of the **End** column, enter `Inf`.

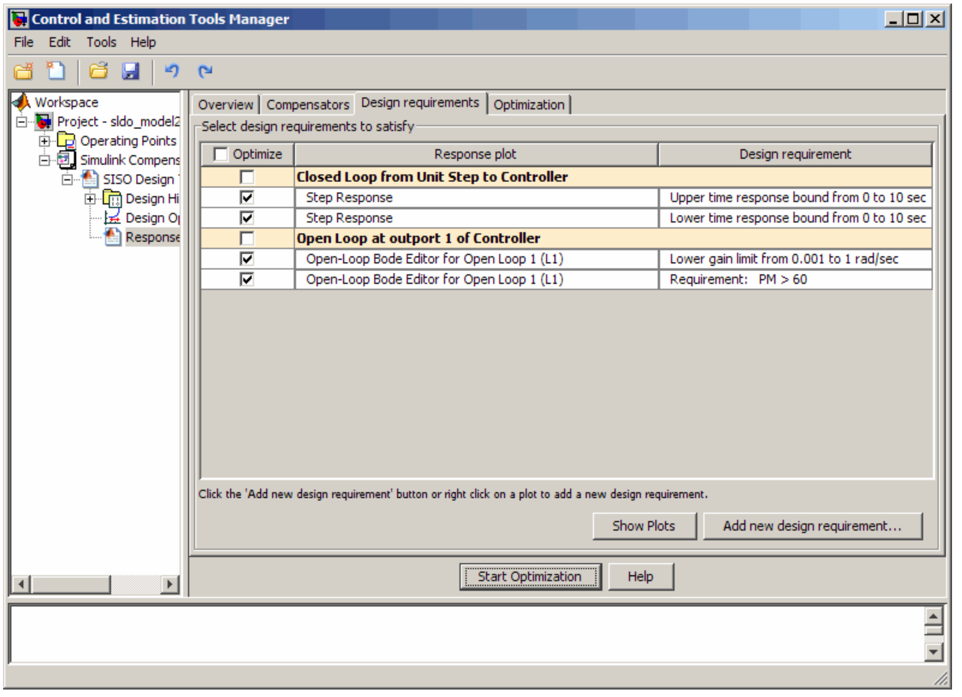
d In the **Amplitude** field of the **Start** column, enter `-250`.

e In the **Amplitude** field of the **End** column, enter `-250`.

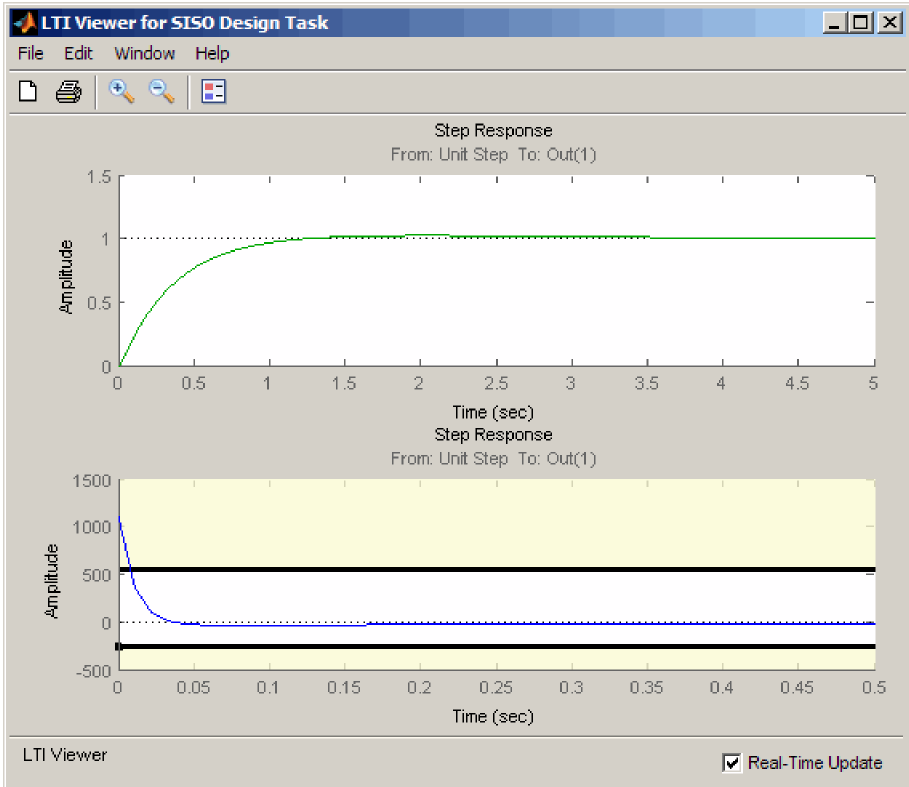
The New Design Requirement dialog box resembles the next figure. Click **OK** to close the dialog box.



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.

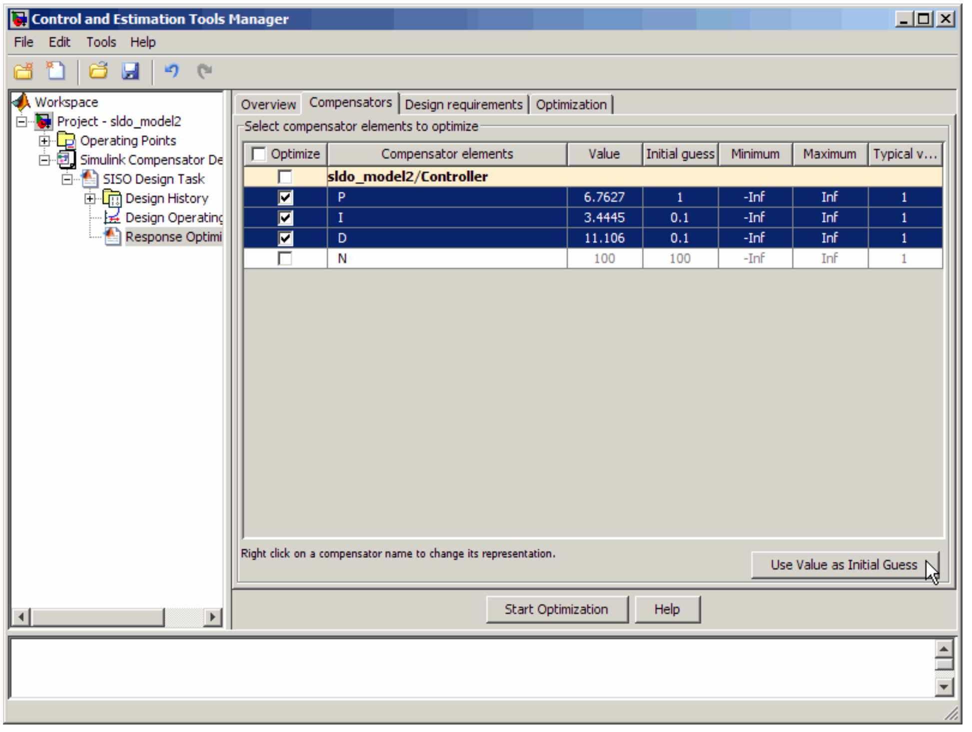


The LTI Viewer also updates to show the lower-bound on the controller's output, as shown in the next figure.



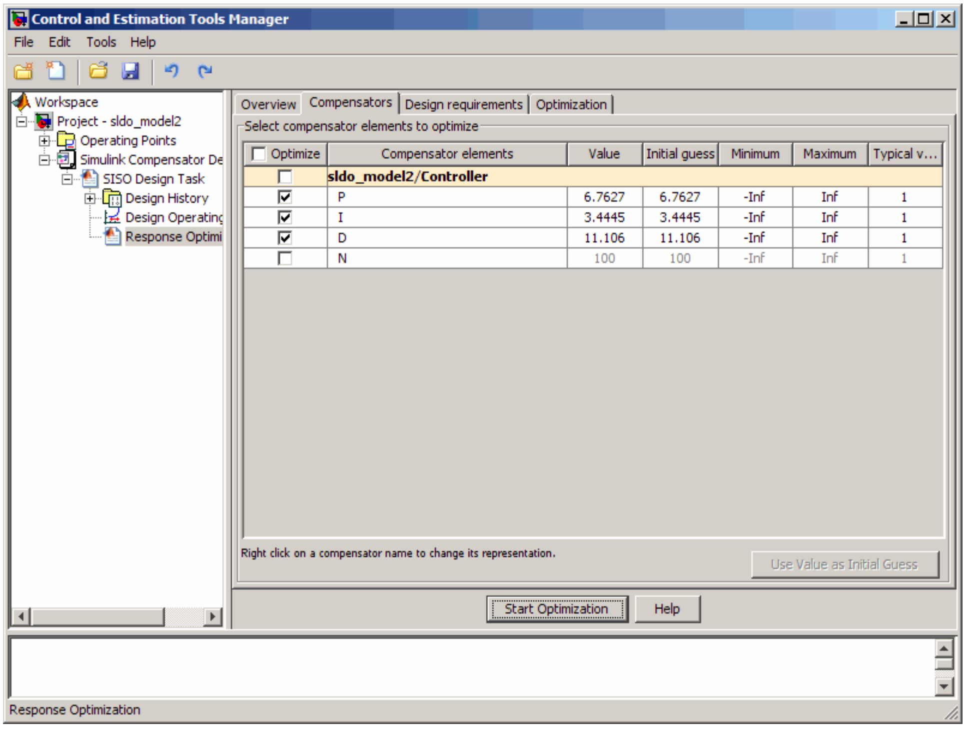
3 Optimize the parameters to meet the design requirements on the controller output:

a In the **Compensators** tab, select the rows containing `P`, `I`, and `D`, and click **Use Value as Initial Guess**.



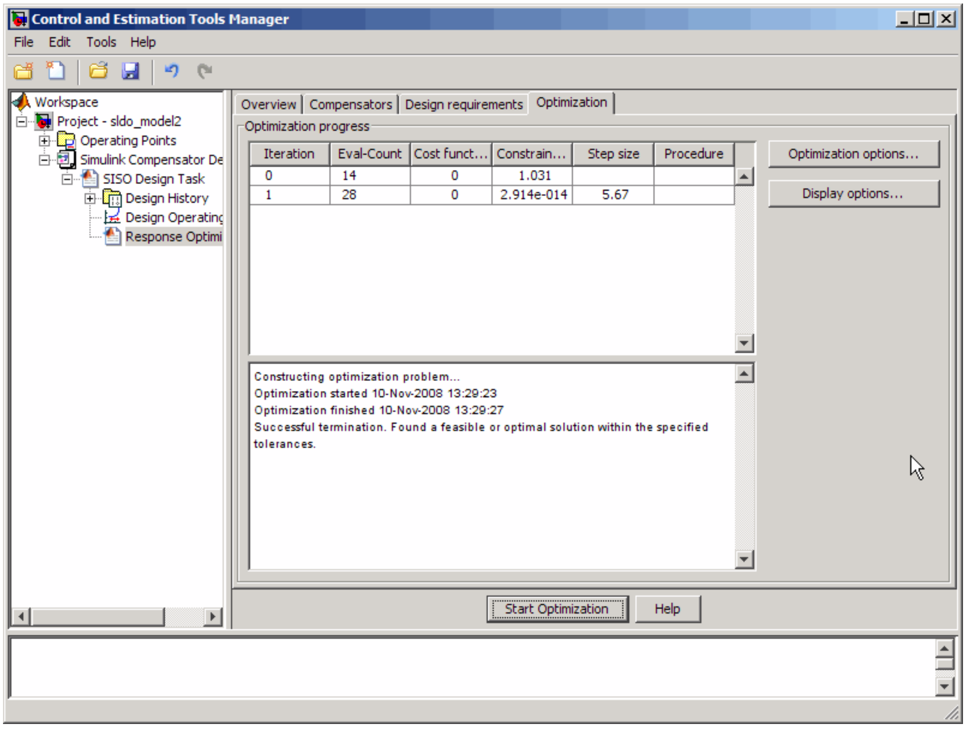
When you run the optimization again, the optimization method uses the updated parameter values as the starting point for refining the values.

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



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

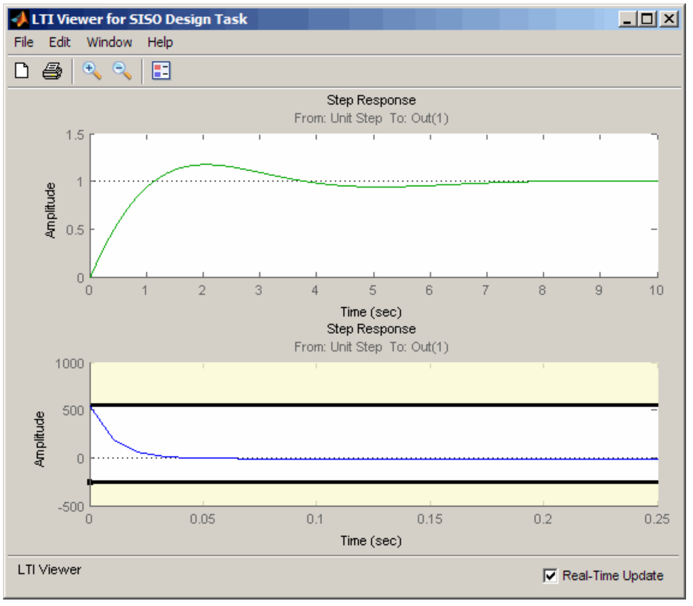
- At every optimization iteration, the optimization method reduces the distance between the current response and the upper and lower bounds on the signal.
- After the optimization completes, the **Optimization** tab resembles the next figure.



4 Examine the parameter values refined controller and the system's response:

a Examine the following response plots:

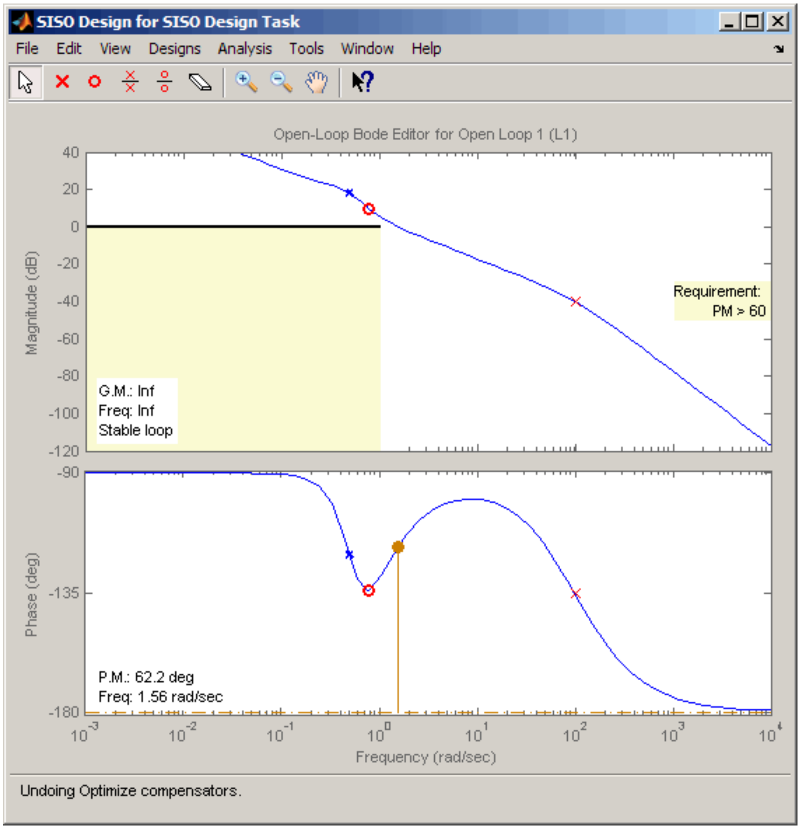
- The LTI viewer



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

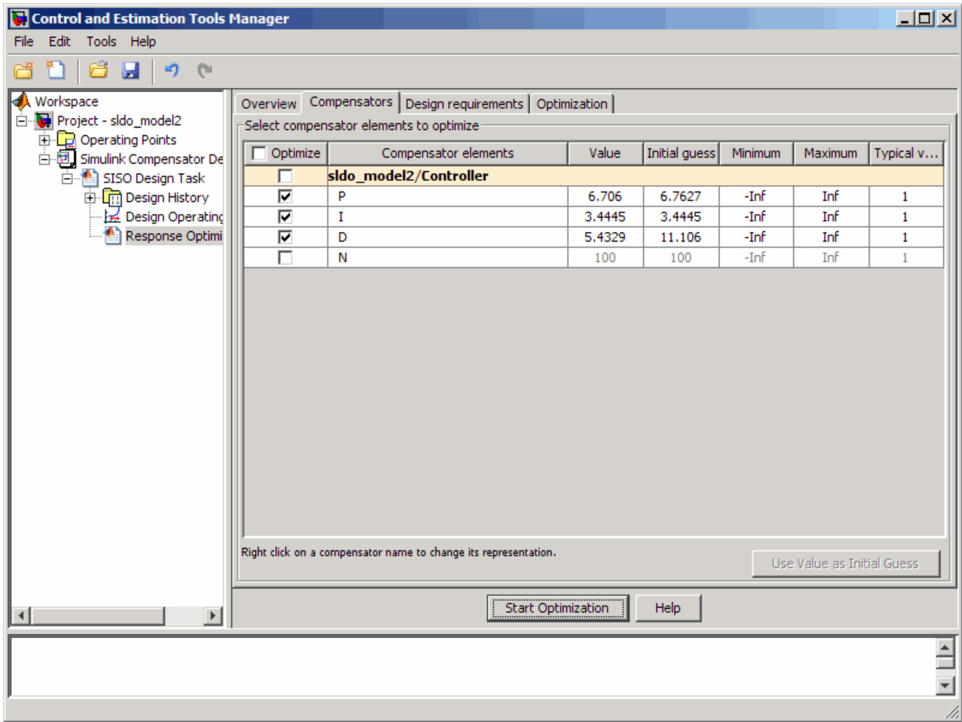
Note You must also check that the closed-loop response, shown in the top plot, remains stable after refining the controller design.

- 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.



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

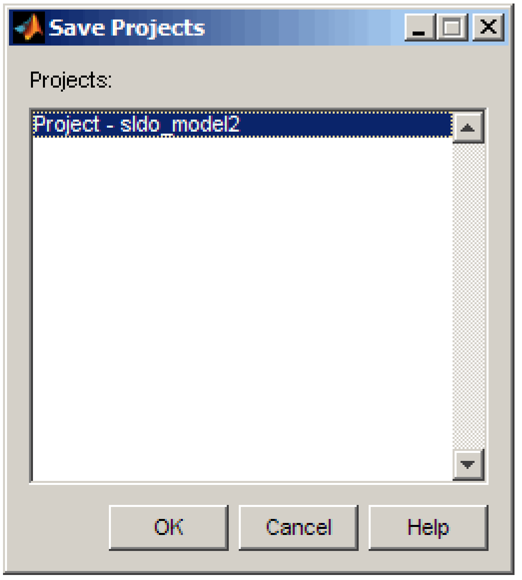
This action writes the optimized controller parameter values to the `Controller` block in the Simulink model.

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.



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.

Examples

Use this list to find examples in the documentation.

Getting Started

Index