ASM Electric Components

Model Description

Release 2021-A – May 2021

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

Important Notice

This publication and the contents hereof are subject to change without notice.

Contents

About This Document

Content

This document shows you how to work with the ASM Electric Components Blockset.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
⚠ DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
⚠ WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
⚠ CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
❔	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
📖	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

`%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>`

or

`%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>`

Documents folder A standard folder for user-specific documents.

`%USERPROFILE%\Documents\dSPACE\<ProductName>\<VersionNumber>`

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

`%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\<ProductName>`

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the 📄 icon in dSPACE Help. The PDF opens on the first page.

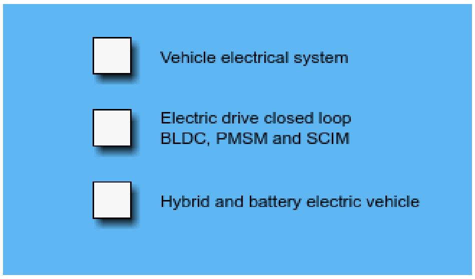
Introduction

Demo Information

Introduction

The demos of ASM Electric Components can be divided into three basic parts:

- The simulation of a vehicle electrical system
- The closed-loop simulation of electric machines with a real ECU or a soft ECU
- The simulation of a fully hybrid or battery electric vehicle


□ Vehicle electrical system
□ Electric drive closed loop BLDC, PMSM and SCIM
□ Hybrid and battery electric vehicle

Vehicle electrical system

The Vehicle electrical System model simulates the battery voltage, depending on the charge and discharge current of the starter, the alternator and different electrical loads, such as air conditioning, the lighting system, PTC heater, and electric machine. Two battery systems are included as the power supply. A standard low voltage battery that represents the lead-acid starter battery and two multicell high voltage lithium ion batteries connected in a parallel circuit. The torque load of the engine crankshaft produced by electric components is also simulated. Refer to Overview of the Vehicle Electrical System Model (ASM Electric Components Reference 📖).

Electric drive closed-loop simulation

ASM provides the following demo models for the closed-loop simulation of electric drives:

Brushless DC machine (BLDC) The Brushless DC machine (BLDC) demo simulates a BLDC with a three-phase inverter, a battery, and a BLDC controller. For more information, refer to Overview of the Electric Drive Closed Loop BLDC Model (ASM Electric Components Reference 📖).

Permanent magnet synchronous machine (PMSM) The Permanent magnet synchronous machine (PMSM) demo simulates a PMSM with a nonlinear behavior of the inductances and a three-phase DCM inverter that is able to simulate natural switching effects, such as a flyback diode, and changing energy flow, such as a passive energy recovery into a battery or a capacitor. A supercapacitor is used as power supply for the system. The voltage is controlled according to its reference by setting the input current. The system is controlled with a PMSM controller. For details, refer to Overview of the Electric Drive closed Loop PMSM Nonlinear Model (ASM Electric Components Reference 📖).

Permanent magnet synchronous machine (PMSM) with 3-level inverter The Permanent magnet synchronous machine (PMSM) with 3-level inverter demo model simulates a PMSM, a three-level three-phase inverter, a capacitive voltage divider and a supercapacitor as power supply. The system is controlled with a three-level PMSM controller. Refer to Overview of the Electric Drive closed Loop PMSM Nonlinear with 3-Level Inverter Model (ASM Electric Components Reference 📖).

Squirrel cage asynchronous machine (SCIM) The Squirrel cage asynchronous machine (SCIM) demo simulates an SCIM and a three-phase DCM inverter that is able to simulate natural switching effects, such as the free-wheeling diode, and changing energy flow, such as passive energy recovery into a battery or a capacitor. The model also includes a three-phase power supply, a three-phase rectifier and a DC link. For more details, see Overview of the Electric Drive Closed Loop SCIM Model (ASM Electric Components Reference 📖).

Hybrid and battery electric vehicle

There are two hybrid vehicle and two battery electric vehicle demo models:

Vehicle dynamics hybrid The Vehicle dynamics hybrid demo simulates the dynamics of a passenger car. The vehicle is composed of engine, electric components, rigid parallel hybrid drivetrain, rigid vehicle body, and four wheels. The model represents the vehicle's longitudinal, lateral, and vertical dynamics and simulates in detail a rigid parallel hybrid drivetrain, suspension kinematics and compliance, tire-road friction forces and moments, steering, and brakes. The combustion process in the engine is not modeled. For more details on the model, refer to Overview of the Vehicle Dynamics Hybrid Model (ASM Electric Components Reference 📖). For information on working with the model, refer to Tutorials on page 11.

Note

To run the Vehicle dynamics hybrid demo model of the ASM Electric Components Library, you also need the ASM Vehicle Dynamics Library and the ASM Environment Library. Otherwise, some Simulink blocks of the demo model will have broken links to the missing libraries.

Engine gasoline hybrid The Engine gasoline hybrid demo model simulates a turbocharged gasoline engine with manifold or direct injection in combination with electric components, and a parallel hybrid drivetrain model including automatic transmission. For more details on the model, refer to Overview of the Gasoline Engine Hybrid Model (ASM Electric Components Reference 📖). For information on working with the model, refer to Tutorials on page 11.

Note

To run the Engine gasoline hybrid demo model of the ASM Electric Components Library, you also need the ASM Engine Gasoline Library and the ASM Drivetrain Basic Library. Otherwise, some Simulink blocks of the demo model will have broken links to the missing libraries.

Battery electric vehicle The Battery electric vehicle (BEV) demo model simulates the dynamics of a passenger car. The vehicle is composed of an energy storage system, one electric drive for the front axle and one for the rear axle, a drivetrain with flexible shafts, a rigid vehicle body four wheels, a serial regenerative brake system, an electric air condition, and a battery charging system with charging station. The model represents the vehicle's longitudinal, lateral, and vertical dynamics and accurately simulates a drivetrain, suspension kinematics and compliance, tire-road friction forces and moments, steering and brakes. Refer to Overview of the Battery Electric Vehicle Model (ASM Electric Components Reference 📖).

Note

To run the BEV demo model, the ASM Electric Components Library must be used in combination with the ASM Vehicle Dynamics Library and the ASM Environment Library. Otherwise, some Simulink blocks of the demo model will have broken links to the missing libraries.

Battery electric vehicle with traffic The Battery electric vehicle with traffic demo model simulates the dynamics of a passenger car, sensors, and traffic objects in the environment. The vehicle is composed of an energy storage system, one electric drive for the front axle and one for the rear axle, a drivetrain with flexible shafts, a rigid vehicle body four wheels, a serial regenerative brake system, an electric air condition, and a battery charging system with charging station as well as several sensors. The model represents the vehicle's longitudinal, lateral, and vertical dynamics and accurately simulates a drivetrain, suspension kinematics and compliance, tire-road friction forces and moments, steering and brakes. It also simulates the movements of fellow vehicles and object detection sensors. The latter can detect the simulated fellows or static objects in the environment and provides sensor information about the mentioned objects. Refer to Overview of the Battery Electric Vehicle with Traffic Model (ASM Electric Components Reference 📖).

Note

To run the **Battery electric vehicle with traffic** demo model, the ASM Electric Components Library must be used in combination with the ASM Vehicle Dynamics Library, the ASM Environment Library, and the ASM Traffic Library. Otherwise, some Simulink blocks of the demo model will have broken links to the missing libraries.

Tutorials

Where to go from here

Information in this section

Handling the Model

Where to go from here

Information in this section

Introduction to the Tutorial

Introduction

For the different demo models of the ASM Electric Components Model, different tutorials are available:

- For the vehicle electrical system model and the electric machine models: Tutorials on page 11
- For the gasoline engine hybrid model: Tutorials on page 11

- For the vehicle dynamics hybrid model: Tutorials on page 11
- For the battery electric vehicle (BEV) model: Tutorials on page 11

How to Open an ASM Library

Objective

After successfully installing ASM on your PC, you can now easily open an ASM library.

This lesson explains how to open the Gasoline Engine library. You can start all ASM products in a similar way. The MATLAB Command Window shows you all the products that are installed and covered by your license.

Method

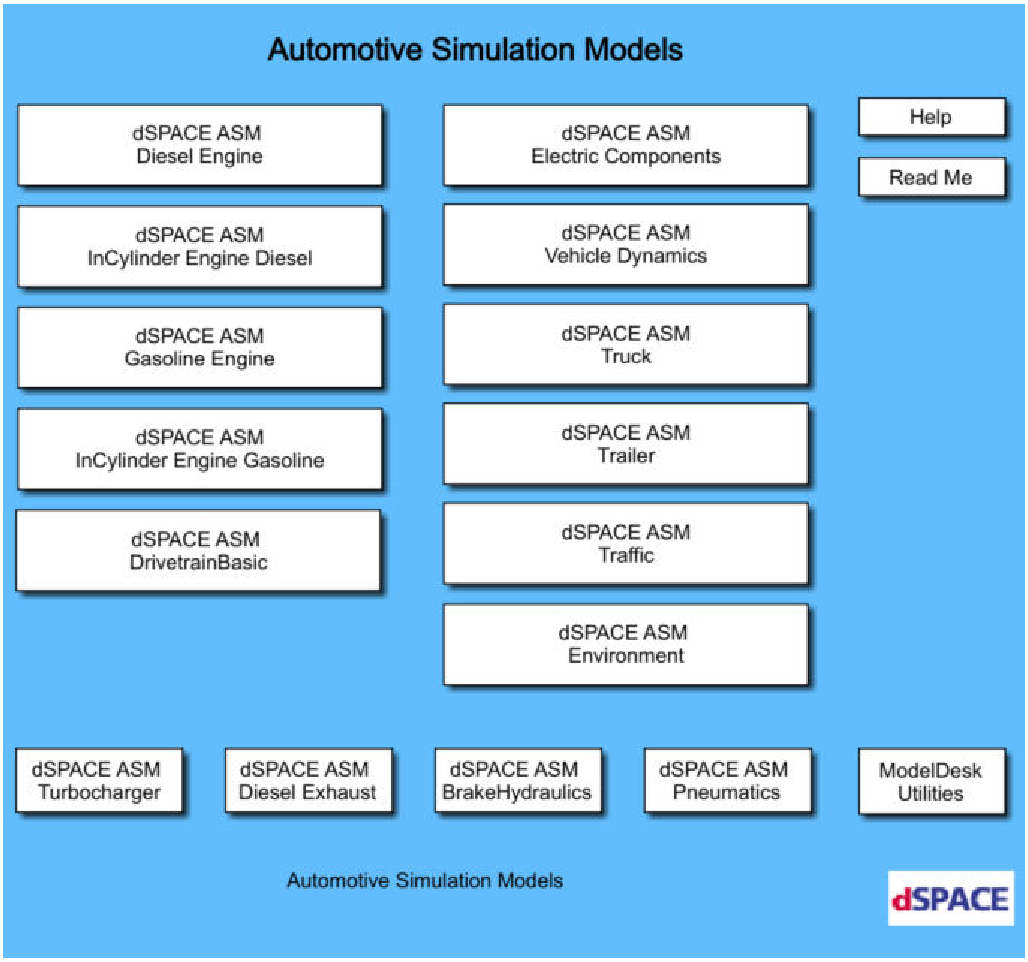
To open an ASM library

1. Start a MATLAB session.

   When MATLAB starts, the MATLAB Command Window lists all the ASM blocksets that are installed on the PC.

2 Type `asm` in your MATLAB Command Window.

The central ASM library opens.


Automotive Simulation Models
dSPACE ASM Diesel Engine
dSPACE ASM InCylinder Engine Diesel
dSPACE ASM Gasoline Engine
dSPACE ASM InCylinder Engine Gasoline
dSPACE ASM DrivetrainBasic
dSPACE ASM Electric Components
dSPACE ASM Vehicle Dynamics
dSPACE ASM Truck
dSPACE ASM Trailer
dSPACE ASM Traffic
dSPACE ASM Environment
Help
Read Me
dSPACE ASM Turbocharger
dSPACE ASM Diesel Exhaust
dSPACE ASM BrakeHydraulics
dSPACE ASM Pneumatics
ModelDesk Utilities
Automotive Simulation Models
dSPACE

It provides an overview of all the accessible major ASM libraries:

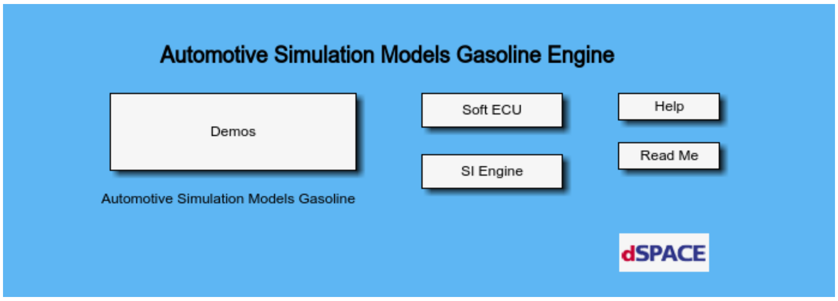
- ASM Diesel Engine
- ASM InCylinder Engine Diesel
- ASM InCylinder Engine Gasoline
- ASM Gasoline Engine
- ASM Drivetrain Basic
- ASM Vehicle Dynamics
- ASM Electric Components
- ASM Trailer
- ASM Traffic
- ASM Truck
- ASM Environment

Depending on your software license, some of the libraries are grayed out. If a library is grayed, you do not have a valid license for it.

3 Double-click the **dSPACE ASM Gasoline Engine** subsystem.

The subsystem opens. Depending on the installed license, you can open the developer or operator library. For details, refer to Operator Version (ASM User Guide 🕮).

4 Double-click the **Developer** or **Operator** subsystem to open the library.


Automotive Simulation Models Gasoline Engine
Demos
Automotive Simulation Models Gasoline
Soft ECU
SI Engine
Help
Read Me
dSPACE

Result

The library opens. It is subdivided into three sections.

- On the left is a subsystem with all the demo models belonging to the library.
- The middle section contains the subsystems that are part of the library and which are contained in the example model as links.
- On the right are links to the documentation and to the last-minute information in the readme section.

Related topics

HowTos

How to Start with an ASM Demo Model via MATLAB.. 15

How to Start with an ASM Demo Model via MATLAB

Objective

After successfully installing ASM on your PC, you can now easily start a demo model of your ASM installation via MATLAB.

This lesson explains how to start the Gasoline Engine model via MATLAB. You can start all ASM products in a similar way. The MATLAB Command Window shows you all the products that are installed and covered by your license.

Methods to start with an ASM demo model

You can start an ASM demo model with the following methods:

- Start an ASM demo model via MATLAB, see below.
- Generate an ASM project by using ModelDesk or by using MATLAB/Simulink. Refer to How to Create a Project Based on an ASM Demo (ModelDesk Project and Experiment Management 🕮).

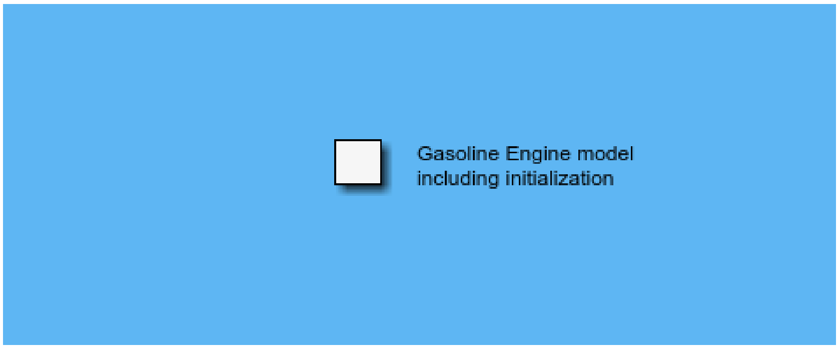
Preconditions

The ASM library is open. Refer to How to Open an ASM Library on page 13.

Method

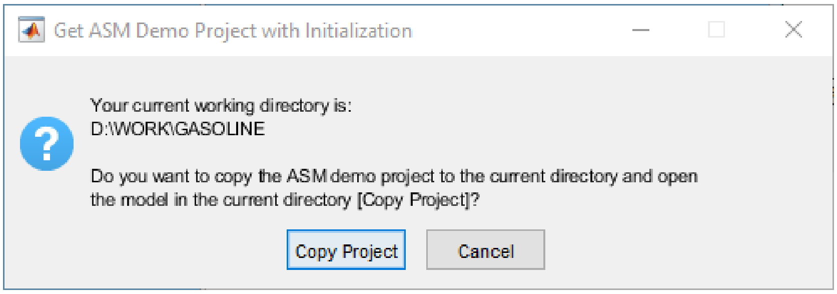
To start with an ASM demo model via MATLAB

1 Double-click the Demos subsystem to open a new window which presents all the demo models that are part of the library.



2 Double-click the demo model to start it.

This opens a dialog that offers you the possibility to copy the project to your current working directory.



Your current working directory is:
D:\WORK\GASOLINE

Do you want to copy the ASM demo project to the current directory and open the model in the current directory [Copy Project]?

3 Click Copy Project to copy the current demo folder to your current MATLAB work folder.

All the files related to the demo model are copied to the current MATLAB work folder. This includes the entire parameterization and the ControlDesk example layouts.

Result

Now the demo model opens. All the initialization files are executed and you can start simulation.

Related topics

HowTos

How to Create a Project Based on an ASM Demo (ModelDesk Project and Experiment Management 🕮)

How to Start with an ASM Demo Model Using ModelDesk

Objective

After successfully installing ASM on your PC, you can now easily start a demo model of your ASM installation using ModelDesk.

Methods to start with an ASM demo model

You can start an ASM demo model with the following methods:

- Generate an ASM project by using ModelDesk or by using MATLAB/Simulink, see below.
- Start an ASM demo model via MATLAB. Refer to How to Start with an ASM Demo Model via MATLAB on page 15.

ASM demos

The ASM installation contains several ASM demos. The ASM demos include all the necessary files for the simulation, for example:

- Simulation model based on the ASM blocks
- Simulation applications for the simulation platforms
- ModelDesk project for parameterizing the model
- ControlDesk project for experimenting with the model
- MotionDesk project for animation (if useful)

Preconditions

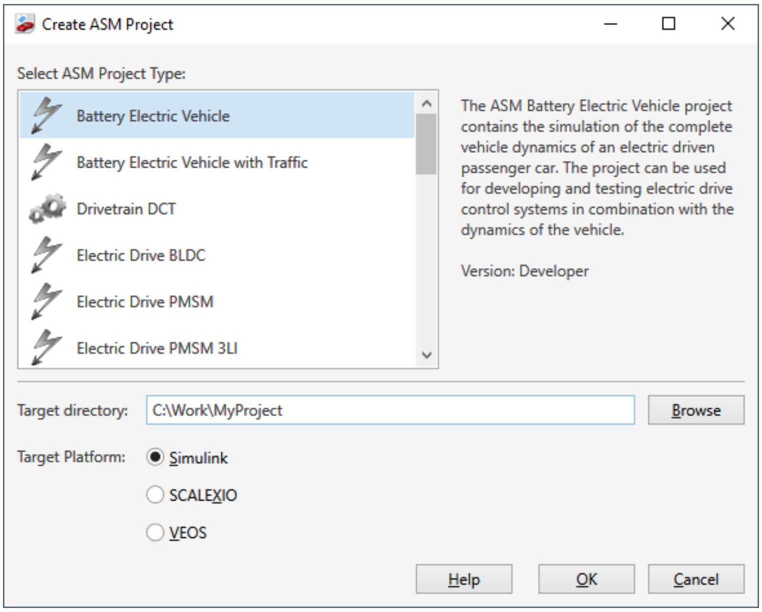
- No other project must be open.
- The license of the ASM library that is used by the ASM demos must be available.
- The ASM library must be decrypted.

Method

To to start with an ASM demo model using ModelDesk

1 On the File ribbon, click New – ASM Project.

The Define a Project dialog opens.



The ASM Battery Electric Vehicle project contains the simulation of the complete vehicle dynamics of an electric driven passenger car. The project can be used for developing and testing electric drive control systems in combination with the dynamics of the vehicle.

Version: Developer

Target directory: C:\Work\MyProject

Target Platform: ● Simulink ○ SCALEXIO ○ VEOS

2 Select an ASM project type.

3 Specify the target directory. Select an empty directory or specify a new directory. You must have write permission to the directory.

4 Select the target platform to be activated in the experiment.

5 Click OK.

Result

ModelDesk copies all files of the selected ASM demo to the specified target folder and opens the project.

How to Open a Parameterization Project

Objective

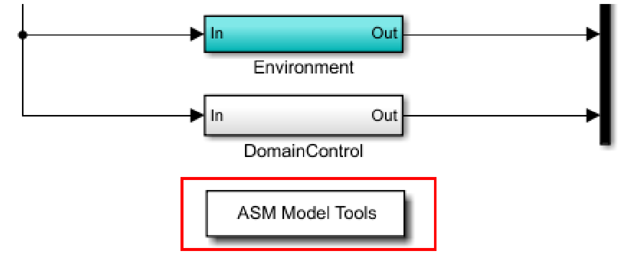
You can open a parameterization project from the model.

The parameterization project for the current model is accessible via the Open ModelDesk Project & Experiment button.

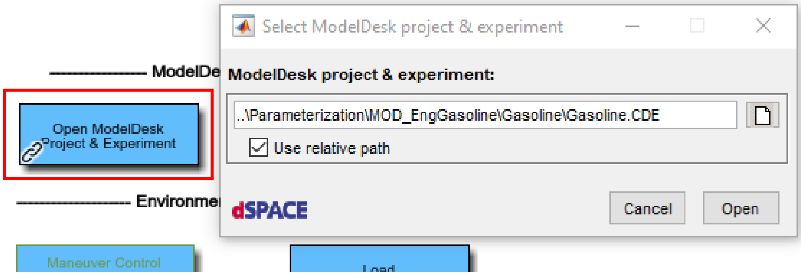
Method

To open a parameterization project

1 In the ASM model, double-click the ASM Model Tools button below the modules.



2 Double-click the Open ModelDesk Project & Experiment button.


ModelDe
Open ModelDesk Project & Experiment
Environme
Maneuver Control
Load
Select ModelDesk project & experiment
dSPACE
Cancel
Open

ModelDesk project & experiment:

..\Parameterization\MOD_EngGasoline\Gasoline\Gasoline.CDE

☑ Use relative path

3 In the dialog, click Open to open the selected ModelDesk project.

Result

ModelDesk is started and the selected project and experiment is loaded.

How to Parameterize ASM via ModelDesk

Objective

The `<Projectfolder>\Parameterization` folder includes a complete ModelDesk project. You can perform the complete parameterization of the ASM model with ModelDesk.

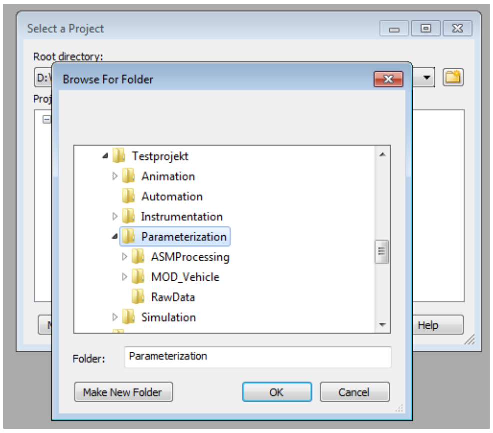
In the following method, an ASM Electric Components demo is used as an example.

Method

To parameterize ASM via ModelDesk

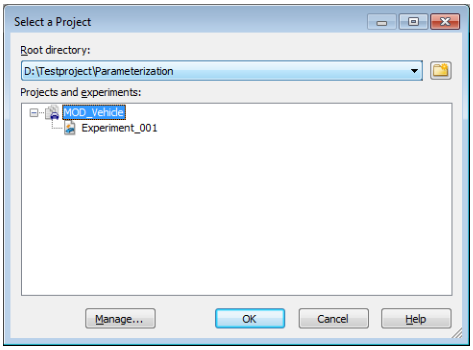
1 Start ModelDesk and open the project from `<Projectfolder>\Parameterization`.

1. Click File – Open – Project + Experiment.
2. Define the root directory in `<Projectfolder>\Parameterization`.



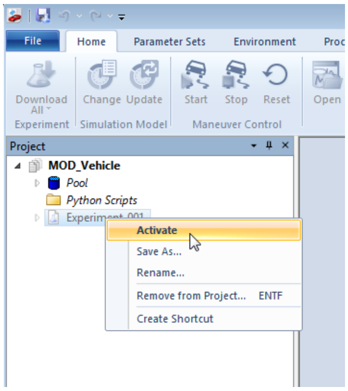
Folder: Parameterization

3. Select the project.



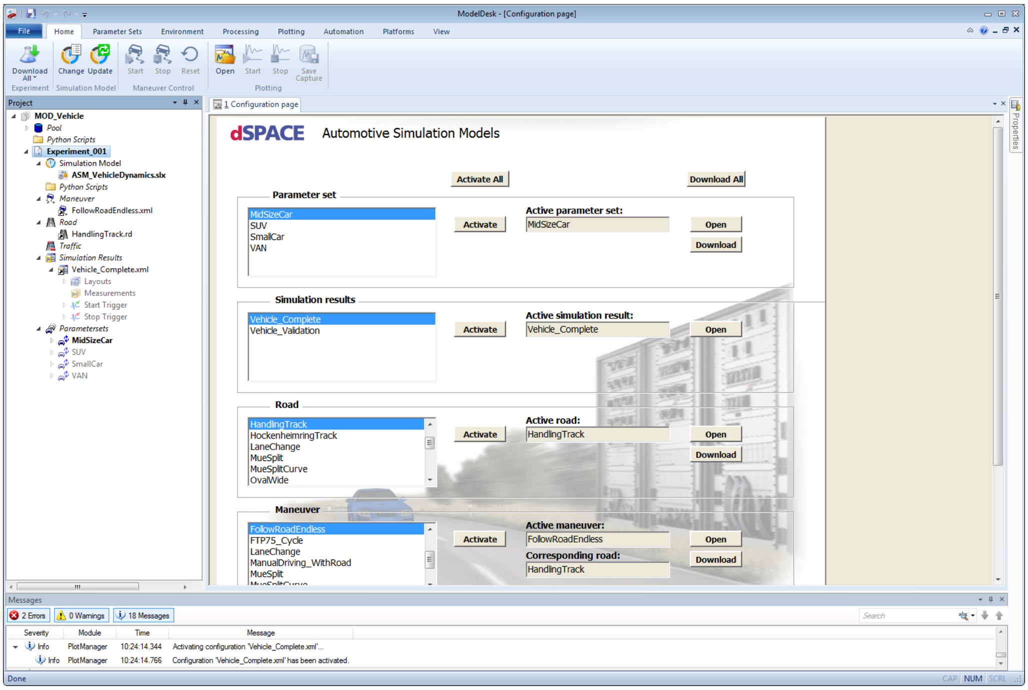
Root directory: D:\Testproject\Parameterization

2 Activate the demo experiment via context menu on the experiment in the Project Navigator.



The project structure is available in the Project Navigator and all parameters are available for configuration.

3 Switch to the experiment's **Configuration** page.



Parameter set: MidSizeCar, SUV, SmallCar, VAN

Active parameter set: MidSizeCar

Simulation results: Vehicle_Complete, Vehicle_Validation

Active simulation result: Vehicle_Complete

Road: HandlingTrack, HockenheimringTrack, LaneChange, MueSplit, MueSplitCurve, OvalWide

Active road: HandlingTrack

Maneuver: FollowRoadEndless, FTP75_Cycle, LaneChange, ManualDriving_WithRoad, MueSplit, MueSplitCurve

Active maneuver: FollowRoadEndless

Corresponding road: HandlingTrack

Messages

2 Errors 0 Warnings 18 Messages

Severity	Module	Time	Message
Info	PlotManager	10:24:14.344	Activating configuration 'Vehicle_Complete.xml'...
Info	PlotManager	10:24:14.766	Configuration 'Vehicle_Complete.xml' has been activated.

Done

The **Configuration** page of the experiment lets you select parameter sets for activating and downloading them to the simulation model or for activating and editing (**Open** button) them.

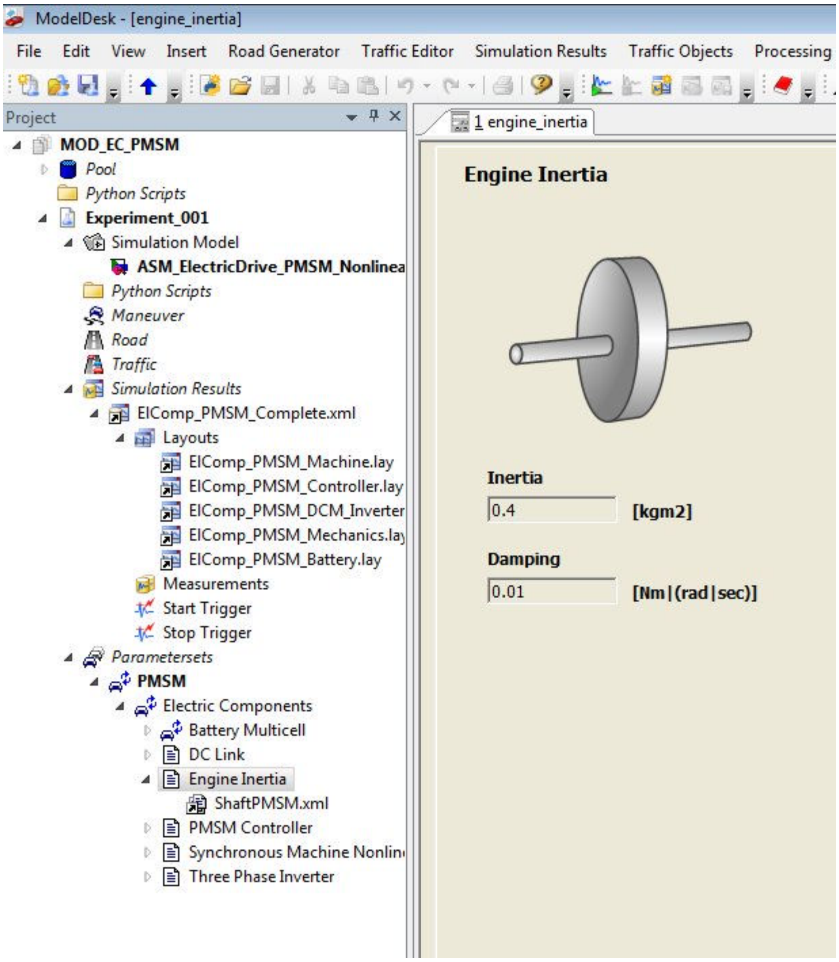
You can also activate plotting sets and open them to obtain simulation results in ModelDesk via the ModelDesk Plotting feature.

4 Click **Activate** and then **Download** to activate and download the parameter set.

5 Start a simulation (if possible in accelerator mode) and observe the results in the Simulink plotters, and/or ModelDesk plotting.

6 Modify, for example, the engine inertia in ModelDesk.

In the Project Navigator, select Parametersets – <ParameterSet name> – ElectricComponents – Engine Inertia.



Inertia: 0.4 [kgm2]

Damping: 0.01 [Nm|(rad|sec)]

7 Use the context menu of the current page to save and download the new setting or switch back to the main experiment page and download the current parameter setting.

8 Start a simulation (if possible in accelerator mode) and observe the results in the Simulink plotters, ModelDesk plotting, and/or MotionDesk.

9 Switch to the MATLAB workspace and look at the value of the engine inertia, `MDL.ElectricComponents.Mechanics.EngineInertia.Const_Inertia`. If you are working with the Simulink simulation, all parameters are available in the structure `MDL` in the MATLAB workspace and ModelDesk can download new values to the `MDL` structure.

```
>> MDL.ElectricComponents.Mechanics.EngineInertia.Const_Inertia
ans =
     Comment: 'Inertia'
      Author: '(null)'
      Origin: ' '
     Version: '2'
LastModified: '16.01.2015 11:21'
       vName: 'Const_Inertia'
       vUnit: '[kgm2]'
           v: [0.4000]
```

Result

You parameterized the model in ModelDesk.

Related topics

Basics

Working with Parameter Sets (ModelDesk Parameterizing 🕮)

References

Settings (ModelDesk Road Creation 🕮)

How to Plot Simulation Results in ModelDesk

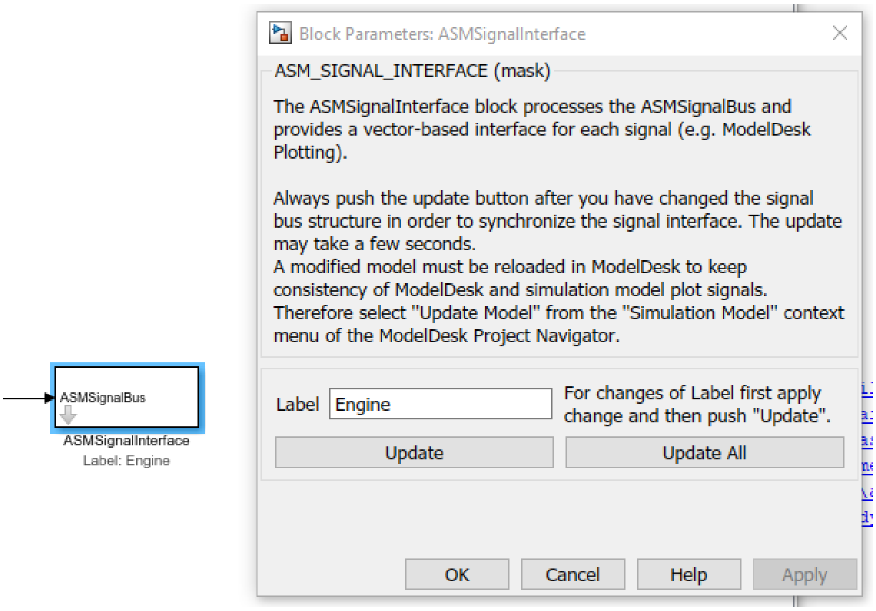
Objective

You can plot simulation results in ModelDesk via the SignalInterfaces.

Description

You can use the ASMSignalInterface block and the ASMSignalInterface Manager to toggle the plotting.

ASMSignalInterface block Simulation data is transferred from the ASM model to ModelDesk via the ASMSignalInterface block. There is one block inside each ASM module. Refer to ASMSignalInterface (ASM User Guide 🕮).



ASM_SIGNAL_INTERFACE (mask)

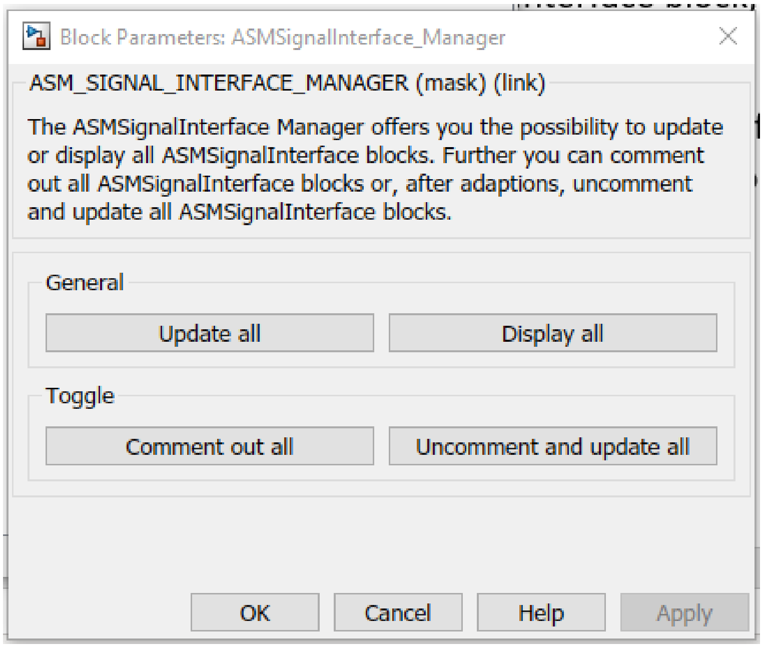
The ASMSignalInterface block processes the ASMSignalBus and provides a vector-based interface for each signal (e.g. ModelDesk Plotting).

Always push the update button after you have changed the signal bus structure in order to synchronize the signal interface. The update may take a few seconds.
A modified model must be reloaded in ModelDesk to keep consistency of ModelDesk and simulation model plot signals.
Therefore select "Update Model" from the "Simulation Model" context menu of the ModelDesk Project Navigator.

Label: Engine

For changes of Label first apply change and then push "Update".

ASMSignal Interface Manager The ASMSignalInterface Manager collects the signals of all modules of your model. You can open it via the ASM Tools. Refer to ASMSignalInterface Manager (ASM User Guide 📖).



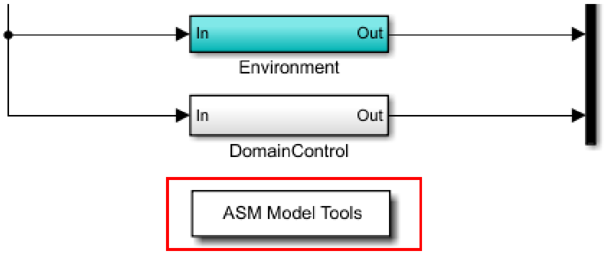
ASM_SIGNAL_INTERFACE_MANAGER (mask) (link)

The ASMSignalInterface Manager offers you the possibility to update or display all ASMSignalInterface blocks. Further you can comment out all ASMSignalInterface blocks or, after adaptions, uncomment and update all ASMSignalInterface blocks.

Method

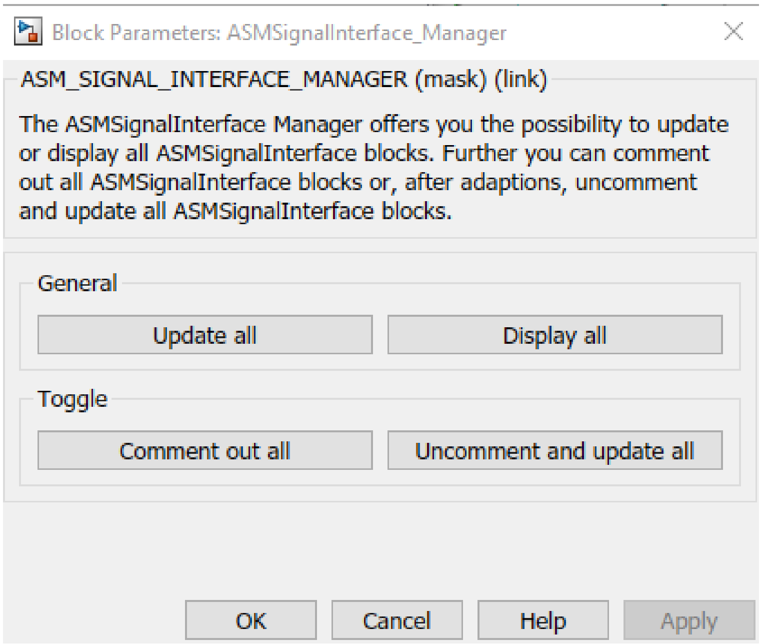
To plot simulation results in ModelDesk

1 In the ASM model, double-click the ASM Model Tools button below the modules.



2 Double-click the ASMSignalInterface Manager button and press one of the two buttons:

- Uncomment and update all: for plotting in ModelDesk.
- Comment out all: if you do not need plotting in ModelDesk. Commenting out the blocks will improve the simulation performance.


Block Parameters: ASMSignalInterface_Manager

ASM_SIGNAL_INTERFACE_MANAGER (mask) (link)

The ASMSignalInterface Manager offers you the possibility to update or display all ASMSignalInterface blocks. Further you can comment out all ASMSignalInterface blocks or, after adaptions, uncomment and update all ASMSignalInterface blocks.

Result

You activated or deactivated plotting in ModelDesk.

Related topics

HowTos

How to Configure a Plotter in ModelDesk for a Simulink Simulation..34
How to Update a Plot Configuration for ModelDesk for a Simulink Simulation........................32

How to Switch the Library Type

Objective

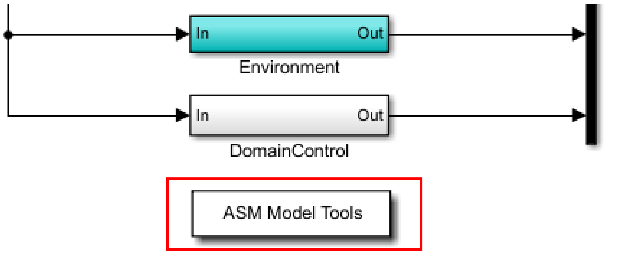
You can switch between the developer library and the operator library.

The developer library is a Simulink library containing the open-source ASM blocks. In the operator library, the ASM blocks are Simulink S-functions for offline simulation.

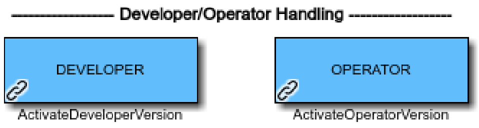
Method

To switch the library type

1 In the ASM model, double-click the ASM Model Tools button below the modules.



2 Depending on the library type, you want to switch to, double-click the DEVELOPER button or the OPERATOR button.



Result

You switched between the developer library and the operator library.

Related topics

References

Activate Developer Version (ASM User Guide 📖)
Activate Operator Version (ASM User Guide 📖)

How to Handle Multi-Instances

Objective

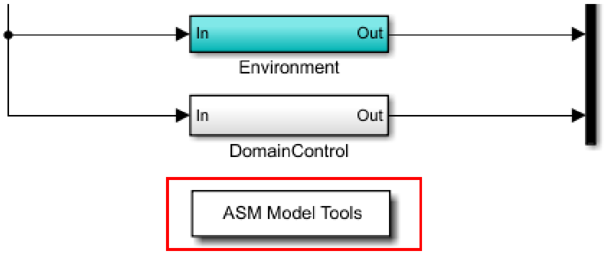
The Multi-Instance Overview button opens a dialog to get an overview of all multi-instance blocks in the model and to set the multi-instance parameters.

For more information on multi-instances, refer to Multi-Instance (ASM User Guide 📖).

Method

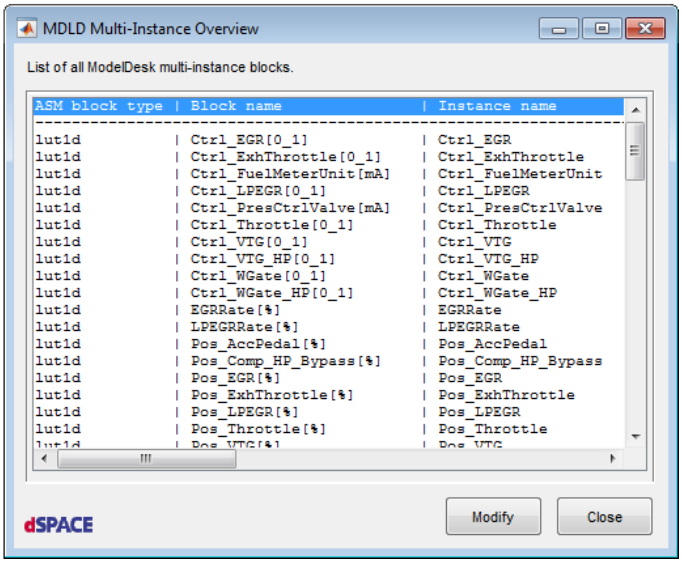
To handle multi-instances

1 In the ASM model, double-click the ASM Model Tools button below the modules.



2 Double-click the Multi-Instance Overview button.

A dialog opens that displays the multi-instances in your model.



List of all ModelDesk multi-instance blocks.

```
ASM block type | Block name                    | Instance name
-------------------------------------------------------------------
lut1d          | Ctrl_EGR[0_1]                 | Ctrl_EGR
lut1d          | Ctrl_ExhThrottle[0_1]         | Ctrl_ExhThrottle
lut1d          | Ctrl_FuelMeterUnit[mA]        | Ctrl_FuelMeterUnit
lut1d          | Ctrl_LPEGR[0_1]               | Ctrl_LPEGR
lut1d          | Ctrl_PresCtrlValve[mA]        | Ctrl_PresCtrlValve
lut1d          | Ctrl_Throttle[0_1]            | Ctrl_Throttle
lut1d          | Ctrl_VTG[0_1]                 | Ctrl_VTG
lut1d          | Ctrl_VTG_HP[0_1]              | Ctrl_VTG_HP
lut1d          | Ctrl_WGate[0_1]               | Ctrl_WGate
lut1d          | Ctrl_WGate_HP[0_1]            | Ctrl_WGate_HP
lut1d          | EGRRate[%]                    | EGRRate
lut1d          | LPEGRRate[%]                  | LPEGRRate
lut1d          | Pos_AccPedal[%]               | Pos_AccPedal
lut1d          | Pos_Comp_HP_Bypass[%]         | Pos_Comp_HP_Bypass
lut1d          | Pos_EGR[%]                    | Pos_EGR
lut1d          | Pos_ExhThrottle[%]            | Pos_ExhThrottle
lut1d          | Pos_LPEGR[%]                  | Pos_LPEGR
lut1d          | Pos_Throttle[%]               | Pos_Throttle
lut1d          | Pos_VTG[%]                    | Pos_VTG
```

3 To modify a multi-instance, select it in the dialog and click Modify.

Result

You modified multi-instances.

Related topics

Basics

Multi-Instance (ASM User Guide 🕮)

How to Display the ModelDesk Parameter Group

Objective

You can modify the grouping in the parameter set that is displayed in the project tree of your ModelDesk experiment.

For more information, refer to Basics on Grouping the View of a Parameter Set (ModelDesk Parameterizing 🕮).

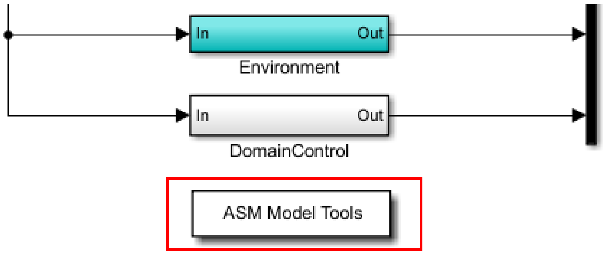
The ModelDesk Parameter Group Overview block in the ASM Tools opens an overview of the ModelDesk parameter group blocks that are contained in the simulation model.

For more information, refer to ModelDesk Parameter Group (ASM User Guide 🕮).

Method

To display the ModelDesk parameter group

1 In the ASM model, double-click the ASM Model Tools button below the modules.



2 Double-click the ModelDesk Parameter Group Overview button.

Result

The MDLD Parameter Group Overview dialog opens. It displays all ModelDesk parameter groups.

Select a group in the list and click Open System to open the group in your model.

Related topics

Basics

Basics on Grouping the View of a Parameter Set (ModelDesk Parameterizing 🕮)
ModelDesk Parameter Group (ASM User Guide 🕮)

Simulating in Simulink

Where to go from here

Information in this section

How to Use the Simulink Accelerator Mode

Objective

You can use the Simulink accelerator mode to speed up the simulation (Pentium 4 Processor with 2 GB RAM: up to 2.5 times faster than real time).

Note

You can perform all steps in this tutorial by using the accelerator mode.

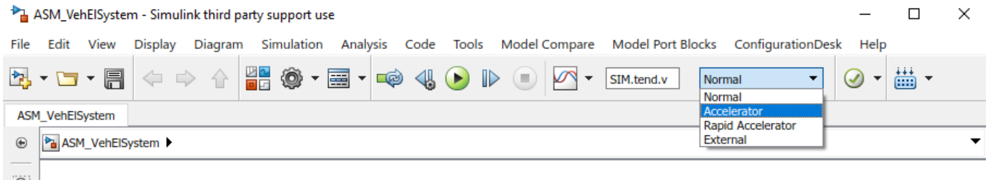
Prerequisite

You need a license for the Simulink Performance Tools.

Method

To use the Simulink Accelerator mode

1 In Simulink, activate the accelerator mode for the ASM demo model on the menu bar.


ASM_VehElSystem - Simulink third party support use
File Edit View Display Diagram Simulation Analysis Code Tools Model Compare Model Port Blocks ConfigurationDesk Help
SIM.tend.v
Normal
Normal
Accelerator
Rapid Accelerator
External
ASM_VehElSystem
ASM_VehElSystem

2 Start the simulation and observe the results on the Simulink plotters.

Result

The model is simulated in accelerator mode.

First, the build process for the accelerator mode is performed, which takes some time.

Whenever you perform an accelerator mode simulation and the model structure has not changed, no new code is generated, even if new parameters for the model are loaded, for example, from ModelDesk.

How to Simulate the Model and Observe the Results in Simulink

Starting simulation

The demo model is prepared for a Simulink simulation. When you start simulation after you opened the demo model, a standard vehicle dynamics maneuver or engine driving cycle starts.

Simulation results

You can observe the simulation results in Simulink using Simulink scopes. The scopes are in the UserInterface/DISP_xyz subsystem of an ASM module. For example, you can find a scope in the Engine/Control/UserInterface/DISP_Control subsystem.

You can also watch the simulation results in oversampled model parts. For more information, refer to Watching the Simulation Results in Oversampled Model Parts (ASM Gasoline Engine InCylinder Model Description 🕮) or Watching the Simulation Results in Oversampled Model Parts (ASM Diesel Engine InCylinder Model Description 🕮).

Method

To simulate the model and observe the results in Simulink

1 In MATLAB, switch to `<Projectfolder>\Simulation` in the project and call `go` to open the model.

2 Start the simulation.

3 Open one of the scopes from the UserInterface subsystem, for example, in Engine/Control/UserInterface/DISP_Control. After a few seconds, look at the scope signals.

Result

You simulated the model and observed the results in Simulink.

How to Handle Simulink Plotters

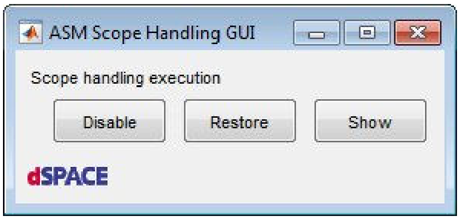
Objective

You can handle the Simulink plotters in your simulation model.

Handling Simulink Plotters

The SCOPE HANDLING GUI button opens a dialog that lets you perform the following actions:

- Disable: Comments out all Simulink plotters, displays, and xy-graphs in the model.
- Restore: Uncomments all Simulink plotters, displays, and xy-graphs in the model.
- Show: Shows a list with all the Simulink plotters, displays, and xy-graphs in the model.



The block makes directly use of the *commented* block property in Simulink. Disabled blocks are excluded from simulation and signals are terminated and grounded.

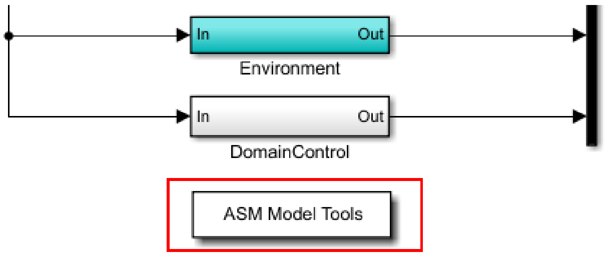
For ASM Engine Models: If you want to simulate your model with `asm_eng_testbench.m`, it is sufficient to store the simulation results in the variables under TEST_CONTROL or TEST_PLANT. No plotters are necessary.

The testbench script contains a postprocessing.

Method

To handle Simulink plotters

1 In the ASM model, double-click the ASM Model Tools button below the modules.



2 Double-click the SCOPE HANDLING GUI button.

The ASM Scope Handling GUI dialog opens.

3 In the ASM Scope Handling GUI dialog, click one of the buttons to handle Simulink plotters.

For information on the options, refer to Handling Simulink Plotters on page 31 above.

Result

You handled the Simulink plotters in your simulation model.

Related topics

References

Scope Handling User Interface (ASM User Guide 🕮)

How to Update a Plot Configuration for ModelDesk for a Simulink Simulation

Objective

The ASM package includes a signal interface block for ModelDesk so that signals of the ASMSignalBus can be plotted. This block must be updated manually if the ASMSignalBus is changed.

If the respective simulation model is an operator version, it is mandatory to perform a manual update of the plotting block and plotting configuration, before using the plotting feature.

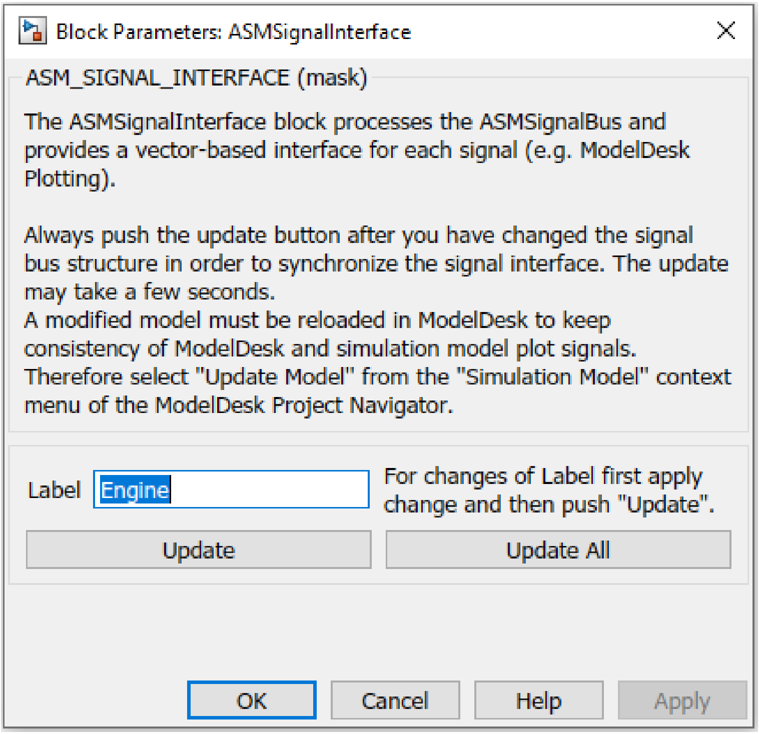
Method

To update a plot configuration for ModelDesk for a Simulink simulation

1 Inside an ASM module, open the SignalInterface subsystem.

2 Double-click the ASMSignalInterface block.

The following dialog appears with varying labels, depending on your model.



Block Parameters: ASMSignalInterface

ASM_SIGNAL_INTERFACE (mask)

The ASMSignalInterface block processes the ASMSignalBus and provides a vector-based interface for each signal (e.g. ModelDesk Plotting).

Always push the update button after you have changed the signal bus structure in order to synchronize the signal interface. The update may take a few seconds.
A modified model must be reloaded in ModelDesk to keep consistency of ModelDesk and simulation model plot signals.
Therefore select "Update Model" from the "Simulation Model" context menu of the ModelDesk Project Navigator.

Label: Engine

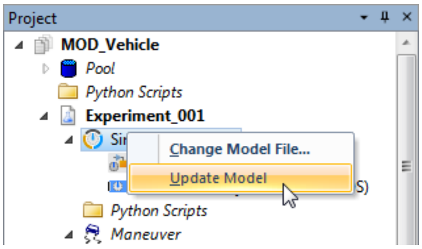
For changes of Label first apply change and then push "Update".

3 Click:

- Update: To update only the current block.
- Update all: To update all blocks in the model that are currently not commented out.

All the ASMSignalInterface blocks are updated. This also changes the model, which you must update in ModelDesk.

4 In ModelDesk's Project Manager, right-click the Simulation model element and select Update Model.



Result

The plot configuration is updated.

Related topics

Basics

Collecting Signals for Plotting in the ASM Model (ModelDesk Plotting 🕮)

References

Signal Selector (ModelDesk Plotting 🕮)

How to Configure a Plotter in ModelDesk for a Simulink Simulation

Objective

To plot a signal from the ASMSignalBus, you have to configure a project and an experiment in ModelDesk.

The `<Projectfolder>\Parameterization\MOD_<Blockset>` folder includes a complete ModelDesk project. You can completely parameterize the engine in ModelDesk. The demo project includes examples of typical plotter signals.

Note

For each simulation step, plotting data is transfered from the simulation platform to ModelDesk. If the model is simulated on real-time hardware with a small sample time, a huge amount of data must be handled. This can lead to a plotting error with the data acquisition being stopped. To avoid this, decrease the number of plotted signals.

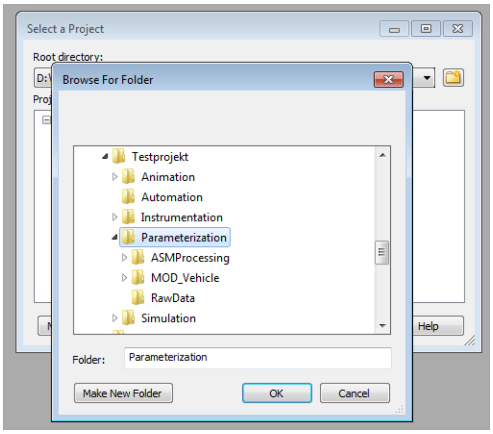
Method

To configure a plotter in ModelDesk for a Simulink simulation

1 Start ModelDesk and open the provided ModelDesk project from `<Projectfolder>\Parameterization`.

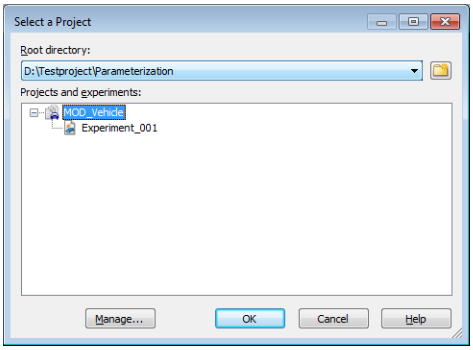
1. Click File – Open - Project + Experiment.

2. Select the `<Projectfolder>/Parameterization` folder.



Folder: Parameterization

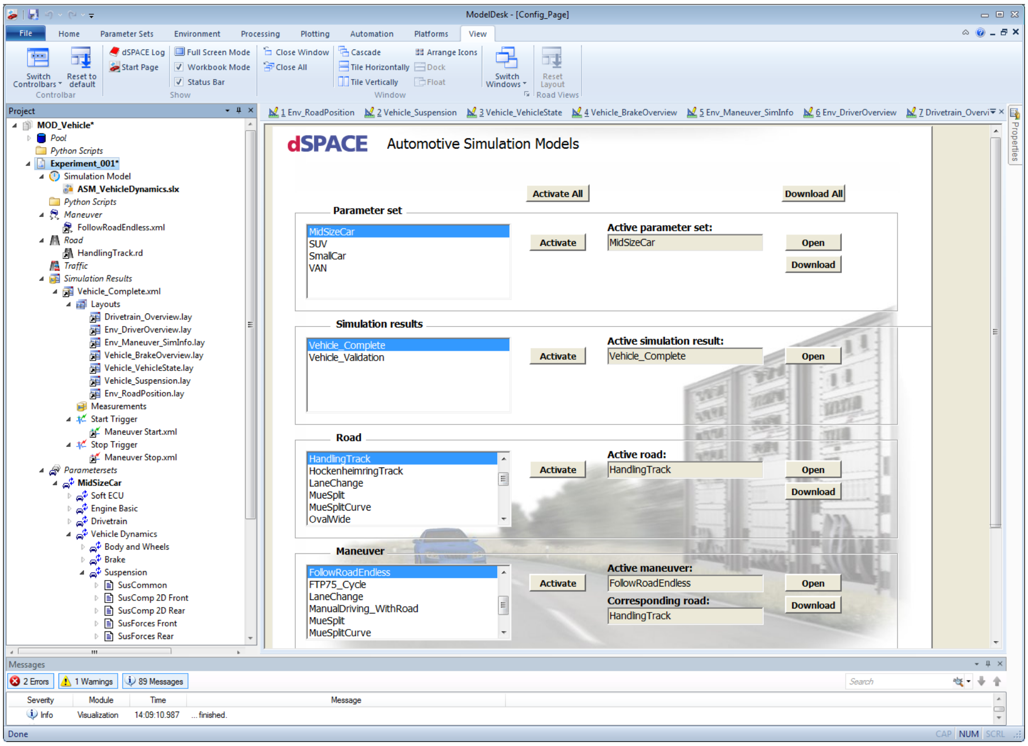
3. Select an experiment from your ModelDesk folder.



Root directory: D:\Testproject\Parameterization

2 Switch to the experiment's Configuration page.

3 Select a plot configuration from the Simulation results list, click Activate and Open.



Active parameter set: MidSizeCar

Active simulation result: Vehicle_Complete

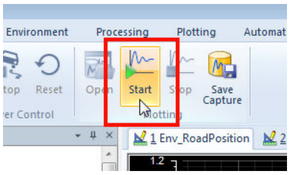
Active road: HandlingTrack

Active maneuver: FollowRoadEndless

Corresponding road: HandlingTrack

Severity	Module	Time	Message
Info	Visualization	14:09:10.987	...finished.

4 To start a simulation, go to the Home ribbon and click Plotting – Start (if possible in Simulink Accelerator mode).



Result

ModelDesk starts the simulation and plots the selected signals. You can observe the results in the ModelDesk plotters.

Related topics

Basics

Plotting Signals (ModelDesk Plotting 📖)

Simulating on dSPACE Platforms

Where to go from here

Information in this section

Model Preparation for Real-Time Simulation (PHS-Bus-Based Platforms)

Model preparation

Before you use the model for real-time simulation, you have to prepare the model. To avoid the real-time process from being stopped by task overruns during the initialization phase of the model, it is recommended to allow a number of queued task calls before the simulation is stopped.

Note

If you use the Operator version of the ASM, you cannot generate code for real-time simulation.

Method

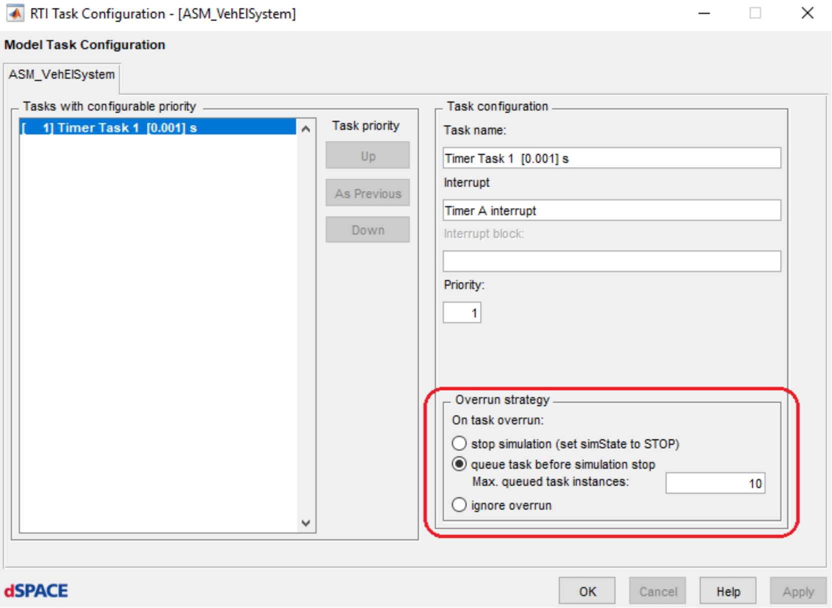
To avoid stopping the real-time process on a singular task overrun

1 In Simulink, select Code – C/C++ Code – Code Generation Options to open the Configuration Parameters dialog.

2 In the Configuration Parameters dialog, select RTI simulation options from the Select tree.

3 Click Task Configuration.

After a few seconds the RTI Task Configuration dialog opens.



RTI Task Configuration - [ASM_VehElSystem]

Model Task Configuration

ASM_VehElSystem

Tasks with configurable priority

[1] Timer Task 1 [0.001] s

Task configuration

Task name: Timer Task 1 [0.001] s

Interrupt: Timer A interrupt

Interrupt block:

Priority: 1

Overrun strategy

On task overrun:

○ stop simulation (set simState to STOP)

● queue task before simulation stop

Max. queued task instances: 10

○ ignore overrun

4 In this dialog, select queue task before simulation stop as the overrun strategy.

5 In Max. queued task instances, enter `10` and click OK.

This setting has no appreciable effect on your normal task overrun behavior, because only a few singular task overruns (for example, 10) are accepted.

How to Generate an OSA File for VEOS

Objective

You can generate an Offline Simulation Application (OSA) for VEOS either manually or automatically.

Possible methods

There are two ways to generate code for VEOS:

- Manually. Refer to Method 1 on page 39.
- Automatically. Refer to Method 2 on page 39.

Method 1

To generate code for VEOS manually

1 Open the model with `go ('simmode','CPT','platform','VEOS')`. This selects the system target `dsrt.tlc` file provided by the Model Interface Package for Simulink.

2 Generate code for the ASM model.

The Model Interface Package for Simulink generates a Simulink implementation container (SIC) file containing ASM model code.

For instructions, refer to Generating Simulink Implementation Containers (Model Interface Package for Simulink - Modeling Guide 🕮).

3 Create a new OSA file:

1. Start the VEOS Player.
2. On the **Home** ribbon, select **New** to create an empty OSA file.

For instructions, refer to Basics on the VEOS Player (VEOS Manual 🕮).

4 Import the SIC file to the VEOS Player to integrate the model in an Simulink simulation application for VEOS.

The VEOS Player builds the OSA for simulation on VEOS.

> **Note**
>
> ASM does not support a 32 bit version of VEOS. Make sure to select **HostPC64** as simulation target on the **Build Options** tab during import to the VEOS Player.
> You can also set **HostPC64** as default simulation target via the Simulation Target Manager in VEOS.

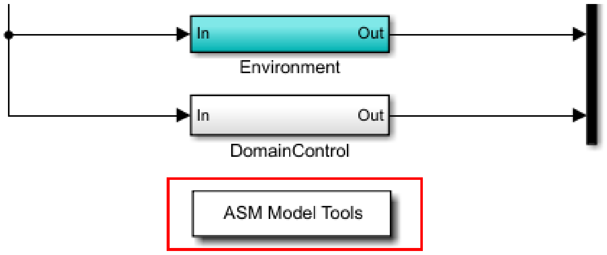
For instructions, refer to How to Import Simulink Implementations (VEOS Manual 🕮).

Method 2

To automate code generation for VEOS

1 Open the model in MATLAB.

2 In the ASM model, double-click the **ASM Model Tools** button below the modules.



3 Double-click the **Generate VEOS Application** button.

4 A **Code Generation** confirmation prompt opens. Click **Yes** to start code generation.

Result

You have generated code for VEOS.

Related topics

References

Generate VEOS (ASM User Guide 🕮)

Generating the Real-Time Application for SCALEXIO

Generating code for SCALEXIO

Generation of the real-time application has to be started from ConfigurationDesk. For information on how to set up a ConfigurationDesk project and start code generation, refer to Building Real-Time Applications (ConfigurationDesk Real-Time Implementation Guide 🕮).

How to Generate a Real-Time Application for PHS-Bus Based Platforms

Objective

If you have a dSPACE RTI (Real-Time Interface) license, you can generate new code of the ASM model for real-time applications. This is needed, if you want to modify the model, for example, after you have added I/O blocks for doing HIL simulation, or if you want different initial settings for the model (settings of your vehicle).

If you have a dSPACE VEOS license, you can generate new code of the ASM model for a VEOS platform (offline simulation application).

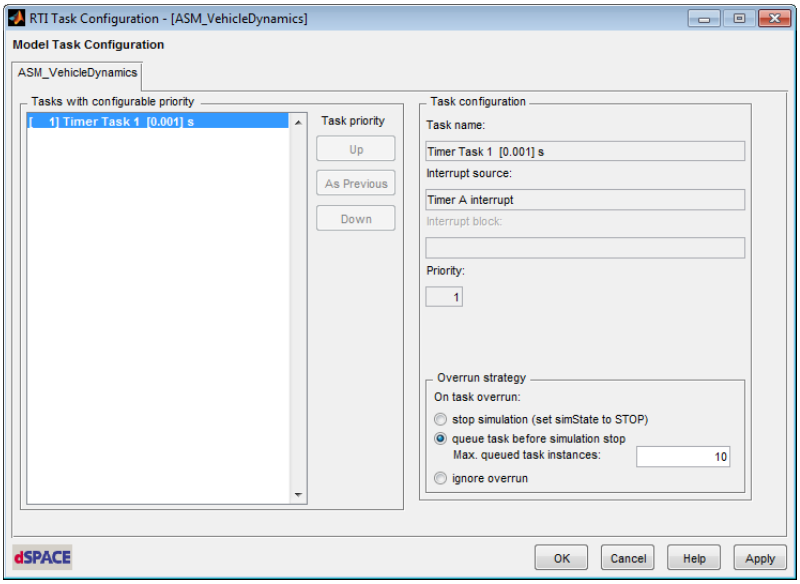
Method

To generate a real-time application for the PHS bus based platforms

1 Start MATLAB, switch to `<Projectfolder>\Simulation` and call **go**. If prompted, select the automatic setting of the preferences. The demo model opens with the initial default parameters specified in the **go** file.

2 From the menu bar, choose Simulation – Model Configuration Parameters or press **`Ctrl+E`** to open the Configuration Parameters dialog.

3 Open the RTI Simulation Options page and click Task Configuration. In the Task Configuration dialog, set the overrun strategy of timer task 1 to queue task before simulation stop and the number of maximum task instances to "10". This setting has to be made, because the initialization of the model on the real-time hardware takes some time, so overruns will occur at the beginning of the simulation. A maximum of 10 permitted overruns will prevent the simulation from stopping during initialization.



[1] Timer Task 1 [0.001] s

Task name: Timer Task 1 [0.001] s

Interrupt source: Timer A interrupt

Interrupt block:

Priority: 1

On task overrun:

○ stop simulation (set simState to STOP)

● queue task before simulation stop

Max. queued task instances: 10

○ ignore overrun

4 To start code generation, click Build Model on the toolbar or press **`Ctrl+B`**.

Result

The complete build process will take some minutes. After the build process is finished, the new real-time application is downloaded to the dSPACE platform. The initial parameters are the parameters which were loaded from the **go** file.

The new real-time application can be used in the same way as the original demo application.

Related topics

References

RTI Task Configuration Dialog (RTI and RTI-MP Implementation Reference 📖)

How to Observe Results in ControlDesk

Objective

When simulating the model, you can observe the results in ControlDesk.

The ASM demo model contains preconfiguered real-time application files for the SCALEXIO and VEOS platforms. You can use these files to observe simulation results with the preconfigured ControlDesk experiment.

Note

The Electric Machine systems need a low simulation step size (e.g. 70µs) for a stable simulation. If measuring in ControlDesk is started and the Time Plotter is used, the data captured by the real-time hardware might be too fast for the PC. This can result in a delayed update of the standard ControlDesk instruments, for example, Display and Bar. To solve this issue, increase the downsampling in the ControlDesk Measurement Configuration controlbar or capture fewer signals to decrease the amount of data.

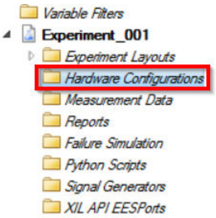
Method

To observe results in ControlDesk

1 Start ControlDesk.

On the ribbon, click File – Open – Open Project and Experiment to open the preconfigured ControlDesk experiment. Select `<Projectfolder>\Instrumentation\CD_<Blockset>\CD_<Blockset>.CDP`.

2 In ControlDesk, right-click Hardware Configurations in the project tree. On the context menu, select Add Platform/Device to add the platform.



Select your simulation platform and click Next. Import the prebuilt real-time application from `<Projectfolder>\Simulation\RealTimeObjects\<platform>\ASM_<Blockset>.sdf`.

3 On the Platforms ribbon, select Register Platforms to register your platform.

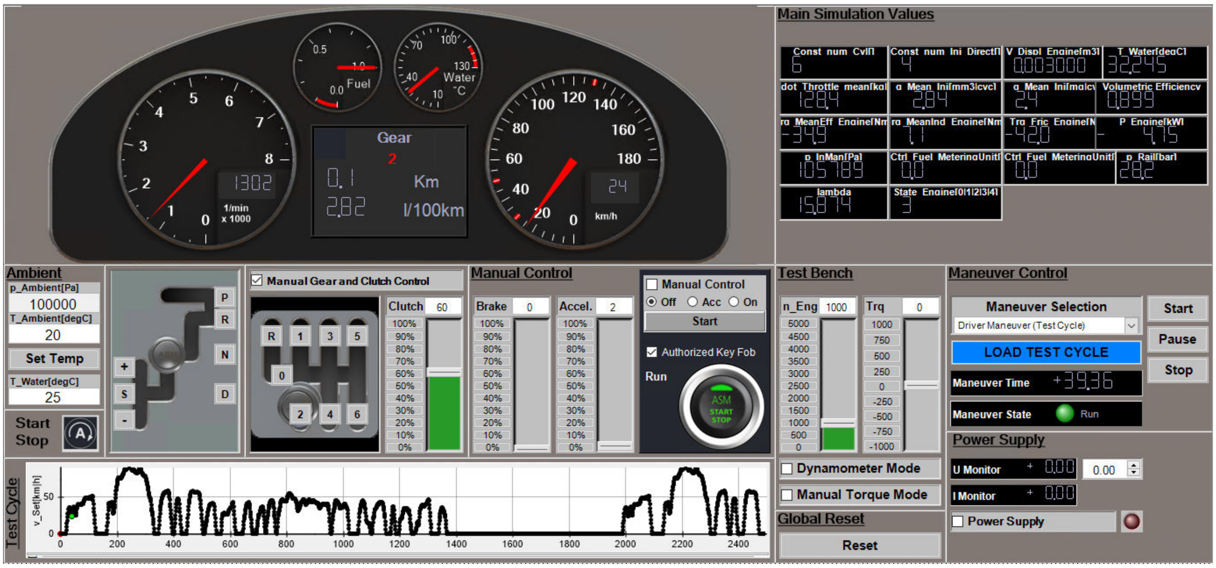
4 Open the platform's context menu.

Open the predefined layout in the Experiment Layouts folder in the project tree of the ControlDesk experiment.

5 Save the experiment.

On the Home ribbon, select Go Online to start online calibration.

The application runs on the selected platform. You can observe the Simulation results on the available layouts.



Main Simulation Values

Const num Cyl[]	Const num Inj Direct[]	V Displ Engine[m3]	T Water[degC]
6	4	0.003000	32.245
dot Throttle mean[kg]	q Mean Inj[mm3/cyc]	q Mean Inj[mg/cy	Volumetric Efficiency
128.4	2.84	2.4	0.899
rq MeanEff Engine[Nm	rq MeanInd Engine[Nm	Trq Fric Engine[N	P Engine[kW]
-34.9	7.1	-42.0	-4.75
p InMan[Pa]	Ctrl Fuel MeteringUnit[	Ctrl Fuel MeteringUnit[	p Rail[bar]
105789	0.0	0.0	28.2
lambda	State Engine[0/1/2/3/4]		
15.874	3		

Ambient: p_Ambient[Pa] 100000; T_Ambient[degC] 20; T_Water[degC] 25

☑ Manual Gear and Clutch Control

Manual Control: Clutch 60; Brake 0; Accel. 2

□ Manual Control ● Off ○ Acc ○ On

☑ Authorized Key Fob

Test Bench: n_Eng 1000; Trq 0

□ Dynamometer Mode

□ Manual Torque Mode

Maneuver Control: Maneuver Selection: Driver Maneuver (Test Cycle); Maneuver Time +39.36; Maneuver State Run

Power Supply: U Monitor 0.00 0.00; I Monitor 0.00

□ Power Supply

6 Use the instruments in the dashboard layout to control the maneuver, to start and stop the engine and to observe variables such as vehicle speed and engine speed.

You can also select whether you would like to drive with the ASM driver model, with stimulated input signals, or with a manual control, by changing the maneuver selection. In real-time simulation, you have to set the pedal positions and the gear switch yourself, for example, by using the instruments of the Manual Control area in the dashboard layout.

Depending on the ASM used, you can use further layouts to observe engine or vehicle dynamics signals.

Result

You observed results in ControlDesk.

Note

If you want to update the real-time application, note the following points: For RTI and VEOS, the real-time code is generated and saved in the MATLAB working directory, which is typically the `Simulation` folder. When you generate the code repeatedly in the same directory, ControlDesk detects the update and prompts you to reload the real-time application when going online.

Note

Generating new code with an RTI platform (e.g., DS1007) creates a host service named HostService by default. The provided ControlDesk experiments expect a host service named Periodic Task1 (default name for SCALEXIO and VEOS platforms). If you do not manually rename the default host service for the RTI platform to Periodic Task1, the ControlDesk Message Viewer displays the following messages:

```
ControlDesk NG MeasurementConfiguration:
Warning …:Could not create the signal for variable 'XYZ’ on
platform 'ABC' with raster 'Periodic Task 1'.
ControlDesk NG MeasurementConfiguration:
Warning …:Could not create the signal for the following
variable on platform ABC with raster Periodic Task 1: XYZ
```

These messages depict the difference between the existing name and the expected name of the host service. However, this behavior has no negative effect on the functionality of the ControlDesk experiment. The signals are connected with the present host service.

How to Update a Plot Configuration for ModelDesk for a dSPACE Platform

Objective

The ASM package includes a signal interface block for ModelDesk so that signals of the ASMSignalBus can be plotted. This block must be updated manually if the ASMSignalBus is changed.

If the respective simulation model is an operator version, it is mandatory to perform a manual update of the plotting block and plotting configuration, before using the plotting feature.

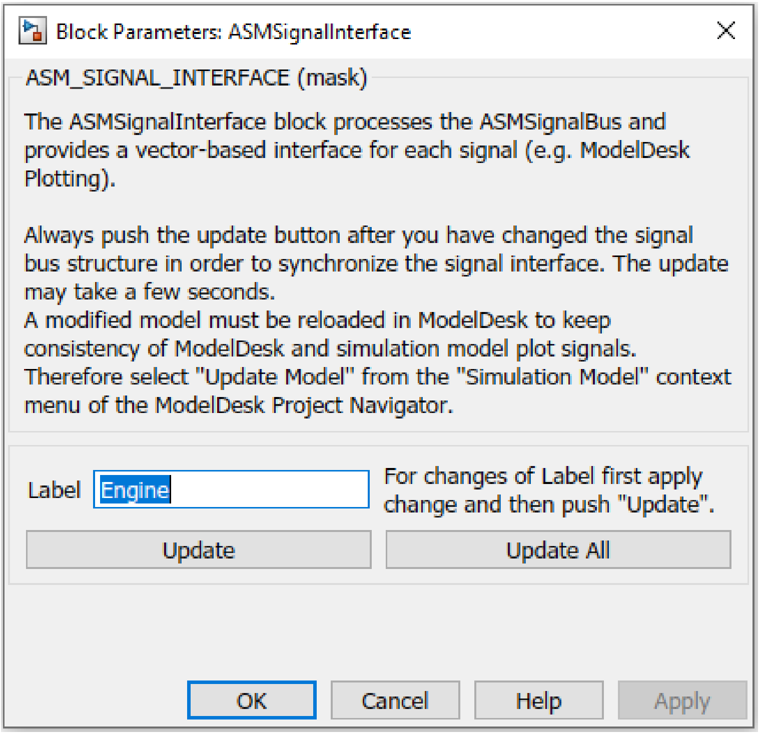
Method

To update a plot configuration for ModelDesk for a real-time simulation

1 Inside an ASM module, open the SignalInterface subsystem.

2 Double-click the ASMSignalInterface block.

The following dialog appears with varying labels, depending on your model.



Block Parameters: ASMSignalInterface

ASM_SIGNAL_INTERFACE (mask)

The ASMSignalInterface block processes the ASMSignalBus and provides a vector-based interface for each signal (e.g. ModelDesk Plotting).

Always push the update button after you have changed the signal bus structure in order to synchronize the signal interface. The update may take a few seconds.
A modified model must be reloaded in ModelDesk to keep consistency of ModelDesk and simulation model plot signals.
Therefore select "Update Model" from the "Simulation Model" context menu of the ModelDesk Project Navigator.

Label: Engine

For changes of Label first apply change and then push "Update".

3 Click:

- Update: To update only the current block.
- Update all: To update all blocks in the model that are currently not commented out.

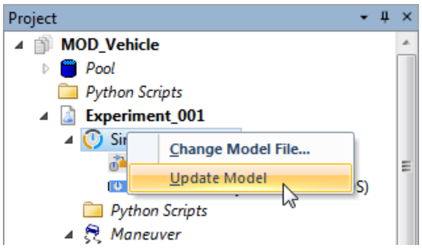
All the ASMSignalInterface blocks are updated. This also changes the model, which you must update in ModelDesk.

4 Start a new code generation process.

You must update the simulation model in ModelDesk to make all changes in the ASMSignalBus visible to ModelDesk.

Make sure your experiment is linked with the correct real-time model SDF file. ModelDesk shows the active model in bold.

5 In ModelDesk's Project Manager, right-click the Simulation model element and select Update Model.



Result

The plot configuration is updated.

Related topics

Basics

Collecting Signals for Plotting in the ASM Model (ModelDesk Plotting 📖)

References

Signal Selector (ModelDesk Plotting 📖)

Changing the Simulation Platform

Introduction

You can change the simulation platform and reuse ControlDesk experiments. For how to change from VEOS to SCALEXIO, for example, refer to Switching the Simulation Platform and Reusing Experiment Parts (ControlDesk Platform Management 📖).

Changing between other platforms works similarly.

How to Use the ModelDesk Project for the Real-Time Simulation

Objective

You can use a ModelDesk project for real-time simulation.

For this, you must make different settings in ModelDesk.

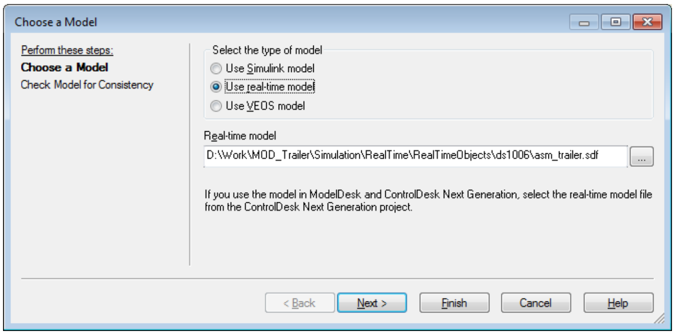
Method

To use the ModelDesk project for the real-time simulation

1 Open the ModelDesk project in:`<Projectfolder>\Parameterization\MOD_EC_ElSystem\MOD_EC_ElSystem.CDP`.

2 In the ModelDesk **Project Navigator**, open the context menu of the **Simulation Model** node and choose **Change Model File**. This has to be done to select the real-time simulation for ModelDesk, so that ModelDesk knows that the parameters should be loaded to the real-time hardware.

3 In the dialog, select **Use real-time model** and enter the path of the corresponding system description file.



Choose a Model

Perform these steps:

Choose a Model

Check Model for Consistency

Select the type of model

○ Use Simulink model

● Use real-time model

○ Use VEOS model

Real-time model: D:\\Work\MOD_Trailer\Simulation\RealTime\RealTimeObjects\ds1006\asm_trailer.sdf

If you use the model in ModelDesk and ControlDesk Next Generation, select the real-time model file from the ControlDesk Next Generation project.

4 Click **Finish** to end this process.

5 Parameterize your model in ModelDesk and switch to the experiment main page and click the **Download all** button to download the currently activated parameters of the ModelDesk project.

6 Switch to ControlDesk and observe the results on the different layouts.

Result

You parameterizes your ModelDesk project for real-time simulation.

Related topics

HowTos

How to Choose a Model and Initialize the Consistency Check (ModelDesk Parameterizing 🕮)

Troubleshooting

Troubleshooting

Vehicle dynamics hybrid demo model

For information on troubleshooting with the vehicle dynamics hybrid demo model, refer to Troubleshooting (ASM Vehicle Dynamics Model Description 📖).

Gasoline engine hybrid demo model

For information on troubleshooting with the gasoline engine hybrid demo model, refer to Troubleshooting (ASM Gasoline Engine Model Description 📖).