ETAS ASCMO-STATIC V5.16

User Guide

www.etas.com

ASCMO-STATIC V5.16 I User Guide R01 EN I 11.2025

Contents

1 Introduction

In this chapter, you can find information about the intended use, the addressed target group, and information about safety and privacy related topics.

Please adhere to the ETAS Safety Advice (**Help › Safety Advice**) and to the safety information given in the user documentation.

ETAS GmbH cannot be made liable for damage which is caused by incorrect use and not adhering to the safety messages.

1.1 Demands on Technical State of the Product

The following special requirements are made to ensure safe operation:

- Take all information on environmental conditions into consideration before setup and operation (see the documentation of your computer, hardware, etc.).

1.2 Intended Use

The ETAS ASCMO tool family is intended for offline data-based modeling, model-based calibration, or efficient optimization of parameters in physics-based models. It is not intended to operate directly in a running system.

With ASCMO-STATIC and ASCMO-DYNAMIC, it is possible to accurately model the behavior of complex systems based on a small set of measurement data. This model can either be used to analyze and optimize input parameters or as a black box plant model in other simulation environments. In contrast, ASCMO-MOCA typically uses existing physics based-models with a defined structure to calibrate and optimize the parameters of the model itself. The results are a suggestion and must be additionally validated before further processing.

ETAS GmbH cannot be held liable for damage which is caused by incorrect use and not adhering to the safety information. See **Help › Safety Advice**.

1.3 Target Group

This product is intended for trained and qualified personnel in the development and calibration sector of motor vehicle ECUs. Technical knowledge in measuring and control unit engineering is a prerequisite.

1.4 Classification of Safety Messages

Safety messages warn of dangers that can lead to personal injury or damage to property:

DANGER

DANGER indicates a hazardous situation that, if not avoided, will result in death or serious injury.

WARNING

WARNING indicates a hazardous situation that, if not avoided, could result in death or serious injury.

CAUTION

CAUTION indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.

NOTICE

NOTICE indicates a situation that, if not avoided, could result in damage to physical property.

ATTENTION

ATTENTION indicates a situation that, if not avoided, could result in damage to digital property like data loss, data corruption and system vulnerability.

1.5 Safety Information

> ***NOTICE***
>
> **Damage due to wrong test plan**
>
> Wrong engine settings in ASCMO-STATIC ExpeDes can lead to engine or test bench damage. Example: the operation point overstresses the engine and causes damage, e.g. by setting an ignition angle that causes extensive knocking.
>
> – The general settings for the test plan must fit the system and the object. Negative example: 10000 rpm are set in the test plan vs. the motor has max. 6000 rpm.
> – Limit the operation points to the allowed values. ETAS ASCMO does not have any knowledge about the engine parameters.
> – Limit the engine load in the general settings before exporting the test plan.
> – Verify the test plan for further use.

For ASCMO-STATIC ExpeDes see 7.11 "Step 9: Export " on page 217and 7.2 "Step 1: General Settings" on page 187.

1.6 Data Protection

If the product contains functions that process personal data, legal requirements of data protection and data privacy laws shall be complied with by the customer. As the data controller, the customer usually designs subsequent processing. Therefore, he must check if the protective measures are sufficient.

1.7 Data and Information Security

To securely handle data in the context of this product, see the next sections about data and storage locations as well as technical and organizational measures.

1.7.1 Data and Storage Locations

The following sections give information about data and their respective storage locations for various use cases.

1.7.1.1 License Management

When using the ETAS License Manager in combination with user-based licenses that are managed on the FNP license server within the customer's network, the following data are stored for license management purposes:

Data

- Communication data: IP address
- User data: Windows user ID

Storage location

- FNP license server log files on the customer network

When using the ETAS License Manager in combination with host-based licenses that are provided as FNE machine-based licenses, the following data are stored for license management purposes:

Data

- Activation data: Activation ID
  - Used only for license activation, but not continuously during license usage

Storage location

- FNE trusted storage

  `C:\ProgramData\ETAS\FlexNet\fne\license\ts`

1.7.2 Technical and Organizational Measures

We recommend that your IT department takes appropriate technical and organizational measures, such as classic theft protection and access protection to hardware and software.

2 About ETAS ASCMO

ASCMO (**A**dvanced **S**imulation for **C**alibration, **M**odeling and **O**ptimization) is a tool for modeling the input/output behavior of unknown systems based on measuring data obtained using methods of the design of experiments.

This data-based modeling is necessary and successful when a precise physical description of the system is not possible. The high model quality that can be achieved allows for mapping even complex relationships, such as the global behavior of an internal combustion engine.

After modeling, ETAS ASCMO offers a variety of possibilities for visualizing the system behavior and for calibration/optimization based on models. The focal point of the calibration is the modeling and optimization of the "internal combustion engine" system in support of the calibration.

However, the modeling and optimization methods can also be applied to any other systems in which the output variables are differentiably dependent on the input variables.

2.1 ASCMO-STATIC Add-ons

ETAS offers several add-ons, which have to be licensed separately. Popular add-ons are:

- **ASCMO-SDK** (Software Developer Kit): 3.5 "P-Code Version" on page 18
- **ASCMO-GO** (Global Optimization): "Global Optimization " on page 73
- **ASCMO-ME** (Model Export): 6.10 "Model Export " on page 170

For more information on licensing,see Licensing.

2.2 Fields of Application

ETAS ASCMO provides various application fields.

Calibration of ECUs

- Engine parameter optimization: emission, oxygen sensor heating...
- Optimization of dynamic functions: driveability, charge pressure
- Parameterization of ECU models (cylinder fill, torque, ...)

The use of ETAS ASCMO in the area of calibration offers a series of advantages:

- Significant increase in efficiency through measuring and analysis efforts
- Improved complexity handling
- Improved data quality
- Multiple use of models

Research, Function and System Development:

- Quick calibration and evaluation of experimental engines
- Use of models of real engines for test and development of new functions (e.g. controller strategies)
- Analysis and optimization of unknown systems:
  - Hybrid vehicles (battery size, engine displacement, ...)
  - Starter-generator systems: modeling of generator current, bearing temperatures, ...
  - Development of injection systems (optimization of the geometry)
- "Meta-Modeling" to accelerate physical simulations.

The advantages in the area of research and development lie primarily in a quicker and improved system understanding, coupled with a variety of possibilities for impact analysis.

2.3 Basics

ETAS ASCMO-STATIC enables you to create data-based models that accurately represent the stationary behavior of complex systems. With a wealth of features and options for visualizing, analyzing, and optimizing system behavior, ASCMO-STATIC also supports the creation of experimental designs based on the DoE (Design of Experiments) methodology.

Using AI methods from the field of machine learning, ASCMO-STATIC allows you to accurately model complex relationships without requiring detailed knowledge of the underlying algorithms. Whether you're a less experienced user who appreciates parameter-free, automated model creation, or an expert who benefits from extensive configuration options, this software is designed to meet your needs.

A typical application of ASCMO-STATIC is the modeling of fuel consumption and pollutant emissions of complex internal combustion engines as a function of engine speed, load and all engine control variables. Based on these models, you can make accurate predictions and implement manual and automatic optimizations to achieve the best compromise between pollutant emissions, fuel consumption, and other operational constraints during engine operation.

2.4 Design of Experiments (DoE)

Design of experiments is a method for data-based modeling of unknown systems.

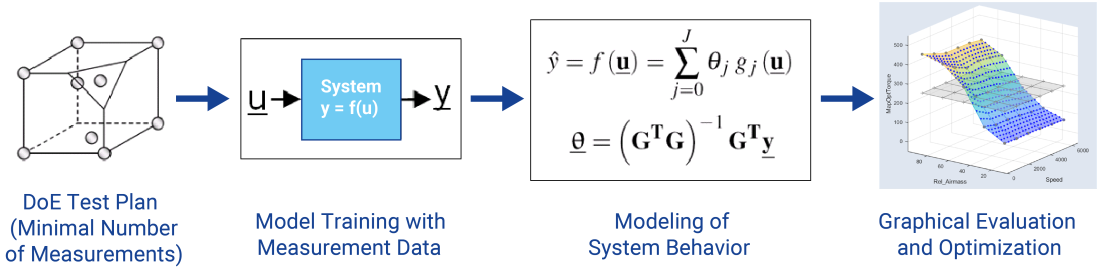


Fig. 2-1: From experiment plan to model-based optimization

The process begins with an experiment plan to obtain data for model training using minimal measuring effort. This dataset is then used to train the model.

The models are based on mathematical approximation methods and are capable of reproducing the behavior of the measured system.

The goal of the modeling is to evaluate and optimize the system's behavior, such as determining the input variables that lead to optimal output variables (maximum performance, minimum consumption/emission) for an internal combustion engine.

2.5 Model-Based Calibration

The calibration of ECUs becomes increasingly more complex and expensive due to several factors.

The main causes are:

- increasing variety of variants of a base version
- decreasing availability of test objects (engine, vehicle)
- increasingly stricter requirements concerning the consumption, emissions and diagnostics.

This growing complexity can no longer be effectively managed using "classical" calibration methods. Even when automating routine processes, a series of tasks must be iteratively executed:

- measurement and variation of ECU parameters
- measurement of responses of the experimental vehicle/engine
- analysis of the measurement
- step-by-step optimization

The entire procedure leads to one optimal data set.

For the model-based calibration with ASCMO-STATIC or ASCMO-DYNAMIC, only one measurement on the real system is necessary (after creating the experiment plan). Everything else is performed on the model:

- After specifying the optimization targets: One optimization run leads to an optimal parameter set.
- Maps can be changed and the resulting behavior can be predicted.

- Depending on the specification, an optimal result is achieved:
  - small vehicle: consumption; sports car: torque
  - sporty driving behavior (torque is quickly available) vs. comfort

In this case, n iterations result in n optimal data sets.

2.6 Finding Out More

In addition to this User Guide, the Online Help is recommended, especially when working with the user interface. It can be accessed via **Help > Online Help** or context-sensitive with F1 in the currently open operating window.

For help on the P-code version functions, use **Help > Interface Help**.

3 Installation

Before installing, make sure your computer meets the system requirements (see System requirements MOCA ASCMO). You must ensure that you have the necessary user rights and a network connection.

If you want to use the product offline, you need to borrow the license in the ETAS License Manager (**LiMa** main window > **License** > **Borrow selected licenses / Borrow all licenses**). See Licensing for more information.

3.1 System Requirements

The following minimum system requirements must be met:

Required Hardware	1,0 GHz PC 4 GB RAM Graphics with a resolution of at least 1024 x 768, 32 MB RAM
Required Operating System	Windows® 10, Windows® 11
Required Free Disk Space	4 GB (not including the size for application data)

The following system requirements are recommended:

Recommended Hardware	4,0 GHz Quad-Core PC or equivalent 32 GB RAM Graphics with a resolution of 1680 x 1050, 128 MB RAM
Recommended Operating System	Windows® 10, Windows® 11
Recommended Free Disk Space	> 4 GB

3.2 Software Requirements

ETAS ASCMO requires and installs the MATLAB® Compiler Runtime 2022b. It also requires the .Net Framework V4.6, which is included with Windows® 10/11.

There are no additional software requirements for the installation of the ETAS ASCMO base product and add-ons. Any missing software components will be installed during the installation.

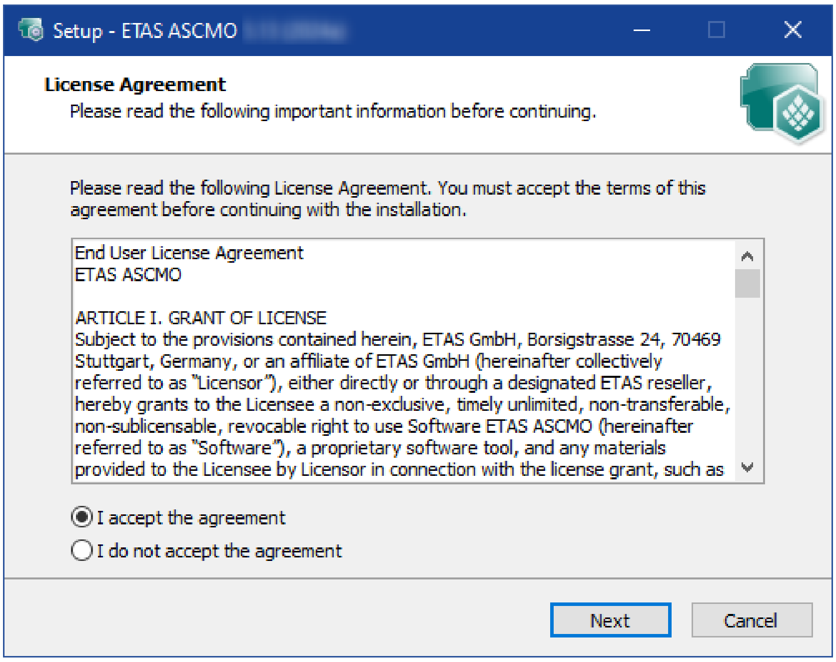
To use the ETAS ASCMO add-on *Software Developer Kit (SDK)*, a MATLAB® version R2021b up to R2023b and the MATLAB® *Optimization Toolbox* and *Statistics Toolbox* are required.

3.3 Installing

Install ETAS ASCMO

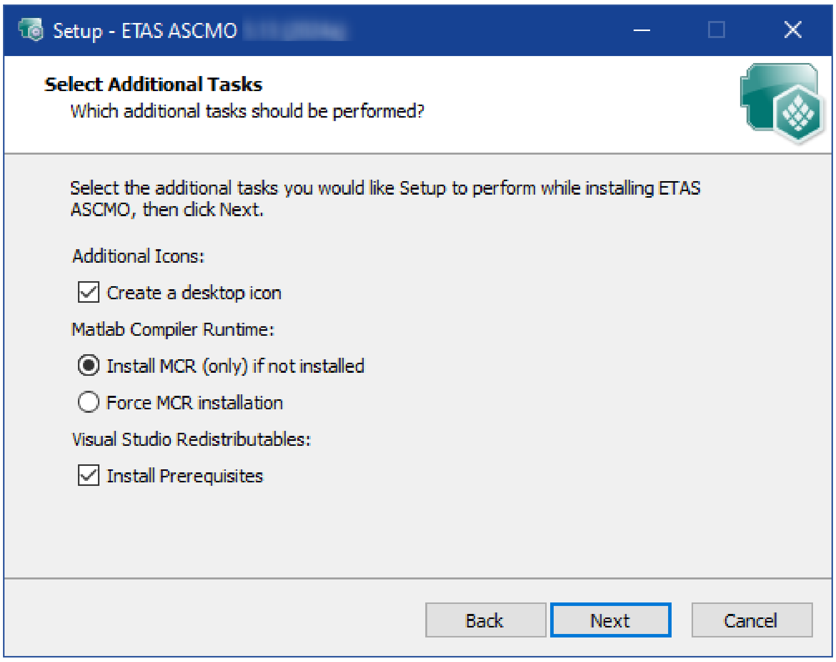
1. Go to the directory where the ASCMOinstallation file is located.
2. Double-click `Setup_ETAS-ASCMO_Vx_x_20xxx.exe`.

   The **License Agreement** window opens.



3. Read the license agreement carefully, then activate the **I accept the agreement** option.
4. Click **Next**.

   If you have already installed an ASCMO version, the path (destination location and start menu folder) of the initial installation will be used and steps 5 to 8 will not be available.
5. In the **Set Destination Location** window, accept the default folder or click **Browse** to select a new directory.
6. Click **Next**.
7. In the **Select Start Menu Folder** window, accept the default folder or click **Browse** to select a new folder.
8. Click **Next**.



i. Activate the **Create a desktop icon** checkbox if you want to create an icon on the desktop.

ii. Choose whether to force the installation of the MATLAB® Compiler Runtime or to install it only if it is not already installed.

iii. If necessary, activate**Install Prerequisites**.

9. Click **Next**.
10. In the **Ready to Install** window, click **Install** to start the installation.

or

If you want to change the settings, click **Back**.

The installation process begins. A progress indicator shows the installation's progress. When the installation is complete, the **Completing the ETAS ASCMO Setup Wizard** window opens.

11. Click **Finish**.

⇨ The installation is complete. ASCMO can be started.

3.4 Files and Directories

All files belonging to the program are located in the *<installation>* directory selected during the installation, and in additional subfolders of this directory.

By default, *<installation>* is `C:\Program Files\ETAS\ASCMO x.x`.

Start Menu

After successful installation, the folder you specified in the **Select Start Menu Folder** window with the following entries is added to the **Windows Start** menu.

- **ASCMO Desk V5.16**

  Starts the ASCMO-DESK window, where you can start the ETAS ASCMO components.
- **ASCMO Dynamic V5.16**

  Starts ASCMO-DYNAMIC.
- **ASCMO ExpeDes Dynamic V5.16**

  Starts ASCMO-DYNAMIC ExpeDes.
- **ASCMO ExpeDes V5.16**

  Starts ASCMO-STATIC ExpeDes.
- **ASCMO MOCA Runtime V5.16**

  Starts the ASCMO-MOCA Runtime environment with limited functionality.
- **ASCMO MOCA V5.16**

  Starts ASCMO-MOCA.
- **ASCMO Static V5.16**

  Starts ASCMO-STATIC.
- **Manuals and Tutorials**

  Opens the ASCMO documentation directory (`<installation>\Manuals`), which contains the following information and documents.
  - `ASCMOInterfaceDoc` – a folder with interface documentation.
  - `Examples` – a folder with different example data (e.g., ASCMO projects, MF4, DCM, XLS or FMU files, templates, plugins, etc.).
  - `HTML folder` – online help files for the installed components (available via <F1>).
  - `ASCMO-DYNAMIC_V5.16_User_Guide_*.pdf` – User Guide with tutorials for the basic functions of ASCMO-DYNAMIC.
  - `ASCMO-STATIC_V5.16_User-Guide_*.pdf` – User Guide with tutorials for the basic functions of ASCMO-STATIC.
  - `ASCMO-MOCA_V5.16_User-Guide_*.pdf` – User Guide with a tutorial for the basic functions of ASCMO-MOCA.

P-code Files

Of special interest are the P-code files for MATLAB® and Simulink® in the `<installation>\pCode\ascmo` directory.

For more information, see "P-Code Version" below.

3.5 P-Code Version

The P-code version also allows you to start ETAS ASCMO within MATLAB®.

Prerequisites

The P-code version requires an installation of MATLAB® R2021b up to R2023b. In addition, the following MATLAB® toolboxes are required:

- Optimization Toolbox™
- Statistics and Machine Learning Toolbox ™

Executing ETAS ASCMO

In MATLAB®, change to the directory `<installation>\pCode\ascmo`. In the command window, enter one of the following commands:

command	action
`AscmoDesk`	Starts ASCMO-DESK.
`ascmo static`	Starts ASCMO-STATIC.
`ascmo expedes`	Starts ASCMO-STATIC ExpeDes.
`ascmo dynamic`	Starts ASCMO-DYNAMIC.
`ascmo expedesdynamic`	Starts ASCMO-DYNAMIC ExpeDes.
`ascmo moca`	Starts ASCMO-MOCA.
`ascmo mocaruntime`	Starts ASCMO-MOCA Runtime.
`ascmo cyclegenerator`	Starts the standalone ASCMO-Cycle Generator.
`ascmo essentials`	Starts ASCMO Essentials.

All further steps in an ETAS ASCMO tool can be automated using commands whose description can be found in the main menu under **Help › Interface Help**.

3.6 Uninstalling

Note

You cannot uninstall specific components. The procedure uninstalls **all** ETAS ASCMO components.

Uninstall ETAS ASCMO

1. Go to the directory where the ASCMOinstallation file is located.

   Start the uninstall procedure.

   A warning message opens.
2. Double-click `unins000.exe`.

   A warning message opens.

3. To completely remove ETAS ASCMO and all its components, click **Yes**.

   The uninstallation process begins. When the process is complete, a message window opens.
4. Click **OK** to complete the uninstallation.

⇨ ETAS ASCMO and all its components are successfully uninstalled.

4 Basics of ASCMO-STATIC

This chapter contains a description of the fundamental concepts of ASCMO-STATIC.

In particular, they include:

- 2.2 "Fields of Application" on page 11

  This section contains information about the application fields of ETAS ASCMO.
- 4.1 "Design of Experiments (DoE)" below

  This section contains information about the topic "design of experiments".
- 4.2 "Model Training in ASCMO-STATIC" on page 28

  This section contains information about the model training in ASCMO-STATIC.
- 4.2.5 " Model Types of ASCMO-STATIC" on page 32

  ASCMO-STATIC may use various methods when generating models. You can define parameters, which are briefly described in this chapter.
- 4.3 "Model Assessment and Improvement " on page 57

  This section contains information about how to evaluate the quality of the models created by ASCMO-STATIC and, if necessary, to improve them.
- 5.1 "User Interface of ASCMO-STATIC" on page 80

  This section provides an overview of the user interface of ASCMO-STATIC.
- 4.4 "Advanced Settings in ASCMO-STATIC " on page 63

  This section gives you an overview on the different advanced parameter and describes how to enable and disable the visibility of the advanced settings.
- 4.5 "Optimization " on page 72

  This section contains a description of the different optimization methods and the optimization criteria that can be used for them.
- 4.6 "Model Screening (Model Evaluation) " on page 75

  This section contains a description of the model screening available via **Extras › Model Screening**.

4.1 Design of Experiments (DoE)

The goal of the *design of experiments* (DoE) is the model-like description of unknown systems based on measuring data. The methodology of the DoE includes the creation of the experiment plan according to statistical aspects, the creation of models and the optimization of modeled systems.

The experiment planning is performed with the purpose of minimizing the measuring effort – for the design of experiments, several variables are modified for each measurement. For the model training, mathematical approximation methods are used.

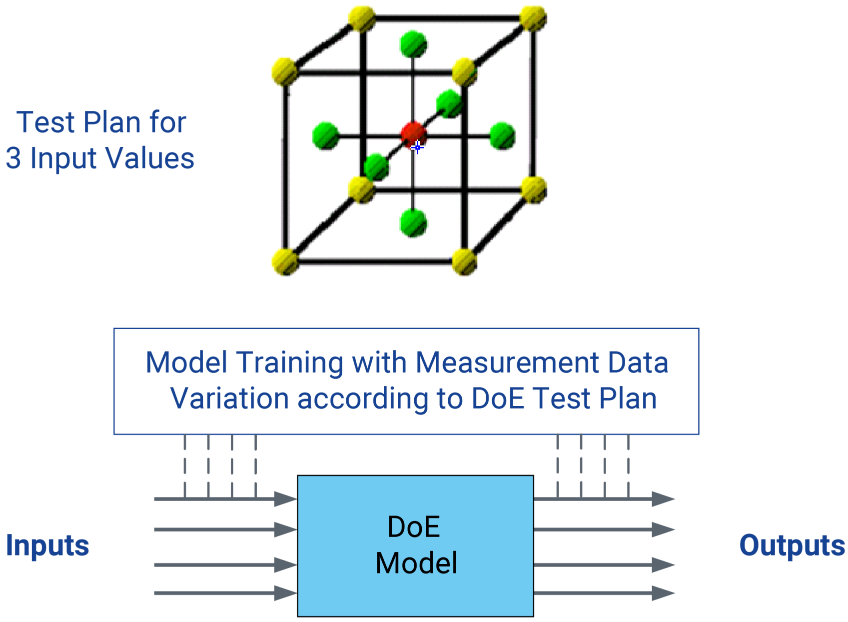


Fig. 4-1: Model training with data from the DoE plan

For example, one application is the description of the behavior of an internal combustion engine by a model. The maps of the ECU are not being optimized using the real system, but a model.

4.1.1 Process for the Experiment Planning

The description of the behavior of a system in the *"classic" DoE of the first generation* is polynomial-based. Advantages are a comprehensible and established process since many tools are available. However, the process only allows mapping simple, local relationships (up to the second order).

Serious disadvantages include the high parameterization effort and the low robustness with respect to outliers.

A more state-of-the-art process is based on *neural networks*. They also allow mapping complex relationships – on top of that, the process is relatively illustrative and a few tools are available.

A disadvantage is once again the parameterization effort and the risk of overfitting. In addition, a high number of measuring data is required (for training as well as validation purposes).

ETAS ASCMO belongs to the *DoE of the 2nd generation*, which is based on Gaussian processes. These processes allow mapping any number of relationships (such as the global behavior of an internal combustion engine).

The outstanding advantage of this process consists of the optimal relationship of measuring effort (low) to model accuracy (high).

Other advantages are:

- No parameterization required
- Robustness with respect to outliers
- User-friendly

Disadvantages are the low illustrativeness of the theory and a relatively large system memory of the PC.

4.1.2 Classic Experiment Plans

Classic experiment plans are based on a grid-shaped or star-shaped measurement of the experimental space.

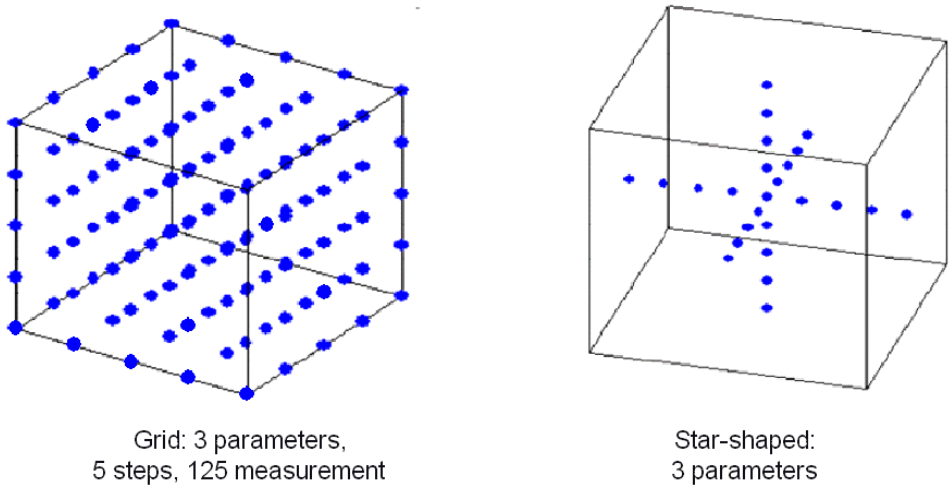


Fig. 4-2: Grid measurement and star-shaped measurement of the experimental space

Grid Measurement

In principle, the grid measurement is suitable for all types of modeling (while an analysis is also possible without a model).

But the exponential increase in required measurements is dramatic: With more than 3-4 parameters to be varied, this procedure is no longer practical (see Tab. 4-1). And despite the high number of measurements, an optimal coverage of the parameter space is not achieved.

Tab. 4-1 shows the comparison of required measurements for classical grid measurement versus DoE method. Despite an increasing number of parameters, the required measurements in DoE method remain low, especially in direct comparison to grid measurement. The shown values are related to medium expected system complexity (selection option in 7.2 "Step 1: General Settings" on page 187).

Number of Parameters	Number of required Measurements	
	Grid Measurement (5 Steps per Parameter)	DoE Method (Space filling)
1	5	5
2	25	10
3	125	20
4	625	35
5	3125	50
6	15625	70
7	78125	90
8	390625	115
9	1953125	145
10	9765625	175

Tab. 4-1: Classic grid measurement and DoE method (Space filling): measuring effort depending on number of parameters

Star-Shaped Measurement

For the star-shaped measurement (Tab. 4-1), only one parameter is being varied at a time which, in particular, translates into a reduced measuring effort (compared to the grid measurement).

However, interactions between the parameters are neglected in the process and actual optima are frequently not found, which renders this process unsuitable for the modeling of complex systems.

4.1.3 Experiment Plans According to DoE

An alternative to the aforementioned processes are the experiment plans according to DoE of the latest generation, the so-called D- (or V-) optimal plans and space-filling experiment plans according to DoE.

D- (or V-) Optimal Plans

These experiment plans are specifically adapted to polynomial models whose precise specification is dependent upon the polynomial order.

However, the creation of such experiment plans requires prior knowledge of the system behavior and the number of parameter levels is low. They show a heavy weighting of the boundary areas and the individual points have a relatively high degree of importance.

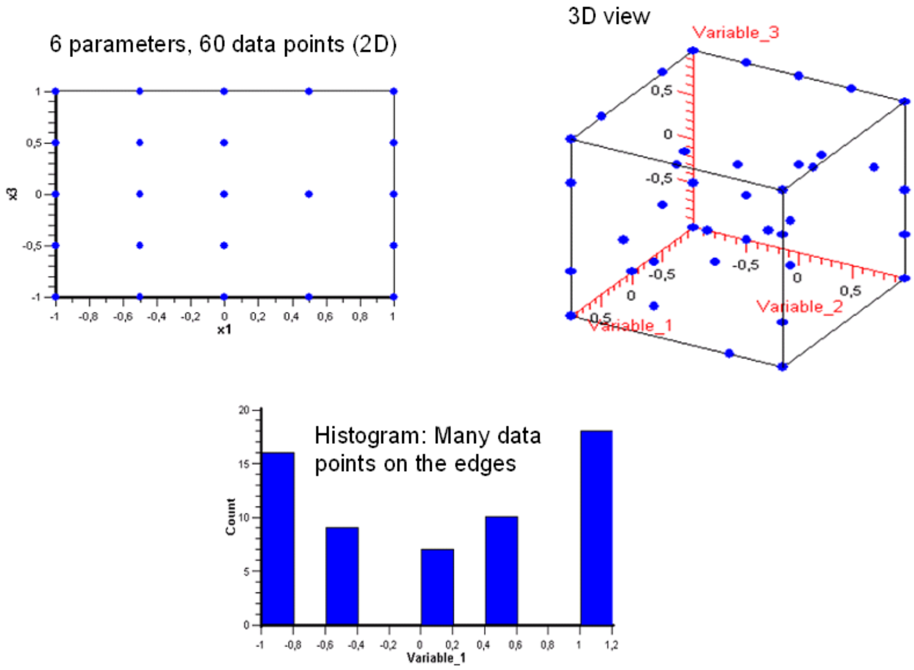


Fig. 4-3: Example for a D-optimal plan

This type of experiment plan is less suitable for the use with ETAS ASCMO – in this case, space-filling plans are optimal.

Space-Filling Plans

Space-filling plans (Sobol, Latin Hypercube, etc.) are characterized by an even distribution of the measuring points in the parameter space and an optimal coverage of all parameter levels.

No previous knowledge about the system to be measured is required, and the data gained are perfectly suited for model training in ASCMO-STATIC.

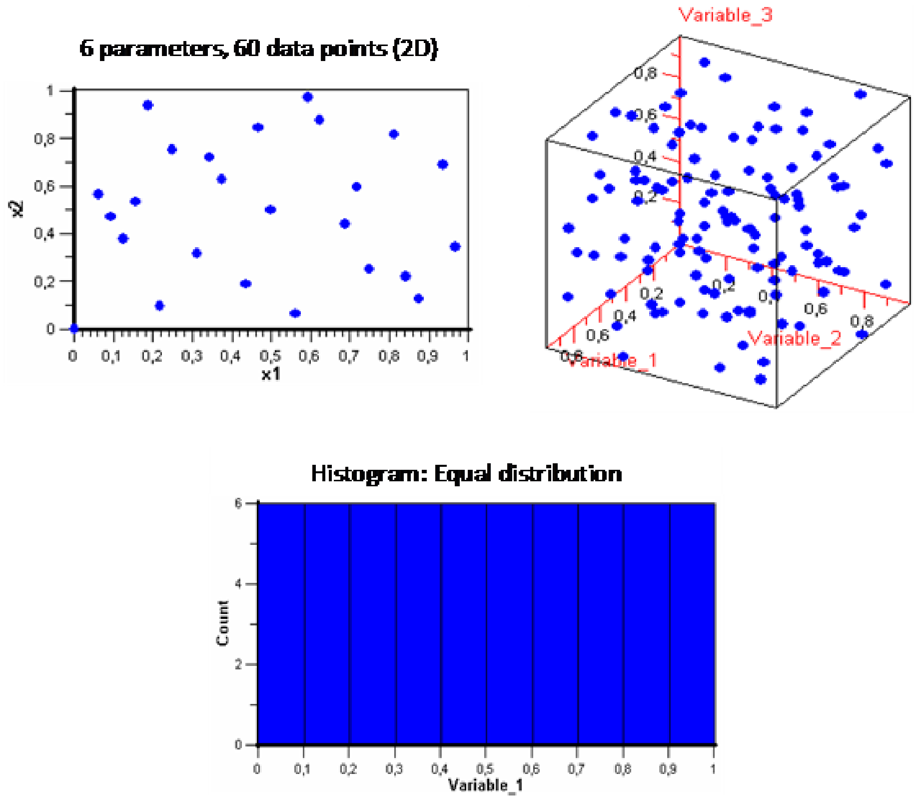


Fig. 4-4: Example for a space-filling experiment plan

The creation of such measuring plans is done, e.g. with ASCMO-STATIC ExpeDes.

4.1.4 Space-Filling Plans with ASCMO-STATIC ExpeDes

ASCMO-STATIC ExpeDes is a tool for creating space-filling experiment plans according to DoE. ASCMO-STATIC ExpeDes is perfectly suited for planning measurements with a space-filling distribution over a grid of operating points (speed/load) as they are required for the model training in ASCMO-STATIC.

Operating range limits can be adapted in a flexible way, measuring points can be compressed in selective ranges.

The creation of an experiment plan with ASCMO-STATIC ExpeDes is described in chapter 7 "Tutorial: Working with ASCMO-STATIC ExpeDes" on page 183.

Fig. 4-5 shows such space-filling measurements for four parameters $x_1,...,x_4$ with the respective operating point-dependent experiment space limits (red and green area).

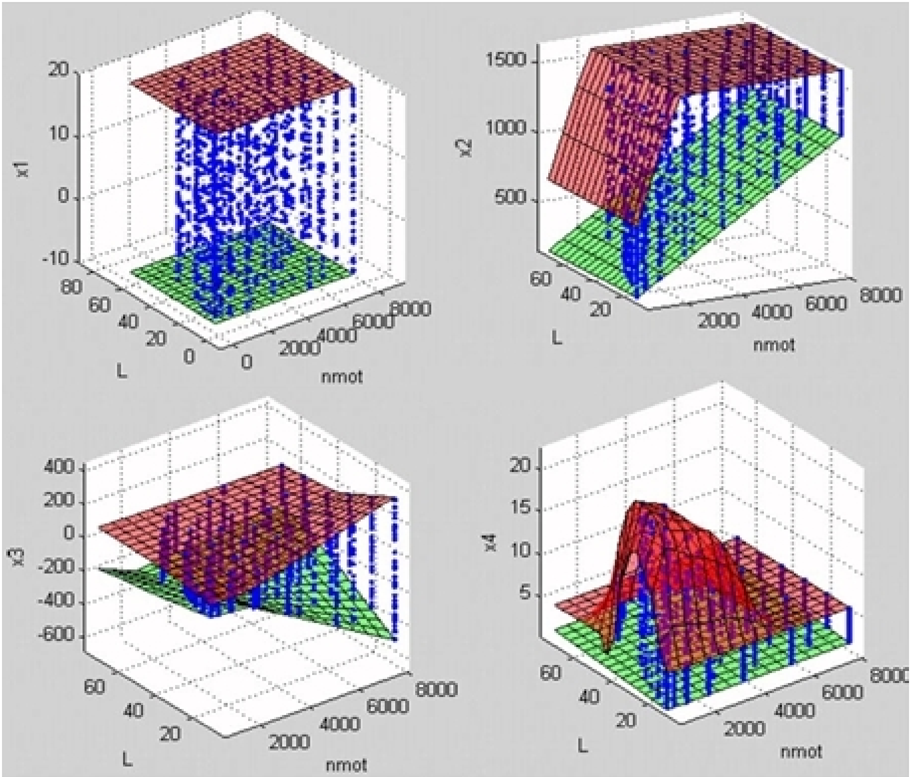


Fig. 4-5: Measuring points and experiment space limits

4.1.5 Important Boundary Conditions for the Experiment Procedure

For the experiment procedure, it must be observed that the results are reproducible. In particular, measured variables with a drift must be ruled out or at least recognized and responded to accordingly.

Experiment Repeatability and Drifts

To determine the experiment repeatability and estimate the time drift, it is necessary to measure several repetition points – preferably at the beginning, in the middle and at the end of the experiment.

In addition, to separate a possible time drift from parameter influences, measurements should not be performed sorted by parameters (ascending or descending), but must be performed "at random" with respect to the order of individual parameters.

Requirement for the Test Bench Automation

The test bench automation should ensure that non-drivable conditions do not lead to an emergency switch-off and subsequently to a cancellation of the measurement.

This requires monitoring critical variables (such as temperatures, peak pressures, soot, misfiring, etc.). If limit violations occur, "intelligent" responses such as returns to safe central points and observation of defined stabilizing times are also required.

4.1.6 Advantages of the DoE Methodology

- The method leads to a significant reduction of the measuring effort.
- Averaging/smoothing results in eliminating the signal noise.
- The decoupling of measurements and analysis/optimization leads to an improved test carrier utilization (test bench, role, vehicle, etc.).
- If optimization criteria are changed, no new measurements are required.
- The models are based on real measurements and not on possibly incorrect assumptions about the system behavior.
- The models permit a description, interpretation, and documentation of the system behavior, such as interactions between parameters.

4.1.7 Limitations of the DoE Methodology

- Differentiable and continuously differentiable signal shapes can be modeled very well. A function is referred to as continuously differentiable if its derivative is differentiable.
- Non-differentiable signal shapes can be described only insufficiently
- If the signal-to-noise ratio is too low, the model has no significance.

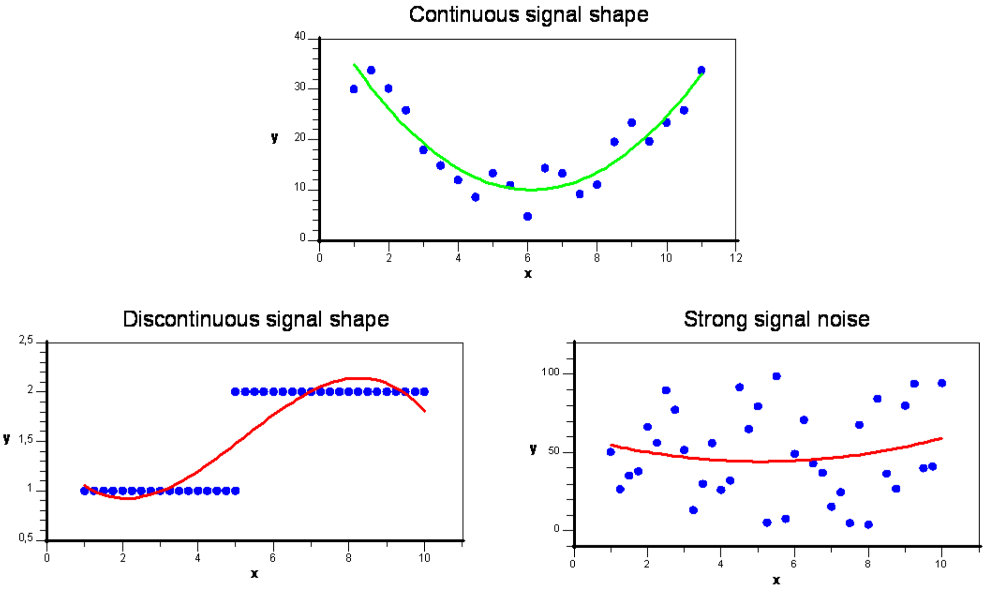


Fig. 4-6: Suitable and unsuitable signal shapes

4.2 Model Training in ASCMO-STATIC

This section contains information about the model training in ASCMO-STATIC.

- 4.2.1 "Requirements for Measuring Data" below
- 4.2.2 "Disturbance Variables, Drift and Experiment Repeatability" on the next page
- 4.2.3 "Global and Local Models" on page 31
- 4.2.4 "Processes for Modeling" on page 31
- 4.2.5 " Model Types of ASCMO-STATIC" on page 32

4.2.1 Requirements for Measuring Data

In principle, one simple rule applies to the successful model training in ASCMO-STATIC: The model can only be as good as it is possible based on the training data. In other words: If the model is trained with incorrect or imperfectly varied data, the result is less useful.

Optimal results are achieved if the training data are obtained based on a DoE plan and if special attention was paid to a meaningful variation of the measurements in the parameter range (see also 4.1 "Design of Experiments (DoE)" on page 21).

For possible error sources during the measurement, see also 4.2.2 "Disturbance Variables, Drift and Experiment Repeatability" on the next page.

Note

The data used for training do not necessarily have to come from a physical experiment, they can also be the result of a computer simulation.

Plausibility Check of the Data

After the measurement and before the model training, it is recommended to subject the training data to a plausibility check.

The following points should be given special attention:

- Have all parameters been varied according to the experiment plan and did the measured system remain in the operating mode intended for this purpose?
- Do the output variables fall in physically meaningful ranges?
- If necessary, delete or add an offset for the data that cannot be <= 0, such as emissions, consumption, etc.
- Have repetition points been measured and what was the result?

Measurement File

Importing the measuring data in ASCMO-STATIC requires a file with the following properties:

- File format: Microsoft Excel
- Input and output variables in columns
- Names (and poss. units) must be located in the first line (or first and second line)

	A	B	C	D	E	F	G	H	I	J	K	L	
1	speed [rpm]	load [bar]	injection [deg CA]	ignition [deg CA]	fuel_pressure [bar]	EGR [%]	ex_cam [deg CA]	in_cam [deg CA]	SCV [-]	Fuel_mass [kg/h]	CoV [%]	Soot [FSN]	NOx_
2	800	0,8	19,4709	0,2545	18,1098	13,238	20,8012	-22,6259	1	0,552916244	5,3774513	0,08650385	
3	800	0,8	22,7538	-2,2528	10,6643	41,578	37,286	-17,0274	1	0,572418779	10,424491	0,30311664	
4	800	0,8	23,5745	1,2991	14,1211	21,564	8,4375	-40,8212	0	0,563930598	4,1069745	0,066854	
5	800	0,8	16,7011	-4,1593	15,5504	21,088	23,0335	-10,5541	1	0,558864104	12,387057	0,11768217	
6	800	0,8	34,1411	1,0119	11,4952	22,045	26,8113	-45,195	1	0,59859547	3,8791776	0,22398064	
7	800	1,7778	35,2775	-1,9005	19,015	9,7125	5,9502	-44,959	0	1,081065738	4,7891371	0,14899586	
8	800	1,7778	24,2238	-4,2493	13,7017	31,525	4,4346	-46,4936	1	0,908954181	7,8313797	0,21828725	
9	800	1,7778	16,7816	-0,7402	17,1201	43,868	37,443	-47,1756	0	0,812965207	4,4316852	0,05167109	
10	800	1,7778	26,1937	-2,9475	14,8164	18,797	39,1271	-32,5121	0	0,881746329	3,9007246	0,23805574	
11	800	1,7778	32,7603	-3,0607	11,9925	43,401	34,4116	-12,7334	0	0,833629265	5,0705967	0,31436919	
12	800	2,7556	13,4006	0,481	19,575	5,982	4,794	-18,3948	0	1,162479088	2,9544179	0,33931497	
13	800	2,7556	37,4154	-3,3695	16,5405	32,021	-0,4523	-7,2413	1	1,384573183	3,6616844	0,95000821	
14	800	2,7556	13,7251	-1,9774	12,2107	26,297	24,2054	-15,8599	1	1,179949204	3,5084389	0,47111527	
15	800	2,7556	33,6292	-3,6361	16,0594	14,793	31,2005	-30,7313	1	1,370680085	3,5836299	0,26899806	
16	800	3,7334	22,8779	-2,2658	16,6584	16,663	26,4276	-43,2946	0	1,612318456	1,3709393	0,14064169	
17	800	3,7334	35,7118	-0,1866	17,0771	4,5399	27,1693	-33,1719	0	2,104364043	2,0054577	0,33186433	
18	800	3,7334	15,633	-1,2262	10,8672	10,278	3,8035	-38,2332	1	1,515028485	3,3482255	0,79382561	
19	800	3,7334	21,9464	-5,4152	12,0882	30,652	25,5003	-34,6903	1	1,587987496	3,2568439	0,30689288	
20	800	3,7334	13,4595	-4,6203	16,9026	20,561	28,8383	-13,4327	1	1,615996618	4,1368152	0,54974639	
21	800	4,7112	16,7607	-1,0074	16,1878	9,4955	8,1898	-20,9447	1	2,204968769	1,190633	1,99608165	
22	800	4,7112	14,9598	-1,8497	17,0387	16,41	11,8535	-24,0508	0	2,285596686	2,4427312	1,71719167	
23	800	4,7112	21,4431	-4,0156	13,3917	14,257	4,5261	-15,3538	1	2,582587511	4,148339	2,29378406	
24	800	4,7112	17,2109	-1,3985	12,9966	14,692	34,019	-28,244	0	2,102591204	2,7290551	0,80713841	
25	1155,5556	5,6063	20,166	-2,3799	18,2145	14,73	5,9772	-38,4912	0	3,562603984	1,3533685	2,18951403	
26	1155,5556	5,6063	28,6861	-2,2914	18,5861	10,996	1,8934	-45,2336	1	4,014029913	3,1715251	1,05900758	
27	1155,5556	5,6063	33,2302	-0,6393	18,7843	8,2267	26,0411	-30,3143	0	4,712343569	1,6206187	2,08932276	
28	1155,5556	4,4762	25,9135	0,0234	17,4984	15,392	6,5119	-43,49	1	2,654935141	3,3937125	0,77749085	
29	1155,5556	4,4762	28,2245	1,2295	13,1383	22,967	19,7176	-20,8562	0	3,242141139	3,9718158	2,27082076	
30	1155,5556	4,4762	12,8179	0,2646	14,1642	19,34	13,8484	-47,4842	1	2,376326545	3,5720996	1,6328346	
31	1155,5556	4,4762	23,6988	-0,8511	11,8238	12,675	43,3779	-24,6839	0	2,712104168	3,3533623	1,37782248	
32	1155,5556	4,4762	18,4991	-1,6351	14,068	14,577	0,4592	-9,7057	1	2,854632856	5,4872199	1,73847319	

Tabelle1

Fig. 4-7: Example for measuring data

Blank lines can be processed by ASCMO-STATIC and do not have to be removed. In addition, individual values of a column can be deleted (it is not necessary to remove the entire line).

4.2.2 Disturbance Variables, Drift and Experiment Repeatability

The quality of the model heavily depends on the quality of the measuring data. During the measurement, e.g. of an engine, disturbance variables such as engine, oil and charge air temperature in addition to a time drift of the measuring arrangement itself can negatively affect the measurement.

For this reason, repetition points should be measured for every measurement series and, in particular, at defined operating points and in such a way that they cover the entire time range of a measurement (start, middle, and end of measurement).

On the one hand, these measurements can be used to determine the experiment reproducibility (RMSE) – no model can be more precise without improving the measurement technique.

On the other hand, it allows identifying a disturbance variable-based drift that can be identified by means of the measuring time or measuring point number. In that case, drift effects could be corrected by inserting the disturbance variable in the model. But a requirement for the disturbance variable correction is that the disturbance variable does not feature any correlation to other model parameters (no "sorted" parameters).

4.2.3 Global and Local Models

The model training and optimization of parameters can be performed for global as well as local models in ASCMO-STATIC.

For a *global model*, there is a grid of operating points on which all input and output variables of interest of the system have first been measured.

A typical example is the measurement of an internal combustion engine with the goal of generating optimized maps with respect to emission, consumption, or driving comfort.

Local models are sufficient if the system does not have any operating points (e.g. oxygen sensor) or they are not of interest in the case at hand.

Examples can be found in the area of system development/advanced engineering where a new injection system is evaluated using only a few operating points (idling, medium and full load) or if the calibration is already finished, but there are problems, e.g. during idling (idle optimization).

4.2.4 Processes for Modeling

Classical processes (DoE of 1st generation, neural networks) limit themselves to classes of functions for modeling the system behavior, such as polynomials or neural networks. However, this leads to a limited flexibility and carries with it the risk of overfitting (see Fig. 4-8).

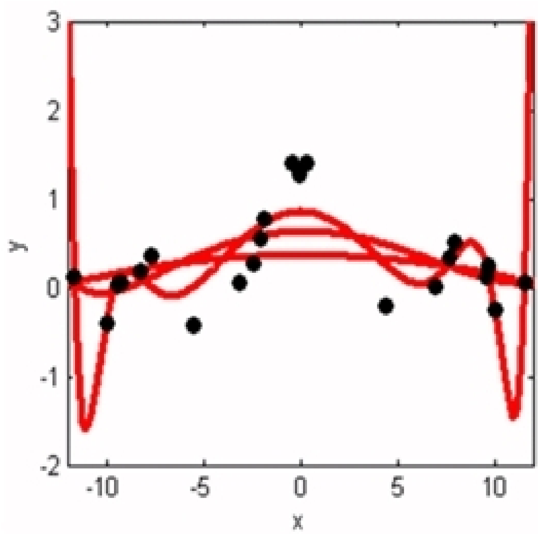


Fig. 4-8: Fit to measuring data using polynomials

Such an overfitting becomes particularly noticeable in a significant deterioration of variables for model evaluation (see 4.3.3 "Variables RMSE and R2" on page 62) if the quality of the modeling is tested with data that have not been involved in the modeling (Leave-One-Out error, see 4.3.2 "Methods and Data for Determining the Model Quality" on page 61).

ASCMO-STATIC, on the other hand, utilizes Gaussian processes for the model training in which all theoretically possible functions are taken into account.

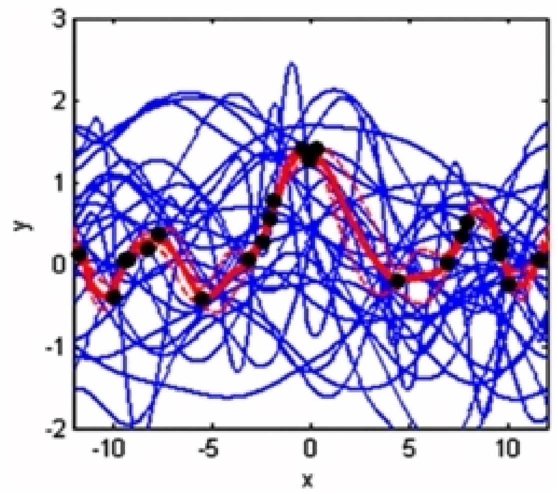


Fig. 4-9: Fit to measuring data in ASCMO-STATIC

In the model training, the functions are evaluated by their errors in comparison with the data and special attention is paid so that they feature a low complexity.

4.2.5 Model Types of ASCMO-STATIC

ASCMO-STATIC may use various methods when generating models. You can define parameters that are described in this section.

The following model types are available for ASCMO-STATIC:

- No Model
- ASC GP Model
- MLP Model

If the advanced settings are enabled(**File › Options › Advanced Settings**), the following additional model types are available:

- Polynom Model
- ASC Compressed Model
- Classification (GP) Model
- Classification (MLP) Model
- ASC GP-SCS Model
- Classification (Random Decision Trees) Model

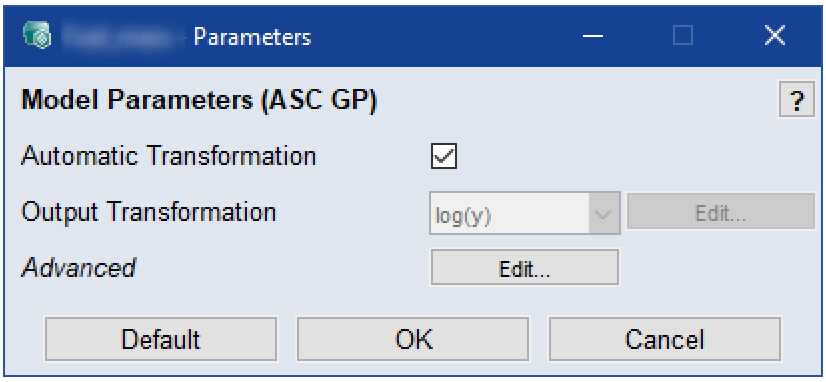
No Model

Only the measurements are displayed as data points. Here you have no possibility to define model parameters.

ASC GP Model

The ASCMO Gaussian Process (ASC GP) model type can be used for data sets with up to 4000 training data points with 15 inputs. This is the default model type. Training time and memory consumption scale especially with the number of data points. For many more data points, other model types may be more suitable.

The following model parameters can be specified:



Automatic Transformation

Activate this checkbox if you want the optimal Box-Cox transformation to be determined automatically during model training.

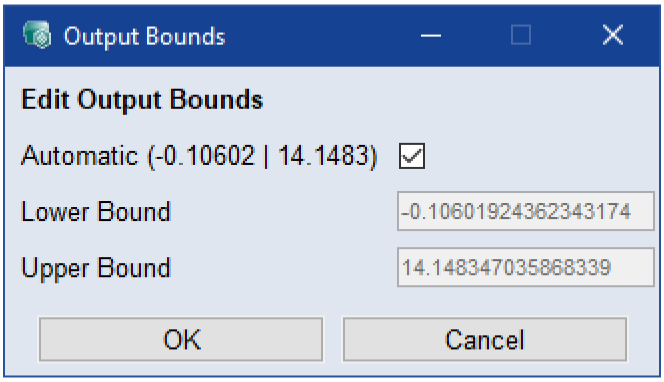
Output Transformation

Select the transformation type of the output. Using a transformation can improve the model prediction. If the **Automatic Transformation** checkbox is activated, the transformation type is selected automatically. Not all transformations are available if the training data has negative or zero values.

You can select from the following choices:

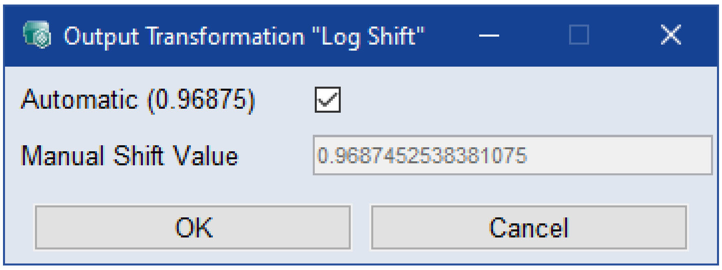
- **none**: no transformation
- **1/y**: inversion
- **1/sqrt(y)**: inverted square root
- **log(y)**: logarithm
- **sqrt(y)**: square root
- **Bounded**: limited to lower and upper bound

  Click **Edit** to view the automatically selected bounds or to define the lower and upper bounds manually. To define them manually, deactivate the **Automatic** checkbox. The bounds must be in the range of the training data.



– **log(y+c)**: logarithm plus constant

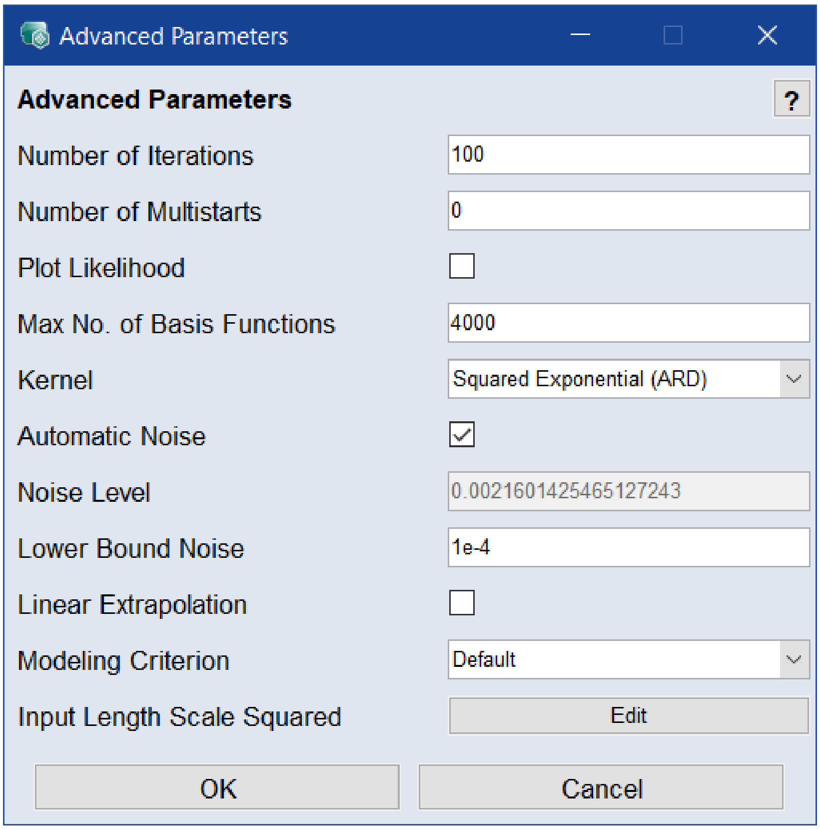
  Click **Edit** to view the automatically selected log shift or to define a manual shift value. To define it manually, deactivate the **Automatic** checkbox.



Advanced

You can find more information in the Online Help (<F1>).

Advanced parameters for the ASC GP model



Number of Iterations

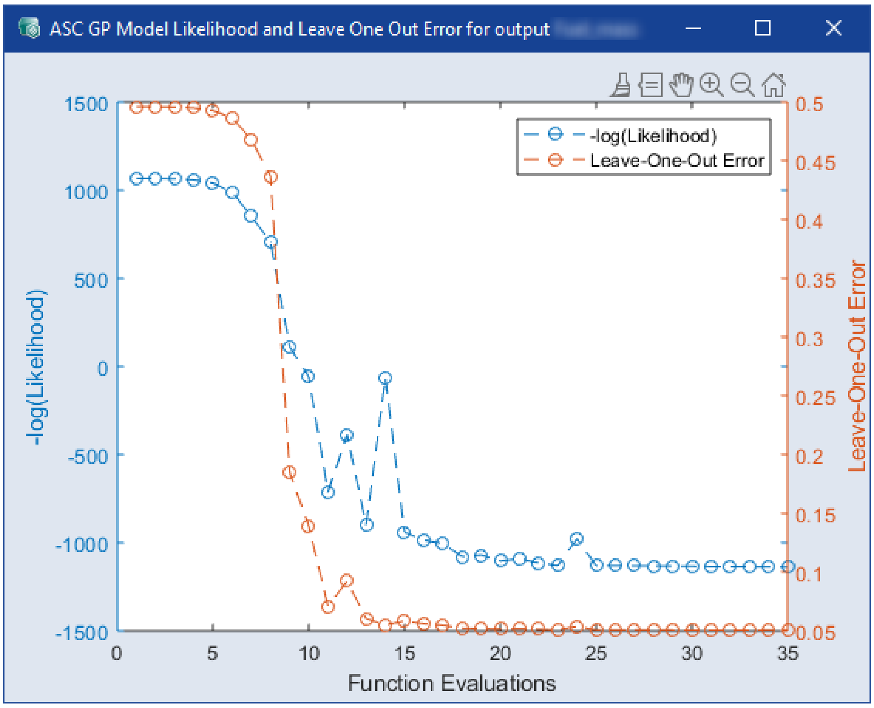
Enter the number of iterations to be performed during model training. If the model performance does not improve within 10 iterations on the validation data, the training will be aborted. In deep learning this is often referred to as number of epochs.

Number of Multistarts

Enter the number of training repetitions with different starting values. A higher value can improve the model quality, but the model training then takes more time. The default value is 3.

Plot Likelihood

Activate if you want the values of the logarithmic Likelihood function and the Leave-One-Out error to be displayed as a function of the runs during model training.



Max No. of Basis Functions

Enter a value as maximum number of basis functions for the model training. A higher value can improve the model quality, but the model training then takes more time. The default value is 4000. The smaller value of either training data points or subset size is used as the number of basic functions in the model. If more training data points are available, all data is used for training, but the resulting model is constrained to 4000 basis functions. A slightly different training algorithm is used in this case, while model predictions are done in the same way.

Sparse Subset Selection Method

Select the method you want to use to reduce the training data points to a spare subset when there are more than "Max No. of Basis Functions" data points.

- **Random** (Default): Random selection of a subset without specific selection criteria.
- **GP-SCS like** Selection using Gaussian process regression to iteratively add relevant features to a subset based on their impact on predictive performance.

Kernel

Select the kernel function to be used during model training.

- **Squared Exponential (ARD)** (Default): Trains models with softer curve characteristics.
- **Matern (ARD)** Trains models with harder curve characteristics. Provides sharper resolution of strong nonlinear effects with small noise. This could lead to overfitting.

Automatic Noise

Activate if you want the optimization of the noise level parameter to be performed automatically.

Noise Level

Enter a value for the maximum noise level to be tolerated. The input field is only accessible if the "Automatic Noise" checkbox is deactivated.

Lower Bound Noise

Enter a value as minimum for the noise level parameter. The input field is only accessible if the "Automatic Noise" checkbox is activated.

Linear Extrapolation

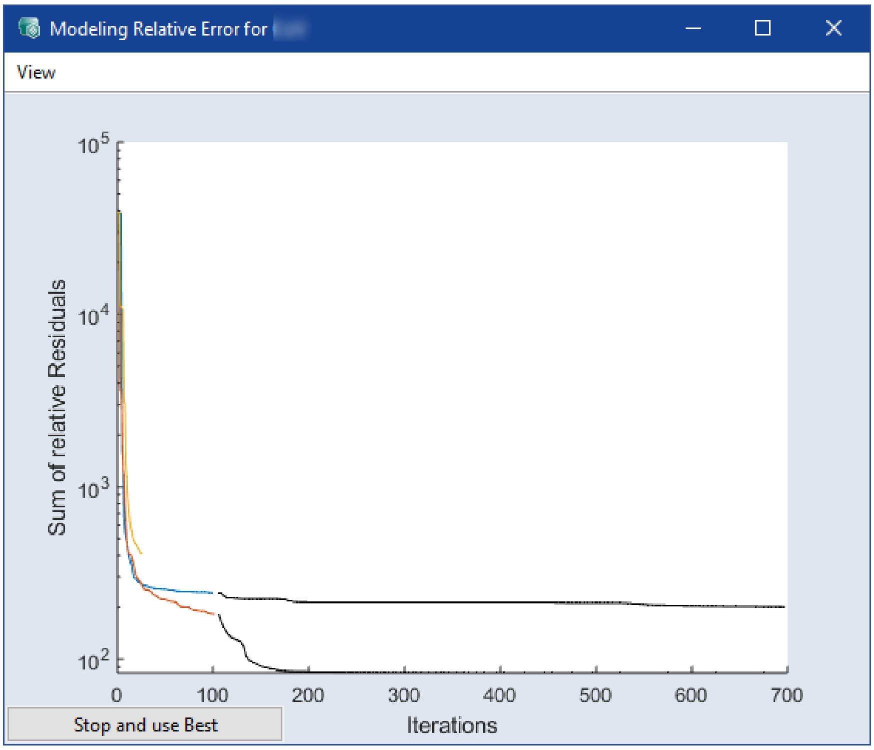
Activate if you want a supplementary linear model to be used, which learns the basic tendency of the data and shows this tendency outside the measured area.

Modeling Criterion

Select the modeling criterion to be used during model training.

- **Default**: Based on likelihood.
- **Relative Error**: Is the quotient of the error and the actual value (`measured data - predicted data / measured data * 100`). During the optimization, a visualization window pops up where you can manually

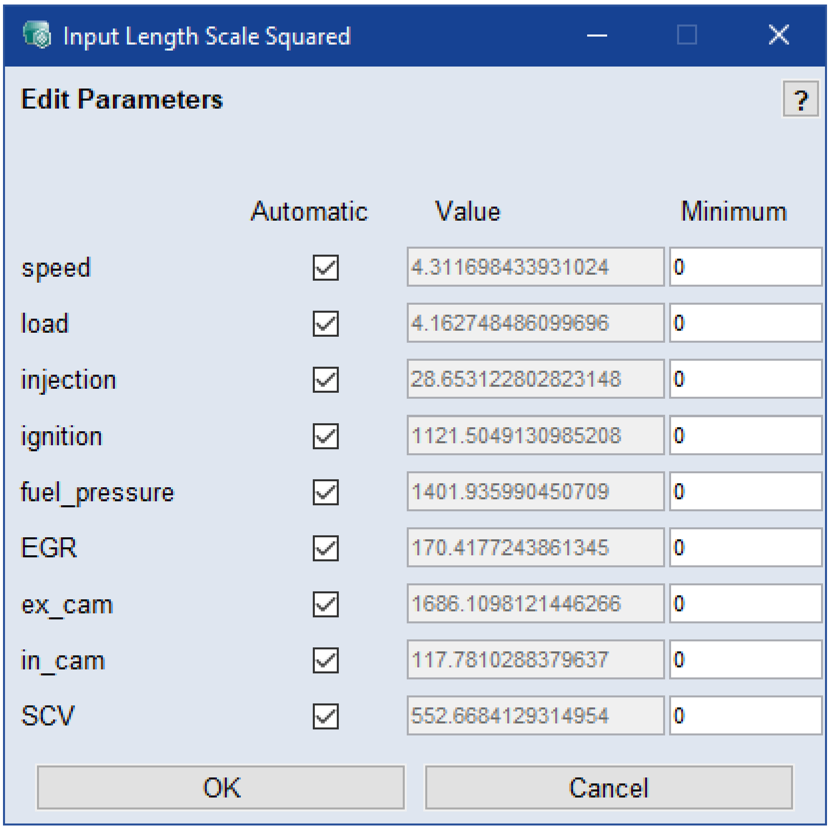
stop the optimization and use the best result so far.



Input Length Scale Squared

Click **Edit** to open the **Input Length Scale Squared** window. You can edit the hyperparameter for each input. If **Automatic** is activated for an input, the respective hyperparameter is set automatically. Otherwise, you can edit the values manually. The hyperparameter per input dimension is the core width of the Gaussian bell. The length scale is r in the following equation, so a smaller value has greater relevance:

$$y(\vec{x}) = \sum_{i=1}^{N} C_i \cdot e^{-\frac{1}{2}\sum_{l=1}^{D}\frac{(X_{il}-x_l)^2}{r_l^2}}$$

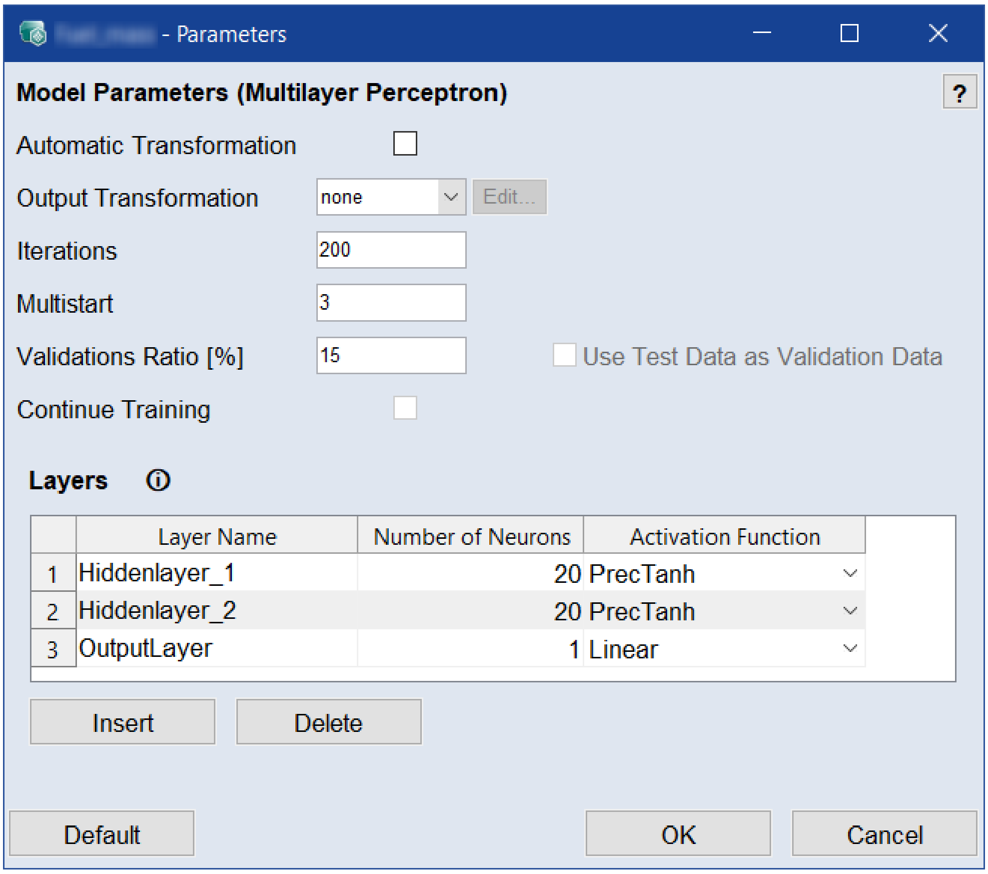


You can display the relevances graphically in the "Relevance of Inputs" Window (see ASCMO Help).

MLP Model

MLP (Multilayer Perceptron) with possibility to export it to flatbuffer format for Bosch ECU.

The following model parameters can be specified:



Automatic Transformation

Activate this checkbox if you want the optimal Box-Cox transformation to be determined automatically during model training.

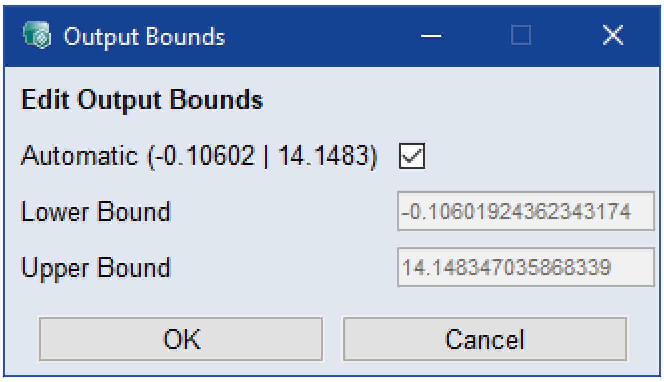
Output Transformation

Select the transformation type of the output. Using a transformation can improve the model prediction. If the **Automatic Transformation** checkbox is activated, the transformation type is selected automatically. Not all transformations are available if the training data has negative or zero values.

You can select from the following choices:

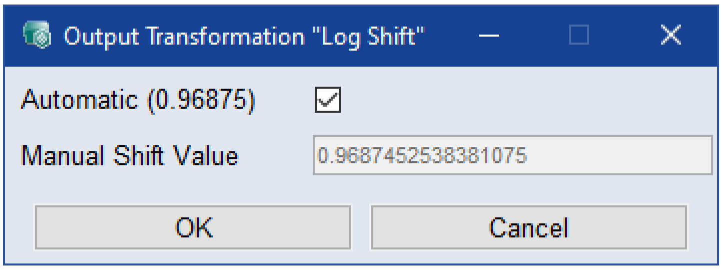
- **none**: no transformation
- **1/y**: inversion
- **1/sqrt(y)**: inverted square root
- **log(y)**: logarithm
- **sqrt(y)**: square root
- **Bounded**: limited to lower and upper bound

  Click **Edit** to view the automatically selected bounds or to define the lower and upper bounds manually. To define them manually, deactivate the **Automatic** checkbox. The bounds must be in the range of the training data.



– **log(y+c)**: logarithm plus constant

  Click **Edit** to view the automatically selected log shift or to define a manual shift value. To define it manually, deactivate the **Automatic** checkbox.



Iterations

Enter the number of iterations to be performed during model training. If the model performance does not improve within 10 iterations on the validation data, the training will be aborted. In deep learning this is often referred to as number of epochs.

Multistart

Enter the number of times to run the optimizer with different starting values during model training. A higher number means a higher probability of finding the optimal model, but it takes more time.

Validations Ratio [%]

Enter the relative number, in percent, of validation samples to be randomly selected from the training data.

Use Test Data as Validation Data

Activate the checkbox if you want to use the test data as validation data.

Continue Training

Activate the checkbox to continue with existing model training and iterations, if possible, instead of starting a new training. You can change the training properties and continue. For example, train with a complex activation function, then switch to a more efficient one (for the ECU) and continue training seamlessly.

Layers

Configure the layers of the multilayer perceptron. There is always at least one hidden layer, usually 1-3, and exactly one output layer.

Select an activation function from the list:

- **Linear**: y = x
- **ReLU**: y = max(0, x)
- **LeakyReLU**: y = max(0.01 * x, x)
- **Sigmoid**: y = 1 / (1 + exp(-x))
- **PrecTanh**: y = 2 / (1 + exp(-2 * x)) - 1
- **Elliotsig**: y = x / (1 + abs(x))

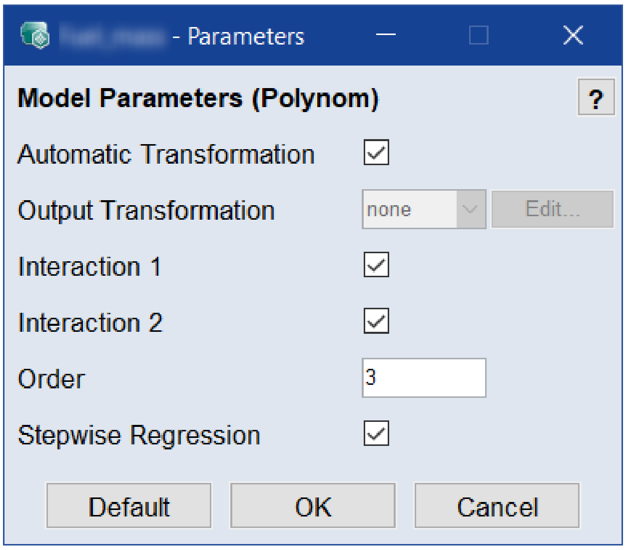
Insert: Click to insert a hidden layer.

Delete: Select one or more layers and delete them.

Polynom Model

> **Note**
>
> The model is only available if you have enabled the advanced settings via **File › Options › Advanced Settings**.

For the **Polynom** model the following parameters can be specified:



Automatic Transformation

Activate this checkbox if you want the optimal Box-Cox transformation to be determined automatically during model training.

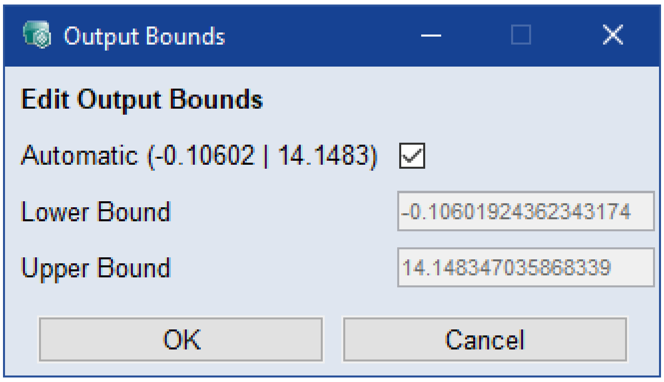
Output Transformation

Select the transformation type of the output. Using a transformation can improve the model prediction. If the **Automatic Transformation** checkbox is activated, the transformation type is selected automatically. Not all transformations are available if the training data has negative or zero values.

You can select from the following choices:

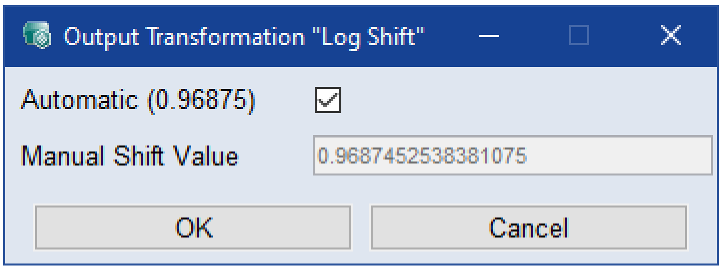
- **none**: no transformation
- **1/y**: inversion
- **1/sqrt(y)**: inverted square root
- **log(y)**: logarithm
- **sqrt(y)**: square root
- **Bounded**: limited to lower and upper bound

  Click **Edit** to view the automatically selected bounds or to define the lower and upper bounds manually. To define them manually, deactivate the **Automatic** checkbox. The bounds must be in the range of the training data.

- **log(y+c)**: logarithm plus constant

  Click **Edit** to view the automatically selected log shift or to define a manual shift value. To define it manually, deactivate the **Automatic** checkbox.

Interaction 1/2

- Interaction of first order:

  $x_i * x_j$ with `i!=j`

- Interaction of second order:

  $x_i * x_j * x_k$ with `i=j` allowed

Order

Maximal order of the polynomial.

Pure terms x_i^n.

Stepwise Regression

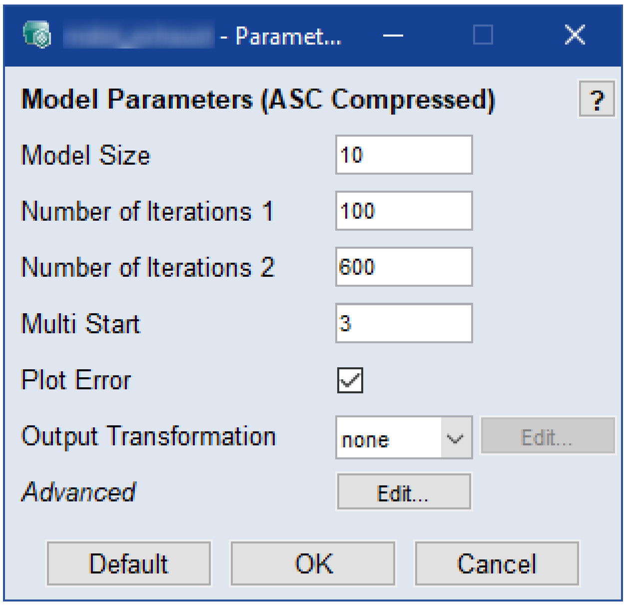
When this option is enabled, the regression model is built up gradually. Terms with a p-value (significance level) less than 5% are included sequentially. Including new terms may change the significance level of an already included term. If the p-value is greater than 10%, terms are removed. This method is also called **forward selection**.

ASC Compressed Model

Note

The model is only available if you have enabled the advanced settings via **File › Options › Advanced Settings**. You will also need the additional license **ASCMO_MODEL_COMPRESSION**.

This model type allows you to limit the number of basis functions within the model. The following model parameters can be specified:



Model Size

Enter the number of basis functions of the compressed model.

Number of Iterations 1/2

Enter the number of iterations to be performed during model training. If the model performance does not improve within 10 iterations on the validation data, the training will be aborted. In deep learning this is often referred to as number of epochs.

Multistart

Enter the number of times to run the optimizer with different starting values during model training. A higher number means a higher probability of finding the optimal model, but it takes more time.

Plot Error

Activate the checkbox to display model error information during model training.

To save the RMSE as a bitmap graphic use **View › Save as Bitmap** in the main window.

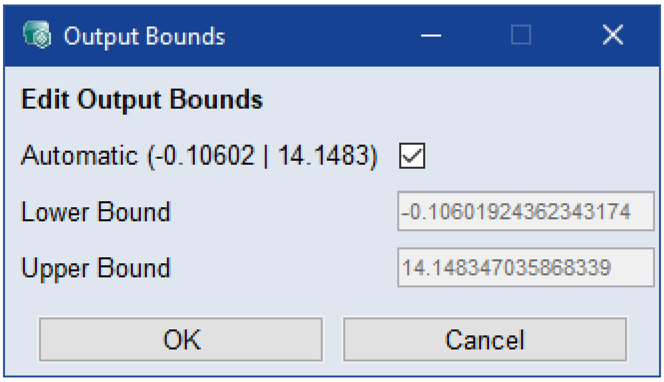
Output Transformation

Select the transformation type of the output. Using a transformation can improve the model prediction. Not all transformations are available if the training data has negative or zero values.

You can select from the following choices:

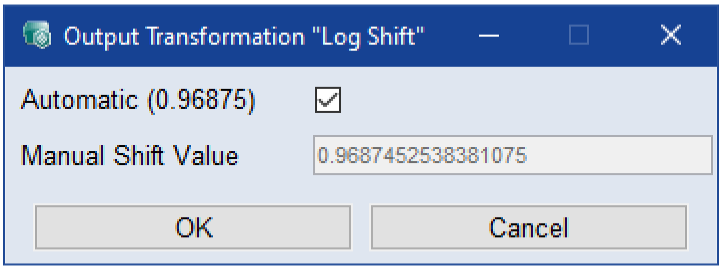
- **none**: no transformation
- **1/y**: inversion
- **1/sqrt(y)**: inverted square root
- **log(y)**: logarithm
- **sqrt(y)**: square root
- **Bounded**: limited to lower and upper bound

  Click **Edit** to view the automatically selected bounds or to define the lower and upper bounds manually. To define them manually, deactivate the **Automatic** checkbox. The bounds must be in the range of the training data.

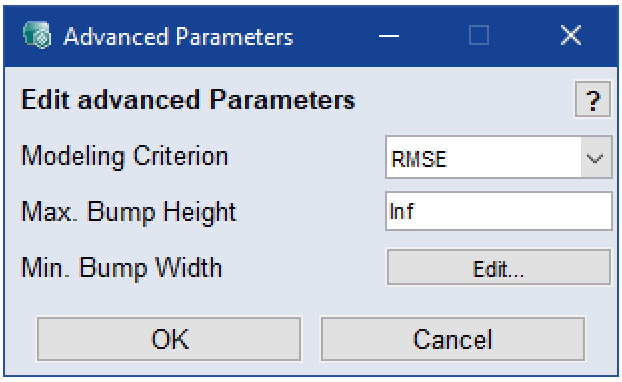
- **log(y+c)**: logarithm plus constant

  Click **Edit** to view the automatically selected log shift or to define a manual shift value. To define it manually, deactivate the **Automatic** checkbox.



Advanced

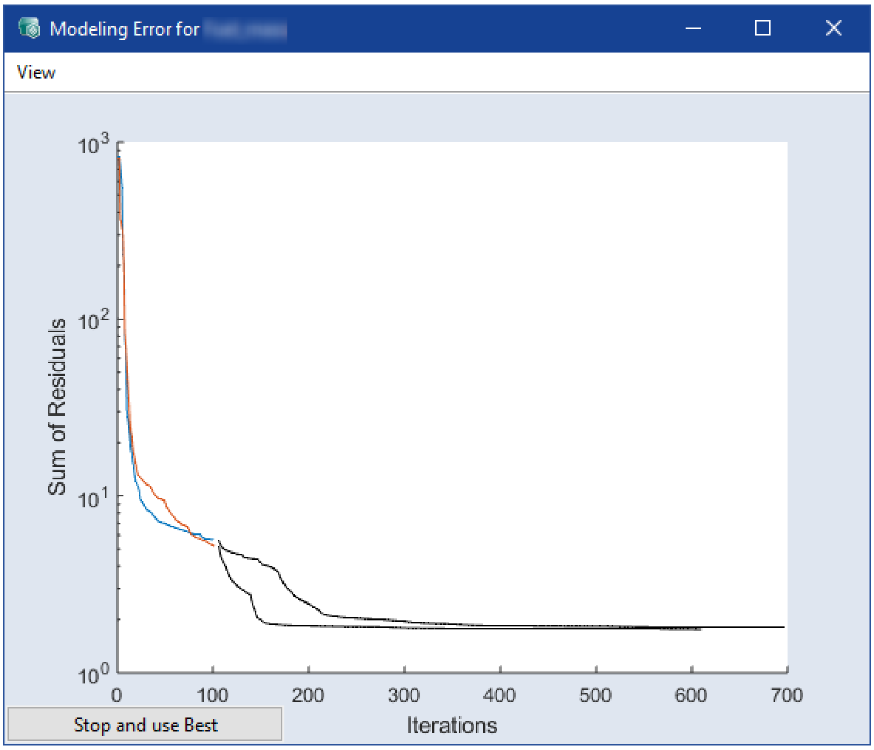
Click **Edit** to edit the advanced parameters.



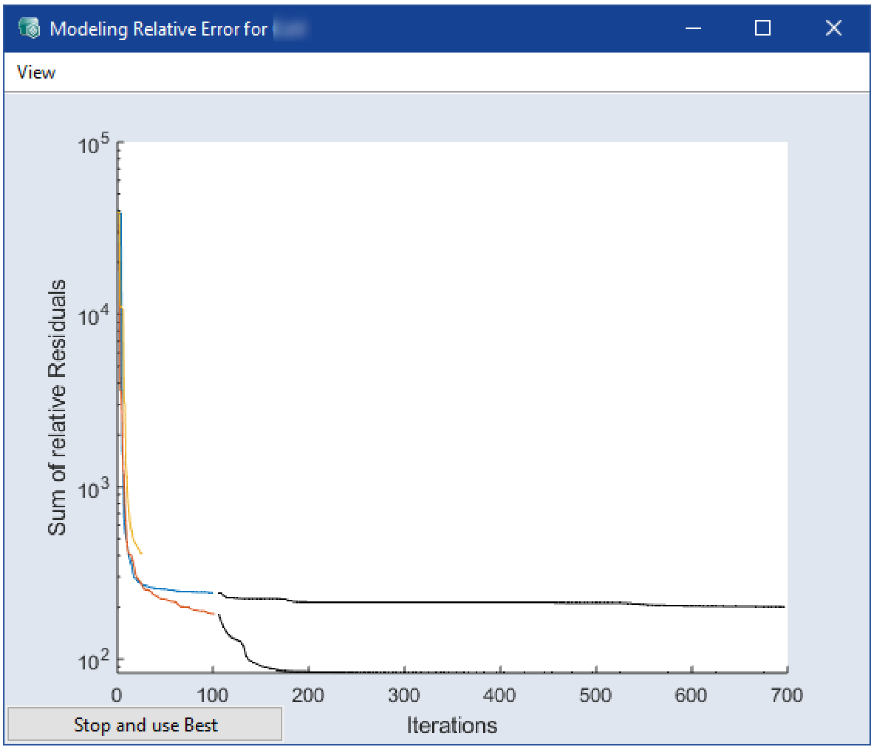
Modeling Criterion

Select the modeling criterion to be used during model training.

- **RMSE**: The Root Mean Square Error is the average size of the error between the predicted and actual values. A second measurement is less than 1 RMSE from the model prediction with 68% probability (95.5% < 2 RMSE, 99.7% < 3 RMSE, etc.). During the optimization, a visualization window pops up where you can manually stop the optimization and use the best result so far.



— **Relative Error**: Is the quotient of the error and the actual value (`measured data - predicted data / measured data * 100`). During the optimization, a visualization window pops up where you can manually stop the optimization and use the best result so far.



Max. Bump Height

Enter a number to limit the height of the Gaussian bump.

Min. Bump Width

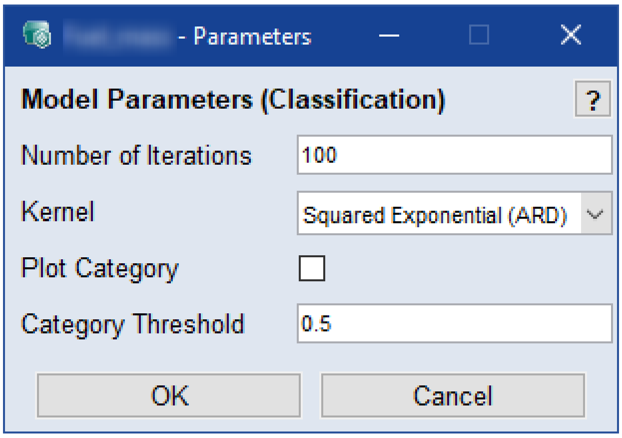
Click **Edit** and enter a number to limit the width of the Gaussian bump per input.

Classification (GP) Model

> **Note**
>
> The model is only available if you have enabled the advanced settings via **File › Options › Advanced Settings**.

This type of model allows you to classify existing measurements (inputs) into two classes, 0 or 1. The curve shown in the classification model is equal to the probability that the current input belongs to class 1. For example, you can determine whether the engine knocks in a certain configuration (class 1) or not (class 2).

The following model parameters can be specified:



Number of Iterations

Enter the number of iterations to be performed during model training. If the model performance does not improve within 10 iterations on the validation data, the training will be aborted. In deep learning this is often referred to as number of epochs.

Kernel

Defines which kernel function is used for model training

- **Squared Exponential (ARD)**: Trains models with softer curve characteristics.
- **Matern (ARD)**: Trains models with harder curve characteristics. This can lead to overfitting.

Export/Plot Category

If deactivated, the probability of mapping the input to class 1 is plotted.

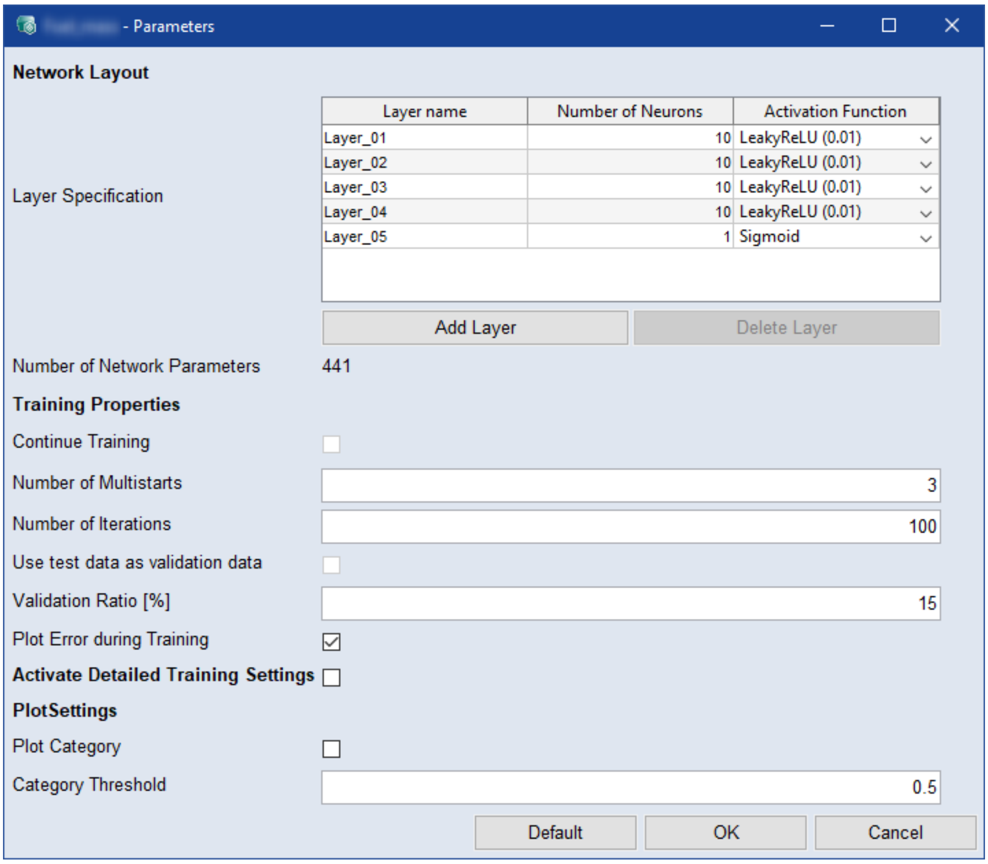
If activated, the probability values greater than or equal to the threshold are mapped to class 1. The class membership is plotted.

Category Threshold

In the model evaluation, measurement data will be assigned to class 1 if it meets or exceeds this threshold. The **Plot Category** checkbox must be activated. The threshold must be a number in [0,1], the default is 0.5.

Classification (MLP) Model

The following model parameters can be specified:



Layers

Configure the layers of the multilayer perceptron.

Select an activation function from the list:

- **Linear**: y = x
- **ReLU**: y = max(0, x)
- **LeakyReLU**: y = max(0.01 * x, x)

- **Sigmoid**: y = 1 / (1 + exp(-x))
- **PrecTanh**: y = 2 / (1 + exp(-2 * x)) - 1
- **Elliotsig**: y = x / (1 + abs(x))

Add Layer: Click to insert a layer.

Layer: Select one or more layers and delete them.

Number of Network Parameters

Dynamically shows the number of parameters used by the model training for current settings.

Continue Training

Activate the checkbox to continue with existing model training and iterations, if possible, instead of starting a new training. You can change the training properties and continue. For example, train with a complex activation function, then switch to a more efficient one (for the ECU) and continue training seamlessly.

Number of Multistarts

Enter the number of training repetitions with different starting values. A higher value can improve the model quality, but the model training then takes more time. The default value is 3.

Number of Iterations

Enter the number of iterations to be performed during model training. If the model performance does not improve within 10 iterations on the validation data, the training will be aborted. In deep learning this is often referred to as number of epochs.

Use Test Data as Validation Data

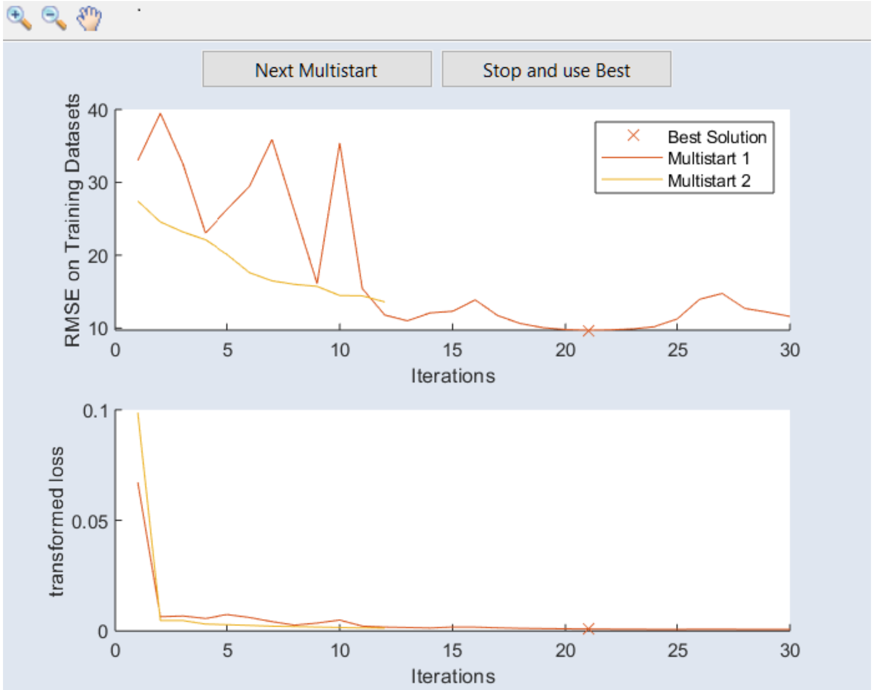
Activate the checkbox if you want to use the test data as validation data.

Validations Ratio [%]

Enter the relative number, in percent, of validation samples to be randomly selected from the training data.

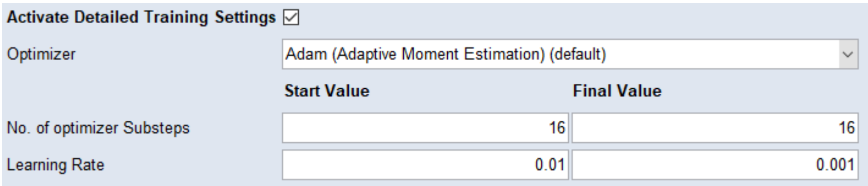
Plot RMSE during Training

Activate if you want the RMSE values for training data and validation data to be displayed during model training.



Activate Detailed Training Settings

Activate the checkbox to display the **Detailed Training Settings** section.



- **Optimizer**

  Select the optimizer used to train the model. If you activate the **Continue Training** checkbox, it is recommended to select **Stochastic Gradient Descent (for continue)**.

The detailed training settings are adjusted for each iteration. For the first iteration the **Start Value** is used, for the last iteration the **Final Value**. The values in between are interpolated.

- **No. of Optimizer Substeps**

  Determines how many sequences of length **Lookback Length** are used for one optimizer update. The default value is 100, which is the batch size used in deep learning. The larger the value, the smaller the batch size and vice versa. If the number is small, the optimizer step is performed less frequently and the training is therefore faster.

- **Learning Rate**

  Enter the size of the optimizer steps. The default value is 0.01. Valid value range is [0, 1].

The larger the learning rate, the faster the training generally will be. However, convergence can be hindered, or even prevented, by large learning rates.

Plot Category

If deactivated, the probability of mapping the input to class 1 is plotted.

If activated, the probability values greater than or equal to the threshold are mapped to class 1. The class membership is plotted.

Category Threshold

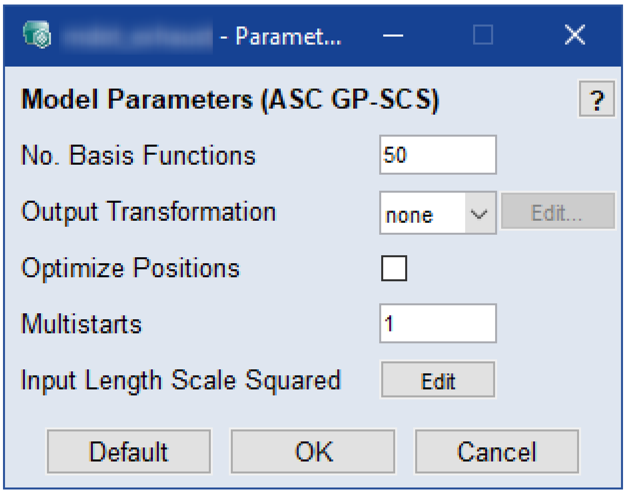
In the model evaluation, measurement data will be assigned to class 1 if it meets or exceeds this threshold. The **Plot Category** checkbox must be activated. The threshold must be a number in [0,1], the default is 0.5.

ASC GP-SCS Model

Note

The model is only available if you have enabled the advanced settings via **File › Options › Advanced Settings**. You will also need the additional license **ASCMO_MODEL_COMPRESSION**.

The ASCMO Gaussian Process Sparse Constant Sigma (ASC GP-SCS) model type should be preferred if the number of training data is large. You can define the following parameters:



No. Basis Functions

For a faster convergence, use a smaller number than in the actual model training (e.g., 20).

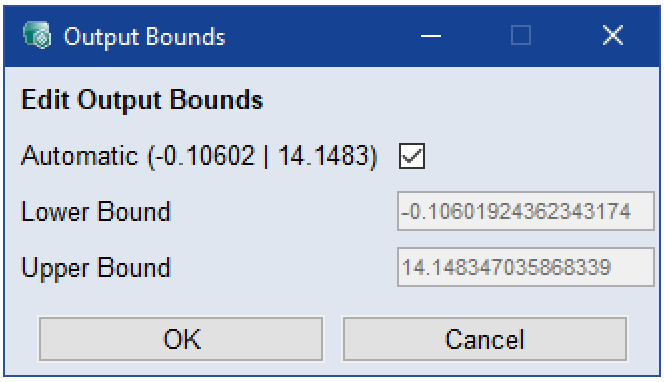
Output Transformation

Select the transformation type of the output. Using a transformation can improve the model prediction. Not all transformations are available if the training data has negative or zero values.

You can select from the following choices:

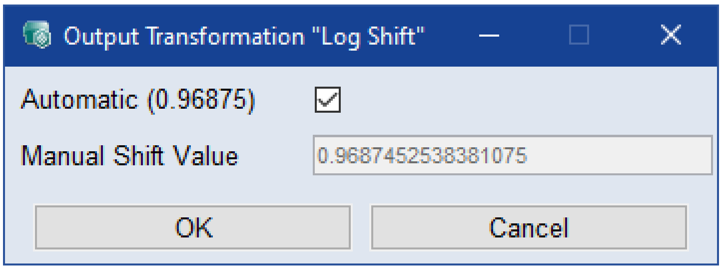
- **none**: no transformation
- **log(y)**: logarithm
- **Bounded**: limited to lower and upper bound

  Click **Edit** to view the automatically selected bounds or to define the lower and upper bounds manually. To define them manually, deactivate the **Automatic** checkbox. The bounds must be in the range of the training data.

- **log(y+c)**: logarithm plus constant

  Click **Edit** to view the automatically selected log shift or to define a manual shift value. To define it manually, deactivate the **Automatic** checkbox.

Optimize Positions

If checkbox is activated, virtual or pseudo inputs at optimized positions are used instead of basis functions at positions of randomly chosen training data.

The number of inputs used for training is given in the "No. Basis Functions" field.

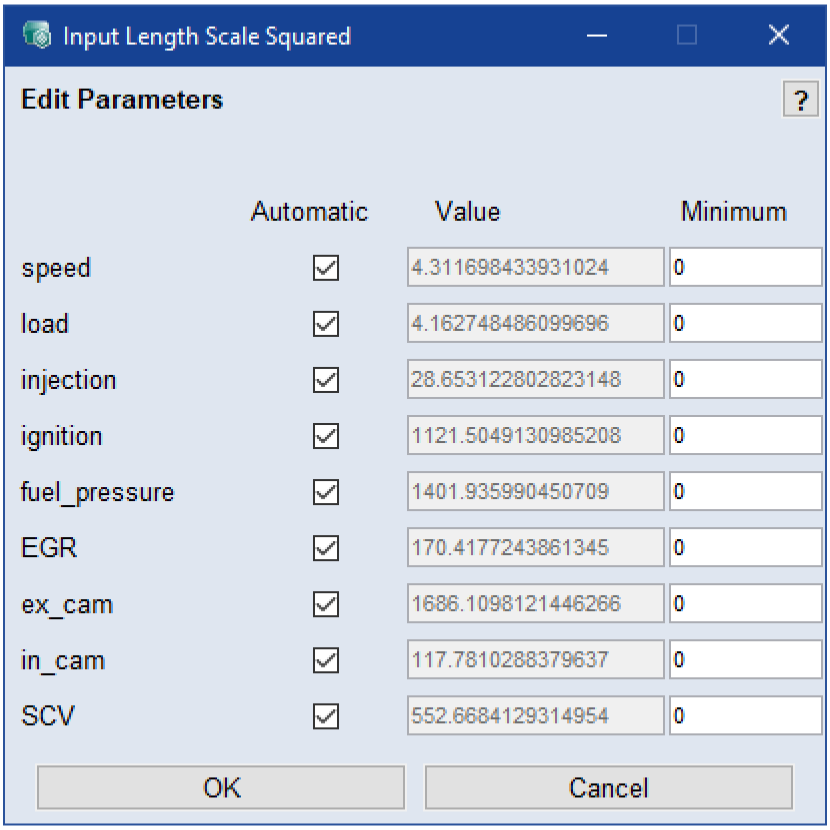
Multistart

Enter the number of times to run the optimizer with different starting values during model training. A higher number means a higher probability of finding the optimal model, but it takes more time.

Input Length Scale Squared

Click **Edit** to open the **Input Length Scale Squared** window. You can edit the hyperparameter for each input. If **Automatic** is activated for an input, the respective hyperparameter is set automatically. Otherwise, you can edit the values manually. The hyperparameter per input dimension is the core width of the Gaussian bell. The length scale is r in the following equation, so a smaller value has greater relevance:

$$y(\vec{x}) = \sum_{i=1}^{N} C_i \cdot e^{-\frac{1}{2}\sum_{l=1}^{D}\frac{(X_{il}-x_l)^2}{r_l^2}}$$



You can display the relevances graphically in the "Relevance of Inputs" Window (see ASCMO Help).

Default

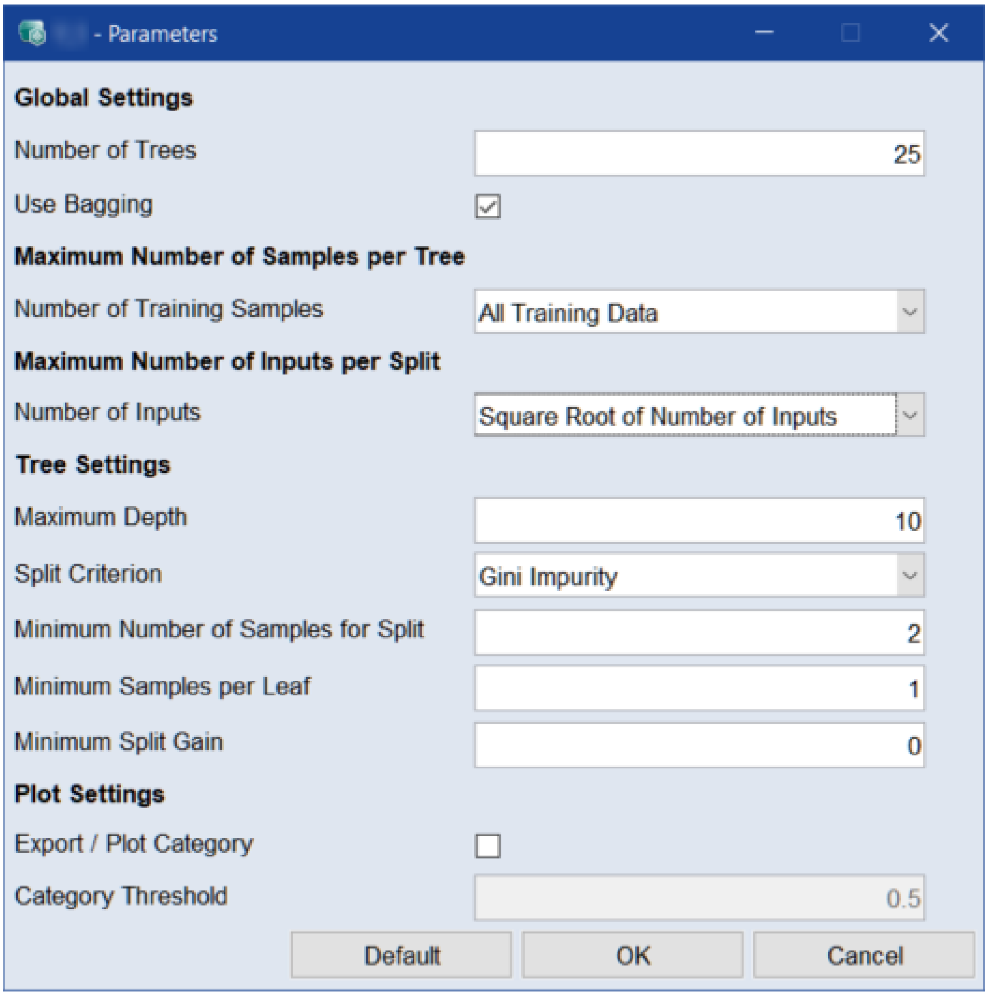
Restores the default settings for all parameters.

Note

Models of type ASC GP-SCS cannot be exported to INCA/MDA.

Classification (Random Decision Trees) Model

Random decision trees classification uses an ensemble of multiple binary decision trees. By using multiple decision trees, it reduces the risk of over-fitting. Different trees use a different subset of the training data, while the same data points can appear in multiple trees, this is called bagging. In addition to bagging, different trees model different features (set of inputs), which is the key point of the algorithm, also called random subspace sampling or feature bagging. You can define the following model parameters:



Global Settings

Number of Trees

Number of trees used by the random decision trees algorithm.

Use Bagging

If activated (recommended), each tree of the random decision trees is trained with a subset of all training data (the size of the subset can be specified by further options).

Bagging reduces the risk of overfitting.

Maximum Number of Samples per Tree

Specifies the number of random draws from the training data to determine the training data of a single tree.

Number of Training Samples

Absolute Number: An exact number can be specified.

Fraction: A value between 0 and 1 can be specified, the number of training data samples for a single tree will then be max(1, fraction * number of training data).

All Training Data: Number of draws is equal to the number of training data.

Maximum Number of Inputs per Split

In the training process, the best split of the data has to be found for each node. To avoid overfitting and to speed up the training process, it can be useful to limit the number of inputs that take part in the split decision (for each split, a subset is randomly selected). Especially for high-dimensional problems, it is recommended to use an option like **Square Root of Number of Inputs**.

Number of Inputs

All Inputs: All inputs are included.

Absolute Number: Specifies the number of features considered in a split decision, in the range [1, number of features].

Fraction: A fraction (a number in [0, 1]) of the total number of inputs is selected for each split, i.e., max(1, round(fraction * number of inputs)).

Square Root of the Number of Inputs: The total number of inputs for a split is the square root of the number of inputs, i.e., round(sqrt(number of inputs)).

2-Logarithm of Number of Inputs: The total number of inputs for a split is the logarithm to the base 2 of the number of inputs, i.e., round(log2(number of inputs)).

Tree Settings

Maximum Depth

Integer value that defines the maximum depth for each decision tree in the random decision trees.

Split Criterion

Gini impurity (between 0 and 0.5).

Shannon information gain (entropy) (between 0 and 1).

Minimum Number of Samples per Split

If the number of samples in a node is less than this value, the node is not split further.

Minimum Samples per Leaf

Splitting a node that results in leaves with less than this value is considered to be invalid.

Minimum Split Gain

Let this value be **v** and the total number of training samples **S**.

For each possible split into a left node with **n** samples and a right node with **m** samples, the gain **g** for the split is calculated internally using the split criterion.

A split is considered to be invalid if **(n+m)/S*g** less or equal than **v**.

Plot Settings

Export/Plot Category

If deactivated, the probability of mapping the input to class 1 is plotted.

If activated, the probability values greater than or equal to the threshold are mapped to class 1. The class membership is plotted.

Category Threshold

In the model evaluation, measurement data will be assigned to class 1 if it meets or exceeds this threshold. The **Plot Category** checkbox must be activated. The threshold must be a number in [0,1], the default is 0.5.

4.3 Model Assessment and Improvement

This section contains information about how to evaluate the quality of the models created by ASCMO-STATIC and, if necessary, to improve them.

- 4.3.1 "Visualization of Model Quality " below
- 4.3.2 "Methods and Data for Determining the Model Quality" on page 61
- "Variables RMSE and R2" on page 62

4.3.1 Visualization of Model Quality

A series of options is available for evaluating the models created with ASCMO-STATIC:

- "Display of Measured Values Compared to Model Prediction" below
- "Error Depending on the Size of the Training Data " on the next page
- "Display of Sigma in Intersection Plots" on page 60
- "Overview of Model Statistics" on page 61

Display of Measured Values Compared to Model Prediction

The **Model** menu provides a series of functions to compare the model predictions for the respective output with the measuring data of the output.

The following window is opened with **Model › Error (Leave-One-Out) › Measured vs. Predicted**.

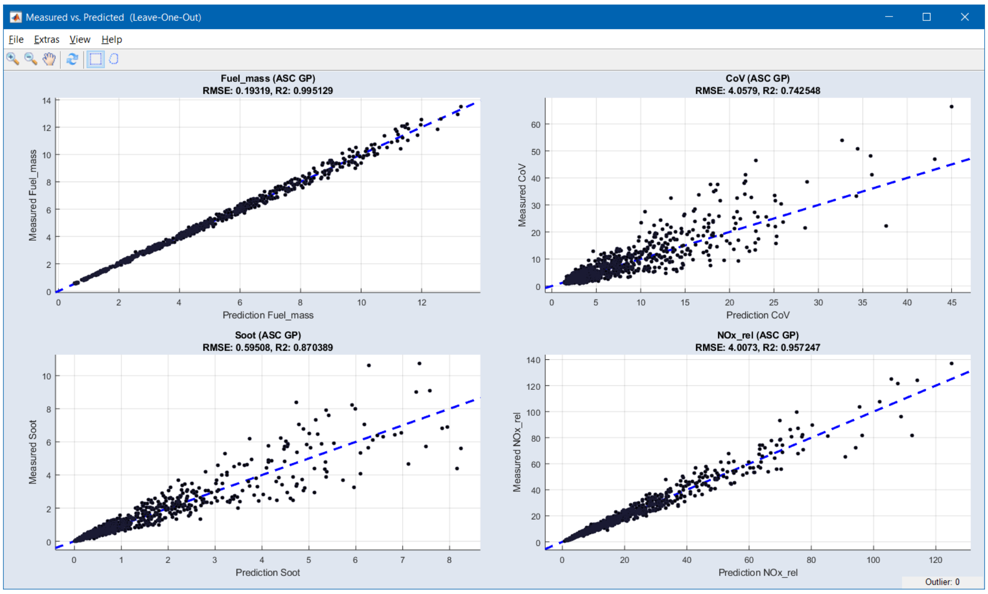


Fig. 4-10: Measuring data and model data

In these plots, the measuring points are displayed on the Y axis and the model prediction on the X axis. A perfect match between the two would result in a "pearl necklace" (y = x) on the line drawn in blue.

The larger the deviation from the blue line, the greater the difference between measurement and model prediction. This makes it possible to visually determine the model quality and to recognize outliers (see 6.5.2 "Model Improvement Through Recognition and Deletion of Outliers" on page 107).

In addition, the parameters RMSE and R^2 are displayed for every output (see 4.3.3 "Variables RMSE and R2" on page 62) – in this example determined with the Leave-One-Out method (see 4.3.2 "Methods and Data for Determining the Model Quality" on page 61).

Additional display options for comparing measuring data and model prediction are:

- Absolute or relative error versus model prediction:

  Model › Error (<method>) › Error vs. Output

  The errors versus the model prediction are displayed here (see, e.g., Fig. 6-4: "Absolute error versus model prediction" on page 109).

- Histograms and normal probability plots (absolute or relative errors):

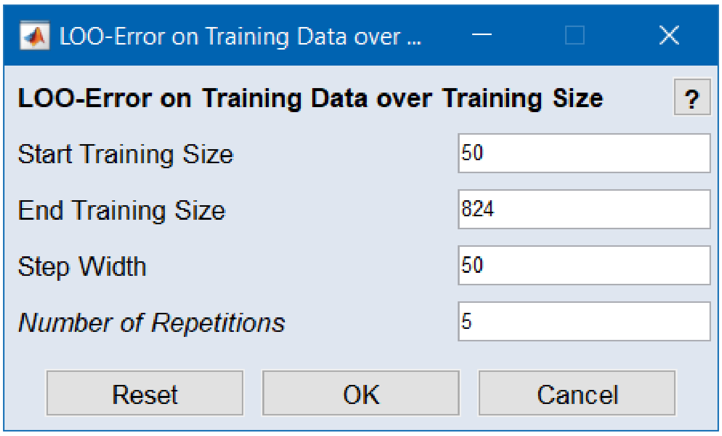
  Model › Error (<method>) › Probability Plot

  The normal probability plot shows whether the data plotted on the X axis feature a normal distribution. The actual distribution of the error (in form of bars) is drawn in the histograms next to a normal distribution (line) (see e.g. Fig. 6-5: "Normal Probability Plot" on page 110).

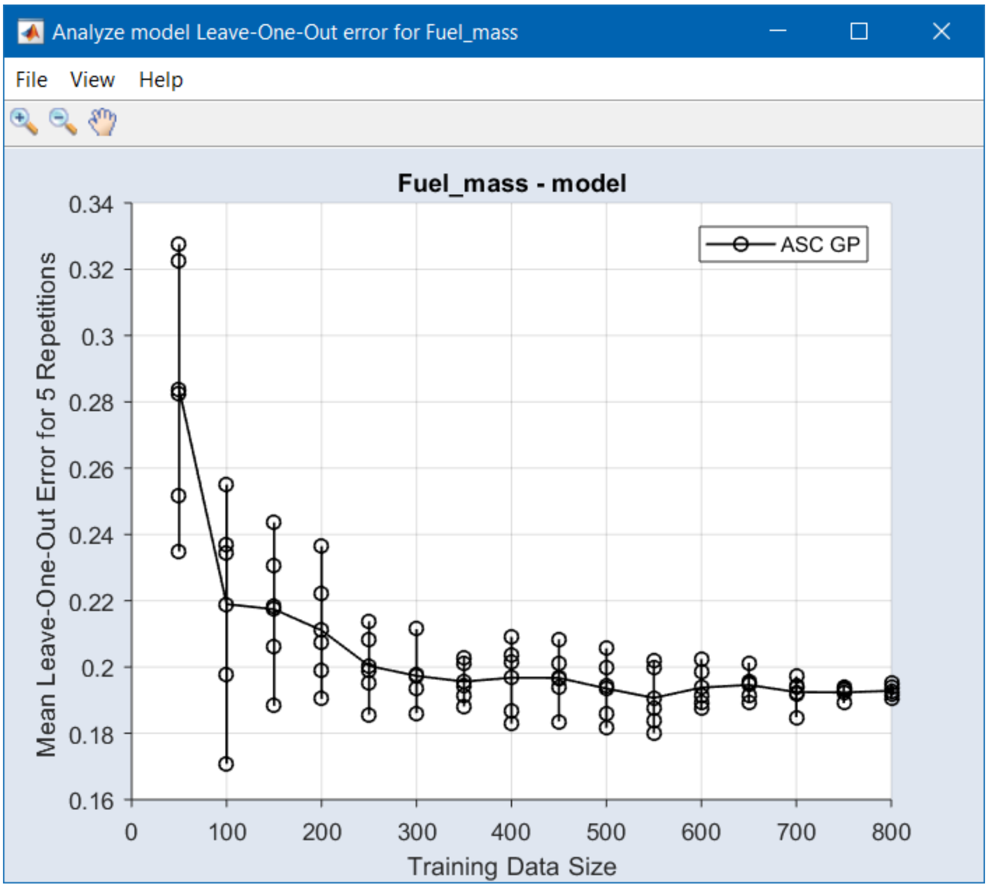
Error Depending on the Size of the Training Data

This function helps in evaluating the degree to which the size of the training data used affects the model quality.

To do so, select **Model › Error (Leave-One-Out) › Error over Training Data Size**. First, you have to specify the start training size for the investigation, the end training size, the step width to the next data size and the number of repetitions with different subsets.



The "Analyze model Leave-One-Out error for *<output>*" window then shows the average model error (RMSE) for every output with respect to the size of the training data used.



Several ("Number of Repetitions") subsets of the training data set are selected for the analysis, and the Leave-One-Out error is determined in each case. The solid line shows the mean value of the results.

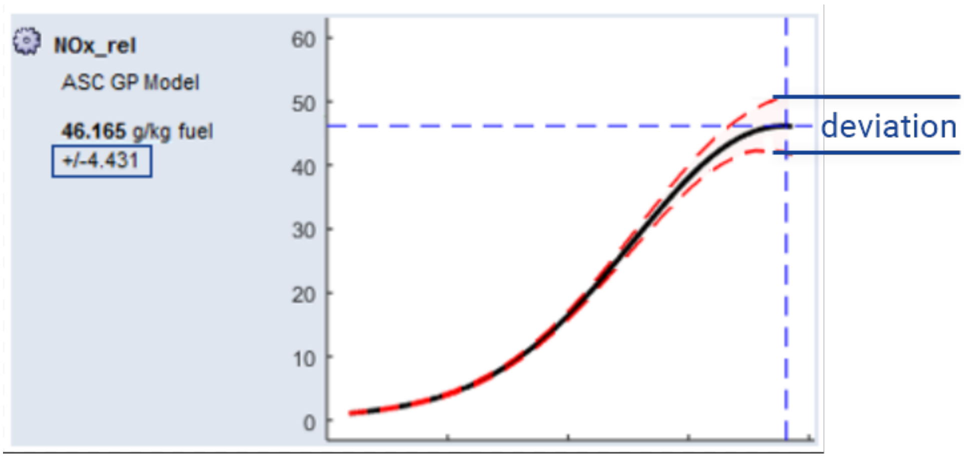
This allows identifying whether the model improves if more training data are used or if the size of the training data can even be reduced since no appreciable model improvement can be achieved starting at a certain size.

Note

The larger the subsets, the more time is required for the calculation.

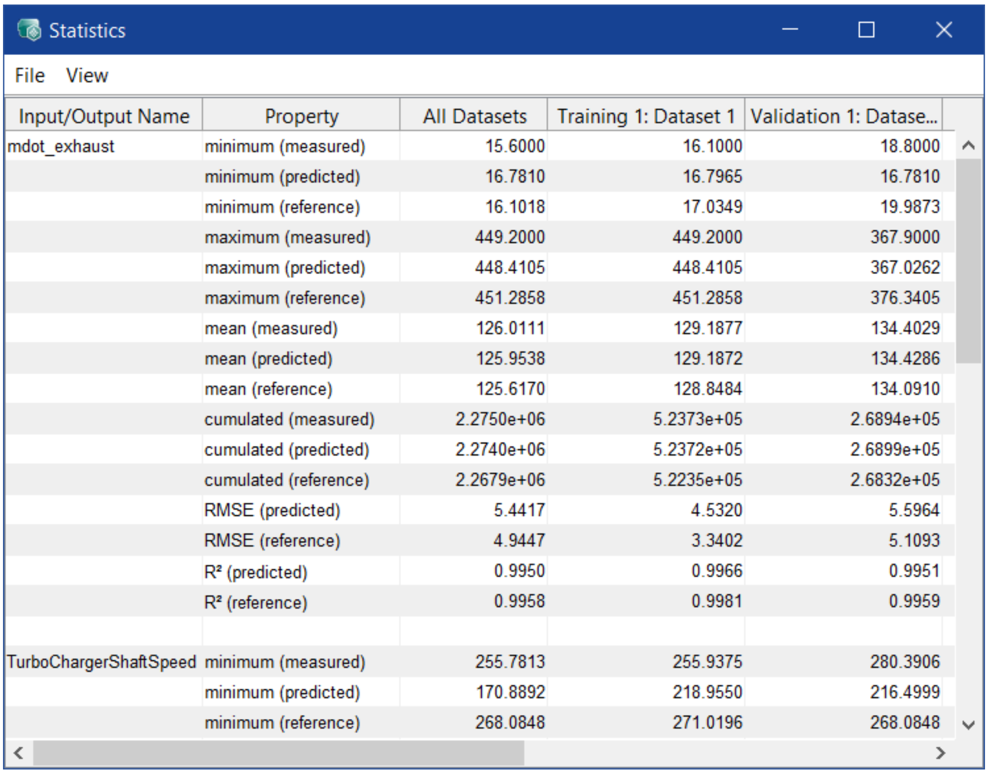
Display of Sigma in Intersection Plots

View › Show Model Sigma activates the display of the standard deviation in the intersection plot. The dashed red lines show the size of a standard deviation. In addition, this value (for the currently selected input value) is displayed next to the output value.



Overview of Model Statistics

Model › Show Statistics opens a window that contains a summary of variables, coverage and normal probability for the modeled outputs.



Input/Output Name	Property	All Datasets	Training 1: Dataset 1	Validation 1: Datase...
mdot_exhaust	minimum (measured)	15.6000	16.1000	18.8000
	minimum (predicted)	16.7810	16.7965	16.7810
	minimum (reference)	16.1018	17.0349	19.9873
	maximum (measured)	449.2000	449.2000	367.9000
	maximum (predicted)	448.4105	448.4105	367.0262
	maximum (reference)	451.2858	451.2858	376.3405
	mean (measured)	126.0111	129.1877	134.4029
	mean (predicted)	125.9538	129.1872	134.4286
	mean (reference)	125.6170	128.8484	134.0910
	cumulated (measured)	2.2750e+06	5.2373e+05	2.6894e+05
	cumulated (predicted)	2.2740e+06	5.2372e+05	2.6899e+05
	cumulated (reference)	2.2679e+06	5.2235e+05	2.6832e+05
	RMSE (predicted)	5.4417	4.5320	5.5964
	RMSE (reference)	4.9447	3.3402	5.1093
	R² (predicted)	0.9950	0.9966	0.9951
	R² (reference)	0.9958	0.9981	0.9959
TurboChargerShaftSpeed	minimum (measured)	255.7813	255.9375	280.3906
	minimum (predicted)	170.8892	218.9550	216.4999
	minimum (reference)	268.0848	271.0196	268.0848

Fig. 4-11: "Model Statistics" window

4.3.2 Methods and Data for Determining the Model Quality

Determining the model quality (calculation of RMSE and R^2, see 4.3.3 "Variables RMSE and R2" on the next page) can be performed using different methods (data). These methods are described in the following sections.

The Leave-One-Out Method

In the *Leave-One-Out method* (LOO), n models, each with n-1 training data, are formed. Afterwards, the model error of the one data point that was not involved in the model training is determined.

The big advantage of this method is that it enables a realistic model evaluation without incurring an additional measuring effort.

Test Data

Another realistic model evaluation can be obtained by verifying the model predictions against a set of test data that was not involved in the model training.

These test data can be obtained by using only a part of the imported data as training data; the (randomly selected) rest automatically becomes the test data set.

Another method, but one that requires additional measuring effort, consists of the use of data of another measurement.

Training Data

This method uses the same data that have been used for the model training.

In the model training based on Gaussian processes as in ASCMO-STATIC, this method is not meaningful since it furnishes an evaluation of the model that is far too positive.

4.3.3 Variables RMSE and R^2

A series of variables is used for quantifying the model quality.

RMSE (Root Mean Square Error)

The *Root Mean Square Error (RMSE)* is defined as follows:

$$RMSE = \sqrt{\frac{SSR}{n}}$$

Equ. 4-1: Root Mean Square Error (RMSE)

whereby n = the number of measuring data and

$$SSR = \sum_{i=1}^{n} (X_{i,pred} - X_{i,meas})^2$$

Equ. 4-2: Sum of Squared Residuals (SSR)

The RMSE describes the variance to be expected (standard deviation) about the model: A second measurement falls less than 1 RMSE from the model prediction with a probability of 68% (with 95.5% < 2 RMSE, 99.7% < 3 RMSE, etc.).

Coefficient of Determination R^2

The *coefficient of determination, R^2*, is derived from the comparison of the variance that remains after the model training (SSR) with the variance concerning the mean value of all measuring data (SST)

$$R^2 = 1 - \frac{SSR}{SST}$$

Equ. 4-3: Coefficient of determination R^2 whereby

$$SST = \sum_{i=1}^{n} (X_{i,meas} - \overline{X}_{meas})^2$$

Equ. 4-4: Total Sum of Squares (SST)

R^2 is a relative measure for evaluating the model error – it indicates which portion of the total variance of the measuring data is described by the model.

Evaluation of the Model Using RMSE and R^2

The most important variable is the coefficient of determination R^2 – this measure results in the following evaluations:

- $0 < R^2 < 0.5$

  The model is not suitable for reliable predictions.
- $0.6 < R^2 < 0.8$

  The model is suitable for qualitative predictions.
- $0.9 < R^2 < 1$

  The model is very good and therefore suitable for quantitative predictions.

The absolute error RMSE must be evaluated individually:

- At best, the RMSE can be as good as the experimental repeatability.
- Despite a good R^2, the RMSE can be too low, e.g. in case of a very large variation range of the modeled variable.
- Despite a low R^2, the RMSE can be good enough, e.g. if the modeled variable features only a minor variance over the input parameters of the model.

4.4 Advanced Settings in ASCMO-STATIC

This section gives you an overview on the different advanced parameters and describes how to enable and disable the visibility of the *advanced settings*.

If you have activated the advanced settings, the added menu items are shown in italics.

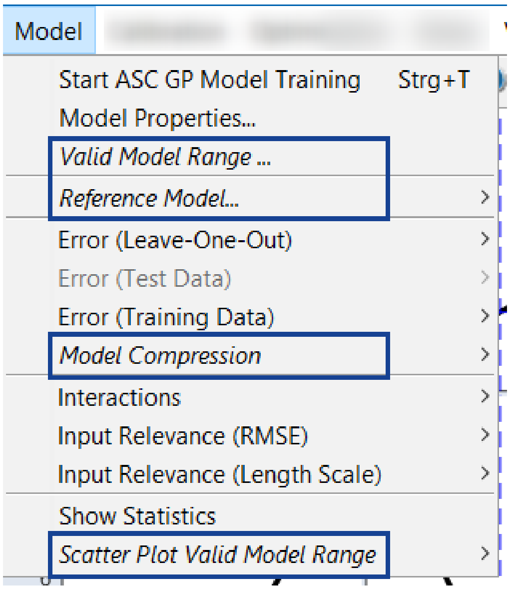


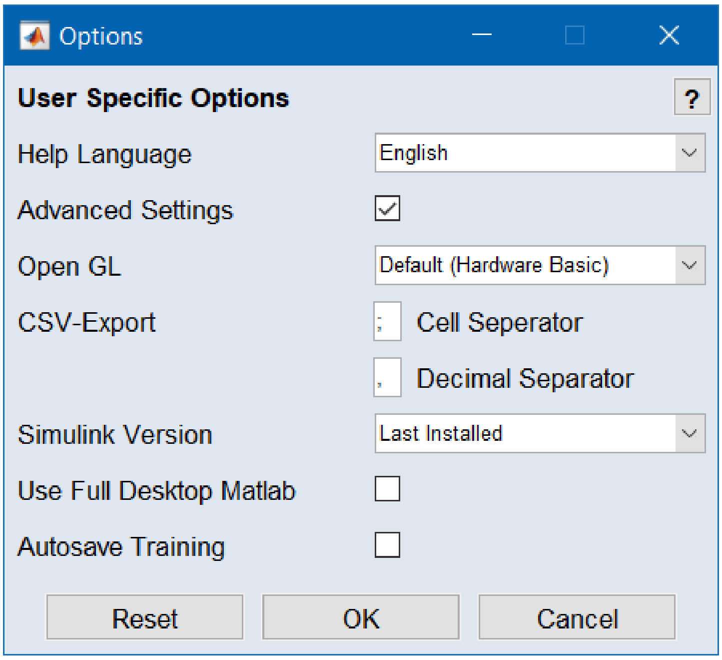
Fig. 4-12: Presentation of the advanced settings menu entries (example)

> **Note**
>
> For more information on advanced settings, see the online help (F1).

4.4.1 Enable/Disable Advanced Settings

1. Select **File › Options**.

   The "Options" window opens.



2. Enable the **Advanced Settings** option.

3. Click on **OK**.

⇨ The advanced settings are enabled and are now visible in ASCMO-STATIC. The "Options" window closes.

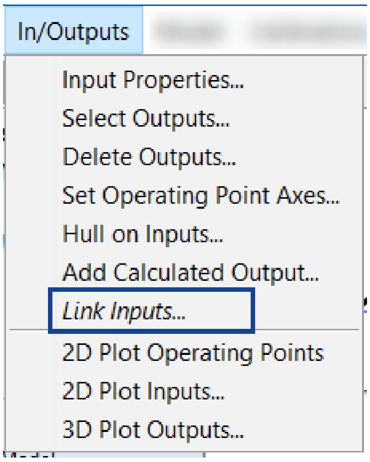
> **Note**
>
> In case of ASCMO-DYNAMIC, enabling/disabling "Advanced Settings" only takes effect after the project is re-opened.

4.4.2 Overview of Advanced Settings

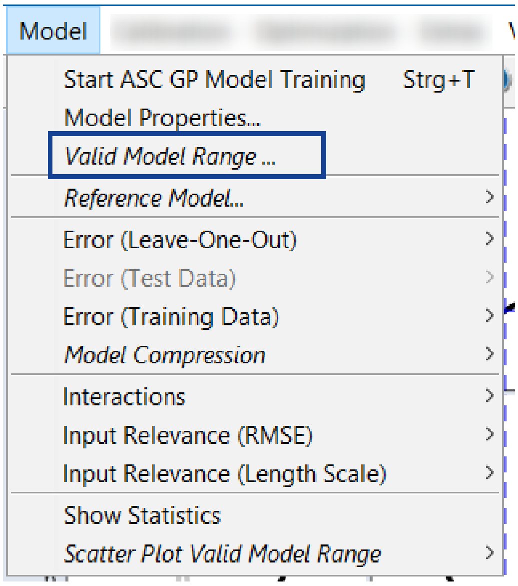
If you have enabled the advanced settings (**File › Options**), the following functions are available in ASCMO-STATIC:

– Menu entries

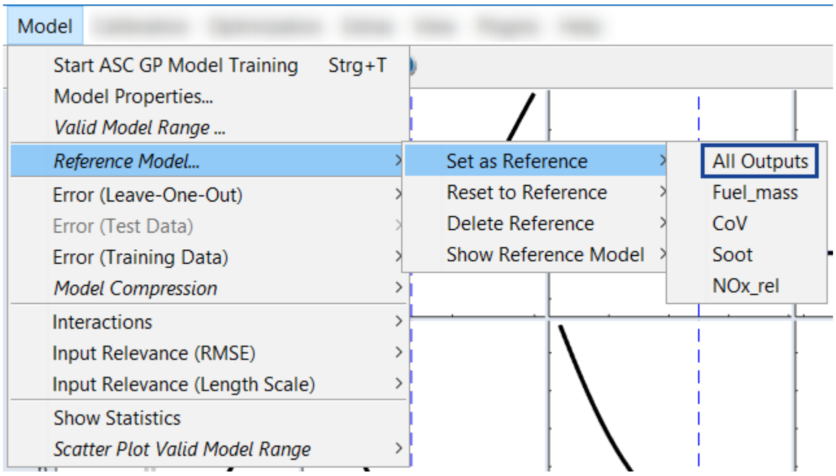
- **In/Outputs › Link Inputs**



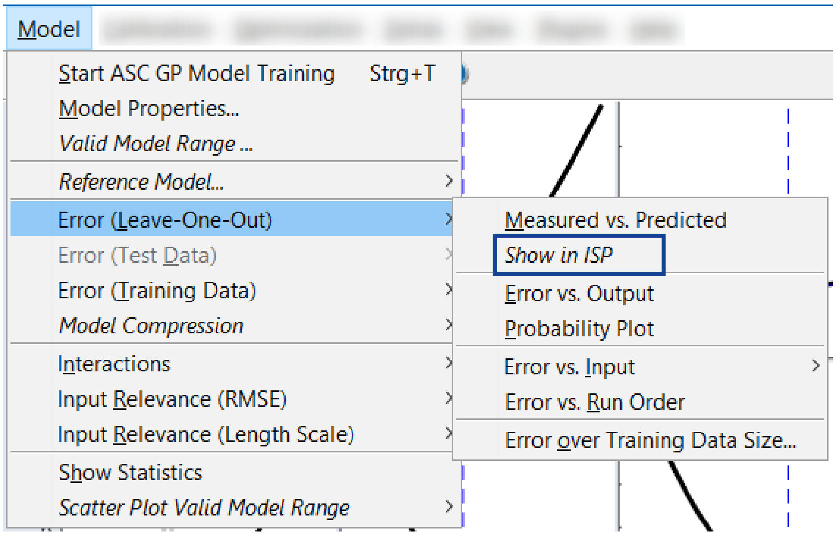
- **Model › Valid Model Range**



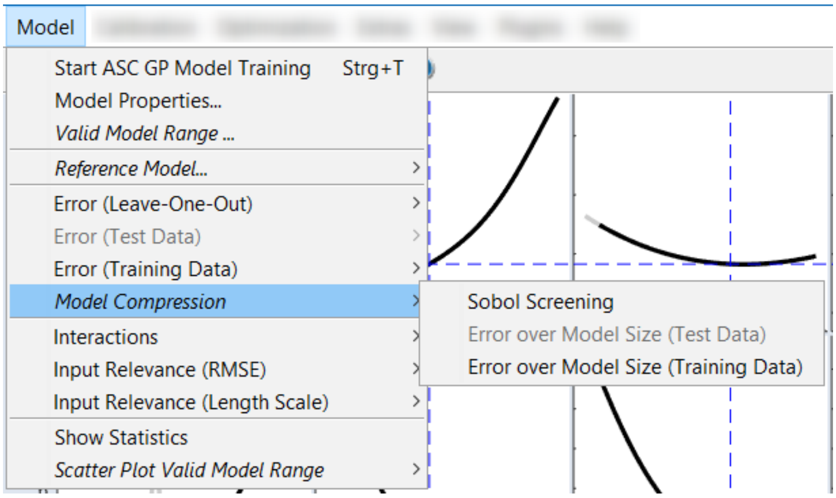
- **Model › Reference Model › * › ***



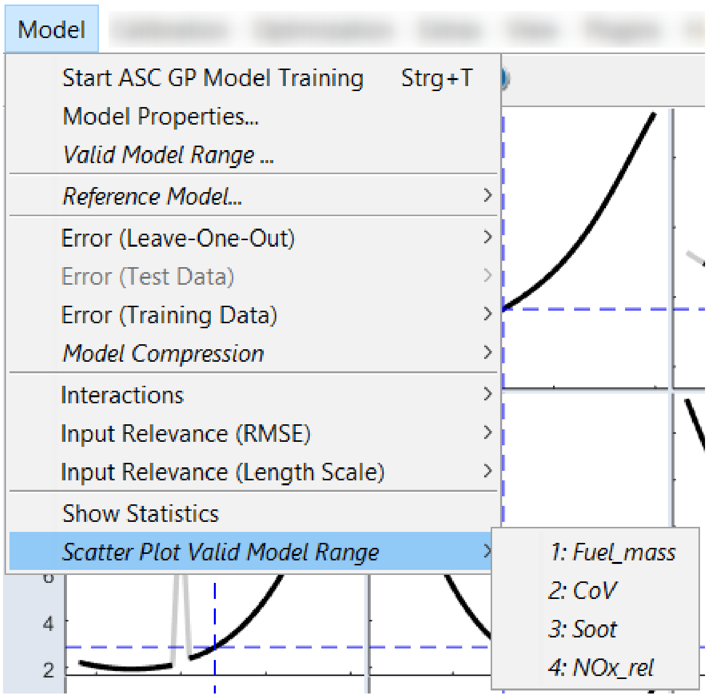
- **Model › Error (Leave One Out) › Show in ISP**

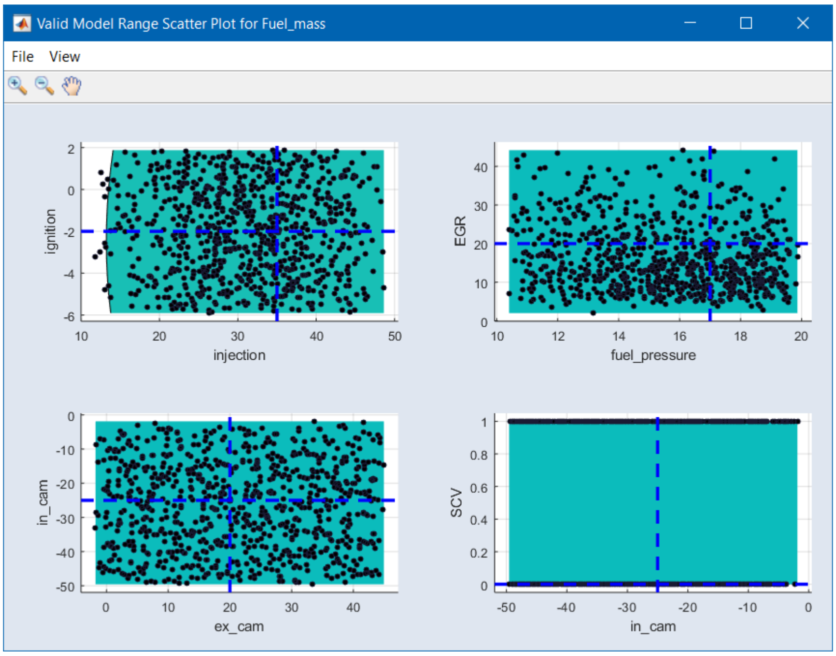


- **Model › Model Compression › ***

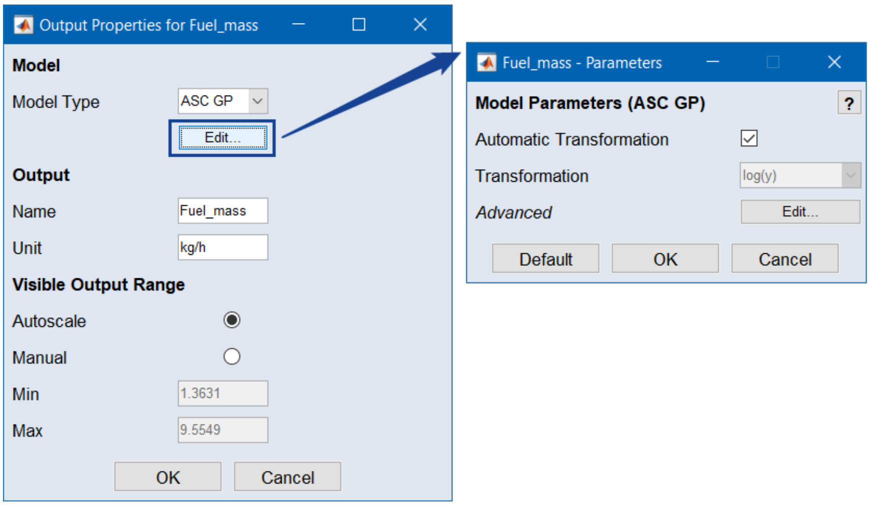


- **Model › Scatter Plot Valid Model Range › ***

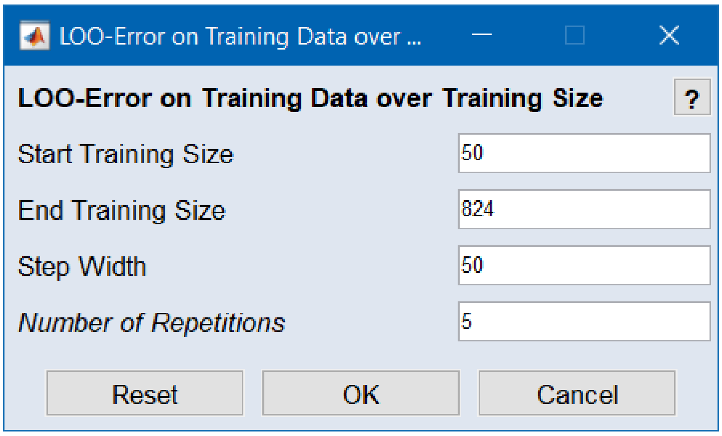




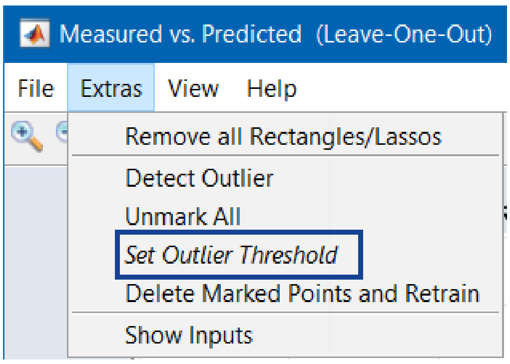
- Dialog windows
  - Advanced Parameter for outputs/models

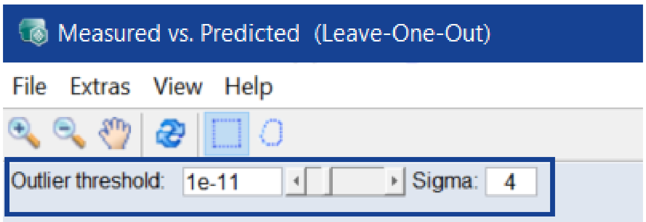


  - **Model › Error (Leave One Out) › Error over Training Data Size**: "Number of Repetitions" field in the "LOO-Error on Training Data over Training Size" window



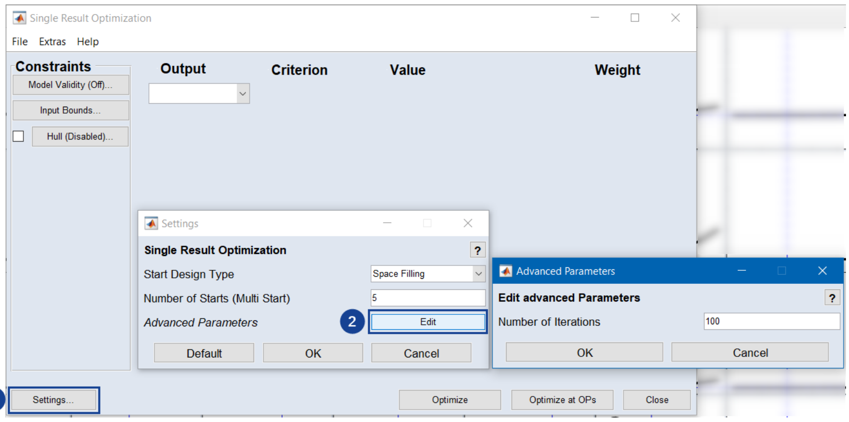
- **Model › Error (Leave One Out) › Measured vs. Predicted**: Menu entry "Set Outlier Threshold" in the "Measured/Error vs. Predicted" window, see **Extras › Set Outlier Threshold**

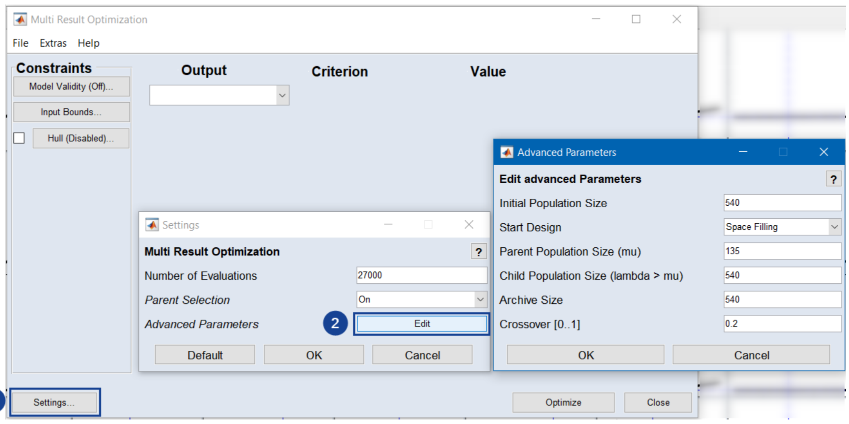




– **Optimization › * Optimization**: Settings in "Single Result Optimization"/"Multi Result Optimization" windows

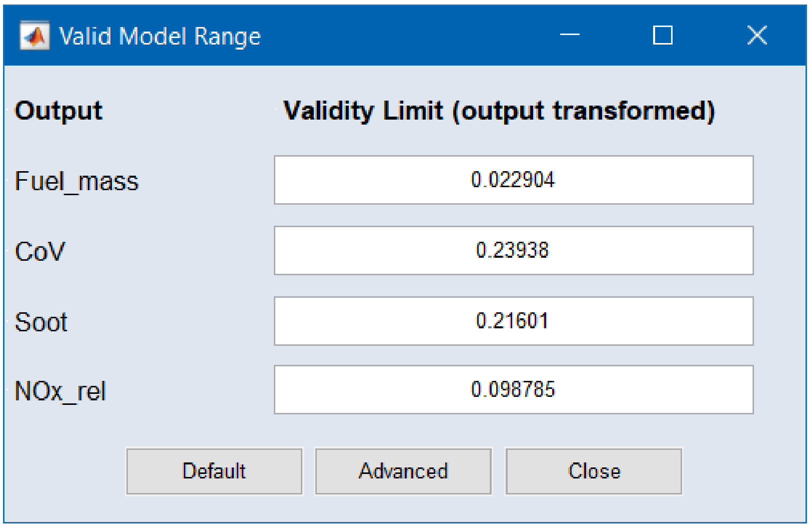
- **Settings › Advanced Parameter: Edit**





– Dialog window "Valid Model Range"

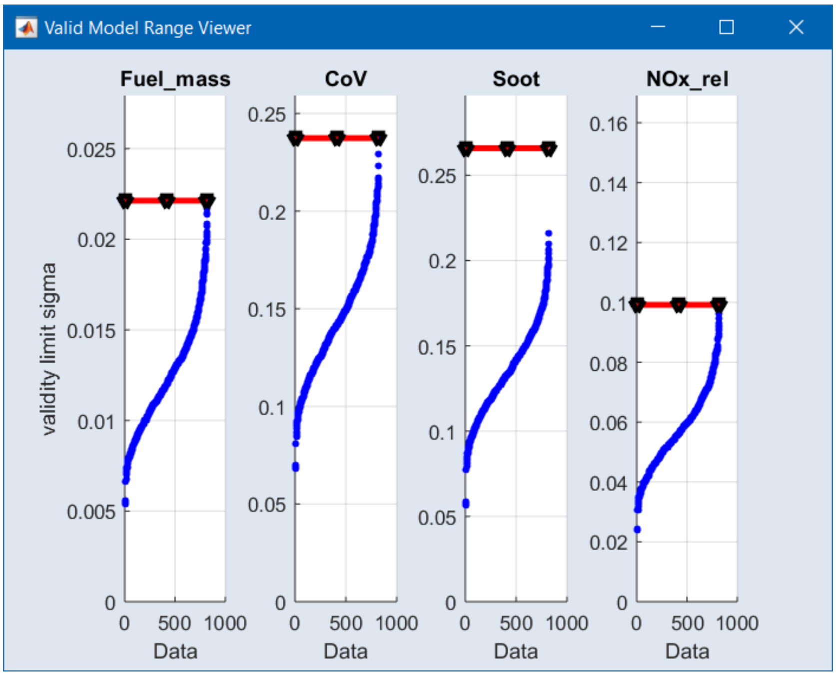
Allows setting a threshold (absolute value of the maximum standard deviation) with which the valid range of the model (output) is determined. The range is displayed with the **Model › Valid Model Range** menu option.



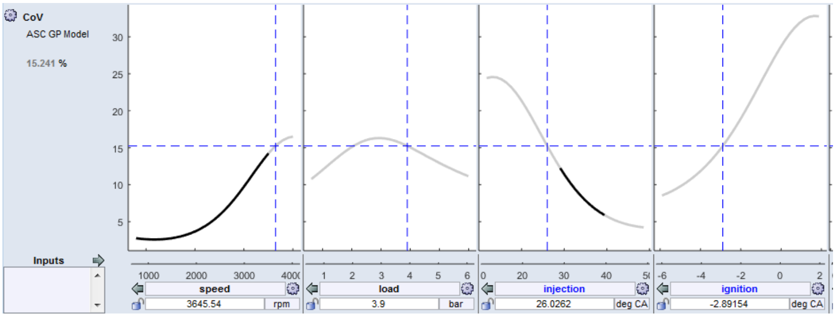
Clicking on **Default** determines a heuristic value for this threshold.

This value corresponds to the maximum standard deviation on the training data of the output. Model predictions with a smaller standard deviation are then rated as valid.

Clicking on **Advanced** opens a viewer in which the data are displayed with the threshold. The red line with the black triangles can be moved with the mouse.



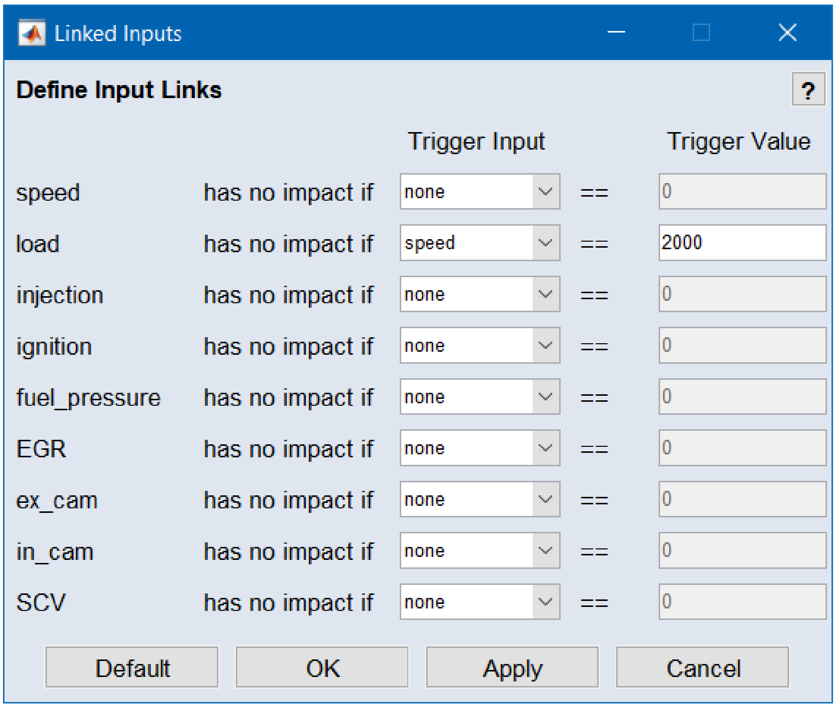
In the ISP View you can also see the invalid values of the Inputs and the invalid areas



– Dialog "Linked Inputs"

This window allows to set conditions for one or more inputs, such that selected outputs have no influence on the model prediction under those conditions.

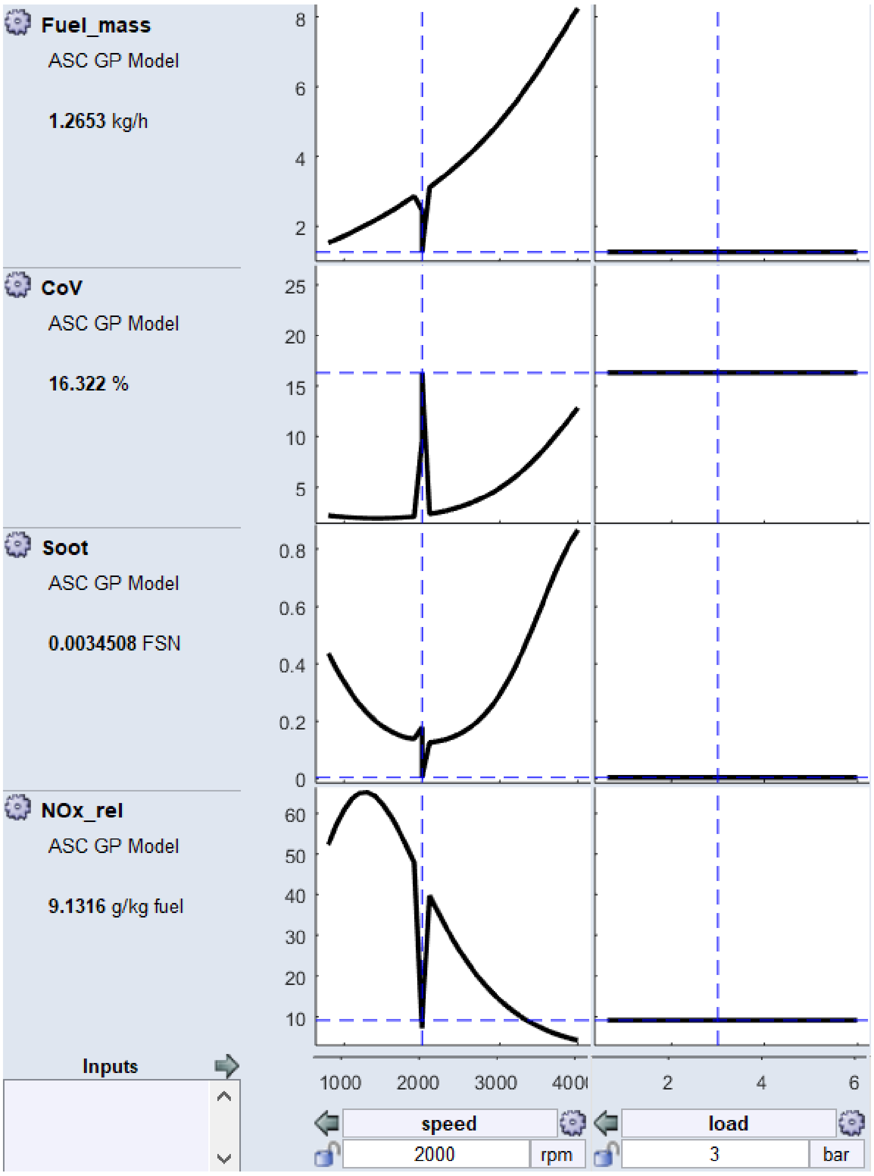
- Example: load has no impact if speed == 2000



Note

You can set several conditions simultaneously.

In the ISP view you can see, that load has no more influence.



4.5 Optimization

This section contains a description of the different optimization methods and the optimization criteria that can be used for them.

4.5.1 Description of Optimization Methods

Single-Criteria Optimization

This is the optimization of a variable or weighted total of several variables according to a gradient descent. The result is a set of setting parameters (input values) for the desired optimization target (see 4.5.2 "Optimization Criteria " on page 74).

An example from engine engineering: Minimize fuel consumption while maintaining limit value for NO_x emission.

The single-criteria optimization is started via **Optimization > Single Result**.

Optimization at Several Operating Points

This is a single-criteria optimization performed multiple times (one for each operating point). If the entire operating range is covered, it can result in the creation of maps.

After defining the operating point grid, the optimization is automatically performed at all of these points. For the resulting maps, it is possible to select different grid nodes.

The optimization at several operating points is called up via **Optimization > Single Result** and clicking **Optimize at OPs**.

Note

The **Optimize at OPs** button in the "Single Result Optimization" window is only available if operating point axes have been selected (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards via **In/Outputs > Set Operating Point Axes**.

Multi-Criteria Optimization

This is a true multiple-target optimization that leads to a set of Pareto-optimal solutions. At that point, the selection of the solution can also be performed by means of other criteria (e.g. the values of other inputs or outputs).

The multi-criteria optimization is called up via **Optimization > Multi Result**.

Global Optimization

Here the optimization procedure is carried out simultaneously (using a gradient descent) at all operating points. The results are calibration maps - as is the case with successive optimization at several operation points.

Regarding the target criteria, prognosis data (e.g. a driving cycle) as well as smoothness of the resulting maps (2nd derivative) can be considered. The contribution of a component to the target criterion (the *weight*) can be adapted to individual requirements.

The global optimization is started via **Optimization > Global Optimization**.

Note

The **Global Optimization** function in the **Optimization** menu is only available if operating point axes have been selected (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards via the menu **In/Outputs > Set Operating Point Axes**.

4.5.2 Optimization Criteria

There are a series of target criteria for the optimization of one or several outputs that are described in the following table.

Criterion	Value	Meaning
Minimize/ Maximize	None	Optimization to minimum/maximum value without further restrictions.
	Hard Upper/ Lower Bound	Optimization to minimum/maximum value with an additional upper/lower limit.
Target		Optimization to target value – the amount of deviation from target is minimized.
Bound		Optimization with limits.
	Hard Upper / Lower Bound	Specifies fixed limits for the optimizer – if these values are overrun/underrun, the optimization concludes without a result. In case of an optimization of the type "Single Result at OPs" the "weak upper/lower bound" criterion is used for further optimization.
	Weak Upper-/Lower Bound	A weakened form of "Hard ... Bound". Minimizing/maximizing up to a threshold, all values below or above it are not optimized any further.

Tab. 4-2: Meaning of optimization criteria

4.5.3 Evolutionary Algorithm (Parent Selection vs. Survivor Selection)

An evolutionary algorithm (EA) is a subset of evolutionary computation, a generic population-based metaheuristic optimization algorithm. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions. Evolution of the population then takes place after the repeated application of the above operators.

Sequence of Evolutionary Algorithms

The evolutionary algorithms used in ASCMO-STATIC consist of an initialization and a loop that will be executed until a defined stop criterion is reached. You can specify the stop criterion (**Number of Evaluations**) for a multi-result optimization in the "Multi Result Optimization" window using the settings (**Optimization › Multi Result › Settings**).

1. Initialization: Generate the initial population of individuals randomly – first generation.
2. Evaluation: Evaluate the fitness of each individual in that population.
3. Repeat on this generation until the stop criterion (Number of Evaluations) is reached:
   i. Selection: Select the best-fit individuals for reproduction.
   ii. Recombination: Breed new individuals through crossover.
   iii. Mutation: Random change of the descendants.
   iv. Evaluation: see above (Step 2)
   v. Selection: Determination of a new generation.

ASCMO-STATIC then distinguish between the following selection processes.

Parent Selection

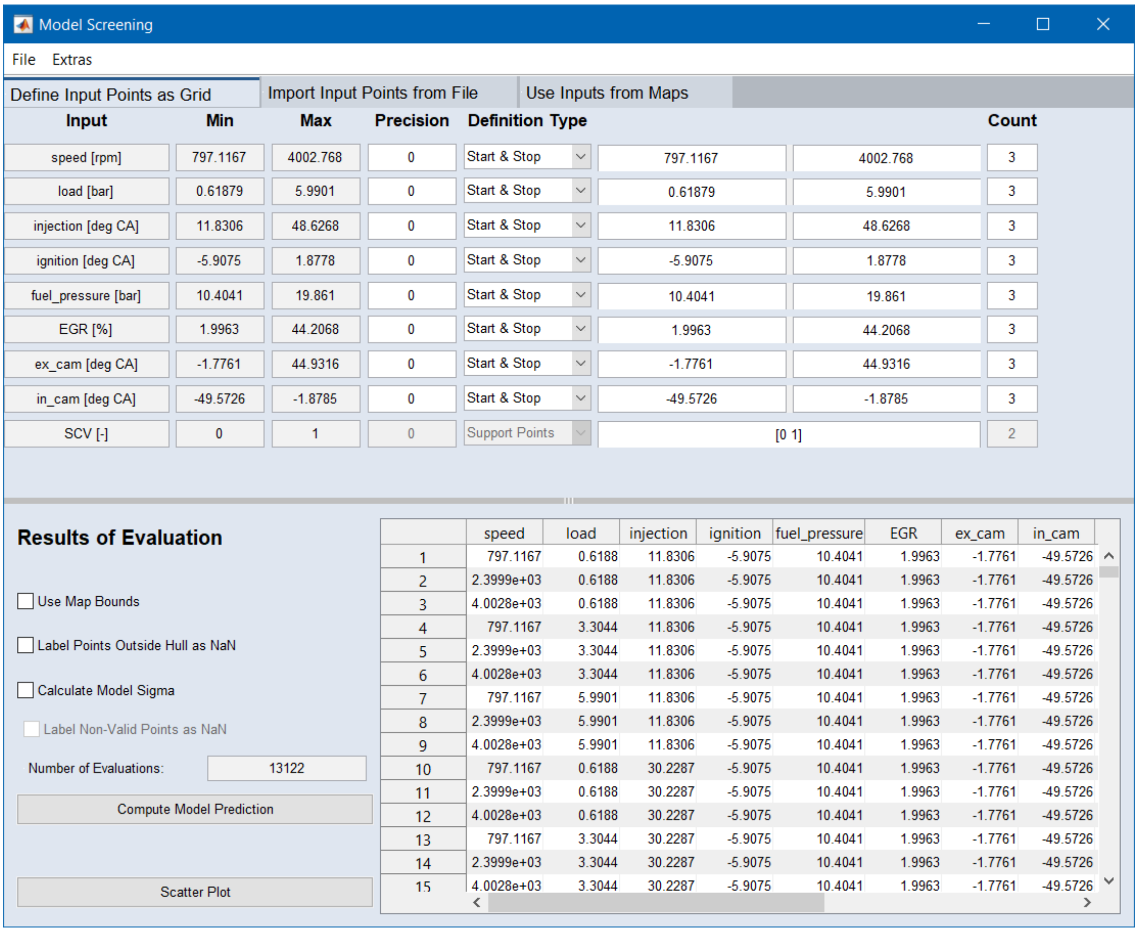
In this selection process, the parents are involved in the evaluation and recombination of new generations. The background to this is that the random mutation of the descendants, thus creating new child generations, not necessarily strives towards the optimization target. They may also be better than the child generation according to a fitness function.

Survivor Selection

In this selection process, the parent generations are not included in the evaluation and recombination.

4.6 Model Screening (Model Evaluation)

The model screening (**Extras > Model Screening**) allows you to make a model evaluation in ASCMO-STATIC. This means that you can represent specific output values at certain input values from the model in the form of a table (**Results of Evaluation**). The result of the model evaluation then can be visualized in a scatter plot or exported in a table (**File > Export Results** or [Strg/ctrl+E]).



On which input values you want to query the respective output data must first be defined in the "Model Screening" window. Here you have the following options:

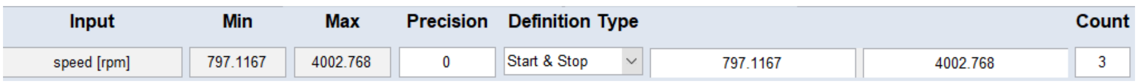
- 4.6.1 "Definition of Operating Points as Grid" below
- 4.6.2 "Import Input Points from File" on the next page
- 4.6.3 "Use Inputs from Maps" on the next page

4.6.1 Definition of Operating Points as Grid

The definition of the operating point grid for which the outputs should be calculated is entered manually. It will be distinguished between two types (**Definition Type**) during the generation of the OP grid.

Definition Type `Start & Stop`

Here, a uniform grid will be generated for the entire range of parameter variation (minimum/maximum value of the input) automatically. You can also define the number of grid nodes to be distributed manually ("Count" column).



For the grid definition presented above, 3 grid points have been generated. The first grid point is located at the minimum input value ("Min" column) and the third point on the maximum input value ("Max" column). The second point is generated by ASCMO-STATIC and in this case is exactly in the middle of the other two points.

Note

If you select "Support Points" in the "Type Definition" field, the automatically calculated value will be displayed.

Definition Type `Support Points`

If you want to add other grid nodes next to the grid nodes evenly distributed as desired, you can specify the appropriate input values in the array.

Input	Min	Max	Precision	Definition Type		Count
speed [rpm]	797.1167	4002.768	0	Support Points	[797.116701773741 2399.94232870996 4002.76795564618]	3

If you change the number of grid nodes in the "Count" column, the newly calculated value will be added to the array. If you enter an input value manually, the number of grid nodes will be automatically updated in the "Count" column.

4.6.2 Import Input Points from File

Here you can specify a file from which the input values will be read. Basically, the file formats `*.xls`, `*.csv`, `*.txt` or `*.ascii` can be loaded. So you can generate, for example, a template that allows you to display the corresponding output values for the input values contained in different models.

Note

With the "Rounding of Values" option, you can round the values to be read from the file.

4.6.3 Use Inputs from Maps

Note

The "Use Inputs from Maps" tab in the "Model Screening" window is only available if operating point axes have been selected (see Assigning Input and Output Variables). You can set the operating point axes afterwards via **In/Outputs › Set Operating Point Axes**.

Here you can specify the operating point lists and grid nodes from maps, from which the values of the inputs will be read. You can select different lists and maps:

- **Manual OP grid**

  Define the operating points grid manually, for which you want to calculate and display the output values. The values for the operating point axes (e.g., speed and load) are defined. The other values will be read from the existing maps.
- **Grid nodes from Maps**

  Select an existing map, from which the grid nodes will be taken for the calculation of the output values.
- **OP lists**

  Select an existing list of operating points from which the operating points will be read for the calculation of the output values.
- **Driving Cycles**

  Select a driving cycle, from which the cycle trajectory data should be transferred as input values for the calculation of the output values.
- **OP list from file**

  If you want to use a list of operating points that is not yet stored in ASCMO-STATIC, you can specify a list of operating points from which the operating points are assumed for the calculation of the output values. This list will only be used for the model prediction and will not be imported for further processing in ASCMO-STATIC.

Note

If you enable the option **Allow duplicates**, operating point duplicates will be included in the model evaluation.

4.6.4 Result of the Model Prediction (Result of Evaluation)

The result of the model evaluation are the values of the outputs, which are determined based on the defined inputs. They are shown in the "Results of Evaluation" area in tabular form.

First, you can define additional settings and restrictions that have an effect on the model evaluation.

- **Use Map Bounds**

  Ignores input values for the model prediction that violate Map Bounds (**Calibration > Map Bounds over OP**).
- **Label Points Outside Hull as NaN**

  Input points that are outside of the selected hull (**In/Outputs > Hull on Inputs**) will be marked as "NaN" (Not a Number) in the table. The hull evaluation will be done per output.

- **Calculate Model Sigma**

  Here the model uncertainty (Sigma) will be calculated for each output value and displayed in the "Sigma `<output>`"column. This option is a prerequisite for enabling the option "Label non-valid Points as NaN".
- **Label non-valid Points as NaN**

  With this option enabled, evaluation results will be marked as "NaN" (Not a Number) where the model validity is false.

Note

This option is only enabled if **Calculate Model Sigma** is activated.

The number of evaluations appears in the "Number of Evaluations" field. The number of evaluations generally result from the number of rows in the table "Result of Evaluation" and is crucial the duration of the model prediction calculation.

The button **Compute Model Prediction** starts the calculation of the model prediction for the defined input values.

5 Working with ASCMO-STATIC

5.1 User Interface of ASCMO-STATIC

This section provides an overview of the user interface of ASCMO-STATIC.

A detailed description of the functions of the main menu and the various dialog windows associated with it is located in the context-sensitive online help (‹F1› or **Help › Online Help**). An introduction to working with the user interface is also provided in the tutorial (see 6.6 "Visualizing" on page 111).

5.2 Elements of the ASCMO-STATIC User Interface

The following figure shows the user interface of ASCMO-STATIC with an opened project. The main working window shows the modeled dependencies of the outputs on the inputs as intersection plots and is also called *ISP view*.

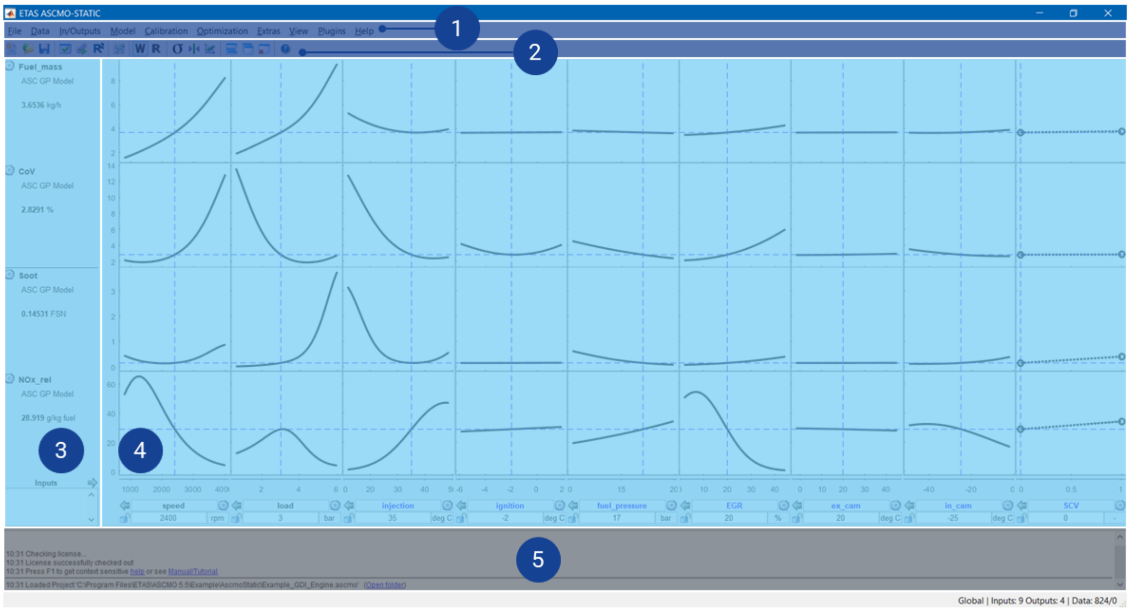


Fig. 5-1: Graphical user interface (GUI) of ASCMO-STATIC (with opened project)

The GUI is divided into 5 different parts:

- ① Main menu
- ② Toolbar
- ③ Outputs
- ④ Main working window: Intersection Plot (ISP View) and Inputs
- ⑤ Log window
- Status bar (footer) with current state information

The main menu offers several operations for user activities. In addition, there are a series of interaction possibilities in the areas with the inputs (bottom) and outputs (left) and the intersection plots (see 5.3 "Intersection Plots " on page 84) that are described below.

Main Menu

A detailed description of the function of the main menu and the dialog windows associated with it is located in the context-sensitive online help (‹F1›).

Inputs

The inputs x_1,..., x_n (under the intersection plots) and the outputs y_1,..., y_m (to the left of them) form a matrix at whose intersections the respective intersection plots $y_1 = f(x_1)$,..., $y_m = f(x_n)$ are displayed.

In the figure, these are "NOx_rel = f(injection)" and "NOx_rel = f(ignition)".

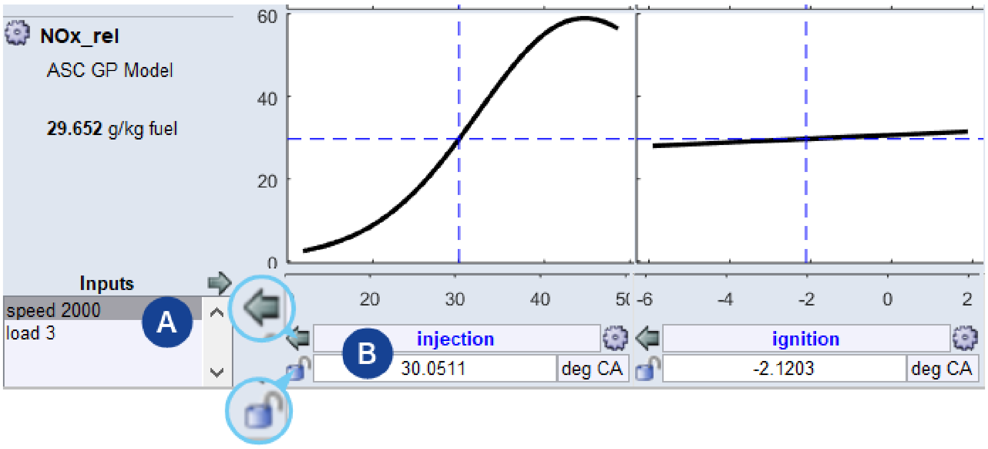


Fig. 5-2: ISP view: Inputs

- **List of hidden inputs (A)**

  All inputs not displayed in the ISP view are listed here. Double-click on an entry to show it in the ISP view.

- **Name of the respective input (B)**

  Clicking on the icon to the right of this field opens the "*‹input›* - Parameters" window.

  In that window, you can change input name and unit, set the visible range, and configure the input.

- **Current value of the respective input (B)**

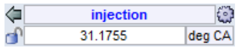
  A change of the value is made with an entry in this field or by clicking in the intersection plot.

- **Hides the corresponding input**

  The status (locked/unlocked/use map) is not changed when you remove an input from the display.

– **Changes the state of the current input**

The following states are available:

unlocked	The value can be changed manually and by the optimization. The input name appears in blue, and the value field (2) can be edited.	
locked	The value is fixed. It cannot be changed manually or by the optimization. The input name appears in gray, and the value field (2) is set to read-only.	
use map	The input value is interpolated from the respective calibration map at the point of the current speed and load. The input name appears in red, and the value field (2) is set to read-only.	

Outputs

The modeled outputs are displayed to the left of the intersection plots.

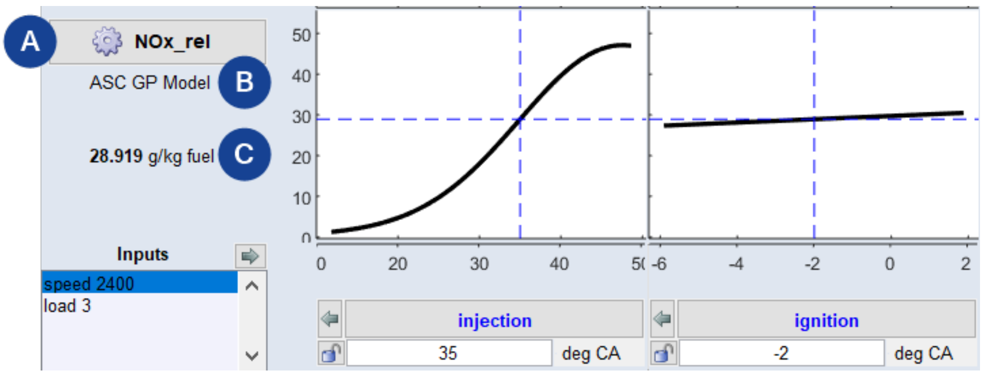
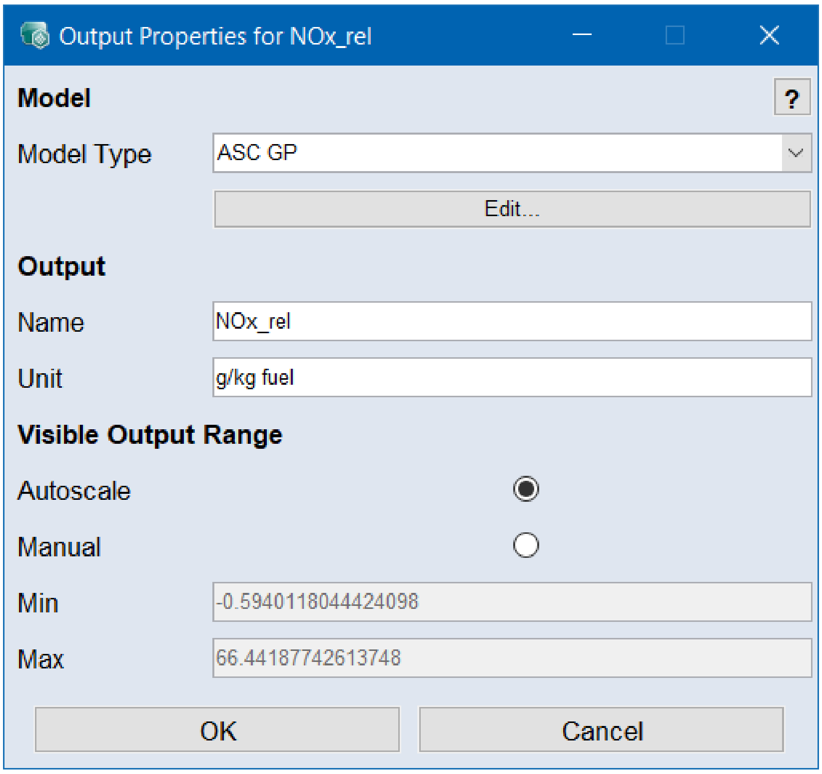


Fig. 5-3: ISP view: Outputs

– **Name of the respective output (A)**

  Clicking the button opens the "Output Properties for *‹output›*" window.



  In that window, you can select the model type, access the model parameters via the **Edit** button, change output name and unit, and set the visible range.

  For example, the model type can be selected here. The transformation of an output (see 6.5.1 "Model Improvement Through Transformation of Output Variables" on page 107) can be manually defined in the model type parameters, accessible via the **Edit** button. Details about the meaning of the other settings can be found in the online help (‹F1›).

– **Model type of the output (B)**

  Clicking on the model type name opens the model parameter window; see 4.2.5 " Model Types of ASCMO-STATIC" on page 32 for further information on the model types.

– **Current value of the output (C)**

  Current value of the output (corresponding to the selected value of the respective input).

Log Window

The bottom part of the main window displays information about executed functions, error messages, etc.

The blue underlined words in the log window are links that give you the ability to access context-sensitive information from the Help system (press F1 to get context-sensitive **help** in HTML format or see **Manual/Tutorial** in PDF format) or provide information about activities and functions that can be performed as part of the optimization process (e.g., "Created new project. You should check your training data.")

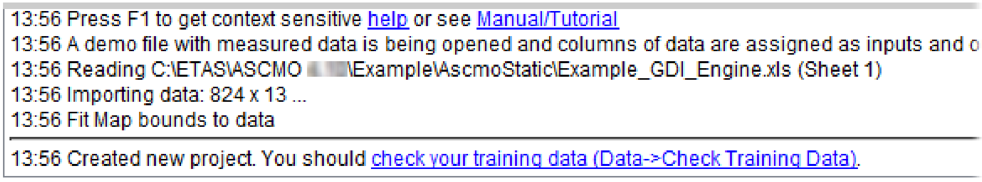


Fig. 5-4: Information in the log window (example)

Save log file

1. Right-click in the log window and select **Save Log to File** from the context menu.

   The "Save Log file As" window opens.

2. Specify the file name and click **Save**.

   The log file is saved.

5.3 Intersection Plots

The model training for an output leads to a function of the following form:

$y_{output} = f(x_{input\_1},...,x_{input\_n})$

With respect to the display, this results in a hyperplane in an n+1-dimensional space that can no longer be graphically displayed as soon as n › 2.

Instead, the main window in ASCMO-STATIC shows n intersections through this hyperplane, the so-called *intersection plots.* With each of these intersections, only one dimension is being varied, while the values of the other dimensions are kept constant, resulting in

$y_{output} = f(x_{input\_1}), ..., y_{output} = f(x_{input\_n})$

With n inputs and m outputs, this results in n x m intersection plots. The inputs x_1, ..., x_n (in the user interface underneath the intersection plots) and the outputs y_1, ..., y_m (to the left of them) form a matrix at whose intersections the respective intersection plots $y_1 = f(x_1)$, ..., $y_m = f(x_n)$ are displayed.

This is shown again in Fig. 5-5 for the case to be displayed of the dependency of one output on two inputs.

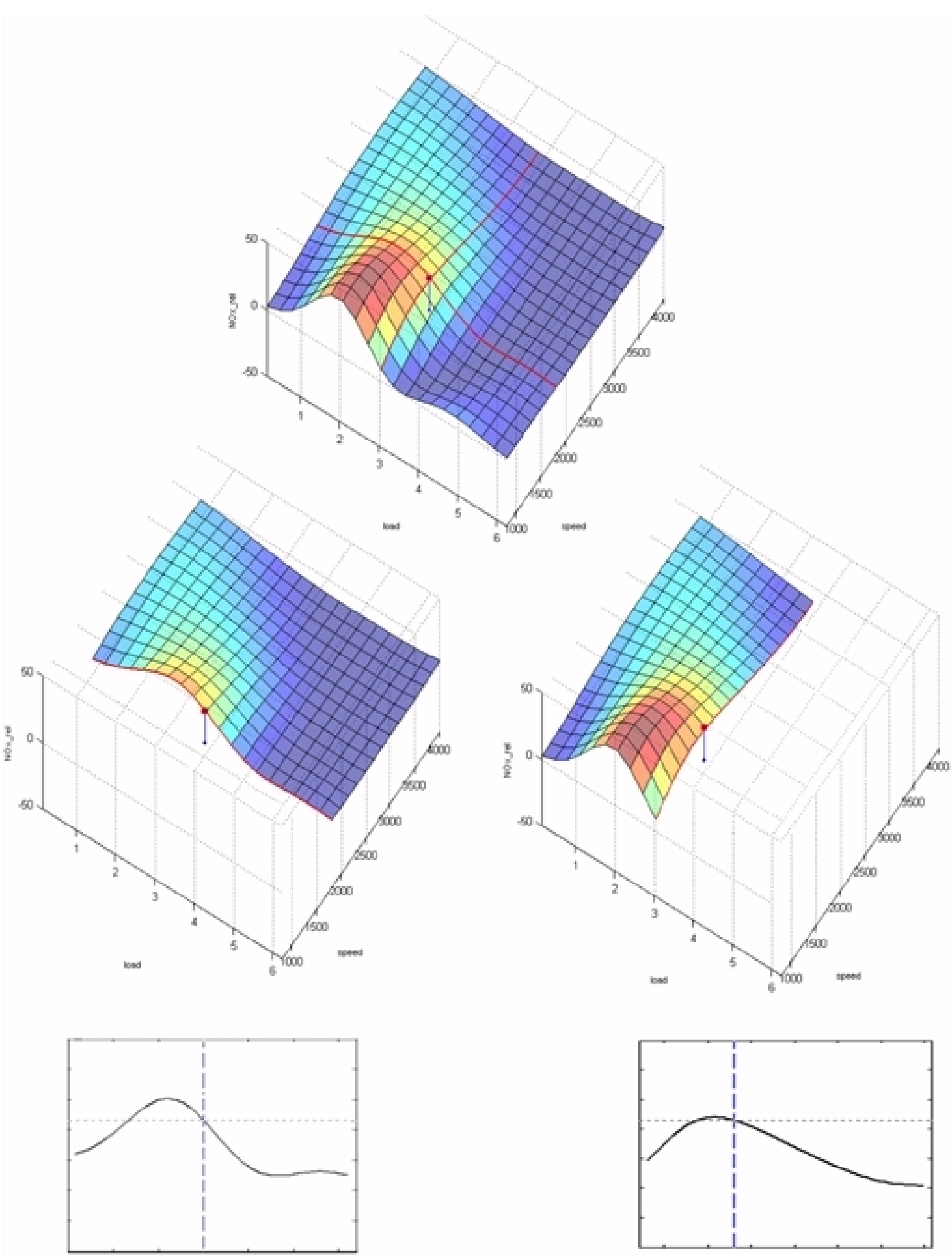


Fig. 5-5: Intersection plot as 2-dimensional intersections in the n+1-dimensional hyperspace (here: n = 2)

The example in the figure shows the model of the dependency of the output `NOx_rel` (relative nitrogen oxide emission) on `speed` and `load`.

The first intersection occurs with a plane of constant speed – the result is the functional dependency of the nitrogen oxide emission on the load (on the left in the figure). All other input parameters (only speed in the example) remain constant.

The second intersection occurs with a plane of constant load – in this case, the intersection plot shows the dependency of emission on speed (on the right in the figure).

6 Tutorial: Working with ASCMO-STATIC

This chapter introduces the basic functions of ASCMO-STATIC by means of an example.

This tutorial is structured as follows:

- 6.1 "Inputs and Outputs of the Measured Engine" on the next page

  This section provides information about the inputs and outputs of the measured engine and the measured data used.
- 6.3 "Before the Model Training" on page 88

  This part of the tutorial describes how to start ASCMO-STATIC and how to evaluate and improve the quality of the training data set. Thus, you can raise the quality of the trained model.
- 6.4 "Model Training" on page 105

  This part of the tutorial describes the direct path from reading the data to the model training
- 6.5 "Model Improvement " on page 107

  This part of the tutorial describes how to assess and improve the trained model.
- 6.6 "Visualizing" on page 111

  The treatment of this section is not absolutely required for the further sequence of the tutorial. However, it is useful to familiarize oneself with the visualization options of ASCMO-STATIC.
- 6.7 "Optimization " on page 120

  In this part of the tutorial, you perform several types of optimization with ASCMO-STATIC. You also learn how to work with the results of a global optimization at several operating points.
- 6.8 "Driving Cycle Forecast " on page 147

  This section provides information about how to use a driving cycle for the definition of the prognosis calculation rules in ASCMO-STATIC.
- 6.9 "Cycle-Based Global Optimization" on page 162

  This section of the tutorial shows how to perform cycle-based global optimization with ASCMO-STATIC, using the model of a diesel engine as an example.
- 6.10 "Model Export " on page 170

  In this section you will learn how to export the ASCMO-STATIC models to MATLAB®, INCA/MDA, Python, Simulink®, Excel, C code, GT-Suite or FMI.

6.1 Inputs and Outputs of the Measured Engine

The tutorial uses direct injection gasoline engine in stratified-charge operation as an example which was measured with respect to the inputs and outputs illustrated in Fig. 6-1 and described in Tab. 6-1.

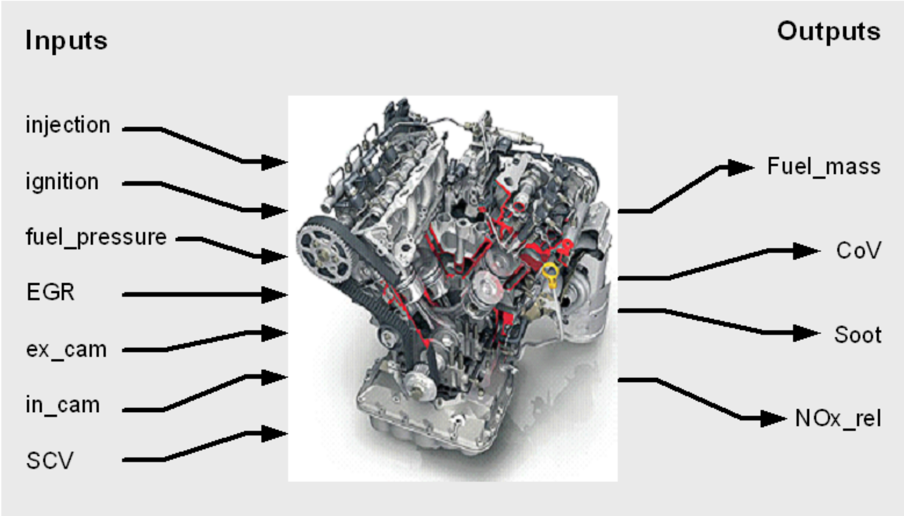


Fig. 6-1: Engine to be measured and modeled

	Name	Meaning
Inputs	`Injection`	End of injection [° KW]
	`Ignition`	Delta "ignition timing after end of injection" [° KW]
	`fuel_pressure`	Fuel pressure [bar]
	`EGR`	EGR rate [%]
	`ex_cam`	Closing of exhaust valves [° KW]
	`in_cam`	Opening of intake valves [° KW]
	`SCV`	Swirl control valve [ ]
Operating points	`speed`	Engine speed [1/min]
	`load`	Load; quantified by mean effective pressure [bar]
Outputs	`fuel_mass`	Fuel mass [kg/h]
	`CoV`	Engine roughness [%]
	`soot`	Smoke number SN [ ]
	`NOx_rel`	relative nitrogen oxide emission [g/kg fuel]

Tab. 6-1: Designation and meaning of inputs and outputs of measured engine

6.2 mail Data for Modeling

The data used in this tutorial are located in the Excel file `Example_GDI_Engine.xls` in the directory `<installation>\Example\AscmoStatic` (By default, *`<installation>`* is `C:\Program\Files\ETAS\ASCMO V5.16\Examples\AscmoStatic`).

These data correspond to the requirements imposed on the training data for modeling in ASCMO-STATIC:

- The engine data were obtained using the DoE method, i.e. the measurements were varied independent of each other and are space-filling.
- The outputs do not have any meaningless values (e.g. values ≤ 0 for consumption or emission, which would render a transformation of the outputs impossible).

6.3 Before the Model Training

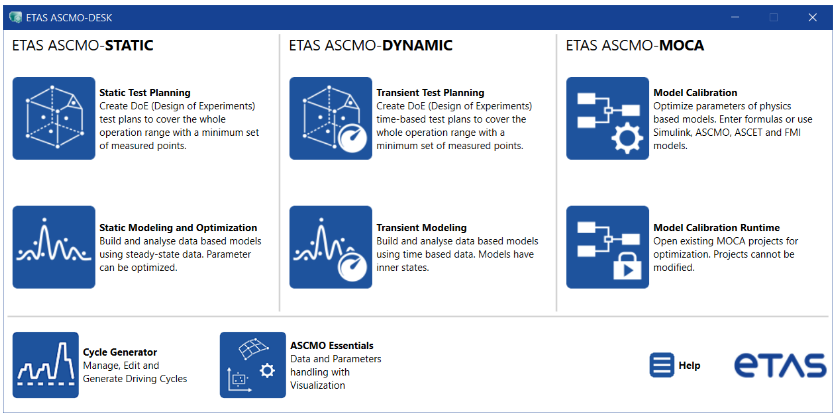
This part of the tutorial describes how to start ASCMO-STATIC and how to evaluate and improve the quality of the training data set. Thus, you can raise the quality of the trained model.

6.3.1 Starting ASCMO-STATIC

After starting ETAS ASCMO, you load either an already existing project or training data for a new project.

1. In the Windows **Start** menu, go to the **ETAS ASCMO V5.16** program group and select **ASCMO Desk V5.16**.

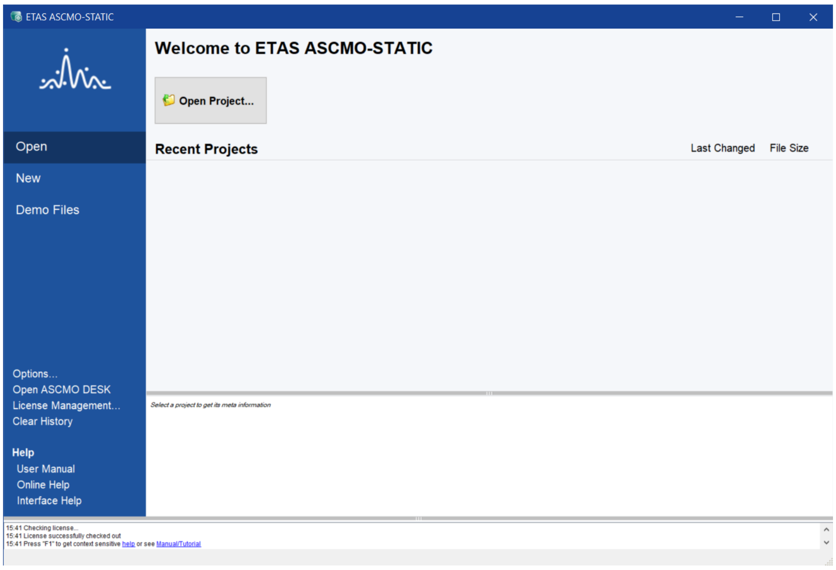
   The **ASCMO-DESK** window opens.

2. In the **ASCMO-DESK** window, click the **Static Modeling and Optim-**

ization tile.

⇨ ASCMO-STATIC opens.



6.3.2 Loading Training Data

If you want to start with a new project, you first have to load the training data required for the model training.

1. Click **New** in the menu panel on the left in the start screen of ASCMO-STATIC.
2. Click 

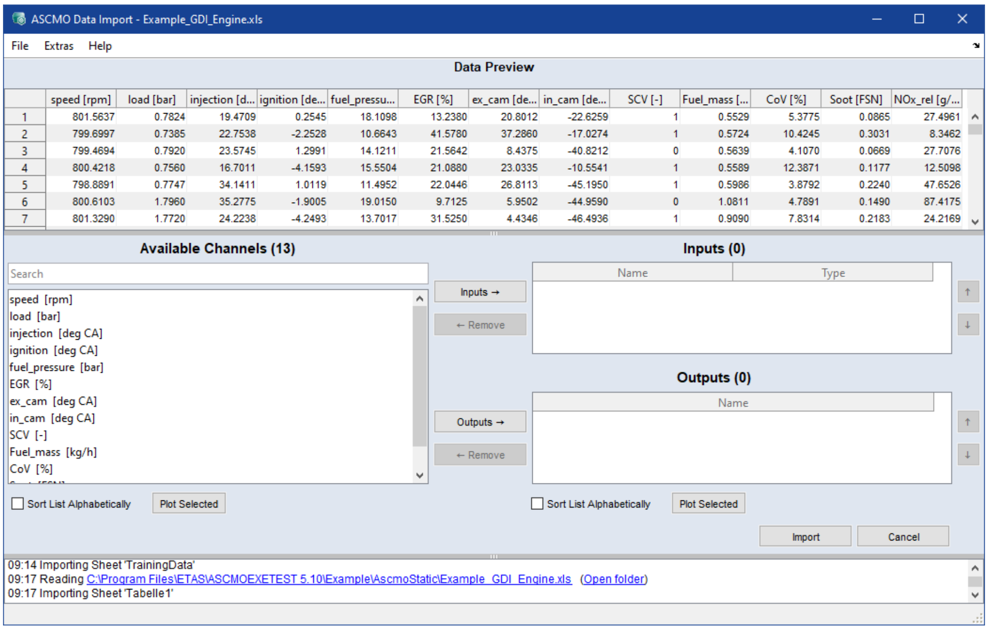
.

   The **ASCMO Data Import** window and a open file dialog opens.
3. In the open file dialog, select the file that contains the training data.
4. If the file contains several work sheets, select the desired work sheet (**Measured Data**).
5. Click **Open**.

⇨ The data are read in.

> **Note**
>
> Depending on the file size, this process may take some time.

After reading in the data, the information from the file is shown in the **ASCMO Data Import** window.



The top part of the window (**Data Preview**) displays the data you imported, the bottom left part (**Available Channels**) displays the names of the columns containing the measured values of the inputs and outputs. They are subsequently assigned to model inputs (**Inputs**) and model outputs (**Outputs**).

6.3.3 Assign Inputs and Outputs

Selecting input variables (Inputs)

To define the input variables of your model, proceed as follows:

1. Select the following measure variables (multiselection is possible):
   - `speed`
   - `load`
   - `injection`
   - `ignition`
   - `fuel_pressure`
   - `EGR`
   - `ex_cam`
   - `in_cam`
   - `SCV`
2. Click Inputs →

⇨ The variables are added to **Inputs**.

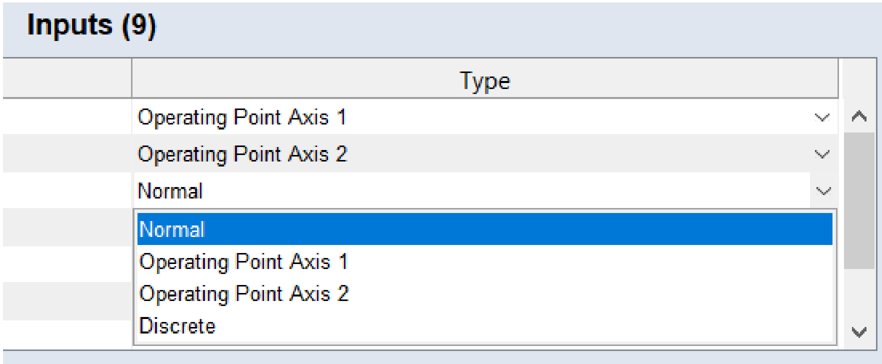
Selecting operating points (for global models)

For global models, the corresponding inputs must be declared as operating points.

1. In the **Type** column for the inputs, select the following types:

Name	Type
speed [rpm]	`Operating Point Axis 1`
load [bar]	`Operating Point Axis 2`
SCV	`Discrete`

The other inputs use the type `Normal`.

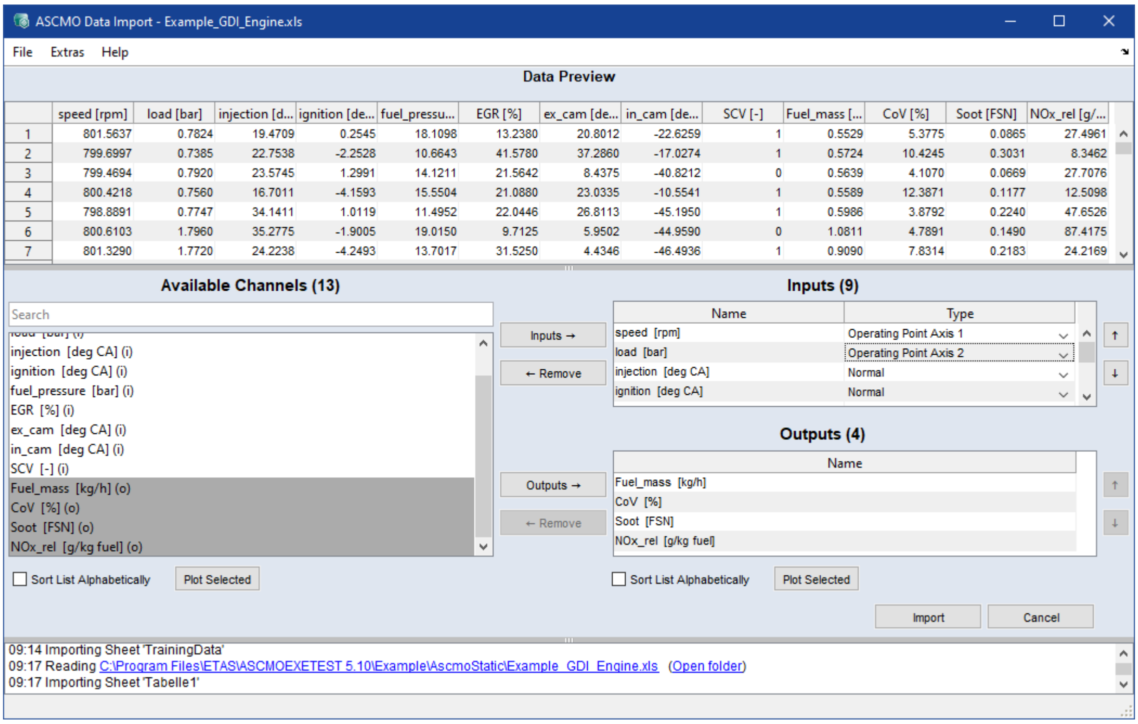


Selecting output variables (Outputs)

Finally, define the output variables to be modeled:

1. Select the following measured variables:
   - `Fuel_mass`
   - `CoV`
   - `Soot`
   - `NOx_rel`
2. Click Outputs → .

➪ The variables are added to **Outputs**. The assignment of the training data is now complete.

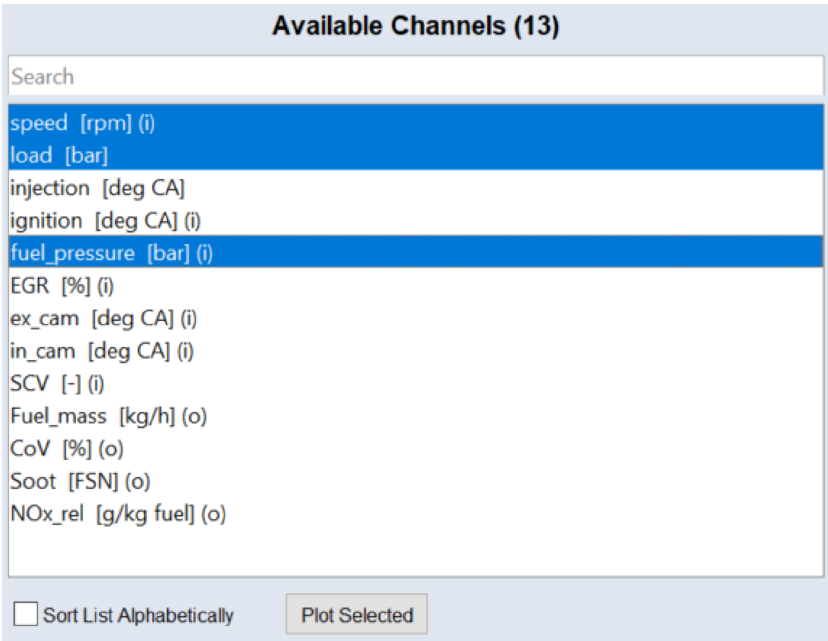


6.3.4 Graphical Plausibility Check

To check the training data again prior to the import, it is possible to graphically display the measuring data.

Displaying measuring data prior to the import

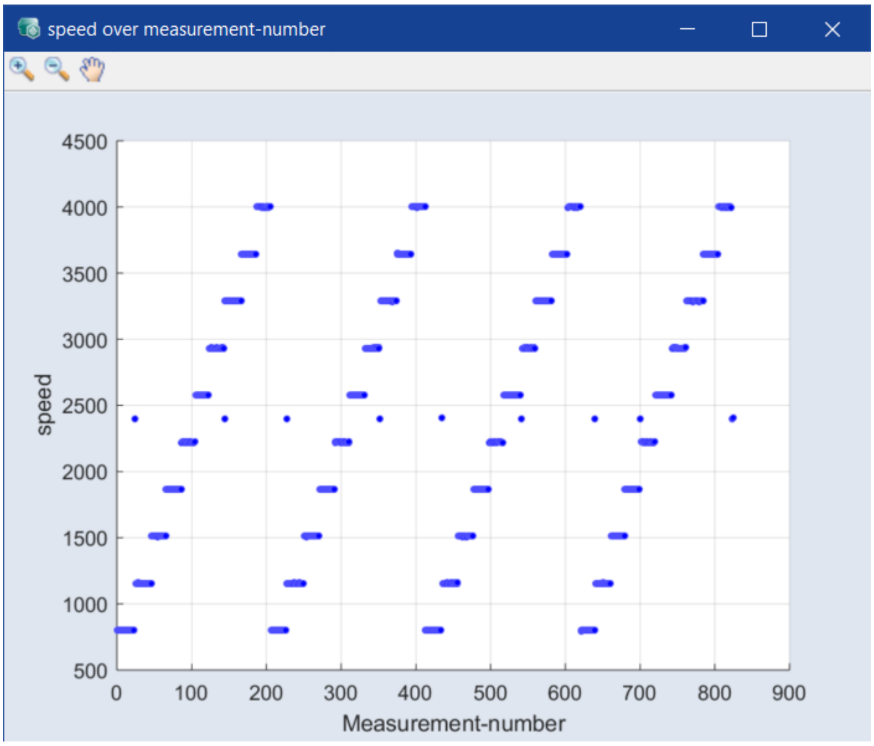
1. In the **Available Channels** field, select one or more measure variables.



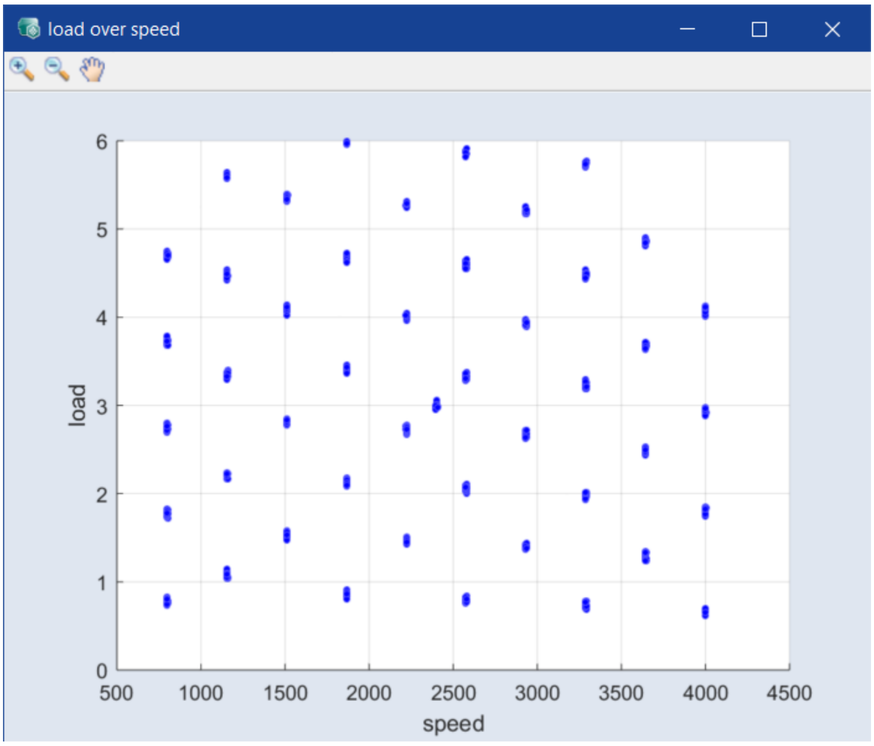
2. Click **Plot Selected** or select **Extras › Plot Selected**.

   A window opens that displays the following (depending on the number of selected variables):

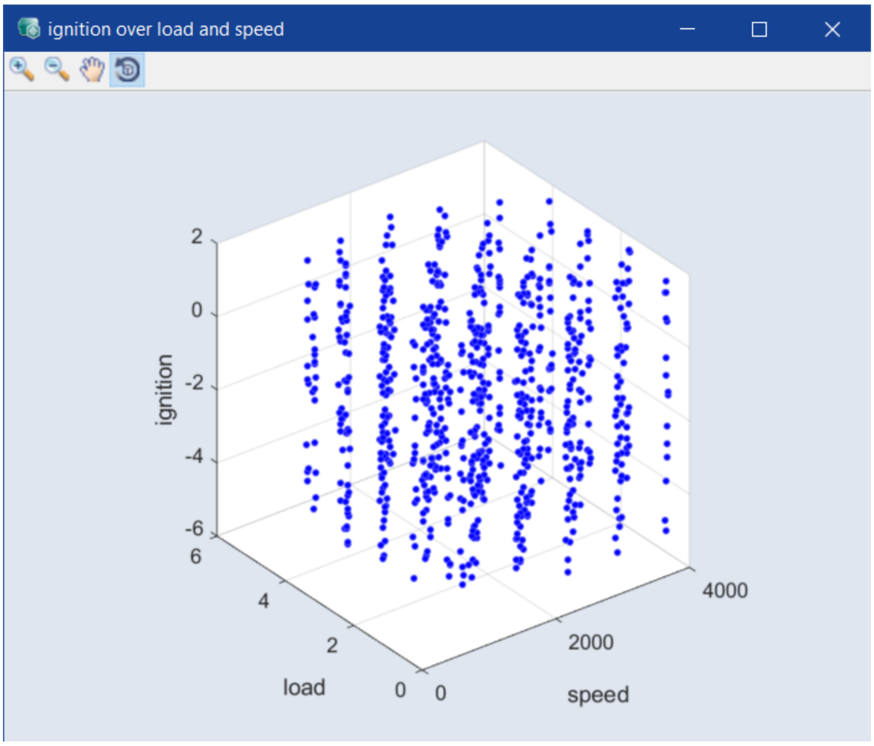
A. *1 variable:* measured data above number of measurement



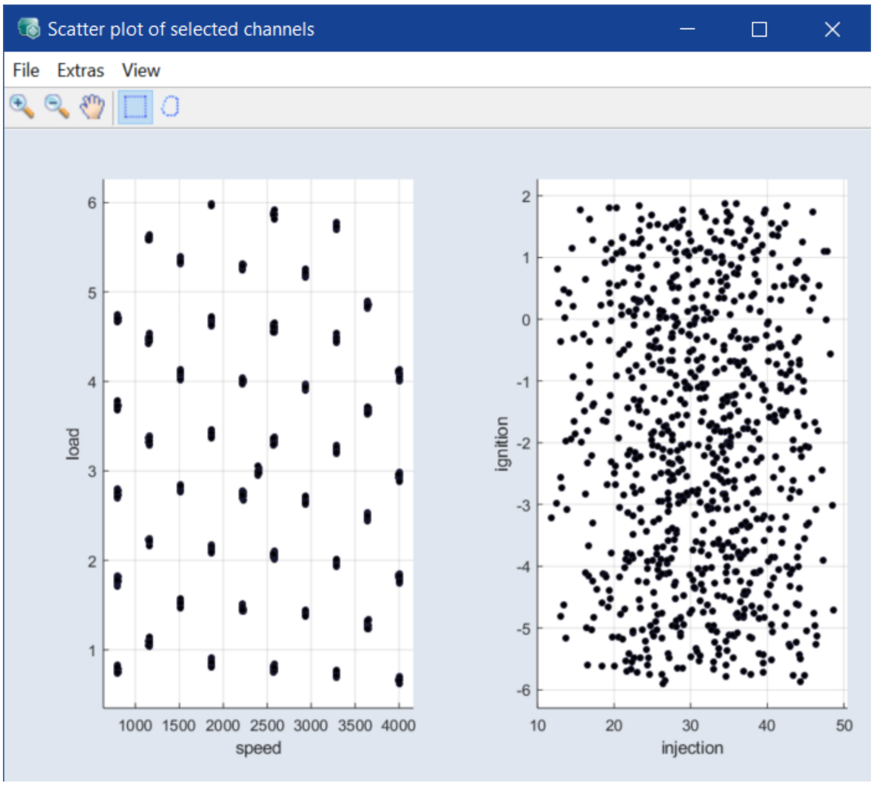
B. *2 variables:* data of one column above data of the other – e.g., load above speed.



C. *3 variables:* data of the third column above the plane set up by the other two – e.g., `ignition` above the `speed-load` plane.



D. *more than 3 variables:* a series of scatter plots



This window can be used, for example, to visualize the equal proportioning of the measuring data.

> **Note**
>
> Detailed instructions for handling such 3D plots are located in section 6.6 "Visualizing" on page 111.

6.3.5 Save and Load a Configuration

A configuration file (`*.ini`) contains the currently selected assignment of individual columns of the training data for the groups **Inputs** (incl. **Type**) and **Outputs**.

Saving and loading a configuration

1. Select **File › Save Channel Config (*.ini)**.
2. In the file selection window, enter path and name of the target file, then click **Save**.
3. To load a previously saved configuration, select **File › Load Channel Config (*.ini, *.lab)**.

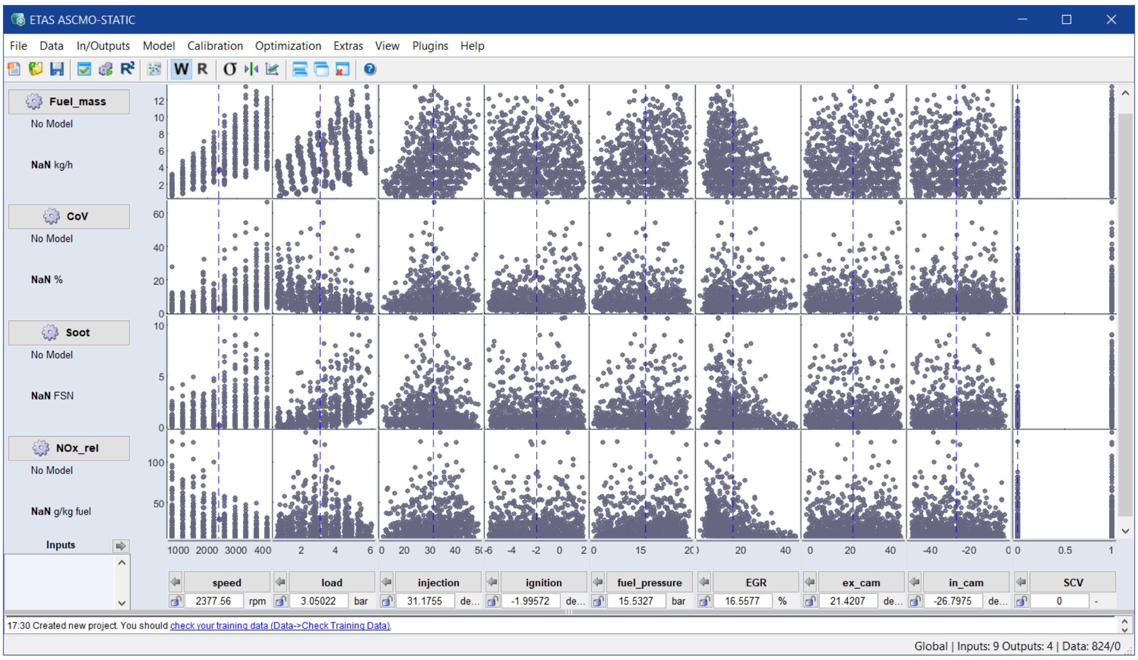
6.3.6 Import of Measurement Data

Now the so-prepared data for model training can be imported.

Importing measuring data

1. In the **ASCMO Data Import** window, click **Import**.

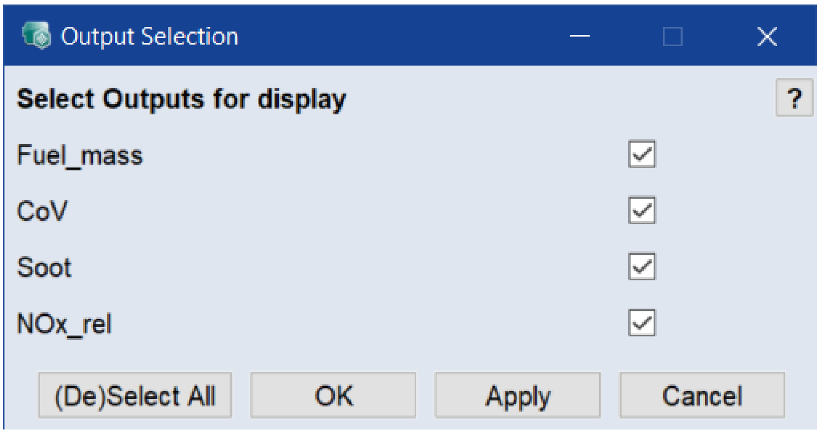
➩ The measuring data are imported and the project is created. The measured data is displayed in the ISP (**I**nter**s**ection **P**lot; see 5.3 "Intersection Plots " on page 84 for details) view.



A detailed description of the user interface is given in 5.1 "User Interface of ASCMO-STATIC" on page 80. To familiarize yourself with the operation using the ISP view, the tutorial contains 6.6.1 "Intersection Plot (ISP)" on page 111.

Determining output variables for modeling

1. Select **In/Outputs › Select Outputs**.



2. Select all outputs and click **OK**.

It is recommended that you save the project before you continue.

Saving the project

1. Select **File › Save** or save the project with a user-defined path and name with **File › Save As**.

6.3.7 Review and Edit the Training Data Set

Even though the actual modeling (see 6.4 "Model Training" on page 105) is straightforward and does not require any data preprocessing or user experience, an analysis of the loaded data set is recommended.

> **Note**
>
> If you want to skip review/editing the training data set, you can directly start the model training (see 6.4 "Model Training" on page 105).

The **Data** menu provides a set of tools that help increase the model quality by checking and modifying the loaded data set.

The measuring data that are not used for the training are selected at random and identified as test data. After selecting the desired quantity, all models are retrained with the reduced training dataset. The number of defined training and test data is also displayed in the main window at the bottom right.

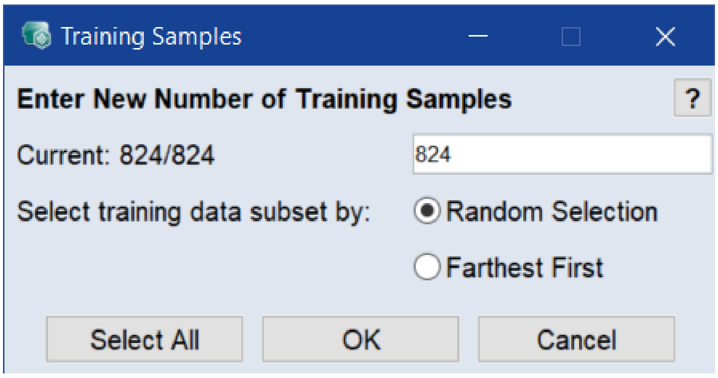
Set Number of Training Samples

At the import of the measurement data, all measuring points are used as training data – this maximum number can be reduced here. A sufficiently large number of measuring points should naturally remain for a successful model training.

Reduce number of training samples

1. Select **Data › Set Number Training Samples**.

   The **Training Samples** window opens. The current number of training samples is displayed.

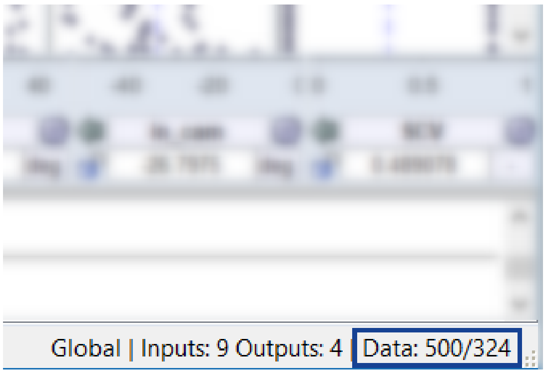
   

2. Change the number of training samples to 500.
3. Activate the **Farthest First** option.

   This activates the Farthest First algorithm that selects a space-filling subset of measurement points. See the online help for more information.

4. Click **OK**.

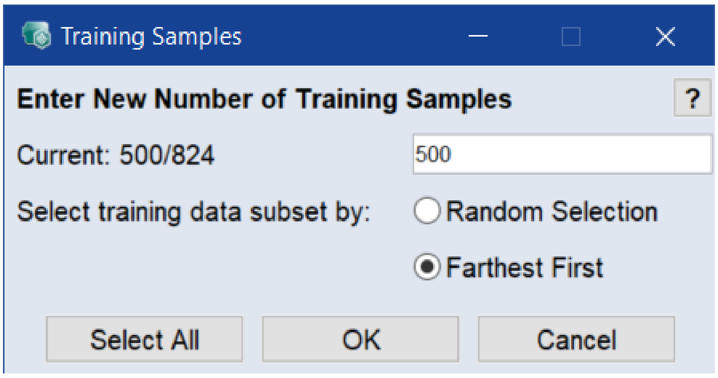
⇨ The **Training Samples** window closes. The number of measurement points is updated in the status bar.



Maximize number of training samples

1. Re-open the **Training Samples** window.

   500 of 824 data samples are used as training data. The model can be tested with 324 data samples.



2. Click **Select All**.

   In the **Training Samples** window, the number of training samples is updated to the maximum number of possible training samples.

3. Click **OK**.

➪ The **Training Samples** window closes. The number of measurement points is updated in the ISP view.

Note

If you have started the model training previously (see 6.4 "Model Training" on page 105), the model will be retrained automatically after you have edited and confirmed the new number of training samples.

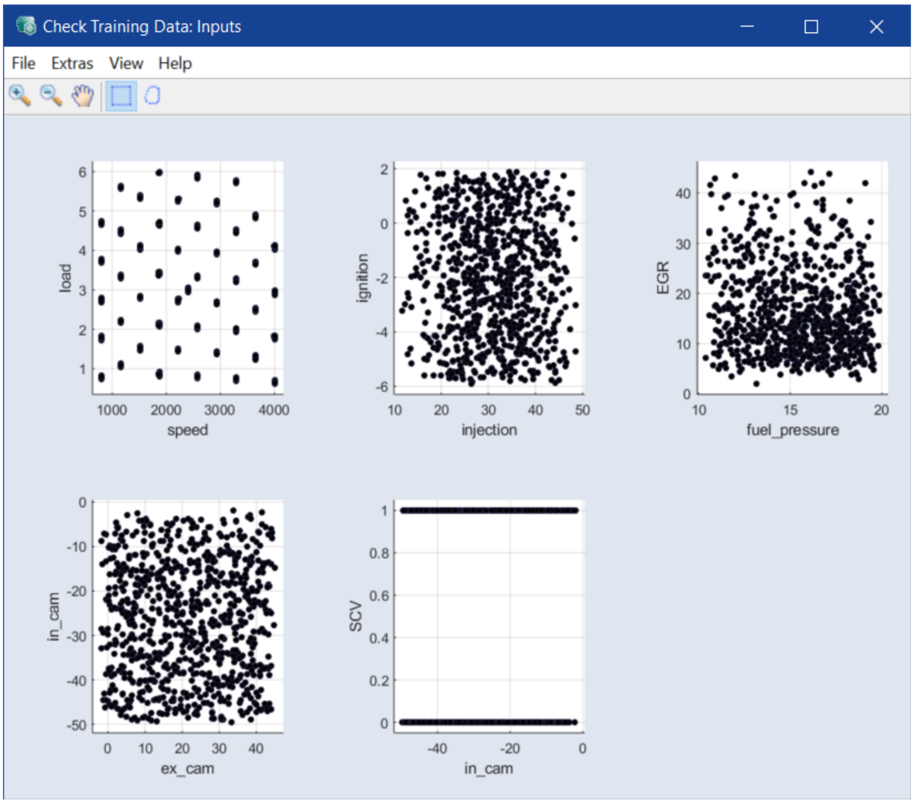
Check Training Data

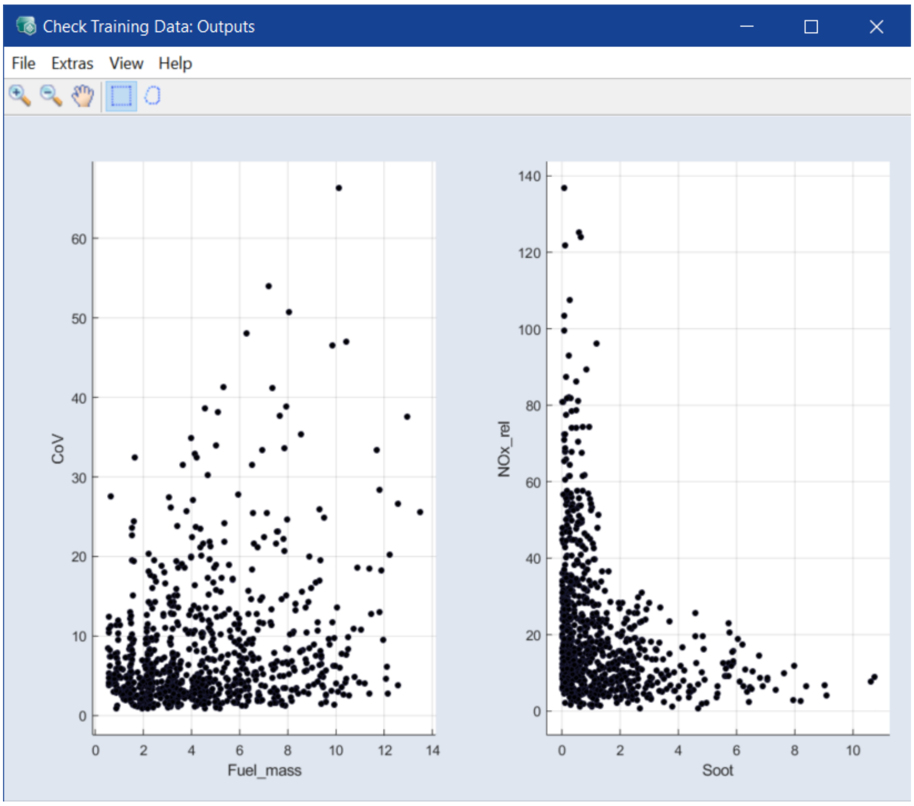
The **Data › Check Training Data** menu option allows checking whether the training data used are distributed in a space-filling way. The windows **Check Training Data: Inputs** and **Check Training Data: Outputs** are opened. In both windows, the measured values of certain inputs/outputs are plotted over other inputs/outputs.

Opening the **Check Training Data** windows

1. Select **Data › Check Training Data**.

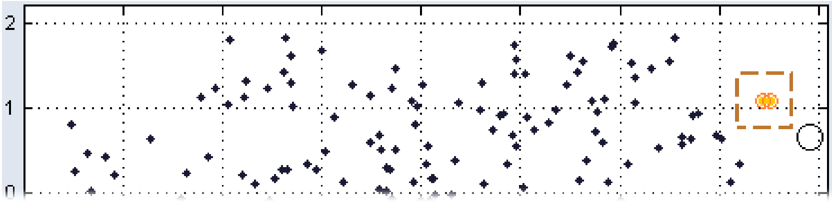
➪ The **Check Training Data: Inputs** and the **Check Training Data: Outputs** windows open.





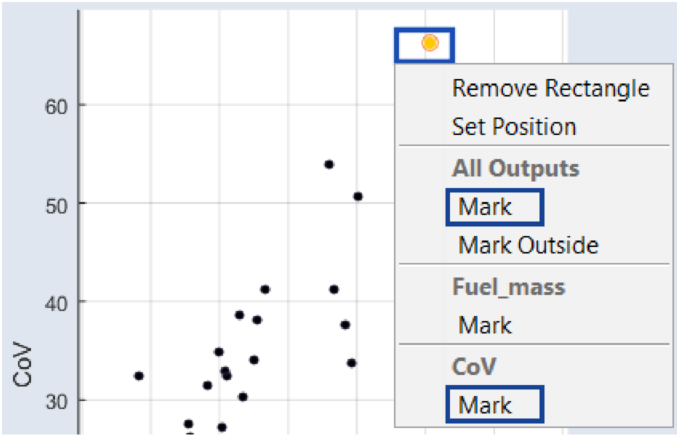
<u>Deleting measurement points within the plot</u>

1. In the plots of the **Check Training Data: *** windows, search for outliers.
2. If you found one or more outliers, click the button, then click in the respective plot and draw a rectangle around the points.



The selected points are highlighted in color in all plots.

3. To mark the points within a specific rectangle as outliers, right-click the edge of that rectangle and select **Mark** from the context menu.



Mark in the **All Outputs** area marks the outliers for all outputs.
Mark in an **<output_name>** area marks the points in the rectangle only for the respective output.
Mark Outside marks all points outside the rectangle.

4. Use **Remove Rectangle** to remove an existing rectangle.

   Marked points remain marked.

5. Select **Extras > Delete Marked Points / Retrain** to delete the marked points and retrain the model.

Repetition Point Analysis

With the *Repetition Point Analysis*, you can display the determined groups of repetition points based on a defined tolerance value (Current Tolerance Level) within a series of measurements.

On the one hand, these measurements can be used to determine the experiment reproducibility (RMSE) – no model can be more precise without improving the measurement technique.

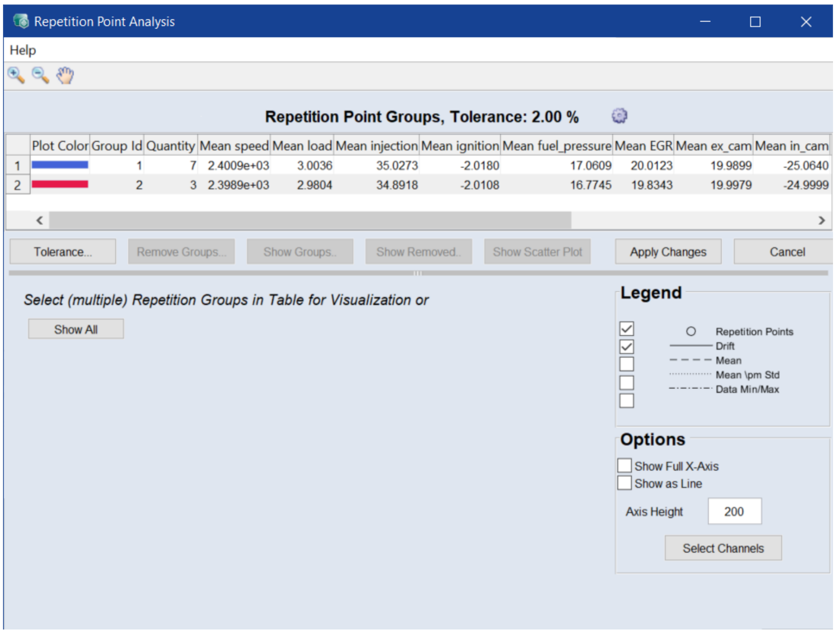
On the other hand, it allows identifying a disturbance variable-based drift which can be identified by means of the measuring time or measuring point number. In that case, drift effects could be corrected by inserting the disturbance variable in the model. But a requirement for the disturbance variable correction is that the disturbance variable does not feature any correlation to other model parameters (no "sorted" parameters).

Opening the **Repetition Point Analysis** window

1. Select **Data › Repetition Point Analysis**.

➪ The **Repetition Point Analysis** window opens. All groups of repetition points are displayed; their number depends on the current tolerance level.

Each group is named by an identifier (**Repetition Point Group ID** column). The **Quantity** column contains the number of points in this group, and the subsequent columns contain the mean of the points.

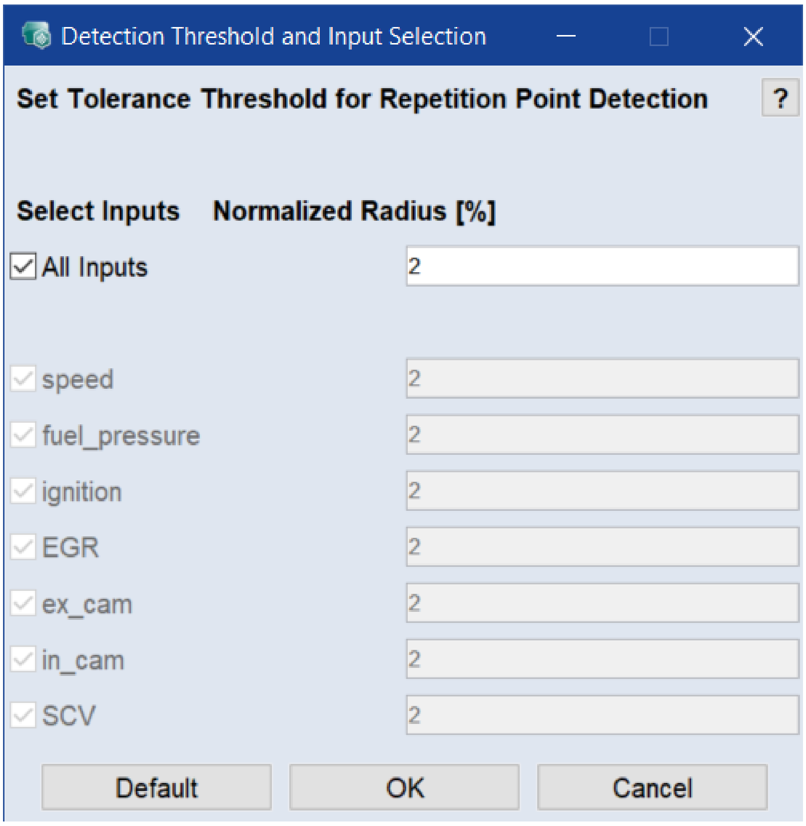


offers different options for editing and displaying repetition points, which are described below.

Selecting the Inputs for the repetition point analysis

1. In the **Repetition Point Analysis** window, select .

The **Detection Threshold and Input Selection** window opens. All inputs are used for the repetition point analysis.



Basically, it is recommended to enable all inputs for the evaluation, because all repetition points can be identified and removed if necessary.

2. Do one of the following:
   - Select the desired inputs for the evaluation.
   - Activate/deactivate **All Inputs** to select/deselect all inputs for the evaluation.
3. Click **OK**.

⇨ The group of repetition points is updated in the **Repetition Point Analysis** window.

Setting the tolerance threshold

The tolerance threshold is the basis for determining the repetition points. A high value results in a higher number of identified repetition points.

1. In the **Repetition Point Analysis** window, select .

   The **Detection Threshold and Input Selection** window opens.
2. Increase the tolerance threshold in the **Normalized Radius in Percent** field.
3. Click **OK**.

   The group of repetition points is updated in the **Repetition Point Analysis** window.

New repetition point groups will appear if the distance of the group members falls below the threshold. Existing groups with a distance above the old threshold and below the new threshold are merged.

4. Set the tolerance threshold back to 2.

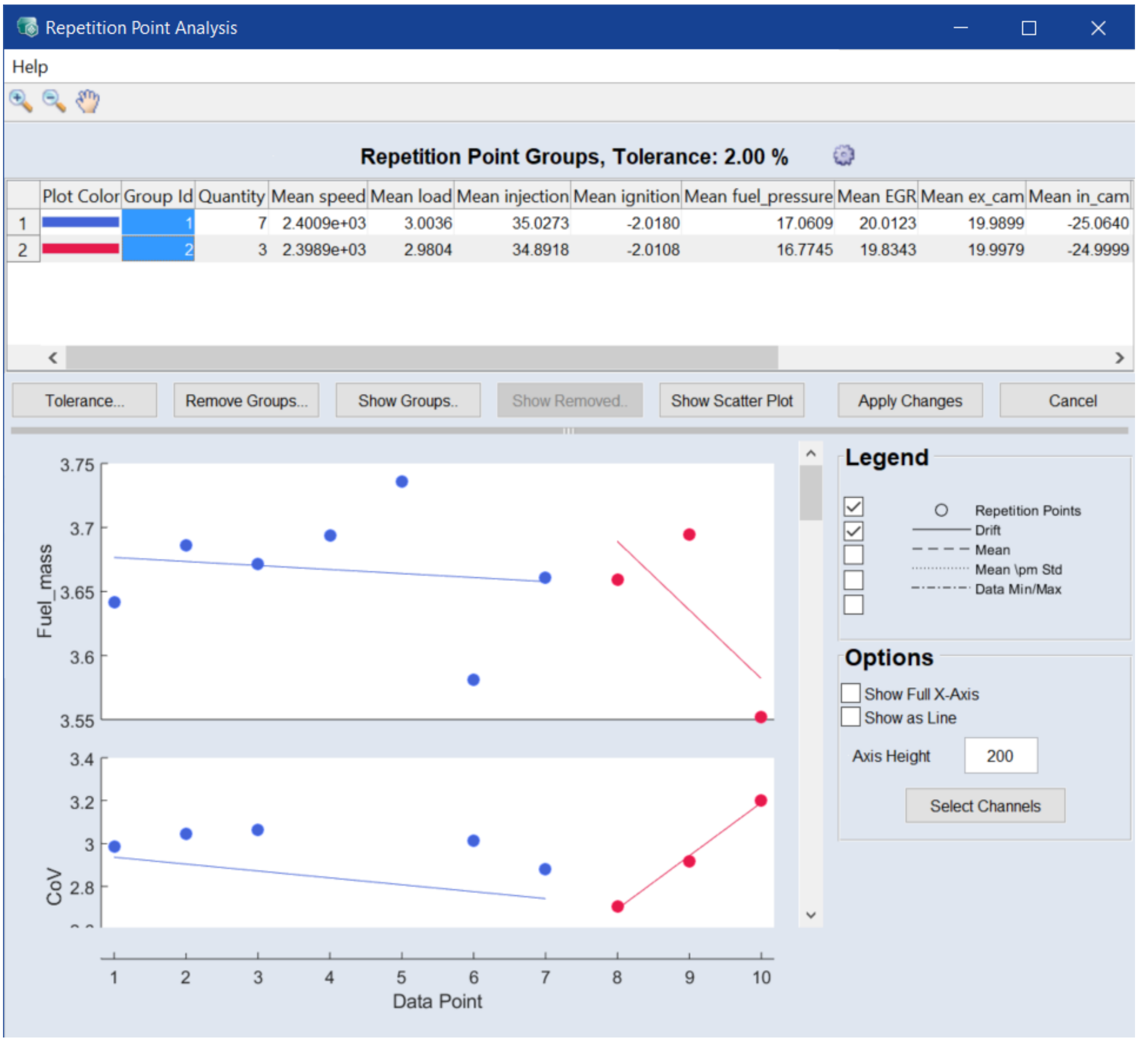
Displaying the drift of the measurement values

You can view details of the measured outputs for each repetition point in the **Repetition Point Analysis** window. Only the *Root Mean Square Error* is shown.

> **Note**
>
> For more information about the drift of the measured data, see 4.2.2 "Disturbance Variables, Drift and Experiment Repeatability" on page 30.

1. In the **Repetition Point Analysis** window, select a group of repetition points by clicking any cell in the desired row.
2. Click the **Drift** checkbox in **Legend**.

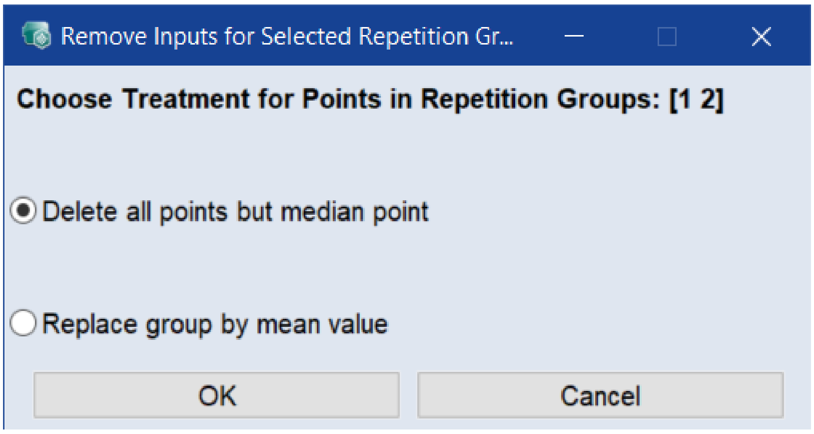
⇨ The Drift is visualized in lines.



Deleting repetition points

1. In the **Repetition Point Analysis** window, select a group of repetition points. Multiple selection is possible.
2. Click **Remove Groups**.

   The **Remove Inputs for the Selected Repetition Groups** window opens. Here you can specify the method on how to replace the determined set of points.

   Delete all points but median point: The individual points are replaced with a median point in the list.

   > **Note**
   >
   > If the number of repetition points in a group is even, the point whose index is closest to the mean index is selected.

   Replace group by mean value: The individual points are replaced with the mean value of the points to be removed.
3. Activate the desired method and click **OK**.

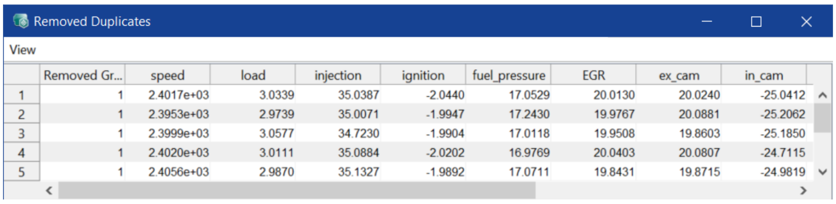
➪ The selected group of repetition points is deleted. The **Remove Inputs for Selected Repetition Groups** window closes.

Showing list of removed repetition points

You can view the removed points later in the **Removed Duplicates** window.

1. In the **Repetition Point Analysis** window, select **Show Removed**.

➪ The **Removed Duplicates** window opens.



	Removed Gr...	speed	load	injection	ignition	fuel_pressure	EGR	ex_cam	in_cam
1	1	2.4017e+03	3.0339	35.0387	-2.0440	17.0529	20.0130	20.0240	-25.0412
2	1	2.3953e+03	2.9739	35.0071	-1.9947	17.2430	19.9767	20.0881	-25.2062
3	1	2.3999e+03	3.0577	34.7230	-1.9904	17.0118	19.9508	19.8603	-25.1850
4	1	2.4020e+03	3.0111	35.0884	-2.0202	16.9769	20.0403	20.0807	-24.7115
5	1	2.4056e+03	2.9870	35.1327	-1.9892	17.0711	19.8431	19.8715	-24.9819

The value of each individual removed data point is provided with an ID (**Removed Group ID** column) and displayed.

6.4 Model Training

This part of the tutorial describes the direct path from reading the data to the model training.

Note

If you want to skip the model training, you can load the project with the trained models (*<installation>\Example\AscmoStatic\Example_GDI_Engine.ascmo*) and continue with 6.5 "Model Improvement " on page 107.

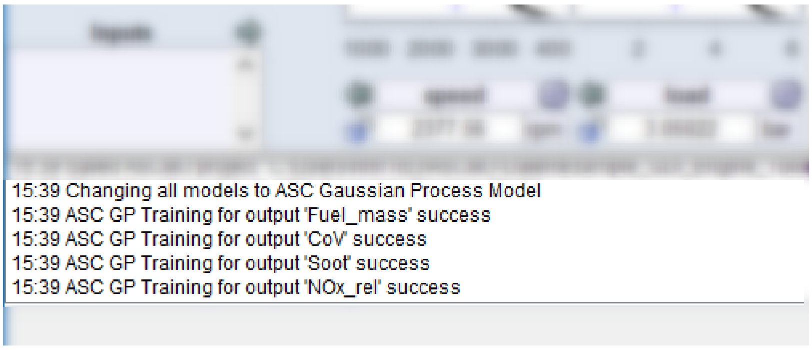
6.4.1 Start Model Training

After these preparations, the model training can be performed.

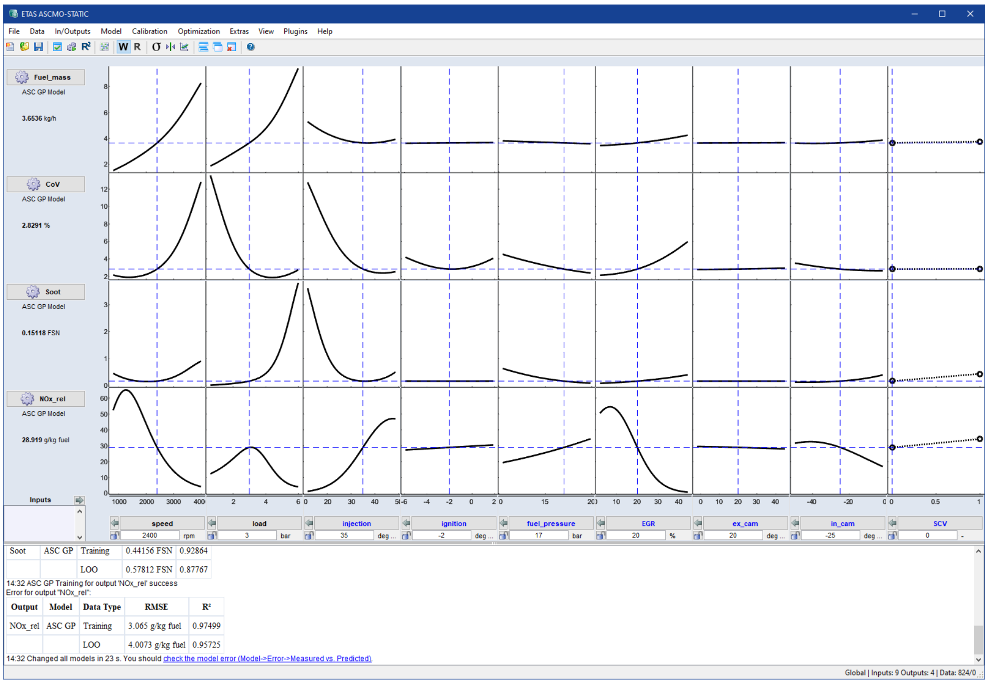
Performing the model training

1. Select **Model > Start ASC GP Model Training**.

   The modeling is started; it may take some time. Information about the progress can be seen in the log window.

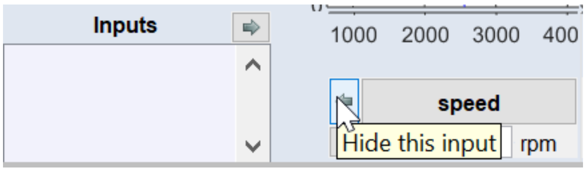
   

   After performing the model training, the intersection plots are updated. The log window displays the RMSE and R^2 for each output. This gives an indication of the quality of the model.



> **Note**
>
> The scaling of the display can be arranged more clearly by deactivating the display of the two variables `speed` and `load` since they exercise by far the greatest influence, e.g. on consumption and emission variables.

2. To do so, click the arrows next to the inputs `speed` and `load`.



➩ It is recommended that you save the project again.

6.4.2 Model Training Summary

This concludes the model training – based on the training data, the dependency of all `m` outputs on `n` inputs was modeled.

The models for the output variables can now be used to find the optimal values of the inputs with respect to optimized output variables such as consumption and emissions.

Before you apply the different optimization methods (see 6.7 "Optimization " on page 120), you should first familiarize yourself with the model assessment and improvement, described in the next section.

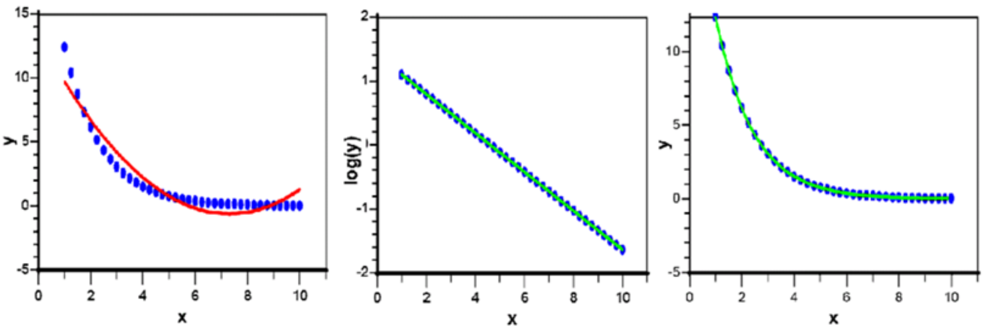
6.5 Model Improvement

This part of the tutorial describes how to assess and improve the trained model.

> **Note**
>
> If you want to skip the model assessment and improvement, you can continue with 6.6 "Visualizing" on page 111 or 6.7 "Optimization " on page 120.

6.5.1 Model Improvement Through Transformation of Output Variables

A further improvement of the model can be achieved with a transformation (also referred to as Box-Cox transformation) of the output variables. In this case, functions such as square root, inversion, and logarithm are applied.



Polynomial of degree 2 (not sufficient) | After transformation | After retransformation: Good description

This allows mapping a variety of engine-based behaviors that do not behave in a linear or squared way over the input variables.

To determine the optimal transformation, all transformations are applied to the output, and the one resulting in the lowest RMSE is selected. In practice, there is already ample experience concerning the application of a specific transformation for certain output variables.

6.5.2 Model Improvement Through Recognition and Deletion of Outliers

Measuring points whose model errors (i.e. the deviation of the measured value from the model prediction) are high are referred to as *outliers*.

Besides the visual evaluation in the corresponding plots (see Fig. 6-2), the term can also be interpreted quantitatively: An outlier exists if the residual is › 3-4 x RMSE.

The visual assessment can take place in **Measured vs. Predicted** displays (such as in the plots opened with **Model › Error (‹method›) › Measured vs. Predicted**, ‹method› = **Leave-One-Out**, **Test Data** or **Training Data**).

The following figure shows the measured data of an engine over the model prediction. The marked points have been identified as outliers by ASCMO-STATIC.

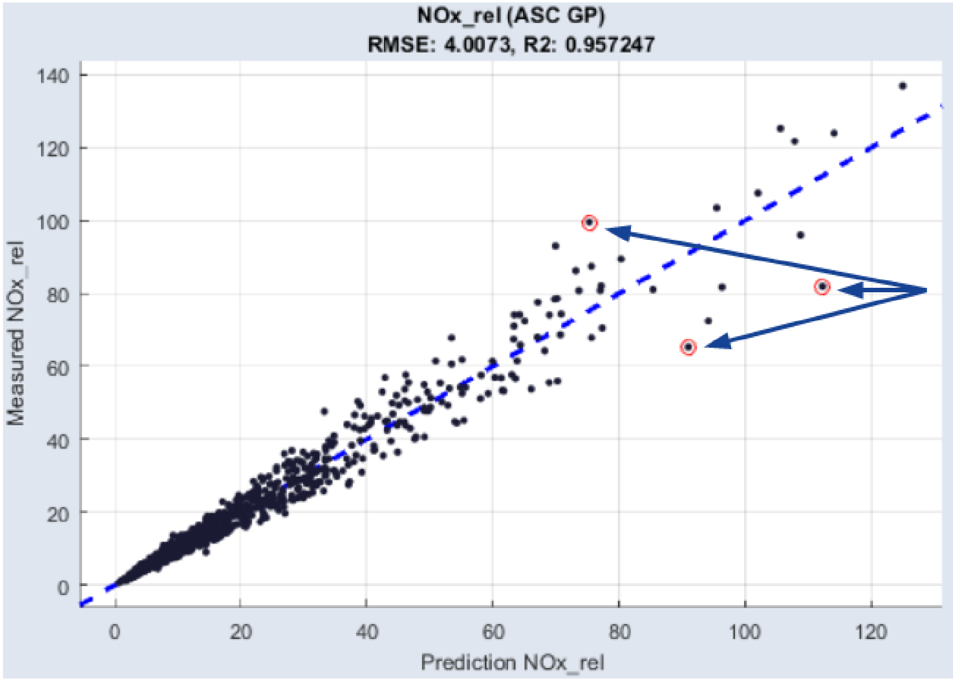


Fig. 6-2: Outliers in the **Measured vs. Predicted** display

The reasons for the occurrence of such deviations can be simple errors in the measurement (e.g. due to defective measuring equipment). Another possibility is that the measurement took place in the limit range of the engine, and hence could not be mapped by the model.

It can easily be seen that such measuring points have a negative effect on the model. While the green graph in Fig. 6-3 results from the modeling based on the blue points, the model training including the red outlier results in a graph (red) that features significant deviations to the measured data.

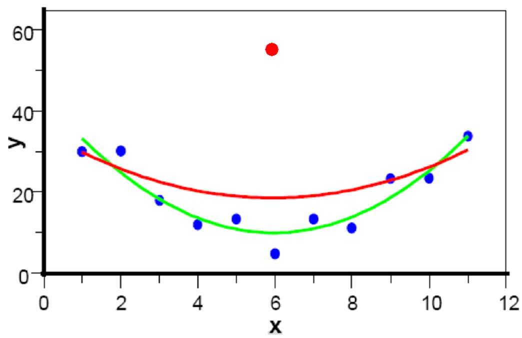


Fig. 6-3: Modeling with (red) and without (green) outlier

The following section contains instructions about how to recognize and remove outliers.

Recognizing outliers

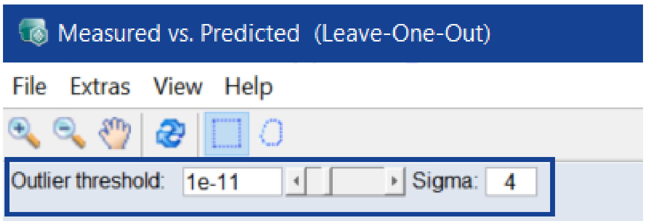
1. Select **Model › Error (Leave-One-Out) › Measured vs. Predicted**.

   The measure values and the model prediction of the outputs are displayed.

2. Select **Extras › Set Outlier Threshold**.

   The menu entry is only visible when the Advanced Settings are enabled. Enable these in the menu of the main window with **File › Options**, if necessary.

   A slider (**Outlier Threshold:**) and an input field (**Sigma**) are displayed.

3. Leave **Sigma** at 4 and increase the significance level to higher values.

   The higher the significance level, the larger the number of outliers (indicated by red circles).

   The outlier becomes even clearer if you allow the absolute or relative error to be displayed, or display the data in a normal probability plot.

4. To do so, do one of the following:

   - Select **Model › Error (Leave-One-Out) › Error vs. Output**
   - Select **Model › Error (Leave-One-Out) › Probability Plot**

⇨ Depending on the menu option you selected, one of the following windows opens.

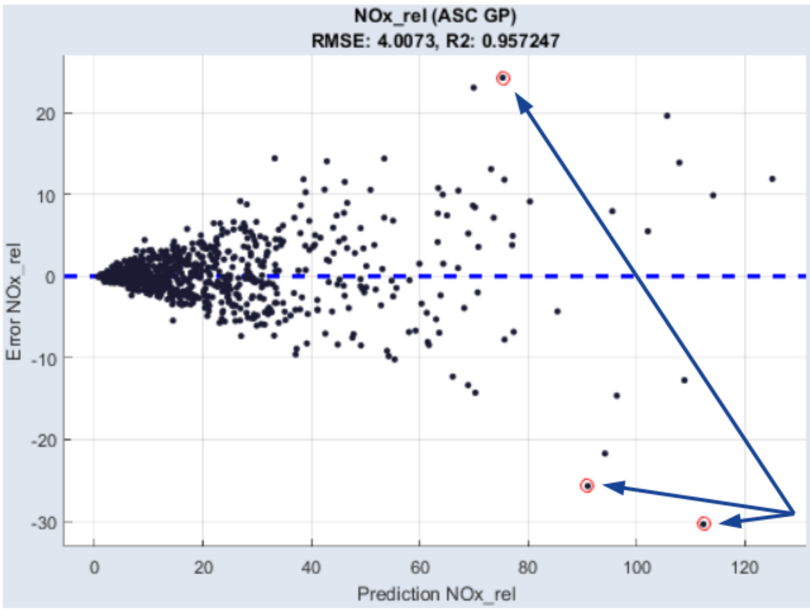


Fig. 6-4: Absolute error versus model prediction

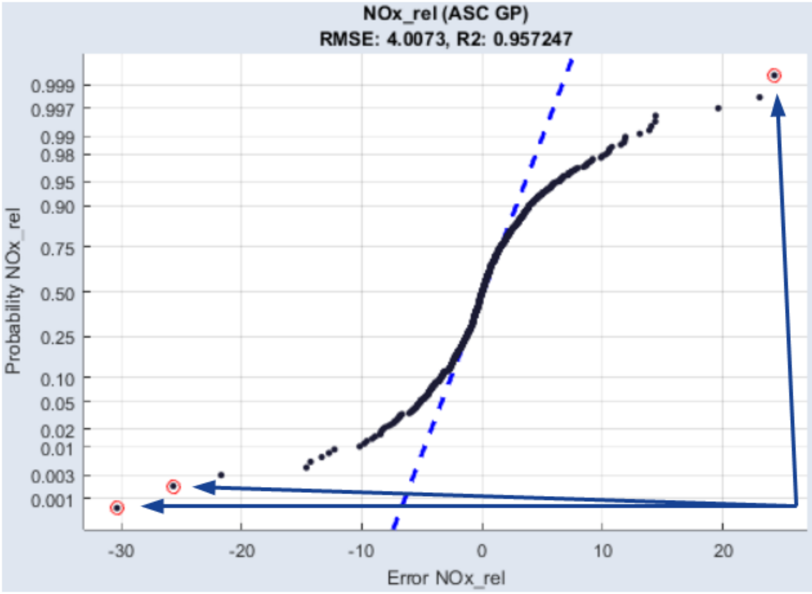
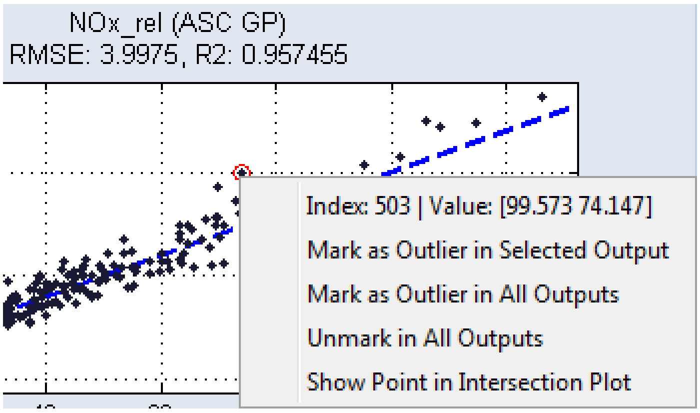


Fig. 6-5: Normal Probability Plot

The fact that it is the same measuring point can be verified by right-clicking on the respective point.



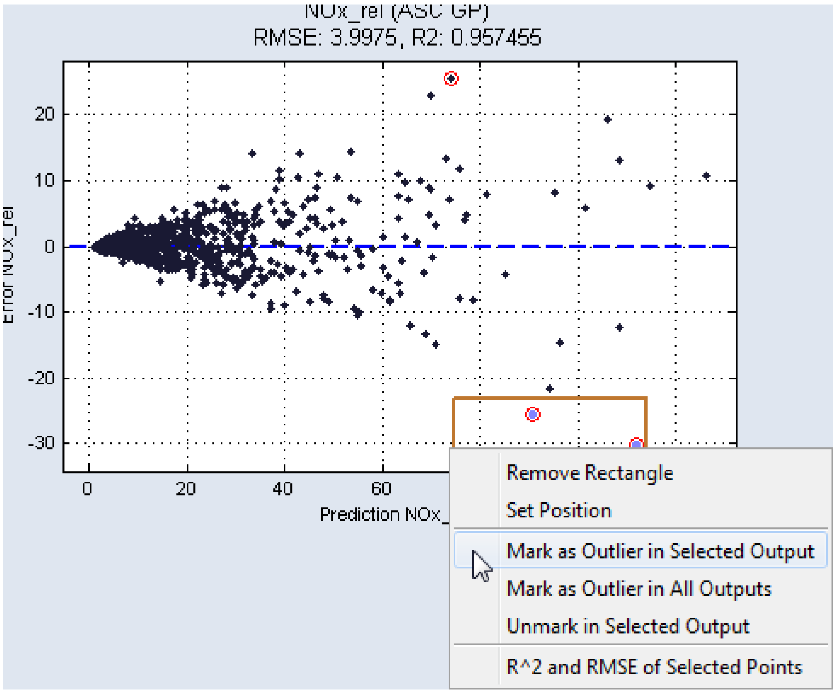
Removing outliers

To remove individual or groups of outliers, proceed as follows.

1. In one of the plots that is opened with **Model › Error (Leave-One-Out) › ***, select one or several points by holding the left mouse button pressed and drawing up a rectangle over these points.

   The selected points are highlighted in color.

2. If necessary, select additional points by drawing up another rectangle.
3. To mark the points within a specific rectangle as outliers, right-click the edge of the particular rectangle and select **Mark as Outlier in Selected/All Outputs**.



You can undo this by selecting **Unmark in Selected Output** from the context menu.

4. To remove the selected measuring points from the set of training data, select **Extras › Delete Marked Points and Retrain**.

⇨ After removing the data, the models are retrained for the outputs.

6.6 Visualizing

The treatment of this section is not absolutely required for the further sequence of the tutorial. However, it is useful to familiarize oneself with the visualization options of ASCMO-STATIC.

6.6.1 Intersection Plot (ISP)

The modeled dependencies of the output variables on the inputs are visualized using **intersection plots**. General information about the intersection plots is given in 5.3 "Intersection Plots " on page 84.

Loading the project

If you saved a previously processed ASCMO-STATIC project, you can continue working with it. To do so, proceed as follows.

1. Do one of the following:
   - In the ASCMO-STATIC start window: click **Open ASCMO Project**. (cf. "Starting ASCMO-STATIC " on page 88)
   - In ASCMO-STATIC user interface: click **File › Open**. (cf. "Elements of the ASCMO-STATIC User Interface" on page 80)

2. In the file selection window, select the project file (`*.ascmo`) and click **Open**.

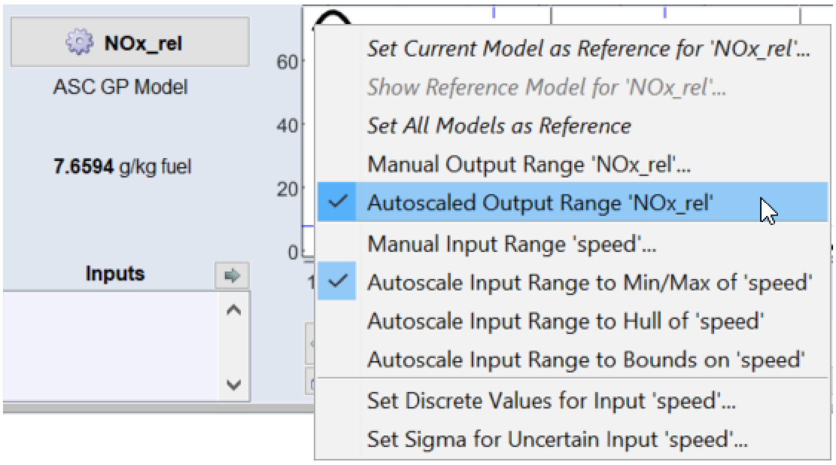
➪ The project opens.

<u>Scaling axes</u>

In principle, the scaling for the Y axes (= outputs) is determined automatically. However, this scaling can also be changed manually.

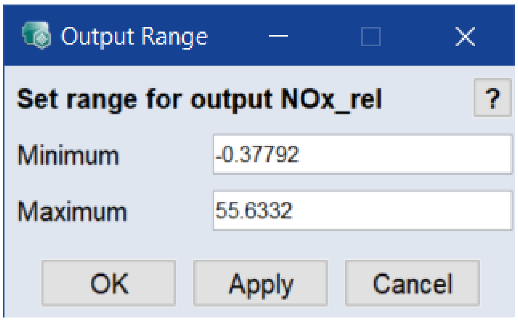
1. Right-click in a plot.

   A context menu opens. The **Autoscaled Output Range ‹output›** option is enabled by default.



2. Select the option **Manual Output Range ‹output›**.

➪ A window for entering the range to be displayed opens.

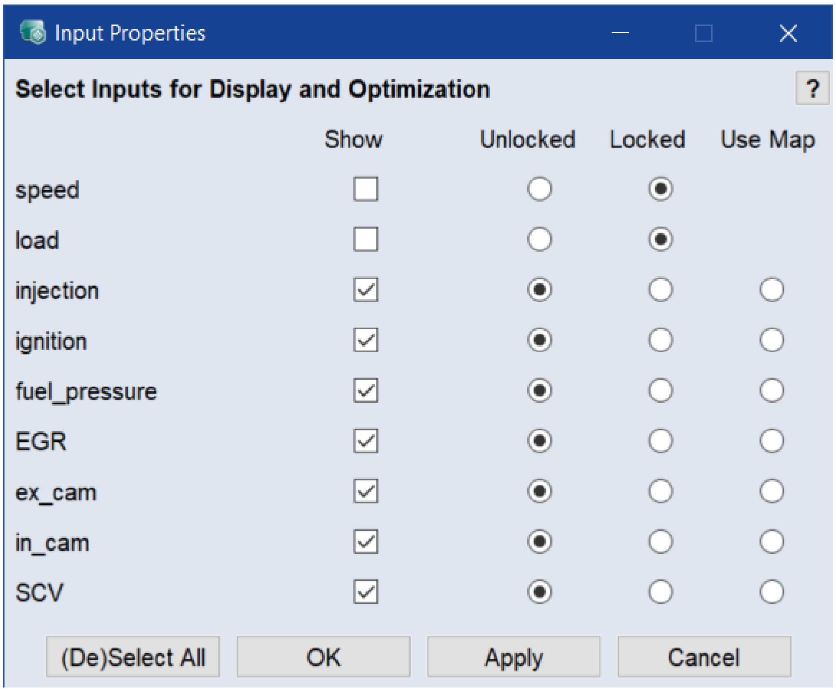


4. Enter minimum and maximum, then click **Apply** or **OK**.

The display range of the inputs can be scaled manually or automatically using the context menu options (right-click in the intersection plot) **Manual Input Range *** and **Autoscaled Input Range ***.

<u>Defining the display of inputs</u>

1. Select **In/Outputs › Input Properties**.
2. In the **Input Properties** window, activate the **Show** checkbox for all inputs you want to display.



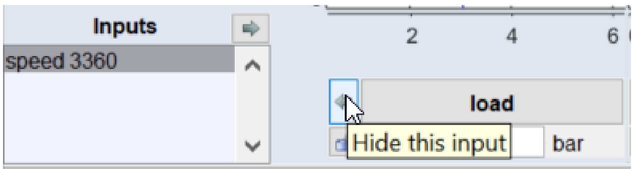
Since only unlocked inputs are optimized, it is recommended that you keep the default setting **unlocked** for all inputs in this tutorial.

3. Click **OK**.

   Intersection plots for the non-selected inputs are no longer displayed. Instead, the inputs are listed at the bottom left.

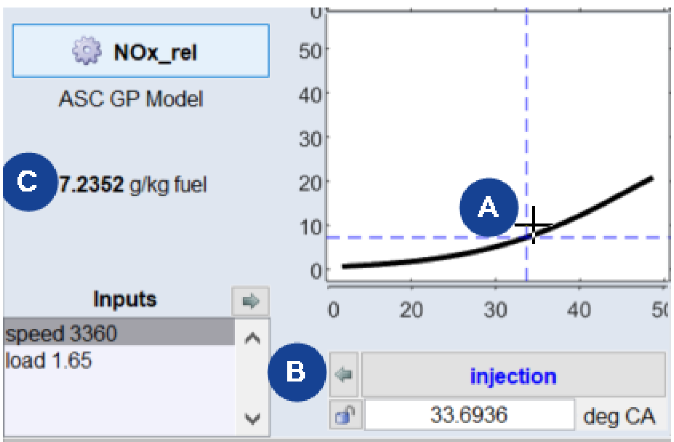
or

As an alternative, click the arrow next to an input name to deactivate this input.



4. Double-click an entry in the **Inputs** list to re-enable that input or select multiple inputs with STRG/CTRL and use the arrow.

Setting values of inputs



– **Setting a value (A)**

  To set an X value in an intersection plot, click a point.

  To continue navigating, hold the left mouse button pressed, drag the cross in the plot, and release the mouse button again at the desired point **(A)**.

– **Current value of the respective input (B)**

  The value of the input can also be entered directly in the field under the name of the input.

– **Current value of the output (C)**

  The corresponding output value is also displayed. The shapes of all other plots are adjusted accordingly.

Optimizing manually and storing the results

You can manually perform simple optimization tasks:

1. Select **In/Outputs > 2D Plot Operating Points** to display the operating points in the **Operating Points Manager** window.

   Operating Points Manager opens (see Fig. 6-6: on page 117).

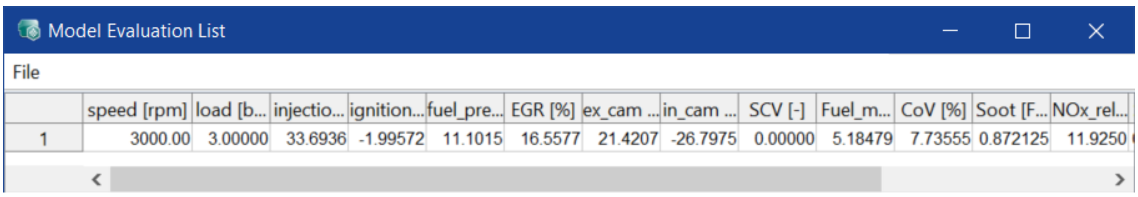
2. Enter 3000 and 3 in the input fields to set `speed` to approx. 3000 and `load` to approx. 3.

3. Adjust all inputs so that the value of the `Fuel_mass` output becomes as small as possible.

   From the intersection plots for the `Fuel_mass` output, it can be seen immediately that the inputs `injection` and `EGR` will have the greatest influence on the fuel consumption.

4. After you have established the desired settings (`Fuel_mass` should be approx. 4), select **Extras > Model Evaluation List > Add Current Setting to List** in the main menu.

   The **Model Evaluation List** window opens and the values currently set in the ISP view are copied into the list.



	speed [rpm]	load [b...	injectio...	ignition...	fuel_pre...	EGR [%]	ex_cam ...	in_cam ...	SCV [-]	Fuel_m...	CoV [%]	Soot [F...	NOx_rel...
1	3000.00	3.00000	33.6936	-1.99572	11.1015	16.5577	21.4207	-26.7975	0.00000	5.18479	7.73555	0.872125	11.9250

5. Optionally: Save the list via **File > Export**.

6.6.2 2D and 3D Visualization of Inputs and Outputs

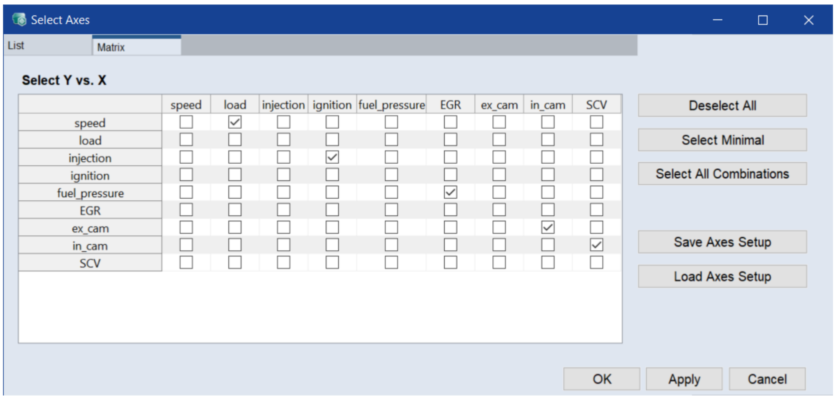
Besides the display of the functional dependencies in the ISP view, it is also possible to display two inputs each in form of 2D plots or one output and two inputs in form of 3D plots.

2D display of two inputs

1. Select **In/Outputs › 2D Plot Inputs**.

   The **Select Axes** window opens.

2. Go to the **Matrix** tab.

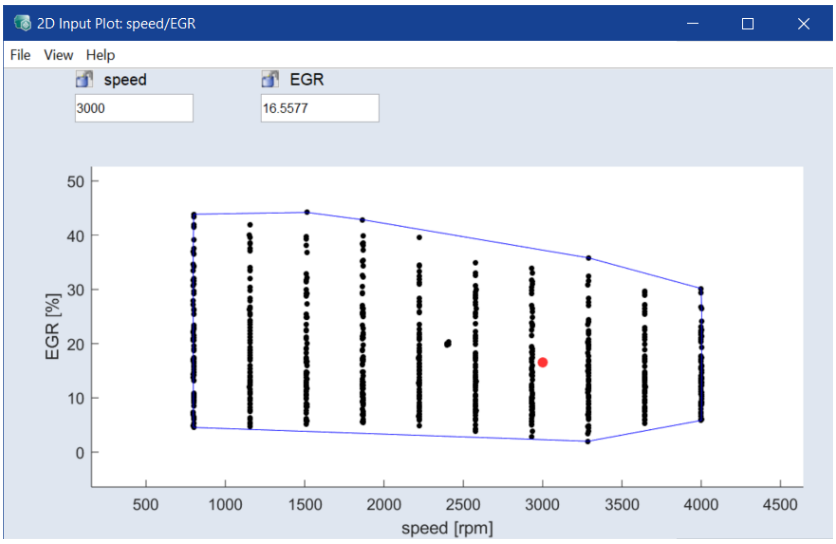


3. To delete the predefined selection, click **Deselect All**.

4. Click the checkbox in the **EGR** row and the **speed** column.

5. Click **OK**.

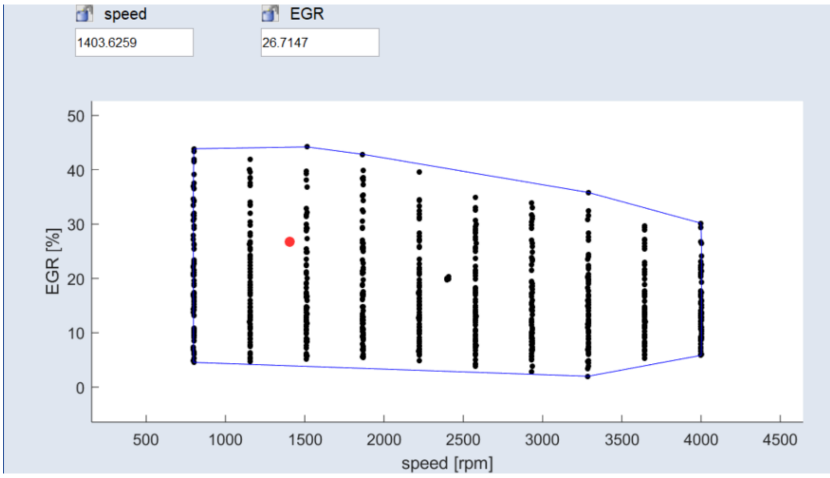
   The measuring data (black points) of the exhaust-gas recirculation at different speeds are displayed in a plot window.



   The red point represents the currently selected value of speed and EGR setting in the ISP view.

6. Click a point in the plot.

⇨ The red point moves to the selected position and the ISP view is updated.



Changes in the ISP view (that apply to the values of speed or EGR) are conversely also updated in the 2D plot.

Special 2D display for operating points

A special 2D display is available for any selected operating points in case of global models. This display is named ***Operating Points Manager*** *window* or *operating points manager*. A separate display of these variables outside of the ISP view is useful since they frequently have the greatest influence on output variables, such as consumption and emissions. For this reason, the dynamics of the influence of other inputs is frequently lost in the ISP view.

1. Select **In/Outputs › 2D Plot Operating Points**.

   The **Operating Points Manager** window opens.

   The navigation (i.e. the selection of operating points) can now be performed in the separate window.
2. If desired, deactivate the display of the operating points in the ISP view.
3. To select a new operating point, do one of the following:
   - Click in the plot.
   - Enter the desired values in the input fields above the plot (**(A)** in Fig. 6-6: on the next page).

     The set operating point can be added to the operating point list via the **Add OP** button.
4. To start a single-criterion optimization after each change of an operating point, select **Extras › Optimize on Move**.
5. To visualize the operating points in the plot (**(B)** in Fig. 6-6: on the next page), select **View › Show OPs**.
6. To visualize the measurement points in the plot (**(C)** in Fig. 6-6: on the next page), select **View › Show Measurement Points**.

7. To visualize the convex hull of the plot (**(D)** in Fig. 6-6: below), select **View › Show Convex Hull**.

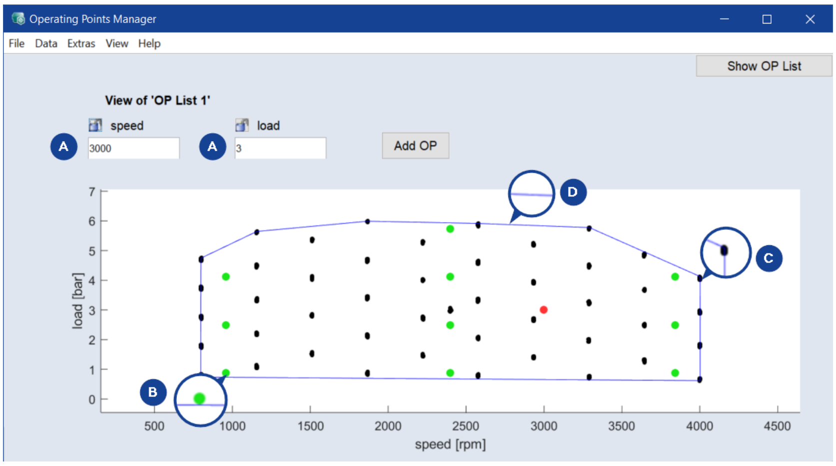


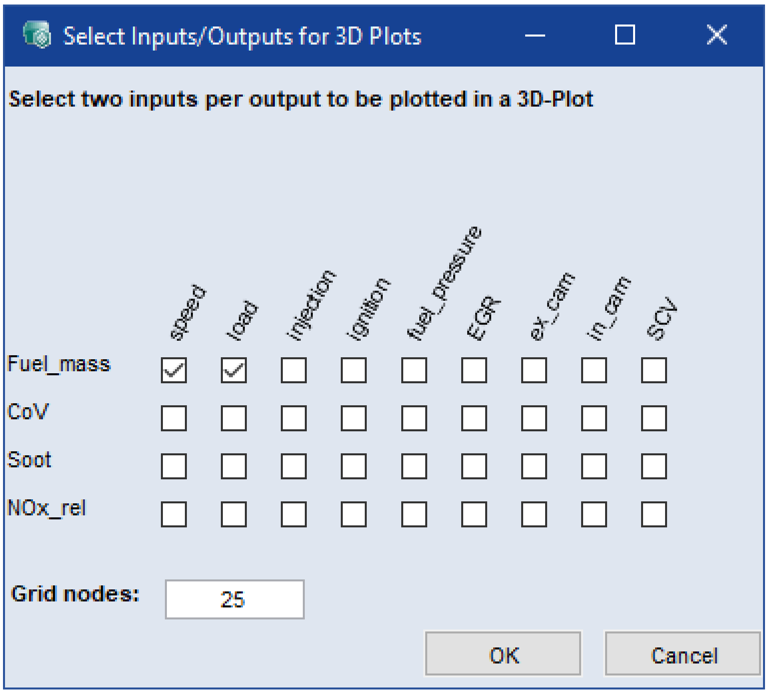
Fig. 6-6: Operating points manager (A: input fields for operating point values, B: operating points, C: measurement points, D: convex hull of the plot)

<u>3D display of the influence of input variables on an output variable</u>

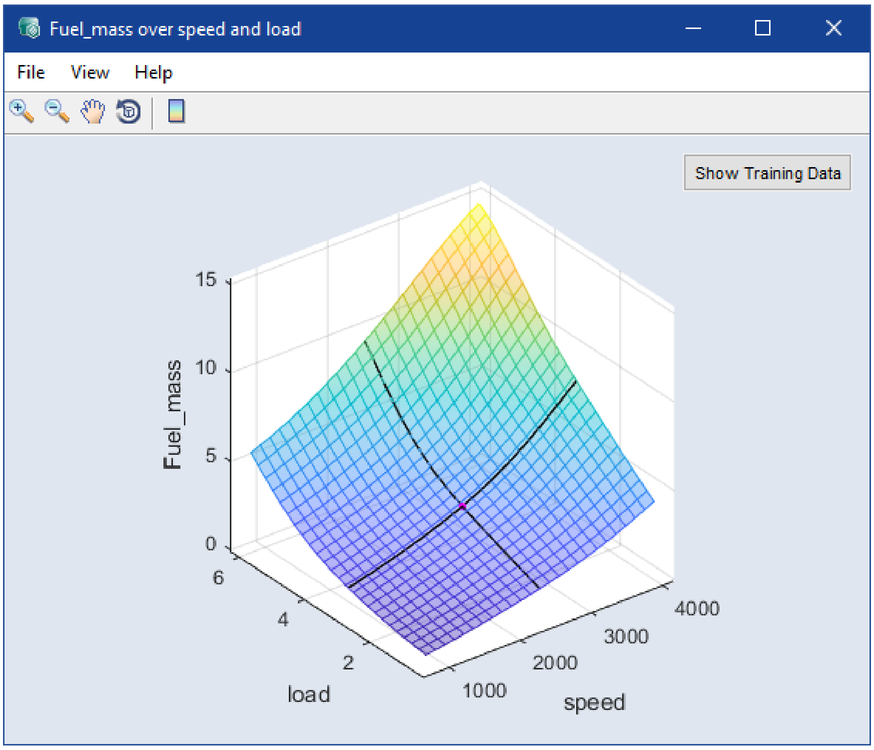
An additional option concerning the graphical display consists of visualizing the influence of two inputs on a modeled output.

1. Select **In/Outputs › 3D Plot Outputs**.

   A dialog window for the selection of inputs (two per output) opens.
2. Activate, for example, **speed** and **load** for the **Fuel_mass** output.
3. In the **Grid nodes** field, enter the number of grid nodes per axis.
4. Click **OK**.



The 3D view of the model of `Fuel_mass` across the speed/load range is displayed.



The intersection of the black lines represents the current values of the inputs, but they cannot be changed in the plot.

In case of changes of the displayed input variables in the ISP view, the position of the point in the plot is being updated.

5. In the toolbar of the 3D view, click **Insert Colorbar** .
6. Click **Show Training Data** to display the training data.

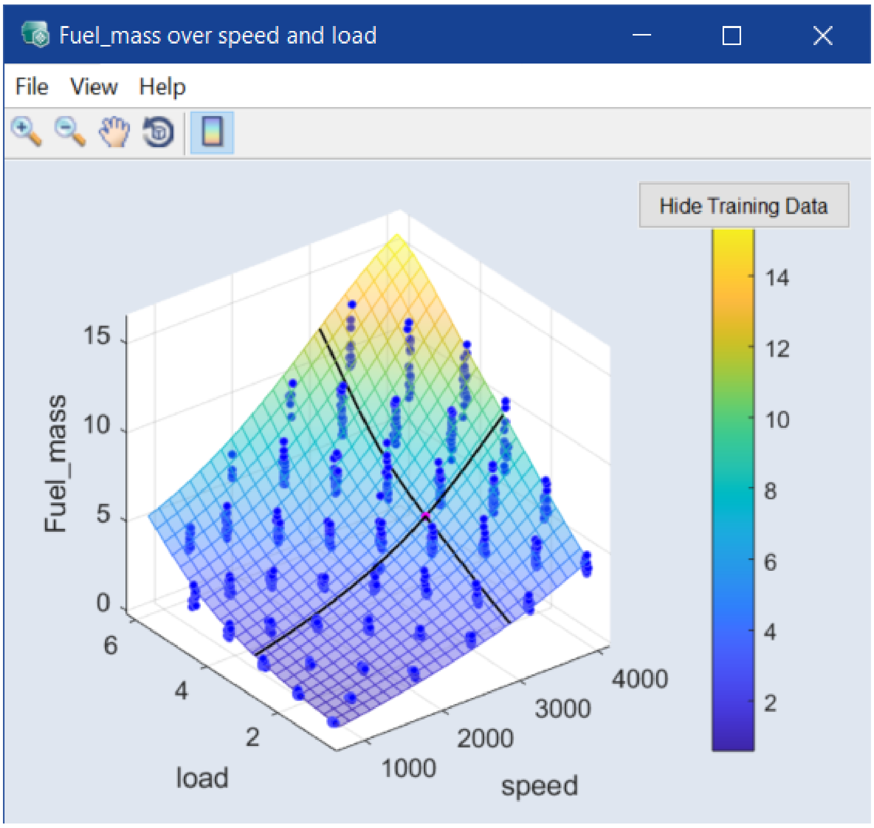


Fig. 6-7: 3D plot of `Fuel_mass` over `speed` and `load`, with training data (blue dots) and color bar

7. Select **View › Show Training Data Bounds** to display upper and lower limit of the training data.

Rotating and flipping the 3D plot

1. To rotate and/or flip the plot, click **Rotate 3D** .

   The mouse pointer changes into a circular arrow.

   After a brief delay, the plot can be rotated into any direction and flipped while holding the left mouse button pressed.
2. To end the rotation mode, click again **Rotate 3D**.

Show 3D-Plot with contour lines and legend

If desired, you can add contour lines to the 3D plot. Contours are also available for calibration maps and result maps; see 6.7.5 "Calibration " on page 141for more information.

1. In the 3D-Plot, select **View › Contour Mode**.

   Contour lines are added to the plot. The plot is rotated so that you see onto the speed-load plane.
2. Rotate the plot to the position you need.

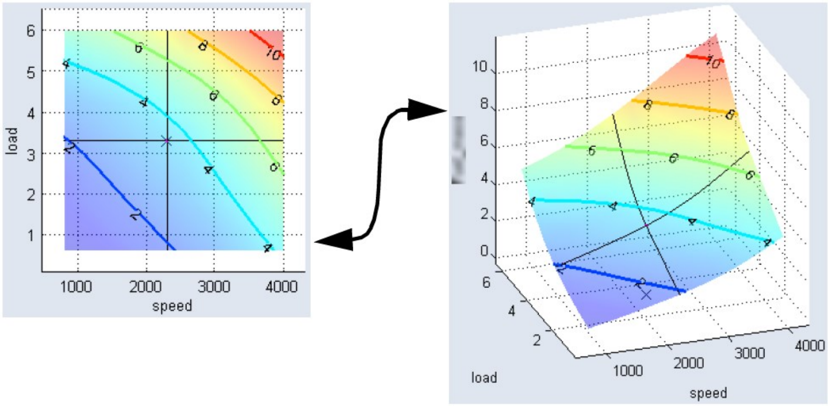


Fig. 6-8: 3D plot of `Fuel_mass` over `speed` and `load`, with contour lines

3. Use the **View › Contour Options › *** menu options to set up the contour mode.

   Details are given in the online help.

The **View › Contour *** commands are also available for calibration maps and result maps; see 6.7.5 "Calibration " on page 141for more information.

6.7 Optimization

In this part of the tutorial, you perform several types of optimization with ASCMO-STATIC. You also learn how to work with the results of a global optimization at several operating points.

This section provides information about the following topics:

- 6.7.1 "Single-Result Optimization with Weighted Total of Single Result " below
- 6.7.2 "Optimization at Several Operating Points " on page 126
- 6.7.3 "Multi-Criteria Optimization" on page 130
- 6.7.4 "Global Optimization " on page 140
- 6.7.5 "Calibration " on page 141

> **Note**
>
> A description of a cycle-based global optimization can be found in 6.9 "Cycle-Based Global Optimization" on page 162.

6.7.1 Single-Result Optimization with Weighted Total of Single Result

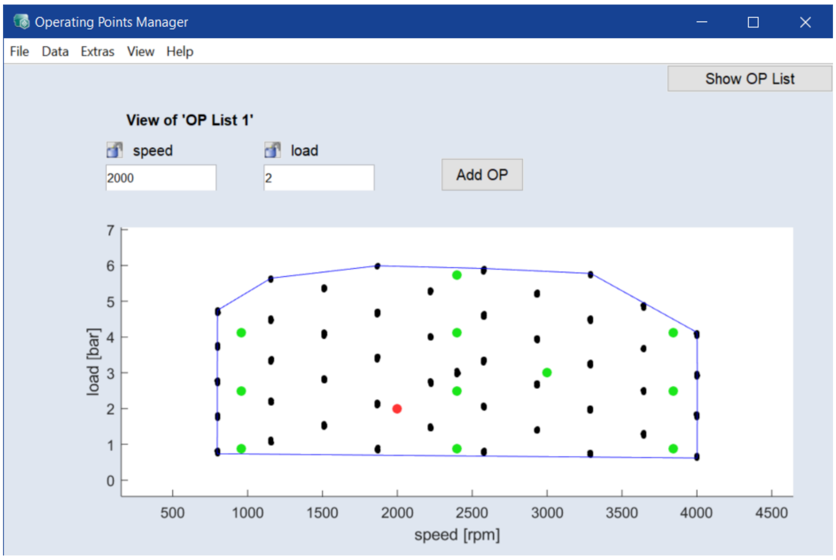
The task: Minimize fuel consumption (`Fuel_mass`) and attempt to keep other variables, such as engine roughness (`CoV`), soot (`Soot`) and nitrogen oxide (`NOx_rel`), below their predefined limits.

Defining the operating point

1. Select **In/Outputs > 2D Plot Operating Points**.

   The two variables `load` and `speed`, which define the operating point, are now displayed in a separate window.

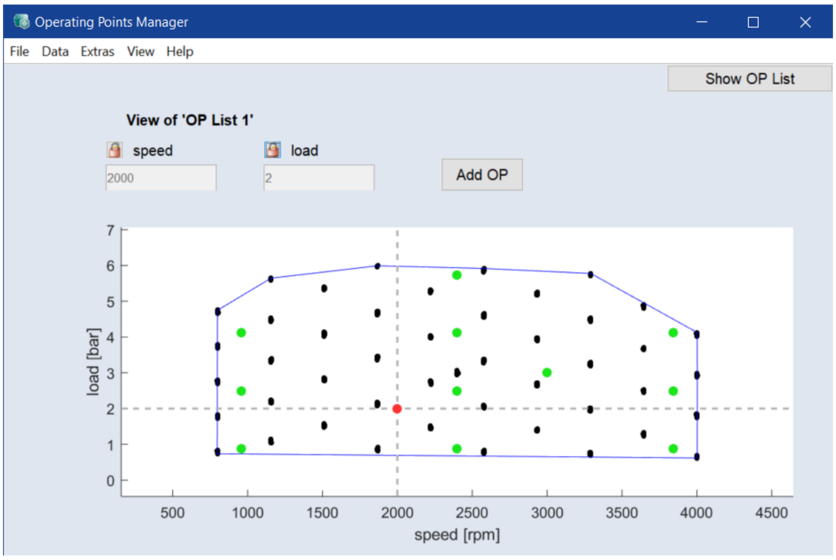
2. In this window, define the operating point at which you want to perform the optimization, e.g., `speed` = 2000 rpm and `load` = 2 bar.

   You have to lock `load` and `speed`, or their values will be varied during optimization.

3. Click the lock icons () next to **speed** and **load**.

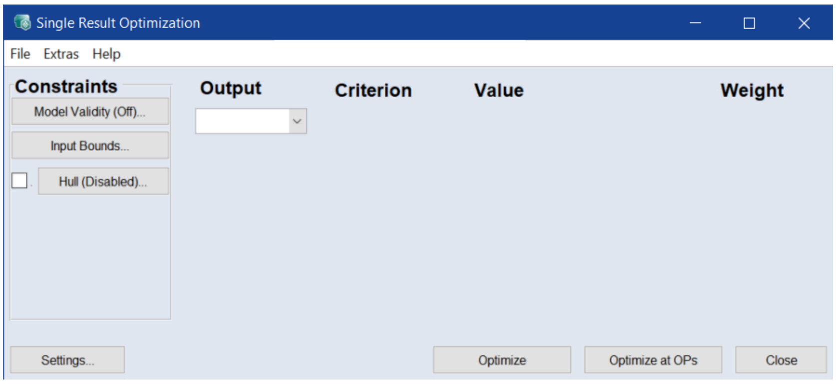
➪ The locks are closed (), the input fields are disabled, and the red dot in the plot is fixed.



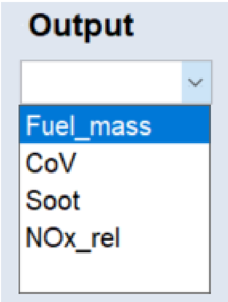
Specifying optimization variables and criteria

1. Select **Optimization › Single Result**.

   The **Single Result Optimization** window opens. The **Output** column is used to select the outputs to be optimized; the other columns are used to set optimization criteria.

2. In the **Output** column, click in the empty cell.

3. Select `Fuel_mass` from the drop-down list.

   The respective optimization criterion is set to the default value. A new empty row is added.

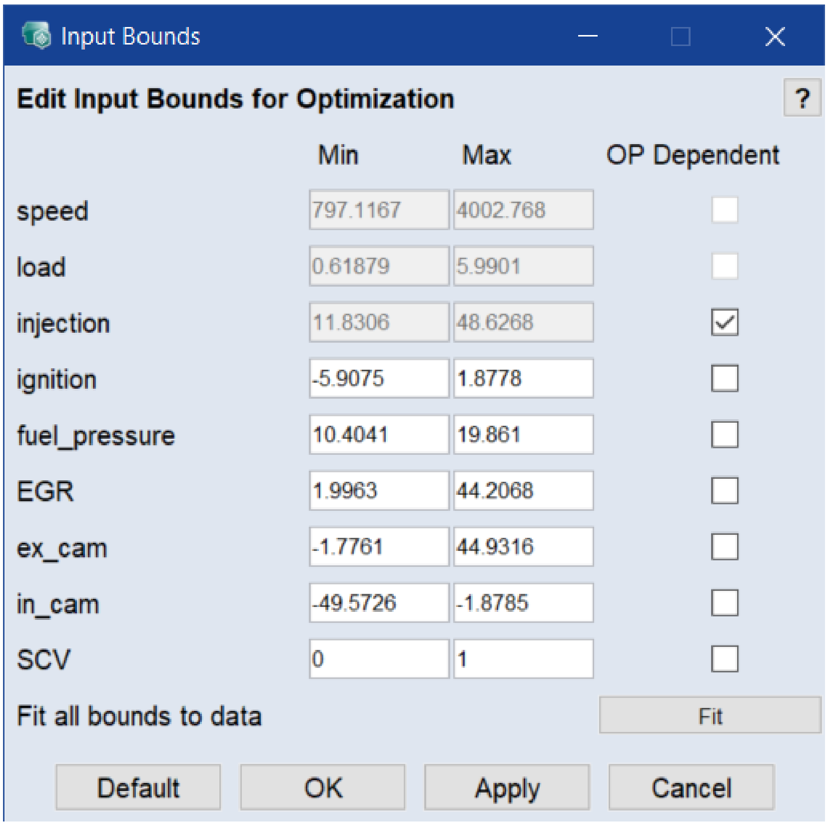
4. Select all remaining outputs, i.e. `CoV`, `Soot` and `NOx_rel`.
5. Enter the following optimization criteria:

Output	Criterion	Value			Weight	
`Fuel_mass`	Minimize	None	-	-	Constant	1
`CoV`	Bound	Weak Upper Bound	Constant	5 [%]	Constant	1
`Soot`	Bound	Weak Upper Bound	Constant	0.3 []	Constant	1
`NOx_rel`	Bound	Weak Upper Bound	Constant	30 [g/kg]	Constant	1

Limiting the optimization results

You can limit the optimization results to the range of valid model output in the **Single Result Optimization** window in different ways.

1. To limit the optimization result automatically to the range of valid model outputs, do the following:
   i. Click the **Model Validity** button.
   The **Valid Model Range** window opens.
   ii. Activate each element you want to limit to the valid model range.
   iii. Click **OK** to apply your settings and close the **Valid Model Range** window.
2. To limit the minimal and maximal values of an input, do the following:
   i. Click the **Input Bounds** button.
   The **Input Bounds** window opens. You can edit the min. and max. values of an input.



   ii. In the **Input Bounds** window, click **Fit** to fit all bounds to measured data.

   Or

   Do one of the following for each input you want to limit:
   - Enter the range of valid parameter variation ("Min"/"Max") for optimization.
   - Activate the **OP Dependent** checkbox.

The **Map Bounds** window opens for the respective input. Here you can manually specify the range of parameter variation using the defined grid points.

iii. In the **Input Bounds** window, click **OK** to apply your settings and close the window.

> **Note**
>
> For more information regarding the definition of valid parameter variation using OP-dependent input bounds, see 6.7.5 "Calibration " on page 141.

3. To limit the range of parameter variation to the data within the specified hull, do the following:

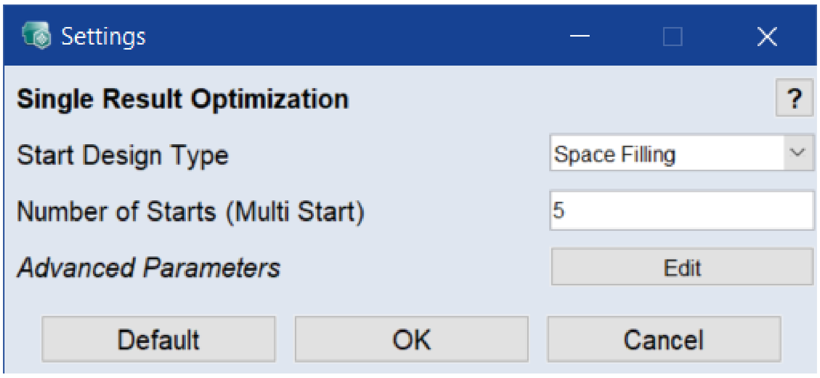
   i. Activate the checkbox next to the **Hull** button.

   This limits the range of parameter variation to the data within the specified hull (**In/Outputs › Hull on Inputs**). Current setting can be checked by clicking the **Hull** button.

Change settings of the Single Result optimization

1. In the **Single Result Optimization** window, click **Settings**.

   The **Settings** window opens.



> **Note**
>
> The **Edit** button to customize the **Advanced Parameters** is only available if the **Advanced Settings** are enabled (see 4.4 "Advanced Settings in ASCMO-STATIC " on page 63.)

2. In the **Start Design Type** drop-down field, specify which start design will be used for optimization.

   The following settings are possible:

   Last: The current settings in the ISP view are taken as start values.

   Space Filling: The start values are selected space-filling by Sobol algorithm.

All Edges: Start values = all corners of the experimental space. The rest of the population is distributed in the parameter space according to the Sobol algorithm.

3. In the input field **Number of Starts (Multi Start)**, define the number of optimization runs.

   To ensure that a global minimum is found, the optimizer should be started multiple times using different start values.

4. Click **OK** to confirm the settings.

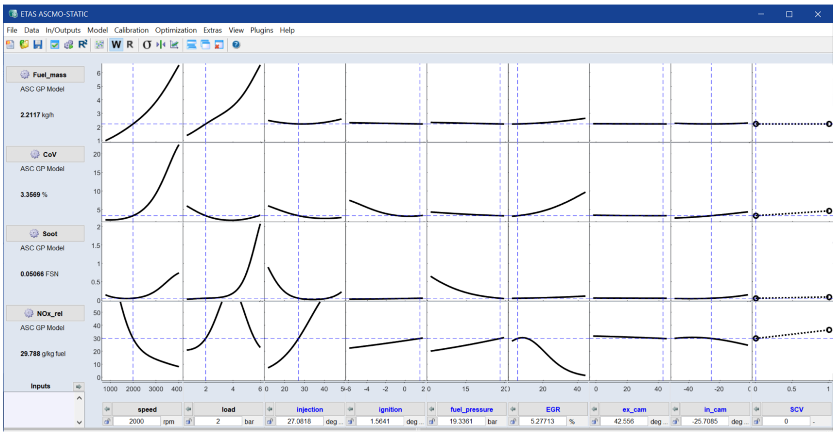
   The settings are applied. A message appears in the log window.

Performing the optimization

1. In the **Single Result Optimization** window, click **Optimize**.

➪ The optimization is performed.

The results of the optimization are displayed in detail in the log window under the ISP view, and the optimized values are also set in the ISP view.



The result manifests itself in a minimized fuel quantity while maintaining the predefined upper limits for the other outputs.

Note

Your results may slightly deviate from the ones achieved here since the models of the outputs may differ slightly depending on the preprocessing of the training data (removal of outliers and repetition points, etc.).

Saving the result of the local optimization

1. Select **Extras › Evaluation List › Add Current Settings to List** to save these optimization data.

   The **Model Evaluation List** window opens and the current optimization result is added to it.

2. In that window, select **File › Export** to save the Model Evaluation List for later use.

3. In the **Save Data** file selection window, enter or select type (e.g. `*.xlsx` or `*.xls` or `*.csv` or `*.ascii`), path and name for the export file, then click **Save**.

Automatic optimization upon selecting the operating point

You can set up ASCMO-STATIC so that a single-result optimization is performed automatically each time you select a new operating point.

1. In the operating points manager (shown in Fig. 6-6: on page 117), select **Extras › Optimize On Move**.
2. Click in the plot to select a new operating point (red point).

⇨ The optimization based on the selected criteria starts for the selected operating point.

6.7.2 Optimization at Several Operating Points

The method described in 6.7.1 "Single-Result Optimization with Weighted Total of Single Result " on page 120 can also be executed in a type of batch mode. The optimizer sequentially runs through a series of operating points, thereby optimizing "globally".

This optimization is the starting point for creating maps based on modeled outputs.

Defining the grid of operating points

First, select the points at which you want to optimize.

1. Select **Calibration › Operating Points**.

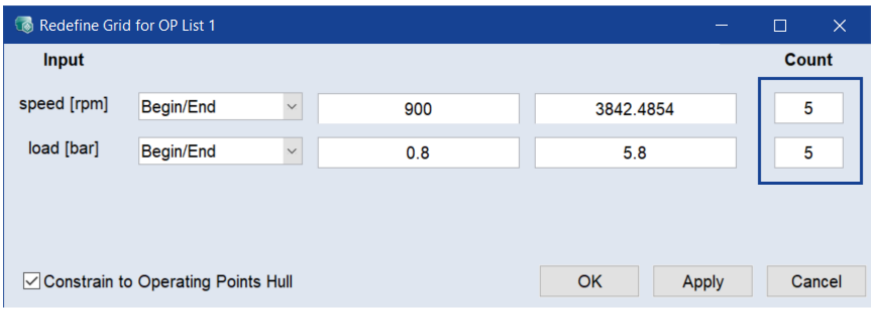
   The **Operating Points Manager** window (see Fig. 6-6: on page 117) opens.
2. In that window, select **Data › Edit List › Redefine Grid** to create a grid that deviates from the default.

   The **Redefine Grid for OP List <n>** window opens.
3. In that window, do the following:

   i. In the drop-down lists of both parameters, select `Begin/End`

   ii. In the input fields, enter the ranges for `speed` and `load`.

   iii. In the **Count** column, enter, e.g., a 5x5 grid.



   iv. If desired, activate the **Constraint to Operating Points Hull** option to ignore operating points outside the hull.

v. Click **OK** to apply your settings and close the window.

A 2-dimensional grid of equidistant points is created. The points are shown in the operating points manager.

It is also possible to import an existing list of operating points via **File › Import** in the operating points manager.

Defining optimization criteria

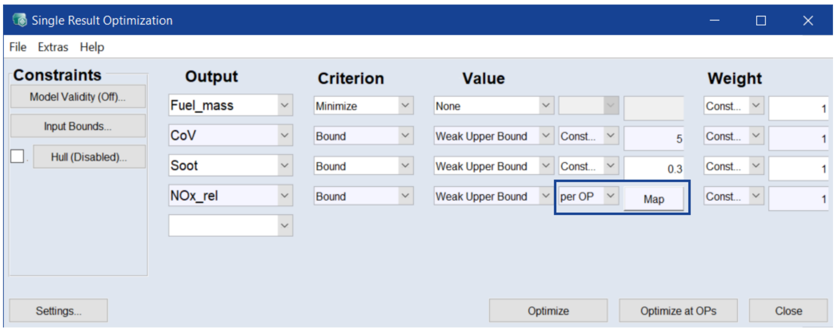
1. Select **Optimization › Single Result**.

   The **Single Result Optimization** window opens. The window contains the criteria of the previous local optimization.

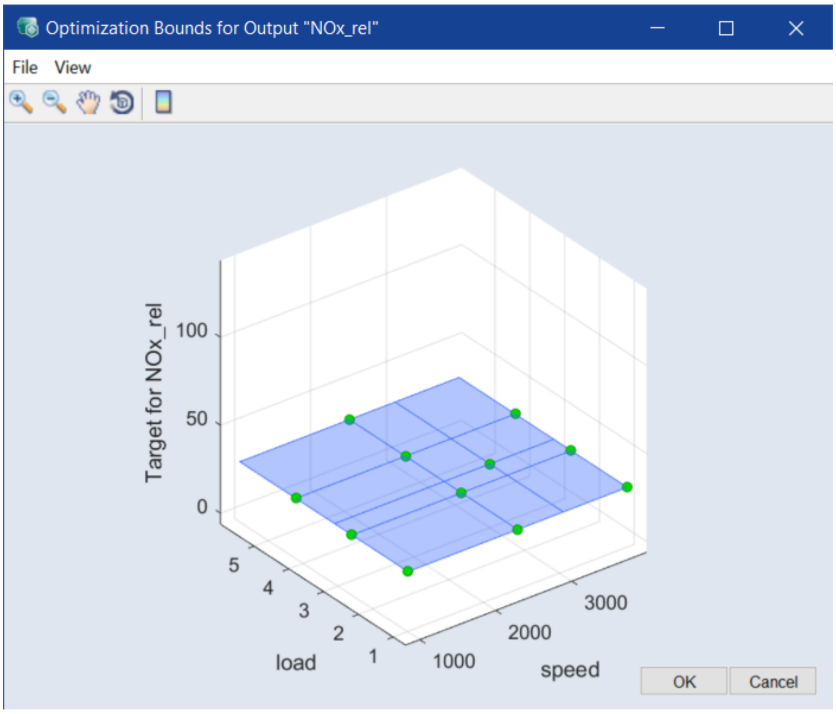
2. For the `NOx_rel` output, change the value in the middle **Value** column from `Constant` to `per OP`.

   The input field in the right **Value** column becomes a button.

3. Click **Map**.



The plot for the optimization settings at all defined operating points opens.



The value 30, which was previously the default for the `Constant` setting, has been copied to all operating points.

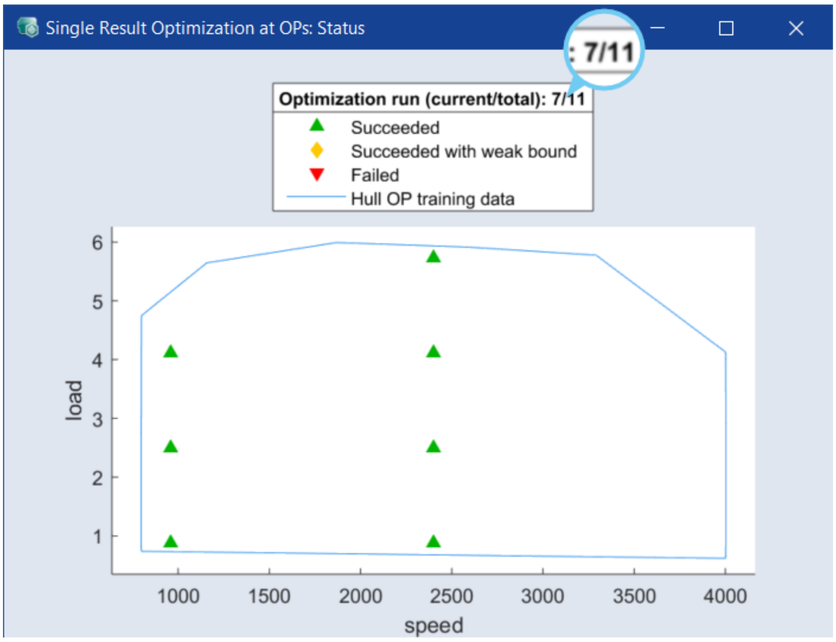
4. To edit the value of a single operating point, do one of the following:
   - Move a point with the mouse.
   - Select **View › Table**, then edit the values in the **Optimization Target Table for Output *** window.
5. If desired, select **File › Import Target Map** to load an Excel file with the settings (values at the respective operating points).
6. To continue with this tutorial, close the window with **Cancel** to discard any changes you made.

Performing the optimization at all operating points

1. In the **Single Result Optimization** window, click **Optimize at OPs**.

   The optimization is performed.

   A window opens in which you can track the progress of the optimization at the individual operating points.

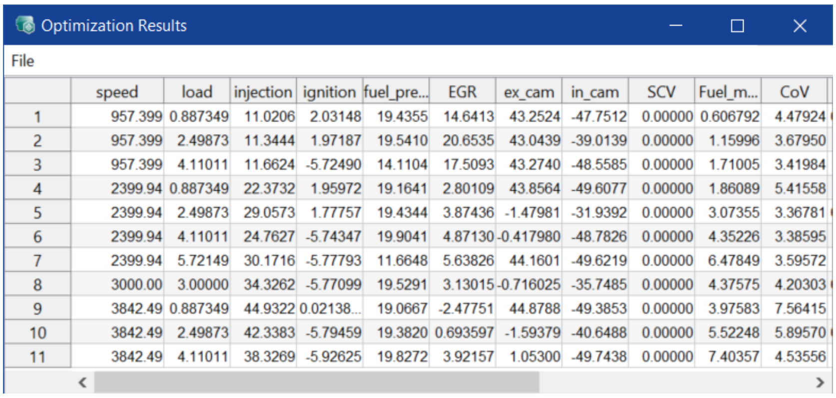


   During the optimization at the respective operating points, the optimization values are set in the ISP view and output in the status window.

   The optimization was successful at the green operating points.

2. To represent the results of the optimization runs at the operating points in a table, select **Extras › Optimization at OPs ›Show Results** in the **Single Result Optimization** window.

   The **Optimization Results** window opens. It contains the optimized results for all operating points.



Optimization Results

File

	speed	load	injection	ignition	fuel_pre...	EGR	ex_cam	in_cam	SCV	Fuel_m...	CoV
1	957.399	0.887349	11.0206	2.03148	19.4355	14.6413	43.2524	-47.7512	0.00000	0.606792	4.47924
2	957.399	2.49873	11.3444	1.97187	19.5410	20.6535	43.0439	-39.0139	0.00000	1.15996	3.67950
3	957.399	4.11011	11.6624	-5.72490	14.1104	17.5093	43.2740	-48.5585	0.00000	1.71005	3.41984
4	2399.94	0.887349	22.3732	1.95972	19.1641	2.80109	43.8564	-49.6077	0.00000	1.86089	5.41558
5	2399.94	2.49873	29.0573	1.77757	19.4344	3.87436	-1.47981	-31.9392	0.00000	3.07355	3.36781
6	2399.94	4.11011	24.7627	-5.74347	19.9041	4.87130	-0.417980	-48.7826	0.00000	4.35226	3.38595
7	2399.94	5.72149	30.1716	-5.77793	11.6648	5.63826	44.1601	-49.6219	0.00000	6.47849	3.59572
8	3000.00	3.00000	34.3262	-5.77099	19.5291	3.13015	-0.716025	-35.7485	0.00000	4.37575	4.20303
9	3842.49	0.887349	44.9322	0.02138...	19.0667	-2.47751	44.8788	-49.3853	0.00000	3.97583	7.56415
10	3842.49	2.49873	42.3383	-5.79459	19.3820	0.693597	-1.59379	-40.6488	0.00000	5.52248	5.89570
11	3842.49	4.11011	38.3269	-5.92625	19.8272	3.92157	1.05300	-49.7438	0.00000	7.40357	4.53556

In addition, the table contains the standard deviations at the respective operating points.

3. In the **Optimization Results** window, mark a row and select **File › Show Current Row in Intersection Plot**.

⇨ The values of the inputs and (as a result) the outputs are set in the ISP view.

This list can also be saved as Excel file (**File › Export**).

Transfer optimization results to Calibration Maps

1. In the **Optimization Results** window, select **File › Apply Results to Calibration Maps**.
2. In the main menu, select **Calibration › Calibration Maps › *** to view the new calibration maps.

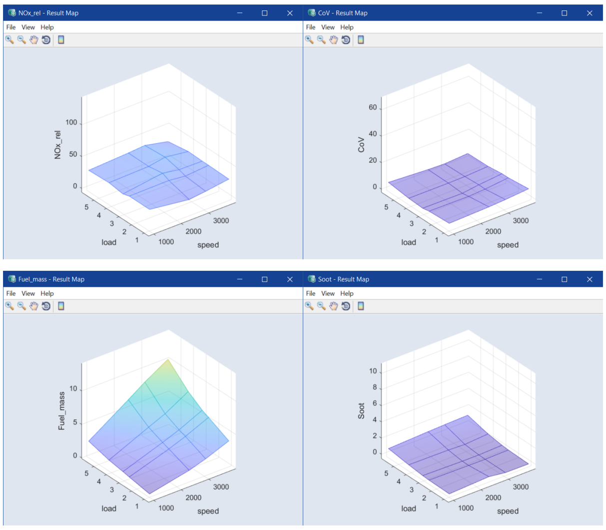
Note

For more information on handling and visualizing calibration maps, see 6.7.5 "Calibration " on page 141.

Viewing optimization results (Result Maps)

1. Select **Calibration › Result Maps › Open all Maps**.

   The plots of the results for all outputs are opened.

   The points represent the operating points at which the optimization has been performed.

2. In the main menu, select **View › Close Child Windows** to close the plot windows and any other child windows.

Note

The current results of the optimization – the maps – are described in section 6.7.5 "Calibration " on page 141.

6.7.3 Multi-Criteria Optimization

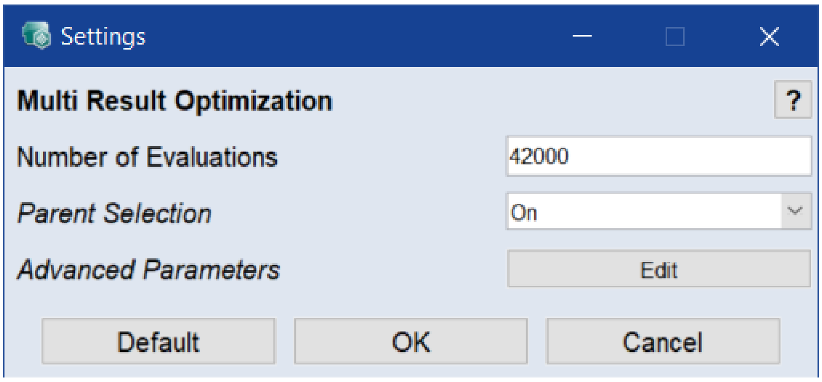
Optimization tasks frequently lead to the situation that the reduction, e.g. of an emission variable, is associated with the increase of other values, such as consumption. The multi-criteria optimization is available for the optimization of such "opposite" outputs.

Change settings of the multi-criteria optimization

1. Select **Optimization › Multi Result**.

   The **Multi Result Optimization** window opens.

2. In the **Multi Result Optimization** window, click **Settings**.



Note

The **Edit** button to customize the advanced parameters is only available if the **Advanced Settings** are enabled (see 4.4.1 "Enable/Disable Advanced Settings " on page 64).

3. In the **Number of Evaluations** field, specify the maximum number of function evaluations across all generations.

   The number of evaluations should be approximately 200 times the number of children-generating parents (**Parent Population Size** in the advanced parameters).

4. In the **Parent Selection** field, activate (`On`) or deactivate (`Off`) involvement of the parents in evaluation and recombinations of new generations.

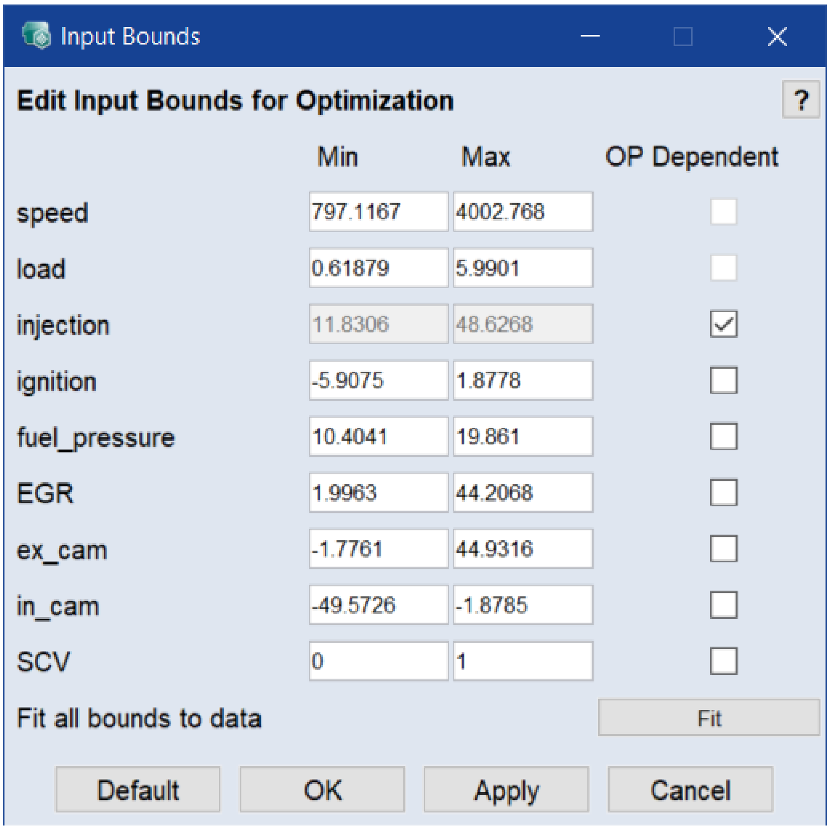
   For more information, see 4.5.3 "Evolutionary Algorithm (Parent Selection vs. Survivor Selection) " on page 74.

5. Click **OK** to confirm the settings.

<u>Restrict the range of parameter variation (Constraints)</u>

For every input, the range to be considered for the modeling can be defined here. To do so, proceed as follows:

1. In the **Multi Result Optimization** window, click **Input Bounds**.



2. In the **Input Bounds** window, click **Fit** to fit all bounds to measured data.

Or

Do one of the following for each input you want to limit:

- Enter the range of valid parameter variation (**Min**/**Max**) for optimization.
- Activate the **OP Dependent** checkbox.

  The **Map Bounds** window opens for the respective input. Here you can manually specify the range of parameter variation using the defined grid points.

3. Click **OK** to apply your settings and close the **Input Bounds** window.

Note

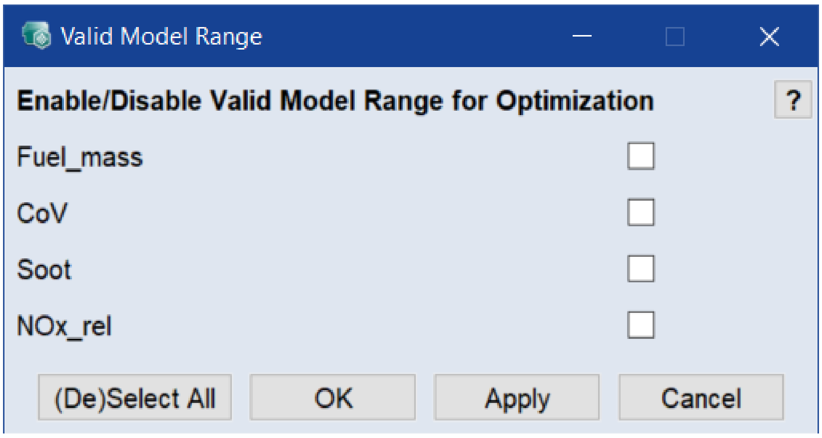
For more information regarding the definition of valid parameter variation using OP-dependent input bounds, see 6.7.5 "Calibration " on page 141.

Limiting the optimization results

You can limit the optimization results to the range of valid model output in different ways.

1. In the **Multi Result Optimization** window, activate the option next to the **Hulls** button.

   This limits the range of parameter variation to the data within the specified hull (**In/Outputs › Hull on Inputs**).
2. Click **Model Validity**.

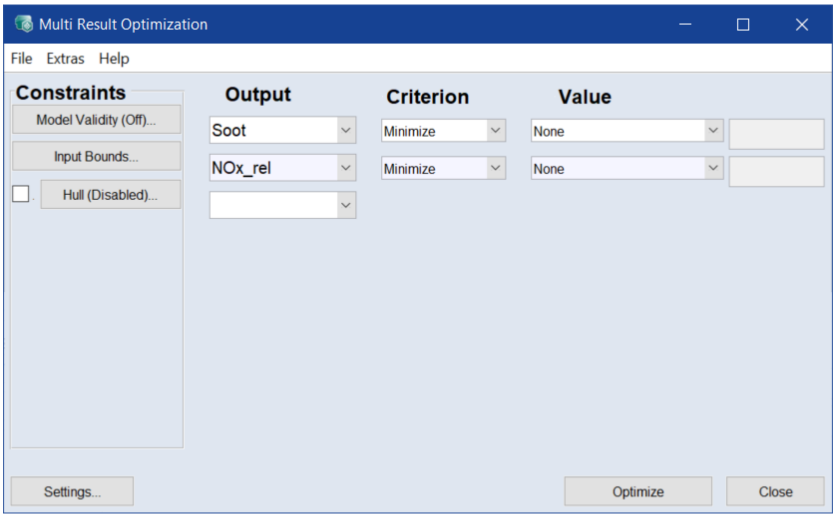
   

3. In that window, activate each element you want to limit to the valid model range.
4. Click **OK** to apply your settings and close the **Valid Model Range** window.

> **Note**
>
> The options in the **Valid Model Range** window should always be enabled so that the solutions found by the optimizer can also be set later. A prerequisite for the use of this option is that the corresponding variables have a meaningful validity range. If not, it would be an indication for a faulty model training.

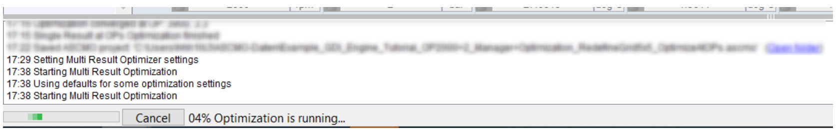
Multi-criteria optimization

1. If necessary, open the **Multi Result Optimization** window.
2. Remove the outputs `Fuel_mass` and `CoV` by selecting `Remove` in the **Output** column.
3. Select the `Minimize` optimization criterion for the two outputs `Soot` and `NOx_rel`.

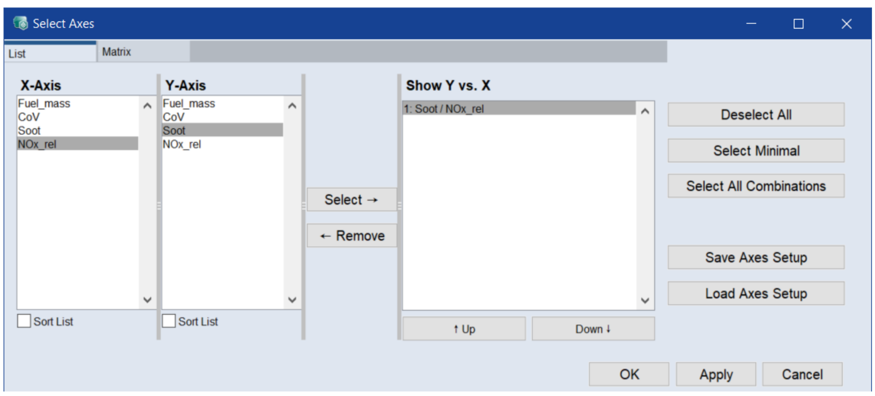


4. Open the operating points manager (cf. Fig. 6-6: on page 117).
5. In that window, select an operating point at which you want to perform the optimization, e.g.,
   - 2000 rpm
   - 2 bar mean effective pressure
6. Lock `speed` and `load` as described in "Defining the operating point" on page 121.
7. In the **Multi Result Optimization** window, click **Optimize**.

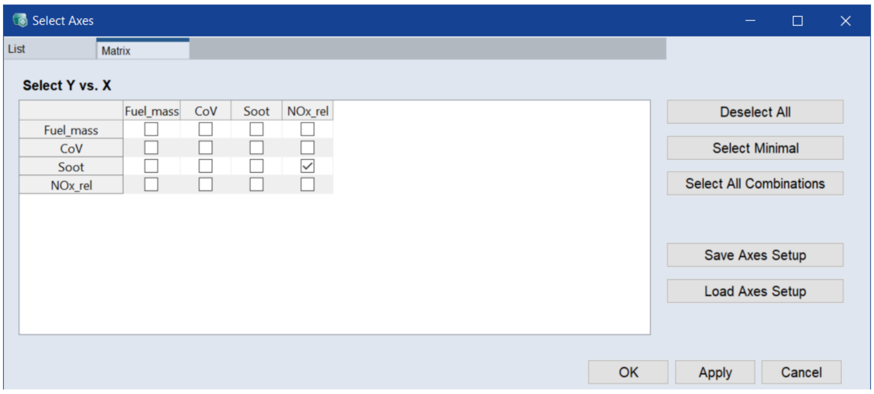
   The optimization starts The progress is displayed in the status bar at the bottom edge of the window.



   Upon completion, the **Select Axes** window opens.
8. In that window, in the **List** tab or in the **Matrix** tab, select the pair of axes to be displayed.
   - In the **List** tab, select `NOx_rel` as X axis and `Soot` as Y axis, then click **Select →**.

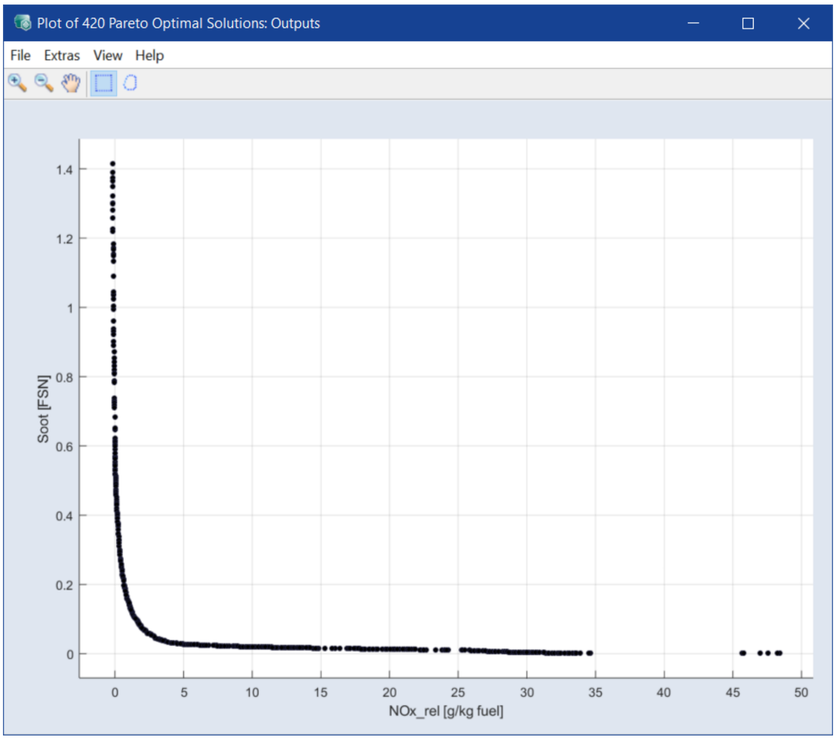


- In the **Matrix** tab, activate the checkbox in the **Soot** row, **NOx_rel** column.



9. Click **OK**.

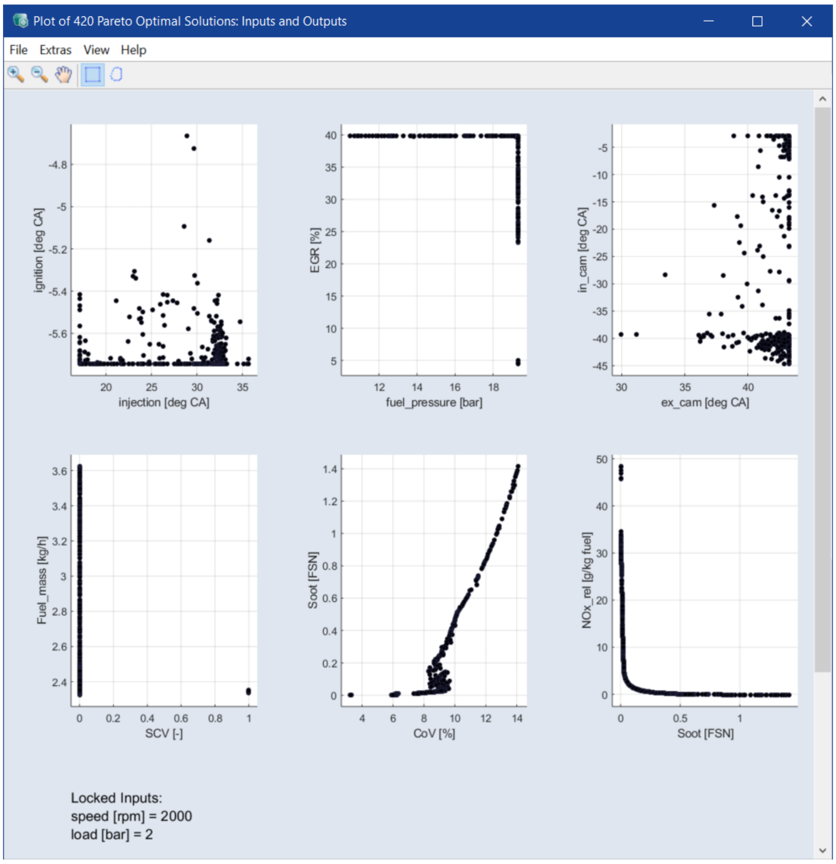
➪ The plot for the mutual dependency between soot and nitrogen oxide emission is displayed.



Note

All points on this curve are Pareto-optimal solutions with respect to the optimization criteria.

In addition, a second window opens in which dependencies between inputs and outputs can also be displayed.



For the function of the **Inputs and Outputs** window, see "Addressing a group of Pareto solutions" on the next page.

Addressing individual Pareto solutions

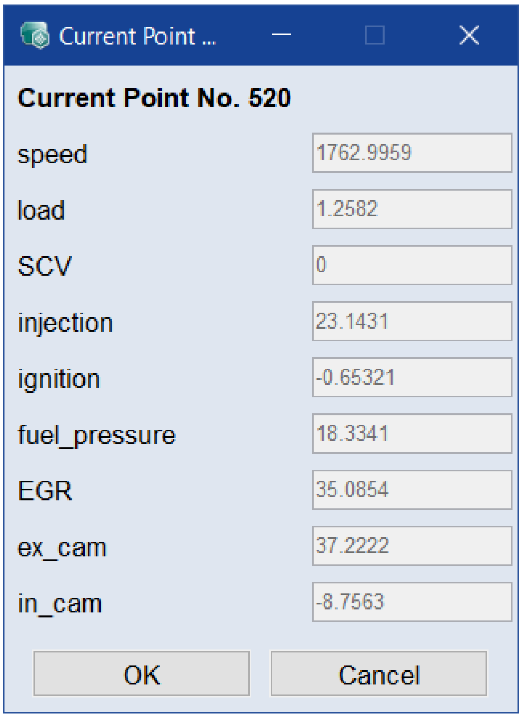
1. Right-click a point of the plot (= one solution of the optimization).

   The selected solution is highlighted and a context menu opens.

2. Select **Show Result in Other Views** to display this solution in the ISP view.

3. Select **Show Result** to display the values of the inputs for this solution in

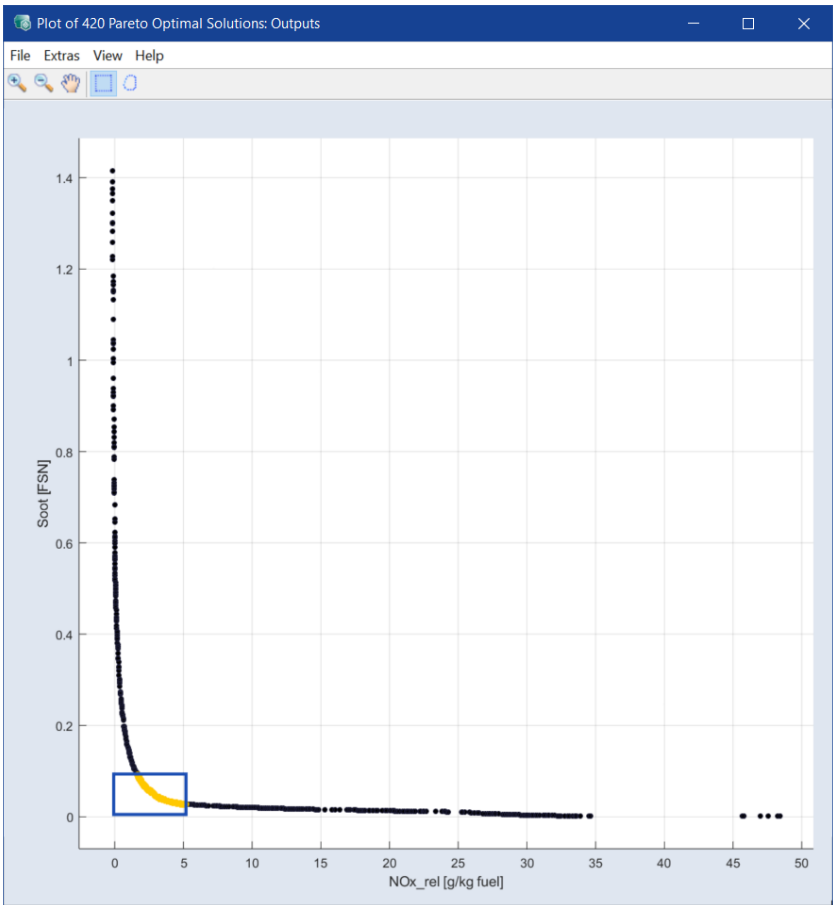
a window.

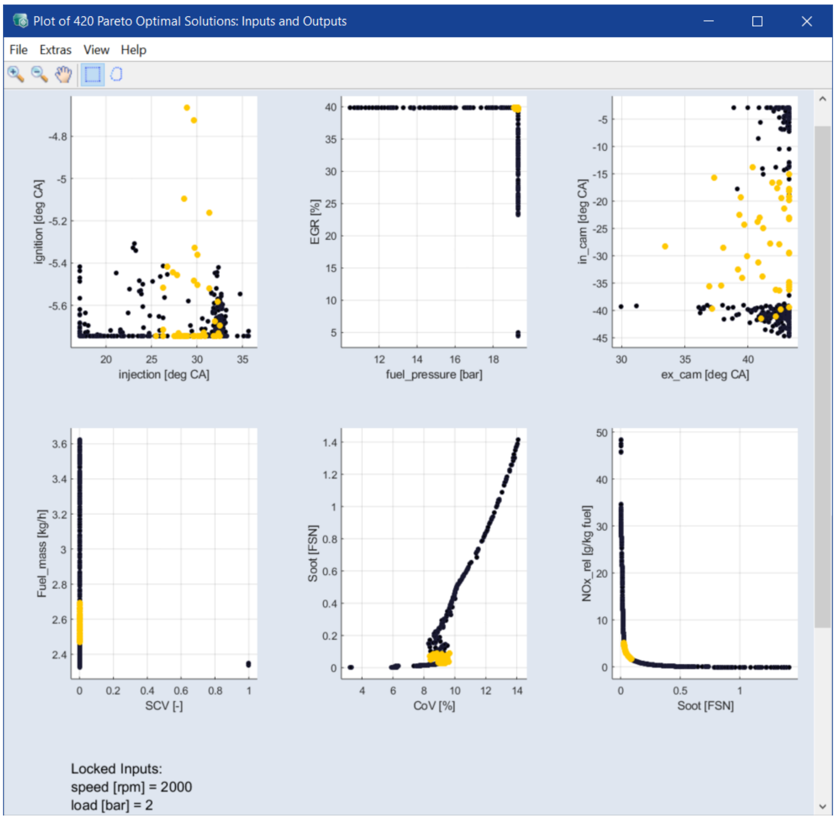


Addressing a group of Pareto solutions

1. Click one of the **Mouse selection *** buttons.
2. Draw a rectangle or lasso around certain solutions.  

   The solutions inside the shape are highlighted.


Plot of 420 Pareto Optimal Solutions: Outputs
File Extras View Help
Soot [FSN]
NOx_rel [g/kg fuel]


Plot of 420 Pareto Optimal Solutions: Inputs and Outputs
File Extras View Help
ignition [deg CA]
injection [deg CA]
EGR [%]
fuel_pressure [bar]
in_cam [deg CA]
ex_cam [deg CA]
Fuel_mass [kg/h]
SCV [-]
Soot [FSN]
CoV [%]
NOx_rel [g/kg fuel]
Soot [FSN]
Locked Inputs:
speed [rpm] = 2000
load [bar] = 2

Note

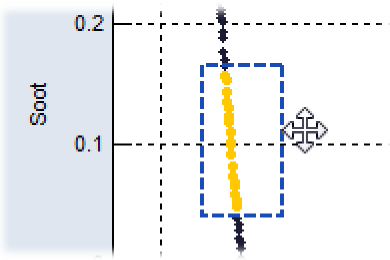
These solutions are also highlighted in the other plots of the **Input and Outputs** windows and the concerned areas of inputs of interest can also be identified.

3. Right-click the frame.

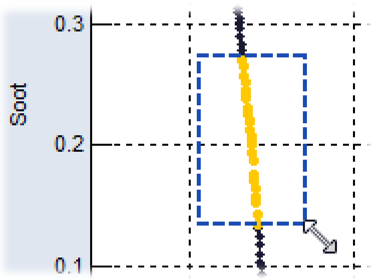
   A context menu opens with which you can perform the following actions:

   - **Remove Rectangle** or **Remove Lasso**: Removes the rectangle or lasso again and therefore the selection of identified solutions.
   - **Set Position**: Only available for rectangles.

     Opens the **Create Rectangle** window where you can edit the rectangle position.
   - **Mark**: Marks the solutions inside the rectangle/lasso for deletion.
   - **Show Number of Solutions**: Displays the number of solutions inside the rectangle/lasso.
4. Click a side of a rectangle/lasso to move it and mark different solutions.



5. Click a corner of a rectangle to resize it.



   You cannot resize a lasso.
6. If you are displaying additional axes with **View › Select Axes**, you can also observe the effect of your selection on these variables.

Saving results

You can save the entire set of Pareto-optimized solutions or solutions in a selected area in a file.

1. In one of the plot windows, select **File › Export All Data / Export Intersection of Selected Data / Export Union of Selected Data**.

   A file selection window opens.

2. Enter a file name and click **Save**.

➪ The solutions are saved in an Excel file.

6.7.4 Global Optimization

For a global optimization, the optimization is performed at all operating points at the same time (not successively like the Single Criteria Optimization, which goes through all operating points in succession in batch mode).

> **Note**
>
> The **Calibration** menu and the **Global Optimization** function in the **Optimization** menu are only available, if operating point axes have been selected during data import (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards in the menu **In/Outputs › Set Operating Point Axes**.

This enables two additional items:

– Consideration of the smoothness of the maps (steepness/gradient of the maps)

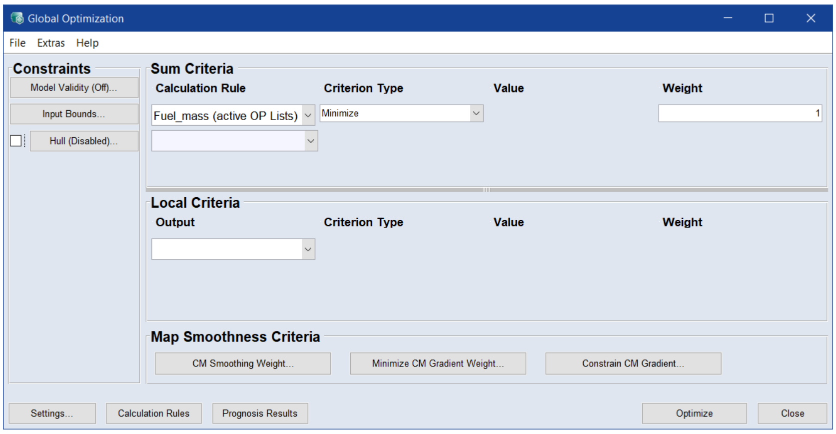
– Observation of total values for driving cycles with a weighting with respect to the duration of stay at the operating points (see also 6.8 "Driving Cycle Forecast " on page 147)

The necessary criteria and limits for global optimization can be set in the ***Global Optimization** window*. For a description of the **Global Optimization** window, see the ASCMO-STATIC online help.

<u>Defining optimization criteria</u>

1. Select **Optimization › Global Optimization**.

   The **Global Optimization** window opens.



2. In the **Sum Criteria** area, select all four outputs, each with criterion `Minimize`.

3. In the **Constraints** area, do the following:

- Use the **Input Bound** button to restrict the range of parameter variation (cf. "Restrict the range of parameter variation (Constraints)" on page 131).
- Activate the checkbox next to the **Hulls** button to limit the range of parameter variation to the data within the specified hull (**In/Outputs › Hull on Inputs**).

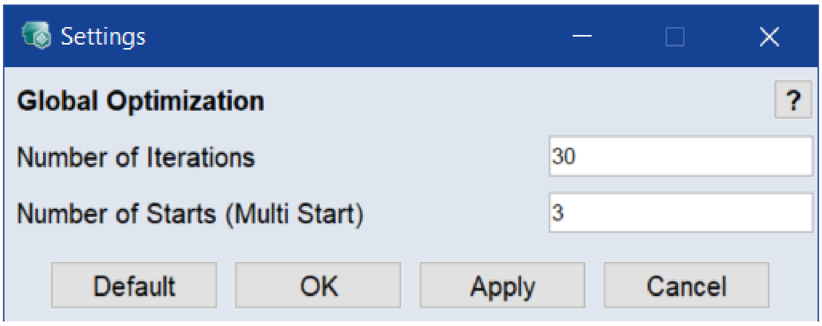
Note

Further information for the definition of the parameter variation for the optimization can be found in 6.7.5 "Calibration " below.

Change settings of the global optimization

1. In the **Global Optimization** window, click **Settings**.

   The **Settings** window opens.



2. In the **Number of Iterations** field, define how many iterations are done during model training.

   A higher number results in better training results, but takes more times.
3. In the **Number of Starts (Multi Start)** field, define the repetitions of the optimization for every operating point with different starting values.
4. Click **Apply** or **OK** to accept the settings.

   Default restores the standard values.

Performing the optimization

1. In the **Global Optimization** window, click **Optimize**.

⇨ The optimization is performed.

Saving criteria

1. In the **Global Optimization** window, select **File › Save Criteria** to save the optimization criteria to a `*.cocrit` file to reuse it in another project.

6.7.5 Calibration

In this section, you edit the maps obtained from the global optimization.

Note

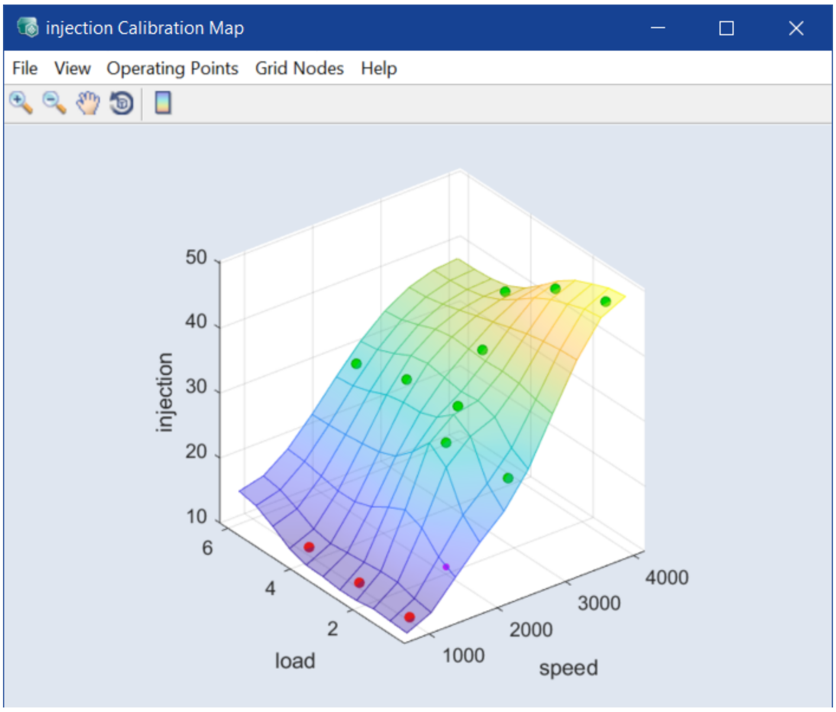
The **Calibration** menu and the **Global Optimization** function in the **Optimization** menu are only available if operating point axes have been selected (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards via the **In/Outputs › Set Operating Point Axes** menu option.

Viewing calibration maps

1. Select **Calibration › Calibration Maps › Open all Maps**.

   Windows with calibration maps of all inputs obtained from the optimization open.

   The following figure shows the calibration map of the `injection` input.

   The colored points are the operating points at which the optimization has been performed.

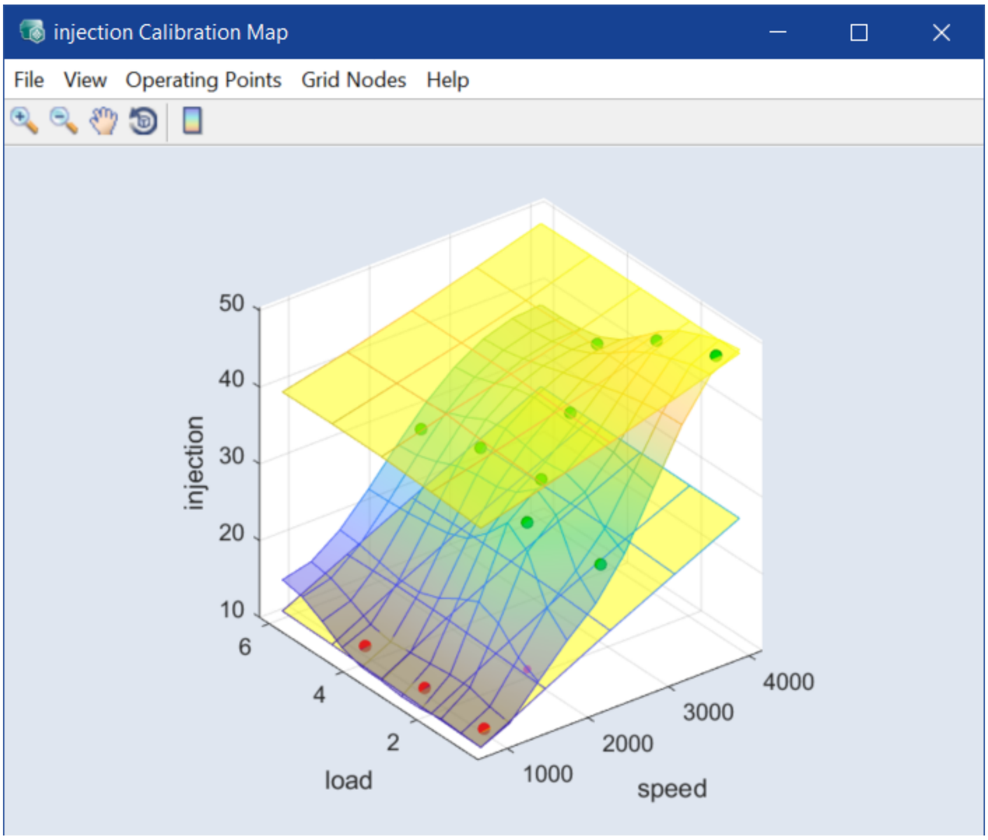
   - Green points indicate that the optimization result falls within the range of the measured data.
   - Points at the limit (= the smallest/largest measured value in this dimension) are marked in red.
   - Yellow is used to mark values that fall between this limit and the measuring range bounds that apply to this operating point.

2. To show the bounds of the measured range, select **View › Map Bounds** in a calibration map window.

Note

These bounds must first have been adjusted to the measured range; see "Adjusting bounds to the measured range" below.

⇨ The bounds of the measured range are drawn in the plot of the calibration map.



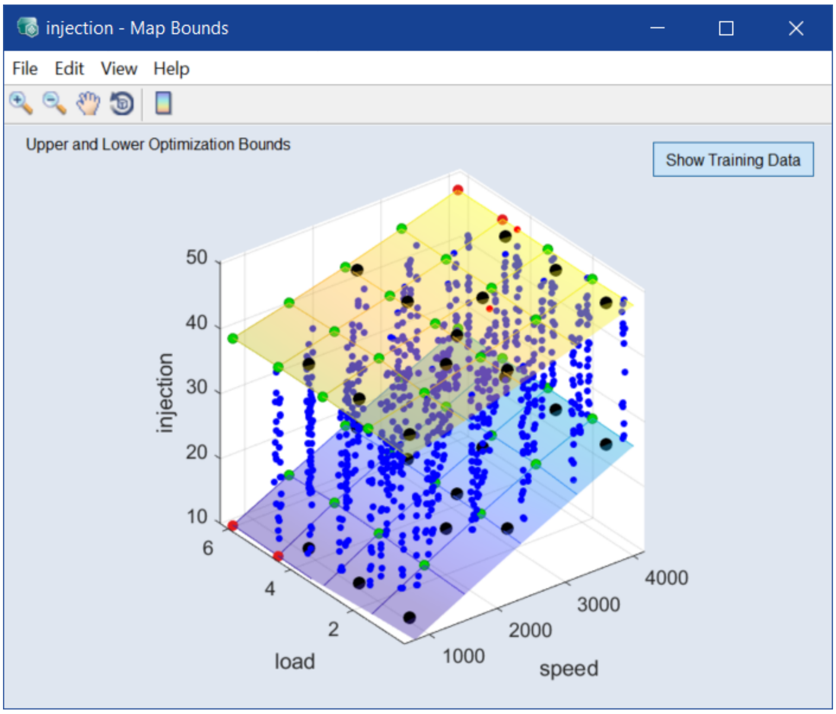
Adjusting bounds to the measured range

1. Select **Calibration › Map Bounds over OP › ‹input_name›**, we use `injection`.

   The **injection - Map Bounds** window opens.
2. Select **Edit › Fit Bounds to Data.**

   The **Fit Map Bounds to Data** window opens.
3. In that window, do one or more of the following:
   - Enter a smoothness factor for the map bounds.
   - In the **Grid Nodes** area, redefine the grid.
   - Activate **Apply to all maps** to apply the changes to all **‹input_name› - Map Bounds** maps.
4. Click **OK** or **Apply** to continue.

   The areas for the lower and upper limit of the measured range of the `injection` input are adapted to the measured data.



5. Close the window.

   Instead of adjusting the bounds for each input separately, you can use **Calibration › Map Bounds over OP › Fit Bounds to Data** or **Fit Bounds to Min/Max** to adjust the bounds for all inputs.

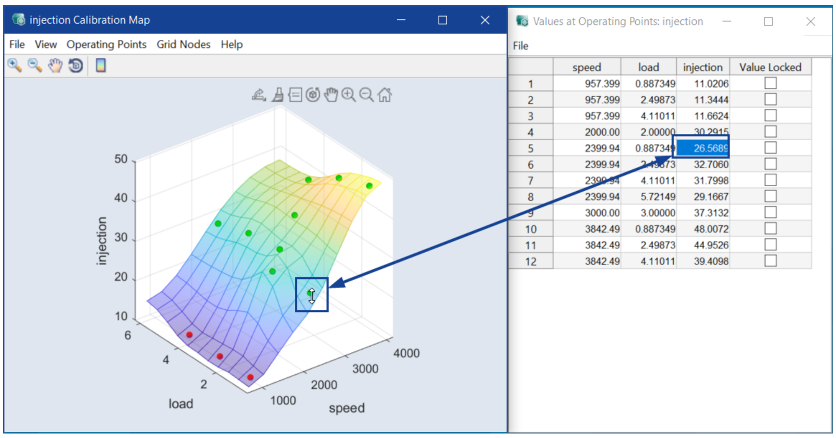
<u>Changing values at operating points</u>

1. In the **injection Calibration Map** window, select **Operating Points › Table of Values at OPs**.

   The **Values at Operating Points: injection** window opens.

2. In the **injection Calibration Map** window, click one of the operating points marked in color and hold the mouse button pressed.

   The mouse pointer changes to a double arrow. At the same time, the cell with the corresponding value is marked in the table.



3. Now, move the point with the mouse.

The value of `injection` is also changed in the table (and in the ISP view).

4. In the Values at **Operating Points: injection** window, change a value of `injection` at another operating point and press <ENTER>.

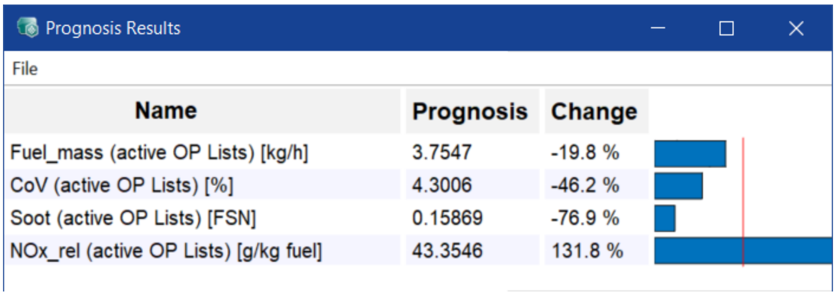
⇨ The plot and (if **View › Update ISP Online** is activated) ISP view are adjusted accordingly.

You can also make predictions of how these changes will affect a predefined driving cycle.

Opening the prognosis window

1. Select **Calibration › Prognosis › Results**.

   The **Prognosis Results** window opens.



2. Move a point in the **injection Calibration Map** window, as shown in "Changing values at operating points" on the previous page.

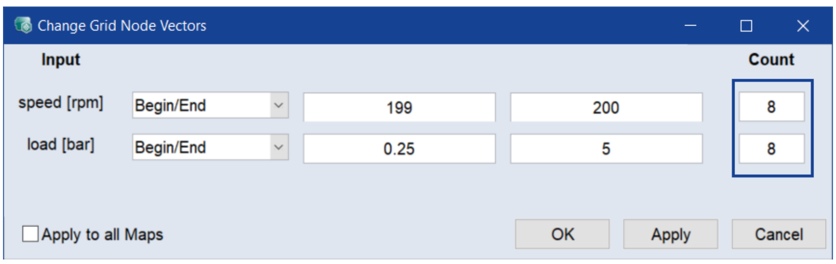
   The effects of this change on the outputs can be seen directly in the **Prognosis Results** window.

Adjusting grid nodes

1. To render the display of the **injection** map more clearly, deactivate **View › Map Bounds** in the plot window.
2. Select **Grid Nodes › Grid Nodes › Define Grid Nodes**.

   The **Change Grid Node Vectors** window opens.
3. To change the number of grid nodes in a calibration map, do the following:

   i. Enter the desired values in the **Count** fields.



   ii. Click **Apply**.

   The 10 x 10 grid for the grid nodes of the map is drawn. The values on each axis are equidistant.

Or

To enter the grid vectors directly, do the following:

i. In the drop-down list for each input, select `Support Vector`.

   The vectors of the grid nodes are displayed in the field of the corresponding input map.

ii. If necessary, edit the vector values and click **OK**.

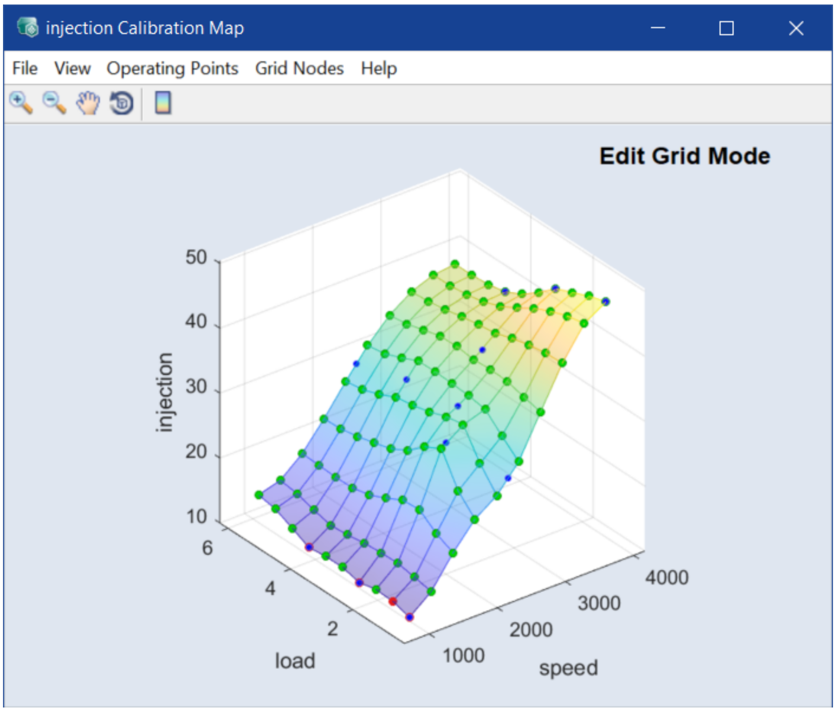
   The grid nodes of the map are adjusted accordingly. The **Change Grid Node Vectors** window closes.

> **Note**
>
> When you add/edit a vector (`Support Points`), the number of nodes in the column **Count** is adjusted automatically.

Editing the map

1. In the **injection Calibration Map** window, select **Grid Nodes › Edit Grid**.

   Similar to the optimized values of `injection` at the operating points, the (interpolated) values at the selected grid nodes of the map are now marked by colored points. The operating points, at which optimizations have been performed, are also displayed (small points).



2. To save the original state of the map, select **File › Set as Reference Page**.
3. The values at the grid nodes can now be edited as described in "Changing values at operating points" on page 144.
4. To reset the state of the map to the optimization result, select **File › Reset from Reference Page**.

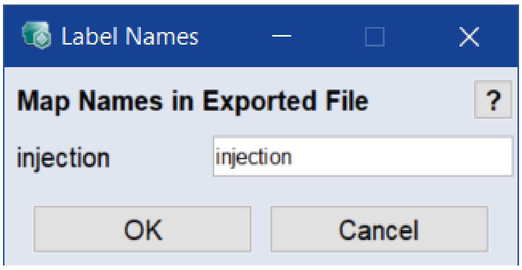
<u>Saving the calibration map</u>

Finally, you can export the optimized and, if necessary, edited map.

1. In the **injection Calibration Map** window, select **File › Export**.

   A file selection window opens in which you can save the map as DCM file (`*.dcm`) or in CSV format (`*.csv`). The default file name is <*project_name*>`_CM_Injection.*`.

2. In the following window, enter a label name and click **OK**.



⇨ The map is saved.

6.8 Driving Cycle Forecast

This section provides information about how to use a driving cycle for the definition of the prognosis calculation rules in ASCMO-STATIC.

A transient driving cycle is a series of data points involving many changes representing the speed of a vehicle versus time. Driving cycles assess the performance of vehicles in various ways, as for example fuel consumption and polluting emissions.

Fuel consumption and emission tests are performed on chassis dynamometers. Tailpipe emissions are collected and measured to indicate the performance of the vehicle. Another use for driving cycles in vehicle simulations such as ASCMO-STATIC.

6.8.1 Driving Cycle Data

The driving cycle data must meet the general requirements of ASCMO-STATIC. This means that the driving cycle data must be spanned over the operating point axis. In ASCMO-STATIC, the operating point axes are defined as rotational speed (`speed`) and load (`load`).

For more information regarding the measurement data requirements, see 4.2.1 "Requirements for Measuring Data" on page 29.

Import driving cycle data

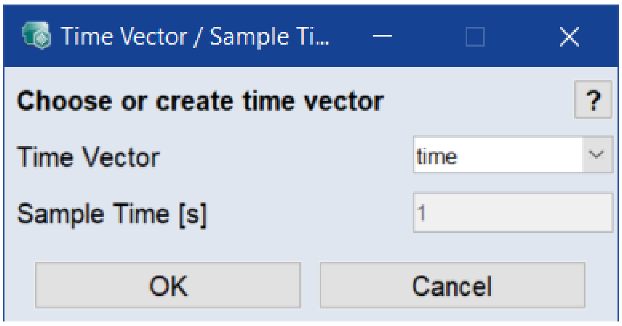
> **Note**
>
> The **Calibration** menu is only available, if operating point axes have been selected during data import (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards in the menu **In/Outputs › Set Operating Point Axes**.

1. Select **Calibration › Driving Cycles**.

   The **Driving Cycle Manager** window opens, displaying the message "`No Driving Cycle loaded, click to import`".
2. In the **Driving Cycle Manager** window, do one of the following:
   - Select **File › Import Driving Cycle**.
   - Click the message "`No Driving Cycle loaded, click to import`".

   A file selection window opens
3. In the file selection window, select the file `Example_Cycle_Trajectory.xls` in the directory *<installation>*`\Example\AscmoStatic`.
4. Click **Open**.
5. Select `time` out of the drop-down list.

   In the **Time Vector** drop-down list, select the column that contains the time information. Otherwise ASCMO-STATIC will automatically create the time vector with the entered **Sample Time**.

6. Click **OK**.

   The **Import Data** window opens. ASCMO-STATIC tries to match the input names of the loaded file to the operating point axes.

7. If the names in the project (**Project Import** column) differ from the names in the file (drop-downs in **Import Name** column), assign them manually.
8. Use default values for **Time Base** and **Other settings**.
9. If you want to use another name than the imported file, enter a name for the driving cycle in the **Dataset Name** field.

   This name is used in the ASCMO-STATIC user interface.
10. Click **Import** to start the import procedure.

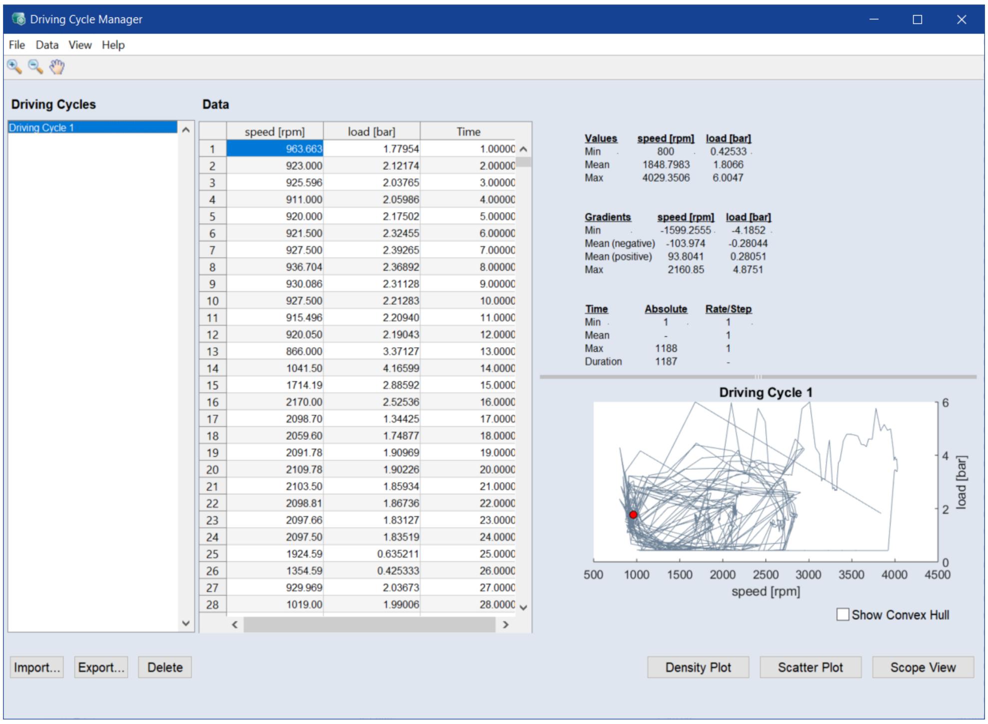
⇨ The driving cycle data file is imported. Once the import is finished, the **Driving Cycle Manager** window shows the driving cycle data in a table view.

> **Note**
>
> If you want to use several driving cycles in your project, import them by consecutively using **File › Import Driving Cycle** and "Modification Mode" `Add` or by loading an already existing database (`*.cycles`) using **File › Load Database**. Then ASCMO-STATIC can use all driving cycle traces for the cycle forecast.

The imported data can be used for global optimization based on Calculation Rules which use these driving cycle data (Main menu: **Calibration › Prognosis › Calculation Rules**: Driving Cycle Based Tab).

"Driving Cycle Manager" Window

Calibration › Driving Cycles menu option in the main menu opens the **Driving Cycle Manager** window, which is used to manage the driving cycles of the project. The window is divided into three parts.



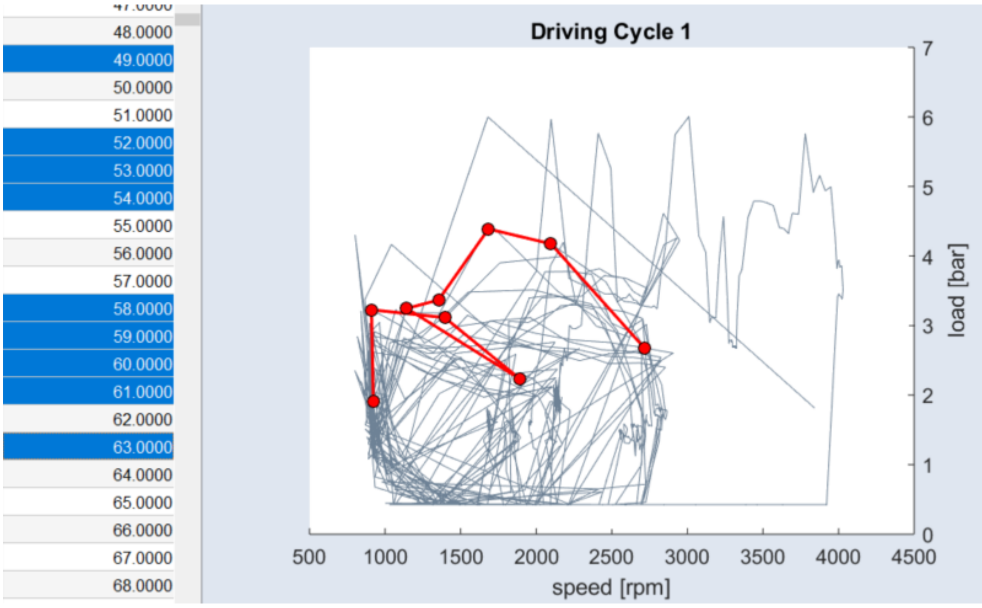
Via the **File** menu, you can **Load/Save Databases** or **Import/Export Driving Cycles**.

For further information see the online help (‹F1› or **Help › Online Help**).

"Driving Cycles" pane: The **Driving Cycles** pane on the left side shows a list of the driving cycles currently available in the project. One or more driving cycles can be selected.

"Data" table: The **Data** table in the middle part shows the selected driving cycles as data table.

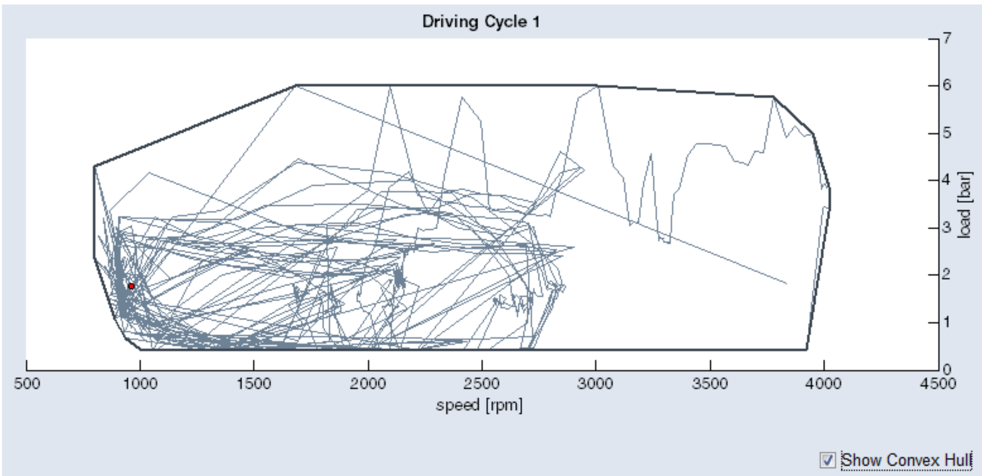
If you select a row in the data table representing a cycle, the respective point is shown as red dot in the plot on the right side. If desired, you can select several rows. In that case, the trace of those points is marked by a red line and the statistics part is updated by just considering the selected points.



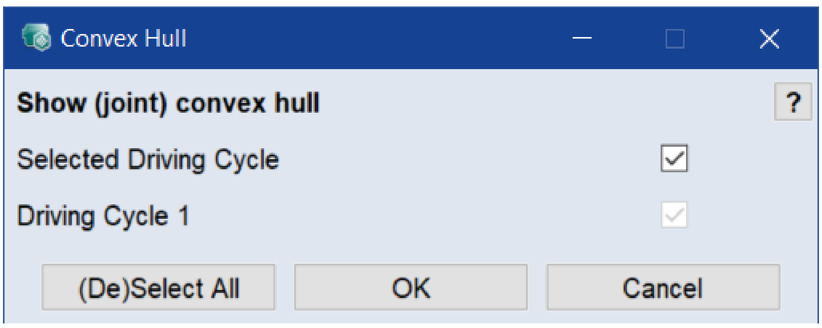
The driving cycles can be edited by editing or deleting rows in the table. Note that the edit mode is deactivated if you selected more than one driving cycle in the list on the left side. It is possible to select a point by clicking it in the plot.

Statistics area: On the right side, there are some statistical evaluations of the driving cycles as well as a plot of the cycles.

If you activate the **Show Convex Hull** checkbox, the convex hull of the selected cycles is plotted.



You can determine which driving cycles are used for the calculation of the convex hull via **View › Convex Hull Selection**.

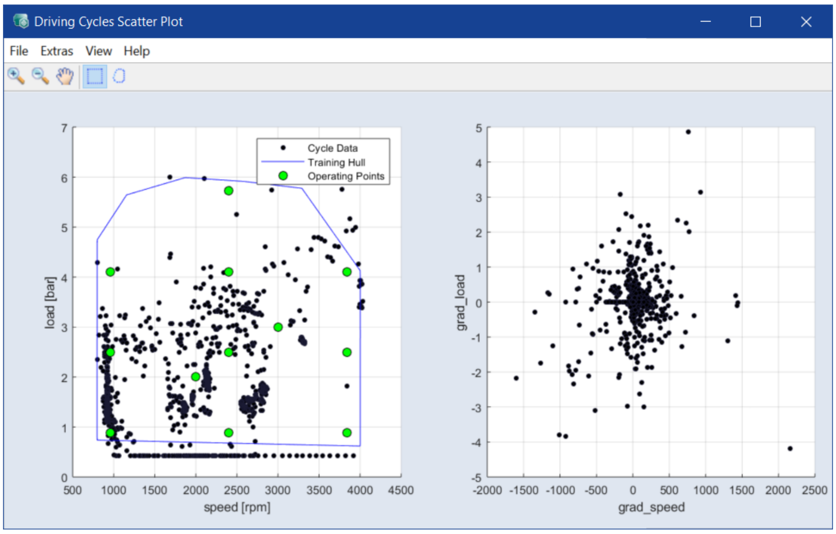


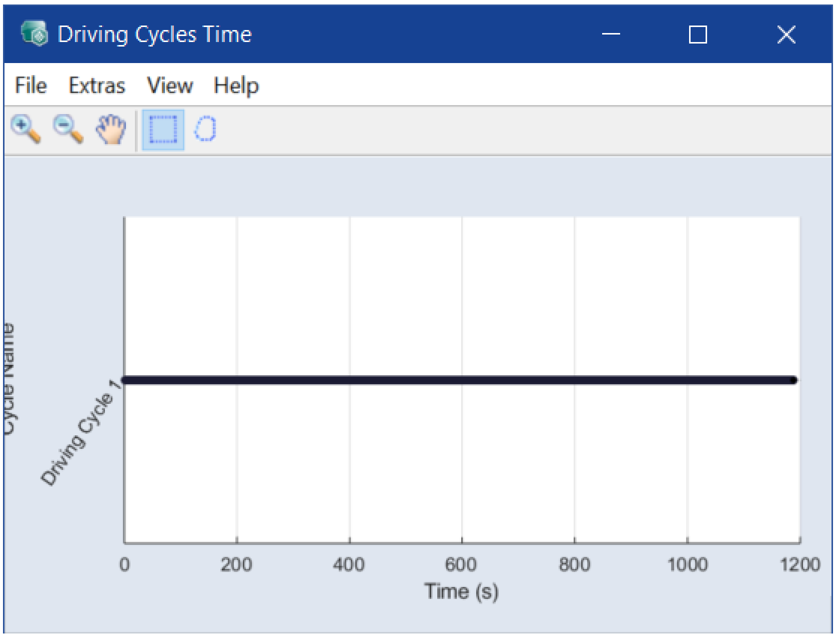
Visualizing driving cycle data in a scatter plot

1. In the **Driving Cycle Manager** window, select **View › Scatter Plot**.
2. In the **Show Scatter Plot** window, select one or more driving cycles to be plotted, then click **OK**.

   As an alternative, you can click the **Scatter Plot** button. In that case, the driving cycles selected in the **Driving Cycles** pane are plotted without inquiry.

   The **Driving Cycle Scatter Plot** window, showing a scatter plot of the operating point axes and the gradients of the selected driving cycles, and the **Driving Cycle Time** window open. You can draw a rectangle in these windows like in all scatter plots.





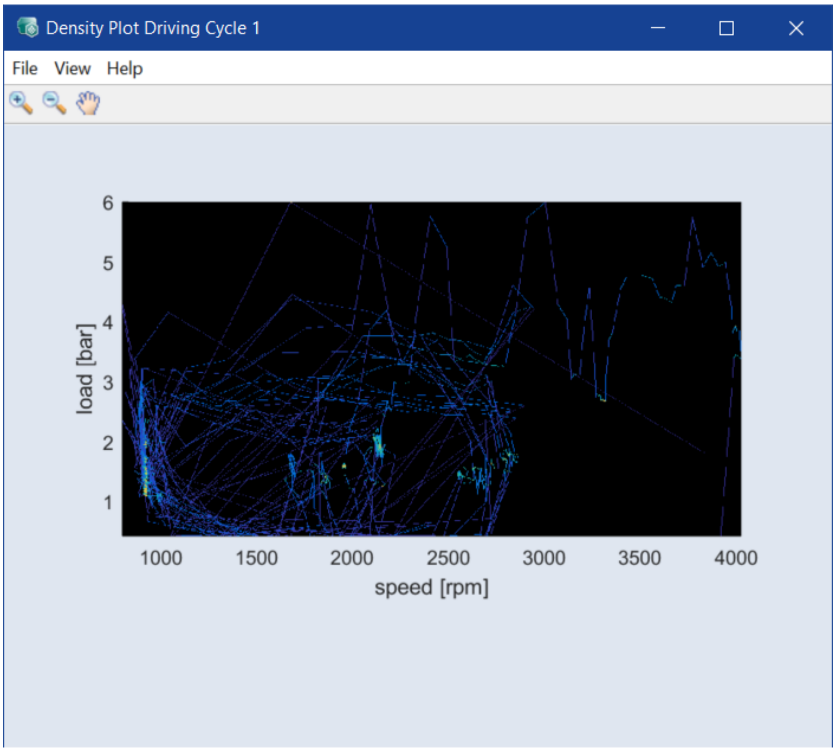
> **Note**
>
> Multiple driving cycles are plotted on the same scatter plot to better see the differences between them.

Visualizing driving cycle data in a density plot

1. In the **Driving Cycle Manager** window, select **View › Density Plot**.
2. In the **Show Density Plot** window, select one or more driving cycles to be plotted, then click **OK**.

   As an alternative, you can click the **Density Plot** button. In that case, the driving cycles selected in the **Driving Cycles** pane are plotted without inquiry.

⇨ A **Density Plot Driving Cycle** window opens for each selected driving cycle.

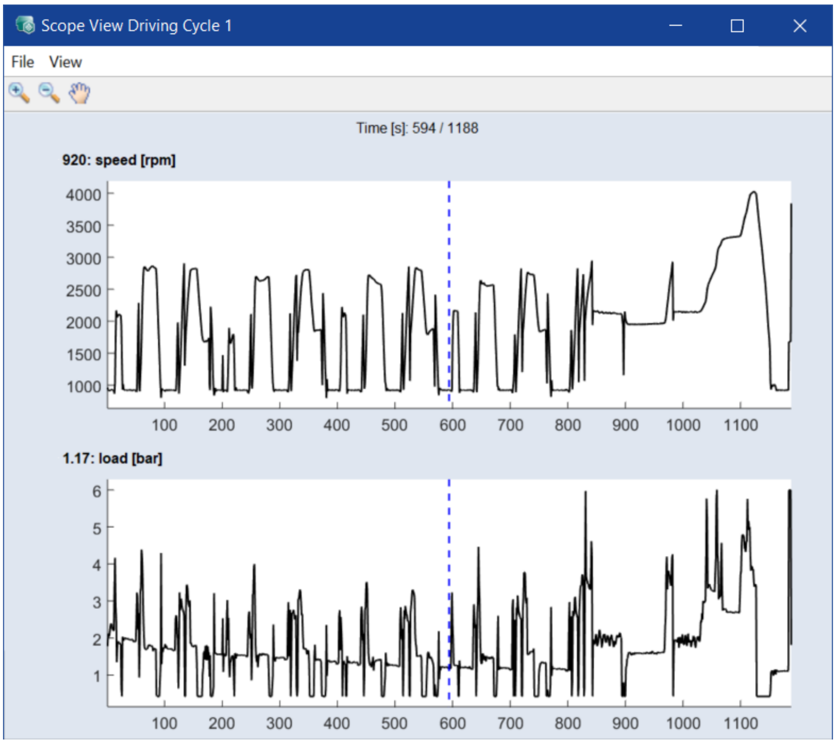


Visualizing driving cycle data in a scope view

1. In the **Driving Cycle Manager** window, select **View › Scope View**.
2. In the **Show Scope View** window, select one or more driving cycles to be plotted, then click **OK**.

   As an alternative, you can click the **Scope View** button. In that case, the driving cycles selected in the **Driving Cycles** pane are plotted without inquiry.

➪ A **Scope View Driving Cycle** window opens for each selected driving cycle.



Generating driving cycles

In many cases, the quality of a model (e.g. for emissions) can be validated by testing its performance on a set of various driving cycles.

Additionally and especially in the context of the Real Driving Emissions (RDE), it is interesting to evaluate models (e.g. for emissions) on a wide variety of different driving cycles in order to ensure that the whole drivable range is covered as good as possible.

ASCMO-STATIC allows the creation of any number of new virtual driving cycles that are based on a set of driving cycles that are already available, e.g. from real measurements from a test bench or the road. The generated driving cycles are similar to the already existing driving cycles and show their typical characteristics.

Note

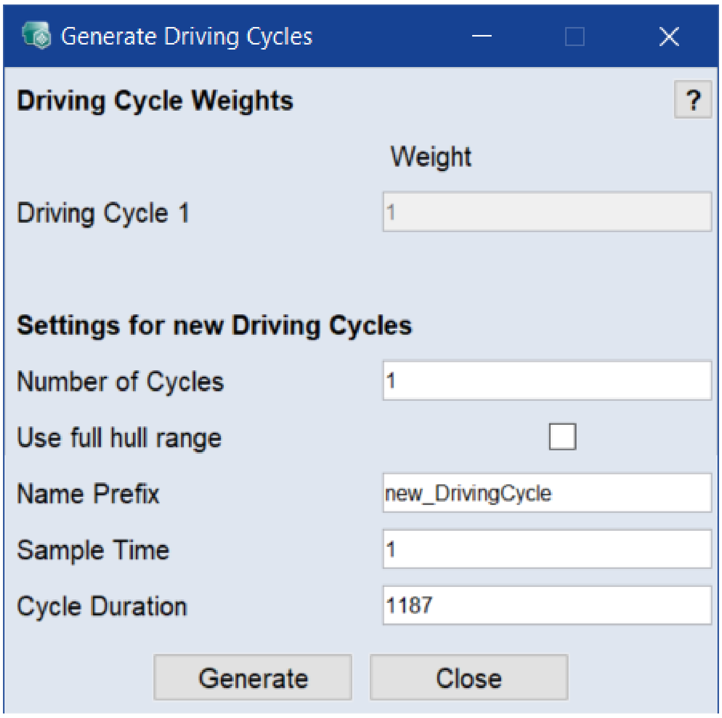
The generated driving cycles with ASCMO-STATIC (GUI: Calibration › Driving Cycles) or ASCMO-Cycle Generator are similar to the already existing driving cycles. They do not meet regulations or fulfill the properties of cycles which they are based on. The generation is based on a probability model (Markov chain).

Generated cycles are not intended for official purposes like RDE homologation or similar purposes.

1. In the **Driving Cycle Manager** window, select **Data › Generate Driving Cycles**.

   A dialog window opens that lists all available driving cycles.
2. Select the existing driving cycles you want to use as base for the new driving cycles.

   Click **OK** to continue.
3. The **Generate Driving Cycles** window opens.

4. Set the parameters according to your needs.

   See the ASCMO-STATIC online help for details on the parameters.
5. Click **Generate**.

➪ The new generated driving cycles appear in the **Driving Cycles** pane in the **Driving Cycle Manager** window.

Visualizing driving cycle traces in a calibration map

Visualizing driving cycle traces in a calibration map (opened via **Calibration › Calibration Maps › *** in the main menu) enables a representation of the detected range of values.

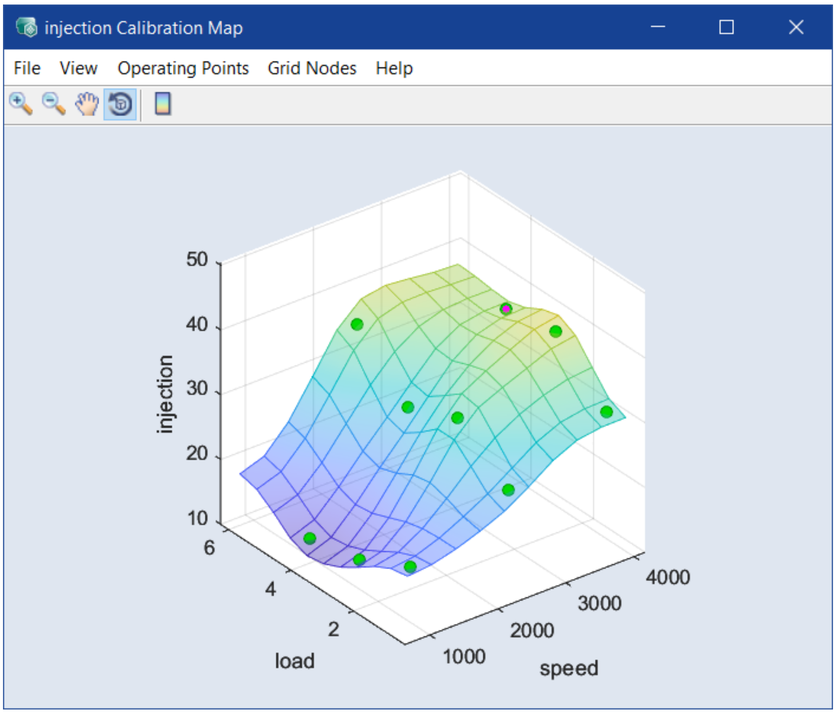
Note

The **Calibration** menu is only available, if operating point axes have been selected during data import (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards in the menu **In/Outputs › Set Operating Point Axes**.

1. In the desired **‹input› Calibration Map** window, select **View › Driving Cycle**.

   If your project contains several driving cycles, the **Driving Cycle in Map** selection window opens.
2. Select the driving cycle you want to display, then click **OK**.

⇨ The driving cycle is indicated by the blue dots in the calibration map.



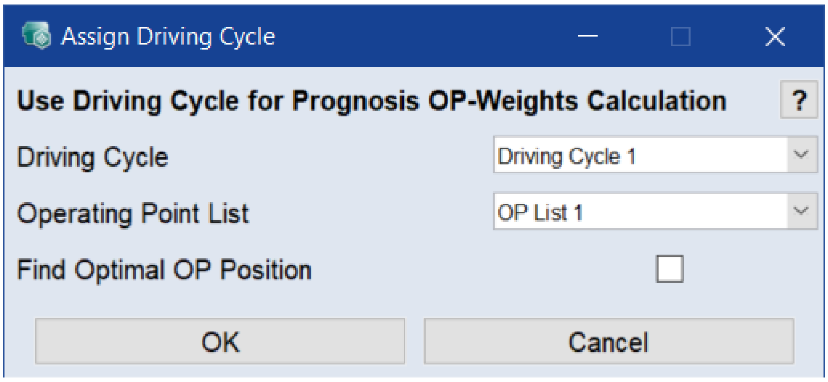
6.8.2 Defining Operating Point Weights Using Driving Cycle Traces

You can use one or more driving cycles to assign weights to the operating points.

Defining operating point weights using driving cycles

1. In the **Driving Cycle Manager** window, select **Data › Assign OP Weights using a Driving Cycle**.

   The **Assign Driving Cycle** window opens.

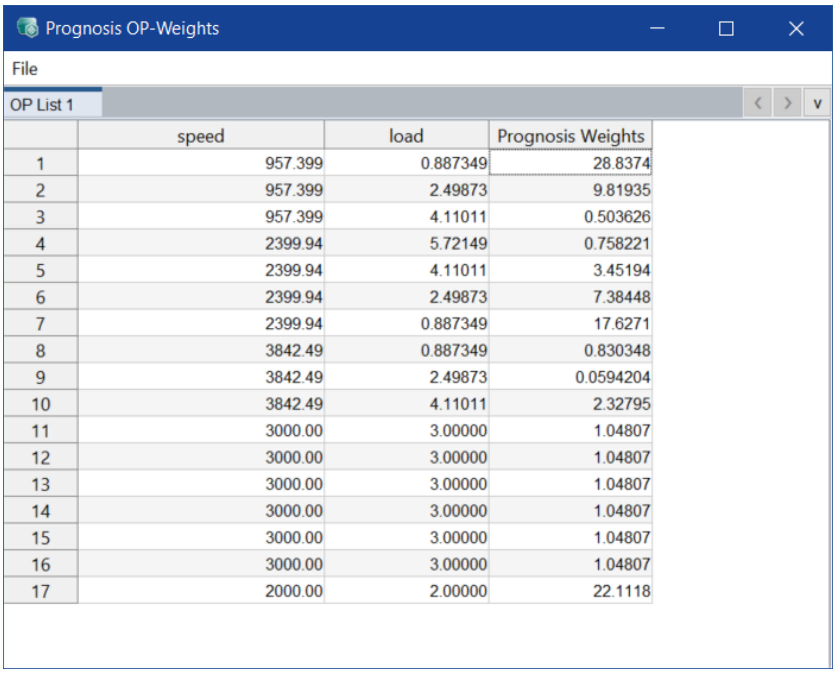


2. In the **Driving Cycle** drop-down list, select the driving cycle you want to use for weighting.
3. In the **Operating Point List** drop-down list, select the operating point list you want to be weighted.

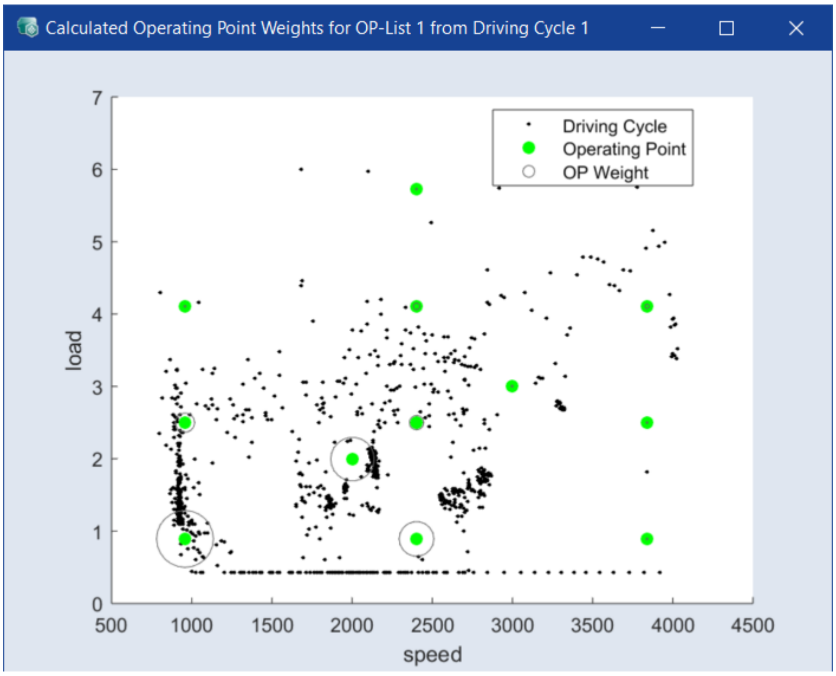
> **Note**
>
> For the purpose of this tutorial, you can accept the predefined settings.

4. Click **OK**.

   The **Prognosis OP-Weights** window and the **Calculated Operating Point Weights from Driving Cycle 1** window open.

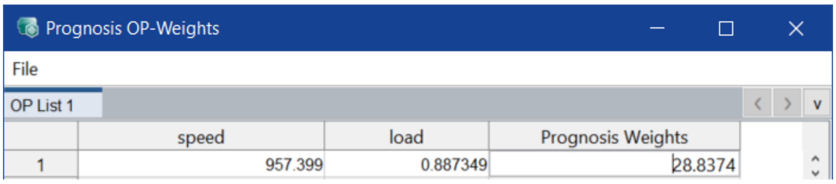

Prognosis OP-Weights
File
OP List 1

	speed	load	Prognosis Weights
1	957.399	0.887349	28.8374
2	957.399	2.49873	9.81935
3	957.399	4.11011	0.503626
4	2399.94	5.72149	0.758221
5	2399.94	4.11011	3.45194
6	2399.94	2.49873	7.38448
7	2399.94	0.887349	17.6271
8	3842.49	0.887349	0.830348
9	3842.49	2.49873	0.0594204
10	3842.49	4.11011	2.32795
11	3000.00	3.00000	1.04807
12	3000.00	3.00000	1.04807
13	3000.00	3.00000	1.04807
14	3000.00	3.00000	1.04807
15	3000.00	3.00000	1.04807
16	3000.00	3.00000	1.04807
17	2000.00	2.00000	22.1118



If desired, you can change the weights of the operating points defined by the driving cycle manually in the **Prognosis OP-Weights** window.

5. Enter the desired weighting value in the **Prognosis Weights** column.



6. Press <ENTER> to confirm the new weighting value.

6.8.3 Defining Operating Point Positions Using Driving Cycle Traces

In addition to the weighting, the positions of the operating points can be optimally adapted to driving cycle traces.

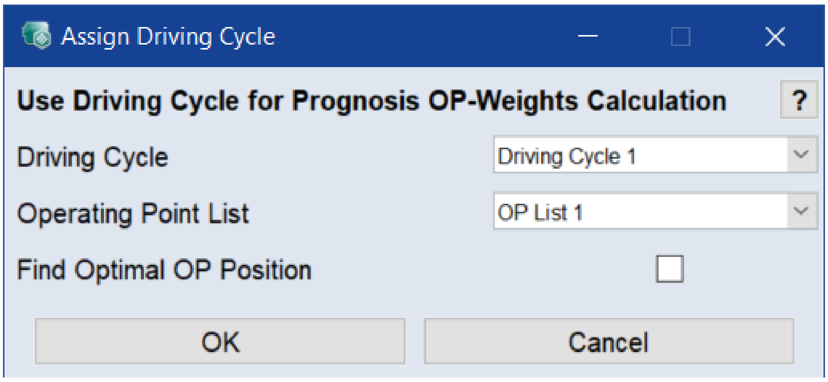
> **Note**
>
> The predetermined weighting and positioning of operating points will be deleted when you assign operating points to driving cycle traces.

Defining operating point positions using driving cycles

1. In the **Driving Cycle Manager** window, select **Data › Assign OP Weights using a Driving Cycle**.

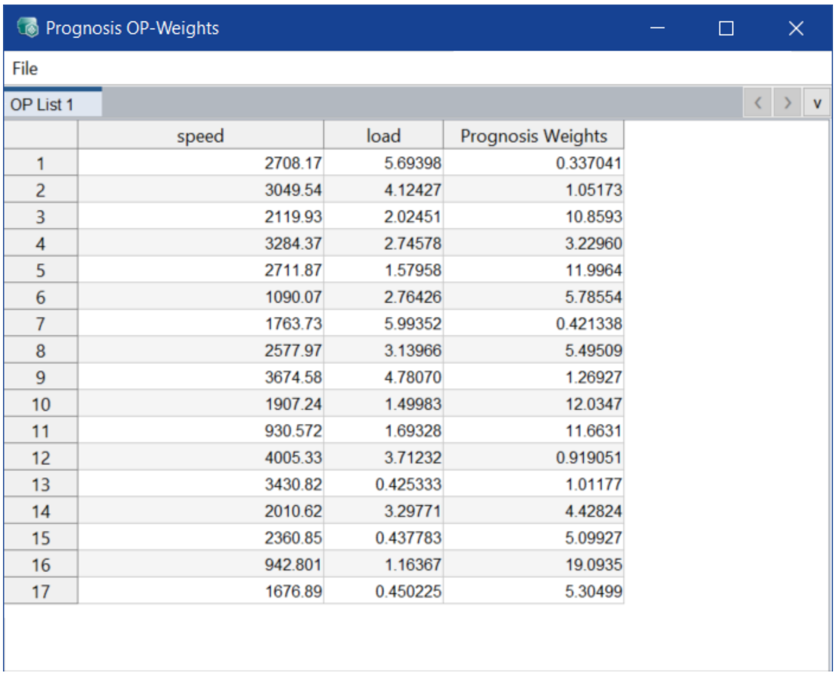
   The **Assign Driving Cycle** window opens.

2. Select the driving cycle and the operating points list you want to use.
3. Activate the **Find Optimal OP Position** checkbox.

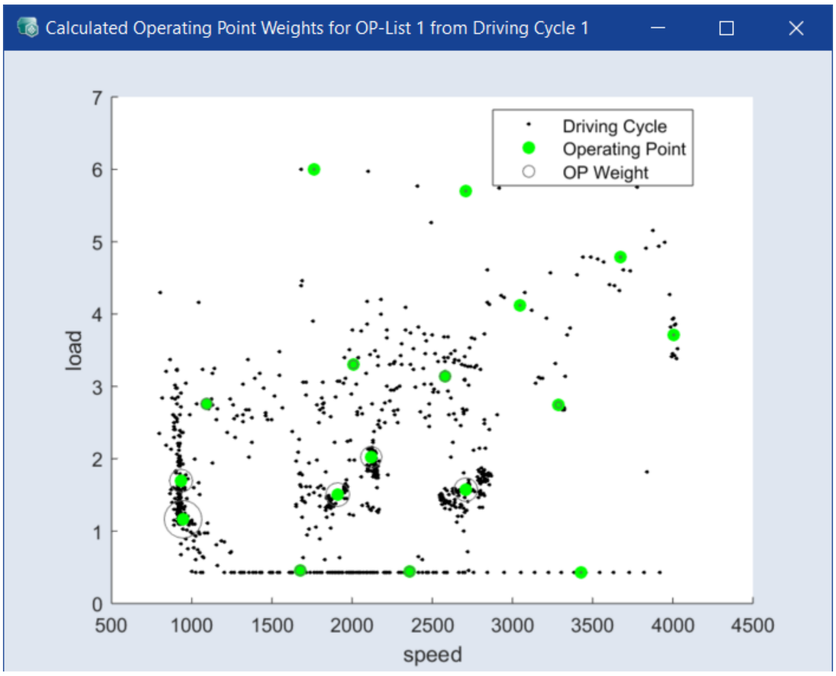


4. Click **OK** to perform the assignment.

⇨ The **Prognosis OP-Weights** window and the **Calculated Operating Point Weights from Driving Cycle 1** window open. The positions of the operating points are changed.



	speed	load	Prognosis Weights
1	2708.17	5.69398	0.337041
2	3049.54	4.12427	1.05173
3	2119.93	2.02451	10.8593
4	3284.37	2.74578	3.22960
5	2711.87	1.57958	11.9964
6	1090.07	2.76426	5.78554
7	1763.73	5.99352	0.421338
8	2577.97	3.13966	5.49509
9	3674.58	4.78070	1.26927
10	1907.24	1.49983	12.0347
11	930.572	1.69328	11.6631
12	4005.33	3.71232	0.919051
13	3430.82	0.425333	1.01177
14	2010.62	3.29771	4.42824
15	2360.85	0.437783	5.09927
16	942.801	1.16367	19.0935
17	1676.89	0.450225	5.30499



6.8.4 Prognosis With Cycle-Based OP-Weighting

With the cycle-based weighting of the operating points, you can already perform the prognosis without editing the calculation rules.

> **Note**
>
> The **Calibration** menu is only available, if operating point axes have been selected during data import (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards in the menu **In/Outputs › Set Operating Point Axes**.

Perform prognosis without editing calculation rules

1. Select **Calibration › Prognosis › Results**.

➩ The **Prognosis Results** window opens. The iterative process of optimization can now be performed.

For more information on how to perform the optimization, see 6.7 "Optimization " on page 120.

6.8.5 Calculation Rules for Cycle-Based Prognosis

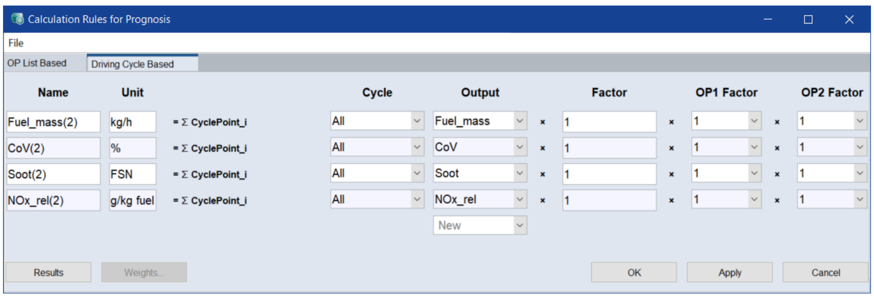
To obtain a cycle value from the stationary measure values, you may well have to make some conversions (e.g. time-based measuring to route-based prognosis). This chapter describes how to define Prognosis Calculation Rules (accessed via **Calibration › Prognosis › Calculation Rules** in the ISP view) using a driving cycle.

> **Note**
>
> The **Calibration** menu is only available, if operating point axes have been selected during data import (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards in the menu **In/Outputs › Set Operating Point Axes**.

Defining the conversion of units

1. Select **Calibration › Prognosis › Calculation Rules**.

   The **Calculation Rules for Prognosis** window opens. The window elements are described in the ASCMO-STATIC online help <F1>.
2. In the **Calculation Rules for Prognosis** window, select the **Driving Cycle Based** tab.
3. In the **Output** column, select all four existing outputs `Fuel_mass`, `CoV`, `Soot` and `NOx_rel` in succession.



4. Click **Apply**.

⇨ An information message appears in the log window that you have changed the prognosis parameters.

Importing the conversion of units

To import a calculation rule for the cycle forecast, proceed as follows.

1. In the **Calculation Rules for Prognosis** window, select **File › Import Settings (*.ini) › Template**.

   The **Load Prognosis Calculation Rule from ini** window opens.
2. Select the `*. ini` file that contains the corresponding calculation rules and click **Open**.

   The **Load Prognosis Calculation Rules** window opens.
3. Assign the template-based calculation rules you want to use to suitable outputs.
4. If desired, activate the **Clear All Before** checkbox.

   Note

   If **Clear All Before** is activated, all existing calculation rules are deleted during import.

   If **Clear All Before** is deactivated, the new calculation rules are added to the existing rules.
5. Click **Apply**.

⇨ In the log window, an information that you have changed the calculation rules appears.

6.8.6 Optimization on Driving Cycle Traces (Global Optimization)

The cycle-dependent weighting and positioning of the operating points with the reference to the calculation rules can now be used for the global optimization. For more information regarding global optimization, see 6.9 "Cycle-Based Global Optimization" on the next page.

6.8.7 Optimization on Driving Cycle Traces Manually

In addition to the automated global optimization, you have the possibility to optimize manually by calibrating the calibration maps. The data obtained from the optimization will be transferred to the calibration maps. Then the operating points of the new calibration maps may be changed in terms of optimization. The adjustment in the calibration maps will be displayed in the **Prognosis Results** window (opened via **Calibration › Prognosis › Results**) directly. For more information on how to calibrate calibration maps for optimization, see 6.7.5 "Calibration " on page 141.

Note

The **Calibration** menu and the **Global Optimization** function in the **Optimization** menu are only available if operating point axes have been selected (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes via **In/Outputs › Set Operating Point Axes**.

6.9 Cycle-Based Global Optimization

For most combustion engines of, for example, cars and trucks, you must be able to prove that legally prescribed thresholds for emissions are adhered to (e.g. in accordance with the Euro 6 standard) in various test cycles. The application's task is thus both to adhere to these thresholds and to attain the best possible consumption rates while adhering to any other constraints such as thresholds for pressure and temperature.

Some driving cycles are prescribed by the legislator as a list of stationary operating points; others are defined as transient driving cycles via speed and load. To be able to make reliable predictions about dynamic results from stationary information, it is common practice to reduce the transient driving cycles to a list of operating points that are as representative as possible.

This section of the tutorial shows how to perform cycle-based global optimization with ASCMO-STATIC, using the model of a diesel engine as an example.

6.9.1 Optimization Problem

The diesel engine model used in cycle-based global optimization has the following inputs and outputs:

Inputs	Operating Points	Outputs
`Airmass` [mg/Str]	`Speed` [1/min]	`NOx` [g/h]
`SOI` [° CA]	`Load` [bar]	`Particle` [g/h]
`pRail` [bar]		`Noise` [dB]
`qPilot1` [mm3/Str]		`Fuelmass` [kg/h]
`qPilot2` [mm3/Str]		`BSFC` [g/kWh]
`tiPilot1` [µs]		
`tiPilot2` [µs]		
`Swirl` [%]		

Modeling Data

The sample diesel engine model used for cycle-based global optimization is in the file `Example_Diesel_Engine_Model.ascmo` in the directory `<installation>\Example\AscmoStatic` (By default, *<installation>* is C:\Program Files\ETAS\ASCMO x.x)

The following files are also part of this tutorial step:

- `Example_Diesel_Engine_OP_List.xls`

  This file contains a list of operating points where optimization is to take place.
- `Example_Diesel_Engine_OP_Weights.xls`

  This file contains the weighting of the individual operating points used for the driving cycle forecast.

Opening a sample model

1. In the start screen of ASCMO-STATIC, click **Open Demo Project** or **Open ASCMO Project**.
2. In the file selection window, select the file `Example_Diesel_Engine_Model.ascmo` from the *<installation>*`\Example\AscmoStatic` directory.

   The model opens. The model is write-protected so it can be reused without any changes having been made to it.
3. To save your changes, save the original model under a different name.

6.9.2 Defining the Operating Points to be Optimized and their Weighting

The operating points to be optimized can be entered manually, but they can also be read in from a file (that can also already contain the weighting of the relevant operating point, see "Defining the weighting of the operating points" on the next page).

Note

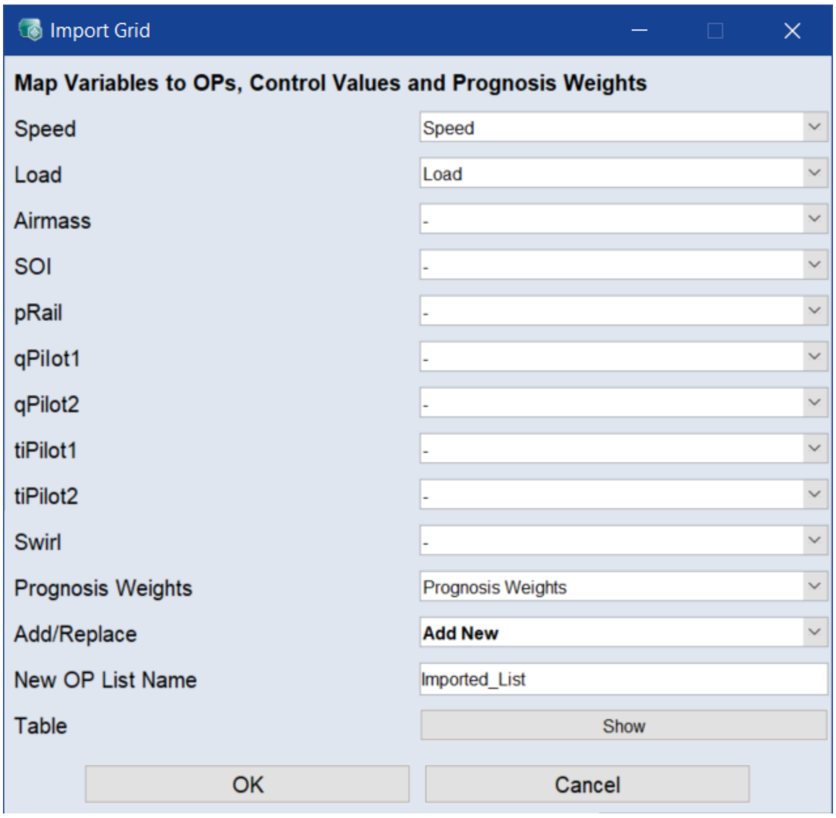
The **Calibration** menu is only available, if operating point axes have been selected during data import (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards in the menu **In/Outputs › Set Operating Point Axes**.

Defining the operating points for optimization

1. Select **Calibration › Operating Points**.

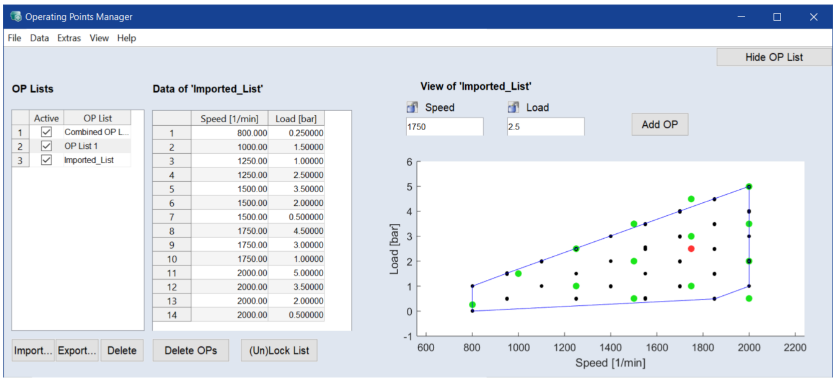
   The **Operating Points Manager** window opens.
2. If necessary, click **Show OP List** to display the operating points lists.
3. To load a list of predefined operating points, select **File › Import**.

4. In the file selection window, select the file `<installation>\Example\AscmoStatic\Example_Diesel_Engine_OP_List.xls`.
5. In the **Import Grid** window, make the following assignments:



6. Click **OK**.

   The list is imported. It is displayed in the **Data of '<list name>'** table and the **View of '<list name>'** plot. Existing lists are kept; the **Combined OP List** contains the merged content of all lists.



7. Alternatively, you can create an operating point grid using **Data > Add New List > ***.

<u>Defining the weighting of the operating points</u>

The weighting of the operating points can be done manually (just like the definition of the operating points themselves) or by reading in the values from a file.

1. Select **Calibration › Prognosis › Weights**.

   The **Prognosis OP-Weights** window opens.

2. Go to the tab with the list you want to edit.

   In this tutorial, you will edit **Imported_List**.

3. In the **Prognosis Weights** column, enter the required values.

Or

4. Select **File › Import**.
5. Select the file `Example_Diesel_Engine_OP_Weights.xls`.

   The **Operating Point Prognosis Weights Import** window opens.

6. Click **OK** to accept the preselected settings.

➪ The information is read in from the file.

The weighting of the operating points can also be calculated by measuring a trajectory (**Data › Assign OP Weights using a Driving Cycle** in the Driving Cycle Manager). This involves every time step in the measurement being assigned to an operating point, and then totaled, which results in the weighting for each of the operating points defined previously. For more information on how to use driving cycle traces for the definition of the calculation rules, see 6.8 "Driving Cycle Forecast " on page 147.

Using several lists of operating points

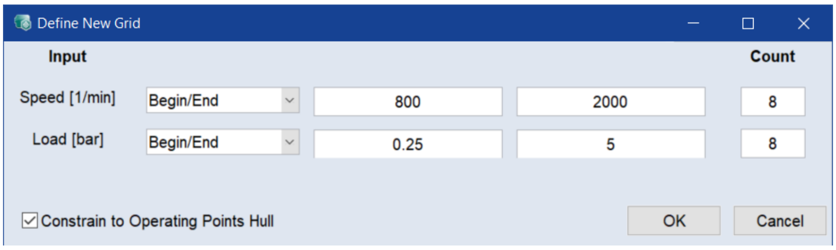
To work with different sets of operating points, several lists can be set up.

1. Select **Calibration › Operating Points**.

   The **Operating Points Manager** window opens.

2. If necessary, display the operating points lists.
3. To set up an additional list, select **Data › Add New List › Define Grid**.

   The **Define New Grid** window opens.

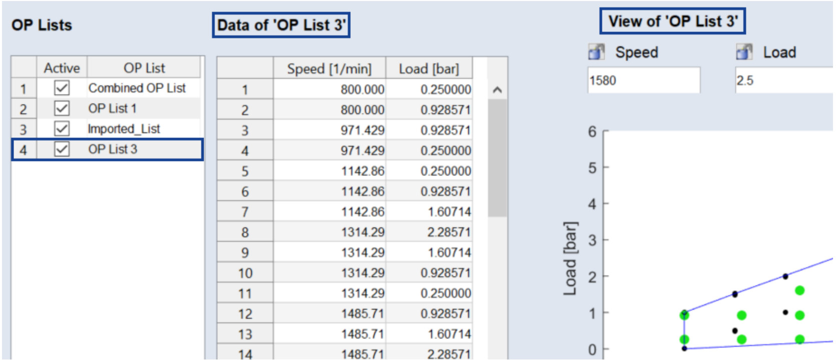


4. Enter the number of operating points per axis.

   For further information, see the online help <F1>.

5. Click **OK**.

➪ A new list is set up and shown in the operating points manager (see below) and in a separate window. The list is selectable separately or in combination with other lists.



If several lists are available, these can be used for cycle forecast hereafter.

6.9.3 Calculation Rules for Cycle Forecast

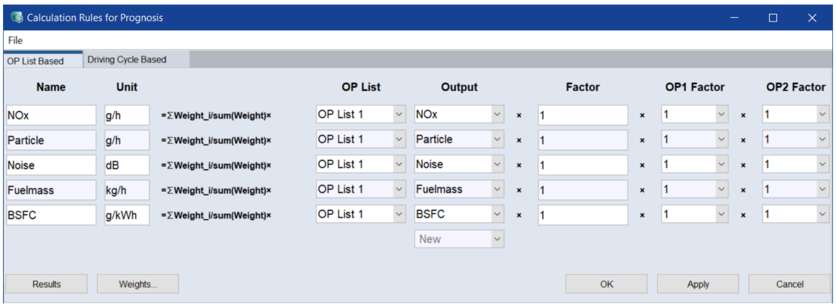
To obtain a cycle value from the stationary measure values, you may well have to make some conversions (e.g. time-based measuring → route-based prognosis).

> **Note**
>
> The **Calibration** menu is only available, if operating point axes have been selected during data import (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards in the menu **In/Outputs › Set Operating Point Axes**.

Defining the conversion of units

1. Select **Calibration › Prognosis › Calculation Rules**.

➪ The **Calculation Rules for Prognosis** window opens.



The meanings of the fields are described in the ASCMO-STATIC online help (‹F1›).

6.9.4 Defining Parameters for Optimization

Before you start optimizing, you should first define the permissible range of parameter variation for all variables.

Note

The **Calibration** menu is only available, if operating point axes have been selected during data import (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards in the menu **In/Outputs › Set Operating Point Axes**.

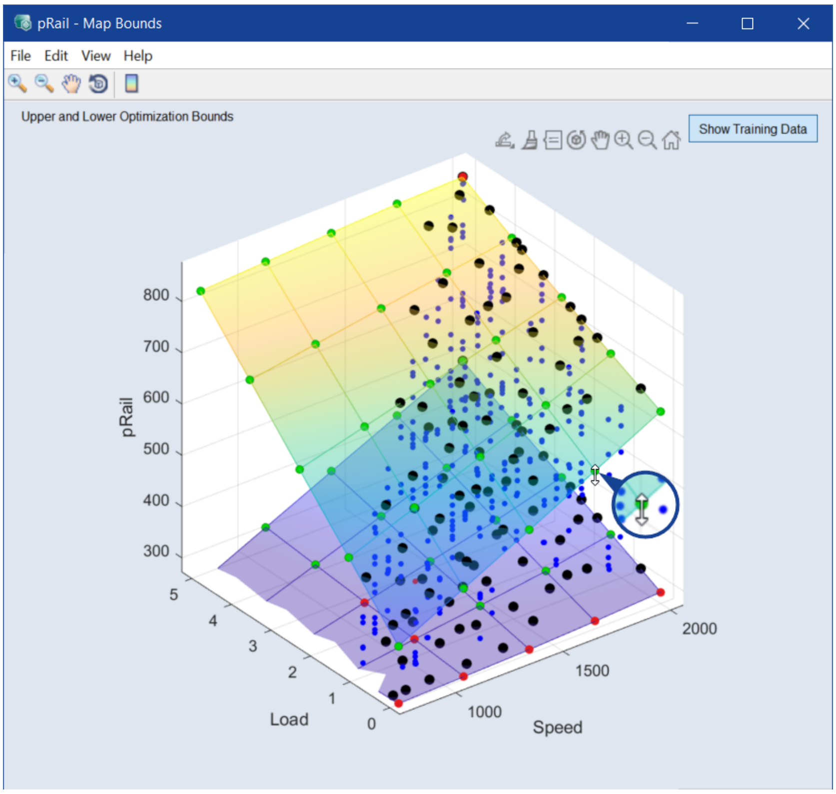
Limiting areas for parameter variation

1. Select **Calibration › Map Bounds over OP › Fit Bounds to Data**.

   All **<input> - Map Bounds** windows and the **Fit Map Bounds to Data** window open. In the latter, **Apply to all Maps** is activated.
2. In the **Fit Map Bounds to Data** window, do one or more of the following:
   - Enter a smoothness factor for the map bounds.
   - In the **Grid Nodes** area, refine the grid.
3. Click **OK** or **Apply** to continue.

   The valid range for all inputs is adapted to the range actually measured. Then, parameters can only be adjusted at all operating points within the measured range.
4. To change the limit manually, click one of the (green) operating points.

⇨ The mouse button becomes a double arrow and the limit can be moved up and down.



Parameterizing optimizers

1. Select **Optimization › Global Optimization**.

   The **Global Optimization** window opens.

2. Enter the required thresholds for pollutant emissions in the **Sum Criteria** area:

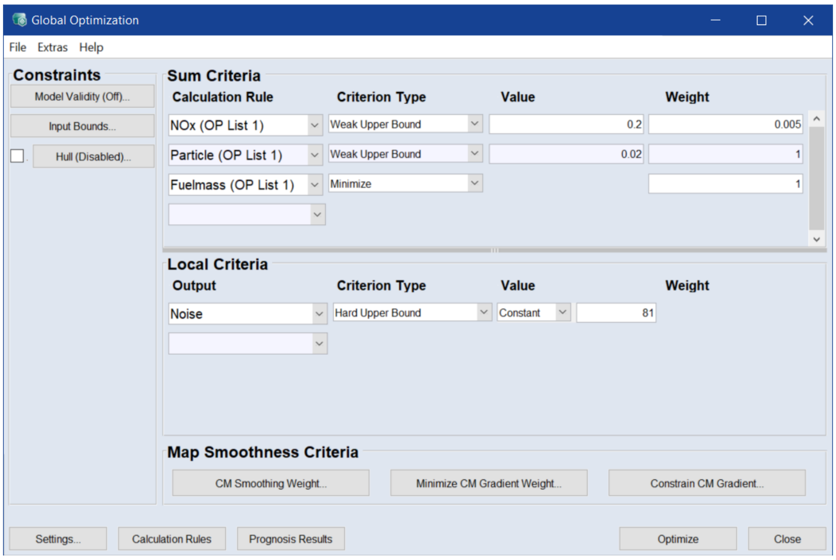
   - `NOx`: Weak Upper Bound / 0.2 g/km
   - `Particle`: Weak Upper Bound / 0.02 g/km

3. Specify the proper calculation rules for the outputs.

   The conversion factor for pollutant emissions measured in the European driving cycle in g/h to g/km can be found in the ASCMO-STATIC online help.

4. Generally, fuel consumption should be minimal – select `Minimize` as criterion for the `Fuelmass` output.

   Further criteria can also be defined, e.g. that engine noise must not exceed 80 dB. This is specified as a **Local Criterion** (Hard Upper Bound / 80 dB) and can be set either for all operating points (**Value** = `Constant`) or for each individual operating point (**Value** = `per OP`).



5. To limit the optimization result to the range of the valid model output, proceed as described in "Limiting the optimization results" on page 123.

➪ The individual optimization criteria are weighted (= prioritized).

Weighting the optimization criteria

1. In the **Weight** column of the **Sum Criteria** area, enter weights for the sum criteria.

2. In the **Weight** column of the **Local Criteria** area, enter weights for the local criteria.

   A higher number means a higher weight.

> **Note**
>
> You cannot enter a weight for criteria of type `Hard Upper Bound` or `Hard Lower Bound`.

6.9.5 Perform Optimization

Once all parameters and conditions have been defined, optimization can be carried out.

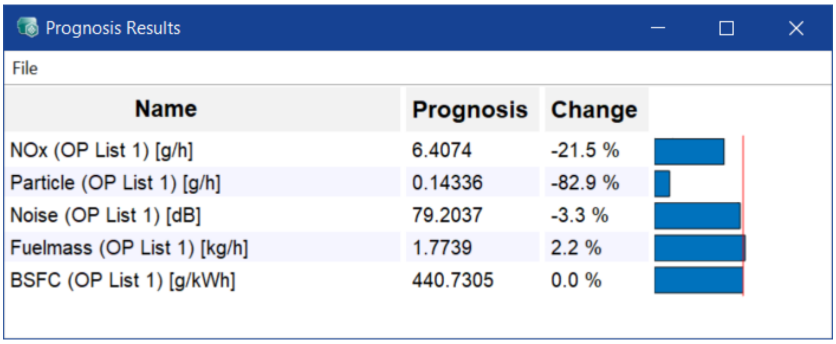
> **Note**
>
> The **Calibration** menu is only available, if operating point axes have been selected during data import (see 6.3.3 "Assign Inputs and Outputs" on page 90). You can set the operating point axes afterwards in the menu **In/Outputs › Set Operating Point Axes**.

Running the optimization

1. Before you start the optimization, open the **Prognosis Results** window (cf. "Perform prognosis without editing calculation rules" on page 160).

   This window shows the forecast for all output values before optimization with the current (non-optimized) application parameters.
2. In the **Global Optimization** window, click **Optimize** to perform the optimization.

   The result is shown in the **Prognosis Results** window after optimization.

3. Edit the weighting of `Fuel_mass` or the other criteria and repeat the optimization.

➪ The prognosis results change.

Checking calibration maps

1. Select **Calibration › Calibration Maps › Open all Maps** to check whether the maps are acceptable as they are or whether they have to be smoother.
2. If necessary, do the following to increase map smoothing:

   i. In the **Global Optimization** window, click **CM Smoothing Weight**.
   ii. In the **Map Smoothing** window, adjust the values for smoothing the maps.

       You can enter values between 0 (no smoothing) and 1 (very strong smoothing).
3. If necessary, do the following to minimize map gradients:
   i. In the **Global Optimization** window, click **Minimize CM Gradient Weight**.
   ii. In the **Minimization of Calibration Map Gradient** window, adjust the weights for minimizing the map gradients for both axes.

       You can enter values between 0 (no minimization) and 1 (very strong minimization).
4. If necessary, do the following to limit map gradients:
   i. In the **Global Optimization** window, click **Constrain CM Gradient**.
   ii. In the **Limitation of Calibration Map Gradient** window, activate the checkboxes for each input whose gradients you want to limit.

       You can limit the gradients on each operating point axis separately.
   iii. In the input fields, enter the lower and upper limit for the gradients of each input and operating point axis.
5. Repeat the optimization.

This process of optimization, checking and changing weighting is carried out until the result corresponds to the requirements.

6.10 Model Export

In this section you will learn how to export the ASCMO-STATIC models to:

> **Note**
>
> An export of calculated models is not possible.

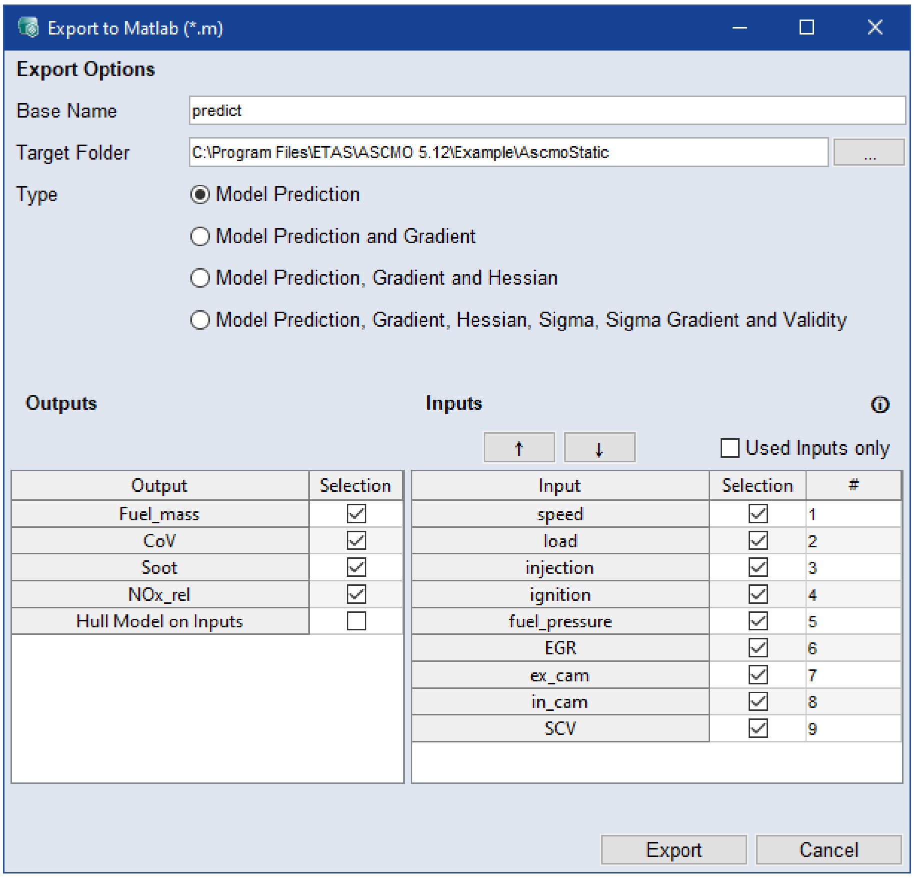
6.10.1 Export to MATLAB®

When you export models to MATLAB®, an M file is generated for each output.

Export a model to MATLAB®

1. **File › Export Model › Matlab**.

   Export to Matlab window



2. Set the **Base Name** (defines the file name *<base name>_<output>*) and the **Target Folder**.
3. Select the **Type** of data you want to export.
4. Select the **Inputs** and **Outputs** to export in the table. The order in which the inputs are passed to the exported model can be changed using the ↑ and ↓ buttons.

   You can use the standard CTRL/SHIFT selection functions in the table, or click and hold LMB and drag the cursor over the cells/rows you want to select. The position of the inputs in the exported model is shown in the **#** column.

   To export only used inputs, select **Used Inputs Only**.
5. Click **Export**.

➪ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

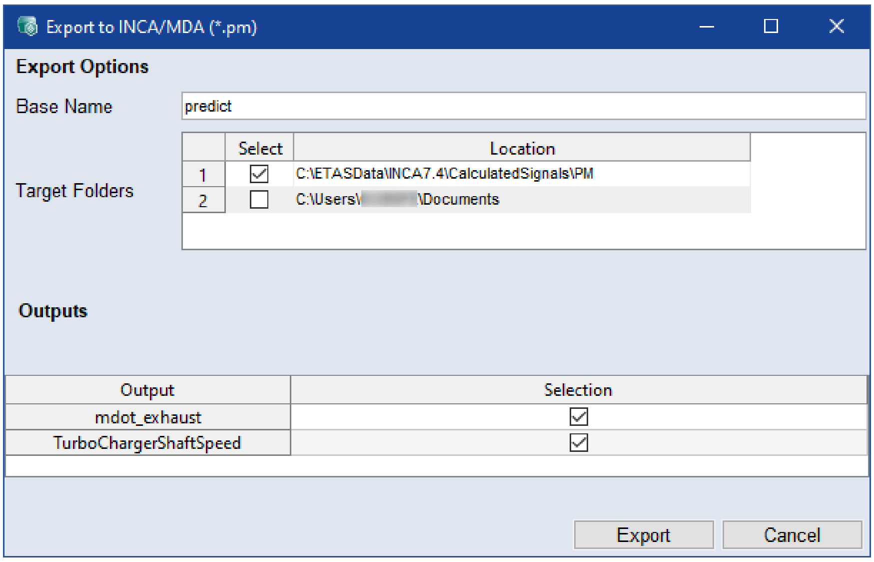
6.10.2 Export to INCA/MDA

The export to INCA/MDA exports the models as "Calculated Signals" for INCA/MDA 6.x. For MDA 8.x . These calculated signals are perl modules (`*.pm`) that INCA/MDA expects in a specific directory.

Export a model to INCA/MDA

1. **File › Export Model › INCA/MDA**.

   Export to INCA/MDA window

2. Set the **Base Name**, which defines the file names (*`<base name>_<output>`*).
3. Select the **Target Folder** from the list of available locations and the corresponding INCA or MDA version.

   > **Note**
   >
   > For the export to INCA/MDA, at least version 6.2 is required. Older versions are not available for selection.

4. Select the **Outputs** to export from the table.
5. Click **Export**.

⇨ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

> **Note**
>
> For information on how to use the exported data in INCA/MDA, please refer to the respective user manuals. The manuals are available from the ETAS Download Center.

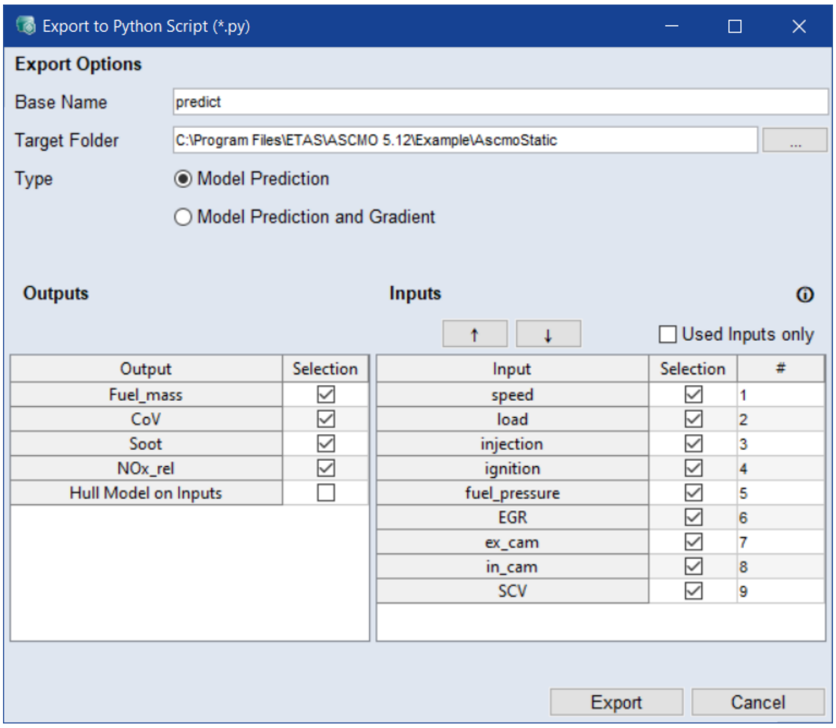
6.10.3 Export to Python Script

When you export models to Python, a `*.py` file is generated for each output, as well as a file containing the call as an example (`ModelEvaluationExample.py`). Depending on the export type, other files are also generated.

Export a model to Python Script

1. **File › Export Model › Python**.

   Export to Python window



2. Set the **Base Name** (defines the file name *<base name>_<output>*) and the **Target Folder**.
3. Select the **Type** of data you want to export.
4. Select the **Inputs** and **Outputs** to export in the table. The order in which the inputs are passed to the exported model can be changed using the ↑ and ↓ buttons.

   You can use the standard CTRL/SHIFT selection functions in the table, or click and hold LMB and drag the cursor over the cells/rows you want to select. The position of the inputs in the exported model is shown in the **#** column.

   To export only used inputs, select **Used Inputs Only**.
5. Click **Export**.

   ➪ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

6.10.4 Export to Simulink® Model

In addition to the Simulink® model file (`*.mdl` or `*.sxl`), an m-S function and TLC file per output is generated.

A Simulink® model is version-specific. If you have more than one version of Simulink® installed on your computer, ASCMO-STATIC will, by default, use the most recently installed version for the export, even if newer versions have been installed. However, you can select a specific version for export.

Select the Simulink® version for export

1. **File > Options**.
2. From the **Simulink Version** drop-down list, select the version you want to use.

   Last installed selects the version most recently installed on your computer, regardless of whether newer versions have been installed before.
3. Click **OK**.

Export a model to Simulink®

1. **File > Export Model > Simulink Model**.
2. Set the **Target File**: Defines the location path and the file name (*<file name>_<model name>*) for the export file.
3. Select the **Inputs** and **Outputs** to export in the table.

   To export only used inputs, select **Used Inputs Only**.
4. Click **Export**.

⇨ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

6.10.5 Export to Simulink® Script

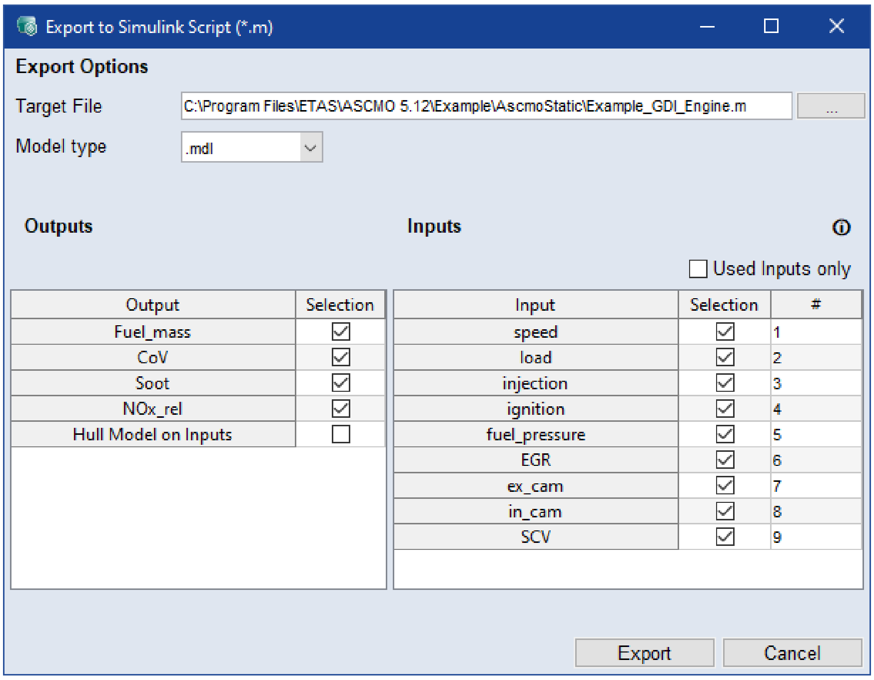
A Simulink installation is not required to perform the export.

When you export a model to **Simulink Script**, an M script file (`*.m`) and a TLC file are generated for each output. The script can later be used to create a Simulink model.

Export a model to Simulink® script

1. **File > Export Model > Simulink Script**.

   Export to Simulink Script (*.m)window

2. Set the **Target File**: Defines the location path and the file name (*<file name>_<model name>*) for the export file.
3. Set the **Model Type**: Specifies the model type of the export file (`*.mdl` or `*.slx`) created when the script is executed.
4. Select the **Inputs** and **Outputs** to export in the table.

   To export only used inputs, select **Used Inputs Only**.
5. Select **Outputs** and **Inputs**. Optionally, change the inputs order and use "Used Inputs only" checkbox.
6. Click **Export**.

⇨ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

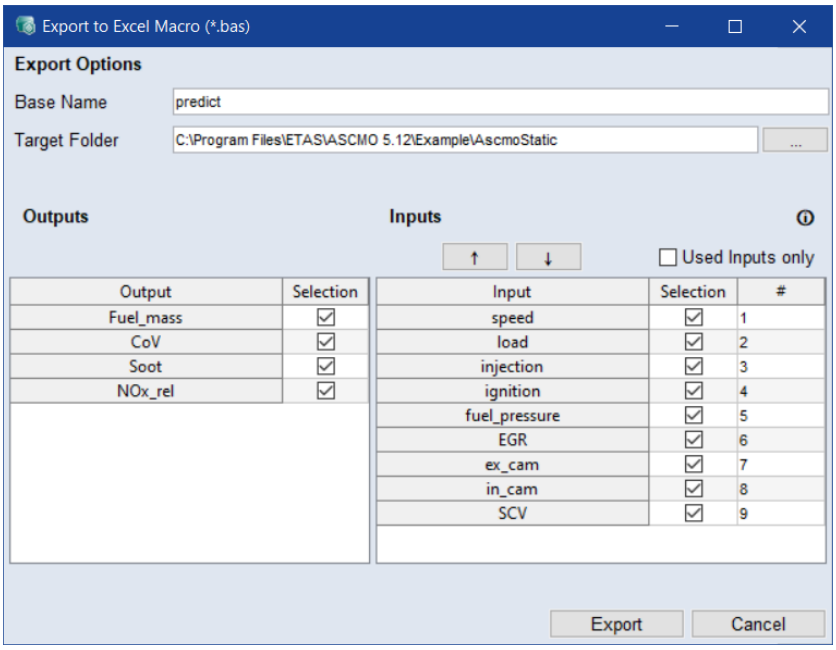
6.10.6 Export to Excel Macro

When you export models to Excel Macro, a VBA script (`*.bas`) is generated for each output.

Export a model to Excel Macro

1. **File > Export Model > Excel Macro**

   Export to Excel Macro window

2. Set the **Base Name** (defines the file name *<base name>_<output>*) and the **Target Folder**.
3. Select the **Type** of data you want to export.
4. Select the **Inputs** and **Outputs** to export in the table. The order in which the inputs are passed to the exported model can be changed using the ↑ and ↓ buttons.

   You can use the standard CTRL/SHIFT selection functions in the table, or click and hold LMB and drag the cursor over the cells/rows you want to select. The position of the inputs in the exported model is shown in the **#** column.

   To export only used inputs, select **Used Inputs Only**.
5. Click **Export**.

➩ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

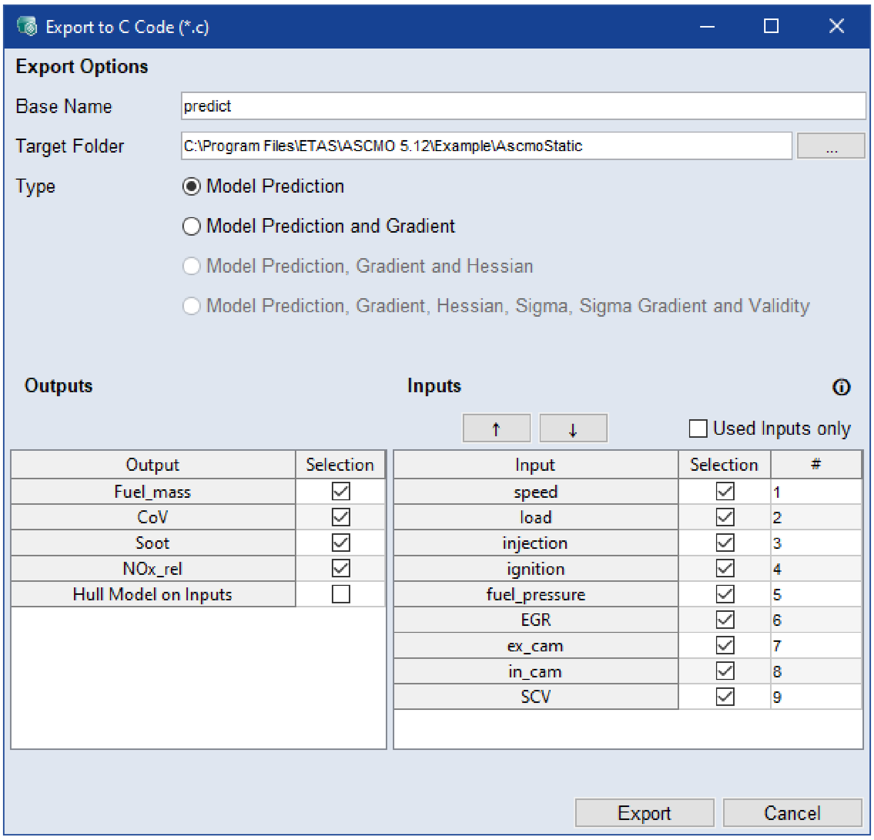
6.10.7 Export to C Code

When you export models to C code, a `*.c` file is generated for each output, as well as model-specific C files.

Export a model to C code

1. **File > Export Model > C Code**

   Export to C Code window



2. Set the **Base Name** (defines the file name *<base name>_<output>*) and the **Target Folder**.
3. Select the **Type** of data you want to export.
4. Select the **Inputs** and **Outputs** to export in the table. The order in which the inputs are passed to the exported model can be changed using the ↑ and ↓ buttons.

   You can use the standard CTRL/SHIFT selection functions in the table, or click and hold LMB and drag the cursor over the cells/rows you want to select. The position of the inputs in the exported model is shown in the **#** column.

   To export only used inputs, select **Used Inputs Only**.
5. Click **Export**.

➪ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

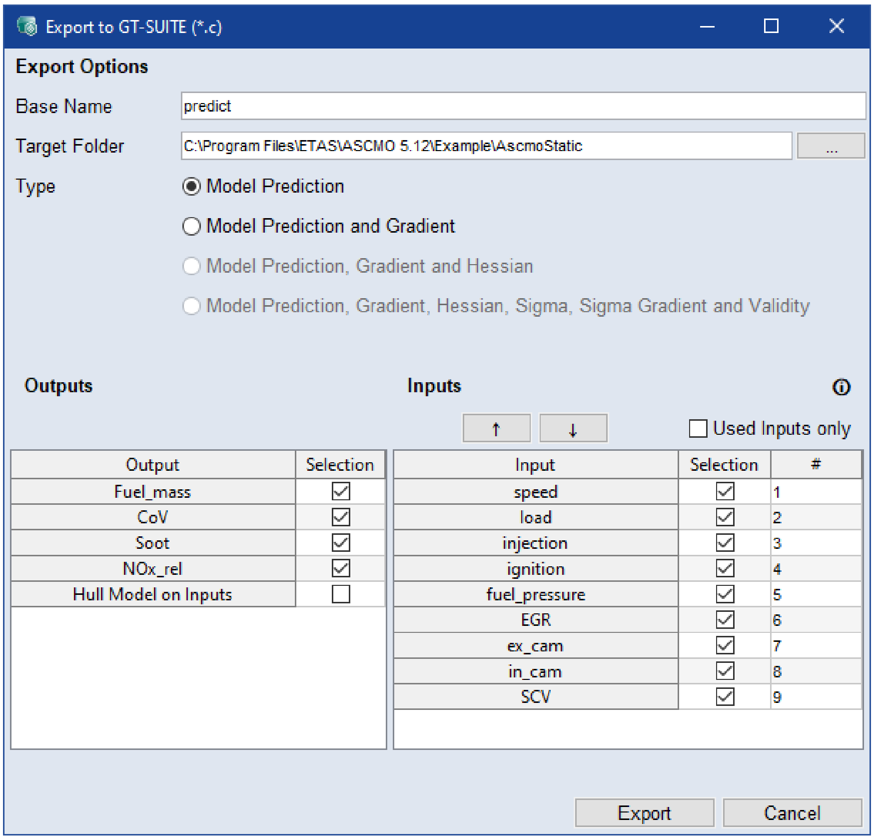
6.10.8 Export to GT-SUITE

The model export to GT-SUITE is an extended version of the C code export that allows you to load and use ASCMO-STATIC models directly in GT-SUITE. When exporting model outputs to GT-SUITE, a C file is generated for each model output.

Export a model to GT-SUITE

1. **File › Export Model › GT-Suite**.

   Export to GT-SUITE (*.c) window



2. Set the **Base Name** (defines the file name `<base name>_<output>`) and the **Target Folder**.
3. Select the **Type** of data you want to export. For more information on the export types see Exporting the Model to GT-SUITE.
4. Select the **Inputs** and **Outputs** to export in the table. The order in which the inputs are passed to the exported model can be changed using the ↑ and ↓ buttons.

   You can use the standard CTRL/SHIFT selection functions in the table, or click and hold LMB and drag the cursor over the cells/rows you want to select. The position of the inputs in the exported model is shown in the **#** column.

   To export only used inputs, select **Used Inputs Only**.

5. Click **Export**.

⇨ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

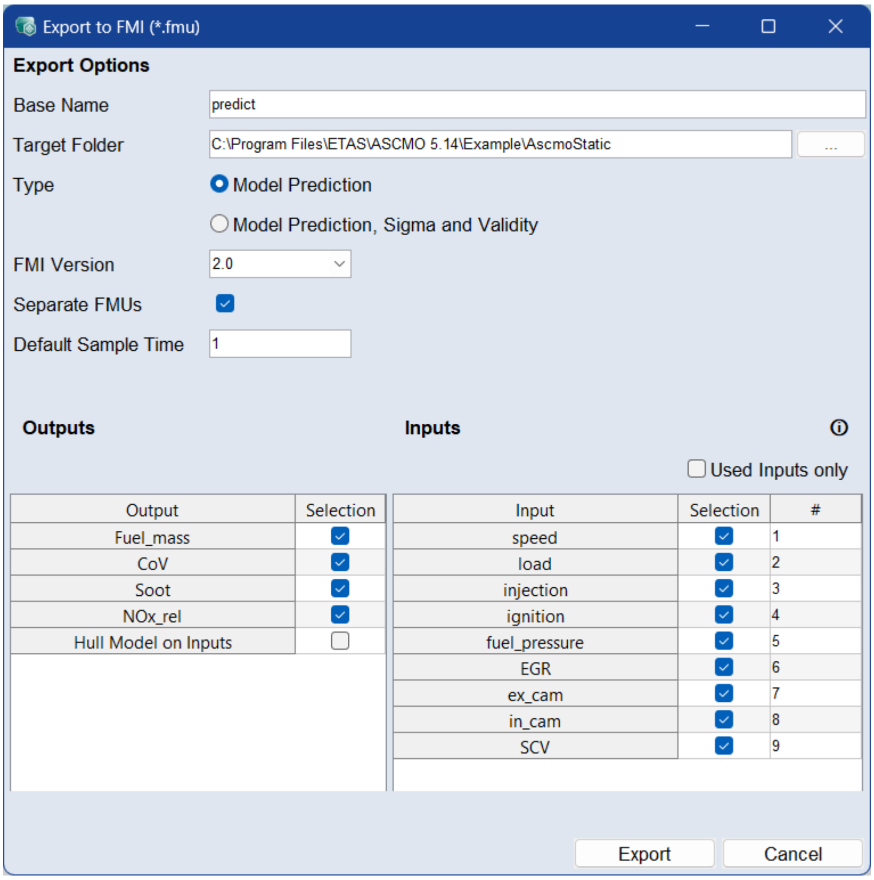
6.10.9 Export to FMI

When you export a model to FMI, a C file and a DLL file for Windows (32/64 bit) are generated for each output with a corresponding XML meta-description for the input and output specification. For an output named `Y1`, a file `predict_Y1.fmu` is generated.

<u>Export a model to FMI</u>

1. **File › Export Model › FMI**.

   Export to FMI (*.fmu) window



2. Set the **Base Name** (defines the file name *<base name>_<output>*) and the **Target Folder**.
3. Select the **Type** of data you want to export.
4. Select the **FMI Version** to which the model is exported.
5. Deselect the **Separate FMUs** checkbox to export the selected outputs as a single FMU file.
6. Set the default sample time for the experiment in seconds, e.g. 0.1 or 0.01.

7. Select the **Inputs** and **Outputs** to export in the table.

   To export only used inputs, select **Used Inputs Only**.
8. Click **Export**.

⇨ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

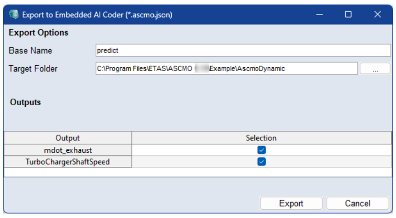
6.10.10 Export to Embedded AI Coder

When you export models to Embedded AI Coder, a JSON file is generated for each output.

Export a model to Embedded AI Coder

1. **File** menu › **Export Model** › **Embedded AI Coder**.

   Export to Embedded AI Coder window

2. Set the **Base Name** (defines the file name *<base name>_<output>*) and the **Target Folder**.
3. Select the outputs you want to export.

   ASCMO will create one JSON per checked output using `<base_name>_<output>.ascmo.json`.
4. Click **Export**.

⇨ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

6.10.11 Export to Bosch Flatbuffers

> Only MLP models without transformation can be exported to Bosch Flat-buffers.

When you export a model to Flatbuffers, a `*.dcm` file is created for each output.

Export a model to Bosch Flatbuffers

1. **File** › **Export Model** › **Bosch Flatbuffers**
2. Set the **Base Name** (defines the file name *<base name>_<output>*) and the **Target Folder**.
3. Select the **Inputs** and **Outputs** to export in the table. The order in which the inputs are passed to the exported model can be changed using the ↑ and ↓ buttons.

You can use the standard CTRL/SHIFT selection functions in the table, or click and hold LMB and drag the cursor over the cells/rows you want to select. The position of the inputs in the exported model is shown in the **#** column.

To export only used inputs, select **Used Inputs Only**.

4. Click **Export**.

⇨ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

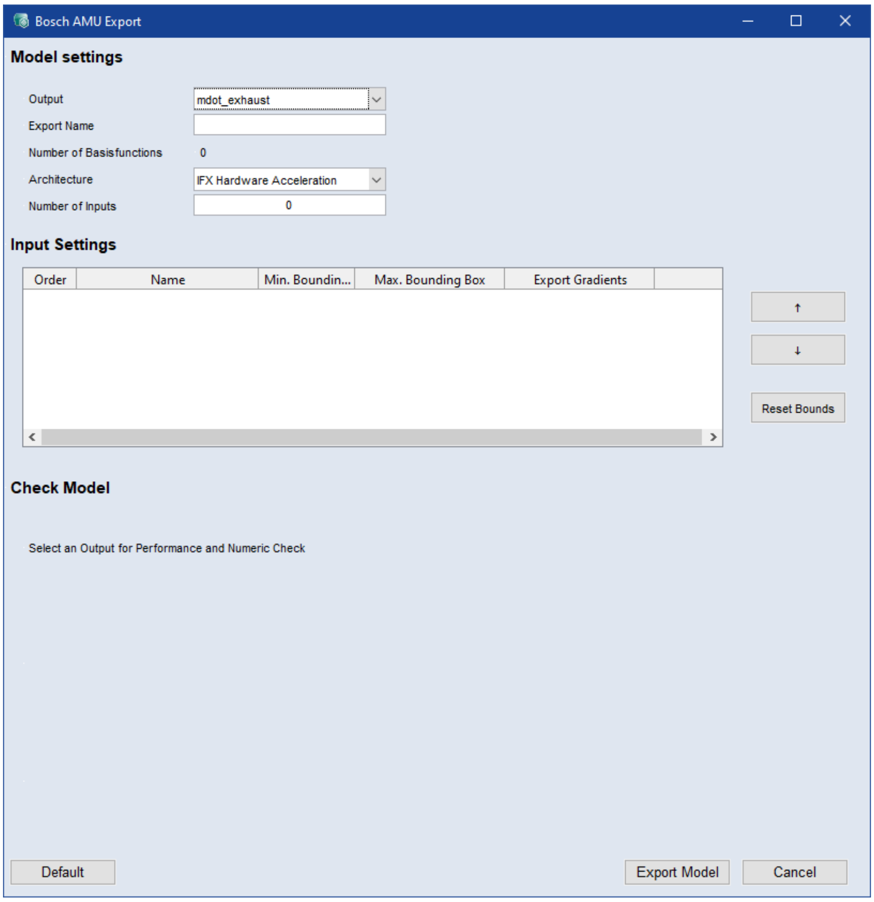
6.10.12 Export to Bosch AMU

Only NARX models with Gaussian process models with squared exponential kernel and logarithmic (or no) transformation are supported by the AMU, e.g., ASC GP, ASC GP-SCS, ASC Compressed.

Export a model to Bosch AMU

1. **File › Export Model › Bosch AMU**.

Bosch AMU Export window



2. Select an output from the **Output** drop-down list to perform a performance check and a numerical check.

The results are displayed in the **Check Model** area.

Check Model

Performance Check

Total flash memory demand (4000x11) 192272 [byte]
Estimated total calculation time (4000x11) 794.2 [µs], (IFX Dev4 B, with transfer time)
Estimated AMU calculation time (4000x11)605.5 [µs], (IFX Dev4 B, without transfer time)

Numeric Check for:	Training Data	Inside Bounding Box
Maximum found at index	3680	-
Value in measurement	162.8 [kg/h]	-
ASCMO-model prediction	161.4 [kg/h]	168.1 [kg/h]
AMU-model prediction	5.315 [kg/h]	5.347 [kg/h]
Worst case AMU-model prediction (upper)	5.316 [kg/h]	5.349 [kg/h]
Worst case AMU-model prediction (lower)	5.313 [kg/h]	5.346 [kg/h]
Max possible numeric error of the model	0.001218 [kg/h] (0.0007547 %)	0.00122 [kg/h] (0.0007256 %)

Worst Case Input Values

	PedalPosition_0	PedalPosition_1	PedalPosition_2	PedalPosition_3	PedalPosition_5	EngineSpeed_0	Eng
Training Data	57.78	57.4	57.4	57.01	56.25	2410	242
Inside Bounding Box	59.21	58.77	60.65	58.76	58.93	2438	241

3. Set the file name in the **Export Name** input field.
4. Select a microcontroller **Architecture** from the drop-down list.

   DFA (Bosch Data Flow Architecture) compatibility is displayed next to the drop-down list. DAF is not supported when gradient export is selected.

   IFX Hardware Acceleration

   JDP Hardware Acceleration

   IFX Pure Software Calculation

   JDP Pure Software Calculation
5. In the **Input Settings** tabel, you can set the **Min.** and **Max. Bounding Box** and whether to **Export** the **Gradient** of an input.
6. Click **Reset Bounds** to reset all bounding boxes values.
7. The order in which the inputs are passed to the exported model can be changed using the ↑ and ↓ buttons. You can use the standard CTRL/SHIFT selection functions in the table, or click and hold LMB and drag the cursor over the cells/rows you want to select. The position of the inputs in the exported model is shown in the **Order** column.
8. Click **Export Model** to export the model to Bosch AMU (`*.dcm`,`*.cdfx`).

⇨ The export starts and the files are saved to the specified path. A link to the export folder is displayed in the log window.

7 Tutorial: Working with ASCMO-STATIC ExpeDes

ASCMO-STATIC ExpeDes is a tool for creating space-filling statistical experiment plans. ASCMO-STATIC ExpeDes is ideally suited for planning measurements with a space-filling distribution over a grid of operating points (speed/load) as is required for model training in ASCMO-STATIC ExpeDes.

Rules for creating an experiment plan with ASCMO-STATIC ExpeDes

The following rules apply when creating an experiment plan:

– The experiment plan is valid at all times when created, and can thus be exported at any time.

– The experiment plan can be updated following modification either automatically when changing to another working step or manually by clicking the **Apply changes and update viewer** button in the toolbar or by executing the relevant menu function **View** → **Update**.

  If any modifications made are not permissible (invalid), you can choose whether to correct (stay in the current dialog window) or reject the changes made (change to another step).

– The steps can be carried out in any order, but we recommend the order described in this chapter.

– During processing, the experiment plan can be visualized at any time (**View** menu – see 7.3 "Visualizing the Experiment Plan " on page 189).

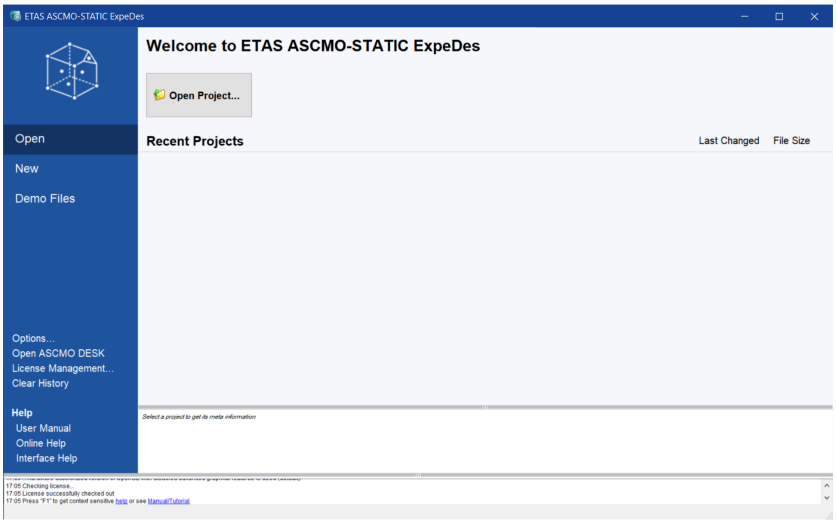
– Warnings in the log window indicate changes that have been made automatically due to dependencies between the working steps. For example, a constraint is automatically deleted if in Step 1 an input involved in the constraint has been deleted.

Starting ASCMO-STATIC ExpeDes

ASCMO-STATIC ExpeDes is started from the Windows Start menu or from the **ASCMO-STATIC ExpeDes** window (shown in "Starting ASCMO-STATIC " on page 88).

1. To start ASCMO-STATIC ExpeDes, do one of the following:
   - In the Windows **Start** menu, go to the **ETAS ASCMOV5.16** program group and select **ASCMO ExpeDes V5.16**.
   - In the **ASCMO-DESK** window, click the **Static Test Planning** title.

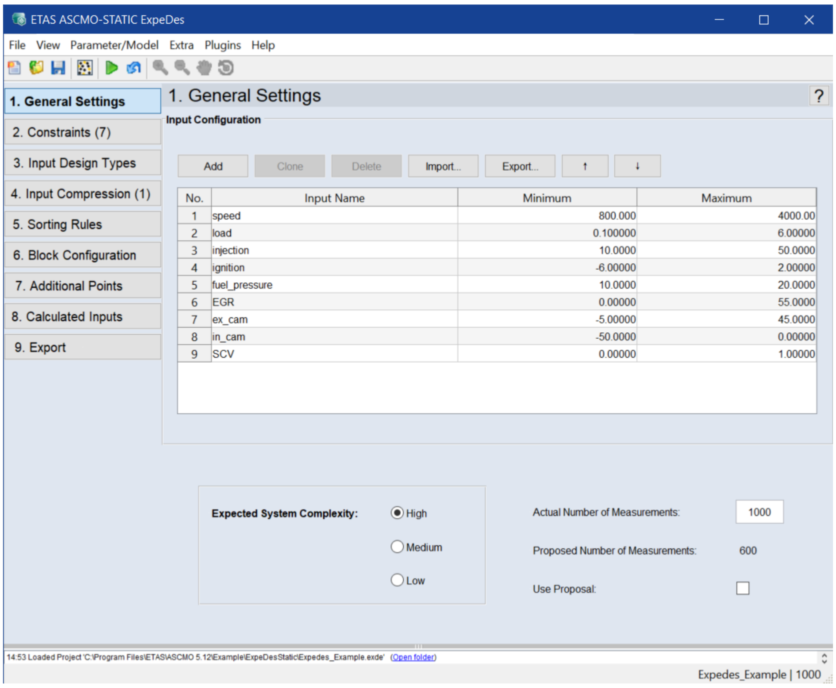
   The ASCMO-STATIC ExpeDes start window opens. You can start ASCMO-STATIC ExpeDes with an empty project (**New** › **Empty Project**), you can open a demo project (**Demo Files**), or you can open an existing project (**Open** › **Open Project**/ Select a recent project).



2. To open the ASCMO-STATIC ExpeDes main window with an empty project:
   a. Click **New** in the menu panel on the left.
   b. Click 

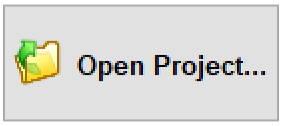
.

   The ASCMO-STATIC ExpeDes main window opens on the first page, with some default settings.



3. To open an existing project in ASCMO-STATIC ExpeDes, do the following:

i. In the ASCMO-STATIC ExpeDes main window, click **Open** in the menu panel on the left.

ii. Click 

.

An open file dialog opens.

iii. Select the `*.exde` file you want to open, then click **Open**.

The selected project is shown in the ASCMO-STATIC ExpeDes main window.

For details of the functions of the main menu and the toolbar, refer to the online help (‹F1› or **Help › Online Help**).

7.1 Working Steps of ASCMO-STATIC ExpeDes

The following sections describe the individual working steps involved in creating an experiment plan with ASCMO-STATIC ExpeDes.

NOTICE

Damage due to wrong test plan

Wrong engine settings in ASCMO-STATIC ExpeDes can lead to engine or test bench damage. Example: the operation point overstresses the engine and causes damage, e.g. by setting an ignition angle that causes extensive knocking.

- The general settings for the test plan must fit the system and the object. Negative example: 10000 rpm are set in the test plan vs. the motor has max. 6000 rpm.
- Limit the operation points to the allowed values. ETAS ASCMO does not have any knowledge about the engine parameters.
- Limit the engine load in the general settings before exporting the test plan.
- Verify the test plan for further use.

For ASCMO-STATIC ExpeDes see 7.11 "Step 9: Export " on page 217and 7.2 "Step 1: General Settings" on page 187.

- **7.2 "Step 1: General Settings" on page 187**

  This is the first step, in which the number of measurements and the number and configuration of the inputs is defined for the experiment plan.

- **7.3 "Visualizing the Experiment Plan " on page 189**

  To be able to evaluate the plan data, you can display it either graphically or as a table at any time.

- **7.4 "Step 2: Constraints " on page 191**

  In this step, constraints of the measurement range can be made for a variable as a function of one or two other variables.
- **7.5 "Step 3: Input Design Types " on page 203**

  In this step, settings can be made to define how individual inputs are to be measured.
- **7.6 "Step 4: Input Compression " on page 209**

  In this step, compressions of measuring points in certain areas of the measuring space can be specified for inputs by area.
- **7.7 "Step 5: Sorting Rules " on page 211**

  In this step, sorting rules can be defined for inputs so that the experiment plan is passed through in a meaningful way (according to the characteristics of the respective system).
- **7.8 "Step 6: Block Configuration" on page 212**

  In this step, the experiment plan can be divided into several parts (blocks) that can be measured separately. Each block by itself corresponds to the requirements of the design of experiments.
- **7.9 "Step 7: Additional Points" on page 215**

  In this step, points can be defined which, in addition to the points of the experiment plan, should be approached repeatedly according to specific criteria and measured, if necessary.
- **7.10 "Step 8: Calculated Inputs" on page 216**

  This step is skipped in this tutorial.
- **7.11 "Step 9: Export " on page 217**

  In this step, the properties of the project and the experiment plan itself are displayed. You can export the data in several formats. In addition, you can display the data as scatter plots, 3D plots or as a table; see Visualizing the Experiment Plan for details.

7.2 Step 1: General Settings

NOTICE

Damage due to wrong test plan

Wrong engine settings in ASCMO-STATIC ExpeDes can lead to engine or test bench damage. Example: the operation point overstresses the engine and causes damage, e.g. by setting an ignition angle that causes extensive knocking.

- The general settings for the test plan must fit the system and the object. Negative example: 10000 rpm are set in the test plan vs. the motor has max. 6000 rpm.
- Limit the operation points to the allowed values. ETAS ASCMO does not have any knowledge about the engine parameters.
- Limit the engine load in the general settings before exporting the test plan.
- Verify the test plan for further use.

For ASCMO-STATIC ExpeDes see 7.11 "Step 9: Export " on page 217and 7.2 "Step 1: General Settings" above.

This is the first step, in which the number of measurements and the number and configuration of the inputs is defined for the experiment plan.

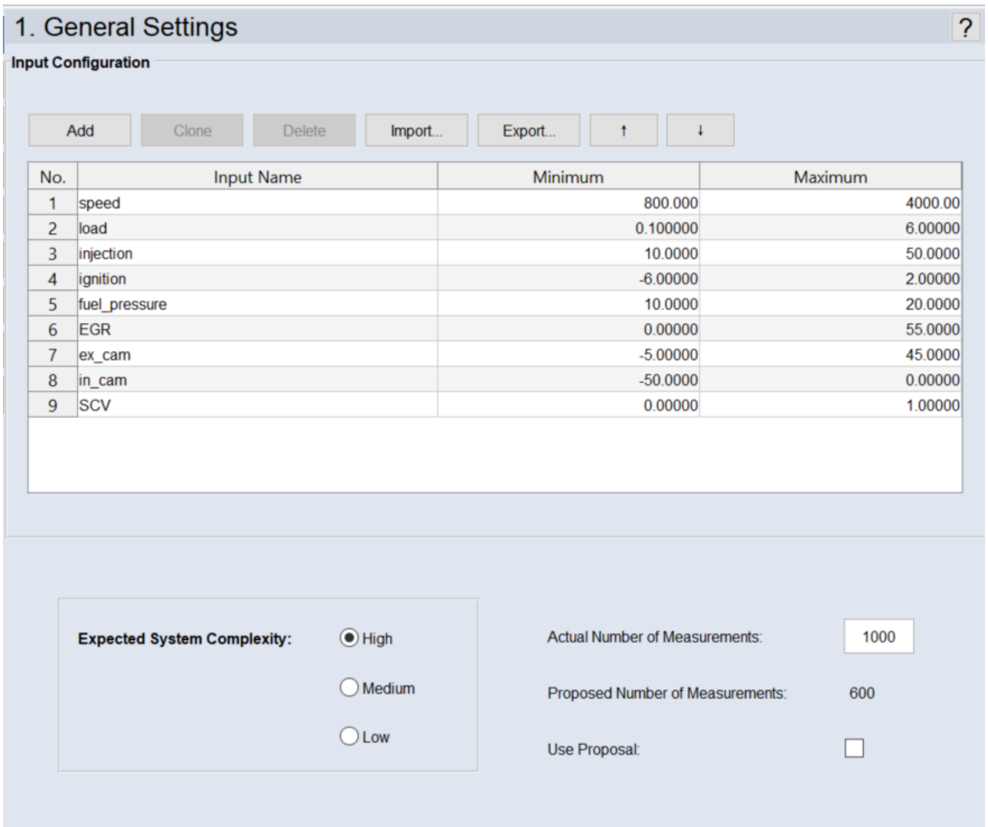


Fig. 7-1: ASCMO-STATIC ExpeDes Step 1: General Settings

7.2.1 Input Configuration

When you create a new ASCMO-STATIC ExpeDes project, several inputs are created automatically. These must be named and configured accordingly, and additional inputs may also be required.

Defining a new input

1. Click **Add** to define a new input.

   A new input is added to the end of the list.

   If you select a specific input (by clicking the row number) before you click **Add**, the new input is inserted directly under the selected row.

2. To duplicate existing inputs, select one or more inputs and click **Clone**.

   The duplicates, named *<input_name>*_2, are added below the respective originals.

Configuring an input

To configure an input, proceed as follows.

1. Click in a cell.

   The cell becomes an input field.

2. Enter the desired value, then press <RETURN>.

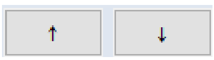
   You can specify the following parameters.

   Input Name: name of the input quantity to be measured

   Minimum / **Maximum**: lower/upper limit of the measurement range for the respective input

   > **Note**
   >
   > Input names must be unique.

3. Use the ↑ and ↓ buttons to move an input to another position in the list.

Importing names of inputs

1. Use the **Import** button to import a list of names available in one of the following file formats:

   - Lab File (`*.lab`), e.g ETAS INCA Variable File
   - MS Excel (`*.xls`, `*.xlsx`)
   - Comma separated values (`*.csv`)
   - Configuration File (`*.ini`), e.g. ETAS ASCMO Channel Config File

⇨ A window opens from which you can select the names file to be imported.

Removing an input

1. Select the row and click **Delete**.

7.2.2 Measurement Size Configuration

You have to define the number of measurements of the experiment plan either manually or via the options in the **Expected System Complexity** area. The following options are available.

- **Low**: For systems with approximately linear dependencies between inputs and outputs.
- **Medium**: For systems whose behavior approximately corresponds to a polynomial of second or third degree (such as a local operating point of an engine, for example).
- **High**: For complex systems which can no longer be represented using polynomials (such as an engine over an extended operating range, for example)

Configuring the number of measurements

1. In the **Actual Number of Measurements** field, enter the desired number.

   Or

   In the **Expected System Complexity** area, activate the option that best describes your system.

   Depending on your selection, a number is written to the **Proposed Number of Measurements** field.

   > **Note**
   >
   > The number proposed by the system depends on the number of inputs and on system complexity.

2. Activate the **Use Proposal** checkbox to use the suggested value.

With a large number of measurements, it is recommended that you split the measurements into smaller blocks (see 7.8 "Step 6: Block Configuration" on page 212) and, during modeling, try out how many blocks of this size are necessary to achieve the desired model quality.

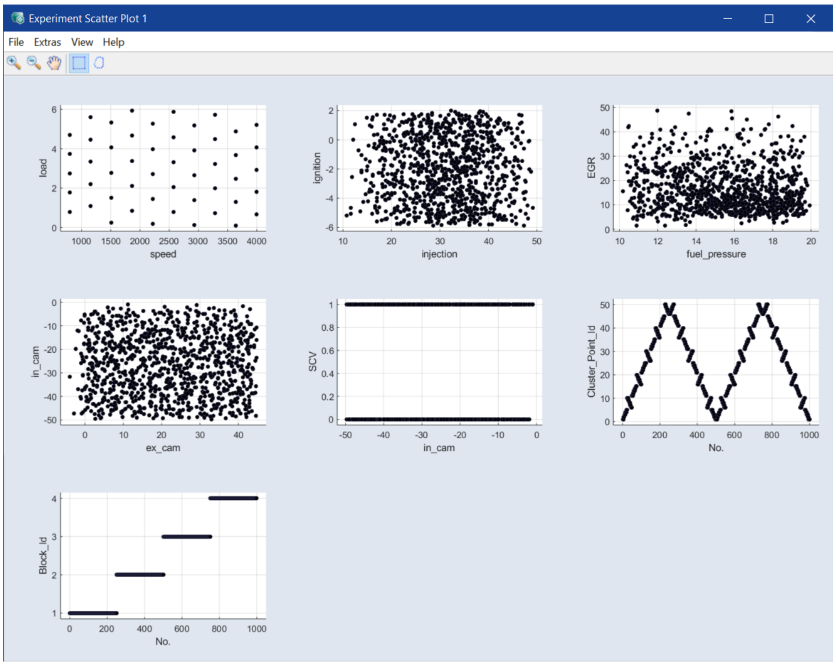
> **Note**
>
> Step **1. General Settings** results in a finished experiment plan that could now be exported as an `*.xls` or `*.csv` file. For more information on how to export an experiment plan, see section 7.11 "Step 9: Export " on page 217.

7.3 Visualizing the Experiment Plan

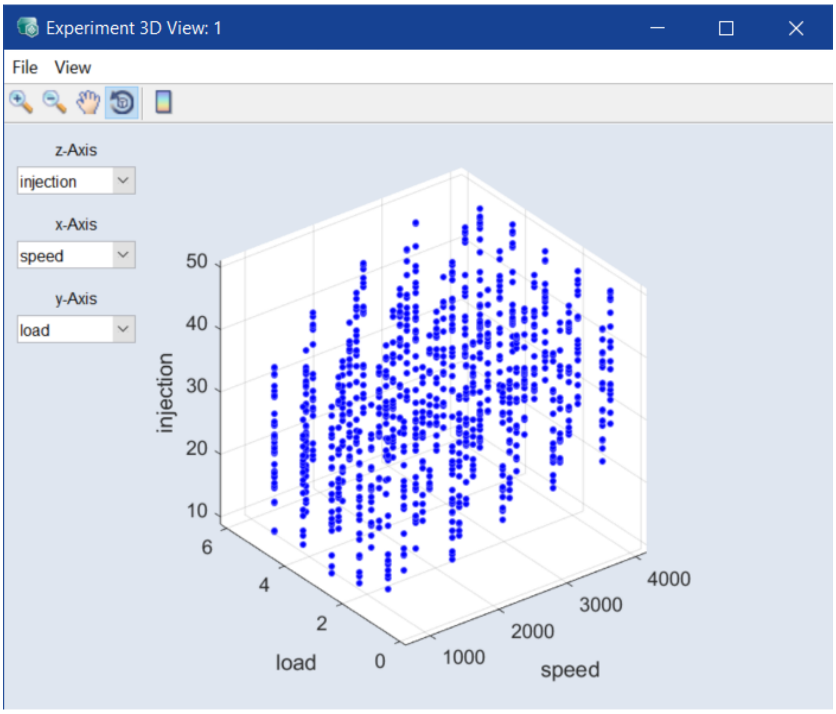
To be able to evaluate the plan data, you can display it either graphically or as a table at any time.

Visualizing the experiment plan

1. In the ASCMO-STATIC ExpeDes main window, select **View › Scatter Plot** to show 2-dimensional representations of the experiment plan.

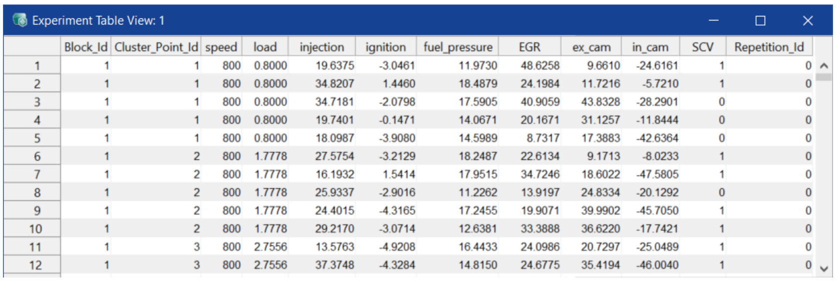


2. In the **Experiment Scatter Plot** window, use **View › Select Axes** to select the axes you want to show.
3. In the ASCMO-STATIC ExpeDes main window, select **View › 3D View** to show 3-dimensional plots.



Here the axes you want to show are selected directly in the plot window.

4. In the ASCMO-STATIC ExpeDes main window, select **View › Table View** to show the data of the measure plan in a table.



Experiment Table View: 1

	Block_Id	Cluster_Point_Id	speed	load	injection	ignition	fuel_pressure	EGR	ex_cam	in_cam	SCV	Repetition_Id
1	1	1	800	0.8000	19.6375	-3.0461	11.9730	48.6258	9.6610	-24.6161	1	0
2	1	1	800	0.8000	34.8207	1.4460	18.4879	24.1984	11.7216	-5.7210	1	0
3	1	1	800	0.8000	34.7181	-2.0798	17.5905	40.9059	43.8328	-28.2901	0	0
4	1	1	800	0.8000	19.7401	-0.1471	14.0671	20.1671	31.1257	-11.8444	0	0
5	1	1	800	0.8000	18.0987	-3.9080	14.5989	8.7317	17.3883	-42.6364	0	0
6	1	2	800	1.7778	27.5754	-3.2129	18.2487	22.6134	9.1713	-8.0233	1	0
7	1	2	800	1.7778	16.1932	1.5414	17.9515	34.7246	18.6022	-47.5805	1	0
8	1	2	800	1.7778	25.9337	-2.9016	11.2262	13.9197	24.8334	-20.1292	0	0
9	1	2	800	1.7778	24.4015	-4.3165	17.2455	19.9071	39.9902	-45.7050	1	0
10	1	2	800	1.7778	29.2170	-3.0714	12.6381	33.3888	36.6220	-17.7421	1	0
11	1	3	800	2.7556	13.5763	-4.9208	16.4433	24.0986	20.7297	-25.0489	1	0
12	1	3	800	2.7556	37.3748	-4.3284	14.8150	24.6775	35.4194	-46.0040	1	0

7.4 Step 2: Constraints

In this step, constraints of the measurement range can be made for a variable as a function of one or two other variables.

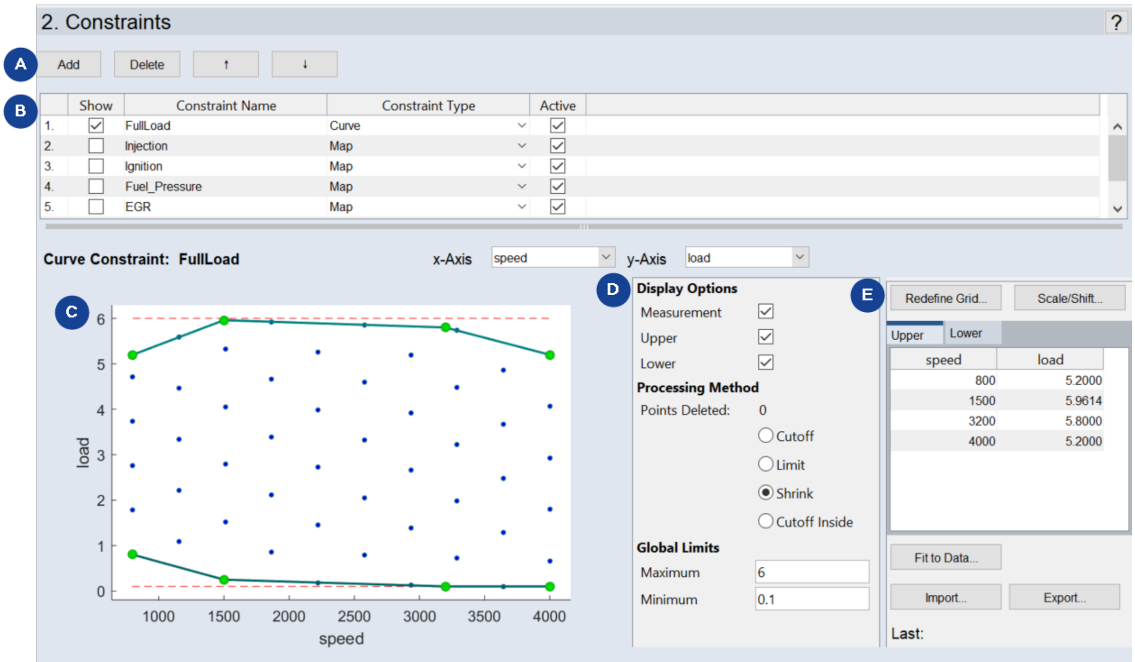


Fig. 7-2: ASCMO-STATIC ExpeDes Step 2: Constraints (Type "Curve")

These constraints can be added (also imported), visualized, configured and deleted again.

<u>Adding, deleting, and managing constraints</u>

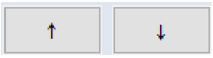
1. Click **Add** to create a new constraint (see **A** in Fig. 7-2: above).

   The new constraint is added to the end of the list.
2. Click in a cell in the **Constraint Name** column and enter a name for the constraint.
3. Click in a cell in the **Constraint Type** column and select the constraint type from the drop-down list.

The following types are available: `Map`, `Curve`, `Formula`, `Hull on imported Data`, and `Classifier`.

See sections 7.4.1 "Constraint Types "Map" and "Curve" " below and 7.4.2 "Constraint Type "Formula" " on page 195 for further information.

4. In the **Active** column, activate the checkbox for each constraint you want to use.
5. Use the ↑ and ↓ buttons to move a constraint to another position in the list.



6. Click **Delete** to remove a selected constraint.

Configuring a constraint of type `Curve` or `Map` is described in 7.4.1 "Constraint Types "Map" and "Curve" " below.

Configuring a constraint of type `Formula` is described in 7.4.2 "Constraint Type "Formula" " on page 195

What is shown in the bottom part of the window (regions **C**, **D**, **E** in Fig. 7-2) depends on the **Constraint Type**.

7.4.1 Constraint Types "Map" and "Curve"

Selecting variables

1. In the **Show** column, activate the checkbox for the constraint you want to edit.

   The constraint is displayed in the lower part of the window. The drop-down lists **x-Axis** and **y-Axis** (and **z-Axis** for maps) are provided.
2. In the drop-down lists, assign the relevant inputs to the axes.

   In the case of a curve, the functional dependency is `y(x)`; in the case of a map, the functional dependency is `z(x,y)`.

Graphical Display of Measurement Points and Constraint

If a completely specified constraint is selected in the list, the measurement points of the current experiment plan are displayed in a 2D (Curve, see Fig. 7-2: on the previous page) or 3D plot (Map).

Since both inputs feature the property "clustered" (see 7.5 "Step 3: Input Design Types " on page 203), the result is a pure grid.

The display of the measurement points is controlled via the **Measurement** option (region **D** in Fig. 7-2: on the previous page). In addition, the thicker points of the grid with which the constraint of areas is controlled are also displayed (see "Table for Displaying and Editing the Grid Nodes" on page 194).

Constraining the measurement range

1. In the **Show** column, activate the checkbox for the constraint you want to edit.

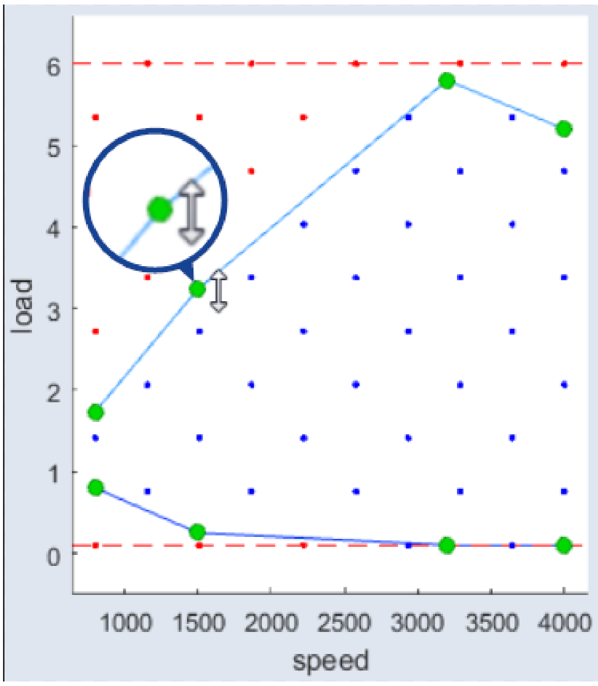
   The constraint is displayed in the lower part of the window.

2. In the constraint plot (region **C** in Fig. 7-2: on page 191), click a point of the constraining line/area and hold the mouse button pressed.

   The mouse pointer changes to a double arrow.

3. Drag the point to the desired position.

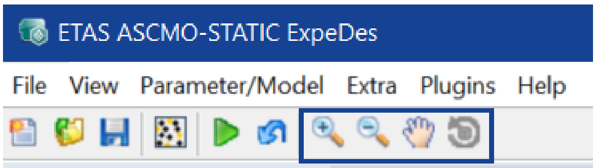
   The figure shows a limitation of the load at lower speeds.



4. In the **Processing Method** area, activate an option to determine the way points outside the constraints are treated.

   See Display Options list for a description of the options.

The display of the plot can also be influenced here with the **Zoom In**, **Zoom Out**, **Pan**, and **Rotate 3D** buttons in the toolbar.



The numeric values of the constraining points are shown in the tables on the right (**Upper** and **Lower** tabs; region **E** in Fig. 7-2: on page 191) and can be processed in these (both in terms of quantity and value); see "Table for Displaying and Editing the Grid Nodes" on the next page.

Further functions for specifying and displaying constraints are:

- **Display Options**
  - **Measurement**: Display of measurement points calculated by all defined constraints
  - **Upper/Lower**: Display of the upper/lower limits defined by the constraint
  - **Points Deleted**: Number of measurement points deleted by the constraint
- **Processing Method**: If the measurement range is constrained, there are several options to handle the number of measurement points
  - **Cutoff**: Removes the points outside the constraint, leaving the points inside the constraint the same. The number of deleted points is shown above (**Points Deleted**).
  - **Limit**: Points outside the constraint are set to the value of the constraint. Points inside the constraint remain unchanged.
  - **Shrink**: All points are scaled proportionally by the range defined by the upper and lower constraints. This option moves the points into the measuring range so that the measuring points are closer together.
  - **Cutoff Inside**: Removes the points inside the constraint and keeps the points outside the constraint. The number of deleted points is shown above (**Points Deleted**).
- **Global Limits**: The global limits of the variables to be limited (as defined in the constraints list; see 7.2 "Step 1: General Settings" on page 187).

Table for Displaying and Editing the Grid Nodes

The grid nodes can be edited in the tables to the right of the plot (region **E** in Fig. 7-2: on page 191). One tab each is available for the upper and lower constraint.

Changing the number of grid nodes

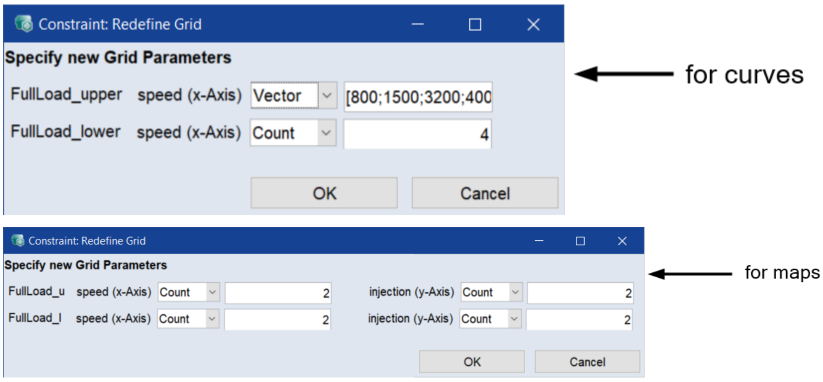
To change the number of points that define the constraining curves/maps, proceed as follows.

Note

The number of points must be in the range [2 .. 20].

1. Click **Redefine Grid**.

   The **Constraint: Redefine Grid** window opens.

2. To change the number of points on a constraint axis, do the following:
   i. In the drop-down list for the constraint axis, select `Count`.

      The number of points is displayed in the input field for the respective constraint axis.

   ii. Enter the desired number.
3. To enter the grid vector for a constraint axis directly, do the following:
   i. In the drop-down list for the constraint axis, select `Vector`.

      The vector is displayed in the input field for the respective constraint axis.

   ii. Edit the vector values as desired.

      Note

      With type `Count`, the grid points are equidistant. With type `Vector`, you can distribute the grid points unevenly.

4. Click **OK.**

   The points are adjusted accordingly. The **Constraint: Redefine Grid** window closes.

7.4.2 Constraint Type "Formula"

In the case of inputs that functionally depend on each other, you must ensure that no-settable states of the system to be measured are not addressed. This can be achieved using formula-based constraints.

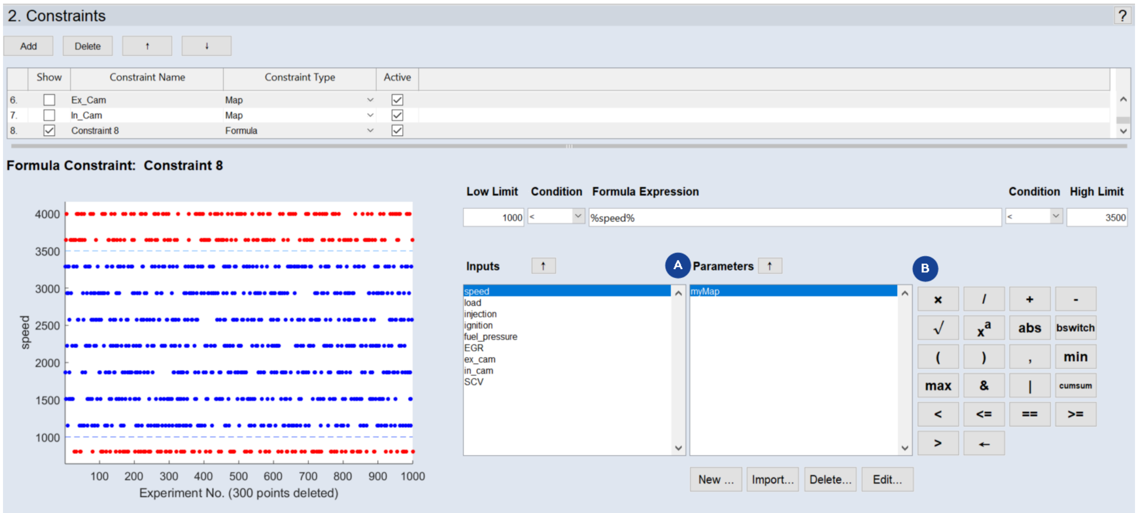


Fig. 7-3: ASCMO-STATIC ExpeDes Step 2: Constraints (Type "Formula")

Formula Definition

Use the **Low Limit** and **High Limit** fields to specify lower and upper limits for the formula. The fields require scalar numerical inputs.

Use the **Condition** drop-down lists to select comparison operators. Available operators are none (if you want to specify no lower or upper limit), < (less than) and <= (less or equal).

Use the **Formula Expression** field to enter the formula. Any MATLAB relational expression can be entered here. If the expression is violated, the corresponding measurement points are rejected. Variables can be either inputs or curves/maps (1D or 2D) that are selected and managed in the **Inputs** area and the **Parameters** area. Operators are entered directly or via the button area (**B** in Fig. 7-3: above).

Example

In the case of a multi-point injection in which the injection times `T_1` and `T_2` are to be measured, you must ensure that, for an injection period `Delta_T`, the following condition is adhered to:

```
T_1 + Delta_T < T_2
```

In addition to the inputs `T_1` and `T_2`, `Delta_T` can be defined as a map that is a function of other inputs (or even of existing maps).

"Inputs" and "Parameters" Lists

- **Inputs**: The **Inputs** list shows all inputs of the experiment plan that can be inserted in the **Formula Expression** field, either by typing their names or via the ↑ button.
- **Parameters**: The **Parameters** list shows all maps of the experiment plan that can be inserted in the **Formula Expression** field, either by typing

their names or via the ↑ button.

The buttons below the **Parameters** list can be used to create, import, delete, and edit a selected map.

7.4.3 Managing Curves and Maps

This section contains information on how to create, edit, delete, import, and export maps/curves used in constraints.

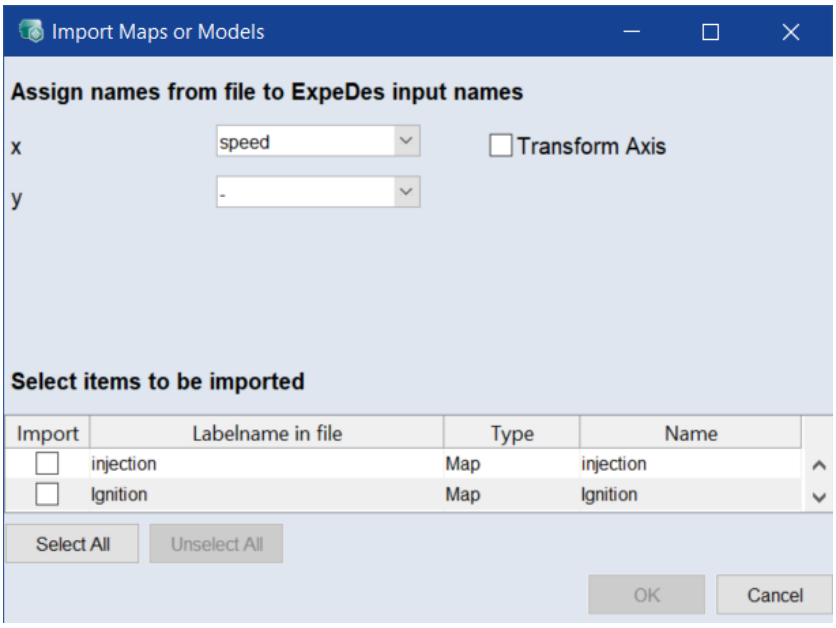
Importing map/curve data from a file

If the data for a constraint curve or map exist as e.g. `*.dcm`, `*.cdfx`, `*.csv`, `*.xls`, `*.xlsx`, `*.xlsm` or `*.ascmo` file, they can be imported.

1. In the **Constraints** step select the **Show** checkbox of the constraint to which you want to import the curve/map.
2. Select **Parameter/Model › Import**.

   A file selection window opens.
3. Select the file you want to import and click **Open**.

   The **Import Maps or Models** window opens. It lists the items in the file.

4. Use the drop-down lists to assign the proper inputs to the X and – for maps only – Y axes.

   If you assign an input to the Y axis for a curve, the assignment is ignored.
5. In the **Import** column, activate the checkboxes of the items you want to import.
6. In the **Name** column, enter a unique name for each item you want to import.
7. Click **OK**.

➪ The selected items are imported to the currently selected constraint.

Afterwards, the borders can be slightly extended by adding scales and shifts for the lower and upper limits via the **Scale/Shift** button.

> **Note**
>
> To extend the **lower** borders, the value of the shift must be **negative**.

To assign an imported map/curve to a constraint, proceed as described in .

Exporting curve/map data

You can export the curve/map used in the currently displayed constraint, or you can export the curves/maps defined in your ASCMO-STATIC ExpeDes project. Available export formats are `*.dcm` and `*.csv`.

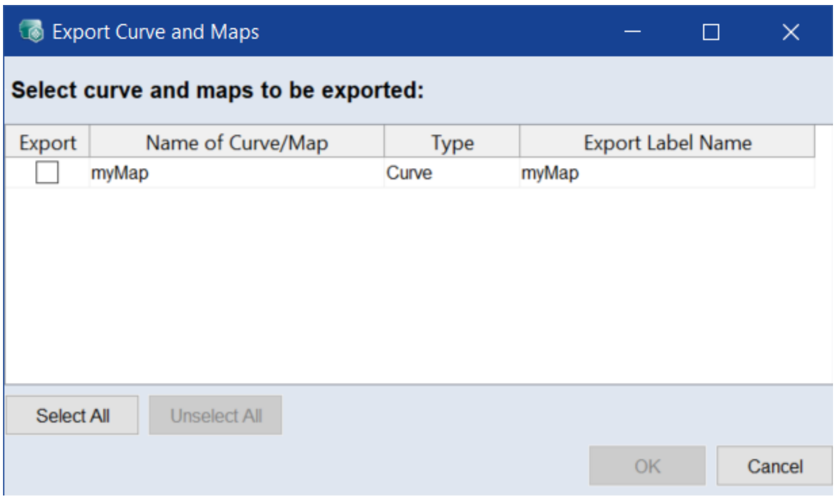
1. Export the curve/map of the currently displayed constraint
   i. In the table area (region **E** in Fig. 7-2: on page 191), click **Export**.

      The **Save Constraint Maps** window opens.
   ii. Select the file type.
   iii. Specify the location and the file name and click **Save**.

      The **Specify label names** window opens.
   iv. In that window, enter labels for the maps/curves.
   v. Click **OK**.

      The maps/curves used as upper and lower constraint are exported.
2. Export the curves/maps in the ASCMO-STATIC ExpeDes project
   i. In the ASCMO-STATIC ExpeDes window, select **Parameter/Model › Export**.

      The **Export Curve and Maps** window opens.



   ii. In the **Export** column, select the maps/curves you want to export.

      You can use **Select All** and **Unselect All** to select/unselect all maps/-curves in the list.

iii. Click **OK**.

The **Save Constraint Maps** window opens.

iv. Select the file type.

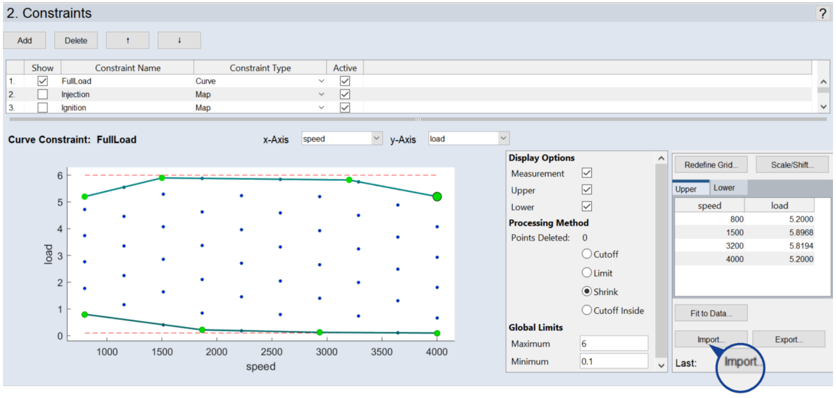
v. Specify the location and the file name and click **Save**.

The maps/curves are exported. The **Export Curve and Maps** window closes.

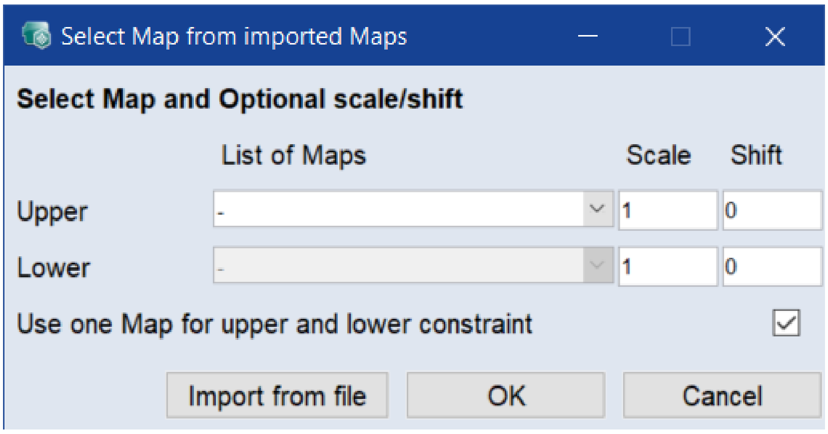
Assigning an existing map/curve to a constraint

If a map/curve already exists in the ASCMO-STATIC ExpeDes project, it can be selected via the **Import** button and assigned to a constraint.

1. Select the constraint to which you want to assign the map/curve data.
2. Click **Import** at the bottom right (see region **E** in Fig. 7-2: on page 191).



The **Select Map from imported Maps** or **Select Curve from imported Curves** window opens.



> **Note**
>
> If no maps/curves are available in the ASCMO-STATIC ExpeDes project, a file selection window opens, in which you can import map data (`*.dcm`, `*.csv`) for further use.

3. To use different maps for the upper and lower limit, deactivate the **Use one map for upper and lower constraint** checkbox.

4. In the **List of Maps/List of Curves** column, select the map/curve to be used for the upper and lower limit.
5. Click **OK**.

   The **Select Map from imported Maps** or **Select Curve from imported Curves** window closes. The upper and lower bounds are assigned to the constraint and displayed.

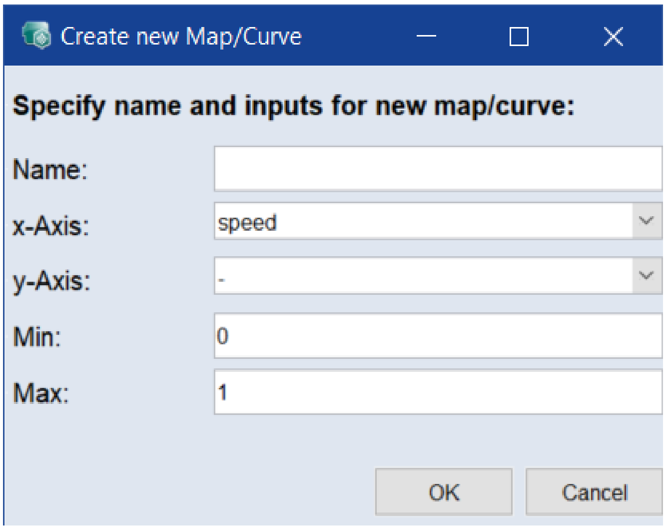
   Afterwards, the borders can be slightly extended by adding scales and shifts for the lower and upper limits via the **Scale/Shift** button.

   > **Note**
   >
   > To extend the lower borders, the value of the shift must be negative.

Creating a map/curve

1. To create a new Map/Curve, do one of the following:
   - In the ASCMO-STATIC ExpeDes window, select **Parameter/Model › New**.
   - In the **Parameters** list of a formula constraint (region **A** Fig. 7-3: on page 196), click **New**.

   The **Create new Map/Curve** window opens.



2. Enter a name, select the inputs at which the Map/Curve is defined and enter upper (**Max**) and lower (**Min**) limit.
3. Click **OK**.

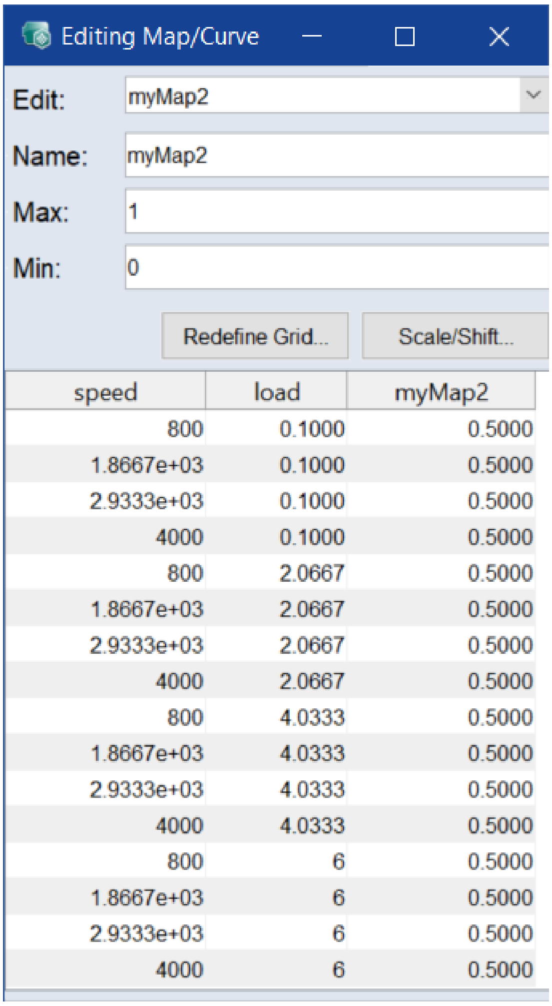
   The **Editing Map/Curve** and the **Map ‹name›** or **Curve ‹name›** windows open. Here you can edit the new map.

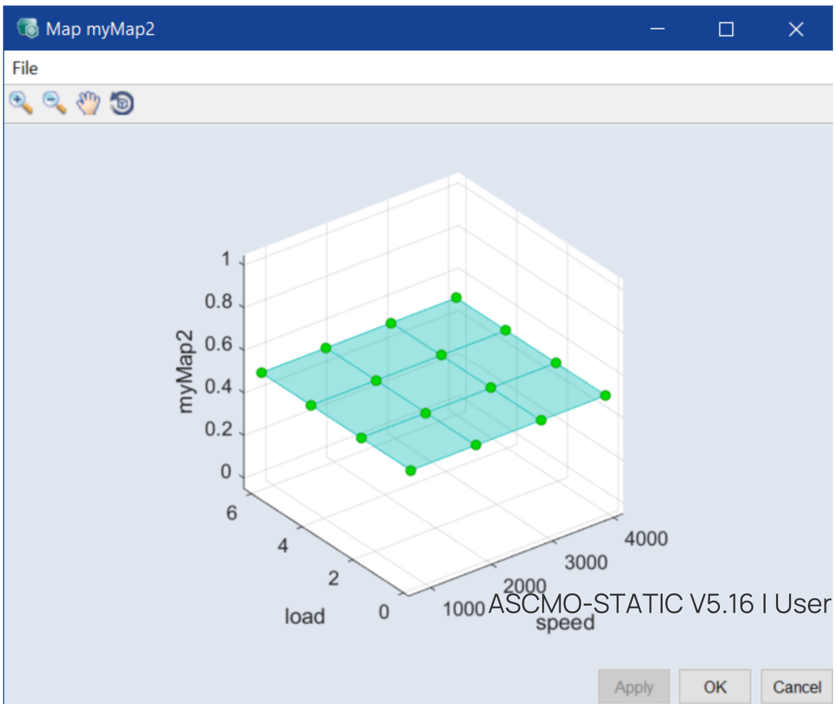
Editing a Map/Curve

Maps/Curves are displayed as a table or plot and can be edited here.

1. To modify an existing Map/Curve, do one of the following:
   - In the ASCMO-STATIC ExpeDes window, select **Parameter/Model › Edit › *‹map/curve name›***.
   - In the **Parameters** list of a formula constraint (region **A** Fig. 7-3: on page 196), click **Edit**.

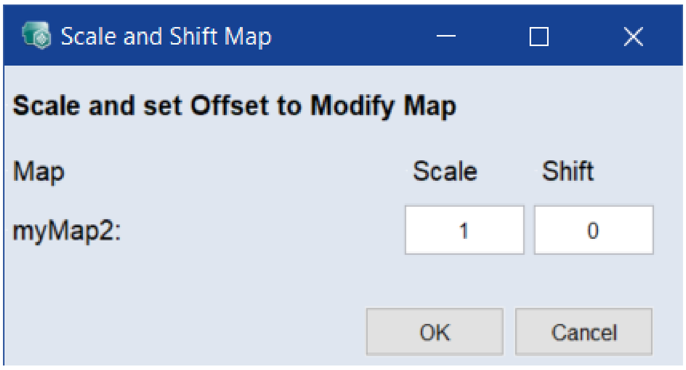
   The **Editing Map/Curve** and the **Map ‹name›** or **Curve ‹name›** windows open.





2. To redefine the grid points, click **Redefine Grid**, then proceed as described in "Changing the number of grid nodes" on page 194.
3. To scale and/or shift the map/curve values, do the following:
   i. Click **Scale/Shift**.

      The **Scale and Shift Map** window opens.

   ii. In that window, enter a scale factor and/or a shift for the map/curve values.

      Note

      To extend the lower borders, the value of the shift must be negative.

   iii. Click **OK**.
4. Move the **Smoothness** slider to smooth the map/curve.

Renaming a map/curve

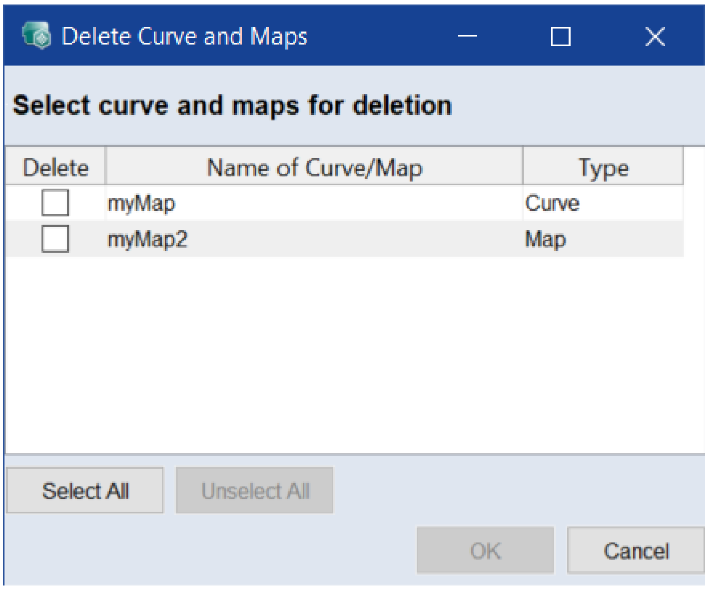
1. In the ASCMO-STATIC ExpeDes window, select **Parameter/Model › Edit › *‹map/curve name›***.

   The **Editing Map/Curve** and the **Map ‹name›** or **Curve ‹name›** windows open.
2. In the **Name** field, edit the name of the map/curve.
3. In the **Map ‹name›** or **Curve ‹name›** window, click **OK** to accept the change and close the windows or **Apply** to accept the change without closing the windows.

Deleting a map/curve

1. To delete a map/curve, do one of the following:
   - In the ASCMO-STATIC ExpeDes window, select **Parameter/Model › Delete**.
   - In the **Parameters** list of a formula constraint (region **A** Fig. 7-3: on page 196), click **Delete**.

   The **Delete Curve and Maps** window opens.



2. In the **Delete** column, select the maps/curves you want to delete.
3. Click **Delete**.
4. Click **Yes** to delete the selected maps/curves.

7.5 Step 3: Input Design Types

In this step, settings can be made to define how individual inputs are to be measured.

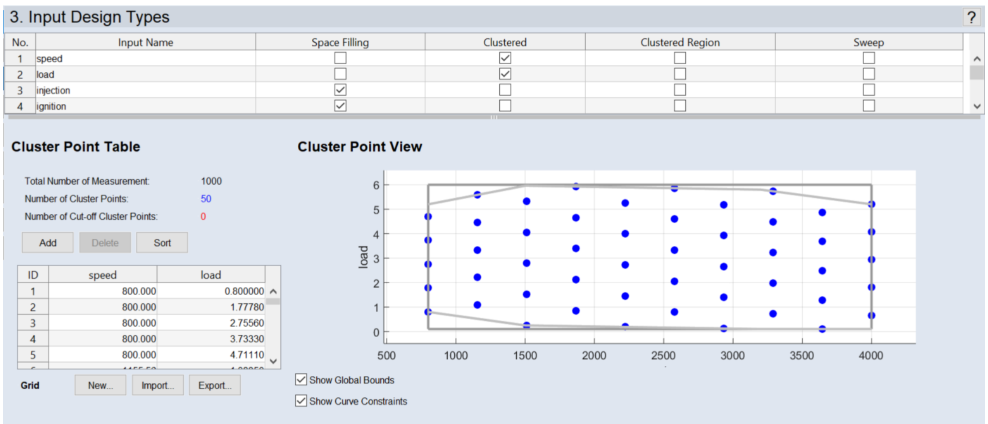


Fig. 7-4: ASCMO-STATIC ExpeDes Step 3: Input Design Types (type **Clustered**)

One of the following three measuring types can be selected for each input.

- **Space Filling**: ASCMO-STATIC ExpeDes creates an experiment plan in which the measuring points are distributed space-filling and, at the same time, quasi-randomly.
- **Clustered**: The concept of cluster points allows the user to disable the space-filling algorithm for certain parameters and, instead, concentrate the many measurements distributed in the space at a few locations (cluster).

This can be useful if the effort for setting an input is very high (e.g., operating point for engine measurement) or if certain parameters can be present only in form of discrete values (e.g., prototype parts for production).

If you configure a parameter as clustered, you are responsible for selecting the measuring points. ASCMO-STATIC ExpeDes provides support in specifying the points insofar as it offers an import of measuring points from files and the creation of an arrangement as a grid (see "Cluster Point Table Area" below).

Note

The number of clustered inputs is limited to three.

– **Sweep**: In the case of inputs of the type **Sweep**, specific points are addressed several times in every measurement.

  If `Input_n` is defined as **Sweep** (consisting of `m` points), an experiment plan is created for `Input_1` ... `Input_n-1` and then extended so that the `n-1` inputs are measured with the `m` values of `Input_n` each time.

  If you configure an input as **Sweep**, you are responsible for selecting the measurement points. ASCMO-STATIC ExpeDes supports the user in defining the points by offering an import of measurement points from files and the generation of a grid arrangement (see on page 207).

Note

Settings from the steps **Constraints** and **Input Compression** also have an effect on inputs of the type **Sweep**.

Cluster Point Table Area

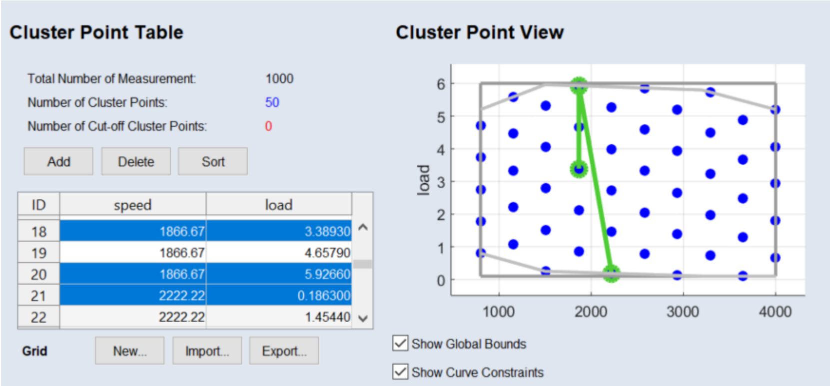
This area displays the table of the current cluster points. Points can be selected, inserted, and deleted.

Selecting points

1. In the **Cluster Point Table** (left area below **Input Design Types** table), click in a row to select a point.

   Multiselection is possible by pressing the <SHIFT> or <CTRL> key.

   Points selected in the table are displayed with green color to the right of the plot.



Inserting points

1. In the cluster point table, select the point after which you want to insert the new point.

   If you do not select a point, the new point is inserted at the end of the list.

   If you select several points, new points are added after each selected point.

2. Click **Add**.

➩ The new point is inserted. If it is placed between two points, its values are determined according to the values in the rows above and below. If it is placed at the end of the list, its values are determined by the mean values of both axes.

Removing points

1. In the cluster point table, select one or more points you want to delete.
2. Click **Delete**.

➩ The points are deleted without warning.

Importing cluster points

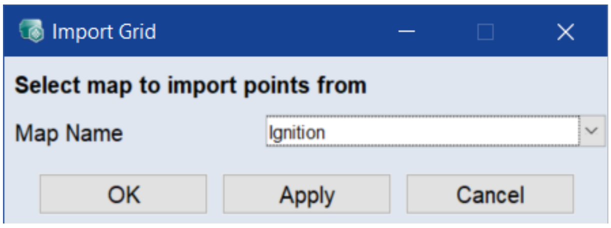
If a cluster point grid exists as `*.dcm` or `*.csv` or `*.xls` file, it can be imported.

1. Below the cluster point table, click **Import**.

   A file selection window opens.

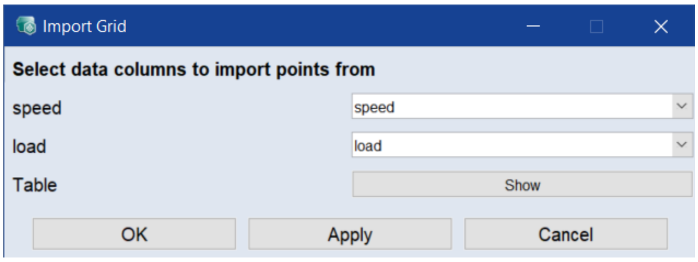
2. Select the file you want to import and click **Open**.

   The **Import Grid** window opens. Its content depends on the format of the import file. The **Map Name** drop-down list contains all maps in the file.



for *.dcm files

for *xlsx/
*.csv files



3. Select the map you want to import.
4. Click **OK**.

⇨ If you selected a curve, an error message is issued and the import aborts.

If you selected a map, the map is imported and used as cluster point grid.

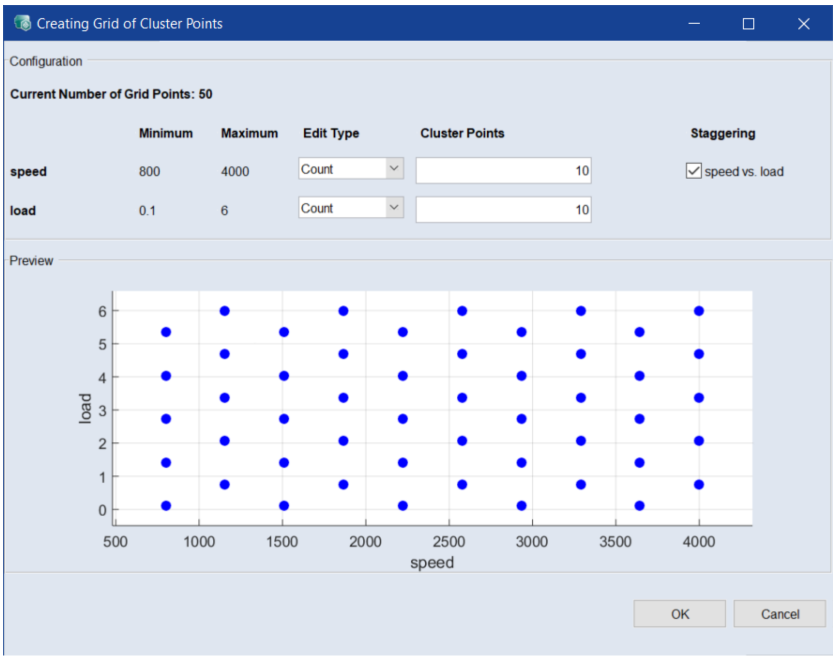
> **Note**
>
> The importer does not warn you if some or all axis values of the selected map are outside the range defined for the clustered inputs. Such axis values are cut off.

Editing cluster points

To change the number of cluster points, proceed as follows.

1. Below the cluster point table, click **New**.

   The **Creating Grid of Cluster Points** window opens.



2. To change the number of points on an axis, do the following:
   i. In the drop-down list for the axis, select `Count`.
      The number of points is displayed in the input field for the respective axis.
   ii. Enter the desired number.
3. To enter the grid vector for an axis directly, do the following:
   i. In the drop-down list for the constraint axis, select `Vector`.
      The vector is displayed in the input field for the respective axis.
   ii. Edit the vector values as desired.

> **Note**
>
> With type `Count`, the grid points are equidistant. With type `Vector`, you can distribute the grid points unevenly.

4. Activate the option below **Staggering** to reduce the number of grid points and thus the variations of speed and load during measurement.
5. Click **OK**.

⇨ The points are adjusted accordingly. The **Creating Grid of Cluster Points** window closes.

Cluster Point View Area

The current values defined in the table are displayed here. Cluster points cut off by global bounds or constraints are shown as red dots. Cluster points selected in the table are highlighted by green circles (see also "Selecting points" on page 204).

Global bounds and constraints can be shown via the options below the plot.

Sweep Value Definition Area

This area shows a table for the input selected from the list with the property **Sweep**. Points can be selected, inserted, and deleted there. Simultaneously, the resulting measurement effort is indicated by the value "Total Number of Measurement".

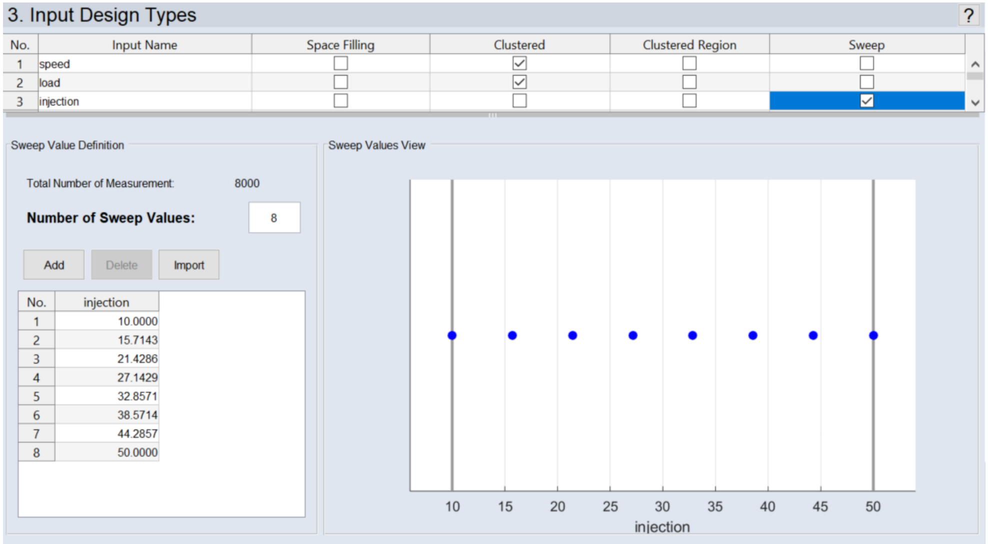


Fig. 7-5: ASCMO-STATIC ExpeDes Step 3: Input Design Types (type **Sweep**)

Selecting points

1. In the **Sweep Value Definition** area (left area below **Input Design Types** table), click in a row to select a point.

   Multiselection is possible by pressing the <SHIFT> or <CTRL> key.

⇨ Points selected in the table are displayed with green color to the right of the plot.

Inserting points

1. Increase the number in the **Number of Sweep Values** field and press <RETURN>.

   If you entered `n`, `n` equidistant points are created.

   Or

   In the sweep value table, select the point after which you want to insert the new point.

   If you do not select a point, the new point is inserted at the end of the list.

   If you select several points, new points are added after each selected point.

2. Click **Add**.

⇨ The new point is inserted. If it is placed between two points, its value is determined according to the values in the rows above and below. If it is placed at the end of the list, it uses the maximum value.

Removing points

1. Decrease the number in the **Number of Sweep Values** field and press <RETURN>.

   If you entered `m`, `m` equidistant points are created.

   Or

In the sweep value table, select one or more points you want to delete.

2. Click **Delete**.

⇨ The points are deleted without warning.

Importing points

If a sweep value list exists as `*.csv` or `*.xls` or `*.xlsx` file, it can be imported.

1. In the sweep value table, click **Import**.

   A file selection window opens.

2. Select the file you want to import and click **Open**.

   The **Import Grid** window opens. The **Map Name** drop-down list contains all maps in the file.

3. Select the list you want to import.
4. Click **OK**.

⇨ The list is imported. If one or more values in the list are outside the allowed range for the input, a warning is issued in the log window:

```
At least one value is outside the defined Min/Max range
of this input and will be discarded!
```

7.6 Step 4: Input Compression

In this step, compressions of measuring points in certain areas of the measuring space can be specified for inputs by area.

The model precision in this area can be improved because the measuring points are closer together.

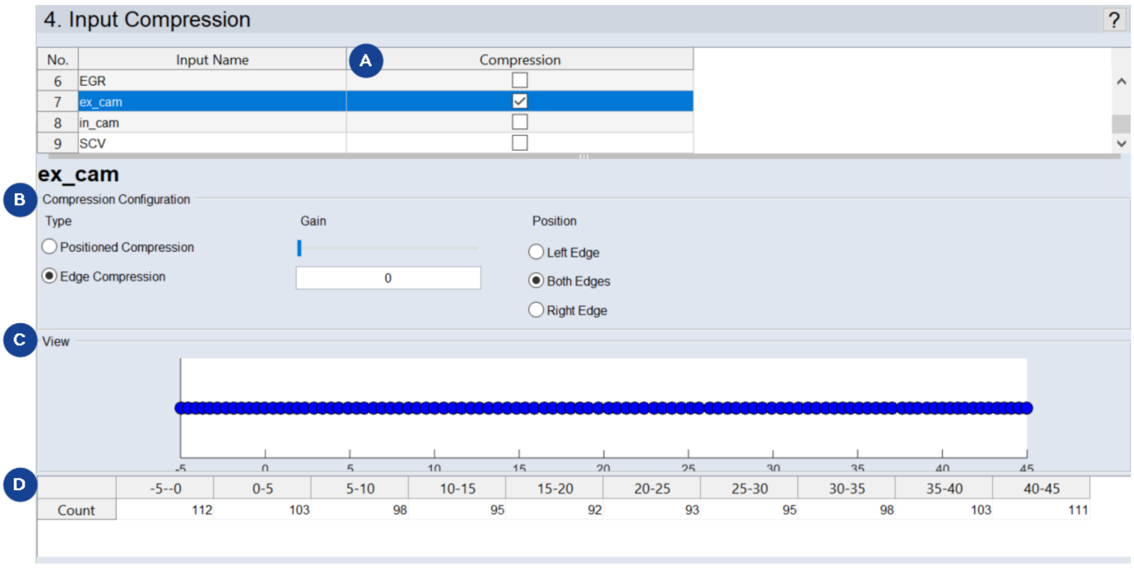


Fig. 7-6: ASCMO-STATIC ExpeDes Step 4: Input compression (type **Edge Compression**)

Selecting inputs to be compressed

1. In the **Compression** column, activate the option for each input you want to compress (see **A** in Fig. 7-6: on the previous page).

➪ The input currently selected in the list is shown below the list.

7.6.1 Compression Configuration Area

In area **B** (see Fig. 7-6: on the previous page), you can define the type and the orientation of compression. There are two types of compression available:

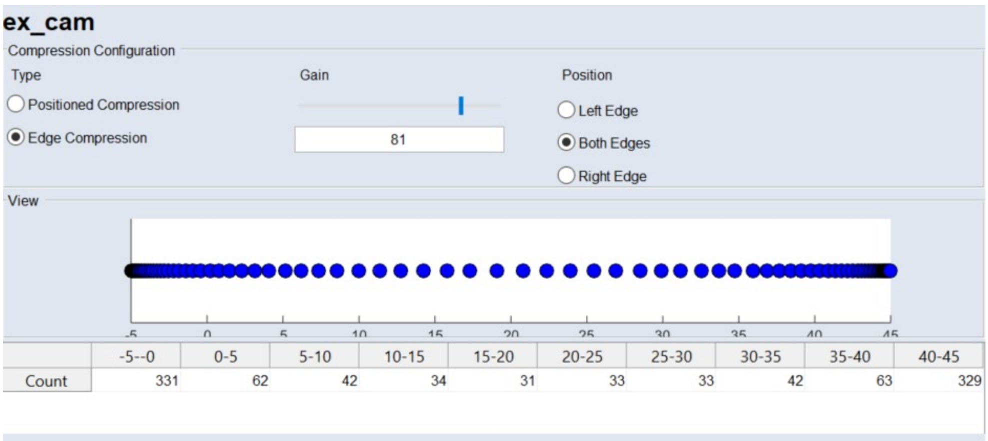
- **Positioned Compression**: In this method, compression focuses on a freely selectable point.
  - **Gain**: Degree of compression for the selected center (**Position**)
  - **Position**: Position of the center of compression
- **Edge Compression**: This allows obtaining a compression to one or both edges of the measurement range.
  - **Gain**: Degree of compression
  - **Position**: The compression is done for one edge (**Left Edge**, **Right Edge**) or both edges (**Both Edges**) of the area.

7.6.2 View Area

In area **C** and **D** (see Fig. 7-6: on the previous page), the position of the measurement points and, therefore, the measure of the selected compression is graphically displayed. Underneath it, the number of measurements in the respective interval resulting from the compression is specified under **Count**.

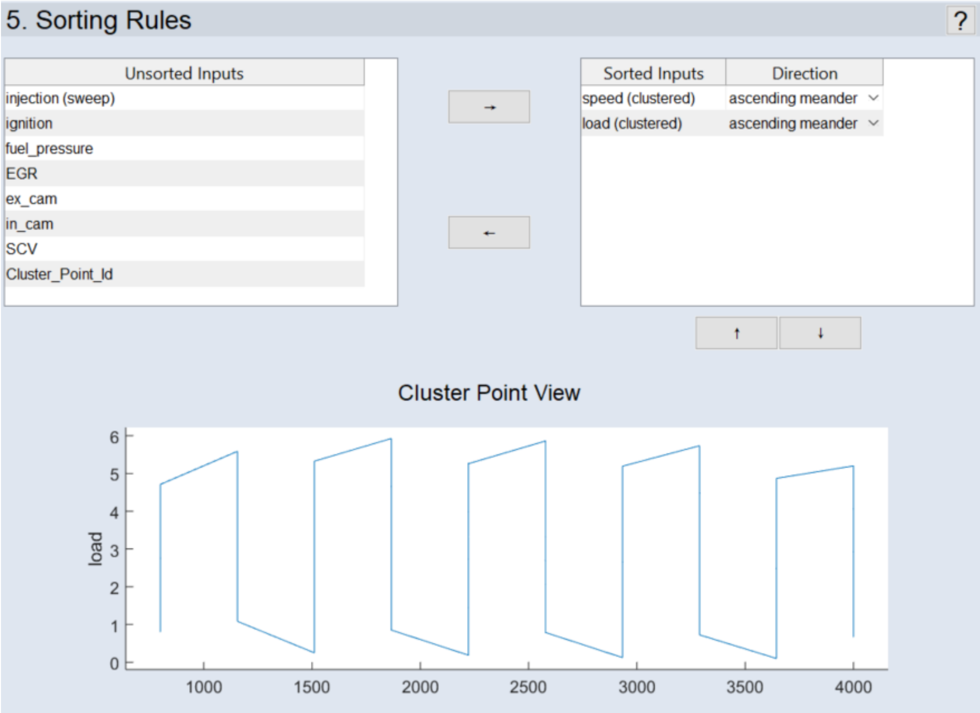
7.6.3 Example of an Applied Input Compression

In the figure below, the measurements of `ex_cam` are compressed with a gain of 81 to both edges of the measurement range.



7.7 Step 5: Sorting Rules

In this step, sorting rules can be defined for inputs so that the experiment plan is passed through in a meaningful way (according to the characteristics of the respective system).



The window consists of two areas. The inputs to be measured can be moved between them with the → and ← arrow buttons ( / ).

Unsorted Inputs Area

All unsorted inputs are listed here.

Sorted Inputs Area

All sorted inputs are listed here.

- **Direction**: The direction through which this input should pass: **Ascending**, **Descending**, or **Alternating start low/middle/high**.
- **Meander**: If this option is activated, the second sorted value does not lead to a reset after the first sorted value has been processed (**ascending** sorting); instead, the value is retained and processed backwards.

  In the screenshot above, no reset takes place to minimum load after measuring the highest load and changing to a higher rotational speed; instead, the load value is retained and the loads (at this rotational speed) are processed in descending order.

Note

The term **Meander** is not meaningful for the first entry in the list of sorted inputs in each case.

The hierarchy within the list of sorted inputs can be modified using the ↑ and ↓ arrow buttons (↑ / ↓).

Cluster Point View Area

The behavior of the measurement configured above is displayed graphically.

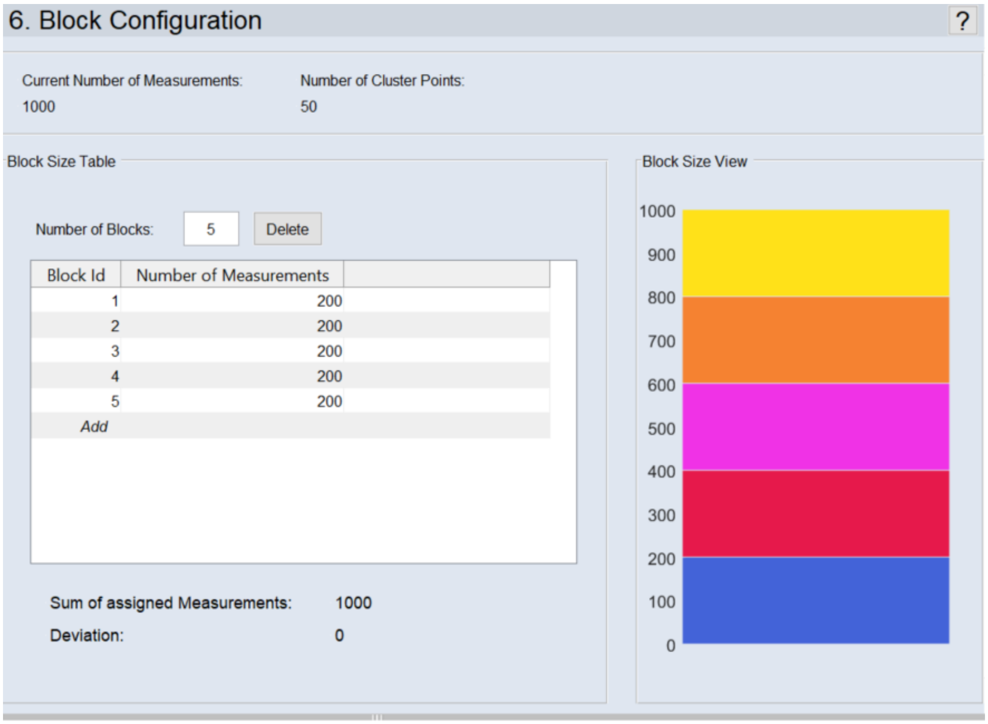
Note

This area is only displayed if clustered inputs was selected for the sorted values.

7.8 Step 6: Block Configuration

In this step, the experiment plan can be divided into several parts (blocks) that can be measured separately. Each block by itself corresponds to the requirements of the design of experiments.

The advantage of block building is that the effort involved in measuring is reduced. By measuring just a few blocks, the model precision achieved is satisfactory.



The total number of measurements (**Current Number of Measurements**) and the **Number of Cluster Points** are shown at the top. The latter number is of particular importance since it is used to define minimum size of a measurement block.

Block Size Table *Area*

The size of the individual blocks is defined here.

The sum of the defined measurement blocks is displayed under Sum of **Assigned Measurements**. If this number is less than or greater than the **Current Number of Measurements**, it is highlighted in red under **Deviation**.

Creating and Editing Blocks

1. In the **Number of Blocks** field, enter a number and press <RETURN>.

   If you entered n, n blocks of equal size are created.

   Or

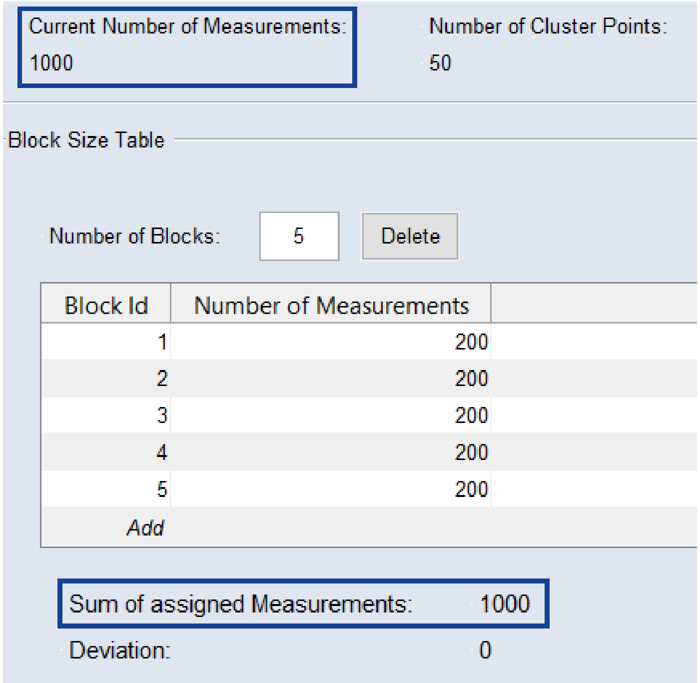
   In the last row of the table, with Block ID = *Add*, click in the Number of **Measurement** cell, enter the number of measurements for the new block and press <RETURN>.

Block Id	Number of Measurements
1	200
2	200
3	200
4	200
5	200
Add	200

   A new block is added.

2. Adjust the number of measurement for the old blocks so that the sum of

assigned measurements equals the current number of measurements.



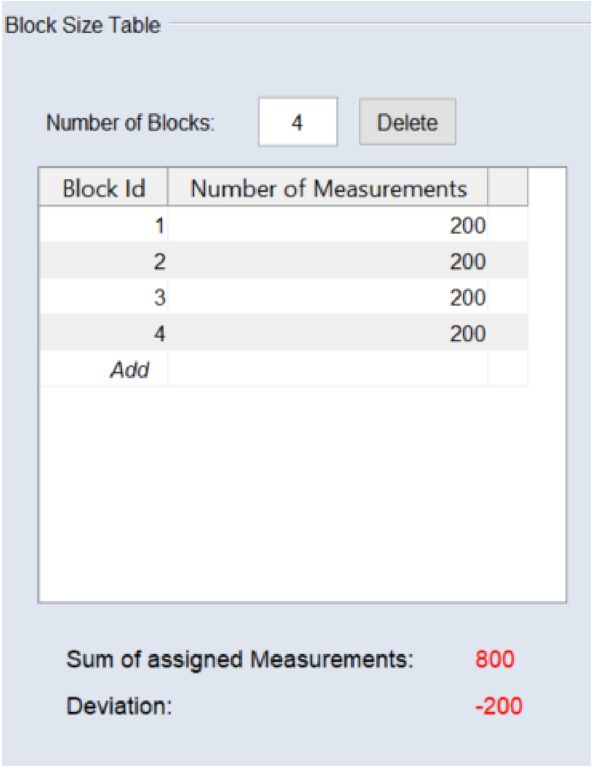
Note

If the sum of assigned measurements deviates from the current number of measurements, the experiment plan is invalid.

Deleting a block

1. In the block size table, select the block you want to delete.
2. Click **Delete**.

   The block is deleted. The number of measurements in the deleted block is displayed as deviation below the table.



3. Adjust the number of measurements in the remaining blocks so that all measurements are assigned to a block.

Block Size View Area

The size of the blocks (defined in the **Block Size Table** area) is graphically displayed here.

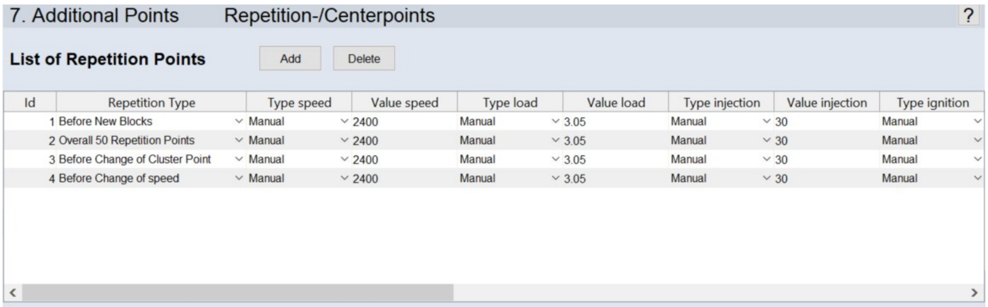
7.9 Step 7: Additional Points

In this step, additional points (also called *repetition points* because they are repeatedly measured) can be defined which, in addition to the points of the experiment plan, will be approached repeatedly according to specific criteria and measured, if necessary.

Reasons for the definition of repetition points are:

- A specific point is to be measured several times – this can help to determine, for example, drift effects over the duration of measuring.
- A repetition point can be used as a stabilization point during test automation or a measurement always has to be started from a specific operating point (e.g. regeneration of the particle filter before measuring).

A repetition point is specified by its location in space, a unique ID and the type of repetition characteristic.



<u>Adding and deleting repetition points</u>

1. Click **Add** to add a new repetition point to the list.
2. In the row of the new repetition point, do the following:
   i. In the **Repetition Type** column, select the type for approaching the repetition point.
   ii. In the **Type *<input>*** columns, select the type for determining the value of *<input>*.
   iii. In the ***<input>*** columns of the inputs with type `Manual`, enter the input value.
3. To remove a repetition point, select the ID of the point and click **Delete**.

List of Repetition Points

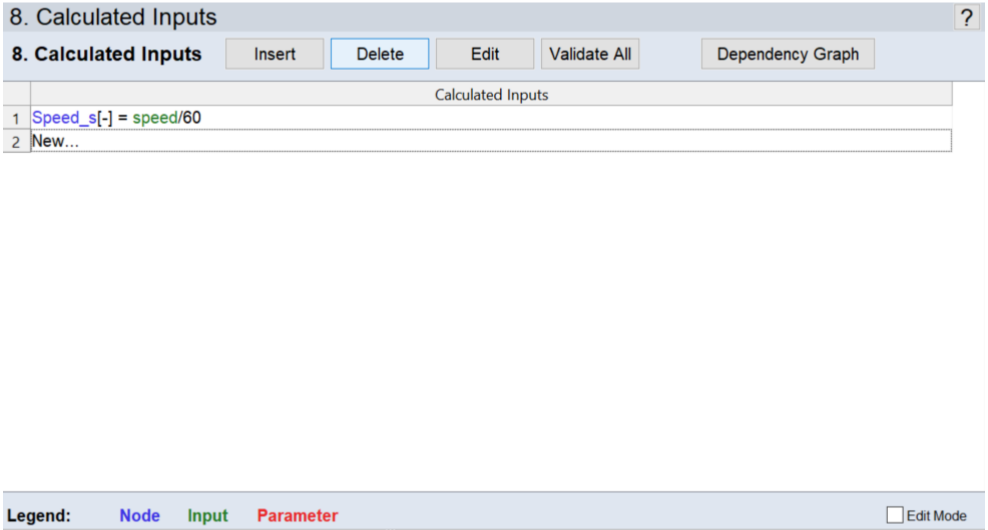
A repetition point is defined by the following parameters.

- **Id**: The unique identifier of the repetition point
- **Repetition Type**: The repetition type defines the condition to be met after which the repetition point is approached in each case. See the online help for a description of the possible repetition types.
- **Type *<input>***: The type of the input is selected here. Available selections are `Manual` and the maps and curves available in the ASCMO-STATIC ExpeDes project.
- **Value *<input>***: The value of *<input>* at the repetition point is defined here. The field can only be edited if the input uses the `Manual` type.

7.10 Step 8: Calculated Inputs

This step allows to add and specify additional inputs calculated from formulas.

The calculated inputs can be seen in all visualizations, e.g. scatter plot, scope view, table view. They are also included in the exported experiment plan.



This step is skipped in the tutorial. See the online help (F1) for further information.

7.11 Step 9: Export

NOTICE

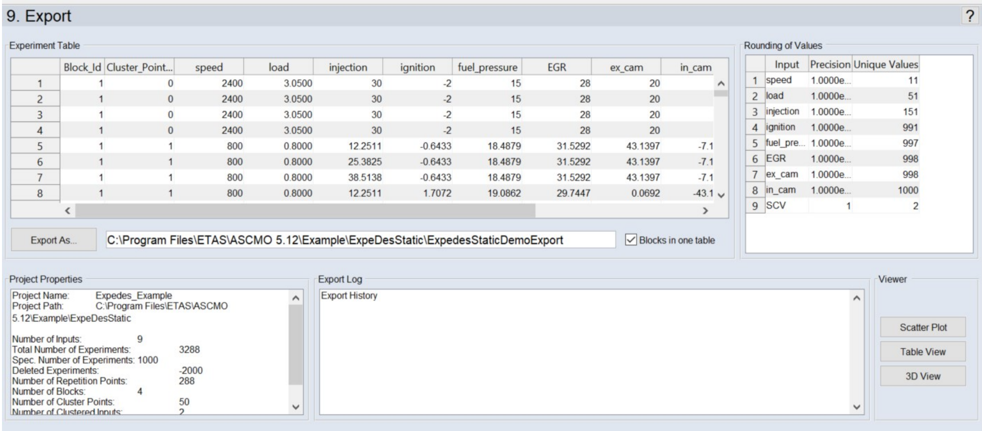
Damage due to wrong test plan

Wrong engine settings in ASCMO-STATIC ExpeDes can lead to engine or test bench damage. Example: the operation point overstresses the engine and causes damage, e.g. by setting an ignition angle that causes extensive knocking.

- The general settings for the test plan must fit the system and the object. Negative example: 10000 rpm are set in the test plan vs. the motor has max. 6000 rpm.
- Limit the operation points to the allowed values. ETAS ASCMO does not have any knowledge about the engine parameters.
- Limit the engine load in the general settings before exporting the test plan.
- Verify the test plan for further use.

For ASCMO-STATIC ExpeDes see 7.11 "Step 9: Export " aboveand 7.2 "Step 1: General Settings" on page 187.

In this step, the properties of the project and the experiment plan itself are displayed. You can export the data in `*.xlsx`, `*.xls` or `*.csv` format. In addition, you can display the data as scatter plots, 3D plots or as a table; see 7.3 "Visualizing the Experiment Plan " on page 189 for details.



Changing the rounding of values

In the **Rounding of Values** area, you specify the precision with which the relevant value can be set (e.g. the speed at an engine test bench).

1. In the **Rounding of Values** area, click in the **Precision** column of the input you want to edit.

   The cell becomes an input field.

2. Enter the desired precision value.

Exporting the plan

1. In the **Experiment Table** area, click **Export As**.

   A file selection window opens.

2. Select the export file type.

   Available formats are `*.xlsx`, `*.xls` and `*.csv`.

3. Enter or select path and file name for the export file.

   An existing file will be overwritten without warning.

4. Click **Save**.

Or

5. Click **Export** to export the plan using the most recently selected path, file name, and format.

⇨ The plan is exported according to your settings.

Experiment Table *Area*

The complete experiment plan is displayed here as it has been created by ASCMO-STATIC ExpeDes according to the settings that were performed in preceding steps.

Project Properties *Area*

In the left field of the **Project Properties** area, the properties of the project and the experiment table are displayed.

In the right field of the **Project Properties** area, the export history of the experiment table is displayed.

Glossary

A

ACF

autocorrelation function

ASC GP

ASCMO Gaussian Process

ASC GP-SCS

ASCMO Gaussian Process Sparse Constant Sigma

ASC GP-Spectrum

ASCMO Gaussian Process Spectrum

C

CCF

cross-correlation function

CNN

Convolutional Neural Networks

D

DoE

Design of Experiment

G

GPU

graphics processing unit

GRU

gated recurrent update

H

Hausdorff distance

The maximum Euclidean distance of all data points of one dataset to the data points of all other datasets.

I

IACF

inverse autocorrelation function

ISP

intersection plot

L

LSTM

long short-term memory

M

MOCA

short for ETAS ASCMO MOCA - a tool for MOdeling and CAlibration of functions with given data

N

NaN

not a number

NARX

Nonlinear Autoregression with Exogenous Inputs

O

ODCM

Online DoE with Constraint Modeling

OP

operating point

R

RDE

Real Driving Emissions

RNN

Recurrent Neural Network

T

TCN

Temporal Convolutional Network

Figures

Index

A

B

C

D

11 Contact Information

Technical Support

For details of your local sales office as well as your local technical support team and product hotlines, take a look at the ETAS website:

www.etas.com/hotlines

For ASCMO-specific inquiries, you can also contact our dedicated support team at:

ascmo.support.de@etas.com

ETAS offers trainings for its products:

www.etas.com/academy

ETAS Headquarters

ETAS GmbH

Borsigstraße 24	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	Internet:	www.etas.com