How to Contact MathWorks

Latest news: www.mathworks.com

Sales and services: www.mathworks.com/sales_and_services

User community: www.mathworks.com/matlabcentral

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

Phone: 508-647-7000

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

Model-Based Calibration Toolbox Model Browser User's Guide

[a] COPYRIGHT 2001 2018 by The MathWorks, Inc.

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

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

Trademarks

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

Patents

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

Revision History

December 2001	Online only	New for Version 1.0 (Release 12.1)
August 2002	Online only	Revised for Version 1.1 (Release 13)
May 2003	Online only	Revised for Version 2.0 (Release 13+)
June 2004	Online only	Revised for Version 2.1 (Release 14)
June 2004	Online only	Revised for Version 2.1.1 (Release 14+)
November 2005	Online only	Revised for Version 3.0 (Release 14SP3+)
September 2006	Online only	Version 3.1 (Release 2006b)
March 2007	Online only	Version 3.2 (Release 2007a)
September 2007	Online only	Revised for Version 3.3 (Release 2007b)
March 2008	Online only	Revised for Version 3.4 (Release 2008a)
October 2008	Online only	Revised for Version 3.4.1 (Release 2008a+)
October 2008	Online only	Revised for Version 3.5 (Release 2008b)
March 2009	Online only	Revised for Version 3.6 (Release 2009a)
September 2009	Online only	Revised for Version 3.7 (Release 2009b)
March 2010	Online only	Revised for Version 4.0 (Release 2010a)
September 2010	Online only	Revised for Version 4.1 (Release 2010b)
April 2011	Online only	Revised for Version 4.2 (Release 2011a)
September 2011	Online only	Revised for Version 4.3 (Release 2011b)
March 2012	Online only	Revised for Version 4.4 (Release 2012a)
September 2012	Online only	Revised for Version 4.5 (Release 2012b)
March 2013	Online only	Revised for Version 4.6 (Release 2013a)
September 2013	Online only	Revised for Version 4.6.1 (Release 2013b)
March 2014	Online only	Revised for Version 4.7 (Release 2014a)
October 2014	Online only	Revised for Version 4.8 (Release 2014b)
March 2015	Online only	Revised for Version 4.8.1 (Release 2015a)
September 2015	Online only	Revised for Version 5.0 (Release 2015b)
March 2016	Online only	Revised for Version 5.1 (Release 2016a)
September 2016	Online only	Revised for Version 5.2 (Release 2016b)
March 2017	Online only	Revised for Version 5.2.1 (Release 2017a)
September 2017	Online only	Revised for Version 5.3 (Release 2017b)
March 2018	Online only	Revised for Version 5.4 (Release 2018a)
September 2018	Online only	Revised for Version 5.5 (Release 2018b)

Contents

1 Work ows For Modeling

What Is the Model Browser? 1-2

Set Up Designs and Models, Resume Work, or Find Engine Modeling Examples 1-4

Fit a One-Stage Model 1-7
- What Is a One-Stage Model? 1-7
- Import Data 1-7
- Fit One-Stage Models 1-7

Fit a Two-Stage Model 1-10
- What Is a Two-Stage Model? 1-10
- Import Data 1-10
- Fit Two-Stage Models 1-11

Fit a Point-by-Point Model 1-14
- What Is a Point-by-Point Model? 1-14
- Import Data 1-14
- Fit Point-by-Point Models 1-15
- Use Cases for Point-by-Point Models 1-17

2 Projects and Test Plans

Create and Reuse Modeling Templates 2-2
- Create a Test Plan 2-2
- Create and Reuse Test Plan Templates 2-2
- Save a New Template 2-3
- Reuse Stored Templates 2-3

Edit Test Plan De nition 2-5
Work With Test Plans 2-5
Edit Model Inputs 2-7
Edit Local, Global, and Response Models 2-8
Design Experiments 2-9
Select New Data 2-11
Choose Summary Statistics 2-11
View and Re ne Boundary Models 2-12
Save the Current Test Plan as a Template 2-12
Test Plan Tools 2-12

3 Designs

About The Design Editor 3-2
Introducing the Design Editor 3-2
Opening the Design Editor 3-2
Design Styles 3-3
Viewing Designs 3-4

Set Up Design Inputs 3-8

De ne Design Constraints 3-10
How to Apply Constraints 3-10
Constraint Types 3-12
Importing Constraints 3-16

Create a Space-Filling Design 3-19
Create a Space-Filling Design 3-19
Sobol Sequence 3-20
Halton Sequence 3-21
Latin Hypercube Sampling 3-22
Lattice 3-23
Strati ed Latin Hypercube 3-23
Augmenting Space-Filling Designs 3-25

Create an Optimal Design 3-28
Introducing Optimal Designs 3-28
Optimal Design: Initial Design Tab 3-29
Optimal Design: Candidate Set Tab 3-31

Optimal Design: Algorithm Tab 3-34
Averaging Optimality Across Multiple Models 3-36

Create a Classical Design 3-37
Setting Up and Viewing a Classical Design 3-39

Manipulate Designs 3-44
Adding and Editing Design Points 3-44
Merging Designs 3-47
Fixing, Deleting, and Sorting Design Points 3-48
Prediction Error Variance Viewer 3-50
Design Evaluation Tool 3-57

Saving, Exporting, and Importing Designs 3-68

Fit Models to Collected Design Data 3-69

4 Data

Using Data 4-2

Load and Edit Data 4-4
Load Data 4-4
Modify Data 4-4
View and Edit Data 4-5

Merge Data 4-10
About Data Loading and Merging 4-10
Loading and Merging Data from File 4-10
Loading Data from the Workspace 4-12
Tailor-Made Excel Sheets 4-13

Create Variables 4-15
How to Create Variables 4-15
New Variables 4-15

Create Filters 4-17
How to Create Filters 4-17
Test Filters and Test Notes 4-17

Filter Editor 4-17
Import Variables, Filters, and Editor Layout 4-18

De ne Test Groupings 4-20

Select Data for Modeling Using the Fit Models Wizard 4-22
Choose a Work ow 4-22
Opening the Fit Models Wizard 4-22
Step 1: Select Data Set 4-22
Step 2: Select Input Signals 4-23
Step 3: Select Response Models 4-24
Step 4: Set Tolerances 4-25

Match Data to Designs 4-28
Introducing the Design Match View 4-28
How to Use the Design Match View 4-28
The Tolerance Editor 4-30
What Will Happen to My Data and Design? 4-31

Data Loading Application Programming Interface 4-34

5 Setting Up Models

What Models Are Available? 5-2
What Is a One-Stage Model? 5-2
What Is a Two-Stage Model? 5-2
What Is a Point-by-Point Model? 5-2
Default Model Types 5-3
Model Types 5-4

Explore Local Model Types 5-7
Alternative Local Model Types 5-7
Local Model Class: Polynomials and Polynomial Splines 5-8
Local Model Class: Linear Models 5-11
Local Model Class: Truncated Power Series 5-13
Local Model Class: Free Knot Spline 5-14
Local Model Class: Growth Models 5-16
Local Model Class: User-De ned Models 5-23
Local Model Class: Transient Models 5-30

Local Model Class: Average Fit 5-37
Transforms 5-38
Covariance Modeling 5-39
Correlation Models 5-40

Assess Boundary Models 5-42
Default Boundary Models 5-42
Plotting Boundary Models 5-43

Explore Boundary Model Types 5-46
Alternative Boundary Model Types 5-46
Creating a New Boundary Model 5-48
Setting Up Local, Global and Response Boundary Models . . . 5-49
Combining Best Boundary Models 5-54
Editing Boundary Model Fit Options 5-57
Saving and Exporting Boundary Models 5-61

Explore Global Model Types 5-62
Alternative Global Model Types 5-62
Global Linear Models: Polynomials and Hybrid Splines 5-63
Global Model Class: Gaussian Process Model 5-68
Global Model Class: Radial Basis Function 5-69
Global Model Class: Hybrid RBF 5-73
Global Model Class: Interpolating RBF 5-74
Global Model Class: Multiple Linear Models 5-76
Global Model Class: Free Knot Spline 5-77
Global Model Class: Neural Network 5-78
Global Model Class: User-De ned and Transient Models 5-80

Add Response Models and Datum Models 5-81
Adding New Response Models 5-81
Datum Models 5-81

Create Alternative Models to Compare 5-84
Build a Selection of Models 5-84
Create a Model Template 5-85
Polynomial Template 5-86
RBF Template 5-87
Hybrid RBF Template 5-87
Free Knot Spline Template 5-88
Gaussian Process Template 5-88
Model Browser Template 5-88

6

Assess High-Level Model Trends 6-2

Assess Local Models 6-5
How to Assess Local Models 6-5
Using Local Model Plots 6-6
Removing Outliers and Updating Fits 6-8
Create Two-Stage Models 6-9
Create Alternative Local and Global Models 6-11
Viewing Local Model Statistics 6-11

Assess One-Stage Models 6-16
Assessing One-Stage, Response Feature or Global Models . . 6-16
Assess Fits Using Model Plots 6-16
Remove and Restore Outliers 6-18
Model-Speci c Tools 6-22
Create Alternative Models 6-23

Compare Alternative Models 6-25
Compare Fits Using Model Plots 6-25
Compare Fits Using Statistics 6-25
Other Model Assessment Windows 6-29

Assess Point-by-Point Models 6-31
Analyze Point-by-Point Models and Choose the Best 6-31
Edit Point-by-Point Model Types 6-32
Assess Point-by-Point Fits Using Model Plots 6-33
Assess Point-by-Point Fits Using Statistics 6-34

Model Selection Window 6-35
Comparing Models 6-35
Select a Best Model 6-36
Plots and Statistics for Comparing Models 6-36

Guidelines for Selecting the Best Model Fit 6-54
Over tting and Under tting 6-54
RMSE 6-54
PRESS RMSE and Other Statistics 6-56
Validation 6-56
Using Information Criteria to Compare Models 6-57

Assess Two-Stage Models 6-59

Model Evaluation Window 6-60
About the Model Evaluation Window 6-60
Using Validation Data 6-61

Stepwise Regression 6-65
What Is Stepwise? 6-65
Automatic Stepwise 6-65
Using the Stepwise Regression Window 6-66
Stepwise in the Model Building Process 6-71
PRESS statistic 6-74

Box-Cox Transformation 6-77

Two-Stage Models for Engines 6-81
Overview of the Mathematics of Two-Stage Models 6-81
Local Models 6-83
Local Covariance Modeling 6-83
Response Features 6-84
Global Models 6-85
Two-Stage Models 6-86
Global Model Selection 6-87
Initial Values for Covariances 6-87
Quasi-Newton Algorithm 6-88
Expectation Maximization Algorithm 6-88
References 6-88
Linear Regression 6-89
Toolbox Terms and Statistics De nitions 6-91

Export Models to Simulink 6-94
Export Models for Simulation 6-94
Use Statistical Models for Plant Modeling and Optimization 6-96
Use Statistical Models for Hardware-in-the-Loop Testing . . . 6-97

Export Models to the Workspace 6-99
Export Models to MATLAB 6-99
Work with Models in the Workspace 6-101
Evaluate Response Models and PEV 6-101
Evaluate Con dence Intervals 6-102
Evaluate Boundary Models in the Workspace 6-103

7

Radial Basis Functions for Model Building 7-2
About Radial Basis Functions 7-2
Types of Radial Basis Functions 7-3
Fitting Routines 7-11
Center Selection Algorithms 7-12
Lambda Selection Algorithms 7-14
Width Selection Algorithms 7-18
Prune Functionality 7-24
Statistics 7-26
Hybrid Radial Basis Functions 7-29
Tips for Modeling with Radial Basis Functions 7-31

Work ows For Modeling

The following sections introduce the Model Browser part of the Model-Based Calibration Toolbox product.

What Is the Model Browser? on page 1-2

Set Up Designs and Models, Resume Work, or Find Engine Modeling Examples on page 1-4

Fit a One-Stage Model on page 1-7

Fit a Two-Stage Model on page 1-10

Fit a Point-by-Point Model on page 1-14

What Is the Model Browser?

The Model-Based Calibration Toolbox product contains tools for design of experiment, statistical modeling, and calibration of complex systems. See Model-Based Calibration Toolbox Product Description . The toolbox has two main apps:

- Model Browser for design of experiment and statistical modeling
- CAGE Browser for analytical calibration

The Model Browser is a exible, powerful, intuitive graphical interface for building and evaluating experimental designs and statistical models:

- Design of experiment tools can drastically reduce expensive data collection time.
- You can create and evaluate optimal, space lling, and classical designs, and constraints can be designed or imported.
- Hierarchical statistical models can capture the nature of variability inherent in engine data, accounting for variation both within and between tests.
- The Model Browser has powerful, exible tools for building, comparing, and evaluating statistical models and experimental designs.
- There is an extensive library of prebuilt model types and the capability to build user-de ned models.
- You can export models to CAGE or to MATLAB®, or Simulink® software.

Starting the Model Browser

To open the application, type

```
mbcmodel
```

at the MATLAB command prompt.

To get started, click Fit models on the Home page. See Set Up Designs and Models, Resume Work, or Find Engine Modeling Examples on page 1-4.

See Also

Related Examples

Set Up Designs and Models, Resume Work, or Find Engine Modeling Examples on page 1-4

Fit a One-Stage Model on page 1-7

Fit a Two-Stage Model on page 1-10

Fit a Point-by-Point Model on page 1-14

Set Up Designs and Models, Resume Work, or Find Engine Modeling Examples

When you open MBC Model Fitting app, the home page helps you get started or resume work faster by accessing frequent tasks, recent projects, and featured examples.

Get started by using the buttons in the right pane for common modeling tasks: design an experiment, import data, or t models.

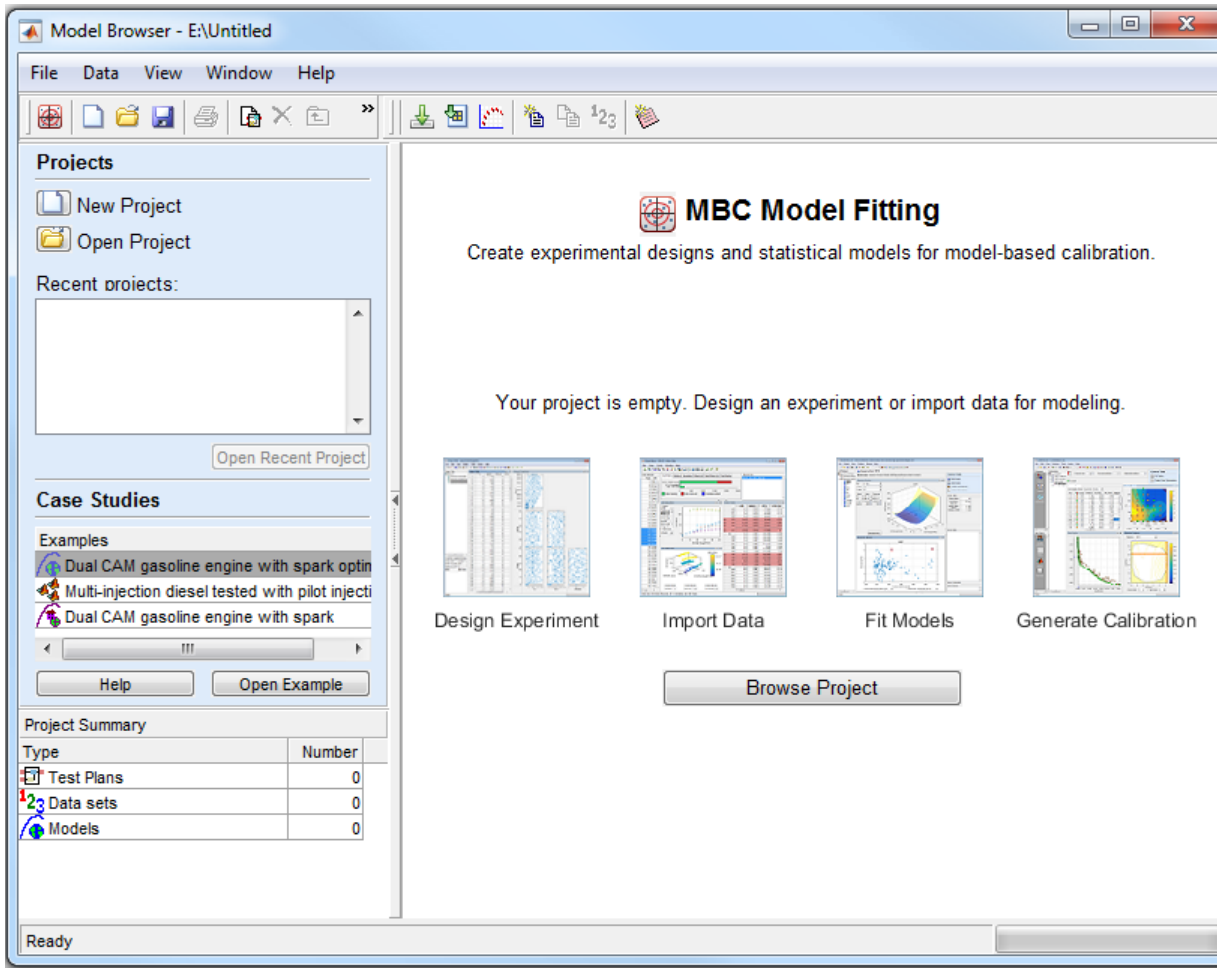
Resume work by opening projects from the Recent projects list.

Open case study examples from the Case Studies list.

View a summary of test plans, data sets, and models in your project. To open other modeling views and browse the model tree, click Browse Project.

From other modeling views, to return to the home page, click the Home toolbar button or select File > Home.

On the home page, click Generate Calibration to open the MBC Optimization App and help you import models for calibrations.



See Also

Related Examples

Fit a One-Stage Model on page 1-7

Fit a Two-Stage Model on page 1-10

Fit a Point-by-Point Model on page 1-14
Create Alternative Models to Compare on page 5-84
Set Up Calibrations, Resume Work, or Find Calibration Examples

Fit a One-Stage Model

In this section...

What Is a One-Stage Model? on page 1-7
Import Data on page 1-7
Fit One-Stage Models on page 1-7

What Is a One-Stage Model?

A one-stage model ts a model to all the data in one process. If your data inputs do not have a hierarchical structure, and all model inputs are global at the same level, then t a one-stage model.

If your data has local and global inputs, where some variables are xed while varying others, then choose a two-stage or point-by-point model instead. See Fit a Two-Stage Model on page 1-10 or Fit a Point-by-Point Model on page 1-14.

Import Data

Prepare your data before model tting.

1 In MATLAB, on the Apps tab, in the Automotive group, click MBC Model Fitting.

2 In the Model Browser home page, click Import Data.

Choose whether to import from le or workspace.

3 Use the le browser to select a le to import.

The Data Editor window opens.

4 Use the Data Editor to inspect and prepare your data. You can lter, group, and edit data, and you can de ne new variables. See Using Data on page 4-2.

Fit One-Stage Models

1 In the Model Browser home page, click Fit Models.

2 In the Fit Models dialog box, select a data set in the project from the Data set list.

If you have no data loaded, you can click Import from le in the Data pane. Use the le browser to select a le to import.

Optionally, you can select validation data as a sample of the tting data or a separate data set.

3 Click the One-Stage test plan icon in the Template pane.

4 In the Inputs and Responses pane, select data channels to use for the responses you want to model, and click the button to add to the responses.

To create a boundary model, leave the Fit boundary model check box selected. A boundary model describing the limits of the operating envelope can be useful when you are creating and evaluating designs, optimization results, and global models.

5 Select data channels to use for the model inputs, and click the button to add to the responses.

6 Click OK to t the default model types to your selected data. The toolbox calculates the t and adds a new model node to the Model Tree. The default response model type is a Gaussian process model (GPM) which can usually produce a good t rst time.

If you are using a template that you created, to override the default models, clear the Use default models for large data option.

Default Model Types	Large Data Settings for >2000 Points
Response model: Gaussian process model (GPM)	Uses the large data behavior for Gaussian process models from Statistics and Machine Learning Toolbox .
Boundary model: Convex hull t to the inputs	Switches to pairwise convex hull. Switch when 8 inputs even when <2000 points.

7 View the model t.

Functionality available for viewing and re ning the model t is described in Assess One-Stage Models on page 6-16 and Guidelines for Selecting the Best Model Fit on page 6-54.

8 After you build a single model, you should create more models for comparison, to search for the best t. Follow the guidelines in Create Alternative Models to Compare on page 5-84.

Tip To view an example project with engine data and nished models, see Gasoline Engine Calibration .

See Also

Related Examples

Assess One-Stage Models on page 6-16

Create Alternative Models to Compare on page 5-84

Set Up Designs and Models, Resume Work, or Find Engine Modeling Examples on page 1-4

Gasoline Engine Calibration

Fit a Two-Stage Model

In this section...

What Is a Two-Stage Model? on page 1-10
Import Data on page 1-10
Fit Two-Stage Models on page 1-11

What Is a Two-Stage Model?

A two-stage model ts a model to data with a hierarchical structure. If your data has local and global inputs, where some variables are xed while varying others, then choose a two-stage model. For example, data collected in the form of spark sweeps is suited to a two-stage model. Each test sweeps a range of spark angles, with xed engine speed, load, and air/fuel ratio within each test.

If your data inputs do not have a hierarchical structure, and all model inputs are global, at the same level, then t a one-stage model instead. See Fit a One-Stage Model on page 1-7

For two-stage models, only specify a single local variable. If you want more local inputs, use a one-stage or point-by-point model instead. See Fit a One-Stage Model on page 1-7 or Fit a Point-by-Point Model on page 1-14.

Import Data

Prepare your data before model tting.

1 In MATLAB, on the Apps tab, in the Automotive group, click MBC Model Fitting.

2 In the Model Browser home page, click Import Data.

Choose whether to import from le or workspace.

3 Use the le browser to select a le to import.

The Data Editor window opens.

4 Use the Data Editor to inspect and prepare your data.

Note You must de ne test groupings before two-stage modeling. See De ne Test Groupings on page 4-20. If you do not de ne test groupings, you are prompted after you try to t models.

You can lter, group, and edit data, and you can de ne new variables. See Using Data on page 4-2.

Fit Two-Stage Models

1 In the Model Browser home page, click Fit Models.

2 In the Fit Models dialog box, select a data set in the project from the Data set list.

If you have no data loaded, you can click Import from le in the Data pane. Use the le browser to select a le to import.

Optionally, you can select validation data as a sample of the tting data or a separate data set.

3 Click the Two-Stage test plan icon in the Template pane.

4 In the Inputs and Responses pane, select data channels to use for the responses you want to model, and click the button to add to the responses.

Note If you are modeling spark sweeps with a datum model, do not de ne responses at this step. Select local and global inputs and then click OK. To set up your datum model and local model types such as polynomial spline, use the Fit Models common task at the test plan node. See Datum Models on page 5-81.

To create a boundary model, leave the Fit boundary model check box selected. A boundary model describing the limits of the operating envelope can be useful when you are creating and evaluating global models and optimization results.

5 Select data channels to use for the local and global model inputs, and click the button to add to the responses.

6 Click OK to t the default model types to your selected data.

If the data does not have test groupings, the Test Groupings dialog box appears with default tests de ned by the global inputs. Verify or change the test groupings and click OK to continue model tting.

The toolbox calculates the t and adds new model nodes to the Model Tree. The default global model is a Hybrid radial-basis function (RBF) which can usually produce a good t rst time.

If you are using a template that you created, to override the default models, clear the Use default models for large data option.

Default Model Types	Large Data Settings for >2000 Tests
Local model: Quadratic Global model: Hybrid radial-basis function (RBF)	Global model switches to quadratic.
Boundary model: Convex hull t to the global inputs, and a two-stage boundary model for the local input	Global boundary model switches to pairwise convex hull. Switch when 8 inputs even when <2000 points.

The Model Browser displays the local model view if you created a single response model, or the test plan node if you created multiple response models.

7 View the t of the local models to each test. Then view the global models at the response feature nodes.

Functionality available for viewing and re ning the model t is described in Assess Local Models on page 6-5, Assess One-Stage Models on page 6-16 and Guidelines for Selecting the Best Model Fit on page 6-54.

8 When you are satis ed with the local and global models, you can build the two-stage model. Click Create Two-Stage in the Common Tasks pane.

Note You can only create the two-stage if there are exactly enough response features for the model. If you add new response features, you must choose the response features to use before you can create the two-stage model.

---

9 You are prompted to calculate the maximum likelihood estimate (MLE) at this point, if your global model types support MLE. You can do this now, or later by selecting Model > Calculate MLE. See Create Two-Stage Models on page 6-9 for a detailed explanation.

At this point, the two-stage model is calculated, and the icon changes at the local node to re ect this.

10 After you build a single model, you should create more models for comparison, to search for the best fit. Follow the guidelines in Create Alternative Models to Compare on page 5-84.

See Two-Stage Models for Engines on page 6-81 for a detailed explanation of two-stage models.

See Also

Related Examples

Assess Local Models on page 6-5

Assess One-Stage Models on page 6-16

Add Response Models and Datum Models on page 5-81

Create Alternative Models to Compare on page 5-84

Set Up Designs and Models, Resume Work, or Find Engine Modeling Examples on page 1-4

Fit a Point-by-Point Model

In this section...

What Is a Point-by-Point Model? on page 1-14
Import Data on page 1-14
Fit Point-by-Point Models on page 1-15
Use Cases for Point-by-Point Models on page 1-17

What Is a Point-by-Point Model?

Point-by-point modeling allows you to build a model at each operating point of an engine with the necessary accuracy to produce an optimal calibration. You often need point-by-point models for multiple injection diesel engines and gasoline direct-injection engines.

You can use point-by-point models to try a variety of models for your data. These can be useful if you want to try several kinds of models, especially if you think there is a lot of variation between operating points in your data. A selection of models (and any others you choose) are tted and the toolbox selects the best one for each test. In this way you can have a variety of models at once. For example, for some tests a Gaussian process model might t best, while for others a quadratic would be acceptable.

With point-by-point models, no predictions are available between operating points. If you need predictions between operating points, use a one-stage model instead. See Fit a One-Stage Model on page 1-7.

Import Data

Prepare your data before model tting.

1 In MATLAB, on the Apps tab, in the Automotive group, click MBC Model Fitting.
2 In the Model Browser home page, click Import Data.

   Choose whether to import from le or workspace.
3 Use the le browser to select a le to import.

   The Data Editor window opens.
4 Use the Data Editor to inspect and prepare your data. You can lter, group, and edit data, and you can de ne new variables. See Using Data on page 4-2.

Note You must de ne test groupings before two-stage modeling. See De ne Test Groupings on page 4-20. If you do not de ne test groupings, you are prompted after you try to t models.

Fit Point-by-Point Models

1 In the Model Browser home page, click Fit models.

2 In the Fit Models dialog box, select a data set in the project from the Data set list.

If you have no data loaded, you can click Import from le in the Data pane. Use the le browser to select a le to import.

Optionally, you can select validation data as a sample of the tting data or a separate data set.

3 Click the Point-by-Point test plan icon in the Template pane. This template lets you create point-by-point test plans with local models at each engine operating point, which is useful when testing is done at xed operating point settings. See Use Cases for Point-by-Point Models on page 1-17.

4 In the Inputs and Responses pane, select data channels to use for the responses you want to model, and click the button to add to the responses.

To create a boundary model, leave the Fit boundary model check box selected. The toolbox will t a separate boundary model of type Convex Hull to each operating point. A boundary model describing the limits of the operating envelope can be useful when you are creating and evaluating models and optimization results.

5 Select data channels to use for the local inputs and operating point inputs, and click the button to add to the responses.

6 Click OK to t the default model types to your selected data.

If you are using a template that you created, to override the default models, clear the Use default models for large data option.

If the data does not have test groupings, the Test Groupings dialog box appears with default tests de ned by the global inputs. Verify or change the test groupings and click OK to continue model tting.

The toolbox calculates the t and adds new model nodes to the Model Tree.

Point-by-point model ts automatically run in parallel if you have Parallel Computing Toolbox software.

Default Model Types	Large Data Settings for any operating point >2000 Points or >100 operating points
The toolbox ts these model types to each operating point and selects the best model: Quadratic with Stepwise: Min PRESS Cubic with Stepwise: Min PRESS Hybrid RBF with nObs/3 Gaussian process models (using defaults)	Switches to tting a single GPM per operating point (no Hybrid RBF or polynomial).
Boundary model: Point-by-point boundary model with a single convex hull t to all inputs at each operating point.	If any operating point has >2000 points, then point-by-point boundary model switches to a convex hull for every pair of inputs. Switch when 8 inputs even when <2000 points.

The toolbox selects the best model type for each test in your data using the PRESS RMSE selection criteria. For example, for some tests a Gaussian process model might t best, while for others a quadratic would be acceptable.

The Model Browser displays the point-by-point model node if you created a single response model, or the test plan node if you created multiple response models.

7 Assess the model t for each operating point at the `Point-by-Point` node.

For details about tools for viewing and re ning the model t, see Assess Point-by-Point Models on page 6-31 and Guidelines for Selecting the Best Model Fit on page 6-54.

8 Export your point-by-point models to CAGE for optimized calibration. From the test plan node, click Generate calibration in the Common Tasks pane.

Tip To view an example project with engine data and nished models, see Multi-Injection Diesel Calibration .

Use Cases for Point-by-Point Models

The point-by-point test plan template provides a convenient mechanism to model a number of tests at di erent operating points using the same set of models. Using the test plan has advantages including:

You can divide the data into tests and model it within a single test plan rather than having a separate one-stage test plan for each operating point. The toolbox does not construct two-stage models or response feature models because it is impossible to choose response features that apply to all tests, when there are di erent model types for di erent tests. You must have at least one global variable (e.g., speed, injection timing, load) and you cannot use covariance modeling.

You can also use point-by-point models in CAGE optimization, by creating an optimization from your models, or you can use the models in an existing optimization provided the global variable values are the same as the global variables used for the local models in the Model Browser.

You can export point-by-point models to le or directly into CAGE, and automatically create an optimization, a tradeo , and a dataset from your point-by-point models.

See Also

Related Examples

Assess Point-by-Point Models on page 6-31

Multi-Injection Diesel Calibration

Set Up Designs and Models, Resume Work, or Find Engine Modeling Examples on page 1-4

2

Projects and Test Plans

Create and Reuse Modeling Templates

In this section...

Create a Test Plan on page 2-2
Create and Reuse Test Plan Templates on page 2-2
Save a New Template on page 2-3
Reuse Stored Templates on page 2-3

Create a Test Plan

You need to select a test plan to construct models or designs.

If you are designing an experiment, see Set Up Design Inputs on page 3-8.

If you want to t models to data, see Model Set Up .

If you click Fit models or Design experiment in the Common Tasks pane, the dialog box guides you through the steps for setting up inputs and models. After setting up your designs or models, if you want more modeling options, you can access more controls using the test plan block diagram. See Edit Test Plan De nition on page 2-5.

Create and Reuse Test Plan Templates

Templates are useful for reusing similar modeling test plans. The procedures to model engines for calibrations are usually repeated for several di erent engine programs. The test plan template lets you reuse the setup for one test plan with another set of data, without rede ning inputs, responses, and default model settings. You can alter the loaded test plan settings without restriction.

You can reuse testplans in the current project without needing to save them to a le. Simply select any current project test plan in the Fit Models or New Test Plan dialog boxes.

A list of test plan templates appears when you t models or design an experiment.

Test plan templates store the following information:

Modeling work ow Whether the model is one- or two-stage or point-by-point, and the default models for each level, e.g., model types for local and global models.

All response models (for example, torque, exhaust temperature, emissions) If they were saved with the template (select the check box in the Test Plan Template Setup dialog box)

Numbers and names of input factors to models

Summary Statistics for display (see Summary Statistics on page 6-28)

Designs If they were saved with the template (select the check box in the Test Plan Template Setup dialog box).

The design for one type of engine might or might not be appropriate for a similar type of engine. You can rede ne or modify the design using the Design Editor.

No model child nodes are included, just the top level of the test plan (response models, and local and global models for two-stage models).

To create a test plan template, see Save a New Template on page 2-3.

To reuse your test plan template, see Reuse Stored Templates on page 2-3.

Save a New Template

From the test plan node that you want to make into a template:

Click the Make Template toolbar button or select TestPlan > Make Template.

The Test Plan Template Setup dialog box appears.

If desired, change the name of the new template and choose whether to include designs and/or response models.

The templates are stored in the folder speci ed in the File > Preferences dialog box. Place your library of templates in your template folder for quick access when creating new test plans.

Reuse Stored Templates

Tip You can reuse test plans in the current project without needing to save them to a le. Simply select any current project testplan in the Fit Models or New Test Plan dialog boxes.

When you click Fit models or Design experiment in the Common Tasks pane, a list of templates appears in the Fit Models and New Test Plan dialog boxes.

You can select any test plan in the current project in the Templates pane.

The Templates pane displays all templates found in the Current Folder, unless you have set a templates folder in the Preferences dialog box (File menu).

Use the Browse button if the required template is not in the folder.

To view some example templates, click Browse and select the `matlab\toolbox\mbc\mbctraining` folder.

Click a template to view the settings. The number of stages, input factors, and responses are displayed. Modify if desired, for example, to add responses. Use stored templates in exactly the same way as the default templates.

Click OK to use the selected test plan template.

Edit Test Plan De nition

Work With Test Plans

If you click Fit models or Design experiment in the Common Tasks pane, the dialog box guides you through the steps for setting up inputs and models. Follow the steps in Model Set Up or Set Up Design Inputs on page 3-8. If you follow those steps, then you do not need to set up anything using the test plan diagram. However, you can later edit settings from the test plan.

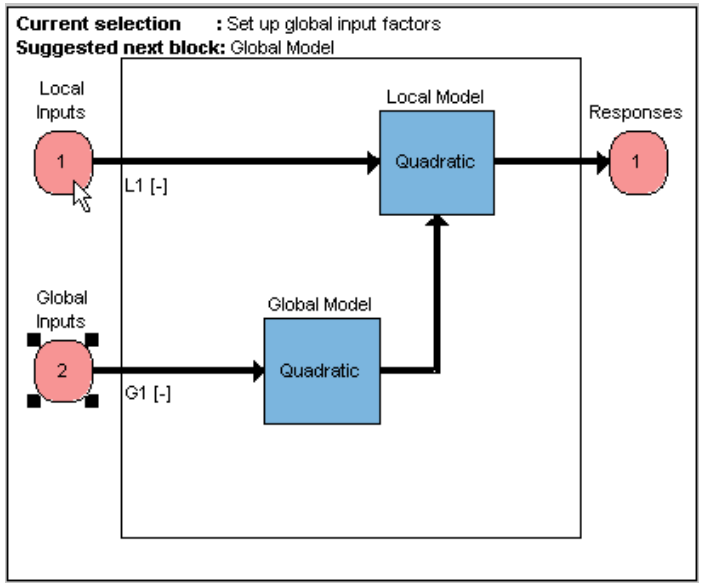
After you t models, the view at the test plan node displays the Response Models tab. View the cross-section plots of all your response models. See Assess High-Level Model Trends on page 6-2.

If you want to edit the test plan settings, click the Test Plan tab to switch back to the test plan diagram.

After you create a test plan, you can use the test plan diagram view to:

- Edit model inputs.
- Edit local, global, and response model types.
- Add new response models.
- View and edit designs, and create designs at the local level.
- View and re ne boundary models.
- Choose summary statistics.
- Select new data for modeling.

When you select a test plan node (with the icon) in the model tree (and the Test Plan tab if you already t models), then this view appears.



This example is a two-stage model. All test plan nodes show this view with a block diagram of the test plan. The diagram provides a graphical interface so you can set up inputs and set up models by double-clicking the blocks in the test plan diagram. You can also use the Test Plan menu.

Use the diagram to edit the test plan settings. Select a model block to choose the stage of the model hierarchy to use with the following menu choices:

- Set Up Model
- Design Experiment
- View Design Data
- View Model
- Summary Statistics

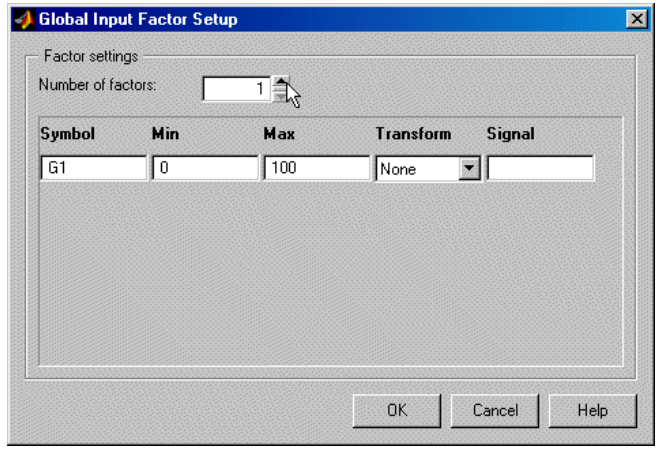
The selected Model block is highlighted in yellow if a Setup dialog box is open; otherwise it is indicated by blocks at the corners.

The following sections describe how to set up models, designs and data from your test plan.

Edit Model Inputs

Edit the number and de nition of model input factors for each stage by double-clicking or right-clicking the inports of the test plan block diagram. You can update ranges and symbols and re t existing models. Setting ranges can be important before you design experiments.

The following example shows the input setup dialog box for the global model. The dialog box for the local model contains exactly the same controls.



You can use the following controls:

Number of Factors

You can change the number of input factors using the buttons at the top.

Symbol

The input symbol is used as a shortened version of the signal name throughout the application. The symbol should contain a maximum of three characters.

Min and Max Model Range

This setting is important before you design experiments. The default range is [0.100]. There is usually some knowledge about realistic ranges for variables. If you are not designing an experiment you can use the data range as the model range later, in the data selection stage. In some cases you might not want to use the data range (for

example, if the data covers too wide a range, or not wide enough) if you are interested in modeling a particular region. In that case you can set the range of interest here.

Transform

You can use input transformations to change the input factor for designing experiments. The available input transformations are `1/x`, `sqrt(x)`, `log10(x)`, `x^2`, `log(x)`.

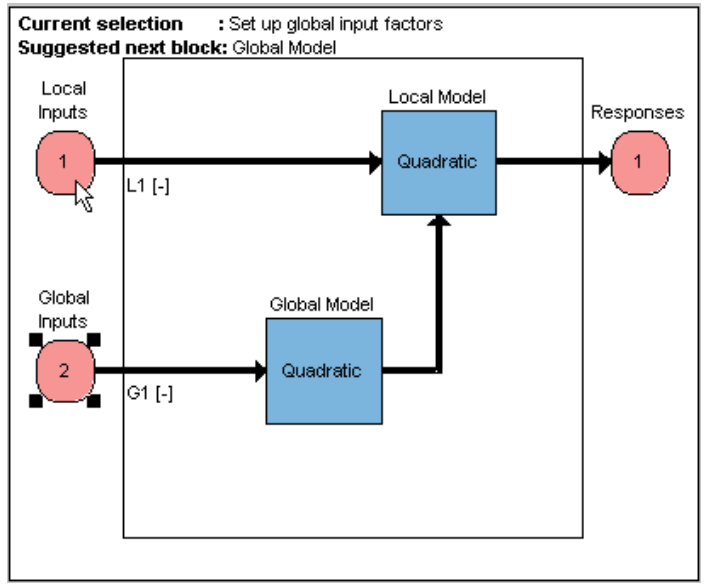
Signal

You can set up the signal name in the input factor setup dialog box. It is not necessary to set this parameter at this stage, as it can be de ned later at the data selection stage (as with the range). However, setting the signal name in this dialog box simpli es the data selection procedures, as the Model Browser looks for matching signal names in loaded data sets. When the number of data set variables is large this can save time.

Edit Local, Global, and Response Models

Set up models by double-clicking the model blocks in the test plan diagram. Select model types to set up the new default models for each stage in the model hierarchy.

The block diagram in the test plan view represents the hierarchical structure of models. Following is an example of a two-stage test plan block diagram.



See Explore Local Model Types on page 5-7 and Explore Global Model Types on page 5-62 for information on all model options.

After you set up model types, you can design an experiment, or select data for tting.

To choose data for tting, double-click the Responses block in the Test Plan diagram to open the Data Wizard. For the same result, you could also click the Select Data toolbar button (or TestPlan > Select Data menu item).

When you rst set up a test plan, the Data Wizard guides you through response model setup after the data matching functions.

To add a new response model to an existing test plan, double-click the Responses outport (or click the New button at test plan level). See Add Response Models and Datum Models on page 5-81.

Design Experiments

Note If you use the Design Experiment common task work ow, you create designs for global inputs only. If you want to create designs at the local level, for example for point-

by-point modeling, then you must open the Design Editor from the local block in the test plan diagram.

You can access the Design Editor from the test plan via the right-click menus on the model blocks, or the TestPlan menu (for a particular model you must select a model or input block before you can design an experiment). View Design Data also opens the Design Editor where you can investigate the statistical design properties of the data.

When the test plan already has a design, the design name is displayed.

You can design experiments for both stages, local and global. You open the Design Editor in several ways from the test plan level:

Right-click a Model block in the test plan diagram and select Design Experiment.

Click a stage to design for (rst or second stage) to enable the following two options:

Click the Design Experiment toolbar button .

Select TestPlan > Design Experiment.

For an existing design, View > Design Data also launches the Design Editor (also in the right-click menu on each Model block). In this case you can only view the current data being used as a design at this stage. If you enter the Design Editor by the other routes, you can view all alternative designs for that stage.

See Design of Experiments .

Viewing Designs

The view design facility enables the user to investigate the statistical properties of the current data.

From the test plan node, select the model stage you are interested in by clicking, then choose View > Design Data. Alternatively, use the right-click menu on a Model block.

This provides access to all the Design Editor and design evaluation utility functions with the current data rather than the pre-speci ed design. If you have done some data-matching to a design, each data point is used as a design point. You can now investigate the statistical properties of this design.

For two-stage models, viewing stage one (local model) designs creates a separate design for each test.

See Design Experiments on page 2-9 or the step-by-step guide in Design of Experiments in the Getting Started documentation.

Select New Data

To load new data, select Test Plan > Fit Models. See Modify Data on page 4-4.

To attach data to the test plan, double-click the Responses block in the test plan diagram to open the Data Wizard (if the project already has data loaded). Alternatively, use TestPlan > Select Data or the toolbar button . If no data is selected, this button opens the Data Wizard, and if a data set is already selected for the test plan, it takes you straight to the Data Selection views in the Data Editor.

In the Data Editor you can select data for modeling and match data to a design. For example, after the design changes, new data matching might be necessary. See Match Data to Designs on page 4-28 for details.

If a test plan already has data attached to it, details of the data set (such as name, number of records) are displayed in the right pane.

You can attach validation data to your test plan using the TestPlan menu. You can use validation data with any model except response features. When you attach validation data to your test plan, Validation RMSE is automatically added to the summary statistics for comparison in the bottom list view of response models in the test plan. See Using Validation Data on page 6-61.

If the test plan already has validation data attached to it, the name is displayed in the right pane.

Choose Summary Statistics

Right-click the global model block in the test plan diagram and select Summary Statistics to reach the Summary Statistics dialog box. In this dialog box you can choose which summary statistics you want displayed to help you evaluate models. See Summary Statistics on page 6-28.

View and Re ne Boundary Models

From the test plan you can access the Boundary Constraint Modeling functionality from the toolbar or TestPlan menu. See Explore Boundary Model Types on page 5-46.

When the test plan already has a boundary model, the right pane displays which boundary models are combined in the best boundary model.

Save the Current Test Plan as a Template

You can save the current test plan as a template using the TestPlan > Make Template command or the toolbar button . This capability can be useful for speeding up creation of subsequent projects. See Create and Reuse Test Plan Templates on page 2-2.

Test Plan Tools

The eight buttons on the left (project and node management, plus the Print and Help buttons) appear in every view level. The right buttons change at di erent levels.

In the test plan level view, the right buttons are as follows:

Design Experiment opens the Design Editor on page 3-2. Only available when a model or input has been selected in the test plan block diagram. You must specify the stage (local or global) you are designing for.

Fit Models opens the Fit Models Wizard. See Select Data for Modeling Using the Fit Models Wizard on page 4-22.

Edit Data opens the Data Editor. See Load and Edit Data on page 4-4.

Edit Boundary opens the Boundary Constraint Editor. See Explore Boundary Model Types on page 5-46.

Export to Simulink opens the Export Models dialog box. See Export Models to Simulink on page 6-94.

Generate Calibration opens CAGE and the Export to CAGE dialog box where you can choose which models to export and/or overwrite in CAGE.

Make Template opens a dialog box to save the current test plan as a template, including any designs and response models. See Create and Reuse Test Plan Templates on page 2-2.

Test Plan Menu

Edit Inputs See Edit Model Inputs on page 2-7.

Set Up Model See Explore Local Model Types on page 5-7 and Explore Global Model Types on page 5-62.

You can also reach these functions by double-clicking the input and model blocks in the test plan diagram, and both can only be used when a Model block is rst selected in the diagram. You must specify the model to set up, local or global.

Design Experiment See About The Design Editor on page 3-2.

This is also available in the toolbar and in the right-click context menu on the blocks in the test plan diagram.

Edit Boundary Opens the Constraint Modeling window. Also available in the toolbar. See Explore Boundary Model Types on page 5-46.

Summary Statistics Only enabled after you click the global model block in the test plan diagram. Opens the Summary Statistics where you can edit the statistics shown for the global models. See Summary Statistics on page 6-28.

Fit Models opens the Fit Models Wizard where you can load new data. See Select Data for Modeling Using the Fit Models Wizard on page 4-22.

Edit Data Opens the Data Editor. See Load and Edit Data on page 4-4.

Validation Data Opens a wizard to select data for validation. See Using Validation Data on page 6-61.

Make Template Opens a dialog box for saving the current test plan as a new template, with or without designs and response models. Same as the toolbar button. See Create and Reuse Test Plan Templates on page 2-2.

Export Point-by-Point Models Only enabled if you have set up a two-stage model with the correct number of inputs. Two global inputs are required (normally speed and load). This provides an interface with the Point-by-Point Tradeo in the CAGE browser part of Model-Based Calibration toolbox. This allows you to calibrate from local maps. See Edit Point-by-Point Model Types on page 6-32 for details.

View Menu (Test Plan Level)

Design Data Opens the Design Editor. The view design facility enables you to investigate the statistical properties of the collected data. This provides access to all the Design Editor and design evaluation utility functions with the current design rather than the pre-speci ed design (after data matching, the data points are used as the new design points). See About The Design Editor on page 3-2.

For two-stage models, viewing level 1 designs creates a separate design for each test.

Model Opens a dialog box showing the terms in the current model.

Both of these are only available when a model or input block is selected in the test plan block diagram.

Designs

This section discusses the following topics:

About The Design Editor on page 3-2
Set Up Design Inputs on page 3-8
De ne Design Constraints on page 3-10
Create a Space-Filling Design on page 3-19
Create an Optimal Design on page 3-28
Create a Classical Design on page 3-37
Manipulate Designs on page 3-44
Saving, Exporting, and Importing Designs on page 3-68
Fit Models to Collected Design Data on page 3-69

About The Design Editor

Introducing the Design Editor

The Design Editor provides prebuilt standard designs to allow a user with a minimal knowledge of the subject to quickly create experiments. You can apply engineering knowledge to de ne variable ranges and apply constraints to exclude impractical points. You can increase modeling sophistication by altering optimality criteria, forcing or removing speci c design points, and optimally augmenting existing designs with additional points.

There is a step-by-step guide to using the Design Editor in Design of Experiments .

Opening the Design Editor

You must rst have a test plan before you can open the Design Editor.

1 From the startup (project) view of the Model Browser, click New and select a one or two-stage test plan.

You can design experiments at both stages, for local models and global models; for most two-stage models the global model is most appropriate for design of experiment.

2 Before you design an experiment we recommend that you set up your input variables, by double-clicking the Inputs blocks on the test plan diagram. See Edit Model Inputs on page 2-7.

You can choose the number of inputs for your model and set up their names and de nitions, then you can design an experiment to collect data. It is much easier to understand your design points if they are labeled with the factor names. Also, if you do not set up model inputs rst, then you can only create designs for the default number of variables (one).

3 If you want to use optimal designs, then the type of model you are going to use to t the data is important, and you should choose a model type before opening the Design Editor. Double-click a model block in the test plan diagram to set up model types. Optimal designs are best for cases with high system knowledge, where previous studies have given con dence on the best type of model to be tted, so in these cases you should pick your model type before designing an experiment.

If you have no idea what model you are going to fit, choose a space-filling design. Model type has no effect on designs that are space-filling or classical, so if you want to create these designs you can leave the model type at the default and open the Design Editor.

You can invoke the Design Editor in several ways from the Edit Test Plan Definition on page 2-5:

1 First you must select the stage (first/local or second/global) for which you want to design an experiment. Click to select the appropriate model block in the test plan diagram.

2 Right-click the model block and select Design Experiment.

Alternatively, click the Design Experiment toolbar icon .

You can also select TestPlan > Design Experiment.

For an existing design, View > Design Data also launches the Design Editor (also in the right-click menu on each Model block). This shows the selected data as a design.

Design Styles

The Design Editor provides the interface for building experimental designs. You can make three different styles of design: classical, space-filling, and optimal.

Classical designs (including full factorial) are very well researched and are suitable for simple regions (hypercube or sphere). See Create a Classical Design on page 3-37.

Space-filling designs are better when there is low system knowledge. In cases where you are not sure what type of model is appropriate, and the constraints are uncertain, space-filling designs collect data in such a way as to maximize coverage of the factors' ranges as quickly as possible. See Create a Space-Filling Design on page 3-19.

Optimal designs are best for cases with high system knowledge, where previous studies have given confidence in the best type of model to be fitted, and the constraints of the system are well understood. See Create an Optimal Design on page 3-28.

You can augment any design by optimally adding points. Working in this way allows new experiments to enhance the original, rather than simply being a second attempt to gain the necessary knowledge. See Adding and Editing Design Points on page 3-44.

Viewing Designs

When you first create or open a design, the main display area shows the default Design Table view of the design (see example above). All the views on the right show the design selected in the left tree (see The Design Tree on page 3-4). There is a context menu for the views on the right, available by right-clicking the title bars, in which you can change the view of the design to 1-D, 2-D, 3-D, 4-D, and Pairwise Projections, 2-D, and 3-D Constraint views, and the Table view (also under View menu). This menu also allows you to split the display either horizontally or vertically so that you simultaneously have two different views on the current design. The split can also be merged again. You can also use the toolbar buttons. After splitting, each view has the same functionality; that is, you can continue to split views until you have as many as you want. When you click a view, its title bar becomes blue to show it is the current active view. See Design Display Options on page 3-5 for more information about how to change your display options.

The information pane, bottom left, displays pieces of information for the current design selected in the tree. The amount of information in this pane can change depending on what the design is capable of; for example, only certain models can support the optimal designs and only these can show current optimal values. You can also see this information and more by selecting File > Properties or using the context menu in the tree.

The Design Editor can display multiple design views at once, so while working on a design you can keep a table of design points open in one corner of the window, a 3D projection of the constraints below it, and a 2D, 3D, or pairwise plot of the current design points as the main plot.

The Design Tree

The currently available designs are displayed on the left in a tree structure.

The tree displays three pieces of information:

- The name of the design, which you can edit by clicking it
- The state of the design
  - The icon changes from if it is empty, to the appropriate icon for the design type when it has design points (for example, optimized, as in the toolbar buttons for Optimal, Classical, and Space-Filling designs).

The icon changes to when design points have been added using a di erent method (for example, augmenting a classical design with optimally chosen points). It becomes a custom design style. You can mix and match all design options in this way.

A padlock appears ( ) if the design is locked. This happens when it has child nodes (to maintain the relationship between designs, so you can retreat back up the design tree to reverse changes).

The design that is selected as best. This is the default design that is used for matching against experimental data. The icon for the selected design is the normal icon turned blue. When you have created more than one design, you should select as best the design to be used in modeling, using the Edit menu. Blue icons are also locked designs, and do not acquire padlocks when they have child nodes.

You can reach a context menu by right-clicking in the design tree pane. Here you can delete or rename designs and add new designs. Choose Evaluate Design to open the Design Evaluation window. Properties opens the Design Properties dialog box, which displays information about the size, constraints, properties (such as optimality values), and modi cation dates of the selected design.

Design Display Options

In the View menu:

Current View Changes the current view to your selection from the sub menu:

- Design Table
- 1D Design Projection
- 2D Design Projection
- 3D Design Projection
- 4D Design Projection
- Pairwise Design Projections
- 2D Constraints
- 3D Constraints
- Model Description

View Options these items depend on the currently selected view:

Plot Properties For 1D, 2D and 3D Design Projections. Opens dialog boxes for con guring details of the current display. You can change basic properties such as color on the projections (1D, 2D, 3D, and 4D). You can rotate all 3D views as usual.

Edit Colormap For the 3D and 4D Design Projections. You can also double-click the color bar to edit the colormap.

Graph Size For the Pairwise Projections, you can choose graph size or to display all graphs.

Value Filter For the table view, you can set up a lter to selectively display certain ranges of values.

Display Design Point Numbers You can select this option to toggle the display of design point numbers in views that support the feature. A design point number is the index of a particular point in the design: this value is permanently displayed in the table view. Views that support the display of design point numbers are

2D Design Projection

3D Design Projection

4D Design Projection

Pairwise Design Projections

Because all these views are projections that use a subset of the design's input factors, it is often the case that the resulting view contains points that have been plotted on top of each other. In this case, the design point numbers will stack up in a column above the common point to aid readability. You can use Display Design Point Count to see at a glance how many points are overlapping in any stack. You can select point count or point numbers but not both.

Note Displaying multiple views with design point numbers for large designs can signi cantly slow down the display. You might want to turn o the design point number display in these cases.

Display Design Point Count If points are plotted on top of each other (in 2D, 3D, 4D, or pairwise plots) this option allows you to see how many points are overlapping in each cluster. A number next to a point indicates that more than one point is plotted there.

Print to Figure This option copies the current view into its own gure, allowing you to use the standard MATLAB plotting tools to annotate and print the display.

In the Tools menu:

Prediction Error Variance Viewer Opens the Prediction Error Variance Viewer where you can evaluate the predictive power of your designs. See Prediction Error Variance Viewer on page 3-50.

Evaluate Designs Opens the Design Evaluation window where you can examine detailed mathematical properties of your design. Also in the context menu in the design tree. See Design Evaluation Tool on page 3-57.

See Also

Related Examples

De ne Design Constraints on page 3-10

Create a Space-Filling Design on page 3-19

Create an Optimal Design on page 3-28

Create a Classical Design on page 3-37

Adding and Editing Design Points on page 3-44

Saving, Exporting, and Importing Designs on page 3-68

Merging Designs on page 3-47

Importing Constraints on page 3-16

Fixing, Deleting, and Sorting Design Points on page 3-48

Set Up Design Inputs

1. Open the Model Browser by typing

   ```
   mbcmodel
   ```

   at the MATLAB command prompt.
2. In the project node view, in the Common Tasks pane, click Design Experiment.
3. In the New Test Plan dialog box, select a template to de ne the structure of your model inputs: one-stage, two-stage, point-by-point, any user-de ned templates, or any test plans in the current project. If you re not sure which to choose, see Model Set Up to learn about one-stage, two-stage, or point-by-point models.
4. Set up your model inputs. Use the following controls:

   Number of Factors

   You can change the number of input factors using the buttons at the top.

   Symbol

   The input symbol is used as a shortened version of the signal name. The symbol should contain a maximum of three characters.

   Min and Max

   De ne the Min and Max model range. This setting is important before you design experiments. The default range is [0.100]. There is usually some knowledge about realistic ranges for variables and you might be interested in modeling a particular region. Set the range of interest here.

   Transform

   You can use input transformations to change the input factor for designing experiments. The available input transformations are `1/x`, `sqrt(x)`, `log10(x)`, `x^2`, `log(x)`.

   Signal

   You can set up the signal name in the input factor setup dialog box. It is not necessary to set this parameter at this stage, as it can be de ned later at the data selection stage (as with the range). However, setting the signal name in this dialog box simpli es the data selection procedures, as the Model Browser looks for matching signal names in loaded data sets. When the number of data set variables is large this can save time.

Click OK.

The Model Browser displays the test plan with your specified inputs.

5 In the test plan node view, in the Common Tasks pane, click Design Experiment.

The Design Editor window opens.

6 Define constraints for the design. See Define Design Constraints on page 3-10.

7 Create designs. See Create a Space-Filling Design on page 3-19, Create an Optimal Design on page 3-28, or Create a Classical Design on page 3-37.

8 Export designs. See Saving, Exporting, and Importing Designs on page 3-68.

9 After you collect data, return to the Model Browser to import data and fit models. In the test plan node view, in the Common Tasks pane, click Fit models. See Fit Models to Collected Design Data on page 3-69.

Define Design Constraints

How to Apply Constraints

In many cases designs might not coincide with the operating region of the system to be tested. For example, an automobile engine normally does not operate in a region of low speed (n) and high exhaust gas recirculation (EGR). You cannot run 15% EGR at 1000 RPM. There is no point selecting design points in impractical regions, so you can constrain the candidate set for test point generation.

Designs can have any number of geometric constraints placed upon them. Each constraint can be one of four types: an ellipsoid, a hyperplane, a 1D lookup table, or a 2D lookup table.

Note When you add constraints, the design type changes to `Custom` (except optimal designs). For space-filling and classical designs you can no longer access the original design generation settings in the Design Properties dialog box. If you want to preserve your original design settings, create a child design to constrain. This is important if you want to augment a space-filling design. See Augmenting Space-Filling Designs on page 3-25.

To add a constraint to a design:

1 Select Edit > Constraints from the Design Editor menus.

2 In the Constraints Manager dialog box, you can add new constraints, and you can delete, edit, duplicate or negate existing constraints. Use the NOT button to negate a constraint, for example if you want to constrain points to be outside a boundary model.

If there are no constraints yet, the Constraints Manager is empty and you can only click Add to construct a new constraint.

To construct a new constraint,

1 Click Add.

2 The Constraint Editor dialog box with available constraints appears. You can select the following from the Constraint Type drop-down menu: Linear, Ellipsoid, 1D Table and 2-Table. See the next section, Constraint Types on page 3-12.

3 After de ning any constraint, click OK. Your new constraint appears in the Constraint Manager list box. Click OK to return to the Design Editor, or Add to de ne more constraints.

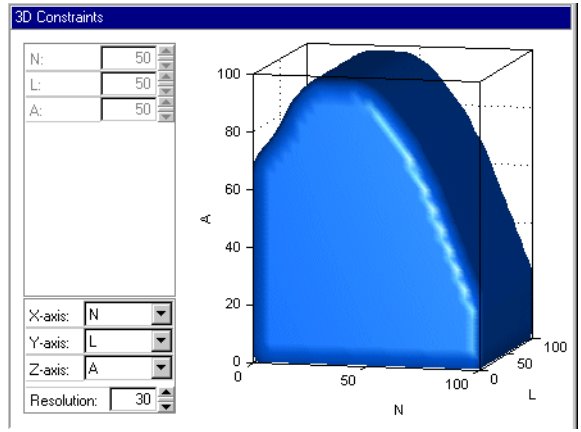
A dialog box appears if there are points in the design that fall outside your newly constrained candidate set. You can simply continue (delete them) or cancel the constraint. Fixed points are not removed by this process. For optimal designs you can also replace them with new random points within the new candidate set, as shown in the preceding example gure.

Note You only get the Replace points option for optimal designs. If you want to replace points removed by constraints from other designs, you can always use Edit > Add Point to add points optimally, randomly, or at chosen places. However, if so many points have been removed by a constraint that there are not enough left to t the current model, optimal addition is not possible. See Adding and Editing Design Points on page 3-44.

To view constraints in the Design Editor:

1 Right-click the Design Editor display pane to reach the context menu.

2 Select Current > View > 3D Constraints. (You can also select 2D constraints view). See the following gure for an example.



These views are intended to give some idea of the region of space that is currently available within the constraint boundaries.

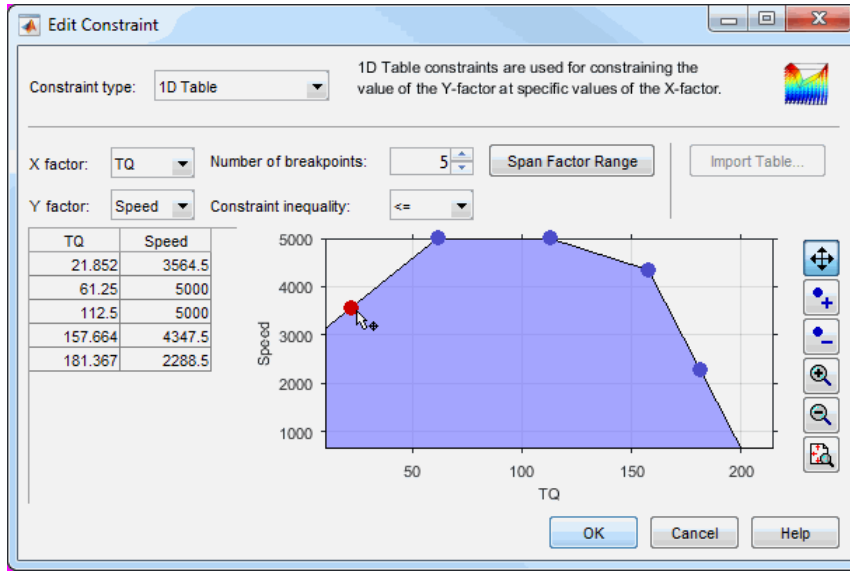
Constraint Types

1D Table Constraints on page 3-12
2D Table Constraints on page 3-15
Linear Constraints on page 3-13
Ellipsoid Constraints on page 3-13

Note These constraint types are the same in the Design Editor and in optimizations in the CAGE Browser part of the Model-Based Calibration Toolbox product.

1D Table Constraints

1D table constraints limit the maximum or minimum setting of one factor as a function of another factor. Linear interpolation between user-de ned points is used to specify the constraint. You can use either the edit boxes or the plot to de ne the constraint.



Select the appropriate factors to use for X and Y, and choose whether to constrain above or below the de ned boundary using the Constraint inequality list.

You can enter the Number of breakpoints, click the button to Span Factor Range to space your breakpoints evenly, and you can enter breakpoint values into the table. You can use a CAGE table or normalizer to de ne a constraint. If CAGE is open and contains a suitable table, you can click Import Table.

On the plot you can add and remove points using the buttons, and click and drag the points to de ne the boundary. You can also enter values in the edit boxes for the selected point.

Linear Constraints

You specify the coe icients of the equation for an (N-1) dimensional hyperplane in the N-factor space. The form of the equation is A.x = b where A is your de ned coe icient vector, x is the vector of values of the factor(s) to be constrained, and b is a scalar. For example,

In two dimensions: A=(1, 2), x=(L, A), b=3

Then A.x = b expands to

1*L + 2*A = 3

Rearranging this, you can write it as

A = -L/2 + 3/2

which corresponds to the traditional equation of a 2D straight line, y = mx + c, with m = -1/2 and c = 3/2. A.x = b is thus the higher dimensional extension of this equation.

The linear constraints work by selecting the region below the de ned plane (that is, A.x <= b). To select a region above the plane, multiply all your values by -1: A -> -A, b -> -b.

For example, to select a simple plane where SPK<50 as a constraint boundary, enter 1 next to SPK and 50 next to b. You can set all the other factors to 0 (or you can remove them on the Inputs tab if you are constraining an optimization).

Ellipsoid Constraints

The ellipsoid constraint allows you to de ne an N-dimensional ellipsoid. You can specify the center of the ellipsoid, the length of each axis, and the rotation of the ellipsoid.

Ellipsoid center

You can specify the center of the ellipsoid by entering values in the Center point columns. These are the values, in natural units, that mark where you want the ellipsoid to

be centered in each of the factor dimensions. The defaults are the midpoint of each factor range.

Axis length

You specify the size of the ellipsoid by entering values along the diagonal of the matrix. The default values create an ellipsoid that touches the edge of the space in each of the factor dimensions. In general, for an entry value X in the diagonal, the ellipsoid size in that factor is 1/sqrt(X).

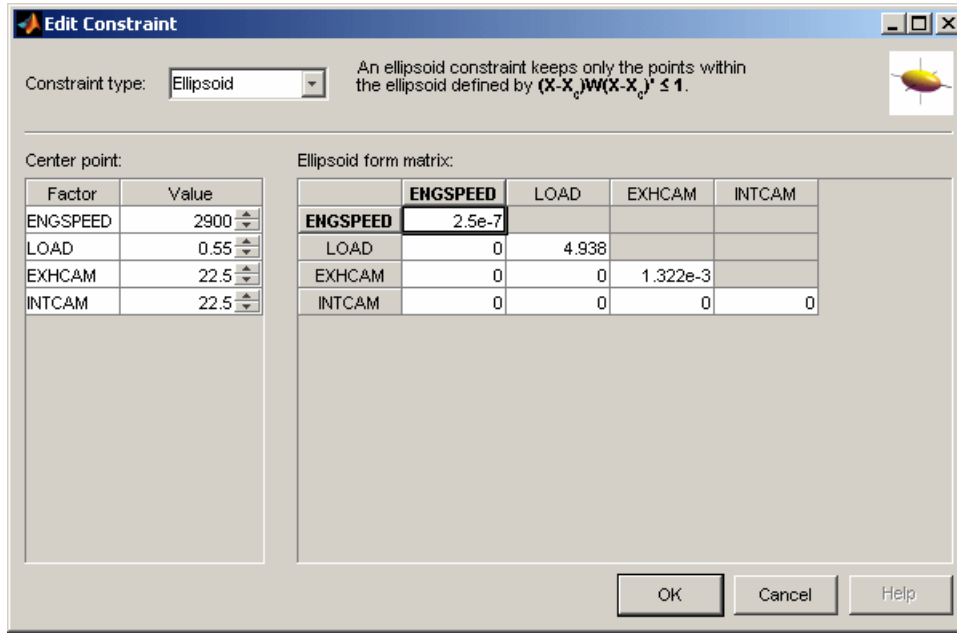
If you want a radius of r in a factor, enter 1/(r^2). For example, if you want to restrict N to a radius of 2000 from the center point, enter 1/2000^2= 2.5e-7, as shown in the example below.

Enter a zero in the diagonal to not constrain with respect to that factor.

Rotation

The matrix entries that are not on the main diagonal control rotation of the ellipsoid.

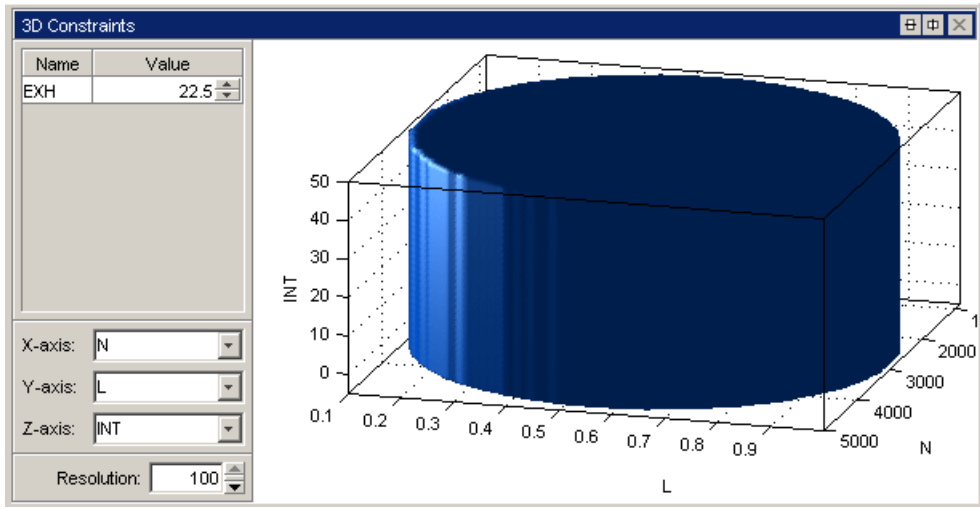
The following example shows a de ned ellipsoid constraint.



You must enter values in the table to de ne the ellipsoid. If you leave the values at the defaults, the candidate set is a sphere.

In this example, entering 2.5e-7 in the ENGSPEED diagonal restricts that axis to 1/ sqrt(2.5e-7) = 2000. Entering zero in the INTCAM diagonal leaves INTCAM unconstrained (that is, the constraint is a cylinder extending to the ends of the INTCAM factor range). The ellipse is not rotated as the non-diagonal matrix entries are all zero.

A 3D display to show the shape of this example constraint in the Design Editor can be seen below.



2D Table Constraints

2D table constraints are an extension of the 1D table. Constraint boundary values for a factor are speci ed over a 2D grid of two other factors.

You can specify these grid locations by entering values in the top row and left column, while the matrix of values for the third factor is entered in the rest of the edit boxes. To specify grid values, you can enter values directly or just choose the number of breakpoints for your grid and space them over the factors' ranges, using the controls described below.

You can specify the number of breakpoints for the X and Y factors.

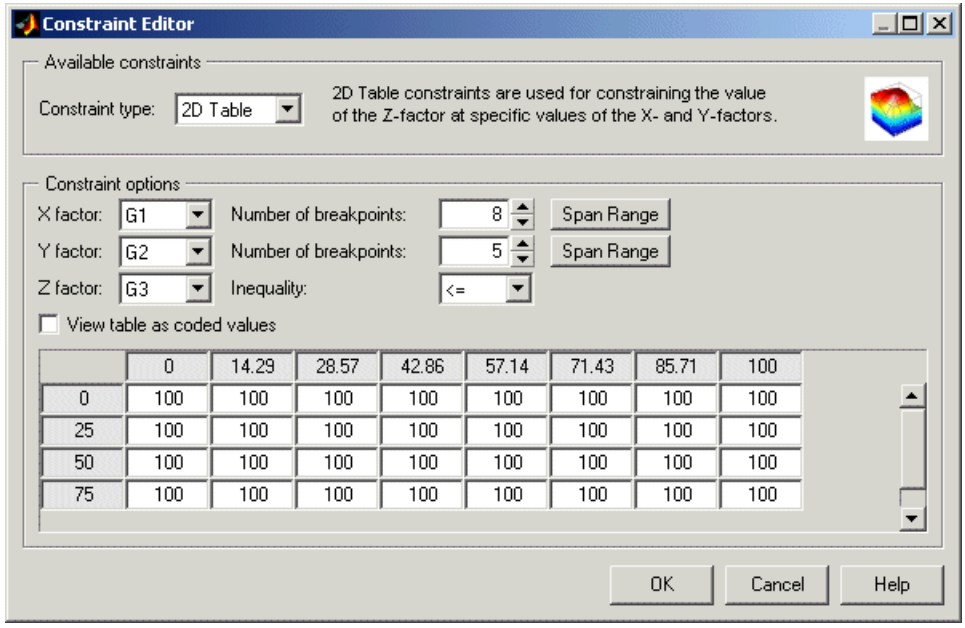
You can click Span Range to space your breakpoints evenly over the range of X or Y. This is useful if you add some breakpoints, as new points are often all at the maximum

value for that factor. It is much quicker to use the Span Range button than to change points manually.

You can specify to keep the region below (<=) or above (>=) the constraint boundary, as for the 1D table, using the Inequality drop-down menu for the Z factor.

You can switch to coded values using the check box. See the example.

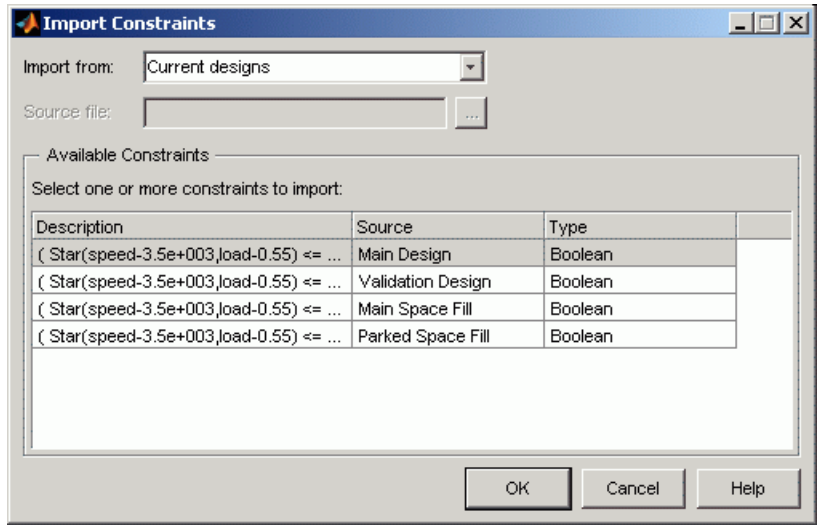
The constraint boundary between the de ned grid points is calculated using bilinear interpolation.



See also 1D Table Constraints on page 3-12

Importing Constraints

Select Edit > Constraints. In the Constraints Manager dialog box, click Import.



In the Import Constraints dialog box, select where you want to import the constraints from.

Import from	Description
Current designs	Import any existing constraints in the design tree.
Design editor file (*.mvd)	Extract constraints from a design le.
Boundary constraints (current project)	Import boundary constraints from the Model Browser project.
Boundary constraints (*.mat file)	Extract boundary constraints from le.

Import from	Description
`Classifiers`	Import classi ers from the workspace. Use classi ers to di erentiate between good and bad operating conditions when you specify boundary constraints for design experiments. To create support vector machines (SVMs) or discriminant classi ers: 1 Use the Statistics and Machine Learning Toolbox to classify data. 2 Use the Classi cation Learner App (Statistics and Machine Learning Toolbox) to export the classi cation model to the workspace.

Note You can only import design constraints from designs that have the same number of factors and have the same coded range for each factor. For designs of N factors you can import boundary constraints with N or less active factors.

1 If importing from a le you can type the lename in the edit box or use the browse button to locate the le.

2 Click to select constraints in the Available Constraints list, or Ctrl+click to select multiple constraints.

3 Click OK to import and apply the constraints.

If importing boundary constraints a dialog box appears (once for each constraint) where you can match up factor names.

See Also

More About

Classi cation Learner App (Statistics and Machine Learning Toolbox)

Support Vector Machine Classi cation (Statistics and Machine Learning Toolbox)

Create a Space-Filling Design

Create a Space-Filling Design

Space- lling designs should be used when there is little or no information about the underlying e ects of factors on responses. For example, they are most useful for a new type of engine, with little knowledge of the operating envelope. These designs do not assume a particular model form. and aim is to spread the points as evenly as possible around the operating space. Space- lling designs ll out the n-dimensional space with points that are in some way regularly spaced. These designs can be especially useful in conjunction with nonparametric models such as radial basis function (a type of neural network).

1 Add a new design by clicking the button in the toolbar or select File > New.

2 Select the node in the tree by clicking. An empty Design Table appears if you have not yet chosen a design. Otherwise, if this is a new child node the display remains the same, because child nodes inherit all the parent design's properties.

3 Select Design > Space Filling > Design Browser, or click the Space Filling Design button on the toolbar.

4 A dialog box appears if there are already points from the previous design. You must choose between replacing and adding to those points or keeping only xed points from the design. The default is replacement of the current points with a new design. Click OK to proceed, or Cancel to change your mind.

The Space Filling Design Browser appears.

Note As with the Classical Design Browser, you can select the types of design you can preview in the Space Filling Design Browser from the Design > Space Filling menu in situations when you already know the type of space- lling design you want.

For all design types, you can edit these settings:

Select from the Design type drop-down menu to choose a space- lling design style. The default Design type is Sobol Sequence.

Specify the Number of points by typing in the edit box or using the controls.

Observe the information displayed to see how many points are excluded by constraints.

The design editor tries to provide points as close as possible to the number of points you specified within the constraints.

You can use the tabs under the display to view 2D, 3D, and 4D previews. The preview is identical to the final design.

When you edit settings for very large designs, you can clear the check box Automatically update preview to avoid waiting for the preview calculation. This check box is cleared automatically if the current design is large enough to cause preview calculation to be very slow. Click the Generate button when you want to create a preview.

You can set the ranges for each factor.

For settings for specific design types, see the sections for each type:

- Sobol Sequence on page 3-20
- Halton Sequence on page 3-21
- Latin Hypercube Sampling on page 3-22
- Lattice on page 3-23
- Stratified Latin Hypercube on page 3-23

Click OK to calculate the design and return to the main Design Editor.

Tip To preserve the space-filling sequence in case you want to add more points later, make a copy of a design before rounding or sorting points.

Sobol Sequence

Sobol sequence designs are generated from the `sobolset` class in the Statistics and Machine Learning Toolbox software. The Sobol sequence is a low-discrepancy (t,s)-sequence in base 2. For more information see the Statistics and Machine Learning Toolbox documentation.

Settings

For Sobol sequence designs, you can choose the following options:

Use the radio buttons to specify whether and how to skip initial points from the sequence:

No skip Do not skip any points.

This property at the command line is `SkipMode 'None'`.

Skip initial 2^k points Automatically chooses the smallest value for k so that 2^k is larger than the number of points requested, and then skip 2^k points.

This property at the command line is `SkipMode '2^k'`.

Custom skip Enter a value to be used as the number of initial points to miss out from the sequence.

This property at the command line is `SkipMode, 'Custom', Skip, Numberofpoints`.

Apply Matousek A ine Owen scramble Performs a linear scramble of the generator matrices for the sequence using random lower-triangular matrices in base 2 and also applies a random digital shift to the points.

This property at the command line is `Scramble`.

Halton Sequence

Halton Sequence designs are generated from the `haltonset` class in the Statistics and Machine Learning Toolbox software. The Halton sequence is a low-discrepancy point set where the coordinate values for each dimension are generated by forming the radical inverse of the point's index, using a di erent prime base for each dimension. For more information see the Statistics and Machine Learning Toolbox documentation.

Settings

For Halton sequence designs, you can choose the following options:

Leap sequence points using prime number Uses only every k-th point in the Halton sequence. k is the next prime number after those used as bases in the radical inverse; i.e., this value is the (NFactors+1) prime number.

This property at the command line is `PrimeLeap`.

Skip zero point Causes the rst point of the sequence, which is always at the lower limit of each input factor, to be skipped. This point is often seen as unbalancing because the upper limits of each input factor can never be produced by the algorithm.

This property at the command line is SkipZero.

Apply RR2 Scramble Sets the scramble to 'RR2', which performs a permutation of the radical inverse coe icients using the RR2 algorithm.

This property at the command line is Scramble.

Latin Hypercube Sampling

Latin Hypercube Sampling (LHS) designs are sets of design points that, for an N point design, project onto N di erent levels in each factor. In this design, the points are generated randomly. You choose a particular Latin Hypercube by trying several such sets of randomly generated points and choosing the one that best satis es user-speci ed criteria.

Settings

For both Latin Hypercube Sampling and Stratified Latin Hypercube, you can choose the following options:

The Selection criteria drop-down menu has these options:

Maximize minimum distance (between points).

Minimize maximum distance (between points)

Minimize discrepancy Minimizes the deviation from the average point density

Minimize RMS variation from CDF This default option minimizes the Root Mean Square (RMS) variation of the Cumulative Distribution Function (CDF) from the ideal CDF.

Minimize maximum variation from CDF Minimizes the maximum variation of the CDF from the ideal CDF

The nal two (CDF variation) options scale best with the number of points and it is advisable to choose one of these options for large designs.

The Enforce Symmetrical Points check box is selected by default. This creates a design in which every design point has a mirror design point on the opposite side of the center of the design volume and an equal distance away. Restricting the design in this way tends to produce better Latin Hypercubes.

Lattice

Lattice designs project onto N di erent levels per factor for N points. The points are not randomly generated but are produced by an algorithm that uses a prime number per factor. If good prime numbers are chosen, the lattice spreads points evenly throughout the design volume. A poor choice of prime numbers results in highly visible lines or planes in the design projections. If all the design points are clustered into one or two planes, it is likely that you cannot estimate all the e ects in a more complex model. When design points are projected onto any axes, there are a large number of factor levels.

For a small number of trials (relative to the number of factors) LHS designs are preferred to Lattice designs. This is because of the way Lattice designs are generated. Lattice designs use prime numbers to generate each successive sampling for each factor in a di erent place. No two factors can have the same generator, because in such cases the lattice points all fall on the main diagonal of that particular pairwise projection, creating the visible lines or planes described above. When the number of points is small relative to the number of factors, the choice of generators is restricted and this can lead to Lattice designs with poor projection properties in some pairwise dimensions, in which the points lie on diagonals or double or triple diagonals. This means that Latin Hypercube designs are a better choice for these cases.

See the illustrations in the following section comparing the properties of good and poor lattices and a hypercube design.

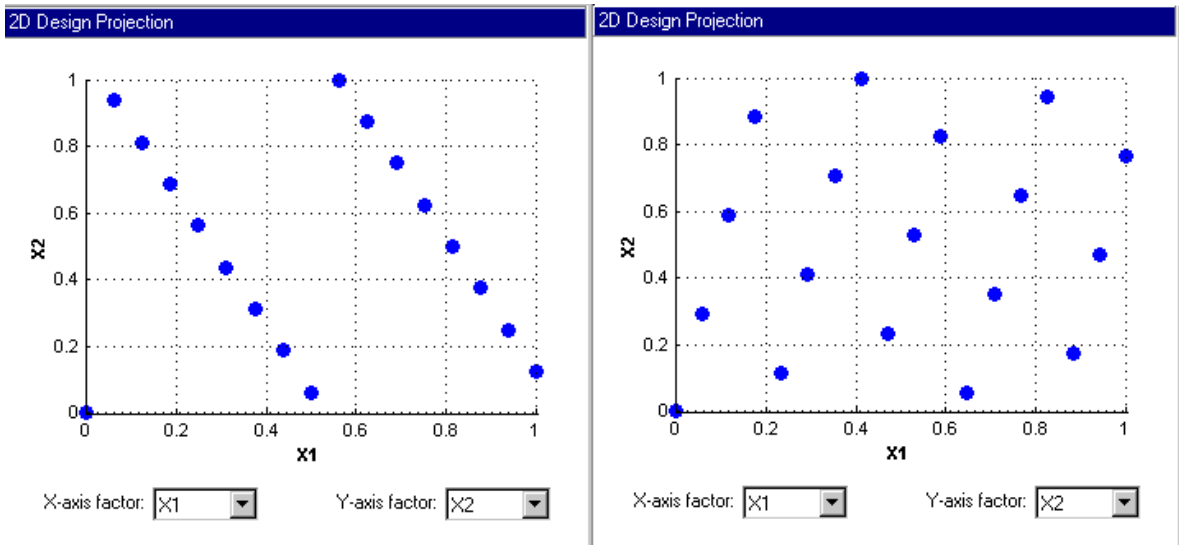
Settings

For a `Lattice` space- lling design, you can choose:

- The Lattice size by using the buttons or typing in the edit box.
- The prime number generator by using the up/down buttons on the Prime number for X edit box.
- The range for each factor.

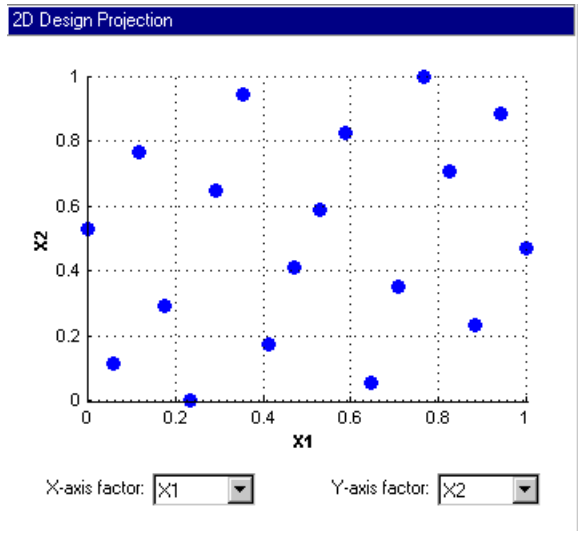
Strati ed Latin Hypercube

Strati ed Latin Hypercubes separate the normal hypercube into N di erent levels on user-speci ed factors. This can be useful for situations where the preferred number of levels for certain factors might be known; more detail might be required to model the behavior of some factors than others. They can also be useful when certain factors can only be run at given levels.



The preceding example shows the di erent properties of a poor lattice (left) and a good lattice (right), with a similar number of points. The poorly chosen prime number produces highly visible planes and does not cover the space well.

An example of an LHS design of the same size is shown for comparison with the preceding lattice examples.



Settings

See Latin Hypercube Sampling on page 3-22, the options are the same.

Comparing Latin Hypercube and Strati ed Latin Hypercube

Latin Hypercube Sampling and Strati ed Latin Hypercube Sampling di er only in that with Strati ed Latin Hypercube Sampling, you can restrict the number of levels available to each factor. If the number of strati cations equals the number of points in the design, then both Latin Hypercube Sampling and Strati ed Latin Hypercube Sampling give the same results. However, if the number of strati cations in a given factor is less than the number of points in the design, then some points will be projected onto the same values in that factor. You can see this change by using the one-dimensional design projection view in the Design Editor.

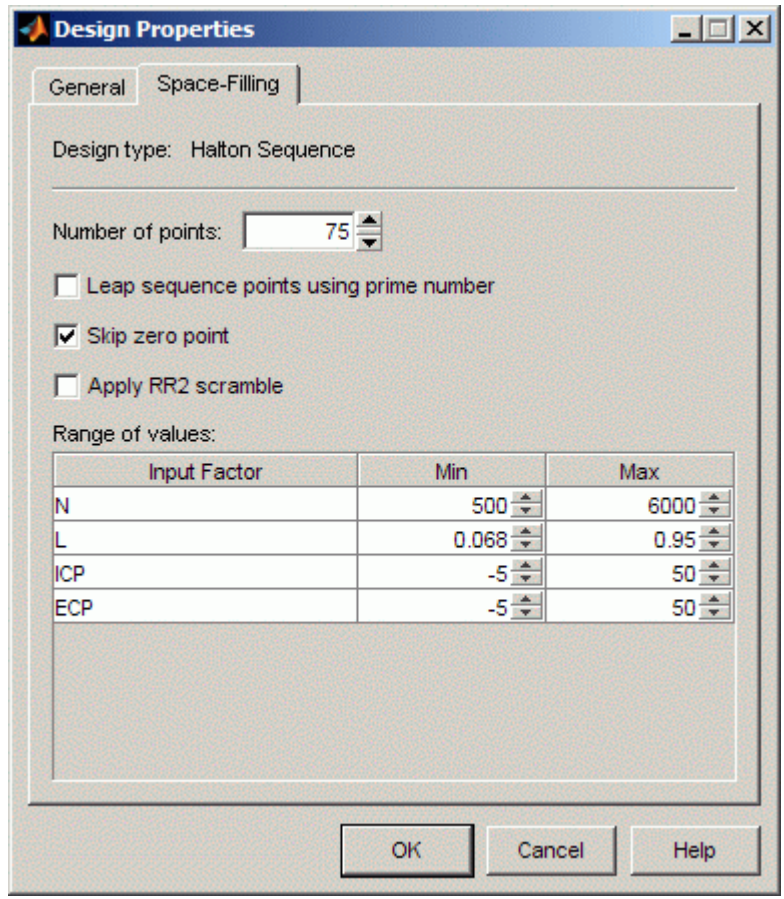
Augmenting Space-Filling Designs

To add points to space- lling designs, you can either use Edit > Add Point, or you can edit the le properties. However the Design Properties dialog box does not contain a preview plot, so instead, use the Add Points dialog box to add to your space- lling design. See Adding and Editing Design Points on page 3-44.

Alternatively, to use the Design Properties dialog box to augment a space- lling design:

1 Create a copy of your design to preserve it, then in the child design, select File > Properties.

2 In the Design Properties dialog box, select the Space-Filling tab.



3 Enter the desired new total number of points in the Number of points edit box.

Leave the other settings unchanged.

4 Click OK.

The Design Editor augments your original design by adding points up to the new total number of points, with the same space- lling sequence parameters as your original Halton or Sobol sequence.

To use a preview plot, see Adding and Editing Design Points on page 3-44 instead.

Space-Filling Design Augmentation Restrictions

You can only augment Halton and Sobol sequence space-filling designs with this method that uses the original sequence settings.

You can augment any Halton sequence, but for Sobol sequences, you must use the default **No skip** setting.

You cannot achieve the same result by selecting any **Design** > **Space Filling** menu option and selecting the **Augment points** option, because doing so will keep your existing points, but will generate the additional points with a new space-filling sequence.

You cannot augment a constrained design in this way. When you add constraints, the design type changes to `Custom`. You can no longer access the original sequence settings. If you want to add constraints, you must create a child design and constrain it. This approach preserves the original space-filling sequence design.

You cannot use this method with Latin Hypercube or Lattice designs as they always create a completely new design.

Create an Optimal Design

Introducing Optimal Designs

Optimal designs are best for cases with high system knowledge, where previous studies have given con dence on the best type of model to be tted, and the constraints of the system are well understood. Optimal designs require linear models.

1 Click the button in the toolbar or select File > New Design. A new node appears in the design tree. It is named according to the model for which you are designing, for example, `Linear Model Design`.

2 Select the node in the tree by clicking. An empty Design Table appears if you have not yet chosen a design. Otherwise, if this is a new child node the display remains the same, because child nodes inherit all the parent design's properties.

3 Set up any constraints at this point. See De ne Design Constraints on page 3-10.

4 Choose an Optimal design by clicking the button in the toolbar, or choose Design > Optimal.

The optimal designs in the Design Editor are formed using the following process:

An initial starting design is chosen at random from a set of de ned candidate points.

p additional points are added to the design, either optimally or at random. These points are chosen from the candidate set.

p points are deleted from the design, either optimally or at random.

If the resulting design is better than the original, it is kept.

This process is repeated until either (a) the maximum number of iterations is exceeded or (b) a certain number of iterations has occurred without an appreciable change in the optimality value for the design.

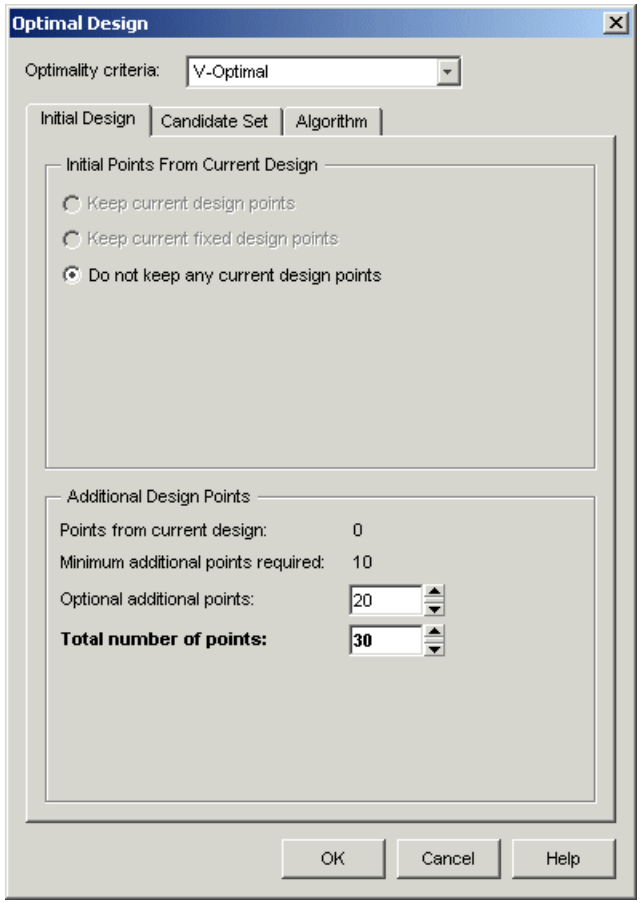
The Optimal Design dialog box consists of three tabs that contain the settings for three main aspects of the design:

Initial Design tab: Starting point and number of points in the design

Candidate Set on page 3-31 tab: Candidate set of points from which the design points are chosen

Algorithm on page 3-34 tab: Options for the algorithm that is used to generate the points

Optimal Design: Initial Design Tab



The Initial Design tab allows you to de ne the composition of the initial design: how many points to keep from the current design and how many total or additional points to choose from the candidate set.

1 Choose the type of the optimal design, using the Optimality criteria drop-down menu:

D-Optimal designs Aims to reduce the volume of the con dence ellipsoid to obtain accurate coe icients. This is set up as a maximization problem, so the progress graph should go up with time.

The D-optimality value used is calculated using the formula

$$D_{eff} = \frac{\log_e(\det(X'X))}{k}$$ where X is the regression matrix and k is the number of terms in the regression matrix.

V-Optimal designs Minimizes the average prediction error variance, to obtain accurate predictions. This is better for calibration modeling problems. This is a minimization process, so the progress graph should go down with time.

The V-optimality value is calculated using the formula

$$V_{eff} = \frac{1}{n_C}\sum_j x_j'(X_C'X_C)^{-1}x_j$$

where x_j are rows in the regression matrix, X_C is the regression matrix for all candidate set points, and n_C is the number of candidate set points.

A-Optimal designs Minimizes the average variance of the parameters and reduces the asphericity of the con dence ellipsoid. The progress graph also goes down with this style of optimal design.

The A-optimality value is calculated using the formula

$$A_{eff} = \text{trace}((X'X)^{-1})$$

where X is the regression matrix.

2 You might already have points in the design (if the new design node is a child node, it inherits all the properties of the parent design). If so, choose from the radio buttons:

Replace the current points with a new initial design

Augment the current design with additional points

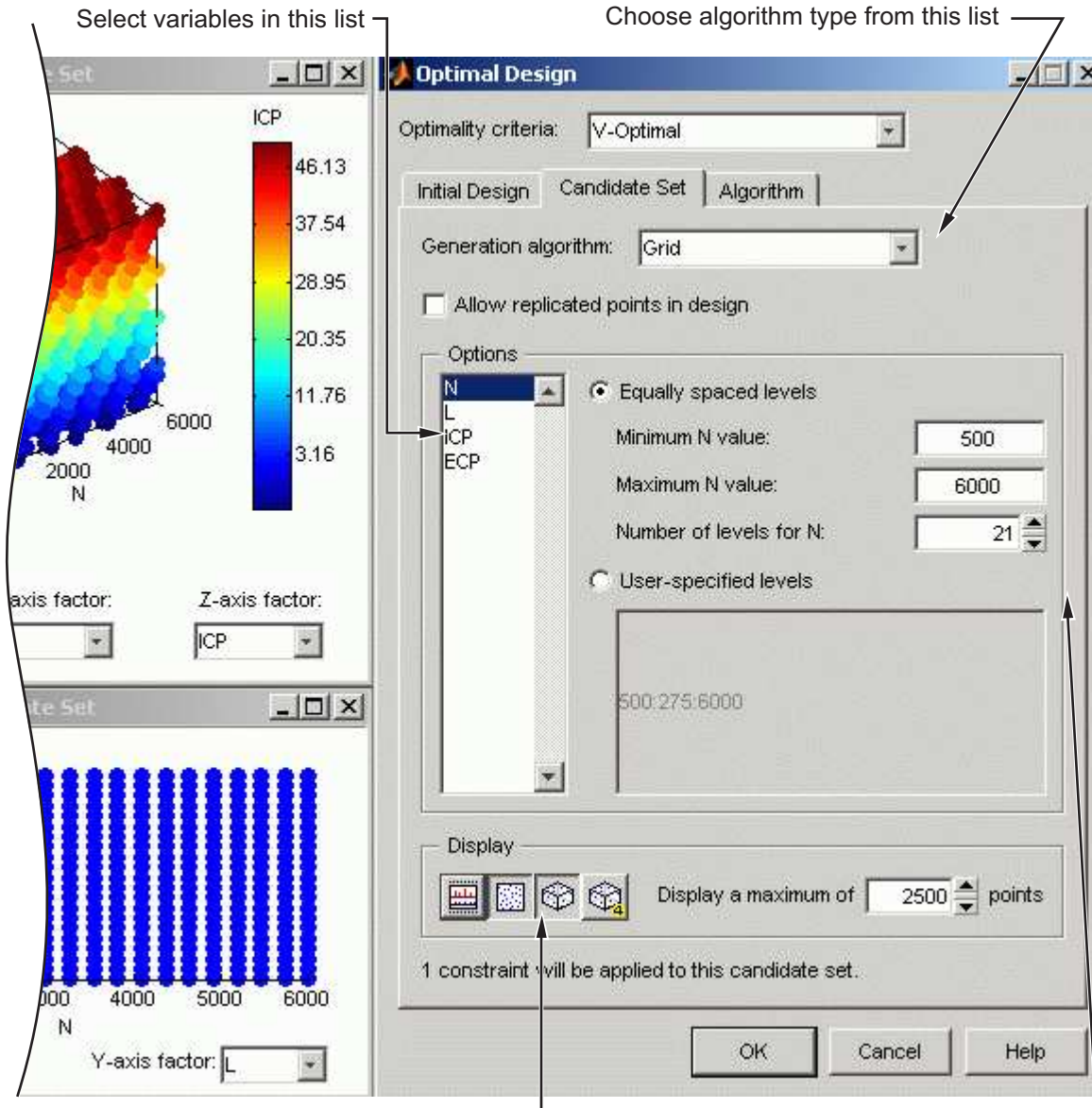
Keep only the xed points from the current design

For information on xed design points, see Fixing, Deleting, and Sorting Design Points on page 3-48.

3 You can choose the total number of points and/or the number of additional points to add by clicking the up/down buttons or by typing directly into the edit boxes for Optional additional points or Total number of points.

Optimal Design: Candidate Set Tab

The Candidate Set tab allows you to set up a candidate set of points for your optimal design. Candidate sets are a set of potential test points. This typically ranges from a few hundred points to several hundred thousand.


Select variables in this list
Choose algorithm type from this list
Optimal Design
Optimality criteria:
V-Optimal
Initial Design
Candidate Set
Algorithm
Generation algorithm:
Grid
Allow replicated points in design
Options
N
L
ICP
ECP
Equally spaced levels
Minimum N value:
500
Maximum N value:
6000
Number of levels for N:
21
User-specified levels
500:275:6000
Display
Display a maximum of
2500
points
1 constraint will be applied to this candidate set.
OK
Cancel
Help
ICP
46.13
37.54
28.95
20.35
11.76
3.16
6000
4000
2000
N
axis factor:
Z-axis factor:
ICP
4000
5000
6000
N
Y-axis factor:
L
Open display windows with these buttons
Change the number of levels of the selected variable here

The set generation schemes are as follows:

`Grid` Full factorial grids of points, with fully customizable levels.

`Grid/Lattice` A hybrid set where the main factors are used to generate a lattice, which is then replicated over a small number of levels for the remaining factors.

`Halton Sequence` Halton Sequence designs are generated from the `haltonset` class in the Statistics and Machine Learning Toolbox software. See Halton Sequence on page 3-21 for more information

`Lattice` These have the same de nition as the space- lling design lattices, but are typically used with about 10,000 points. The advantage of a lattice is that the number of points does not increase as the number of factors increases; however, you do have to try di erent prime number generators to achieve a good lattice. See Lattice on page 3-23.

`Sobol Sequence` Sobol sequence designs are generated from the `sobolset` class in the Statistics and Machine Learning Toolbox software. See Sobol Sequence on page 3-20 for more information.

`Stratified Lattice` Another method of using a lattice when some factors cannot be set to arbitrary values. Strati ed lattices ensure that the required number of levels is present for the chosen factor. Note that you cannot set more than one factor to stratify to the same N levels. This is because forcing the same number of levels would also force the factors to have the same generator. As for a lattice space- lling design, no two factors can have the same generator, because in such cases the lattice points all fall on the main diagonal of that particular pairwise projection, creating highly visible planes in the points and poor coverage of the space. For illustrations of this e ect, see Lattice on page 3-23.

`User-defined` Import custom matrices of points from MATLAB software or MAT- les.

For each factor you can de ne the range and number of di erent levels within that range to select points.

1 Choose a type of generation algorithm from the drop-down menu. Note that you could choose di erent parameters for di erent factors (within an overall scheme such as `Grid`).

2 This tab also has buttons for creating plots of the candidate sets. Try them to preview your candidate set settings. If you have created a custom candidate set you can check it here. The edit box sets the maximum number of points that will be plotted in the preview windows. Candidate sets with many factors can quickly become very large, and attempting to display the entire set will take too long. If the candidate set has

more points than you set as a maximum, only every Nth point is displayed, where N is chosen such that (a) the total displayed is less than the maximum and (b) N is prime. If you think that the candidate set preview is not displaying an adequate representation of your settings, try increasing the maximum number of points displayed.

3 Notice that you can see 1D, 2D, 3D, and 4D displays (fourth factor is color) all at the same time as they appear in separate windows (see the example following). Move the display windows (click and drag the title bars) so you can see them while changing the number of levels for the di erent factors.

4 You can change the factor ranges and the number of levels using the edit boxes or buttons.

Optimal Design: Algorithm Tab

The Algorithm tab has the following algorithm details:

Augmentation method Random or Optimal Optimal can be very slow (searches the entire candidate set for points) but converges using fewer iterations. Random is much faster per iteration, but requires a larger number of iterations. The Random setting does also have the ability to lower the optimal criteria further when the Optimal setting has found a local minimum.

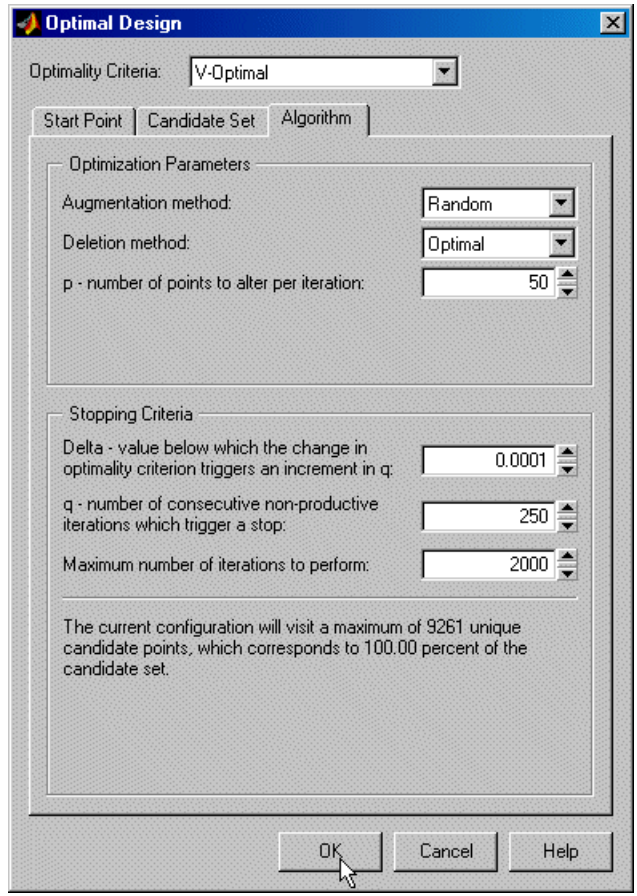
Deletion method Random or Optimal Optimal deletion is much faster than augmentation, because only the design points are searched.

p number of points to alter per iteration The number of points added/removed per iteration. For optimal augmentation this is best kept smaller (~5); for optimal deletion only it is best to set it larger.

Delta value below which the change in optimality criteria triggers an increment in q This is the size of change below which changes in the optimality criteria are considered to be not signi cant.

q number of consecutive non-productive iterations which trigger a stop Number of consecutive iterations to allow that do not increase the optimality of the design. This only has an e ect if random augmentation or deletion is chosen.

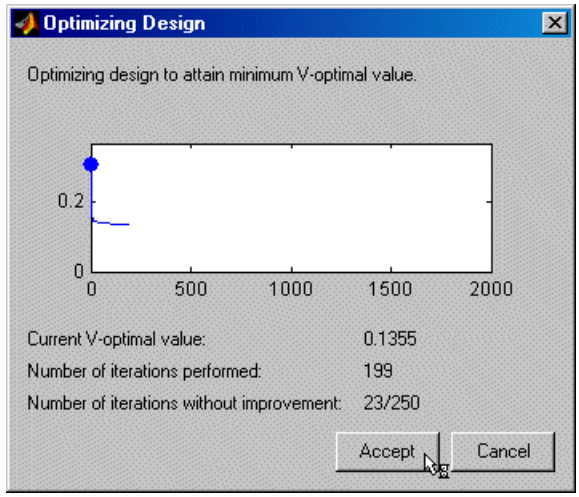
Maximum number of iterations to perform Overall maximum number of iterations.



1 Choose the augmentation and deletion methods from the drop-down menus (or leave at the defaults).
2 You can alter the other parameters by using the buttons or typing directly in the edit boxes.
3 Click OK to start optimizing the design.

When you click the OK button on the Optimal Design dialog box, another window appears that contains a graph. This window shows the progress of the optimization and has two buttons: Accept and Cancel. Accept stops the optimization early and takes the current design from it. Cancel stops the optimization and reverts to the original design.

4 You can click Accept at any time, but it is most useful to wait until iterations are not producing noticeable improvements; that is, the graph becomes very at.



You can always return to the Optimal Design dialog box (following the same steps) and choose to keep the current points while adding more.

Averaging Optimality Across Multiple Models

The Design Editor can average optimality across several linear models. This is a exible way to design experiments using optimal designs. If you have no idea what model you are going to t, you would choose a space- lling design. However, if you have some idea what to expect, but are not sure which model to use, you can specify a number of possible models. The Design Editor can average an optimal design across each model.

For example, if you expect a quadratic and cubic for three factors but are unsure about a third, you can specify several alternative polynomials. You can change the weighting of each model as you want (for example, 0.5 each for two models you think equally likely). This weighting is then taken into account in the optimization process in the Design Editor. See Global Model Class: Multiple Linear Models on page 5-76.

Create a Classical Design

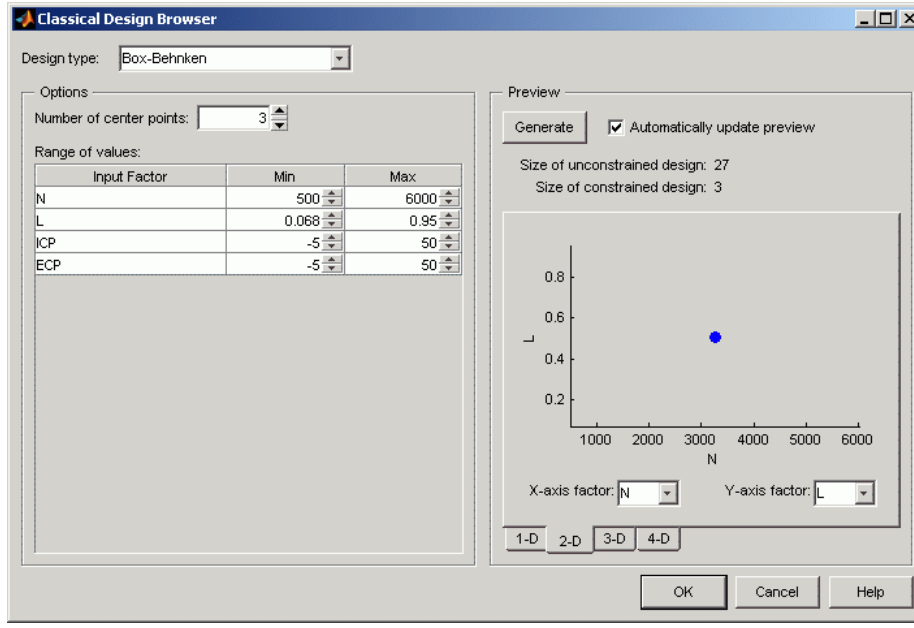
1. Add a new design by clicking the button in the toolbar or select File > New.
2. Select the new design node in the tree. An empty Design Table appears if you have not yet chosen a design. Otherwise if this is a new child node the display remains the same, because child nodes inherit all the parent design's properties. All the points from the previous design remain, to be deleted or added to as necessary. The new design inherits all its initial settings from the currently selected design and becomes a child node of that design.
3. Click the button in the toolbar or select Design > Classical > Design Browser.

Note In cases where the preferred type of classical design is known, you can go straight to one of the ve options under Design > Classical. Choosing the Design Browser option allows you to see graphical previews of these same ve options before making a choice.

---

4. A dialog box appears if there are already points from the previous design. You must choose between replacing and adding to those points or keeping only xed points from the design. The default is replacement of the current points with a new design. Click OK to proceed, or Cancel to change your mind.

The Classical Design Browser appears.



In the Design Style drop-down menu there are five classical design options:

`Central Composite`

Generates a design that has a center point, a point at each of the design volume corners, and a point at the center of each of the design volume faces. The options are `Face-center cube`, `Spherical`, `Rotatable`, or `Custom`. If you choose `Custom`, you can then choose a ratio value (α) between the corner points and the face points for each factor and the number of center points to add. Five levels for each factor are used. You can set the ranges for each factor. `Inscribe star points` scales all points within the coded values of 1 and -1 (instead of plus or minus α outside that range). When this box is not selected, the points are circumscribed.

A Central-Composite design consists of factorial points in the corners of the space, plus axial points in the direction of the design face centers that can have their distance varied by the alpha factor. For values of alpha, see Classical Design Properties on the `Properties (for design generators)` reference page.

`Spherical` arranges the axial design points so that both they and the factorial points lie on the same geometric circle/sphere/hypersphere.

Rotatable means that the prediction variance pattern of the design is spherically-symmetric, that is, rotating the design in any direction has no impact on the prediction quality of a model that results from the experiment.

With 2 factors the rotatable design is also circular, but in higher dimensions the rotatable designs have closer axial points than the spherical designs.

Box-Behnken

Similar to Central Composite designs, but only three levels per factor are required, and the design is always spherical in shape. All the design points (except the center point) lie on the same sphere, so you should choose at least three to five runs at the center point. There are no face points. These designs are particularly suited to spherical regions, when prediction at the corners is not required. You can set the ranges of each factor.

Full Factorial

Generates an n-dimensional grid of points. You can choose the number of levels for each factor, the number of additional center points to add, and the ranges for each factor.

Plackett Burman

These are screening designs. They are two-level designs that are designed to allow you to work out which factors are contributing any effect to the model while using the minimum number of runs. For example, for a 30-factor problem this can be done with 32 runs. They are constructed from Hadamard matrices and are a class of two-level orthogonal array.

Regular Simplex

These designs are generated by taking the vertices of a k-dimensional regular simplex (k = number of factors). For two factors a simplex is a triangle; for three it is a tetrahedron. Above that are hyperdimensional simplices. These are economical first-order designs that are a possible alternative to Plackett Burman or full factorials.

Setting Up and Viewing a Classical Design

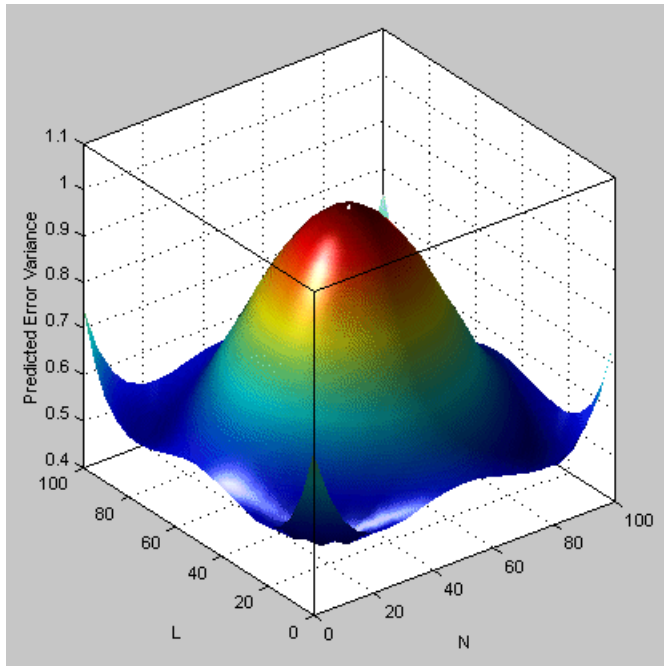
1 Choose a Box-Behnken design.

2 Reduce the number of center points to 1.

3 View your design in different projections using the tabs under the display.

4 Click OK to return to the Design Editor.

5 Use the Prediction Error Variance Viewer to see how well this design performs compared to the optimal design created previously; see the following illustration.

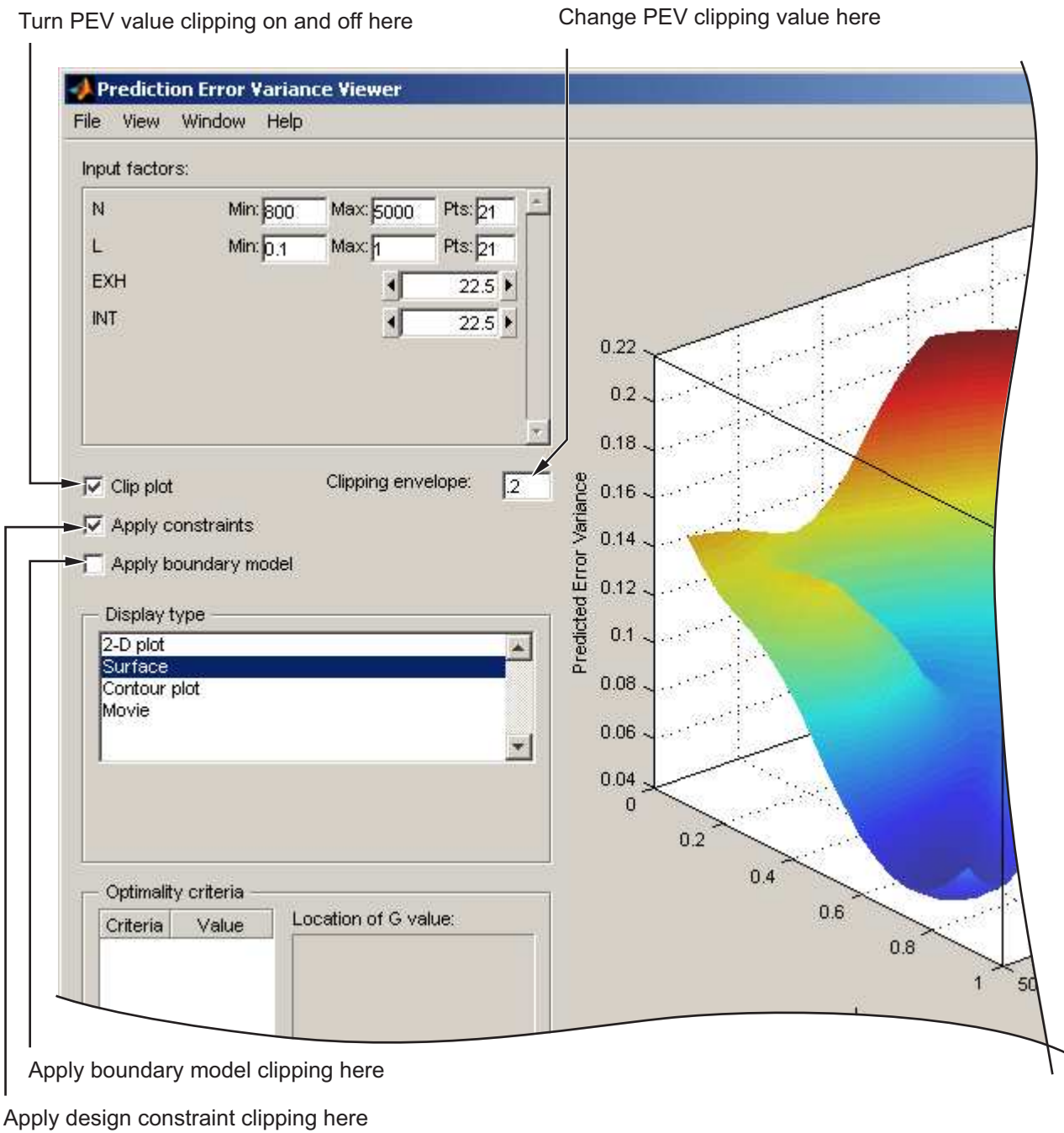


As you can see, this is not a realistic comparison, as this design has only 13 points (you can find this information in the bottom left of the main Design Editor display), whereas the previous optimal design had 100, but this is a good illustration of leverage. A single point in the center is very bad for the design, as illustrated in the Prediction Error Variance Viewer surface plot. This point is crucial and needs far more certainty for there to be any confidence in the design, as every other point lies on the edge of the space. This is also the case for Central Composite designs if you choose the spherical option. These are good designs for cases where you are not able to collect data points in the corners of the operating space.

If you look at the PEV surface plot, you should see a spot of white at the center. This is where the predicted error variance reaches 1. For surfaces that go above 1, the contour at 1 shows as a white line, as a useful visual guide to areas where prediction error is large.

1 Select Movie, and you see this white contour line as the surface moves through the plane of value 1.

2 Select the Clip Plot check box. Areas that move above the value of 1 are removed. The following example explains the controls.


Turn PEV value clipping on and off here
Change PEV clipping value here
Prediction Error Variance Viewer
File View Window Help
Input factors:
N Min: 800 Max: 5000 Pts: 21
L Min: 0.1 Max: 1 Pts: 21
EXH 22.5
INT 22.5
Clip plot
Clipping envelope: .2
Apply constraints
Apply boundary model
Display type
2-D plot
Surface
Contour plot
Movie
Optimality criteria
Criteria Value
Location of G value:
Predicted Error Variance
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0
0.2
0.4
0.6
0.8
1
Apply boundary model clipping here
Apply design constraint clipping here

See Also

Related Examples

Create a Constrained Space-Filling Design

View Design Displays

More About

Design of Experiments

Manipulate Designs

Adding and Editing Design Points

Adding Design Points

In any design, you can add points by selecting Edit > Add Point. You can choose how to add points: extend a space- lling sequence (for space- lling designs), optimally (for optimal designs), randomly, or at speci ed values. For space- lling designs, when you want to collect more data, you can add design points that continue the same space- lling sequence in your original design. This allows you to collect more data lling in the gaps between your previous design points. You can progressively augment Halton and Sobol sequence space- lling designs to add points with the same sequence parameters. You can add points to your original space- lling sequence, preserving the original points and adding new ones with the same sequence parameters.

Tip In case you want to add more points later, to preserve the space- lling sequence of a design, copy the design before constraining or editing the design. If you do not create a copy of the design, when you edit, you lose the original space- lling settings and then you cannot extend the sequence. When design type changes to `Custom`, you cannot access the original sequence settings to add new points.

---

1 To preserve your original design when you add new points, create a child design of your original unconstrained design. Select your design, and click the New Design button in the toolbar, or select File > New.

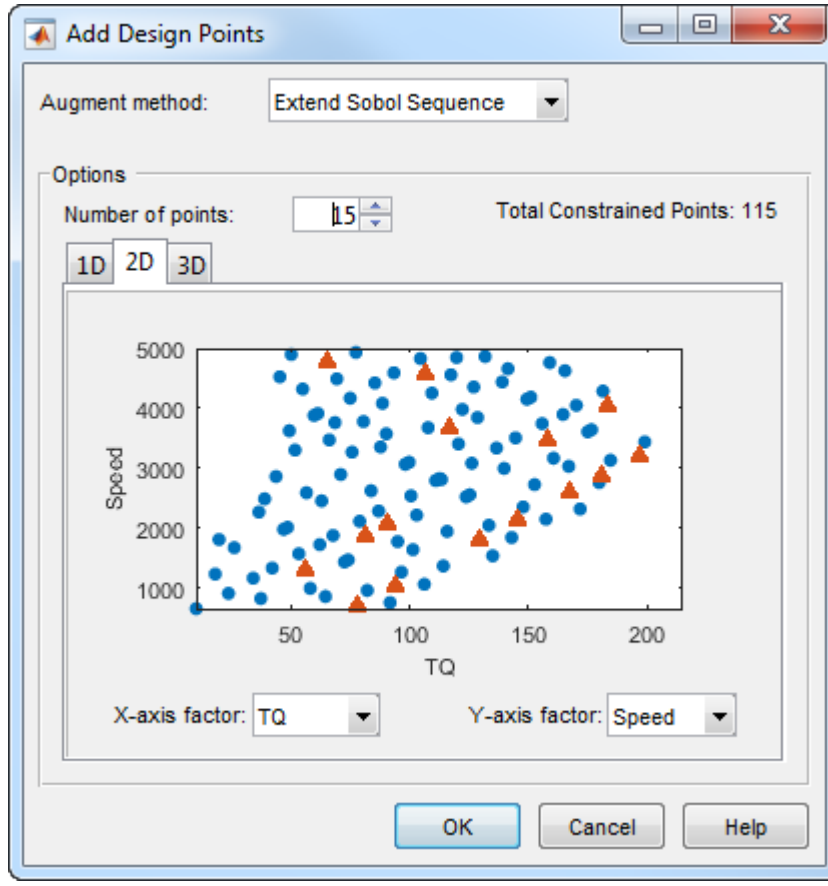
   The new child design, identical to the parent, is selected in the tree.
2 To add space- lling, optimal, custom, or random points, select Edit > Add Point or click the button. A dialog box appears, as shown following.
3 Choose an augmentation method from the drop-down menu. Options depend on your design type and include: extend sobol or halton sequence, optimal (D,V, or A), random, or user-speci ed.

   Note You can add points optimally to any design based on a linear or multilinear model, as long as it has the minimum number of points required to t that model.

This means that after adding a constraint you might remove so many points that a subsequent replace operation does not allow optimal addition.

For space- lling designs, use `Extend Sobol Sequence` or `Extend Halton Sequence` to preserve the original points and add new ones with the same sequence parameters.

4 Choose the number of points to add, using the buttons or typing into the edit box. For space- lling designs, observe the new points on the plot.



For user-speci ed custom points, enter the values of each factor for each point you want to add.

5 If you choose an optimal augmentation method and click Edit, the Candidate Set dialog box appears. Edit the ranges and levels of each factor and which generation algorithm to use. These are the same controls you see on the Candidate Set on page 3-31 tab of the Optimal Design dialog box. Click OK.
6 Click OK to add the new points and close the Add Design Points dialog box.

In the Design Editor, for space- lling designs, these steps can be helpful:

Use the pairwise view, switching between the parent and child designs to visually verify that the new points have been added while preserving your original points.

Create a copy child design to add constraints. Preserving your unconstrained parent design allows you to add more points later if you need to. See also Space-Filling Design Augmentation Restrictions on page 3-27.

New points are added to the end of the list of existing points. If you want to extract only the augmented points for testing, select Edit > Delete Point to open a dialog box in which you can choose the points to delete. See Fixing, Deleting, and Sorting Design Points on page 3-48.

Round and sort your data before sending it out for testing. Select Edit > Round Factor to limit decimal places of factors. Select Edit > Sort to sort the points for test e iciency, because operators often test in order of speed followed by load. See Fixing, Deleting, and Sorting Design Points on page 3-48.

Editing Design Points

Tip In case you want to add more points later, to preserve the space- lling sequence of a design, copy the design before editing points. If you do not create a copy of the design, when you edit points, you lose the original space- lling settings and then you cannot extend the sequence. When design type changes to Custom, you cannot access the original sequence settings to add new points.

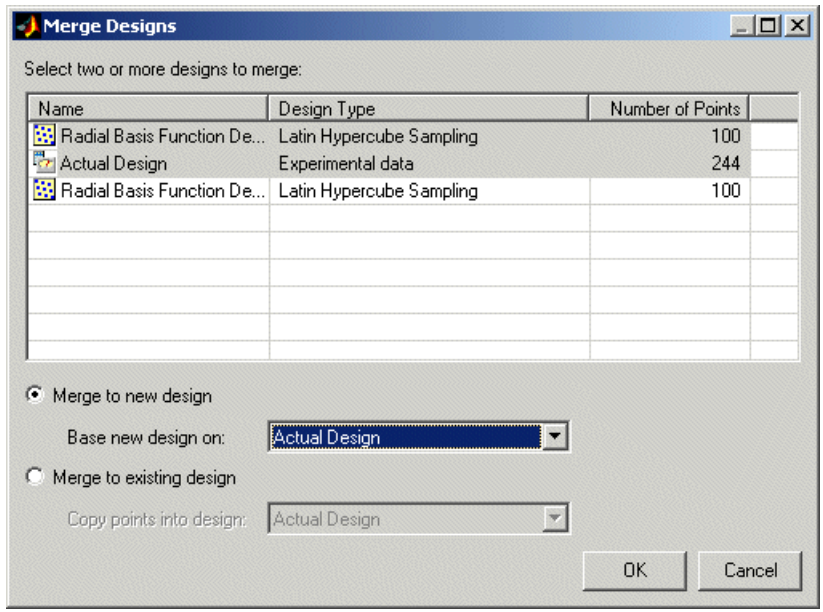
To edit particular points,

1 Right-click the title bar of one of the Design Editor views and select Current View > Design Table to change to the Table view. This gives a numbered list of every point in the design, so you can see where points are in the design.
2 To edit points, click to select table cells and type new values. You can also right-click table cells and select Copy or Paste. You can click and drag to select multiple cells to copy or paste.

Merging Designs

You can merge the points from two or more designs together using the File menu. You can merge designs together to form a new design, or merge points into one of the chosen designs. Points that are merged retain their fixed status in the new design.

1 Select File > Merge Designs. A dialog box appears, as shown.



2 A list of all the designs from the Design Editor is shown, along with the associated design style and number of points. Select two or more designs from the list by dragging with the mouse or Ctrl+clicking.

3 When at least two designs have been selected, the options at the bottom of the dialog box are enabled. Choose whether you want to create a new design or put the design points into an existing design.

4 If you choose to create a new design, you must also choose one of the selected designs to act as a base. Properties such as the model, constraints, and any optimal design setup are copied from this base design. If you choose to reuse an existing design you must choose one of the selected designs to receive the points from other designs.

5 Click OK to perform the merging process and return to the main display. If you choose to create a new design, it appears at the end of the design tree.

Fixing, Deleting, and Sorting Design Points

You can fix or delete points using the Edit menu. You can also sort points or clear all points from the design.

Tip In case you want to add more points later, to preserve the space-filling sequence of a design, copy the design before editing, rounding, or sorting points. If you do not create a copy of the design, when you edit points, you lose the original space-filling settings and then you cannot extend the sequence. When design type changes to Custom, you cannot access the original sequence settings to add new points.

Fixed points become red in the main Design Editor table display. If you have matched data to a design or used experimental data as design points, those points are automatically fixed. You already have the data points, so you do not want them to be changed or deleted. Design points that have been matched to collected data are also fixed. Since these points have already been run they cannot be freed you will not see them in the Fix Design Points dialog box. Once you have fixed points, they are not moved by design optimization processes. This automatic fixing makes it easy for you to optimally augment the fixed design points.

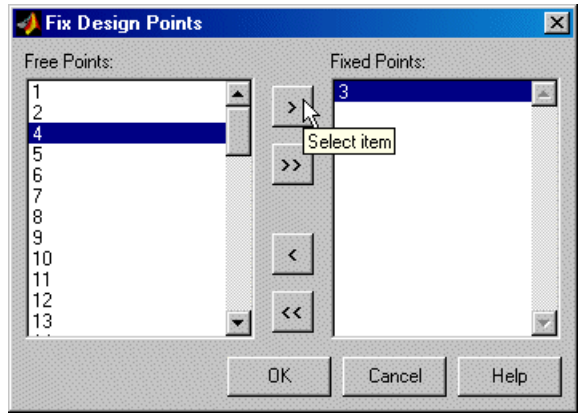
To fix or delete points:

1 To delete all points in the current design, select Edit > Clear.

2 If you want to fix or delete particular points, first change a Design Editor display pane to the Table view. This gives a numbered list of every point in the design, so you can see where points are in the design.

3 Select Edit > Fix/Free Points or Edit > Delete Point (or click the toolbar button).

4 A dialog box appears in which you can choose the points to fix or delete.



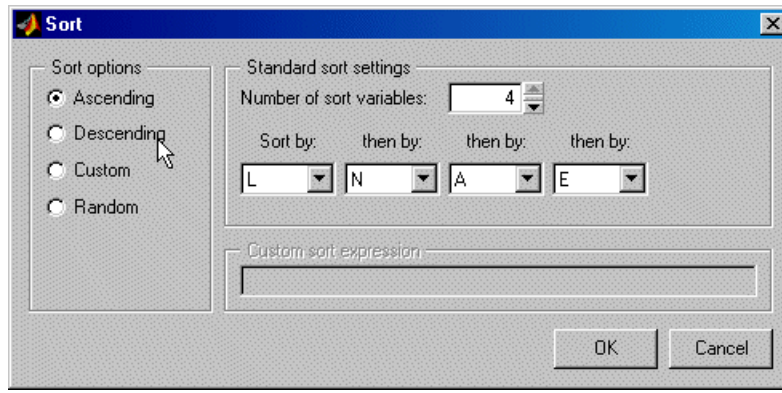
The example above shows the dialog box for fixing or freeing points. The dialog box for deleting points has the same controls for moving points between the Keep Points list and the Delete Points list.

5 Move points from the Free Points list to the Fixed Points list; or from the Keep Points list to the Delete Points list, by using the buttons.

6 Click OK to complete the changes specified in the list boxes, or click Cancel to return to the unchanged design.

To sort points:

1 Select Edit > Sort. A dialog box appears (see example following) for sorting the current design by ascending or descending factor values, randomly, or by a custom expression.



2 To sort by custom expression you can use MATLAB expressions (such as `abs(N)` for the absolute value of `N`) using the input symbols. Note that sorts are made using coded units (from -1 to 1) so remember that points in the center of your design space will be near zero in coded units, and those near the edge will tend to 1 or -1.

3 Select Edit > Randomize as a quick way of randomly resorting the points in the current design. This is a shortcut to the same functionality provided by the Random option in the Sort dialog box.

Prediction Error Variance Viewer

Introducing the Prediction Error Variance Viewer on page 3-50

Display Options on page 3-53

Prediction Error Variance on page 3-54

Prediction Error Variance for Two-Stage Models on page 3-56

Introducing the Prediction Error Variance Viewer

You can use the Prediction Error Variance (PEV) viewer to examine the quality of the model predictions. You can examine the properties of designs or global models. When you open it from the Design Editor, you can see how well the underlying model predicts over the design region. When you open it from a global model, you can view how well the current global model predicts. A low PEV (tending to zero) means that good predictions are obtained at that point.

The Prediction Error Variance Viewer is only available for linear models and radial basis functions.

When designs are rank de cient, the Prediction Error Variance Viewer appears but is empty; that is, the PEV values cannot be evaluated because there are not enough points to t the model.

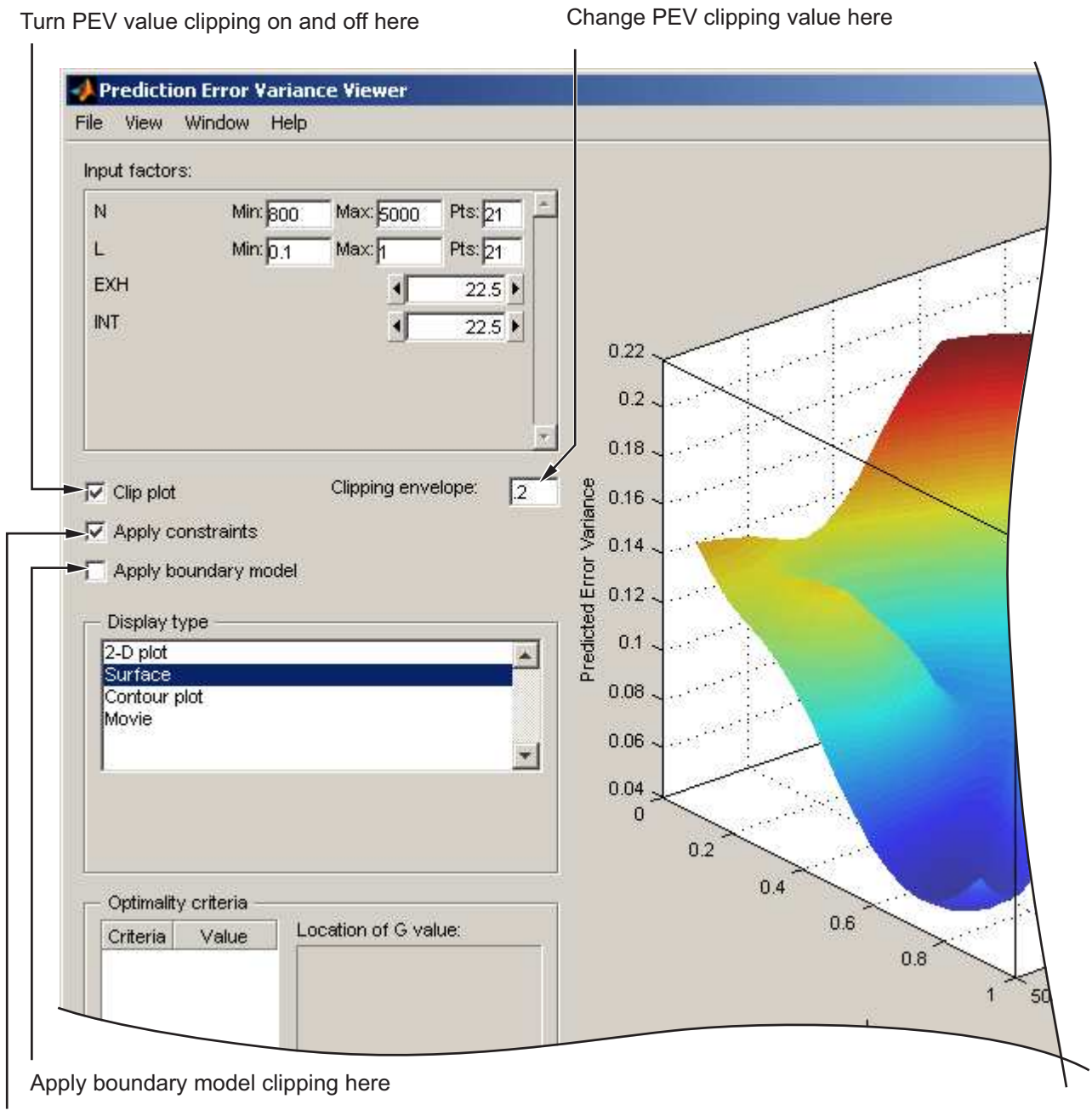
From the Design Editor, select Tools > Prediction Error Variance Viewer.

From the global level of the Model Browser, if the selected global model is linear or a radial basis function,

Click the toolbar button to open the Prediction Error Variance Viewer.

Alternatively, select Model > Utilities > Prediction Error Variance Viewer.

If a model has child nodes you can only select the Prediction Error Variance Viewer from the child models.


Turn PEV value clipping on and off here
Change PEV clipping value here
Prediction Error Variance Viewer
File View Window Help
Input factors:
N Min: 300 Max: 5000 Pts: 21
L Min: 0.1 Max: 1 Pts: 21
EXH 22.5
INT 22.5
Clip plot
Clipping envelope: .2
Apply constraints
Apply boundary model
Display type
2-D plot
Surface
Contour plot
Movie
Optimality criteria
Criteria Value
Location of G value:
Predicted Error Variance
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0
0.2
0.4
0.6
0.8
1
Apply boundary model clipping here
Apply design constraint clipping here

The default view is a 3D plot of the PEV surface.

The plot shows where the model predictions are best. The model predicts well where the PEV values are lowest.

If you have transformed the output data (e.g. using a Box-Cox transform), then the Prediction Error Variance Viewer displays the predicted variance of the transformed model.

Display Options

The View menu has many options to change the look of the plots.

You can change the factors displayed in the 2D and 3D plots. The drop-down menus below the plot select the factors, while the unselected factors are held constant. You can change the values of the unselected factors using the buttons or edit boxes in the frame, top left.

The Movie option shows a sequence of surface plots as a third input factor's value is changed. You can change the factors, replay, and change the frame rate.

You can change the number, position, and color of the contours on the contour plot with the Contours button. See the contour plot section (in Response Surface View on page 6-41) for a description of the controls.

You can select the Clip Plot check box, as shown in the preceding example. Areas that move above the PEV value in the Clipping envelope edit box are removed. You can enter the value for the clipping envelope. If you do not select Clip Plot, a white contour line is shown on the plot where the PEV values pass through the clipping value.

You can also clip with the boundary model or design constraints if available. Select the check boxes Apply constraint or Apply boundary model to clip the plot.

When you use the Prediction Error Variance Viewer to see design properties, optimality values for the design appear in the Optimality criteria frame.

Note that you can choose Prediction Error shading in the Response Feature view (in Model Selection or Model Evaluation). This shades the model surface according to Prediction Error values (sqrt(PEV)). This is not the same as the Prediction Error Variance Viewer, which shows the shape of a surface de ned by the PEV values. See Response Surface View on page 6-41.

Optimality Criteria

No optimality values appear in the Optimality criteria frame until you click Calculate. Clicking Calculate opens the Optimality Calculations dialog box. Here iterations of the optimization process are displayed.

In the Optimality criteria frame in the Prediction Error Variance Viewer are listed the D, V, G and A optimality criteria values, and the values of the input factors at the point of maximum PEV (Location of G value). This is the point where the model has its maximum PEV value, which is the G-optimality criteria. The D, V and A values are functions of the entire design space and do not have a corresponding point.

For statistical information about how PEV is calculated, see the next section Prediction Error Variance on page 3-54.

Prediction Error Variance

Prediction Error Variance (PEV) is a very useful way to investigate the predictive capability of your model. It gives a measure of the precision of a model's predictions.

You can examine PEV for designs and for models. It is useful to remember that:

PEV (model) = PEV (design) * MSE

So the accuracy of your model's predictions is dependent on the design PEV and the mean square errors in the data. You should try to make PEV for your design as low as possible, as it is multiplied by the error on your model to give the overall PEV for your model. A low PEV (close to zero) means that good predictions are obtained at that point.

You can think of the design PEV as multiplying the errors in the data. If the design PEV < 1, then the errors are reduced by the model tting process. If design PEV >1, then any errors in the data measurements are multiplied. Overall the predictive power of the model will be more accurate if PEV is closer to zero.

You start with the regression (or design) matrix, for example, for a quadratic in N (engine speed) and L (load or relative air charge):

$$X = \begin{bmatrix} 1 & L_1 & N_1 & L^2_1 & L_1N_1 & N^2_1 \\ 1 & L_2 & N_2 & L^2_2 & L_2N_2 & N^2_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & L_n & N_n & L^2_n & L_nN_n & N^2_n \end{bmatrix}$$

If you knew the actual model, you would know the actual model coe icients β. In this case the observations would be:

$$y = X\beta + \varepsilon$$

where ε is the measurement error with variance

$$\mathrm{var}(\varepsilon) = \mathrm{MSE}$$

However you can only ever know the predicted coe icients:

$$\hat{\beta} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$$

which have variance

$$\mathrm{var}\hat{\beta} = (X^{\mathrm{T}}X)^{-1}\mathrm{MSE}$$

Let x be the regression matrix for some new point where you want to evaluate the model, for example:

$$x = \begin{bmatrix} 1 & L_{new} & N_{new} & L^2_{new} & L_{new}N_{new} & N^2_{new} \end{bmatrix}$$

Then the model prediction for this point is:

$$\hat{y} = x\hat{\beta} = x(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$$

Now you can calculate PEV as follows:

$$\mathrm{PEV}(x) = \mathrm{var}(\hat{y}) = (x(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}})(X(X^{\mathrm{T}}X)^{-1}x^{\mathrm{T}})\mathrm{MSE}$$

$$\mathrm{PEV}(x) = x(X^{\mathrm{T}}X)^{-1}x^{\mathrm{T}}\mathrm{MSE}$$

Note the only dependence on the observed values is in the variance (MSE) of the measurement error. You can look at the PEV(x) for a design (without MSE, as you don't yet have any observations) and see what e ct it will have on the measurement error - if it is greater than 1 it will magnify the error, and the closer it is to 0 the more it will reduce the error.

You can examine PEV for designs or global models using the Prediction Error Variance viewer. When you open it from the Design Editor, you can see how well the underlying model predicts over the design region. When you open it from a global model, you can view how well the current global model predicts. A low PEV (tending to zero) means that good predictions are obtained at that point. See Prediction Error Variance Viewer on page 3-50.

For information on the calculation of PEV for two-stage models, see Prediction Error Variance for Two-Stage Models on page 3-56.

Prediction Error Variance for Two-Stage Models

It is very useful to evaluate a measure of the precision of the model's predictions. You can do this by looking at Prediction Error Variance (PEV). Prediction error variance will tend to grow rapidly in areas outside the original design space. The following section describes how PEV is calculated for two-stage models.

For linear global models applying the variance operator to Equation 6-15 yields:

$$\mathrm{Var}(\beta) = (Z^T W^{-1} Z)^{-1} Z^T W^{-1} \mathrm{Var}(P) W^{-1} Z (Z^T W^{-1} Z)^{-1}$$

so

$$\mathrm{Var}(\beta) = (Z^T W^{-1} Z)^{-1} \tag{3-1}$$

since Var(P) = W. Assume that it is required to calculate both the response features and their associated prediction error variance for the i^{th} test. the predicted response features are given by:

$$\hat{p}_i = z_i \hat{\beta} \tag{3-2}$$

where z_i is an appropriate global covariate matrix. Applying the variance operator to Equation 3-2 yields:

$$\mathrm{Var}(\hat{p}_i) = z_i \mathrm{Var}(\hat{\beta}) z_i^T = z_i (Z^T W^{-1} Z)^{-1} z_i^T \tag{3-3}$$

In general, the response features are non-linear functions of the local t coe icients. Let g denote the non-linear function mapping θ_i onto p_i. Similarly let h denote the inverse mapping.

$$\hat{\theta}_i = h(\hat{p}_i) \tag{3-4}$$

Approximating h using a rst order Taylor series expanded about p_i (the true and unknown xed population value) and after applying the variance operator to the result:

$$\mathrm{Var}(\hat{\theta}_i) = \dot{h}\mathrm{Var}(\hat{p}_i)\dot{h}^T \tag{3-5}$$

where the dot notation denotes the Jacobian matrix with respect to the response features, p_i. This implies that $\dot{h}$ is of dimension (pxp). Finally the predicted response values are calculated from:

$$\hat{y}_i = f(\theta_i) \tag{3-6}$$

Again, after approximating f by a rst order Taylor series and applying the variance operator to the result:

$$\mathrm{Var}(\hat{y}_i) = \left[\frac{\partial f}{\partial \theta}\right]\Bigg|_{\hat{\theta}} \dot{h}\mathrm{Var}(\hat{p}_i)\dot{h}^T \left[\frac{\partial f}{\partial \theta}\right]^T\Bigg|_{\hat{\theta}} \tag{3-7}$$

After substituting Equation 3-3 into Equation 3-7 the desired result is obtained:

$$\mathrm{Var}(\hat{y}_i) = \left[\frac{\partial f}{\partial \theta}\right]\Bigg|_{\hat{\theta}} \dot{h} z_i (Z^T W^{-1} Z)^{-1} z_i^T \dot{h}^T \left[\frac{\partial f}{\partial \theta}\right]^T\Bigg|_{\hat{\theta}} \tag{3-8}$$

This equation gives the value of Prediction Error Variance.

Design Evaluation Tool

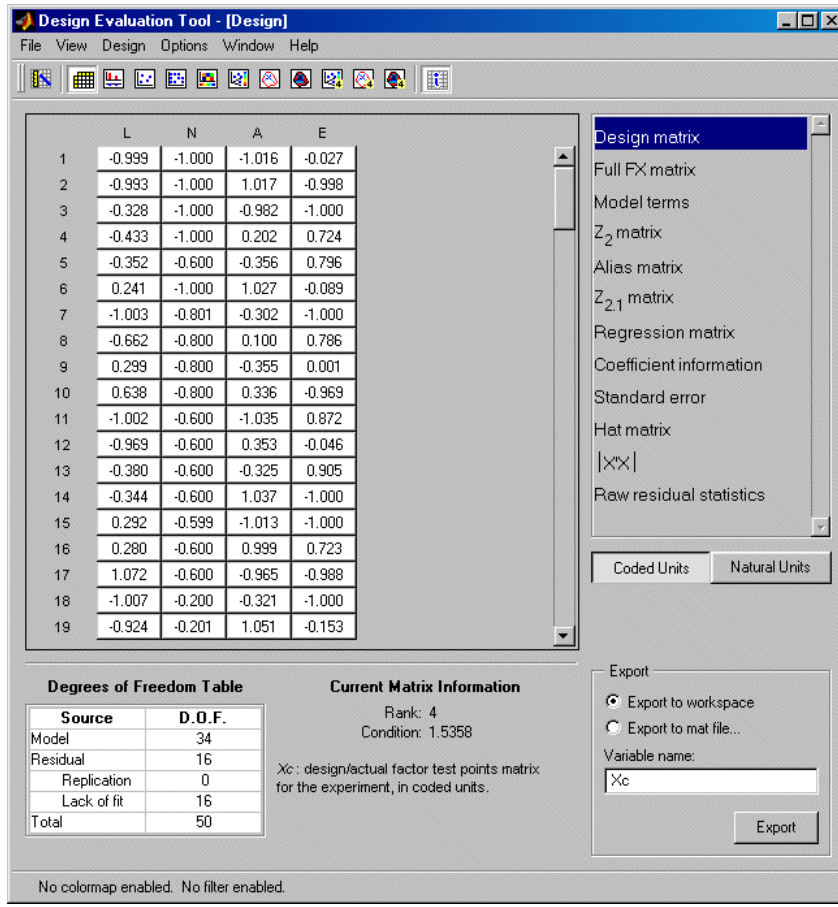
Introducing the Design Evaluation Tool on page 3-58
Table Options on page 3-60
Design Matrix on page 3-60
Full FX Matrix on page 3-60

Model Terms on page 3-61
Z2 Matrix on page 3-61
Alias Matrix on page 3-61
Z2.1 Matrix on page 3-62
Regression Matrix on page 3-62
Coe icient Information on page 3-62
Standard Error on page 3-64
Hat Matrix on page 3-64
|X'X| on page 3-64
Raw Residual Statistic on page 3-65
Degrees of Freedom Table on page 3-65
Design Evaluation Graphical Displays on page 3-66
Export of Design Evaluation Information on page 3-67

Introducing the Design Evaluation Tool

The Design Evaluation tool is only available for linear models.

You can open the Design Evaluation tool from the Design Editor or from the Model Browser windows. From the Design Editor select Tools > Evaluate Designs and choose the design you want to evaluate. From the Model Browser global view, you can click the button.



In the Design Evaluation tool you can view all the information on correlations, covariance, confounding, and variance in ation factors (VIFs). You can investigate the e ects of including or excluding model terms aided by this information (you must remove them in the Stepwise window). Interpretation is aided by color-coded values based on the magnitude of the numbers. You can specify changes to these criteria.

When you open the Design Evaluation tool, the default view is a table, as shown in the preceding example. You choose the elements to display from the list on the right. Click any of the items in the list described below to change the display. Some of the items have a choice of buttons that appear underneath the list box.

To see information about each display, click the toolbar button or select View > Matrix Information.

Table Options

You can apply color maps and filters to any items in a table view, and change the precision of the display.

To apply a color map or edit an existing one:

1 Select Options > Table > Colors. The Table Colors dialog box appears.
2 Select the check box Use a colormap for rendering matrix values.
3 Click the Define colormap button. The Colormap Editor dialog box appears, where you can choose how many levels to color map, and the colors and values to use to define the levels. Some tables have default color maps to aid analysis of the information, described below.

You can also use the Options > Table menu to change the precision (number of significant figures displayed) and to apply filters that remove specific values or values above or below a specific value from the display.

The status bar at bottom left displays whether color maps and filters are active in the current view.

When evaluating several designs, you can switch between them with the Next design toolbar button or the Design menu.

Design Matrix

Xn/Xc: design/actual factor test points matrix for the experiment, in natural or coded units. You can toggle between natural and coded units with the buttons on the right.

Full FX Matrix

Full model matrix, showing all possible terms in the model. You can include and exclude terms from the model here, by clicking on the green column headings. When you click one to remove a term, the column heading becomes red and the whole column is grayed.

The Full FX matrix is the same as the Jacobian for linear models if all terms included. Jacobian only includes 'in' terms. In the general case, the Jacobian is expressed:

$(i, j) = df/dp_j (x_i)$

In the case of linear models and RBFs this simpli es to:

(i,j) = j^{th} term evaluated at i^{th} data point = Jacobian matrix.

Model Terms

You can select terms for inclusion in or exclusion from the model here by clicking. You can toggle the button for each term by clicking. This changes the button from `in` (green) to `out` (red) and vice versa. You can then view the e ect of these changes in the other displays.

Note Removal of model terms only a ects displays within the Design Evaluation tool. If you decide the proposed changes would be bene cial to your model, you must return to the Stepwise window and make the changes there to t the new model.

Z2 Matrix

Z_2: Matrix of terms that have been removed from the model. If you haven't removed any terms, the main display is blank apart from the message All terms are currently included in the model.

Alias Matrix

Like the Z_2 matrix, the alias matrix also displays terms that are not included in the model (and is therefore not available if all terms are included in the model). The purpose of the alias matrix is to show the pattern of confounding in the design.

A zero in a box indicates that the row term (currently included in the model) is not confounded with the column term (currently not in the model). A complete row of zeros indicates that the term in the model is not confounded with any of the terms excluded from the model. A column of zeros also indicates that the column term (currently not in the model) could be included (but at the cost of a reduction in the residual degrees of freedom).

A: the alias matrix is de ned by the expression

$$A = (X'X)^{-1}X'Z_2$$

Z2.1 Matrix

As this matrix also uses the terms not included in the model, it is not available if all terms are included.

$Z_{2.1}$: Matrix de ned by the expression $Z_{2.1} = Z_2 - XA$

Regression Matrix

Regression matrix. Consists of terms included in the model. $n \times p$ matrix where n is the number of test points in the design and p is the number of terms in the model.

Coe cient Information

When you select `Coefficient information`, six buttons appear below the list box. Covariance is displayed by default; click the buttons to select any of the others for display.

Covariance

Cov(b): variance-covariance matrix for the regression coe icient vector b.

$$Cov(b) = (X'X)^{-1}$$

Correlation

Corr(b): correlation matrix for the regression coe icient vector b.

$$Corr(b)_{ij} = \frac{Cov(b)_{ij}}{\sqrt{(Cov(b)_{ii})}\sqrt{(Cov-(b)_{jj})}}$$

Correlation has a color map to aid analysis. You can view and edit the color map using Options > Table > Colors.

Partial VIFs

Variance In ation Factors (VIFs) are a measure of the non-orthogonality of the design with respect to the selected model. A fully orthogonal design has all VIFs equal to unity.

The Partial VIFs are calculated from the o -diagonal elements of Corr(b) as

$$VIF_{ij} = \frac{1}{(1 - Corr(b)_{ij}^2)} \quad \text{for} \quad p \geq i > j > 1$$

Partial VIFs also has a default color map active (<1.2 black, >1.2<1.4 orange, >1.4 red). A lter is also applied, removing all values within 0.1 of 1. In regular designs such as Box-Behnken, many of the elements are exactly 1 and so need not be displayed; this plus the color coding makes it easier for you to see the important VIF values. You can always edit or remove color maps and lters.

Multiple VIFs

Measure of the non-orthogonality of the design. The Multiple VIFs are de ned as the diagonal elements of Corr(b):

$$VIF_i = \{Corr(b)^{-1}\}_{ii}$$

Multiple VIFs also has a default color map active (<8 black, 8><10 orange, >10 red). A lter is also applied, removing all values within 0.1 of 1. Once again this makes it easier to see values of interest.

2 Column Corr.

Corr(X); correlation for two columns of X.

$$w_{ij} = \frac{x_{ij} - \bar{x}_j}{\sqrt{\sum_{i=1}^{N} (x_{ij} - \bar{x})^2}}$$

Let W denote the matrix of w_{ij} values. Then the correlation matrix for the columns of X (excluding column 1) is Corr(X), de ned as

Corr(X) = W'W

2 Column Correlation has the same default color map active as Correlation.

Single Term VIFs

Measure of the nonorthogonality of the design. The Single Term VIFs are de ned as

$$VIF_{ij} = \frac{1}{(1 - Corr(X)_{ij}^2)} \quad \text{for} \quad p \geq i > j > 1$$

Single term VIFs have a default color map active (<2 black, 2>red) and values within 0.1 of 1 are ltered out, to highlight values of interest.

Standard Error

σ_j : Standard error of the j[th] coe icient relative to the RMSE.

Hat Matrix

Full Hat matrix

H: The Hat matrix.

H = QQ'

where Q results from a QR decomposition of X. Q is an $n \times n$ orthonormal matrix and R is an $n \times p$ matrix.

Leverage values

The leverage values are the terms on the leading diagonal of H (the Hat matrix). Leverage values have a color map active (<0.8 black, 0.8>orange<0.9, >0.9 red).

|X'X|

D; determinant of X'X.

D can be calculated from the QR decomposition of X as follows:

$$D = \left[\prod_{i=1}^{p} (R_1)_{ii}\right]^2$$

where p is the number of terms in the currently selected model.

This can be displayed in three forms:

$|X'X|$

$\log(|X'X|)$

$|X'X|^{(1/p)}$

Raw Residual Statistic

Covariance

Cov(e): Variance-covariance matrix for the residuals.

`Cov(e) = (I-H)`

Correlation

Corr(e) : Correlation matrix for the residuals.

$$Corr(e)_{ij} = \frac{Cov(e)_{ij}}{\sqrt{(Cov(e)_{ii})}\sqrt{(Cov(e)_{jj})}}$$

Degrees of Freedom Table

To see the Degrees of Freedom table (and the information about each display), click the toolbar button or select View > Matrix Information.

Degrees of Freedom Table

Source	D.O.F.
Model	5
Residual	14
Replication	0
Lack of fit	14
Total	19

Source	D.F.
Model	p
Residual	n-p
Replication	by calculation
Lack of t	by calculation
Total	n

Replication is de ned as follows:

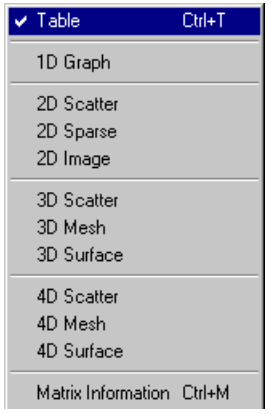
Let there be n_j (>1) replications at the j^{th} replicated point. Then the degrees of freedom for replication are

$$\sum_j (n_j - 1)$$

and Lack of t is given by n - p - degrees of freedom for replication.

Note: replication exists where two rows of X are identical. In regular designs the factor levels are clearly spaced and the concept of replication is unambiguous. However, in some situations spacing can be less clear, so a tolerance is imposed of 0.005 (coded units) in all factors. Points must fall within this tolerance to be considered replicated.

Design Evaluation Graphical Displays



The Design Evaluation tool has options for 1D, 2D, 3D, and 4D displays. You can switch to these by clicking the toolbar buttons or using the View menu.

Which displays are available depends on the information category selected in the list box. For the Design matrix, (with su icient inputs) all options are available. For the Model terms, there are no display options.

You can edit the properties of all displays using the Options menu. You can con gure the grid lines and background colors. In the 2D image display you can click points in the image to see their values. All 3D displays can be rotated as usual. You can edit all color map bars by double-clicking.

Export of Design Evaluation Information

All information displayed in the Design Evaluation tool can be exported to the workspace or to a `.mat` file using the radio buttons and Export button at the bottom right. You can enter a variable name in the edit box.

Saving, Exporting, and Importing Designs

To save your designs with your project and close the Design Editor, select File > Save and Close or the toolbar button.

You do not need to save your designs separately from the project. When you save your project in the Model Browser, your designs remain part of that project. You can also export designs to a file or the workspace.

To export your design to a file:

1 Select File > Export Design. The selected design only will be exported.

2 Choose an export option:

   `Comma separated format file (*.csv)` exports the matrix of design points to a CSV (comma-separated-values) file. You can include factor symbols by selecting the check boxes.

   `Design Editor file (*.mvd)` generates a Design Editor file (`.mvd`).

   `Workspace` exports the design matrix to the workspace. You can convert design points to a range of (1, -1) by selecting the check box.

3 Choose the destination file or variable by typing in the edit box or using the browse button.

Import designs by selecting File > Import. The controls on the dialog box are very similar to the Export Design dialog box: you can import from Design Editor files, CSV files, or the workspace, and you can convert design points from a (1,-1) range.

See Also

Related Examples

Design of Experiments

Create a Constrained Space-Filling Design

Fit Models to Collected Design Data

After you collect data at your design points, return to the Model Browser to import data and fit models.

1. Open the Model Browser by typing

   ```
   mbcmodel
   ```

2. Open the project containing your designs.
3. In the test plan node view, in the Common Tasks pane, click Fit models.
4. Follow the prompts in the Fit Models to Data wizard to select data, optionally match data to design points, and fit models to the data.
5. Evaluate models.

See Also

Related Examples

Select Data for Modeling Using the Fit Models Wizard on page 4-22

Match Data to Designs on page 4-28

More About

Model Assessment

4

Data

Using Data

This section describes all aspects of loading, manipulating, and selecting data in the Model Browser. The Data Editor provides a powerful graphical interface for dealing with data:

Plot, filter, group, and edit data, and you can define new variables. You can match data to designs. You can reach the Data Editor from every node in the model tree, so you can also examine and export your modeling data. The Data Editor contains various tools for these tasks:

Data Import Wizard for loading and merging data

Variable Editor, Filter Editor, Test Filter Editor, and Test Notes Editor are dialog boxes for creating and editing new variables and data filters.

Storage dialog box for storing new variables, data filters, and plot settings

Test Groupings dialog box can be used for plotting and manipulating data groups.

Within the Data Editor there are 2-D, 3-D, and multiple data plots for viewing data, and design match plots for viewing data and design points.

You use the Data Editor for matching data to experimental designs. You can set tolerances for automatic selection of the nearest data points to the specified design points, or select data points manually.

You use the Data Wizard to select data for modeling. You can also set up matching data to designs by setting tolerances and automatically opening the Design Match views within the Data Editor. You reach the Data Wizard from test plan level.

You can load and merge data from the following:

From files (Excel®, Concerto, MATLAB)

From the workspace

You can also write your own data-loading functions.

In the Data Editor, you can do the following:

View plots, edit and add data records.

Define new variables.

Apply filters to remove unwanted records, filter by test, and apply notes to tests that fulfill set criteria.

Store and retrieve user-de ned variables and lters.

Collect data into groups.

Match data to experimental designs using the Design Match views.

Export data and modeling output to le and to the workspace.

Use the Data Wizard to:

Select the data set and design to use for modeling.

Select the data signals to use for model input factors (one-stage, or local and global for two-stage).

Select matching tolerances (if matching data to a design).

Select data signals for response model input factors.

See Also

More About

Create Variables on page 4-15

Create Filters on page 4-17

Data Loading Application Programming Interface on page 4-34

Data Wizard on page 4-22

De ne Test Groupings on page 4-20

Load and Edit Data on page 4-4

Merge Data on page 4-10

View and Edit Data on page 4-5

Load and Edit Data

In this section...

Load Data on page 4-4
Modify Data on page 4-4
View and Edit Data on page 4-5

Load Data

Choose a modeling work ow:

If you want to t models to data, open the Model Browser home page if necessary (use the toolbar button or File > Home), then click Import Data. Choose le or workspace, then use the le browser to select a le or les to import.

If you have designed an experiment, collected data, and want to import data for modeling, from the test plan node, in the Common Tasks pane, click Fit models. The Fit Models Wizard guides you through selecting data.

Alternatively, to load data without following a modeling work ow task:

To load data from a le or les, click Import Data on the home page, or select Data > Import Data from File.

To load data from the Workspace, select Data > Import Data from Workspace.

Modify Data

To load or merge new data, do one of the following:

If you already loaded data and want to merge in more data, click Import Data on the home page. The wizard prompts you with merging options.

If you want to re t existing models to a new data set, from the test plan node, in the Common Tasks pane, click Fit models. In the Fit Models Wizard, you can import new data or select another data set in the project.

If you already loaded data, you can open the Data Editor to view, modify, or copy the data as follows:

From the project node, double-click a data set in the Data Sets pane.

From the project node, select Data > Edit Data, or Copy Data.

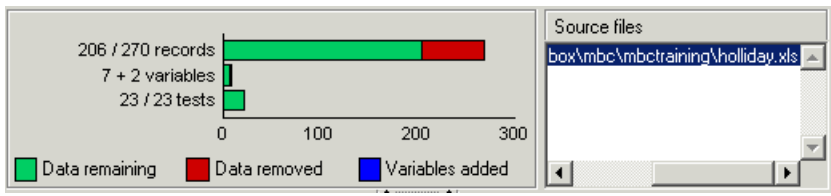
From the test plan node, select TestPlan > Edit Data.

From any modeling node, click the View Modeling Data toolbar button.

You can split the view to display several plots at once. Use the context menu to split the views. You can choose 2-D plots, 3-D plots, multiple data plots, data tables, and list views of filters, variables, test filters, and test notes.

View and Edit Data

You can open multiple different views at the same time in the Data Editor. You can display 2-D plots, 3-D plots, multiple data plots, data tables, and views of your filters, variables, and test notes. Use the toolbar buttons, the View menu, or right-click a view title bar to split views and change view types.



The list box at the top right contains the source file information for the data, and other information is displayed on the left. The Summary tab lists the numbers of records, variables, and tests. The bars and figures show the proportion of records removed by any filters applied, and the number of user-defined variables is shown. For this example with two user-defined variables added to a data set originally containing seven variables, you see '`7 + 2 variables`.'

The Test Selector list pane on the left is constant for two-stage and point-by-point data, but not needed for one-stage data. Tests selected here apply to 3-D plots, multiple data plots, and tables. It does not apply to 2-D plots because they have their own independent test controls. If you are viewing read-only local modeling data, the selected test is shown in the Tests pane and remains synchronized if you change test in the Model Browser.

By default new data sets are called Data Object. You can change the names of data sets at the project node by select-clicking a data set in the Data Sets list, or by pressing F2 (as when selecting to rename in Windows Explorer).

To edit data plot properties (2-D, 3-D, or multiple), you can right-click the plot and select Properties. Here you can choose to show the legend and the grid. You can choose the line style if you want to connect the data points and the data marker point style, if any. Reorder X Data (2-D plots) redraws the line joining the points in order from left to right. The line might not make sense when drawn in the order of the records.

2-D Plots

In the 2-D plot view you can select combinations of variables and tests to plot from the list boxes on the left. Multiple selection of tests and y-axis variables is possible as show in the following gure, multiple tests are selected to view several tests simultaneously.

To edit data plot properties, right-click the plot and select Properties. Here you can choose to show the legend and the grid. You can choose the line style if you want to connect the data points and the data marker point style, if any. Reorder X Data redraws the line joining the points in order from left to right. The line might not make sense when drawn in the order of the records. You can use the Show Removed Data check box to plot outliers you have removed.

Click points to select outliers, and remove them by selecting Tools > Remove Data(or use the keyboard shortcut Ctrl+A). Select Tools > Restore Data(or use the keyboard shortcut Ctrl+Z) to open a dialog box where you can choose to restore any or all removed points.

3-D Plots

In 3-D data plot views you can select the variables for each axis from the drop-down menus, and rotate the plot by clicking and dragging. To edit data plot properties, you can right-click the plot and select Properties. Here you can choose the color and style of the axes, whether to show the grid in each axis, and perspective or orthographic axes projection.

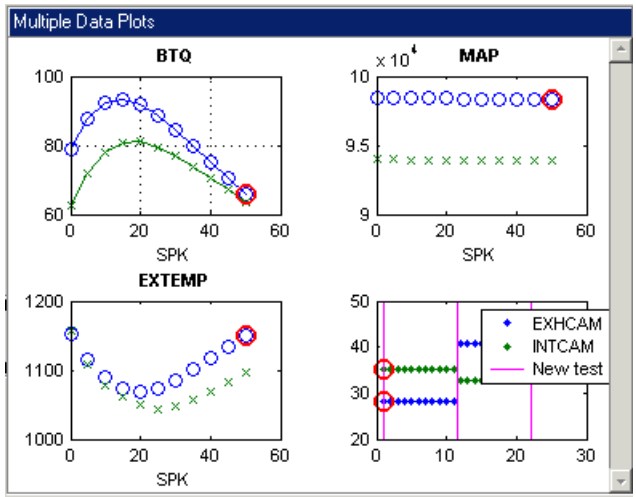
Multiple Data Plots

Here you can add as many 2-D plots as desired to the same view, plotting the same selection of tests in a variety of di erent plots. Use the right-click context menu to add and remove plots, select plot variables, and edit plot properties. You can select single or multiple Y variables to plot against a single X variable (or no X variable) in the Plot Variables dialog box. Select tests to display in the list on the left of the Data Editor, as for 2-D and 3-D plots and the table view.

Note that you can select di erent plot properties and variables for each plot within the Multiple Data Plots view, as shown in the example following. Click to select a plot (or

right-click) before selecting Plot Variables or Plot Properties. For each plot you can use the same plot properties options as for the single 2-D data plots. You can choose to show the legend and the grid. You can choose the line style if you want to connect the data points and the data marker point style, if any. Reorder X Data redraws the line joining the points in order from left to right. The line might not make sense when drawn in the order of the records. To view removed data in the Multiple Data Plots, select Properties and select the box Show removed data.

Click points to select outliers, and remove them by selecting Tools > Remove Data (or use the keyboard shortcut Ctrl+A). Selected outliers are outlined in red, as shown in the following example.



Select Tools > Restore Data (or use the keyboard shortcut Ctrl+Z) to open a dialog box where you can choose to restore any or all removed points.

Design Match Plots

You use these views for matching data to design points. Use the list in the design match view to examine your data and design. Click points in the plot to select them across the Data Editor that is, the selected points are displayed in the table view and other data plot views (except 2-D plots, which have separate controls).

Summary tabs Statistics, Variables, Filters, Test Filters, and Test Notes

These tabs show lists and information such as variable and filter definitions, the notes applied to filtered tests, and the data and design points in selected clusters. Variable and Filter tabs show the definitions of each variable or filter. Double-click to select particular filters or variables to edit.

The Test Notes List view shows the rules used to define notes on the data, along with the actual note, the color of the note, and the number of tests to which that note applies. The specified rule is applied to each test in turn to decide if that test should be noted: e.g., mean(TQ) > 0 evaluates the mean torque for each test and notes those tests where the value is greater than zero.

Table View

In the Table view, you can view your data, edit, and add records.

Points you have selected in plots (by clicking) are outlined in red in the table. Points you have removed as outliers are light red in the table with a filter icon next to the row number. Edited cells become blue.

Tip To view removed data in the table view, right-click and select Allow Editing. Removed records are red.

Right-click the title bar for these options.

Duplicate Selected Records First select one or more records, then use this option to duplicate them. Each duplicate appears directly underneath the parent record. Edit duplicates to create new records. You must first select Allow Editing to enable this option.

Undo Edits in Selected Region You can click and drag to highlight a region, then use this option to reverse any edits in the highlighted area. You must first select Allow Editing to enable this option.

Allow Editing Toggles editing, and, as a side effect, causes all records to be shown, including those which are filtered out. Records which are filtered out appear light red in the table, with a filter icon next to the row number. You can alter records by clicking a cell and then typing a new value. Edited cells become blue. Editing the value of a cell may cause that row to be filtered out. If so, the background color of the row will change after the cell has been edited.

Select Columns To Display Opens a dialog box where you can use the check boxes to select the columns to display in the table. Select a column, then press Ctrl+A to select all columns, and then you can select or clear all check boxes with one click. You can click and drag column headers in the table view to rearrange columns.

See Also

Related Examples

Create Variables on page 4-15

Create Filters on page 4-17

De ne Test Groupings on page 4-20

Match Data to Designs on page 4-28

Merge Data on page 4-10

Select Data for Modeling Using the Fit Models Wizard on page 4-22

Merge Data

In this section...

About Data Loading and Merging on page 4-10
Loading and Merging Data from File on page 4-10
Loading Data from the Workspace on page 4-12
Tailor-Made Excel Sheets on page 4-13

About Data Loading and Merging

You can load and merge data from files, from the workspace, and from tailor-made Excel sheets, as described in the following sections.

A test plan can only use a single data set, so if you want to use more than one source file, you need to use the merge functions to combine data variables in order to incorporate desired variables into one model. You can import and merge multiple files at once.

Loading and Merging Data from File

Note Only use this workflow to merge data. Otherwise, see Load Data on page 4-4.

Use the following workflow if you want to merge data.

Data Import

If you already have data loaded and want to merge in more data, use the following workflow.

1 In the Data Editor, select File > Import > File. To import data from a file, enter the file pathname in the edit box, or use the Browse button to find and select the data file or files.

   The drop-down menu contains the file types recognized by the Model Browser (Excel, Delimited text files, MATLAB). The default tries to determine the type of file by looking at the file extension.

2 For Excel files with multiple sheets, you must next select the sheet you want to use and click OK.

3 The Import Wizard now displays a summary screen showing the total number of records and variables imported, and you can view each variable's range, mean, standard deviation, and units in the list box. You can edit variable names and units in this list. Click Finish to accept the data, unless you have data to merge.

Merging Data

1 If you already have some data loaded, you cannot click Finish but must click Next instead. This brings you to the data merging screen of the wizard.

2 Here you must choose one of the radio buttons:

Append new records (all variable names and units match current data)

Append new records (add NaNs to unmatched variables)

Add new variables to current data

Overwrite current data

3 To accept the data and return to the Data Editor, click Finish.

Note The merge might not succeed if the data sets do not contain the same variables. A message appears if the merge is impossible and you must make another choice.

Text File Formats

Select `Delimited Text File` from the Open As list to read delimited text files into MBC. These files can be delimited by tabs, |, commas or spaces. They may optionally contain a variable name header and units header line.

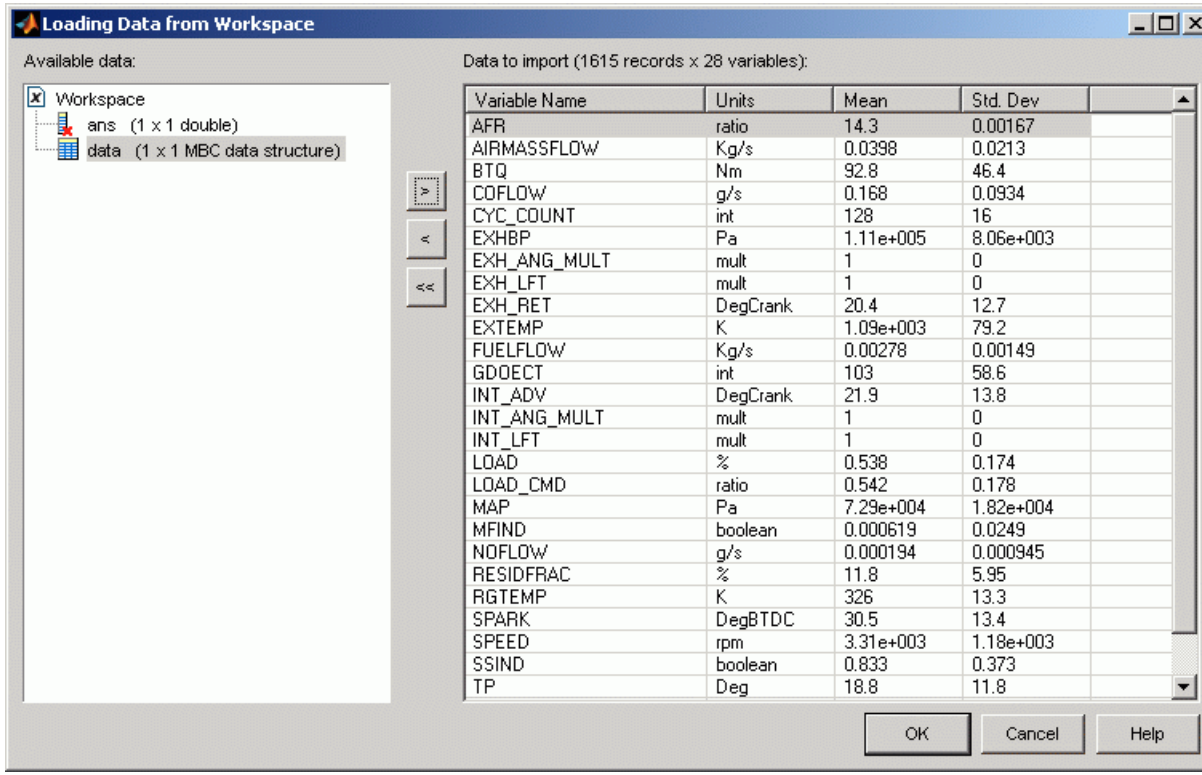
Select `Delimited Text Sweep File` from the Open As list to read delimited text files with multiple blocks of data into MBC. These files can be delimited by tabs, |, commas or spaces. Each block of data may optionally contain a variable name header and units header line. The first block's names and headers are used, subsequent rows of non-numeric data are stripped out. The data can be split into multiple blocks where each block contains data for a separate sweep. Within each block, a column that consists of a value in the first row and empty values in the remaining rows for initial columns will be assumed as meaning that for these columns each row should take its value from the first row in the block. Note that the data is imported as a single array and you must use the Test Groupings window to decide how to sort the data into sweeps. See Define Test Groupings on page 4-20.

Loading Data from the Workspace

1 From the project node choose Data > Import Data from Workspace.The Data Editor on page 4-4 opens.

You can also import variables in the Data Editor by choosing File > Import > Workspace.

The Loading Data from Workspace dialog box appears.



Variable Name	Units	Mean	Std. Dev
AFR	ratio	14.3	0.00167
AIRMASSFLOW	Kg/s	0.0398	0.0213
BTQ	Nm	92.8	46.4
COFLOW	g/s	0.168	0.0934
CYC_COUNT	int	128	16
EXHBP	Pa	1.11e+005	8.06e+003
EXH_ANG_MULT	mult	1	0
EXH_LFT	mult	1	0
EXH_RET	DegCrank	20.4	12.7
EXTEMP	K	1.09e+003	79.2
FUELFLOW	Kg/s	0.00278	0.00149
GDOECT	int	103	58.6
INT_ADV	DegCrank	21.9	13.8
INT_ANG_MULT	mult	1	0
INT_LFT	mult	1	0
LOAD	%	0.538	0.174
LOAD_CMD	ratio	0.542	0.178
MAP	Pa	7.29e+004	1.82e+004
MFIND	boolean	0.000619	0.0249
NOFLOW	g/s	0.000194	0.000945
RESIDFRAC	%	11.8	5.95
RGTEMP	K	326	13.3
SPARK	DegBTDC	30.5	13.4
SPEED	rpm	3.31e+003	1.18e+003
SSIND	boolean	0.833	0.373
TP	Deg	18.8	11.8

Variables in the workspace are displayed in hierarchical form in the top left pane.

1 Select a variable to import in the tree at top left.

2 Click the Add button.

The number of records and variables appears in the Output Data pane. You can add variables (one at a time) as many times as you like (as long as there are no name con icts; su ixes _1 are added to repeated names).

You can double click to edit variable names and units.

3 You can use the Remove button to remove selected variables one at a time from the right list, or click or Remove all to remove all variables at once.

4 Click OK to accept the data to import and return to the Data Editor.

Data Merging

If you already have data loaded, the Data Merging Wizard appears, where you must choose one of these radio buttons:

Append new records (all variable names and units match current data)

Append new records (add NaNs to unmatched variables)

Add new variables to current data

Overwrite current data

Click Finish.

Note The merge might not succeed if the data sets do not contain the same variables. A message appears if the merge is impossible and you must make another choice.

Click Finish to accept the data and return to the Data Editor.

Tailor-Made Excel Sheets

The Data Editor can create a tailor-made Excel sheet for you to ll with data and then import. This sheet will be in the format the Data Editor expects to import data.

1 Select File > Import Excel.

Excel is opened with a new sheet created, containing the labeled rows `Name`, `Unit`, and `Data`.

2 Copy your data and variable names into this sheet, then click Next in the wizard to import into the Data Editor.

If the data has been entered in the columns in a way that the Data Editor expects, a summary screen shows you information about the numbers, ranges, means, units, and standard deviations of the records and variables you can import.

3 Click Finish to import the data.

See Also

More About

Create Variables on page 4-15

Create Filters on page 4-17

Data Editor on page 4-4

De ne Test Groupings on page 4-20

Create Variables

In this section...

How to Create Variables on page 4-15
New Variables on page 4-15

How to Create Variables

You create variables by doing the following:

Click the Edit user-de ned variables toolbar button.

Selecting Tools > Variables

Alternatively, click the Variables tab, then press Insert.

You can also load user-de ned variables.

View your user-de ned variables in the Variables tab.

To edit existing variables:

Rename directly by select-clicking in the Variables tab or press F2 (as in renaming in Windows Explorer).

To edit in the Variable Editor, double-click a variable in the Variables tab.

Select Tools > Variables

Delete variables by selecting them in the Variables tab and pressing Delete

New Variables

You can de ne new variables in terms of existing variables:

De ne the new variable by writing an equation in the edit box at the top of the Variable Editor dialog box.

You can type directly in the edit box or add variable names by double-clicking them. In the case of variable names especially, this latter method avoids typing mistakes. Variable names are case sensitive.

For example, to define a new variable called `POWER` that is defined as the product of two existing variables, `tq` and `n`, enter `POWER = tq x n`.

Click **OK** to add the new variable to the current data set.

To edit a variable, click a variable in the left list, or click the button to add a new item to the list if you want to add a new variable.

Note The computation of variable values is vectorized and occurs prior to filtering and clustering. The result must be either a vector the same length as the dataset or a scalar, in which case the value is repeated for every record. Variables are used on a record-by-record basis.

See Also

More About

Variable Editor on page 4-15

Vectorization (MATLAB)

Create Filters

How to Create Filters

You can create filters

- By clicking the toolbar button
- By selecting Tools > Filters
- Alternatively, by selecting an existing Filter List view by clicking in it, then pressing Insert

You can also import user-defined filters.

View filters in the Filters tab. Filter effects are shown graphically in the Summary tab removed data is shown in red.

After you create them, filters can be edited in the same way as variables:

- Directly, after you select-click them in a Filters tab, or by pressing F2
- Using Tools > Filters, which opens the Filter Editor
- By double-clicking, which also opens the Filter Editor
- Delete filters by selecting them and pressing Delete.

Test Filters and Test Notes

Similarly you can add test filters (to filter out entire tests, instead of individual observations) and test notes (to mark every test that fulfills set criteria). The Test Filter Editor and Test Notes Editor can be reached from the Tools menu. You can view these filters in the Data Editor in the same way as the other filters by selecting Test Filters or Test Notes tabs. You define, edit, store, and delete these filters in the same way, and store and import them.

Filter Editor

A filter is a constraint on the data set used to exclude some records. You define the filter using logical operators on the existing variables.

For example, if you enter `n>1000`, the e ect of this lter is to keep all records with speed (n) greater than 1000.

Click OK to impose new lters on the current data set.

The Filter Editor looks di erent depending on whether you opened it to create a new lter or edit an existing one. The example above shows the editor when adding a new lter. If you open the editor to edit a lter there is an additional list on the left. You can choose which of your existing lters to edit from this list, or click the button to add a new item to the list if you want to add a new lter.

Import Variables, Filters, and Editor Layout

You can store and import plot preferences, user-de ned variables, lters, and test notes so that you can apply them to other data sets loaded later in the session.

To store expressions in a le, in the Data Editor, select Tools > Export Expressions and select a le name.

To import, either:

Select the menu Tools > Import Expressions

Use the toolbar button

In the Import Variables, Filters, and Editor Layout dialog box, use the toolbar buttons to import variables, lters and plot layouts. Import from other data sets in the current project, or from MBC project les, or from les exported from the Data Editor.

To use imported expressions in your current project, select items in the lists and click the toolbar button to apply in the data editor.

See Also

More About

Load and Edit Data on page 4-4

Merge Data on page 4-10

Using Data on page 4-2

De ne Test Groupings

The De ne Test Groupings dialog box collects records of the current data set into groups; these groups are referred to as tests. Test groupings are used to de ne hierarchical structure in the data for two-stage modeling.

Open the dialog box from the Data Editor by doing one of the following:

Using the menu Tools > Test Groups

Using the toolbar button

1 Select a variable in the list box to use in de ning groups within the data.

2 The Add Variable button ( ) adds the currently selected variable in the Variables list to the list view on the left. Alternatively, double-click variables to add them.

You can now use this variable to de ne groups in the data.

The maximum and minimum values of grouping variables are displayed.

Use the Tolerance to de ne groups: on reading through the data, when the value of n changes by more than the tolerance, a new group is de ned. You can change the tolerance by typing directly in the edit box.

You can de ne additional groups by selecting another variable and choosing a tolerance. Data records are then grouped by any of the grouping variables changing outside their tolerances.

You can plot variables without using them to de ne groups by clearing the Group By check box.

Remove variables from grouping by selecting the unwanted variable in the list view (the selection is highlighted in blue) and clicking the Remove Variable button .

The color of the Tolerance text corresponds to the color of data points in the plot. Vertical pink bars show the tests (groups). You can zoom the plot by Shift-click-dragging or middle-click-dragging the mouse.

One-stage data de nes one test per record, regardless of any other grouping. Select this to use the data in creating one-stage models.

Sort records before grouping allows you to reorder records in the data set. Otherwise, the groups are de ned using the order of records in the original data set.

Show original test groups displays the original test groupings if any were de ned.

Test number variable contains a drop-down menu showing all the variables in the current data set. You can select any of these to number the tests (for example, lognumber could be useful (instead of 1,2,3...) if the data was taken in numbered tests and you want access to that information during modeling).

Every record in a test must share the same test number to identify it, so when you are using a variable to number tests, the value of that variable is taken in the rst record in each test.

Test numbers must be unique, so if any values in the chosen variable are the same, they are assigned new test numbers for the purposes of modeling. (This does not change the underlying data, which retains the correct lognumber or other variable.)

Click OK to accept the test groupings de ned and close the dialog box.

Select Data for Modeling Using the Fit Models Wizard

In this section...

Choose a Work ow on page 4-22
Opening the Fit Models Wizard on page 4-22
Step 1: Select Data Set on page 4-22
Step 2: Select Input Signals on page 4-23
Step 3: Select Response Models on page 4-24
Step 4: Set Tolerances on page 4-25

Choose a Work ow

Use the work ow on this page if you have designed an experiment, collected data, and want to use that data for tting models. The Fit Models Wizard guides you through the steps. You can also re t existing models to new data using the Fit Models Wizard.

If you want to t models to data and do not have an existing test plan in your project, use the Import Data common task work ow instead. Follow the steps in Model Set Up .

Opening the Fit Models Wizard

Set up a test plan for designs by following the steps in Set Up Design Inputs on page 3-8. After you collect data, return to the Model Browser to import data and t models. In any test plan, to select new data to t models to, do either of the following:

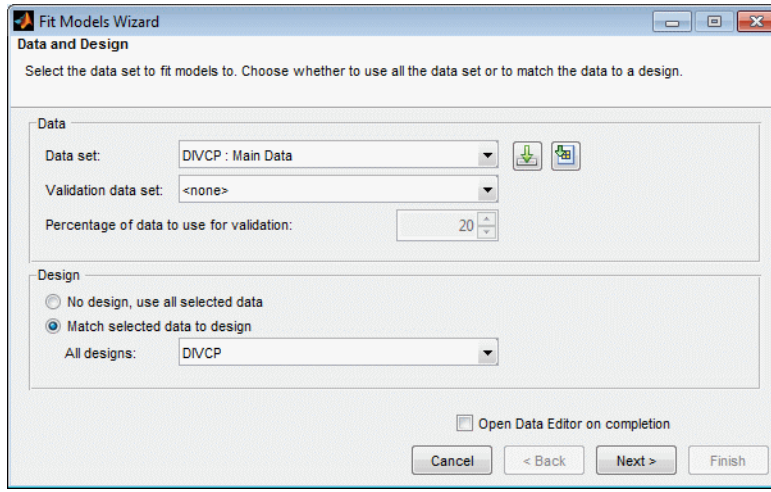
- In the Model Browser test plan view, in the Common Tasks pane, click Fit models, or select TestPlan > Fit Models.
- Double-click the Responses block in the test plan diagram.

The Fit Models Wizard opens.

Step 1: Select Data Set

Select the data set to t models to. Select a data set in the project from the list, or click the button to load a new data set if needed. If you load new data, select the check box Open Data Editor on completion to inspect or edit the data before modeling.

If the test plan contains any designs, choose whether to use all the data set or to match the data to a design. If matching to a design, select a design in the list.

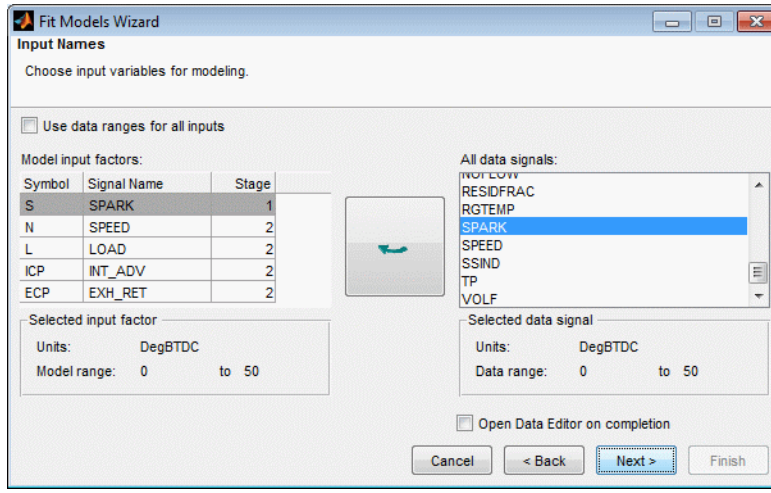


Step 2: Select Input Signals

Select the input signals for the model (on all stages of the hierarchical model) from the list box of data signals on the right, and match them to the correct model variables using the big button with the arrow. Double-click an item in the data signals list to select that signal for the currently selected input factor and then move to the next input.

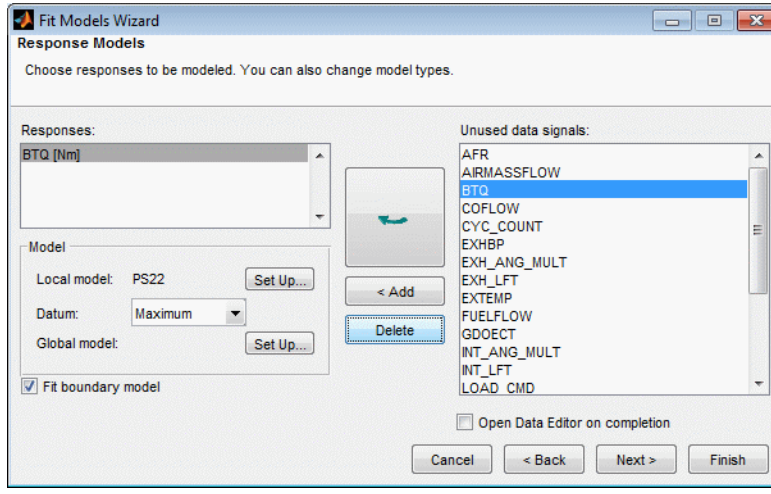
If you entered the correct signal name at the model setup stage, the appropriate signal is automatically selected as each model input factor is selected. This can be time-saving if there are many data signals in a data set. If the signal name is not correct, you must click to select the correct variable for each input.

If you want to use the range of the data signals for model input ranges, instead of the ranges set in the design inputs, then select the Use data ranges for all inputs check box.



To continue to selecting response models, click Next.

Step 3: Select Response Models



Use the following controls to set up your models:

If starting with an empty Responses list box, select the desired response in the Unused data signals list, and click Add.

If you are using a test plan template or selecting new data in a test plan, responses may already be specified in the Responses list box. If you want to change a response, select a signal and click the large button with the arrow to replace the current selected response. The previous response appears in brackets to inform you what has changed.

When there is already a response in the list box, clicking Add does not replace the selection but increases the number of responses. The replace button (with the arrow) is not available when the Responses box is empty.

You can use Delete to remove selected responses.

You can select Datum models (if the local model supports them), and you can use the Set Up buttons to change the local and global models. See Explore Local Model Types on page 5-7, Explore Global Model Types on page 5-62, and Datum Models on page 5-81 for details.

To create a boundary model, leave the Fit boundary model check box selected.

To continue:

Click Finish to fit the models.

If you need to define test groupings for two-stage or point-by-point models, the Test Groupings dialog box opens. Verify or change the test groupings and click OK to continue model fitting. See Define Test Groupings on page 4-20.

If you selected the check box Open Data Editor on completion, the Data Editor opens to inspect or edit data before modeling. For next steps working with data, see Using Data on page 4-2. The models are fitted after you close the Data Editor.

For next steps evaluating your models, see Model Assessment.

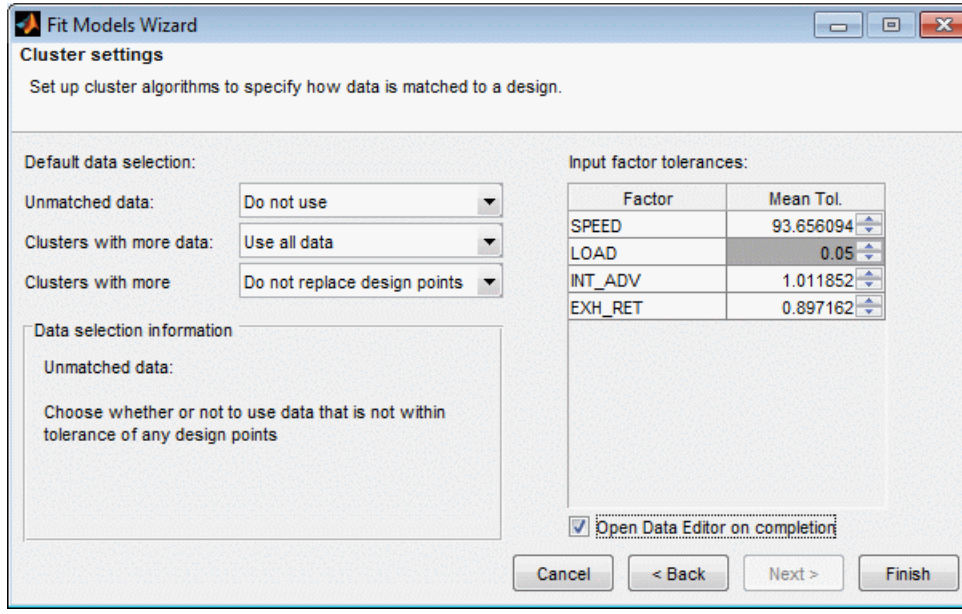
If you are matching data to a design, click Next. See Step 4: Set Tolerances on page 4-25.

Step 4: Set Tolerances

You can only reach Step 4 if you are matching data to a design. Setting tolerances is only relevant if you are matching data to a design. You can only match data to designs for global models.

You can also edit tolerances later using the context menu in the Data Editor window. See The Tolerance Editor on page 4-30 for definitions of clusters.

Set Input factor tolerances for each variable to determine the size of the tolerance in each dimension. This is used for selecting data. The tolerance in each dimension determines the size of clusters centered on each design point. Data points that lie within tolerance of a design point are included in that cluster. Data points that fall inside the tolerance of more than one design point form a single cluster containing all those design and data points. If no data points fall within tolerance of a particular design point, they remain unmatched. Default tolerance values are related to variable ranges.



The choices you make in the Default data selection options determine how the cluster algorithm is rst run to select matching data and design points. This only a ects the status of the check boxes for data and design points when you rst see the Design Match view in the Data Editor. You can always alter the results of this later in the Design Match list, where you can manually select the data and design points you want to use.

When you exit the Data Editor these selections determine what data is used for modeling and how design points are augmented and replaced. Selected data is used for modeling and added to the design. Data you have decided to exclude is not used for modeling or added to the design.

Unmatched data Use or Do not use

Data that does not lie within tolerance of any design point is unmatched. You can decide what to do with these. If you select `Use`, this data is selected for modeling and added to your design.

If you select `Do not use`, then unmatched data is not used for modeling or added to the design; it is excluded data.

Clusters with more data `Use all data` or `Use closest match only`

This refers to clusters containing more data points than design points. If you choose `Use all data`, all the data points in these clusters are selected for modeling and added to the design, replacing the design points in those clusters.

If you choose `Use closest match only`, then a one-to-one match of the data point closest to each design point is selected, and these are the only points that are selected for modeling and added to the design (replacing a design point each).

Cluster with more design `Do not replace design points` or `Replace design with closest`

All data from these clusters is selected for modeling. The setting here only affects selections for the design.

Where clusters contain more design points than data points, you can choose to leave the design unchanged by selecting `Do not replace design points`.

If you choose `Replace design with closest`, this replaces the design points where possible with the closest data point and leaves the rest of the design points unchanged.

Remember you can override any of these selections manually in the Data Editor; changes are only applied when you close the Data Editor after matching. The Design Match view in the Data Editor window appears by default when you close the Fit Models Wizard while you are matching data to designs. See Match Data to Designs on page 4-28.

Match Data to Designs

In this section...

Introducing the Design Match View on page 4-28
How to Use the Design Match View on page 4-28
The Tolerance Editor on page 4-30
What Will Happen to My Data and Design? on page 4-31

Introducing the Design Match View

You can use the Design Match view for matching data to experimental designs for global models. Here you can select data for modeling. You can use an iterative process: make a design, collect some data, match that data with your design points, modify your design accordingly, then collect more data, and so on. You can use this process to optimize your data collection process in order to obtain the most robust models possible with the minimum amount of data.

Use the Design Match view to select data for modeling. All data you select is also added to a new design called `Actual Design`. You can use the matching process to produce an `Actual Design` that accurately re ects your current data. You can then use this new design to decide the best points to use if you want to augment your current design in order to collect more data.

For instructions, see the following section, How to Use the Design Match View on page 4-28.

How to Use the Design Match View

Tip For a step-by-step guide to matching data to a design using an example project, see Match Data to Designs on page 4-28 in the Getting Started documentation.

You can Shift+click (or center+click) and drag to zoom in on clusters of interest. Double-click the plot to return to full size.

Use the following sequence as a guideline for matching data to designs using the Design Match plot:

1 It is unlikely that you will get the tolerances right immediately. Open the Tolerance Editor using the context menu and try di erent values for di erent variables. These values determine the size of clusters centered on each design point. Data points that lie within tolerance of any design point in a cluster are matched to that cluster. See The Tolerance Editor on page 4-30 for cluster de nitions.

2 For matching data to designs, you might want to clear the check box in the Design Match for green clusters (with equal data and design points). These clusters are matched; you are more likely to be interested in unmatched points and clusters with uneven numbers of data and design points. Removing the green clusters allows you to focus on these points of interest. If you want your new Actual Design to accurately re ect your current data, your aim is to get as many data points matched to design points as possible, that is, as few red clusters as possible. See Red Clusters on page 4-30.

3 You can see the values of variables at di erent points by clicking and holding. Selected points have a pink border. Once points are selected, you can change the plot variables using the `X-` and `Y-axis factor` drop-down menus to track those points through the di erent dimensions.

This can give you a good idea of which tolerances to change in order to match points. Remember that points that do not form a cluster can appear to be perfectly matched when viewed in one pair of dimensions; you must view them in other dimensions to nd out where they are separated beyond the tolerance value. Use this tracking process to decide whether you want particular pairs of points to be matched, and then change the tolerances until they form part of a cluster.

Remember that points you select in the design match view are selected across the Data Editor, so if you have other data plots or a table view open you can investigate the same points in di erent views.

4 Once you have found useful values for the tolerances by trial and error, you can make selections of points within clusters that have uneven numbers of data and design points. These clusters are blue (more data than design) or red (more design than data). Select any cluster by clicking it. The details of every data and design point contained in the selected cluster appear in the Cluster Information list. Choose the points you want to keep or discard by selecting or clearing the check boxes next to each point. Notice that your selections can cause clusters to change color as you adjust the numbers of data and design points within them.

5 You can also select unmatched points by right-clicking and selecting Select Unmatched Data. All unmatched points then appear in the list view. You can decide whether to include or exclude them in the same way as points within clusters, by

using the check boxes in the list. If you decide to exclude data points (within clusters or not) they appear on the plot as black crosses (if the Excluded Data check box is selected for display).

Note that it is a single fast operation to multiple-select points before selecting or clearing a check box, rather than selecting points individually. To do this, use Shift +click to select multiple points and hold the Shift key when clicking one of the check boxes.

You can right-click and select Show Labels to see design and data point numbers on the plot (also in the View menu).

Continue this process of altering tolerances and making selections of points until you are satis ed that you have selected all the data you want for modeling. All selected data is also added to your new Actual Design, except that in red clusters.

Red Clusters

These contain more design points than data points. These data points are not added to your design, because the algorithm cannot choose the design points to replace, so you must manually make selections to deal with red clusters if you want to use these data points in your design. If you don't care about the Actual Design (for example, if you do not intend to collect more data) and you are just selecting data for modeling, then you can ignore red clusters. The data points in red clusters are selected for modeling. For information about the e ects of your selections, see What Will Happen to My Data and Design? on page 4-31.

The Tolerance Editor

Open the Tolerance Editor by selecting Tolerances in the context menu.

Here you can edit the tolerance for selecting data points. You can choose values for each variable to determine the size of tolerance in each dimension.

Data points within the tolerance of a design point are included in that cluster.

Data points that fall inside the tolerance of more than one design point form a single cluster containing all those design and data points.

Excluded data (shown as black crosses) that lies within tolerance appears in the list when that cluster is selected. You can then choose whether to use it or continue to exclude it.

Data in Design (pink crosses) is the only type of data that is not included in clusters.

Note Tolerances are set for global variables. Data used for matching uses test means of global variables, not individual records, unlike other Data Editor views. Click points to inspect values of global variables.

Using the Tolerance Editor is the same process as setting tolerances within the Data Wizard. In the Data Wizard you can also choose in advance what to do with unmatched data and clusters with uneven numbers of data and design points. These choices a ect how the cluster algorithm is rst run; you can always change selections later in the Data Editor. See Step 4: Set Tolerances on page 4-25.

Note If you modify the data in any way while the Design Match view is open (e.g., by applying a lter) the cluster algorithm will be rerun. You might lose your design point selections.

See the next section, What Will Happen to My Data and Design? on page 4-31, for information about what happens to your data set and design when you close the Data Editor after data selection in the Design Match view.

What Will Happen to My Data and Design?

As with everywhere else in the Data Editor, the changes you make are only applied to the data set when you exit. When you close the Data Editor, your choices in the Design Match plot are applied to the data set and a new design called `Actual Design` is created. All the changes are determined by your check box selections for data and design points.

Note All data points with a selected check box are selected for modeling. All data points with a cleared check box are excluded from the data set.

All data points with a selected check box are put into the new `Actual Design` except those in red clusters. See below.

When you close the Data Editor, these changes are applied:

Green clusters equal number of data and design points

The design points are replaced by the equal number of data points. These points become xed design points (red in the Design Editor table) and appear as Data in Design (pink crosses) when you reopen the Data Editor.

This means that these points are not included in clusters when matching again. These xed points are also not changed in the Design Editor when you add points, although you can unlock xed points if you want. This can be very useful if you want to optimally augment a design, taking into account the data you have already obtained.

Blue clusters more data than design points

The design points are replaced by all the data points.

Note Design points with selected check boxes in green or blue clusters are the points that will be replaced by your selected data points. You may have cleared the check boxes of other design points in these clusters, and these points will be left unchanged.

Red Clusters more design than data points

Red clusters indicate that you should make a decision if you want your new Actual Design to re ect your most current data. The algorithm cannot choose the design points to replace with the data points, so no action is taken. Red clusters do not make any changes to the design when you close the data editor. The existing design points remain in the design. The data points are included or excluded from the data set depending on your selections in the Cluster Information list, but they are not added to the design.

Unmatched Design Points

These remain in the design.

Unmatched Data Points

If you have selected the check boxes for unmatched data, they become new xed design points, which are red in the Design Editor. When you reopen the Data Editor these points are Data in Design, which appear as pink crosses. Note that in the Data Wizard you could choose Use to select all these initially, or you could choose Do not use, which clears all their check boxes. See Step 4: Set Tolerances on page 4-25.

Data in Design

These remain in the design.

Excluded Data

These data points are removed from the data set and are not displayed in any other views. If you want to return them to the data set you can only do so by selecting them in the Design Match view.

Tip For a step-by-step guide to matching data to a design using an example project, see Match Data to Designs on page 4-28 in the Getting Started documentation.

Data Loading Application Programming Interface

Data Loading API Specification

You can use the data loading API (application programming interface) to write your own data loading function, plug these functions into the toolbox, and subsequently use data loaded by these functions within the toolbox. To allow this, there are several stages that need to be followed as described below. For an example, see `xregReadConcerto.m` (in the `mbctools` directory).

Data Function Prototype

A function to successfully load data has the following prototype:

```
[OK, msg, out] = dataLoadingFcn(filename, protoOut)
```

Input Arguments

`filename` is the full path to the file to be loaded.

`protoOut` is an empty structure with the fields expected in the return argument `out`. This allows the data loading API to be easily extended without the need for data loading functions to change when the toolbox changes.

Output Arguments

The first return argument, `OK`, allows the function to signal that it has successfully loaded the data file. A value of 1 signals success, and 0 failure. If the function fails, it can return a message, `msg`, to indicate the reason for the failure. This message is displayed in a warning dialog box if the function fails. If the function is successful, the return argument `out` contains the data necessary for the toolbox.

`out.varNames` is a cell array of strings that hold the names of the variables in the data (`1 x n` or `n x 1`).

`out.varUnits` is a cell array of strings that hold the units associated with the variables in `varNames` (`1 x n` or `n x 1`). This array can be empty, in which case no units are defined.

`out.data` is an array that holds the values of the variables (`m x n`).

`out.comment` is an optional string holding comment information about the data.

Data Function Check In

Once you have written the function, you need to check it into the toolbox, using the `mbccheckindataloadingfcn` function. This function has the following prototype:

```
OK= mbccheckindataloadingfcn(fun, filterSpec, fileType, filename)
```

`fun` is a string that is the function to call to load the data. This function must be on the MATLAB path.

`filterSpec` is a `1 x 2` element cell array that contains the extensions that this function loads and the descriptions of those files. This cell array is used in the `uigetfile` function, for example, `{'*.m;*.fig;*.mat;', 'All MATLAB Files'}` or `{'*.m', 'M-files (*.m)'}`. MBC attempts to decide automatically which sort of file is loaded, based on the extension. In the case of duplicate extensions, the first in the list is selected; however, it is always possible to override the automatic selection with a user selection. You will see a warning message if there is any ambiguity.

`fileType` is a string that describes the file type, for example, `'MATLAB file'` or `'Excel file'`.

See Also

More About

- Automate Design and Modeling With Scripts

Setting Up Models

This section discusses the following topics:

What Models Are Available? on page 5-2
Explore Local Model Types on page 5-7
Assess Boundary Models on page 5-42
Explore Boundary Model Types on page 5-46
Explore Global Model Types on page 5-62
Add Response Models and Datum Models on page 5-81
Create Alternative Models to Compare on page 5-84

What Models Are Available?

In this section...

What Is a One-Stage Model? on page 5-2
What Is a Two-Stage Model? on page 5-2
What Is a Point-by-Point Model? on page 5-2
Default Model Types on page 5-3
Model Types on page 5-4

What Is a One-Stage Model?

A one-stage model fits a model to all the data in one process. If your data inputs do not have a hierarchical structure, and all model inputs are global at the same level, then fit a one-stage model.

If your data has local and global inputs, where some variables are fixed while varying others, then choose a two-stage or point-by-point model instead.

What Is a Two-Stage Model?

A two-stage model fits a model to data with a hierarchical structure. If your data has local and global inputs, where some variables are fixed while varying others, then choose a two-stage model. For example, data collected in the form of spark sweeps is suited to a two-stage model. Each test sweeps a range of spark angles, with fixed engine speed, load, and air/fuel ratio within each test.

If your data inputs do not have a hierarchical structure, and all model inputs are global, at the same level, then fit a one-stage model instead.

For two-stage models, only specify a single local variable. If you want more local inputs, use a one-stage or point-by-point model instead.

What Is a Point-by-Point Model?

Point-by-point modeling allows you to build a model at each operating point of an engine with the necessary accuracy to produce an optimal calibration. You often need point-by-point models for multiple injection diesel engines and gasoline direct-injection engines.

With point-by-point models, no predictions are available between operating points. If you need predictions between operating points, use a one-stage model instead.

Default Model Types

Model Type	Default Model Fits	Large Data Settings
One-stage	Response model: Gaussian process model (GPM)	For >2000 points, uses the large data behavior for Gaussian process models from Statistics and Machine Learning Toolbox.
	Boundary model: Convex Hull t to the inputs	For >2000 points, switches to pairwise convex hull (one for every pair of inputs). Switch when 8 inputs even when <2000 points.
Two-stage	Local model: Quadratic Global model: Hybrid radial-basis function (RBF)	For >2000 tests, global model switches to quadratic.
	Boundary model: Convex Hull t to the global inputs, and a two-stage boundary model for the local input.	For >2000 tests, global boundary model switches to pairwise convex hull. Switch when 8 inputs even when <2000 points.

Model Type	Default Model Fits	Large Data Settings
Point-by-point	The toolbox ts these model types to each operating point and selects the best model: Quadratic with Stepwise: Min PRESS Cubic with Stepwise: Min PRESS Hybrid RBF with nObs/3 Gaussian process models (using defaults)	For any operating point >2000 Points or >100 operating points, switches to tting a single GPM per operating point (no Hybrid RBF or polynomial).
	Boundary model: Point-by-point boundary model with a single Convex Hull t to all inputs at each operating point.	If any operating point has >2000 points, then point-by-point boundary model switches to a pairwise convex hull. Switch when 8 inputs even when <2000 points.

If you are using a template that you created, you can override the default models when you t the model. On the Fit Models dialog box, clear the Use default models for large data option.

Model Types

The following table shows the model types available for one-stage and two-stage modeling.

Model Type	One-Stage and Two-Stage Global	Two-Stage: Local
Linear model	Yes	Yes
Radial basis function (RBF)	Yes	
Hybrid RBF	Yes	

Model Type	One-Stage and Two-Stage Global	Two-Stage: Local
Interpolating RBF	Yes	
Multiple linear models	Yes	
Free knot spline	Yes, one factor only	Yes, one factor only
Neural net (requires Deep Learning Toolbox software)	Yes	
Average t		Yes
Point-by-point models*		Yes
Growth models		Yes, one factor only
Polynomial**		Yes, one factor only
Polynomial spline**		Yes, one factor only
Truncated power series		Yes, one factor only
User de ned#	Yes (the example is one factor only)	Yes
Transient#	Yes (the example is two factors only)	Yes

*Point-by-point models give you access to global model types for your local model.

** Polynomial and polynomial spline are two special case linear models for local models with one input factor. You can use polynomial and polynomial spline models (with more settings) for local models with more factors by choosing Linear Models.

User de ned and transient models must be checked into the toolbox before you can use them. They are available only for the number of factors you speci ed. There is an example user-de ned model for a single factor preregistered with the toolbox. The example transient model provided must have exactly two factors, one of which must be time.

See Also

Related Examples

Fit a One-Stage Model on page 1-7
Fit a Two-Stage Model on page 1-10
Fit a Point-by-Point Model on page 1-14

More About

Explore Local Model Types on page 5-7
Explore Global Model Types on page 5-62
Explore Local Model Types on page 5-7
Local Model Class: User-De ned Models on page 5-23
Local Model Class: Transient Models on page 5-30

Explore Local Model Types

In this section...

Alternative Local Model Types on page 5-7
Local Model Class: Polynomials and Polynomial Splines on page 5-8
Local Model Class: Linear Models on page 5-11
Local Model Class: Truncated Power Series on page 5-13
Local Model Class: Free Knot Spline on page 5-14
Local Model Class: Growth Models on page 5-16
Local Model Class: User-De ned Models on page 5-23
Local Model Class: Transient Models on page 5-30
Local Model Class: Average Fit on page 5-37
Transforms on page 5-38
Covariance Modeling on page 5-39
Correlation Models on page 5-40

Alternative Local Model Types

First, try tting the defaults using the Fit models common task button.

If you want to try alternative local model types, select the response node, then in the Common Tasks pane, click New Local Model. This opens the Local Model Setup dialog box. Browse the model types on this page.

To examine ts, see Assess Local Models on page 6-5.

The available models depend on the number of input factors. Polynomial, polynomial spline, truncated power series, free knot spline, and growth models are for one factor only.

You can choose additional response features at this stage using the Response Features tab of the Local Model Setup dialog box. These can also be added later. The Model Browser chooses su icient response features for the current model.

See each local model type for statistical details and available response features.

See also

Covariance Modeling on page 5-39

Correlation Models on page 5-40

Transforms on page 5-38

Local Model Class: Polynomials and Polynomial Splines

Polynomials

At the local level, if you have one input factor, you can choose Polynomial directly from the list of local model classes. Here you can choose the order of polynomials used, and you can de ne a datum model for this kind of local model.

If there is more than one input factor, you can choose Linear Models from the Local Model Class list, then you can choose Polynomial or Hybrid Spline. This is a di erent polynomial model where you can change more settings such as Stepwise, the Term Editor (where you can remove any model terms) and you can choose di erent orders for di erent factors (as with the global level polynomial models). See Local Model Class: Linear Models on page 5-11.

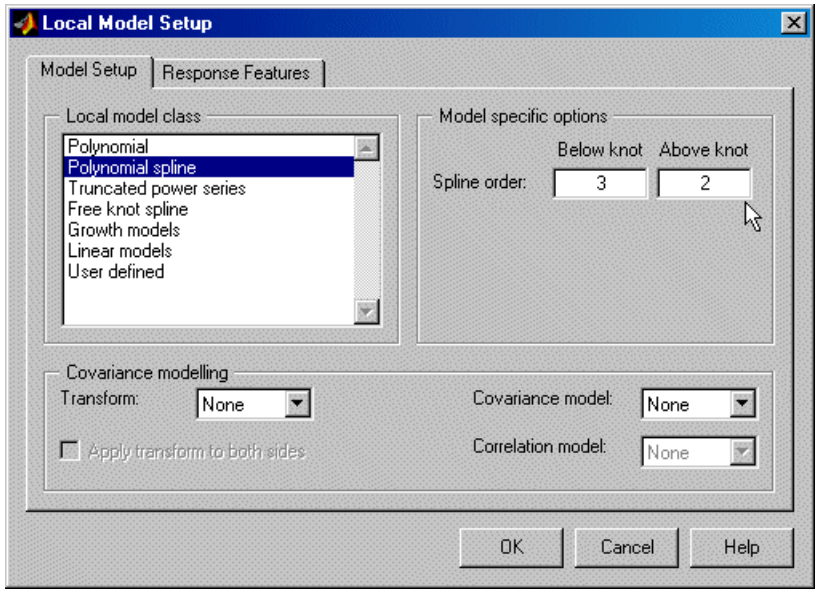
Di erent response features are available for this polynomial model and the Linear Models: Polynomial choice. You can view these by clicking the Response Features tab on the Local Model Setup dialog box. Single input polynomials can have a datum model, and you can de ne response features relative to the datum. See Datum Models on page 5-81.

The following response features are permitted for the polynomial model class:

- Location of the maximum or minimum value.
- Value of the t function at a user-speci ed x-ordinate. When datum models are used, the value is relative to the datum (for example, mbt - x).
- The n^{th} derivative at a user-speci ed x-ordinate, for n = 1, 2, ..., d where d is the degree of the polynomial.

See the global model section Polynomials on page 5-63 for a general description of polynomial models.

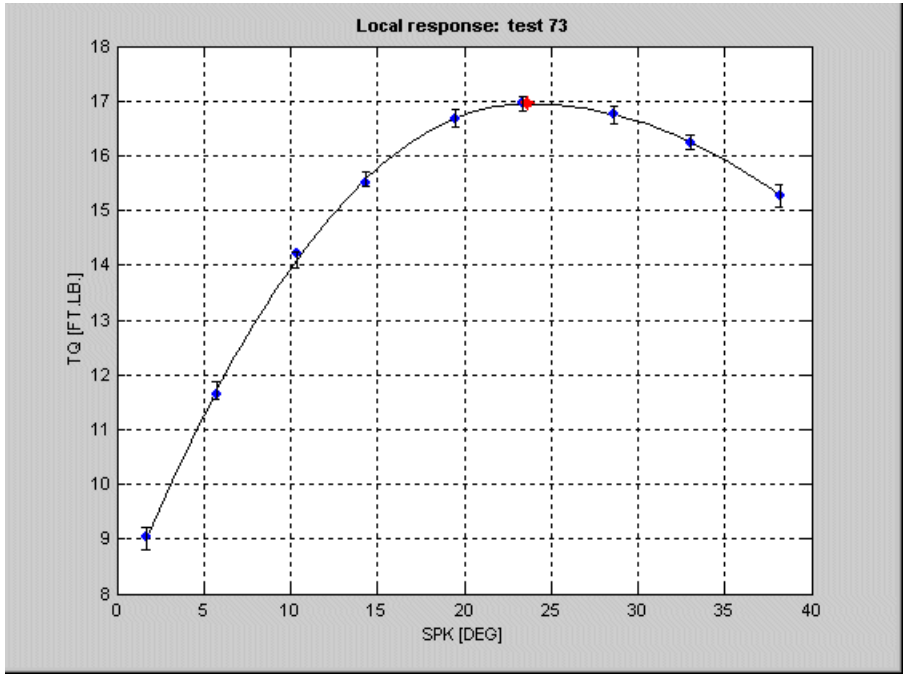
Polynomial Spline



A spline is a piecewise polynomial, where di erent sections of polynomial are tted smoothly together. The location of each break is called a knot. Polynomial splines are essential for modeling torque/spark curves.

This model has only one knot. You can choose the orders of the polynomials above and below the knot. See also Hybrid Splines on page 5-66. These global models also use splines, but use the same order polynomial throughout.

Polynomial splines are only available for single input factors. The following example shows a typical torque/spark curve, which requires a spline to t properly. The knot is shown as a red spot at the maximum, and the curvature above and below the knot is di erent. In this case, there is a cubic basis function below the knot and a quadratic above.



To model responses that are characterized in appearance by a single and well-de ned stationary point with asymmetrical curvature either side of the local minimum or maximum, de ne the following spline class,

$$y_{ij} = \beta_o + \sum_{a=2}^{c} \beta_{Low_a} (x_j - k)_-^a + \sum_{b=2}^{h} \beta_{High_b} (x_j - k)_+^b$$

where k is the knot location, β denotes a regression coe icient,

$$(x_j - k)_- = \min\{0, (x_j - k)\}, \quad (x_j - k)_+ = \max\{0, (x_j - k)\}$$

where c is the user-speci ed degree for the left polynomial, h is the user-speci ed degree for the right polynomial, and the subscripts Low and High denote to the left (below) and right of (above) the knot, respectively.

Excluding terms in $(x_j - k)_-$ and $(x_j - k)_+$ ensures that the first derivative at the knot position is continuous. In addition, by definition the constant β_0 must be equal to the value of the fit function at the knot, that is, the value at the stationary point.

For this model class, response features can be chosen as

Fit constants $\{\beta_0, \beta_{Low_2}, \ldots, \beta_{Low_p}, \beta_{High_2}, \beta_{High_q}\}$

Knot position $\{k\}$

Value of the fit function at a user-specified delta

$$\{\Delta a_j = x_j - k\}$$

from the knot position $\{f(\pm \Delta a_j)\}$ if the datum is defined, otherwise the value is absolute.

Difference between the value of the fit function at a user-specified delta from the knot position and the value of the fit function at the knot

$$\{f(\pm \Delta a_j) - f(k)\}$$

Local Model Class: Linear Models

Select `Linear Models` and then click `Setup`.

You can now set up polynomial or hybrid spline models. The settings are exactly the same as the global linear models.

These models are for multiple input factors - for single input factors you can use a different polynomial model from the Local Model Class list, where you can only change the polynomial order. See Local Model Class: Polynomials and Polynomial Splines on page 5-8.

If there is more than one input factor, you can choose Linear Models from the Local Model Class list, then you can choose Polynomial or Hybrid Spline. This polynomial is a different model where you can change more settings such as Stepwise, the Term Editor

(where you can remove any model terms) and you can choose di erent orders for di erent factors (as with the global level polynomial models).

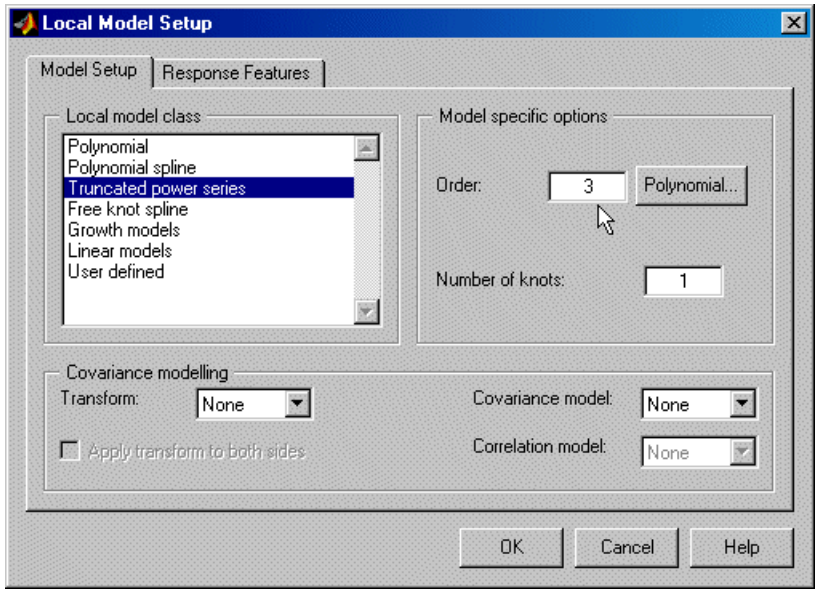
See Global Linear Models: Polynomials and Hybrid Splines on page 5-63 for details.

By default inputs are transformed to [-1, 1] before tting and evaluating polynomials. This is important as di erences in scales between inputs can cause numerical problems. Generally it is a good idea to transform inputs as this alleviates problems with variables of di erent scales. In some circumstances, you might be concerned with the values of polynomials (for example if your strategy requires raw polynomial coe icients). In this case, you can clear the Transform input range check box to use untransformed units to calculate natural polynomials.

Di erent response features are available for this Linear Models: Polynomial model and the other Polynomial choice (for single input factors). You can view these by clicking the Response Features tab on the Local Model Setup dialog box. Single input polynomials can have a datum model, and you can de ne response features relative to the datum. See Datum Models on page 5-81.

These linear models are labeled `Quadratic`, `Cubic`, and so on, on the test plan block diagram. The single input type of polynomials are labeled `Poly2`, `Poly3`, and so on. For higher orders, both types are labeled `Poly` n.

Local Model Class: Truncated Power Series



This is only available for a single input factor.

You can choose the order of the polynomial basis function and the number of knots for Truncated Power Series Basis Splines. A spline is a piecewise polynomial, where di erent sections of polynomial are tted smoothly together. The point of each break is called a knot. The truncated power series changes the coe icient for the highest power when the input passes above the knot value.

It is truncated because the power series is an approximation to an in nite sum of polynomial terms. You can use in nite sums to approximate arbitrary functions, but clearly it is not feasible to t all the coe icients.

Click Polynomial to see (and remove, if you want) the polynomial terms. One use of the remove polynomial term function is to make the function linear until the knot, and then quadratic above the knot. In this case, remove the quadratic coe icient.

See also

Polynomial Spline on page 5-9, where you can choose di erent order basis functions either side of the knot

Hybrid Splines on page 5-66, a global model where you can choose a spline order for one of the factors (the rest have polynomials tted)

Local Model Class: Free Knot Spline on page 5-14, free knot splines, where you can choose the number of knots and the order of the basis functions

Truncated Power Series Basis (TPSBS) Splines

A general class of spline functions with arbitrary (but strictly increasing) knot sequence:

$$\mathbf{k} = \{k_1, k_2, \ldots, k_k\}^T : \quad f(x) = \sum_{i=0}^{m} \beta_i x^i + \sum_{i=0}^{k} \beta_{m+i}(x - k_i)_+^m$$

This de nes a spline of order m with knot sequence $\mathbf{k} = \{k_1, k_2, \ldots, k_k\}^T$

For this model class, response features can be chosen as

Fit constants $\{\beta_0, \beta_1, \ldots, \beta_{m-1+k}\}$

Knot position vector $\{\mathbf{k}\}$.

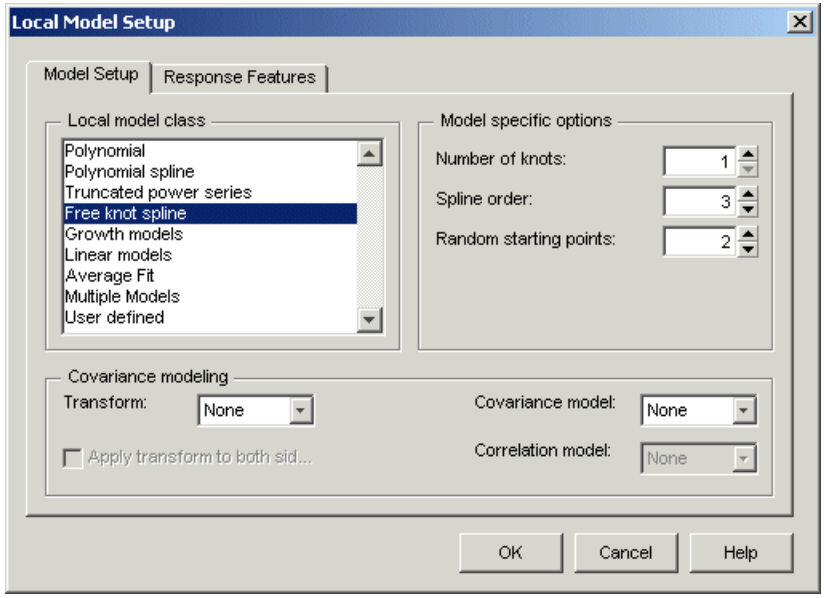
Value of the t function $\{f(a_j)\}$ at a user-speci ed value $\{a_j\}$

Value of the nth derivative of the t function with respect to x_j $\{f(a_j)^n\}$ at a user-speci ed value $\{a_j\}$, with n = 1, 2, ..., m-2

You can remove any of the polynomial terms from the model.

Local Model Class: Free Knot Spline

These are the same as the Global Model Class: Free Knot Spline on page 5-77 (which is also only available for one input factor). See the global free knot splines for an example curve shape.



A spline is a piecewise polynomial, where di erent sections of polynomial are tted smoothly together. The point of the join is called the knot.

You can choose the number of knots. You can choose the order of polynomial tted (in all curve sections) from 1 to 3. The default is cubic.

You can set the number of Random starting points. These are the number of initial guesses at the knot positions.

See also

- Polynomial Spline on page 5-9, where you can choose di erent order basis functions for either side of the knot
- Local Model Class: Truncated Power Series on page 5-13, where you can choose the order of the basis function
- Hybrid Splines on page 5-66, a global model where you can choose a spline order for one of the factors (the rest have polynomials tted)

Free Knot Splines

The $(x_j - k_i)_+^{m-1}$ basis is not the best suited for the purposes of estimation and evaluation, as the design matrix might be poorly conditioned. In addition, the number of arithmetic operations required to evaluate the spline function depends on the location of x_i relative to the knots. These properties can lead to numeric inaccuracies, especially when the number of knots is large. You can reduce such problems by employing B-splines.

The most important aspect is that for moderate m the design matrix expressed in terms of B-splines is relatively well conditioned.

For this model class, response features can be chosen as

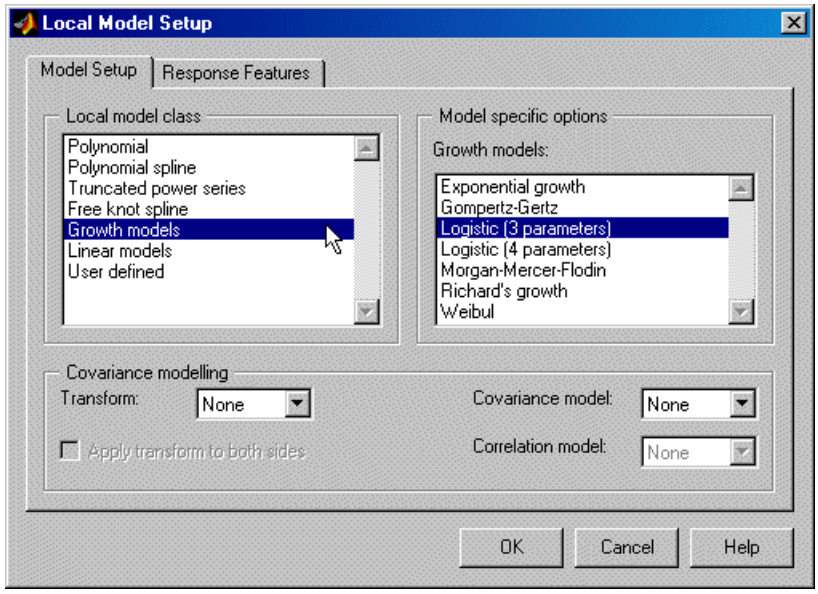
Fit constants $\{\beta_{-(m-1)}, \beta_{-m}, \ldots, \beta_k\}$

Knot position vector $\{k\}$.

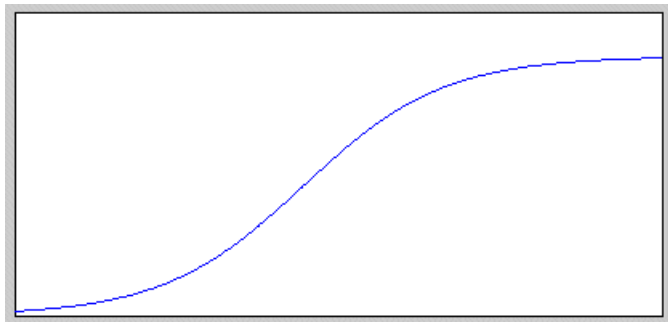
Value of the t function $\{f(a_j)\}$ at a user speci ed value $\{a_j\}$

Local Model Class: Growth Models

Growth models have several varieties available. They are only available for single input factors.



These are all varieties of sigmoidal curves between two asymptotes, like the following example.



Growth models are often the most appropriate curve shape for air charge engine modeling.

See the following sections for mathematical details on the di erences between these growth models:

Three Parameter Logistic Model on page 5-18

Morgan-Mercer-Flodin Model on page 5-18
Four-Parameter Logistic Curve on page 5-19
Richards Curves on page 5-20
Weibul Growth Curve on page 5-21
Exponential Growth Curve on page 5-22
Gompertz Growth Model on page 5-22

Three Parameter Logistic Model

The equation $y_j = \frac{\alpha}{1+\exp[-\kappa(x_j-\gamma)]^r}$ de nes the three parameter logistic curve.

where α is the nal size achieved, K is a scale parameter, and γ is the x-ordinate of the point of in ection of the curve.

The curve has asymptotes $y_j = 0$ as $x_j \rightarrow$ - and $y_j = \alpha$ as $x_j \rightarrow$. Growth rate is at a maximum when $y_j =$ /2, which occurs when $x_j = \gamma$. Maximum growth rate corresponds to

$$\frac{\kappa\alpha}{4}$$

The following constraints apply to the t coe icients:

> 0, $K > 0$, $\gamma > 0$.

The response feature vector g for the three parameter logistic function is de ned as

$$g = \left[\alpha \quad \gamma \quad \kappa \quad \frac{\kappa\alpha}{4}\right]^T$$

Morgan-Mercer-Flodin Model

This equation de nes the Morgan-Mercer-Flodin (MMF) growth model.

$$y_j = \alpha - \frac{\alpha-\beta}{\left(1+\left(\kappa x_j\right)^{\delta}\right)}$$

where is the value of the upper asymptote, is the value of the lower asymptote, K is a scaling parameter, and is a parameter that controls the location of the point of in ection for the curve. The point of in ection is located at

$$x = \left[\frac{\delta - 1}{\delta + 1}\right]^{1/\delta}$$

$$y = \frac{\delta - 2}{2\delta}$$

for $\delta \geq 1$

There is no point of in ection for < 1. All the MMF curves are sublogistic, in the sense that the point of in ection is always located below 50% growth (0.5). The following constraints apply to the t coe icient values:

> 0, > 0, K > 0, > 0

>

This equation provides the response feature vector g.

$$g = \left[\alpha \quad \beta \quad \kappa \quad \delta \quad \frac{\delta - 1}{2\delta}\right]^{T}$$

Four-Parameter Logistic Curve

This equation de nes the four-parameter logistic model.

$$\beta - \frac{(\alpha - \beta)}{\left(1 + \frac{e^{\gamma\kappa}}{x^{\kappa}}\right)}$$

with constraints $\alpha, \beta, \kappa, \gamma > 0$, $\beta < \alpha$ and $\beta < \gamma < \alpha$. Again, α is the value of the upper asymptote, κ is a scaling factor, and γ is a factor that locates the x-ordinate of the point of in ection at

$$\exp\left(\frac{\kappa\gamma - \log\left(\frac{1+\kappa}{\kappa-1}\right)}{k}\right)$$

The following constraints apply to the t coe icient values:

All parameters > 0

>

This is the available response feature vector:

$$g = \left[\alpha \quad \beta \quad \kappa \quad \gamma \quad \frac{\left(\kappa\gamma - \log\left(\frac{1+\kappa}{\kappa-1}\right)\right)}{\kappa}\right]^T$$

Richards Curves

This equation de nes the Richards curves family of growth models.

$$y_j = \alpha\left[1 + (\delta - 1)\exp\left(-\kappa\left(x_j - \gamma\right)\right)\right]^{1/(1-\delta)} \qquad 1$$

where is the upper asymptote, γ is the location of the point of in ection on the x axis, K is a scaling factor, and is a parameter that indirectly locates the point of in ection.

The y-ordinate of the point of in ection is determined from

$$\frac{\alpha}{\delta^{1/(\delta-1)}} \quad \delta > 0$$

Richards also derived the average normalized growth rate for the curve as

$$\frac{\kappa}{2(\delta+1)}$$

The following constraints apply to the t coe icient values:

> 0, γ > 0, K > 0, > 0

> γ

1

Finally, the response feature vector g for Richards family of growth curves is de ned as

$$g = \left[\alpha \quad \gamma \quad \kappa \quad \frac{\kappa}{2(\delta+1)} \right]^T$$

Weibul Growth Curve

This equation de nes the Weibul growth curve.

$$y_j = \alpha - (\alpha - \beta)\exp\left(- (\kappa x_j)^{\delta} \right)$$

where α is the value of the upper curve asymptote, β is the value of the lower curve asymptote, κ is a scaling parameter, and δ is a parameter that controls the x-ordinate for the point of in ection for the curve at

$$\left(\frac{1}{\kappa}\right)\left(\frac{\delta-1}{\delta}\right)^{1/\delta}$$

The following constraints apply to the curve t parameters:

> 0, > 0, K > 0, > 0

>

The associated response feature vector is

$$g = \left[\alpha \quad \beta \quad \kappa \quad \delta \quad \left(\frac{1}{\kappa}\right)\left(\frac{\delta-1}{\delta}\right)^{1/\delta} \right]^T$$

Exponential Growth Curve

This equation de nes the exponential growth model.

$$y_j = \alpha - (\alpha - \beta)\exp(-\kappa x_j)$$

where is the value of the upper asymptote, is the initial size, and K is a scale parameter (time constant controlling the growth rate). The following constraints apply to the t coe icients:

> 0, > 0, K > 0

>

The response feature vector **g** for the exponential growth model is de ned as

$$\mathbf{g} = [\alpha \quad \beta \quad \kappa]^T$$

Gompertz Growth Model

Another useful formulation that does not exhibit a symmetric point of in ection is the Gompertz growth model. The de ning equation is

$$y_j = \alpha\exp\left(-e^{-\kappa(x_j-\gamma)}\right)$$

where is the nal size, K is a scaling factor, and γ is the x-ordinate of the point of in ection. The corresponding y-ordinate of the point of in ection occurs at

$$\frac{\alpha}{e}$$

With maximum growth rate

$$\frac{\kappa\alpha}{e}$$

The following constraints apply to the selection of parameter values for the Gompertz model:

> 0, K > 0, γ > 0.

The response feature vector g for the Gompertz growth model is de ned as

$$g = \left[\alpha \quad \gamma \quad \kappa \quad \frac{\kappa\alpha}{e} \right]^{T}$$

Local Model Class: User-De ned Models

You can create user-de ned models to use for local, global, or one-stage models.

The user-de ned model type allows you to de ne your equation in a le to use for tting in the toolbox.

To create and check in a user-de ned model:

1 Copy the user-de ned template le (`functemplate.m`) to a location on the MATLAB path (but not under *`matlabroot`*`\toolbox\mbc`). Find the le at this location:

   `<`*`matlabroot`*`>\toolbox\mbc\mbcmodels\@xregusermod\functemplate.m`

2 Modify the copy of the template le for your model. Code all the compulsory functions. The template comments describe all the compulsory and optional functions.

   It is not compulsory to modify every optional section. If you do not wish to use an optional section, leave it unmodi ed.

3 Check your model into the toolbox with this command:

```
m = checkin(xregusermod,'MfileName',TestInputData)
```

Note Your user-de ned models can have as many factors as you choose.

After you check in your user-de ned models, they appear as options on the Model Setup dialog box when you are using data with the right number of factors.

To remove your checked-in model from the toolbox, enter:

```
remove(xregusermod,'MfileName')
```

Note If any of your global models are user-de ned you cannot use MLE (maximum likelihood estimation) for your two-stage model.

You can examine this example user-de ned model le as a guide:

<MATLAB root>\toolbox\mbc\mbcmodels\@xregusermod\weibul.m

This le de nes the local model Weibul (Growth Model) in the toolbox and hence should not be changed. Make sure that you do not overwrite the le. If you alter the le, save it under another name.

The following sections explain how to examine and check in the Weibul model:

Examine the Example User-De ned Model on page 5-24
Check the Model into the Toolbox on page 5-29
Verify That the Model Is Checked In on page 5-29
Recover from Editing Errors on page 5-30

Examine the Example User-De ned Model

This example demonstrates how to create a user-de ned model for the Weibul function:

```
y = alpha - (alpha - beta).*exp(-(kappa.*x).^delta)
```

1 Open the following example le

```
<MATLAB root>\toolbox\mbc\mbcmodels\@xregusermod\weibul.m
```

Observe that the le is called by the toolbox using

```
varargout= weibul(m,x,varargin)
```

Where the variables are given by

```
m = the xregusermod object
x = input data as a column vector for fast eval
```

The rst function in the template le is a vectorized evaluation of the function. First, the model parameters are extracted:

```
b= double(m);
```

Then, the evaluation occurs:

```
y = b(1) - (b(1)-b(2)).*exp(-(b(3).*x).^b(4));
```

Note The parameters are always referred to and declared in the same order.

Compulsory Local Functions

Edit these local functions as follows:

Enter the number of input factors. For functions y = f(x) this is 1.

```
function n= i_nfactors(U,b)
n= 1;
```

Enter the number of tted parameters. In this example, there are four parameters in the Weibul model.

```
function n= i_numparams(U,b)
n= 4;
```

The following local function returns a column vector of initial values (`param`) for the parameters to t. The initial values can be de ned to be data-dependent; hence there is a ag to signal if the data is not worth tting (`OK`). In `weibul.m` there is a routine for calculating a data-dependent initial parameter estimate. If no data is supplied, the default parameter values are used.

```
function [param,OK]= i_initial(U,b,X,Y)
param= [2 1 2 5]';
OK=1;
```

Optional Local Functions

You can use the following optional local functions to set additional parameters

1 You can state lower and upper bounds for the model parameters. These bounds appear in the same order that the parameters are declared and used throughout the template le.

   ```
   function [LB,UB,A,c,nlcon,optparams]=i_constraints(U,b,varargin)
   LB=[eps eps eps eps]';
   UB=[1e10 1e10 1e10 1e10]';
   ```

   You can also de ne linear constraints on the parameters. This code produces the constraint (alpha+beta) is less than or equal to zero:

```
A= [-1 1 0 0];
c= [0];
```

`nlcon` defines the number of nonlinear constraints (here declared to be zero). If the number of nonlinear constraints is not zero, the nonlinear constraints are calculated in `i_nlconstraints`.

```
nlcon= 0;
```

`optparams` defines any optional parameters. No optional parameters are declared for the cost function.

```
optparams= [];
```

2 `i_foptions` defines the fit options. The fit options are always based on the input `fopts`. See MATLAB help on the function `optimset` for more information on fit options. When there are no constraints, the toolbox uses the MATLAB function `lsqnonlin` for fitting, otherwise `fmincon` is used.

```
function fopts= i_foptions(U,b,fopts)

    fopts= optimset(fopts,'Display','none');
```

3 `i_jacobian` can supply an analytic Jacobian to speed up the fitting algorithm.

```
function J= i_jacobian(U,b,x)

    x = x(:);
    J= zeros(length(x),4);

    a=b(1); beta=b(2); k=b(3); d=b(4);

    ekd= exp(-(k.*x).^d);
    j2= (a-beta).*(k.*x).^d.*ekd;

    J(:,1)= 1-ekd;
    J(:,2)= ekd;
    J(:,3)= j2.*d./k;
    J(:,4)= j2.*log(k.*x);
```

4 `i_labels` defines the labels used on plots. You can use LaTeX notation and it is formatted.

```
function c= i_labels(U,b)
c={'\alpha','\beta','\kappa','\delta'};
```

5 i_char is the display equation string and can contain LaTeX expressions. The current values of model parameters appear.

```
function str= i_char(U,b)

s= get(U,'symbol');
str=sprintf('%.3g - (%.3g-%.3g)*exp(-(%.3g*x)^{%.3g})',...
        b([1 1 2 3]), detex(s{1}), b(4));
```

6 i_str_func displays the function definition with labels appearing in place of the parameters (not numerical values).

```
function str= i_str_func(U,b, TeX)

s= get(U,'symbol');
if nargin == 2 || TeX
    s = detex(s)
end
% This can contain TeX expressions supported by HG text objects
lab= labels(U);
str= sprintf('%s - (%s - %s)*exp(-(%s*%s)^{%s})',...
        lab{1},lab{1},lab{2},lab{3},s{1},lab{4});
```

7 i_rfnames defines response feature names. This example shows a defined response feature that is not one of the parameters (in this case it is also nonlinear).

You do not need to define rname (it can return an empty cell array).

```
function [rname, default] = i_rfnames(U,b)
% response feature names
rname= {'inflex'};

if nargout>1
    default = [1 2 5 4]; %{'Alpha','Beta','Inflex','Delta'};
end
```

8 i_rfvals defines response feature values. This example defines the response feature labeled as INFLEX in previous step. The Jacobian matrix is also defined here as dG.

```
function [rf,dG]= i_rfvals(U,b)
%I_RECONSTRUCT nonlinear reconstruction
%
% p = i_reconstruct(m,b,Yrf,dG,rfuser)
%  Inputs
%     Yrf   response feature values
```

```
%     dG    Jacobian of response features with respect to
%              parameters
%     rfuser index to the user-defined response features
%              so you can find out which response features
%              are which. rfuser(i) = 0 if the rf is a parameter.
%
% If all response features are linear in model parameters
%  then you do not need to define 'i_reconstruct'.

% this is an example of how to implement a nonlinear response
% feature definition

K = b(3);
D = b(4);
if D>=1
   rf= (1/K)*((D-1)/D)^(1/D);
else
   rf= NaN;
end

if nargout>1
   % Jacobian of response features with respect to model
   %  parameters
   if D>=1
      dG= [0, 0, -((D-1)/D)^(1/D)/K^2,...
            1/K*((D-1)/D)^(1/D)*(-1/D^2*log((D-1)/D)+...
            (1/D-(D-1)/D^2)/(D-1))];
   else
      dG= [0, 0, 1 1];
   end
end
```

9 The `i_reconstruct` local function allows the model parameters to be reconstructed from the response features that have been given. If all response features are linear in the parameters, then you do not need to de ne this function. The code rst identi es which response features (if any) are user de ned.

```
function p= i_reconstruct(U,b,Yrf,dG,rfuser)
%I_RECONSTRUCT nonlinear reconstruction
%
% p = i_reconstruct(m,b,Yrf,dG,rfuser)
%  Inputs
%     Yrf   response feature values
%     dG    default Jacobian of response features
%     rfuser index to the user-defined response features so you can find
%     out which response features are which. rfuser(i) = 0 if the
%     rf is a parameter.
```

```
%
% If all response features are linear in model parameters then you do not
% need to define 'i_reconstruct'.

% reconstruct linear response features using this line
p= Yrf/dG';

% find which response feature is a nonlinear user-defined response feature
f= rfuser>size(p,2);
if ~any(rfuser==3)
   % need to use delta (must be > 1) for reconstruction to work
   p(:,4)= max(p(:,4),1+16*eps);

   p(:,3)= ((p(:,4)-1)./p(:,4)).^(1./p(:,4))./Yrf(:,f);
end
```

Check the Model into the Toolbox

After you create a user-de ned model le, save it somewhere on the path.

Check in the model. The check in process ensures that the model you have de ned provides an interface that allows the toolbox to evaluate and t it. If the checkin procedure succeeds, the toolbox registers the model and makes it available for tting when you use appropriate input data.

At the command-line, with the example le on the path, you need to create a model and some input data, then call `checkin`. For the user-de ned Weibul function, call `checkin` with the example model, its name, and some appropriate data.

```
checkin(xregusermod, 'weibul', [0.1:0.01:0.2]');
```

This returns some command-line output, and a gure appears with the model name and variable names displayed over graphs of input and model evaluation output. The nal command-line output:

```
Model successfully registered for
Model-Based Calibration Toolbox software.
```

Verify That the Model Is Checked In

To verify a user-de ned model is checked in to the toolbox:

1. Start the Model Browser and load data that has the necessary format to evaluate your user-de ned model.
2. Set up a Test Plan with N Stage1 input factors, where N is the number of input factors de ned in `i_nfactors`.

3 Open the Local Model Setup dialog box, and select User-de ned models in the Local Model Class list. Your user-de ned model should appear in the right list.

To verify that the `weibul` example user-de ned model is checked in:

1 Open `gasoline_project.mat`, and select the PS22 test plan node in the tree.
2 Double-click the Local Model icon in the test plan diagram.
3 In the Local Model Setup dialog box, select User-de ned models in the Local Model Class list. The `Weibul` user-de ned model appears in the right list, as checked in.

Recover from Editing Errors

If you modify the le that de nes a checked-in model that you are currently using, the Model Browser tries to continue.

If the user-de ned model has serious errors (e.g., you have altered the number of inputs or parameters, model evaluation, or the le cannot be found), the Model Browser changes the status of the current model to 'Not tted'.

To recover from this state, rst correct the problem in the le. Then (for Local Models) you can select Model > Fit Local to re t. You do not need to check the model in again, but if you have problems the error messages produced during a checkin might be useful in diagnosing faults.

If there is a minor fault with the le, then the Model Browser continues, using default values. Minor errors can include items such as label de nitions, that is, faults in `i_char`, `i_label`, `i_str_func`, or `i_foptions`. A message appears in the command window providing details of these faults.

Local Model Class: Transient Models

Use transient models for local, global, or one-stage models. The toolbox supports transient models with multiple input factors where time is one of the factors. You can de ne a dynamic model using Simulink software and a le that describes parameters to t in this model. The toolbox provides an example called `fuelPuddle` which is already checked in to the toolbox. You can use this example for modeling. The example provided requires two input factors. The process of creating and checking in your own transient models is described in the following sections using this example.

Create Folders on page 5-31

Create Simulink Model on page 5-31
Create a User-De ned Transient Model File on page 5-33
Compulsory Local Functions on page 5-33
Optional Local Functions on page 5-34
Check In to Toolbox on page 5-35
Using and Removing Transient Models on page 5-36

Locate the example Simulink model:

<MATLAB root>\toolbox\mbc\mbcsimulink\fuelPuddle

and the example user-de ned transient model le:

<MATLAB root>\toolbox\mbc\mbcmodels\@xregtransient\fuelPuddle.m

Create Folders

You can de ne a dynamic model using a Simulink model and a user-de ned transient model le that describes parameters to t in this model. Create folders for these les that are on the MATLAB path. Then, you must check the les into the Model-Based Calibration Toolbox product before you can use them for modeling.

1 Create a folder and add it to the MATLAB path.

   For example: `D:\MyTransient`.

   Put your Simulink model in this folder. See Create Simulink Model on page 5-31.

2 Inside the new folder, create a folder called @xregtransient. (e.g., D:\MyTransient \@xregtransient). You must call this folder @xregtransient.

   Put your user-de ned transient model le in the new @xregtransient folder. This le must have the same name as your Simulink model. See Create a User-De ned Transient Model File on page 5-33.

Create Simulink Model

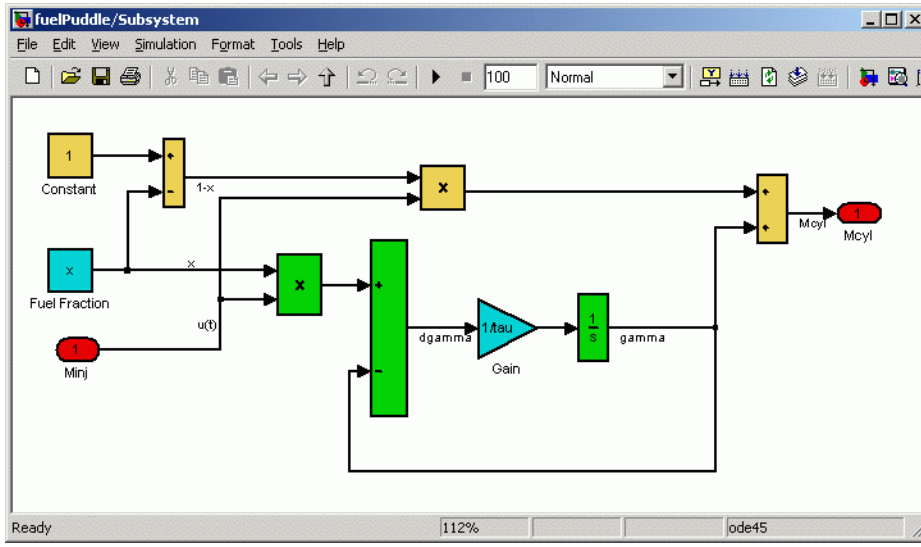
Create a Simulink model to represent your transient system.

Simulink inports represent the model inputs. Time is the rst input to the model in Model-Based Calibration Toolbox software, but it is not an inport in the model. The model output

is represented by a Simulink output. Only one scalar output is supported in the Model Browser.

Your Simulink model must have some free parameters that Model-Based Calibration Toolbox software will fit by nonlinear least squares. Use the same variable names in the model and the user-defined transient model file.

The Simulink model for the `fuelPuddle` example appears in the following figure. You will examine the example file (`fuelPuddle.m`) in a later section.



The block labels are not important. The model returns a single scalar output (here labeled `Mcyl`).

The toolbox defines the model parameters in the Model Workspace. You can see this in the Model Explorer. A vector of all parameters called `p` is also available.

Model Requirements

For both continuous and discrete models, the following rules apply:

- For Simulink models with a Fixed Step solver the step size is set to the sample time of the data.
- For Variable step solvers the simulation step size is set by the Simulink model.

You see an error if the size of the output from simulating the Simulink model does not does not match the size of the input data provided by the toolbox.

Create a User-De ned Transient Model File

De ne a transient model le for your Simulink model, and this le must have the same name as your Simulink model.

1 Copy the transient model template le, found at:

```
<MATLAB root>\toolbox\mbc\mbcmodels\@xregtransient\functemplate.m
```

Copy the le to the new *MyTransientfolder*`\@xregtransient` folder you created.

2 Modify the copy of the template le for your model.

The commented code describes the compulsory and optional functions available in the template le.

You do not have to modify every section. If you do not want to use an optional section, leave it unmodi ed, and the defaults are su icient.

Open the example le `fuelPuddle.m` (for the `fuelPuddle` model) to examine the code described in the following sections. Find the le here:

```
<MATLAB root>\toolbox\mbc\mbcmodels\@xregtransient\fuelPuddle.m
```

Compulsory Local Functions

Specify the following local functions:

1 Specify the parameters of the Simulink model to t.

```
function vars= i_simvars(m,b,varargin);
vars = {'tau','x'};
```

This local function must return a cell array of strings. These are the parameter names that the Simulink model requires from the workspace. These strings must match the parameter names declared in the Simulink model.

2 If your Simulink model requires constant parameters to be de ned, do so here:

```
function [vars,vals]= i_simconstants(m,b,varargin);
vars = {};
vals = [];
```

These are constant parameters required by the Simulink model from the workspace, and are not fitted. These parameters must be the same as those in the Simulink model, and all names must match. Here `fuelPuddle` requires no such parameters, and returns an empty cell array and empty matrix.

3 Specify Initial conditions for the integrators.

```
function [ic]= i_initcond(m,b,X);
ic=[];
```

Initial conditions are based on the current parameters, and inputs could be calculated here. Leaving ic = [] means that Simulink uses the definition of initial conditions in the Simulink model.

4 Specify the number of input factors, including time.

```
function n= i_nfactors(m,b);
n= 2;
```

Time is the first input for transient models. `fuelPuddle` has the input `X = [t, u(t)]`, and so the number of input factors is 2.

5 This local function returns a column vector of initial values for the fitted parameters.

```
function [b0,OK]= i_initial(m,b,X,Y)
b0= [0.5 0.1]';
OK=1;
```

Check `nargin` before using X and Y as this local function is called with 2 or 4 parameters: (m,b) or (m,b,X,Y).

You could do some calculations here to estimate good initial values from the X, Y data. The initial values can be defined to be data dependent, hence there is an OK flag to signal if the data is not worth fitting. The example defines some default initial values for `x` and `tau`.

Optional Local Functions

You do not need to edit the remaining local functions unless they are required. The comments in the code describe the role of each function. You use these functions most often when you are creating a user-defined model (see Local Model Class: User-Defined Models on page 5-23).

These labels are used on plots:

```
function c= i_labels(m,b)
b= {'\tau','x'};
```

You can use LaTeX notation, and it is correctly formatted. By default, the variable names de ned in `i_simvars` are used to specify parameter names.

Check In to Toolbox

Having created a Simulink model and the corresponding function le, you must save each on the path, as described in Create Folders on page 5-31.

To ensure that the transient model you have de ned provides an interface that allows the toolbox to evaluate and t it, you check in the model. If this procedure succeeds, the toolbox registers the model and makes it available for tting when you use appropriate input data.

At the command-line, with both template le and Simulink models on the path, create some input data, then call `checkin`. The procedure follows.

1 Create some appropriate input data. The particular data used is not important; you are using it only to check whether the model can be evaluated.

   For the `fuelPuddle` example, enter the following input at the command-line:

   ```
   TestInputData = [0:0.1:10; ones(1,101)]';
   ```

2 Call `checkin` with the transient model, its name, and the data. For the example, enter:

   ```
   m = checkin(xregtransient,'fuelPuddle',TestInputData)
   ```

   The string you use for your new transient model name appears in the Model Setup dialog box when you set up transient models using the appropriate number of inputs.

   `xregtransient` creates a transient model object suitable for use in checking in and removing from the toolbox.

A successful checkin creates some command-line output, and a gure appears with the model name and variable names displayed over graphs of input and model evaluation output. The nal command-line output (if `checkin` is called with a semicolon as in the previous example) is

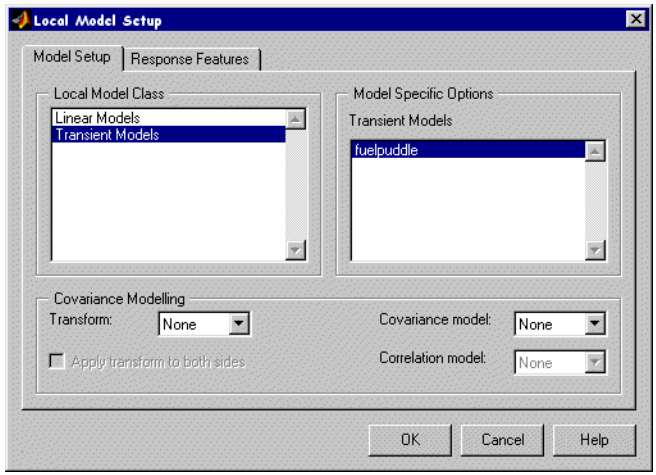
```
Model successfully registered for
Model-Based Calibration Toolbox software
```

Note You might see errors if you modify the le or model that de nes a checked-in transient model that you are currently using in the Model Browser. See Recover from Editing Errors on page 5-30.

Using and Removing Transient Models

The `fuelPuddle` model is already checked in for you to use. Open the Model Browser and load data that has the necessary format to evaluate this transient model (two local inputs). Time input data must be increasing.

Create a Test Plan with two local input factors (the same number of input factors required by the `fuelPuddle` model). The Local Model Setup dialog box now o ers the following options:



Select the `fuelpuddle` model, and click OK.

On building a response model with `fuelPuddle` as the local model, the toolbox ts the two parameters `tau` and `x` across all the tests.

Removing Checked-In Models

To remove a checked-in model, close the Model Browser, and type the following command at the command-line:

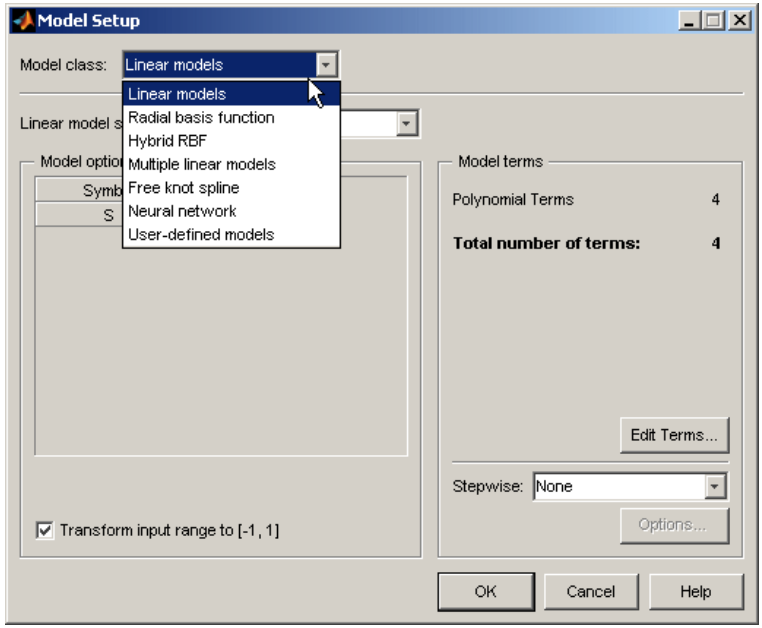
```
remove(xregtransient, 'ModelName')
```

When you restart the toolbox, the command removes the checked-in transient model.

Local Model Class: Average Fit

You can use this local model class to fit the same model to all tests. Sometimes it is desirable to try to fit a single one-stage model to two-stage data (for example, fitting an RBF over the whole operating region of spark, speed, load, air/fuel ratio and exhaust gas recirculation). However, it can still be useful to be able to examine the model fit on a test-by-test basis. You can use Average Fit for this purpose.

Select `Average Fit` and click Setup. The Model Setup dialog box appears.



In the Model class drop-down menu is a list of available models. This list contains the same models that you would find in the global model setup of a one-stage model. Note the number of inputs changes which models are available. A local model with only one input can access all the models.

The Average Fit local model class allows you to use any of these global model options to fit to all tests. In the same way that global models are fitted to all the data simultaneously,

using average t allows you to t the same model to every test in your data, instead of tting a separate local model for each test.

The advantage of this is that you can use these one stage models to t your data while also being able to view the t to each test individually. Set up your global model with an input such as record number or a dummy variable. Make all the variables you want to model local inputs. It does not matter what dummy variable or model type you use for the global input - it is only there to distinguish the Local Average Fit model from a one-stage model. The dummy global variable has no in uence on the model t. The Average Fit model is tted in the same way as a one-stage model (across all tests simultaneously) but the main di erence is that you can analyze the t to each test individually. You cannot perform such analysis when tting one-stage models.

Note No two-stage model is available with local Average Fit models, and you cannot use covariance modeling.

Transforms

The following example shows the transforms available.

Input transformation can be useful for modeling. For example, if you are trying to t

$$y = e^{a+bx+cx^2}$$

using the log transform turns this into a linear regression:

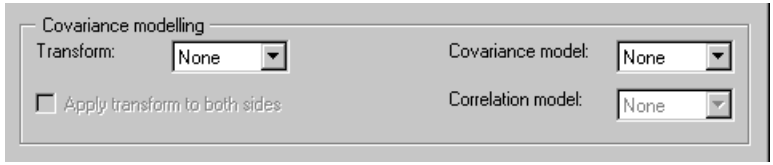
$$\log(y) = a + bx + cx^2$$

Transforms available are logarithmic, exponential, square root, y^2, $\frac{1}{y}$, and `Other`. If you choose `Other`, an edit box appears and you can enter a function.

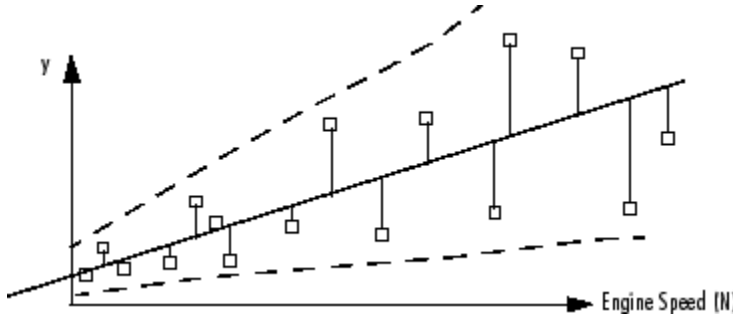
Apply transform to both sides is only available for nonlinear models, and transforms both the input and the output. This is good for changing the error structure without changing the model function. For instance, a log transform might make the errors relatively smaller for higher values of x. Where there is heteroscedasticity, as in the Covariance Modeling example, this transform can sometimes remove the problem.

Covariance Modeling

This frame is visible no matter what form of local model is selected in the list.



Covariance modeling is used when there is heteroscedasticity. This means that the variance around the regression line is not the same for all values of the predictor variable, for example, where lower values of engine speed have a smaller error, and higher values have larger errors. If so, data points at low speed are statistically more trustworthy, and should be given greater weight when modeling. Covariance models are used to capture this structure in the errors.



You can t a model by nding the smallest least squares error statistic when there is homoscedasticity (the variance has no relationship to the variables). Least squares tting tries to minimize the least squares error statistic $\sum \varepsilon_i^2$, where ε_i^2.

is the error squared at point i.

When there is heteroscedasticity, covariance modeling weights the errors in favor of the more statistically useful points (in this example, at low engine speed N). The weights are determined using a function of the form

$$\sum \frac{\varepsilon_i^2}{W_i}$$

where W_i is a function of the predictive variable (in this example, engine speed N).

There are three covariance model types.

Power

These determine the weights using a function of the form $W_i = \hat{y}^{\alpha}$. Fitting the covariance model involves estimating the parameter α.

Exponential

These determine the weights using $W_i = e^{\alpha \hat{y}}$.

Mixed

These determine the weights using $W_i = \alpha + \hat{y}^{\beta}$. There are two parameters to estimate, therefore using up another degree of freedom. This might be in uential when you choose a covariance model if you are using a small data set.

Correlation Models

These are only supported for equally spaced data in the Model-Based Calibration Toolbox product. When data is correlated with previous data points, the error is also correlated.

There are three methods available.

MA(1) - The Moving Average method has the form $\varepsilon_n = \alpha_1 \xi_{n-1} + \xi_n$.

AR(1) - The Autoregressive method has the form $\varepsilon_n = \alpha_1 \varepsilon_{n-1} + \xi_n$.

AR(2) - The Autoregressive method of the form $\varepsilon_n = \alpha_1 \varepsilon_{n-1} + \alpha_2 \varepsilon_{n-2} + \xi_n$ ξ is a stochastic input, $\xi_n - \mathrm{N}(0, \sigma_\xi^2)$.

Assess Boundary Models

In this section...

Default Boundary Models on page 5-42
Plotting Boundary Models on page 5-43

Default Boundary Models

Boundary models are nonparametric surfaces you can use as a visual aid to understanding complex operating envelopes. You can use boundaries to guide modeling and constrain optimizations. A boundary model describing the limits of the operating envelope can be useful when you are creating and evaluating designs, optimization results, and models.

Tip You t a boundary model by default when you use the Fit models common task button. Browse this page only if you want to assess a boundary model or try alternative boundary model types.

- For one-stage models, the default boundary model is a Convex Hull t to the inputs.
- For two-stage models, the default boundary model is a Convex Hull t to the global inputs and a two-stage boundary model for the local input.
- For point-by-point models, the default is a separate boundary model of type Convex Hull to each operating point.

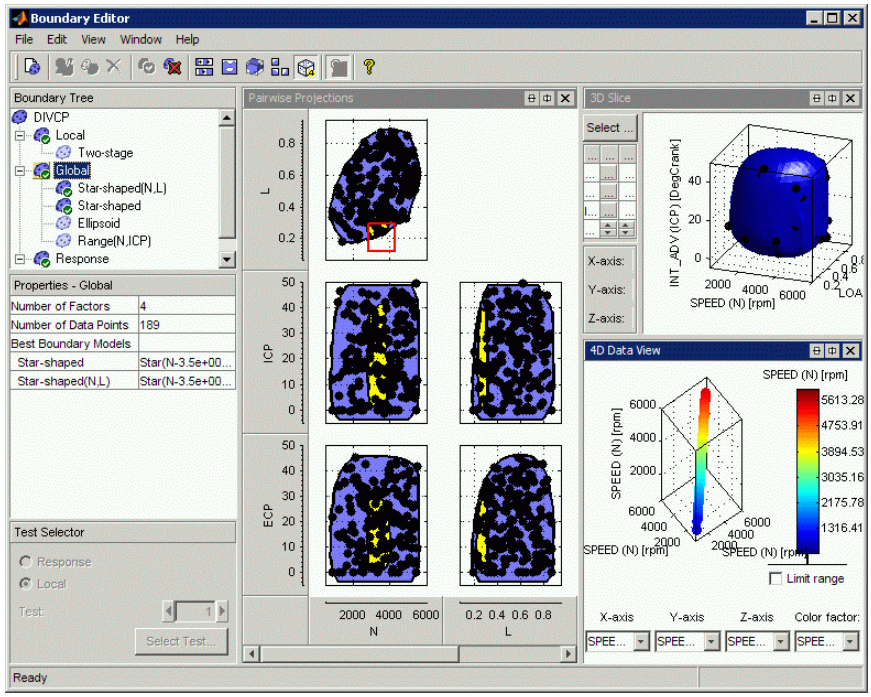
These boundary models are integrated with the rest of the toolbox. You can view them in the model plots and in CAGE in optimizations, tradeo , model views, and the Surface Viewer. You can import boundary models into the Design Editor to use as constraints. You can also use them to clip models to view only the area of interest, to constrain models and designs to realistic engine operating envelopes, or to designate the most valid areas for optimization, tradeo , and calibration.

To assess a boundary model, see Plotting Boundary Models on page 5-43.

To create alternative boundary models, see Explore Boundary Model Types on page 5-46.

Plotting Boundary Models

To edit existing boundary models or add new ones, from the test plan level, select TestPlan > Boundary Models, or the toolbar button Edit Boundary Models. The Boundary Editor appears.



In the Boundary Editor, you can edit or construct boundary models from your data.

Use the following tools to plot and highlight boundary models and points:

1-D Slice Shows a 1-D slice through your model and data. This function also appears in the View menu.

You can select values for variables in the edit boxes. These values determine the point at which the slice through the boundary is plotted.

You can change the tolerance values in the Tolerance edit boxes next to each variable to set the width of the slice. Data points within the tolerances are

displayed with the slice. This display is similar to the one in the Cross Section model selection view, where the Tolerance either side of the displayed model slice determines how near data points must be to the model slice to be displayed. See Cross Section View on page 6-50 for comparison.

You can click and hold on data points to view the values of the inputs at that point, and the distance from the boundary. Double-click a data point to move the slice view to that point, or click Select Data Point to choose a particular point.

2-D View Shows a 2-D slice through your model and data. This function also appears in the View menu. You can choose which variables to plot on the X and Y drop-down menus. Similarly to the 1-D Slice view, you can change the values and tolerances of the other variables in the edit boxes to determine where the boundary slice is plotted and how much data is also displayed.

You can click and hold on data points to view the values of the inputs at that point, and the distance from the boundary. Double-click a data point to move the slice view to that point, or click Select Data Point to choose a particular point.

3-D Slice Shows a 3-D slice of your model and data. This function also appears in the View menu. You can choose which variables to use for the three axes using the drop-down menus and set the resolution of the grid (number of points) to display in the value edit boxes for each factor. You can set the value of other variables in the edit boxes as for the 2-D view.

You can click and hold on data points to view the values of the inputs at that point, and the distance from the boundary. Double-click a data point to move the slice view to that point, or click Select Data Point to choose a particular point.

Pairwise View Shows a pairwise projection of your boundary and data. This function also appears in the View menu. Clicking this button displays a plot of the entire range for each pair of variables in turn. You can click and drag to select a region. Do not click on points, instead, click in the blue or white regions and then drag to de ne a region. The same region is then highlighted in yellow in each projection, so you can see how your data is distributed in each dimension.

Note If some points appear to be outside the boundary, select View > Set Resolution to increase the resolution of the pairwise plots. Increasing the number of evaluation points displays more detail, but takes longer to calculate.

---

4 D Data Projection A 3 D plot of the data points colored by a fourth factor. You can use the drop-down menus to select plot and color bar inputs. This function also appears in the View menu.

Highlight Boundary Points Highlights in red all data points that are on the boundary surface. This highlighting applies to all views. The points highlighted red (on the boundary) should have a `Distance` value of zero (distance to the boundary) when you click and hold on these points. Due to rounding errors, the value might not be exactly zero, although it is small. This button is only enabled for leaf nodes. Highlight Boundary Points is also in the View menu and the right-click context menu on views.

Highlight Validation Points If your testplan has validation data, plots all validation data as triangles, and highlights in red all validation points that outside the boundary surface. This highlighting applies to all views. Highlight Validation Points is also in the View menu and the right-click context menu on views.

In the View menu:

- Under Current View and Split View there are submenus where you can select any of the views available in the toolbar 1D, 2D, 3D slices, the pairwise view, and the 4D data view.
- You can select graph size (for views with multiple graphs).
- You can set resolution for the pairwise view.
- You can toggle boundary point highlighting on and o .
- You can split the currently selected view horizontally and vertically, and close the current view, as in the Design Editor and Data Editor.

All these options are available in the right-click context menu on the views.

To create and compare alternative boundary models, and export boundary models, see Explore Boundary Model Types on page 5-46.

See Also

Related Examples

Explore Boundary Model Types on page 5-46

Explore Boundary Model Types

In this section...

Alternative Boundary Model Types on page 5-46
Creating a New Boundary Model on page 5-48
Setting Up Local, Global and Response Boundary Models on page 5-49
Combining Best Boundary Models on page 5-54
Editing Boundary Model Fit Options on page 5-57
Saving and Exporting Boundary Models on page 5-61

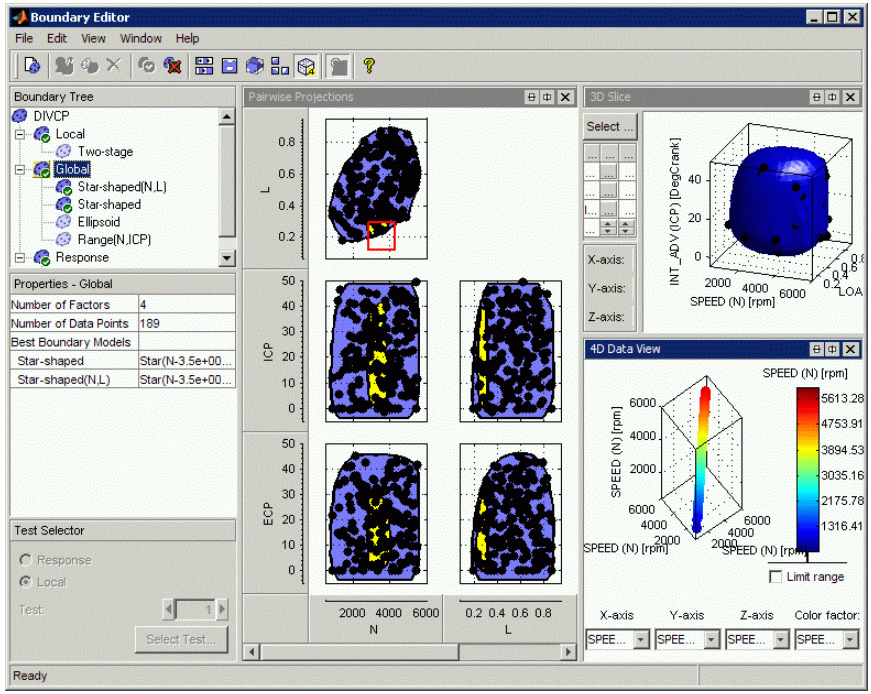
Alternative Boundary Model Types

A boundary model describing the limits of the operating envelope can be useful when you are creating and evaluating designs, optimization results, and models.

Tip You t a boundary model by default when you use the Fit models common task button. Browse this page only if you want to explore alternative boundary model types.

- For one-stage models, the default boundary model is a Convex Hull t to the inputs.
- For two-stage models, the default boundary model is a Convex Hull t to the global inputs and a two-stage boundary model for the local input.
- For point-by-point models, the default is a separate boundary model of type Convex Hull to each operating point.

To edit existing boundary models or add new ones, from the test plan level, select TestPlan > Boundary Models, or the toolbar button Edit Boundary Models. The Boundary Editor appears.



In this editor, you can construct boundary models from your data. Boundary models are nonparametric surfaces you can use as a visual aid to understanding complex operating envelopes. You can use boundaries to guide modeling and constrain optimizations.

These boundary models are integrated with the rest of the toolbox. You can view them in the model plots and in CAGE in optimizations, tradeo , model views, and the Surface Viewer. You can import boundary models into the Design Editor to use as constraints. You can also use them to clip models to view only the area of interest, to constrain models and designs to realistic engine operating envelopes, or to designate the most valid areas for optimization, tradeo , and calibration.

The toolbox saves boundary models implicitly as part of your test plan. You do not need to save them separately before closing the Boundary Editor.

From the Boundary Editor, you can export boundary models to Simulink.

From the Model Browser you can export boundary models with all your other models, from the test plan level. Use the File menu to export to CAGE, the Workspace or Simulink.

For local boundary models only, export from the test plan by selecting TestPlan > Export Point-by-Point Models.

You can use the function `ceval` to evaluate a boundary model exported to the Workspace. For example, if your exported model is `M`, then `ceval(M, X)` evaluates the boundary model attached to `M` at the points given by the matrix `X` (values less than zero are inside the boundary). See Evaluate Boundary Models in the Workspace on page 6-103 for more details.

You can import boundary models to use as constraints in the Design Editor.

Creating a New Boundary Model

You t a boundary model by default when you use the Fit models common task button. To create a boundary model, leave the Fit boundary model check box selected in the Fit Models dialog box.

To edit existing boundary models or add new ones, from the Model Browser test plan level, select TestPlan > Boundary Models, or the toolbar button Edit Boundary Models. The Boundary Editor appears.

To create a boundary model:

1 In the Boundary Editor, get started by clicking New Boundary Model in the toolbar, or select File > New Boundary.

You cannot access these options for leaf nodes. You can only add new boundary models at the root node and at the second-level nodes (local, global, or response).

2 For two-stage models, the Choose Level dialog box appears. Select a radio button to specify whether you want to model the boundary of the Local, Global, or Response values, and click OK.

Note You can skip this step if you rst select the Local, Global, or Response nodes in the Boundary Tree and then create a model. You get a new model appropriate for your current tree selection.

---

3 The controls available depend on the type of boundary model. See Setting Up Local, Global and Response Boundary Models on page 5-49.

After you create a boundary model, see Plotting Boundary Models on page 5-43.

Setting Up Local, Global and Response Boundary Models

What Are Local, Global and Response Boundary Models? on page 5-49

Setting Up Global and Response Boundary Models on page 5-49

Setting Up Local Boundary Models on page 5-51

Adding, Duplicating, and Deleting Boundary Models on page 5-54

What Are Local, Global and Response Boundary Models?

Response boundary models are built to cover the combination of the local and global variable spaces.

Global boundary models are built in the global variable space.

If you only select the global variables as active inputs to model, the di erences between a global boundary model and a response boundary model are:

- For the response boundary model, the data includes all records.
- For the global boundary model, the data is one point per test (the average value of the global variables for that test).

Local boundary models t the boundary of the local inputs. Local boundary models can be either two-stage or point-by-point global evaluation type.

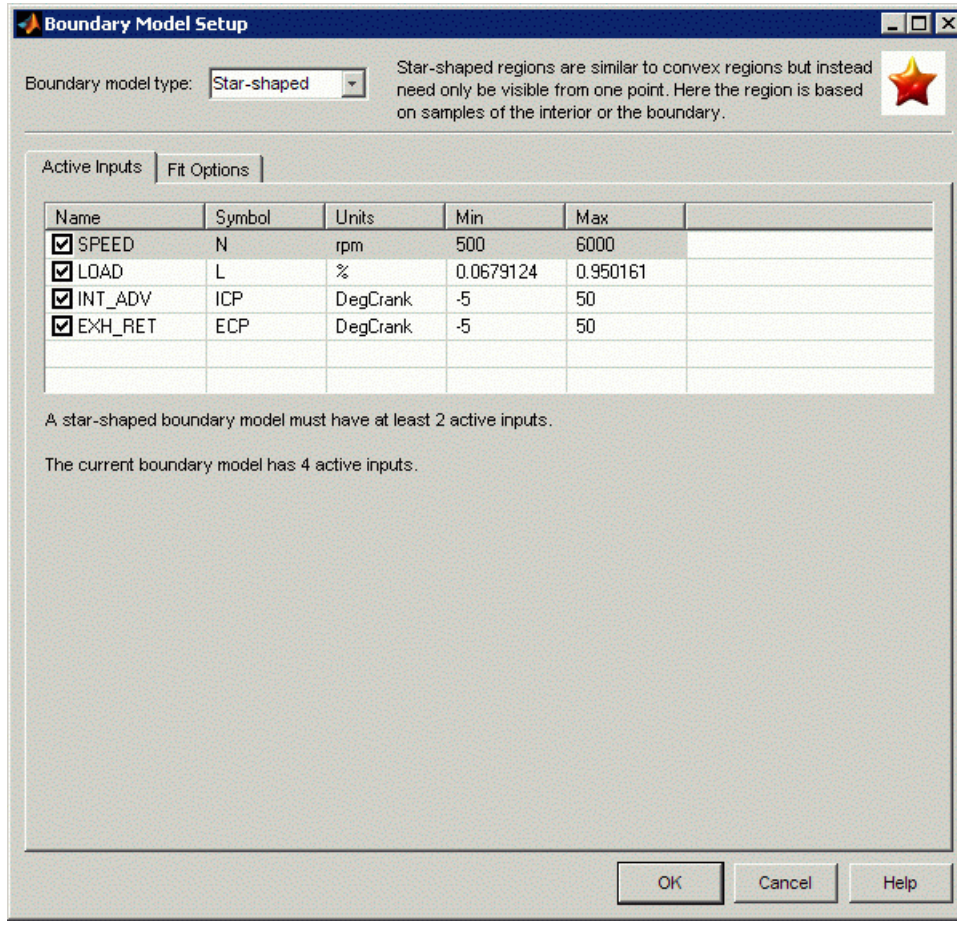
- Two-stage boundary models t the local boundary model parameters as a function of the global inputs. The toolbox uses an interpolating RBF for the function of the global inputs. For example, `min_spark = f_1(speed, load)` and `max_spark = f_2(speed, load)`. This is useful for modeling borderline spark, for example.

  Two-stage boundaries are valid at any operating point.

- Point-by-point boundary models are separate boundary models tted to the data collected at each operating point. Point-by-point boundary models are only valid at the observed operating points. The toolbox uses the global input values are used to select which local boundary model to use.

Setting Up Global and Response Boundary Models

When you set up a Global or Response boundary model, the Boundary Model Setup dialog box opens, displaying the controls shown in the following gure.



1 Select a type of boundary model: Range, Star-shaped, Ellipsoid, or Convex Hull.

The Range model finds the furthest extent of points for each variable and draws a hyper-rectangle to enclose all points.

Range is the only type that you can use with only one input.

The Ellipsoid model forms an ellipse to enclose all points.

The Convex Hull model forms the minimal convex set containing the data points.

The Star-shaped model is a more complex model with various settings that determine how your boundary model is calculated. This calculation occurs in three stages: determining the center of the data; deciding which points are on the boundary, and interpolating between those points. The star-shaped model is the only model type that can t non-convex regions.

2 Select a set of input factors to model using the Active Inputs check boxes. The required and selected number of inputs is displayed underneath. You might nd it useful to build boundary models using subsets of input factors. You can then combine them for the most accurate boundary. This approach can be more e ective than including all inputs.

3 The Fit Options tab is only enabled if your selected boundary type has any options you can set.

Range models do not have any further settings you can alter.

Star-shaped, convex hull, and ellipsoid models have several parameters that you can alter, see Editing Boundary Model Fit Options on page 5-57. Try the defaults before experimenting with these.

4 Click OK, and the toolbox calculates the boundary model.

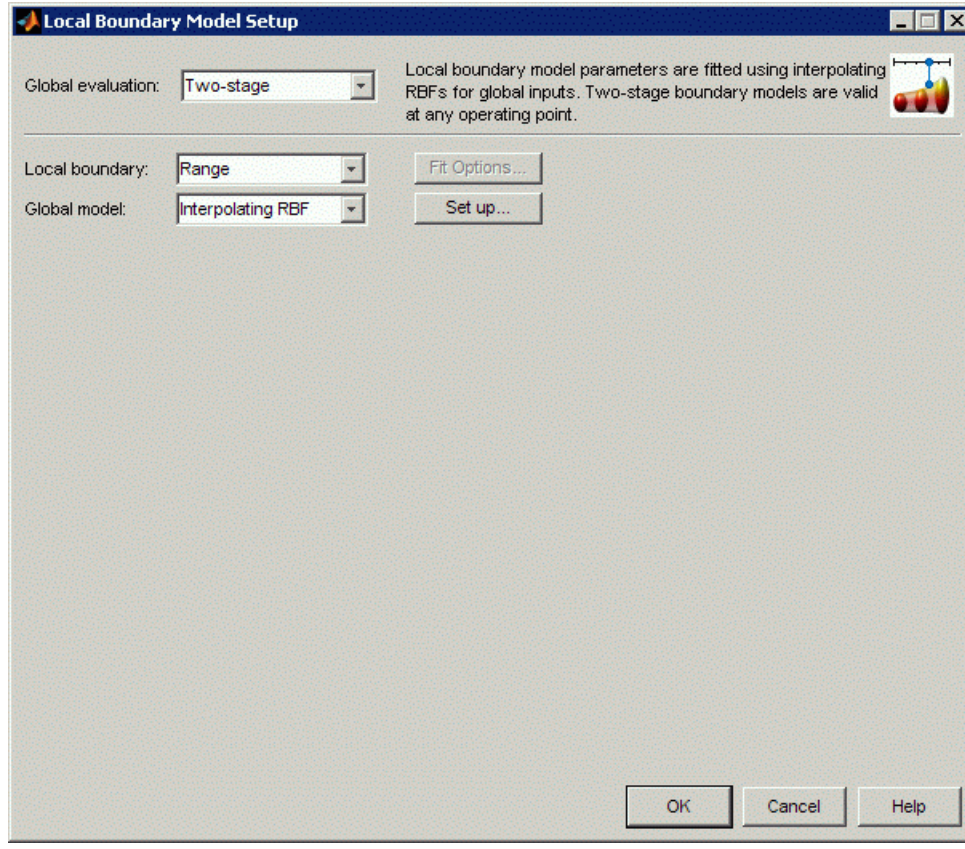
Setting Up Local Boundary Models

For Local boundary models you can set up a two-stage boundary or a point-by-point boundary. When you set up a local boundary model, the Local Boundary Model Setup dialog box opens.

Two-Stage Boundaries

To set up a two-stage boundary model:

1 Leave the default setting Two-stage in the Global evaluation list. You see the following controls.



2 You can select `Range` or `Ellipsoid` for the `Local Boundary` (to t to the local inputs).

   You can select `Ellipsoid` only for more than one local variable. You can select `Fit Options` only for `Ellipsoid` models. Click `Fit Options` to see the parameters that you can alter. Try the defaults before experimenting with these. See Editing Boundary Model Fit Options on page 5-57.

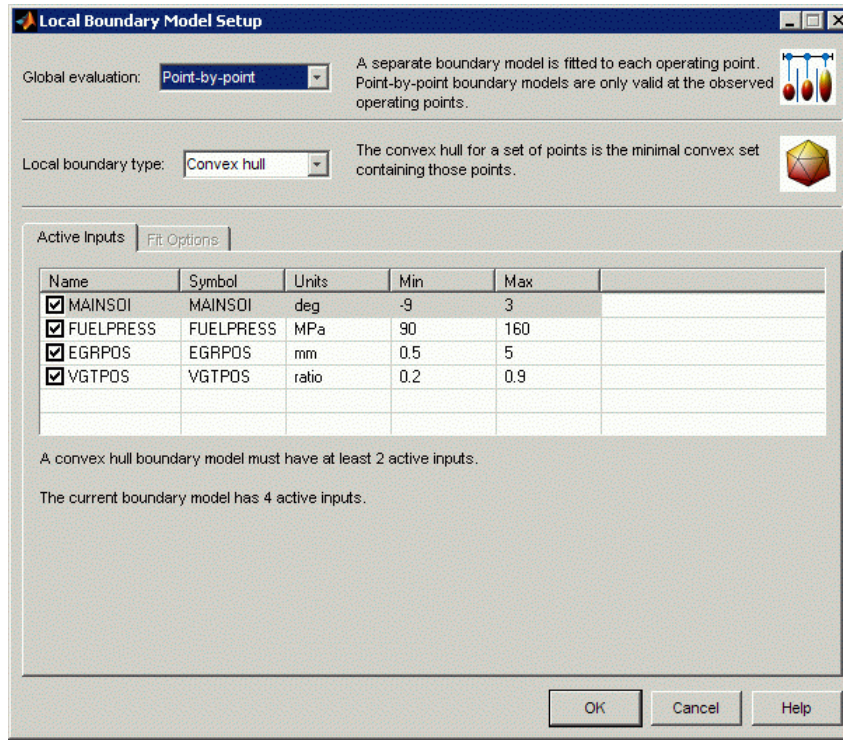
3 The `Global model` must be an interpolating RBF that interpolates across the global inputs between these local boundaries. You can click `Set Up` to change the parameters for the interpolating RBF.

4 Click `OK`, and the toolbox calculates the boundary model.

Point-by-Point Boundaries

To set up a point-by-point boundary model:

1 Select `Point-by-point` in the Global evaluation list. You see the controls shown in the following gure.



2 Select the boundary settings. The available settings for point-by-point boundary models are the same as for global or response boundary models:

a Select a Local boundary type: Range, Star-shaped, Ellipsoid, or Convex Hull.

b Select a set of input factors to model using the Active Inputs check boxes.

c Optionally, view or edit settings on the Fit Options tab, if enabled for your boundary type.

See Setting Up Global and Response Boundary Models on page 5-49 for more details.

3 Click OK, and the toolbox calculates the boundary model.

Adding, Duplicating, and Deleting Boundary Models

New boundary model (also in the File menu) Opens a setup dialog box, and, when you click OK, adds a child node (containing a tted boundary model) to the current node. This button is not enabled at leaf nodes. Similarly to the model tree in the Model Browser, the new child nodes are di erent depending on the location of your current selection in the tree.

For one-stage test plans, new child nodes of the root (top) node are boundary models (leaf nodes).

For two-stage test plans, new child nodes di er depending on the parent node. From the top or root node, you can choose a new child node of either local, global, or response type. From local, global, or response nodes, new child nodes are boundary models of the same type as their parent nod local, global, or response. You can add as many boundary models of each type as you want.

These toolbar buttons are only available for leaf nodes: (also in the Edit menu):

Duplicate boundary model Duplicates the current node.

Delete boundary model Deletes the current node.

Combining Best Boundary Models

You can select a single boundary model node as best or you can combine models by including them in your best selections. You might nd it useful to build boundary models using subsets of inputs and di erent boundary types. You can then combine these models to achieve the most accurate boundary. This approach can be more e ective than including all inputs.

Note Only models selected as best are exported. Select one or more models as best, or you cannot export boundary models.

Look at the tree icons to see which boundary models are included as best. Included tree nodes have a check mark on their icon. In the following example, Response is included in best, and the child node Star-shaped is not included.

For two-stage or point-by-point test plans, the Local, Global, and Response nodes are included in best by default (even though they are empty to begin with). You can include or exclude any tree node except the root node.

Each parent node displays the combination of child nodes that you have selected as best (if any; otherwise the parent node is empty.) Use the toolbar and Edit menu items Add to Best and Remove From Best to include only the nodes you require at the root node.

For example, if you pick two leaf nodes and select Add to Best for each, the parent node shows a combination of the two boundary models. You might want your final model to combine boundaries of different types, e.g., a star-shaped and a range boundary. You can view the results at the parent node: the combined model is clipped to fit within the ranges defined by both boundaries. You can combine as many leaf nodes as you like.

Note You can always see which boundaries have been combined at the currently selected node by viewing the Properties pane.

Use the following toolbar buttons or the Edit menu to combine and remove boundary models. These toolbar buttons are not available for the root node (only leaf and second-level nodes).

Add to best Include selected node in best. Only enabled where the selected node is excluded from best.

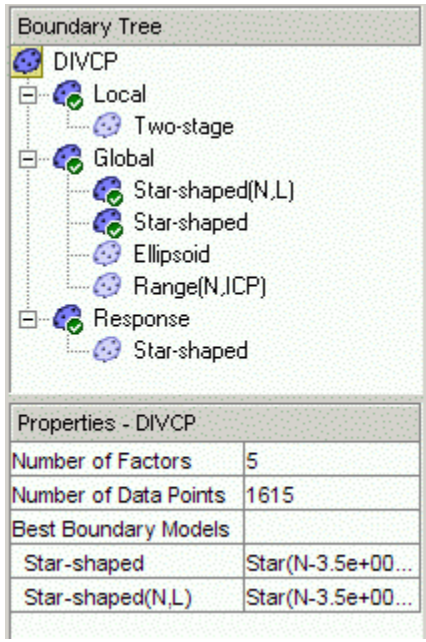
Remove from best Exclude selected node from best. Only enabled where the selected node is included in best.

You can combine the best models for one-stage or two-stage (local, global, and response) leaf nodes.

The Local node displays the child node or nodes selected as best (or is empty if none are selected as best). The Global and Response nodes also display their child node or

combination of nodes selected as best. You can only combine local leaf nodes with other local leaf nodes, and global leaf nodes with other global leaf nodes, etc., because you can only have one best model at the `Local` node, one best model at the `Global` node, and one best model at the `Response` node. However you can choose to combine any of the `Local`, `Global`, or `Response` nodes as best for the root node. You can see which child model (or combination) is best from a parent node (e.g., `Global`) or the root node by looking under `Best Boundary Model` in the bottom left `Properties` pane.

See the following figure for an example.



In this example, the root node (`DIVCP`) is selected the icon is outlined. Look at the `Properties` pane to see which leaf node (or nodes) you have included in the set of best boundary models for the selected node in this case, it is `Star-shaped` (all inputs) and the `Star-shaped(N,L)` global nodes. The icons in the rest of the tree show the path of combined nodes.

Looking one level down in the tree, Local, Global, and Response are all included in best, but Local and Response are empty because they do not contain any child nodes selected as best. The Local and Response nodes are included in the best selection for the root

node, but they have no e ect because they are currently empty. Selections at di erent levels of the tree (branch and leaf) are independent.

Only the Global node contains any child nodes included in best, and therefore only that combination of global child nodes selected as best is displayed at the parent node (the root).

The Global node has the `Star-shaped` (all inputs) and the `Star-shaped(N,L)` child nodes selected as best. Therefore the `Global` node contains the combination of the `Star-shaped` (all inputs) and the `Star-shaped(N,L)` boundary models. The Global node is included in the best for the root node, so the root node also contains the `Star-shaped` (all inputs) and the `Star-shaped(N,L)` boundary models.

Editing Boundary Model Fit Options

To edit settings for existing boundary models, you can reach the Boundary Model Setup dialog boxes by selecting Edit > Set Up Boundary or the equivalent toolbar button, . This action opens the Boundary Model Setup dialog box where you can edit settings for the selected model. The new model is tted when you click OK. You can edit the boundary type and active inputs, and settings where available.

You can view the type and details of the selected boundary model in the Properties pane. This pane displays information about the boundary model such as: number of data points; number of boundary, interior, and exterior points; the type of boundary model and summary of settings (e.g. center point of star). For root and branch nodes, you can see which model or combination of models you have selected as best.

See Combining Best Boundary Models on page 5-54 for an example.

You can reach the Fit Options settings in the Boundary Model Setup dialog box, either when creating a boundary model, or when editing an existing boundary model, by selecting Edit > Set Up Boundary or the equivalent toolbar button.

The Fit Options tab (or button for local boundaries) is only enabled if your selected boundary type has any options you can set.

Range models have no additional settings you can alter.

Star-shaped, convex hull, and ellipsoid models have parameters that you can alter, as the following topics describe. Try the defaults before experimenting with these.

Ellipsoid Settings

The ellipsoid boundary has several optimization settings you can alter if you are having problems getting a good fit. Change the display setting to `iter` or `final` to see output messages at the command-line during the fit, and try different tolerances or numbers of iterations. For details on these settings see `optimset` in the MATLAB Reference.

Convex Hull Setting

The default convex hull boundary model keeps only the most useful facets and can discard around 30% of the facets that contribute only a small amount to the boundary. Discarding these facets is more efficient and includes all the data points within the boundary, with the total volume increasing by around 1%.

If you want the minimal volume at the expense of a much larger number of facets and a larger project file size, select the Keep All Facets check box. The total number of facets can be many thousands. In a typical example, using 8 factors and 70 points results in a convex hull with around 35,000 facets.

Star-Shaped Settings

The `Star-shaped` boundary is a more complex model with various settings that determine how your boundary model is calculated. This determination occurs in three stages: determining the center of the data; deciding which points are on the boundary; and interpolating between those points.

Special Points > Center This setting is not the same as the Center Selection settings for RBFs, instead, the toolbox uses this setting as the method for determining the center of the boundary model sphere. Think of the boundary model as a deformed sphere. You can choose `Mean`, `Median`, `Mid Range`, `MinEllipse` or `User Defined`. If you select `User Defined`, you can enter a value for each input.

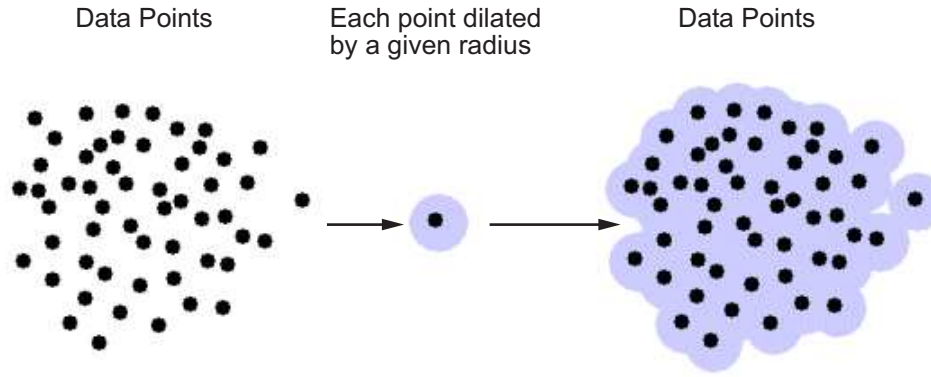
Boundary Points These settings determine how to decide which points are on the boundary.

`Interior` Choose this option to specify that not all points should be on the boundary.

`Boundary Only` Places all points on the boundary. This setting can save you time, if this is suitable for your data.

Dilation Radius If you choose `Interior` points, then Dilation Radius is used to determine which points are on the boundary. The model-fitting calculation

expands each point to a sphere until the intersection of all those spheres forms a boundary shape.

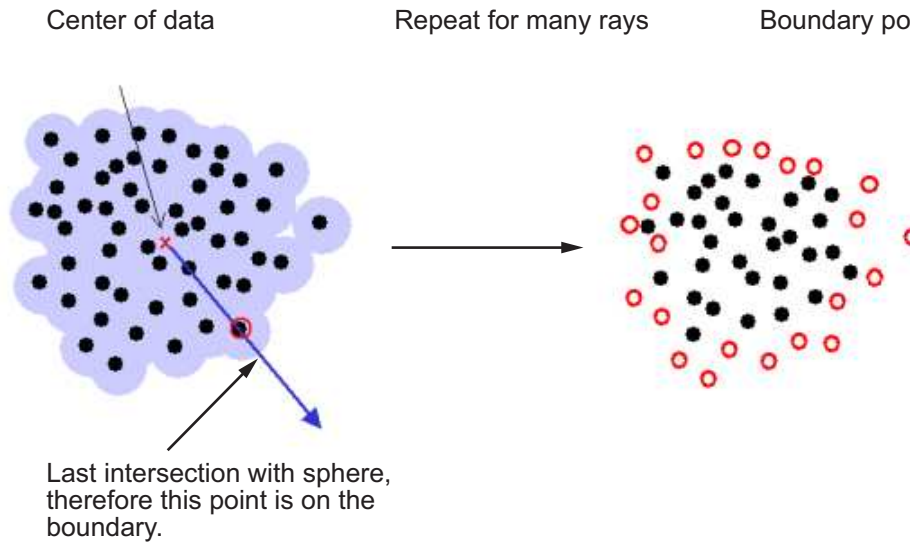


Dilation Radius settings:

`Auto` This setting selects the dilation radius (how much to expand each point) by checking all the minimum distances between points, and then choosing the largest of those.

`Manual` You can manually set the dilation radius in the edit box. The default is 1. This value might seem large as model range is from -1 through 1, but all points are expanded equally so you will still detect the points on the edge. However, large spheres will intersect and obscure points that should be detected as boundary points.

Ray Casting The toolbox draws rays from the center of the boundary model to determine which points are on the edge. The last point intersected on each ray is a boundary point. The ray actually intersects a sphere, given that the dilation radius expands the search point.



Ray casting settings:

`From Data` This option uses the same number of rays as there are data points and sends one ray in the direction of each point. If you have dense data or a large number of points, it might be better to use the `Manual` setting to choose a smaller number of rays.

`Manual` You can set a value in the `Number of Rays` edit box. This number of rays is used in random directions. A good guideline is to use about twice the number of data points, although if you have a large number (many hundreds), the model fit becomes slow, and you might run out of memory. In most situations, more than 1000 is too many.

Constraint Fit.

`Transform` `None`, `Log`, or `McCallum`. The default is `None`. Depending on the shape of your boundary, you might need to use a transform to prevent self intersections near the center of the model.

`RBF Kernel` Radial Basis Function (RBF) settings. You can choose RBF kernels, width, and continuity as when setting up models. See Global Model Class: Radial Basis Function on page 5-69 for more information. After the boundary points have been determined, each of those points is used as an RBF center and the toolbox obtains the boundary surface by interpolating radial basis functions

between all those centers. The width and continuity settings depend on which kernel you choose.

RBF Algorithm

These options control the interpolating RBF model settings. You can leave the defaults unless you have a large data set (several thousand points). With large data sets, you can improve the speed and robustness of fitting if you try a different Algorithm setting, such as GMRES, first and then vary the tolerance and number of iterations.

Saving and Exporting Boundary Models

File > Close Closes the Boundary Editor and saves your boundary models with the test plan.

File > Export to Simulink Exports the currently selected node as a block in a Simulink model. If the currently selected node is a leaf node, the toolbox exports a single boundary model. If the selected node is a parent node (root, local, global, or response), the toolbox might export one boundary or a combination of several depending on which nodes you have assigned best and added to best.

See Also

Related Examples

Assess Boundary Models on page 5-42

Plotting Boundary Models on page 5-43

Explore Global Model Types

In this section...

Alternative Global Model Types on page 5-62
Global Linear Models: Polynomials and Hybrid Splines on page 5-63
Global Model Class: Gaussian Process Model on page 5-68
Global Model Class: Radial Basis Function on page 5-69
Global Model Class: Hybrid RBF on page 5-73
Global Model Class: Interpolating RBF on page 5-74
Global Model Class: Multiple Linear Models on page 5-76
Global Model Class: Free Knot Spline on page 5-77
Global Model Class: Neural Network on page 5-78
Global Model Class: User-De ned and Transient Models on page 5-80

Alternative Global Model Types

First, try tting the defaults using the Fit models common task button.

To build a selection of global models to compare, in the Common Tasks pane, click Create Alternatives. See Create Alternative Models to Compare on page 5-84 for details.

If you want to try a single alternative global model type, click Add Model in the Common Tasks pane at any global model node. This opens the Global Model Setup dialog box.

Browse all the available model types on this page.

Some global models are only available in the Model class list when using the appropriate number of inputs. The example of a user-de ned model is for a single input; the example transient model is for two inputs. You can check in your own user-de ned models and transient models with as many factors as you choose; these only appear as options when the appropriate number of inputs are present.

Global Linear Models: Polynomials and Hybrid Splines

Global linear models can be polynomials or hybrid splines. Options are described in the following sections:

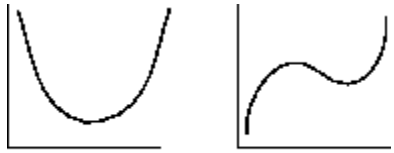
Polynomials on page 5-63
Hybrid Splines on page 5-66

Polynomials

Polynomials of order n are of the form

$$\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 \ldots \beta_n x^n$$

You can choose any order you like for each input factor.

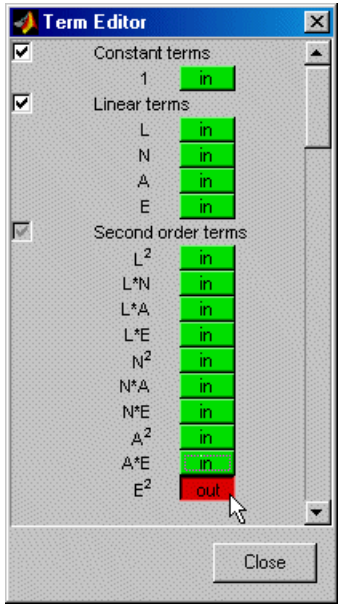


Quadratic curve Cubic curve

A quadratic polynomial $y = ax^2 + bx + c$ can have a single turning point, and a cubic curve $y = ax^3 + bx^2 + cx + d$ can have two. As the order of a polynomial increases, it is possible to t more turning points. The curves produced can have up to (n-1) bends for polynomials of order n.

See also the local model Polynomials on page 5-8 for information about di erent settings available.

Click the Edit Terms button to see the terms in the model. This opens the Term Editor dialog box. Here you can remove any of the terms.



Interaction: You can choose the interaction level on both linear model subclasses (polynomial and hybrid spline). The maximum interaction level is the same as the polynomial order (for example, three for cubics).

The interaction level determines the allowed order of cross-terms included.

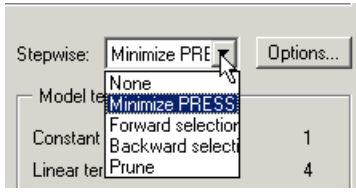
You can use the Term Editor to see the e ects of changing the interaction level. Click the Edit Terms button. The number of constant, linear, second-, and third-order (and above) terms can be seen in the Model Terms frame.

For polynomials, with an interaction level of 1, there are no terms in the model involving more than one factor. For example, for a four-factor cubic, for factor L, you see the terms for L, L^2, and L^3, but no terms involving L and other factors. In other words, there are no cross-terms included.

If you increase the interaction level to 2, under second-order terms you see L^2 and also L multiplied by each of the other factors; that is, second-order cross-terms (for example, LN, LA, and LS).

Increase the interaction to 3, and under third-order terms you see L^2 multiplied by each of the other factors (L^2N, L^2A, L^2S), L multiplied by other pairs of factors (LNA, LNS, LAS), and L multiplied by each of the other factors squared (N^2L, A^2L, S^2L). Interaction level three includes all third-order cross-terms.

The preceding also applies to all four factors in the model, not just L.



Stepwise: Take care not to over t the data; that is, you do not want to use complex models that chase points in an attempt to model random e ects.

The Stepwise feature can help. Stepwise selects the terms using various criteria. Stepwise generally means that terms are removed in steps (one at a time). The stepwise algorithms are Minimize Press, Forward Selection, Backwards Selection, and Prune. The most commonly used stepwise algorithm is `Minimize PRESS`, where at each step the term that improves the PRESS statistic the most is removed or included. `Minimize PRESS` throws away terms in the model to improve its predictive quality, removing those terms that reduce the PRESS of the model. Forward and Backwards Selection uses statistical signi cance at the alpha % level.

Predicted sum of squares error (PRESS) is a measure of the predictive quality of a model. See PRESS statistic on page 6-74 for an explanation of PRESS and Guidelines for Selecting the Best Model Fit on page 6-54 for more information on why it is useful as a diagnostic statistic.

Prune is one of the alternative algorithms for stepwise. The order of the terms matter, and the terms are removed from the end, provided they improve the quality of the t (measured by various criteria: PRESS, GCV etc.). The other stepwise algorithms do not have this restriction - they can remove any term in any order. Removing terms only from the end is valid when there is ordering in the terms, e.g., polynomials (from low-order terms to high-order terms) or RBFs where the tting algorithms select the most important terms rst.

Click Options to open a dialog box containing further settings for the selected Stepwise option. Choose from the list a criteria for removing terms (PRESS, RMSE, AIC, BIC etc.).

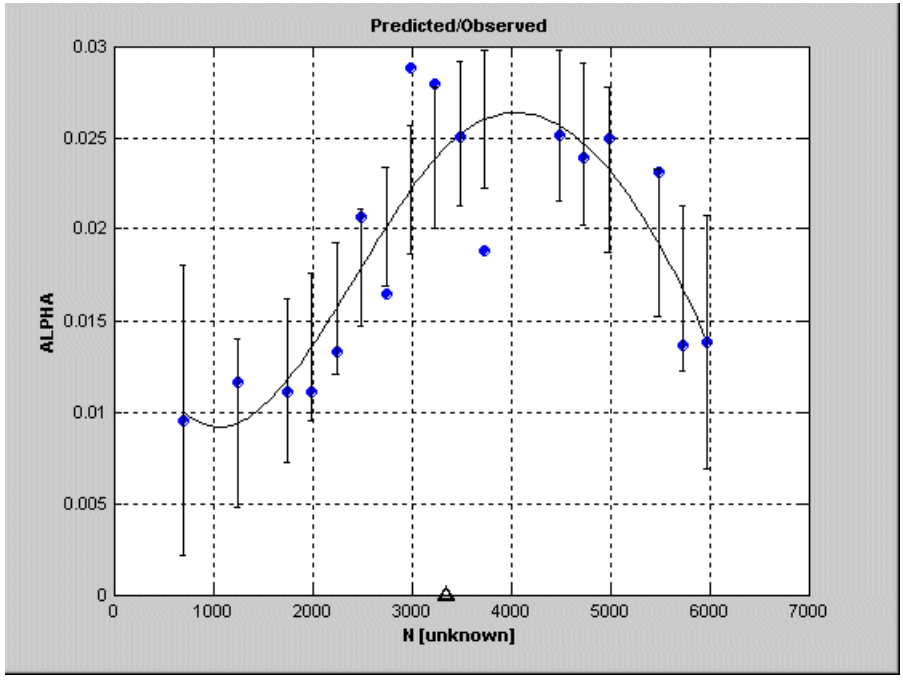
For the Prune settings, see Global Model Class: Radial Basis Function on page 5-69. For a guide to all the settings in the Stepwise window (which explains the other Stepwise settings available here), see Stepwise Regression on page 6-65. Note you can also use the Stepwise window after model fitting to try other Stepwise settings, and replace excluded model terms if you want.

Hybrid Splines

You can use the Hybrid Spline model to fit a spline to one factor and polynomials to all other factors.

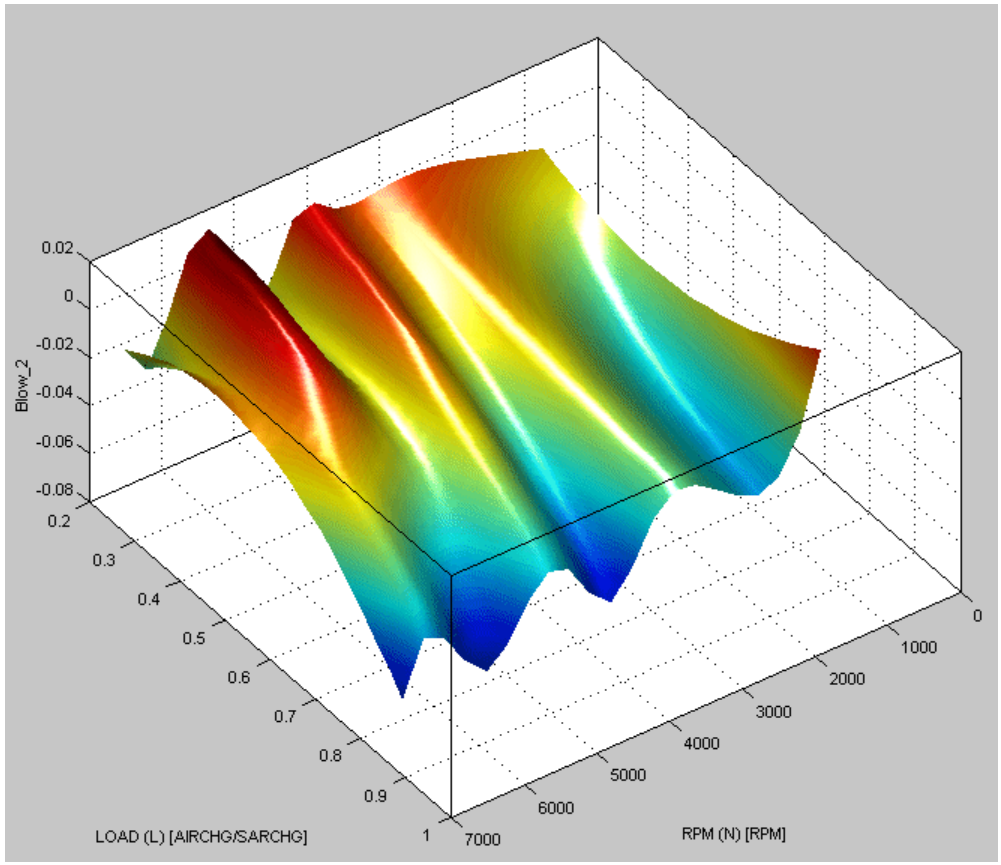
A spline is a piecewise polynomial function, where different sections of polynomials are fitted smoothly together. The locations of the breaks are called knots. You can choose the required number of knots (up to a maximum of 50) and their positions. In this case all the pieces of curves between the knots are formed from polynomials of the same order. You can choose the order (up to 3).

The following example illustrates the shape of a spline curve with one knot and third-order basis functions. The knot position is marked on the N axis.



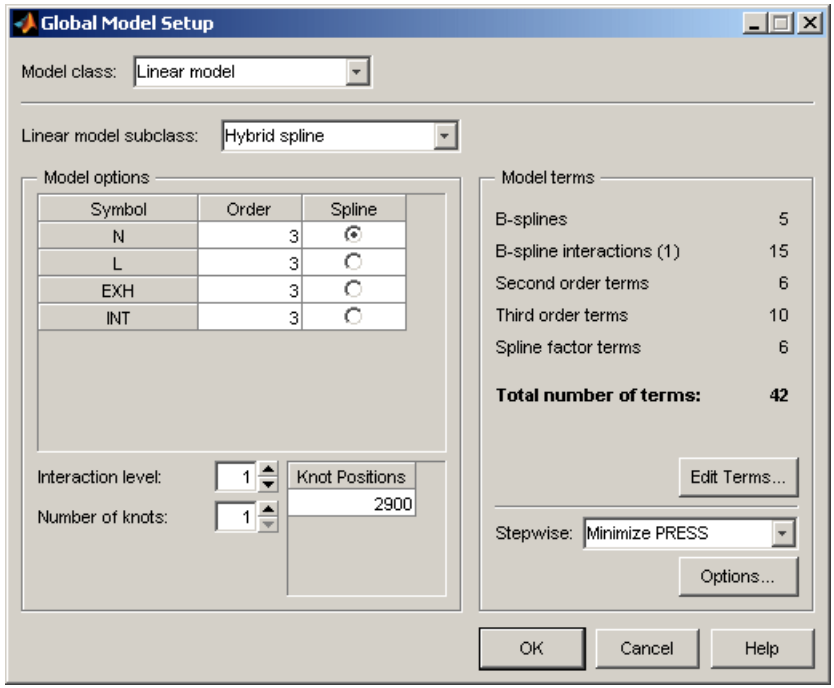
You can fit more complicated curves using splines, so they can be useful for the factor you expect to behave in the most complex way. This allows you to model detailed fluctuations in the response for one factor, while simpler models are sufficient to describe the other factors.

The following example clearly shows that the response (Blow_2 in this case) is quadratic in the Load (L) axis and much more complex in the RPM (N) axis.



You can choose the order of the polynomial for each factor and the factor to fit the spline to. The maximum order for each factor is cubic. Use the radio buttons to select which factor is modeled with a spline. Select the order for each factor in the edit boxes.

The following example shows the options available for the Hybrid Spline linear model subclass.



See also Local Model Class: Polynomials and Polynomial Splines on page 5-8.

For hybrid splines, the interaction function is different to polynomials: it refers only to cross-term interactions between the spline term and the other variables. For example, at interaction order 0, raw spline terms only; interaction 1, raw terms, and the spline terms x the first-order terms; interaction 2, includes spline terms x the second-order terms.

Global Model Class: Gaussian Process Model

Gaussian process models (GPM) are popular non-parametric regression models used in model-based calibration. These models can usually produce a good fit without needing to tune lots of parameters.

Settings:

Kernel function

Explicit basis function

If you want to try all kernels and basis function options, use the Gaussian Process model template to build a selection of Gaussian process models. Click Create Alternatives in the Common Tasks pane, and see Gaussian Process Template on page 5-88.

For large data sets (>2000 points), Gaussian process models use the default large data settings from Statistics and Machine Learning Toolbox.

If you have a large data set (>2000 observations), in the Model Setup dialog box you can try the large data fit options to see if other sparse methods result in better fits.

1. Select the Show large data fit options check box. This displays further options that can be helpful for larger data sets.
2. To apply these options when fitting, edit the Threshold value to less than the number of observations in your data set.

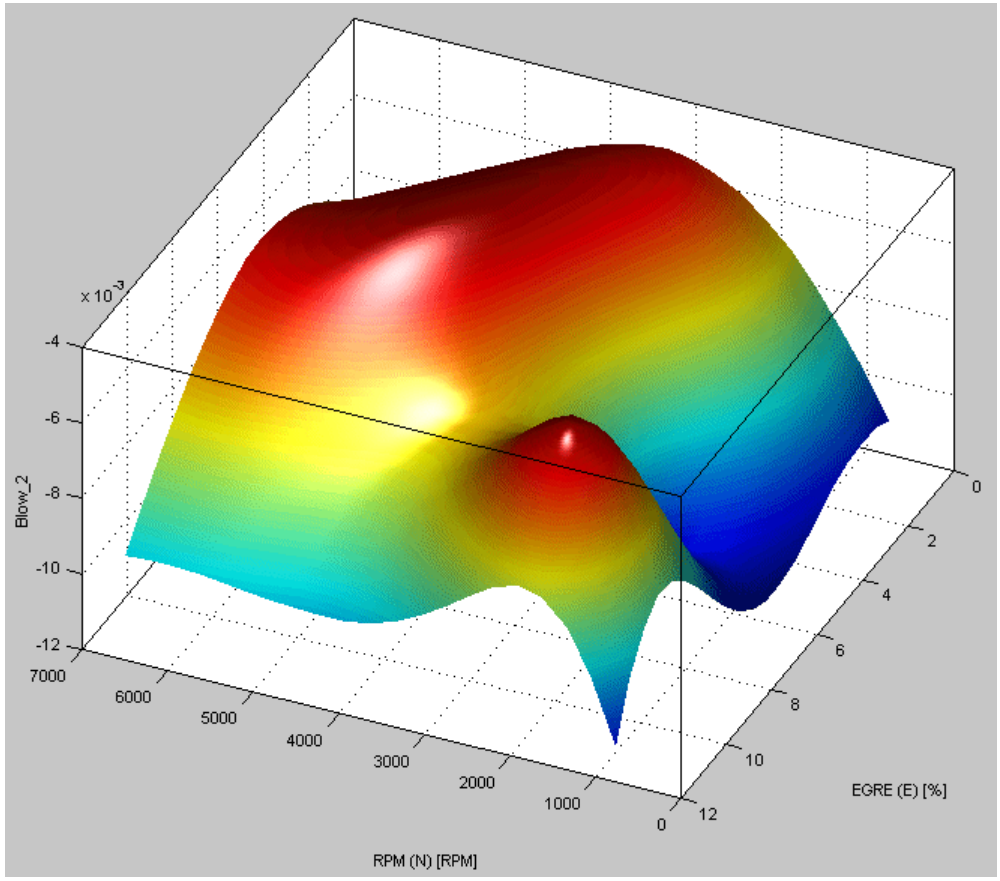
Gaussian process models do not support the AICc selection criteria or MLE.

Global Model Class: Radial Basis Function

Several radial basis functions (RBFs) are available in MBC. They are all radially symmetrical functions that can be visualized as mapping a flexible surface across a selection of hills or bowls, which can be circular or elliptical.

Networks of RBFs can model a wide variety of surfaces. You can optimize on the number of centers and their position, height, and width. You can have different widths of centers in different factors. RBFs can be useful for investigating the shapes of surfaces when system knowledge is low. Combining several RBFs allows complicated surfaces to be modeled with relatively few parameters.

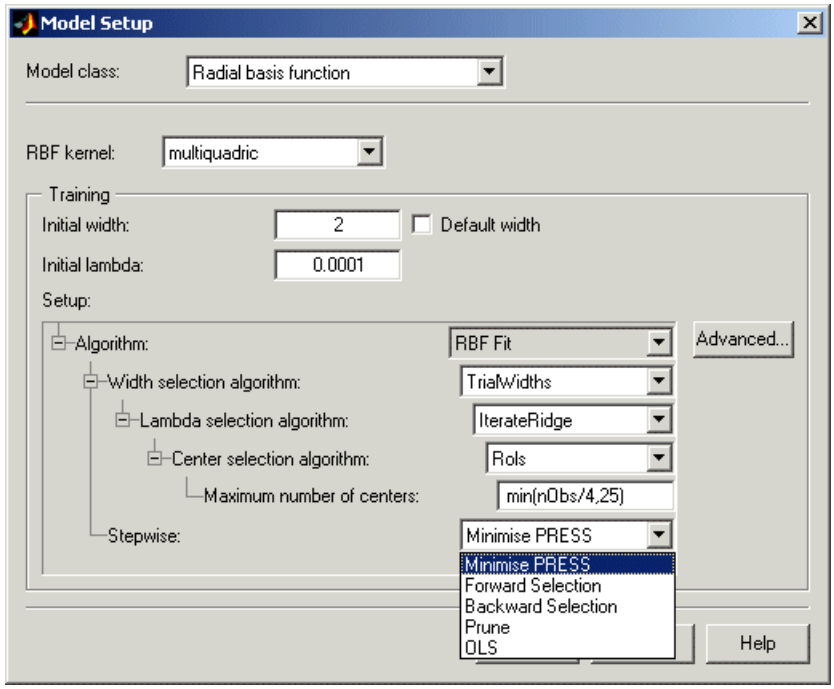
The following example shows a surface of an RBF model.



There is a detailed user guide for modeling using RBFs in Radial Basis Functions for Model Building on page 7-2. See especially Tips for Modeling with Radial Basis Functions on page 7-31 for guidelines and Types of Radial Basis Functions on page 7-3 for illustrations of di erent types.

The statistical basis for each setting in the RBF global models is explained in detail in Radial Basis Functions for Model Building on page 7-2.

The following example illustrates the basic RBF settings available.



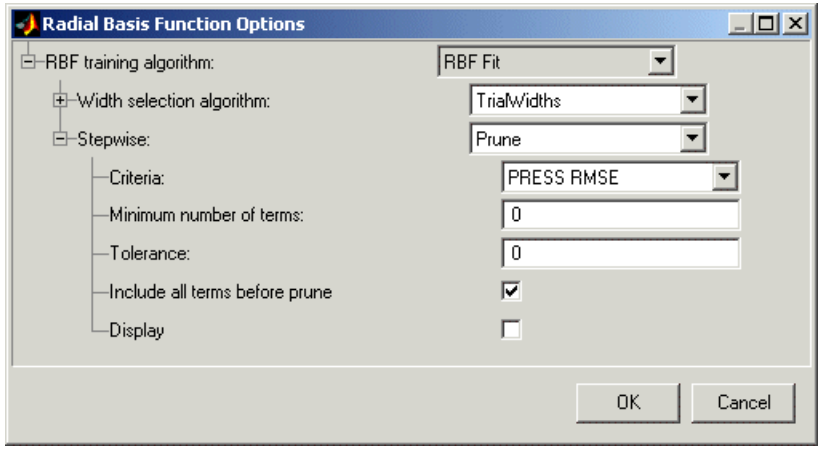
You can use the drop-down menus to set RBF kernel type, initial width and lambda, width, lambda, and center selection algorithm and maximum number of centers. After you have tted a model once to get some idea of what to expect, you can try di erent maximum numbers of centers as a useful method for homing in on a good t. There are more options for ne-tuning in the Advanced options dialog box, but you can use the main controls from here to narrow down the search for the best model.

For most algorithms the Initial width is a single value. However, for `WidPerDim` (available in the Width selection algorithm pull-down), you can specify a vector of widths to use as starting widths. `WidPerDim` produces elliptical basis functions that can have a di erent width in each factor direction. If supplying a vector of widths, there should be the same number as the number of global variables, and they must be in the same order as speci ed in the test plan. If you provide a single width, then all dimensions start o from the same initial width, but are likely to move from there to a vector of widths during model tting.

You can use the last drop-down menu to choose to run Stepwise at the end of the center/width selection algorithm to remove less useful model terms, Ordinary Least Squares

(OLS), or the Prune algorithm to home in on the best number of centers (using your choice of the Summary Statistics as selection criteria).

Of you choose Prune, there are further settings you need which can be found by clicking Advanced. This opens the Radial Basis Function Options dialog box.



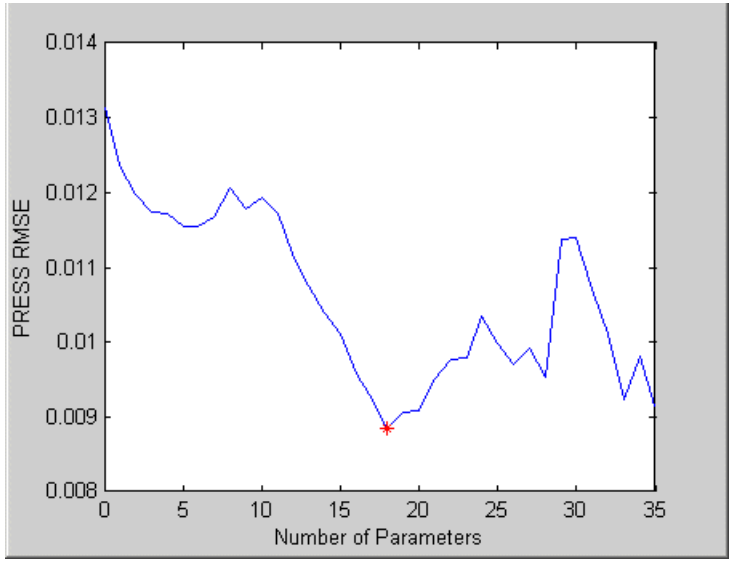
All the settings under Width selection algorithm are for ne-tuning the RBF model. See Tips for Modeling with Radial Basis Functions on page 7-31 for guidelines, and details on speci c parameters for di erent algorithms can be found in Radial Basis Functions for Model Building on page 7-2.

The options in the Stepwise drop-down menu are the same as the main Model Setup dialog box Min. PRESS, Forward, Backward, Prune, and OLS (Ordinary Least Squares). If you choose Prune, there are more options. Choose one of the Summary Statistics as selection criteria for the Prune algorithm. All the Summary Statistics options are available as criteria, and do not depend on your choices of these statistics in the Summary Statistics dialog box. See Summary Statistics on page 6-28 for more information.

We recommend that you select the check box to Include all terms before prune (otherwise the current number of terms is used at the start). You can choose a Minimum number of terms, and the Tolerance you set determines how far from this number of terms the algorithm can go within the limits of the tolerance, the algorithm searches for fewer terms that reduce the value of your selection criteria.

If you select the Display check box a gure appears illustrating the Prune process, like the example shown following, plotting the number of parameters against the selection

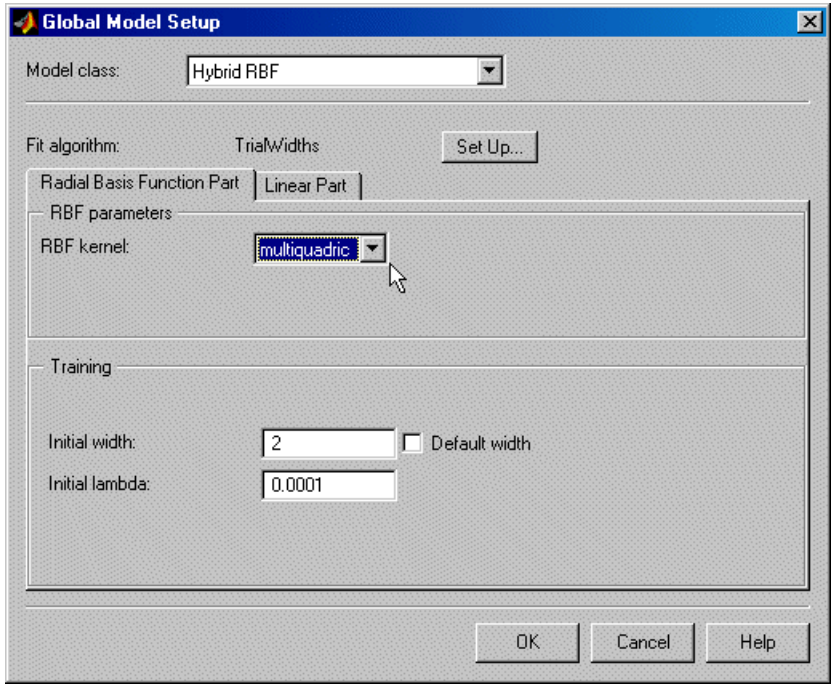
criteria, in this case PRESS RMSE. You can use this information to determine if you need to change the minimum number of terms and the tolerance and re t to avoid a local minimum.



Note Once you have a global model, you can use the RBF template in the Create Alternative Models to Compare on page 5-84 to build several radial basis function models with varying maximum numbers of centers and/or di erent kernels.

Global Model Class: Hybrid RBF

This option combines an RBF model with a linear model.



The RBF kernel drop-down menu o ers the same options as for normal RBF.

The Linear Part tab contains the same options as the other global linear models; see Global Linear Models: Polynomials and Hybrid Splines on page 5-63.

See Hybrid Radial Basis Functions on page 7-29.

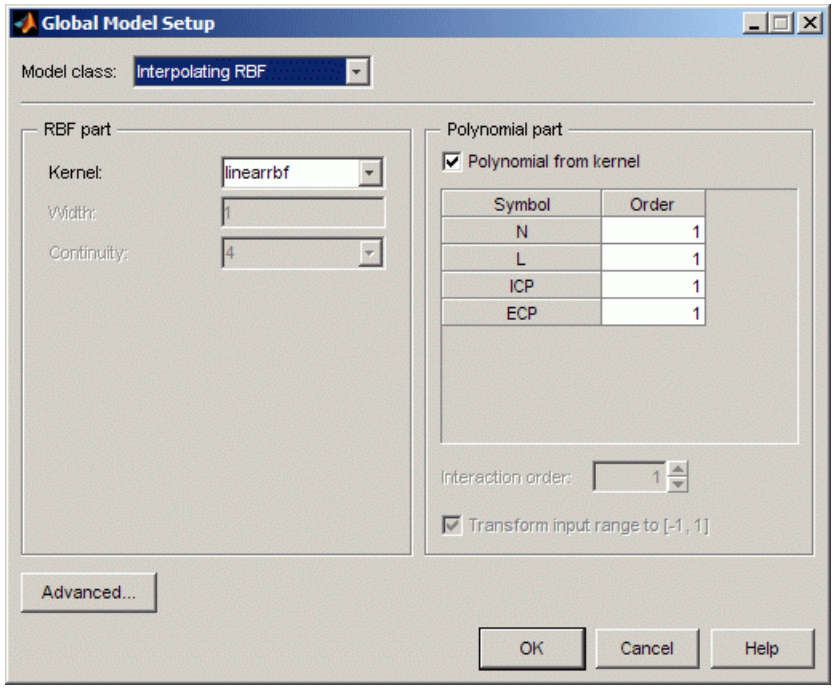
Click Set Up to reach the Hybrid RBF Options dialog box where you can change all the settings for the RBF part of the model. Here you can also choose to run Stepwise, OLS, or Prune. These settings are the same as the Radial Basis Functions Options dialog box, see Global Model Class: Radial Basis Function on page 5-69 for details.

See also Radial Basis Functions for Model Building on page 7-2 for a detailed guide to the use of all the available RBFs.

Global Model Class: Interpolating RBF

The Interpolating RBF model type ts an interpolating surface that passes through every data point. Each point is used as a radial basis function center and the toolbox

interpolates RBFs between all those centers. This model type is also used by the Boundary Editor for creating boundary models.



The Kernel drop-down menu o ers the same options as for normal RBF. The Width and Continuity parameters are only enabled for speci c kernels.

The Polynomial Part tab contains the same options as the other global linear models (order and interaction); you cannot edit them unless you clear the check box to create the Polynomial from kernel. See Global Linear Models: Polynomials and Hybrid Splines on page 5-63.

You can click Advanced to reach more interpolating RBF model settings. You can leave the defaults unless you have a large data set (several thousand points). With large data sets, you can improve the speed and robustness of tting if you try a di erent Algorithm setting, such as GMRES, rst and then vary the tolerance and number of iterations. The Algorithm setting speci es which linear solver to use in solving the linear system of equations for the interpolation.

Global Model Class: Multiple Linear Models

The following example shows the defaults for multiple linear models. You can add linear models (as seen in the single linear model settings).

This is primarily for designing experiments using optimal designs. If you have no idea what model you are going to fit, you would choose a space-filling design. However, if you have some idea what to expect, but are not sure exactly which model to use, you can specify several possible models here. The Design Editor can average optimality across each model.

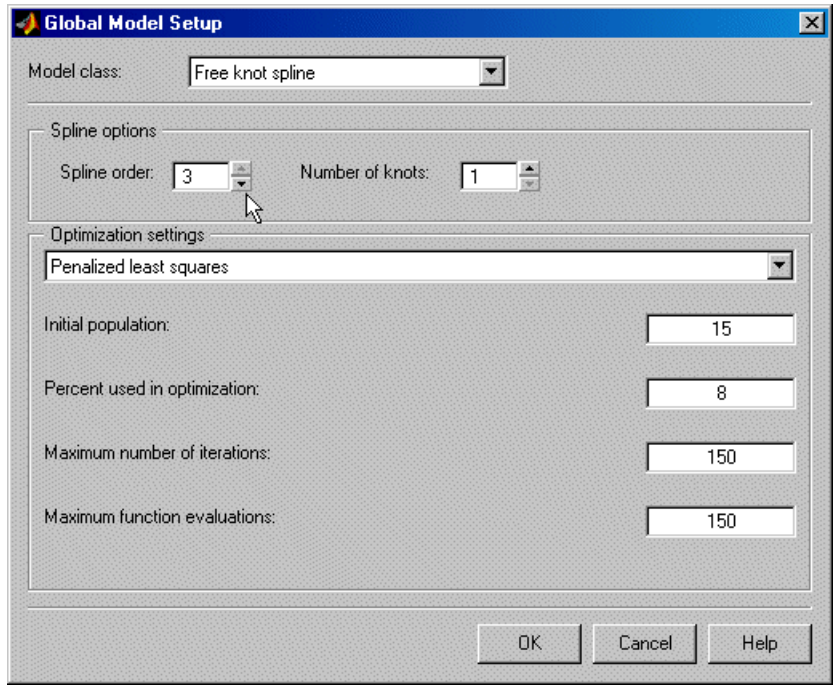
For example, if you expect a quadratic and cubic for two factors but are unsure about a third, you can enter several alternative polynomials here. You can change the weighting of each model as you want (for example, 0.5 each for two models you think equally likely). The optimization process in the Design Editor uses the weighting.

The model that appears in the model tree is the one you select, listed as Primary model. Click the model in the list, then click Use Selected. The Primary model changes to the desired model. If you do not select a primary model, the default is the first in the list.

When the model has been fitted, you can view the primary model at the global node. To compare the fit of all the alternatives, click Create Alternatives in the toolbar, select Multiple Linear Models in the dialog box, and click OK. One of each model is then built as a selection of child nodes.

See also "Polynomials" on page 5-63, and "Edit Point-by-Point Model Types" on page 6-32.

Global Model Class: Free Knot Spline



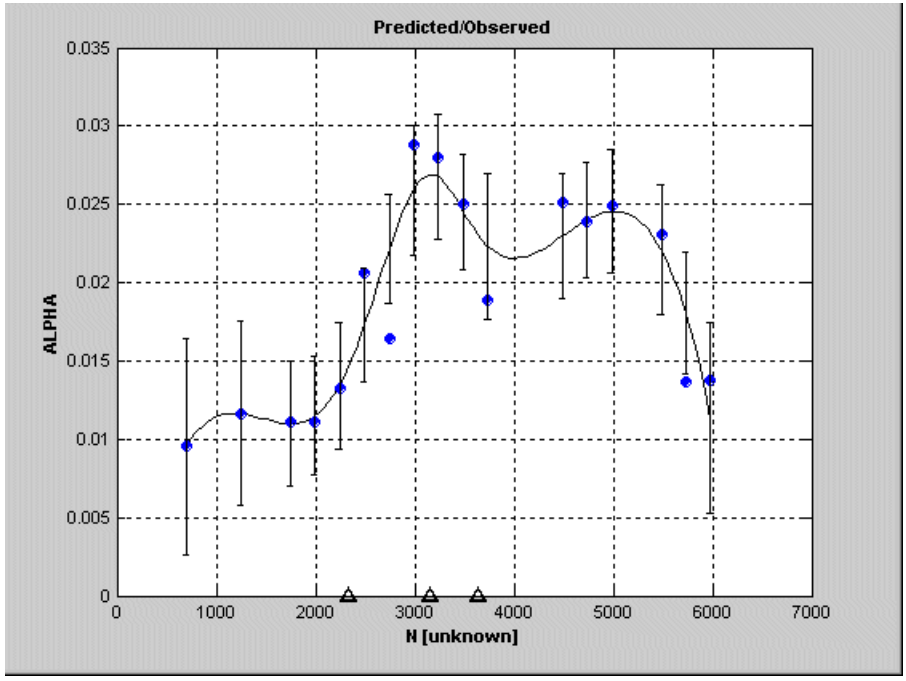
This option is only available for global (and local) models with only one input factor. See also Hybrid Splines on page 5-66 for a description of splines. The major di erence is that you choose the position of the knots for hybrid splines; here the optimal knot positions are calculated as part of the tting routine.

You can set the number of knots and the spline order can be between one and three.

There are three di erent algorithms under Optimization settings: `Penalized least squares`, `Genetic algorithm`, and `Constrained least squares`.

For all three methods, you can set the Initial population. This is the number of initial guesses at the knot positions. The other settings put limits on how long the optimization takes.

The following example shows a free knot spline model with three knots. The position of the knots is marked on the N axis.



See also the local models involving splines:

Polynomial Spline on page 5-9

Local Model Class: Truncated Power Series on page 5-13

Local Model Class: Free Knot Spline on page 5-14

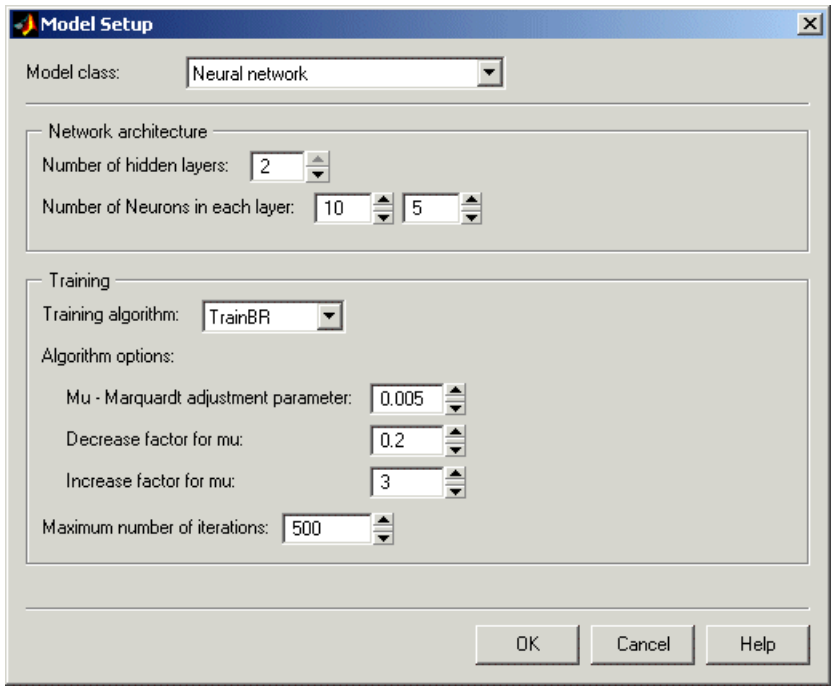
Global Model Class: Neural Network

Neural network models require the Deep Learning Toolbox product. If any of your global models are neural nets, you cannot use MLE (maximum likelihood estimation) for your two-stage model.

Neural nets contain no preconceptions of the model shape, so they are ideal for cases with low system knowledge. They are useful for functional prediction and system modeling where the physical processes are not understood or are highly complex.

The disadvantage of neural nets is that they require much data to give good con dence in the results, so they are not suitable for small data sets. Also, with higher numbers of inputs, the number of connections and hence the complexity increase rapidly.

MBC provides an interface to some of the neural network capability of the Deep Learning Toolbox product. Therefore these functions are only available if the Deep Learning Toolbox product is installed.



For help on the neural net models implemented in the Model-Based Calibration Toolbox product, see the Deep Learning Toolbox documentation. At the MATLAB command-line, enter

```
>>doc nnet
```

The training algorithms available in the Model-Based Calibration Toolbox product are `traingdm, trainlm, trainbr`.

These algorithms are a subset of the ones available in the Deep Learning Toolbox product. (The names indicate the type: gradient with momentum, named after the two authors, and

Bayesian reduction). Neural networks are inspired by biology, and attempt to emulate learning processes in the brain.

Global Model Class: User-De ned and Transient Models

These models can be local, global, or one-stage models. For set up information see Local Model Class: User-De ned Models on page 5-23 and Local Model Class: Transient Models on page 5-30.

Add Response Models and Datum Models

In this section...

Adding New Response Models on page 5-81
Datum Models on page 5-81

Adding New Response Models

Use the Fit models common task to set up response models.

To add new response models to an existing test plan, in the Test Plan tab view, double-click the Responses output port of the block diagram on the test plan tab, or select File > New Response Model.

The Response Model Setup dialog box has a list box containing all the variables in the selected data set except the inputs to the local and global models; you cannot use an input also as a response.

You can also change the local and global models also by clicking the Set Up buttons to open the Local Model and Global Model Setup dialog boxes (see Explore Local Model Types on page 5-7 and Explore Global Model Types on page 5-62). You can add Datum Models on page 5-81 (maximum or minimum) if the local model supports this.

You can return to the local or global setup options individually at any time by double-clicking the block in the test plan diagram.

In the local model view, in the Common Tasks pane, you can click Edit Model. In the Local Model Setup dialog box, select the Response Features tab, and the Name list shows available response features.

Datum Models

A datum model tracks the maximum or minimum of the local models. This is equivalent to adding the maximum or minimum as a response feature, which can be useful for analysis if those points are interesting from an engineering point of view.

If you are modeling spark sweeps with a datum model, use the work ow in Fit a Two-Stage Model on page 1-10. In the Fit Models dialog box, do not de ne responses at the project level. Instead click OK to nish. To set up your datum model and local model type,

use the Fit Models common task at the test plan node. In the Fit Models Wizard, on the Response Models screen, set up your local model and add a datum. Datum models are only available for some local models polynomial splines and polynomials (but see Linked datum models following). Other local models cannot have a datum model, because they do not necessarily have a unique turning point.

You can also choose a datum model when setting up a new response model.

The Datum options are

None

Maximum This can be useful in cases using polyspline modeling of torque against spark. The maximum is often a point of engineering interest.

Minimum - This can be useful for cases where the object is to minimize factors such as fuel consumption or emissions.

Linked datum model This is only available to subsequent two-stage models within a test plan in which the rst two-stage model has a datum model de ned. In this case, you can use that datum model. The linked datum option carries the name of the response of the rst two-stage model, where it originated.

If the maximum and minimum are at points of engineering interest, like MBT or minimum fuel consumption, you can add other response features later using the datum model (for example, MBT plus or minus 10 degrees of spark angle) and track these across local models too. It can be useful to know the value of MBT when modeling exhaust temperature, so you can use a linked datum model from a previous torque/spark model. Having responses relative to datum means that the response features are more likely to relate to a feature within the range of the data points.

You can also export the datum model along with local, global, and response models.

Fitting Process for Polynomial Splines With a Datum Model

The tting process for a polynomial spline with a maximum datum is:

1 The toolbox ts a quadratic polynomial to the data.

2 The toolbox nds the x-location of the maximum of this polynomial (if it does not have a maximum, then the model will not be tted).

3 The toolbox uses this x-value as a starting point in an optimization to nd the best knot position for the polynomial spline. Note this optimization does not have any constraint that Bhigh2 stays negative.

4 The toolbox checks the result to see if the new knot position is still at the maximum of the curve. If so, then nish.

If not, then the algorithm returns to the quadratic polynomial tted in step 1, which has the required maximum.

Create Alternative Models to Compare

In this section...

Build a Selection of Models on page 5-84
Create a Model Template on page 5-85
Polynomial Template on page 5-86
RBF Template on page 5-87
Hybrid RBF Template on page 5-87
Free Knot Spline Template on page 5-88
Gaussian Process Template on page 5-88
Model Browser Template on page 5-88

Build a Selection of Models

After you have tted and examined a single model (either one- or two-stage or point-by-point), you will normally want to create more models to search for the best t.

To create alternative models, in the Model Browser model views, use the Common Tasks links.

Use the Create Alternatives ( ) option to build a selection of alternative models to compare. Find Create Alternatives in the Common Tasks pane, the toolbar, or the Model menu.

1 From any global model node (before calculating MLE), click Create Alternatives in the Common Tasks pane.

2 In the Model Template dialog box, select a template to build a selection of models. There are prede ned templates for polynomials, radial basis functions, hybrid radial basis functions, and Gaussian process models. You can also create your own templates of any models you choose, or use model types in the current project.

   To build a selection of model types, click New, then click OK.

3 Observe the default list of several models. Add models if desired, then click OK.

4 In the Model Selection dialog box, select the criterion for choosing the best child node e.g., PRESS RMSE, and click OK.

The toolbox builds the models and selects the best using your selection criteria.

Note The toolbox builds models in parallel if you have Parallel Computing Toolbox.

5 Assess all the ts in the Alternative Models list in case you want to choose an alternative as a better t.

Note After you start building models from any template, you can always click Stop to abort model building if the process is taking too long.

You can also create individual new models. Create child nodes by clicking the Add Model common task in modeling nodes. The Model Setup dialog box appears where you can change the type and settings. Repeat this for multiple child nodes to create a selection of di erent model types tted to the same data.

Common Tasks links for creating alternative models vary depending on the model type:

To add a selection of alternatives for one-stage models, click Create Alternatives.

For two-stage models:

To add a selection of alternatives for each response feature node, at the local node, click Build Global Models.

To add local models, at the response node, click New Local Model.

To add point-by-point models, click Edit Model and then add to the list of models.

Create a Model Template

Create a template to build several model types, and save the template for reuse.

1 From any global model node (before calculating MLE), click Create Alternatives in the Common Tasks pane.

2 In the Model Template dialog box, click New then click OK.

The Multiple Model Setup dialog box shows the default list of point-by-point model types so that you can try several models:

Poly2 with Stepwise: Min PRESS

Poly3 with Stepwise: Min PRESS

Hybrid RBF with nObs/3 centers

Gaussian process models (using defaults)

3 To add models to the list, click Add to open the Model Setup dialog box, where you can select any model type available for the number of inputs. Click OK to add the model and return to the Multiple Model Setup dialog box.

4 Click Add again to repeat the process to add as many di erent models as you like. Click Edit Model to change the settings for any models in the list.

5 When you are satis ed with the list of model types, click OK in the Multi-Model Settings dialog box. The toolbox builds your chosen selection of model types as a selection of child nodes of the currently selected model node.

6 The Model Selection dialog box appears, where you can select the criterion for choosing the best child node. Use the drop-down menu to select from the available criteria (such as from PRESS RMSE, RMSE, Box-Cox, Observations, or Parameters). You can select additional criteria to appear here using the Summary Statistics options, from the Model menu. See Summary Statistics on page 6-28. Click OK to accept the chosen criterion.

You can also save templates of models you have already built.

1 From any global or one-stage model with child nodes, select Model > Create Template. You can save the child node model types of your currently selected modeling node as a template.

2 To nd your user-de ned templates and build all those model types again for any global model you choose, use the Model Template dialog box. You can set the default directory where the toolbox looks for templates (and models, data, and projects) using File > Preferences

You can use the Browse button to nd stored templates that are not in the default folder. Select your template and click OK. The models are built and the Model Selection dialog box appears, where you can select the criterion for choosing the best child node.

Polynomial Template

Use the polynomial template in the Model Template dialog box to build several polynomials of di erent orders.

1 Select Polynomials and click OK. The Model Building Options dialog box opens where you can specify the model settings.

2 Choose the minimum and maximum order of the polynomials you want to build, and whether to use Stepwise settings. For example, if you choose 1 and 5 as the minimum and maximum polynomial order, 5 child node models are built (linear, quadratic, cubic, and so on). If you choose a Stepwise setting (e.g. Minimize PRESS), it is applied to all child models.

3 Click Build and the models are built. The Model Selection dialog box appears; select the criterion for choosing the best child node.

RBF Template

In the Model Template dialog box, you can use the RBF template to build several radial basis function models with varying maximum numbers of centers and/or di erent kernels.

1 Select RBF and click OK. The Model Building Options dialog box opens where you can specify the model settings.

2 Enter a vector in the edit box to specify the maximum numbers of centers for each child model. This can be a MATLAB expression including the number of observations, e.g. 10:10:nObs/2.

3 If the current model node is an RBF, the same model settings are used by default. Click Model Settings to open the Radial Basis Function Model Settings dialog box, where you can view and change all the model parameters such as kernel and widths. See Types of Radial Basis Functions on page 7-3.

4 Select the check box Build all kernels to create child models with the speci ed range of centers for each kernel type.

5 Click Build and the models are built. The Model Selection dialog box appears, where you can select the criterion for choosing the best child node.

Hybrid RBF Template

In the Model Template dialog box, you can use the Hybrid RBF template to build several hybrid radial basis function models with varying maximum numbers of centers and/or di erent kernels.

1 Select Hybrid RBF and click OK. The Model Building Options dialog box opens where you can specify the model settings.

2 Enter a vector in the edit box to specify the maximum numbers of centers for each child model. This can be a MATLAB expression including the number of observations, e.g. 10:10:nObs/2.

3 If the current model node is a hybrid RBF, the same model settings are used by default. Click Model Settings to open the Hybrid RBF Model Settings dialog box, where you can view and change all the model parameters such as kernel and widths (and the order of the polynomial part of the model on the Linear Part tab). See Hybrid Radial Basis Functions on page 7-29.

4 Select the check box Build all kernels to create child models with the specified range of centers for each kernel type.

5 Click Build and the models are built. The Model Selection dialog box appears, where you can select the criterion for choosing the best child node.

Free Knot Spline Template

In the Model Template dialog box, you can use the Free Knot Spline template to build several free knot spline models of different numbers of knots. Only available for models with a single input factor.

1 Select Free Knot Spline and click OK. The Model Building Options dialog box opens where you can specify the model settings. If the current model node is a free knot spline, the same model settings are used by default.

2 Choose the initial and final number of knots. For example, if you specify the initial and final numbers of knots as 1 and 5, five child nodes are built, one with one knot, one with two.

3 Click Build and the models are built. The Model Selection dialog box appears, where you can select the criterion for choosing the best child node.

Gaussian Process Template

In the Model Template dialog box, select the Gaussian Process template to build a selection of Gaussian process models. You can choose to build all kernel functions and basis functions, or select a kernel or basis function.

Model Browser Template

In the Model Template dialog box, you can use the Model Browser template to build a copy of an existing set of child model types in the current project.

1 Select Model Browser and click OK.

2 The Model Tree dialog box opens. Select a model from the model tree that has the child node model types you want to build. You can also use this template to select a local multiple model node to copy. Click OK to return to the Model Template dialog box.

3 Click Build and the models are built. The Model Selection dialog box appears, where you can select the criterion for choosing the best child node.

Selecting Models

This section discusses the following topics:

Assess High-Level Model Trends on page 6-2
Assess Local Models on page 6-5
Assess One-Stage Models on page 6-16
Compare Alternative Models on page 6-25
Assess Point-by-Point Models on page 6-31
Model Selection Window on page 6-35
Guidelines for Selecting the Best Model Fit on page 6-54
Assess Two-Stage Models on page 6-59
Model Evaluation Window on page 6-60
Stepwise Regression on page 6-65
Box-Cox Transformation on page 6-77
Two-Stage Models for Engines on page 6-81
Export Models to Simulink on page 6-94
Export Models to the Workspace on page 6-99

Assess High-Level Model Trends

After tting models in the Model Browser, to assess high-level model trends, use the Response Models plots at the test plan node.

After you t models, the view at the test plan node displays the Response Models tab. If you are two-stage modeling you must create a two-stage model to see the Response Models plots.

View the cross-section plots of all your response models. You can use the following options:

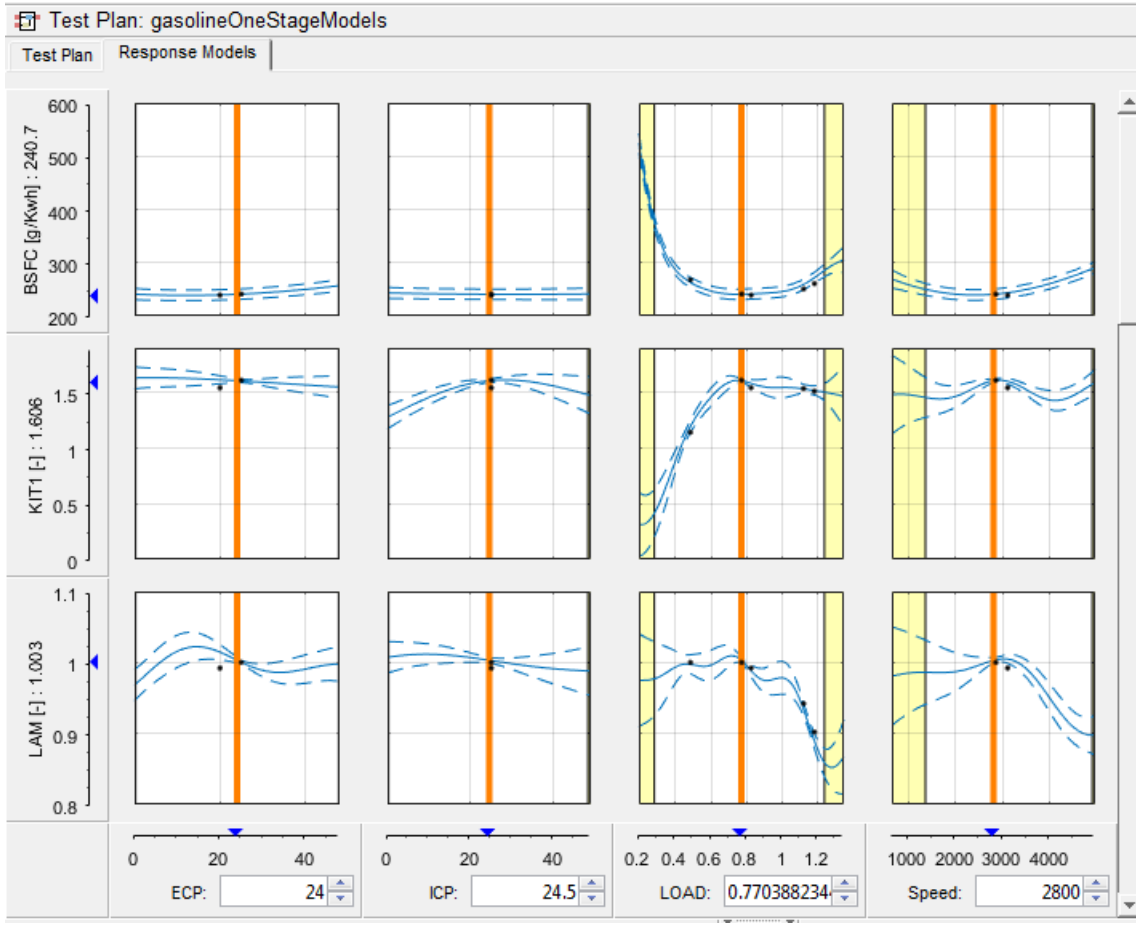
To select a value of an input, either drag the orange line in a plot, change the value in the edit box under the graph, or change the value in the Value box on the right.

To change the range for displaying data points within tolerance of the input values, edit the Tolerance boxes on the right.

To set the axes limits to the boundary constraint boundary, select the Zoom to boundary constraint check box. This zooms the plots inside the constraint boundaries, so you can explore models in valid regions only.

The plots display con dence levels, common Y limits, boundary constraints, and data points within the Tolerance values. Edit these settings using the controls on the right.

When you are displaying a point-by-point model, select the operating point to display using the Test controls. For point-by-point models, the plots show only the cross-sections of the local inputs.



See Also

Related Examples

Assess One-Stage Models on page 6-16

Assess Point-by-Point Models on page 6-31

Assess Local Models on page 6-5

Assess Two-Stage Models on page 6-59

More About

What Models Are Available? on page 5-2

Assess Local Models

In this section...
How to Assess Local Models on page 6-5 Using Local Model Plots on page 6-6 Removing Outliers and Updating Fits on page 6-8 Create Two-Stage Models on page 6-9 Create Alternative Local and Global Models on page 6-11 Viewing Local Model Statistics on page 6-11

How to Assess Local Models

After tting models using a two-stage test plan in the Model Browser, you must assess local models, then global models, and then create the two-stage model. When you select a local node (with the icon) in the Model Browser tree, the local level view appears. At the local level you can:

- View local model plots and statistics, and scroll through all local models test by test. See Using Local Model Plots on page 6-6, and Viewing Local Model Statistics on page 6-11.
- Look for problem tests with the RMSE Plots. See Using the RMSE Plot with Local Models on page 6-7.
- You can use the Test Notes pane to record information on particular tests.
- Remove and restore outliers and update ts. See Removing Outliers and Updating Fits on page 6-8.
- Calculate two-stage models, and add or remove response features. After calculating the two-stage model, you can compare the local t and the two-stage t on the local level plots. See Create Two-Stage Models on page 6-9.

Note that after the two-stage model is calculated the local node icon changes to a two-stage icon ( ) to re ect this. The response node also has a two-stage icon, but produces the response level view on page 6-59 instead.

Using Local Model Plots

Sweep Plot

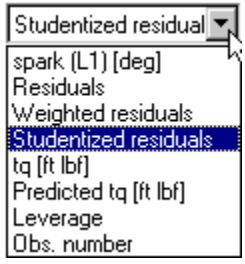
You can scroll through all the local models by using the up and down test buttons, type directly in the edit box, or go directly to test numbers by clicking Select Test.

The sweep plot shows the local model fit to the data for the current test only, with the datum point if there is a datum model. If there are multiple inputs to the local model, a predicted/observed plot is displayed. In this case to examine the model surface in more detail you can use Model > Evaluate. See Model Evaluation Window on page 6-60.

To examine the local fit in more detail, close other plots, or zoom in on parts of the plot by Shift-click-dragging or middle-click-dragging on the place of interest on the plot. Return to full size by double-clicking.

Diagnostic Statistics Plot

The Diagnostic Statistics plot can show various scatter plots of statistics for assessing goodness-of-fit for the current local model shown. The statistics available for plotting are model dependent.



The preceding is an example drop-down menu on the scatter plot for changing x and y factors. In this case `spark` is the local input factor and `torque` is the response. The local inputs, the response, and the predicted response are always available in these menus. The observation number is also always available.

The other options are statistics that are model dependent, and can include residuals, weighted residuals, studentized residuals, and leverage. At local level these are internally studentized residuals.

Using the RMSE Plot with Local Models

Use the RMSE Plot to quickly identify problem tests and navigate to a test of interest. The plot shows the standard errors of all the tests, both overall and by response feature. Navigate to a test of interest by double-clicking a point in the plot to select the test in the other plots in the local model view.

The plot displays one value of standard error per test, overall and for each response feature. As a best practice, first plot RMSE against test number to get an idea of how the error is distributed and locate any tests with much higher errors. Right-click to toggle display of test numbers. Ideally, all the standard errors should be roughly the same value to satisfy the statistical assumptions for two-stage models. If these assumptions are not satisfied, error estimates for two-stage models may not be valid.

You can also use the X- and Y-axis factor drop-down lists to plot these standard errors against the global variables to examine the global distribution of error.

Additional Plots

You can add or change plots by clicking the toolbar buttons, split buttons in plot title bars, or selecting an option from Current View in the context menu or View menu. You can add:

Data Plots View plots of the data for the current test. Select View > Plot Variables to choose variables to plot. You can choose to view any of the data signals in the data set for the current test (including signals not being used in modeling). You can plot a pair of variables or plot a variable against record number. You can add more data plots if you want.

Note You can also view values of global variables in the Global variables pane.

`Normal Plot` Normal plots are a useful graph for assessing whether data comes from a normal distribution. For more information, see Normal Probability Plots (Statistics and Machine Learning Toolbox).

`Validation Data` If you are using validation data, the plot shows the local model validation residuals if there is validation data for the current test (the global variables must match). If there is a two-stage model, the two-stage validation residuals are also shown. Validation data must be attached at the Edit Test Plan Definition on page 2-5. See Using Validation Data on page 6-61.

`Model Definition` View the parameters and coefficients of the model formula and the scaling details.

Removing Outliers and Updating Fits

Removing and Restoring Outliers on page 6-8
Updating Other Fits on page 6-8

Removing and Restoring Outliers

You can use the right-click context menus on plots or the Outliers menu to remove and restore outliers. For available options, see Remove and Restore Outliers on page 6-18.

Local models have an additional option to remove a whole test: Outliers > Remove All Data. This option leaves the current local model with no data, so entirely removes the current test. This test is removed from all the global models.

Updating Other Fits

When you remove an outlier from your local model, it re ts immediately. Other dependent ts also need updates. You can choose when to update the other ts. Removing an outlier can a ect several other models. Removing an outlier from a best local model changes all the response features for that two-stage model. The global models all change; therefore the two-stage model must be recalculated. For this reason the local model node returns to the local (house) icon and the response node becomes blank again. If the two-stage model has a datum model de ned, and other models within the test plan are using a datum link model, they are similarly a ected.

To update ts, either:

In the local view, use the Update Fit toolbar button to update all the dependent ts.

When you select another model node, you are prompted to update or defer updates.

When leaving the local node, a dialog box asks if you want to update all dependent ts. Click Yes to update all global models, or No to delay lengthy updates of dependent ts. Delaying updates can be useful when you want to examine only a particular global model after removing an outlier at the local level. With the defer option, you can avoid waiting while updating all other dependent ts.

If you defer updating ts and you go to a response feature node, the toolbox re ts only that node, so you can inspect that global model t. Other response features do not update unless you click them. When you return to the local node again the Update Fit button is enabled. Until you update ts, a status message at the bottom of the browser tells you that you have deferred updates.

Create Two-Stage Models

To create a two-stage model, click Create Two-Stage in the Common Tasks pane.

If your model supports it, you are prompted to calculate the two-stage model using maximum likelihood estimation (MLE). This takes correlations between response features into account.

Note You need the right number of response features to create a two-stage model. You are prompted if you need to select response features, then the toolbox creates the two-stage model.

After creating the two-stage model, compare the local fit and the two-stage fit on the local level plots.

MLE Settings

Calculating MLE: For an ordinary (univariate) two-stage model, the global models are created in isolation without accounting for any correlations between the response features.

Using MLE (maximum likelihood estimation) to fit the two-stage model takes account of possible correlations between response features.

In cases where such correlations occur, using MLE significantly improves the two-stage model.

When you click Create Two-Stage Model in the common tasks pane, and your model support MLE, then a dialog box asks if you want to calculate MLE. If you click Cancel at this point, you can calculate MLE later as follows:

1. From the local node, click the MLE icon in the toolbar .

   Alternatively, choose Model > Calculate MLE.
2. The MLE dialog box appears. Click Start.
3. After you click Start a series of progress messages appears, then a new Two-Stage RMSE (root mean square error) value is reported.
4. You can perform more iterations by clicking Start again to see how the RMSE value changes, or you can click Stop at any time.

5 Clicking OK returns you to the Model Browser, where you can view the new MLE model fit.

After calculating MLE, notice that the plots and the icons in the model tree for the whole two-stage model (response node, local node, and all response feature nodes) have turned purple.

You can select all response features in turn to inspect their properties graphically; the plots are all purple to symbolize MLE. At the local node the plots show the purple MLE curves against the black local fit and the blue data.

From the response feature nodes, at any time, click the MLE toolbar icon to recalculate MLE and perform more iterations.

From the local node, you can open the Model Selection on page 6-36 window to compare the MLE model with the previous univariate model (without correlations), and choose the best. Here you can select the univariate model and click Assign Best to undo MLE and return to the previous model.

MLE dialog box settings:

Algorithm

The algorithm drop-down menu offers a choice between two covariance estimation algorithms, Quasi-Newton and Expectation Maximization. These are algorithms for estimating the covariance matrix for the global models.

Quasi-Newton is recommended for smaller problems (< 5 response features and < 100 tests). Quasi-Newton usually produces better answers (smaller values of -logL) and hence is the default for small problems.

Expectation Maximization is an iterative method for calculating the global covariance (as described in Davidian and Giltinan (1995); see References in Two-Stage Models for Engines on page 6-81). This algorithm has slow convergence, so you might want to use the Stop button.

Tolerance

You can edit the tolerance value. Tolerance is used to specify a stopping condition for the algorithm. The default values are usually appropriate, and if calculation is taking too long you can always click Stop.

Initialize with previous estimate

When you recalculate MLE (that is, perform more iterations), there is a check box you can use to initialize with the previous estimate.

Predict missing values

The other check box (selected by default) predicts missing values. When it is selected, response features that are outliers for the univariate global model are replaced by the predicted value. This allows tests to be used for MLE even if one of the response features is missing. If all the response features for a particular test are missing or the check box is unselected, the whole test is removed from MLE calculation.

Create Alternative Local and Global Models

To quickly build a selection of alternative global models to compare, in the Common Tasks pane, click Build Global Models. This opens the Model Template dialog box. For local models, you build a selection of child nodes for each response feature node. See Create Alternative Models to Compare on page 5-84 for details.

The toolbox selects the best model for each response feature, based on your selection criteria (such as AICc). Assess all the ts in case you want to choose an alternative.

To change the current local model type, in the Common Tasks pane, click Edit Model. This opens the Model Setup dialog box, where you can choose another model type. See Explore Local Model Types on page 5-7.

Model > Fit Local Opens the Local Model Fit Tool dialog box. If you are covariance modeling, you can choose three algorithms: REML (Restricted Maximum Likelihood - the default), Pseudo-likelihood, or Absolute residuals. Using the Fit button might take several steps to converge, but if you use the One Step button only one step in the optimization process is taken. Every time you run a process, the initial and nal Generalized Least Squares parameter values are displayed for each iteration.

Without covariance modeling, you can only click Fit. The Ordinary Least Squares (OLS) parameters are displayed. Click Fit to rerun. You can enter a di erent change in parameters in the edit box.

After creating alternative models, for next steps, see Compare Alternative Models on page 6-25.

Viewing Local Model Statistics

Local Statistics Pane on page 6-12

Pooled Statistics on page 6-12

Local Statistics Pane

You can select the information to display in the Local Statistics pane.

Summary statistics RMSE for the current test, number of observations, degrees of freedom on the error, R squared, Cond(J) (the condition index for the Jacobian matrix).

Note Check for high values of Cond(J) (e.g., > 10^8). High values of this condition indicator can be a sign of numerical instability.

If there is validation data for the current test, Validation RMSE for the current test also appears here.

Parameters Shows the values and standard errors of the parameters in the local model for the current test selected.

Correlations Shows a table of the correlations between the parameters.

Response Features Shows the values and standard errors of the response features de ned for this local model, from the current test (often some or all of them are the same as the parameters; others are derived from the parameters).

Global Covariance For MLE models, shows a covariance matrix for the response features at the global level.

The Global variables pane shows the values and standard errors of the global variables at the position of the current test.

Pooled Statistics

These are seen at the local node (when two-stage modeling) in the Pooled Statistics table, and at the response node in the list of local models. If you have a selection of local or two-stage models, use these statistics to help you choose which model is best.

Local RMSE	Root mean squared error between the local model and the data for all tests. The divisor used for RMSE is the number of observations minus the number of parameters.

Two-Stage RMSE	Root mean squared error between the two-stage model and the data for all tests. You want this error to be small for a good model fit.
PRESS RMSE	Root mean squared error of predicted errors, useful for indicating overfitting; see PRESS statistic on page 6-74. The divisor used for PRESS RMSE is the number of observations. Not displayed for MLE models because the simple univariate formula cannot be used.
Two-Stage T^2	T^2 is a normalized sum of squared errors for all the response features models. You can see the basic formula on the Likelihood view of the Model Selection window. $T^2 = (y_{rf} - \hat{y})^T \Sigma^{-1} (y_{rf} - \hat{y})$ Where $\Sigma = \text{blockdiag}(C_i + D)$, where C_i is the local covariance for test i. See blockdiag diagram following. A large T^2 value indicates that there is a problem with the response feature models.

-log L	Log-likelihood function: the probability of a set of observations given the value of some parameters. You want the likelihood to be large, tending towards -in nity, so large negative is good. For n observations $x_1, x_2, .. x_n$, with probability distribution $f(x, \theta)$, the likelihood is: $L = \prod_{i=1}^{n} f(x_i, \theta)$ This is the basis of MLE. See Create Two-Stage Models on page 6-9. $\log L = \log(\det(\Sigma)) + (y_{rf} - \hat{y})^T \Sigma^{-1} (y_{rf} - \hat{y})$ which is the same as: $\log L = \log(\det(\Sigma)) + T^2$ This assumes a normal distribution. You can view plots of -log L in the Model Selection window, see Likelihood View on page 6-44.
Validation RMSE	Root mean squared error between the two-stage model and the validation data for all tests.

To explain blockdiag as it appears under T^2 in the Pooled statistics table:

$\Sigma = \text{blockdiag}(C_i + D)$, where C_i is the local covariance for test i, is calculated as shown below.

$$blockdiag(C_i + D) = \begin{bmatrix} C_1 + D & & & \\ & C_2 + D & & \\ & & C_3 + D & \\ & & & C_i + D \end{bmatrix}$$

See Also

Related Examples

Assess High-Level Model Trends on page 6-2
Assess One-Stage Models on page 6-16
Compare Alternative Models on page 6-25

Assess One-Stage Models

In this section...

Assessing One-Stage, Response Feature or Global Models on page 6-16
Assess Fits Using Model Plots on page 6-16
Remove and Restore Outliers on page 6-18
Model-Speci c Tools on page 6-22
Create Alternative Models on page 6-23

Assessing One-Stage, Response Feature or Global Models

After you t models in the Model Browser, use the model views to assess ts. When you select a one-stage model node (or a response feature node when two-stage modeling) in the All Models tree, or any child nodes of these models, you see the global model view.

These kinds of models all have a global icon ( ), so this is referred to as global level. To learn more about model types, see What Models Are Available? on page 5-2. Plot settings are shared between all global models in your test plan. For example, if you select a contour plot and some variables to plot in the Response Surface plot, you see the same plots when you switch to another global model in your test plan.

Use the plots and statistics to assess ts, and use the Common Tasks pane to build alternative models to compare.

Assess Fits Using Model Plots

Use the plots to assess model ts. It can be helpful to click to highlight an outlier so that you can view the same point highlighted in other plots. You can display test numbers using the context menu. You can remove outliers with the plot context menus. See Remove and Restore Outliers on page 6-18.

Response Surface Plot

This view shows the model surface in a variety of ways. The default view is a 3-D plot of the model surface.

You can choose which input factors to display by using the drop-down menus left of the plot. The unselected input factors are held constant and you can change their values

using the controls at the left of the plot (either by clicking the arrow buttons or by typing directly in the edit box). Click Select Data Point to choose a point to plot.

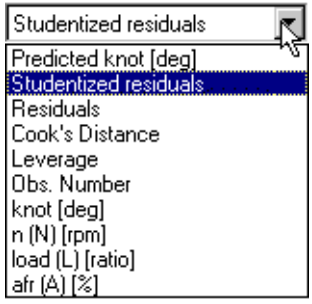
To plot the region inside the boundary model only, right-click and select Zoom to Boundary.

Select the Plot list to switch to a Line, Contour, or Multiline plot.

Diagnostic Statistics Plot

The lower default plot is the Diagnostic Statistics plot, that shows various scatter plots of statistics for assessing goodness-of- t for the current model.

The statistics and factors available for plotting are model dependent. Choose the x- and y-axis factors using the drop-down menus. Following is an example.



In this example knot is the response feature node selected. The model output is knot, so knot and Predicted knot are available in the menu. (For child nodes of knot, the model output is still knot.) The global inputs, the model output, the predicted model output and the observation number are always available.

The other options are statistics that are model dependent, and can include: Residuals, Weighted Residuals, Studentized Residuals, Leverage, and Cook's Distance. You can use any of these as criteria for selection of outliers, see Remove and Restore Outliers on page 6-18. At global (or one-stage) level these are externally studentized residuals.

Additional Plots

You can add or change plots by clicking the toolbar buttons, split buttons in plot title bars, or selecting an option from Current View in the context menu or View menu. The browser remembers your layout per testplan. You can add:

`Predicted/Observed` plot. Where there is only one input factor, the plot shows the model fit and the data against the input factor.

When there is more than one input factor it becomes impossible to display the fit in the same way, so the data for the response feature is plotted against the values predicted by the global model. The line of predicted=observed is shown. With a perfect fit, each point would be exactly on this line. The distances of the points from the line (the residuals) show how well the model fits the data.

To examine plots in more detail, close other plots, or zoom in on parts of the plot by Shift-click-dragging or middle-click-dragging on the place of interest on the plot. Return to full size by double-clicking.

Note When two-stage modeling, for response feature models, each data point is the value taken by this response feature for some local model fit (of this two-stage model). Note that response features are not necessarily coefficients of the local curves, but are always derived from them in some way. Right-click a point in the plots to open a figure plot of that particular test.

`Normal Plot` Normal plots are a useful graph for assessing whether data comes from a normal distribution. For more information, see Normal Probability Plots (Statistics and Machine Learning Toolbox).

`Validation Data` For one-stage models. If you are using validation data, the plot shows the one-stage model validation residuals. Validation data must be attached during model setup or at the Edit Test Plan Definition on page 2-5. See Using Validation Data on page 6-61.

`Model Definition` View the parameters and coefficients of the model formula and the scaling details.

For any radial basis function model you can see the kernel type, number of centers, width, and regularization parameter. For radial basis function models, see also How to Find RBF Model Formula on page 7-36.

Remove and Restore Outliers

To remove and restore outliers, you can use the right-click context menus on plots (except the Response Surface and Validation Data) or the Outliers menu. The toolbox outlines in red possible outliers, where studentized residuals >3. You can remove outliers in the Diagnostic Statistics, Normal, and Predicted/Observed plots.

When you remove an outlier from your model, it re ts immediately.

Apply to All Responses Select to apply Clear Outliers, Remove Outliers, or Restore Removed Data to all response models.

Clear Outliers Returns all data points to the unselected state as possible outliers.

Remove Outliers Removes red-outlined data points from the t. Re ts the current local t only. Use the Update Fit toolbar button or Model > Update Fit to also re t the global models. You can update or defer when another node is selected.

Restore Removed Data Opens the Restore Removed Data dialog box, where you can choose the points to restore from the list by record number, or restore all available points. Select points in the left list and use the buttons to move points between the lists. Clicking OK re ts the model, including all restored data.

If you are two-stage modeling and have removed an entire test at the local level, or you could not re t the local model, then you see tests marked with an asterisk (*) in the Removed Data pane. Go to the local level to restore removed tests. Double-click on any removed test number to display a plot of the test in a gure window.

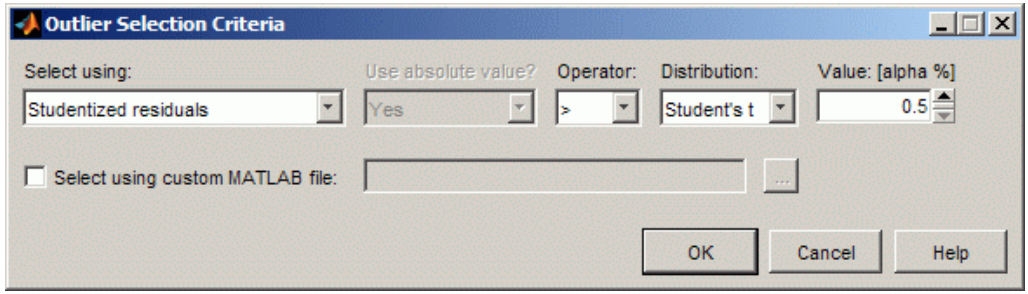
Copy Outliers From Opens the Copy Outliers dialog box, where you can choose which outliers to copy. Select a model (of the same type local or global) in the tree and click OK, and the current model (and other models a ected) are re tted using the outlier selections for that model.

Selection Criteria Opens the Outlier Selection Criteria dialog box where you can set the criteria for the automatic selection of outliers. Disabled for MLE models.

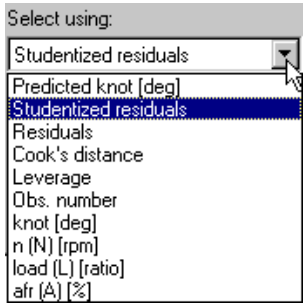
Specify Automatic Outlier Selection Criteria

You can change the criteria for automatically selecting outliers. You can specify outliers as satisfying a condition on the value of some statistic (for example, residual > 3), or select those points that fall in a region of the distribution of values of that statistic.

For example, assume that residuals are normally distributed and select those with p-value > 0.9. You can also select outliers using the values of model input factors.



To change the criteria, select the drop-down list `Select using` and view the available criteria.



The options available change depending on the type of model currently selected. The options are the same as those found in the drop-down menus for the x- and y-axis factors of the scatter plots in the Model Browser.

In the preceding example, the model selected is the knot response feature, so `knot` and `Predicted knot` appear in the criteria list, plus the global input factors; and it is a linear non-MLE model, so `Cook's Distance` and `Leverage` are also available.

The range of the selected criteria (for the current data) is indicated above the `Value` edit box, to give an indication of suitable values. You can type directly in the edit box. You can also use the up/down buttons on this box to change the value (incrementing by about 10% of the range).

You can use the `Distribution` drop-down menu to remove a proportion of the tail ends of the normal or t distribution. For example, to select residuals found in the tails of the distribution making up 10% of the total area:

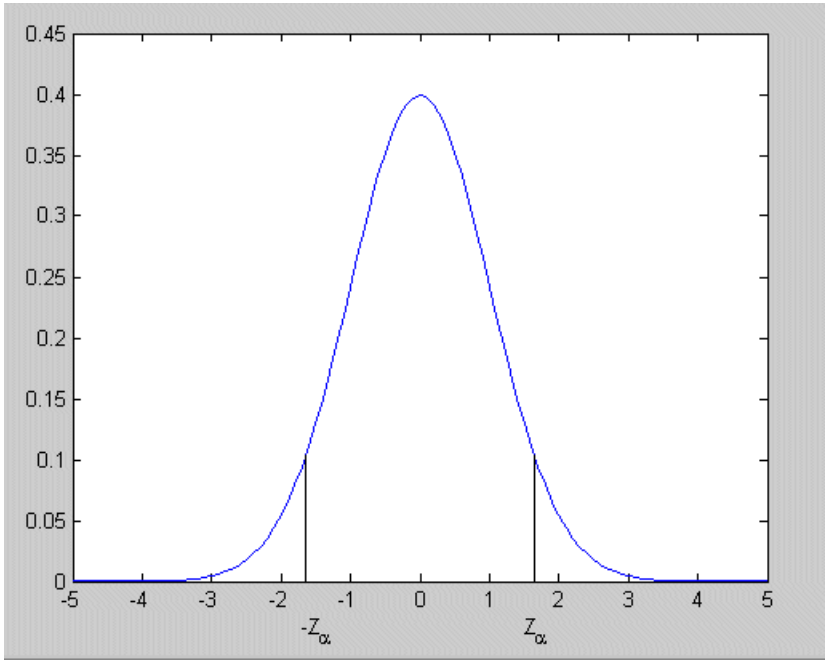
Select Normal in the Distribution drop-down menu.

Select the operator >.

Enter 10 as the α% value in the edit box.

Residuals found in the tails of the distribution that make up 10% of the total area are selected. If you had a vast data set, approximately 10% of the residuals would be selected as outliers.

As shown, residuals found beyond the value of Z_α in the distribution are selected as outliers. α is a measure of signi cance; that is, the probability of nding residuals beyond Z_α is less than 10%. Absolute value is used (the modulus) so outliers are selected in both tails of the distribution.



The t distribution is used for limited degrees of freedom.

If you select None in the Distribution drop-down menu, you can choose whether or not to use the absolute value. That is, you are selecting outliers using the actual values rather than a distribution. Using absolute value allows you to select using magnitude only

without taking sign into account (for example, both plus and minus ranges). You can select No here if you are only interested in one direction: positive or negative values, above or below the value entered. For example, selecting only values of speed below 2000 rpm.

The Select using custom MATLAB file check box enables the adjacent edit box. Here you can choose a function file that selects outliers. Type the name of the file and path into the edit box, or use the browse button.

In this file you define a MATLAB function of the form:

```
function outIndices = funcname (Model, Data, Names)
```

`Model` is the current MBC model.

`Data` is the data used in the scatter plots. For example, if there are currently 10 items in the drop-down menus on the scatter plot and 70 data points, the data make up a 70 x 10 array.

`Names` is a cell array containing the strings from the drop-down menus on the scatter plot. These label the columns in the data (for example, spark, residuals, leverage, and so on).

The output, `outIndices`, must be an array of logical indices, the same size as one column in the input `Data`, so that it contains one index for each data point. Those points where index = 1 in `outIndices` are highlighted as outliers; the remainder are not highlighted.

Model-Specific Tools

Linear Model and Multiple Linear Models

You can find model-specific tools in the Model > Utilities menu or in the toolbar.

Stepwise This opens the Stepwise Regression window, where you can view the effects of removing and restoring model terms on the PRESS statistic (Predicted Error Sum of Squares), which is a measure of the predictive quality of a model. You can also use Min PRESS to remove all at once model terms that do not improve the predictive qualities of the model. See "Stepwise Regression" on page 6-65 for further discussion of the statistical effects of the Stepwise feature.

Design Evaluation Opens the Design Evaluation tool, where you can view properties of the design. See "Design Evaluation Tool" on page 3-57.

Prediction Error Variance Viewer - Opens the Prediction Error Variance Viewer. See Prediction Error Variance Viewer on page 3-50.

Radial Basis Function Models

Radial basis function models have some model-specific toolbar buttons.

Update Fit refits the RBF widths and centers. See Tips for Modeling with Radial Basis Functions on page 7-31 and Fitting Routines on page 7-11 in the Radial Basis Functions chapter.

View Centers opens a dialog box where you can view the position of the radial basis function's centers graphically and in table form.

Prune opens the Number of Centers Selector where you can minimize various error statistics by decreasing the number of centers. See Prune Functionality on page 7-24.

Hybrid RBFs have the same toolbar buttons as linear models.

MLE Models

If you are viewing an MLE model and have removed outliers, you can recalculate MLE using the toolbar button. This opens the MLE dialog box, where you can perform more iterations to try to refine the MLE model fit. See Create Two-Stage Models on page 6-9 for more details.

Create Alternative Models

After you have fitted and assessed a single model fit, you will often want to create more models to search for the best fit.

To quickly build a selection of alternative models to compare, in the Common Tasks pane, click Create Alternatives. See Create Alternative Models to Compare on page 5-84 for details.

After you create a variety of models to compare, the alternative models list appears at the top. The toolbox selects the best model, based on your selection criteria (such as AICc). Assess all the fits in case you want to choose an alternative.

The table lists the child models of the currently selected model, the number of parameters and observations, and the summary statistics for each model. Compare the child models and choose the best by selecting the Best Model check box.

Use the plots and summary statistics table to help you assess and compare ts and help you choose the best. See Assess Fits Using Model Plots on page 6-16 and Compare Fits Using Statistics on page 6-25

To change the current model type, in the Common Tasks pane, click Edit Model. This opens the Model Setup dialog box, where you can choose another model type. See Explore Global Model Types on page 5-62.

To reset to the default model type, select Model > Reset. This opens a con rmation dialog box so you cannot unintentionally reset your model. When you con rm you want to continue, the model is reset to the global model default, that is, the global model speci ed at the test plan stage, restoring any removed outliers and removing any transforms.

After creating alternative models, for next steps, see Compare Alternative Models on page 6-25.

See Also

Related Examples

Assess High-Level Model Trends on page 6-2

Compare Alternative Models on page 6-25

More About

What Models Are Available? on page 5-2

Compare Alternative Models

In this section...

Compare Fits Using Model Plots on page 6-25
Compare Fits Using Statistics on page 6-25
Other Model Assessment Windows on page 6-29

When you use MBC Model Fitting app to t statistical models to experimental data, the app automatically ts models to your data. The default models can usually produce a good t, but consider assessing ts and exploring alternative models. After assessing the initial model t, use the app tools to create more models to search for the best t.

To create alternative models, in the Model Browser model views, use the Common Tasks links. After you create models to compare, you see an Alternative Models list in the model view. To help you assess each model and decide which model to choose as best, use the plots and diagnostic statistics of the Model Browser.

After you use the Model Template dialog box to create child nodes, the toolbox selects the best model from the alternatives (indicated with a blue icon) based on the selection criteria you choose. For local nodes, the best child node of each response feature is chosen.

Assess all the ts in the Alternative Models list in case you want to choose an alternative as a better t.

Compare Fits Using Model Plots

To compare the plots and statistics and select the best model, select each model in the list of alternative models. To determine the best t, you examine both the plots and statistics. Start with a visual inspection of the model plots. In the Model Browser, use the Response Model plots at the test plan node. For more information about using plots to compare ts, see Assess High-Level Model Trends on page 6-2.

Compare Fits Using Statistics

Use the summary table to assess the currently selected model. When you have a list of alternative models, use the same statistics in the list to compare the models and select the best.

The summary table can include the following information.

Statistic	Description	Assess Model Fits
Observations	Number of observations used to estimate model.	NA
Parameters	Number of parameters in model. Gaussian process models display the e ective number of parameters, so it can be noninteger.	Look for models with fewer parameters than observations. If the quantities are close, this indicates possible over tting.
PRESS RMSE	Root mean squared error of predicted errors. PRESS RMSE is a measure of the predictive power of your models (for linear models only). The divisor used for PRESS RMSE is the number of observations. The residuals are in untransformed values to enable comparison between alternative models with di erent Box-Cox transformations.	Look for lower PRESS RMSE values to indicate better ts without over tting. Compare PRESS RMSE with RMSE. If the value of PRESS RMSE is much bigger than the RMSE, then you are over tting.

Statistic	Description	Assess Model Fits
RMSE	Root mean squared error. The divisor used for RMSE is the number of observations minus the number of parameters. The residuals are in untransformed values, to enable comparison between alternative models with di erent Box-Cox transformations.	Look for lower RMSE values to indicate better ts, but beware of over tting. See PRESS RMSE.
AICc	Information criteria. Not available for Gaussian process models.	Look for lower values of AICc. Only the di erence between the AICc values for two models is meaningful, not the absolute value: if this di erence is greater than about 10, then discard the worse model.
Box-Cox	Power transform used for box-cox transformation. 1 indicates no transform. 0 indicates a log transform is used.	1 indicates no transform. 0 indicates a log transform is used.
Validation RMSE (one-stage models only)	Root mean squared error between the one-stage model and the validation data.	Look for lower values of the validation RMSE.

For linear models, use the stepwise functions (choose a Stepwise option during model setup) to re ne your models and remove less useful model terms. Make sure that you examine outliers, but do not automatically remove them without good reason.

After you evaluate your models and make your selections, export your models to CAGE for optimized calibration. From the test plan node, click Generate calibration in the Common Tasks pane.

Summary Statistics

To help you evaluate models, you can select additional statistics in the Summary Statistics dialog box. The standard summary statistics are PRESS RMSE (for linear models only) and RMSE.

1 To open the Summary Statistics dialog box:

   From any global model node, select Model > Summary Statistics.

   From the test plan, right-click the global model block and select Summary Statistics. If you want the summary statistics to apply to all the models within the test plan, use this option before building models. When you create a child node, the summary statistics inherit the test plan node or the parent node statistics.

2 Choose additional statistics by selecting the check boxes.

3 Click OK. Changes made from a global model node are applied immediately to the Summary table and the Models list pane.

<table>
<tr><th>Available Statistics</th><th>Description</th><th>Assess Model Fits</th></tr>
<tr><td>GCV

Weighted PRESS

-2LogL

R^2

R^2 adj

PRESS R^2

DW</td><td>Names and formula are in the dialog box.</td><td>In general, look for lower values. This table provides general guidance.
<table>
<tr><th>Statistic</th><th>Value</th><th>Model Fit</th></tr>
<tr><td rowspan="4">R^2

R^2 adj

PRESS R^2</td><td>Less than 0.5</td><td>Poor</td></tr>
<tr><td>Greater than 0.5 and less than 0.9</td><td>Moderate</td></tr>
<tr><td>Greater than 0.9 and less than 1.0</td><td>Good</td></tr>
<tr><td>Close to 1.0</td><td>Potentially over t</td></tr>
</table></td></tr>
<tr><td>cond(J)</td><td>Condition indicator.</td><td>High values (e.g., > 10^8) can be a sign of numerical instability.</td></tr>
</table>

Available Statistics	Description	Assess Model Fits
AIC AICc (small sample) BIC	Information criteria. Available for comparison only if the same data is used for all models. Not available for Gaussian process models.	The di erences between two information criteria are of interest, not the absolute value. As a rule of thumb, a di erence of 2 or less means models are roughly the same quality of t. All the information criteria impose a penalty for over tting, which increases linearly with the number of parameters in the model. As a rough guide, typical models favoured by BIC are less complicated than those chosen by AIC or AICc. AICc (small sample) is designed for cases where parameters/ observations < 40. Use AIC or AICc, but not both.
Durbin-Watson	Correlation of adjacent observations, usually the residuals of a time series regression.	Value of 2.0 suggests that there is no time correlation.

Note

For ordinary least squares cases, 'p' is the number of parameters, but for non-ordinary least squares cases (rols and ridge least squares) 'p' is the e ective number of parameters (p = N-df).

The total sum of squares (SST) calculation depends on a model constant. If your model has a constant, the total sum of squares (SST) calculation depends on the mean (SST=sum(y-ymean)^2). If your model does not have a constant, the calculation does not depend on the mean (SST=sum(y)^2).

Other Model Assessment Windows

To view the information listed in the table, you can other model assessment windows.

To	Select
View a read-only version of the inputs, predicted, and actual responses.	View > Modeling Data to open the Data Editor.
Displaying the t relative to the selected data.	Model > Evaluate to open a wizard that helps you select the data, and then opens the Model Evaluation window.
Plot multiple models on the same plot, or a table view.	Model > Selection Window to open the Model Selection window.

See Also

Related Examples

Assess High-Level Model Trends on page 6-2

Guidelines for Selecting the Best Model Fit on page 6-54

Toolbox Terms and Statistics De nitions on page 6-91

Box-Cox Transformation on page 6-77

Stepwise Regression on page 6-65

Assess Point-by-Point Models

In this section...

Analyze Point-by-Point Models and Choose the Best on page 6-31
Edit Point-by-Point Model Types on page 6-32
Assess Point-by-Point Fits Using Model Plots on page 6-33
Assess Point-by-Point Fits Using Statistics on page 6-34

Analyze Point-by-Point Models and Choose the Best

In the point-by-point model view, you nd controls and menu items speci c to point-by-point models (using the `Point-by-Point` test plan). Use the following tools to choose the best models:

To assess high-level model trends, use the Response Models tab at the test plan node. View the cross-section plots of all your response models at once. For details, see Assess High-Level Model Trends on page 6-2.

In the point-by-point model node, you can assess all the alternative models for each test, and decide which model type to choose for the selected test. Click in the alternative models list to view and compare the plots and statistics for each t. For details see Assess Point-by-Point Fits Using Model Plots on page 6-33.

The alternative models list displays the value of your selection criteria (e.g., PRESS RMSE) for each model type, with the Best Model check box selected for the currently selected best model for the test. You select criteria when you create local multiple models. The toolbox automatically selects a best model for each test based on your selection criteria. Assess all the ts, and if desired, change the selected check box in the Best Model column.

Click Add Local Model in the Common Tasks pane to try adding one more model type. In the Model Setup dialog box, choose a model type to add. When you click OK the toolbox ts the new model type to all tests, and then selects it as best if it is better (by your selection criteria) than any of the alternatives for a test. A dialog box informs you which tests (if any) have a new best model.

Click Edit Model in the Common Tasks pane to change the list of alternative model types for every test. See Edit Point-by-Point Model Types on page 6-32.

Select Model > Summary Statistics to open the Summary Statistics dialog box. In this dialog box, select statistics to display in the alternative models list and in the

Local summary statistics table. See Summary Statistics on page 6-28. If you are using validation data, the validation RMSE appears in the summary table for the test, if there is validation data for the current test (global variables must match), for comparison with the model fit RMSE. See Using Validation Data on page 6-61.

Edit Point-by-Point Model Types

Click Edit Model in the Common Tasks pane to change the list of alternative model types for every test.

In the Point-by-Point Model Setup dialog box, choose model types to add or edit the existing model list. You can use any model available as one-stage models. You can choose the summary statistic to use as the selection criteria for deciding which model fits best to each test.

View the list of default point-by-point model types.

Click Add or Edit to add and change models.

When you add models, in the Model Setup dialog box you can choose from all the global models available for a one-stage model with the same number of inputs as your current point-by-point model.

To choose a template to build a selection of models, click Template. There are predefined templates for polynomials, radial basis functions, hybrid radial basis functions, Gaussian process models, and you can also save your own templates of any models you choose. See Create Alternative Models to Compare on page 5-84.

Select a statistic for selecting the best model in the Criteria list (such as RMSE or PRESS RMSE). The toolbox uses the statistic to select the best model type for each test. You can also change the choice of model for each test after the models are fitted.

Select statistics to display in the point-by-point model view by clicking Statistics.

To use data-based ranges for each test, leave the Automatic input ranges check box selected. Clear the check box only if you want to use instead the range for every local model that you set up in the test plan. Using data-based ranges for each test is helpful when the local input ranges for local tests vary between tests. This option is useful for diesel modeling as often there are a number of inputs at the local level (e.g., main injection timing, pilot injection timing, rail pressure, boost pressure and EGR are common variables). The ranges of these variables vary over the global input space (e.g., torque and speed or fuel and speed). Adjusting the ranges for each test means that the inputs are scaled for modeling leading to better conditioned models.

After you add models to your list and close the dialog box, all the models you have chosen are fitted to each test individually, and the best fit to each test is chosen by the selection criteria you picked. Assess all the alternative models for each test.

Assess Point-by-Point Fits Using Model Plots

RMSE Plot

Use the RMSE Plot to quickly identify problem tests and navigate to an operating point of interest. Navigate to a test of interest by double-clicking a point in the plot to select the test in the other plots in the model view.

Response Surface Plot

This view shows the model surface in a variety of ways. The default view is a 3-D plot of the model surface.

You can choose which input factors to display by using the drop-down menus left of the plot. The unselected input factors are held constant and you can change their values using the controls at the left of the plot (either by clicking the arrow buttons or by typing directly in the edit box). Click Select Data Point to choose a point to plot.

Select the Plot list to switch to a `Line`, `Contour`, or `Multiline` plot.

Diagnostic Statistics Plot

The Diagnostic Statistics plot shows various scatter plots of statistics for assessing goodness-of-fit for the current model.

The statistics and factors available for plotting are model dependent. Choose the x- and y-axis factors using the drop-down menus.

Additional Plots

You can add or change plots by clicking the toolbar buttons, split buttons in plot title bars, or selecting an option from Current View in the context menu or View menu. The browser remembers your layout per testplan. You can add:

- `Predicted/Observed`
- `Normal Plot`
- `Validation Data`

Model Definition

These plots are also used for one-stage models. For details see Assess One-Stage Models on page 6-16.

To view plots of the data for the current test, add Data Plots. Select View > Plot Variables to choose variables to plot. You can choose to view any of the data signals in the data set for the current test (including signals not being used in modeling). You can plot a pair of variables or plot a variable against record number. You can add more data plots if you want.

You can also view values of input variables in the Operating Point pane.

Assess Point-by-Point Fits Using Statistics

For advice on using the statistics such as PRESS RMSE to compare point-by-point models, see Compare Fits Using Statistics on page 6-25.

See Also

Related Examples

Fit a Point-by-Point Model on page 1-14

Assess High-Level Model Trends on page 6-2

Compare Alternative Models on page 6-25

Guidelines for Selecting the Best Model Fit on page 6-54

Model Selection Window

In this section...

Comparing Models on page 6-35
Select a Best Model on page 6-36
Plots and Statistics for Comparing Models on page 6-36

Comparing Models

You can use the Model Selection window to help you select a best model by comparing several candidate models on the same plot.

Select Model > Selection Figure to open the Model Selection window and compare the child nodes of your current view.

You can select among the following:

- Local models
- Response features
- Submodels of response features
- Global models

However, you cannot select between response models or test plans.

Note If you click Create Two-Stage in the Common Tasks pane at the local node and you need to select response features, then the toolbox opens the Model Selection window. You need the right number of response features to create a two-stage model, so you must choose which combination of response features to use to create the two-stage model. Compare the possible two-stage models and select the best.

After calculating a two-stage model with MLE, you can use the Model Selection window to compare the MLE model with the previous univariate model, and you can choose the best.

Model Selection might not be available if you are not ready to choose among the child nodes. For example, at the response node, the child nodes must have models assigned as best before you can select among them. Also, if a response feature has child nodes of

alternate models, you must select the best, or the Browser cannot tell which to use to calculate that response feature.

Use the Model Selection window for visual comparison of several models. From the response level you can compare several two-stage models. From the local level, if you have added new response features you can compare the di erent two-stage models (constructed using di erent combinations of response feature models). If you have added child nodes to response feature models, you can compare them all using the Model Selection window.

When a model is selected as best it is copied up a level in the tree together with the outliers for that model t.

A tree node is automatically selected as best if it is the only child.

If a best model node is changed the parent node loses best model status (but the automatic selection process will reselect that best model if it is the only child node).

Note You can assign models as best in the Model Browser without needing to open the Model Selection window. See Compare Alternative Models on page 6-25.

Select a Best Model

In the Model Selection window, click the Assign Best button at the bottom of the window to mark the currently selected model as best, or you can double-click a model in the list.

To choose which model to select as best, use the plots and statistics in the Model Selection window described in the next section. To determine the best t, you should examine both the graphical and numerical t results. When you can no longer eliminate ts by examining them graphically, you should examine the statistics. For guidance on how to use the statistics for assessing models, see Guidelines for Selecting the Best Model Fit on page 6-54.

Plots and Statistics for Comparing Models

You can display several di erent views in the Model Selection window, depending on the type of models you are comparing:

Tests View on page 6-37

Predicted/Observed View on page 6-40
Response Surface View on page 6-41
Likelihood View on page 6-44
RMSE View on page 6-46
Residuals View on page 6-48
Cross Section View on page 6-50

You can change to any available view in the Model Selection window using the View menu or by clicking the buttons of the toolbar.

Information about each candidate model is displayed in the list at the bottom. The information includes categories such as the number of observations and parameters, and various diagnostic statistics such as RMSE and PRESS RMSE. You can click on column headers in this list to sort models by that category for example, clicking on the column header for PRESS RMSE sorts the models in order of increasing PRESS RMSE. As this statistic is an indication of the predictive power of the model, it is a useful diagnostic statistic to look at (the lower the better), but remember to also look at other factors.

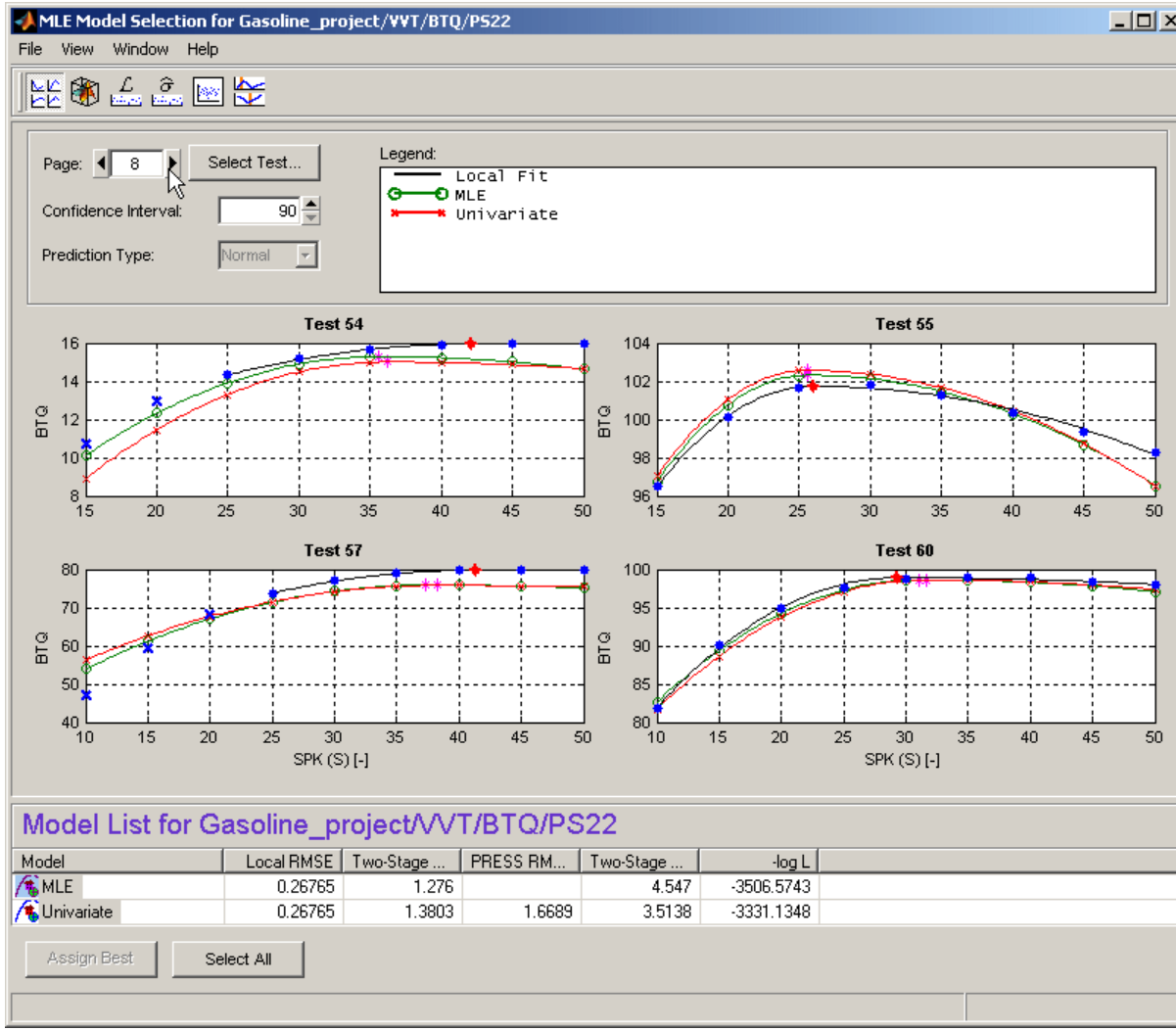
To print the current view, use the File > Printmenu item or its hot key equivalent Ctrl+P. In the Response Surface view you can also use the right-click context menu.

To close the Model Selection window, use the File > Close menu item or its hot key equivalent Ctrl+W. Model Selection is intended to help you select a best model by comparing several candidate models, so when you close the window you are asked to con rm the model you chose as best.

See also Model Evaluation Window on page 6-60, which includes some of the same views you see in the Model Selection window, and where you can use validation data.

Tests View

For a two-stage model the initial view is as follows:

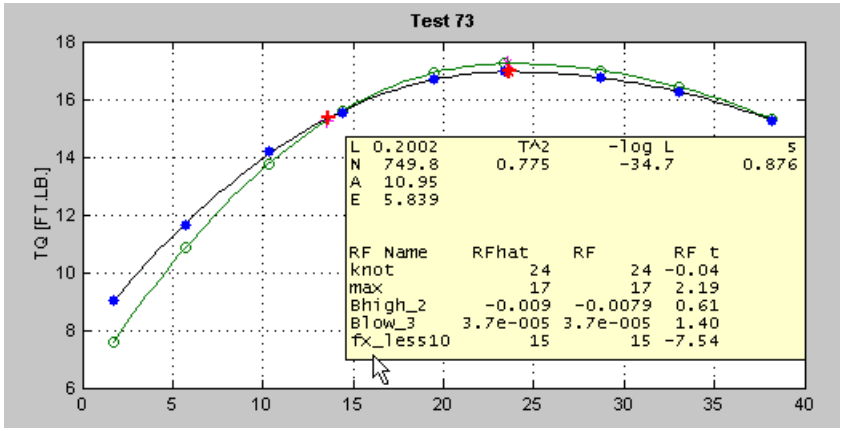


The tests view shows the data being modeled (blue dots) and models that have been tted to this data. The black line shows the local model that has been tted to each test separately. The green line and red lines in this case show an MLE two-stage model and the Univariate two-stage model: you can see the local model curves reconstructed using response feature values taken from the global models, and compare the ts.

This view allows you to compare several models simultaneously. Using standard Windows multiselect behavior (Shift+click and Ctrl+click) in the list view, or by clicking the Select All button, you can view several two-stage models together. A maximum of ve models can be selected at once. The legend allows you to identify the di erent plot lines.

If the local input has more than one factor, a Predicted/Observed View on page 6-40 appears instead.

Clicking one of the plots (and holding the mouse button down) displays information about the data for that test. For example:

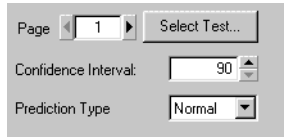


Here you see the values of the global variables for this test and some diagnostic statistics describing the model t. Also displayed are the values (for this test) of the response features used to build this two-stage model and the two-stage model's estimation of these response features.

The controls allow navigation between tests.

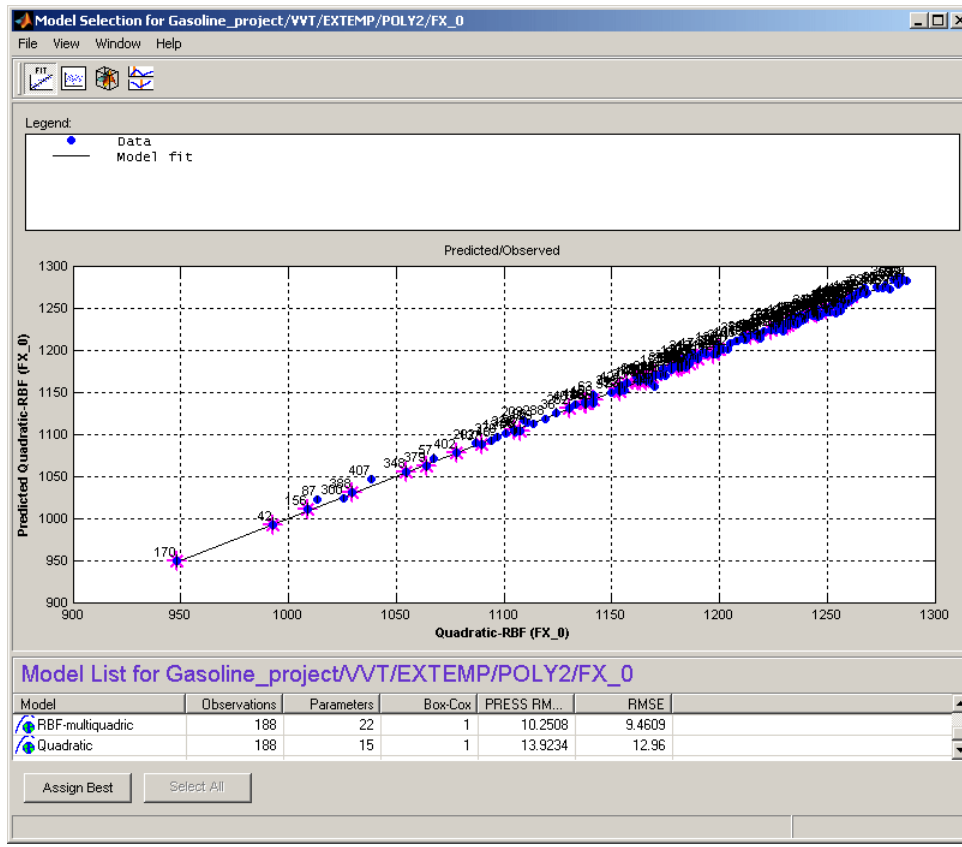
You can change the size of the con dence intervals; these are displayed using a right-click menu on the plots themselves.

The prediction type allows a choice of Normal or PRESS (Predicted Error Sum of Squares) although not if you entered this view through model evaluation (rather than model selection). PRESS predictions give an indication of the model t if that test was not used in tting the model. For more on PRESS see PRESS statistic on page 6-74, Guidelines for Selecting the Best Model Fit on page 6-54, and Stepwise Regression on page 6-65.



Predicted/Observed View

For a one-stage model, or when you are comparing di erent models for one Response Feature, the initial view is as follows:



The plot shows the data used to t this model, against the predicted values found by evaluating the model at these data points. The straight black line is the plot of y=x. If the

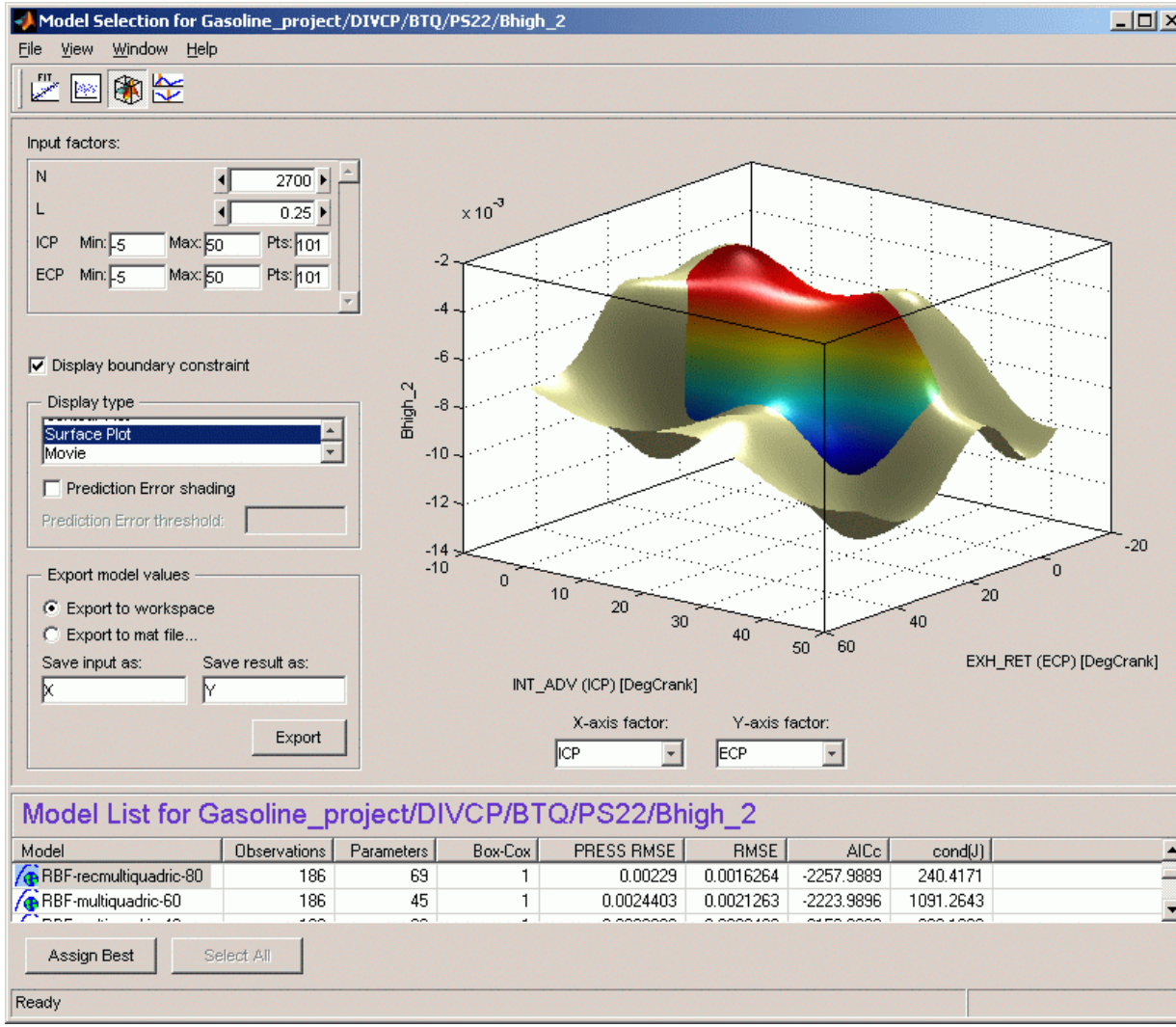
model fitted the data exactly, all the blue points would lie on this line. The error bars show the 95% confidence interval of the model fit.

For single inputs, the response is plotted directly against the input.

The Predicted/Observed view only allows single selection of models for display. Right-click to toggle test number display, as you can on most plots.

Response Surface View

This view shows the model surface in a variety of ways.



The default view is a 3-D plot of the model surface, as in the example. This model has ve dependent factors; you can see these in the controls at the top left (there is a scroll bar as only four can be seen at once at this size of window).

You can choose which input factors to display by using the drop-down menus below the plot. The unselected input factors are held constant and you can change their values

using the controls at the top left of the view (either by clicking the arrow buttons or by typing directly in the edit box).

Display using (S - datum) If a datum model is being displayed, this check box appears. The datum variable here is spark angle, S. When you select this box, the model is displayed in terms of spark angle relative to the datum. The appropriate local variable name appears here. See Datum Models on page 5-81.

Display boundary constraint If you have boundary models you can display them by selecting the check box. Areas outside the boundary are yellow, as shown in the example. Areas outside the boundary are yellow (or gray in table view). They are shown on all display types (contour, 2-D, surface, movie and table).

Display Type Changes the model plot. Display options are available for some of these views and are described under the relevant view. The choices are as follows:

A table showing the model evaluated at a series of input factor values.

A 2-D plot against one input factor.

A 2-D plot with several lines on it (called a multiline plot); this shows variation against two input factors.

A contour plot.

The Contours. button opens the Contour Values dialog box. Here you can set the number, position, and coloring of contour lines.

Fill Contour colors each space between contours a di erent color.

Contour Labels toggles the contour value numbers on and o . Without labels a color bar is shown to give you a scale.

Auto (the default) automatically generates contours across the model range.

N Contour Lines opens an edit box where you can enter any number of contour lines you want.

Specify values opens an edit box where you can enter the start and end values where you want contour lines to appear, separated by a colon. For example, entering `5:15` gives you 10 contour lines from 5 to 15. You can also enter the interval between the start and end values; for example `1:100:600` gives you contour lines between 1 and 600 at intervals of 100.

A surface (shown in the example).

Prediction Error shading Colors the surface in terms of the prediction error (sqrt (Prediction Error Variance)) of the model at each point. A color bar appears, to show the value associated with each color.

Note For datum models, Prediction Error shading is only available when the Display using (local variable - datum) check box is not selected.

Prediction Error threshold To see good color contrast in the range of PE of interest, you can set the upper limit of the coloring range. All values above this threshold are colored as maximum P.E.

A movie: this is a sequence of surfaces as a third input factor's value changes.

Replay replays the movie.

Frame/second selects the speed of movie replay.

The number of frames in the movie is de ned by the number of points in the input factor control (in the array at the top left) that corresponds to the Time factor below the plot.

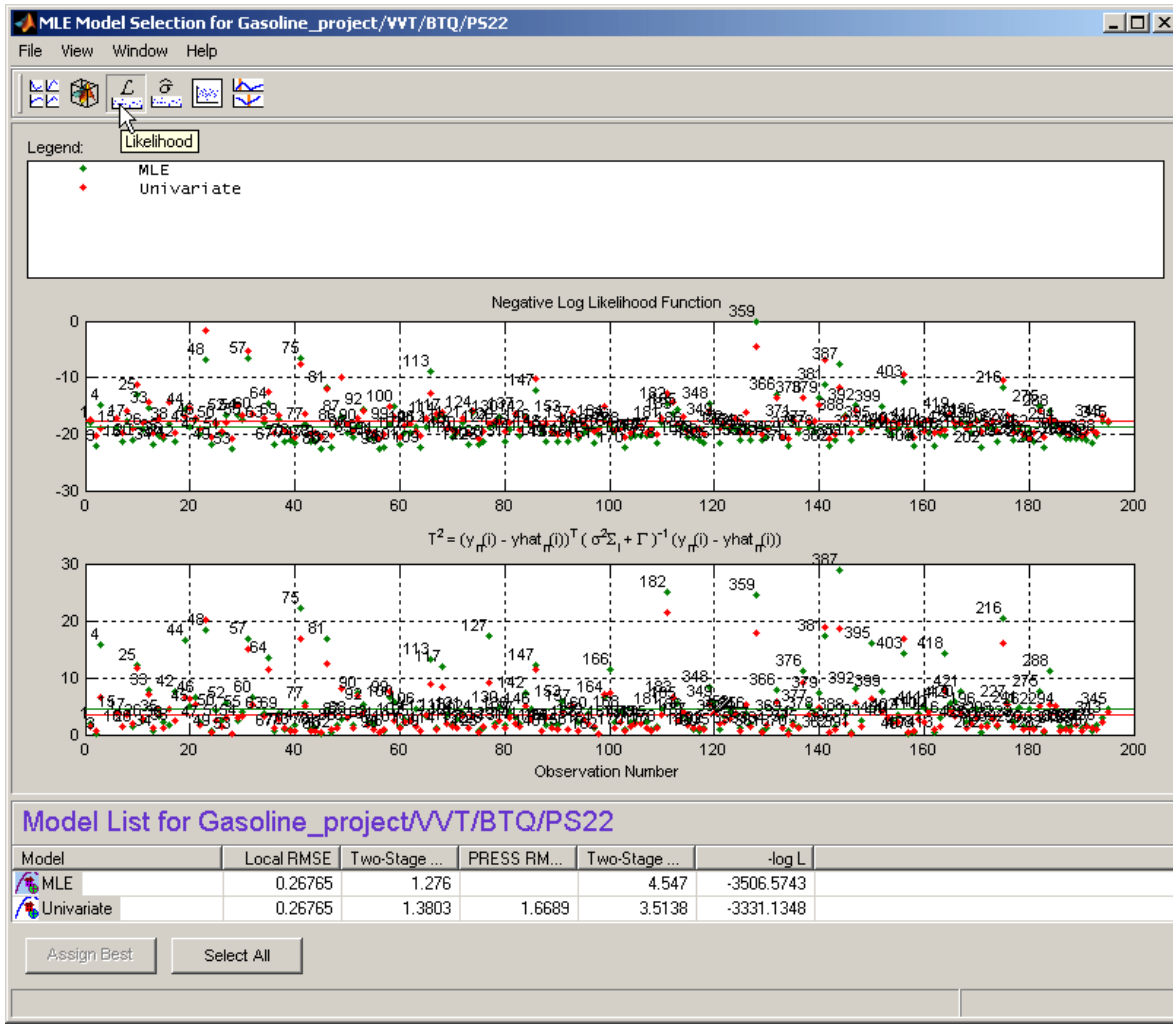
Export model values allows the currently displayed model surface to be saved to a MAT le or to the MATLAB workspace.

Right-click on the plot to reach the context menu and change many display properties (lighting, colormap etc.) and print to gure.

Within a test plan the memory is retained of the evaluation region, plot type and the number of points resolution last displayed in the Response Surface view.

Likelihood View

The likelihood view shows two plots relating to the log likelihood function evaluated at each test. It is useful for identifying problem tests for maximum likelihood estimation (MLE).



Each plot has a right-click menu that allows test numbers to be displayed on the plots and also o ers autoscaling of the plots. You can also Print to Figure.

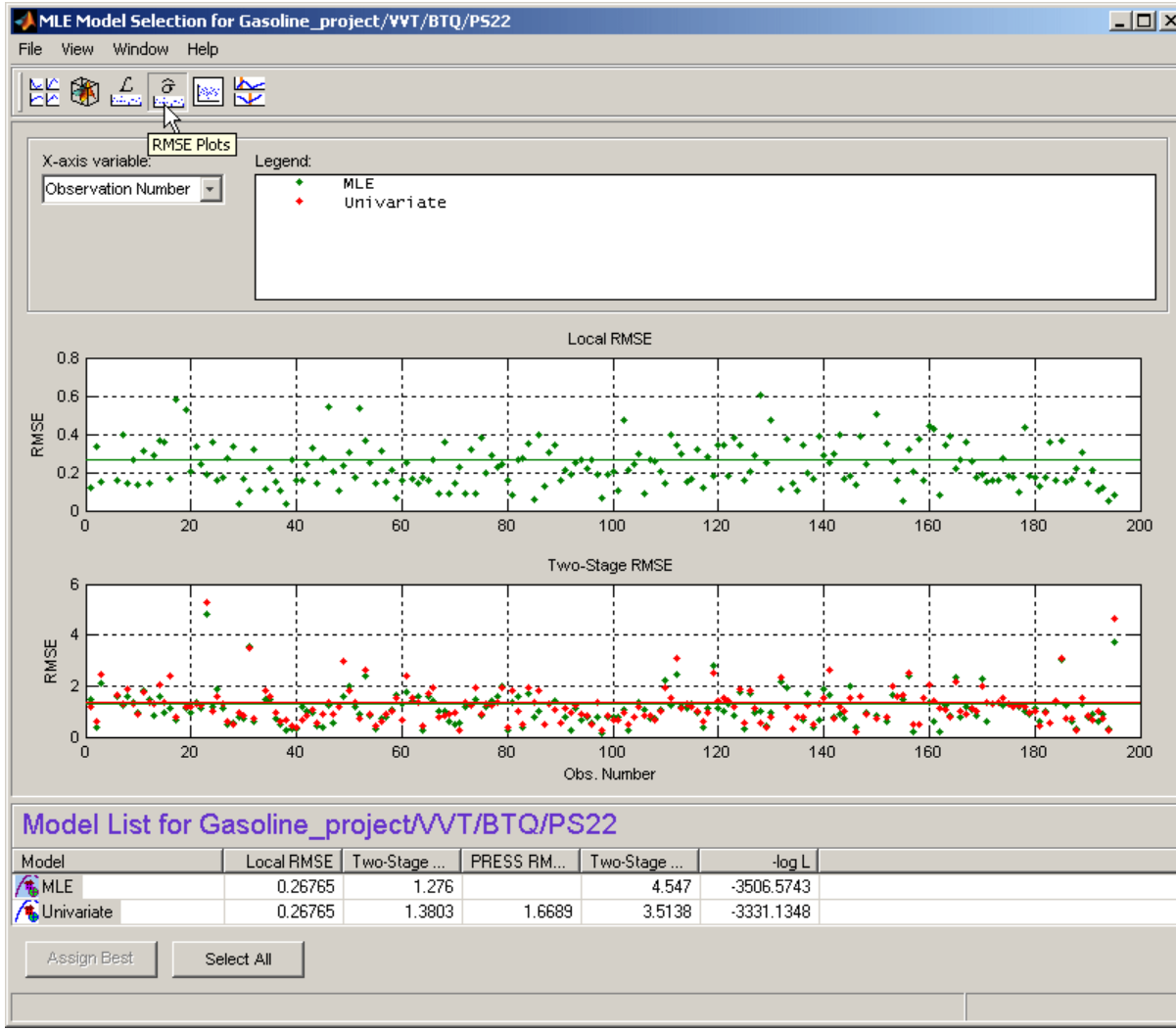
The likelihood view allows several models to be displayed simultaneously; click the Select All button at the bottom of the window or, in the model list view, Shift+click or Ctrl +click to select the models for display.

The upper plot shows values of the negative log likelihood function for each test. This shows the contribution of each test to the overall negative log likelihood function for the model, as compared with the average, as indicated by the horizontal green line.

The lower plot shows values of the T-squared statistic for each test. This is a weighted sum squared error of the response feature models for each test. As above, the purpose of this plot is to show how each test contributes to the overall T-squared statistic for this model. The horizontal line indicates the average.

RMSE View

The Root Mean Square Errors view has three di erent plots, each showing standard errors in the model t for each test.



Each plot has a right-click menu that allows test numbers to be displayed on the plots, and you can Print to Figure.

The X variable menu allows you to use di erent variables as the x-axis of these plots.

The RMSE view allows several models to be displayed simultaneously; click the Select All button at the bottom of the window or, in the model list view, Shift+click or Ctrl +click to select the models for display.

Local RMSE shows the root mean squared error in the local model fit for each test.

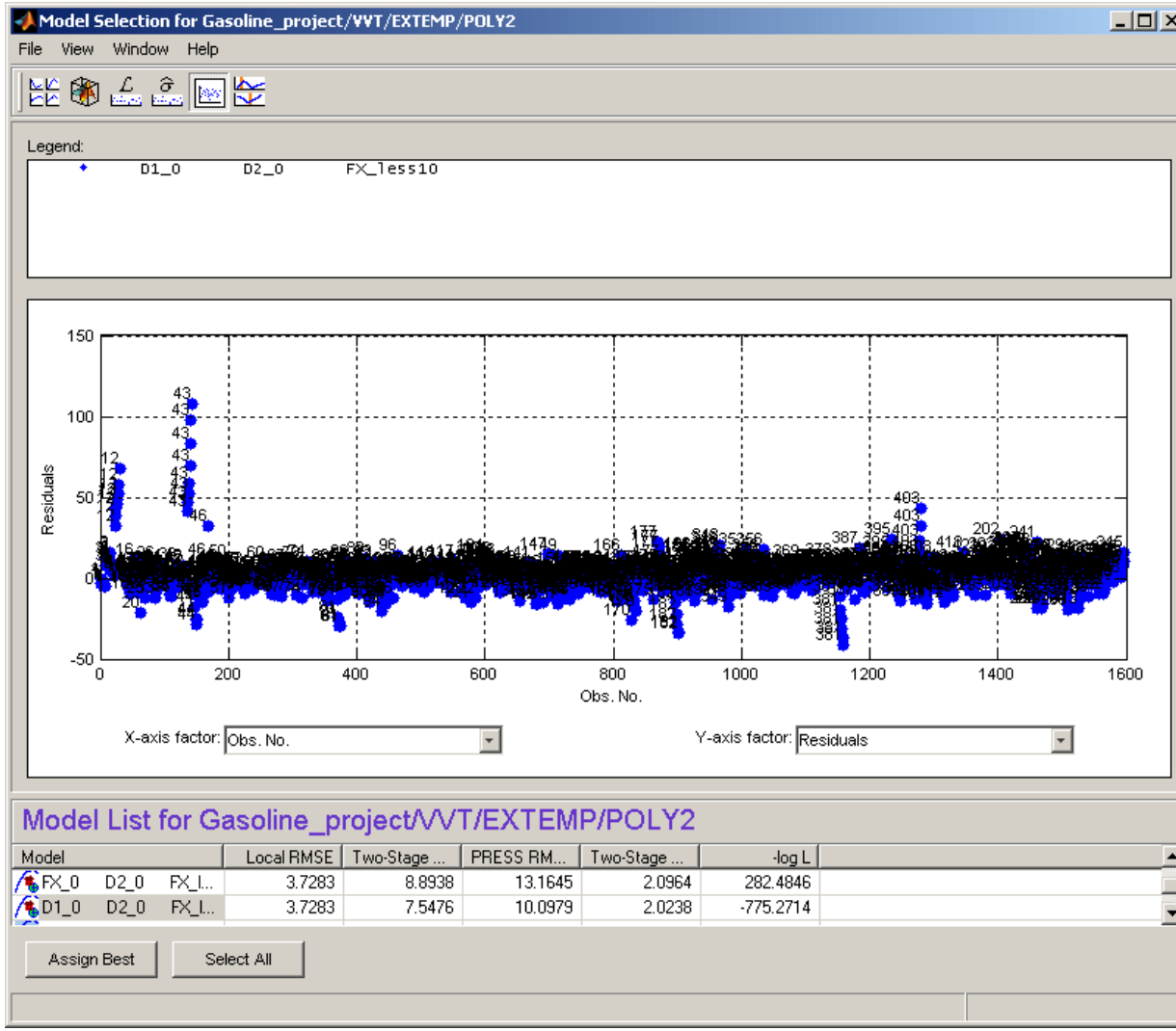
Two-Stage RMSE shows the root mean squared error in the two-stage model fit to the data for each test. You should expect this to be higher than the local RMSE.

PRESS RMSE is available when all response feature models are linear. This plot shows the root mean squared error in the PRESS two-stage model fit at each test.

For information on PRESS RMSE see Guidelines for Selecting the Best Model Fit on page 6-54.

Residuals View

The residuals view shows the scatter plots of observation number, predicted and observed response, input factors, and residuals.



This view allows several models to be displayed simultaneously; click the Select All button at the bottom of the window or, in the model list view, Shift+click or Ctrl+click to select the models for display.

A right-click menu allows the test number of each point to be displayed when only one model is being displayed, as shown.

The X-axis factor and Y-axis factor menus allow you to display various statistics.

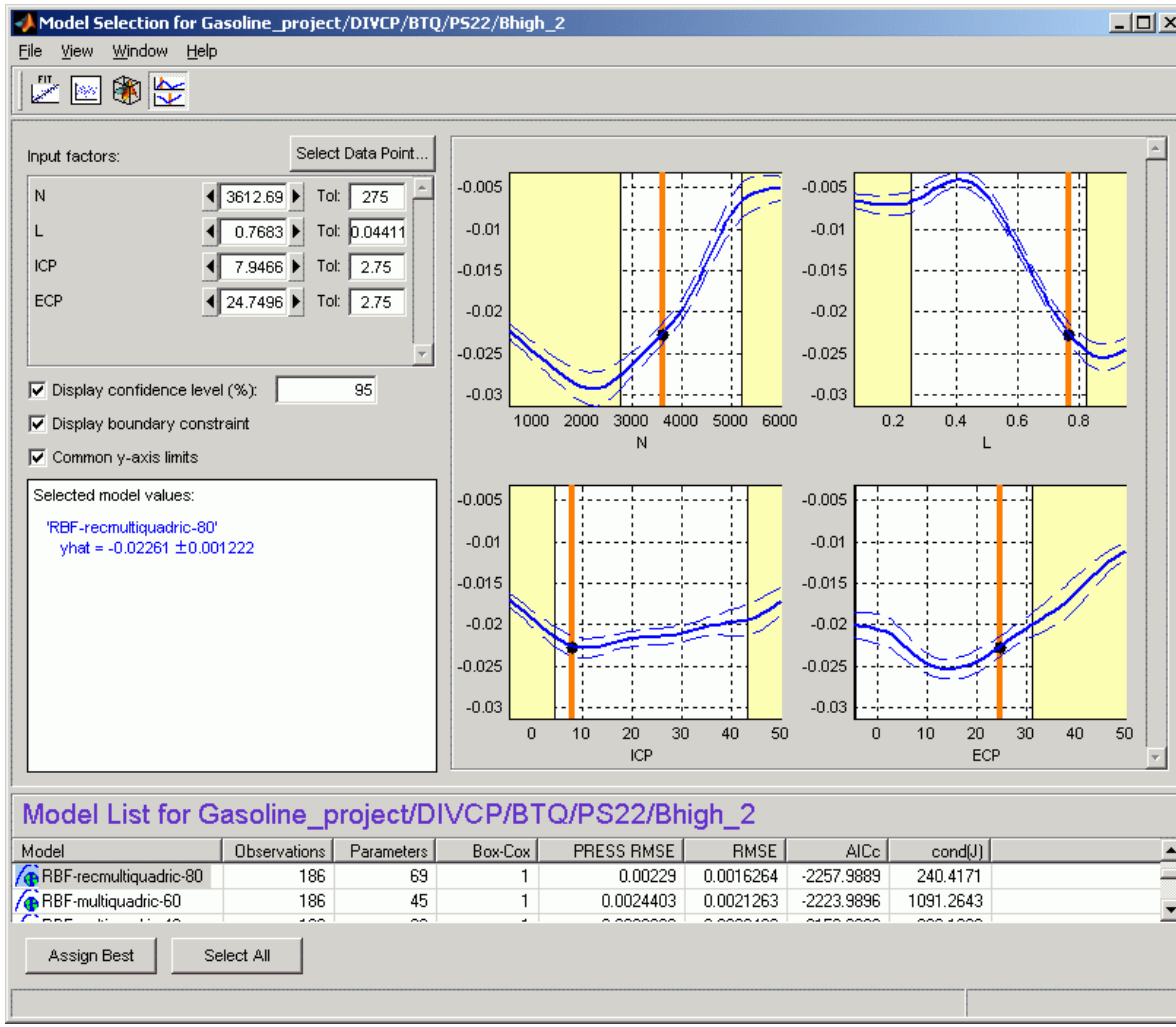
Cross Section View

The cross-section view shows an array of cross sections through the model surface. You can choose the point of cross section in each factor. Data points near cross sections are displayed, and you can alter the tolerances to determine how much data is shown. The only exception is when you evaluate a model without data; in this case no data points are displayed.

You can select individual data points by test number (using the Select Data Point button). You can double-click a data point in a graph to take the display directly to that point. You can choose to use a common Y-axis limit for all graphs using the check box.

If you have boundary models you can choose to display them here using the check box; regions outside the boundary are yellow, as shown in the example.

Within a test plan the memory is retained of the point last displayed in the Cross Section view; when you reopen the view you return to the same point.



The number of plots is the same as the number of input factors to the model. The plot in S shows the value of the model for a range of values of S while the other input factors are held constant. Their values are displayed in the controls at the top left, and are indicated on the plots by the vertical orange bars.

You can change the values of the input factors by dragging the orange bars on the plots, using the buttons on the controls, or by typing directly into the edit boxes.

For example, changing the value of N to 1000 (in any of these ways) does nothing to the graph of N, but all the other factor plots now show cross sections through the model surface at N = 1000 (and the values of the other variables shown in the controls).

On the plots, the dotted lines indicate a con dence interval around the model. You de ne the con dence associated with these bounding lines using the Display con dence level (%) edit box. You can toggle con dence intervals on and o using the check box on this control.

For each model displayed, the value of the model and the con dence interval around this are recorded in the legend at the lower left. The text colors match the plot colors. In the example shown, two models are selected for display, resulting in blue (PS22 model) and green (POLY2 model) legends on the left to correspond with the blue and green plots. You can select multiple models to display in the list at the bottom using Ctrl+click, or click Select All. The values of the input factors (for which the model is evaluated) can be found in the controls (in the Input factors pane) and seen as the orange lines on the plots.

Data points are displayed when they fall within the tolerance limit near each cross section. You can set the tolerance in the Tol edit boxes.

For example, if N is set to 1000, and the tolerance for N is set to 500, all data points with values between N = 500 and N = 1500 appear on the plots of the other factors.

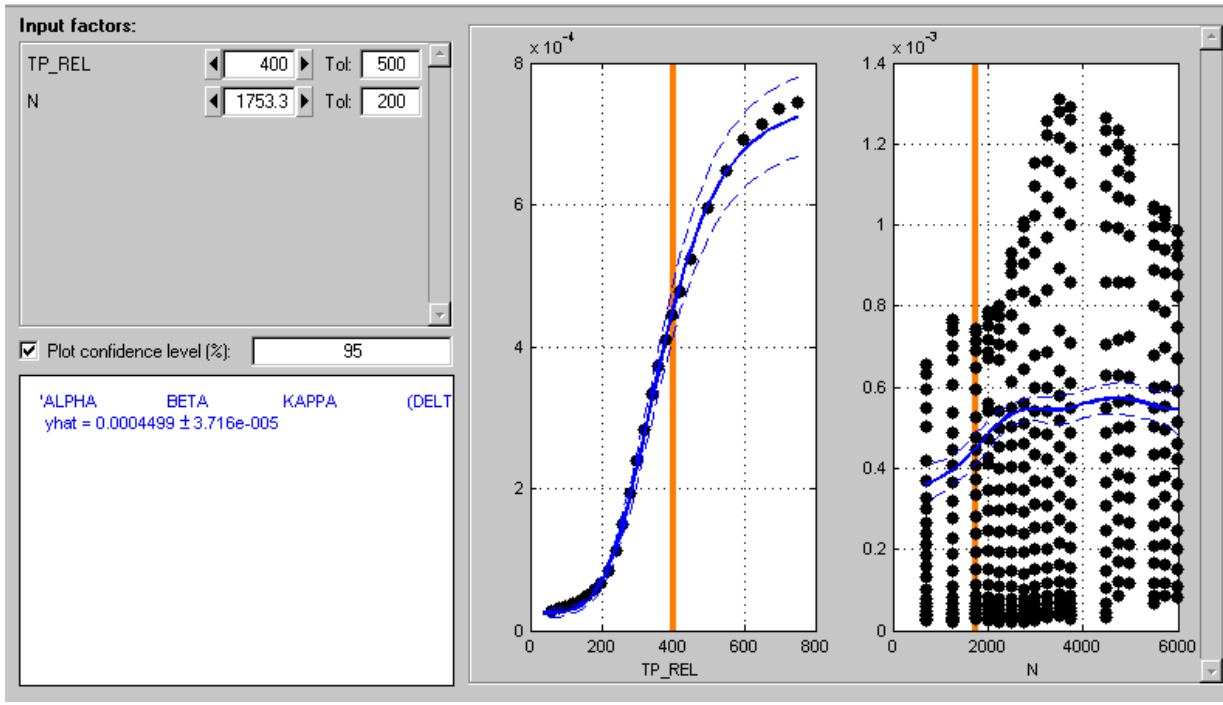
This means that changing the tolerance in one factor a ects the data points that appear on the plots of all the other factors. It does not a ect the plots of that factor.

You can click data points in the plots to see their values. Several points can mask each other; in this case the values of all coincident data points are displayed. Double-click to move the display directly to a data point.

The following example illustrates how the tolerance level determines which data points are displayed. The tolerance for TP_REL (500) includes all points in the data set (this is an extreme example). The plot for N therefore shows the data points for all the tests. Note that you can see the structure of the data as each test shows as a vertical line of points.

You can see that the orange line on the N plot passes through a test. This orange line shows the value of N for the cross-section plot of TP_REL. You can also read the value in the edit box (N=1753.3). The tolerance for N (200) only includes data points of this test. Data in adjacent tests fall outside this tolerance. Therefore the TP_REL plot shows the data points from one test only.

Increasing the tolerance on N will mean that more data points fall within the tolerance and so would appear on the TP_REL plot.



Guidelines for Selecting the Best Model Fit

To determine which model provides the best fit for a data set, use the plots and statistics in the Model Browser views. To determine the best fit, first examine the graphical results. When you can no longer eliminate model fits by examining them graphically, use the statistical results. Use these guidelines to help assess the models and determine the best fit.

Overfitting and Underfitting

When the Model-Based Calibration Toolbox fits noisy data, the model can overfit or underfit the data. During the fit, the toolbox balances bias and variance. Bias measures how well the model fit follows the data trends. Variance and the root mean squared error (RMSE) both measure how well the model fit matches the data. RMSE is the square root of the variance.

To determine if a model is underfit or overfit, consider the number of parameters in a model. As this number increases, so does the model complexity. By default, Model-Based Calibration Toolbox uses a Gaussian process model (GPM) to fit data. If the number of parameters in a GPM approaches the number of observations, the model might be overfit.

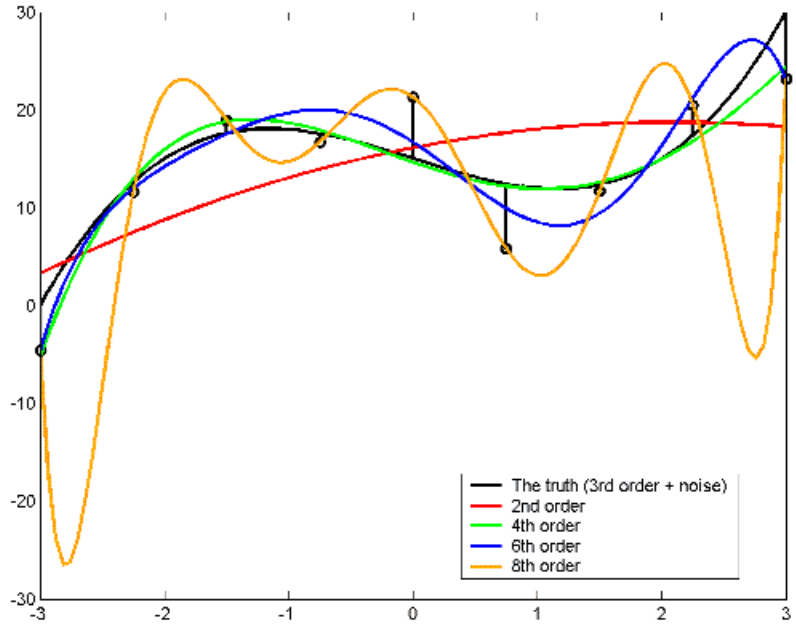
Model	Description	Bias	Variance/RMSE
Underfit	Model does not capture the data trends. Model contains fewer parameters than the data justifies.	High	High
Overfit	Model fits the data points too closely, following the noise rather capturing the trend. Model contains more parameters than the data justifies.	Low	Low

RMSE

The root mean squared error measures the average mismatch between each data point and the model. To inspect the fit quality, start with the RMSE values. High RMSE values

can indicate problems. Low RMSE values indicate a close match with the data. If a model predicts each data point exactly, then the RMSE is zero.

This illustration shows how increasing the number of parameters in a model can result in over tting while maintaining a low RMSE. The nine "truth" data points are generated from a cubic polynomial with a known amount of noise. A cubic polynomial has four parameters. In this case, the 4th order model provides the best t.



Model	Parameters	Over t	Under t	RMSE
The truth (3rd order plus noise)	4	NA		
2nd order	3			High
4th order	5			Low
6th order	7			Low

Model	Parameters	Over t	Under t	RMSE
8th order	9			Low

PRESS RMSE and Other Statistics

If you rely solely on the RMSE to assess a model t, the model might be over t and perform poorly in regions that do not contain data points. Consider using the predicted residual error sum of squares (PRESS) root mean squared error (RMSE) and information criteria statistics, which measure the model over t.

Statistic	Description	Assess Model Fits
PRESS RMSE Predicted residual error sum of squares (PRESS) root mean squared error (RMSE)	For each data point, the statistic calculates how well the model ts the data point when it is not included in the t. The PRESS RMSE is the average of the results.	If the PRESS RMSE is larger than the RMSE, the model might be over t. In general, use PRESS RMSE for smaller data sets.
AIC and AICc Akaike Information Criteria BIC Bayesian Information Criteria	Statistics that combine an RMSE term with a term that rises with the number of parameters in the model. This penalizes a model for an increase in its level of structure. AIC, AICc, and BIC are approximations, which get more accurate as the number of data points increases.	For better ts, minimize the information criteria statistics. In general, do not use them unless the ratio of the data points to parameters is greater than 40:1.[1] Use AICc for smaller data sets. AIC the most appropriate information criterion for most problems in engine calibration.

Validation

These statistics help you select a model that makes reasonable predictions at the data points and the regions between the data points. To validate your model, collect additional

validation data. Then use your model to measure how well the model predicts that validation data. Comparing a validation RMSE with the RMSE based on the modeling data is a good model selection statistic. Use the Model Evaluation window to validate models against other data. You can use validation data throughout a test plan.

Using Information Criteria to Compare Models

To help you use information criteria to compare models, this section provides background information about the Akaike Information Criteria (AIC and AICc) and the Bayes Information Criterion (BIC).

Information Criteria	Description
AIC-type criteria	Based on the di erence in Kullback-Leibler information between two models, or their K-L distance. K-L distance is a useful measure because it compares the information content of two curves by calculating the entropy in each. Akaike and others found ways to estimate K-L distance based on the results of a maximum likelihood estimate of the parameters of a model, given some data. These estimates are the information criteria, and become more accurate as the sample size increases.[1]
BIC	Derived from Bayes theorem. Applies the Occam e ect to select a preferred model. If two models provide an equally good t with some data, then the simpler model is the likelier. For models with greater complexity, it is less remarkable that they are able to t a given data set well. Conversely, for a simple model, if you encounter a data set for which the model provides an acceptable t, it would seem a coincidence. Therefore, for data matching both models well, the odds are that the simpler one is closer to the truth. [4]
Bayes factors (evidence ratios)	Measure the relative probabilities of two models. In the context of Model-Based Calibration Toolbox, BIC is an estimate of Bayes factors based on the results of a maximum likelihood estimate, and, like AIC, increases in accuracy in the limit of large sample size. Although priors often spring to mind in the context of Bayes theorem, the Occam e ect still applies.[3]

AIC and BIC improve as estimators of their statistical measures as the sample size increases, with relative errors of O(n-1), where n is the sample size. AIC is obtained from

a 1st order Taylor expansion. AICc is a 2nd order correction for the special case of Gaussian Likelihood (no general 2nd order correction). Use AICc when the ratio of data samples to model parameters (in the largest model for nested sets) is less than about 40:1.[2], [5]

Most problems in Model-Based Calibration Toolbox are not so simple that the model contains the closed-form solutions to the dynamic equations. In terms of the number of samples per model parameter, AIC is seldom likely to be a reliable statistic. Use AICc instead. If you prefer a more conservative estimate of the complexity of the model, consider using the BIC.

References

[1] Burnham, Kenneth P., and David R. Anderson. Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. 2nd edition. New York: Springer-Verlag, 2002.

[2] Draper, Norman R, and Harry Smith. Applied Regression Analysis. 3rd edition. New York: John Wiley & Sons, 1998.

[3] Kass, Robert E., and Adrian E. Raftery. Bayes Factors. Journal of the American Statistical Association. Volume 90, Number 430, 1995, pp. 773 795.

[4] Leonard, Thomas and John S.J. Hsu. Bayesian Methods. Cambridge: Cambridge University Press, 2001.

[5] MacKay, David. Information Theory, Inference, and Learning Algorithms. Cambridge: Cambridge University Press, 2003.

See Also

Related Examples

Compare Alternative Models on page 6-25

Summary Statistics on page 6-28

Using Validation Data on page 6-61

Assess Two-Stage Models

View response models by selecting a response model node (with a two-stage icon a house and a globe) in the model tree.

The plots show the data and the two-stage model t. You can scroll through the tests using the test controls, as at the local level on page 6-5: by clicking the up/down page number buttons, typing directly in the edit box, or clicking Select Test to go directly to a particular test.

The Local Models list view shows all child models and summary statistics for comparison if you have multiple local models. See Pooled Statistics on page 6-12 for information on the diagnostic statistics in the list (such as log likelihood and T^2). The statistics in this list can be seen in the Pooled Statistics table at the local model level.

To switch to the local model view, in the Common Tasks pane, click View Local Model. Assess Local Models on page 6-5.

To try a di erent local model to compare, in the Common Tasks pane, click New Local Model.

To view the Model De nition dialog box displaying the model terms, click View Model in the toolbar.

Note The response node remains empty until you have created a two-stage model at the local level. The two-stage model then appears at the response node. See Create Two-Stage Models on page 6-9.

Model Evaluation Window

In this section...

About the Model Evaluation Window on page 6-60
Using Validation Data on page 6-61

About the Model Evaluation Window

The Model Evaluation Window is intended to help you evaluate the t of your model. You can evaluate against the t data, validation data, other data, or without reference to any data. The window displays some of the same views you see in the Plots and Statistics for Comparing Models on page 6-36. The views available depend on what kind of model you are evaluating and the evaluation mode you choose.

You can access the Model Evaluation window via the menu items under Model > Evaluate from any of the modeling nodes: the one-stage or two-stage model node, local model node, response feature nodes, or any child model nodes. You can use validation data with any model except response features.

The Model Selection window allows you to compare di erent models with each other and with the data used to create these models. The Model Evaluation window also allows you either to examine a model without data or to validate it against data other than that used in creating the model. For any model node, model evaluation is a quick way to examine the model in more ways than those available in the main Browser views. For example, local models with more than one input factor can only be viewed in the Predicted/Observed view in the main Browser, and the Model Selection window only shows you the two-stage model, so you go to Model Evaluation to view the local model itself in more detail. For other models, such as childless response feature nodes or their child nodes, the Model Selection window is not available, so Model Evaluation is the way to view these models in detail.

There are four modes for evaluation, determined by your selection from the Model > Evaluate sub menu:

Fit data (or the hot key Ctrl+E) The evaluation window appears, and the data shown along with the model surface is the data that was used to create this model. Summary Statistics on page 6-28 are shown in the model list. The views available are

Residuals

Response surface

Cross section

Tests Two-stage models only

Predicted/observed One-stage or response feature only

Validation data The evaluation window appears (titled Model Validation). You can compare the model with the validation data. This option is only available if you have attached validation data to the test plan. See Using Validation Data on page 6-61. You can only use validation data with two-stage (response), local, and one-stage models (not response features). The views available are

Residuals

Response surface

Cross section

Tests two-stage models only. The local fit is not shown, as the local model was fitted to different data.

Validation RMSE appears in the model list for comparison with the Fit RMSE.

No data The evaluation window appears with only the model surface view and the cross-section view. These do not show the model fit against data. You can investigate the shape of the model surface.

Other data Opens the Select Data for Evaluation wizard, where you can choose the data that is shown along with the model views. The steps are the same as selecting validation data to the test plan. Select a data set, match signals if necessary (only if signal names do not match), and select tests you want to use (the default includes all tests). See Using Validation Data on page 6-61. When you click Finish, the evaluation window then appears with the same views shown for validation data.

For more information about each view, see Plots and Statistics for Comparing Models on page 6-36.

Using Validation Data

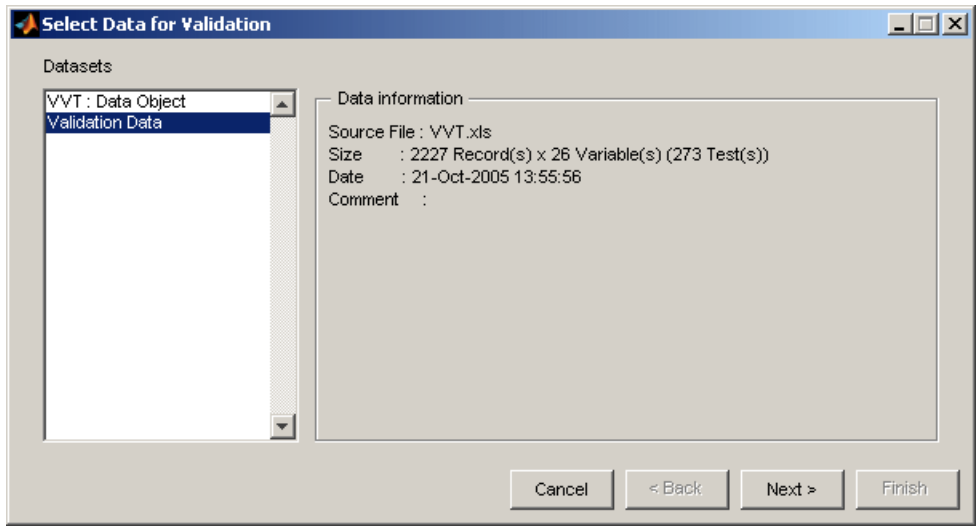
These statistics help you select a model that makes reasonable predictions at the data points and the regions between the data points. To validate your model, collect additional validation data. Then use your model to measure how well the model predicts that validation data. Comparing a validation RMSE with the RMSE based on the modeling data

is a good model selection statistic. Use the Model Evaluation window to validate models against other data. You can use validation data throughout a test plan.

Attach validation data to your test plan, then use it to validate your models. Validation RMSE appears in the statistics tables, you can view plots of validation residuals, and you can open the Model Evaluation window to investigate your models with validation data.

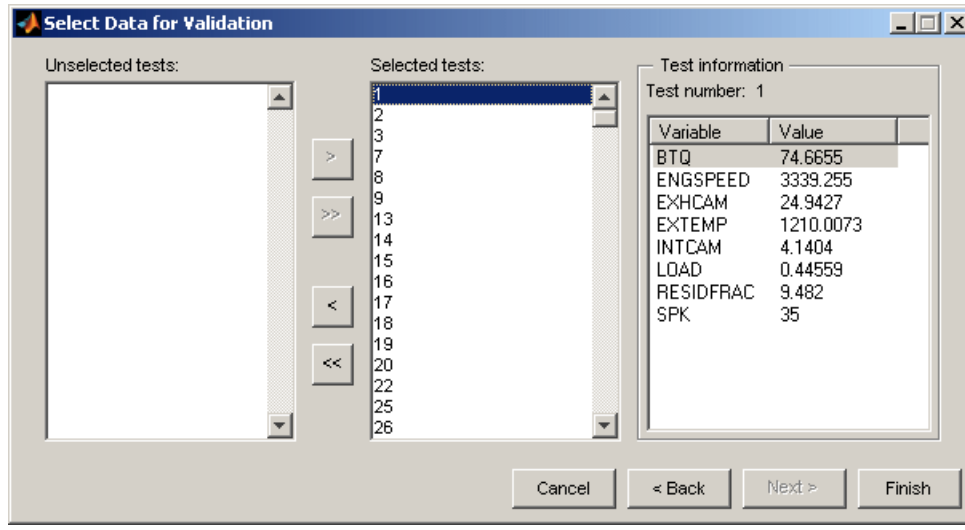
To attach a data set to your test plan for validation:

1 At the test plan level, select TestPlan > Validation Data. The Select Data for Validation wizard appears.



2 Select a data set, and click Next.

3 If the input factors and responses required to evaluate the model do not appear in the selected data set, the next screen allows you to match signal names in the selected data set to those in the model. If signal names match you next see the screen to select tests.



Choose the tests from this data set to use. By default all tests are selected. For the currently selected test, the mean test values of all variables in this data set are displayed on the right.

4 Click Finish to use the selected tests to validate models in this test plan.

The validation data set appears in the Data Set information pane for the test plan. Validation RMSE is automatically added to the summary statistics for comparison in the bottom list view of response models in the test plan.

You can now use the validation data to validate all models except response features. You can see validation statistics in the following places:

Model List `Validation RMSE` appears in the summary statistics in the lower list of models at the test plan, response and one-stage nodes

At the local node view:

Pooled Statistics `Validation RMSE` The root mean squared error between the two-stage model and the validation data for all tests

Diagnostic Statistics > Local Diagnostics Local model `Validation RMSE` for the currently selected test (if validation data is available for the current test global variables must match)

Diagnostic Statistics > Summary Table `Validation RMSE` for the current test (if available) appears for local multiple models

Summary Table `Validation RMSE` for one-stage models

You can view validation plots in the following places:

Plots of `Validation residuals` For local and one-stage models in the Model Browser

From any model node except response features, you can select Model > Evaluate > Validation Data to open the Model Evaluation window and investigate the model with the selected validation data.

Similarly you can use the Model Evaluation window to investigate your models with other data, by using the Model > Evaluate > Other Data menu choice from a modeling node. The steps required are the same: select a data set, match signal names if necessary, and select tests to use.

Stepwise Regression

In this section...

What Is Stepwise? on page 6-65
Automatic Stepwise on page 6-65
Using the Stepwise Regression Window on page 6-66
Stepwise in the Model Building Process on page 6-71
PRESS statistic on page 6-74

What Is Stepwise?

Use the Stepwise functions to help you search for a good model fit. The goal of the stepwise search is to minimize PRESS. Minimizing Predicted Error Sum of Squares (PRESS) is a good method for working toward a regression model that provides good predictive capability over the experimental factor space. See PRESS statistic on page 6-74.

The use of PRESS is a key indicator of the predictive quality of a model. The predicted error uses predictions calculated without using the observed value for that observation. PRESS is known as `Delete-1` statistics in the Statistics and Machine Learning Toolbox product. See also Two-Stage Models for Engines on page 6-81.

You can choose automatic stepwise during model setup, or manual control using the Stepwise window.

Use the Stepwise menu in the Model Setup dialog boxes to run stepwise automatically when building linear models.

Open the stepwise regression window through the toolbar icon when you are in the global level view. The Stepwise tool provides a number of methods of selecting the model terms that should be included.

Automatic Stepwise

You can set the Minimize PRESS, Forward, and Backward selection routines to run automatically without the need to enter the stepwise figure.

You can set these options in the Model Setup dialog box, when you initially set up your test plan, or from the global level as follows:

1 Select Model > Set Up.

2 The Global Model Setup dialog box has a drop-down menu Stepwise, with the options None, Minimize PRESS, Forward selection, and Backward selection.

Using the Stepwise Regression Window

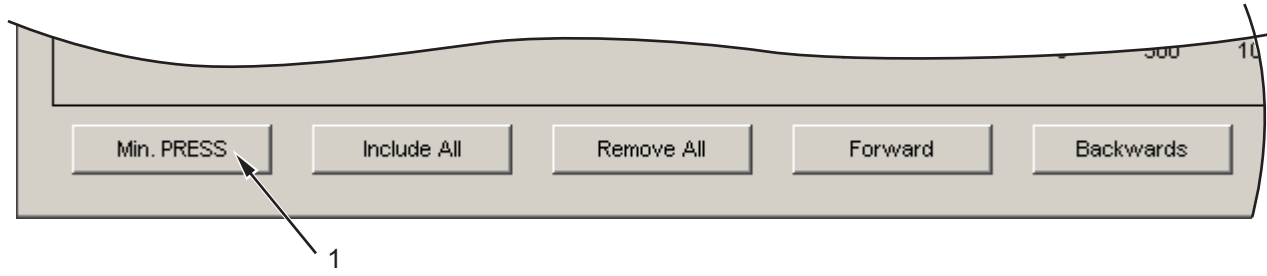
1 Use the stepwise command buttons at the bottom of the window (also available in the Regression menu) as follows:

Click Min. PRESS (labeled 1 in the following figure) to automatically include or remove terms to minimize PRESS. This procedure provides a model with improved predictive capability.

Include All terms in the model (except the terms flagged with Status as Never). This option is useful in conjunction with Min. PRESS and backward selection. For example, first click Include All, then Min. PRESS. Then you can click Include All again, then Backwards, to compare which gives the best result.

Remove All terms in the model (except the terms flagged with Status as Always). This option is useful in conjunction with forward selection (click Remove All, then Forwards).

Forwards selection adds all terms to the model that would result in statistically significant terms at the α% level (see Step 4 for alpha). The addition of terms is repeated until all the terms in the model are statistically significant.

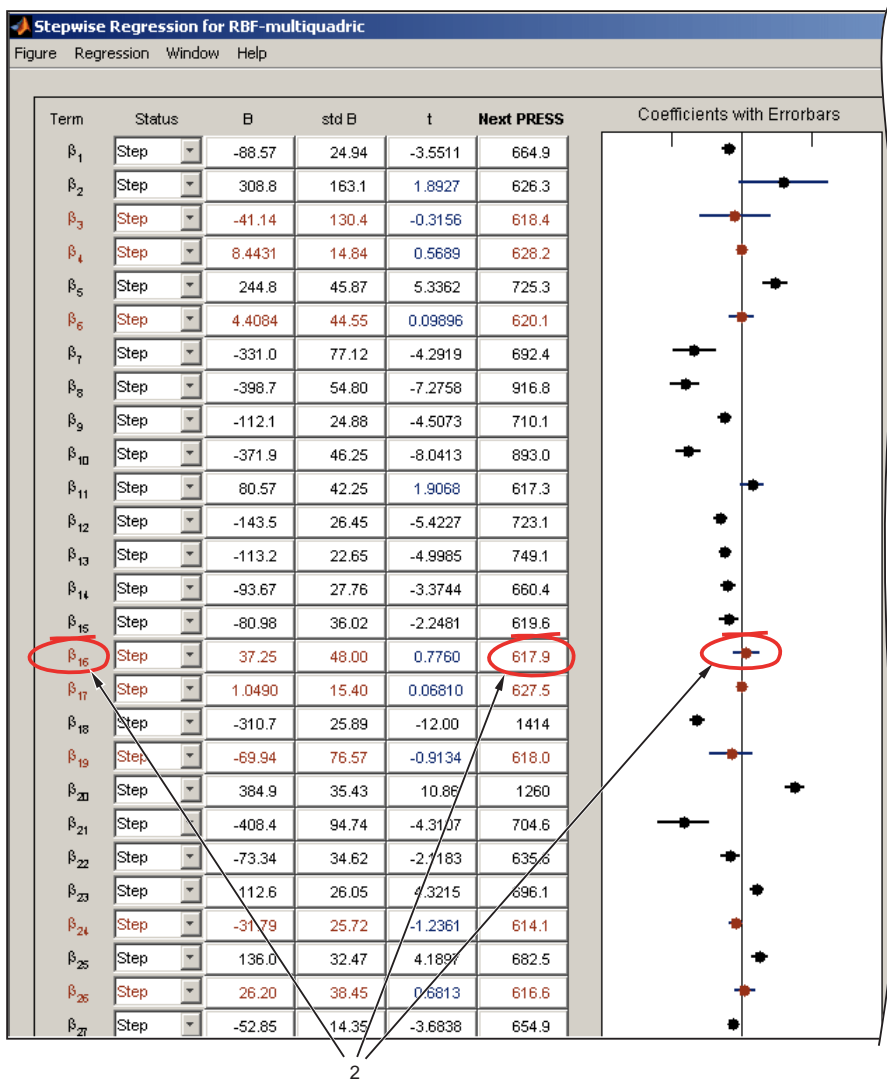


Backwards selection removes all terms from the model that are not statistically significant at the α% level. The removal of terms is repeated until all the terms in the model are statistically significant.

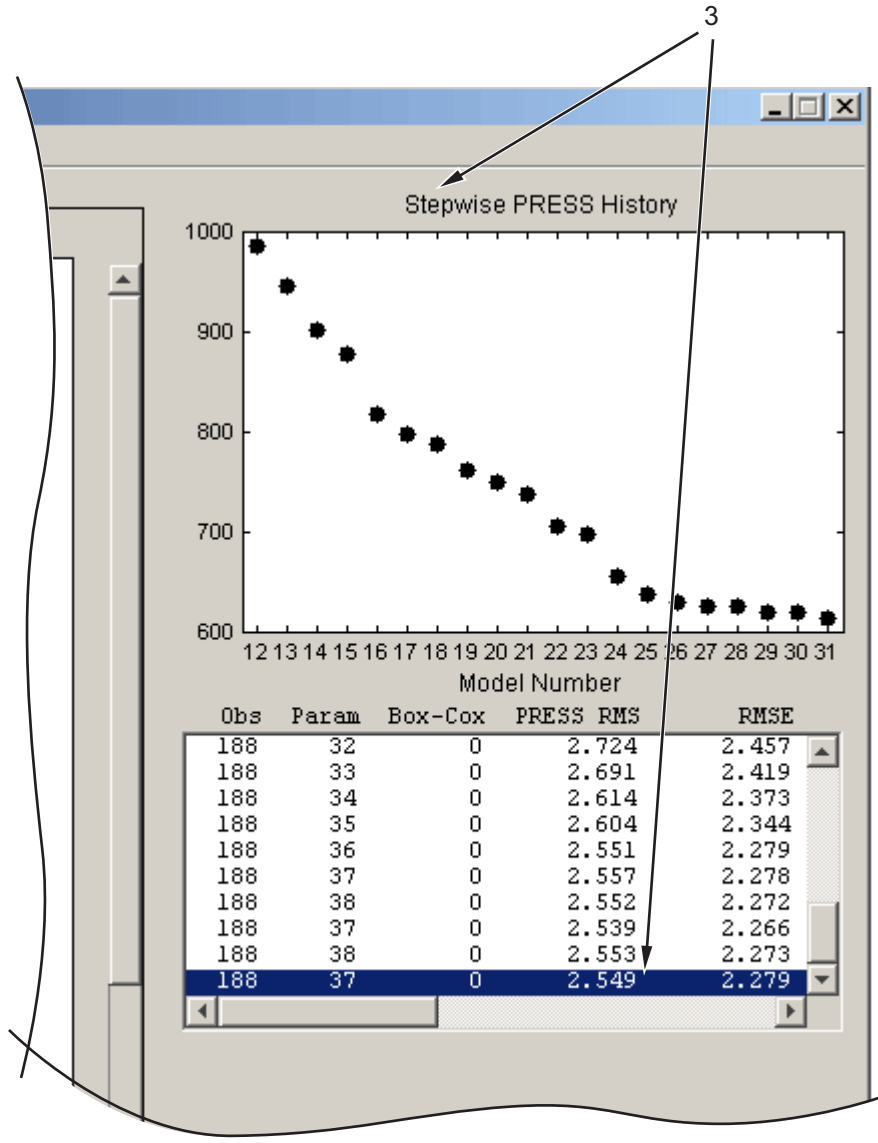
2 Terms can be also be manually included or removed from the model by clicking on the Term, Next PRESS, or coe icient error bar line (labeled 2 in the following gure).

The con dence intervals for all the coe icients are shown to the right of the table. Note that the interval for the constant term is not displayed, as the value of this coe icient is often signi cantly larger than other coe icients.

Terms that are currently not included in the model are displayed in red. See Stepwise Table for the meaning of the column headings.

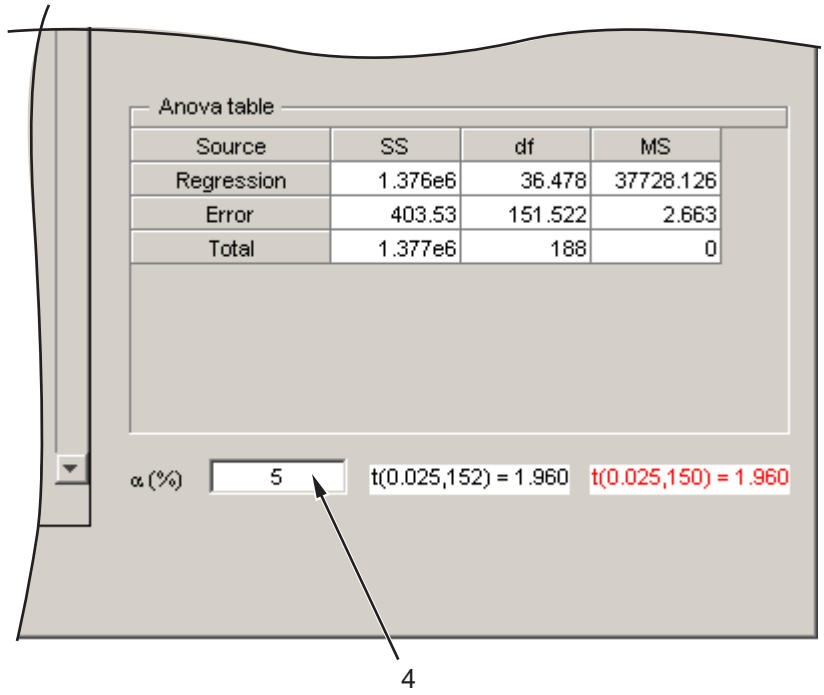


3 A history of the PRESS and summary statistics is shown on the right of the stepwise gure. You can return to a previous model by clicking an item in the list box or a point on the Stepwise PRESS History plot (labeled 3 in the following gure).



4 The critical values for testing whether a coe icient is statistically di erent from zero at the α% level are displayed at the bottom right side of the stepwise gure. You can

enter the value of α in the edit box (labeled 4 in the following figure) to the left of the critical values. The default is 5%. The ANOVA table is shown for the current model .



Any changes made in the stepwise figure automatically update the diagnostic plots in the Model Browser.

You can revert to the starting model when closing the Stepwise window. When you exit the Stepwise window, the Confirm Stepwise Exit dialog box asks `Do you want to update regression results?` You can click Yes (the default), No (to revert to the starting model), or Cancel (to return to the Stepwise window).

Stepwise Table

Term	Label for Coe cient
Status	Always. Stepwise does not remove this term. Never. Stepwise does not add this term. Step. Stepwise considers this term for addition or removal.
B	Value of coe icient. When the term is not in the model the value of the coe icient if it is added to the model is displayed in red.
stdB	Standard error of coe icient.
t	t value to test whether the coe icient is statistically di erent from zero. The t value is highlighted in blue if it is less than the critical value speci ed in the α% edit box (at bottom right).
Next PRESS	The value of PRESS if the inclusion or exclusion of this term is changed at the next iteration. A yellow highlighted cell indicates the next recommended term to change. Including or excluding the highlighted term (depending on its current state) will lead to the greatest reduction in the PRESS value. If there is no yellow highlighting this means that the PRESS value is already minimized. If there is a yellow cell, the column header is also yellow to alert you that you could make a change to achieve a smaller PRESS value. The column header is highlighted because you may need to scroll to nd the yellow cell.

The preceding table describes the meanings of the column headings in the Stepwise Regression window.

Stepwise in the Model Building Process

Once you have set up a model, you should create several alternative models, use the Stepwise functions and examine the diagnostic statistics to search for a good model t. For each response feature,

1 Begin by conducting a stepwise search.

You can do this automatically or by using the Stepwise window.

The goal of the stepwise search is to minimize PRESS. Usually not one but several candidate models per response features arise, each with a very similar PRESS R^2. The predictive capability of a model with a PRESS R^2 of 0.91 cannot be assumed superior in any meaningful engineering sense to a model with a PRESS R^2 of 0.909. Further, the nature of the model building process is that the improvement in PRESS R^2 o ered by the last few terms is often very small. Consequently, several candidate models can arise. You can store each of the candidate models and associated diagnostic information separately for subsequent review. Do this by making a selection of child nodes for the response feature.

However, experience has shown that a model with a PRESS R^2 of less than 0.8, say, is of little use as a predictive tool for engine mapping purposes. This criterion must be viewed with caution. Low PRESS R^2 values can result from a poor choice of the original factors but also from the presence of outlying or in uential points in the data set. Rather than relying on PRESS R^2 alone, a safer strategy is to study the model diagnostic information to discern the nature of any fundamental issues and then take appropriate corrective action.

2 Once the stepwise process is complete, the diagnostic data should be reviewed for each candidate model.

It might be that these data alone are su icient to provide a means of selecting a single model. This would be the case if one model clearly exhibited more ideal behavior than the others. Remember that the interpretation of diagnostic plots is subjective.

3 You should also remove outlying data at this stage. You can set criteria for detecting outlying data. The default criterion is any case where the absolute value of the external studentized residual is greater than 3.

4 After removing outlying data, continue the model building process in an attempt to remove further terms.

High-order terms might have been retained in the model in an attempt to follow the outlying data. Even after removing outlying data, there is no guarantee that the diagnostic data will suggest that a suitable candidate model has been found. Under these circumstances,

5 A transform of the response feature might prove bene cial.

A useful set of transformations is provided by the Box and Cox family, see Box-Cox Transformation on page 6-77. Note that the Box-Cox algorithm is model dependent and as such is always carried out using the (Nxq) regression matrix X.

6 After you select a transform, you should repeat the stepwise PRESS search and select a suitable subset of candidate models.

7 After this you should analyze the respective diagnostic data for each model.

It might not be apparent why the original stepwise search was carried out in the natural metric. Why not proceed directly to taking a transformation? This seems sensible when it is appreciated that the Box-Cox algorithm often, but not always, suggests that a contractive transform such as the square root or log be applied. There are two main reasons for this:

The primary reason for selecting response features is that they possess a natural engineering interpretation. It is unlikely that the behavior of a transformed version of a response feature is as intuitively easy to understand.

Outlying data can strongly in uence the type of transformation selected. Applying a transformation to allow the model to t bad data well does not seem like a prudent strategy. By bad data it is assumed that the data is truly abnormal and a reason has been discovered as to why the data is outlying; for example, The emission analyzer was purging while the results were taken.

Finally, if you cannot nd a suitable candidate model on completion of the stepwise search with the transformed metric, then a serious problem exists either with the data or with the current level of engineering knowledge of the system. Model augmentation or an alternative experimental or modeling strategy should be applied in these circumstances.

After these steps it is most useful to validate your model against other data (if any is available). See Model Evaluation Window on page 6-60.

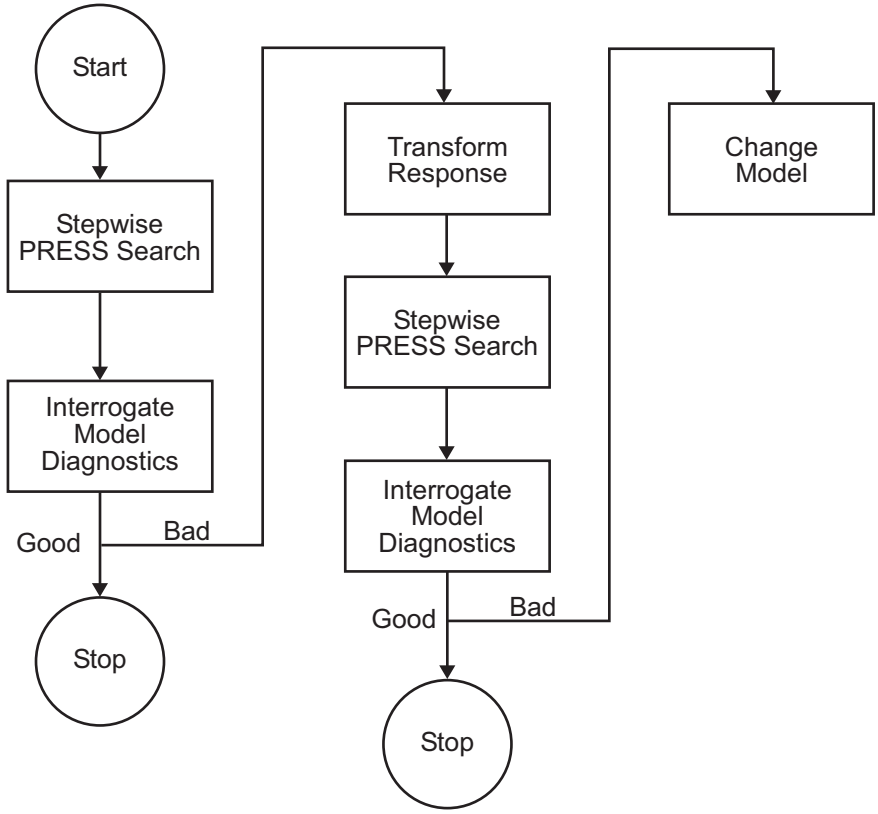
See also these guideline pages with links to information about each of the steps involved in creating one and two-stage models and then searching for the best t:

Fit a One-Stage Model on page 1-7

Fit a Two-Stage Model on page 1-10

Create Alternative Models to Compare on page 5-84

The recommended overall stepwise process is best viewed graphically, as shown in the following ow chart.



Note that the process depicted in the preceding diagram should be carried out for each member of the set of response features associated with a given response and then repeated for the remaining responses.

PRESS statistic

With n runs in the data set, the model equation is tted to n-1 runs and a prediction taken from this model for the remaining one. The di erence between the recorded data value and the value given by the model (at the value of the omitted run) is called a prediction residual. PRESS is the sum of squares of the prediction residuals. The square root of PRESS/n is PRESS RMSE (root mean square prediction error).

Note that the prediction residual is di erent from the ordinary residual, which is the di erence between the recorded value and the value of the model when tted to the whole data set.

The PRESS statistic gives a good indication of the predictive power of your model, which is why minimizing PRESS is desirable. It is useful to compare PRESS RMSE with RMSE as this may indicate problems with over tting. RMSE is minimized when the model gets very close to each data point; 'chasing' the data will therefore improve RMSE. However chasing the data can sometimes lead to strong oscillations in the model between the data points; this behavior can give good values of RMSE but is not representative of the data and will not give reliable prediction values where you do not already have data. The PRESS RMSE statistic guards against this by testing how well the current model would predict each of the points in the data set (in turn) if they were not included in the regression. To get a small PRESS RMSE usually indicates that the model is not overly sensitive to any single data point.

For more information see Guidelines for Selecting the Best Model Fit on page 6-54, Stepwise Regression on page 6-65 and Toolbox Terms and Statistics De nitions on page 6-91.

Note that calculating PRESS for the two-stage model applies the same principle (tting the model to n-1 runs and taking a prediction from this model for the remaining one) but in this case the predicted values are rst found for response features instead of data points. The predicted value, omitting each test in turn, for each response feature is estimated. The predicted response features are then used to reconstruct the local curve for the test and this curve is used to obtain the two-stage predictions. This is applied as follows:

To calculate two stage PRESS:

1 For each test, S, do the following steps:

   For each of the response features, calculate what the response feature predictions would be for S (with the response features for S removed from the calculation).

   This gives a local prediction curve C based on all tests except S.

   For each data point in the test, calculate the di erence between the observed value and the value predicted by C.

2 Repeat for all tests.

Sum the square of all of the di erences found and divide by the total number of data points.

Box-Cox Transformation

You can apply a Box-Cox transform to any one-stage or response feature model node (any models with a global icon) by selecting Model > Set Up and entering a number for lambda in the Box-Cox edit box.

For linear models (polynomials, polynomial splines, and RBFs) you can also use the Box-Cox Transformation dialog box described in the following section, by using the toolbar button or Model menu item Box-Cox Transform.

You might want to transform a response feature either to correct for nonnormality and/or a heteroscedastic variance structure. A useful class of transformations for this purpose is the power transform y^{λ}, where is a parameter to be determined. Box and Cox (1964) showed how and the regression coe icients themselves could be estimated simultaneously using the method of maximum likelihood. The procedure consists of conducting a standard least squares t using

$$y^{(\lambda)} = \frac{y^{\lambda} - 1}{\lambda y^{\lambda - 1}} \qquad \text{for} \qquad \lambda \neq 0$$

$$y^{(\lambda)} = y \ln(y) \qquad \text{for} \qquad \lambda = 0$$

where the so called geometric mean of the observations is given by

$$\dot{y} = \exp\left[\frac{\sum_{i=1}^{N} \ln(y_i)}{N}\right]$$

The maximum likelihood estimate of corresponds to the value for which the SSE( ) from the tted model is a minimum. This value of is determined by tting a model (assumed throughout to be de ned by the regression matrix for the full model - X) for various levels of and choosing the value corresponding to the minimum SSE( ). A plot of SSE( ) versus is often used to facilitate this choice.

The parameter is swept between the range of -3 to 3 in increments of 0.5.

You can enter a value for lambda in the edit box that approaches the point on the plot with the smallest SSE.

Although SSE( ) is a continuous function of , simple choices for are recommended. This is because the practical di erence between 0.5 and 0.593, say, is likely to be very small but a simple transform like 0.5 is much easier to interpret.

You can also nd an approximate 100(1-α) con dence interval on l by computing

$$SS^{*} = SSE(\lambda)\left[1 + \frac{t_{\alpha/(2,\upsilon)}}{\upsilon}\right]$$

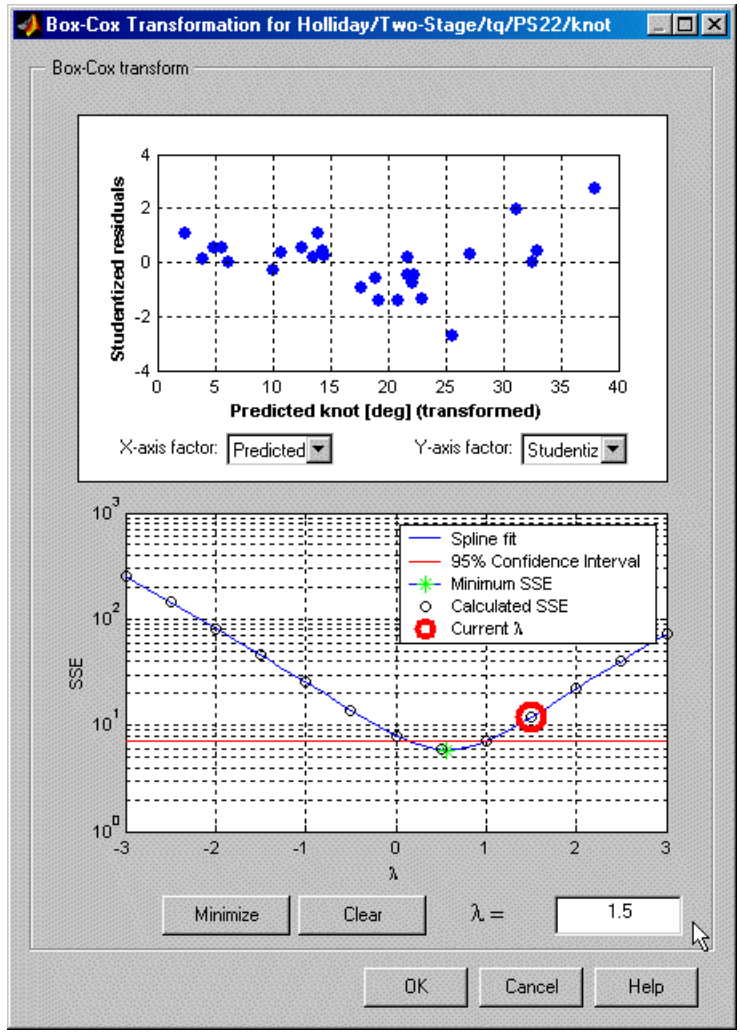
where υ is the number of residual degrees of freedom equal to (N-q).

In this formula is understood to be the value that minimizes SSE( ). Note that this con dence interval might encompass more than one incremental value for . In this case, any of these values is as valid as any other and you can select any of these transformations from which to develop trial models.

You should always look at the residuals plots at the top to see the e ect of di erent transforms.

You can create several child nodes of a single model and choose di erent transforms for each to compare them using the rest of the Model Browser tools.

For the sake of clarity, consider the following example, which illustrates the results of applying the Box-Cox algorithm to a polyspline torque model.



In this example the minimum value of SSE( ) occurs near to =0. The minimum is marked in green. The 95% con dence limit has been calculated and drawn on the gure as a red solid line. It is apparent in this example that, after rounding to the nearest incremental value contained within the con dence interval, any in the range $0 \le \lambda \le 1$ is appropriate. Of the three possible increments, 0, 0.5, and 1, = 0.5 is the closest to the minimum SSE.

You can select any point on the plot by clicking. The chosen point (current lambda) is then outlined in red. You can also enter values of lambda directly in the edit box and press Enter.

Two-Stage Models for Engines

In this section...

Overview of the Mathematics of Two-Stage Models on page 6-81
Local Models on page 6-83
Local Covariance Modeling on page 6-83
Response Features on page 6-84
Global Models on page 6-85
Two-Stage Models on page 6-86
Global Model Selection on page 6-87
Initial Values for Covariances on page 6-87
Quasi-Newton Algorithm on page 6-88
Expectation Maximization Algorithm on page 6-88
References on page 6-88
Linear Regression on page 6-89
Toolbox Terms and Statistics De nitions on page 6-91

Overview of the Mathematics of Two-Stage Models

This section contains an overview of the mathematics of two-stage models. A comprehensive reference for two-stage modeling is Davidian and Giltinan [3]. The information is divided into the following sections:

Lindstrom and Bates [6] de ne repeated measurements as data generated by observing a number of individuals repeatedly under various experimental conditions, where the individuals are assumed to constitute a random sample from a population of interest. An important class of repeated measurements is longitudinal data where the observations are ordered by time or position in space. More generally, longitudinal data is de ned as repeated measurements where the observations on a single individual are not, or cannot be, randomly assigned to the levels of a treatment of interest.

Modeling data of this kind usually involves the characterization of the relationship between the measured response, y, and the repeated measurement factor, or covariate x. Frequently, the underlying systematic relationship between y and x is nonlinear. In some cases the relevant nonlinear model can be derived on physical or mechanistic grounds.

However, in other contexts a nonlinear relationship might be imposed simply to provide a convenient empirical description for the data. The presence of repeated observations on an individual requires particular care in characterizing the variation in the experimental data. In particular, it is important to represent two sources of variation explicitly: random variation among measurements within a given individual (intraindividual) and random variation among individuals (interindividual). Inferential procedures accommodate these di erent variance components within the framework of an appropriate hierarchical statistical model. This is the fundamental idea behind the analysis of repeated measurement data.

Holliday [1,2] was perhaps the rst to apply nonlinear repeated measurements analysis procedures to spark ignition engine data. The focus of Holliday's work was the modeling of data taken from engine mapping experiments. In these experiments, engine speed, load, and air/fuel ratio were held constant while spark was varied. Various engine response characteristics, for example, torque or emission quantities, were measured at each spark setting. Holliday modeled the response characteristics for each sweep as a function of spark advance. Variations in the individual sweep parameters were then modeled as a function of the global engine operating variables speed, load, and air/fuel ratio. Conceptually, variations in the measurements taken within a sweep represent the intraindividual component of variance. Similarly, variation in the sweep-speci c parameters between sweeps represents the interindividual component of variance. You can generalize these principles to other steady-state engine modeling exercises where the nature of data collection usually involves sweeping a single engine control variable while the remainder are held at xed values. These points suggest that nonlinear repeated measurements analysis represents a general approach to the parameterization of mean value engines models for controls-oriented development.

Another application for models of this form is the ow equations for a throttle body. Assuming the ow equations are based upon the usual one-dimensional isentropic ow principle, then they must be modi ed by an e ective area term, A_e, which accounts for the fact that the true ow is multidimensional and irreversible. You can map the throttle ow characteristics by sweeping the throttle position at xed engine speed. This data collection methodology naturally imposes a hierarchy the analysis of which is consistent with the application of nonlinear repeated measures. Experience in modeling e ective area suggests that free knot spline or biological growth models provide good local predictions. The global phase of the modeling procedure is concerned with predicting the systematic variation in the response features across engine speed. A free knot spline model has proven useful for this purpose.

Local Models

Modeling responses locally within a sweep as a function of the independent variable only. That is,

$$y_i^j = f_i(s_i^j, \theta_i) + \varepsilon_i^j \qquad \text{for } j = 1, 2, \ldots m_i \tag{6-1}$$

where the subscript i refers to individual tests and j to data within a test, s_i^j is the j[th] independent value, θ_i is a (rx1) parameter vector, y_i^j is the j [th] response, and ε_i^j is a normally distributed random variable with zero mean and variance ^{2}. Note that equation (4 1) can be either a linear or a nonlinear function of the curve t parameters. The assumption of independently normally distributed errors implies that the least squares estimates of θ are also maximum likelihood parameters.

Local Covariance Modeling

The local model describes both the systematic and random variation associated with measurements taken during the i[th] test. Systematic variation is characterized through the function f while variation is characterized via the distributional assumptions made on the vector of random errors e_i. Hence, speci cation of a model for the distribution of e_i completes the description of the intratest model. The Model-Based Calibration Toolbox product allows a very general speci cation of the local covariance:

$$e_i - N(0, \sigma^2 C_i(\beta_i, \xi_i)) \tag{6-2}$$

where C_i is an (n_i x n_i) covariance matrix, σ^2 is the coe icient of variation, and $_i$ is a (q-by-1) vector of dispersion parameters that account for heterogeneity of variance and the possibility of serially correlated data. The speci cation is very general and a ords considerable exibility in terms of specifying a covariance model to adequately describe the random component of the intratest variation.

The Model-Based Calibration Toolbox product supports the following covariance models:

Power Variance Model:

$$C_i = diag\left\{ f(x_i, \beta_i)^{\xi_1} \right\} \tag{6-3}$$

Exponential Variance Model:

$$C_i = diag\{\exp(f(x_i,\beta_i)\xi_1)\} \tag{6-4}$$

Mixed Variance Model:

$$C_i = diag\left\{\xi_1 + f(x_i,\theta_i)^{\xi_2}\right\} \tag{6-5}$$

where diag{x} is a diagonal matrix.

Correlation models are only available for equispaced data in the Model-Based Calibration Toolbox product. It is possible to combine correlation models with models with the variance models such as power.

One of the simplest structures that can be used to account for serially correlated errors is the AR(m) model (autoregressive model with lag m). The general form of the AR(m) model is

$$e_j = \phi_1 e_{j-1} + \phi_2 e_{j-2} + \ldots + \phi_m e_{j-m} + v_j \tag{6-6}$$

where ϕ_k is the k[th] lag coe icient and v_j is an exogenous stochastic input identically and independently distributed as $N(0,\sigma_v^2)$. First- and second-order autoregressive models are implemented in the Model-Based Calibration Toolbox product.

Another possibility is a moving average model (MA). The general structure is

$$e_j = \phi_1 v_{j-1} + \phi_2 v_{j-2} + \ldots + \phi_m v_{j-m} + v_j \tag{6-7}$$

where ϕ_k is the k[th] lag coe icient and v_j is an exogenous stochastic input identically and independently distributed as $N(0,\sigma_v^2)$. Only a rst-order moving average model is implemented in the Model-Based Calibration Toolbox product.

Response Features

From an engineering perspective, the curve t parameters do not usually have any intuitive interpretation. Rather characteristic geometric features of the curve are of interest. The terminology response features of Crowder and Hand [8] is used to

describe these geometric features of interest. In general, the response feature vector p_i for the i^{th} sweep is a nonlinear function (g) of the corresponding curve t parameter vector θ_i, such that

$$p_i = g(\theta_i) \tag{6-8}$$

Global Models

Modeling the variation in the response features as a function of the global variables. The response features are carried through to the second stage of the modeling procedure rather than the curve t parameters because they have an engineering interpretation. This ensures that the second stage of the modeling process remains relatively intuitive. It is much more likely that an engineer will have better knowledge of how a response feature such as MBT behaves throughout the engine operating range (at least on a main e ects basis) as opposed to an esoteric curve t parameter estimate.

The global relationship is represented by one of the global models available in the Model-Based Calibration Toolbox product. In this section we only consider linear models that can be represented as

$$P_i = X_i\beta + \gamma_i \text{ for } i=1,2,..,r \tag{6-9}$$

where the X_i contains the information about the engine operating conditions at the i^{th} spark sweep, is the vector of global parameter estimates that must be estimated by the tting procedure, and γ_i is a vector of normally distributed random errors. It is necessary to make some assumption about the error distribution for γ, and this is typically a normal distribution with

$$\gamma_i - N_r(0,D) \tag{6-10}$$

where r is the number of response features. The dimensions of D are (rxr) and, being a variance-covariance matrix, D is both symmetric and positive de nite. Terms on the leading diagonal of D represent the test-to-test variance associated with the estimate of the individual response features. O -diagonal terms represent the covariance between pairs of response features. The estimation of these additional covariance terms in a multivariate analysis improves the precision of the parameter estimates.

Two-Stage Models

To unite the two models, it is first necessary to review the distributional assumptions pertaining to the response feature vector p_i. The variance of p_i (Var(p_i)) is given by

$$Var(\mathbf{p}_i) = \left[\frac{\partial g(\theta_i)}{\partial \theta}\right]\sigma^2 C_i\left[\frac{\partial g(\theta_i)}{\partial \theta}\right]^T \tag{6-11}$$

For the sake of simplicity, the notation ^2C_i is to denote Var(p_i). Thus, p_{ii} is distributed as

$$\mathbf{p}_i - N_r(\mathbf{p}_i, \sigma^2 C_i) \tag{6-12}$$

where C_i depends on f_i through the variance of θ_i and also on g_i through the conversion of θ_i to the response features p_i. Two standard assumptions are used in determining C_i: the asymptotic approximation for the variance of maximum likelihood estimates and the approximation for the variance of functions of maximum likelihood estimates, which is based on a Taylor series expansion of g_i. In addition, for nonlinear f_i or g_i, C_i depends on the unknown θ_i ; therefore, we will use the estimate $\hat{\theta}_i$ in its place. These approximations are likely to be good in the case where ^{2} is small or the number of points per sweep (m_i) is large. In either case we assume that these approximations are valid throughout.

We now return to the issue of parameter estimation. Assume that the γ_i are independent of the ε_i^j. Then, allowing for the additive replication error in response features, the response features are distributed as

$$\mathbf{p}_i - N(X_i\beta, \sigma^2 C_i + D) \tag{6-13}$$

When all the tests are considered simultaneously, equation (6-13) can be written in the compact form

$$P - N(Z\beta, W(\varpi)) \tag{6-14}$$

where P is the vector formed by stacking the n vectors p_i on top of each other, Z is the matrix formed by stacking the n X_i matrices, W is the block diagonal weighting matrix with the matrices on the diagonal being $^2C_i + D$, and is a vector of dispersion

parameters. For the multivariate normal distribution (6-14) the negative log likelihood function can be written:

$$\log L(\beta, \varpi) = \log|W| + (P - Z\beta)' W^{-1} (P - Z\beta) \tag{6-15}$$

Thus, the maximum likelihood estimates are the vectors $_{ML}$ and $_{ML}$ that minimize logL(,). Usually there are many more t parameters than dispersion parameters; that is, the dimension of is much larger than . As such, it is advantageous to reduce the number of parameters involved in the minimization of logL(,). The key is to realize that equation (6-15) is conditionally linear with respect to . Hence, given estimates of , equation (6-15) can be di erentiated directly with respect to and the resulting expression set to zero. This equation can be solved directly for as follows:

$$\beta = (Z' W^{-1} Z)^{-1} (Z' W^{-1} P) \tag{6-16}$$

The key point is that now the likelihood depends only upon the dispersion parameter vector , which as already discussed has only modest dimensions. Once the likelihood is minimized to yield $_{ML}$, then, since W($_{ML}$) is then known, equation (6-16) can subsequently be used to determine $_{ML}$.

Global Model Selection

Before undertaking the minimization of Equation 6-15 (see Two-Stage Models on page 6-86) it is rst necessary to establish the form of the X_i matrix. This is equivalent to establishing a global expression for each of the response features a priori. Univariate stepwise regression is used to select the form of the global model for each response feature. Minimization of the appropriate PRESS statistic is used as a model building principle, as speci ed in Stepwise in the Model Building Process on page 6-71. The underlying principle is that having used univariate methods to establish possible models, maximum likelihood methods are subsequently used to estimate their parameters.

Initial Values for Covariances

An initial estimate of the global covariance is obtained using the standard two-stage estimate of Steimer et al. [10],

$$D_{STS} = \frac{1}{r-1} \sum_{i=1}^{r} (p_i - X_i \beta)(p_i - X_i \beta)^T \tag{6-17}$$

where are the estimates from all the univariate global models. This estimate is biased.

Quasi-Newton Algorithm

Implicit to the minimization of equation (6-17) is that D is positive de nite. It is a simple matter to ensure this by noting that D is positive de nite if and only if there is an upper triangular matrix, G, say, such that

$$D = G'G \tag{6-18}$$

This factorization is used in the Quasi-Newton algorithm. Primarily, the advantage of this approach is that the resulting search in G, as opposed to D, is unconstrained.

Expectation Maximization Algorithm

The expectation maximization algorithm is an iterative method that converges toward the maximal solution of the likelihood function. Each iteration has two steps:

1 Expectation Step Produce re ned estimates of the response features given the current parameter estimates.

2 Maximization Step Obtain new estimates of the parameters (global model parameters and covariance matrix) for the new response features.

These steps are repeated until the improvement in value of the log likelihood function is less than the tolerance. Details of the algorithm can be found in [3, Chapter 5].

References

1 Holliday, T., The Design and Analysis of Engine Mapping Experiments: A Two-Stage Approach, Ph.D. thesis, University of Birmingham, 1995.

2 Holliday, T., Lawrance, A. J., Davis, T. P., Engine-Mapping Experiments: A Two-Stage Regression Approach, Technometrics, 1998, Vol. 40, pp 120-126.

3 Davidian, M., Giltinan, D. M., Nonlinear Models for Repeated Measurement Data, Chapman & Hall, First Edition, 1995.

4 Davidian, M., Giltinan, D. M., Analysis of repeated measurement data using the nonlinear mixed e ects model, Chemometrics and Intelligent Laboratory Systems, 1993, Vol. 20, pp 1-24.

5 Davidian, M., Giltinan, D. M., Analysis of repeated measurement data using the nonlinear mixed e ects model, Journal of Biopharmaceutical Statistics, 1993, Vol. 3, part 1, pp 23-55.

6 Lindstrom, M. J., Bates, D. M., Nonlinear Mixed E ects Models for Repeated Measures Data, Biometrics, 1990, Vol. 46, pp 673-687.

7 Davidian, M., Giltinan, D. M., Some Simple Methods for Estimating Intraindividual Variability in Nonlinear Mixed E ects Models, Biometrics, 1993, Vol. 49, pp 59-73.

8 Hand, D. J., Crowder, M. J., Practical Longitudinal Data Analysis, Chapman and Hall, First Edition, 1996.

9 Franklin, G.F., Powell, J.D., Workman, M.L., Digital Control of Dynamic Systems, Addison-Wesley, Second Edition, 1990.

10 Steimer, J.-L., Mallet, A., Golmard, J.L., and Boisvieux, J.F., Alternative approaches to estimation of population pharmacokinetic parameters: Comparison with the nonlinear mixed e ect model. Drug Metabolism Reviews, 1984, 15, 265-292.

Linear Regression

Building a regression model that includes only a subset of the total number of available terms involves a trade-o between two con icting objectives:

Increasing the number of model terms always reduces the Sum Squared Error.

However, you do not want so many model terms that you over t by chasing points and trying to t the model to signal noise. This reduces the predictive value of your model.

The best regression equation is the one that provides a satisfactory trade-o between these con icting goals, at least in the mind of the analyst. It is well known that there is no unique de nition of best. Di erent model building criteria (for example, forward selection, backward selection, PRESS search, stepwise search, Mallows C_p Statistic...) yield di erent models. In addition, even if the optimal value of the model building statistic is found, there is no guarantee that the resulting model will be optimal in any other of the accepted senses.

Principally the purpose of building the regression model for calibration is for predicting future observations of the mean value of the response feature. Therefore the aim is to select the subset of regression terms such that PRESS is minimized. Minimizing PRESS is consistent with the goal of obtaining a regression model that provides good predictive capability over the experimental factor space. This approach can be applied to both polynomial and spline models. In either case the model building process is identical.

1 The regression matrix can be viewed in the Design Evaluation Tool. Terms in this matrix de ne the full model. In general, the stepwise model is a subset of this full term set.

2 All regressions are carried out with the factors represented on their coded scales (-1,1).

Toolbox Terms and Statistics De nitions

De nitions

Symbol	De nition
N	Number of data points
p	Number of terms currently included in the model
q	Total number of possible model parameters (q=p+r)
r	Number of terms not currently included from the model
y	(Nx1) response vector
X	Regression matrix. X has dimensions (Nxq)
X_p	(Nxp) model matrix corresponding to terms currently included in the model
X_r	(Nxr) matrix corresponding to terms currently excluded from the model
β_p	(px1) vector of model coe icients $\beta_p = \{\beta_1, \beta_2, \ldots, \beta_p\}$ $\hat{\beta} = (X^T X)^{-1} X^T y$ $\mathrm{var}\hat{\beta} = (X^T X)^{-1} \mathrm{MSE}$
PEV	Prediction Error Variance $\mathrm{PEV}(x) = \mathrm{var}y(\hat{y}) = x(X^T X)^{-1} x^T \mathrm{MSE}$
α	User-de ned threshold criteria for automatically rejecting terms
$\hat{y}$	(Nx1) vector of predicted responses. $\hat{y} = X_p \beta_p$
e	(Nx1) residual vector. $e = (y - \hat{y})$

Symbol	Definition
$e_{(i)}$	(Nx1) vector of PRESS residuals. $e_{(i)} = e_i/(1-H_{ii})$
H	Hat matrix. $X(X'X)^{-1}X'$
L	(Nx1) vector of leverage values. $L = \{l_1, l_2, \ldots, l_N\}' = \{H_{11}, H_{22}, \ldots, H_{NN}\}$
VIF	Variance Inflation Factors
SSE	Error Sum of Squares. SSE = e'e
SSR	Regression Sum of Squares. SSE = $\sum e_i^2$
SST	Total Sum of Squares. SST = y'y - $N\bar{y}^2$
MSE	Mean Square Error. MSE = SSE/(N-p)
MSR	Mean Square of Regression. MSR = SSR/P
F	F-statistic. F = MSR/MSE
$MSE_{(i)}$	MSE calculated with i^{th} point removed from the data set. $MSE_{(i)} = \frac{(N-p)MSE - e_i^2/(1-H_{ii})}{N-p-1}$
RMSE	Root Mean Squared Error: the standard deviation of regression. $RMSE = \sqrt{MSE}$
s_i	i^{th} R-Student or Externally Scaled Studentized Residual. $s_i = \frac{e_i}{\sqrt{MSE_{(i)}(1-H_{ii})}}$

Symbol	Definition
r_i	i^{th} Standardized or Internally Scaled Studentized Residual. $$r_i = \frac{e_i}{\sqrt{MSE(1-H_{ii})}}$$
D	Cook's D Influence Diagnostic. $$D_i = \frac{r_i^2 H_{ii}}{p(1-H_{ii})}$$
SEBETA	(px1) vector of model coefficient standard errors. $$SEBETA = MSE\{\sqrt{c_{11}}, \sqrt{c_{22}}, \ldots, \sqrt{c_{pp}}\}' \quad \text{where} \quad c = (X^T X)^{-1}$$
PRESS	Predicted Error Sum of Squares. PRESS = $e'_{(i)}e_{(i)}$

For more on PRESS and other displayed statistics, see PRESS statistic on page 6-74, Guidelines for Selecting the Best Model Fit on page 6-54, and Pooled Statistics on page 6-12.

Export Models to Simulink

In this section...

Export Models for Simulation on page 6-94
Use Statistical Models for Plant Modeling and Optimization on page 6-96
Use Statistical Models for Hardware-in-the-Loop Testing on page 6-97

Export Models for Simulation

You can export statistical models developed in the Model Browser to a Simulink model for simulation and hardware-in-the-loop (HIL) testing.

1 From the Model Browser test plan or any model node, select File > Export Models. The Export Models dialog box opens.