www.dspace.com

dSPACE MAGAZINE

3/2015

CLAAS

Assistance Systems Optimize the Harvest | page 6

ZF TRW – Validating Steering Systems with a Simulator and Test Bench | page 12

Rimac Automobili – Concept_One: Completely Electric Supercar | page 18

User Feedback

"The Powertrain department of Jaguar Land Rover has used dSPACE Simulators in the hardware-in-the-loop (HIL) lab for many years. I personally have over 7 years' experience in dealing with and using dSPACE technology. We find dSPACE Simulators to be very flexible, easy to configure, and robust in operation. My team extensively uses dSPACE software, namely dSPACE ControlDesk and AutomationDesk. Overall we receive good pre- and post-sales support. This is critical, as the Powertrain test facilities have to support the engine management, transmission and driveline programs for Jaguar Land Rover cars. dSPACE plays an important role in providing simulation technology for our business."

Dr. Nancy Liu, HIL team leader for EMS, Transmission and Driveline, Jaguar Land Rover Limited

Photo credits: © Jaguar Land Rover

"With ASM, we were able to achieve unparalleled versatility."

An important factor for dSPACE's success has always been combining different disciplines while keeping the control engineer's interdisciplinary view of things. Some might have been surprised when we released our own line of mathematical models exactly 10 years ago, the Automotive Simulation Models (ASM). Before that, we successfully integrated third-party models in hardware-in-the-loop (HIL) simulators. But modeling is a dynamic task, and when putting HIL test benches into operation, one should also have access to the inner workings of the models to be able to work efficiently.

We first took on vehicle dynamics and drivetrain models. Our goal was to offer high-quality models for HIL simulation: models that are real-time-capable, integrated, can be extended by the customers themselves, and are so precise that the ECUs do not notice any difference to being used in the vehicle. We did not aspire to achieving the extreme precision required for design and dimensioning. Our ambition was to reach and even surpass the best models from specialized providers, for vehicle dynamics and drivetrains alike. We are confident that we have achieved this goal and, most of all, offer unparalleled versatility. The dSPACE models are no longer defined only for vehicle dynamics and drivetrains. Today, they also simulate electric motors, electric components and complex traffic environments. What is more, they are no longer restricted to HIL simulation: ASMs are also being used offline, for example, together with the PC-based simulation platform dSPACE VEOS.

The most recent developments are in the area of electromobility, driver assistance systems, and Car2Car communication. Just enter the words 'dSPACE' and 'ASM' in the search field of a well-known video portal and you will find an abundance of demos showing you all that can be done with ASMs. We are proud that our birthday child has replaced other models at many customers, even for very complex and demanding applications. Happy birthday, ASM!

Dr. Herbert Hanselmann

Milestones 10 years ASM:
www.dspace.com/go/dMag_20153_ASME

IMPRINT

dSPACE MAGAZINE is published periodically by:

dSPACE GmbH · Rathenaustraße 26
33102 Paderborn · Germany
Tel.: +49 5251 1638-0 · Fax: +49 5251 16198-0
dspace-magazine@dspace.com · www.dspace.com

Responsible in terms of press law:
Bernd Schäfers-Maiwald
Project Manager: André Klein

Authors:
Thorsten Bödeker, Michael Lagemann, Ralf Lieberwirth, Sonja Ziegert, Thomas Pöhlmann, Julia Reinbach, Dr. Gerhard Reiß

Co-Workers on this issue:
Dr. Ulrich Eisemann, Gregor Hordys, Doreen Krob, Carsten Rustemeier, Thomas Sander, Tino Schulze

Editors and Translators:
Robert Bevington, Stefanie Bock, Anna-Lena Huthmacher, Dr. Michelle Kloppenburg

Design and Layout:
Jens Rackow, Sabine Stephan

Print: Media-Print Group GmbH, Paderborn

Cover photo © CLAAS

Contents

Customers

Products

Business

Compact News

dSPACE on Board

Automated assistance functions for more harvesting efficiency

Controlled Harvesting

A new assistance system for CLAAS combine harvesters permanently monitors the harvesting process and automatically adjusts the machine settings to the current conditions – faster and more exactly than an operator ever could. This automated optimization is based on a distributed control system. dSPACE's production code generator, TargetLink, supports the developers in this complex project.

The annual harvest season has just a very small time slot. Wheat, rye, barley, and corn have to be harvested at just the right point of maturity. Once this harvest maturity has been reached, the combine harvesters work the fields day and night. But running a combine is a highly complex task. Up to 50 parameters from the reel to the chopper influence the harvest yield. The operator has to continuously monitor and evaluate around a dozen of these parameters. Hardly any operator is capable of keeping an eye on everything and tapping the machine's full potential.

Less Complexity on the Field

The high number of settings and parameter dependencies is the result of many different environmental factors, such as climate and terrain, and basic objectives concerning throughput, fuel consumption, and threshing quality. This makes optimizing the harvesting process very complex. Operators can barely master the challenge of choosing settings and checking the displayed values time and time again. To take off some of their burden so they only have to enter the most important settings, some functions can be partially automated. The new assistance system CEMOS AUTOMATIC (CLAAS Electronic Machine Optimization System) tackles the challenge of optimizing the crop yield. CEMOS monitors the harvesting process, regulates the process parameters and continuously adjusts the machine to the harvesting conditions. Its key to success is online modeling: The assistance system permanently computes the machine and environment model, analyzes the parameters, and determines the optimal parameter set. It then passes these parameters to the machine.

Automated Harvesting

Before the combine is on the field, the driver enters the harvest objectives via a graphical dialog-based user interface. CEMOS AUTOMATIC analyzes the objectives, sensor data, and machine settings and very quickly determines the optimal combination of parameters. The harvesting conditions change throughout the day, so the system checks these optimal settings time after time and adjusts them continuously. This means that CEMOS AUTOMATIC makes permanent readjustments to an extent that operators could never perform on their own. These automated parameter adjustments yield optimal results, such as maximum throughput with the highest grain quality, grain cleanliness, and minimum fuel consumption. In combination with other assistance systems, such as the CRUISE PILOT for controlling driving speed and the LASER PILOT for steering, a combine becomes fully automated.

CEMOS AUTOMATIC with the subsystems AUTO SEPARATION, AUTO CLEANING, and CRUISE PILOT.

ECU System for Optimal Harvesting

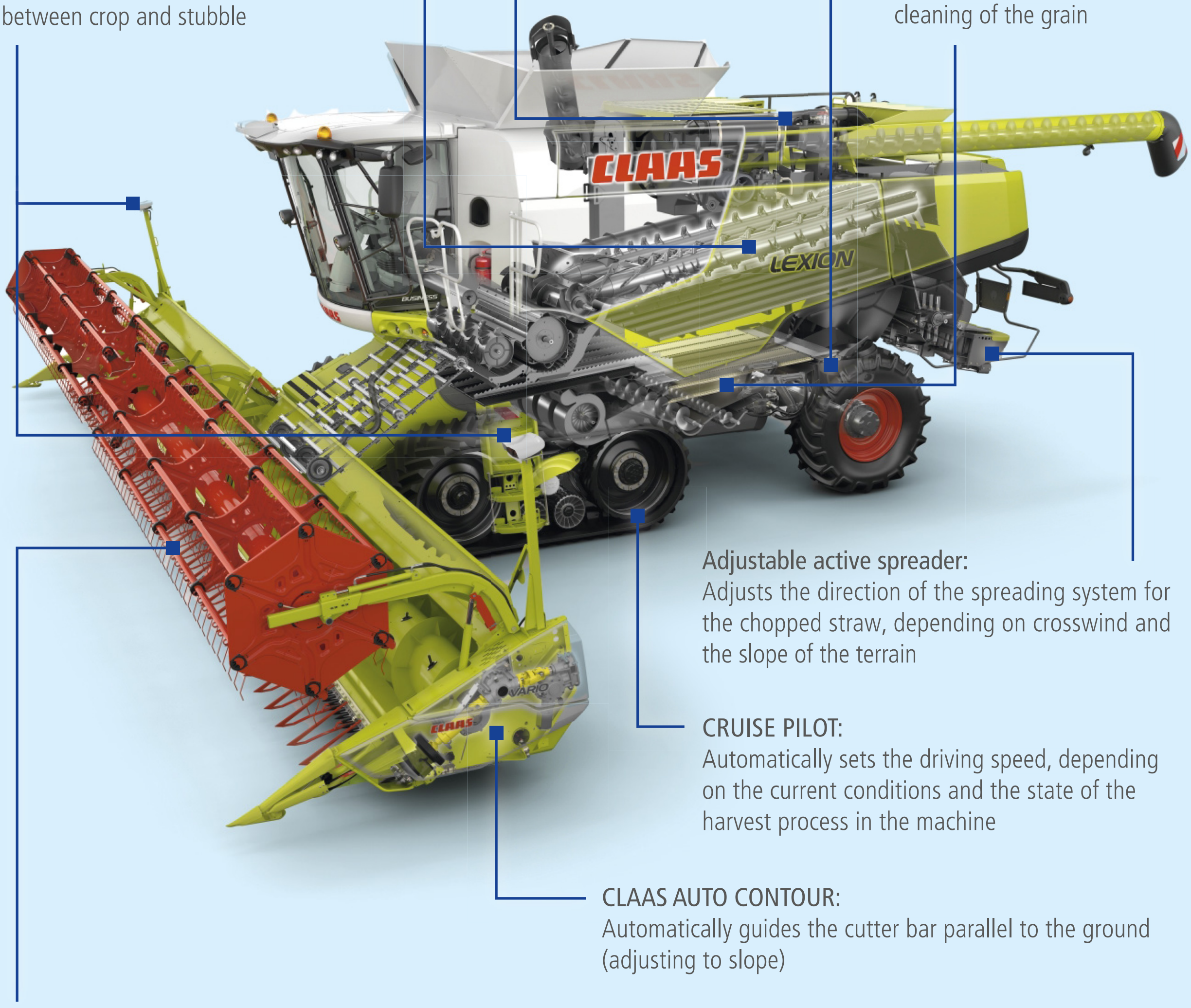
Optimal harvests are reached when all of the combine's systems interact in coordination. This is done by an automatic mechanism that works on a level above the ECUs of the individual systems. The ECU network consists of control units for basic control tasks and superordinate ECUs. A fully equipped LEXION 780 combine harvester includes 35 ECUs that are connected via CAN bus. For the systems directly involved in the harvesting process, CLAAS develops the ECUs itself. Purchased systems, such as the combustion engine, are equipped with ECUs from their suppliers. Depending on the task, ECUs with processors based on fixed-point or floating-point arithmetic are inserted. The ECU of CEMOS AUTOMATIC is designed for a 32-bit PowerPC.

Model-Based ECU Software Development

ECU software development is model-based. All of the large control units are developed with MATLAB®/Simulink® and dSPACE TargetLink®. The neces-

>>

Assistance Functions Developed with TargetLink



DYNAMIC COOLING:
Automatically sets the combine’s cooling system (for diesel motor and the hydraulics system), depending on the required cooling power

MONTANA:
Automatic chassis that swings the axle portals to compensate cross tilt up to 17% and longitudinal tilt up to 6% for operation on steep terrain

CEMOS AUTO SEPARATION:
Automatically adjusts the residual grain separation

CLAAS LASER PILOT:
Uses a laser scanner to automatically steer the machine along the edge between crop and stubble

CEMOS AUTO CLEANING:
Automatically adjusts the cleaning of the grain

Adjustable active spreader:
Adjusts the direction of the spreading system for the chopped straw, depending on crosswind and the slope of the terrain

CRUISE PILOT:
Automatically sets the driving speed, depending on the current conditions and the state of the harvest process in the machine

CLAAS AUTO CONTOUR:
Automatically guides the cutter bar parallel to the ground (adjusting to slope)

Automatic reel speed control:
Synchronizes the reel's circumferential speed with the machine's driving speed (2001: pilot project for TargetLink, 5 days of development time from the available hydraulics components to integrating the first prototype into the ECU)

Basic features:
- Various speed settings
- Various position settings

The graphical user interface makes entering objectives easy.

>> sary machine functions are designed according to the principle of a distributed automation represented by one overall model. This complex model has a size of 50 megabytes. Task control and the communication between the system functions are based on the OSEK (Open Systems and the Corresponding Interfaces for Automotive Electronics) operating system. The OSEK module for TargetLink is used to define interfaces and tasks. The developed assistance function therefore only has to be connected to the existing environment.

Code Generation and Offline Testing

After the model-based development of new functions, the code can be generated for the controllers either incrementally or completely. It is possible to generate code only for individual functions (incremental code generation) or for the overall controller function (complete code generation). TargetLink supports different simulation modes so the new functions can be tested as soon as possible. This validation phase before the harvest is extremely important. When harvest season is underway, the development team does not have any time to look for implementation errors while on the field. The harvesting needs to start with well-tested software. The validation phase includes integration tests and also intensive functional validation, which uses complex plant models. The offline test scenarios use the large amount of data that is collected on the field during each harvesting phase.

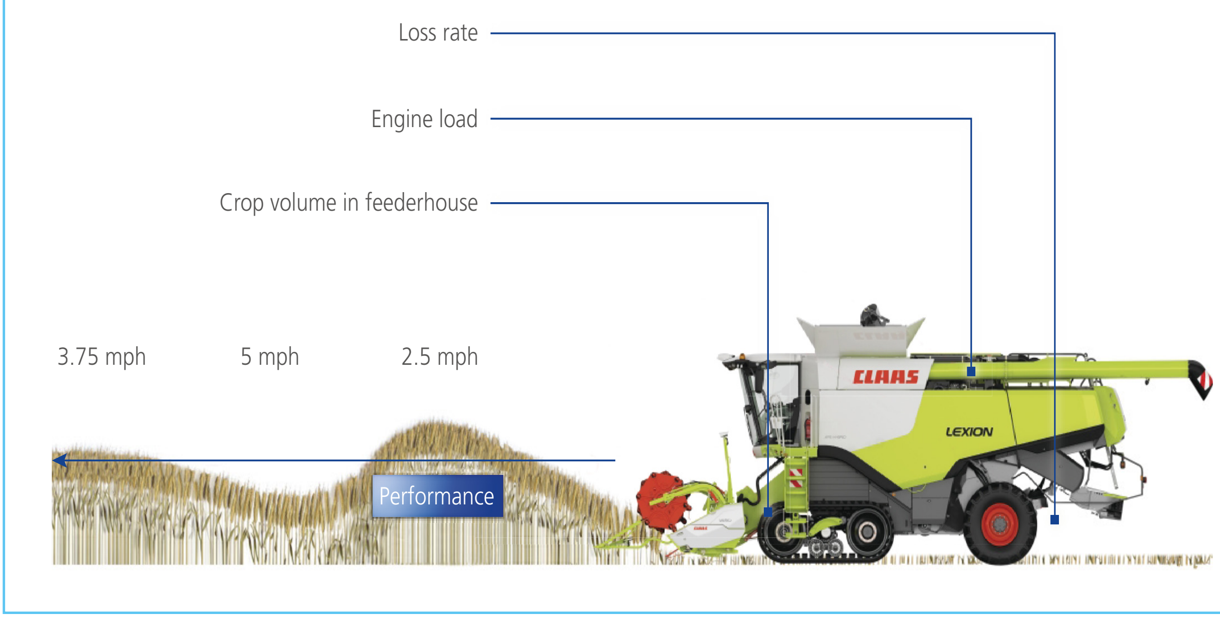
Virtual ECUs

With the CLAAS online simulator, the behavior of the combine can be tested prior to the harvest and under different conditions. The operators can even use the simulation scenarios to get used to the combine or to polish up and improve how they handle it. PC-based simulation makes this possible. The online simulator uses the virtual representation of the different machine components and the process model that contains the data from many years of harvesting experience. Real-time-capable virtual ECUs run in the background, with software that matches the real ECUs. These possibilities for simulation before the harvest is another important way to minimize the risk of machine damage and operating errors during the harvest.

TargetLink Evaluated

For years, CLAAS has successfully developed basic functions with TargetLink. These functions include an automatic reel speed control – the first pilot project using TargetLink. Even back then, the short development time of only 5 days for going from the available hydraulics components to integrating the first prototype function into the ECU was astonishing (TargetLink Goes to the Fields, dSPACE NEWS 2001/2). Because the ECU systems of combines are becoming increasingly complex, their development no longer focuses only on easy handling and a steep learning curve but also on other aspects. This made it easy to integrate the functions for CEMOS AUTOMATIC into the model and the complex ECU network. It is still important that the software can be tested in the laboratory with TargetLink's own instruments. This way, functions can be validated very early on. Even special functions, such as defining multi-rate tasks or background tasks, can be described

> “New assistance functions make our combine harvesters much more efficient. For their development we rely on dSPACE’s production code generator, TargetLink."
>
> *Andreas Wilken, CLAAS*



Harvesting process with environmental influences, driving behavior, and machine parameters.

and implemented precisely with TargetLink. For CLAAS, the combination of TargetLink and the OSEK module means that developers can concentrate on the essential development tasks. Despite the complexity of the model, efficient code can be generated quickly with TargetLink, both for individual functions (incremental code generation) and for the entire ECU network.

CEMOS AUTOMATIC Assistance System

The CEMOS AUTOMATIC assistance system is a shining example of how important software is in the commercial vehicle sector. The system first went into series production as an optional feature for the LEXION 740-780 series of combine harvesters. Implementing and testing the new function became easy with model-based development and production code generation. The generated code is reliable and error-free and lets the developers focus on the most important task during their short and valuable time on the machine: the final function test on the fields. ■

Andreas Wilken, CLAAS

Conclusion and Outlook

There is a steady increase in the demands on modern combine harvesters regarding efficiency and the economical use of fuel. Assistance systems such as CEMOS AUTOMATIC (CLAAS Electronic Machine Optimization System) are a tried-and-tested solution and are becoming more and more important. The production code generator TargetLink is a firmly established part of the development process for such assistance systems. TargetLink's easy-to-handle functions support the development of complex control systems and make it possible to generate reliable production code. For learning purposes, the TargetLink models are also used on virtual ECUs on a PC so the operators can become familiar with the combine before the harvest. Tools like TargetLink are the ideal basis for spurring innovation. In the future, new standards such as AUTOSAR will lay the groundwork for developing distributed controllers and easily reusing software.

Andreas Wilken
Andreas Wilken works in predevelopment in the area of function systems at CLAAS Selbstfahrende Erntemaschinen GmbH in Harsewinkel, Germany.

Testing Steering Systems

Real-time-based testing at ZF TRW Steering Development

Developing and validating safety-critical mechatronic systems such as power steering requires a suitable test environment. ZF TRW Tech Center in Duesseldorf uses test instances with increasing hardware content.

Also entire ZF TRW steering systems are tested in a real-time environment.

The demands on the functional performance and safety of today's steering systems are continuously increasing – namely, functional performance in the context of autonomous driving and safety in relation to the stricter requirements for steering support availability. Both aspects have to be integrated into a validation process that complies with ISO 26262. Furthermore, a comprehensive validation procedure also needs to consider aspects arising from the high number of vehicle platform variants. Proof of series maturity has to be delivered as soon as possible and at competitive costs.

> "The openness of the dSPACE environment gives us decisive advantages for implementing our own models and using self-developed test bench components."
>
> *Dr. Michael Moczala, ZF TRW*

Virtualization as the Solution

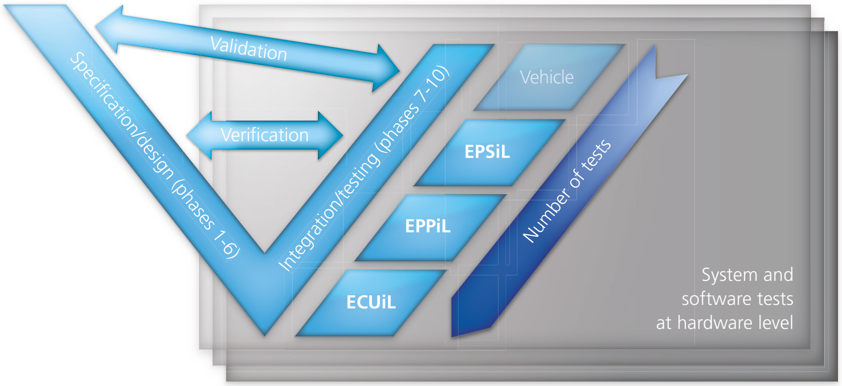
To tackle these challenges the ZF TRW Tech Center in Duesseldorf uses virtualization technology and Hardware-in-the-Loop simulators. This means that real components (hardware) are used in a control loop that is closed by virtual components. In particular, these virtual components include precise models of the steering sensor, the steering gear, the driver's arms and vehicle communication. These specialized submodels can easily be combined with the components of the Automotive Simulation Models (ASM) Library from dSPACE. This lets you simulate vehicle dynamics realistically and in real time. The real-time capability of ASM enables the testing of ECU hardware in a virtual vehicle environment. This makes it possible to perform model-based testing when the real systems or components are not available yet. Errors and failures can be simulated without any risk or damage to the product, and automating tests becomes both easier and more efficient. A further benefit is that tests are easy to reproduce and are independent of weather conditions, which is important for in-vehicle tests.

Three Test Instances

On the basis of the system integration strategy, ZF TRW set up three HiL test instances with different hardware-to-model ratios: ECUiL, EPPiL and EPSiL. The three abbreviations stand for: Electronic-Control-Unit-in-the-Loop, Electric-Power-Pack-in-the-Loop and Electrically-Powered-Steering-in-the-Loop. In the EPP, the ECU and the electric motor together form a functional unit. The EPS then combines the ECU, the motor, a steering sensor, and mechanical power transmission elements to form one steering system. The amount of hardware used in the tests increases from one test bench to the next, while the amount of modeled components decreases. The test benches are then used successively according to the ISO 26262 test phases (figure 1). This means that the extensive tests of the early validation phases are carried out cost-effectively on the ECUiL simulators, of which there are several. The test benches dedicated to higher integration levels have higher acquisition and operating costs.

>>

Figure 1: The three test benches are used for the ISO-26262-compliant functional validation of the steering algorithms at the hardware level – verification via bidirectional traceability. The validation steps require additional information from superordinate specification phases.



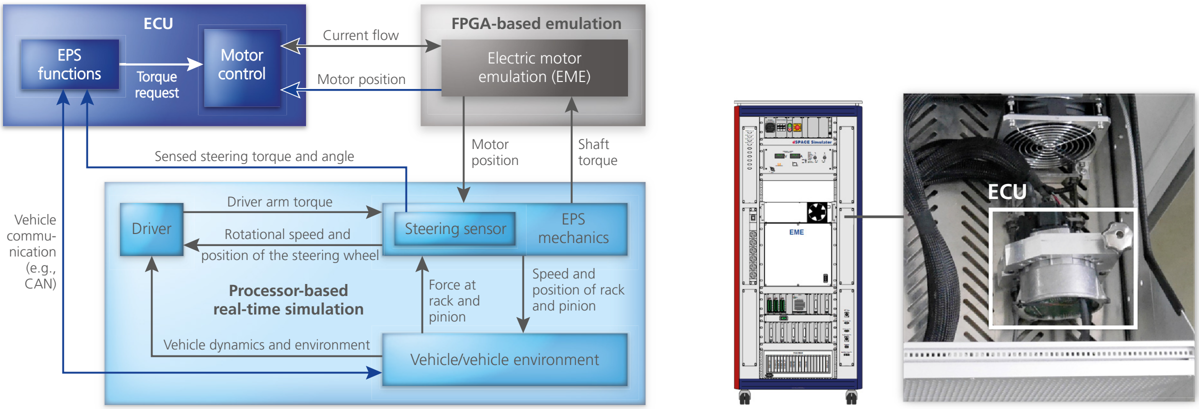


Figure 2: The number of virtual components is largest in the ECUiL systems. The ECU under test is electrically connected to the simulation environment.

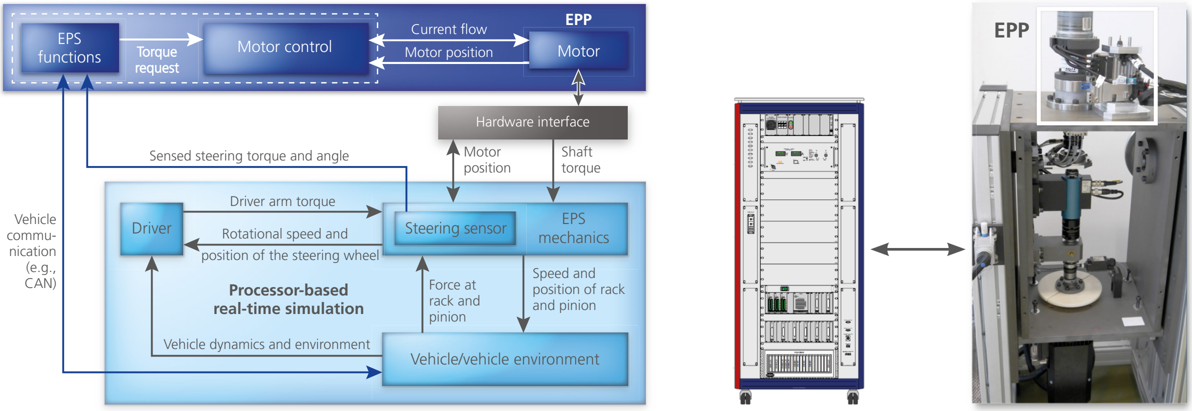


Figure 3: The EPPiL system contains the ECU and the steering motor as hardware. An actuator and sensors are used to integrate the motor into the simulation environment.

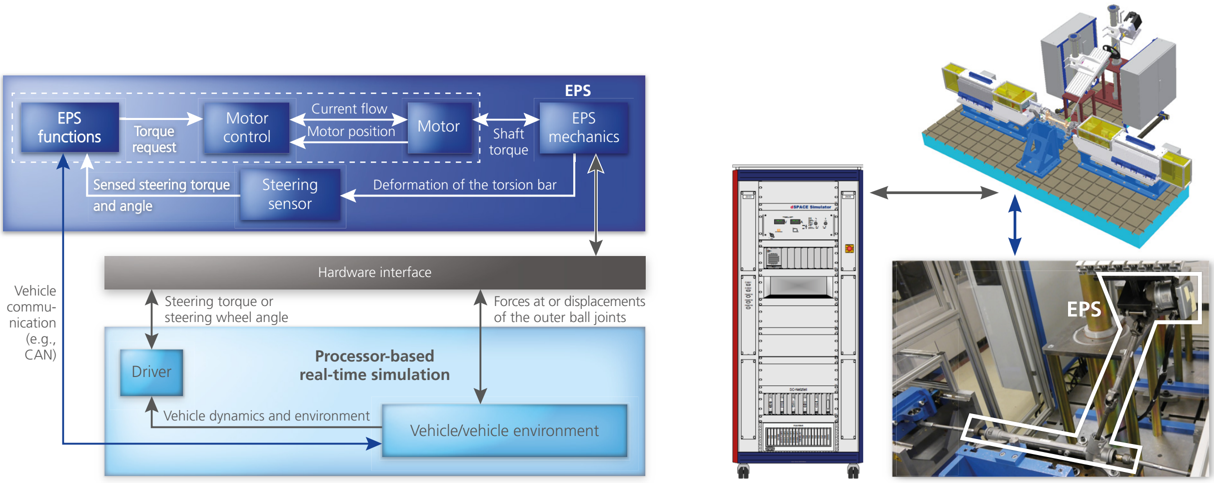


Figure 4: The EPSiL simulator tests the entire steering system. The mechanical interface to the model is therefore complex.

> “The flexibility of dSPACE’s real-time systems is impressive. We use our HiL infrastructure for application projects and predevelopment.”
>
> *Dr. Michael Moczala, ZF TRW*

On the other hand, the higher the degree of integration, the lower the amount of testing.

Instance 1: Electronic-Control-Unit-in-the-Loop

The ECU used in the steering system contains the numeric algorithms of the steering functions and the actual controller of the electric motor (figure 2). The motor generates a torque that is adjusted to the driving situations and to the driver requests. Simulating this interaction requires the electrically precise emulation of the motor in such an ECUiL test environment. The required precision can be achieved by the electric motor emulator (EME), which is integrated into the dSPACE real-time system. With this solution, the developers and testers also have the ability to adjust the motor characteristics. They can access the motor parameters, the measured signals, and all other data of the control model via integrative dSPACE software.

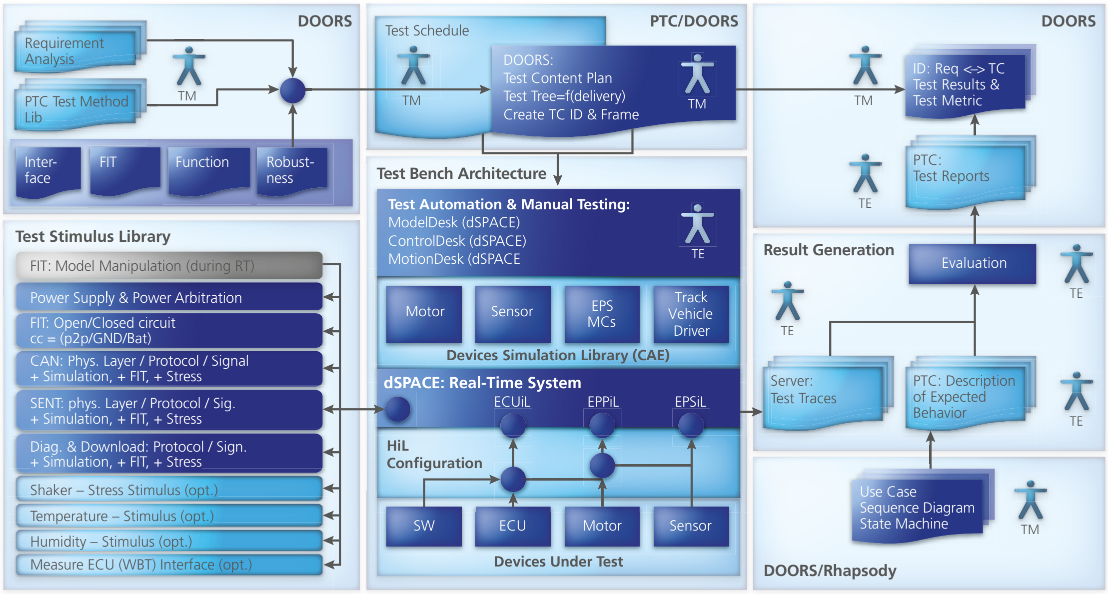
Instance 2: Electric-Power-Pack-in-the-Loop

For EPPiL testing, the ECU is connected to the real steering motor (figure 3). The motor interacts with a rotary actuator according to the simulated load case. The matching sensors, which are a part of the hardware interface, provide the actual state, and the actuator’s inverter receives the target values generated in the control model. The HiL loop is then closed by virtual components, similar to the ECUiL systems.

Instance 3: Electrically-Powered-Steering-in-the-Loop

The EPSiL simulator contains the fewest virtual components (figure 4). However, providing the mechanical interfaces of a complete steering system to the model requires a large number of active elements and measurement devices. For example, the driver torque and the rotational angle of the steering wheel are controlled by a direct rotary actuator. The linear actuators, which are connected to the two tie rods, are used as direct electric drives as well. The close connection between real-time hardware and drive controllers via the TWINsync protocol guarantees the high-precision and dynamic transfer of the control variables to the steering hardware under test. Measurement values from several force, torque, position and acceleration sensors feed the steering states back into the real-time model and close the control loop. The EPSiL simulator is a one-stop solution from dSPACE.

Figure 5: EPS HiL tool architecture using dSPACE tools that provide a complete solution for editing and managing test artifacts.



TM: test manager, TE: test engineer

> "The open dSPACE architecture – real-time hardware and software tools – enabled us to map our integration strategy to the HiL infrastructure."
>
> *Thomas Maur, ZF TRW*

Flexible Test Environment

The control environment of all three HiL test benches is designed so that it simulates not only complex interactions between driver, vehicle and road in a virtual environment, but also offers the possibility for simple force and trajectory specifications at the related hardware interfaces. Synthetic stimuli or recorded measurement data can be fed to the unit under test via the hardware interface. The modular character of the real-time systems creates a flexible environment. For example, only little effort is required to connect the real-time systems to existing test benches.

Summary and Evaluation

To validate the electrically powered steering (EPS) systems, ZF TRW uses a test concept that consists of several HiL instances, from ECU testing (ECU-in-the-Loop) to testing the entire steering system on a test bench (EPS-in-the-Loop). In combination with the HiL simulators' hardware, the Tech Center in Duesseldorf created a complete data and software infrastructure that can be accessed by all HiL test benches. Models, user layouts and test automation scripts are developed with the entire system in mind. One great advantage at this place is the seamlessness and flexibility of the dSPACE products. The combination of the dSPACE hardware with tools such as ControlDesk® Next Generation, AutomationDesk, ASM or ModelDesk gives developers and users maximum flexibility. The open setup of the real-time application, which is described in MATLAB®/Simulink®, makes it possible to extend it with specialized submodels. These detailed models let developers master the demanding development and test tasks of the EPS systems. The data management system SYNECT® completes the software infrastructure and provides the required interfaces to the test management in PTC® Integrity and the requirements management in DOORS® (figure 5). The HiL infrastructure makes it possible to develop and test EPS systems according to the ISO standard for the functional safety of road vehicles efficiently and reliably. The test instances (ECUiL, EPPiL, and EPSiL) reflect the integration strategy defined by ISO 26262. ■

Dr. Michael Moczala,
Thomas Maur, ZF TRW

> "By integrating SYNECT in our dSPACE-based HiL setups we closed the gap between requirements management and testing. 'Testing between DOORS & DOORS' is our new motto."
>
> *Thomas Maur, ZF TRW*

Dr. Michael Moczala
Dr.-Ing. Michael Moczala is CAE Specialist at ZF TRW Active & Passive Safety Technology in Duesseldorf, Germany.

Thomas Maur
Dipl.-Ing. (FH) Thomas Maur is Head of System Integration & Testing department at ZF TRW Active & Passive Safety Technology in Duesseldorf, Germany.

Concept_One

Insights into the drive concepts of a purely electric supercar

The Rimac Automobili Concept_One was designed from the ground up to be a fully electric supercar – the first of its kind. Not impressed? How does 1088 hp and 4 independent electric motors sound? All this power is kept in check by a MicroAutoBox.

The story behind Concept_One is unique. It is the brainchild of Mate Rimac, a young engineer and inventor from Croatia. He is now 27 years old and employs over 80 people in Rimac Automobili, the Croatian company that designs, engineers, builds, and sells high-performance electric vehicles and technologies around the world. The Concept_One was introduced in Frankfurt in 2011. It is the first electric supercar in the world, with staggering performance figures.

> "dSPACE ControlDesk has proven invaluable in real-world testing conditions."
>
> *Kruno Hrvatinić, vehicle dynamics engineer, Rimac Automobili*

Powertrain Design and Concept

Its drivetrain is what makes the Concept_One so special. Rimac Automobili started with a blank sheet of paper and examined each potential component closely in an effort to decide which one could have a useful place in the car and which would just add weight. Since a four-wheel drive is the only way to take advantage of a full tire grip, all four wheels are driven. Several powerful electric motors provide more power and weigh less than one large electric machine, so each wheel has its own motor. "This also eliminated the need for a traditional clutch and differential, but we decided the gearbox needed to stay. This makes the Concept_One the only electric car with a two-speed gearbox for each of the rear wheels, allowing the car to achieve amazing acceleration and still reach a top speed of 325 km/h," states Mr. Kruno Hrvatinić from Rimac Automobili's vehicle dynamics team.

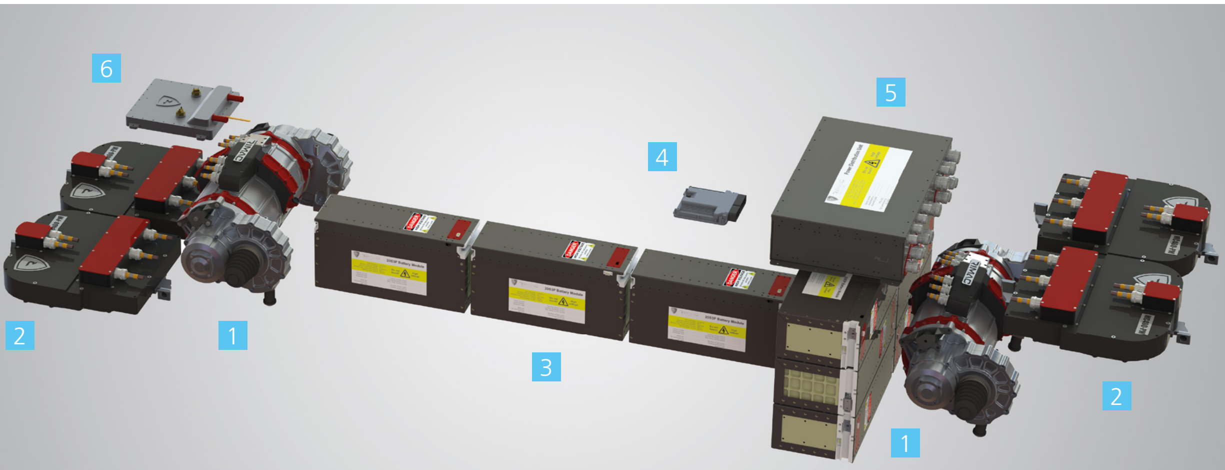
Performance Package with Synchronous Motor and High-Voltage Battery

"We opted for two pairs of permanent magnet synchronous motors, each front motor capable of delivering up to 330 Nm of torque and each rear motor up to 440 Nm – a combined 1.540 Nm. There are two proprietary Rimac Automobili motors in each housing, sharing their cooling system to save space and weight. Each motor has its own gearbox, single-speed for the front and two-speed dual-clutch for the rear. At full power, the front motors can draw up to 400 kW of power and the rear up to 600 kW, which adds up to a full megawatt of battery power," says Mr. Hrvatinić. The battery pack consists of several hundred lithium-ion battery cells operating at 650 V. At full power, the motors draw nearly 1600 A of current. But when driving with a range-focused power consumption setting, the battery pack's 82 kWh of energy is estimated to last for about 330 km. The cell voltages and temperatures are managed by the Rimac Active Battery Management System, developed fully in-house by Rimac Automobili and controlled from the central vehicle control unit via CAN bus communication.

Why Go Electric?

The best thing about a drivetrain with four independently controllable motors is the degree of freedom such

The 'drivetrain' consists of two motor units with transmissions (1), inverters (2), and a battery pack (3). Further components: central controller (4), power distribution unit (5), and heating (6).



a drivetrain allows. A gasoline-powered car must rely on a differential to transmit torque from a central source of power (the engine) to each of the wheels, which means mechanically redirecting power to where it is needed. This incurs losses and is frequently limited by what the differential can physically accomplish. Beyond that, conventional cars have no choice but to utilize the hydraulic brakes, which wastes energy, wears out brake discs and, worst of all, slows the car down more than necessary. Mr. Hrvatinić explains, "Electric motors do not have this problem. Four electric motors means we can adjust the torque to each wheel a hundred times each second and have the motor respond almost as fast as the commands are sent. In electric motors, the direction does not play a role either – and can give negative torque just as easily as positive. This is called regenerative braking and it produces not only heat but also electrical energy, reclaiming part of the power used to accelerate the car. Add to this the 95% efficiency of an electric motor compared to 35% for a gasoline engine and you get a powerful, efficient, flexible, and easily controllable drivetrain."

Torque Vectoring

The next step was to design a control algorithm that would allow Rimac Automobili to make full use of this drivetrain's great power and flexibility. "Once again starting with a blank sheet of paper, we drew up a physical model of the car and started analyzing its passive behavior and defining the targets we wanted to accomplish with its active control systems. To build a model we could use for testing, we used all the data we were able to measure, from basic physical dimensions to suspension geometry and tire characteristics. This data was then transferred into a software suite for automotive physical simulation where we were able to validate the model data and start developing our dynamic vehicle control algorithm, Rimac All Wheel Torque Vectoring (R-AWTV)," states Mr. Tomislav Šimunić, head of the vehicle dynamics team.

Vehicle Behavior Under Electronic Control

"R-AWTV combines longitudinal and lateral control into one cohesive whole. It monitors and adjusts the forces on each wheel to produce a driving experience suited to each individual driver and situation. This is done solely by controlling the torque delivered to/from each of the motors in a way that it results in an improvement over the passive dynamics of

Concept_One is powered purely electrically and is the world's first all-electric hypercar, not only by performance but by the technology in it.

Mate Rimac, inventor of the Concept_One, beside the opened vehicle rear showing the power distribution unit.

>>

the car. We use high-precision physical sensors such as accelerometers, gyroscopes and wheel speed and steering wheel angle sensors that are fed into estimation algorithms to monitor the vehicle behavior. This way, we get a clear picture of the vehicle's physical state. The amount of grip, or total available force, on each wheel is estimated and either used as a limit to ensure maximum traction or deliberately exceeded to let the car enter a controlled lateral slide."

Driver-Controlled Lateral Dynamics

Rimac Automobili wanted a system that keeps the average driver safe and stable at high speeds and sharp turns, but also one that an expert driver would not feel was overbearing. This meant that the system had to be more configurable and offer more than the standard on-off switch most production cars provide. Concept_One's aluminum center console is designed to let drivers easily switch between operating modes, i.e., stable or dynamic driving behavior, by simply turning a knob. The innovative HMI solution enables the driver to fine-tune the torque distribution so that the car can operate as a pure front-wheel drive or rear-wheel drive, or any setting in between.

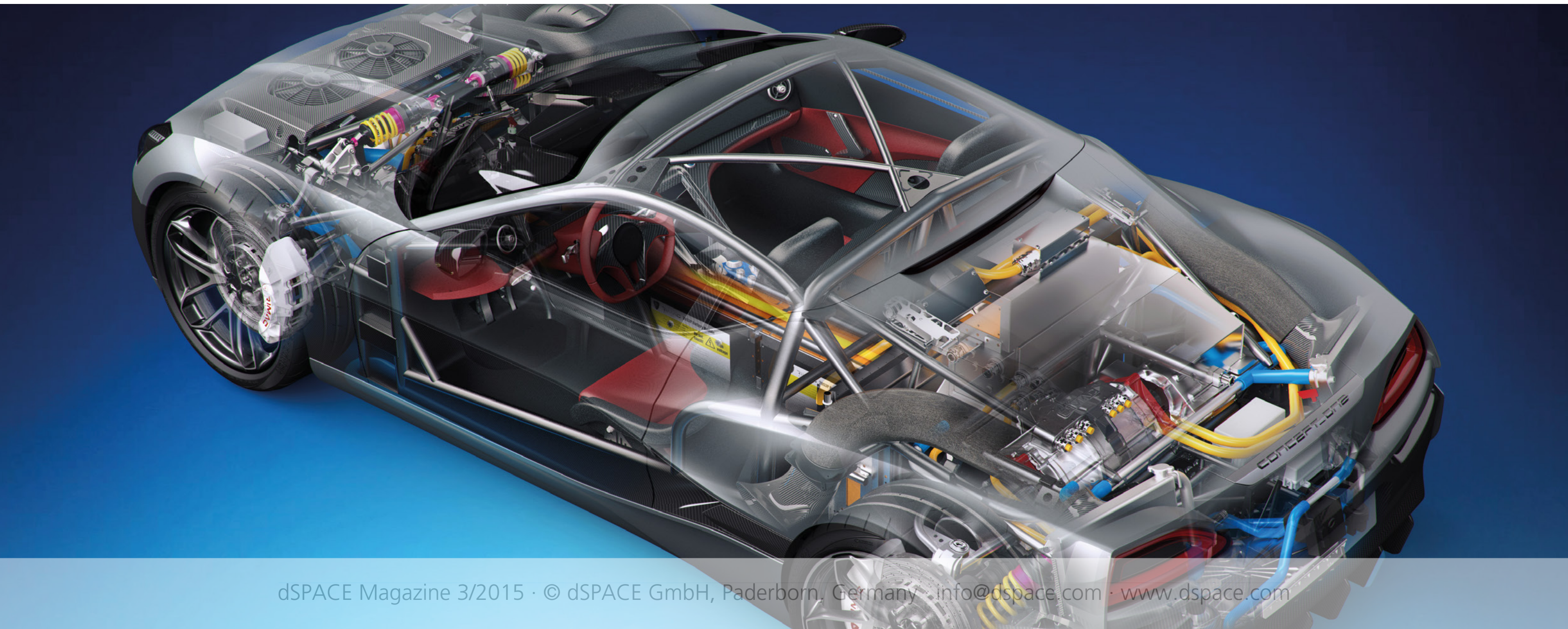
Role of the MicroAutoBox

"Of course, in order to make use of the drivetrain's great power and flexibility, clever control solutions had to be implemented on a platform reliable and fast enough to process them. This is why we chose the MicroAutoBox as the prototyping system to develop the central controller for the Concept_One. Its task is to coordinate the distributed network of secondary control units, provide safety-critical features such as over-temperature protection and device error detection, handle driver inputs and, of course, implement the Rimac All Wheel Torque Vectoring system," Mr. Hrvatinić says. The Concept_One makes full use of the MicroAutoBox's four CAN bus channels for communicating with the Rimac Active Battery Management System, the power distribution unit and charger, the four inverters and various chassis control units. The dSPACE RTI CAN Blockset is especially helpful here, making it easy to track and manage the nearly 200 CAN messages sent and received by various devices on the bus. For this, the standard serial communication channels and most of the analog and digital inputs are used. "Converting the control algorithm from a Simulink model compatible with our physical simulation software into a program that can be run on the MicroAutoBox is simple and straightforward, allowing control engineers to do their job without worrying about the underlying C code," concludes Mr. Šimunić.

Precise Signal Analysis with ControlDesk

Mr. Hrvatinić remarks, "dSPACE ControlDesk has proven invaluable

Packed with the latest technology: To connect the various systems of Concept_One, Rimac Automobili uses a central control unit that was developed using a dSPACE MicroAutoBox.



> "Working with the MicroAutoBox is simple and straightforward, allowing control engineers to focus on controller development without worrying about the underlying code."
>
> *Tomislav Šimunić, head of the vehicle dynamics team, Rimac Automobili*

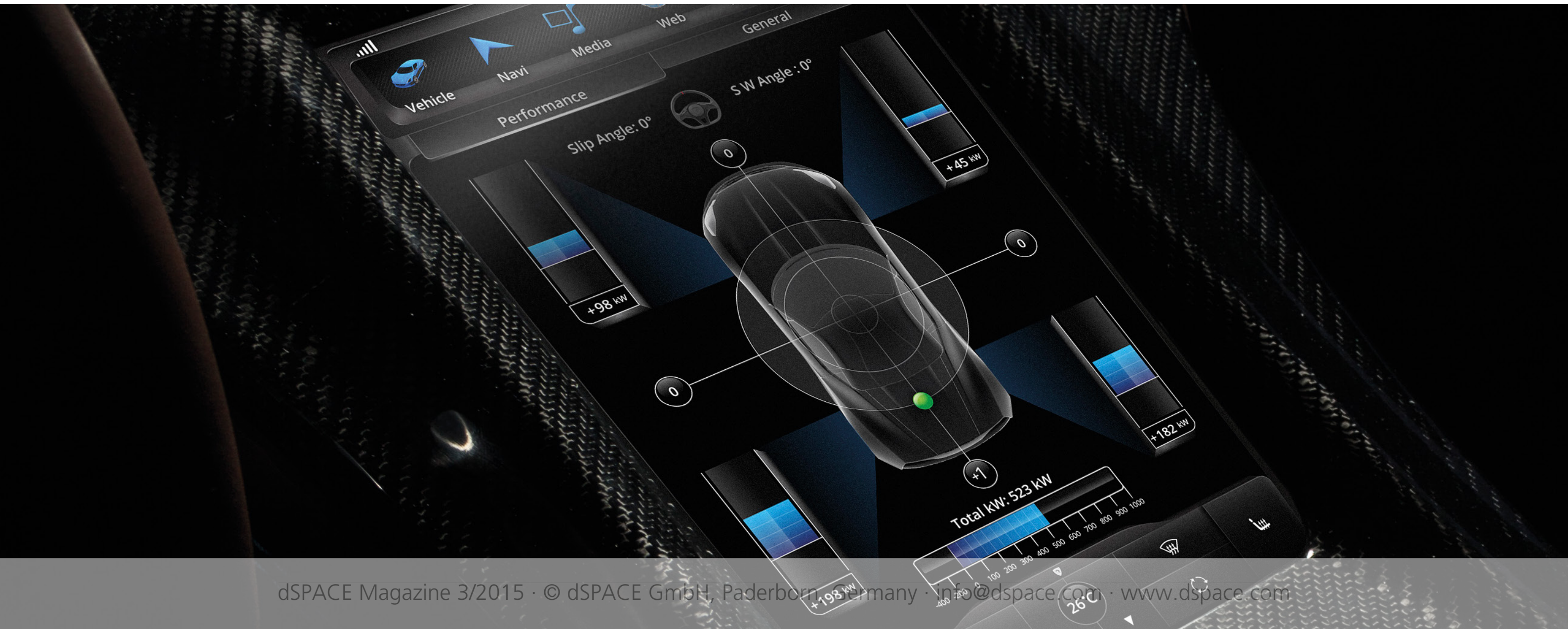
in real-world testing conditions. The ability to check the value of each signal in real time and record it greatly simplifies debugging, and is also very useful when evaluating the performance of dynamic control systems. We can access and review test data immediately after a test run is completed, which allows us to make the best possible use of a full day on the track." ControlDesk is also very useful for manually fine-tuning algorithm parameters on the fly. Since the values of parameters can also be changed easily, the time between test runs with different controller setups is reduced. Different experimental subsystems can be turned on and off without the need for any structural changes to the controller model.

Outlook

Only eight supercars in the first series Concept_One World Edition are scheduled to be produced, but the design and control algorithms are constantly being improved. The expertise and components developed and produced in-house for the Concept_One, such as the infotainment system, the powertrain and the battery system, to name just a few, are also finding a variety of applications in different B2B projects. Going forward, Rimac Automobili will continue to design and build the most powerful and sophisticated electric vehicles in the world, and find new ways to implement its cutting-edge technology in different fields and industries. ■

Courtesy of Rimac Automobili, Croatia

A touch screen shows the power values during engine operation and provides accurate settings.



The Joined Wing

AFRL, Boeing and NASA: success in wind tunnel testing of Joined-Wing SensorCraft vehicle

Photo credits: NASA

A goal of the Air Force Research Laboratory's SensorCraft project is to develop technologies for future high-altitude, long-endurance unmanned surveillance platforms. A component of this research project intended to develop technologies relevant to these large, flexible vehicles was the Aerodynamic Efficiency Improvement (AEI) program. The goals of the AEI program included the demonstration of flutter suppression, gust load alleviation (GLA), and reduced static margin, which are potentially enabling technologies for a SensorCraft vehicle that will allow it to have reduced structural weight, thereby increasing endurance, range, and payload capacity. The AEI program included a series of wind tunnel tests of two SensorCraft designs in the NASA Transonic Dynamics Tunnel. The configurations tested were a flying wing and a joined wing (shown in figure 1) described in references 1 and 2, respectively. Each configuration required a model support that provided rigid body degrees of freedom. The need to include rigid body degrees of freedom required the models to be 'flown' in the wind tunnel which added a great deal of complexity and risk to the project. The wind tunnel tests of these models were successfully and safely completed using two digital controller systems, one for the flight control laws and the other for the servo control loops, WatchDog system, and emergency control law. This article describes the controls' architecture and implementation using dSPACE systems, focusing on the joined-wing test. The test team included the Air Force Research Lab (AFRL), The Boeing Company, and the NASA Langley Research Center.

>>

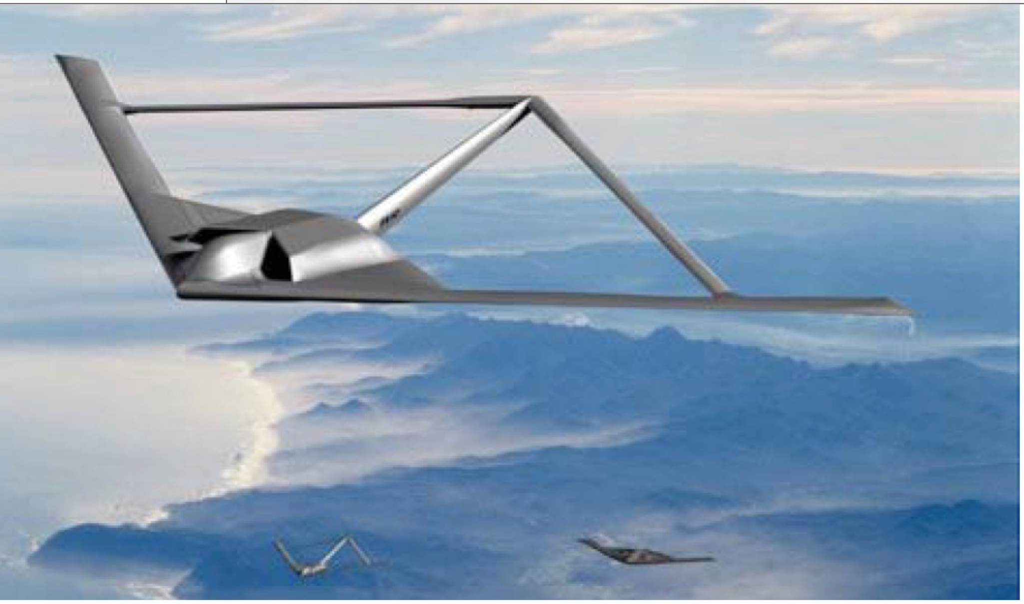


Figure 1: Joined-Wing SensorCraft configuration.

Experimental Setup

The Joined-Wing SensorCraft (JWS) tests were conducted in the NASA Langley Transonic Dynamics Tunnel (TDT). The TDT is a unique national facility dedicated to identifying, understanding, and solving relevant aeroelastic and aeroservoelastic problems. The TDT is a closed-circuit, continuous-flow, variable-pressure wind tunnel with a 16-foot square test section with cropped corners. The tunnel uses either air or a heavy gas as the test medium and can operate over a Mach number range from near zero to 1.2. The TDT is specially configured for flutter testing, with excellent model visibility from the control room and a rapid tunnel shutdown capability for model safety. The TDT has an Airstream Oscillation System to generate simulated gusts that was used extensively for this series of wind tunnel tests. Figure 2 shows an illustration of the JWS and the support system. The pitch and plunge degrees of freedom are indicated by the two arrows.

Figure 2: Illustration of the AEI Joined-Wing SensorCraft model mated to a two-DOF support system installed in the NASA Langley TDT.

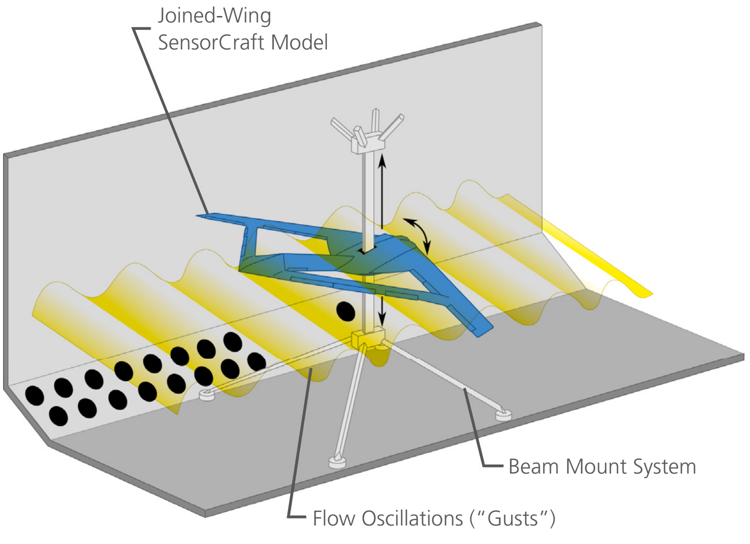


Figure 3 shows a photo of the JWS model installed in the TDT. The support system is comprised of a beam and carriage, and for safety, the carriage was equipped with a plunge brake and pitch displacement limiter to remotely lock out rigid body motion. The model was also equipped with a large instrumentation suite consisting of accelerometers, strain gages, rate gyros, potentiometers, and a total of 13 hydraulically actuated, high-bandwidth control surfaces, each with a Rotary Variable Differential Transformer (RVDT) position sensor. The control surfaces were located on the trailing edges of the wings with six each on the forward and aft wings and a rudder. External to the JWS wind-tunnel model, a variety of components had to be integrated to support the test. These included two dSPACE digital control systems, a commercial signal conditioning system, custom signal conditioners for the RVDTs and the Moog servo valves, various power supplies, and a custom snubber control system. A schematic showing how the wind-tunnel model and the various systems were connected is shown in figure 4. The signals external to dSPACE units are all analog signals. The anti-aliasing filters were set to 400 Hz for the RVDTs, as they were routed only to a digital control system running at a 1,000 Hz frame rate (dSPACE 1). All other signals were filtered at 100 Hz to be compatible with the other digital control system running at a 200 Hz frame rate (dSPACE 2). Data were acquired using the TDT Data Acquisition System (DAS) sampling at 500 Hz.

Control Systems

The control tasks for the JWS wind-tunnel test were divided between two dSPACE systems. The servo control loops for positioning the control surfaces using the RVDTs, servo valves and the WatchDog system

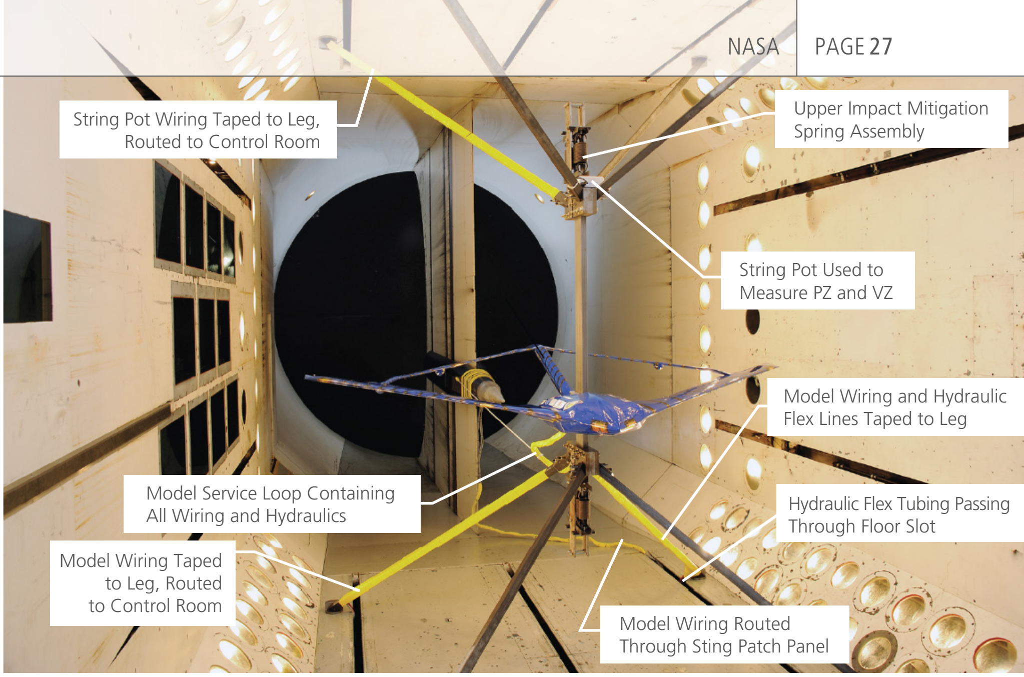
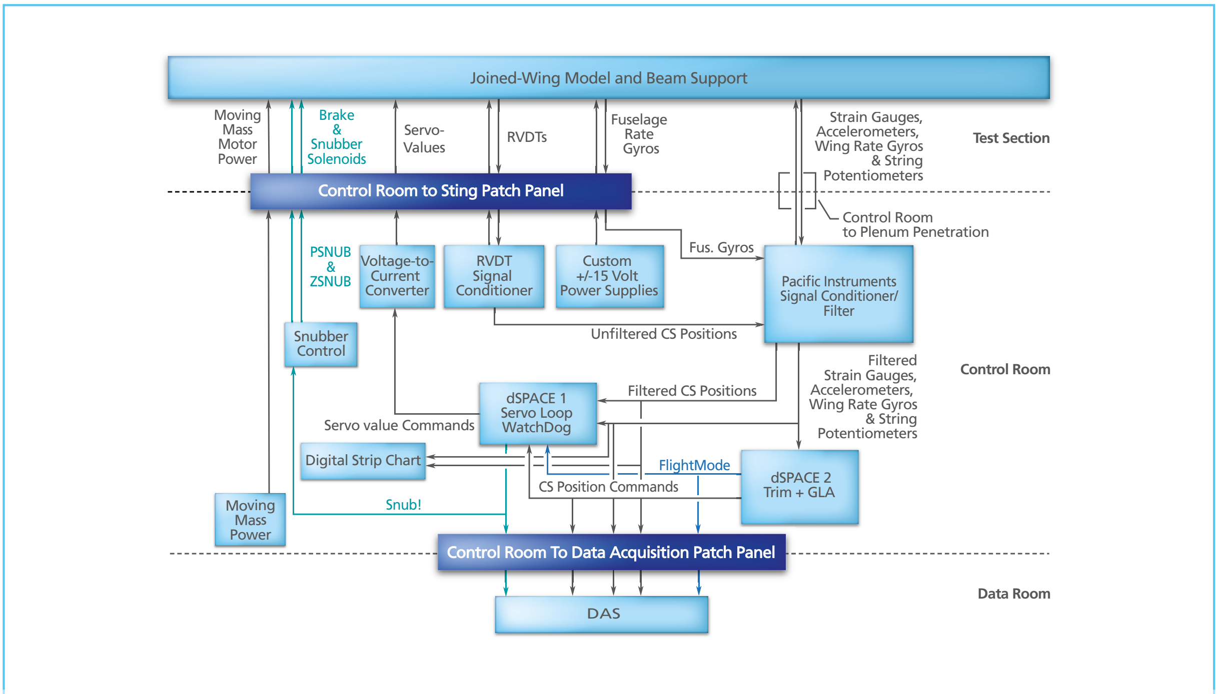


Figure 3: Joined-Wing SensorCraft in the NASA Langley Transonic Dynamics Tunnel.

were both implemented on dSPACE 1, while dSPACE 2 was used for flight control (trim and GLA). The snubber control system was custom built to support this wind tunnel test, and it consists of a latching circuit, several switches, and a power supply. The snubber control system combined with solenoid valves and hydraulic actuators onboard the model was used to lock out the rigid body motion. This system could be tripped manually, but this feature was never used. Instead, the automated WatchDog system running on dSPACE 1 reliably issued the "Snub!" command to keep the model safe. Key features of the two dSPACE systems and the snubber control system are shown in figure 5. User input to these systems is shown in dark gray. The gust load alleviation and trim control laws were implemented within the flight control block on dSPACE 2. This flight control block included a GUI interface and programming logic for controlling or initiating certain events like resetting the system or initiating a takeoff sequence. An excitation signal could be added to various combinations of the control surface commands when conducting parameter identification testing. These commands were output as analog signals to dSPACE 1. The servo-control loops and the WatchDog system were implemented on dSPACE 1. The servo-loops were independent PID control loops equipped with output saturation blocks to prevent overdriving the actuators. The WatchDog system monitored the model signals, and when a fault was detected, it would issue a "Snub!" command and transfer control to the emergency control law via the switch shown in figure 5. For the joined-wing tests, the emergency controller consisted simply of 0° control surface commands, but a closed-loop controller was used in the test described in reference 1. As shown in figure 5, the various systems communicated with each other via the status

>>

Figure 4: Signal routing used in the joined-wing tests.



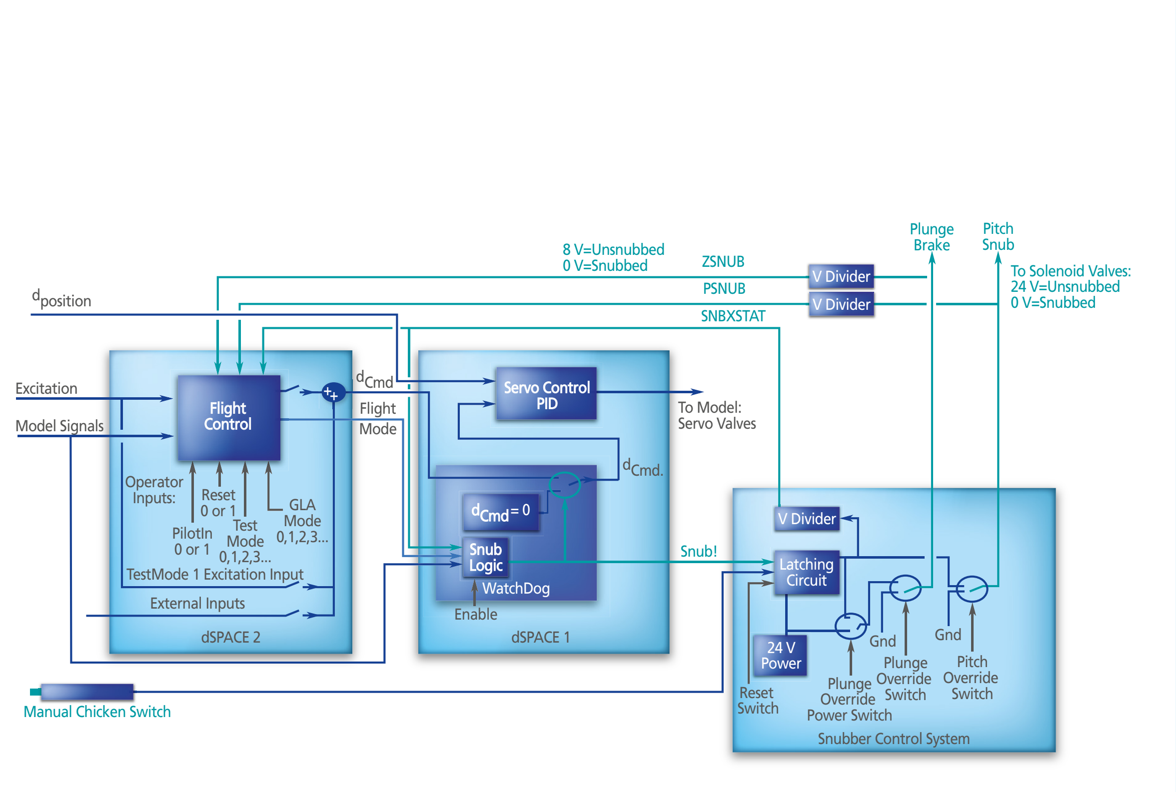
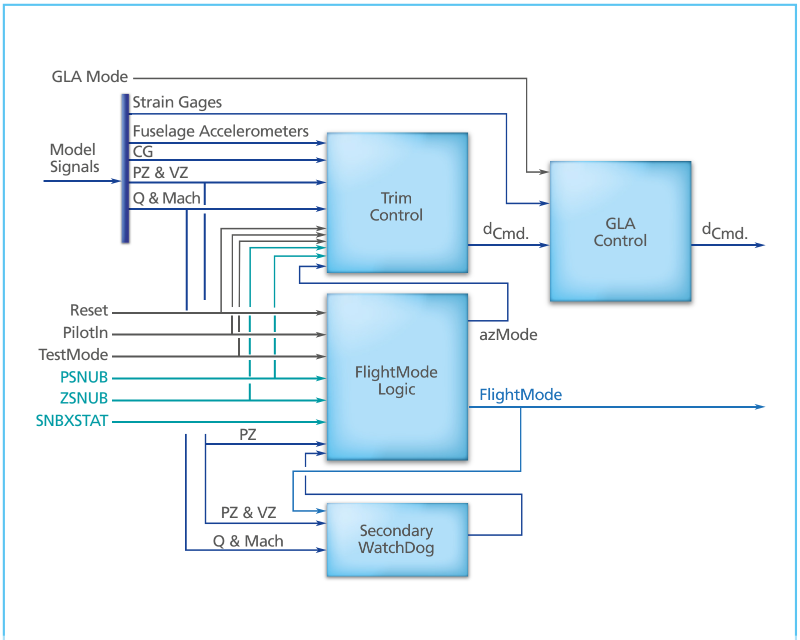


Figure 5: Block diagram of dSPACE systems and snubber control.

signals coming from the snubber control system to dSPACE 1, the FlightMode signal from dSPACE 2 to dSPACE 1, and the Snub! signal from dSPACE 1 to the snubber control system. These means of communication along with proper user inputs solved one of the key challenges of the joined-wing wind-tunnel test, transitioning the model from being at the bottom of its range of vertical travel with plunge brake and pitch limiter engaged to flying freely at the center of the vertical travel range. The steps involved are detailed in reference 2 and beyond the scope of this article. Key features of the dSPACE 2 flight control block are shown in figure 6. The primary components of this block are the trim and gust load alleviation (GLA) blocks, the FlightMode logic block, and the fault detection block. Model signals, snubber-related signals, operator inputs, and the relevant model signals are shown. The trim controller has two modes of operation: Theta hold and altitude (Z) hold. The exact mode of operation is determined by the user inputs and the model vertical position. Logic for ramping the vertical position set point from the lower stop to tunnel centerline is also contained in the trim control block. The operation of the GLA control block is controlled by the user input GLAMode. When GLAMode is set to 0, the GLA control block simply passes through the control surface commands. When this parameter is greater than 0, strain gauge feedback was used to generate GLA control surface commands that are added to the trim controller outputs. The trim controller was designed to launch, fly and land the model in the wind tunnel and to serve as the reference for GLA reduction. The trim control-

Figure 6: Block diagram of the flight control system.



ler consisted of two main elements, a vertical (Z) loop and a pitch (θ) loop. The vertical loop consisted of a simple PID controller that generated an acceleration command. The pitch loop consisted of a PID plus acceleration feed-forward controller to generate an elevator surface command from the acceleration command. These two loops used gains that were scheduled based on the model's center of gravity. The single control surface command was passed through a third-order low pass filter to attenuate the response at higher frequencies.

dSPACE Systems

Each dSPACE Digital Control System (DCS) consists of a rack containing a host computer, a target system, a keyboard, a monitor, BNC patch panels for I/O, and an uninterruptible power source. The heart of the DCS is the target system that includes a dSPACE DS1006 control processor board utilizing a 2.6 GHz AMD Opteron™ processor connected to three dSPACE DS2002 Multi-Channel A/D Boards and one dSPACE DS2103 Multi-Channel D/A Board. The A/D boards each have 32 channels utilizing 16-bit quantization with an input range of ±10 volts. The D/A board contains 32 channels of 14 quantization bits designed for ±10 volts and a settling time of 10 µsec. The controller software is developed within the MATLAB®/Simulink® environment, then compiled and downloaded to the target processor via the dSPACE and MATLAB Real-Time Interface. An integral component of the dSPACE tools is the ControlDesk® application. ControlDesk provides the user interface to the target processor for the development and implementation of the GUI. The host computer runs the GUI and controls all communications between the processors. ■

David A. Coulson and Robert C. Scott, NASA

Results

Closed-loop wind tunnel tests were conducted over a period of approximately six weeks. Throughout testing, both the trim controller and the GLA control law were continually refined. The general process was to design, implement, test, and evaluate the trim and GLA controllers. This process was repeated multiple times during the testing period. As the testing progressed, improved testing procedures were developed that allowed better parameter identification data sets to be acquired. These data sets were used to further refine the analytical models, helping to improve the trim and GLA controller designs. Ultimately, trimmed flight at -10 percent static margin and a reduction of structural response of at least 50 percent were successfully demonstrated. The ability to customize and reconfigure the dSPACE systems, and their compatibility with the industry-standard MATLAB computing environment were key success factors in this wind tunnel test. This is exemplified by the fact that the control law development work was done by a team member located on the West Coast, while the TDT is located on the East Coast. Updated control systems could easily be delivered to the TDT as Simulink models, dropped into the existing framework, compiled, and be ready to run in a matter of minutes. Numerous control design iterations could thus be attempted, leading to a successful outcome.

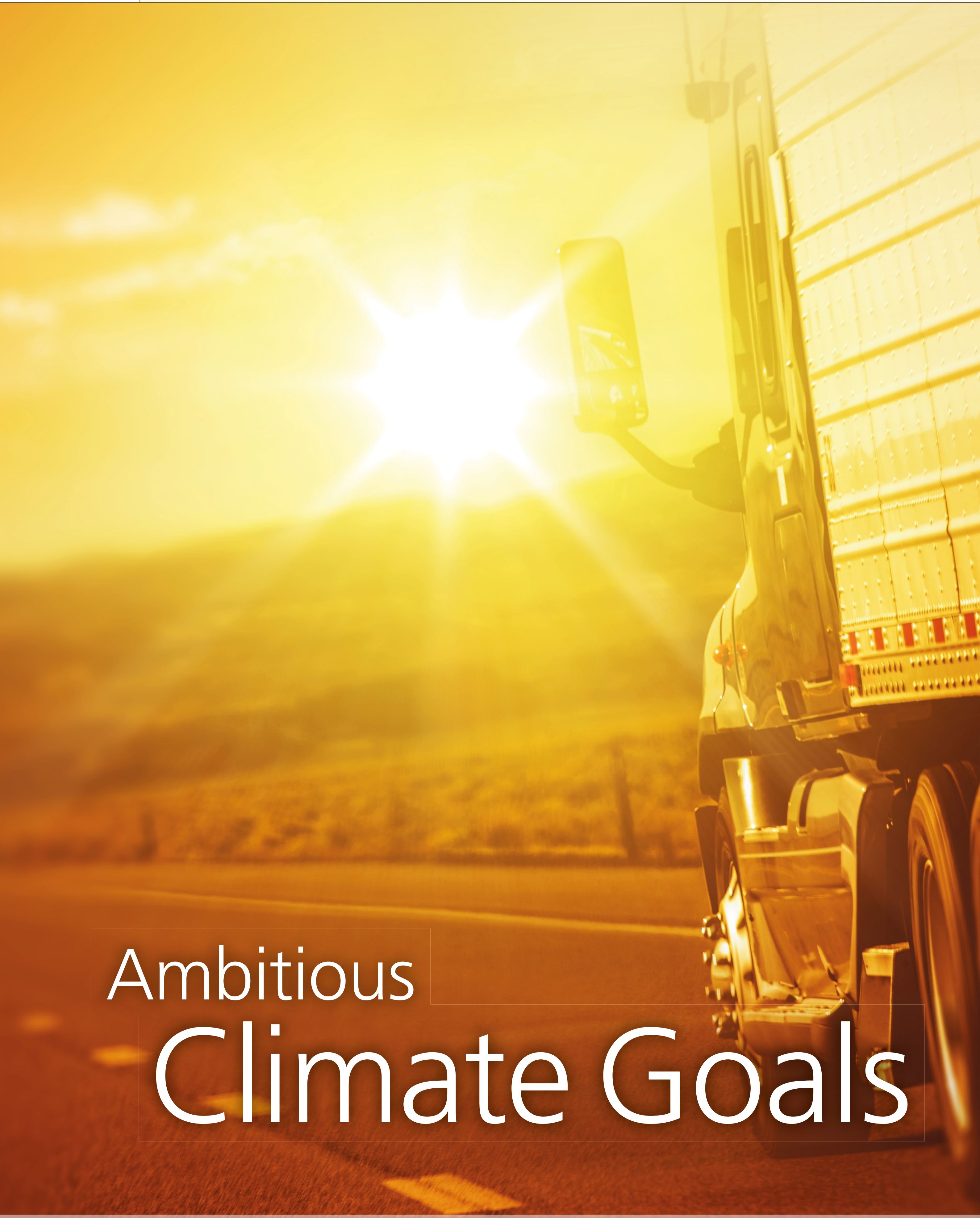
References

[1] Scott, R., Vetter, T., Penning, K., Coulson, D., and Heeg, J., "Aeroservoelastic Testing of Free Flying Wind-Tunnel Models Part 1: A Sidewall Supported Semispan Model Tested for Gust Load Alleviation and Flutter Suppression," NASA/TP-2013-218051, Oct. 2013.

[2] Scott, R., Castelluccio, M., Coulson, D., and Heeg, J., "Aeroservoelastic Testing of Free Flying Wind-Tunnel Models Part 2: A Centerline Supported Fullspan Model Test for Gust Load Alleviation," NASA/TP-2014-218170, Feb. 2014.

David A. Coulson
Senior Research Engineer, Active Controls at the NASA TDT, Analytical Services & Materials, Inc., Hampton, VA, USA.

Robert C. Scott
Senior Aerospace Engineer, Aeroelasticity Branch, NASA Langley Research Center, Hampton, VA, USA.



Ambitious Climate Goals

Bergstrom aims at developing 85% of its control software virtually

Precise control of the cab temperature is the be-all and end-all of developing vehicle climate control systems. Bergstrom aims at reducing the time-consuming calibration of these systems by 80%. For this, the climate experts use a model-based approach with various dSPACE tools.

Commercial and specialty vehicles always have to perform their demanding tasks reliably, even in extreme climate conditions. In order for drivers to ‘keep cool’ and be able to focus on their job, their driver cabins have to provide an optimal working environment. Bergstrom Inc. significantly contributes to such reliable climate conditions.

Ambitious Goal

Bergstrom offers two commercial products in the cab climate systems branch: one classic engine-powered system and one battery-powered system (known as NITE, No-Idle Thermal Environment) for air conditioning in a parked vehicle. A core component of all Bergstrom systems is the automated temperature electronic control unit (ECU) whose tests

>>

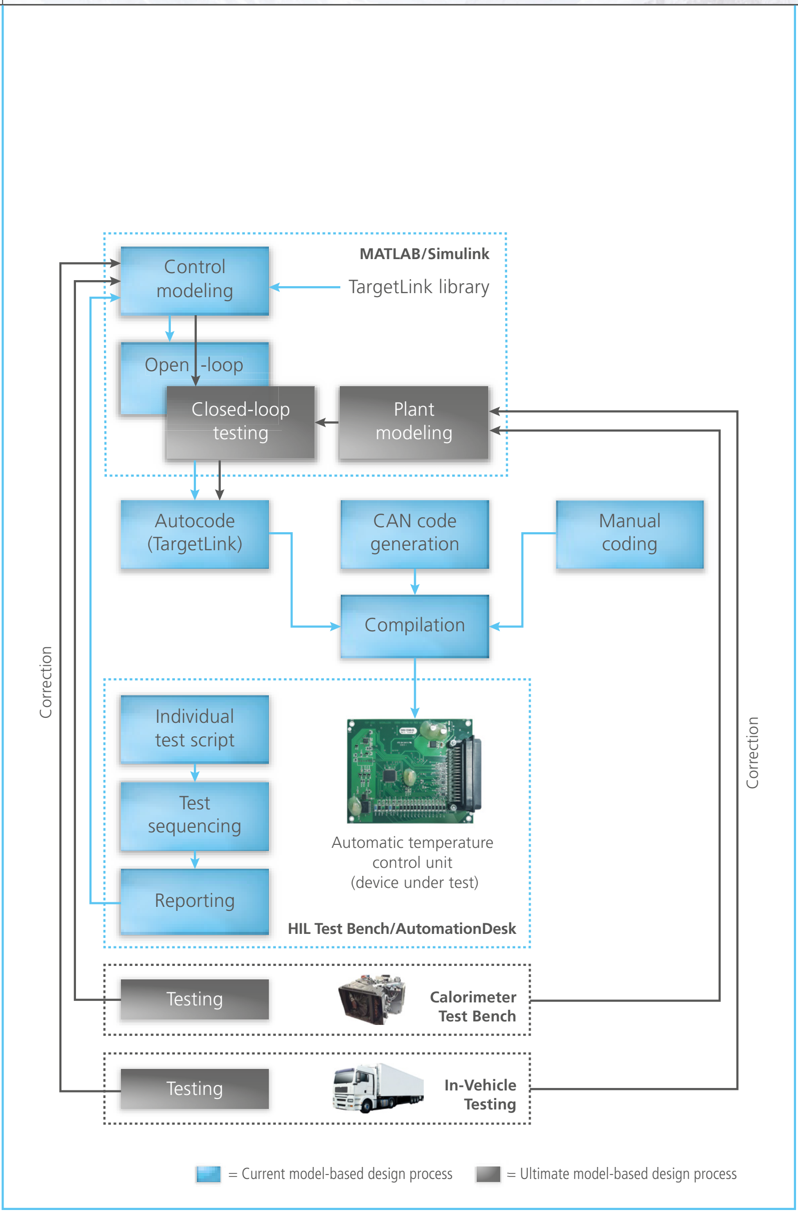


Figure 1: Current (blue) and ultimate (gray) model-based development process at Bergstrom.

and calibration have always required long iteration times. This is why Bergstrom aims at accelerating the ECU's development in the long term and performing more calibration steps offline in a new, model-based development process. The company is very ambitious right from the start: In the near future, Bergstrom wants to develop 85% of its control software virtually and reduce the calibration time by 80%.

Powerful Tool Chain Required

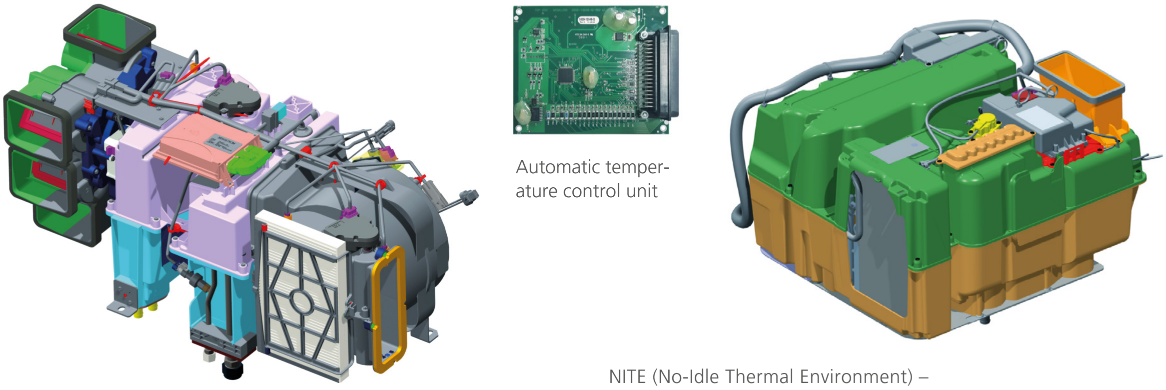
After setting the course for the model-based design journey, Bergstrom had to choose the right instruments for it. These instruments had to provide efficient processes and a well-proven, powerful tool chain. The company opted for state-of-the-art development tools, including:

- A data management system to keep all data (models, documents, specifications, software, figures, tests, etc.) organized, up-to-date, and in one central location accessible to all development teams worldwide
- A software version control system for traceability across the product's life cycle
- A requirements management system for documenting, analyzing, tracing, and prioritizing the requirements
- A modeling environment (MATLAB®/Simulink®) for model-based software development
- The production code generator dSPACE TargetLink® for automatically generating efficient, production-ready code from the models
- The test automation software dSPACE AutomationDesk in combination with a dSPACE hardware-in-the-loop (HIL) simulation system for automating software tests.

New Structure for the Development Process

When these tools were first included in a basic model-based development process (figure 1, blue elements), one of the first tasks was to capture and document the requirements of existing products via reverse engineering. The obtained design information allowed Bergstrom to start modeling the control algorithms in MATLAB/Simulink. The same platform was used for immediate open-

> "We embarked on our journey to model-based development with tools from dSPACE, because as the market leader, the company offers state-of-the-art products."
>
> *Bjorn Hansson, Bergstrom Inc.*



Automatic temperature control unit

Classic HVAC system

NITE (No-Idle Thermal Environment) –
battery-powered HVAC system

Figure 2: The automatic temperature control unit (center) is one of the key components in both Bergstrom's classic heating ventilation and air conditioning (HVAC) system (left) and the NITE (No-Idle Thermal Environment, right) battery-powered HVAC system.

loop function tests of the modeled control unit, giving the developers a first glimpse at the actual behavior of the ECU's algorithms in the design phase. If the tests were successful, the control models were translated into production code with TargetLink. This production code was then compiled and implemented on the ECU together with additional communication interfaces (CAN code), a calibration protocol (XCP), and additional, hand-written code. The next step was to thoroughly test the ECU on a HIL simulator. To make these tests easier, Bergstrom created a series of automated test cases with AutomationDesk. The developers were able to use the test reports to correct and adjust the control model.

First In-Vehicle Tests

After a prototype was tested on the HIL simulator, it was integrated into the driver cabin of a real truck. For this, TargetLink converted the controller and interface models into production code, which was then implemented on the ECU. The developers used a calibration system to fine-tune the ECU in the vehicle via the XCP protocol. The entire prototype functioned correctly in the vehicle, so the integration tests were successful.

Promising Interim Result

The successful in-vehicle tests showed that the software controller already fulfills all the requirements that were formulated in advance. But a functioning model is only the first step. Additional development effort is required to calibrate it and make sure it functions correctly. Currently, the Bergstrom engineers are tasked to transfer real-world insights to the models.

Path to the Ultimate Development Process

In the future, Bergstrom will fine-tune established model-based workflows and add further test procedures (figure 1, gray elements). For example, the correction loops that further improve the models will also consider test results from the calorimeter test bench and in-vehicle tests. Therefore, the actual calorimeters and vehicles will also be modeled in MATLAB/Simulink to perform closed-loop tests on controllers in the future. The model's functions can then be validated much earlier and more extensively. In the long term, these methods aim at achieving Bergstrom's goal of 85% virtual development and validation of control software before in-vehicle testing, thus reducing calibration time at the mechanical level by 80%. From this, Bergstrom expects distinct time and cost savings. Hence, the drivers of commercial vehicles will not be the only ones to keep cool. At the end of the journey to model-based development, Bergstrom's financial controlling department can do so as well.

Bjorn Hansson,
Bergstrom Inc.

Bjorn Hansson
Bjorn Hansson is Chief Mechatronics Engineer at Bergstrom Inc. in Rockford, Illinois, USA.

OSU at EcoCar 2 –
hybrid power takes first place

Model-Based Winning

The Ohio State University drove home first-place honors in the EcoCAR 2 advanced vehicle technology competition finale for delivering a re-engineered plug-in hybrid that impressed judges across the board. During the three-year project, the students used state-of-the-art industry tools to redesign a 2013 Chevrolet Malibu, implementing vehicle energy storage, electric drive and ethanol-(E85)-fueled engine technology.

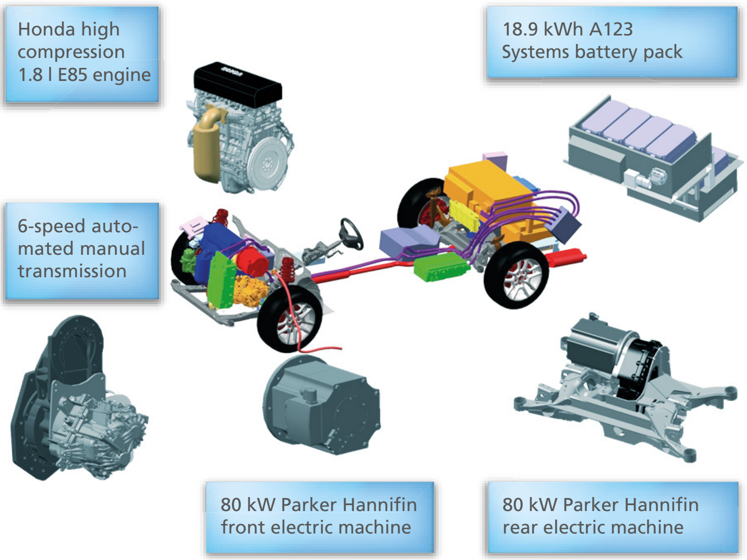
Photo by Myles Regan/CC BY-ND 2.0
https://www.flickr.com/photos/doeavtc/14338888602/in/album-72157644984645925

"Using one's own concept to give a production vehicle such a reduction in fuel consumption and emissions – while retaining full performance – is an incredible result. And on top of that, it was students who accomplished this as part of their studies," said Santhosh Jogi, Director of Technology at dSPACE Inc. "Earning first place in the overall standings of the EcoCAR 2 competition was more than deserved. And we are, of course, proud that tools from dSPACE have played such a decisive role in the development process." "In addition," Jogi continued, "we also awarded the team with the 1st place dSPACE Embedded Success Award for fully embracing the concepts behind product development, the model-based development process and tool use, and combining them effectively."

Competition for the Makers of Tomorrow

"Placing first overall was a big deal," said M.J. Yatsko, EcoCAR 3 Co-Team Leader and EcoCAR 2 HIL Development Leader. "The whole three-year EcoCAR 2 competition with 15 North American university teams, sponsored by the U.S. Department of Energy (DOE), General Motors (GM) and several other institutions and companies, gave us students an up-close feel of the current and future challenges in the automotive industry." One of the main underlying goals of the EcoCAR 2 competition was for teams to come up with creative ways to further optimize the energy efficiency and environmental compatibility of the 2013 Chevrolet Malibu – an already established production vehicle. Each team was given three years to envision, develop, and implement their vehicle designs, without compromising performance, safety, and consumer acceptability features. Throughout the competition, each team was required to put their vehicles through the same level of industry testing used by GM for their production vehicles. With their plug-in hybrid vehicle design, the OSU team achieved an impressive 50 miles per gallon (4.7 l/100 km) gas equivalent, while using 315 watt-hours per mile (196 watt-hours per kilometer) of electricity and were able to significantly reduce their vehicle's emission levels.

Figure 1: The vehicle architecture – flexible power on both axles.



In the Heart of EcoCAR 2

"For the vehicle architecture we chose a plug-in hybrid concept," explained Jason Ward, Project Manager of the OSU team. "There are many sources for power: the front axle is powered by a Honda 1.8-liter ethanol combustion engine with a 6-speed automated manual transmission. Additional torque is provided by an 80 kW electric motor which is coupled with the transmission via a belt. The rear axle is driven by an additional 80 kW electric drive." Andrew Huster, Electrical Team Leader, presented the key advantages: "The various drive components can be combined flexibly to set up drive modes as combustion engines, hybrid drives, and purely electric drives. According to the particular drive mode, the battery pack can be recharged, maintained by energy recovery or discharged during operation." The OSU team performed extensive tests to ensure smooth transitions from one mode to another.

The plug-in hybrid controller has a hierarchic architecture. A dSPACE MicroAutoBox® II acts as the central supervisory control system. Below the supervisory level, connected via CAN bus interfaces, are the low-level controllers for the engine, battery, brakes, transmission, electric motors, etc. The architecture allows easy expandability, is fault tolerant, and makes it easier to try out various controller variants.

HIL Tests with dSPACE Simulator

In the first year of the competition, the OSU students focused on the vehicle architecture and vehicle subsystems. In addition to performing SIL tests (primarily with their own

> "dSPACE tools were a significant contribution towards making our team reach the milestones and specification targets for EcoCAR 2. With the dSPACE tools, we were able to test the control code easily, while designing and creating the mechanical/electrical subsystems at the same time."
>
> *Matthew Yard, former OSU EcoCAR 2 Team Leader*

developed SIL simulator), they also carried out intensive HIL tests with hardware and software from dSPACE. The second year involved the actual completion of a prototype vehicle and the integration of components. Finally, in the third year of the competition, accompanied by numerous road tests, the car was further optimized by using the constructed tool chain to achieve the desired fuel consumption, emissions, performance, and driveability goals defined in the team's Vehicle Technical Specification (VTS).

During the HIL testing phase, the OSU team worked with four different HIL configurations, utilizing a dSPACE Simulator Mid-Size to develop and validate subsystem-to-system functions. This included thoroughly checking each of the controllers that they implemented themselves. The team utilized these HIL configurations with closed-loop dynamic plant models to test functional behavior, failure detection and mitigation, communication between controllers, and many other component and vehicle-level features. The four HIL configurations were:

- *Case 1:* Validating the main controller, implemented on a dSPACE MicroAutoBox II. Simulation models from various manufacturers were used.

>>

Figure 2: ControlDesk® Next Generation was used for HIL tests with a dSPACE Simulator and for controller applications on the dSPACE MicroAutoBox II.

Excerpt from original photo by Myles Regan/CC BY-ND 2.0
https://www.flickr.com/photos/doeavtc/14221296830/in/album-72157644984645925/

Figure 3: Ohio State's Katherine Bovee shows U.S. DOE's Michael Knotek their vehicle.

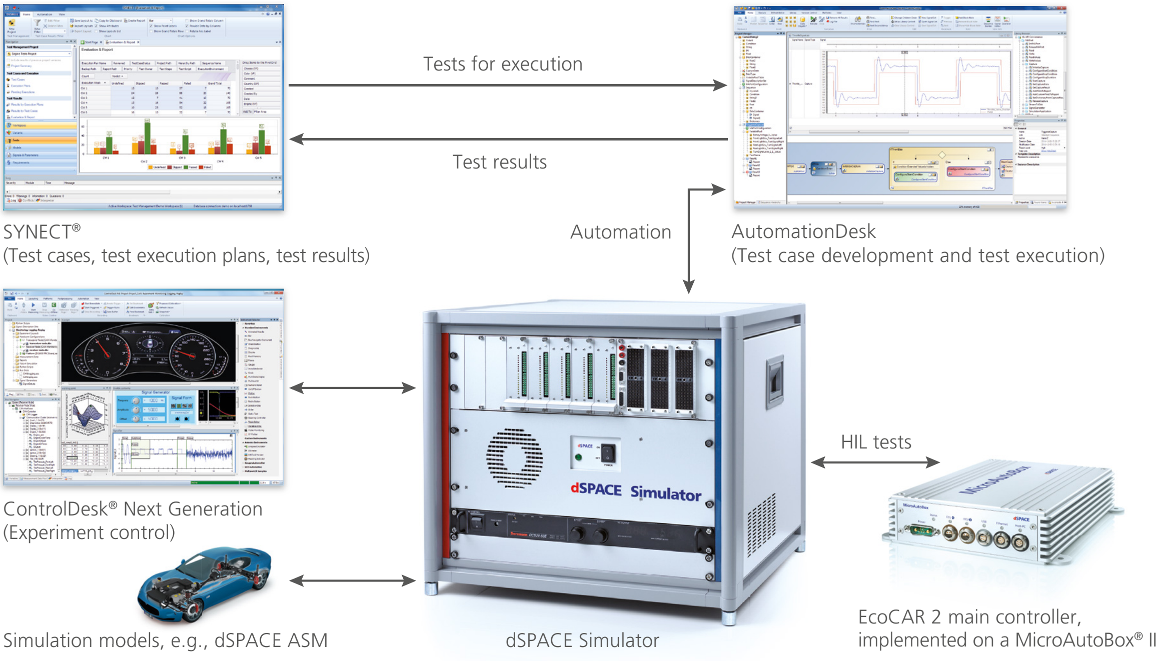


Figure 4: A comprehensive dSPACE tool chain with SYNECT as the central data management software.

- *Case 2:* Validating the combustion engine control, with dSPACE Automotive Simulation Models (ASM) as the simulation model and with their own development ECU.
- *Case 3:* Validating the transmission control, with dSPACE ASMs as the simulation model and a 128-pin Woodward MotoTron as the controller.
- *Case 4:* Validating the CAN communication of the entire controller network, developed by the OSU team.

"Initially, our team performed the tests manually, but it quickly became clear that 'more' was needed to reach the safety of the control code," said Amanda Hyde, former OSU EcoCAR 2 Fault Diagnosis Team Leader. "Extensive, automated regression testing was necessary for each new version of the code, considering the full controller functionality. The solution was to build a powerful tool chain with dSPACE SYNECT®, dSPACE AutomationDesk and the dSPACE Simulator. This automation gave our team a crucial time advantage for in-vehicle tests. Altogether, a total of 74% of our HIL tests were automated."

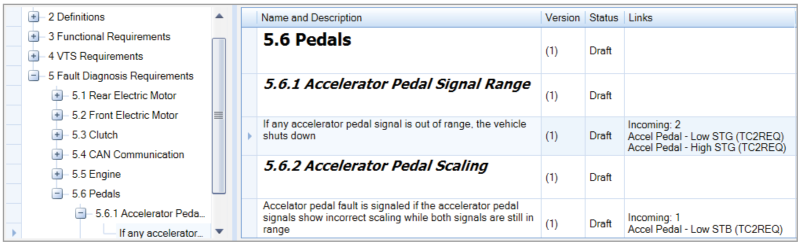
AutomationDesk – Test Authoring and Automation

The test cases and scripts were implemented in AutomationDesk through its graphical programming environment. Utilizing AutomationDesk's integrated debugger and the ability to insert breakpoints and investigate tests step-by-step ensured fast error-finding and resulted in reliable test sequences. Overall, the team used 76 automated tests that were covered by only 16 test scripts, due to skillful test grouping and parameterization.

> "dSPACE SYNECT was a great help for our regression tests in the third year of the competition. It let us focus more on the in-vehicle tests and on the overall optimization of the vehicle, while providing a central tool for managing our development data and test runs."
>
> *Amanda Hyde, former OSU EcoCAR 2 Fault Diagnosis Team Leader*

SYNECT – Superior Data Management for Automated Tests

SYNECT, the data management software from dSPACE, played a key role in the area of automated testing. First, the OSU team loaded the list of requirements from the "Control and Validation Requirements Document" into SYNECT. To define the tests, the students imported parameterized test sequences from AutomationDesk. The defined test cases, linked to the requirements for optimal traceability, were then processed comfortably step-by-step in SYNECT via the test execution plans. With the specially-configured test reports, it was easy to trace the success of the tests during the development period. Changes in the requirements, the associated test cases, and AutomationDesk automation scripts could be updated at any time with a few clicks in SYNECT. ■



Name and Description	Version	Status	Links
5.6 Pedals	(1)	Draft	
5.6.1 Accelerator Pedal Signal Range	(1)	Draft	
If any accelerator pedal signal is out of range, the vehicle shuts down	(1)	Draft	Incoming: 2 Accel Pedal - Low STG (TC2REQ) Accel Pedal - High STG (TC2REQ)
5.6.2 Accelerator Pedal Scaling	(1)	Draft	
Accelator pedal fault is signaled if the accelerator pedal signals show incorrect scaling while both signals are still in range	(1)	Draft	Incoming: 1 Accel Pedal - Low STB (TC2REQ)

Figure 5: List of requirements in SYNECT, with linked test cases.

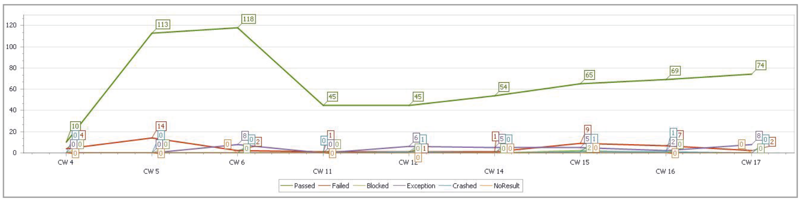


Figure 6: Clear trace of the test progress in SYNECT.

Published with the kind permission of the Ohio State University EcoCAR 2 team.

Conclusion and Outlook

"With the high achievement goals, right from the beginning, and the significant complexity of the plug-in hybrid, our team could not have achieved success without a highly efficient development process and professional tools," concluded Matthew Yard, former OSU EcoCAR 2 Team Leader. The OSU team succeeded in optimally managing time and resources to achieve its goals. In a short amount of time, the students became accustomed to the development process and tools and gained confident control of the dSPACE tool chain, which consisted of a dSPACE Simulator, MicroAutoBox II, SYNECT, AutomationDesk, and ControlDesk Next Generation. While the EcoCAR 2 competition is over, the OSU team is now in full swing with the next advanced vehicle technology competition – EcoCAR 3. In this competition, students have four years to optimize a 2016 Chevrolet Camaro and the bar for requirements has been set even higher with added criteria for costs and the degree of innovation. The OSU team, with its new team members, continues to impress and already won the first year of this new competition. dSPACE congratulates the Ohio State University team for their outstanding performance and wishes them future success!

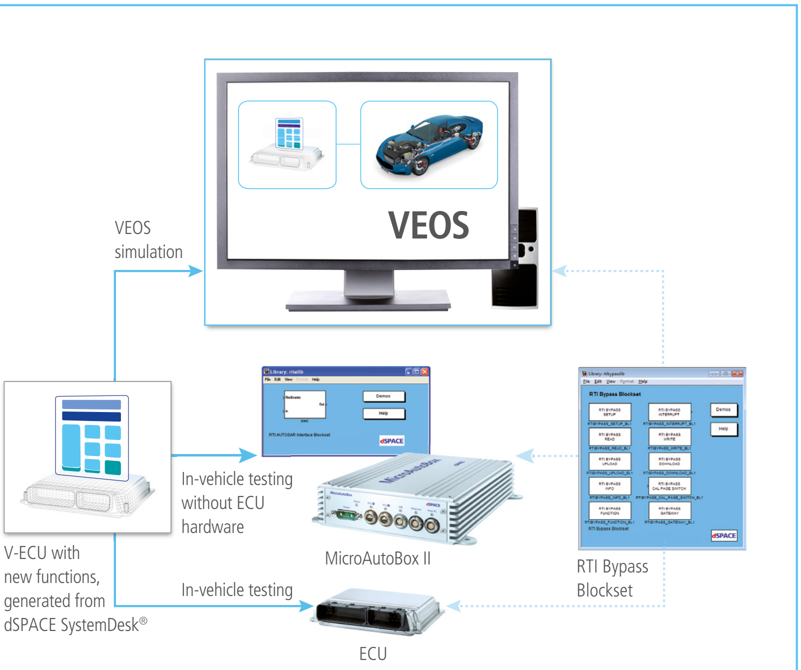
Figure 7: The smile of champions – The Ohio State University EcoCAR 2 team received first-place honors for their plug-in hybrid concept.

The new control function has been completed in Simulink – the next step is to test it (more) easily and fast(er) in combination with the existing ECU software. dSPACE's tool chain for virtual bypassing offers a highly efficient solution.

The MATLAB®/Simulink®-based approach for developing control algorithms further is the prevailing method worldwide. Once the first control designs are done, their interaction with other software components or even complete electronic control unit (ECU) software has to be tested realistically. Until now, function developers had to wait for suitable prototypes of the production ECU to become available. But the required number of prototypes is not available until late in the development process.

However, the later testing begins, the less time developers have for integration, error search, corrections, and optimizations. This increases the time pressure, because the deadlines for new production ECUs are usually very tight and because corporate management and customers both have high expectations.

Figure 1: With the RTI Bypass Blockset, new ECU functions can be used on different platforms, such as VEOS, MicroAutoBox II, or the ECU prototype.



Frontloading Tests with Virtual Bypassing

Here is the approach: When new functions are integrated into existing ECU software or a virtual ECU (V-ECU) via virtual bypassing and are simulated in this environment on the developer PC, function tests can be performed much earlier. This means that the developers can test much earlier whether their changes have the desired effect – without ECU hardware or access to the physical controlled system. Virtual bypassing makes integrating new functions into existing ECU software quick and easy, because the new function is used simply by selecting it; the ECU source code does not have to be modified. Therefore, ECU software does not have to be recompiled either, which saves developers long build times and allows for considerably more development iterations.

Virtual Bypassing Tool Chain

Virtual bypassing is made possible by the dSPACE RTI Bypass Blockset, the same blockset used for external bypassing and on-target prototyping, and by VEOS®, dSPACE's PC-based simulation platform. With VEOS, developers can simulate entire virtual ECUs on the PC together with complex plant models such as the dSPACE Automotive Simulation Models (ASM).

>>

Earlier test results with virtual bypassing

Taking Function Development to the Next Level

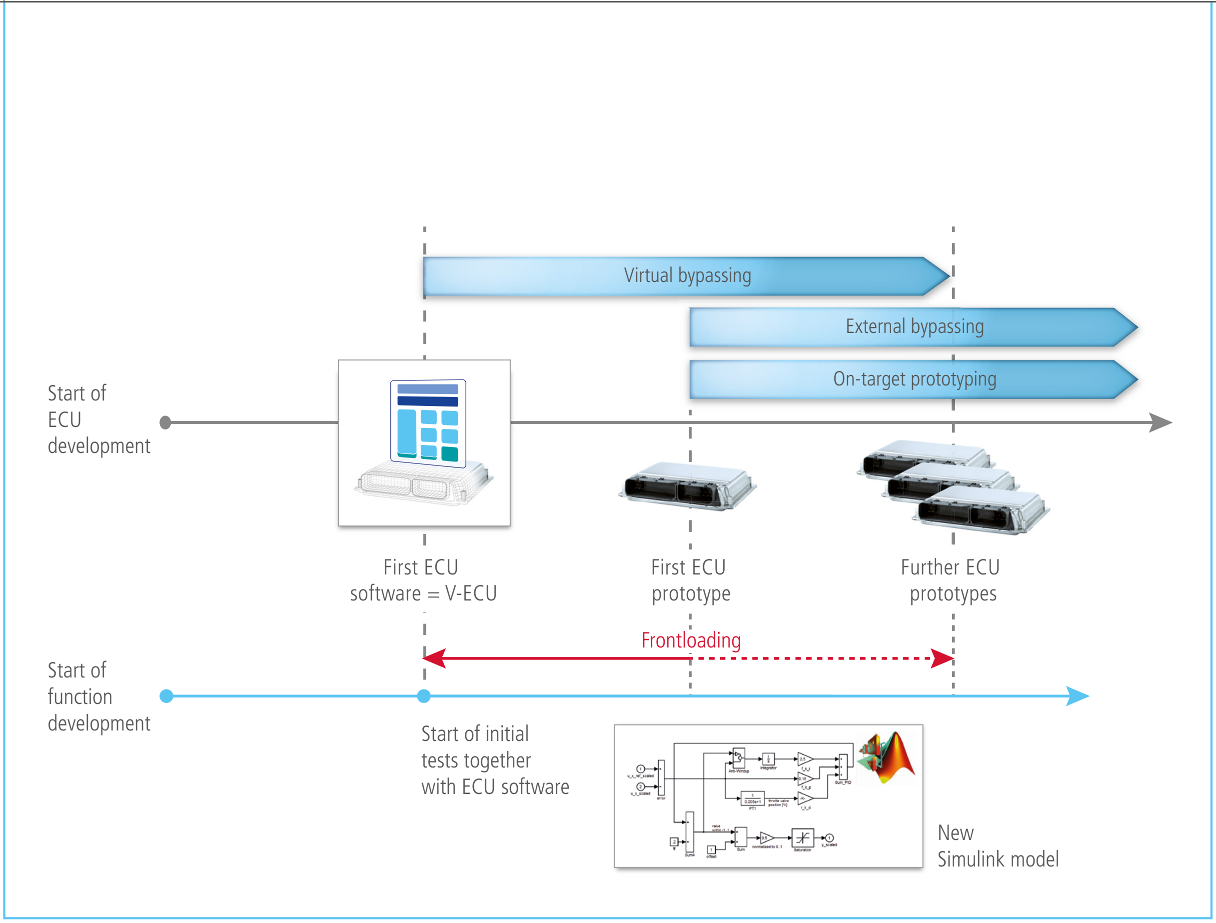


Figure 2: Using virtual ECUs makes it possible to start function tests earlier and achieves a better software quality faster.

The V-ECU can be provided by a software integration expert. The function developers just have to use the RTI Bypass Blockset to connect their Simulink functions with the functions of the ECU software – without any special software skills or integration know-how. This lets them focus completely on implementing and optimizing their functions and testing them in combination with other software components. It is also possible for several developers to use the same V-ECU for very different controller components, without having to generate it again (figure 2). Virtual bypassing therefore eliminates redundant work.

Validating More Iterations Faster

The RTI Bypass Blockset not only allows for integrating new functions into the V-ECU without a new software build. It also lets developers replace controller models during a running simulation. Different variants of a controller can therefore be tested and compared without a simulation restart, i.e., without causing a delay. This approach is even more efficient because VEOS can simulate faster than in real time. Another benefit of virtual bypassing is the possibility to run tests before a hardware prototype is available. Tests can therefore be performed earlier. This frontloading means that developers have more time for developing and testing. It also mitigates project risks.

Offline and Online

As soon as the real ECU becomes available, developers can perform real-time tests on it together with the physical controlled system, in the lab or in the vehicle. The users switch from virtual bypassing to either external or internal bypassing. To do this, the new control function is integrated into the final ECU software on the real ECU. The transition is seamless, and is also performed using the dSPACE RTI Bypass Blockset so that users do not have to become familiar with new software. In the blockset itself, users simply select a different execution platform, such as an ECU instead of the V-ECU (figure 1). With ControlDesk® Next Generation, developers can then use measurement and calibration data and experiment layouts for all platforms.

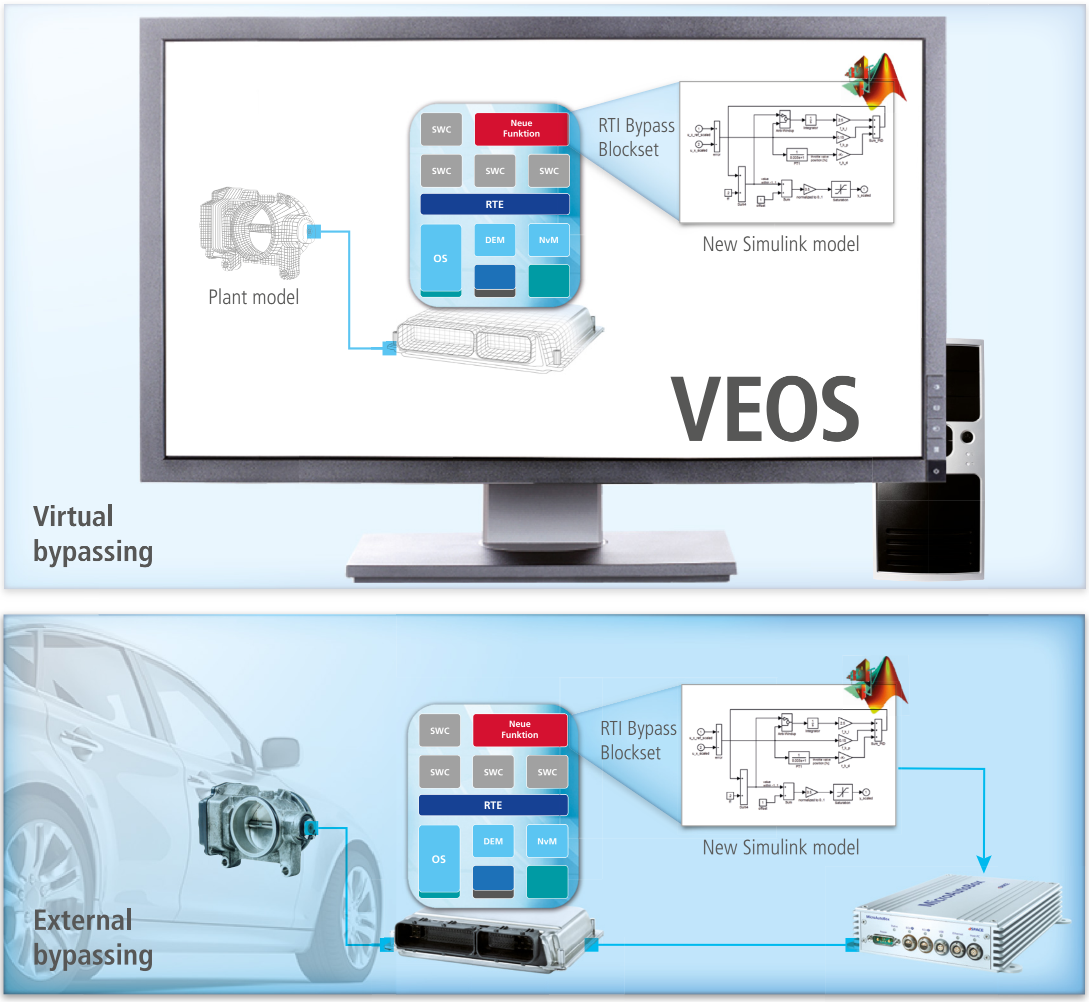
Real-Time Testing Without the ECU

If real-time testing is required when the ECU prototype is not available yet, dSPACE's MicroAutoBox II prototyping system replaces the ECU. With the RTI AUTOSAR Blockset, the V-ECU is transferred to the MicroAutoBox and used in the vehicle. Virtual bypassing can also be used here to extend the functionality of the V-ECU. The Simulink model of a new function, including the bypass blocks, can remain completely unchanged, which allows for a seamless transition from VEOS.

Conclusion

Virtual bypassing makes it possible to frontload function tests to the PC-based simulation with dSPACE VEOS and start them earlier. This approach enables more and faster development iterations, without a real ECU or physical controlled systems. The RTI Bypass Blockset allows for a seamless transition between the different dSPACE development platforms, providing a continuous, highly efficient development process with a very short training period.

Figure 3: The dSPACE RTI Bypass Blockset allows for a seamless transition from virtual bypassing with VEOS to external bypassing with MicroAutoBox II and the production ECU.



AutomationDesk is a powerful test authoring and automation tool for testing electronic control units (ECUs) and supports the graphical definition of test sequences. Now, AutomationDesk becomes even more powerful. Its new, unique signal-based test description makes it possible to create and execute tests quickly and clearly in real time.

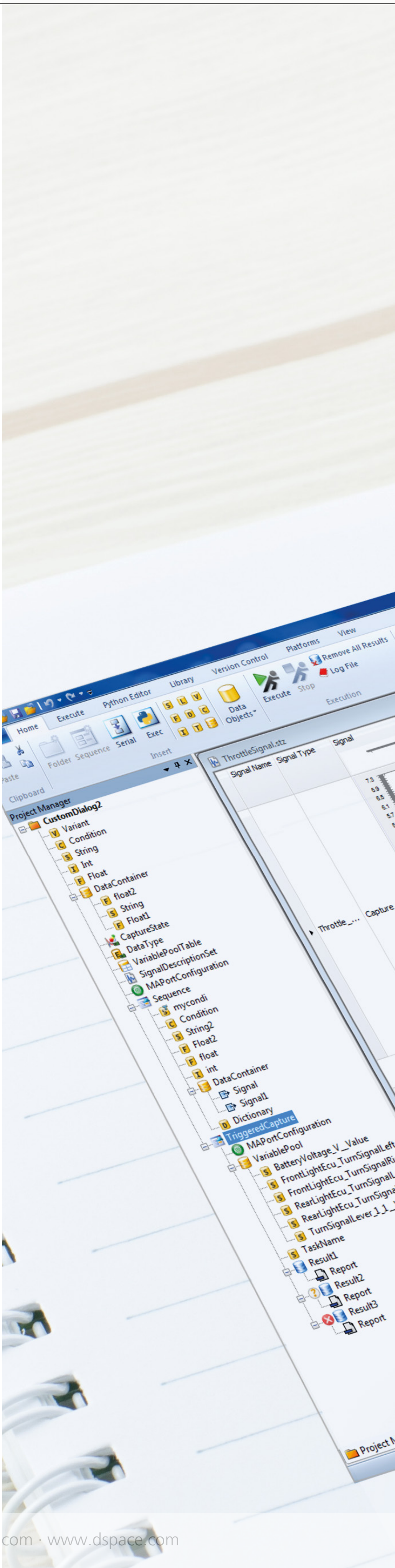
AutomationDesk has long established block-based testing, i.e., testing a combination of graphical function blocks, which has been used successfully in thousands of projects. But some test scenarios are better described by signal behaviors. These scenarios include:

- Test descriptions in which signal behaviors serve as a reference for evaluating measurement variables
- Tests in which stimuli have to be added in real time or in which requirements have to be evaluated in real time

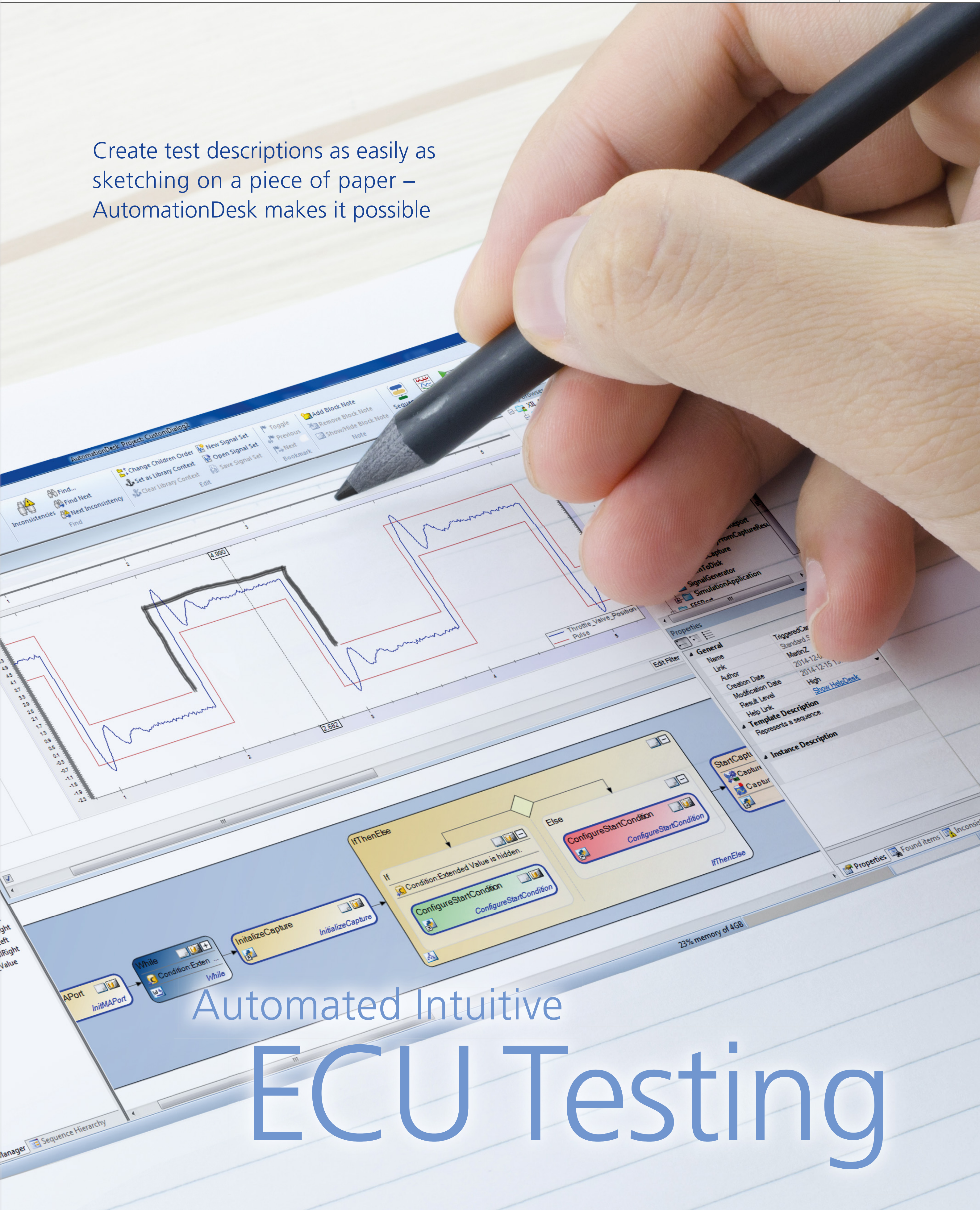
This is where signal-based testing comes into play. Signal-based tests offer a new kind of test description that is as easy and intuitive to create as if it were on a sheet of paper. With signal-based testing, stimulus and reference signals for simulation variables can be described intuitively, in a plotter-like editor. The documentation of the performed tests provides a report with informative plots and parameter information.

The main advantage of this new method is increased transparency. Users can create the test specifications in an editor, and the reports have a similar layout as the test specifications, because reference and signal behavior are displayed more precisely. They can see the test criteria and results at first glance. This is what makes signal-based testing so intuitive.

>>



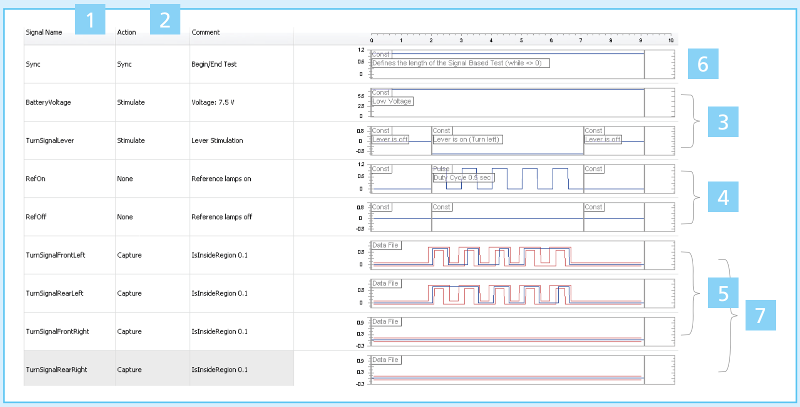
Create test descriptions as easily as sketching on a piece of paper – AutomationDesk makes it possible



Automated Intuitive
ECU Testing

Figure 1 (left): This is how signal-based testing works: In this example, the behavior of the turn signal is tested at a low onboard voltage (7.5 V) and after the turn-signal lever was activated.

Figure 2 (below): Stimulus and reference signals for simulation variables are described intuitively in a plotter-like editor. The simulation results are displayed directly in the graphical test description.

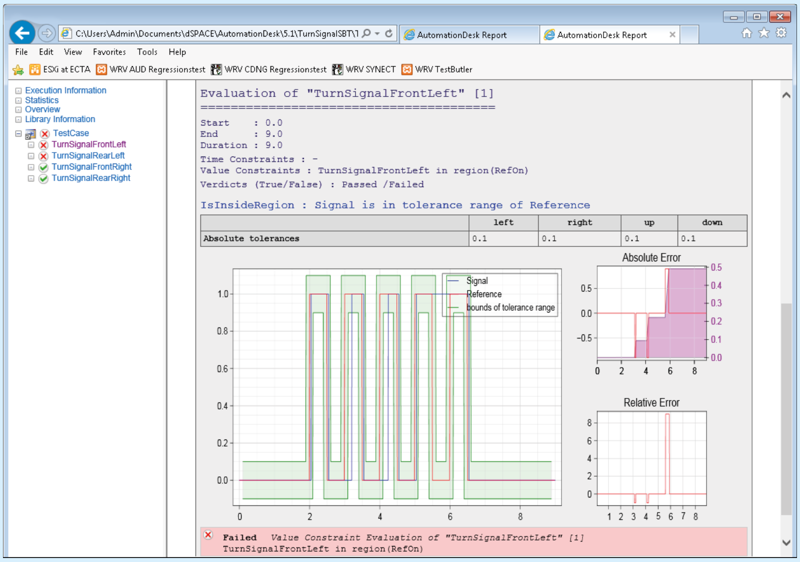


Signal Name	Action	Comment
Sync	Sync	Begin/End Test
BatteryVoltage	Stimulate	Voltage: 7.5 V
TurnSignalLever	Stimulate	Lever Stimulation
RefOn	None	Reference lamps on
RefOff	None	Reference lamps off
TurnSignalFrontLeft	Capture	IsInsideRegion 0.1
TurnSignalRearLeft	Capture	IsInsideRegion 0.1
TurnSignalFrontRight	Capture	IsInsideRegion 0.1
TurnSignalRearRight	Capture	IsInsideRegion 0.1

>> Creating Signal-Based Tests

Signal-based testing with AutomationDesk usually comprises seven steps:

1. Assigning variables:
   Which variables of the simulation model are assigned to which signal behavior in the test?
2. Specifying actions:
   Which signal serves as a stimulus, measurement or reference?
3. Defining stimulus signals:
   What segments should a stimulus signal consist of (e.g., step, ramp, sine)?
4. Defining reference signals:
   What segments should a reference signal consist of (e.g., step, ramp, sine)? The same description segments are used for stimulus and reference signals.
5. Defining evaluation methods for reference signals:
   Which range (tolerance) must the measured values be in so the test is passed?
6. Defining the test duration:
   What is the longest run time of the test?
7. Test execution and evaluation:
   Are the signal behaviors really inside the specified tolerance?



```
Evaluation of "TurnSignalFrontLeft" [1]
=========================================
Start    : 0.0
End      : 9.0
Duration : 9.0
Time Constraints : -
Value Constraints : TurnSignalFrontLeft in region(RefOn)
Verdicts (True/False) : Passed /Failed

IsInsideRegion : Signal is in tolerance range of Reference
```

	left	right	up	down
Absolute tolerances	0.1	0.1	0.1	0.1

```
Failed   Value Constraint Evaluation of "TurnSignalFrontLeft" [1]
         TurnSignalFrontLeft in region(RefOn)
```

Figure 3: As expected, for a low onboard voltage the signal of the turn signal does not meet the requirements and the test result is 'failed'. The combined display of the tolerance range and result behavior together with error curves ensures a high transparency.

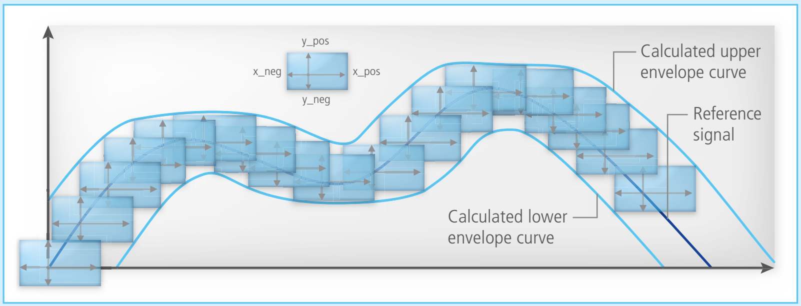
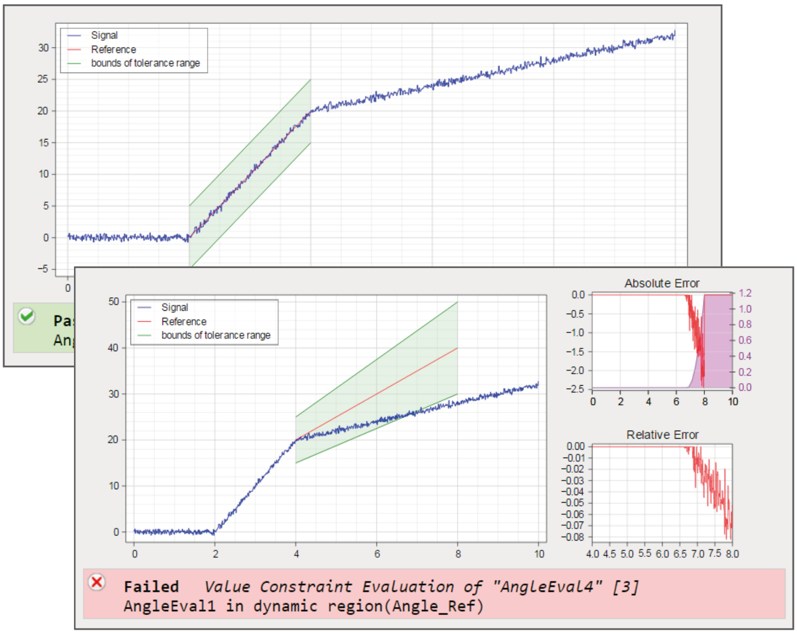


Figure 4: When you define tolerance values, a valid, rectangular, parameterizable area is defined around each point of the reference signal. The measurement signal has to be in this area. If you connect the corners of all possible rectangles, you get the upper and lower envelope curve of the valid signal behavior.

As if on Paper

In signal-based testing, all signal behaviors are created and edited graphically, in an intuitive editor. The test execution itself is performed in AutomationDesk as usual, with the tried and tested mechanisms that are also available for block-based testing.

The possibility to define evaluation boundaries for the signal amplitude and time via the reference signal and tolerance values is particularly useful. The result is an evaluation criterion in the form of an envelope curve. The measurement signal must be inside this curve. But tolerance values can also be specified relative to a variable value. The envelope curves can then expand or contract, depending on the signal behavior.



```
Failed   Value Constraint Evaluation of "AngleEval4" [3]
AngleEval1 in dynamic region(Angle_Ref)
```

Figure 5: Different evaluation rules can be applied to individual segments. This example shows an absolute tolerance value of 2-4 seconds and a relative, calculated tolerance value of 4-8 seconds.

XIL-API Compliance

In signal-based testing with AutomationDesk, test descriptions comply with the XIL API standard and therefore offer a standardized access to the simulation platform. This means that you can execute the tests on any XIL-API-compliant hardware. The test descriptions are thus platform-independent and can be used in other simulation environments as well.

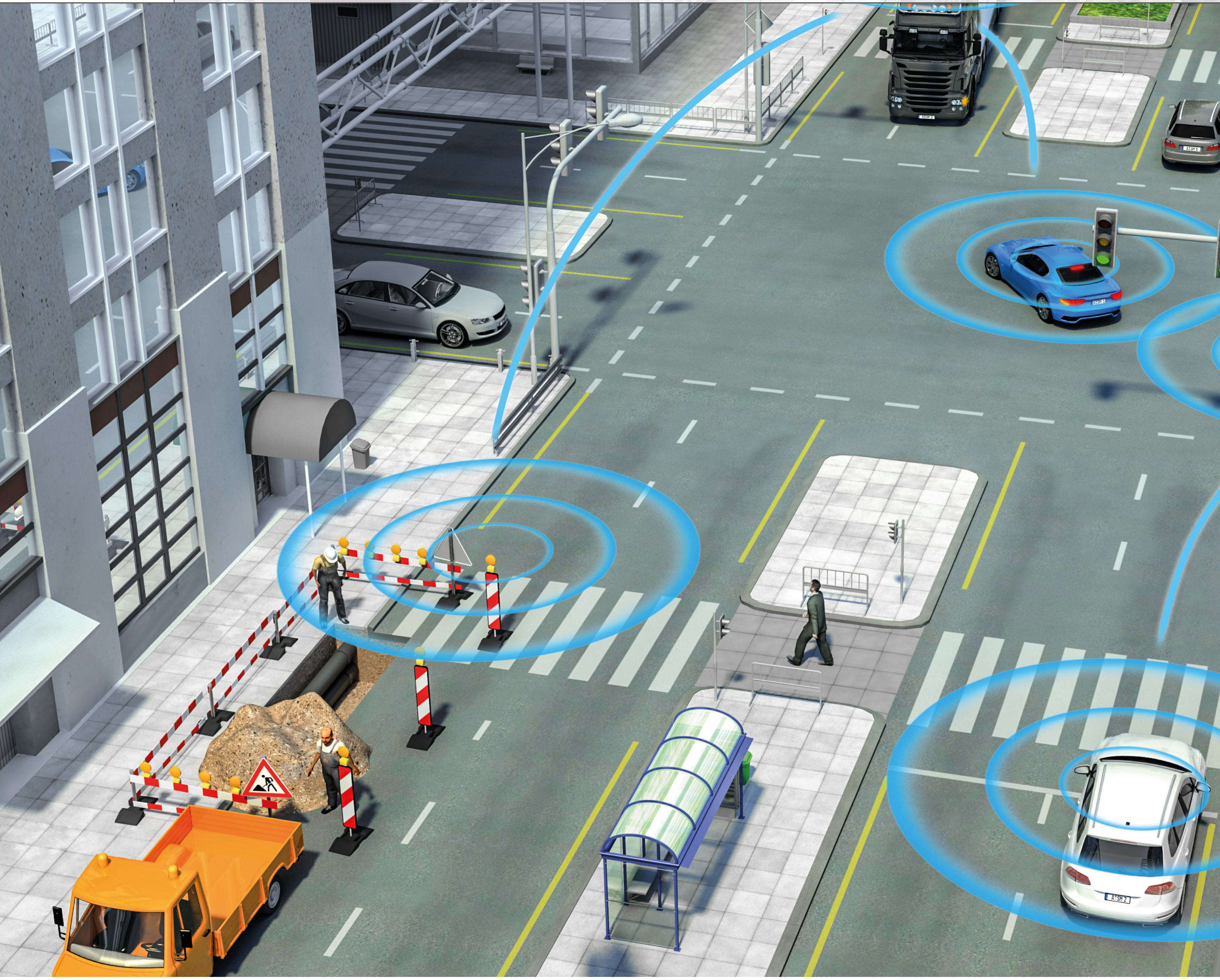
The describing elements (segments, signals, conditions, etc.) for signal-based testing are also based on ASAM XIL. This lets AutomationDesk customers use their valuable accumulated ASAM XIL know-how and workflows for stimulus definition when defining reference signals.

Segment-Based Testing

Signal-based tests can be divided into four different segments. Segments let users apply test criteria more precisely. In addition to a quick and easy evaluation description for an entire signal, developers can also describe individual segments of quality functions, e.g., to exclude the beginning (startup) and end (shutdown) of a signal from the evaluation. ■

> "AutomationDesk's Signal-Based Testing library makes defining our test cases both easy and precise, and gives us a meaningful test report. Signal-based testing ensures that measured signals follow a stimulus within 10 milliseconds. This method let us achieve a major goal: to check as many signals as possible at the same time."

Dr. Yoon Kwon Hwang,
Principal Research Engineer, Advanced Test & Development Team, Hyundai MOBIS, South Korea



Model-based development of V2X applications

Everything on Screen

The introduction of V2X technology opens up numerous possibilities for enhancing safety and comfort on the roads while decreasing the amount of time and fuel spent. dSPACE has a tailored solution for developing and testing the related applications efficiently.

Today's driver assistance systems use environment sensors such as radar and cameras to scan the vehicle environment. But if the view of these systems is blocked, e.g., by other vehicles or large buildings at an intersection without a clear view, some information about the environment is missing (figure 1). With the introduction of V2X technology, these restrictions might become a relic of the past. The 'X' stands for other objects in the vehicle environment, not only other vehicles but also parts of the infrastructure, such as traffic lights and road signs. V2X technology, often referred to as C2X or Car2X, enables the exchange of information between all of these objects via ITS-G5 (IEEE 802.11p), the WLAN-based ad-hoc network standard. The exchanged data packages contain information about the position, speed, and driving direction or sudden events such as traffic jams, construction sites, or slick roads. The goal of introducing V2X technology is to increase traffic safety and driving comfort and to optimize the flow of traffic. The technology is therefore a further step towards autonomous vehicles.

Crucial: Cross-Border Strategies for Market Introduction

Introducing V2X technology is a great challenge for automobile manufacturers, because in order to achieve the goals mentioned above, 10% of the vehicles on the market have to use V2X communication. This is why vehicle manufacturers, suppliers, and tool providers such as dSPACE work together in the CAR 2 CAR Communication Consortium (C2C-CC) to formulate a joint strategy for introducing V2X and to define a European standard. This is done in close cooperation with the standardization groups ETSI and CEN, and harmonization groups from the EU, the US, and Japan. The main focus is not just on wireless communication, but also on aspects such as defining supported applications; standardized criteria for detecting traffic jams, fog, or slick roads; the definition of required data protocols; and a comprehensive data security concept. Companies in the US also put great effort into introducing V2X. The US standards resemble the European solution in many areas. In contrast to Europe, however, the US is discussing legally binding regulations. V2X will likely be introduced on the European and US markets in this decade.

Developing V2X Applications

The functions of V2X applications are usually developed through model-based development, e.g., with MATLAB®/Simulink®. The engineers focus on implementing and testing the actual application, not on implementing specific protocols and standards in the model. The new dSPACE V2X Blockset for Simulink therefore supports and provides easy access to the V2X world, from fast function development (rapid control prototyping) to testing complete applications (figure 2). The blockset provides dedicated blocks for preparing, coding, transmitting, decoding, and managing V2X messages (CAMs or DENMs). The contents of each message are provided as signal vectors in Simulink. To have a clear overview, the users can configure a filter so only the message contents that are required for an application are displayed in the model. The coding and

>>

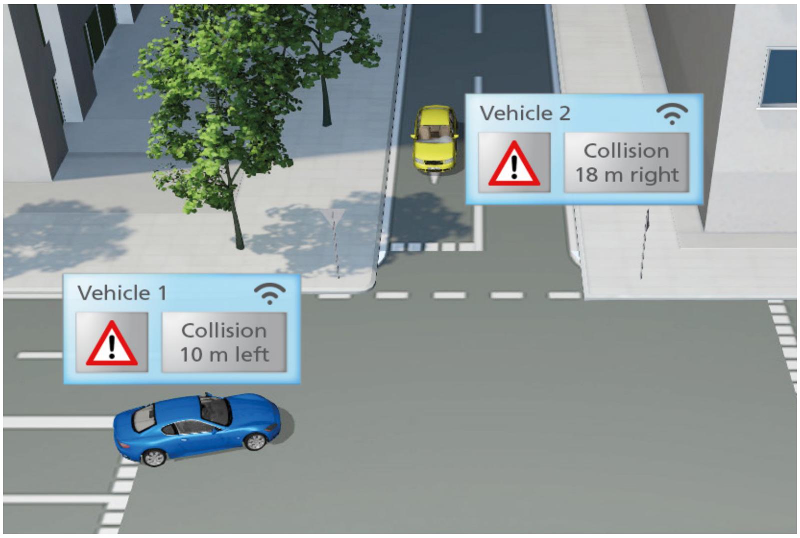


Figure 1: Vehicles exchange their motion data and calculate the probable paths. This example of an intersection assistant shows that the drivers are warned in time when there is the risk of a collision.

decoding blocks are automatically generated from the ASN.1 description standardized by ETSI. This ensures that the V2X blockset can easily be adjusted to new versions of the description file. The development platform, test platform, and dSPACE V2X Blockset are connected to the radio channel via a V2X hardware adapter, such as the MK5-OBU by Cohda Wireless. The adapter is connected via Ethernet UDP/IP and uses the standardized Basic Transport Protocol (BTP) to transmit messages. The GPS receiver in the MK5-OBU can be used to capture position data. Developers can also use a dedicated blockset for evaluating GPS data according to the NMEA-0183 standard.

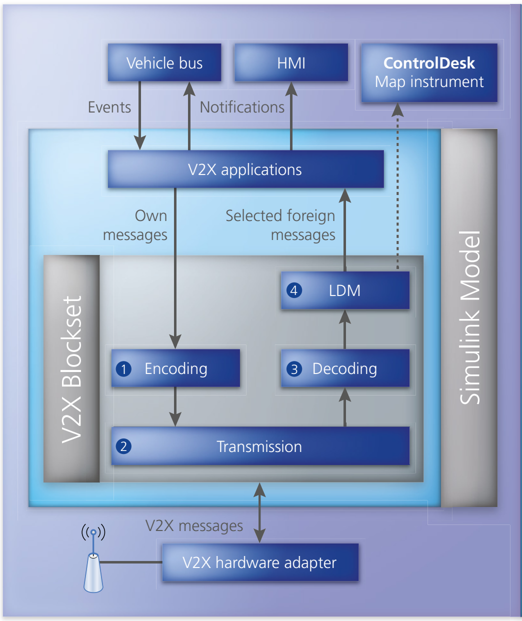
Efficient Message Management with Local Dynamic Map

A key factor for receiving V2X messages is the Local Dynamic Map (LDM). The map stores, manages and distributes all relevant information on the local traffic situation (vehicle positions, speed, the state of traffic lights, weather information, road slickness, etc.) and is updated continuously. The V2X applications first register at the LDM for receiving specific message contents, such as all DENMs with warnings against broken-down cars. The LDM then automatically assigns relevant information to the applications. If messages are obsolete or refer to objects that are too far away, they are discarded automatically.

Highlight: Map Instrument in ControlDesk

The V2X solution adds a specially developed map instrument to the

Figure 2: dSPACE V2X Blockset for developing and testing V2X applications.



① In the first step, if a V2X application detects a relevant event, e.g., the end of a traffic jam, appropriate messages are prepared for sending. The messages are then converted to a compact format by using dedicated encoder blocks for CAM and DENM. After that, the messages are sent.

② Transmission blocks are the link between the Simulink model and the V2X hardware adapter. They transmit messages via Ethernet (UDP/IP) in both directions using the Basic Transport Protocol (BTP) standardized by ETSI. The V2X adapter sends and receives messages in the form of WLAN packages according to the IEEE-802.11p standard.

③ Dedicated decoder blocks for CAM and DENM unpack received messages and forward them to the LDM or directly to a V2X application.

④ The LDM is the central collection and distribution unit for all received messages. It assigns only the required messages to each registered V2X application according to a standardized filter method. The messages created by the applications are transmitted to the final recipients via the vehicle bus or the HMI.

familiar ControlDesk features that support the application and test engineers in manipulating message contents, recording data, etc. The LDM feeds information into the map instrument and uses a map to show the current traffic participants in a V2X network and their movements. The instrument very clearly displays what the V2X application sees. The intuitive handling of the map instrument makes the data analysis much easier. ■

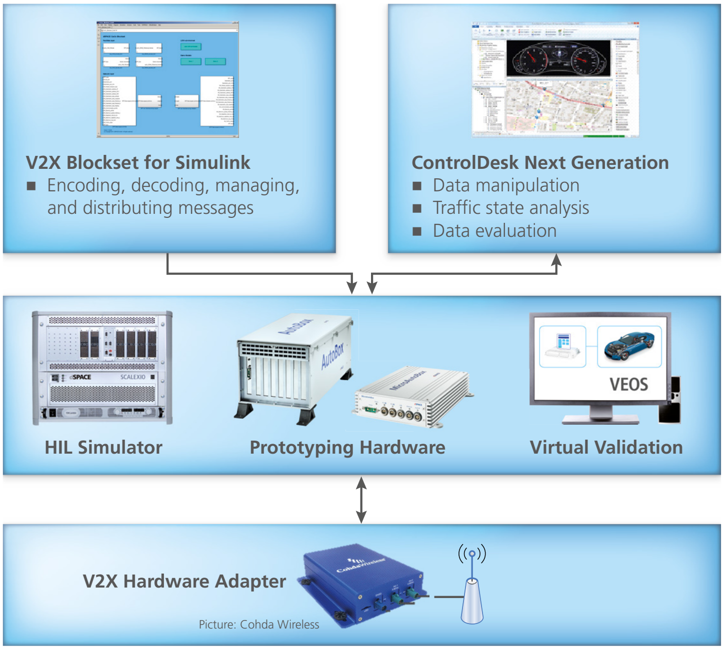


Figure 3: Development and test environment for V2X applications.

Summary and Outlook

With the new V2X solution, dSPACE meets the new requirements that V2X technology brings for development and test systems. The solution can be integrated seamlessly into existing tool chains and offers comprehensive support, from implementing a V2X application up to testing (figure 3). dSPACE is also planning to release a test catalog with a selection of the tests specified by the C2C-CC.

Glossary

Ad-hoc network	Wireless communication network that establishes itself spontaneously and configures itself independently.
ASN.1	Abstract Syntax Notation One. Description language for describing data structures.
BTP	Basic Transport Protocol. Data transport protocol for use in intelligent traffic systems.
C2C-CC	CAR 2 CAR Communication Consortium. Consortium of automotive manufacturers, suppliers, tool suppliers and research institutions with the goal of increasing the safety and efficiency of traffic on the roads through the use of cooperative and intelligent systems based on V2X.
C2X (Car2X)	Car-to-X. Synonym for ad-hoc communication in a traffic system. The 'X' stands for other vehicles and parts of the infrastructure, such as traffic lights and road signs (see V2X).
CAM	Cooperative Awareness Message. Message about the position, speed, type designation, state, etc., which each participant of the V2X network continually sends.
CEN	Comité Européen de Normalisation. European committee for standardization in all technical fields except electrical engineering and telecommunications (see ETSI).
DENM	Decentralized Environmental Notification Message. Message for specific events: e.g., accidents, danger spots, etc.
ETSI	European Telecommunications Standards Institute. European institute for telecommunications standards.
HMI	Human-Machine Interface. The interface between a machine and the person operating that machine.
IEEE 802.11p	Standard for establishing WLAN technology in vehicle ad-hoc networks. Known in Europe as ITS-G5.
LDM	Local Dynamic Map. Database for storing the current state of traffic in the vehicle environment.
NMEA 0183	A communications standard defined by the National Marine Electronics Association, which is also used for communication between GPS receivers and PCs or mobile devices.
OBU	Onboard Unit.
V2X	Vehicle-to-X (see C2X or Car2X).

AUTOSAR Central

Moving from AUTOSAR 3.x to AUTOSAR 4.x

Change with Ease

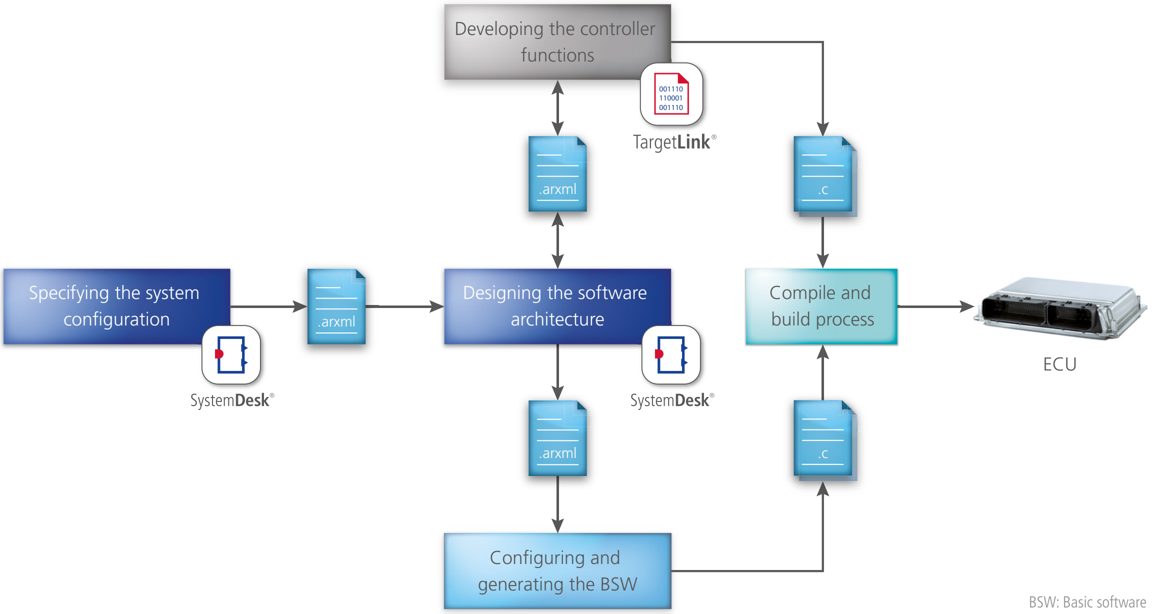
The AUTOSAR standard stands for safety and for exchanging and reusing software components. dSPACE offers comprehensive support for switching up from AUTOSAR 3 to AUTOSAR 4.

AUTOSAR 4 provides many more functions than AUTOSAR 3 in areas such as functional safety, multicore applications, and describing timing requirements. Many companies wish to use the new functions and therefore switch up to the new version. Others, like some suppliers, have to use the new version when their customers migrate to AUTOSAR 4.

Reuse through Change

A main feature of the AUTOSAR standard is that developers can reuse tried-and-tested components, which reduces the amount of development work for follow-up projects. Switching up

>>



BSW: Basic software

Figure 1: Elements of a potential AUTOSAR tool chain that have to be considered during migration.

from version 3 to 4 pursues the same goal. Ideally, existing models that were modeled according to AUTOSAR 3 should be migrated to AUTOSAR 4 automatically. Then, the new features and functions of AUTOSAR 4 can be used for further development. A complete migration involves several work steps, and dSPACE provides seamless support if needed.

Migrating the Component Code

For migrating individual software components TargetLink®, dSPACE's production code generator, is more than just suitable. In TargetLink, when generating production code from the individual functions, the user can define which AUTOSAR version the code will be based on. Because the algorithms for open and

Table 1: dSPACE provides comprehensive support and consultation for the AUTOSAR migration.

Service (Examples)	Details (Extract)
Integrating the AUTOSAR tools into the development process	▪ Integration with basic software configuration tools ▪ Integration with TargetLink and other behavior modeling tools ▪ Adjustments to project-specific solutions for data management (1-D and 3-D motion platforms, steering test benches, piggyback modules for FPGA Base Board)
Migrating existing architectures	▪ Importing existing system and software information from non-AUTOSAR formats ▪ Integrating non-AUTOSAR code
AUTOSAR support	▪ Support for introducing the AUTOSAR-compliant development of software architectures
Model analysis and advice	▪ Support for analyzing your models and modeling types with regard to efficiency, safety, and reusability ▪ Support for creating and optimizing specific guidelines and in applying industry-proven standards
Automation	▪ Developing project-specific scripts (e.g., mapping architecture elements based on naming conventions, connecting software components based on customer rules)

closed loop controls are independent of AUTOSAR versions, they only have to be connected to version-specific data. This means that only this connected data has to be adjusted for AUTOSAR 4. TargetLink stores the data in a version-independent Data Dictionary, so AUTOSAR 4-compliant code can be generated just by changing one global property.

Migrating the Architecture

Due to the large and complex differences between the two AUTOSAR versions, complete system architectures, and larger AUTOSAR system, extracts cannot be migrated at the click of a button. Some processes can be automated via scripts, but these scripts have to be adjusted to each individual case. A converter makes it possible to automatically transform all AUTOSAR 3 elements that are part of the AUTOSAR software component template. This ensures that no information is lost. However, the converter does not create new elements that were added with AUTOSAR 4. dSPACE confers with the customer about what the architecture should look like with AUTOSAR 4. With this information, dSPACE Engineering Services creates customized scripts that generate the desired architecture via the SystemDesk automation interface. Conversion takes place either at dSPACE or at the customer. dSPACE offers various engineering services for the migration (table 1).

Adjusting the Tool Chain

For a successful migration, developers not only have to keep the AUTOSAR files in mind, but also the entire underlying tool chain. When files are migrated from AUTOSAR 3 to 4, it might be necessary to also update the AUTOSAR tools that are being used, so the new AUTOSAR 4 files can be edited later (figure 1). Here, dSPACE offers a mature tool chain as well as individual advice and support, letting users benefit from dSPACE's long-standing project experience. In close cooperation with the customer, dSPACE checks which data needs to be migrated, what type of data exists, and whether a single software component or a complete software architecture needs to be migrated. The solution for the migration is then tailored to the individual project needs. dSPACE also offers training on AUTOSAR 4 to make customers familiar with the changes in the new version.

Validating with SystemDesk

After the migration, dSPACE SystemDesk® can be used for extensive validation. Either the complete system architecture is imported into SystemDesk or individual software components are imported, connected, and integrated to ECU software. SystemDesk Version 4 supports the complete AUTOSAR 4 data model and offers multi-user support. Integrated validation processes let users check the project for consistency and completeness. SystemDesk can also generate virtual ECUs (V-ECUs) from ECU software. The V-ECUs can be simulated on the developer PC without additional hardware, with the simulation platform dSPACE VEOS. ■

Differences Between AUTOSAR 3 and 4

Some features of AUTOSAR 3 cannot be mapped automatically to AUTOSAR 4 features. Some manual effort is involved in their migration. Some examples for AUTOSAR 4 include the application data types (ADT) for physical information such as units, limits, or scalings, and the implementation data types (IDT) for defining the data type, such as integer. A data type mapping set assigns an implementation data type to each application data type for a software component. AUTOSAR 3 provides only data types (DT) that contain both types of information. There are many different ways to generate ADTs, IDTs, and data type mapping sets for AUTOSAR 4 from the AUTOSAR 3 data types. The mapping can therefore not be automated. It must be defined for each project.

Action and Analysis

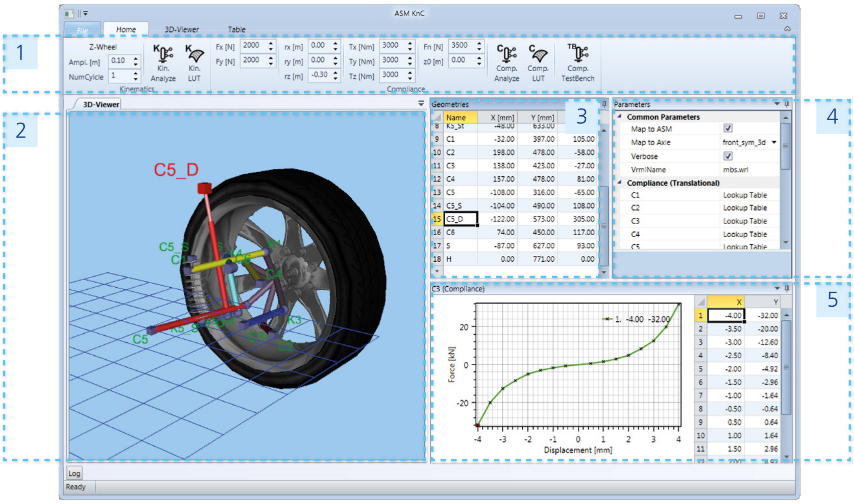
Designing and testing wheel suspensions more efficiently

Virtual test rigs shift complex mechanical axle designs and a majority of their trials away from the test track onto the developer's desk. In virtual test drives, the modeled suspensions have to prove what they are capable of.

Automotive Simulation Models (ASM) are a tool suite for the real-time simulation of automotive applications, such as in the field of vehicle dynamics. If the wheel suspensions of the simulated vehicle need to be examined more closely during the vehicle dynamics simulation, the special tool ASM KnC provides readily available help. ASM KnC (Kinematics and Compliance) is a virtual axle test rig that supports the design and analysis of wheel suspensions. It gives engineers the ability to run virtual tests on the wheel suspensions of many vehicle variants, optimize the suspensions, and reuse them for hardware-in-the-loop (HIL) tests.

Intuitive Graphical Handling

The current version of ASM KnC, 7.0, comes equipped with a completely redesigned user interface and improved user navigation. From the included templates, users choose the suspension type. The templates include customary suspensions, such as McPherson, double wishbone, 3-link, 4-link, multi-link, and more. The exact geometry, the pivot points, and the bushing stiffness can be defined intuitively, either graphically or numerically. CAD data or informa-



Ampl. [m] 0.10
NumCylcle 1
Fx [N] 2000
Fy [N] 2000
rx [m] 0.00
ry [m] 0.00
rz [m] -0.30
Tx [Nm] 3000
Ty [Nm] 3000
Tz [Nm] 3000
Fn [N] 3500
z0 [m] 0.00

	Name	X [mm]	Y [mm]	
8	K5_St	-48.00	633.00	
9	C1	-32.00	397.00	105.00
10	C2	198.00	478.00	-58.00
11	C3	138.00	423.00	-27.00
12	C4	157.00	478.00	81.00
13	C5	-108.00	316.00	-65.00
14	C5_S	-104.00	490.00	108.00
15	C5_D	-122.00	573.00	305.00
16	C6	74.00	450.00	117.00
17	S	-87.00	627.00	93.00
18	H	0.00	771.00	0.00

Common Parameters
Map to ASM ☑
Map to Axle front_sym_3d
Verbose ☑
VrmlName mbs.wrl
Compliance (Translational)
C1 Lookup Table
C2 Lookup Table
C3 Lookup Table
C4 Lookup Table
C5 Lookup Table

	X	Y
1	-4.00	-32.00
2	-3.50	-20.00
3	-3.00	-12.60
4	-2.50	-8.40
5	-2.00	-4.92
6	-1.50	-2.96
7	-1.00	-1.64
8	-0.50	-0.64
9	0.50	0.64
10	1.00	1.64
11	1.50	2.96

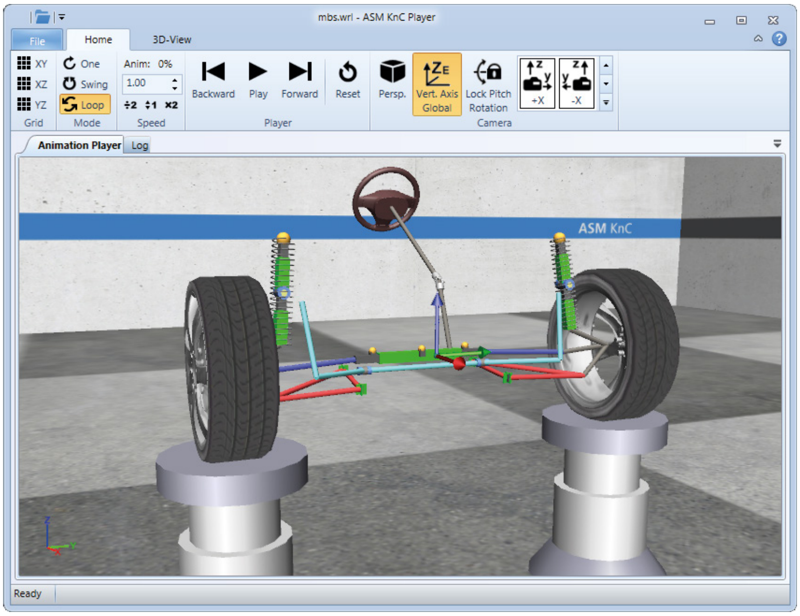
The graphical user interface of ASM KnC:
1) test rig control, 2) interactive 3-D preview window, 3) definition of axle geometry
4) configuration management, 5) definition of bushing stiffness.

tion from the supplier's data sheets, for example, provide the basis for this. The axle design can be animated immediately with a relevant wheel excitation on the test rig. Thus, the design can be rotated freely in space and inspected visually.

Workflow and Advantages

An ASM KnC test rig lets users simulate and examine the kinematic rotations and displacements of the wheel under the influence of vertical deflection and steering rod movement/displacement, and the elastokinematic dependencies under the influence of forces and torques. The defined design can be inserted as look-up tables into the ASM Vehicle Dynamics simulation model, where it is used for real-time-capable vehicle dynamics simulations. Because ASM KnC can be completely automated, users can conduct parameter studies iteratively without manual adjustments. For example, a script can be used to automatically alter a coupling point and analyze the influence on the vehicle dynamics simulation. This helps determine the most suitable axle design for the defined driving maneuvers, thereby reducing the test effort involved in using test vehicles and real test rigs. ASM KnC is therefore one of the key factors in frontloading tests, speeding up vehicle development. ■

This video shows the workflow with ASM KnC.
www.dspace.com/go/dMag_20153_KnC



Anim: 0%
1.00

Animation of the front wheel suspension and steering.

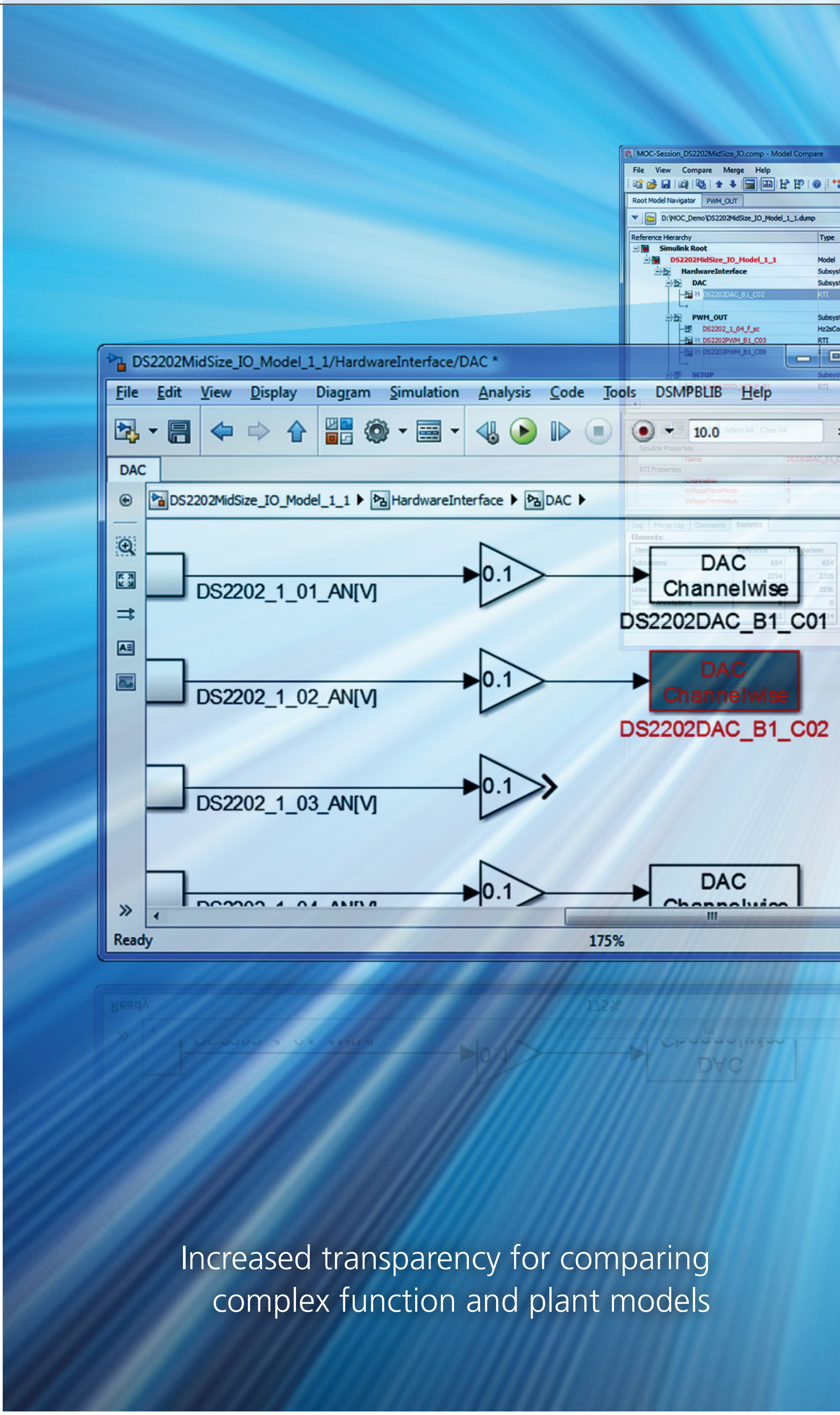
Application Examples

Model parameterization –
Generating kinematics and compliance look-up tables for vehicle dynamics models.

Analyzing wheel suspensions –
Checking the axle modifications via clear visualizations.

Analyzing vehicle dynamics –
Checking the effect of axle modifications (kinematics and bushing compliance) in complete vehicle dynamics models. Faster than in real time.

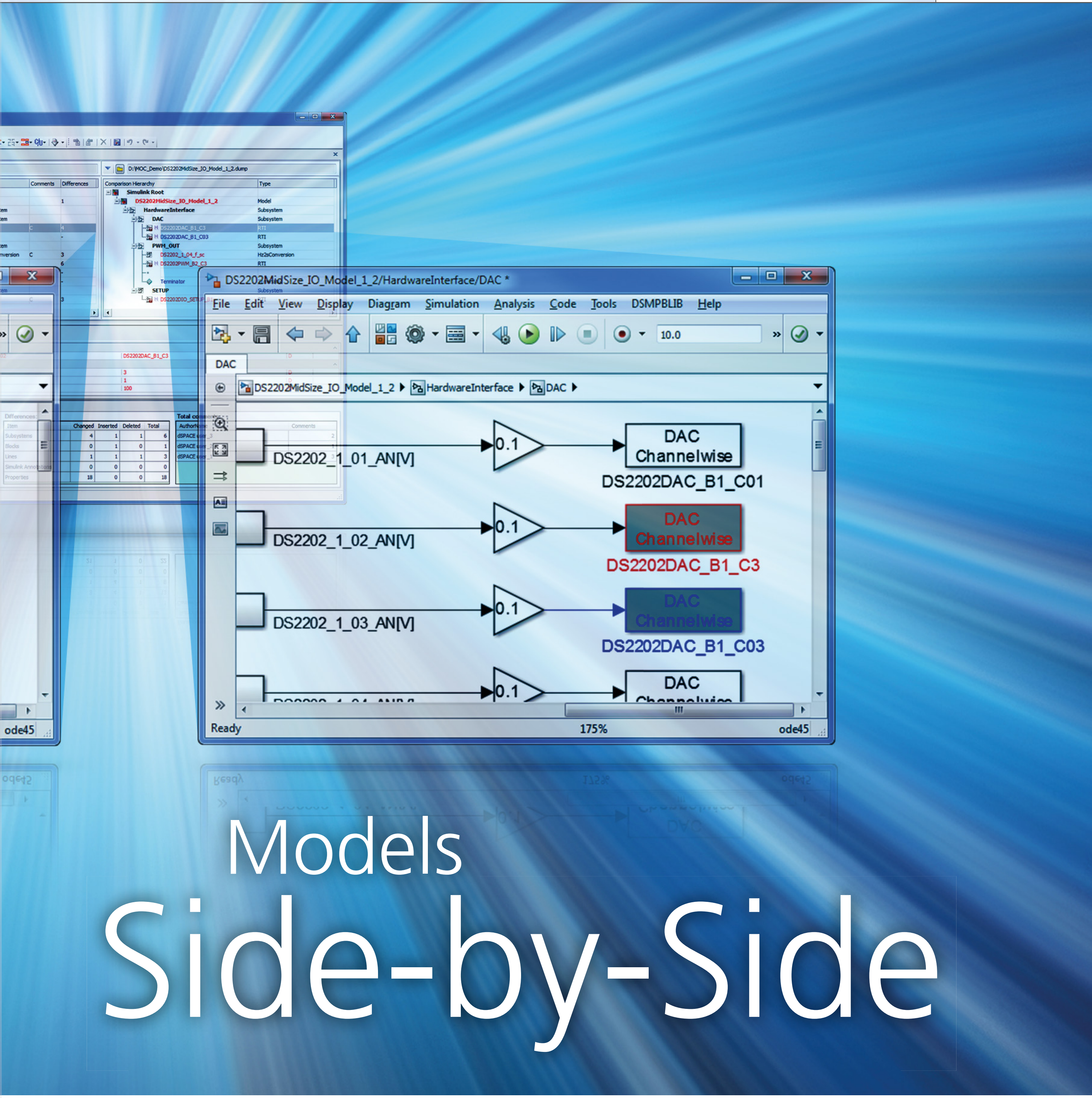
Virtual optimizations –
Optimizing wheel suspensions automatically. The goal: Improving the behavior of the vehicle's dynamics early on.



Increased transparency for comparing complex function and plant models

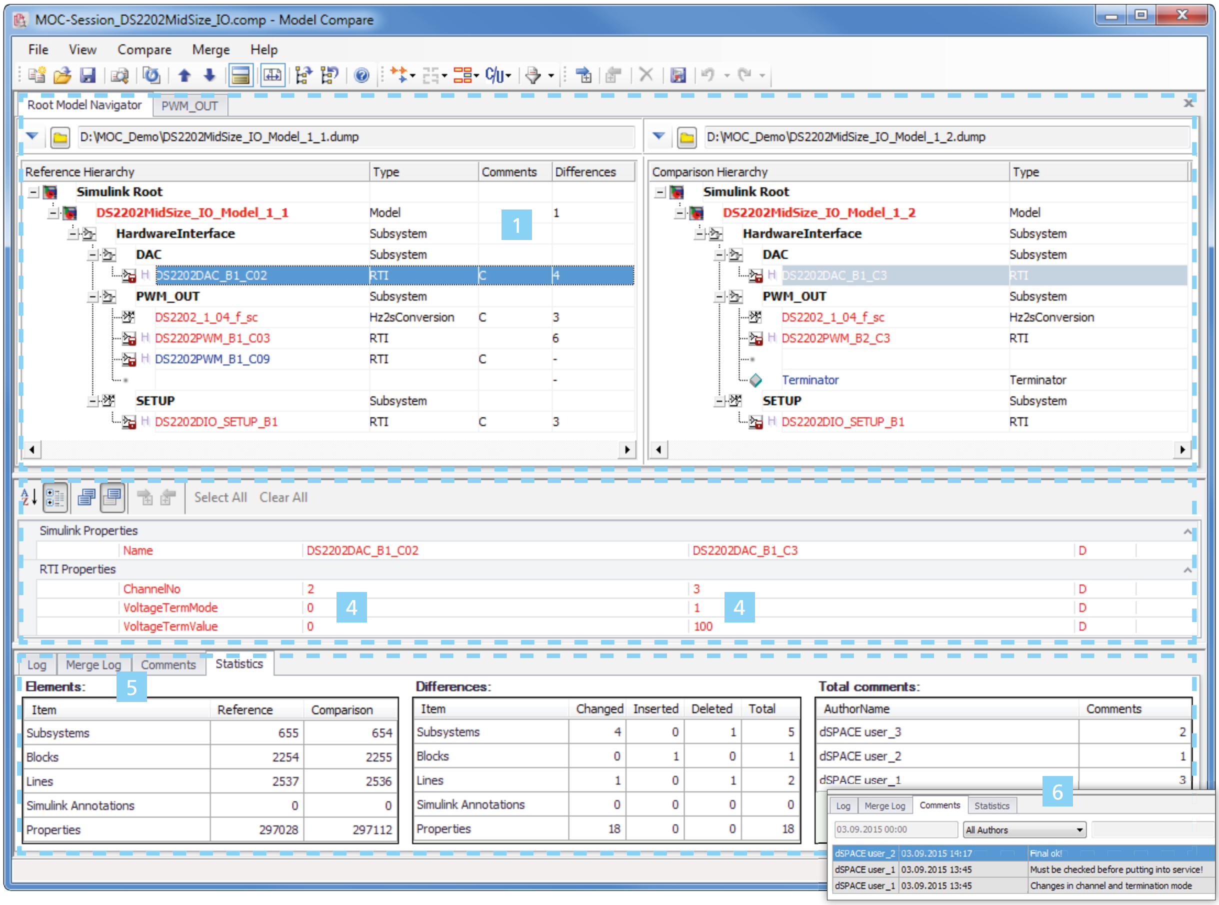
Find and structure all the differences between two model versions quickly and easily: That is what the well-established dSPACE tool Model Compare does. Even large models can be visually compared in minutes, which would be practically impossible without tool support. A very powerful comparison algorithm ensures that all model differences are detected and block correspondences are identified, even if their positions, individual properties or even names have changed. Thanks to the integrated TargetLink support, Model Compare gives a precise overview of the model differences that are relevant to the users of the dSPACE production code generator. Redundant and implementation-specific data, such as data underneath block masks, is hidden by default. This makes comparing models not only effective but also very efficient. A new mechanism in Model Compare 2.6 lets you use this easily readable and sustainable overview also for other Simulink-based model libraries. This means that developers from other areas, such as rapid control prototyping (RCP) and hardware-in-the-loop (HIL) simulation, can take advantage of the useful Model Compare functions even more efficiently than before.

>>



Models Side-by-Side

Model-based development often involves different versions of one model. Sometimes it is difficult to keep up. What exactly were the differences between the versions? Model Compare creates clarity – with version 2.6 now for HIL and many other models.



D:\MOC_Demo\DS2202MidSize_IO_Model_1_1.dump

Reference Hierarchy	Type	Comments	Differences
Simulink Root			
DS2202MidSize_IO_Model_1_1	Model		1
HardwareInterface	Subsystem		
DAC	Subsystem		
DS2202DAC_B1_C02	RTI	C	4
PWM_OUT	Subsystem		
DS2202_1_04_f_sc	Hz2sConversion	C	3
DS2202PWM_B1_C03	RTI		6
DS2202PWM_B1_C09	RTI	C	-
			-
SETUP	Subsystem		
DS2202DIO_SETUP_B1	RTI	C	3

D:\MOC_Demo\DS2202MidSize_IO_Model_1_2.dump

Comparison Hierarchy	Type
Simulink Root	
DS2202MidSize_IO_Model_1_2	Model
HardwareInterface	Subsystem
DAC	Subsystem
DS2202DAC_B1_C3	RTI
PWM_OUT	Subsystem
DS2202_1_04_f_sc	Hz2sConversion
DS2202PWM_B2_C3	RTI
Terminator	Terminator
SETUP	Subsystem
DS2202DIO_SETUP_B1	RTI

Simulink Properties

Name	DS2202DAC_B1_C02	DS2202DAC_B1_C3	D

RTI Properties

ChannelNo	2	3	D
VoltageTermMode	0	1	D
VoltageTermValue	0	100	D

Elements:

Item	Reference	Comparison
Subsystems	655	654
Blocks	2254	2255
Lines	2537	2536
Simulink Annotations	0	0
Properties	297028	297112

Differences:

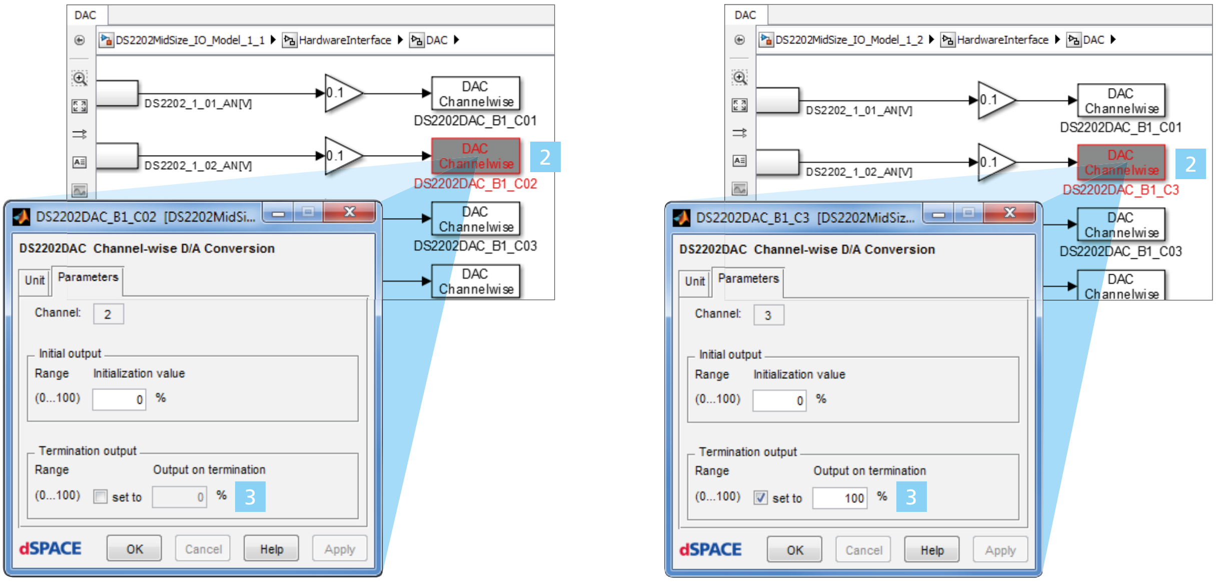
Item	Changed	Inserted	Deleted	Total
Subsystems	4	0	1	5
Blocks	0	1	0	1
Lines	1	0	1	2
Simulink Annotations	0	0	0	0
Properties	18	0	0	18

Total comments:

AuthorName	Comments
dSPACE user_3	2
dSPACE user_2	1
dSPACE user_1	3

03.09.2015 00:00 All Authors

dSPACE user_2	03.09.2015 14:17	Final ok!
dSPACE user_1	03.09.2015 13:45	Must be checked before putting into service!
dSPACE user_1	03.09.2015 13:45	Changes in channel and termination mode



Channel: 2

Initial output
Range (0...100) Initialization value 0 %

Termination output
Range (0...100) Output on termination ☐ set to 0 %

Channel: 3

Initial output
Range (0...100) Initialization value 0 %

Termination output
Range (0...100) Output on termination ☑ set to 100 %

1 The synchronized hierarchy trees in Model Compare’s user interface show all the differences in the models.

2 Comparison of RTI blocks. Model Compare detects the settings 3 made in the configuration dialogs and presents them clearly in the Property Inspector 4 .

5 The Statistics Viewer of the Tool Window provides a summary of all the discovered differences and of the models themselves.

6 Model Compare supports review comments and complex review sessions involving several participants.

> Complex models can be compared in minutes, which would be practically impossible without tool support.

All Differences at a Glance

The convenient graphical user interface of Model Compare shows the comparison results in well-structured, color-coded synchronized hierarchy trees (page 60, No. 1). These trees immediately show you which model elements belong together and which elements were changed, added, or deleted. Initialization routines for the models and their environment or tool chains can also be considered. To illustrate differences in particular, they can be highlighted in different colors directly in the Simulink®/TargetLink® models (page 60, No. 2). This makes it easy for you to graphically inspect them in their respective model context. The displays of the model and the hierarchy tree are connected so that you can trace the differences in both directions with just one click.

Comprehensive Filter Options

To make your work as efficient as possible, Model Compare provides many different filter options. For example, you can use different display filters to specify which model elements to show in the central display area (page 60, No. 1): only the changed blocks or signal lines, the added or deleted ones, or a combination of the two. Model Compare also offers several predefined filters that let you focus on a specific kind of difference. For TargetLink models, e.g., you can focus on all functional changes or only on implementation-specific changes. The tool also lets you define your own filters to exclude one or several element properties, or even entire model elements from the comparison. To reuse the defined filter settings in other projects, you can save them as favorites.

Review and Merge Support

During a model review with Model Compare, developers can add review comments to differences at the block or property level. The tool automatically adds the time stamps and author information to support even complex reviews with several participants (page 60, No. 6). If you want to merge parallel development branches or transfer changes from one model variant to another, you can do this right in the comparison views via user-friendly commands, such as Copy to Right or Copy to Left. Merging is possible at the element and property levels. You can merge individual properties, model elements or entire subsystems. During the merge process, a smart line handling option makes sure that the appropriate connections are copied or deleted as well. All of the merge operations are logged in the Merge Log Viewer, which is part of Model Compare's Tool Window (page 60, No. 5).

Documenting the Comparison Results and Tool Automation

The comparison results can be saved as a PDF, HTML or XML report so you can pass them to other colleagues or archive them. Review comments, filter settings and screenshots of the models can be integrated in the reports so that you can also use them for model reviews. A powerful application programming interface (API) lets you automatically start model comparison or create comparison reports. It also makes it easy to integrate Model Compare into existing tool chains.

New Add-on Mechanism

In addition to pure Simulink, Stateflow and TargetLink models, Model Compare can also compare models with any Simulink-based block library. A new add-on mechanism in Model Compare 2.6 lets you use hook scripts to integrate block-specific knowledge in the comparison of any number of models. Model differences in mask variables or block dialog parameters can therefore be displayed immediately (page 60, No. 4). This means that Model Compare now provides precise and efficient model comparisons, e.g., for RCP or HIL models. This of course includes the Real-Time Interface (RTI) blocksets by dSPACE. ■

Conclusion

With the new add-on mechanism, Model Compare 2.6 now provides an effective and efficient model comparison for models with any Simulink-based block libraries. This means that developers from different areas, such as rapid control prototyping (RCP) or hardware-in-the-loop (HIL) simulation, benefit from the useful Model Compare functions and use the tool to compare their plant or I/O models.

Limited availability outside Europe and Asia.
Please contact dSPACE.

How can I test everything from complex networked functions up to complete vehicles as early and flexibly as possible? How can I save costs by reusing test artifacts from one process step in the next one? dSPACE test systems support you in all of your current and future challenges.

No one can exactly predict how vehicles will develop in the long term, but if the innovations of recent years do show a general direction, the path is leading to more, and more complex electrics/electronics (E/E) functions in the vehicles. In today's vehicles, 100 million lines of code for an entire ECU network are already a reality in many places, and the complexity is continuing to grow due to factors such as driver assistance systems. The challenges on validation and test systems to validate this complexity are also increasing.

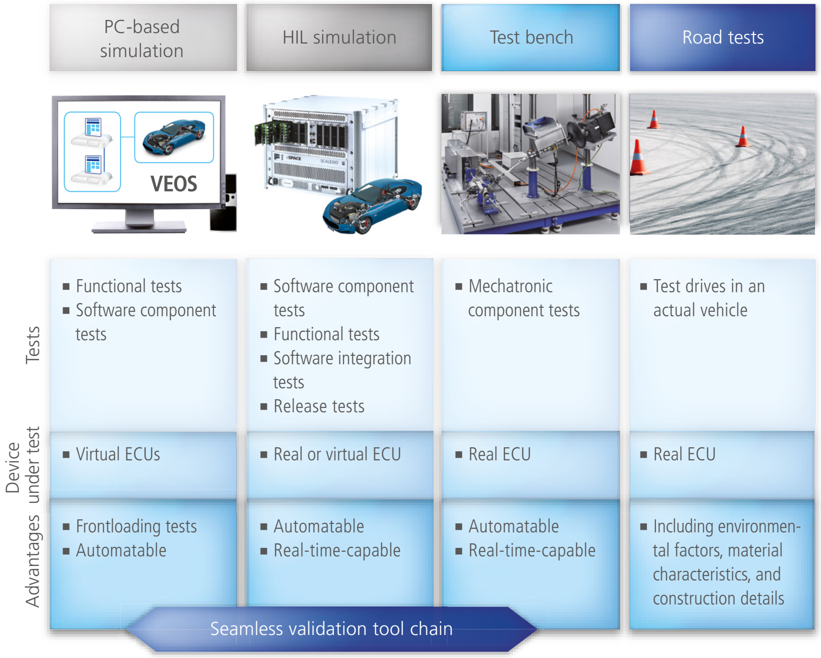
Enormous Challenges

There are several factors that affect future test processes and test systems:

- *Networked functions*
  The new driver assistance functions require networked sensors and actuators that give them information about the environment and other road users. This requires detailed simulation models of the vehicles, the sensors, and the environments. The numerous electronic control units (ECUs) interact tightly with each other. Besides the classic vehicle bus systems, new communication networks such as CAN FD and Ethernet are also being used. Their behavior must also be validated in the tests.
- *Vehicle and model variants*
  The increasing level of networking between functions and ECUs is accompanied by a wide range of variants and models and also new drive concepts such as electric and hybrid vehicles. This increases the diversity of the ECUs and embedded software to be validated, because ECUs are used across several vehicle variants. More and more, the decisive factor for the test system is an intelligent data management which provides process reliability.
- *Special requirements for combustion engines and electric motors*
  New technologies in the field of battery management and electric motors are changing the validation process because, in comparison to conventional drives, significantly higher currents and faster control algorithms have to be considered. For internal combustion engines, new emissions laws are causing an increased use of exhaust gas treatment systems and more precise injection systems which, in turn, need to be included in validation tests.
- *Standards and norms*
  Test systems are increasingly influenced by binding standards and norms, such as ISO 26262 for the development of safety-relevant E/E systems in motor vehicles.

>>

Figure 1: The dSPACE tool chain is used seamlessly throughout several test phases.



	PC-based simulation	HIL simulation	Test bench	Road tests
Tests	▪ Functional tests ▪ Software component tests	▪ Software component tests ▪ Functional tests ▪ Software integration tests ▪ Release tests	▪ Mechatronic component tests	▪ Test drives in an actual vehicle
Device under test	▪ Virtual ECUs	▪ Real or virtual ECU	▪ Real ECU	▪ Real ECU
Advantages	▪ Frontloading tests ▪ Automatable	▪ Automatable ▪ Real-time-capable	▪ Automatable ▪ Real-time-capable	▪ Including environmental factors, material characteristics, and construction details

Seamless validation tool chain

Quo Vadis, Test?

Single-source test solutions

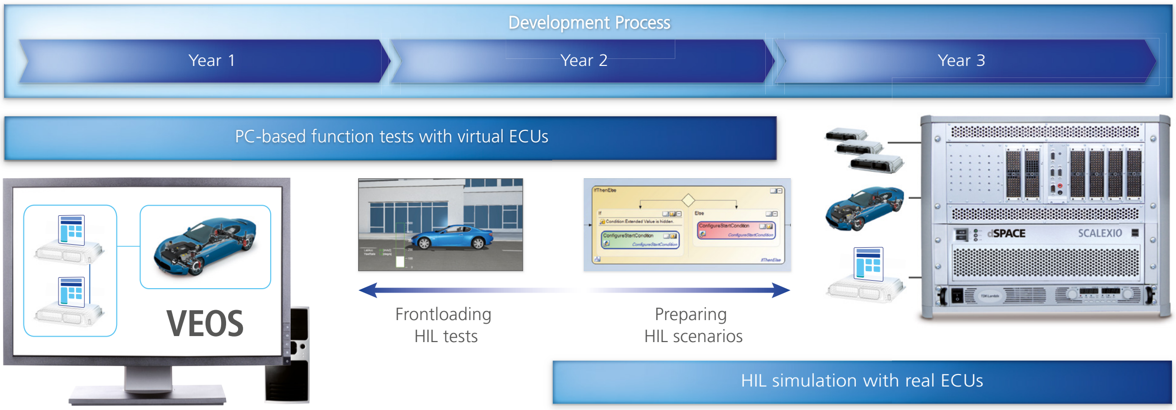


Figure 2: The combination of PC-based simulation and HIL tests enables early function validation and test scenario preparation throughout the entire development process.

Seamless Test Systems

To make the complex challenges manageable, dSPACE offers seamless test system solutions from a single source. In these solutions, hardware-in-the-loop (HIL) tests and driving tests play an important role, complemented by pure software simulation with virtual ECUs (figure 1).

Early Validation on the PC

The increasing number of model variants, high variety of functions, and shorter development cycles are making it more difficult, even impossible, to use prototype vehicles to perform all the tests. Beside HIL simulation, validation on a PC makes earlier tests possible, as functional tests are frontloaded to the earlier development stages. dSPACE provides the PC-based simulation platform VEOS® for this task. This gives function developers their own test platforms to perform function tests with virtual ECUs. They can verify the development steps at any time and cost-effectively.

Reliable Real-time Tests on HIL Simulators

With the HIL simulator SCALEXIO®, HIL tests follow directly after PC-based simulation. HIL simulation is a well-established, efficient process step for automatically validating the ECUs that are built into the vehicles later on. In particular, bus communication via CAN/CAN FD, LIN, and Ethernet SOME/IP, for example, can be validated with a dSPACE HIL system reliably and reproducibly in a simulated vehicle environment. dSPACE offers tailored hardware for applications with special requirements. For example, to test electric drives, which require short control cycles and high currents, the dSPACE Automotive Simulation Models (ASM) and FPGA-based hardware are available. For driver assistance applications, dSPACE provides simulations for the vehicle environment and sensors so that the HIL simulator can be used to test the wide variety of traffic scenarios in the virtual world.

Mechatronic Tests on the Test Bench

In some HIL test cases, it is not possible to access the ECU through its electric interfaces alone. As a result, mechanical access is essential for setting up mechanical loads on electric drives, stimulating the integrated sensors for mechatronic ECUs, or stimulating human-machine interfaces, for example. For these tasks, dSPACE provides highly dynamic test benches for mechatronic components and systems in connection with the real-time simulation. For more information, read the interview on page 66.

Process-Reliable Tool Chain

Standards and norms are becoming more important for test systems. For example, ISO 26262 explicitly names HIL tests as a validation step. For process-reliable test environments, dSPACE offers not only appropriate test systems, but also software that meets the standards. The test automation software AutomationDesk has been certified by TÜV SÜD for testing safety-related systems according to ISO 26262 and IEC 61508. This certificate confirms that this software tool is suitable for developing and testing safety-relevant systems in the automobile industry, commercial vehicles, aerospace and many other fields. AutomationDesk is the first test automation software in the area of HIL simulation that has been granted such a certificate.

Openness Through Standards

Test systems often need to be integrated into existing software environments. With its products, dSPACE supports several standards such as AUTOSAR, the Functional Mock-up Interface (FMI), and ASAM XIL API. Standardized interfaces to the dSPACE test systems make it easier to exchange simulation models, for example, between the OEM and supplier (figure 3).

Convenient Data Management with SYNECT

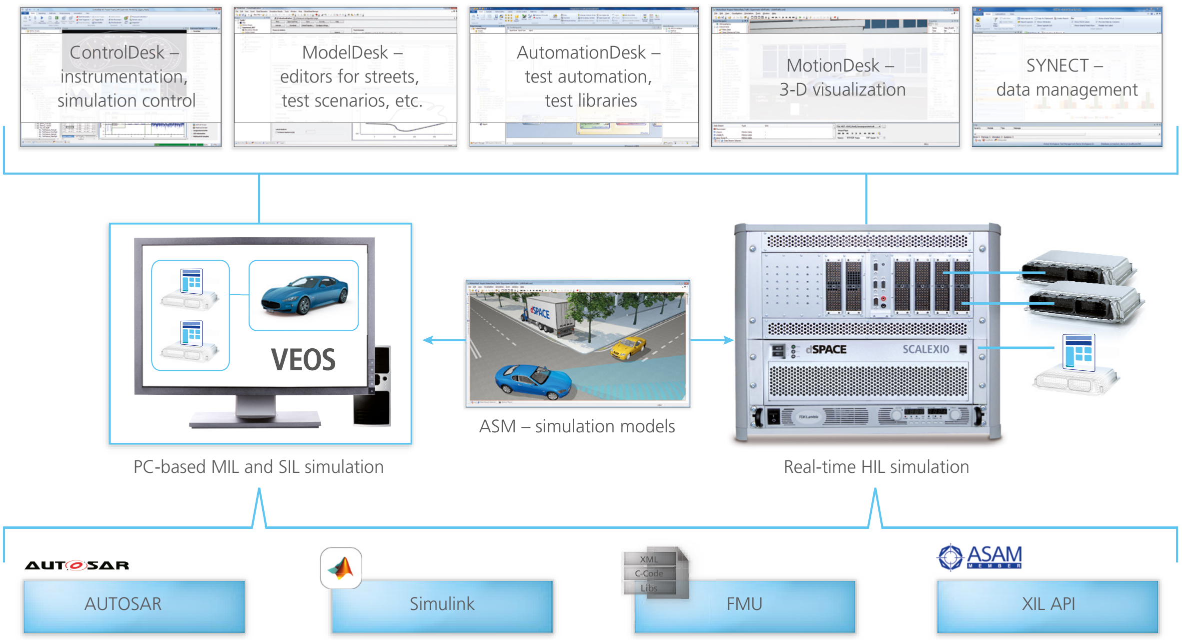
The complexity of the test tasks and test systems generates a huge amount of data. To make the testing process as efficient as possible, the test scenarios, test variants, models and test results must be managed, versioned and stored for easy retrieval. dSPACE SYNECT® is the data management software tool that is designed exactly for these tasks, focusing on the model-based development and ECU tests. SYNECT manages both the data throughout the development process (e.g., models, signals, parameters, tests, test results) and the data dependencies, versions, and variants, including links to the underlying requirements.

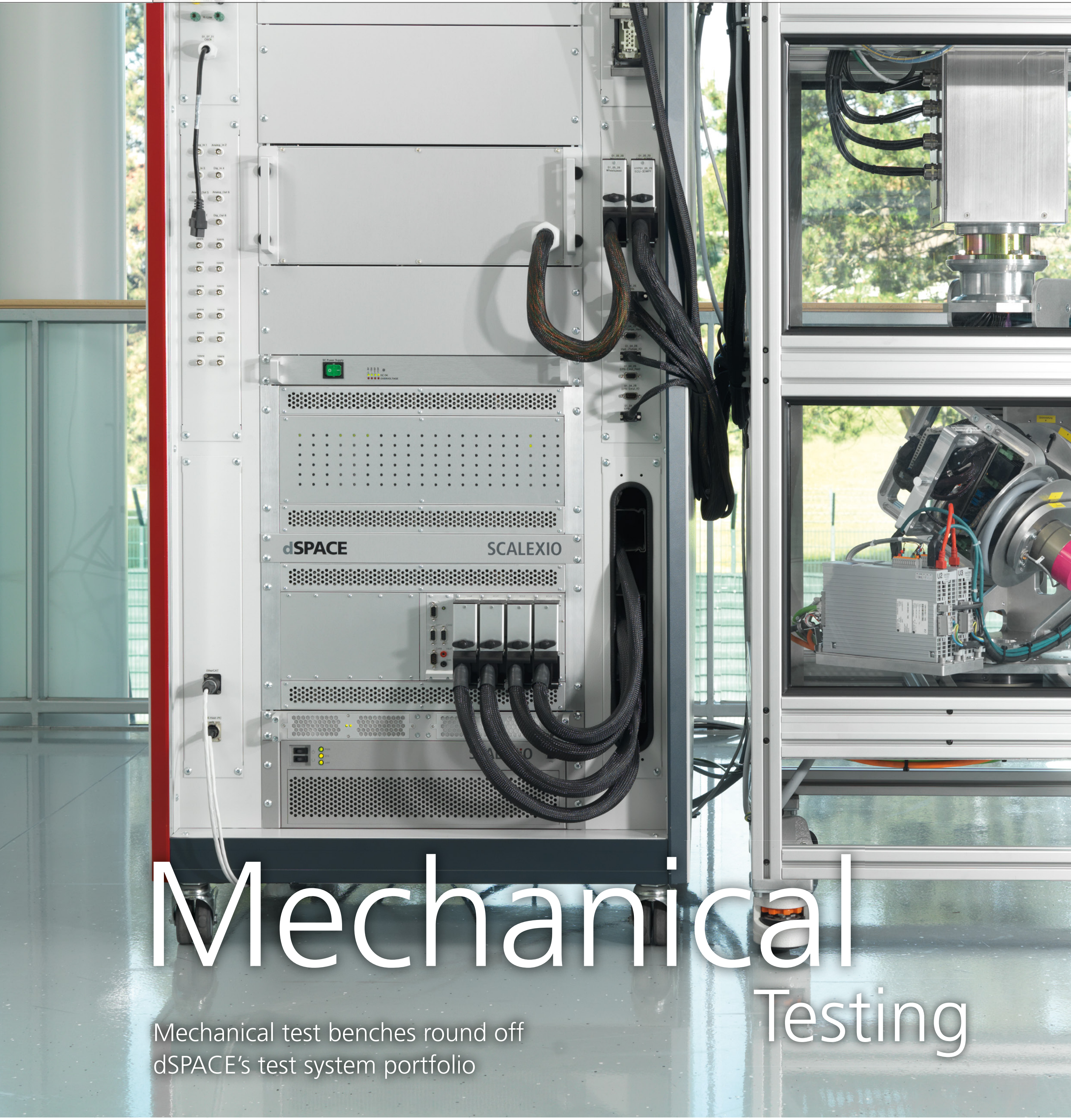
All Solutions from One Source

From PC-based simulation, to HIL test systems, up to mechatronic test benches, dSPACE offers a harmonious hardware and software tool chain. Together with its decades of accumulated experience in setting up systems with many thousands of applications, dSPACE helps customers design effective testing processes, today and in the future.

For further information, visit the dSPACE website at: www.dspace.com/go/dMag_20153_HILE

Figure 3: The seamless tool chain and support of various standards enables the early exchange of test scenarios, models and configurations.



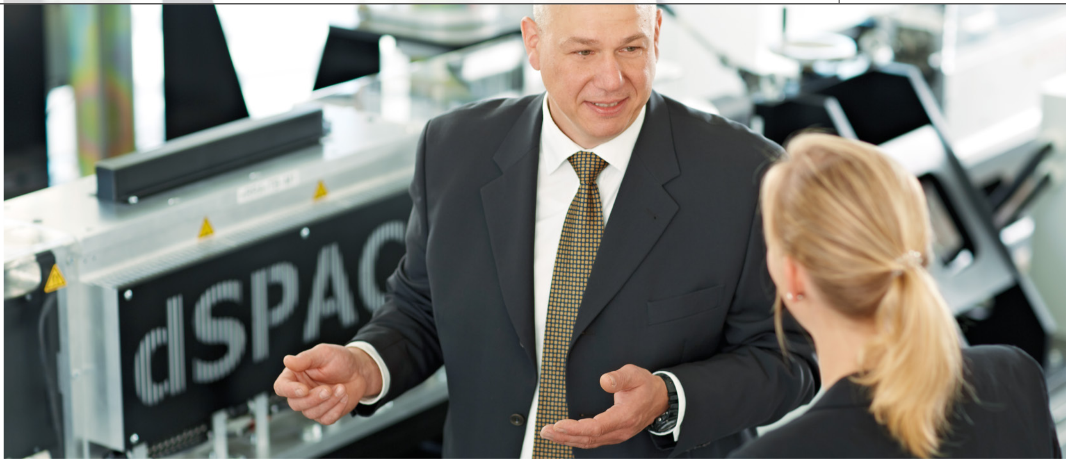


Mechanical Testing

Mechanical test benches round off dSPACE's test system portfolio

Completely validating a complex control system often also includes putting the mechanical components to the test. Matthias Deter, responsible for the setup of mechanical test benches at dSPACE, illustrates the key role that dSPACE test benches play for testing mechatronic systems.

3-D platform for electronic stability control (ESC) tests.



Mr. Deter, dSPACE is well-known as a provider of rapid control prototyping (RCP) systems and hardware-in-the-loop (HIL) simulators for ECU development. Why does dSPACE also provide test benches?
Completely validating an electronic control unit (ECU) under laboratory conditions is often possible only if the test system completely simulates the real ECU environment. More and more ECUs also need a stimulus to their mechanical interfaces and sensors in order to be completely integrated in the test environment. One example is an ECU for ESC, with an integrated yaw rate sensor. Since the goal of dSPACE has always been to offer its customers turn-key HIL test systems, from our point of view it is natural that we also provide test benches, offering complete systems that are optimally adapted to our customers' needs. When we set up these test benches, our experience and products in rapid control prototyping (RCP) are a great help, because the RCP systems drive the necessary load machines.

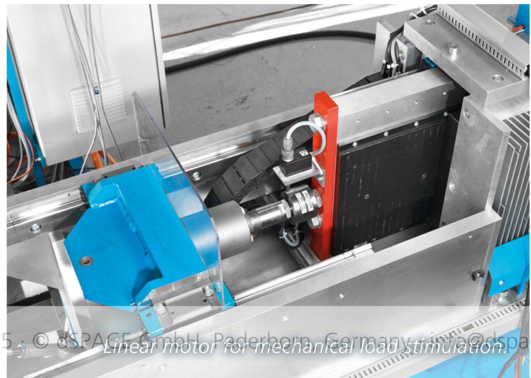
Are test benches a new dSPACE business field?
dSPACE already has a lot of experience in this field. In the past 7 years, we have set up more than 50 test benches and delivered them as turn-key systems to our customers. Test benches are one of our fastest-growing fields.

What are typical applications?
Most of the typical applications are electric power steering systems, and also brake boosters, 3-D motion platforms for vehicle dynamics control systems, and mechanical loads for real pump motors. We also cover smaller automotive applications, such as seat controls, fans, belt tensioners, and electric tank caps. We are not afraid of any application.

How does dSPACE stand out from its market competitors in the field of test benches?
First of all, it is important to know that at dSPACE, customers get all their components from a single source. With other providers, this is

> Turn-key simulators for mechatronic systems from a single source.

Test bench for steering systems (single-sided linear motor).



Linear motor for mechanical load stimulation.

Test bench for a steering wheel.

usually not the case. dSPACE offers a comprehensive portfolio of real-time hardware and simulation models for HIL applications. We also provide customized engineering services for test bench design, construction and commissioning. We plan the mechanical components ourselves and have them manufactured by local specialized companies. Our customers always have exactly one contact: dSPACE. And then there is one test bench characteristic that really stands out: Dynamics! Customers tell us that dSPACE test benches surpass their expectations influenced by experience with other providers regarding control dynamics and accuracy. We rely on the open TWINsync protocol from the company LTI. With this protocol, our real-time hardware offers decisive advantages such as low-latency control with 8 kHz and synchronicity for pulse width modulation of the drive motors. And we naturally are also experienced with other protocols and industrial buses.

Why does dSPACE rely on electric drives for the test benches?

Electric drives have an optimal balance of dynamics and energy efficiency. They also have a manageable, simple infrastructure and are easy to control. If we need to, we can even reach a two-digit kilonewton range for linear motor forces – representing the dynamics that act on a tie rod at high speeds and uneven surfaces, like cobblestones.

How does dSPACE make sure that the test benches also cover the customer's needs?

The test benches are implemented in customer projects. This means that we work closely with our customers to meet and understand their specific requirements in detail. The customer's requirements go directly into the concept of the test bench so that a customized, optimal solution is always the result. In turn, new challenges naturally lead to innovations in our hardware and software. So dSPACE always has an optimal mix of innovative products and customer-specific engineering units.

What measures has dSPACE taken to position itself for processing such projects?

Our internal expertise, tools and processes have been expanded to include the development of mechanical setups. In particular, adequately handling 3-D construction data brings new demands. We also have processes for production release, safety requirements, and quality assurance. Because we are highly experienced in these elementary topics, we can provide test benches that operate safely and efficiently for the system users. Of course we are happy that our customers appreciate our efforts, which the following statement from a premium manufacturer confirms: "I've never experienced this on-schedule delivery and accurate implementation of such a complex project when other companies were involved."

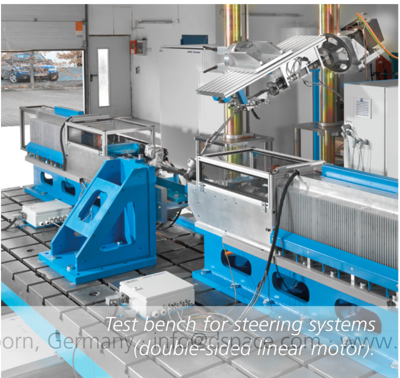
Mr. Deter, thank you for talking to us!

Matthias Deter is Group Manager Engineering, responsible for customer projects with mechanical test benches, at dSPACE in Paderborn, Germany.

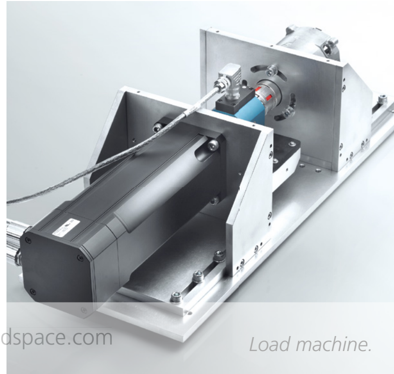
Dynamics is the prime discipline of dSPACE's test benches.

Driving simulator/cockpit.

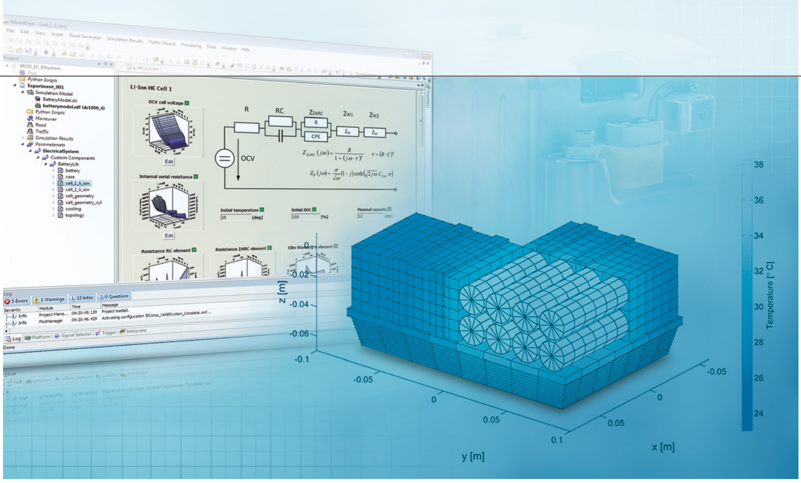
Test bench for electromechanical brakes.



Test bench for steering systems (double-sided linear motor).



Load machine.



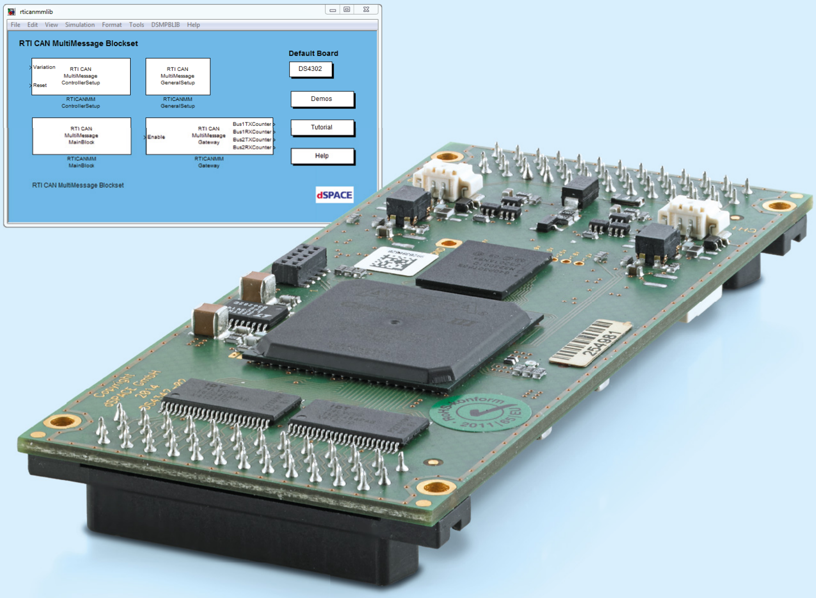
European Union
European Structural
and Investment Funds

Electrical and Thermal Battery Simulation

dSPACE and the Institute for Power Electronics and Electrical Drives (ISEA) of RWTH Aachen University have developed a simulation environment for electrical energy storage systems, called Toolbox Speichersysteme (Energy Storage Toolbox), that supports a particularly broad range of physical battery characteristics. Toolbox Speichersysteme is designed for realistically and conveniently simulating the thermal and electrical behavior of different batteries, supercapacitors and other electrical energy storage systems, depending on their design and cooling system. The simulation environment provides many different settings that include aspects such as battery technology, the geometric shape of a battery, the number and arrangement of storage cells or the peripheral cooling system. Developers can therefore analyze the cooling strategy and identify potential hot spots that would occur during battery operation. The development, which is sponsored by the European Union and the German state of North Rhine-Westphalia is integrated into the dSPACE tool chain and can therefore be seamlessly implemented in the model-based development process. ■

dSPACE Supports ISO CAN FD

As of dSPACE Release 2015-B, dSPACE supports not only ‘non-ISO CAN FD’ but also its revised version, ‘ISO CAN FD’. The CAN FD (flexible data rate) bus protocol provides a much higher data transfer rate and longer payload data lengths for dSPACE rapid control prototyping and hardware-in-the-loop systems than the classic CAN. The dSPACE DS4342 CAN FD Interface Module supports both protocol versions, ISO CAN FD and non-ISO CAN FD, in addition to the classic CAN, so users do not need new hardware for the transition. Existing systems can be updated conveniently via software. Regardless of the use case, the RTI CAN MultiMessage Blockset will always be used as the implementation software, so no additional learning costs arise. ■



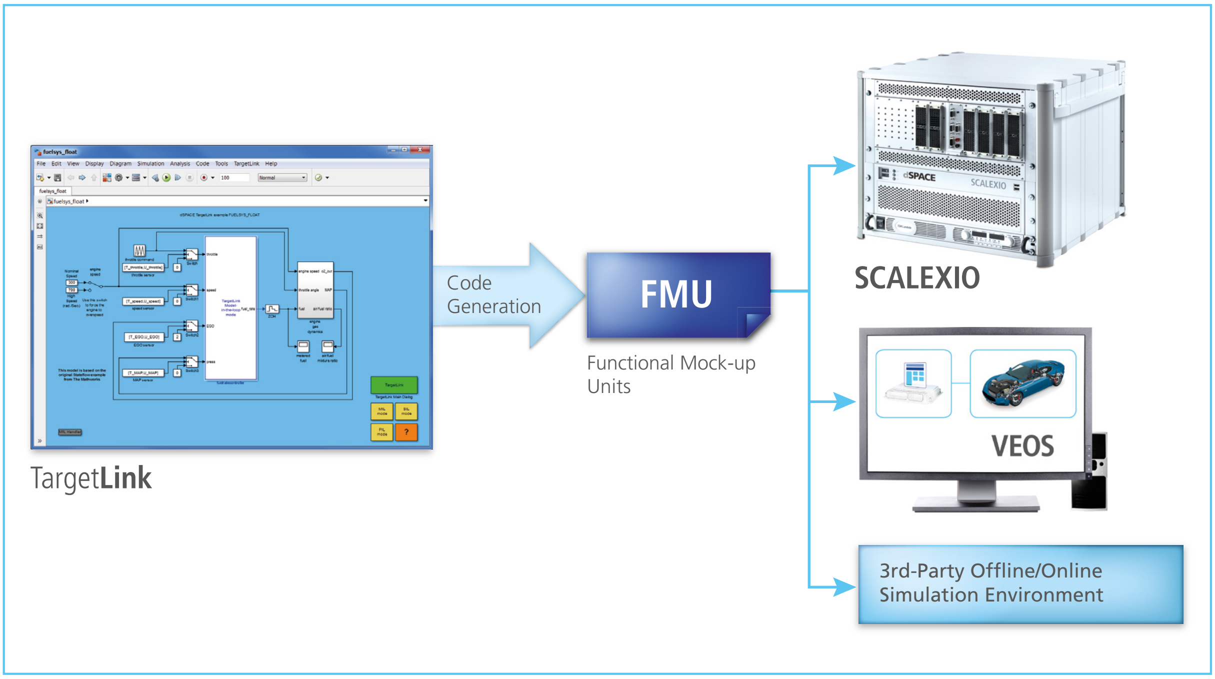
TargetLink 4.1: AUTOSAR Extensions, FMI Support and Much More

The new version of dSPACE's production code generator, TargetLink 4.1, provides substantial extensions for AUTOSAR-compliant development. In addition to supporting AUTOSAR version 4.2, TargetLink also offers functionality for efficient read and write access to NVRAM (nonvolatile RAM) via NvData interfaces. TargetLink 4.1 also supports AUTOSAR transformers, which enable end-to-end communication protection for safety-relevant applications and communication via automotive Ethernet by using SOME/IP (scalable service-oriented middleware over IP). Other new features of TargetLink 4.1 involve modeling in Simulink®/Stateflow®. In particular, the production code generator now supports simplified initialization mode in Simulink with its clear initialization semantics, simplified modeling with buses via the Bus Assignment block, buses at the Simulink/Stateflow interface, structures in the Stateflow Action Language, using Signal Conversion blocks, etc.

The new version also provides crucial additions for Code Generator functionality. MISRA-C:2004/MISRA-C:2012 compliance and code efficiency have been improved, and TargetLink's powerful function reuse mechanism for reusing functions in different projects can now be used for incremental code generation, without any loss in performance. You can use this functionality for parameterized, referenced models or for reusable subsystems that were generated incrementally.

With TargetLink 4.1, you can take Functional Mock-up Units (FMUs) based on the Functional Mock-up Interface (FMI) standard 2.0 and export them from TargetLink models. The standard serves to integrate and simulate models from different modeling environments. By using the FMI 2.0 for Co-Simulation standard, you can execute TargetLink-generated code with dSPACE VEOS®, dSPACE SCALEXIO®, and offline and real-time simulators from third parties.

For information on FMI/FMU, visit *www.dspace.com/go/dMag_20153_fmi*



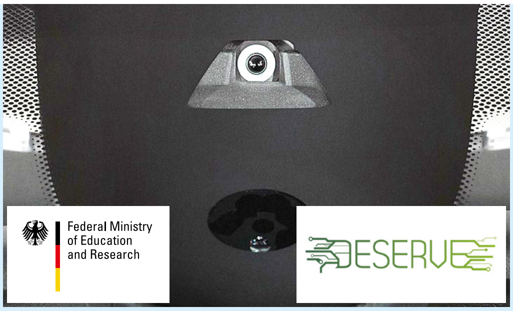
Exporting Functional Mock-up Units from TargetLink to simulation environments that support FMI.

dSPACE on Board

Discover intriguing and innovative applications, achieved with dSPACE development tools

Grasping the Environment

As part of the DESERVE project (Development Platform for Safe and Efficient Drive), a development platform for driver assistance systems was designed that also supports the processing of camera data. dSPACE provides a MicroAutoBox® platform for prototyping the ADAS algorithms, and has extended that platform with a powerful embedded PC and a fast Kintex®-7 FPGA board to handle algorithms for image processing and data fusion.



Camera-based assistance systems observe the vehicle's close surroundings and detect objects that are relevant to driving the vehicle.

Algorithms for image processing and data fusion are developed with the help of a special prototyping platform from dSPACE.

Autonomous on Freeways

A technology study by BMW demonstrates highly automated driving in real traffic on highways. The demonstration vehicle is equipped with sensors such as radar, camera, laser scanner and ultrasonic sensors. dSPACE software and hardware help controlling the vehicle that has to adhere to traffic laws and master new traffic situations.

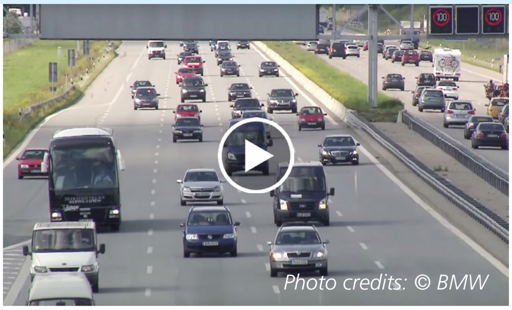


Photo credits: © BMW

Highly automated driving means successfully mastering various traffic situations.
www.dspace.com/go/dMag_20153_BMW

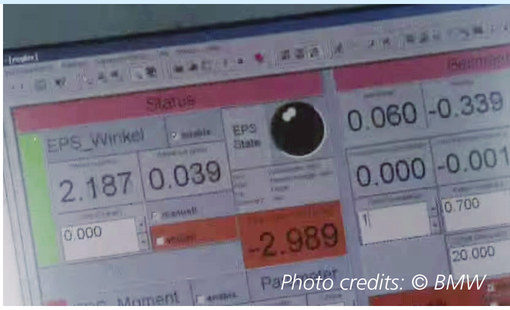


Photo credits: © BMW

ControlDesk is part of the autonomous driving equipment installed into the prototype vehicle.

Autonomous at 160 km/h

AUDI AG uses an Audi RS7 to investigate all aspects of autonomous driving. On the test track, the vehicle already masters a many different driving maneuvers at a high velocity, such as emergency braking and dodging obstacles. A dSPACE MicroAutoBox plays a central role in computing the driving commands.

Photo credits: © ARD-aktuell

The Audi RS7 cornering at high speed on the test track.
www.dspace.com/go/dMag_20153_ARD

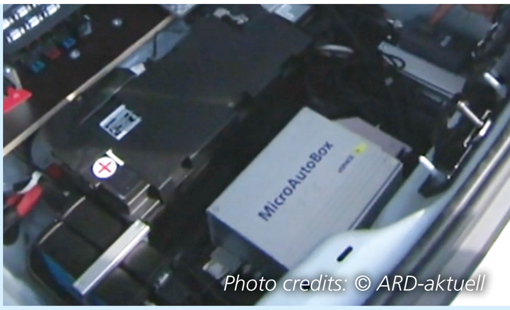
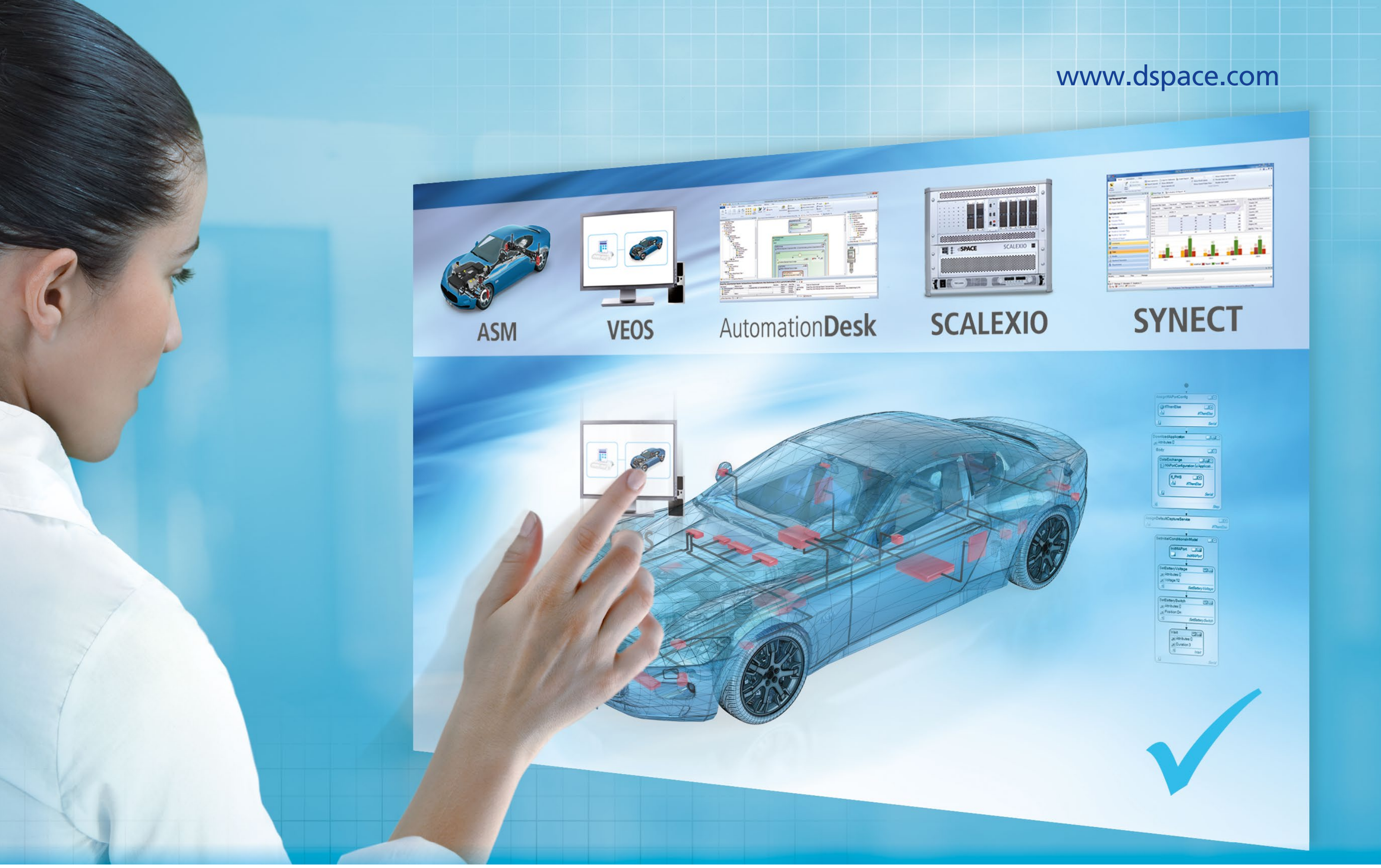


Photo credits: © ARD-aktuell

A dSPACE MicroAutoBox is a central component in controlling the vehicle.

Learn more about these applications online, via videos, photos, and reports:
www.dspace.com/go/dMag_20153_REF_E



The Universe of Testing – with dSPACE

ASM
SCALEXIO
AutomationDesk
SYNECT
VEOS

Advanced driver assistance systems, self-driving vehicles, Car2Car communication – these ever more complex systems require millions of lines of code, while innovation and test cycles are becoming shorter.

How do you validate new functionalities today? And in the future?

With the dSPACE test tools for an early validation:
On the PC without a hardware prototype, with small to large hardware-in-the-loop test benches, open simulation models, a graphical test description, and centralized test data management.

For advanced driver assistance systems, drive, body or other applications – dSPACE has the solution!

For more product information, visit
www.dspace.com/go/testing

Embedded Success **dSPACE**