ETAS ASCMO-DYNAMIC V5.16

User Guide

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

Contents

1 **Introduction** 6

1.1 Demands on Technical State of the Product 6

1.2 Intended Use 6

1.3 Target Group 6

1.4 Classification of Safety Messages 6

1.5 Safety Information 8

1.6 Data Protection 8

1.7 Data and Information Security 8

1.7.1 Data and Storage Locations 8

1.7.1.1 License Management 9

1.7.2 Technical and Organizational Measures 9

2 **About ETAS ASCMO** 10

2.1 Fields of Application 10

2.2 Basics 11

2.3 Design of Experiments (DoE) 11

2.4 Model-Based Calibration 12

2.5 Nonlinear Autoregression With Exogenous Inputs (NARX) 13

2.6 Recurrent Neural Networks (RNN) 14

2.7 Finding Out More 14

3 **Installation** 15

3.1 System Requirements 15

3.2 Software Requirements 15

3.3 Installing 16

3.4 Files and Directories 17

3.5 P-Code Version 18

3.6 Uninstalling 19

4 **Working with ASCMO-DYNAMIC** 21

4.1 User Interface of ASCMO-DYNAMIC 21

4.2 Elements of the ASCMO-DYNAMIC User Interface 21

5 **Tutorial: Working with ASCMO-DYNAMIC** 25

5.1 Inputs and Outputs of the Measured Engine 25

5.2 Data Import 26

5.3 Data Analysis ... 29
5.3.1 Training Data Table ... 29
5.3.2 Scatter Plots ... 29
5.3.3 Resampling ... 30
5.3.4 Noise Filtering ... 31
5.3.5 Cross Correlation Inputs ... 32
5.3.6 Amplitude Spectra ... 33
5.3.7 Phase Plot, ACF and IACF ... 34
5.4 Model Training ... 36
5.4.1 Modeling Method and Algorithm ... 39
5.4.2 Feedback Structure (NARX) ... 40
5.4.3 Dimensionality Reduction ... 40
5.5 Model Prediction ... 43
5.5.1 Sigma Prediction ... 44
5.5.2 Multi-Step/One-Step Ahead Prediction ... 44
5.6 Model Validation ... 45
5.6.1 Model Validation with Scatter Plot ... 46
5.6.2 Measured vs. Predicted ... 47
5.6.3 Cross Validation ... 48
5.6.4 Cross-Correlation of the Residuals (CCR) Validation ... 49
5.7 Model Export ... 51
5.7.1 Export to MATLAB® ... 51
5.7.2 Export to Python Script ... 52
5.7.3 Export to Simulink® Model ... 52
5.7.4 Export to Simulink® Script ... 53
5.7.5 Export to INCA/MDA ... 54
5.7.6 Export to C Code ... 55
5.7.7 Export to GT-Suite ... 56
5.7.8 Export to FMI ... 57
5.7.9 Export to Keras ... 58
5.7.10 Export to Embedded AI Coder ... 59
5.7.11 Export to Bosch AMU ... 60
5.7.12 Export to Bosch Flatbuffers ... 61
6 Tutorial: Working With ASCMO-DYNAMIC ExpeDes ... 63
6.1 Starting ASCMO-DYNAMIC ExpeDes ... 64
6.2 Step 1: General Settings ... 66
6.2.1 Input Configuration ... 67
6.2.2 Measurement Size Configuration ... 68

6.3 Visualizing the Dynamic Experimental Plan 68

6.4 Step 2: Constraints 71

6.4.1 Constraint Types "Map" and "Curve" 73

6.4.2 Managing Curves and Maps 75

6.5 Step 3: Input Compression 81

6.5.1 Compression Configuration Area 82

6.5.2 View Area 82

6.5.3 Example of an Applied Input Compression 83

6.6 Step 4: Steady State Points 83

6.6.1 "Select Axes" Area 84

6.6.2 "User Defined Positions" Area 84

6.6.3 "Random Positions" Area 89

6.7 Step 5: Block Configuration 89

6.8 Step 6: Snippet Configuration 91

6.9 Step 7: Calculated Inputs 92

6.10 Step 8: Export 93

Glossary **99**

Figures **101**

Index **102**

10 Contact Information **105**

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-DYNAMIC 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-DYNAMIC ExpeDes see 6.10 "Step 8: Export" on page 93 and 6.2 "Step 1: General Settings" on page 66.

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 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 time-dependent, 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.

2.2 Basics

ETAS ASCMO-DYNAMIC enables you to create data-based models that capture the dynamic/transient behavior of complex systems. ASCMO-DYNAMIC offers a wide range of functions and options for visualizing and analyzing system behavior. Additionally, it allows for exporting to ASCMO-MOCA for optimizations and supports the creation of experimental designs based on the DoE methodology (design of experiments).

Using AI methods from the field of machine learning, ASCMO-DYNAMIC allows you to accurately model complex relationships without requiring precise knowledge of the underlying algorithms. This flexibility makes the software suitable for a wide range of users, including less experienced users who appreciate parameter-free, automated model creation, as well as experts who benefit from extensive configuration options.

A typical application of ASCMO-DYNAMIC is the modeling of transient processes in internal combustion engines, particularly in the context of Real Driving Emissions (RDE), where purely stationary consideration of emissions and fuel consumption may not be sufficient. Dynamic effects, such as peaks, can significantly influence the overall outcome. When modeling relevant variables like fuel consumption and pollutant emissions, historical values and the rate of change are taken into account, enabling a detailed, time-resolved analysis of the influence of dynamic driving maneuvers on the relevant output variables.

The methodology used in ASCMO-DYNAMIC is not limited to the internal combustion engine, making the tool applicable in areas such as electric mobility (e.g., battery modeling).

2.3 Design of Experiments (DoE)

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

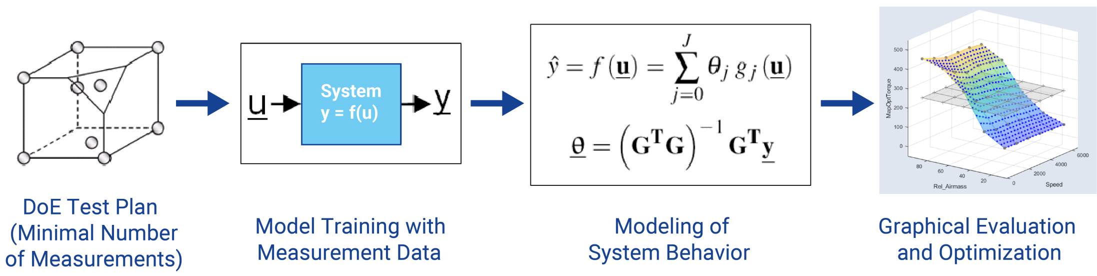


Fig. 2-1: From experiment plan to model-based optimization

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.

2.4 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.

2.5 Nonlinear Autoregression With Exogenous Inputs (NARX)

Dynamic effects with a data-driven modeling algorithm, e.g. the ASC modeling part of ASCMO-DYNAMIC, can be taken into consideration by using a superordinate model structure. In the discrete time case, the system input space is expanded with the feedback of past input and output values up to a certain time horizon, as shown in Fig. 2-2. This is referred to in literature as *nonlinear autoregression with exogenous inputs* (NARX). The feedback values are referred to as *features* in the following.

With this approach, the dynamic identification problem is transformed into a quasi-stationary relationship:

`y(k)=f(x_1 (k),x_1 (k-1),…,x_2 (k),x_2 (k-1),…,y(k-1) ,…)`

where k indicates the discrete time-step.

Based on a continuously measured data set of the system inputs and outputs, the ASC modeling algorithm, or any other data-driven regression, can be applied for the modeling of the functional relationship `f( )`.

Multi-Step vs. One-Step Ahead Prediction

After model training, we can distinguish between two scenarios for the application of the model in the prediction (see Fig. 2-2):

- **One-step ahead prediction**

  In the case of a one step-ahead prediction, the past system outputs are known and given by actual measurements, e.g. through sensors. The model has to predict just the upcoming time step.

- **Multi-step ahead prediction**

  In the multi-step ahead prediction, the past system output values are replaced by the model’s predictions. This corresponds to an offline simulation and is the standard use case, where the user wants the model response to a given set of input variations.

  Compared to the one-step ahead implementation, the multi-step case usually results in a worse model quality due to the accumulation of prediction errors.

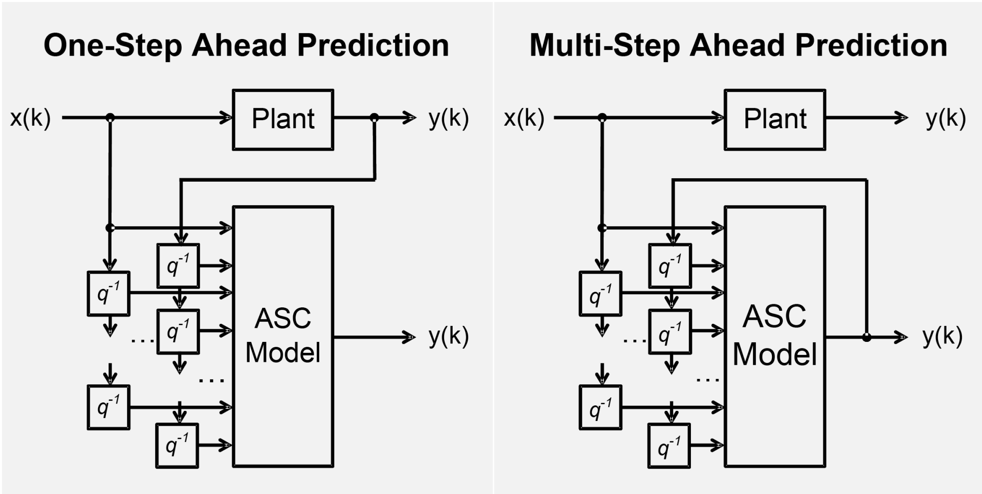


Fig. 2-2: Model structure for nonlinear autoregression with exogenous inputs (NARX)

See also section 5.5.2 "Multi-Step/One-Step Ahead Prediction " on page 44.

2.6 Recurrent Neural Networks (RNN)

ASCMO-DYNAMIC offers the possibility to use *recurrent neural networks* (RNN) for transient modeling.

The underlying basis for this model type is the open-source machine learning platform Tensorflow.

Recurrent neural networks are distinct from traditional neural networks, such as simple feed-forward networks like the single hidden layer network that can be found in one of the ASCMO-MOCA example projects. In contrast to such so-called perceptrons, RNN cells maintain a state that is reused, in addition to trainable weights and biases. This state is updated when an RNN is looking at an input, and it is made available to the cell when looking at the next input handed to it. Consequently, the cell can be thought of as having memory, since the results of previous evaluations of inputs can now influence the results of input evaluations at later times. Therefore, RNNs are very interesting candidates for building models from sequential data.

More information on RNN can be found in the online help (Press F1 and navigate to **ASCMO-DYNAMIC › Working with ASCMO-DYNAMIC › Model Training › Modeling with RNN**).

2.7 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.

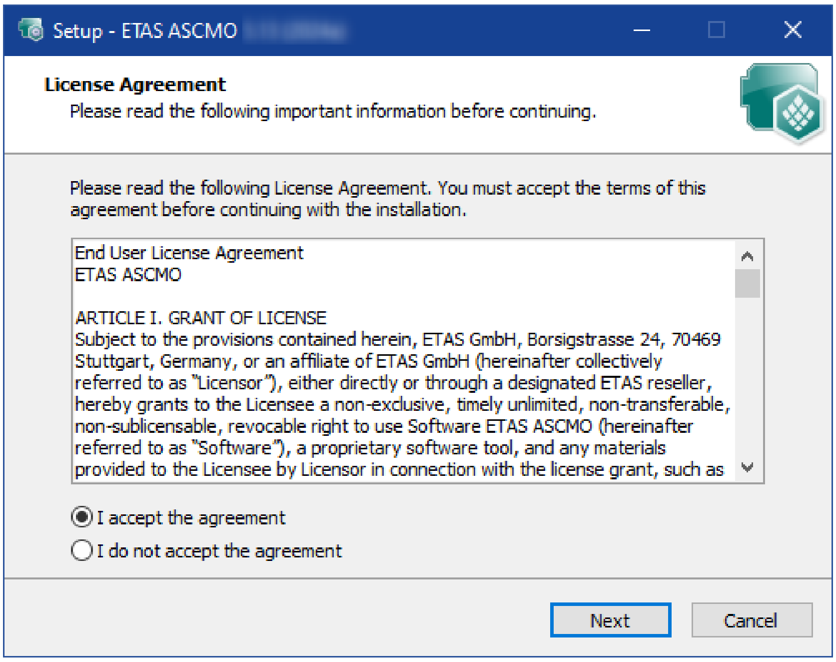
To use the ETAS ASCMO add-on *Software Developer Kit (SDK)*, a MATLAB® version R2021b up to R2023b and the MATLAB® *Optimization Toolbox* and *Statistics Toolbox* are required.

3.3 Installing

Install ETAS ASCMO

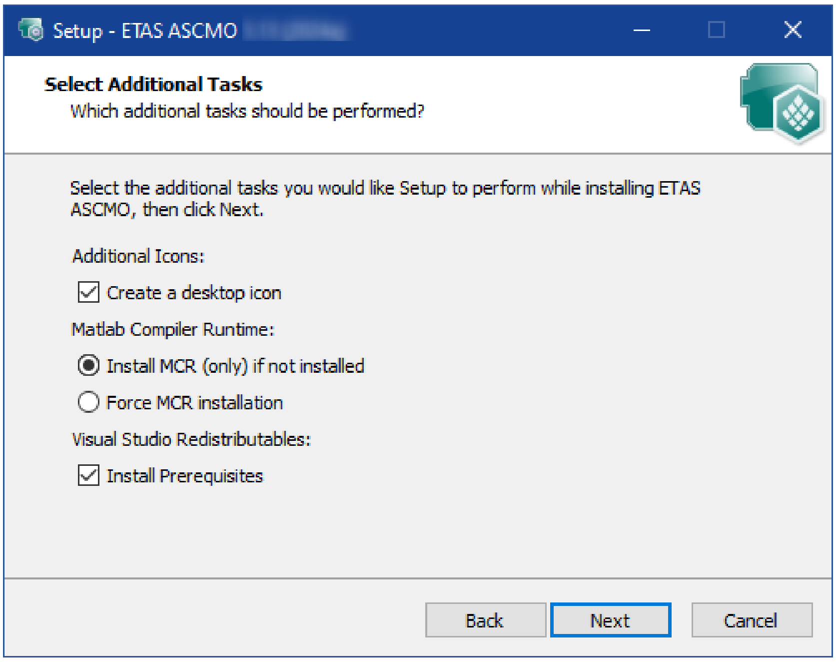
1. Go to the directory where the ASCMOinstallation file is located.
2. Double-click `Setup_ETAS-ASCMO_Vx_x_20xxx.exe`.

   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 install-ation'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 Working with ASCMO-DYNAMIC

4.1 User Interface of ASCMO-DYNAMIC

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

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**).

4.2 Elements of the ASCMO-DYNAMIC User Interface

The main window of ASCMO-DYNAMIC consists of the following areas:

- ① Main menu
- ② Toolbar
- ③ Inputs

  with "Dataset" combo box and ⚙ button (directly below the toolbar)
- ④ Outputs
- ⑤ Log window
- Status bar (footer) with current state information

"Data Set" Drop-down

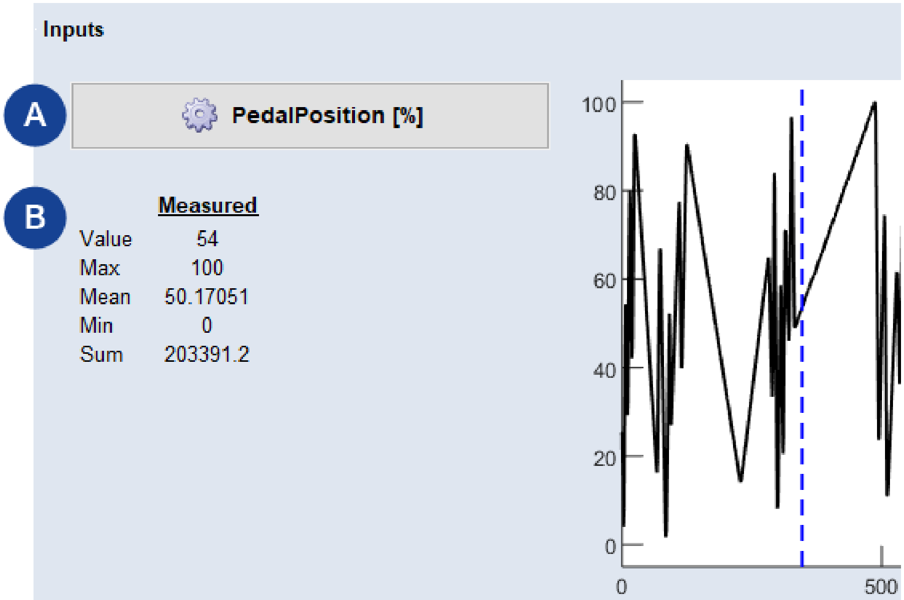
This drop-down offers all sets of training or test data for selection. The selec-ted dataset is displayed in the "Inputs" and "Outputs" areas.

The ⚙ button opens the "Manage Datasets" window.

"Inputs" Area

Information on the inputs x_1,..., x_n is displayed on the left of the time-based plot in the upper area of the ASCMO-DYNAMIC main window.

> **Note**
>
> You can hide information about the inputs by selecting **View › Show ***.



- **A** Input name and unit

  Click the button to change the name and unit.

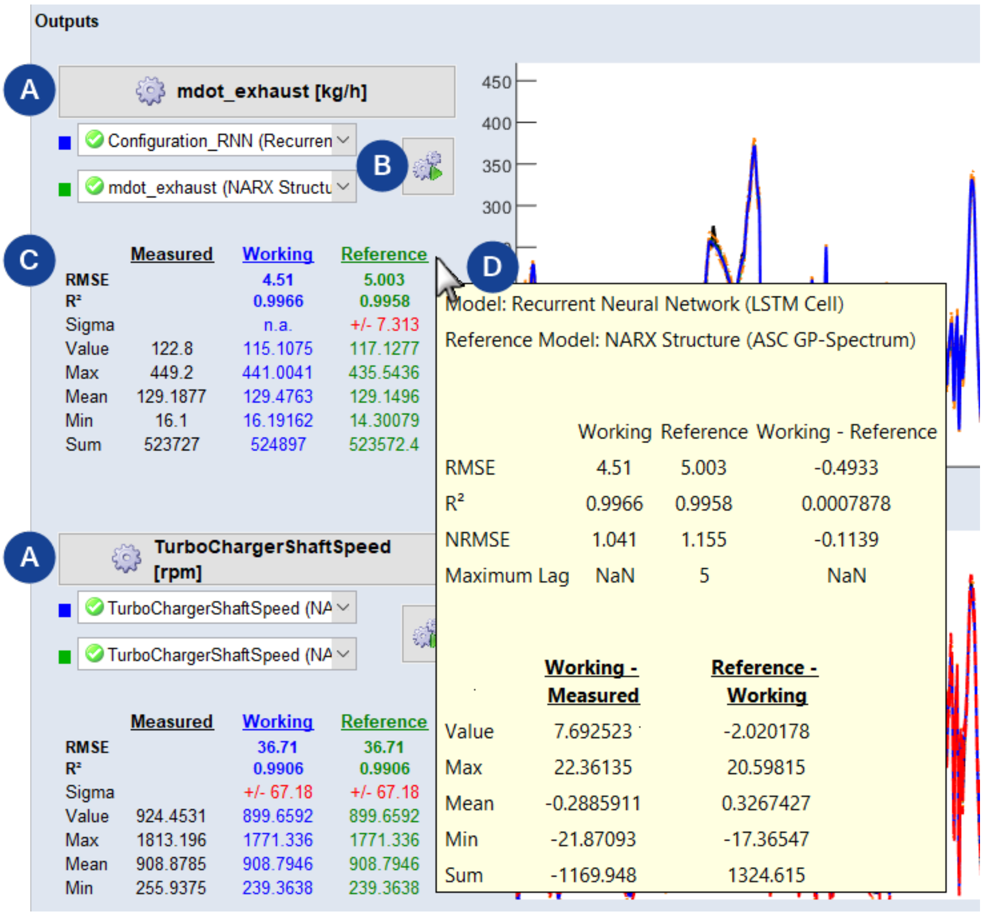
- **B** Current values of the input
  - **Value**: The current value at the specified position in the plot.
  - **Max**, **Mean**, **Min**: The maximum, mean, and minimum values of the respective input.
  - **Sum**: The sum of all input values.

If you want to display the value for another time position in the plot, click on the desired position with the cursor.

"Outputs" Area

Information on the modeled outputs is displayed on the left of the time-based plot in the lower area of the ASCMO-DYNAMIC main window.

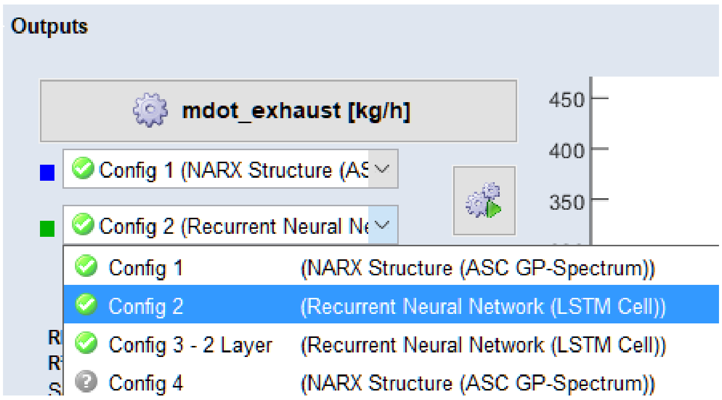
> **Note**
>
> You can hide information about the outputs by selecting **View › Show *** .



– **A** Output name

Click the button to open the "Model Configurations" window.

The respective model type selection for working (blue) and reference (green) can be set in the drop-down. The modeling method and the associated model type is set in the configuration of the output (**Model › Configurations**).



Multiple models can be trained and easily compared for a specific output. You can define several configurations for an output (**Model › Configurations**). Within the configuration the modeling method and model type are defined. That enables you to apply different modeling methods for an output in an easy way. Additionally, a configuration can also be reused and applied to other outputs for the working or reference model. The concept and settings will be described in detail in the upcoming online help.

- **B** Model Training

  Trains models with selected configurations. You can choose between working model, reference model or both.

- **C** Current values of the output

  - **RMSE**: Displays the root mean square error.
  - **R^2**: Displays the model's uncertainty.
  - **Sigma**: Displays the model's uncertainty at the specified time in the plot.
  - **Value**: The current output value at the specified position in the plot.
  - **Max**, **Mean**, **Min**: The maximum, mean, and minimum values of the respective output.
  - **Sum**: The sum of all output values.

  > **Note**
  >
  > If you have defined a model as reference model and selected in the drop-down, you can see the values of the reference model.

- **D** Tooltip

  Shows the actual values of the output and the difference between Working - Measured and Reference - Working in an overview.

If you want to display the value or the model sigma for another time position in the plot, click on the desired position with the cursor.

Log Window

In this area, notes, status messages and results (e.g. of optimization runs) are shown.

5 Tutorial: Working with ASCMO-DYNAMIC

This tutorial introduces the basic functions of *ASCMO-DYNAMIC*, the ETAS ASCMO toolbox for dynamic system identification. The tutorial is structured as follows:

- 5.1 "Inputs and Outputs of the Measured Engine" below
- 5.2 "Data Import" on the next page
- 5.3 "Data Analysis " on page 29
- 5.4 " Model Training" on page 36
- 5.5 "Model Prediction " on page 43
- 5.6 "Model Validation" on page 45
- 5.7 "Model Export " on page 51

5.1 Inputs and Outputs of the Measured Engine

The example (a diesel engine) measured with respect to the inputs and outputs given in Fig. 5-1: below and Tab. 5-1: below.

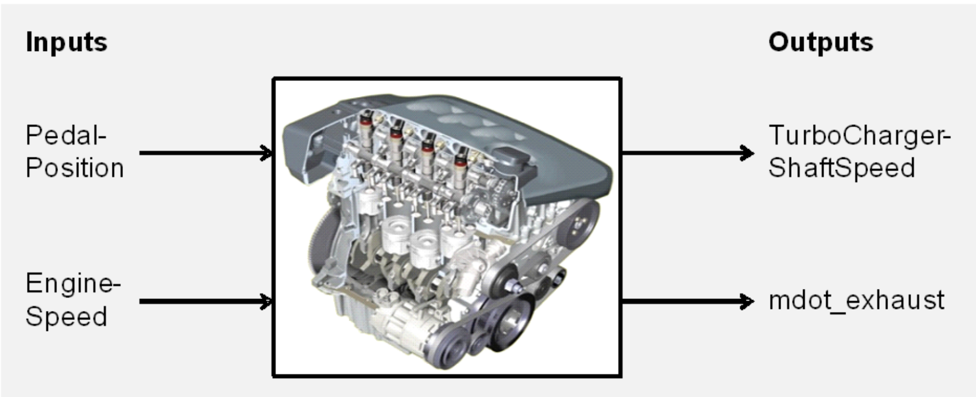


Fig. 5-1: Illustration of the Measured Diesel Engine

	Name	Meaning
Inputs	PedalPosition	Position of the throttle [%]
	EngineSpeed	Engine speed [rpm]
Outputs	TurboChargerShaftSpeed	Speed of the turbocharger shaft [rpm]
	mdot_exhaust	Exhaust gas mass flow [kg/h]

Tab. 5-1: Meaning of Inputs and Outputs of the Measured Diesel Engine

Data for Modeling

The available data set consists of a continuous data stream of 18000 measurements, where 4000 samples are used for modeling and the remaining 14000 points representing the test data set. The sampling interval was 0.1s.

5.2 Data Import

This section describes the training data import.

Start ASCMO-DYNAMIC

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.

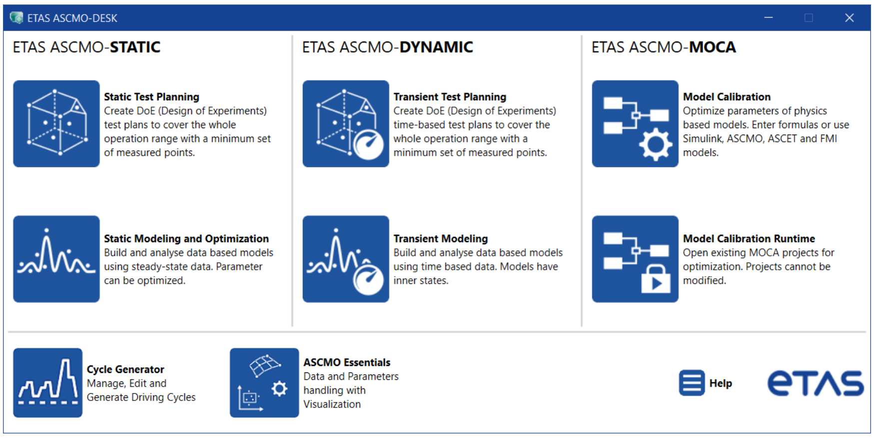


Fig. 5-2: ASCMO-DESK window

2. In the **ASCMO-DESK** window, click the **Transient Modeling** tile.

➪ ASCMO-DYNAMIC opens.

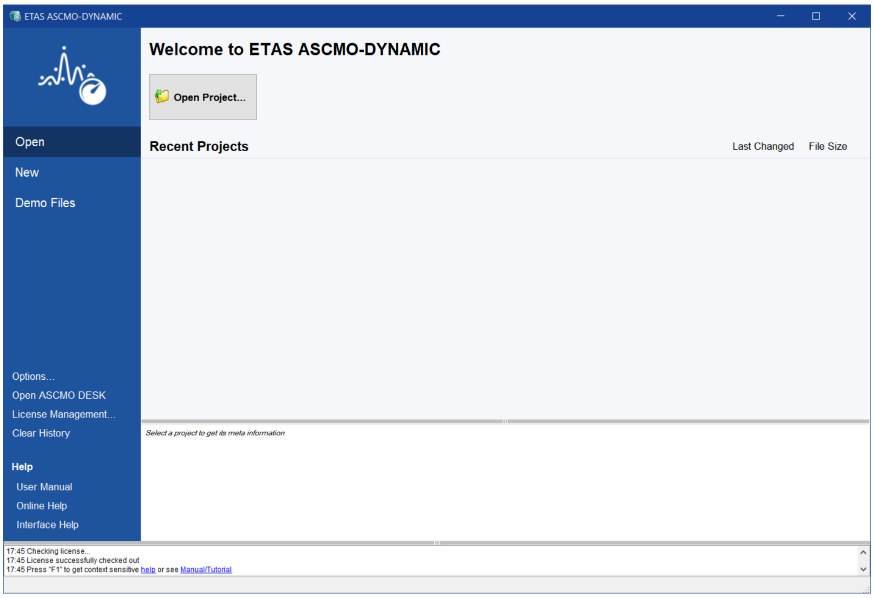


Fig. 5-3: ASCMO-DYNAMIC start window

Load and import training data

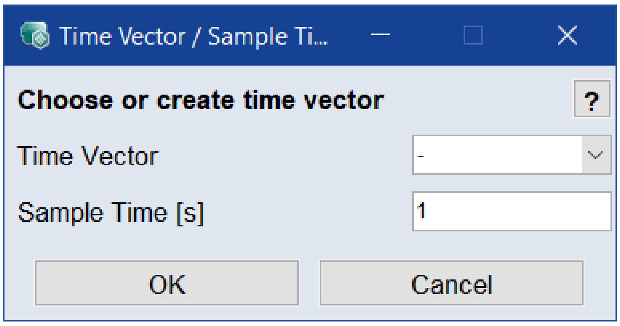
If you want to start with a new project, you first have to load the training data required for the model training.

1. In the ASCMO-DYNAMIC start window, click **New** in the menu panel on the left.
2. Click 

.

   The **ASCMO Data Import** window and an open file dialog opens.
3. In the open file dialog, select the file that contains the training data.
4. Click **Open**.
5. If you selected a `*.xls` file, continue as follows:
   i. If the file contains several work sheets, select the desired work sheet (**Training Data**).
   ii. Click **OK**.

      The **Time Vector / Sample Time** window opens.



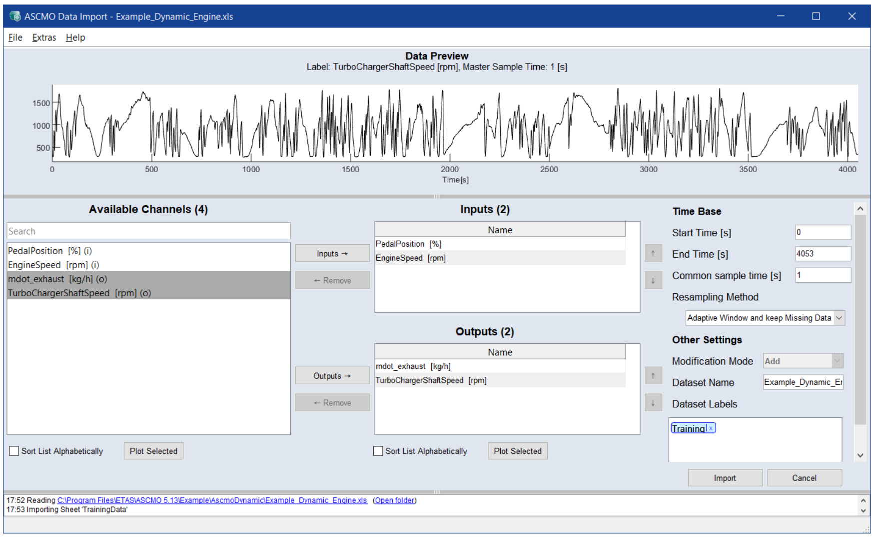
   iii. In the **Sample Time** field, enter the sample time (in seconds) of the training data.

      Since the training data table contains no time column, you can ignore the **Time Vector** field in this tutorial.
   iv. Click **OK**.

      The data are read in.

      Note

      Depending on the file size, this process may take some time.
6. Select `PedalPosition` and `EngineSpeed` as inputs and `mdot_exhaust` and `TurboChargerShaftSpeed` as outputs.



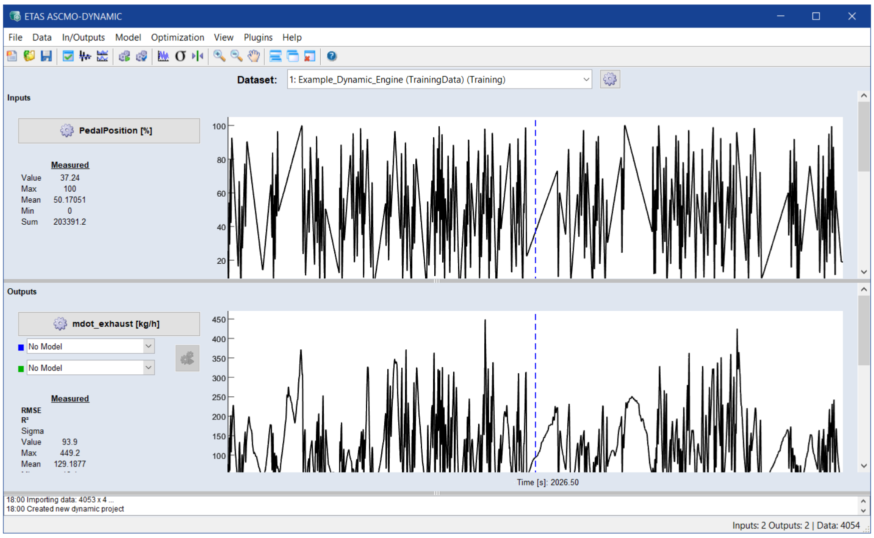
Note

As opposed to ASCMO-STATIC, ASCMO-DYNAMIC does not consider operating points (only **Normal** is possible).

Similar to ASCMO-STATIC, the import data interface also provides the user with a graphical plausibility check (**Plot Selected**) and the saving/loading of the selected input-output configuration (**File › Save/Load Channel Config (*.ini)**).

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

➩ The measuring data are imported and the project is created. After finishing the data import, the window displays the corresponding time series of the inputs (top) and outputs (bottom).



Save a configuration

1. Select **File › Export Data › Channel Config...**.
2. In the file selection window, enter path and name of the target file, then click **Save**.

Load a configuration

1. To load a previously saved configuration, select **File › Import Data › Training / Validation / Test**.
2. In the new window, select **File › Load Channel Config (*.ini, *.lab)**.

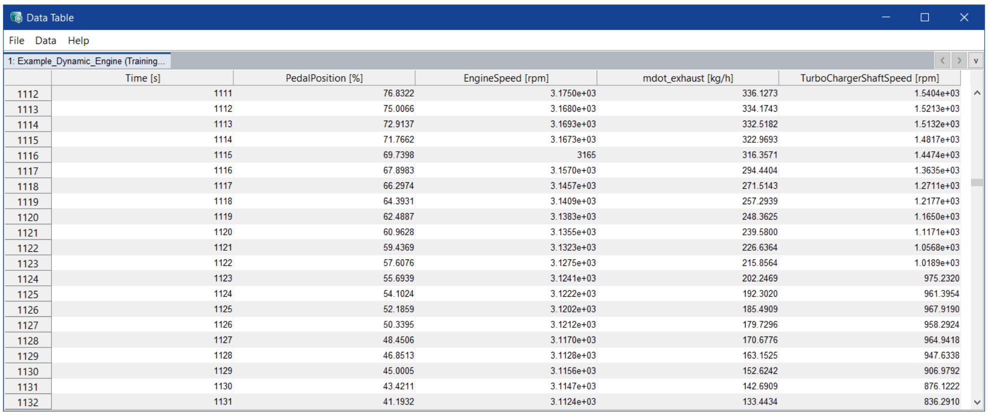
5.3 Data Analysis

Even though the actual modeling of ASCMO-DYNAMIC (see 5.4 " Model Training" on page 36) is straightforward and does not require any data preprocessing or user experience, an analysis of the loaded data set is recommended.

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

5.3.1 Training Data Table

A table of the time series can be found under **Data › Table › Training Data**.



	Time [s]	PedalPosition [%]	EngineSpeed [rpm]	mdot_exhaust [kg/h]	TurboChargerShaftSpeed [rpm]
1112	1111	76.8322	3.1750e+03	336.1273	1.5404e+03
1113	1112	75.0066	3.1680e+03	334.1743	1.5213e+03
1114	1113	72.9137	3.1693e+03	332.5182	1.5132e+03
1115	1114	71.7662	3.1673e+03	322.9693	1.4817e+03
1116	1115	69.7398	3165	316.3571	1.4474e+03
1117	1116	67.8983	3.1570e+03	294.4404	1.3635e+03
1118	1117	66.2974	3.1457e+03	271.5143	1.2711e+03
1119	1118	64.3931	3.1409e+03	257.2939	1.2177e+03
1120	1119	62.4887	3.1383e+03	248.3625	1.1650e+03
1121	1120	60.9628	3.1355e+03	239.5800	1.1171e+03
1122	1121	59.4369	3.1323e+03	226.6364	1.0568e+03
1123	1122	57.6076	3.1275e+03	215.8564	1.0189e+03
1124	1123	55.6939	3.1241e+03	202.2469	975.2320
1125	1124	54.1024	3.1222e+03	192.3020	961.3954
1126	1125	52.1859	3.1202e+03	185.4909	967.9190
1127	1126	50.3395	3.1212e+03	179.7296	958.2924
1128	1127	48.4506	3.1170e+03	170.6776	964.9418
1129	1128	46.8513	3.1128e+03	163.1525	947.6338
1130	1129	45.0005	3.1156e+03	152.6242	906.9792
1131	1130	43.4211	3.1147e+03	142.6909	876.1222
1132	1131	41.1932	3.1124e+03	133.4434	836.2910

5.3.2 Scatter Plots

The scatter plots (shown via **Data › Scatter Plot › Training Data**) show the coverage of the training data in the possible input and output range. By marking a block in the **Data Set vs. Time plot**, the corresponding input and output values at these time steps of the time series can be visualized.

The scatter plots will be useful later during model validation (see section 5.6 "Model Validation" on page 45) to check whether the data is in the range of the input data. This is crucial since ASCMO-DYNAMIC uses a data-driven modeling algorithm, which cannot guarantee a reliable extrapolation into areas not covered.

The content of the scatter plots can be changed with **Configuration › X-Axis / Y-Axes** in the scatter plot windows.

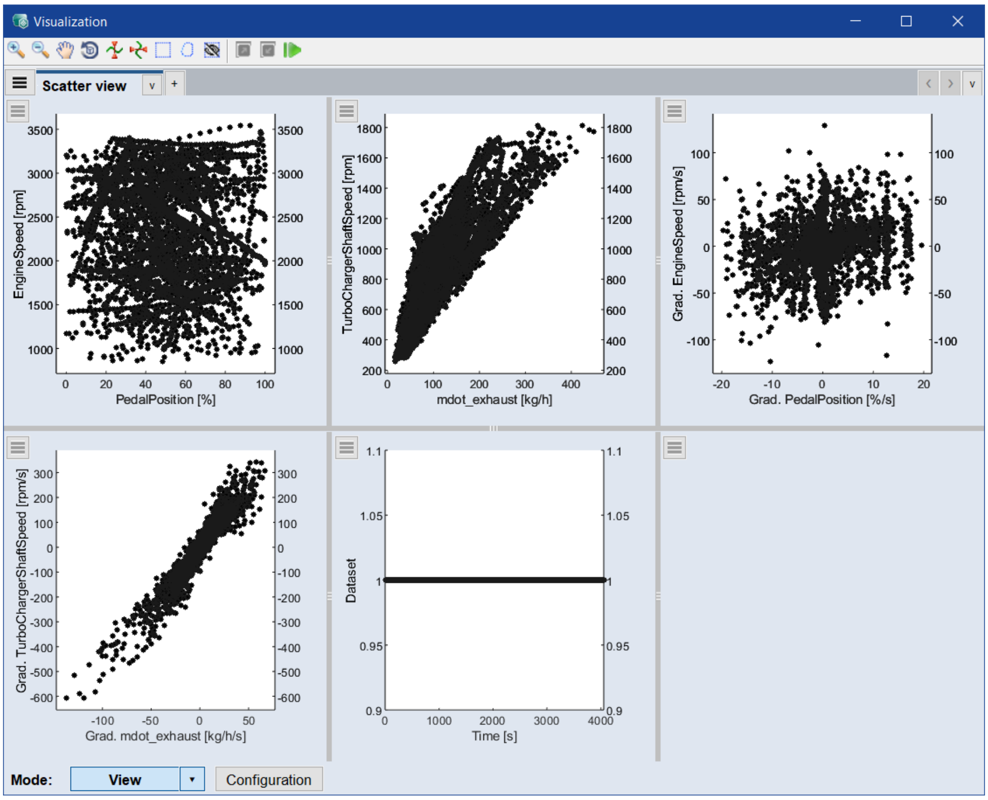
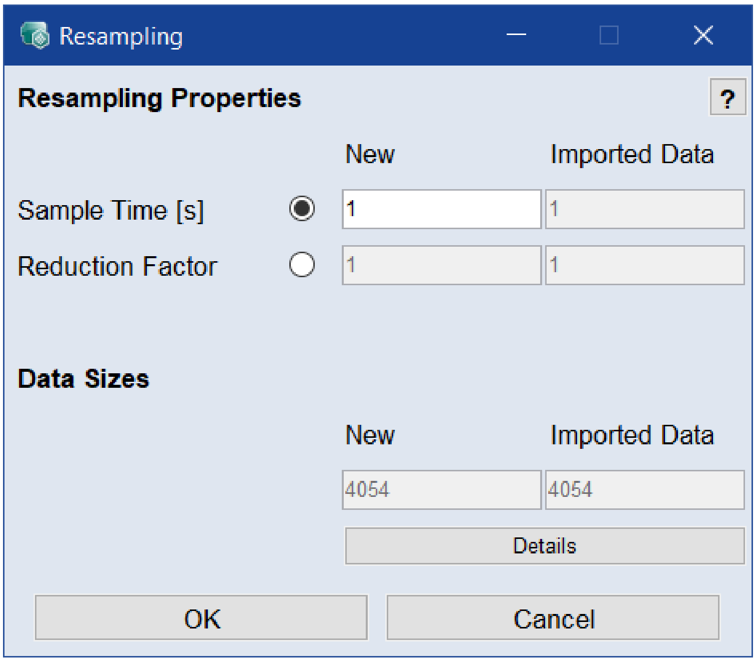


Fig. 5-4: ASCMO-DYNAMIC scatter plots (default plots)

5.3.3 Resampling

Sometimes the measured data is sampled within too small a sampling time `Ts` (sampling frequency too high). A proper sampling time is about `Ts ≤ T/10`, where `T` is the system's dominant time constant. The down-sampled data set will also benefit the later modeling by reducing the training time. The resampling process is started with **Data › Resampling**.

The user only provides the resampling factor `n`, which means that the new time series is generated by picking every n-th data point out of the original series.



By setting the **Sample Time**, the absolute time difference in seconds between two samples of the signal can be defined. The **Reduction Factor** is a factor relative to the sampling rate of the original loaded signal (e.g., a reduction factor of 2 doubles the absolute time difference between two samples of the original data). Both settings lead to a resampling of the original data, which is possible based on linear interpolation.

The information fields concerning the data sizes show the number of samples using the current sampling rate as well as the original one.

The new and the original training data will be calculated and displayed automatically in the **Resampling** window.

Note

For the diesel engine example, resampling is not necessary.

5.3.4 Noise Filtering

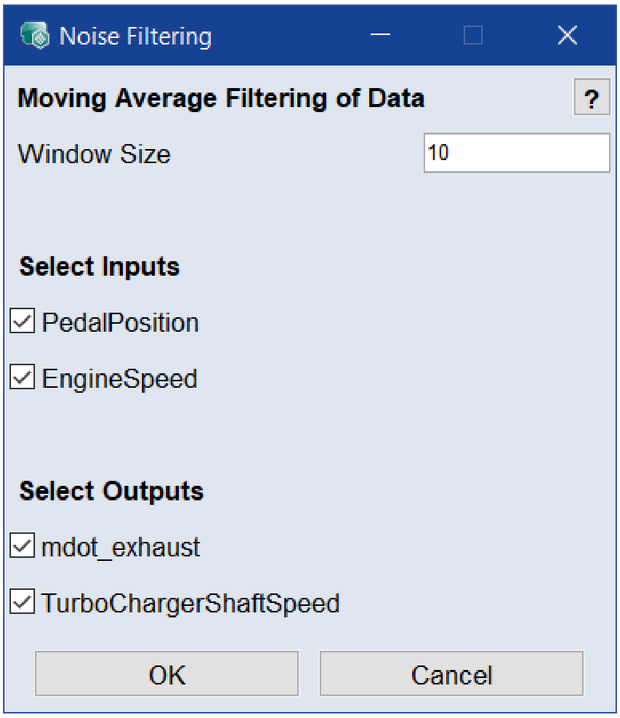
To get rid of signal disturbances, the dataset can be modified by a moving average filter (**Data › Noise Filtering**), where the new signal is generated as follows:

$$x(k) = \frac{1}{n+1}\sum_{t=0}^{n} x(k-t)$$

n = filter window size

The window size n is given in time steps, so it depends on the current sample time.

If the sample time is 0.1s and the window size is 10, the moving average window has a width of 1s.



A measurement noise can be identified, for instance, with the amplitude spectrum of the signal (see 5.3.6 "Amplitude Spectra" on the next page).

Note

For the diesel engine example, noise filtering is not necessary.

5.3.5 Cross Correlation Inputs

Ideally, all inputs are uncorrelated. This can be checked with a cross-correlation analysis (via **Data › Cross Correlation Inputs**).

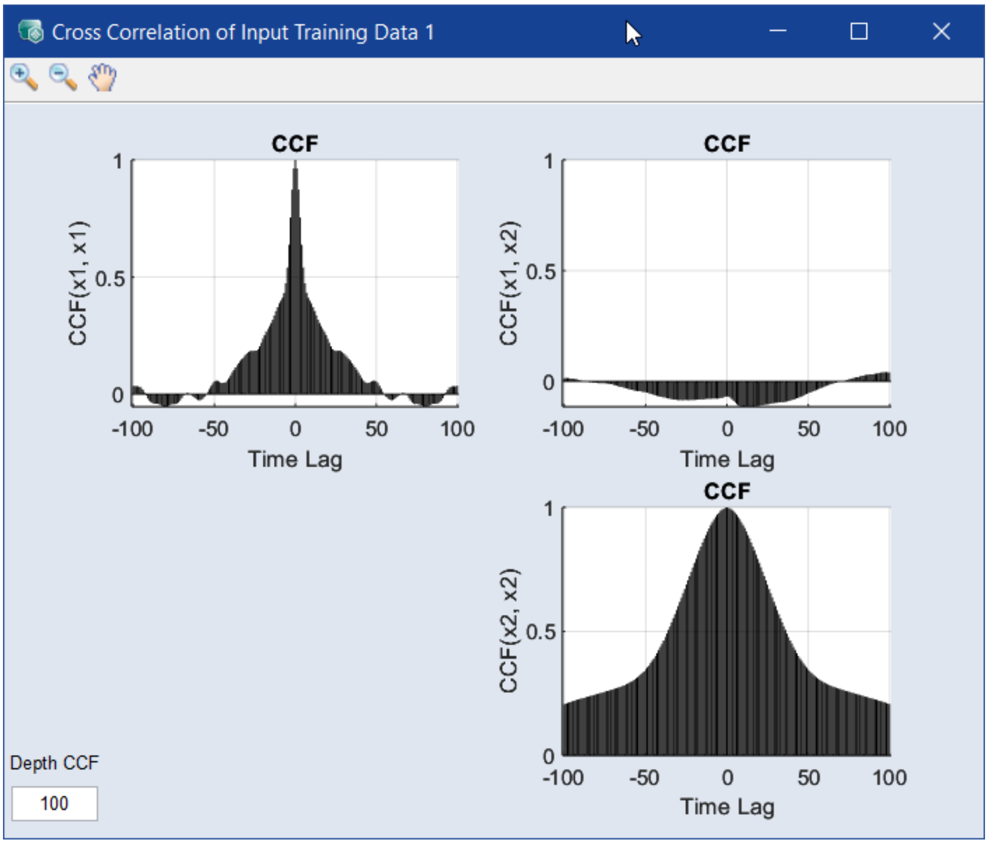
Note

The cross-correlation analysis is only available if the Advanced Settings are enabled.

The desired behavior is shown below using the diesel engine example. The plots $CCF(x_i, x_i)$ show the cross-correlation of each input with itself, which is actually the auto-correlation of the signal, depending on the respective time lag.

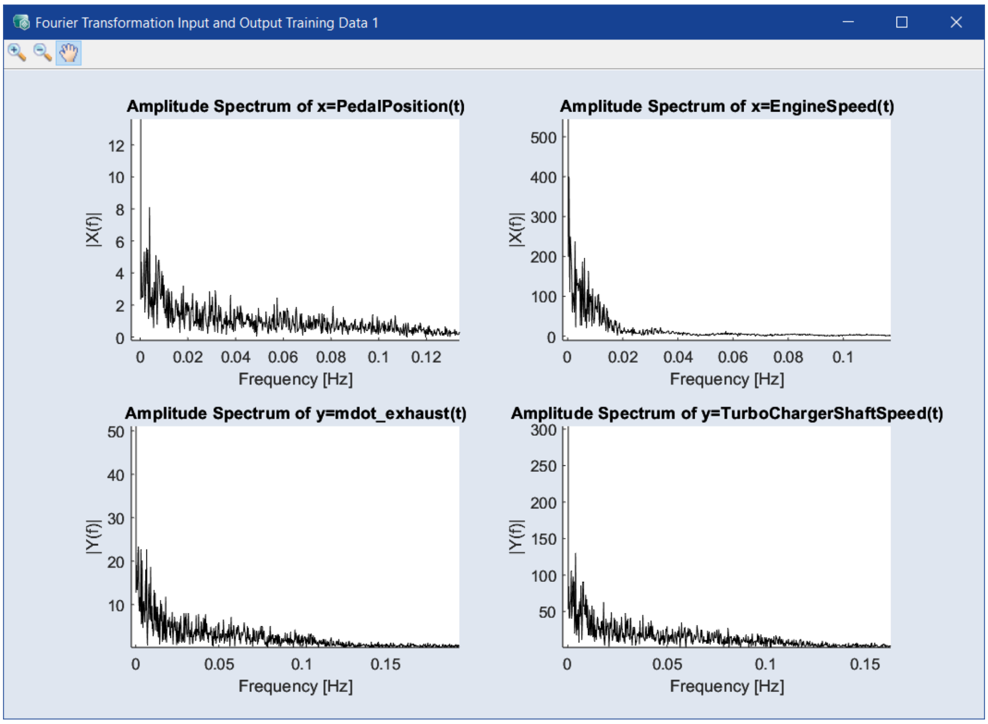
In an ideal case, one would get an exponentially decreasing function with a (full) correlation of 1.0 at time lag 0. The slower the CCF/ACF decreases, the smaller the signal changes are over time.

The more interesting plots are the CCFs among the inputs, which are displayed off-diagonal. The desired case is no correlation, as in the diesel engine example. Any peaks within the cross correlation point to correlated input channels – a cross correlation of 1.0 would indicate redundant signals.



5.3.6 Amplitude Spectra

With **Data › Spectrum of Channels** you can display the amplitude spectra of the inputs and outputs. More or less narrow peaks indicate the presence of noise of more or less defined frequency.



5.3.7 Phase Plot, ACF and IACF

Data › Phase Plot and IACF Outputs makes it possible to identify the time dependency of the identification task. Each output is shown in a different window.

Phase Plot

A scatter plot of `output(k)` over `output(k-t)`; the phase/lag `t` can be adjusted in the lower left corner of the window (the standard value is 1).

This plot (top half of Fig. 5-5: on the next page) shows the changes of the output values over time from one step to the `t`-th following. Many points on the diagonal with slight scattering indicate a strong dependency between the `output(k)` and the phase-shifted `output(k-t)`. This is normally the case for a phase of `t = 1`. If you increase `t` in steps, the point distribution approximates equal distribution – this means you can estimate the time dependency of an output.

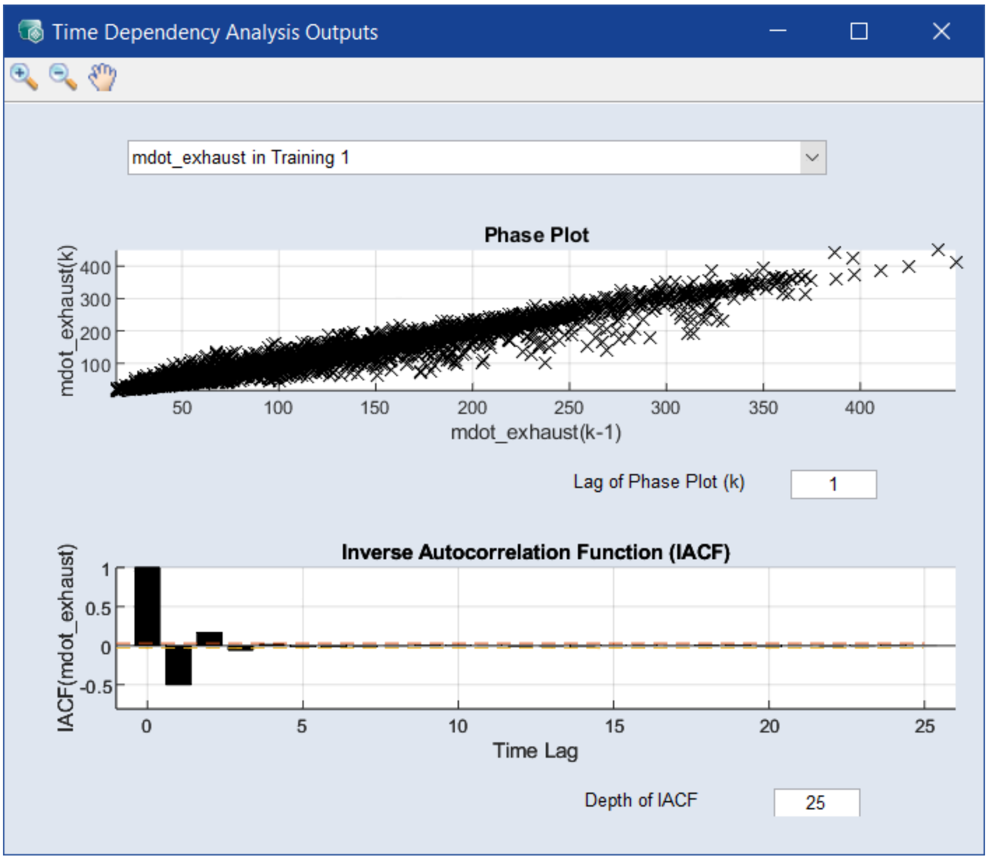


Fig. 5-5: Phase plot and ACF/IACF plot for an output

Autocorrelation Function (ACF)/Inverse Autocorrelation Function (IACF)

The analysis with ACF and IACF assumes a linear time series with no external input variables involved. Of course, this is not the case in the identification of non-linear systems, e.g. the diesel engine example, but we can draw some first assumptions from this about the present time dependency.

The expected shape of the ACF is exponentially decreasing or sinusoidal. A sudden decay after a few time lags would indicate no autoregressive part (no time dependency) within the time series.

The IACF plot (bottom half of Fig. 5-5: above) should decay after a few steps, which gives the order of the autoregressive part. The dashed line represents the cut-off criterion, which is the 95% confidence interval:

$$\frac{1.96}{\sqrt{Length\ of\ data\ set}}$$

Exponential or sinusoidal behavior of the IACF would indicate no autoregression. In the diesel engine example, the IACF decays after a signal shift of 3 time lags. This is the time lag horizon to be considered for the upcoming model training.

5.4 Model Training

There are two ways to start the model training, either via **Model › Configurations** ("Model Configurations window" on the next page) or via **Model › TrainModels** ("Model Configuration All Outputs window" on page 38).

Starting the training via **Model › Configurations** allows you to set the model properties individually for each output. They concern the following topics:

- modeling method (for this tutorial: **NARX Structure**)
- method parameters

  For the **NARX Structure** method, these parameters are:

  - modeling algorithm (see also "Setting global model configurations and starting the training" on page 38)
  - feedback (NARX) structure (see also Feedback Structure)
  - dimensionality reduction (see also Dimensionality Reduction)

<u>Setting individual model properties and starting the training</u>

1. Select **Model › Configurations**.

   The **Model Configurations** window opens. It contains a separate tab for each output.

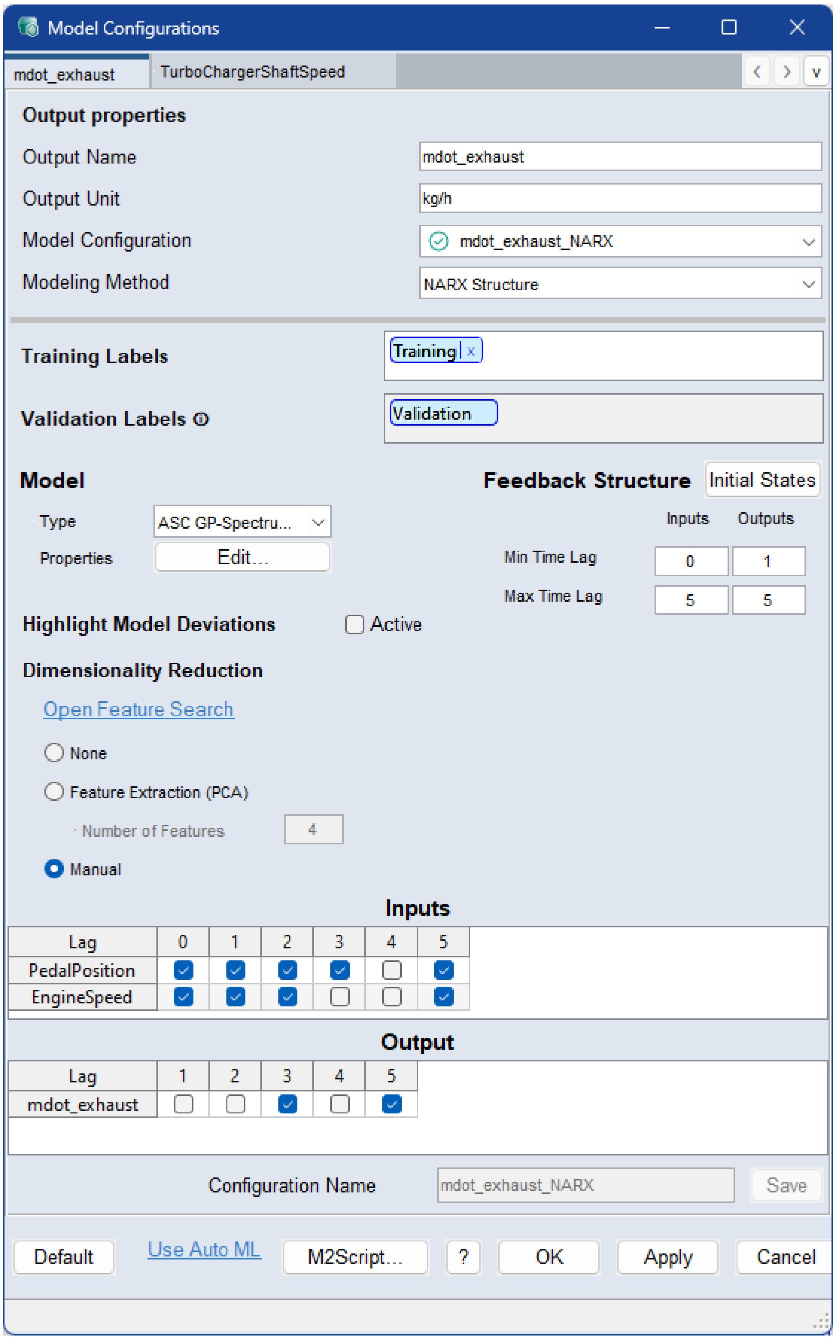


Fig. 5-6: Model Configurations window

2. In the **Modeling Method** drop-down list, select `NARX Structure`.
3. In the **Model** area, do the following:
   i. In the drop-down list, select a model type. See also 5.4.1 " Modeling Method and Algorithm" on page 39.

      The **<output> - Parameters** window opens. Its content depends on the selected model type.

   ii. In the **<output> - Parameters** window, set the model parameters.

      See "NARX Model Types" in the online help for a description of the window.

iii. Click **OK** to close the **‹output› - Parameters** window.

You can also open the **‹output› - Parameters** window with the **Edit** button.

4. In the **Feedback Structure** area, enter the minimum and maximum time lags for inputs and outputs.

   See also 5.4.2 "Feedback Structure (NARX)" on page 40.

5. In the **Dimensionality Reduction** area, activate **None**.

   None is sufficient for the diesel example. For further information, see 5.4.3 "Dimensionality Reduction " on page 40

6. Once you have set model parameters for all outputs, click **OK** or **Apply** to start the training.

<u>Setting global model configurations and starting the training</u>

Starting the training via **Model › Configuration for All** allows you to set model parameters for all outputs.

1. Select **Model › Configuration for All**.

   The **Model Configuration All Outputs** window opens.

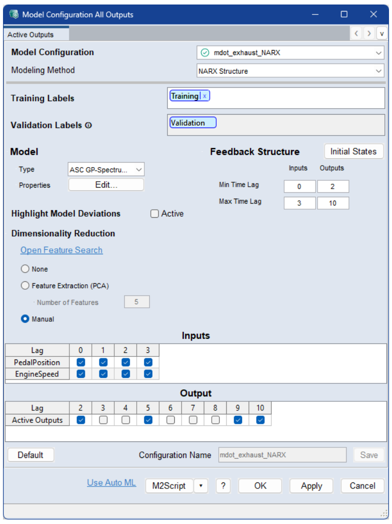


Fig. 5-7: Model Configuration All Outputs window

   This window contains the same elements as the **Model Configurations** window (Fig. 5-6: on the previous page), except that there is only one tab for all outputs.

2. Set the model properties as described in "Setting individual model properties and starting the training" on page 36.

> **Note**
>
> The settings you make affect **all** outputs.

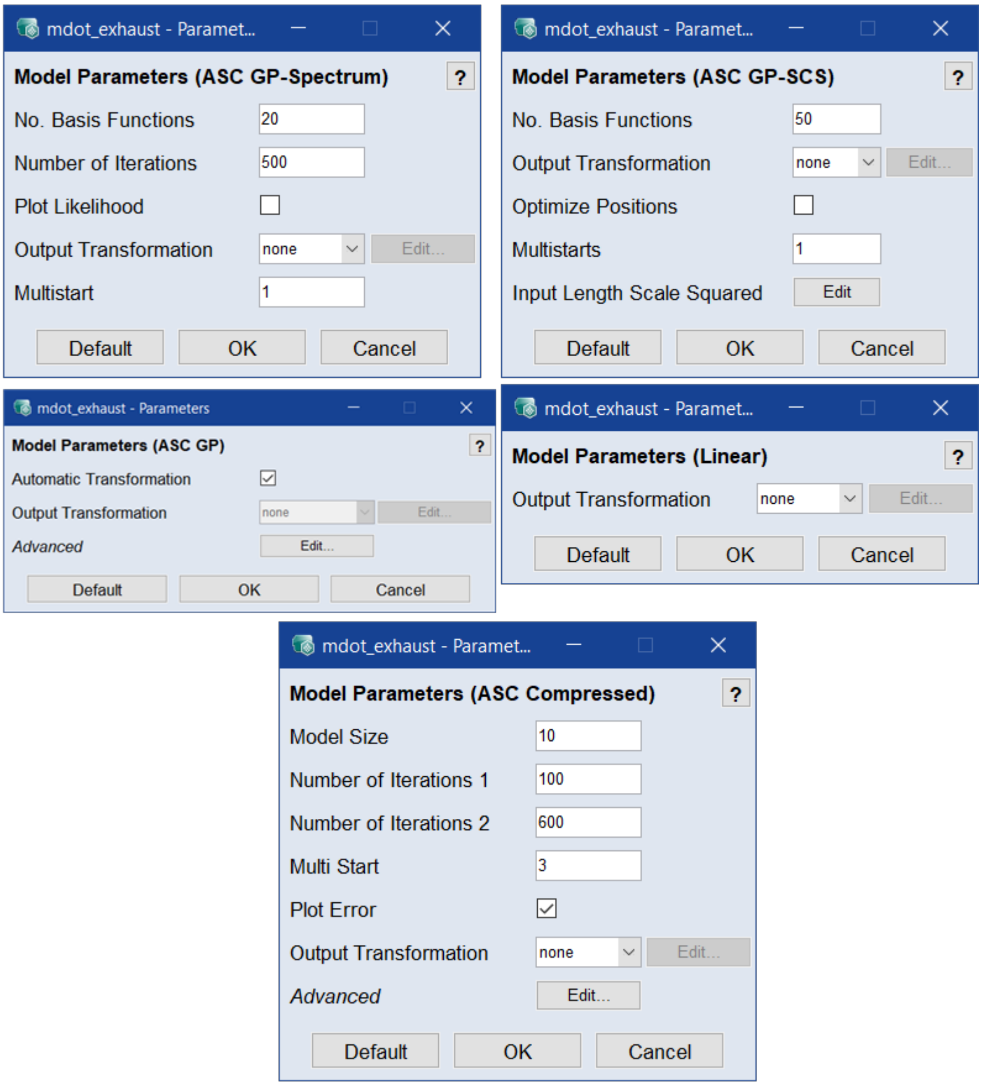
3. Click **OK** to start the training.

5.4.1 Modeling Method and Algorithm

The default modeling method is NARX Structure, the default modeling algorithm is ASCMO Gaussian Process Spectrum (ASC GP-Spectrum), which represents the recent improvement of the standard ASC algorithm to cope with large data sets.

ASC GP-Spectrum requires a number of Basis Functions `s` set by the user. During the model training, the information contained within the entire data set will be transformed to a set of `s` virtual basis points. The bigger the size `s`, the better the modeling result, but this goes along with an increase in modeling time. The recommended range is `50 < s < 200`.

The number of Basis Functions, as well as other model parameters, are set in the **‹output› - Parameters** window. This window opens automatically when you select the model type `ASC GP-Spectrum`; in addition, you can open it with the **Edit** button.

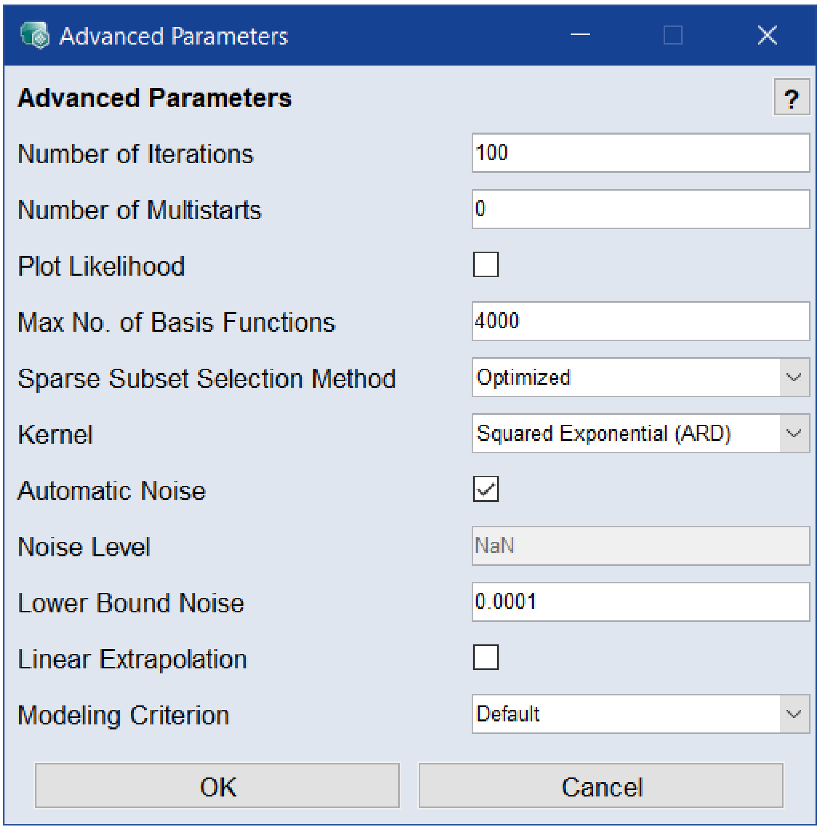


The ASCMO Gaussian Process Sparse Constant Sigma (`ASC GP-SCS`) model type is recommended for large numbers of training data.

The ASCMO Gaussian Process (`ASC GP`) model type is recommended for small numbers of training data.

A `Linear` model type is provided to train a linear model. For the linear model, the **‹output› - Parameters** window offers fewer parameters to edit.

Advanced Parameters for ASC GP Models can be set by using the **Edit** button, **Advanced Settings** must be activated via **File › Options**.



For more information on modeling methods and algorithms, see the online help (‹F1›).

5.4.2 Feedback Structure (NARX)

Definition of the maximum time lag for all inputs and outputs to be considered within the NARX structure. The feedback values will be called features in the following.

The results of the IACF analysis (see 5.3.7 "Phase Plot, ACF and IACF" on page 34), usually serve as a good starting point. In the diesel example, both values are set to 3.

5.4.3 Dimensionality Reduction

Reducing the number of feedback values (features) can increase model quality and accelerate model training.

Several options are available in the **Model properties** window (Fig. 5-6: on page 37).

- **None**: Use all features in the NARX structure in accordance with the set maximum time lags. This is sufficient for the diesel example.
- **Feature Extraction (PCA)**: Reduces the dimensionality of the feedback structure to the given number of features using a principle component analysis (PCA).

  > **Note**
  >
  > In many cases, not all features of the feedback structure are necessary, e.g. due to redundancies. A smaller number of features supports faster model training.

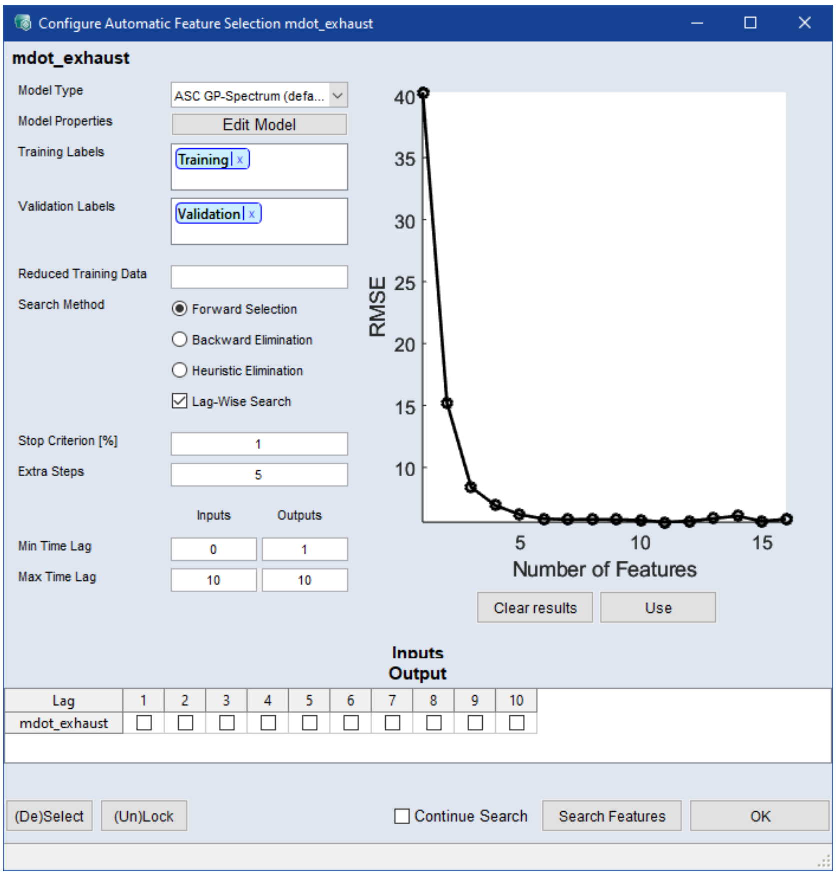
- **Automatic Feature Selection**: As an alternative to manual feature selection, you can find the proper feature setting via an automatic search.
- **Manual**: Manual feature selection allows the user the explicit selection of the feature set by activating the corresponding checkboxes in the **Inputs** and **Output** table (in Fig. 5-6: on page 37).

  This is recommended either when the model training with **Dimensionality Reduction None** yields insufficient model quality or when the system-specific time dependencies are known.

Use Automatic Feature Selection

> **Note**
>
> In the case of more than one system output, this procedure has to be performed individually for each output.

1. Select **Model › NARX Feature Search › Configure**.



2.
3. In the **Model Type** drop-down, select a model type you want to use for Automatic Feature Selection.

   For best results, the selected type should be the same as in the actual model training.
4. If desired, click **Edit Model** to open the **‹output› - Parameters window** (5.4 on page 36) and edit the model parameters.
5. In the **Search Method** area, activate the option for the method you want to use.

– **Forward Selection**

  Starts with an empty feature set and iteratively adds the feature that gives the maximum increase in model quality. Recommended setting for faster convergence.

– **Backward Elimination**

  Uses all features at the beginning and iteratively removes the feature with the least impact on the model. The advantage of backward selection is that it identifies interdependencies between features.

– **Heuristic Elimination**

Uses all features in the beginning and iteratively removes the feature with the least impact on the model based on heuristic input relevance. This is a faster method (compared to the other methods, especially Backward Elimination) to automatically find the NARX structure.

- **Lag-Wise Search**

  When this checkbox is active, the inputs are searched lag-wise. This speeds up the feature search.

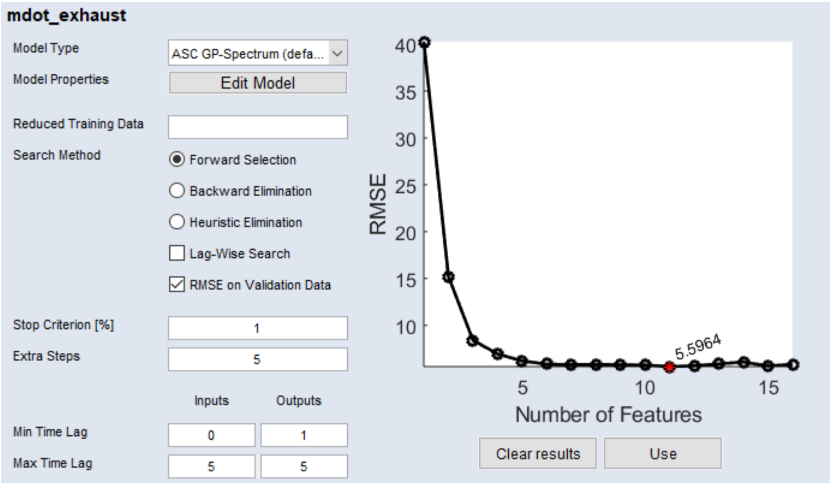
6. In the **Stop Criteria [%]** field, enter a value.

   The search stops if the model quality cannot be increased further (Forward Selection) or becomes worse (Backward Elimination) than the given stopping value.

7. In the **Min Time Lag** and **Max Time Lag** fields, enter the minimum and maximum time lags for inputs and outputs.

8. Click **Search Features** to start automatic feature selection. The search may take some time. Progress is shown in the plot on the top right of the **Automatic Feature Selection <output>** window.

   When the search is complete, the best result is marked.

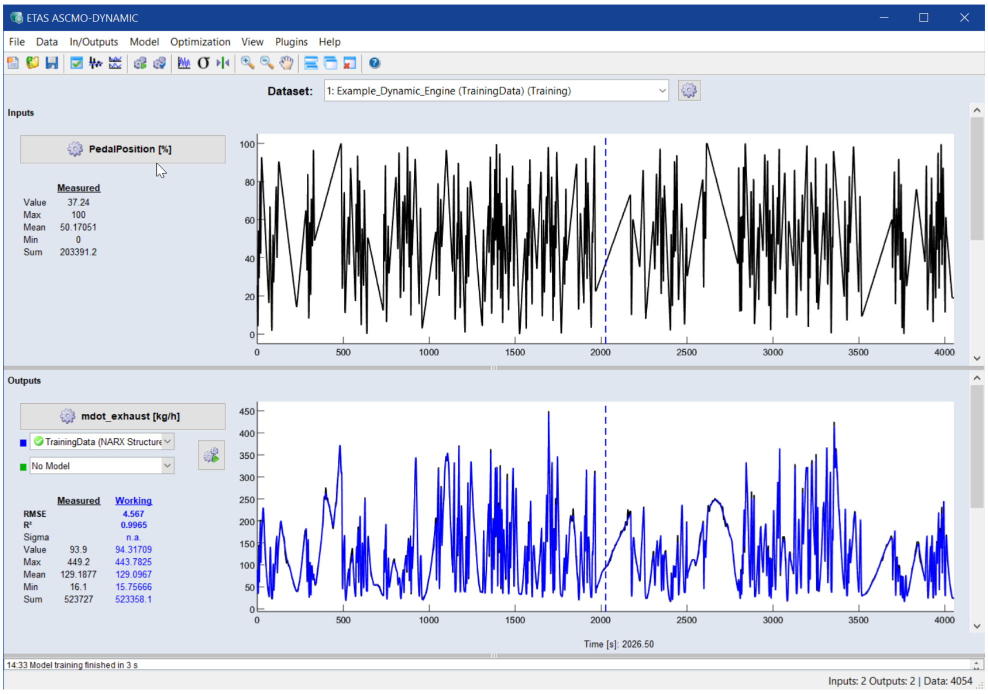


9. Click **Apply** to apply the result.

When you have selected features for all outputs, you can return to the **Model Properties** window and start model training with the **OK** button.

5.5 Model Prediction

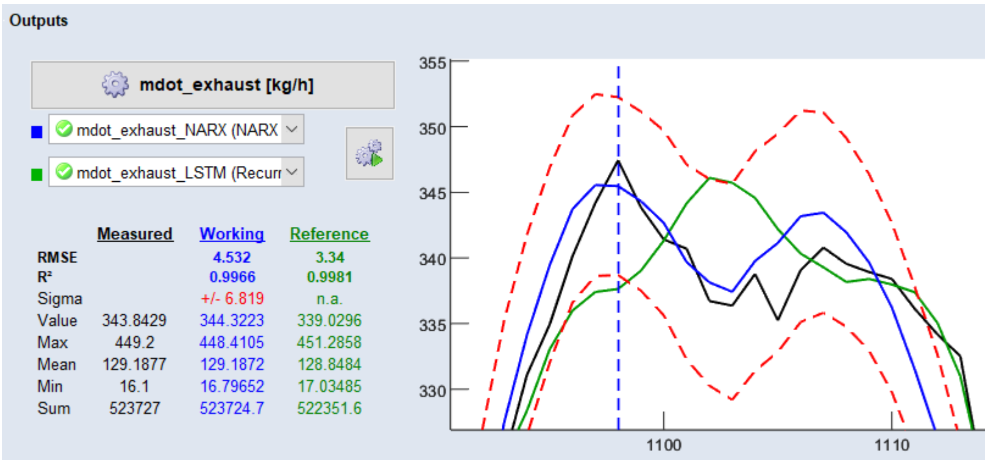
After model training, the model prediction is automatically plotted over the output measurements of the training dataset. The model quality is given with the RMSE and R^2 measure.



5.5.1 Sigma Prediction

The ASC Dyn algorithm also provides a sigma prediction (**View › Show Sigma**), which shows the model's uncertainty.

A weaker model behavior (i.e. a smaller value of R^2) corresponds to increasing sigma values. In the diesel engine example, the very good model quality is connected with a small sigma.



5.5.2 Multi-Step/One-Step Ahead Prediction

The mechanism of the NARX model prediction can be changed between one-step and multi-step ahead prediction (**Model › NARX Model Options › One Step Ahead Prediction / Multi Step Ahead Prediction**).

Note

The important difference is that the multi-step ahead prediction uses past model predictions within the feedback (NARX) structure instead of the measured values (see Fig. 2-2: on page 14). Multi-step ahead prediction is the standard setting and corresponds to a real offline simulation, where no system responses are available during model prediction.

5.6 Model Validation

To check the model quality on an independent data set, a second data set (**Test data**) can be imported.

To import test data

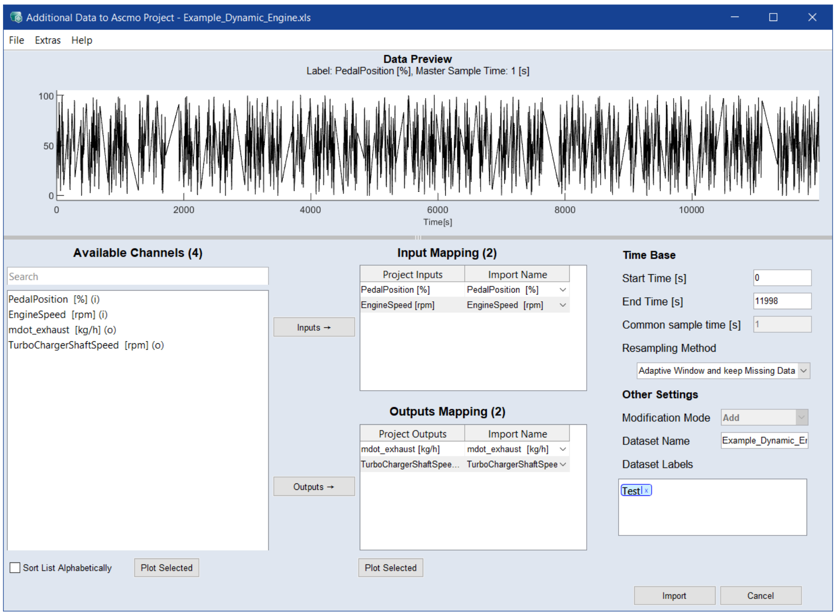
1. Select **File › Import Data › Test**.

   A file selection window opens.
2. Navigate to the folder *<installation>*`\Example\AscmoDynamic` and select the file `Example_Dynamic_Engine.xls`
3. Click **Open**.
4. In the **Select Sheet** window, select the sheet **Test Data**.
5. Click **OK**.

   The **Time Vector / Sample Time** window opens.
6. In the **Sample Time** field, enter the sample time (in seconds) of the test data (cf. "Data for Modeling" on page 26).

   Since the test data table contains no time column, you can ignore the **Time Vector** field in this tutorial.
7. Click **OK**.

   The **Additional Data to ASCMO project** window opens.



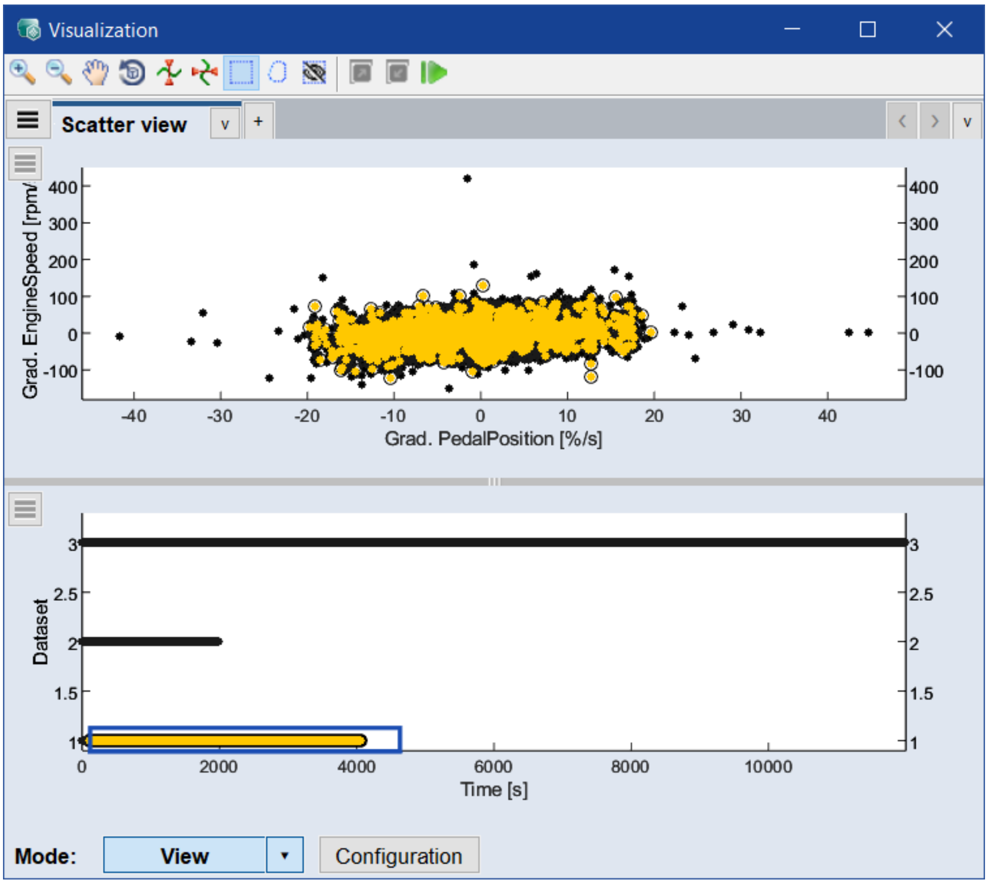
Because training and test data are part of the same measurement cycle, mapping is straightforward.

8. Click **OK** to import the test data.

   After loading the data set, the model prediction is automatically shown in the ASCMO-DYNAMIC window.

5.6.1 Model Validation with Scatter Plot

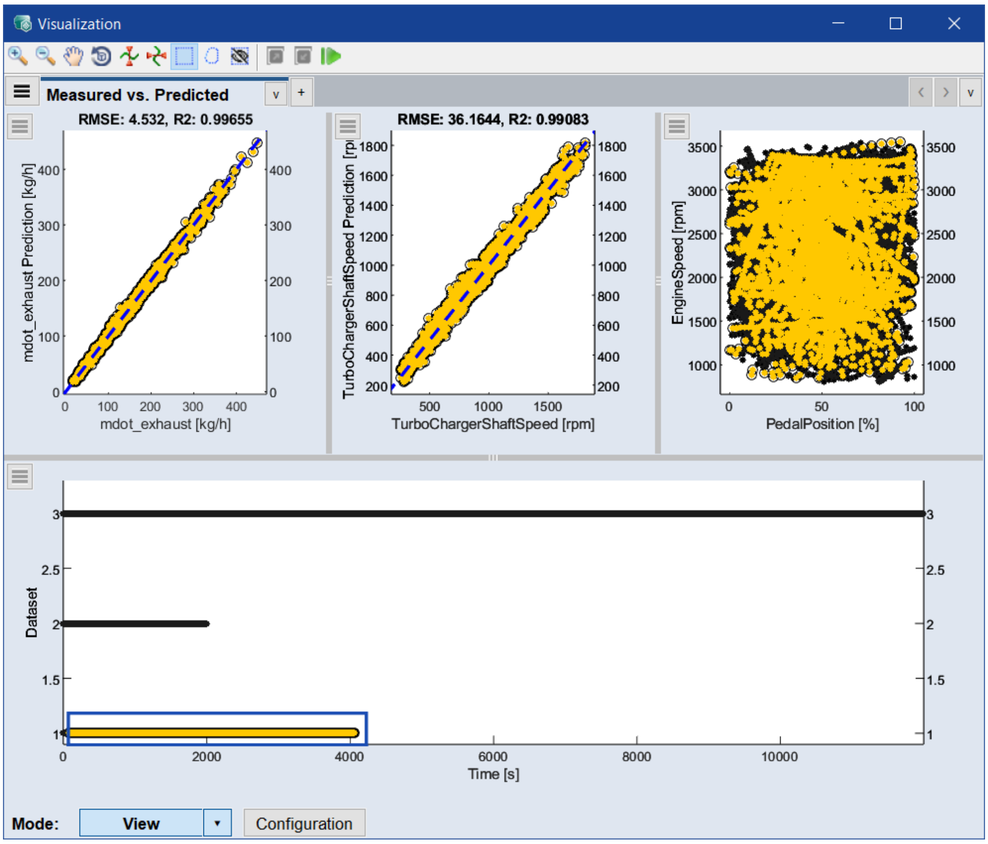
You can use the functions **Data › Scatter Plot › All Data** and **View › Select Axes** to check the distribution of the test data and its gradients.



To avoid extrapolation, the test data (black in the above screenshot) is expected to be within the range of the training data (yellow in the above screenshot), which is the case in the diesel engine example.

5.6.2 Measured vs. Predicted

The menu options **Model › Measured * vs. Predicted** and **Model › All Measured Data vs. Predicted** (* = **Training Data** or **Validation Data** or **Test Data**) plot the measured values of the training or test data or both against the model prediction. The expected result is a diagonal line, which would indicate a perfect match of the prediction and the measurements.

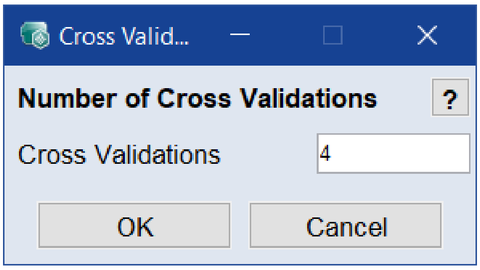


5.6.3 Cross Validation

If no test data set is available, an n-fold cross-validation can be carried out on the training data set. To do so, proceed as follows.

Cross-validation

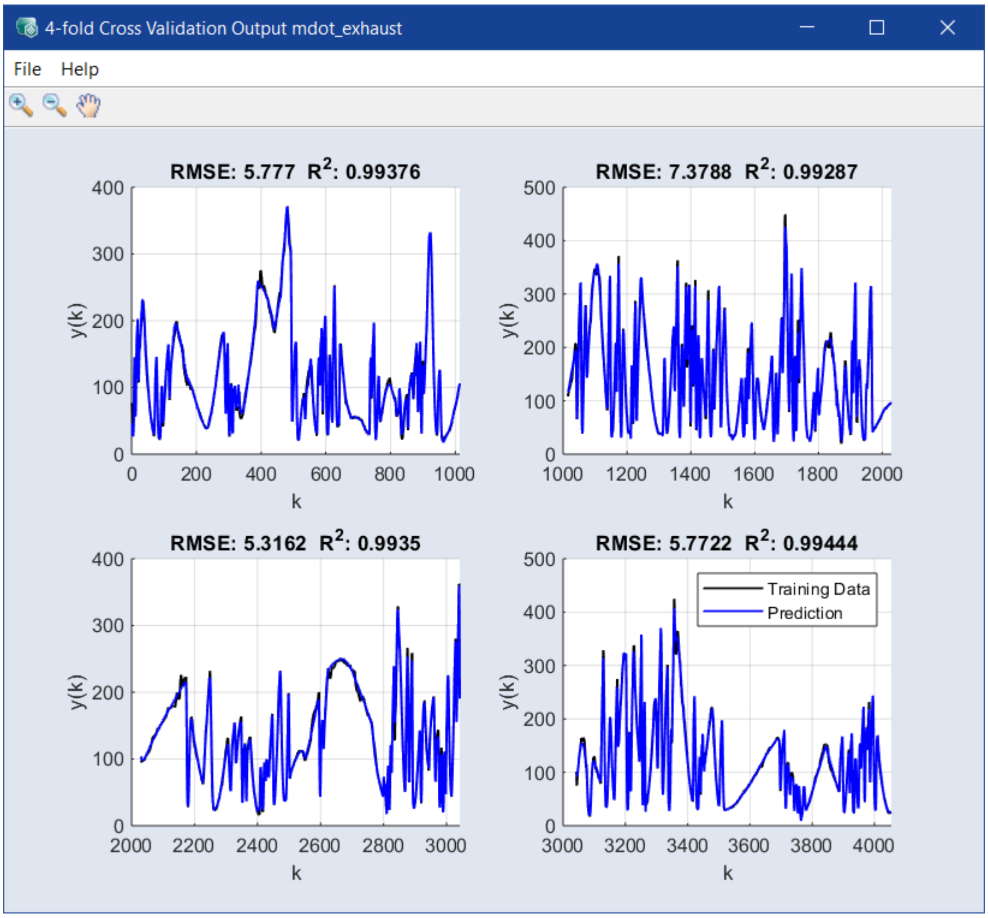
1. Select **Model › Cross Validation on Training Set**.



2. In the **Cross Validation** window, enter the number of data sets you want to use for cross-validation.
3. Click **OK**.

   The data set is split into n equal parts, according to the specified number of cross validations.

For each output, n plots are displayed. Each plot is the result of an individual model training on n-1 data sets and the prediction on the data set not used for modeling.



5.6.4 Cross-Correlation of the Residuals (CCR) Validation

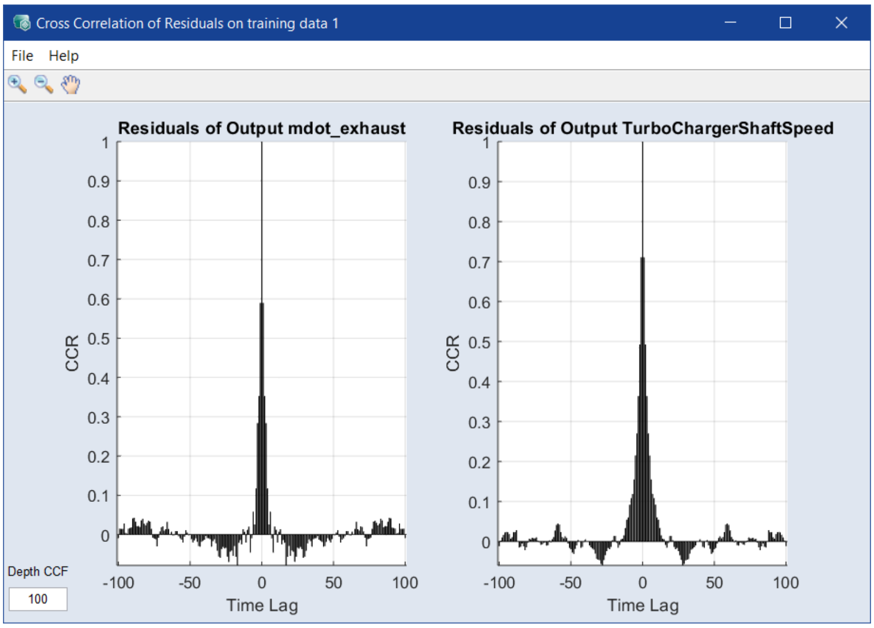
The CCR analysis checks the **c**ross-**c**orrelation of the **r**esiduals.

> **Note**
>
> CCR Validation is only available if you have activated the **Advanced Settings**.

– **Model** › **CCR Validation** › **Residuals * Data**

***** = **Training** or **Test**

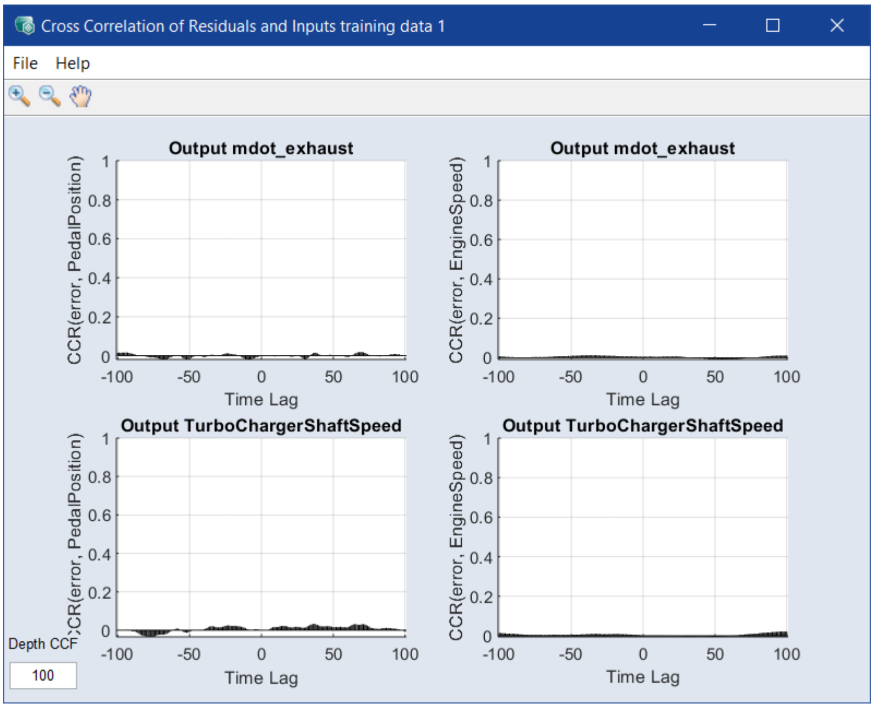
Shows the auto-correlation of the prediction's residuals regarding the training or test data. The desired behavior is an exponentially decreasing correlation with a correlation of 1.0 at time lag 0.



- **Model › CCR Validation › Residuals and Inputs * Data**

  ***** = **Training** or **Test**

  The expected cross correlation between the residuals and the inputs is zero, without any significant peaks.



5.7 Model Export

Note

The model export feature is an add-on to ETAS ASCMO for which a special license ("ASCMO_EXPORT") is required.

In this section you will learn how to export the ASCMO-DYNAMIC models to:

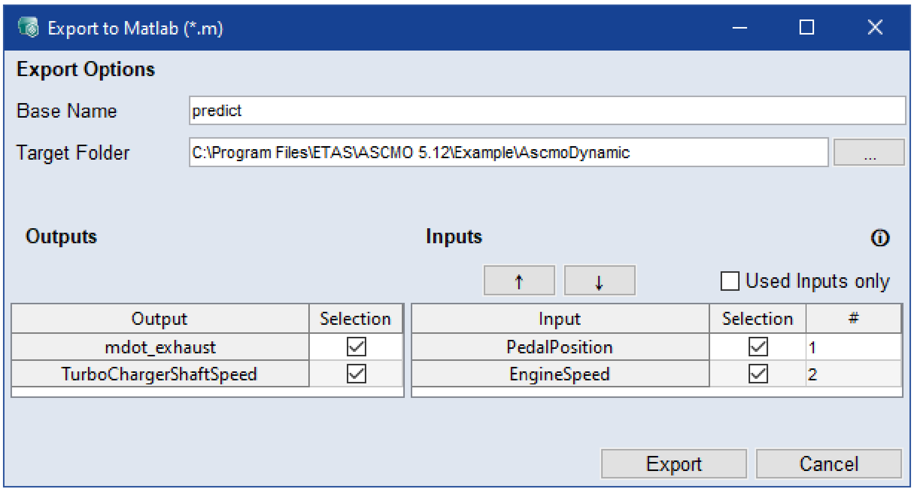
5.7.1 Export to MATLAB®

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 **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.

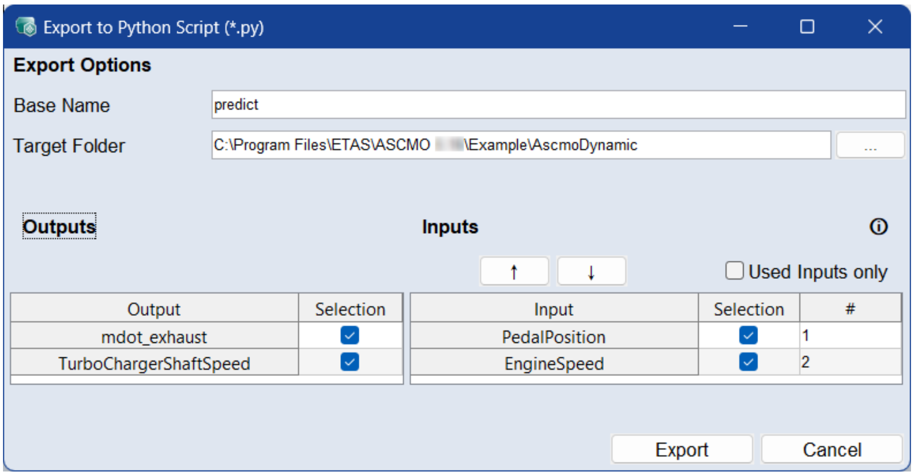
5.7.2 Export to Python Script

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 **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.

5.7.3 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-DYNAMIC 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.

5.7.4 Export to Simulink® Script

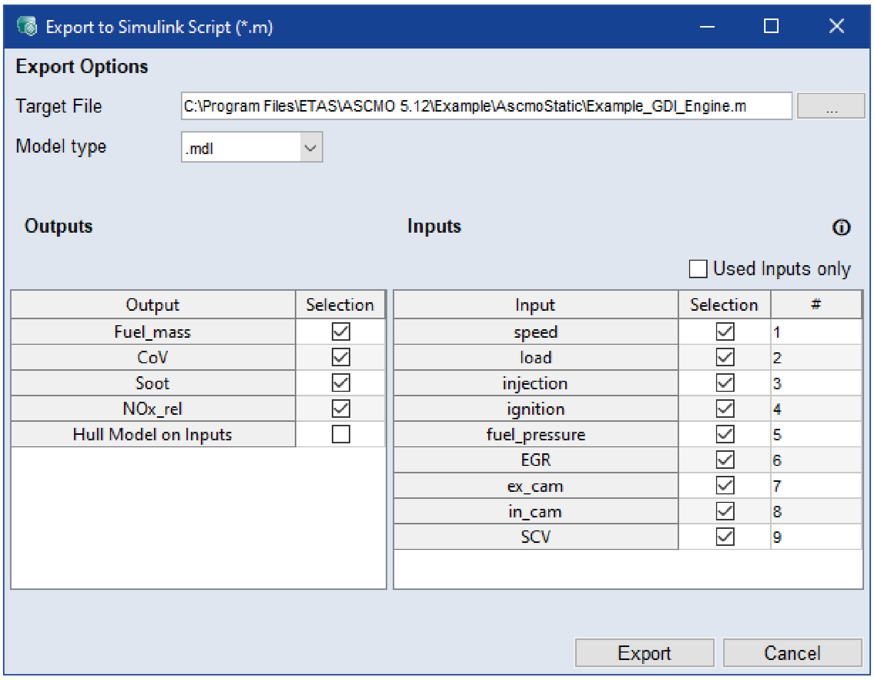
A Simulink installation is not required to perform the export.

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.

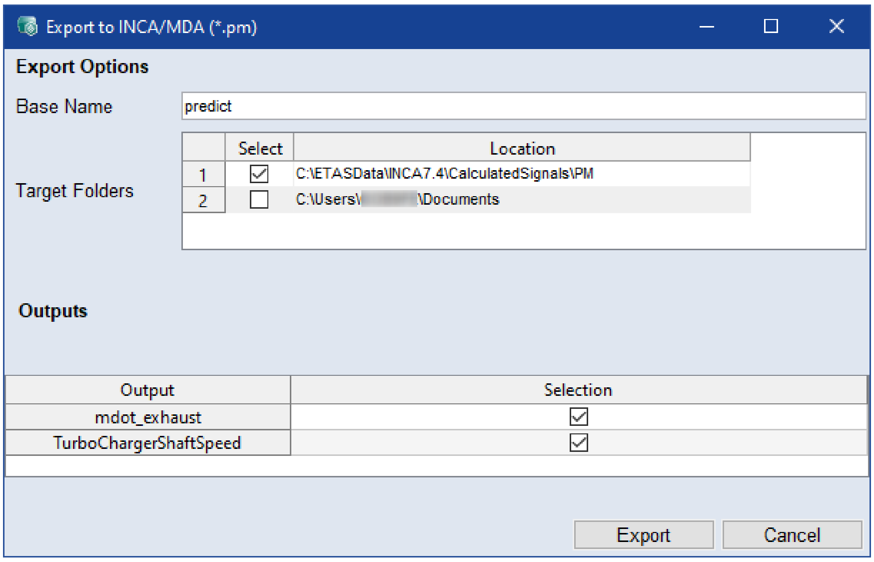
5.7.5 Export to INCA/MDA

The export to INCA/MDA exports the models as "Calculated Signals" for INCA/MDA 6.x. For MDA 8.x export to FMU. 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.

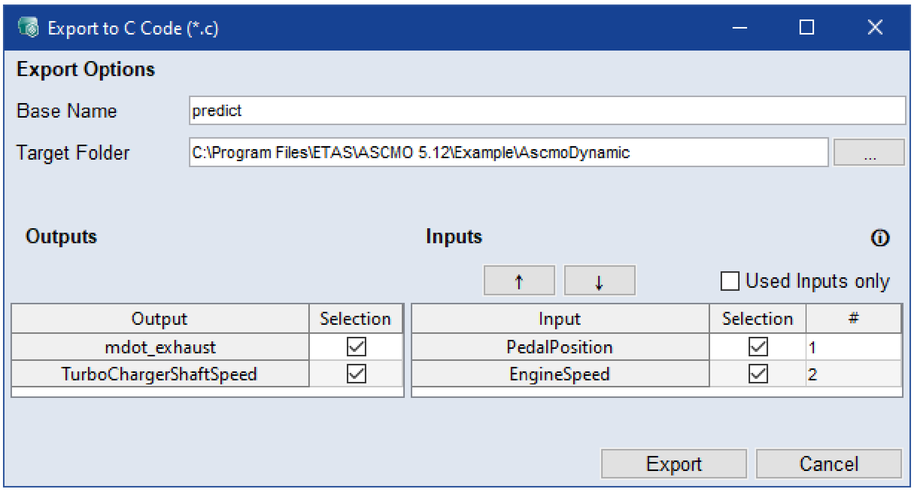
5.7.6 Export to C Code

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 **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.

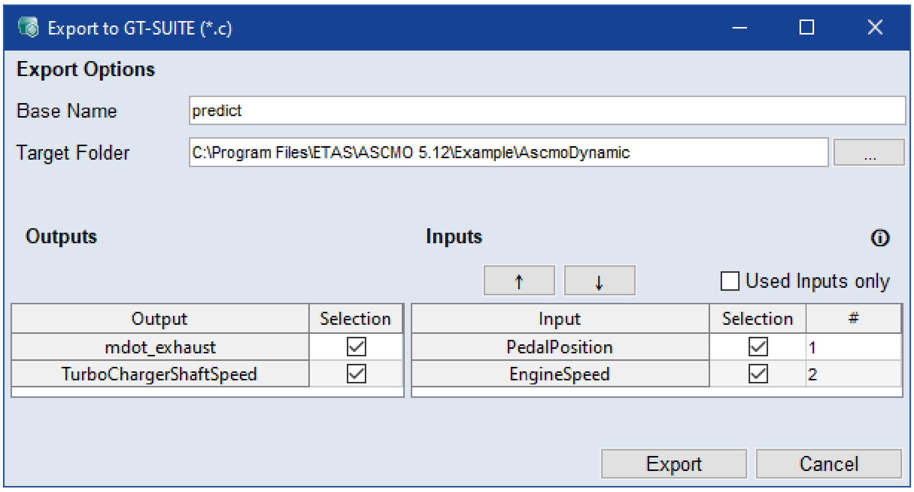
5.7.7 Export to GT-Suite

The model export to GT-SUITE is an extended version of the C code export that allows you to load and use ASCMO-DYNAMIC 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 **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.

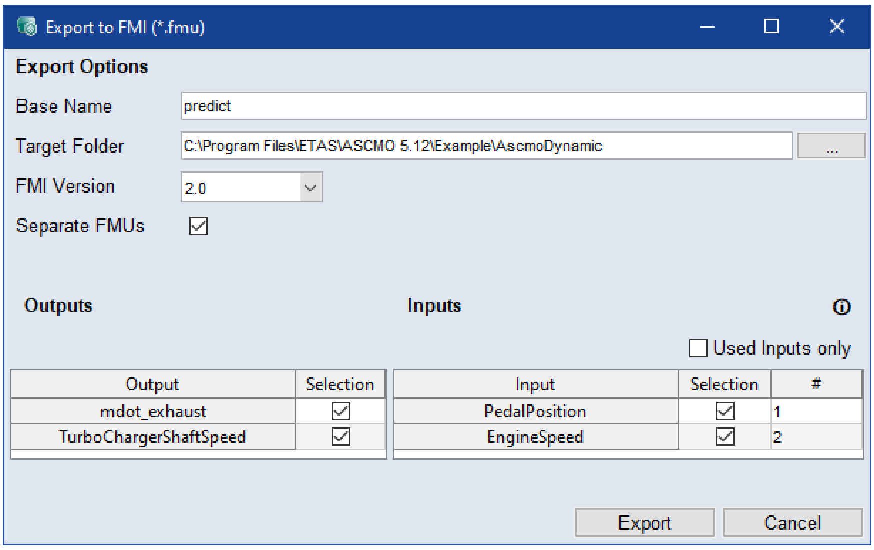
5.7.8 Export to FMI

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.

Export a model to FMI

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 **FMI Version** to which the model is exported.
4. Deselect the **Separate FMUs** checkbox to export the selected outputs as a single FMU file.
5. Select the **Inputs** and **Outputs** to export in the table.

   To export only used inputs, select **Used Inputs Only**.
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.

5.7.9 Export to Keras

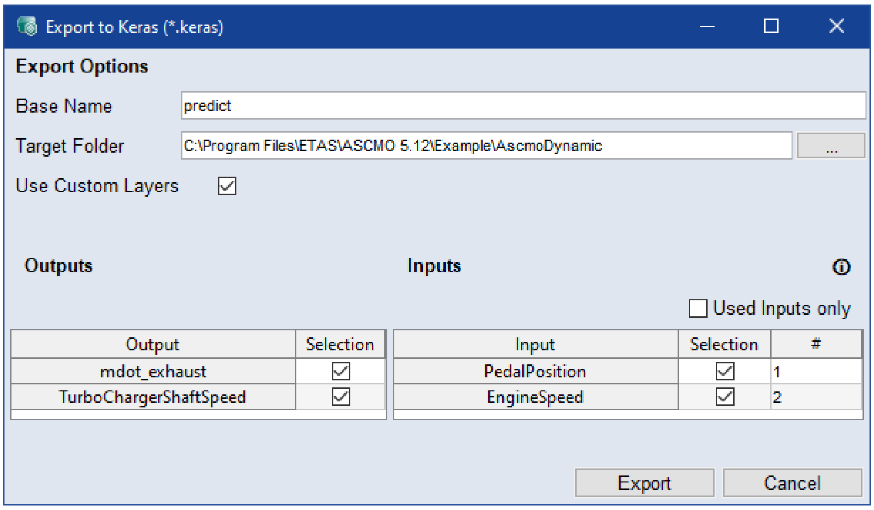
> Only Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) models can be exported to Keras.

When you export a model to Keras, a `*.keras` file is created for each output, as well as the `transformation.py` file.

Export a model to Keras

1. **File** › **Export Model** › **Keras**

   Export to Keras window

2. Set the **Base Name** (defines the file name `<base name>_<output>`) and the **Target Folder**.
3. Deactivate the **Use Custom Layers** checkbox if you want a model without custom layers.
4. Select the **Inputs** and **Outputs** to export in the table.

   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.

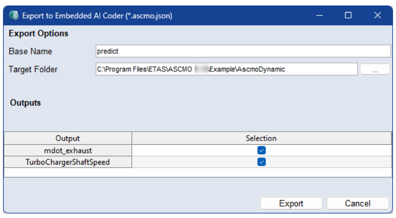
5.7.10 Export to Embedded AI Coder

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.

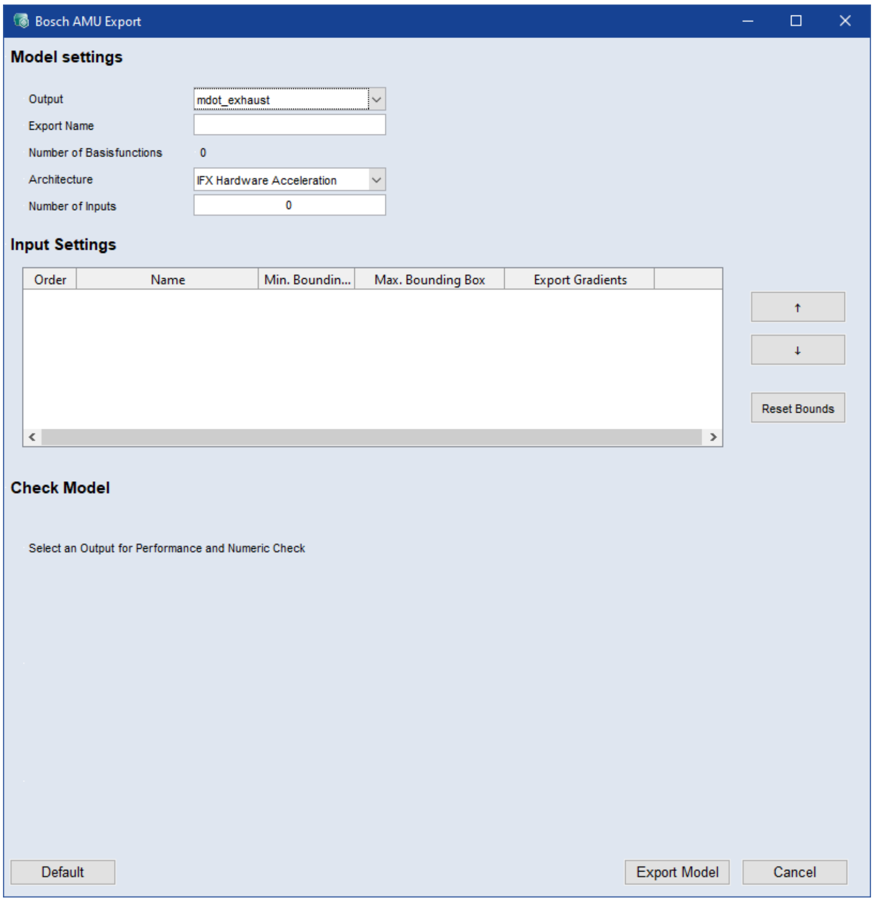
5.7.11 Export to Bosch AMU

> 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

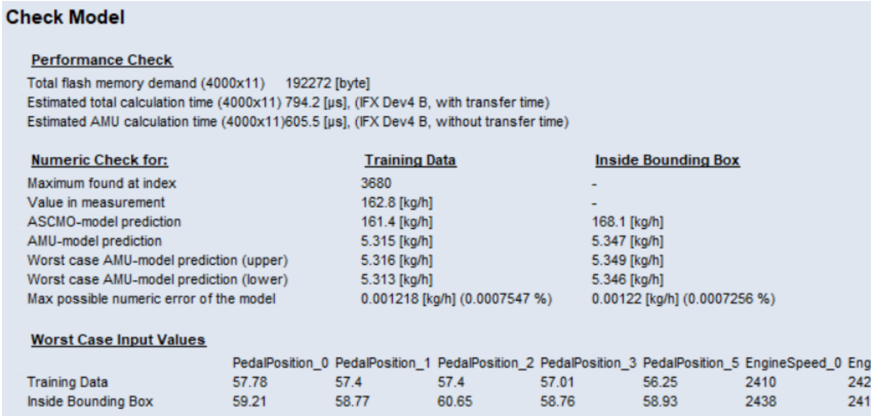
1. **File › Export Model › Bosch AMU**.

   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.

5.7.12 Export to Bosch Flatbuffers

Only Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) models can be exported to Bosch Flatbuffers.

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 Tutorial: Working With ASCMO-DYNAMIC ExpeDes

ASCMO-DYNAMIC ExpeDes is a tool for creating space-filling statistical experiment plans. It is ideally suited for planning measurements with a space-filling distribution of target values and gradients of the measurement channels, as required for model training in ASCMO-DYNAMIC.

This chapter describes the individual working steps involved in creating a dynamic experiment plan with ASCMO-DYNAMIC ExpeDes.

- **6.2 "Step 1: General Settings" on page 66**

  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.
- **6.3 "Visualizing the Dynamic Experimental Plan " on page 68**

  To be able to evaluate the plan data, you can display it either graphically or as a table at any time.
- **6.4 "Step 2: Constraints " on page 71**

  In this step, constraints of the measurement range can be made for a variable as a function of one or two other variables.
- **6.5 "Step 3: Input Compression " on page 81**

  In this step, compressions of measuring points in certain areas of the measuring space can be specified for inputs by area.
- **6.6 "Step 4: Steady State Points " on page 83**

  This step allows you to define operating points for the system, which have a frequency of stay in the experiment plan. In addition, you can define steady state phases and the duration of the steady state phases.
- **6.7 "Step 5: Block Configuration" on page 89**

  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 and is space filling.
- **6.8 "Step 6: Snippet Configuration" on page 91**

  This step is skipped in this tutorial.
- **6.9 "Step 7: Calculated Inputs" on page 92**

  This step is skipped in this tutorial.
- **6.10 "Step 8: Export" on page 93**

  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, density plots, scope views, or as a table.

6.1 Starting ASCMO-DYNAMIC ExpeDes

ASCMO-DYNAMIC ExpeDes is started from the Windows Start menu or from the **ASCMO-DESK** window (shown in Fig. 5-2: on page 26).

Start ASCMO-DYNAMIC ExpeDes

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

   The ASCMO-DYNAMIC ExpeDes start window opens.

2. You can start ASCMO-DYNAMIC ExpeDes with an empty project (**New** > **Open Project**), you can open a demo project (**Demo Files**), or you can open an existing project (**Open** > **Open Project**/Select a recent project).

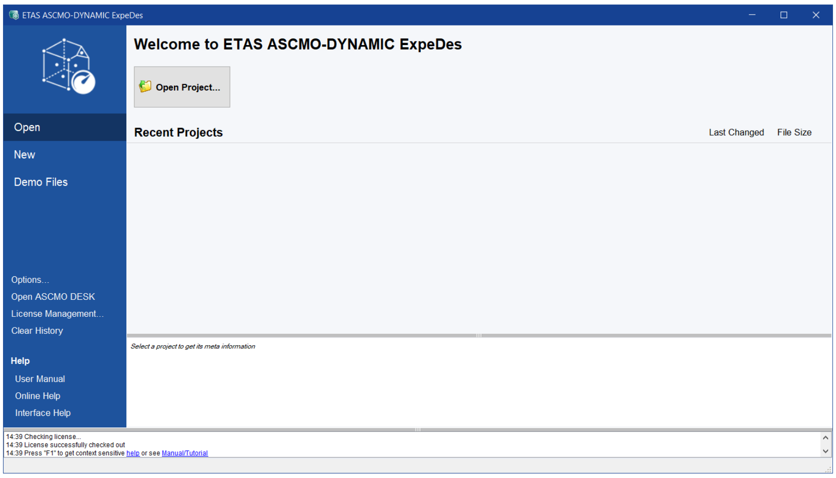


Fig. 6-1: ASCMO-DYNAMIC ExpeDes start window

3. To start ASCMO-DYNAMIC ExpeDes with an empty project:

   i. Click **New** in the menu panel on the left.

   ii. Click 

.

   The ASCMO-DYNAMIC ExpeDes main window opens on the first page, with some default settings.

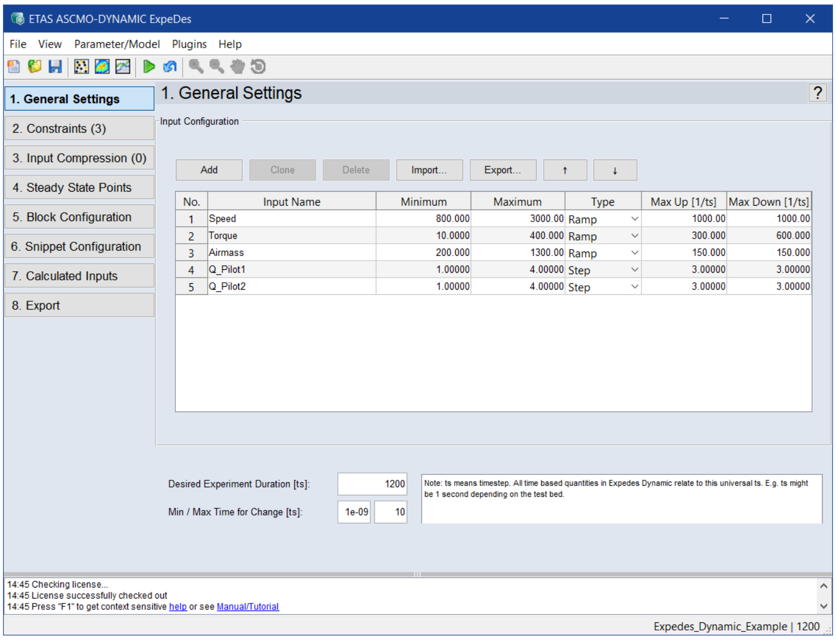
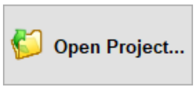


Fig. 6-2: ASCMO-DYNAMIC ExpeDes main window

4. To open an existing project in ASCMO-DYNAMIC ExpeDes, do the following:

   i. In the ASCMO-DYNAMIC 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-DYNAMIC ExpeDes main window.

6.2 Step 1: General Settings

NOTICE

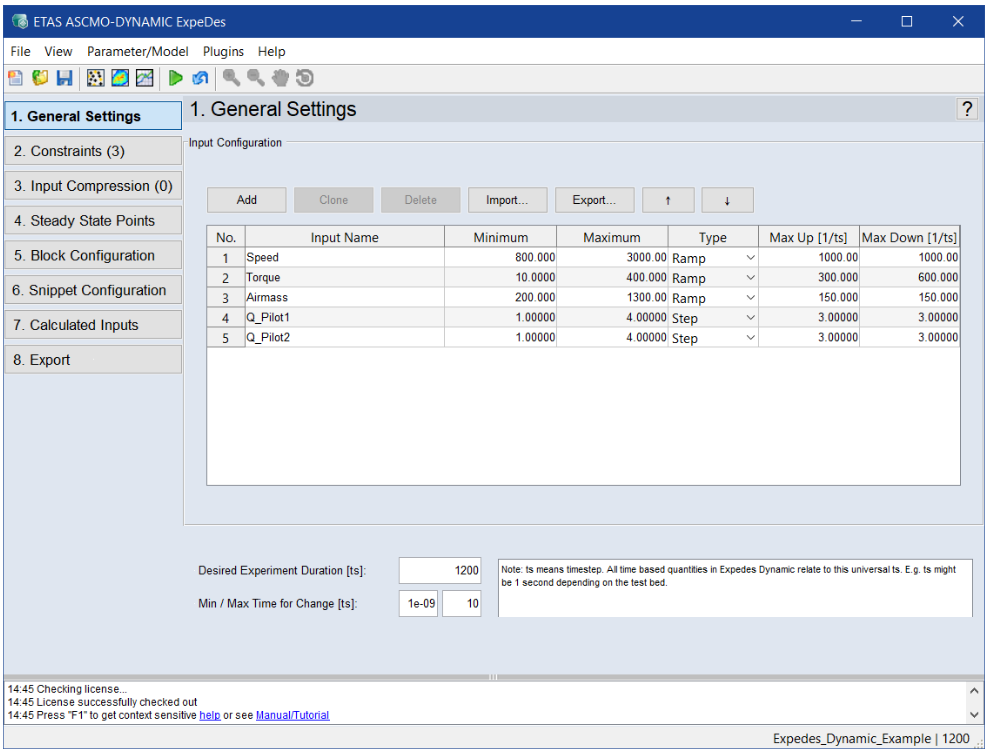
Damage due to wrong test plan

Wrong engine settings in ASCMO-DYNAMIC 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-DYNAMIC ExpeDes see 6.10 "Step 8: Export" on page 93 and 6.2 "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.



6.2.1 Input Configuration

When you create a new ASCMO-DYNAMIC 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 using STRG/CTRL or SHIFT and click **Clone**.

   The duplicates, named *<input_name>*_2, are added below the respective originals.

Configuring an input

The inputs defined in the last step must be configured to generate a dynamic experimental plan. To configure an input, proceed as follows.

1. Click in a cell.

   The cell becomes an input field or a drop-down list.

2. Enter or select the desired value, then press <ENTER>.

   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.
Type	The type of the input. Possible selections are the types `Ramp` and `Step`.
Max Up **Max Down**	Type `Ramp`: maximum possible positive or negative slope of the gradient. Type `Step`: maximum possible positive or negative jump height.

> **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.

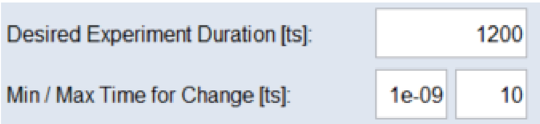
Removing an input

1. To remove an input, select the corresponding row and click **Delete**.

> **Note**
>
> By pressing ‹CTRL›, you can select and remove multiple inputs simultaneously.

6.2.2 Measurement Size Configuration

Below the **Input Configuration** field, you can set the probable duration of the experiment (**Desired Experiment Duration**) and the absolute time for the processing of a line (**Min / Max Time for Change**).



> **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 a dynamic experiment plan, see 6.10 "Step 8: Export" on page 93.

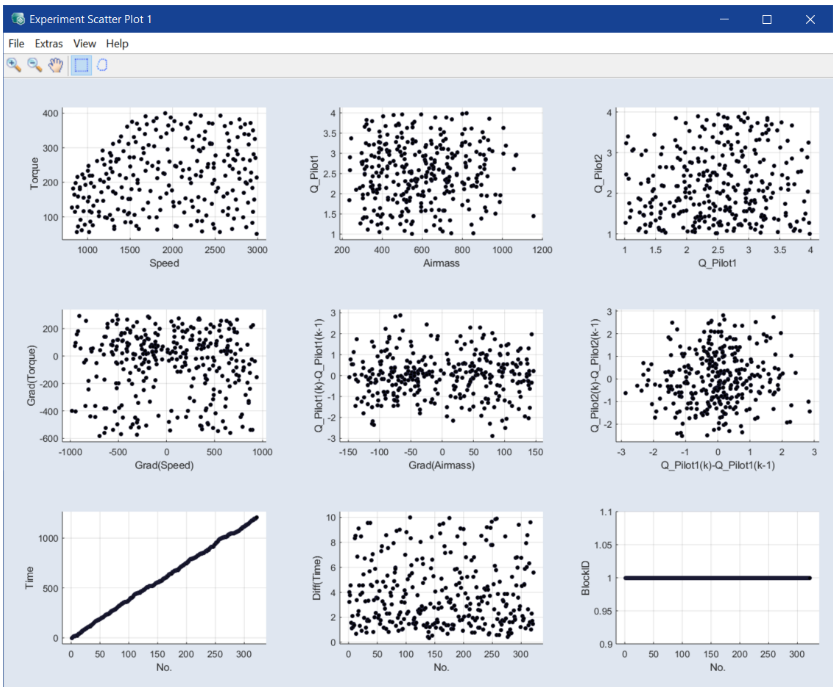
6.3 Visualizing the Dynamic Experimental Plan

To be able to evaluate the plan data, you can display it either graphically or as a table at any time.

For the execution of the tutorial, section 6.3 "Visualizing the Dynamic Experimental Plan " above, is not absolutely necessary. You can directly continue with 6.4 "Step 2: Constraints " on page 71.

Visualizing the experiment plan in a scatter plot

1. In the ASCMO-DYNAMIC ExpeDes main window, select **View › Scatter Plot** to show 2-dimensional representations of the experiment plan.

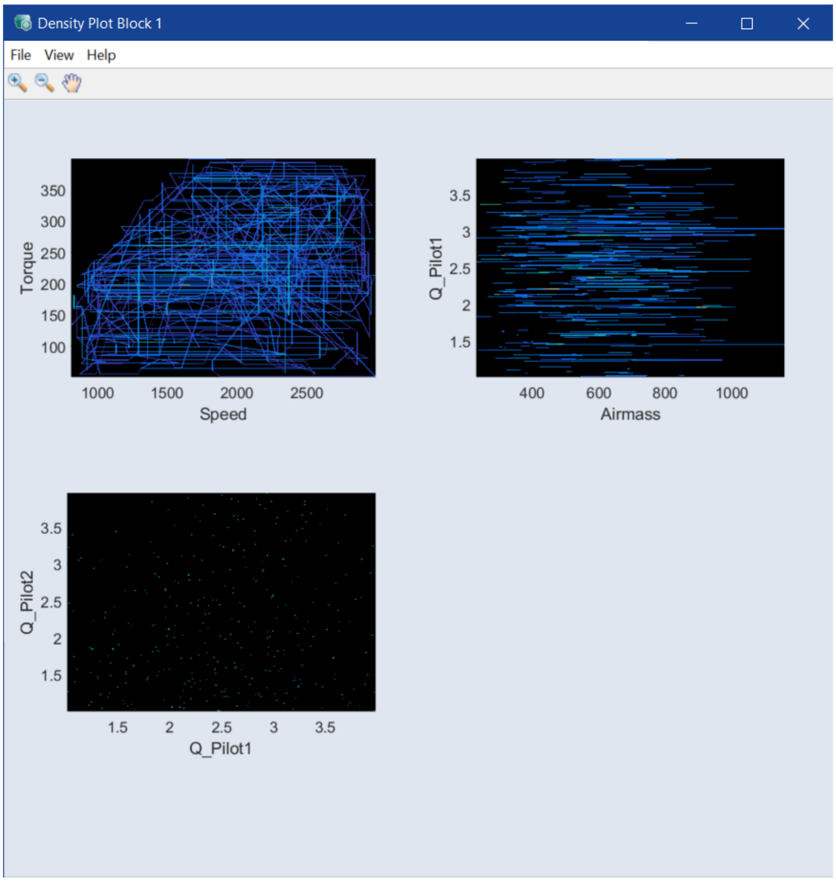


2. In the **Experiment Scatter Plot** window, select the axes you want to show with **View › Select Axes**.

Visualizing the experiment plan in a Density Plot

A density plot visualizes the dynamic of the experiment plan by showing the trajectories of the plan. With the lightness graduation of the trajectories, you can see the frequency of stay on the different points (the brighter the point is plotted, the higher is the frequency of stay in the experiment plan). The visualization is independent of the time.

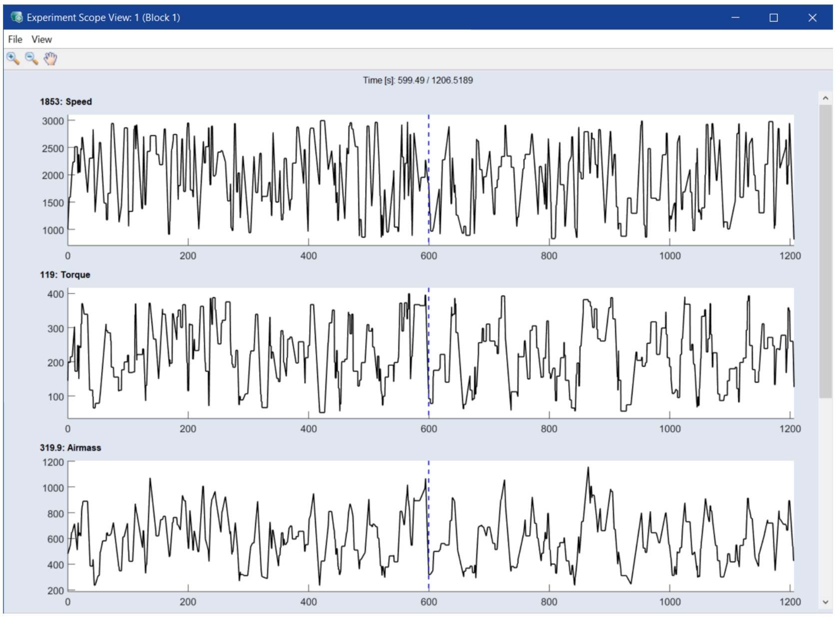
1. In the ASCMO-DYNAMIC ExpeDes main window, select **View › Density Plot** to show density plots of the experiment plan.



2. In the **Experiment Density Plot** window, select the axes you want to show with **View › Select Axes**.

Visualizing the experiment plan in a time-based plot (Scope View)

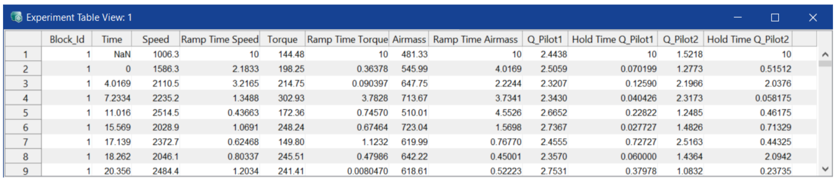
1. In the ASCMO-DYNAMIC ExpeDes main window, select **View › Scope View** to show the experiment plan in a time-based plot.



> **Note**
>
> If you click in the time-based plot, the time-dependent value will be displayed. If you hold down the left mouse button, you can retrieve the time-dependent.values dynamically along the time axis.

Visualizing the experiment plan in a table view

1. To show the data of the measure plan in a table, select **View › Table View**.


Experiment Table View: 1

	Block_Id	Time	Speed	Ramp Time Speed	Torque	Ramp Time Torque	Airmass	Ramp Time Airmass	Q_Pilot1	Hold Time Q_Pilot1	Q_Pilot2	Hold Time Q_Pilot2
1	1	NaN	1006.3	10	144.48	10	481.33	10	2.4438	10	1.5218	10
2	1	0	1586.3	2.1833	198.25	0.36378	545.99	4.0169	2.5059	0.070199	1.2773	0.51512
3	1	4.0169	2110.5	3.2165	214.75	0.090397	647.75	2.2244	2.3207	0.12590	2.1966	2.0376
4	1	7.2334	2235.2	1.3488	302.93	3.7828	713.67	3.7341	2.3430	0.040426	2.3173	0.058175
5	1	11.016	2514.5	0.43663	172.36	0.74570	510.01	4.5526	2.6652	0.22822	1.2485	0.46175
6	1	15.569	2028.9	1.0691	248.24	0.67464	723.04	1.5698	2.7367	0.027727	1.4826	0.71329
7	1	17.139	2372.7	0.62468	149.80	1.1232	619.99	0.76770	2.4555	0.72727	2.5163	0.44325
8	1	18.262	2046.1	0.80337	245.51	0.47986	642.22	0.45001	2.3570	0.060000	1.4364	2.0942
9	1	20.356	2484.4	1.2034	241.41	0.0080470	618.61	0.52223	2.7531	0.37978	1.0832	0.23735

6.4 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.

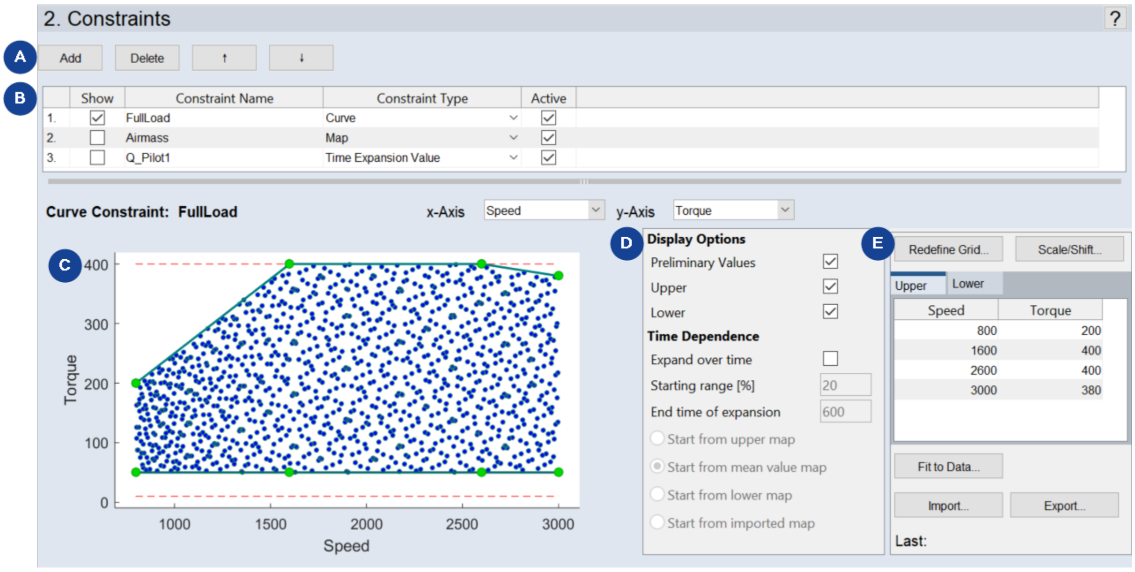


Fig. 6-3: ASCMO-DYNAMIC ExpeDes Step 2: Constraints (Type **Curve**)

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** Fig. 6-3: 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`, `Time Expansion Value` and `Time Expansion Gradient`.

   See section 6.4.1 "Constraint Types "Map" and "Curve" " on the next page for further information on curve and map constraints.

   `Time Expansion Value` and `Time Expansion Gradient` are described in the online help (F1).

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 section 6.4.1 "Constraint Types "Map" and "Curve" " on the next page.

6.4.1 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 Preliminary Values 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. 6-3: on the previous page) or 3D plot (Map).

The display of the preliminary values is controlled via the **Preliminary Values** checkbox (region **D** in Fig. 6-3: on the previous page). Only a part is shown to ensure a smooth operation. In addition, the thicker points of the grid with which the constraint of areas is controlled are also displayed.

Constraining the measurement range

1. In the **Show** column, activate the checkbox for the constraint you want to edit.

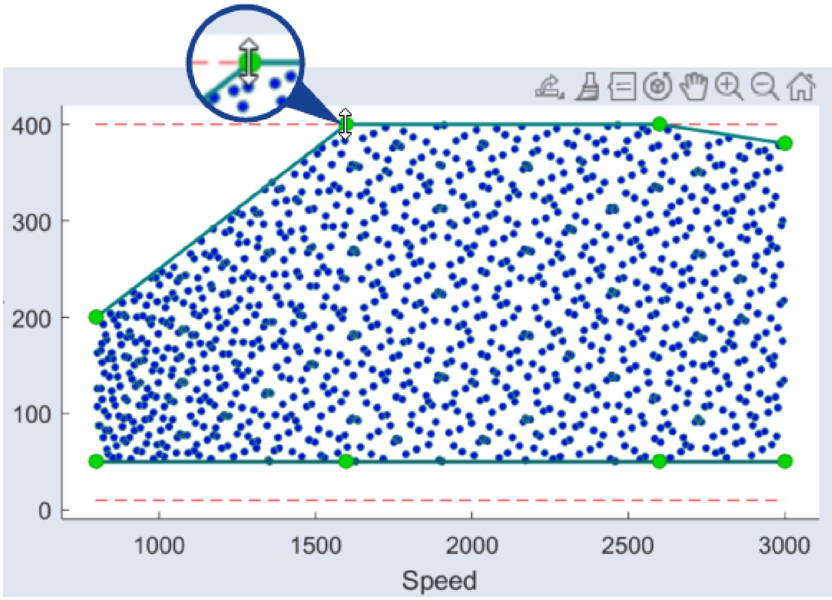
   The constraint is displayed in the lower part of the window.

2. In the constraint plot (region **C** in Fig. 6-3: on the previous page), click a point of the constraining line/area and hold the mouse button pressed.

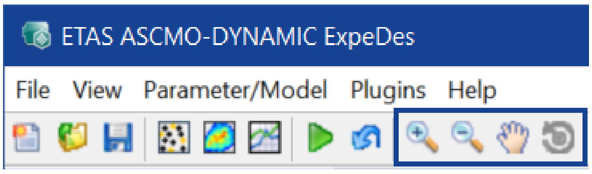
   The mouse pointer changes to a double arrow.

3. Drag the point to the desired position.

The figure shows a limitation of the torque at lower speeds.



The display of the plot can 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. 6-3: on page 72) and can be processed in these (both in terms of quantity and value); see section "Table for Displaying and Editing the Grid Nodes" below.

Further functions for specifying and displaying constraints are.

- **Display Options**
  - **Preliminary Values**: Display of preliminary values calculated by all defined constraints
  - **Upper/Lower**: Display of the upper/lower limits defined by the constraint
- **Time Dependence**: The options and fields in this area can be used to constrain the value range or gradient range of a variable as a function of time. See the online help for details.

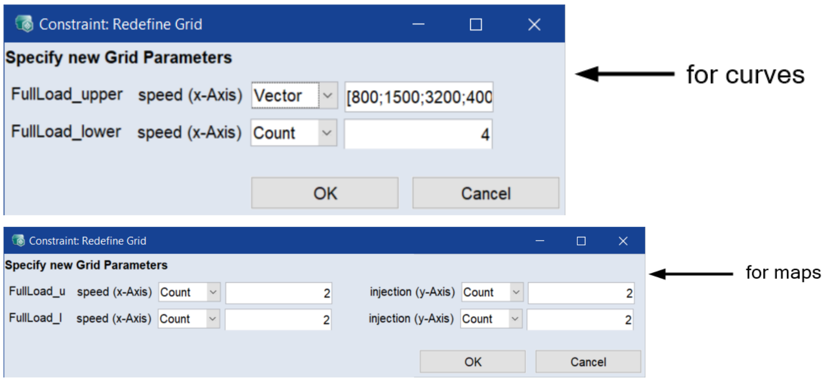
Table for Displaying and Editing the Grid Nodes

The grid nodes can be edited in the tables to the right of the plot (region **E** in Fig. 6-3: on page 72). One tab each is available for the upper and lower constraint.

Changing the number of grid nodes

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 **Constraints: Redefine Grid** window closes.

6.4.2 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

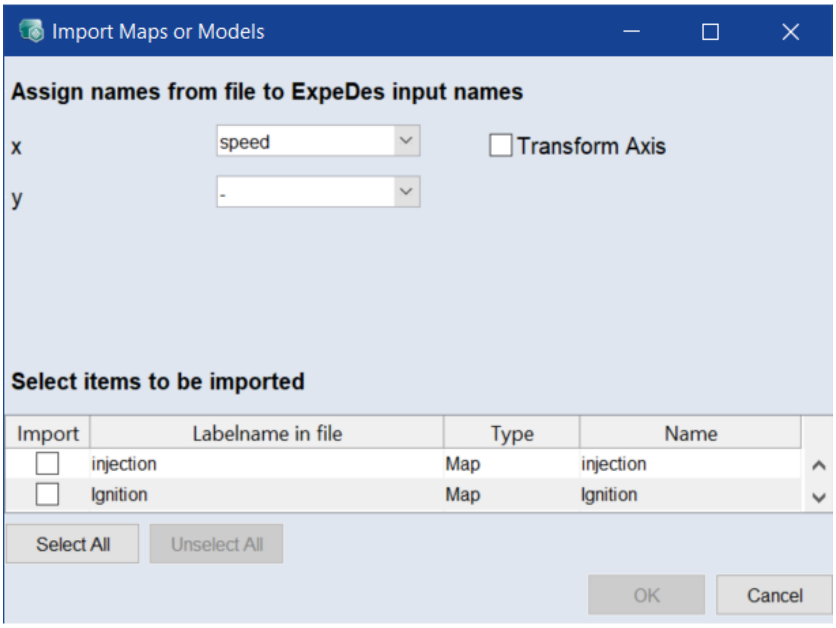
If the data for a constraint curve or map exist as e.g. `*.dcm`, `*.cdfx`, `*.csv`, `*.xls`, `*.xlsx`, `*.xlsm` or `*.ascmo` file, they can be imported.

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 "Assigning an existing map/curve to a constraint" on the next page.

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-DYNAMIC ExpeDes project. Available export formats are for example `*.dcm`, `*.cdfx` and `*.csv`.

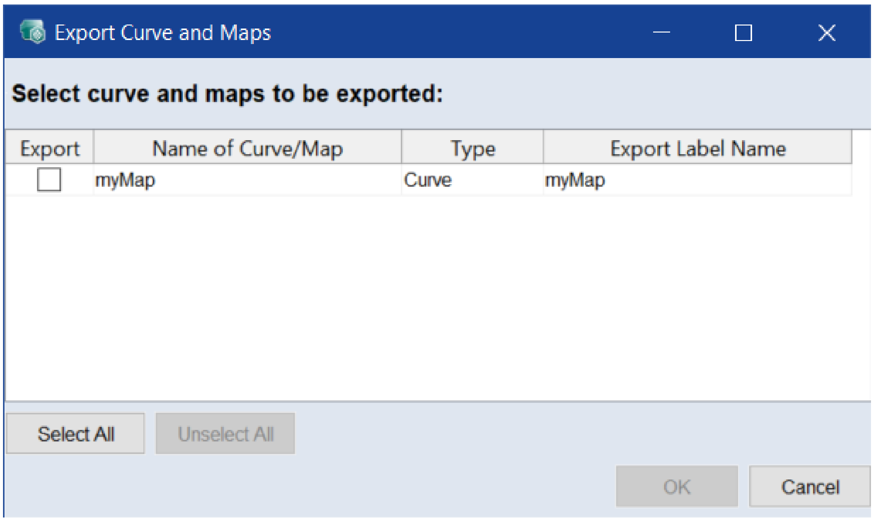
1. Export the curve/map of the currently displayed constraint
   i. In the table area (region **E** in Fig. 6-3: on page 72), click **Export**.

   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-DYNAMIC ExpeDes project
   i. 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**.

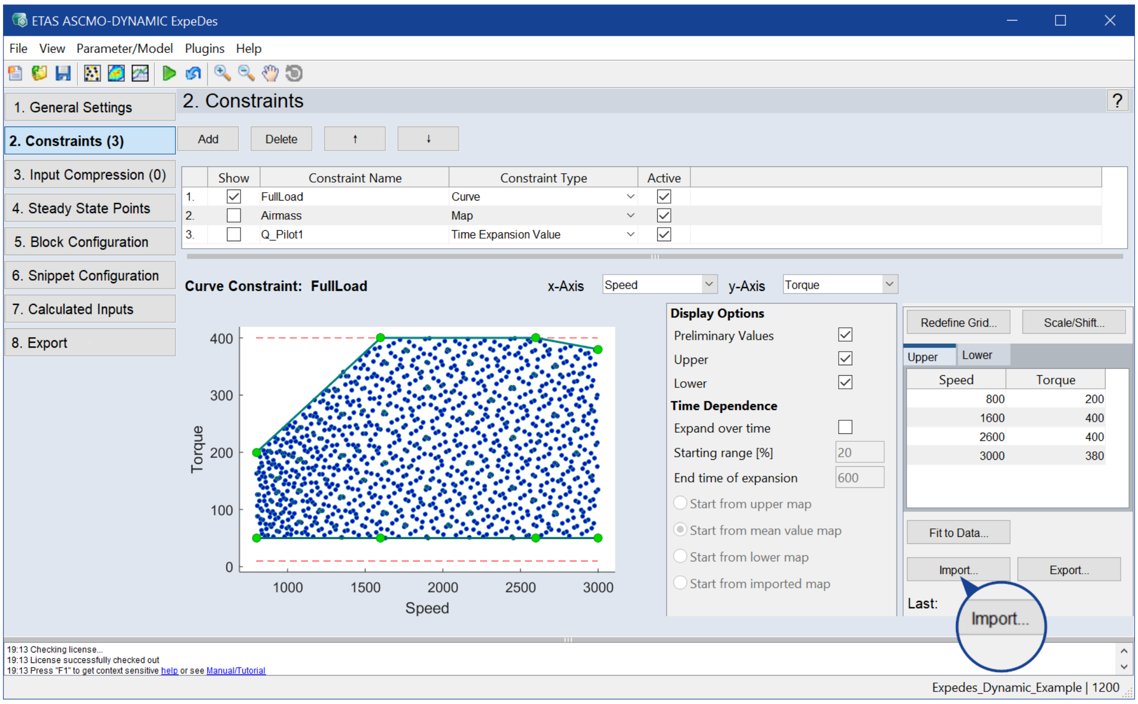
   The **Save Constraint Maps** window opens.
   iv. Select the file type.
   v. Specify the location and the file name and click **Save**.

   The maps/curves are exported. The **Export Curve and Maps** window closes.

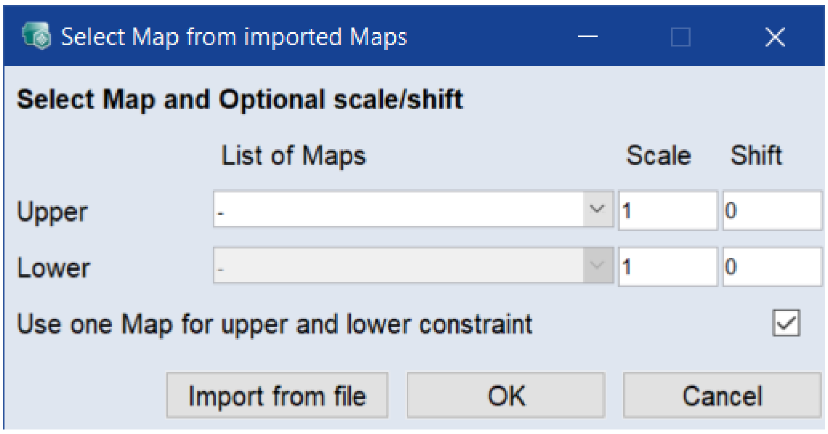
Assigning an existing map/curve to a constraint

If a map/curve already exists in the ASCMO-DYNAMIC 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. 6-3: on page 72).



The **Select Map from imported Maps** or **Select Curve from imported Curves** window opens.



Note

If no maps/curves are available in the ASCMO-DYNAMIC ExpeDes project, a file selection window opens, in which you can import map data (e.g. `*.dcm`, `*.cdfx`, `*.csv`, `*.xls`, `*.xlsx`, `*.xlsm` or `*.ascmo`) 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.

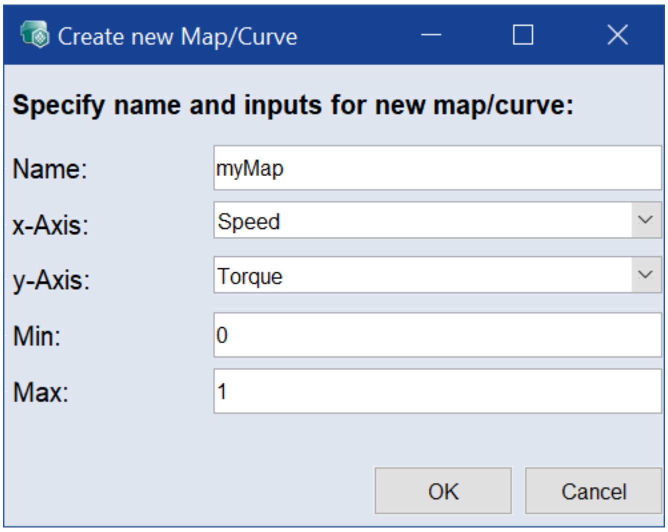
   Afterwards, the borders can be slightly extended by adding scales and shifts for the lower and upper limits via the **Scale/Shift** button.

   > **Note**
   >
   > To extend the lower borders, the value of the shift must be negative.

<u>Creating a map/curve</u>

1. In the ASCMO-DYNAMIC ExpeDes window, select **Parameter/Model › New**.

   The **Create new Map/Curve** window opens.

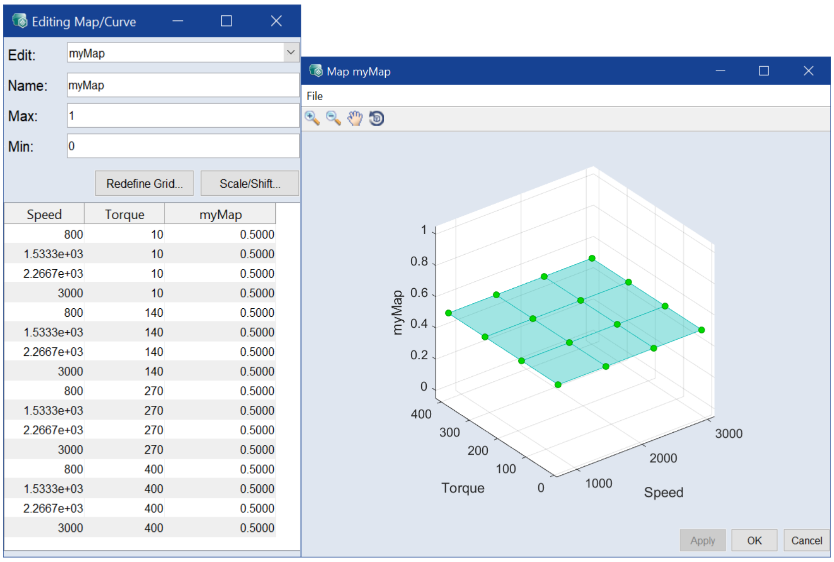
2. Enter a name.
3. Select the inputs at which the map/curve is defined.
4. Enter upper (**Max**) and lower (**Min**) limit.
5. Click **OK**.

   The **Editing Map/Curve** and the **Map *‹name›*** or **Curve *‹name›*** windows open. Here you can edit the new map.

<u>Editing a map/curve</u>

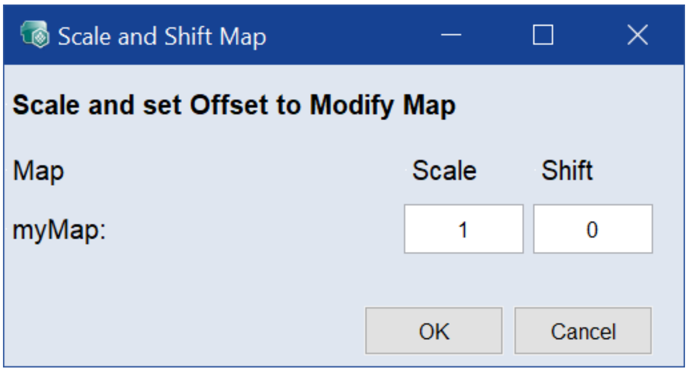
1. In the ASCMO-DYNAMIC ExpeDes main window, select **Parameter/Model › Edit › *‹map/curve name›***.

   The **Editing Map/Curve** and the **Map *‹name›*** or **Curve *‹name›*** windows open.



2. To redefine the grid points, do the following:
   i. In the **Editing Map/Curve** window, click **Redefine Grid**.
   ii. Then proceed as described in "Changing the number of grid nodes" on page 75.
3. To scale and/or shift the map/curve values, do the following:
   i. In the **Editing Map/Curve** window, 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**.

Renaming a map/curve

1. In the ASCMO-DYNAMIC ExpeDes main window, select **Parameter/Model › Edit › *‹map/curve name›***.

   The **Editing Map/Curve** and the **Map *‹name›*** or **Curve *‹name›*** windows open.

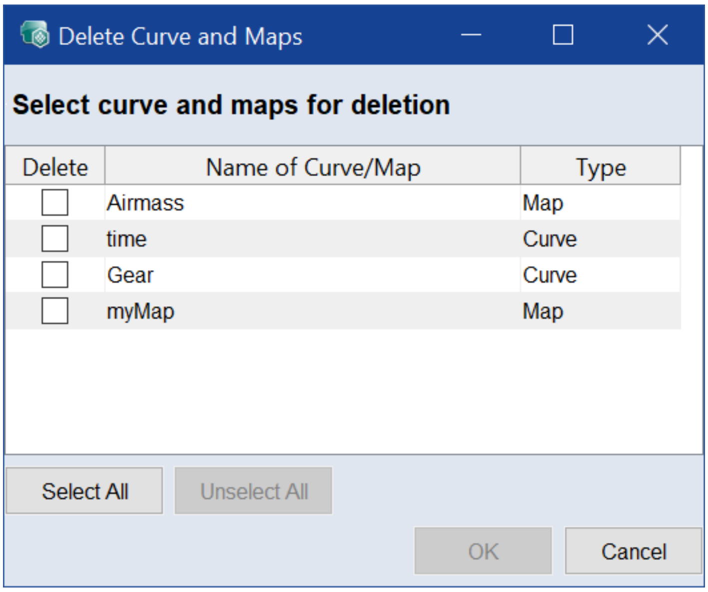
2. In the **Name** field, edit the name of the map/curve.
3. In the **Map *‹name›*** or **Curve *‹name›*** window, click **OK** to accept the change and close the windows.

   Apply accepts the changes without closing the windows.

<u>Deleting a map/curve</u>

1. In the ASCMO-DYNAMIC ExpeDes main window, select **Parameter/Model › Delete**.

   The **Delete Curve and Maps** window opens.

2. In the **Delete** column, select the maps/curves you want to delete.
3. Click **Delete**.

   The **Confirm** window opens. Here you can confirm or cancel the deletion.
4. Click **Yes** to delete the selected maps/curves.

6.5 Step 3: 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.

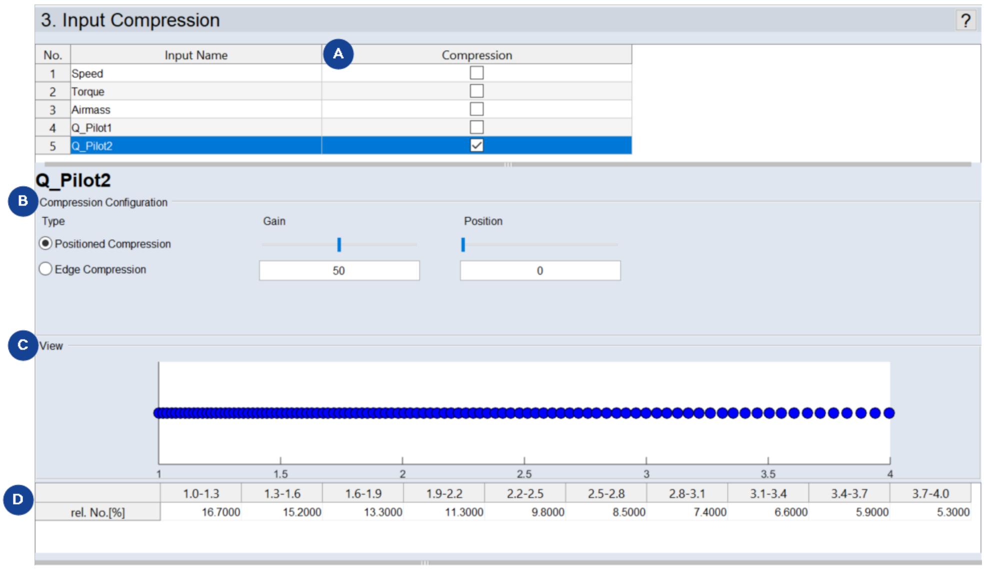


Fig. 6-4: ASCMO-DYNAMIC ExpeDes Step 3: Input compression (type "Positioned Compression")

<u>Selecting inputs to be compressed</u>

1. In the **Compression** column, activate the checkbox for each input you want to compress (see **A** in Fig. 6-4: above).

⇨ The input currently selected in the list is shown below the list.

6.5.1 Compression Configuration Area

In the **Compression Configuration** area (**B** in Fig. 6-4: above), you can define the type and the orientation of compression. There are two types of compression.

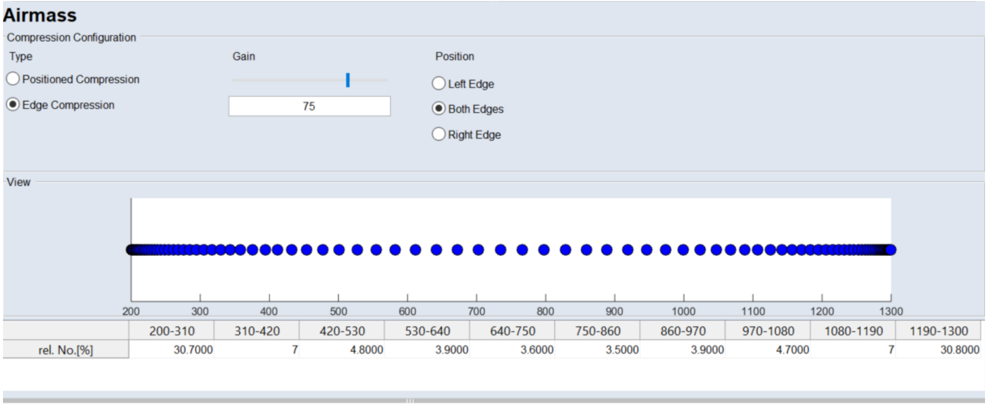
- **Positioned Compression**: With this type, compression focuses on a freely selectable point.
  - **Gain**: Degree of compression for the selected center (**Position**)
  - **Position**: Position of the center of compression
- **Edge Compression**: This type allows obtaining a compression to one or 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.

6.5.2 View Area

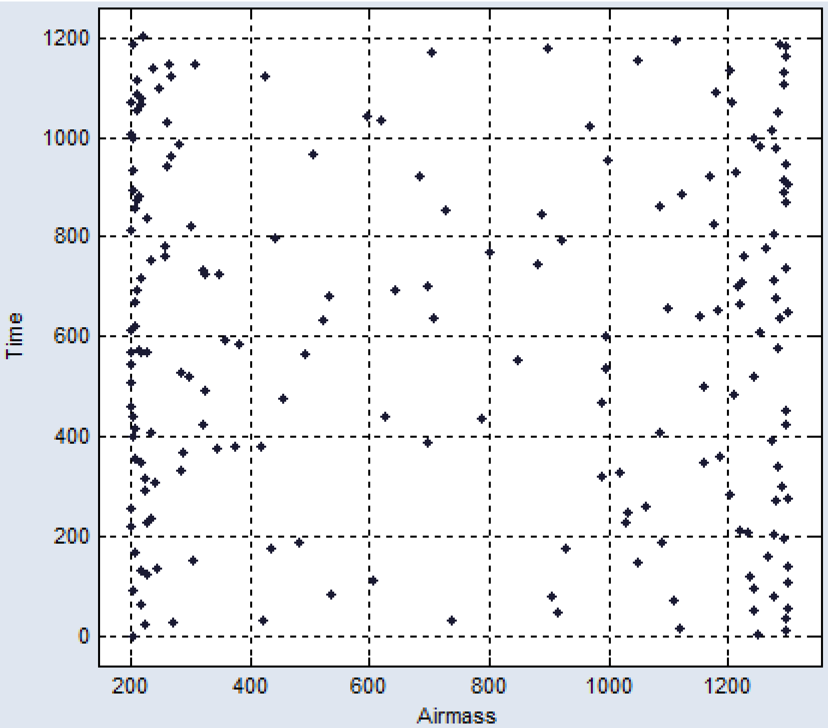
In the **View** area (**C** and **D** in Fig. 6-4: above), the position of the measurement points and, therefore, the measure of the selected compression is graphically displayed. Underneath it, the number of measurements in the respective interval resulting from the compression is specified under **rel. Size in %**.

6.5.3 Example of an Applied Input Compression

In the figure below, the measurements of `Airmass` (with no constraint) are compressed using **Edge Compression** with a gain of 75 to both edges of the measurement range.



If you visualize the applied compression in a scatter plot (**View › Scatter Plot**), you can see that the space-filling values of the experimental plan are now compressed in the direction of the edges.



6.6 Step 4: Steady State Points

This step allows you to define operating points for the system, which have a frequency of stay in the experiment plan. In addition, you can define steady state phases and the duration of the steady state phases.

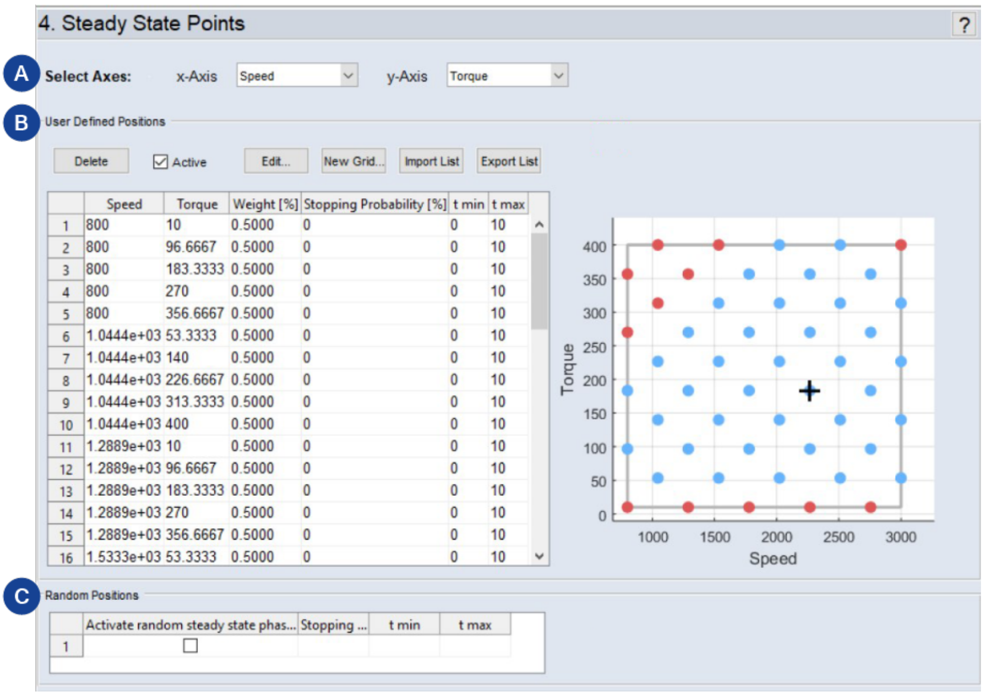


Fig. 6-5: ASCMO-DYNAMIC ExpeDes Step 4: Steady State Points

6.6.1 "Select Axes" Area

In the **Select Axes** area (**A** in Fig. 6-5: above), you can set the axes for the steady state points. With them, you can define the operating points, which have to be considered explicitly in the experiment plan. The values of the operating point axes can be specified in the **User Defined Positions** area, and the remaining inputs will be varied according to the experiment plan.

6.6.2 "User Defined Positions" Area

Area **B** (Fig. 6-5: above) shows steady state points in a table and a plot. You can add, import/export, edit, and delete steady state points.

<u>Adding a steady state point manually</u>

> **Note**
>
> To activate the **User Defined Positions** area, you have to define input axes in the **Select Axes** area.

1. In the `New` row, click in a cell.

   The cell becomes an input field.

2. Enter the value for the column.

All columns must be edited for a valid steady state point.

Column	Meaning
Input names (defined as `x-Axis` and `y-Axis`)	Values of the steady state point (e.g. Speed, Torque).
Weight	Relative frequency of how often the steady state point is hit in the DoE. The higher the value, the more often the point is hit.

Note

The sum of all weights must be ≤ **75**.

Column	Meaning
Stopping Probability	Probability (in [0 .. 100]) that a steady state phase occurs when the point is hit. 0 means steady state phase never occurs 100 means steady state phase always occurs
t min, t max	[t min .. t max] is the time interval from which the stopping time is drawn.

The steady state point is shown in the plot on the right side of the **User Defined Positions** area.

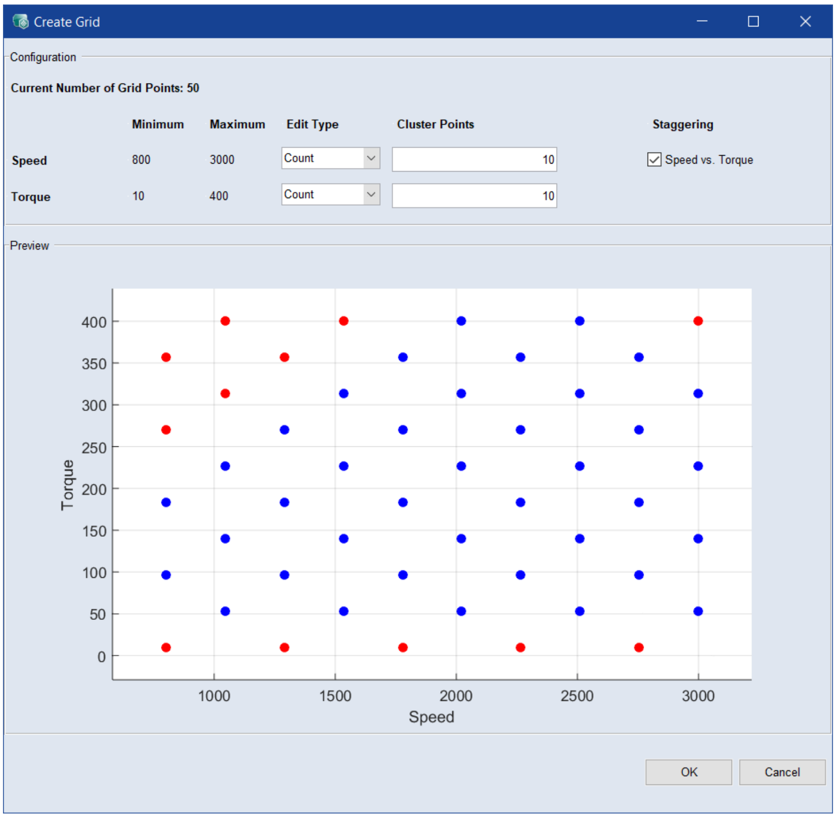
Adding a grid of steady state points

Note

Existing steady state points are deleted when you add a grid of steady state

1. In the **User Defined Positions** area, click **New Grid**.

   The **Create Grid** window opens.

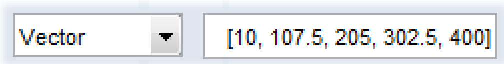


2. To change the number of points on an axis, do the following:
   i. In the drop-down list for the axis, select `Count`.

      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.
5. Click **OK**.

The steady state points are added to the table and the plot. The **Create Grid** window closes.

Activating/deactivating steady state points

1. To activate all steady state points, activate the **Active** option.

   Each steady state point is considered in the experiment plan, unless it is deactivated individually.

2. To deactivate a single steady state point, set its weight to `0`.
3. To deactivate all steady state points, deactivate the **Active** option.

   You can still add, edit, import/export, and delete steady state points, but they will not be considered in the experiment plan.

Editing a single steady state point

1. In the desired row, click in the cell you want to edit.

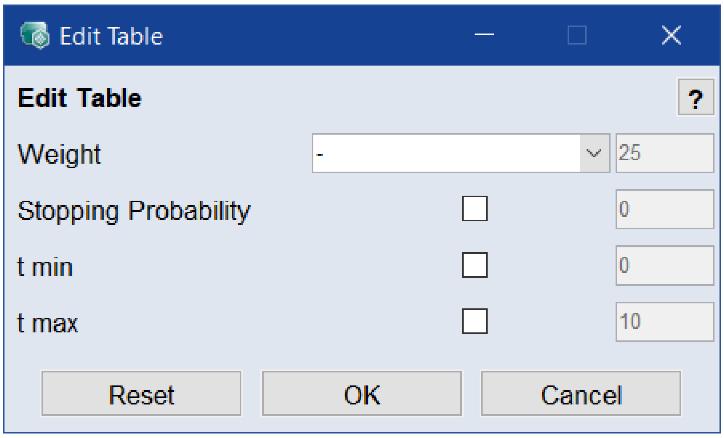
   The cell becomes an input field.

2. Enter the desired value for the column.

Editing all steady state points

1. In the **User Defined Positions** area, click **Edit**.

   The **Edit Table** window opens.



2. In the **Weight** drop-down list, select the way you want to change the weights.

   `Equally distribute weights with total quantity of ...`: The weights of all n steady state points are set to the same fraction sum/n of the desired sum of all weights.

   `Renorm current weights to total quantity of ...`: The weights of all steady state points are scaled so that the sum reaches the desired value. The ratio of the weights is kept.

3. In the input field, enter the sum of all weights.
4. In the **Stopping Probability**, **t min** and **t max** rows, do the following:

   i. Activate the checkbox if you want to apply the options.

   ii. In the input field, enter the value you want to assign to all steady state points.

5. Click **OK**.

⇨ The steady state points are changed according to your settings.

Deleting steady state points

1. In the steady state point table, select the row(s) you want to delete.
2. Click **Delete**.

   The steady state points are deleted without further notice.

Importing steady state points

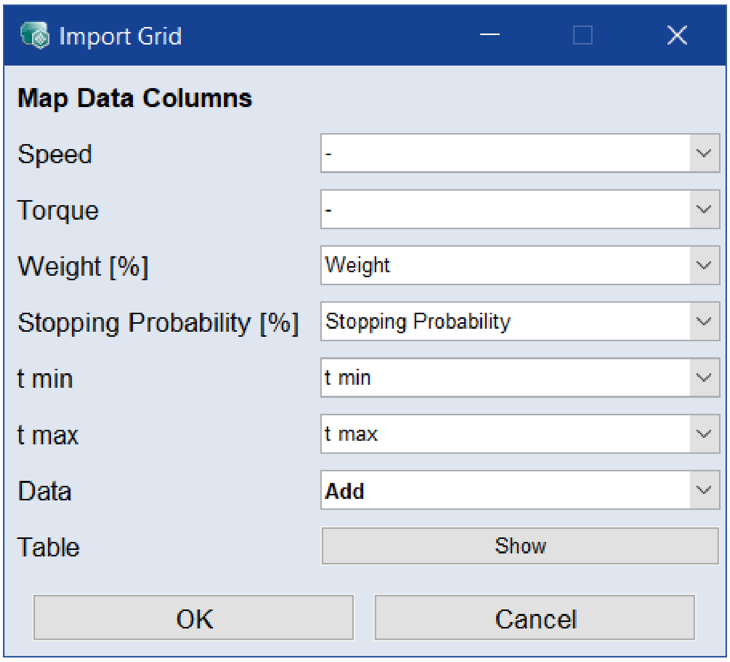
If a steady state point grid exists as `*.csv` or `*.xls` / `*.xlsx` / `*.xlsm` file, it can be imported.

1. In the **User Defined Positions** area, click **Import List**.

   A file selection window opens.
2. Select the file you want to import and click **Open**.

   If you selected a file with several work sheets, the **Select Sheet** window opens.
3. Select the desired work sheet and click **OK**.

   The **Import Grid** window opens. The drop-down lists contain the column names found in the import file.



4. In the **Import Grid** window, map the columns of the steady state point table to appropriate columns of the import file.
5. In the **Data** drop-down list, select the import mode (`Add` or `Replace`).
6. Click **OK** to start the input.

Exporting steady state points

You can export the entire list of steady state points. Exporting only parts of the list is not possible. Available export formats are e.g. `*.csv`, `*.xls` and `*.xlsx`.

1. In the **User Defined Positions** area, click **Export List**

   A file selection window opens.
2. Select the file type.
3. Specify the location and the file name and click **Save**.

6.6.3 "Random Positions" Area

In the **Random Positions** area (in Fig. 6-5: on page 84), you can enable random steady state phases, i.e. steady state phases, that occur with a given probability.

Enabling random steady state phases

1. In the **Random Positions** area, activate **Activate random steady state phases**.
2. In the **Stopping Probability** column, enter a value for the probability (in [0 .. 100]) that a steady state phase occurs.

   0 means steady state phase never occurs
   100 means steady state phase always occurs
3. In the **t min** and **t max** fields, enter the borders of the time interval from which the stopping time is drawn.

6.7 Step 5: 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 and is space filling.

The advantage of block building is that the effort involved in measuring is reduced. By measuring just a few blocks and if the model precision achieved is satisfactory, measuring can be stopped early without measuring all blocks.

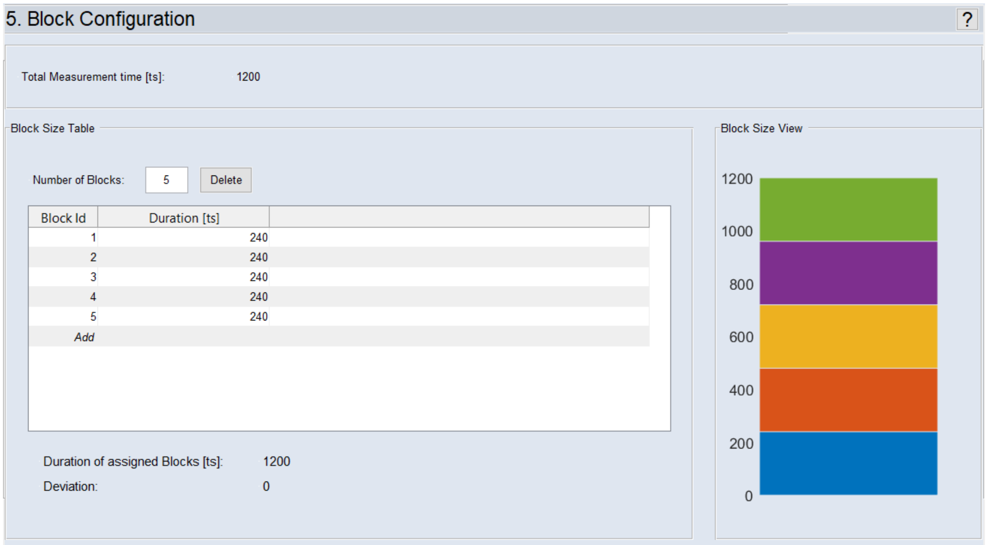


Fig. 6-6: ASCMO-DYNAMIC ExpeDes Step 5: Block Configuration

"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**.

"Block Size View" Area

The size of the blocks (defined in the **Block Size Table** area) is graphically displayed here.

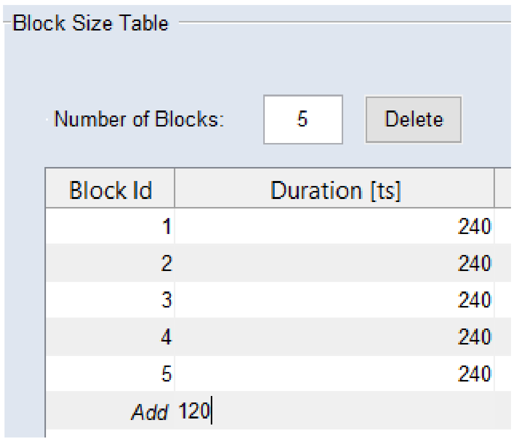
Creating and Editing Blocks

1. In the **Number of Blocks** field, enter a number and press ‹RETURN›.

   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›.



A new block is added.

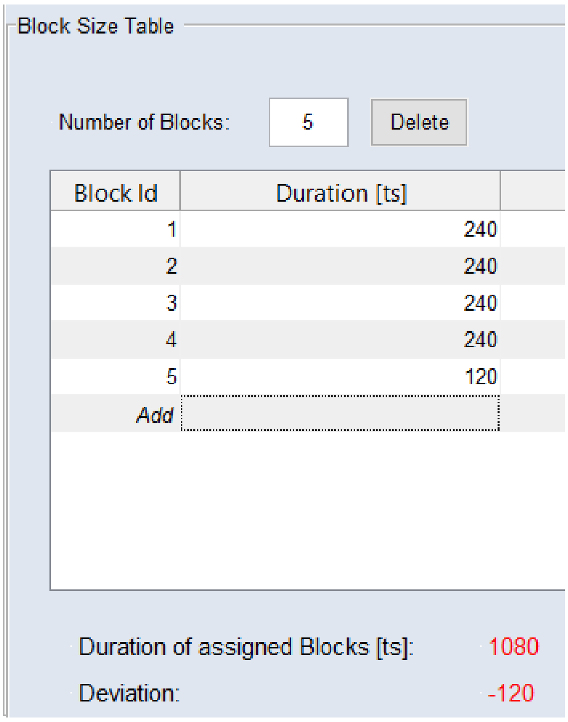
2. Adjust the **Duration [ts]** for the previous blocks, so that the **Duration of assigned Blocks [ts]** equals the **Total Measurement time [ts]**.

> **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 Duration 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.

6.8 Step 6: Snippet Configuration

This step allows you to include external cycles into your experiment plan.

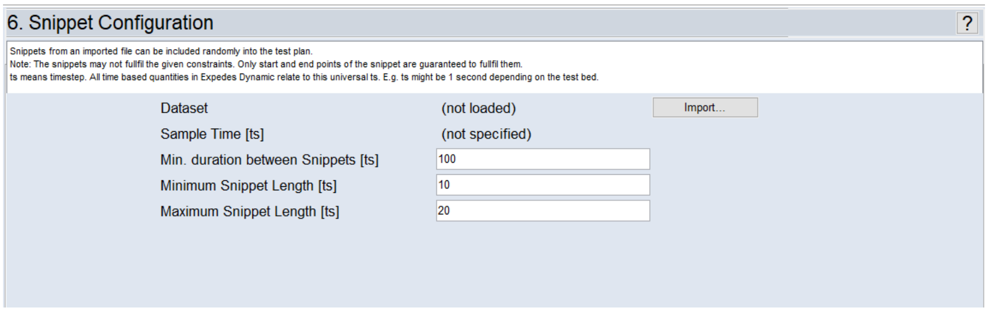


Fig. 6-7: ASCMO-DYNAMIC ExpeDes Step 6: Snippet Configuration

This step is skipped in the tutorial. See the online help for further information (F1).

6.9 Step 7: 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.

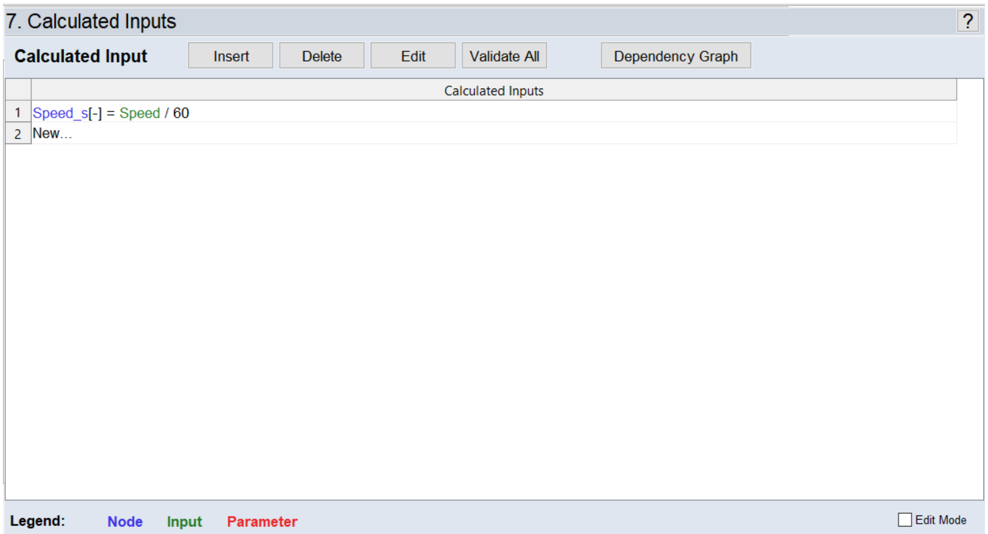


Fig. 6-8: ASCMO-DYNAMIC ExpeDes Step 7: Calculated Inputs

This step is skipped in the tutorial. See the online help for further information (F1).

6.10 Step 8: Export

> ***NOTICE***
>
> **Damage due to wrong test plan**
>
> Wrong engine settings in ASCMO-DYNAMIC 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-DYNAMIC ExpeDes see 6.10 "Step 8: Export" above and 6.2 "Step 1: General Settings" on page 66.

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, density plots, scope views, or as a table.

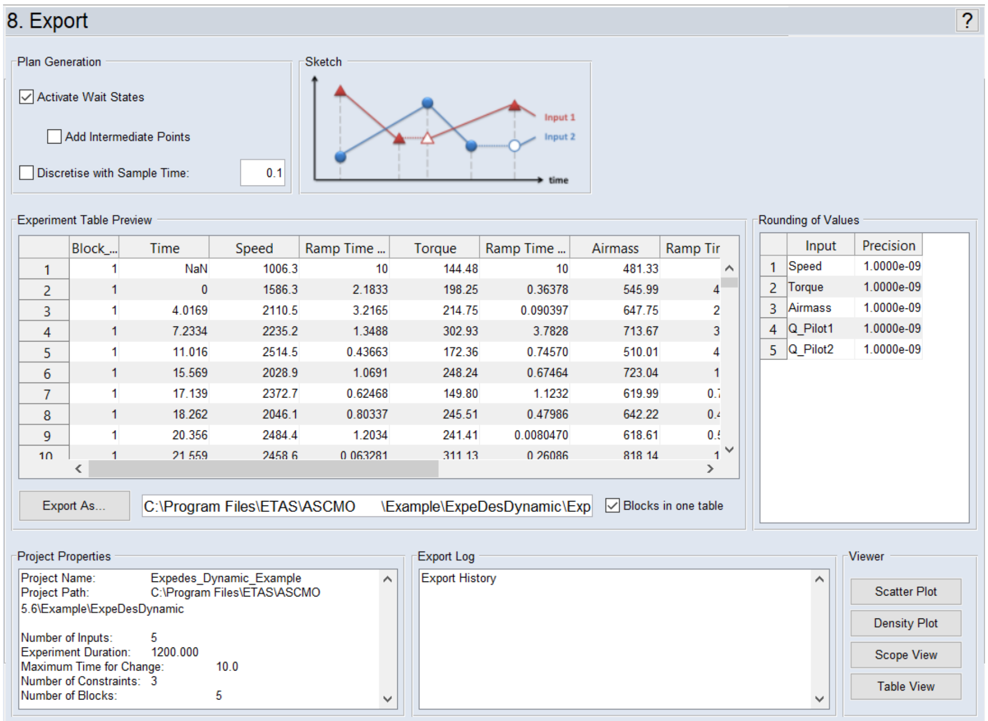
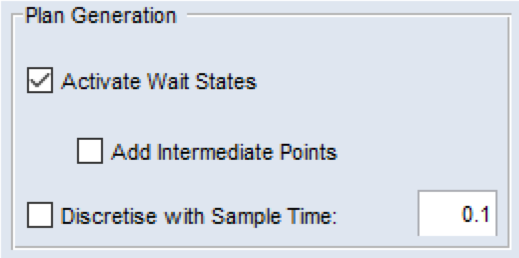


Fig. 6-9: ASCMO-DYNAMIC ExpeDes Step 8: Export

Plan Generation

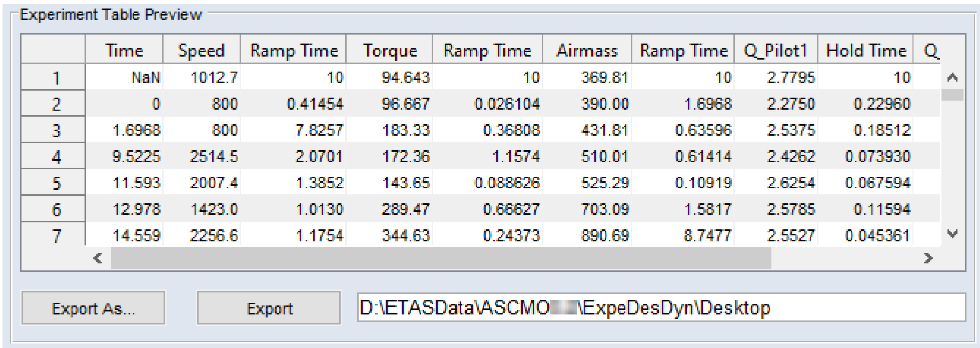
In the **Plan Generation** area, you can define which values are to be included in the export table of the experimental design.

Plan Generation – Type 1



The target parameters for the engine dynamometer are already specified. The absolute time (**Time** column), until all the parameters for a line have been processed by the test rig is indicated in the export table.

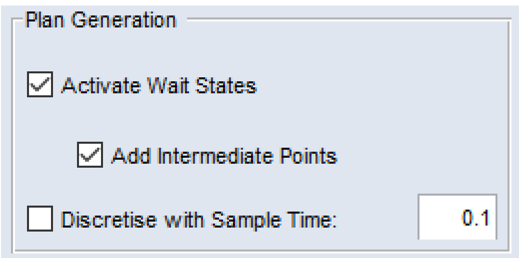
If an input is defined as the type **Ramp**, the **Ramp Time** column to its right indicates the amount of time specified by the test for reaching the target value. The **Ramp time** depends on the predefined gradients. For the input type **Step**, the **Hold Time** column on its right specifies the length of holding the respective value.



	Time	Speed	Ramp Time	Torque	Ramp Time	Airmass	Ramp Time	Q_Pilot1	Hold Time	Q
1	NaN	1012.7	10	94.643	10	369.81	10	2.7795	10	
2	0	800	0.41454	96.667	0.026104	390.00	1.6968	2.2750	0.22960	
3	1.6968	800	7.8257	183.33	0.36808	431.81	0.63596	2.5375	0.18512	
4	9.5225	2514.5	2.0701	172.36	1.1574	510.01	0.61414	2.4262	0.073930	
5	11.593	2007.4	1.3852	143.65	0.088626	525.29	0.10919	2.6254	0.067594	
6	12.978	1423.0	1.0130	289.47	0.66627	703.09	1.5817	2.5785	0.11594	
7	14.559	2256.6	1.1754	344.63	0.24373	890.69	8.7477	2.5527	0.045361	

Fig. 6-10: Preview for the export table format "Target Values with waitstates"

Plan Generation – Type 2



Similar to Type 1, but intermediate points are added to the table.

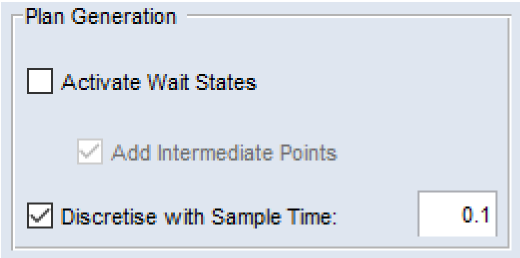
Experiment Table Preview

	Time	Ramp Time	Speed	Torque	Airmass	Q_Pilot1	Q_Pilot2
1	0	0.026104	1012.7	94.643	369.81	2.7795	2.8967
2	0.026104	0.38843	999.30	96.667	370.12	2.2750	3.3405
3	0.41454	1.2822	800	96.667	374.74	2.2750	3.3405
4	1.6968	0.36808	800	96.667	390.00	2.2750	3.3405
5	2.0649	0.26788	800	183.33	414.20	2.5375	1.8648
6	2.3327	7.1897	800	183.33	431.81	2.5375	1.8648
7	9.5225	0.61414	800	183.33	431.81	2.5375	1.8648
8	10.137	0.54329	1308.6	177.51	510.01	2.4262	1.6988
9	10.680	0.91263	1758.6	172.36	510.01	2.4262	1.6988
10	11.593	0.088626	2514.5	172.36	510.01	2.4262	1.6988
11	11.681	0.020559	2482.0	143.65	522.42	2.6254	1.5099
12	11.702	1.2760	2474.5	143.65	525.29	2.6254	1.5099
13	12.978	0.66627	2007.4	143.65	525.29	2.6254	1.5099
14	13.644	0.34673	1623.1	289.47	600.19	2.5785	3.3870
15	13.991	0.56866	1423.0	289.47	639.17	2.5785	3.3870
16	14.559	0.24373	1423.0	289.47	703.09	2.5785	3.3870

Export As... | Export | D:\ETASData\ASCMO[illegible]\ExpeDesDyn\Desktop

Fig. 6-11: Preview for the export table format "Target Values with waitstates and intermediate points"

Plan Generation – Type 3



All output values are dependent on the information in the **sample time [s]** field. The **sample time** defines the time delay until a new measurement point for all inputs will be detected. In this connection it does not matter if an input parameter has been reached by the engine dynamometer.

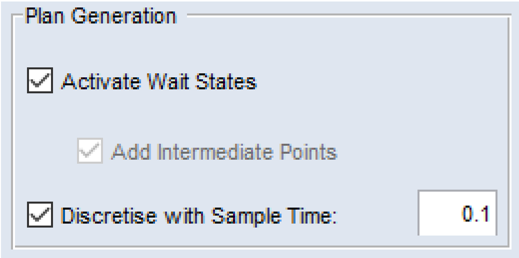
Experiment Table Preview

	Time	Speed	Torque	Airmass	Q_Pilot1	Q_Pilot2
1	0	1464.9	83.816	268.87	2.6819	1.4758
2	0.10000	1480.5	89.279	283.83	2.6819	1.4758
3	0.20000	1496.1	94.742	298.79	2.6819	1.4758
4	0.30000	1511.7	100.20	313.75	2.6819	1.4758
5	0.40000	1527.3	105.67	328.71	2.6819	1.4758
6	0.50000	1542.9	111.13	343.68	2.6819	1.4758
7	0.60000	1558.4	116.59	358.64	2.6819	1.4758

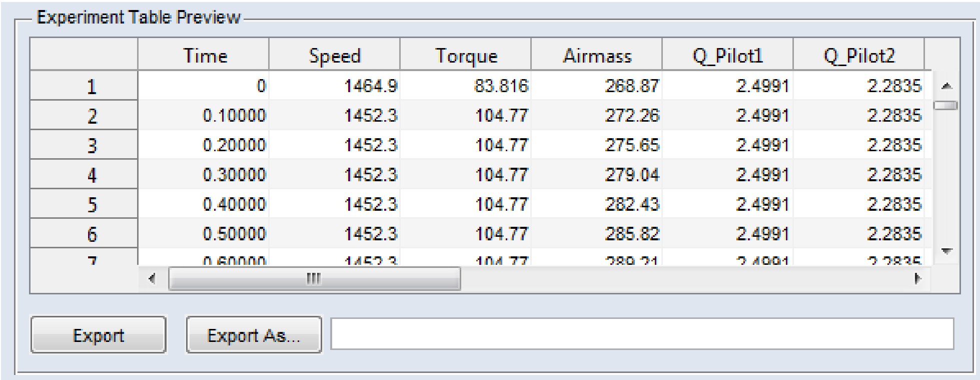
Export | Export As...

Fig. 6-12: Preview for the export table format "Target Values"

Plan Generation – Type 4

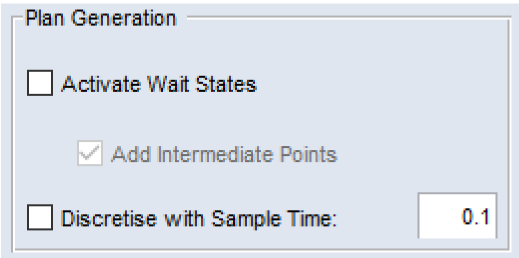


This Plan Generation is similar to the Activate Wait States checkbox. But the path will be recalculated depending on the **Sample Time**.

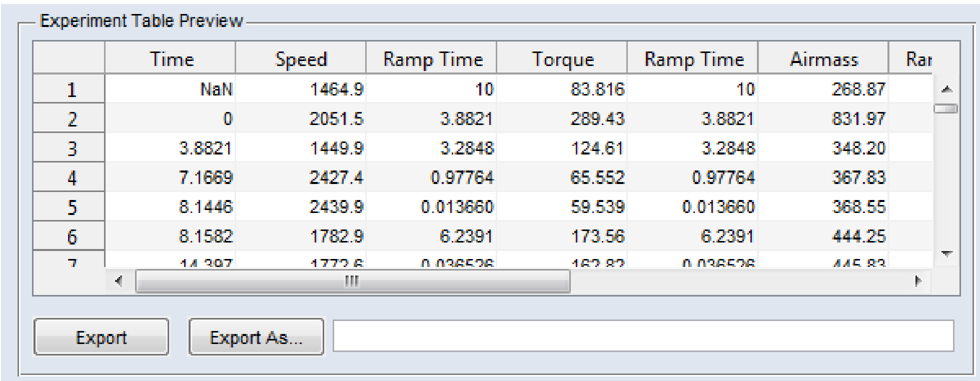


	Time	Speed	Torque	Airmass	Q_Pilot1	Q_Pilot2
1	0	1464.9	83.816	268.87	2.4991	2.2835
2	0.10000	1452.3	104.77	272.26	2.4991	2.2835
3	0.20000	1452.3	104.77	275.65	2.4991	2.2835
4	0.30000	1452.3	104.77	279.04	2.4991	2.2835
5	0.40000	1452.3	104.77	282.43	2.4991	2.2835
6	0.50000	1452.3	104.77	285.82	2.4991	2.2835

Plan Generation – Type 5



When an input has reached the predetermined target value, the present value of the other inputs will be detected.



	Time	Speed	Ramp Time	Torque	Ramp Time	Airmass	Rar
1	NaN	1464.9	10	83.816	10	268.87	
2	0	2051.5	3.8821	289.43	3.8821	831.97	
3	3.8821	1449.9	3.2848	124.61	3.2848	348.20	
4	7.1669	2427.4	0.97764	65.552	0.97764	367.83	
5	8.1446	2439.9	0.013660	59.539	0.013660	368.55	
6	8.1582	1782.9	6.2391	173.56	6.2391	444.25	

Fig. 6-13: Preview for the export table format "Full Trace"

> **Note**
>
> The **Sample Time** can only be defined if you have specified the export table format Type 3/4. Doing this, the time must not be higher than the information in the **Maximum Time for Change** field.
>
> If you enter a higher value, the **Conflict** window opens. Use the drop-down list to decide which value you want to keep (Maximum time for change (<m>) or Sample time (<n>)). The selected value is assigned to both **Sample Time** and **Maximum Time for Change**.

Precision

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).

Changing the rounding of values

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 Experiment Plan

When you selected the plan generation mode and the input precisions, you can export the experiment plan.

Exporting the plan

To export the experiment plan, proceed as follows.

1. Click **Export** to export the plan using the most recently selected path, file name, and format.

   Or

   In the **Experiment Table Preview** area, click **Export As**.

   A file selection window opens.

2. Select the export file type.

   Available file 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 inquiry.

4. Click **Save**.

   The plan is exported according to your settings.

"Project Properties" and "Export Log" Areas

In the **Project Properties** area , the properties of the project and the experiment plan are displayed.

The **Export Log** area contains information on previous exports.

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

Fig. 2-1: From experiment plan to model-based optimization ... 11
Fig. 2-2: Model structure for nonlinear autoregression with exogenous inputs (NARX) ... 14
Fig. 5-1: Illustration of the Measured Diesel Engine ... 25
Fig. 5-2: ASCMO-DESK window ... 26
Fig. 5-3: ASCMO-DYNAMIC start window ... 26
Fig. 5-4: ASCMO-DYNAMIC scatter plots (default plots) ... 30
Fig. 5-5: Phase plot and ACF/IACF plot for an output ... 35
Fig. 5-6: Model Configurations window ... 37
Fig. 5-7: Model Configuration All Outputs window ... 38
Fig. 6-1: ASCMO-DYNAMIC ExpeDes start window ... 64
Fig. 6-2: ASCMO-DYNAMIC ExpeDes main window ... 65
Fig. 6-3: ASCMO-DYNAMIC ExpeDes Step 2: Constraints (Type Curve) ... 72
Fig. 6-4: ASCMO-DYNAMIC ExpeDes Step 3: Input compression (type "Positioned Compression") ... 82
Fig. 6-5: ASCMO-DYNAMIC ExpeDes Step 4: Steady State Points ... 84
Fig. 6-6: ASCMO-DYNAMIC ExpeDes Step 5: Block Configuration ... 90
Fig. 6-7: ASCMO-DYNAMIC ExpeDes Step 6: Snippet Configuration ... 92
Fig. 6-8: ASCMO-DYNAMIC ExpeDes Step 7: Calculated Inputs ... 92
Fig. 6-9: ASCMO-DYNAMIC ExpeDes Step 8: Export ... 93
Fig. 6-10: Preview for the export table format "Target Values with waitstates" ... 94
Fig. 6-11: Preview for the export table format "Target Values with waitstates and intermediate points" ... 95
Fig. 6-12: Preview for the export table format "Target Values" ... 95
Fig. 6-13: Preview for the export table format "Full Trace" ... 96

Index

A

ASCMO
- p-code version ... 18

ASCMO-DYNAMIC
- start ... 26
- tutorial ... 25

ASCMO-STATIC
- design of experiments ... 11
- user interface ... 21

B

Block configuration (ExpeDes)
- delete block ... 91

C

Compression
- see Input compression ... 81

Concept
- multi-step ahead prediction ... 13
- one-step ahead prediction ... 13

Constraints (ExpeDes Dynamic) ... 71
- add ... 72
- assign curve ... 77
- assign map ... 77
- curve ... 73
- delete ... 72
- manage ... 72
- map ... 73

Contact Information ... 105

Cross validation ... 48

Curve
- assign to constraint ... 77
- change number of grid nodes ... 75
- constraint ... 73
- create ... 79
- delete ... 81
- edit ... 79
- export data ... 76
- import ... 75
- rename ... 80

D

Data analysis ... 29
- amplitude spectra ... 33
- cross correlation input ... 32
- inverse autocorrelation function ... 35
- noise filtering ... 31
- phase plot ... 34
- resampling ... 30
- scatter plot ... 29

Data import ... 26

Design of experiments ... 11

Dimensionality reduction ... 40
- feature selection (automatic) ... 42
- feature selection (manual) ... 41

E

ETAS
- Contact Information ... 105

ExpeDes Dynamic
- constraints ... 71
- general settings ... 66
- input compression ... 81
- input configuration ... 67
- measurement size ... 68
- start ... 64
- steady state point ... 83
- tutorial ... 63,98
- visualize experiment plan ... 68

Experiment plan
- visualize ... 68

Export ... 53-54
- curve data ... 76
- map data ... 76
- model (ASCMO Dynamic) ... 51
- steady state point ... 89

G

glossary ... 99

I

Import
- curve ... 75
- map ... 75
- steady state point ... 88

- test data 45
- training data 27

Input compression 81
- edge 82
- example 83
- positioned 82
- select input 82

Input configuration (ExpeDes Dynamic) 67

Installation
- directories 17
- files 17
- license agreement 16
- uninstall MOCA 19

Introduction 6

Inverse autocorrelation function 35

M

Map
- assign to constraint 77
- change number of grid nodes 75
- constraint 73
- create 79
- delete 81
- edit 79
- export data 76
- import 75
- rename 80

Measurement size
- configure (ExpeDes Dynamic) 68

MOCA
- uninstall 19

Model export (ASCMO Dynamic) 51
- C code 55
- FMI 57
- GT-Suite 56
- Simulink 52

Model export (ASCMO Static)
- Matlab 51
- Python 52

Model prediction 43
- multi-step ahead prediction 44
- one-step ahead prediction 44
- sigma prediction 44

Model training 36
- dimensionality reduction 40
- feedback structure 40
- model properties (details) 36
- model properties (few) 38
- modeling algorithm 39
- start 36,38

Model validation 45
- cross validation 48
- measured vs. predicted 47
- scatter plot 46

Modeling algorithm 39

Multi-step ahead prediction 13,44

N

Noise filtering 31

O

One-step ahead prediction 13,44

P

Phase plot 34

S

Sigma prediction 44

Start
- ASCMO Dynamic 26
- ExpeDes Dynamic 64

Steady state point 83
- activate 87
- add manually 84
- deactivate 87
- delete 88
- edit 87
- edit all 87
- export 89
- import 88
- random 89

T

Test data
- import 45

Training data
- import 27

Tutorial 25,98
- constraints (ExpeDes Dynamic) 71
- data analysis 29
- data import 26
- general settings (ExpeDes Dynamic) 66

input compression (ExpeDes Dynamic) ... 81
model export ... 51
model prediction ... 43
model training ... 36
model validation ... 45
steady state point ... 83
visualize experiment plan (ExpeDes Dynamic) ... 68

U

User interface ... 21

10 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