ASM Drivetrain Basic

Model Description

Release 2021-A – May 2021

dSPACE

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

Contents

About This Document 5

Introduction to the Dual Clutch Transmission Demo 7

Introduction ... 7
Transmission Mechanics ... 8
Transmission Hydraulics ... 9

Tutorials 15

Handling the Model ... 16
How to Open an ASM Library ... 16
How to Start with an ASM Demo Model via MATLAB ... 18
How to Start with an ASM Demo Model Using ModelDesk ... 20
How to Open a Parameterization Project ... 21
How to Parameterize ASM via ModelDesk ... 22
How to Select Maneuver Conditions ... 26
How to Select a Test Cycle ... 28
How to Plot Simulation Results in ModelDesk ... 30
How to Switch the Library Type ... 32
How to Handle Multi-Instances ... 33
How to Display the ModelDesk Parameter Group ... 35

Simulating in Simulink ... 36
How to Use the Simulink Accelerator Mode ... 36
How to Simulate the Model Manually in Simulink ... 37
How to Simulate the Model and Observe the Results in Simulink ... 40
How to Handle Simulink Plotters ... 40
How to Update a Plot Configuration for ModelDesk for a Simulink Simulation ... 42
How to Configure a Plotter in ModelDesk for a Simulink Simulation ... 43

Simulating on dSPACE Platforms ... 46
How to Generate an OSA File for VEOS ... 46
Generating the Real-Time Application for SCALEXIO ... 48
Generating a Real-Time Application for PHS-Bus-Based Platforms ... 48
How to Observe Results in ControlDesk ... 49

How to Update a Plot Configuration for ModelDesk for a dSPACE Platform ... 51
Changing the Simulation Platform ... 52

Troubleshooting 53

Troubleshooting in MATLAB/Simulink ... 53
Troubleshooting in ModelDesk ... 54
Troubleshooting in ControlDesk ... 55
Troubleshooting with Rising Turnaround Times at Steady Operating Point ... 55

Index 57

About This Document

Content

This document shows you how to work with the ASM Drivetrain Basic Model.

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 to the Dual Clutch Transmission Demo

Where to go from here

Information in this section

Introduction 7
The Dual Clutch Transmission demo builds a detailed dual clutch transmission with hydraulic actuation.

Transmission Mechanics 8
The mechanics consists mainly of a combination of two manual gearboxes with two clutches

Transmission Hydraulics 9
The hydraulic system of the transmission can be divided into hydraulic supply, clutch actuation, and gearshift actuation.

Introduction

Demo

The Dual Clutch Transmission (DCT) demo can be used to build a detailed dual clutch transmission with hydraulic actuation. The default variant is a seven-gears transmission.

Hardware requirements

For real-time simulation, the ASM Dual Clutch Transmission demo can be used with the boards supported by ASM.

Working principle

The gear and clutch engagement are hydraulically managed. In normal driving situations, one gear is always engaged (with the related closed clutch) and another is preselected (with the related open clutch). If a gearshift is requested, the open clutch closes and the closed one opens. The continuous control of clutch actuation is essential to ensure a comfortable and soft gearshift process. The following illustrates the vehicle start and gearshift process in detail.

When the engine is running, the brake pedal is being pressed and the selector lever changes to D position, the first gear will be engaged. Both clutches are open and no torque is transmitted to the wheels. Depending on the driver's request, i.e., accelerator and brake pedal positions, the vehicle starts moving by engaging the first clutch. During the vehicle start-up, the second clutch remains open, while the second gear is preselected. After reaching a certain speed, the transmission ECU (TCU) controls both clutches to engage the second gear, i.e., opening the odd clutch and closing the even clutch. If the vehicle keeps accelerating, the first gear will be disengaged and the third gear is preselected. Thus, there are always one odd and one even engaged gear.

The gear preselection is always performed with the open clutch, so that the process does not damage the synchronizers or other gearbox parts.

Related topics

References

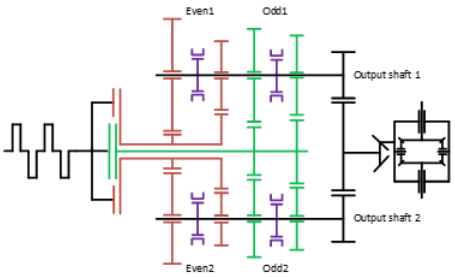
Transmission Hydraulics..... 9
Transmission Mechanics..... 8

Transmission Mechanics

Introduction

The mechanics consists mainly of a combination of two manual gearboxes with two clutches. Additionally, the two output shafts of the transmission combination are connected to the final drive with different ratios.

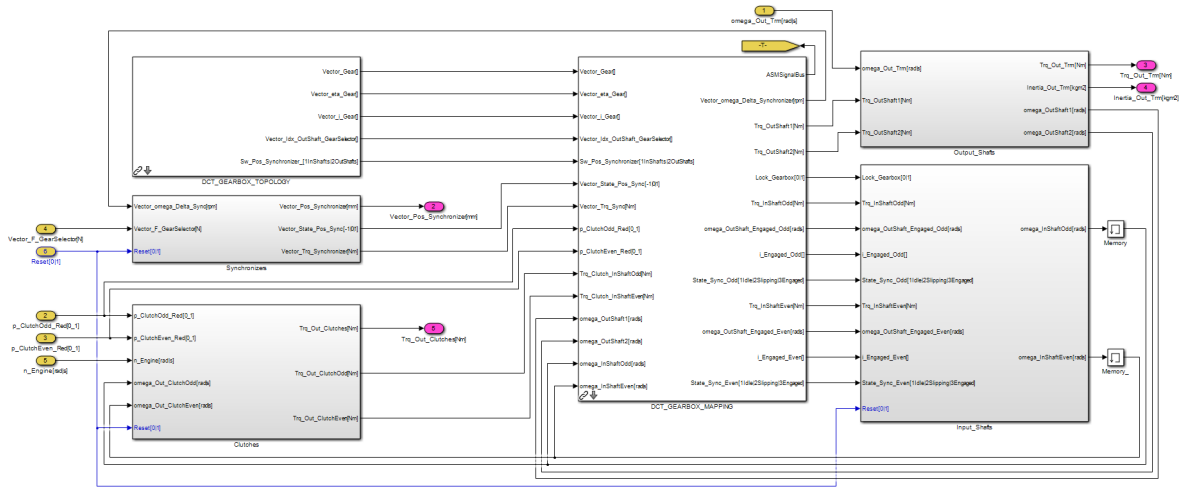
The following illustration shows an example of the mechanical structure of a dual clutch transmission.



There are two input shafts, a rigid and a hollow one. The rigid shaft connects the odd clutch to the odd gear wheels. The hollow shaft connects the even clutch to the even gear wheels.

The gear wheels connect the input shafts to the output shafts by means of synchronizers. Both output shafts are finally connected to the final drive via two final gear trains.

The following illustration shows the first level of the Simulink models of the transmission mechanics.



Related topics

Basics

Introduction 7

References

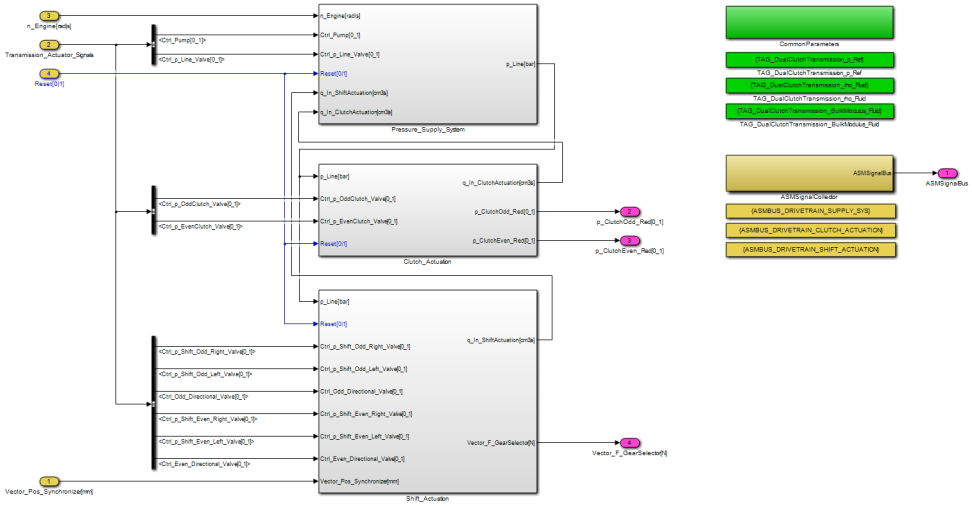
Transmission Hydraulics 9

Transmission Hydraulics

Introduction

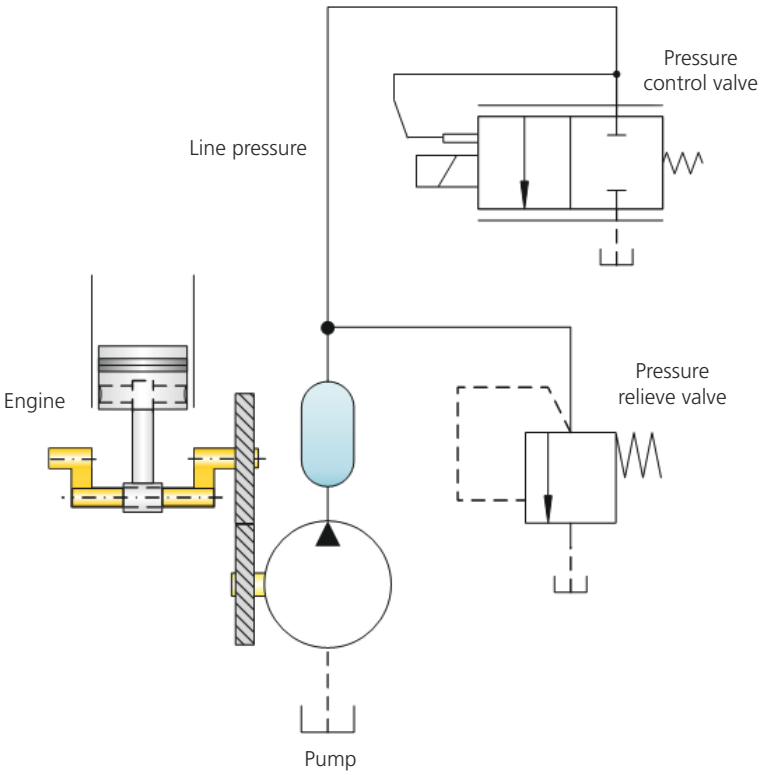
The hydraulic system of the transmission can be divided into hydraulic supply, clutch actuation, and gearshift actuation. The following illustration shows the hydraulic structure of the default variant of the DCT demo actuation system.

The following illustration shows the first level of the Simulink models for the transmission hydraulics.



Pressure supply system

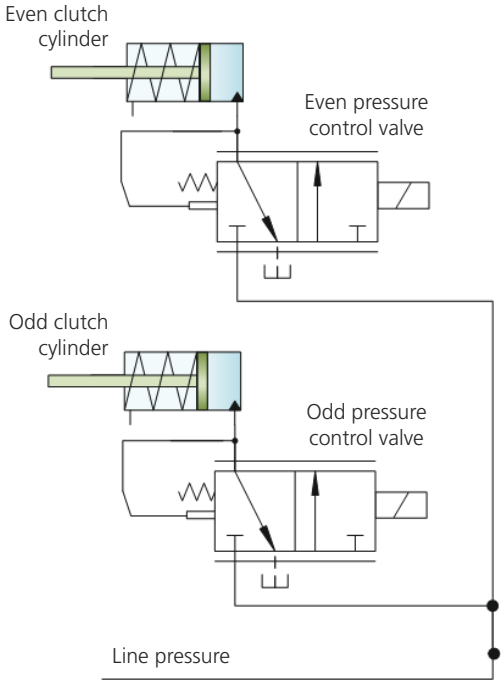
The pressure supply system mainly consists of a mechanical pump and a pressure regulation system. The mechanical pump is driven directly by the combustion engine via a gear train.



The pressure regulation system is used to control the system main line pressure. It consists of a pressure relief valve and a pressure control valve, which is controlled directly by the TCU. The function of the pressure relief valve is to prevent the system from high pressure. The pressure control valve keeps the main line pressure at a certain working point.

Clutch actuation

Two identical pressure control valves are used for clutch actuation. The valves use their output pressure as feedback and are controlled directly by the transmissíon ECU (TCU). They are used to control the pressure in the odd and even clutch cylinders.

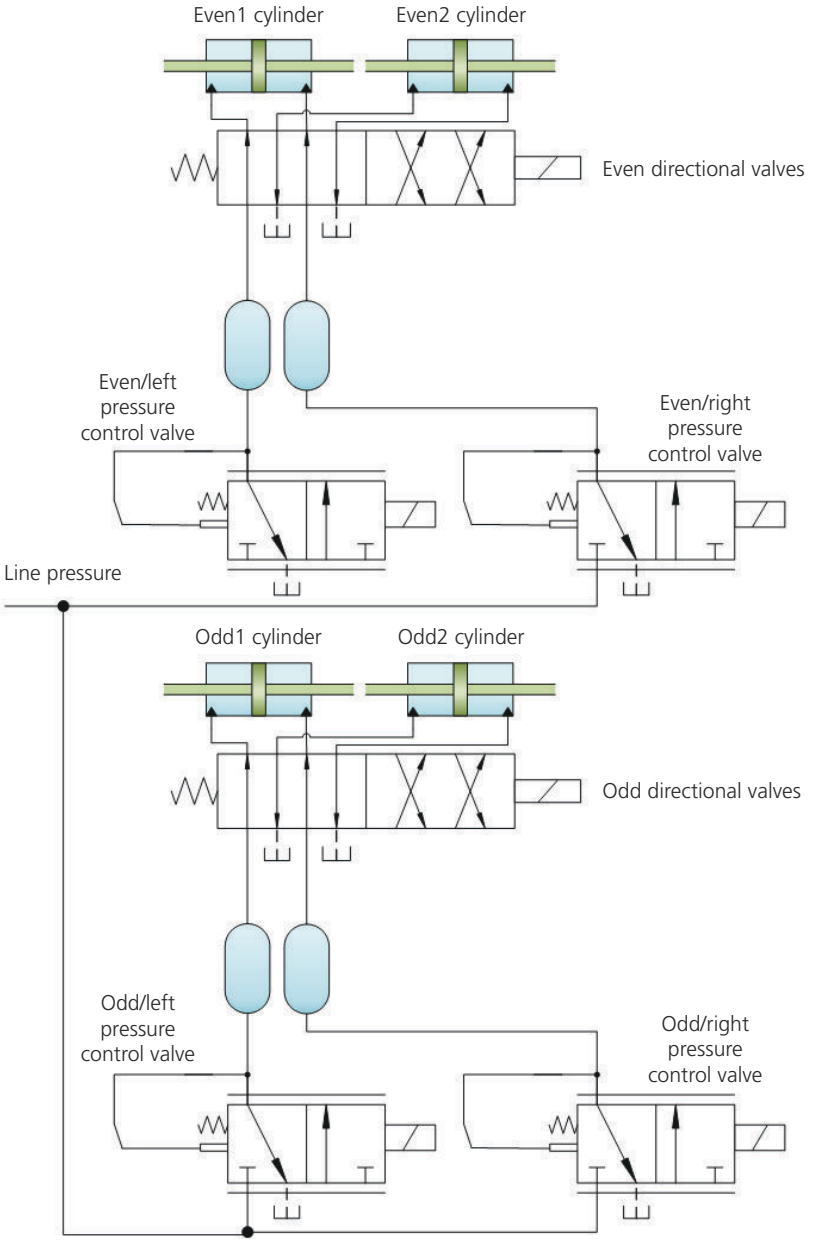


Shift actuation

The system controls the gearshift process and the synchronization process. It consists of actuation subsystems for odd gears and even gears, so that odd and even gears can be shifted in parallel. Each subsystem has a shift pressure control valve to control the gearshift pressure and a flow directional valve to switch between the actuated gear selectors.

A double acting hydraulic cylinder is used for the actuator movement. Each cylinder can move right and left, i.e., it is responsible for engaging one gear on each side.

The following illustration shows the default variant of the shift actuation used in the DCT demo.



Related topics

Basics

Introduction 7

References

Transmission Mechanics 8

Tutorials

Where to go from here

Information in this section

Handling the Model....16
Provides information on how to work with the model.

Simulating in Simulink....36
You can simulate an ASM model in Simulink.

Simulating on dSPACE Platforms....46
You can simulate an ASM model on dSPACE platforms.

Handling the Model

Where to go from here

Information in this section

How to Open an ASM Library....16
After successfully installing ASM on your PC, you can now easily open an ASM library.

How to Start with an ASM Demo Model via MATLAB....18
After successfully installing ASM on your PC, you can easily start an example model of your ASM installation via MATLAB.

How to Start with an ASM Demo Model Using ModelDesk....20
After successfully installing ASM on your PC, you can now easily start a demo model of your ASM installation using ModelDesk.

How to Open a Parameterization Project....21
You can open a parameterization project from the model.

How to Parameterize ASM via ModelDesk....22
You can parameterize ASM with ModelDesk.

How to Select Maneuver Conditions....26
You can select different maneuver conditions.

How to Select a Test Cycle....28
You can select different test cycles.

How to Plot Simulation Results in ModelDesk....30
You can plot simulation results in ModelDesk via the SignalInterfaces.

How to Switch the Library Type....32
You can switch between the developer library and the operator library.

How to Handle Multi-Instances....33
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.

How to Display the ModelDesk Parameter Group....35
You can modify the grouping in the parameter set that is displayed in the project tree of your ModelDesk experiment.

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

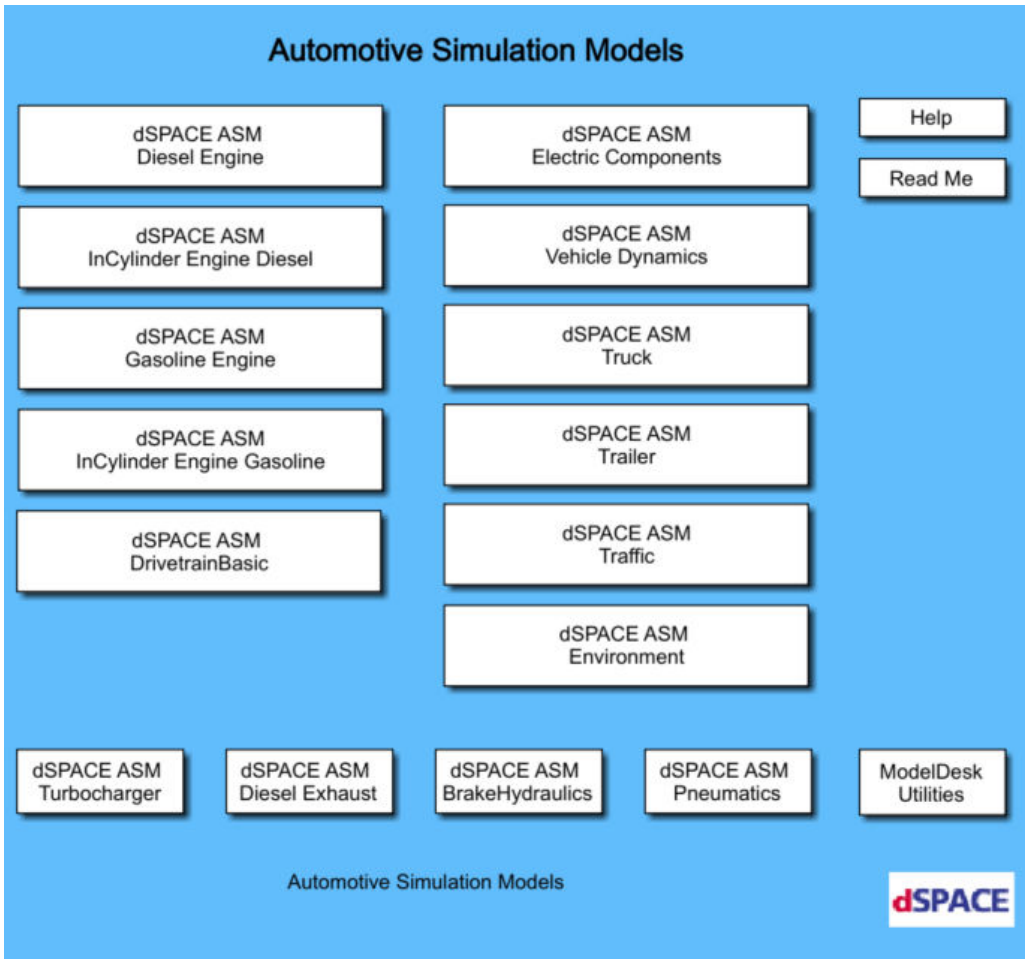
To open an ASM library

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.



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

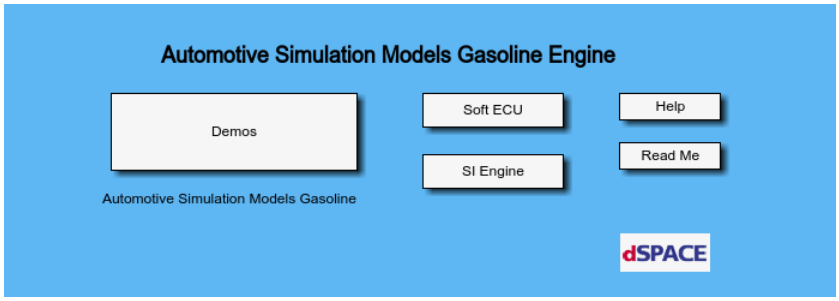
- ASM Electric Components
- ASM Trailer
- ASM Traffic
- ASM Truck
- ASM Environment

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.



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.....18

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 🕮).

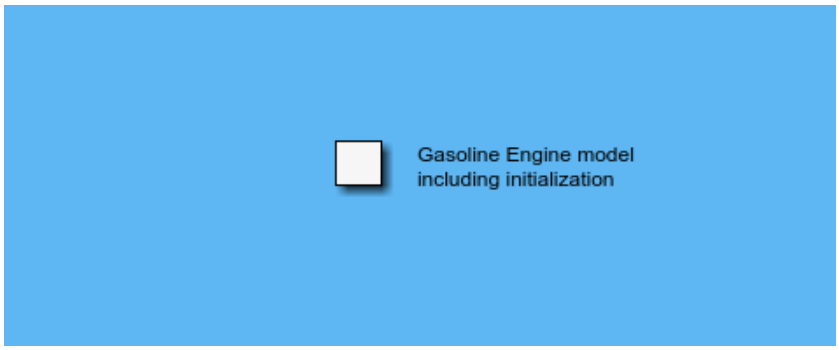
Preconditions

The ASM library is open. Refer to How to Open an ASM Library on page 16.

Method

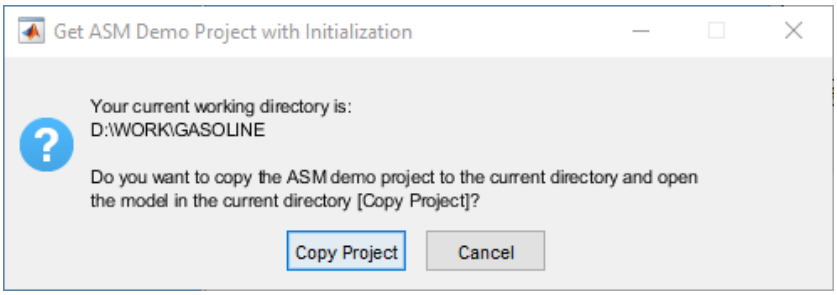
To start with an ASM demo model via MATLAB

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.



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 18.

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

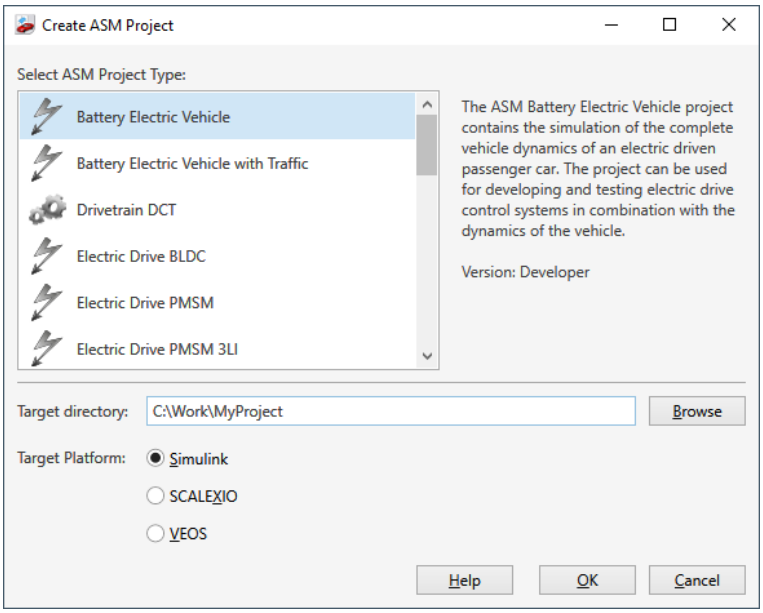
- No other project must be open.
- The license of the ASM library that is used by the ASM demos must be available.
- The ASM library must be decrypted.

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.



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

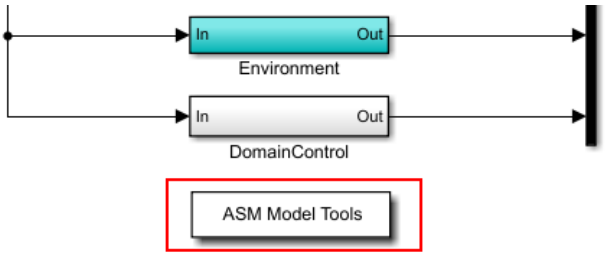
You can open a parameterization project from the model.

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

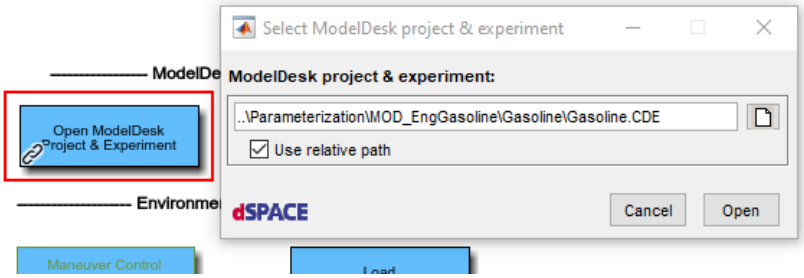
Method

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.



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.

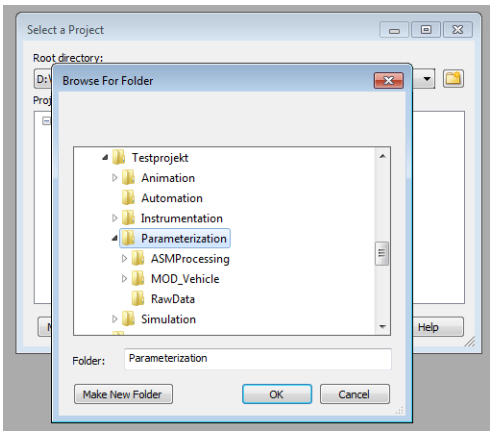
In the following method, an ASM Electric Components demo is used as an example.

Method

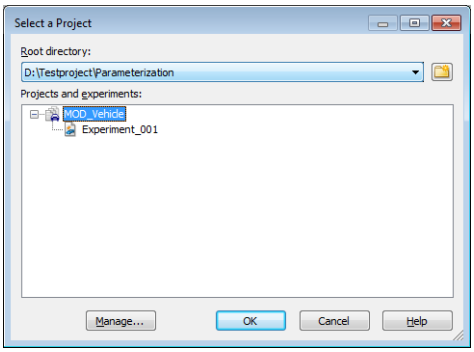
To parameterize ASM via ModelDesk

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

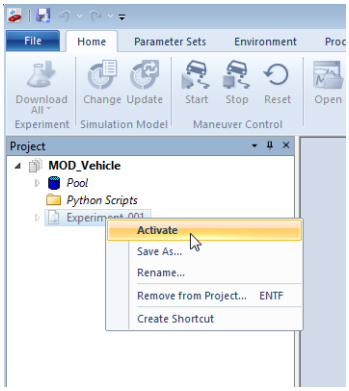
1. Click File – Open – Project + Experiment.
2. Define the root directory in `<Projectfolder>\Parameterization`.



3. Select the project.

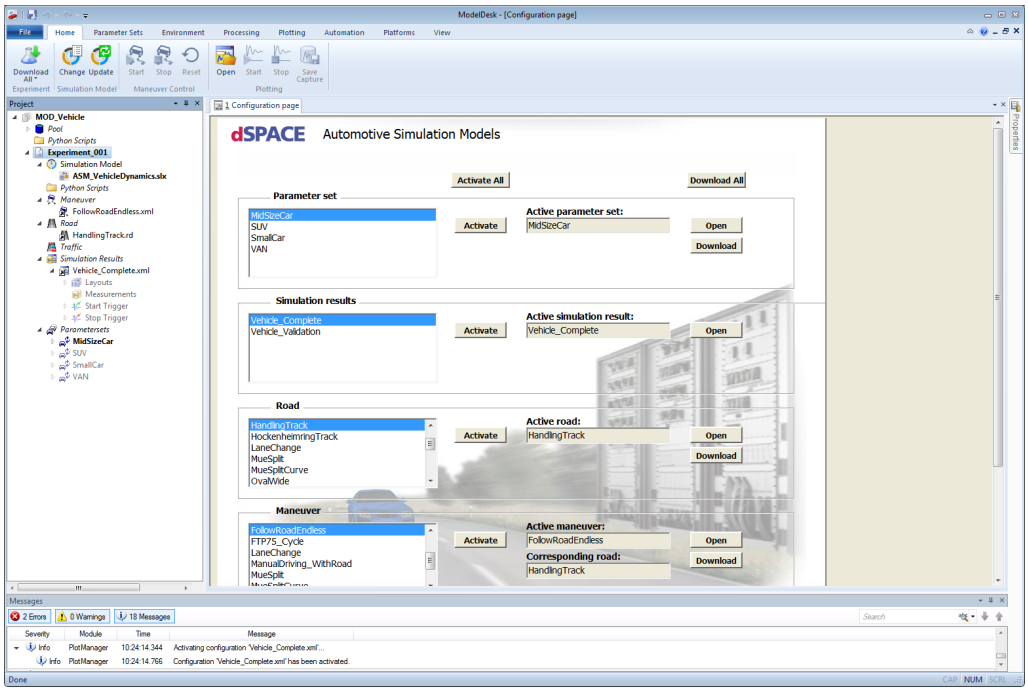


2 Activate the demo experiment via context menu on the experiment in the Project Navigator.



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

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



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.

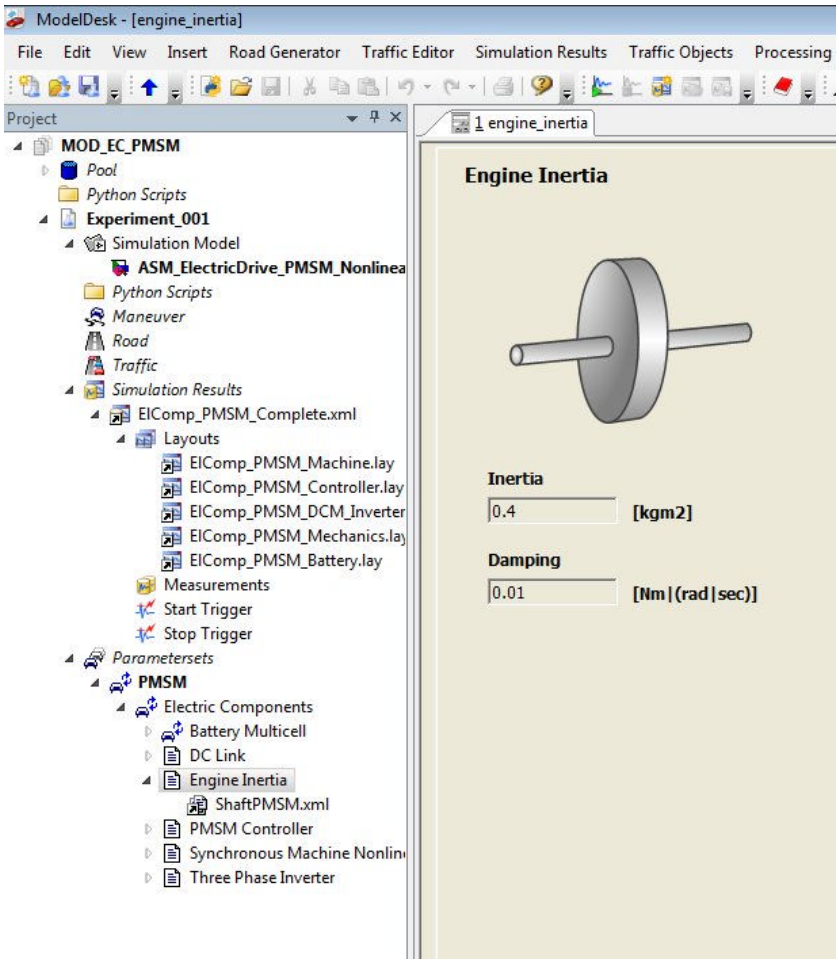
You can also activate plotting sets and open them to obtain simulation results in ModelDesk via the ModelDesk Plotting feature.

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.



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 Select Maneuver Conditions

Objective

You can select different maneuver conditions.

Maneuver type

The following table describes the options for the maneuver type:

Maneuver Type	Description
Manual	To simulate the model manually in ControlDesk on a dSPACE platform or in Simulink on a standard PC.
Stimulus	To define a time-dependent stimulus maneuver to test the behavior of the model.
Test cycle	Contains the longitudinal driving and engine cycles, which can be defined by a MATLAB function that is part of your installation.

Maneuver state

Different states are possible for the Test cycle and Stimulus maneuver types.

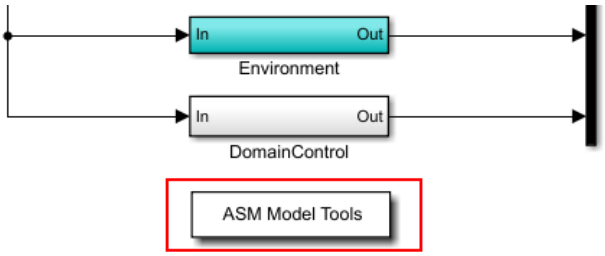
The following table describes the options for the maneuver state:

Maneuver State	Description
Stop	Maneuver stopped.
Start	Maneuver started.

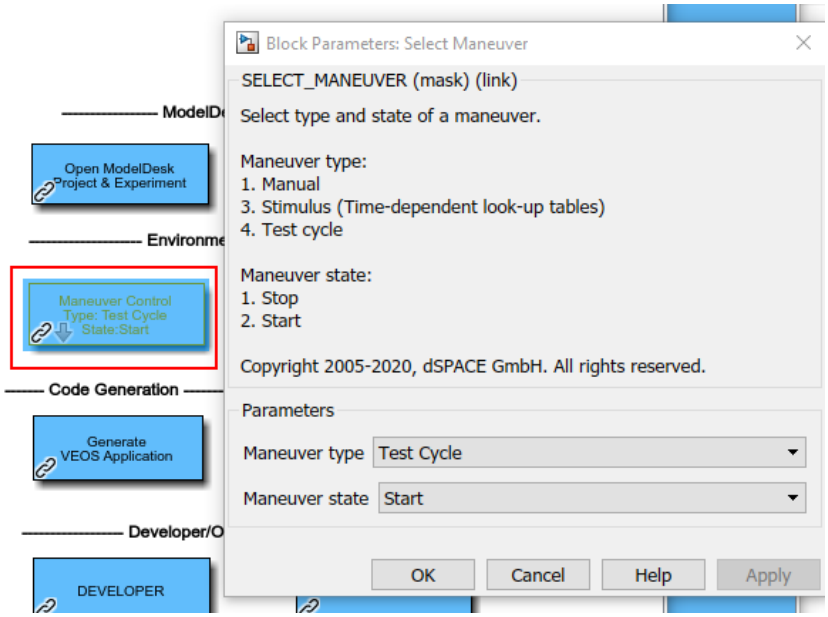
Method

To select maneuver conditions

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



2 Double-click the Test cycle Start button.



3 In the Simulink dialog, adjust the maneuver type and the maneuver state and click **OK**.

Result

You selected maneuver conditions.

Related topics

References

Select Maneuver (ASM User Guide 🕮)

How to Select a Test Cycle

Objective

You can select different test cycles.

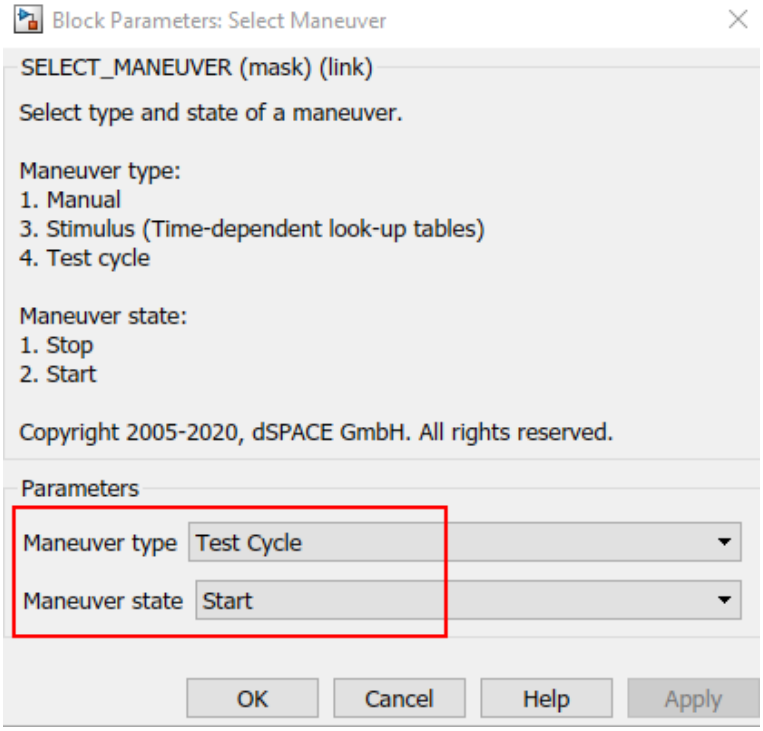
Description

The test cycles are stored in the ASM project at `<Projectfolder>\Simulation\IniFiles\DrivingCycles`. You can select a test cycle via a Simulink dialog.

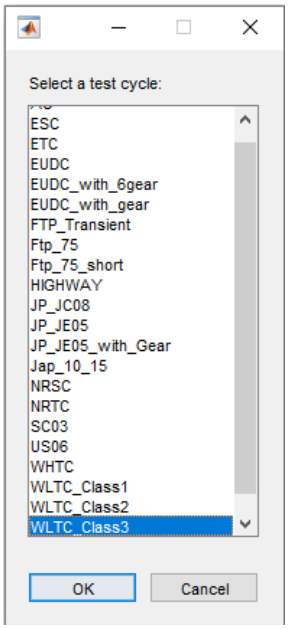
Method

To select a test cycle

1 In the ASM Tools, set the maneuver type to *Test Cycle* and the maneuver state to *Start*. Refer to How to Select Maneuver Conditions on page 26.



2 In the ASM Tools, double-click the Load Test Cycles button to open the Simulink dialog which lists all the driving cycles.



3 In the Simulink dialog, select a new test cycle and click OK.

4 Restart the simulation with the new test cycle.

Result

You selected the test cycle.

Related topics

References

Test Cycle (ASM User Guide 🕮)

How to Plot Simulation Results in ModelDesk

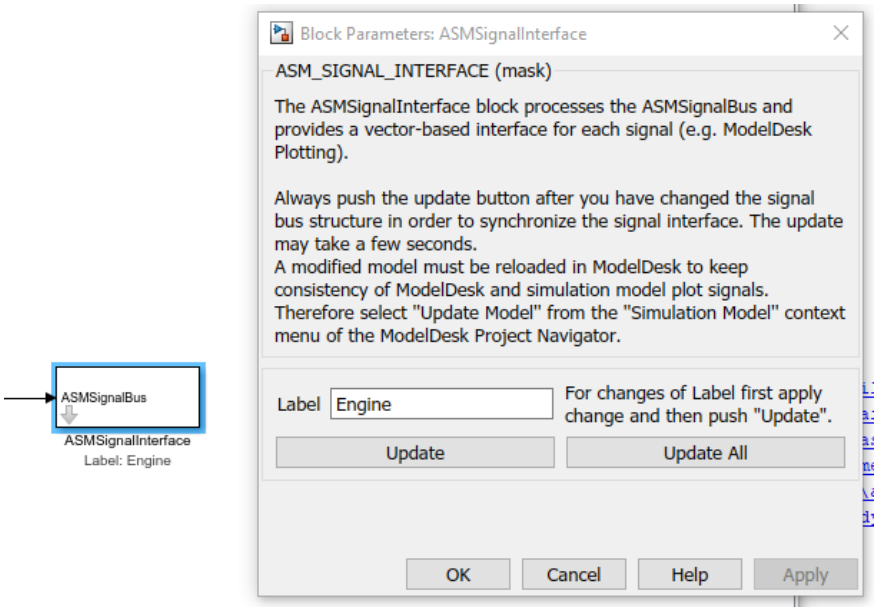
Objective

You can plot simulation results in ModelDesk via the SignalInterfaces.

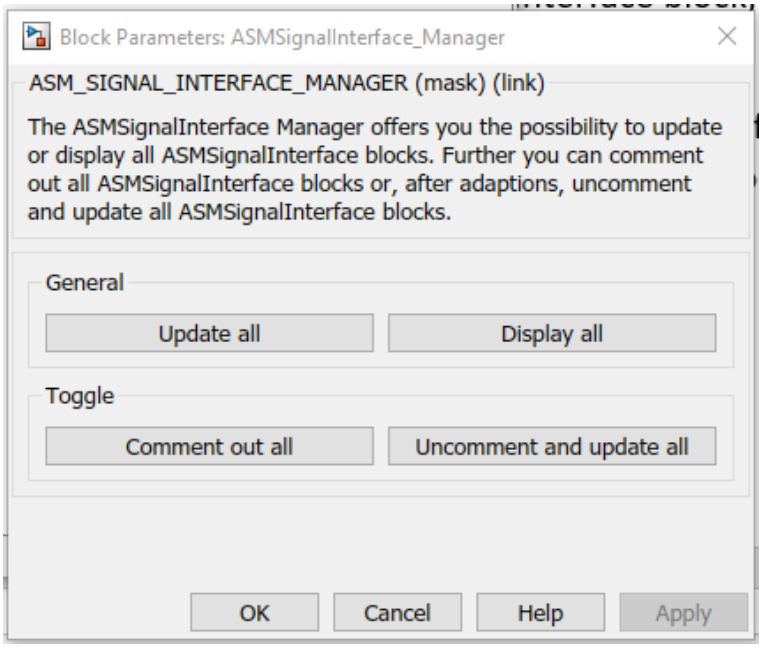
Description

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 🕮).



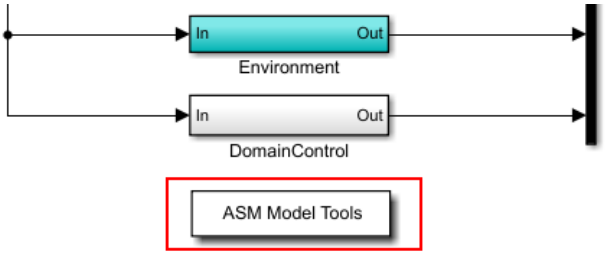
ASMSignal Interface Manager The ASMSignalInterface Manager collects the signals of all modules of your model. You can open it via the ASM Tools. Refer to ASMSignalInterface Manager (ASM User Guide 🕮).



Method

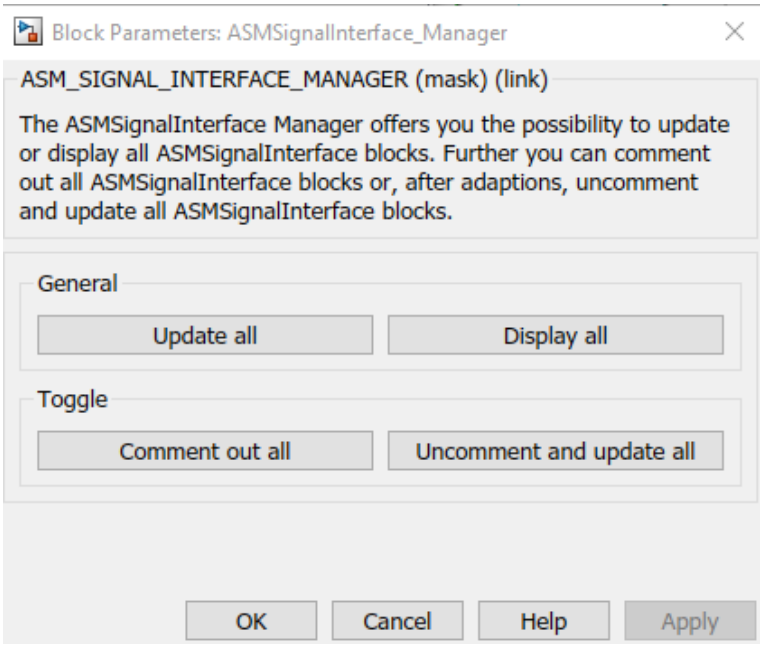
To plot simulation results in ModelDesk

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.



Result

You activated or deactivated plotting in ModelDesk.

Related topics

HowTos

How to Configure a Plotter in ModelDesk for a Simulink Simulation.....43
How to Update a Plot Configuration for ModelDesk for a Simulink Simulation.....42

How to Switch the Library Type

Objective

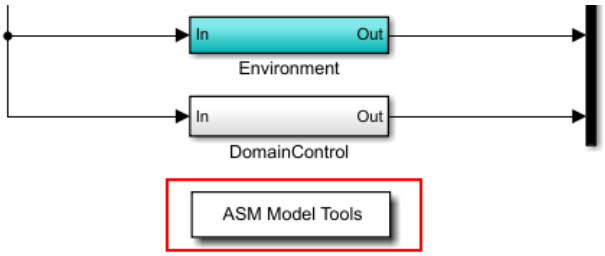
You can switch between the developer library and the operator library.

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.

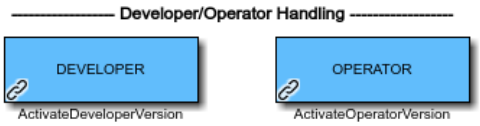
Method

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

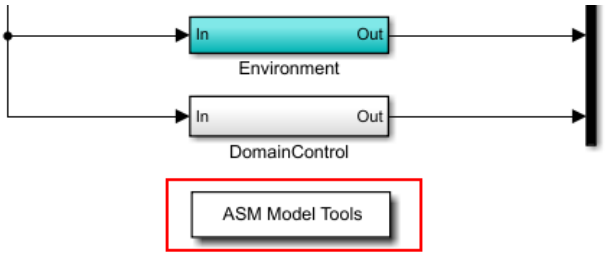
The Multi-Instance Overview button opens a dialog to get an overview of all multi-instance blocks in the model and to set the multi-instance parameters.

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

Method

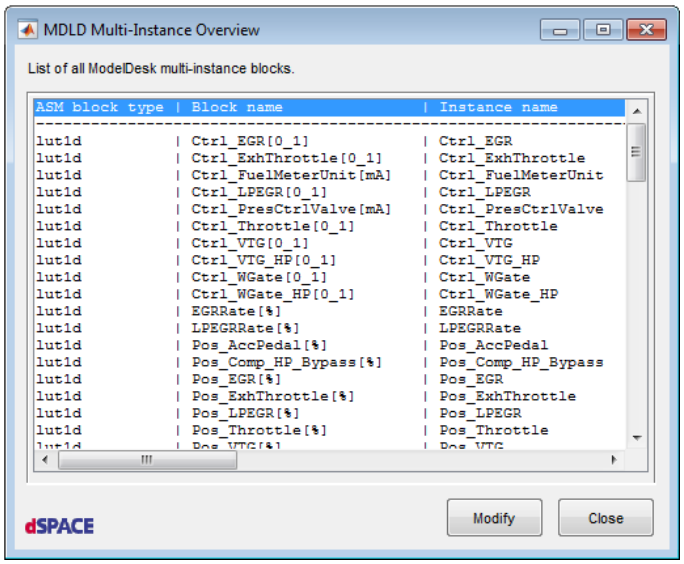
To handle multi-instances

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.



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 🕮).

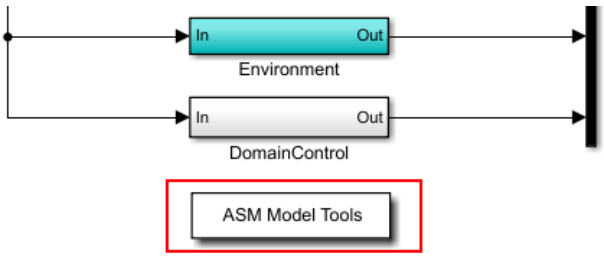
The ModelDesk Parameter Group Overview block in the ASM Tools opens an overview of the ModelDesk parameter group blocks that are contained in the simulation model.

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....36
You can speed up the simulation by using the Simulink Accelerator mode.

How to Simulate the Model Manually in Simulink....37
You can manually simulate your model in Simulink with prepared dashboard instruments.

How to Simulate the Model and Observe the Results in Simulink....40
You can observe the simulation results in Simulink by using Simulink scopes.

How to Handle Simulink Plotters....40
You can handle the Simulink plotters in your simulation model.

How to Update a Plot Configuration for ModelDesk for a Simulink Simulation....42
If the ASMSignalBus is changed, you must update the plotting interface block and the model in ModelDesk.

How to Configure a Plotter in ModelDesk for a Simulink Simulation....43
To plot a signal from the ASMSignalBus, you have to configure a project and an experiment in ModelDesk.

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.

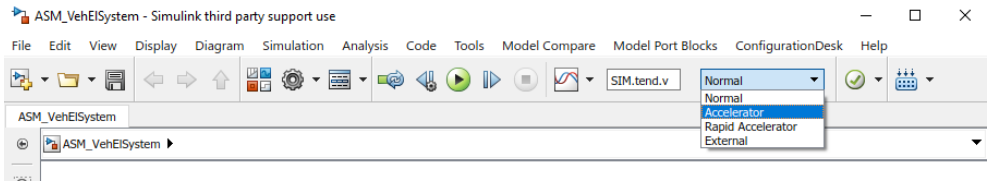
Prerequisite

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.



2 Start the simulation and observe the results on the Simulink plotters, and if available, in MotionDesk (perform a simulation of 200 s simulation time).

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 Manually in Simulink

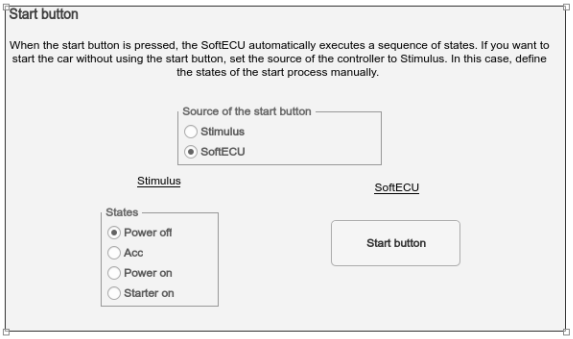
Objective

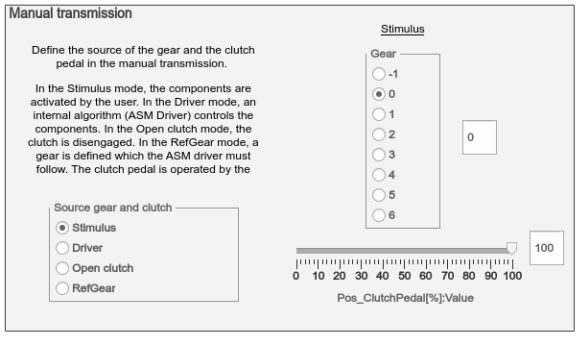
You can manually simulate your model in Simulink with prepared dashboard instruments that come with your ASM model.

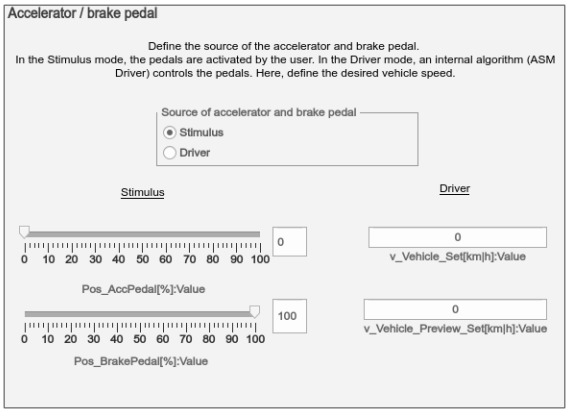
Dashboard instruments

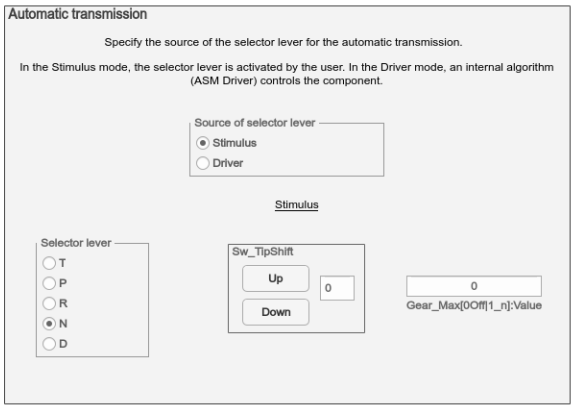
You can find the dashboard instruments in: `/Environment/Plant/UserInterface/PAR_Plant/Manual_Controller`.

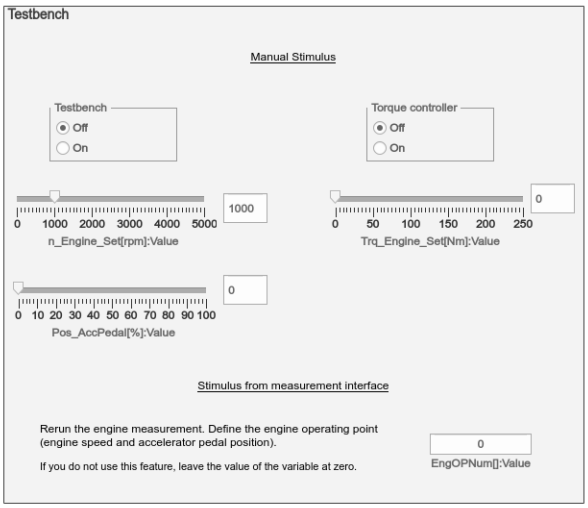
There are the following instruments:

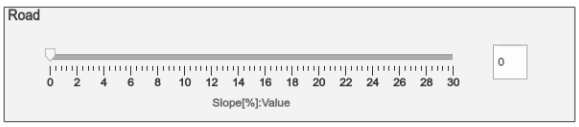




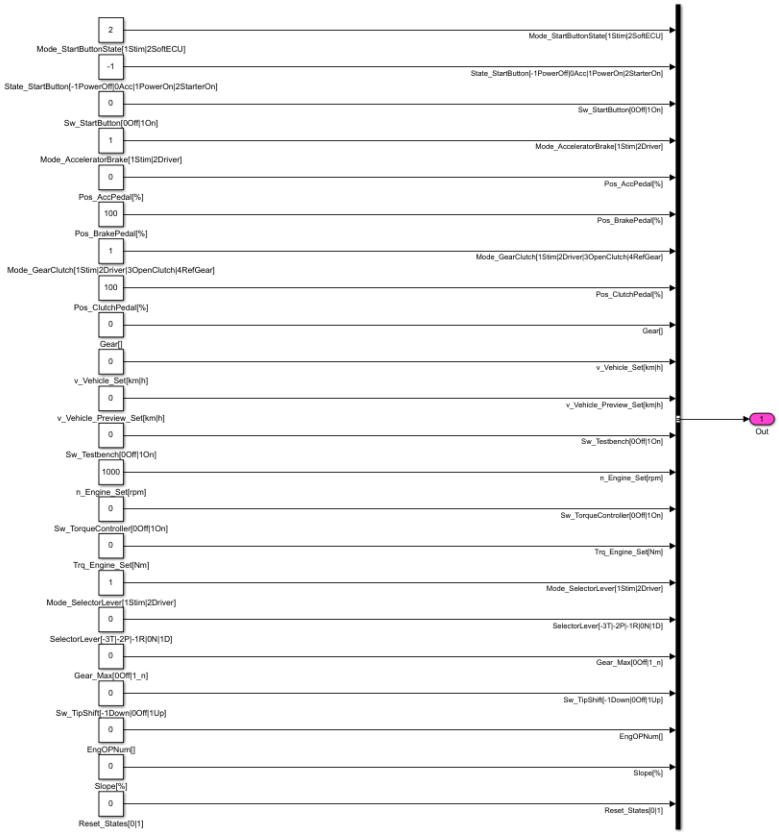








With these instruments, you can change the settings of the model controllers in `/Environment/Plant/UserInterface/PAR_Plant/Manual_Controller/Controller`.



Method

To simulate the model manually in Simulink

1. In the ASM Model Tools, set the maneuver type to *Manual*. This lets you run the model manually.
2. Start the simulation in Simulink.
3. Use the instruments on the dashboard to manipulate parameters in the model.

Result

You can now observe the results of the simulation in Simulink. Refer to How to Simulate the Model and Observe the Results in Simulink on page 40.

Related topics

HowTos

How to Select Maneuver Conditions.....26

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.

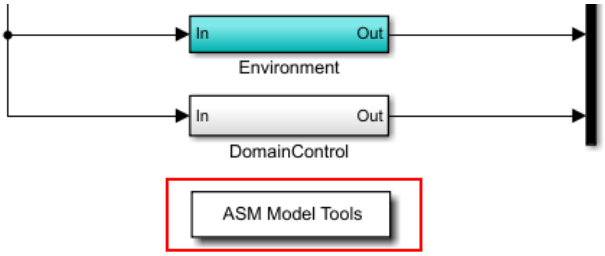
For ASM Engine Models: If you want to simulate your model with `asm_eng_testbench.m`, it is sufficient to store the simulation results in the variables under TEST_CONTROL or TEST_PLANT. No plotters are necessary.

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 40 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.

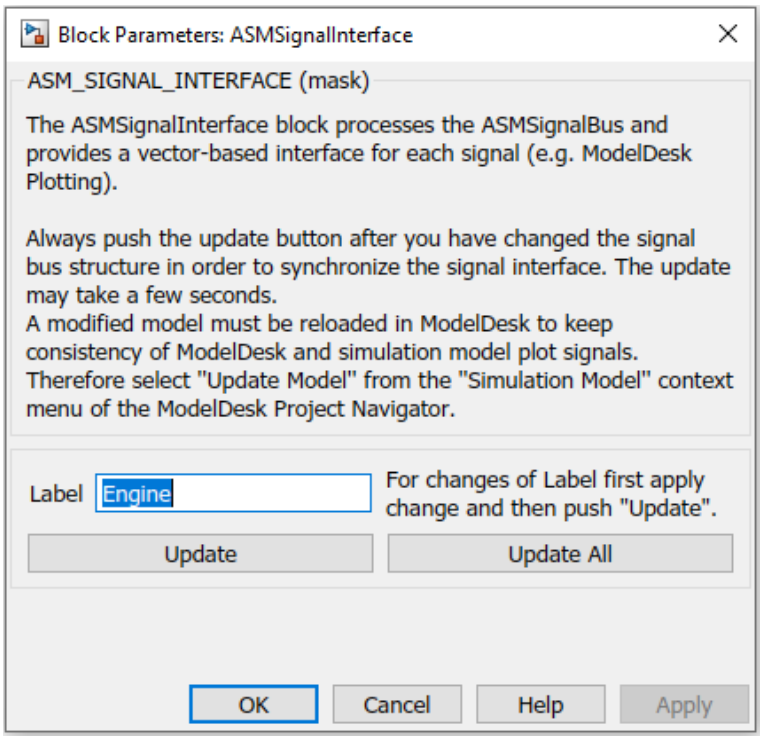
Method

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.

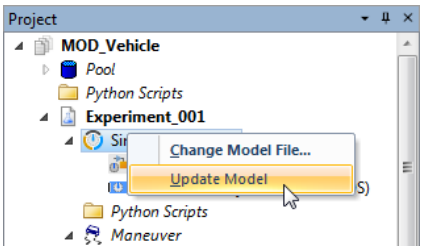


3 Click:

- 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.

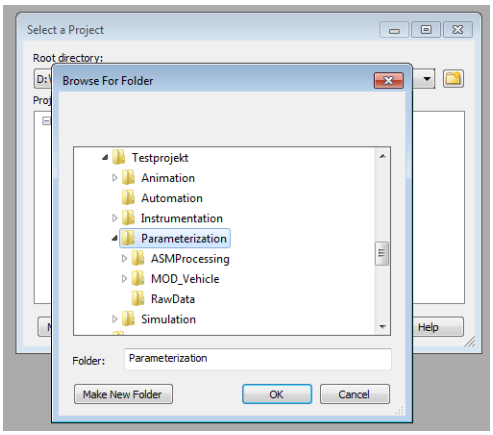
Method

To configure a plotter in ModelDesk for a Simulink simulation

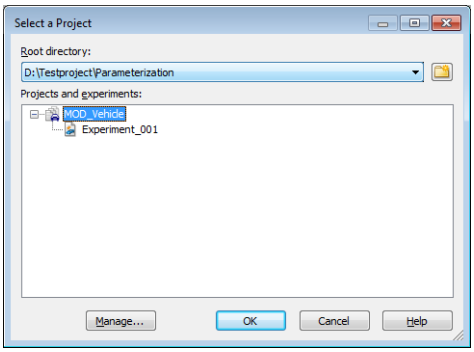
1 Start ModelDesk and open the provided ModelDesk project from `<Projectfolder>\Parameterization`.

1. Click File – Open - Project + Experiment.

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

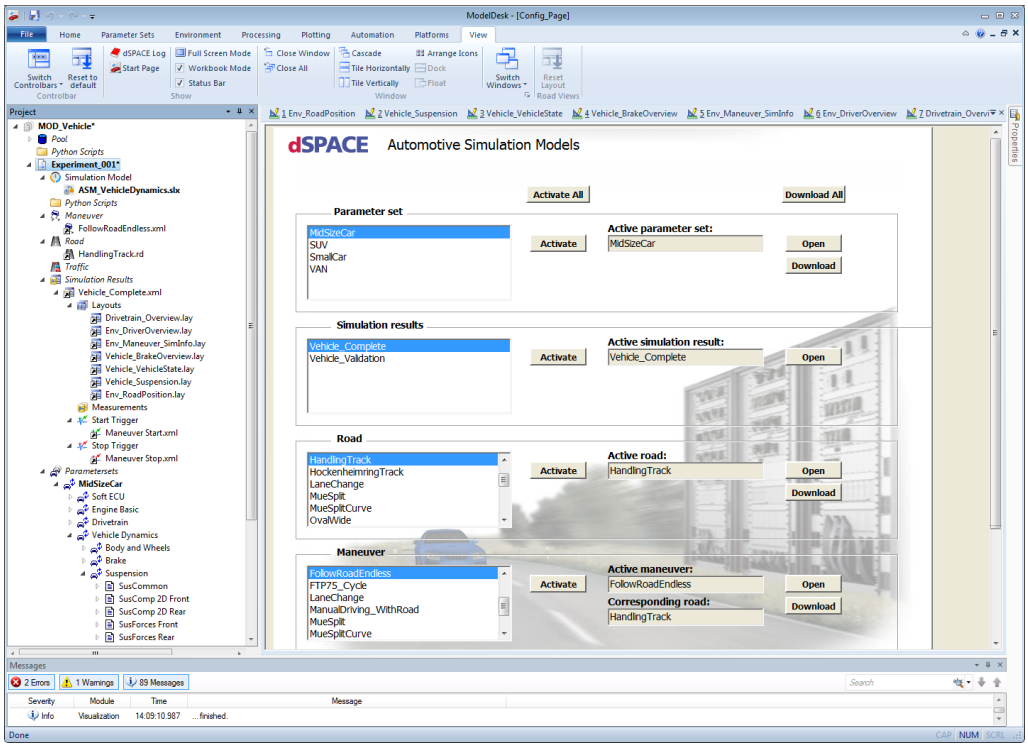


3. Select an experiment from your ModelDesk folder.

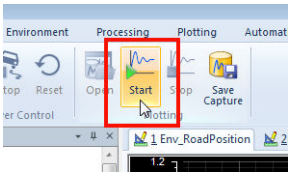


2 Switch to the experiment's **Configuration** page.

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



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

How to Generate an OSA File for VEOS....46
You can generate an Offline Simulation Application (OSA) file for VEOS either manually or automatically.

Generating the Real-Time Application for SCALEXIO....48
You can generate the real-time application for SCALEXIO in ConfigurationDesk.

Generating a Real-Time Application for PHS-Bus-Based Platforms....48
If you have a dSPACE Real-Time Interface (RTI) license, you can generate a new real-time application for the ASM.

How to Observe Results in ControlDesk....49
When simulating the model, you can observe the results in ControlDesk.

How to Update a Plot Configuration for ModelDesk for a dSPACE Platform....51
If the ASMSignalBus is changed, you must update the plotting interface block and the model in ModelDesk.

Changing the Simulation Platform....52
You can change the platform and reuse the experiment.

How to Generate an OSA File for VEOS

Objective

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

Model preparation for VEOS

If the model has an oversampled engine part, it must be implemented as a Simulink For-Iterator subsystem for VEOS simulation. This 'offline' mode is automatically inserted if the model is opened with `go ('simmode', 'PC')`. For more information on the different oversampling strategies, refer to Basic Model Concept (ASM Gasoline Engine InCylinder Model Description 🕮).

Possible methods

There are two ways to generate code for VEOS:

- Manually. Refer to Method 1 on page 47.
- Automatically. Refer to Method 2 on page 47.

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.

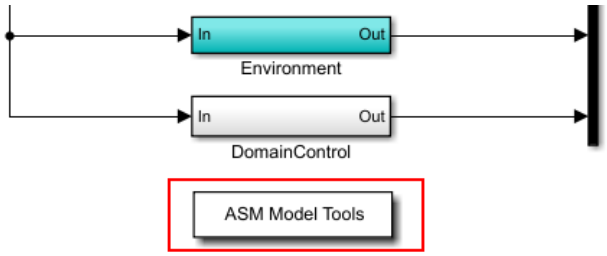
For instructions, refer to How to Import Simulink Implementations (VEOS Manual 🕮).

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 🕮).

Generating a Real-Time Application for PHS-Bus-Based Platforms

Introduction

If you have a dSPACE Real-Time Interface (RTI) license, you can generate a new real-time application for the ASM. This is needed if you want to modify the model, for example, by adding I/O blocks for HIL simulation, or to have different initial settings for the model (engine settings).

Note

RTI code generation of ASM models is currently not supported if the *Accelerator* simulation mode is selected.

For real-time simulation, the model must be adapted. The model can be switched to online (real-time) mode by opening it with the following:

```
go('simmode','HIL', 'platform','RTI')
```

Note

Make sure that the MATLAB current folder is in the Simulation folder of your project, `<Project>\Simulation`

If the online mode is activated, code generation can start. This can be done via the **Simulink Preferences** or with **Ctrl+B** when clicking a Simulink model.

Code is always generated in the current folder.

To generate real-time code to another folder, change the MATLAB current folder after running the **go** file.

Generating code for PHS-bus-based platforms

For information on how to generate code, refer to Building and Downloading the Model (RTI and RTI-MP Implementation Guide 🕮).

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.

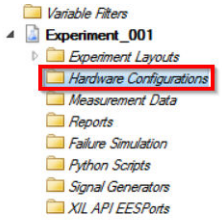
Method

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.

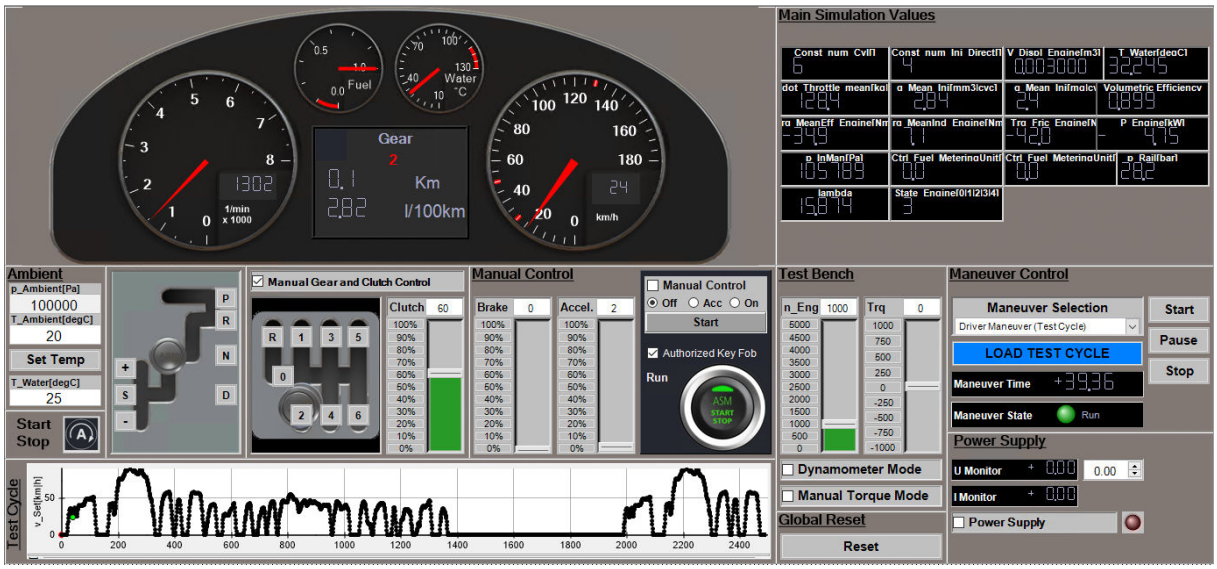
4. Open the platform's context menu.

   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.



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

For some ControlDesk projects, a map layout is available that provides a browser instrument to show a street map with the current position of the simulated vehicle depending on its GPS coordinates. The visualization of the street map is based on OpenStreetMap data.

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.

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.

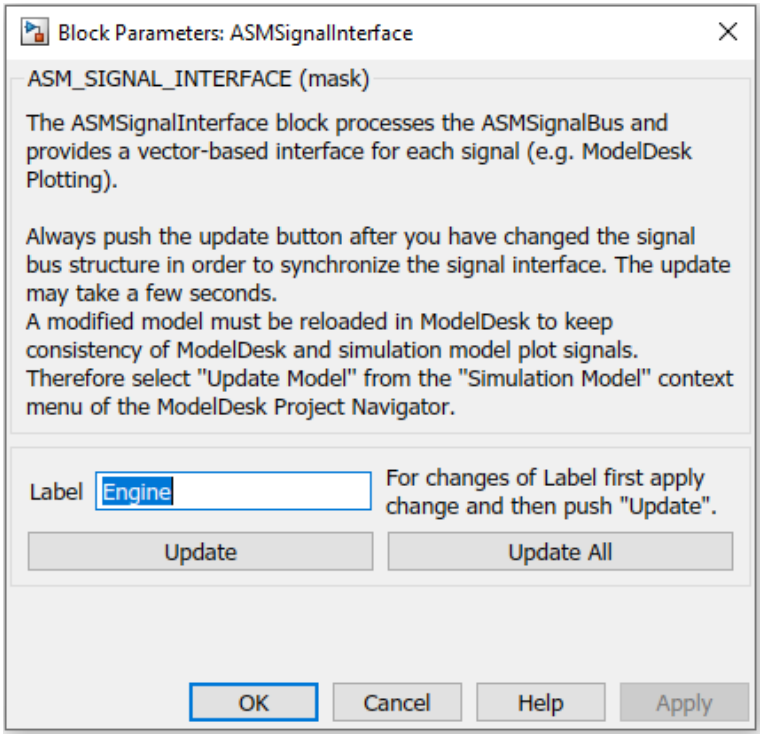
Method

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.



3 Click:

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

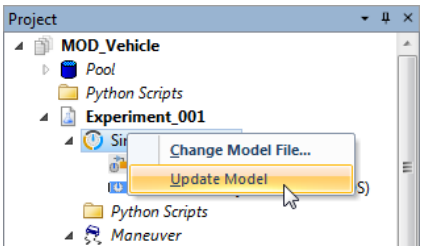
All the ASMSignalInterface blocks are updated. This also changes the model, which you must update in ModelDesk.

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.

Troubleshooting

Where to go from here

Information in this section

Troubleshooting in MATLAB/Simulink 53
These problems might occur when working with ASM and MATLAB/Simulink.

Troubleshooting in ModelDesk 54
These problems might occur when working with ASM and ModelDesk.

Troubleshooting in ControlDesk 55
These problems might occur when working with ASM and ControlDesk.

Troubleshooting with Rising Turnaround Times at Steady Operating Point 55
Simulating the model on a real-time processor board at a steady state, might lead to a significant increase in turnaround time after several seconds of simulating this operating point.

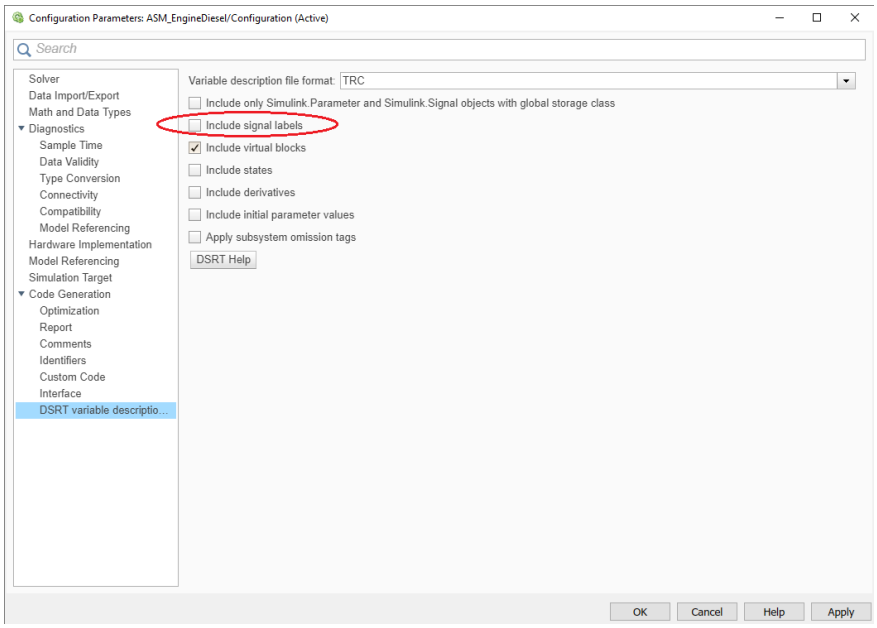
Troubleshooting in MATLAB/Simulink

Out of memory

When the real-time application is built, an Out-of-Memory error at MATLAB might occur: for example, "TLC Compiler encountered an OUT_OF_MEMORY condition." This error message results from an insufficient memory allocation of the MATLAB session.

To avoid this error, one of the following actions can help:

- Disable the **Include signal labels** option at the **RTI variable description** page, see the following illustration.



- In some cases, it is a temporary solution to close MATLAB, and continue with the new MATLAB session.

Related topics

References

Code Generation Dialog (Model Configuration Parameters Dialogs) (RTI and RTI-MP Implementation Reference 🕮)

Troubleshooting in ModelDesk

Wrong model

If downloading new parameters from ModelDesk has no effect on the simulation behavior, check if the selected simulation model in ModelDesk points to the correct model (either Simulink model or real-time model).

In ModelDesk's Project Navigator, open the context menu of the Simulation model and choose **Change Model File**.

Related topics

References

Change Model File (ModelDesk Parameterizing 🕮)

Troubleshooting in ControlDesk

Overrun message

If you get an overrun message in ControlDesk after the real-time application was loaded to the real-time platform, check the following points:

- The processor board must fit the minimum requirements (DS1006 or SCALEXIO).
- For a DS1006 board, the RTI task configuration options must be set correct. Refer to How to Generate a Real-Time Application for PHS-Bus Based Platforms (ASM Vehicle Dynamics Model Description 🕮).
- For a SCALEXIO system, the task priority, step size and offset must be set correct. Refer to Generating the Real-Time Application for SCALEXIO on page 48.

Troubleshooting with Rising Turnaround Times at Steady Operating Point

Description

Simulating the model on a real-time processor board at a steady state, might lead to a significant increase in turnaround time after several seconds of simulating this operating point. When changing to another operating point, the turnaround time decreases to the original lower value. For example, in case of an engine model this behavior might occur when you drag the engine to test bench mode at a fixed operating point.

For information on possible reasons and workarounds, refer to Troubleshooting with Rising Turnaround Times at Steady Operating Points (ASM User Guide 🕮).

A

accelerator mode 36
ASM example model
  starting via MATLAB 18
ASM library
  opening 16

C

code generation
  for PHS-bus-based platforms 48
  for SCALEXIO 48
  for VEOS 46
Common Program Data folder 6
ControlDesk 49

D

Documents folder 6

H

handling
  Simulink plotters 40
handling multi-instances 33

L

library types 32
Local Program Data folder 6

M

maneuver condition
  selecting 26
ModelDesk
  parameterizing 22
ModelDesk plotting 30
multi-instances 33

O

opening
  ASM library 16
opening parameterization project 21

P

parameterizing using ModelDesk 22
PHS-bus-based platforms 48
plot configuration
  for real-time simulation 51
  for VEOS 51
  update 42
plotter
  in ModelDesk 43
plotting 30

S

SCALEXIO
  code generation 48
scopes 40
selecting maneuver conditions 26
selecting test cycle 28
Simulink plotters
  handling 40
starting via MATLAB
  ASM example model 18
switching library types 32

T

test cycles 28
troubleshooting 53

U

using the accelerator mode 36

V

VEOS
  code generation 46

W

watching simulation results 40