Virtual Validation with dSPACE

Benefits the entire ECU development process
# Virtual Validation

## Contents

- Virtual Validation ............................................. 3
- Virtual ECUs .................................................. 4
- Tool Chain for Virtual Validation ......................... 5
- Versatility by Supporting Automotive Standards ........ 5
- Support of Functional Mock-up Interface (FMI) ........... 6
- Standard Interfaces ........................................... 7

### Use Cases

- Generate V-ECUs ............................................. 8
- Simulate V-ECUs .............................................. 9
- Integrate V-ECUs .............................................. 10
- Testing and Experimenting with V-ECUs ................. 11

### Products for Virtual Validation

- SystemDesk V-ECU Generation Module .................. 12
- VEOS .......................................................... 14
- Bus Manager .................................................. 17
- Automotive Simulation Models (ASM) .................... 19
- Further Products for Virtual Validation ................... 21
Virtual Validation
Benefits the entire ECU development process

Highlights

- Develop new functions and validate ECU software much earlier by using virtual ECUs (V-ECUs)
- Use identical tools throughout the entire development process
- Achieve versatility through the support of automotive standards

Virtual Validation in Brief

- Running PC-based simulations to validate, verify, and test ECU software in the form of V-ECUs
- No additional hardware is needed.
- Preparing HIL tests and scenarios on a PC
- Frontloading HIL tests to a PC
- Using V-ECUs during function development: verifying new control algorithms in the context of existing ECU software
- Covering the large amount of test runs needed for autonomous driving (AD) with cluster simulation

By using virtual validation, you can perform development, verification and validation tasks much earlier and reduce the number of additional test systems and ECU prototypes needed. This answers the need for early simulation that the automotive and aerospace industries are currently experiencing.

dSPACE tools cover all your virtual validation requirements: SystemDesk for generating V-ECUs from the ECU software architecture, VEOS for PC-based simulation, as well as software for experiments, visualization, and test automation.

Application Areas

Virtual validation can be used throughout the entire ECU software development process.

- Generate ECUs: Application Software Tests, Basic Software Integration and Continuous Integration, see p. 8
- Simulate ECUs: Tests During Development, Real-Time Independent Simulation and Virtual Bus Simulation, see p. 9
- Integrate ECUs: Virtual Test Drive, Virtual Integration Test and Co-Simulation, see p. 10
- Testing ECUs: Preparing HIL Tests, Frontloading HIL Tests and Cluster Simulation, see p. 11

Key Benefits

- You can develop and test complex new functions in a completely virtual environment instead of using expensive test benches.
- You can simulate the ECU code on a PC before a hardware prototype is available.
- You can prepare simulation models and test libraries on a PC, which reduces the preparation time at the HIL simulator.
- You can reuse the experiment software for instrumenting and controlling the HIL simulation when you run simulations on the PC.
- You can integrate flexible V-ECU generation into the continuous integration process.
Virtual ECUs

A virtual ECU (V-ECU) is software that represents a real ECU in a simulation. Unlike a soft ECU, which uses only a simplified Simulink®/Stateflow® model, a V-ECU consists of real production code. It consists of code only and does not require any special hardware. In SIL scenarios, V-ECUs are used instead of real ECUs or controller models.

V-ECUs can have different levels of abstraction, depending on what they are used for:
- V-ECUs at the application level contain selected parts of the application software, the operating system, the RTE and required parts of the basic software typically provided by dSPACE.
- V-ECUs can also include the application software and parts of the production basic software, such as Dem, NvM, and COM.

Generating V-ECUs

There are different ways to create a V-ECU, depending on what they are used for, the project needs, and on whether the development is based on AUTOSAR:

Function and software developers who only work with single components can create a V-ECU directly with Simulink or TargetLink. The result is a simple V-ECU with only a specific part of the application layer of the ECU software. It enables basic functional tests.

Software integrators who want to test a more complex network of functions can combine software components, functions or non-AUTOSAR code from different sources in SystemDesk to create the ECU’s software architecture. They can then use the SystemDesk V-ECU Generation Module to create a complete V-ECU. This includes the run-time environment (RTE) and, if needed, basic software in addition to the application layer. The V-ECUs are used for PC-based simulation with VEOS.

V-ECUs can include the complete application software and hardware-independent basic software, except modules for the Microcontroller Abstraction Layer (MCAL).
Tool Chain for Virtual Validation

All the products in the dSPACE tool chain for virtual validation interact with each other, covering every process step from generating a virtual ECU, integrating it with other V-ECUs, building and simulating an overall system including environment models, to visualizing and automating the simulation. For example, VEOS is an easy-to-use PC-based simulation platform for virtual validation. Models can be exchanged between the different simulation platforms without further modifications, and data, layouts and experiments from ControlDesk or tests from AutomationDesk can be reused. Because VEOS and SCALEXIO support automotive standards such as AUTOSAR, ASAM (XIL API, XCP), or Functional Mock-up Interface (FMI), the tools can easily be integrated into the existing tool chain, so you can continue to use all the tools you are familiar with.

Versatility by Supporting Automotive Standards

VEOS can easily be integrated into existing tool chains because it supports various automotive standards. Therefore, you can keep existing tools when you add VEOS to your rapid control prototyping or HIL tool chain to perform PC-based simulation. By using dSPACE software and hardware, you gain high flexibility and investment protection for new projects and challenges.

ASAM

In July 2009, the Association for Standardisation of Automation and Measuring Systems (ASAM) released the new XIL API standard, defining an interface for connecting test automation tools like AutomationDesk with any simulation platform, such as VEOS or SCALEXIO. The standard enables truly platform-independent test development.

AUTOSAR

AUTomotive Open System Architecture (AUTOSAR) is an open de-facto industry standard for automotive electrics/electronics (E/E) architectures. dSPACE joined the AUTOSAR partnership as a Premium Member in April 2004 and is active in defining and developing parts of the architecture and its specifications.

Functional Mock-up Interface (FMI)

The Functional Mock-up Interface (FMI) is an open standard for the exchange and integration of plant models provided by different tool vendors. dSPACE has signed the Codex of PLM Openness and works actively in the ProSTEP Smart Systems Engineering project, the Modelica Association FMI project to further develop the FMI standard, and the Modelica Association project for System Structure and Parameterization of Components for Virtual System Design (SSP). Through these activities, dSPACE gathers the necessary knowledge and insights to support our customers in projects using FMI.
Support of Functional Mock-up Interface (FMI)

Efficient Integration of Different Modeling Approaches
Compliance with FMI ensures that models created in different modeling tools can be exported as Functional Mock-up Units (FMUs) based on the FMI standard. The FMUs can readily be integrated in simulation environments with FMI support. This makes it easier to use best-in-class tools for specific modeling tasks and consistently reuse models in different development phases (e.g., virtual validation and HIL simulations) and different company departments.

VEOS also supports FMUs based on FMI for Co-Simulation. FMUs can be integrated in a comprehensive virtual validation project together with other FMUs, V-ECUs, and Simulink models. The workflow in VEOS for importing and connecting the FMUs to V-ECUs and other model interfaces is identical to the user-friendly workflow for V-ECUs and Simulink models. New modeling approaches can therefore be integrated into new or existing projects fast and efficiently. The reliable dSPACE tool chain ensures consistent simulation and parameter access in different use cases. dSPACE ensures smooth interfacing between all the tools in the dSPACE tool chain for virtual validation and HIL projects. This means you can reuse not only the real-time-capable FMUs, but also corresponding tests and experiments based on tools such as AutomationDesk and ControlDesk.

Services Concerning FMI
dSPACE also provides additional services concerning FMI to customize the FMI-based workflow according to your requirements. For example, you can make the legacy code of existing plant models available as FMUs to benefit from the advantages of FMI with the existing code base.

Example Workflow
The following illustration shows an example workflow in comprehensive virtual validation projects that use FMI. An automatic gearbox is modeled with detailed elastic and frictional behavior via a Modelica-based physical modeling approach and then exported as an FMU based on FMI for Co-Simulation. This FMU is integrated in an overall system model that includes the V-ECU representation of the automatic gearbox controller described in AUTOSAR and the Simulink-based ASM model of the vehicle dynamics. ControlDesk is used to access and monitor all the parameters and variables of the integrated system model simulated by VEOS on a standard PC.
Standard Interfaces

VEOS provides various standard interfaces so you can integrate it into an existing tool chain. The software can be coupled with any tool that supports these interfaces, for example, XCP or XIL API.

For measurement, stimulation, and calibration, VEOS supports description files commonly used in the automotive industry:

- TRC files for accessing plant models, e.g., during HIL simulation.
- A2L files for accessing V-ECUs. The same format is used for real ECUs.

This makes for a continuous tool chain that lets you use the same configuration and layouts for the calibration and automation software, no matter whether you are working on virtual validation or hardware-in-the-loop projects.
Use Case – Generate V-ECUs

**The Advantages**
- Generate V-ECUs from production code for both AUTOSAR and non-AUTOSAR ECU development
- Flexibly create V-ECUs as needed, independently of whether you are testing a new application software functionality or the complete ECU code including a third-party basic software (BSW) application
- Integrate V-ECU generation into your existing continuous integration tool chain and use V-ECUs that contain the latest release of the production code

**Application Software Tests**
As VEOS (p. 14) is a PC-based simulation platform, developers can test the latest production code directly on the computer. This makes it possible to directly validate the function code and application layer for V-ECUs without the need to integrate the complete basic software stack. The tests can be performed in open- or closed-loop test scenarios.

**BSW Integration**
You can integrate basic software and its configuration (e.g., the COM stack or modules for diagnostics) as third-party basic software in combination with the application software in SystemDesk to generate a V-ECU that contains both BSW and application software. This V-ECU can be used to test the integration of both the application and the production code of the basic software.

**Continuous Integration**
The V-ECU generation process can be integrated into existing tool chains by automating the V-ECU configuration and generation. The generation process will always provide V-ECUs with the latest version of production code. Therefore, the V-ECU can be continuously integrated. In this scenario, every check-in of new code will trigger the generation of a new V-ECU build, including the latest version of the code. The V-ECU will then automatically be tested and the results are delivered to the developer or a database, and serve as input for further automated test and integration processes.
Use Case – Simulate V-ECUs

The Advantages
- VEOS allows you to run deterministic simulations of V-ECUs faster than real time
- Connect V-ECUs on bus level, e.g., for ECUs including a COM stack, or on signal level, e.g., for application level V-ECUs or for stimulation in open-loop testing
- Break points can be set to pause the simulation for debugging purposes exactly as required
- Simulate the bus behavior for CAN, CAN FD and LIN, and use an idealized bus for FlexRay

Tests During Development
Function developers can run V-ECU simulations directly on their development PCs without the need of additional hardware. Thus, they can pause the simulation and debug the code of the V-ECU within the simulation step-by-step.

Real-Time Independent Simulation
Simulating V-ECUs does not depend on real-time execution. Scaling up the simulation to more powerful CPUs lets you run more tests in the same amount of time.

Virtual Bus Simulation
You can simulate V-ECUs including a complete COM Stack. The V-ECUs can then be connected to a virtual bus for simulating distributed functions and ECU networks without real hardware. The COM stack can either be generated automatically for CAN and LIN networks or you can include a third-party CAN or CAN FD COM stack. If the V-ECU contains diagnostic basic software, you can access the virtual bus by using ControlDesk and run virtual diagnostic tests, for example.
Use Case – Integrate V-ECUs

**The Advantages**
- Simulate Simulink models, such as Automotive Simulation Models (ASM), including engine models and traffic scenarios in VEOS
- Simulate V-ECUs with plant or environment models in the same environment
- Integrate V-ECUs and environment models from different sources in the same simulator
- Restbus models can be generated by the Bus Manager and integrated with environment models and V-ECUs
- Integrate third-party models based on FMI for Co-Simulation 2.0
- Integrate V-ECUs on different levels, whether you are testing a new application software functionality or the integration of hardware-independent software

**Virtual Test Drive**
To run a virtual test drive, model a driving scenario using ASM traffic and import it to VEOS. In VEOS, integrate the virtual ECU with a third-party engine model as an FMU into the driving scenario. The overall simulation runs completely in VEOS, so there is no need for any additional simulation hardware or software.

**Virtual Integration Test**
To test whether the integration of V-ECUs will work, implement a system model: In VEOS, you can import V-ECUs and connect them. Additionally, you can create a rest bus simulation to mimic missing ECUs to VEOS. Afterwards, integration tests can be run using the system model.

**Co-Simulation**
Integrate your VEOS simulation systems with specialized simulators such as RTMaps from Intempora for sensor simulation using co-simulation. For example, you can simultaneously run a sensor fusion algorithm in RTMaps and have both simulators communicating via a network connection within the context of the project.
Use Case – Testing and Experimenting with V-ECUs

The Advantages
- Support of relevant automotive standards, such as XCP and ASAM XIL-API and usable as an additional platform, connectable by its IP address
- Access V-ECUs that run in VEOS using XCP over Ethernet based on A2L variable descriptions

Use the same tooling for experiments and tests that you use for Hardware-in-the-Loop (HIL) systems
- Make use of scenario-based testing in your VEOS test setups

Preparing Hardware-in-the-Loop (HIL) Tests
The PC-based simulation platform VEOS has the same standard interfaces as a HIL simulator, so you can design and validate test scenarios and layouts for the HIL simulation in advance. In addition, you can develop, parameterize, and test plant models without using a HIL system.

Layouts that were previously developed for the HIL system can be reused and adapted to new requirements. Failures of the actual HIL tests due to faulty test sequences can be avoided. Layouts and tests are created once and can be reused across the different development stages to save both time and effort.

Frontloading HIL Tests
With the PC-based simulation platform VEOS, a V-ECU can be tested comprehensively on a PC in real HIL test scenarios, including automated test sequences. If necessary, two or more virtual ECUs can be combined with environment models to perform system integration tests. VEOS can simulate automated ECU test sequences on a PC that can be reused with a HIL system. Since you do not need additional hardware, you can easily distribute virtual ECUs. Frontloading tests this way lets you improve the quality of the ECU software even before starting any of the subsequent HIL tests.

Cluster Simulation
Using a PC-based simulation platform does not only offer the option to run test scenarios on a single PC, but on multiple execution modes simultaneously. This way, you can easily scale a test execution to your needs, and perform the huge amount of test runs needed, e.g., for autonomous driving.
SystemDesk V-ECU Generation Module
Generation of virtual ECUs

Highlights
- Generating simulation platform-independent V-ECUs for validation and verification
- Creating V-ECUs from external code
- Several editors and process support for V-ECU convenient configuration and generation

Application Areas
For virtual validation applications, the SystemDesk V-ECU Generation Module lets you configure and generate virtual ECUs (V-ECUs). To create a full V-ECU, software components, functions or C code from different sources can be combined in SystemDesk. To test the system’s overall behavior, V-ECUs can be used for PC-based simulation with VEOS as soon as the C code implementation is available.

Depending on your test focus, you can do the following:
- Integrate dSPACE basic software into the V-ECU generation to test the application software.
- If the BSW modules available with the dSPACE basic software do not suffice, you can use your own production BSW to test the BSW itself or functions that require realistic BSW behavior.

Key Benefits
- Guided creation of V-ECUs on the basis of the software architecture
- Support of AUTOSAR-based and non-AUTOSAR-based approaches
- Automatic configuration of dSPACE basic software for the simple and fast preparation of V-ECUs
- Automatic configuration of production BSW that can be generated with third-party BSW generator tools and used to generate V-ECUs (available for some modules)
- Integration of production BSW above the Microcontroller Abstraction Layer (MCAL)
- Automatic processes for V-ECU generation possible due to a complete automation API
- Comprehensive validation of the software architecture model for direct feedback in case of problems

Order Information

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemDesk V-ECU Generation Module</td>
<td>Generating V-ECUs for application software tests or with your own basic software</td>
<td>SYD_GEN</td>
</tr>
</tbody>
</table>
Feature Overview

V-ECUs created with SystemDesk can include components of the application and the basic software as well as functionalities comparable to those of real ECUs. The SystemDesk V-ECU Generation Module lets you generate V-ECUs for different use cases. Therefore, you can configure and generate BSW modules, e.g., the RTE, OS, the ECU State Manager or the NVRAM Manager. In order to generate V-ECUs as fast and conveniently as possible, you can choose to let SystemDesk automatically configure the BSW modules, which is often sufficient for simulation purposes. If you want to configure a module specifically to your needs, SystemDesk also offers convenient editors, for example, for mapping RTE events to tasks. You can integrate production BSW into V-ECUs if you want to test the BSW or if your application software relies strongly on the BSW. You can integrate just some BSW functionalities or the complete ECU code on the basis on the dSPACE MCAL module for VEOS. Preparing the V-ECU for connection to plant models is easy with SystemDesk, both on the RTE and the MCAL level. You can even include intervention points to insert errors for stimulating the RTE, which is much easier than in a real ECU. This makes it easy to test the application software in various error scenarios.

SystemDesk not only lets you configure and generate V-ECUs. It also supports you during the whole development process. A powerful validation functionality lets you check the AUTOSAR architecture and reports problems in the input ARXML file, enabling you to fix these problems instead of running into them later during RTE generation. And, thanks to the complete automation API, SystemDesk can also be used in automated processes for generating the V-ECUs.

Use Cases

- **Simulation at the Application Software Level**
  With SystemDesk and VEOS, you can verify and test application software in an early phase, even without considering the underlying hardware topology (virtual functional bus (VFB) mode). Thus, functional errors or error behavior in the communication between software components can be found and fixed as soon as possible. SystemDesk provides the basic software modules needed for simulation, such as RTE and OS, and offers support for conveniently configuring BSW modules. Multiple V-ECUs can be connected at the signal level and simulated together.

- **Bus Simulation**
  You can also import predefined communication matrices and generate V-ECUs including a COM stack. The V-ECUs can then be connected with each other or with a restbus model via a virtual bus. For each ECU, SystemDesk provides the required basic software components. You can use VEOS to simulate the bus communication between the ECUs. Arbitration effects, transfer times and transmission delays of bus messages or ECU and bus breakdowns can be simulated for CAN and LIN buses and verified in an early phase of development. Networks including FlexRay buses can be modeled and simulated as idealized buses.

- **Tests of Diagnostics Functions**
  You can frontload tests of diagnostics functions from the HIL simulator to VEOS. For this, the V-ECU has to include the application software as well as specific BSW modules, such as the COM stack (CAN), NvRAM, or diagnosis modules (Dem). As in HIL tests, you can use dSPACE ControlDesk to access the diagnostics functions while simulating them with VEOS.

- **ECU Integration Tests**
  The focus of ECU integration tests is to simulate and test the complete ECU code, including the hardware-independent BSW modules. By using V-ECUs, you can perform these tests on your PC and immediately debug them, if necessary. You also benefit from a continuous integration, meaning that new functions are integrated and can be tested as soon as they are available.
VEOS
Platform for PC-based simulation of models and ECU network communication

Highlights
- Early validation of ECU software by means of PC-based simulation
- Precise simulation of ECU network communication for CAN, CAN FD, and LIN buses
- Off-the-shelf integration into RCP and HIL tool chains
- Support of the relevant standards such as AUTOSAR and Functional Mock-up Interface
- Support of multi-model scenarios

Application Areas
dSPACE VEOS is a PC-based simulation platform that promotes virtual validation for the development of electronic control units (ECUs). VEOS makes it possible to simulate a wide range of different models, such as function models, Functional Mock-up Units (FMUs), virtual ECUs (V-ECUs), and vehicle models, in the early development stages independent of any specific simulation hardware.

For multi-model scenarios VEOS supports importing, connecting, and running any number of function and plant models based on Simulink or Functional Mock-up Interface (FMI), thereby extending the scope of the applications.

Key Benefits
VEOS runs on standard PCs, which gives function developers, software architects, and ECU testers a variety of new options for virtual validation during the early project phases.
- Software-software integration can be tested from component to system level.
- Sophisticated environment models can be integrated with virtual ECUs to validate complex controller strategies or simulate and test entire virtual vehicles.
- In preparation for hardware-in-the-loop simulation, models and tests can be created, set up, and run on a PC, independent of the hardware-in-the-loop (HIL) system.

Systematic Extension of the dSPACE Tool Chain
VEOS works hand in hand with other dSPACE products to provide an entire tool chain for the development and testing process. This means that tools and models that are commonly used in rapid control prototyping and hardware-in-the-loop simulation can also be used virtually. Similarly, layouts from HIL simulations can be reused in PC-based simulations with VEOS and vice versa. VEOS also provides open interfaces to connect and use existing tools.
- Simulink® and dSPACE Run-Time Target for generating C-code-based simulations
- TargetLink for generating AUTOSAR and non-AUTOSAR simulations based on production code
- SystemDesk (p. 12) for generating virtual ECUs (p. 4)
- Automotive Simulation Models (p. 19) for sophisticated environment models
- ModelDesk for graphically configuring and parameterizing environment models
- ControlDesk for experimenting and visualizing simulations and for monitoring bus communication with ControlDesk Bus Navigator
- MotionDesk for visualizing simulation scenarios
- AutomationDesk for creating tests and automating simulation runs
- SYNECT for managing test data and results
Functionality Overview

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC-based simulation</td>
<td>Simulation of heterogeneous models, from function models to virtual ECUs, bus systems, and vehicle models</td>
</tr>
<tr>
<td></td>
<td>No additional hardware needed for simulation</td>
</tr>
<tr>
<td>Simulink support</td>
<td>Simulation of Simulink implementation containers (SICs) generated from different projects</td>
</tr>
<tr>
<td>FMI support</td>
<td>Simulation of Functional Mock-up Units (FMUs) based on the Functional Mock-up Interface (FMI) for co-simulation</td>
</tr>
<tr>
<td></td>
<td>Support of FMI 2.0 functionalities and access/monitoring for all variables and parameters defined by an FMU</td>
</tr>
<tr>
<td>TargetLink support</td>
<td>Simulation of TargetLink-generated code as virtual ECUs (V-ECUs) or FMUs</td>
</tr>
<tr>
<td></td>
<td>Support for AUTOSAR as well as non-AUTOSAR TargetLink code</td>
</tr>
<tr>
<td>AUTOSAR support</td>
<td>Simulation of virtual ECUs generated by SystemDesk</td>
</tr>
<tr>
<td></td>
<td>AUTOSAR-compliant operating system</td>
</tr>
<tr>
<td></td>
<td>Support of AUTOSAR basic software modules, such as ECU State Manager</td>
</tr>
<tr>
<td></td>
<td>Development according to AUTOSAR R4</td>
</tr>
<tr>
<td></td>
<td>Support of virtual ECUs based on the AUTOSAR Adaptive Platform</td>
</tr>
<tr>
<td>Bus simulation</td>
<td>Simulation of ECU network communication on CAN, CAN FD, and LIN buses, including messages, scheduling, and arbitration</td>
</tr>
<tr>
<td></td>
<td>Optimized bus simulation for FlexRay</td>
</tr>
<tr>
<td>XIL API support</td>
<td>Support of XIL API Model Access Port</td>
</tr>
<tr>
<td>XCP support</td>
<td>Access to Simulink and TargetLink models as well as V-ECUs via XCP on Ethernet</td>
</tr>
<tr>
<td>Debugging</td>
<td>C code debugging in a running simulation</td>
</tr>
<tr>
<td>Code coverage</td>
<td>Analyzing the extent to which code has been tested with CTC++ from Testwell</td>
</tr>
<tr>
<td>Tool chain integration</td>
<td>Off-the-shelf integration into the dSPACE rapid control prototyping (RCP) and hardware-in-the-loop (HIL) tool chain</td>
</tr>
</tbody>
</table>

Order Information

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation of one or more Simulink</td>
<td>Simulation capability for executing Simulink and FMI models</td>
<td>VEOS_BASE</td>
</tr>
<tr>
<td>models or FMUs</td>
<td>The dSPACE Run-Time Target is needed for generating Simulink implementation containers (SICs) from Simulink models (Simulink Coder required)</td>
<td></td>
</tr>
<tr>
<td>Simulation of one or more virtual</td>
<td>Add-on to VEOS_BASE</td>
<td>VEOS_ECU</td>
</tr>
<tr>
<td>ECUs</td>
<td>Simulation capability for executing ECU models</td>
<td></td>
</tr>
<tr>
<td>Simulation of ECU network</td>
<td>Add-on to VEOS_ECU</td>
<td>VEOS_CAN</td>
</tr>
<tr>
<td>communication</td>
<td>Bus simulation capability</td>
<td>VEOS_LIN</td>
</tr>
<tr>
<td></td>
<td>Supported bus protocols: CAN, CAN FD, and LIN</td>
<td></td>
</tr>
</tbody>
</table>
Features and Benefits

PC-Based Simulation of Heterogeneous Models
With VEOS, you can simulate Simulink and TargetLink models, FMUs, AUTOSAR software components, virtual ECUs, and ECU networks in a single environment directly on your PC. This allows for a fast integration and validation process for your ECU software at the very early development stages. You can detect errors long before the first hardware prototype exists. Another advantage of a PC-based simulation platform is that parameters, models, and results can easily be exchanged between different kinds of user groups throughout the development process. If an error is found by software architects, integrators, or testers, it can be understood, investigated, and fixed much more easily by function developers if they can use the same simulation and testing environment.

Convenient Model Exchange
To simplify exchanging simulation models, dSPACE offers a Model Interface Package for Simulink® (MIPS) for generating Simulink implementation container (SIC) files. With the free-of-charge MIPS, modeling experts can generate the SIC file with Simulink Coder, without the need for a VEOS or ConfigurationDesk license. From the Simulink models and in combination with dSPACE Run-Time Target, they can generate code and create ZIP files that contain all the necessary code and artifacts for executing the models on different simulation platforms, such as VEOS and SCALEXIO. Model integrators that use SIC files do not have to generate code again for building the simulation, which significantly reduces the effort when using them in other projects.

Comprehensive Bus Simulation
By using VEOS, you can also simulate a network of virtual AUTOSAR ECUs. These include a realistic operating system and can be extended with basic software such as NVRAM or the ECU state manager, depending on the requirements of the simulation scenario. CAN and LIN buses and their bus-specific effects can be simulated on a PC using VEOS without any additional hardware. This gives you a precise simulation of distributed functions, including ECU network communication, very early in the development process.

Openness Through Automotive Standards
VEOS can easily be integrated into your existing tool chain, because it supports relevant automotive standards such as
- ASAM
- AUTOSAR
- FMI

When you add VEOS to your rapid control prototyping or HIL tool chain to perform PC-based simulation, you can keep your existing tools. By using VEOS, you gain high flexibility and protect your investments for new projects and challenges.
Bus Manager
Configuration for LIN, CAN, and CAN FD bus simulation

Highlights
- One configuration tool for different bus systems
- Work with several communication matrices for one configuration
- Customizable restbus configuration with tool automation interface

Application Areas
The dSPACE Bus Manager is the central tool for configuring bus communication for simulation purposes, e.g., restbus simulation, and for implementing the bus communication in real-time applications for dSPACE SCALEXIO and VEOS systems. It supports different bus systems, such as LIN, CAN, and CAN FD. The Bus Manager is available as an add-on for dSPACE ConfigurationDesk to configure hardware-in-the-loop (HIL) and rapid control prototyping (RCP) applications for dSPACE SCALEXIO hardware. You can also use a stand-alone version to configure PC-based restbus simulations with VEOS.

Key Benefits
- One tool for homogeneously configuring several bus systems at the same time
- Easy bus configuration via drag & drop
- Work with several communication matrices for one configuration
- Modeling-tool-independent model interface
- Tool automation interface
- Consistent bus simulation for CAN and LIN with VEOS

Workflow
The Bus Manager offers a convenient and straightforward workflow for implementing bus simulations. In general, the same workflow applies to virtual scenarios with VEOS and real-time systems with SCALEXIO:
- Import one or more bus communication matrices. All relevant information is extracted automatically for the subsequent bus configuration.
- Create a bus configuration. For the configuration, different views for the dedicated tasks are available.
- Assign the communication matrices completely or in part to the bus configurations.
- Specify the real-time hardware access (this step only applies to RCP and HIL scenarios).
- If required, you can configure different parameters and properties of the simulated elements. For example, you can enable the access to signal values during run time via experiment software such as ControlDesk. If the simulation requires signals whose values must change dynamically during run time, you can use behavior models, e.g., MATLAB®/Simulink® behavior models or Functional Mock-up Units (FMUs) to use behavior models designed in another modeling tool.
- Finally, start the build process, and download (applies to RCP and HIL scenarios) and execute the real-time application.
Functionality Overview

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Description</th>
</tr>
</thead>
</table>
| General | - Import of several communication matrices, such as AUTOSAR system description files, FIBEX, DBC, and LDF files  
- Versatile communication matrix visualization with different views, folding, filtering, searching  
- Easy restbus configuration across communication matrices via drag & drop  
- Selectable and customizable restbus configuration view  
- Tree view and property grid for bus-element-based configuration  
- Table view for multiple-element configuration  
- Configuration of the bus hardware  
- Modeling-tool-independent model interface (model port blocks)  
- Support of Functional Mock-up Units (FMUs)  
- Convenient update of existing configurations with new communication matrices |
| LIN | - Send and receive of unconditional, event-triggered and sporadic frames/PDUs  
- Generation of LIN schedules |
| CAN | - Triggered/cyclic send and receive of CAN frames/PDUs  
- Support of multiplexed PDUs |
| CAN FD | - Triggered/cyclic send and receive of CAN FD frames/PDUs  
- Support of multiplexed PDUs |
| Bus simulation | - Export of bus configurations as bus configuration containers for import into VEOS |

Order Information

<table>
<thead>
<tr>
<th>Product</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus Manager (add-on for ConfigurationDesk)</td>
<td>BUS_MANAGER_BASE</td>
</tr>
<tr>
<td>Bus Manager (stand-alone version for use with VEOS)</td>
<td>BUS_MANAGER_BASE</td>
</tr>
<tr>
<td>Bus Manager (add-on for Bus Manager for inspection and manipulation functions)</td>
<td>BUS_MANAGER_PLUS</td>
</tr>
</tbody>
</table>
Automotive Simulation Models

Engine, Vehicle Dynamics and Virtual Vehicle Simulation

Highlights

- Open MATLAB®/Simulink® models
- Real-time simulation and offline simulation
- Engine models with torque generation based on mean-value or thermodynamic approaches
- Vehicle dynamics with road, maneuvers, and driver plus optional traffic and trailer simulation
- Intuitive graphical parameterization, and road and maneuver creation in ModelDesk

Application Areas

The Automotive Simulation Models (ASM) are open Simulink models for the real-time simulation of key automotive systems such as engines and vehicle dynamics plus extensions such as brake hydraulics, turbocharger, electrical system, exhaust system, traffic and trailer. A virtual vehicle can be created by combining models. The ASMs are typically used on a dSPACE Simulator for hardware-in-the-loop testing of electronic control units (ECUs) or during the design phase of controller algorithms for early validation by offline simulation.

Key Benefits

All the Simulink blocks in the models are visible, so it is easy to add or replace components with custom models to adapt the properties of modeled components perfectly to individual requirements. ASMs standardized interfaces make it easy to expand a single model, for example, engine or body, or even create a whole virtual vehicle. Roads and driving maneuvers can be easily and intuitively created using graphical tools with preview and clear visualization.

Vehicle Model Characteristics

The actual physical vehicle characteristics are represented by a multibody system with 24 degrees of freedom. It consists of a drivetrain with elastic shafts, a table-based engine, two semi-empirical tire models, a nonlinear or table-based vehicle multibody system with geometrical suspension kinematics and aerodynamics, and a steering model. An environment with a road, maneuvers, and an open- and closed-loop driver is included as well. All parameters can be altered during run time. The included brake hydraulics model consists of a dual-circuit hydraulics system.

Engine Model Characteristics

Combustion models with mean-value and thermodynamic approaches are available. They offer crankshaft-angle-based torque generation, dynamic simulation of the air path, and several injection models. To simulate the engine in a vehicle system in a closed control loop, the engine models have a drivetrain with longitudinal dynamics and either manual or automatic transmission, plus models for the driver and the soft ECUs. The engine models can be expanded by a physical turbocharger model.
ASM Tool Suite

Combining Simulation Models
The dSPACE Automotive Simulation Models consist of various packages and libraries for specific application areas. They can be combined as needed to create the simulation model for a specific project.

Parameterization and Visualization
For parametrizing the simulation models, dSPACE ModelDesk is used. The graphical user interface also provides project handling and allows parameter sets to be downloaded for offline and online simulations. During the simulation run itself, dSPACE MotionDesk visualizes the simulation in a virtual world that exactly represents the simulation scenario.
## Further Products for Virtual Validation

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
</table>
| TargetLink       | - Production code generation software  
|                  | - AUTOSAR software components developed with TargetLink can be exported to SystemDesk and integrated into virtual ECUs there.  
|                  | - Offers a direct link to VEOS for simulating the generated production code as a virtual ECU in offline simulation scenarios.                                                                             |
| ConfigurationDesk| - Configuration and implementation software for dSPACE SCALEXIO hardware  
|                  | - Configuring real-time applications graphically  
|                  | - Managing signal paths between external devices (like ECUs or loads) and behavior model interfaces  
|                  | - Implementing behavior model code and I/O function code on dSPACE hardware  
|                  | - Managing multicore applications and importing virtual ECUs                                                                                                                                         |
| ModelDesk        | - Software for parameterizing the Automotive Simulation Models graphically  
|                  | - Parameterization during online (simulator) and offline (Simulink) simulations  
|                  | - Managing parameter sets and entire projects                                                                                                                                                          |
| ControlDesk      | - Universal, modular experiment and instrumentation software for accessing simulation platforms, such as VEOS and SCALEXIO  
|                  | - Experimenting and visualizing the simulations  
|                  | - Access to all variables of the virtual ECUs and environment models  
|                  | - Powerful layouting, instrumentation, measurement and postprocessing  
|                  | - Failure simulation model for controlling failure insertion units in real time  
|                  | - Monitoring bus communication  
|                  | - Integrated ECU calibration, measurement and diagnostics access (CCP, XCP, ODX)                                                                                                                                 |
| AutomationDesk   | - Environment for powerful and convenient test automation  
|                  | - Automation of virtual ECUs tests  
|                  | - Connects to VEOS and SCALEXIO simulation platforms via XIL/HIL API                                                                                                                                     |
| Real-Time Testing| - Real-time tests are executed synchronously with the simulation model  
|                  | - Model variables can be observed and changed in every simulation step  
|                  | - Test programming via standard Python scripting language                                                                                                                                              |
| MotionDesk       | - Visualizing driving maneuvers for vehicle dynamics simulation  
|                  | - Perfectly suited for visualizing ADAS scenarios, with a new rendering engine and features like rainfall and snow                                                                                       |
| SYNECT           | - Systematically planning and controlling test execution  
|                  | - Managing test data and scenarios                                                                                                                                                                       |