www.dspace.com

dSPACE MAGAZINE

2/2015

Porsche – Virtual Driving

Customer Feedback

Piloted Driving on the Race Track

The Audi RS 7 piloted driving concept is an innovative high-technology platform that demonstrates autonomous driving even on the race track – at racing speed.

> "A MicroAutoBox performs the signal processing to control the actuators of the RS 7 piloted driving concept – in real time."

Peter Bergmiller, Development of Driver Assistance Systems, AUDI AG

*This **video** shows piloted driving at the extremes with the Audi RS 7 piloted driving concept on the Hockenheimring race track. (www.dspace.com/go/dMag_20152_RS7).*

Photo credits: © AUDI AG

Despite its 'spacy' name, dSPACE is heavily automotive because of the enormous demand from the automotive industry and because we strive to hold the pole positions that we have reached in many areas. But we are always just as enthusiastic about applications from other areas. Aerospace is traditionally one of the larger segments that we back by providing special interfaces and bus support, to name a few examples.

In this edition, a total of three articles are devoted to aerospace applications. No, the driving dynamics application at Porsche is not one of them. A Porsche is not meant to take off in the air at all.

The floating platform from Airbus Defence and Space, however, is meant to stably lift off and land afterwards. And it does, with rapid control prototyping in a rapid 4 years of development instead of 15 years. When you watch the video, you really gain respect for this jet-propelled rocket ride, controlled by models in on-board MicroAutoBoxes. It's a wonderful proof of confidence in our hardware and software - but we're not thinking of renaming them as MicroSpaceBoxes.

Developing the HondaJet took much longer. dSPACE came on board in 2008 by helping plan a simulator for system integration tests, which was already in productive use a short time later. In 2009, Honda Aircraft International publically reported on the simulator's contribution to the successful first flight. This year, the aircraft is now ready for sale. I was able to visit the testing facility myself a few years ago. Back then, the avionics and their tests were already stable, but there were a number of challenging issues because of the exceptionally high number of signals and the extremely wide-ranged automated tests. I congratulate the HondaJet team on their success. Staying power pays off. dSPACE knows this as well.

Dr. Herbert Hanselmann

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 Lillwitz, Thomas Pöhlmann, Julia Reinbach, Dr. Gerhard Reiß

Co-Workers on this issue:
Jace Allen, Dr. Ulrich Eisemann, Doreen Krob, Dr. Klaus Lamberg, Thomas Sander

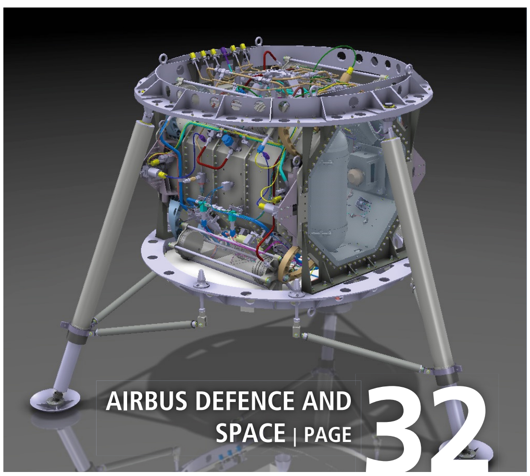
Editors and Translators:
Robert Bevington, Stefanie Bock, Anna-Lena Huthmacher, Dr. Michelle Kloppenburg

Design and Layout:
Jens Rackow, Sabine Stephan

Print: Media-Print Group GmbH, Paderborn

Cover photo © Porsche

Contents



PEFC certified
The paper for this magazine is from sustainably managed forests and controlled sources.
www.pefc.org
PEFC/04-31-0810

ClimatePartner
climate neutral
Print | ID: 53446-1505-1001

Honda Aircraft Company develops a fully automated Advanced Systems Integration Test Facility to get its new business jet into the air in record time.

First Flight in Real Time

Honda Aircraft Company in 2008 laid the foundation for a world-class simulation and test facility for its HondaJet at the company's R&D facility in Greensboro, North Carolina. This Advanced Systems Integration Test Facility (ASITF) is Honda's core facility for developing, validating and verifying engineering systems. After a rigorous evaluation of different hardware-in-the-loop (HIL) vendors on the market, Honda Aircraft selected dSPACE HIL systems to provide real-time simulation capability. In the following months, dSPACE developed a new HIL interconnection architecture that lets Honda Aircraft fully test its interfaces for avionics and full authority digital engine control (FADEC), either through closed-loop simulation or point-to-point breakout and analysis. The HondaJet team accomplished the first successful flight of the HondaJet ASITF on October 9, 2009, approximately 14 months before the actual maiden flight of the first FAA-conforming HondaJet – a critical milestone for the program and a significant step toward the first flight of a conforming aircraft.

Research Phase

Honda began its research in aviation in the late 1980s, so its engineers now have more than 20 years of experience in researching and developing advanced aerospace technologies. The initial design for the HondaJet was sketched in 1997 by Michimasa Fujino, who is now President and CEO of Honda Aircraft Company. Only a year later, research and configuration design for the HondaJet began, giving the jet its characteristic engine mount design, the Over-The-Wing Engine Mount (OTWEM) configuration. >>

The test lab including the iron bird (left) and the three dSPACE HIL simulators (center).

Many more milestones have followed:

- *2003:* First flight of the proof-of-concept HondaJet
- *2005:* World debut of the HondaJet at EAA AirVenture, Oshkosh
- *2010:* Maiden flight of the first FAA-conforming HondaJet
- *2012:* Start of HondaJet production
- *2013:* FAA Type Inspection Authorization for the HondaJet, clearing the way for FAA pilots to begin on-board testing
- *2014:* Initial flight of the first production HondaJet

The HondaJet combines many technological innovations of aviation design. Its characteristic Over-The-Wing Engine Mount configuration, for example, dramatically improves aircraft performance and fuel efficiency by significantly reducing aerodynamic drag. This original airframe design also reduces the noise of aircraft flying overhead and allows for a more spacious cabin and greater cargo capacity. The HondaJet is powered by two highly fuel-efficient GE Honda HF120 turbofan jet engines.

Advanced Systems Integration Test Facility (ASITF)

The HondaJet ASITF consists of two major elements: the aircraft test facility and the Real-time Test and Simulation System (RTSS). Its development was led by Masa Hirvonen, Senior Manager of Systems Integration. At the ASITF, actual aircraft system hardware and software is set up in a spatially representative manner and connected with actual aircraft cable harnesses. The RTSS simulates the avionics systems, the environment and aerodynamics. It is supported by dSPACE real-time hardware and I/O that runs Honda-developed, high-fidelity 6-DOF (degrees of freedom) aerodynamics models and real-time Simulink engine models. Additional simulation capabilities were integrated to provide real-time simulation of navigation RF data, including GPS and VOR/ILS (VHF omnidirectional radio range/instrument landing system) signals. The HondaJet ASITF also includes an electronic control loading system (ECLS) that is integrated with a fully representative primary flight control system (PFCS). The avionics configuration for the HondaJet is based on the integrated Garmin G3000™ avionics suite with two touchscreen controllers.

"Test automation and traceability from test execution all the way back to the system requirements are the keys to efficient, repeatable, and fully traceable test operations for systems development and integration. Being able to implement these key features has given us a significant advantage over many other aircraft programs."

Masa Hirvonen, Senior Manager Systems Integration, Honda Aircraft Company

"The hardware and software tools provided by dSPACE have allowed us to quickly implement the advanced simulation and test capabilities needed for the HondaJet program. The open and expandable architecture lets us develop and expand the test facility to meet our growing needs. Without the exceptional tools, service, and support that dSPACE provides, our small team would not have been able to accomplish the things we have."

Benjamin Hager, Real-time Control Design and Simulation Engineer, Honda Aircraft Company

Other real parts used in the integration system include the flap actuation system (FAS), fuel quantity system (FQS), automatic flight control system (AFCS), nose wheel steering system (NWSS), and electrical power system (EPS). These individual components combine the abilities of an engineering simulator, an integration test facility, and an iron bird in a single facility.

Setting up the Test System

dSPACE Inc. supplied the RTSS, which provides the advanced test interface, test automation and high-fidelity simulation abilities. It essentially comprises three full-size HIL racks: two racks for avionics interfaces (RTSS1 and RTSS2) and one for the aircraft engines (Engine HIL for FADEC interfaces). The RTSS systems use six DS1006 Processor Boards in five expansion boxes and have the following I/O and communication support:

- 820 channels of discrete I/O (inputs and outputs)
- 180 channels of ARINC429 (90 TX, 90 RX)
- 240 channels of analog I/O (inputs and outputs)
- Multiple serial data bus interfaces
- Ethernet interfaces (UDP and TCP/IP)

The RTSS uses a special signal breakout interface system that enables system testing and simulation in four operation modes:

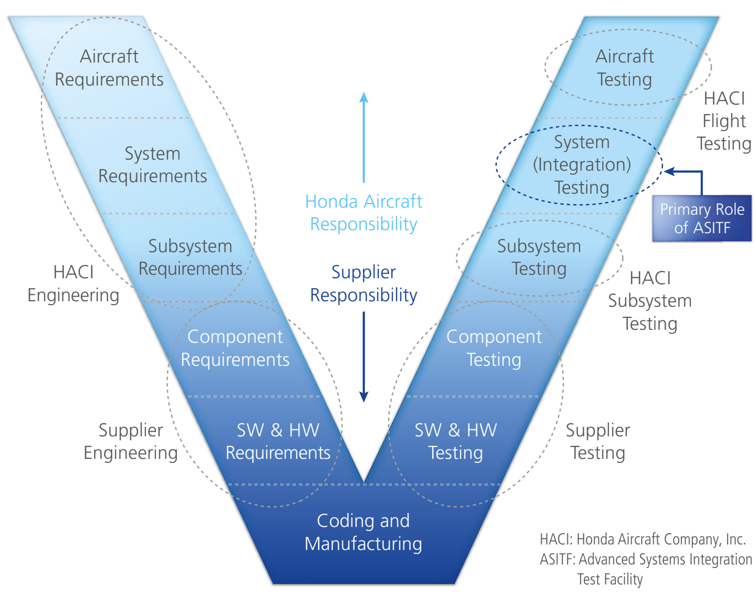
1. Manual testing (traditional system break-out)
2. Computer-assisted testing (using ControlDesk®)
3. Automated testing (using AutomationDesk)
4. Pilot-in-the-loop testing (including iron-bird capability)

The RTSS uses a harness interconnect and break-out box (BoB) system with which the signals can be either sent straight through, broken out for analysis, bypassed for simulation, or connected directly for iron-bird testing.

System Simulation Control and Interfaces

The core of the RTSS system is a high-fidelity 6-DOF aircraft simulation model that is integrated with engine simulation models and various aircraft system test interfaces. The model is spread across the multiprocessor system to allow for full-bandwidth model control and parallelization of ARINC message >>

Benjamin Hager, Real-time Control Design and Simulation Engineer at Honda Aircraft Company, operates the test system using dSPACE ControlDesk equipped with avionics instruments (Primary Flight Display) and dSPACE AutomationDesk.



The development process for the HondaJet program follows the V-cycle. The dSPACE simulators are mainly used during the system integration testing phase.

and I/O throughput. The closed-loop simulation executed in the RTSS can be used to fly the ASITF 'aircraft' in real time. The Real-time Engine Model (RTEM) simulation connects to real FADEC hardware so the HF-120 turbofan engine can be simulated closed-loop. Real-time GPS and VOR/ILS simulations (RF signals) provide navigation data during the piloted missions. The pilot-in-the-loop mode uses the interface and animation abilities of the visual system to provide direct flight simulation interaction. This out-of-the-window visual system is key for pilot-in-the-loop testing. Honda Aircraft engineers worked with the ControlDesk-programmable user interfaces to create quick access interfaces for the most important aircraft subsystems, making testing easier. The engineers used the related API to build extensions for ControlDesk that allow for calibration trim, provide interfaces for the real-time model, and make test setups easy to prepare. dSPACE test automation blocks from the ApplTools Solution are used to help manage the complexity of I/O interfacing and usage. By automating the layouting and model creation from ARINC labels, using the myriad of messages in the system became much simpler. Signal definitions are now taken from the models, and Python scripts use XML to auto-map the signals to HondaJet custom instrument configurations. Interface control definitions (ICDs) are used to manage all system interface signals, which are used to work with signals within the models and RTSS system. All software used with the RTSS is managed and controlled in IBM® Rational® ClearCase®, and IBM Rational ClearQuest® is used for internal issue and change tracking.

Development According to V-Cycle

Honda uses the established V-cycle for systems development and defines the responsibilities of customer and supplier via subsystems

> "Our Advanced Systems Integration Test Facility is a powerful tool that continues to support the development and certification program for the HondaJet. The ASITF lets us evaluate the entire system integration, which increases aircraft safety and ultimately helps us develop an advanced light jet that will exceed our customers' expectations and bring them lasting joy."
>
> *Michimasa Fujino, President and CEO of Honda Aircraft Company.*

and systems development methodologies. The core responsibility of the ASITF is system integration and validation, as shown in the following diagram. The ASITF mission is defined by the following tasks and responsibilities:

- Hardware-in-the-loop (HIL) test facility, including iron-bird capability
- Integration tests of aircraft subsystems
- Functional tests of aircraft systems
- Pilot-in-the-loop (PIL) tests
- Flight test, production and fleet support

Automated Test Runs

Automated test scripts written with dSPACE AutomationDesk let the HondaJet ASITF team perform 'lights-out testing'. This means that the team can test flight control and avionics systems around the clock, at night and on weekends, with minimal supervision. "This ability lets the Honda Aircraft team conduct more thorough and in-depth testing at the ASITF, while keeping track of requirements traceability," says Jace Allen, Lead Technical Specialist at dSPACE, Inc. Honda Aircraft uses IBM Rational DOORS® to write requirement specifications and test plans. Specific requirements are linked to test cases within DOORS, and dSPACE Connect&Sync links HondaJet test cases to AutomationDesk tests and projects. The test results can reimported into the DOORS document after test cases are run in AutomationDesk. This integration gives Honda Aircraft traceability in their test process, allowing them to directly connect requirements to current test results. Honda Aircraft is considering to expand this traceability to help extend the testing capability with an integrated test management using dSPACE SYNECT® Test Management. ■

By kind permission of Honda Aircraft Company, Inc.

Watch the first production HondaJet take to the sky: www.dspace.com/go/dMag_20152_HJET

Summary of the HondaJet Project

Honda Aircraft Company set up a world-class simulation and test facility for the HondaJet. This Advanced Systems Integration Test Facility (ASITF) is the core facility for engineering systems development, validation, and verification. By using tools like dSPACE simulators and the test automation software dSPACE AutomationDesk, Honda Aircraft is able to test all of its flight control and avionics systems. The engineers at Honda Aircraft are convinced that hardware and software tools provided by dSPACE were an important contribution to verifying and validating avionics reliably and efficiently. With dSPACE tools, the developers were able to use the following test methods:

- Manual testing
- Computer-assisted testing using dSPACE ControlDesk
- Automated testing using dSPACE AutomationDesk
- Pilot-in-the-loop testing (including iron-bird capability)

HIL pilot project for exchanging submodels based on FMI

Always the Right Model

Realistic hardware-in-the-loop tests require real-time-capable models that can simulate the required system behavior with the right degree of accuracy. Together, BMW, ITI, and dSPACE test the transfer and use of a real-time-capable model part via an open, tool-independent interface.

For a joint project, BMW, dSPACE, and ITI used a model of an automatic transmission to analyze whether the Functional Mock-up Interface (FMI) standard (2.0) is suitable for use in real-time hardware-in-the-loop simulation. The companies also tested a process prototype for replacing submodels in an existing simulation model with FMI-based submodels.

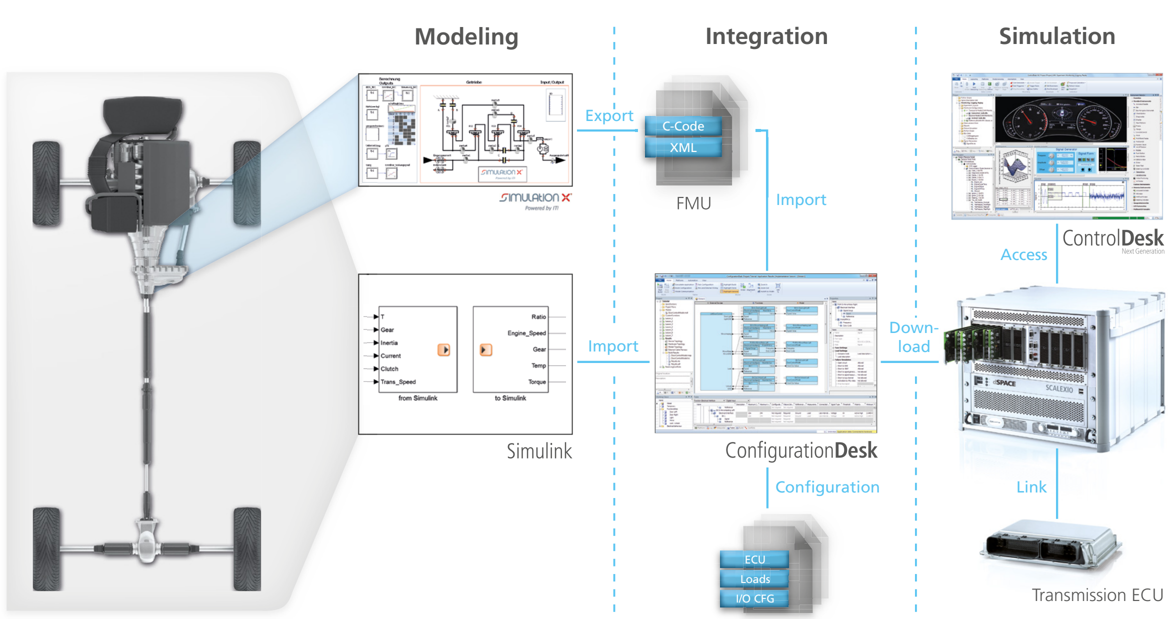
Test System Setup

To make the simulation realistic enough and prevent entries in the fault memory of the ECU when it is being tested with a HIL simulator, two things are needed: the correct interaction of the ECU interface with the HIL simulator and realistic simulation models. This is needed to test the behavior of an ECU realistically. Because the tests in the project were to be executed for realistic use cases, the project uses the electrical interface and environment simulation models of an existing HIL setup for the ECU of an 8-gear automatic transmission, adjusted for use in dSPACE's HIL simulator SCALEXIO®. In addition to MATLAB®/Simulink® and also SimulationX® from ITI for modeling, the project uses dSPACE's ConfigurationDesk® software for configuring the SCALEXIO system and dSPACE ControlDesk® Next Generation for controlling the HIL simulation (figure 1).

The Idea Behind FMI

The basic idea of FMI is to avoid redundant work: Its goal is to reuse existing simulation models, which can come from different suppliers, in different development phases and in different departments of a company. This requires a standard that makes it easier to transfer and integrate environment models that were developed with development tools from different providers. Functional Mock-up Interface (FMI) is such an open, tool-independent standard. It lets engineers use the perfect modeling approach for each submodel and easily combine these submodels in a project. The submodels are transferred via Functional Mock-up Units (FMUs). FMUs are compressed folder structures containing the model functionality and the required interface description as ANSI-C-compatible API and XML file. Users can also add documentation and other data that the model requires. The new functions of the FMI 2.0 standard are beneficial for HIL projects, such as the possibility to adjust parameter values during simulation run time and the direct definition of step sizes for real-time-optimized solvers. Therefore, FMI 2.0 for Co-Simulation is used in this project. In the FMI for Co-Simulation

Figure 1: Setup of the overall system. The Simulink model and the FMU from ITI are integrated to one overall model in ConfigurationDesk. The overall model is then configured and loaded to SCALEXIO for HIL simulation. The simulation is controlled via ControlDesk Next Generation.



variant, the suitable real-time-optimized solver is already included in the FMU.

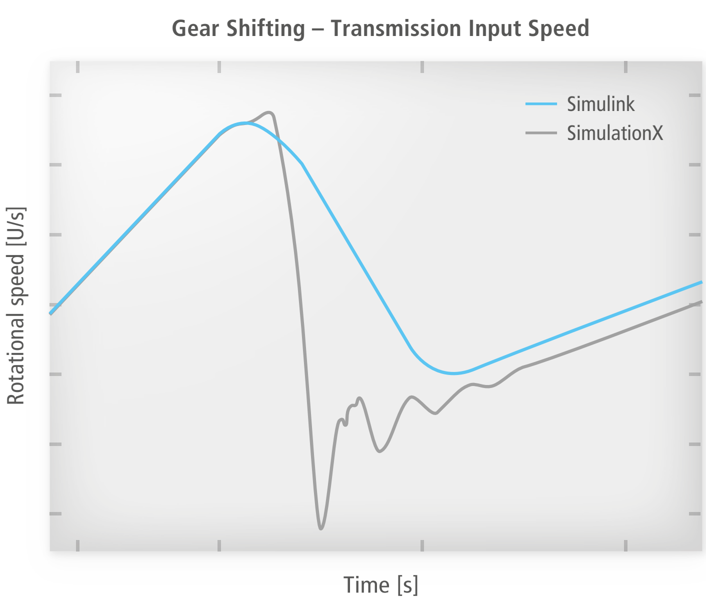
Benefits of the New Model

The project aims at using the described advantages of FMI to integrate a new, Modelica-based, acausal, dynamic transmission model in the Simulink-based overall drivetrain model. The goal is to better represent the elastic behavior of the transmission when simulating the overall behavior. To this end, ITI modeled four sets of planetary gears with variable gear ratios in SimulationX, including the associated inertias and elasticities, and the input, output and friction torques. The switchable clutches are modeled as friction surfaces with the appropriate physical friction behavior, including the alternation between the locked and unlocked states. The clutches are controlled by a gear shift logic with a speed- and torque-dependent gear shift diagram. Figure 2 highlights how different the simulation precision is for the dynamic transmission behavior when it is simulated with the simple Simulink model or with the SimulationX model. The environment model of the drivetrain is computed in real time with simulation steps of 1 ms. The new transmission submodel that is modeled in SimulationX and exported as an FMU has to meet these real-time requirements.

Integrating the New Model

At the start of the project, the model interface for integrating the transmission model was defined and then used by all project partners throughout the project (figure 3). To integrate the new transmission model into the existing, Simulink-based drivetrain environment model, the developers performed the following steps: >>

Figure 2: The blue curve (Simulink) indicates the idealized rev interval during a gear shift in the existing, signal-flow-oriented transmission model. The gray curve (SimulationX) indicates the transmission behavior in acausal modeling, which is more realistic because it considers vibrations.

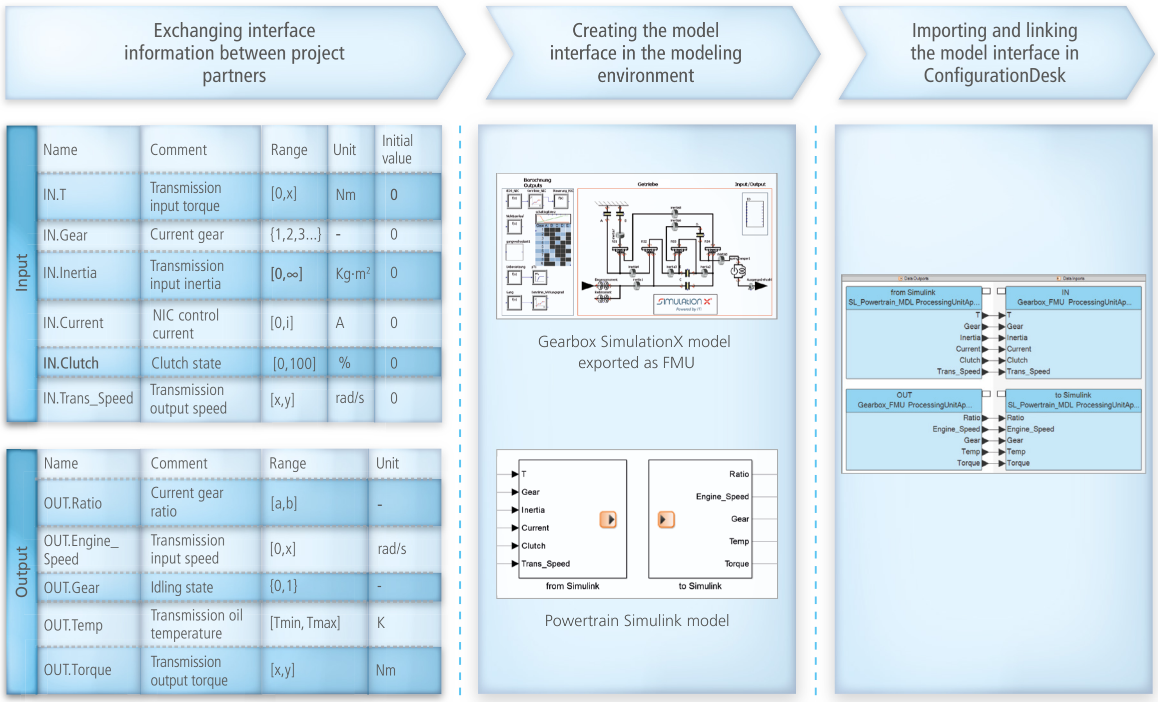


About FMI

The FMI standard was first defined in the MODELISAR project in 2011 and is currently being developed further by the FMI project of the Modelica Association. The standard focuses on the exchange of models of dynamic systems: i.e., models that are defined by differential, algebraic, and discrete equations.
Its current version, 2.0., includes new functions that benefit HIL simulation. One major advantage is the possibility to define the step size for real-time-optimized solvers and tunable parameters that allow changing the parameter values during simulation run time. This is necessary for interactive experiments and HIL simulations that cannot simply be restarted because the control loop includes real hardware.
www.dspace.com/go/fmi

About ProSTEP

The ProSTEP iViP Association is an international association that is committed towards developing innovative approaches to solving problems and modern standards for product data management and virtual product creation.
www.prostep.org



Name	Comment	Range	Unit	Initial value
IN.T	Transmission input torque	[0,x]	Nm	0
IN.Gear	Current gear	{1,2,3...}	-	0
IN.Inertia	Transmission input inertia	[0,∞]	Kg·m²	0
IN.Current	NIC control current	[0,i]	A	0
IN.Clutch	Clutch state	[0,100]	%	0
IN.Trans_Speed	Transmission output speed	[x,y]	rad/s	0

Name	Comment	Range	Unit
OUT.Ratio	Current gear ratio	[a,b]	-
OUT.Engine_Speed	Transmission input speed	[0,x]	rad/s
OUT.Gear	Idling state	{0,1}	-
OUT.Temp	Transmission oil temperature	[Tmin, Tmax]	K
OUT.Torque	Transmission output torque	[x,y]	Nm

Figure 3: The predefined modeling interface is considered and implemented in each work step.

1. Cropping the existing, simple transmission from the Simulink model
2. Adding the necessary model interfaces to the Simulink model via dSPACE Model Port blocks
3. Forwarding the data required for creating the new transmission model to ITI. This data included:
   a. An interface description file
   b. The required model functionality
   c. The technological framework (C compiler, real-time requirements, etc.)
4. Based on this data, ITI developed a physical SimulationX model and provided an FMU and a Simulink-based S-function as a test benchmark.
5. First, the S-function was included in the model to test whether the SimulationX transmission model functions correctly (variant 1). Then, the FMU was integrated to test the interface's function (variant 2).

Tests and Results

The project partners used ControlDesk Next Generation to perform closed-loop tests on the dSPACE HIL simulator SCALEXIO for the two variants: i.e., for the integration via Simulink using an S-function (variant 1) and for the integration via an FMU, with ConfigurationDesk (variant 2). The computation times for the model under test were nearly identical in both cases and the test results were bit-identical. It was possible to compute the new transmission model variant, which considers vibration phenomena of up to 40 Hz, with numeric stability in real time at a step width of 1 ms. FMI 2.0 for Co-Simulation therefore shows that it is fit for the standardized, tool-independent transfer of real-time-capable models. This means that the FMI standard can help simplify exchanging environment models between departments.

Tool Chain Integration

But exchanging model parts is only a first step; it is also important to test how the exchanged element can be used in an existing tool chain. To make the adaptation process as fast and resource-saving as possible, major adjustments to the configuration and experiment tools must be avoided. Since the configuration software ConfigurationDesk supports both Simulink and FMI as import formats for environment models, exchanging the model part via the established model interface was quick and easy. ConfigurationDesk recognized the new model interface, making it possible to use the interface for forwarding the model signals. In ControlDesk Next Generation, the model parameters and variables of the FMU are available in the same way as in Simulink-based models. This made it easy to adjust and reuse the existing tests and the experiment layout for the transmission HIL system. Once the new, FMI-based transmission model was integrated in the overall project, the existing workflow and the associated HIL tests can be reused within the project without significant modifications.

Outlook

ProSTEP used the experience gained in this pilot project in their Smart Systems Engineering Project for the developed workflow describing the collaboration of various partners during FMI-based model exchange. It is planned to continue the project described in this article in order to analyze the workflow for exchanging intellectual-property-protected FMUs for HIL testing. ■

By kind permission of BMW AG

Summary

BMW, dSPACE, and ITI used a model of an automatic transmission in a pilot project to analyze whether the Functional Mock-up Interface (FMI) standard is suitable for use in real-time hardware-in-the-loop simulation. It was possible to compute the new FMI-based variant of the transmission model, modeled in SimulationX, with numeric stability in real time with a step size of 1 ms. FMI 2.0 for Co-Simulation therefore appears to be suitable for exchanging real-time-capable models. The easy integration of FMI-based models into the dSPACE tool chain made it possible to reuse existing tests and experiment layouts with little additional effort. Because the ECU of the automatic transmission runs on the SCALEXIO HIL system without entries in the fault memory, detailed tests that consider the dynamic effects of the transmission can now be performed in real time.

In the future, the FMI standard can be used to combine only the most suitable model elements, regardless of who manufactured them.

Dynamic Models

Vehicle dynamics are the showpiece of car manufacturing. Porsche uses an efficient, seamless vehicle development process to pass on excellent vehicle dynamics genes from the first development steps to the final product.


918 Spyder

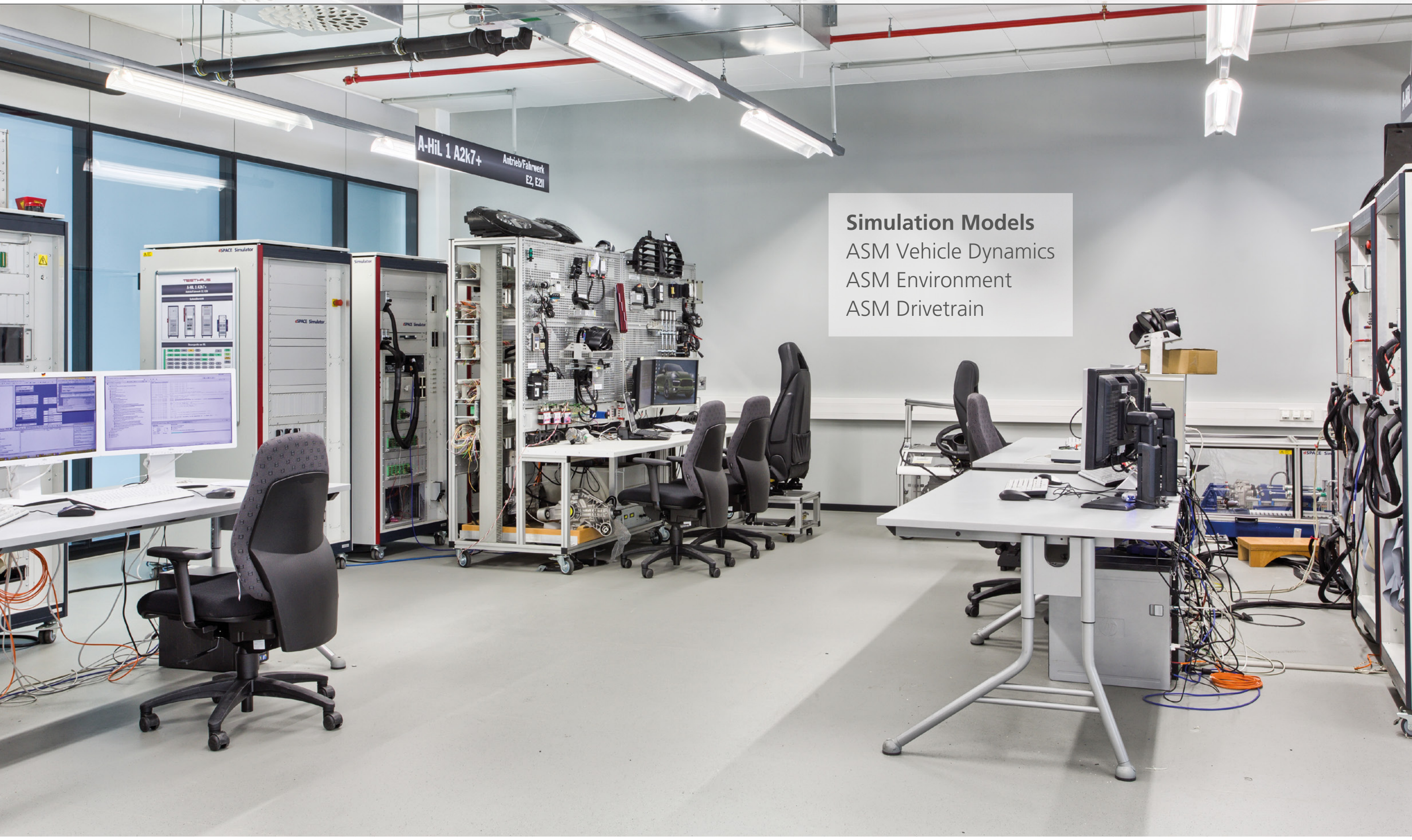


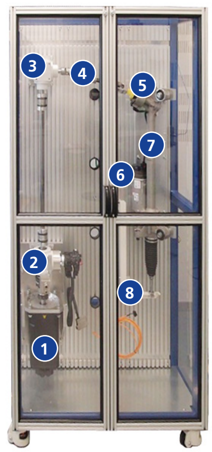
Figure 1: HIL test benches for validating the drive/chassis applications. A setup for installing ECUs and for integrating real components (throttle valves, injectors, transmission valves, actuators for the electronic parking brake, radiator shutters, etc.) is located at the center back.

Porsche vehicles are known for their outstanding vehicle dynamics. To achieve this high level of quality, all vehicle components must be very finely tuned to each other, especially the body, chassis and the wheels. Active chassis components are playing an increasingly important role in achieving Porsche's ambitious vehicle dynamics goals. These active components include the electronic stability program (ESP), active damping control and the Porsche Dynamic Chassis Control (PDCC), which reduces a vehicle's roll in curves to almost zero, improving agility and comfort. Because the number of vehicle variants increases steadily while development cycles are becoming shorter, the virtual simulation of the vehicle is gaining importance for the development process.

Function and ECU Tests with HIL Test Benches

At Porsche, hardware-in-the-loop (HIL) simulation has earned its place for automated testing of the electronic control units (ECUs) during the development process. It has become a core component in the creation of new products. Networked HIL test benches are used for the integration and function validation of applications in drives and chassis as well as body and infotainment. These test benches include all ECUs of the relevant application (figure 1). While relatively simple simulation models are sufficient for body and infotainment applications, the modeling effort for drive and chassis applications is much higher, because they require more complex models, such as for combustion engines and transmissions. These models simulate all system variables (such as sensor signals or bus signals), which can then be used for ECU testing via the HIL simulator. However, systems whose ECUs cannot be separated from the actuators simply cannot be simulated in this way (e.g., electric steering systems).

> "For the HIL validation of vehicle dynamics ECUs, we rely on dSPACE's ASM Vehicle Dynamics."
>
> *Dr. Günter Hetzel, Porsche AG*



1. Motor for superimposing the driver steering torque
2. Angle gear for steering angle sensor (only on the test rig)
3. Angle gear (only on the test rig)
4. Steering column
5. Steering box
6. Servomotor for steering support
7. Rack and pinion
8. Measuring device for analyzing the specified path

Figure 2: Setup for testing an electric steering system.

In these cases, the entire mechatronics system is integrated into the setup, including the ECU, the electric machine, and the steering column. A servomotor takes on the role of the driver and generates the steering torque (figure 2). The high complexity of the chassis ECU functions place high demands on the vehicle models used with the HIL test bench. Testing and validating the vehicle dynamics functions of the ECU network in particular requires validated models that emulate the real vehicle behavior as precisely as possible. The simulation models used for the drive/chassis test benches consist of a combination of Porsche's own models (e.g., for the transmission) and dSPACE's Automotive Simulation Models (ASM). The vehicle dynamics are simulated with ASM Vehicle Dynamics.

Simulating Vehicle Dynamics in the Development Process

Simulation models can support the objective evaluation of vehicle dynamics even in early project phases. They allow for a systematic, automated and therefore efficient and comprehensive validation of all relevant criteria. Depending on the task, different model classes are used for chassis/vehicle dynamics testing at Porsche AG's development center in Weissach:

- Reference models with a high-fidelity simulation of the chassis and vehicle components, but with increased computation times
- Function models with a simplified representation of the parts, which can be simulated in real time or faster
- Property models with a summarizing description of the components and a complexity comparable to that of a function model, but with very short calculation times
- Component models without a model of the overall vehicle to observe individual components, as in a test bench model

To use these model classes equally for the associated tasks and without any additional effort, Porsche introduced a process of seamless parameterization and uniform validation (figure 3). This approach makes it possible to create the models from the respective superordinate classes and validate them based on the same driving maneuvers. Because only open-loop driving maneuvers are possible, such as specifying steering and speed profiles, objectively capturing the vehicle properties becomes easy in both the real test drives and the simulation. At the same time, the stationary and the dynamic behavior of the vehicle are considered.

Combining Strengths on the HIL Test Bench

To benefit from the uniform model preparation process while being able to use the mature test automation >>

Type	Real-Time Capability	Parameterization Effort	Component Models
Reference model	No	High	Exact simulation
Function model	Yes	Partly automated parameterization from the reference model	Simplified
Property model	Yes	Partly automated parameterization from the reference model, function model or measurements	Simplified, component summary
Component model	Model-dependent	Model-dependent	Model-dependent

Table 1: Classes of the vehicle dynamics models.

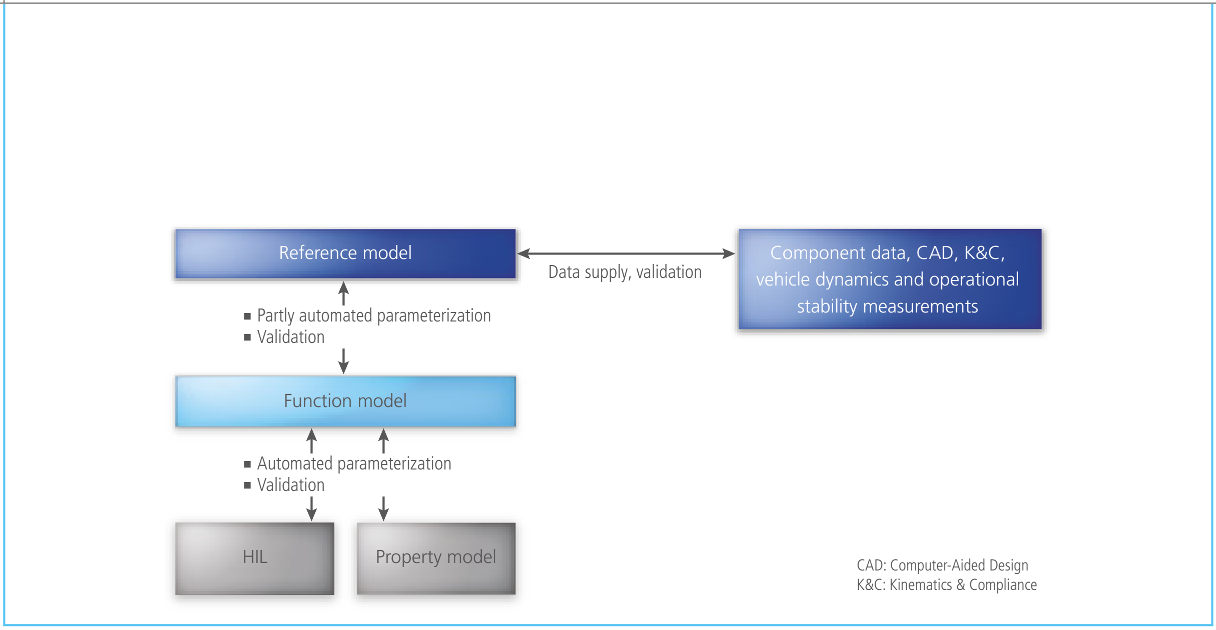


Figure 3: Seamless model parameterization.

and robust dSPACE ASM environment, Porsche created a function for translating the function model's parameters. This function can now be used to create fully automated model data sets for the HIL test bench that have practically the same vehicle dynamics properties as the reference model. Valid vehicle dynamics models let developers perform comprehensive tests of the ECU code's functional aspects with the desired accuracy.

A Double Win

Due to the process described here, the completely parameterized chassis models that are needed for the HIL test bench are already available during the established model preparation process. ECU development and validation therefore benefits from validated chassis models in early development stages. The simulations carried out in the ECU network can also be used for validating a software-in-the-loop (SIL) environment in which software models of the ECUs can be tested with the same vehicle data sets. Porsche therefore has a powerful and robust tool chain for all phases of vehicle development.

Outlook

Our goal is a seamless range of models for all future vehicle projects. In the next step, we will aim at automating parameter preparation and the validation and comparison of the HIL and SIL simulation results. This will make the work processes for virtual validation even more efficient. ▪

Dr. Günter Hetzel, Florian Strecker,
Dr. Ing. h.c. F. Porsche AG

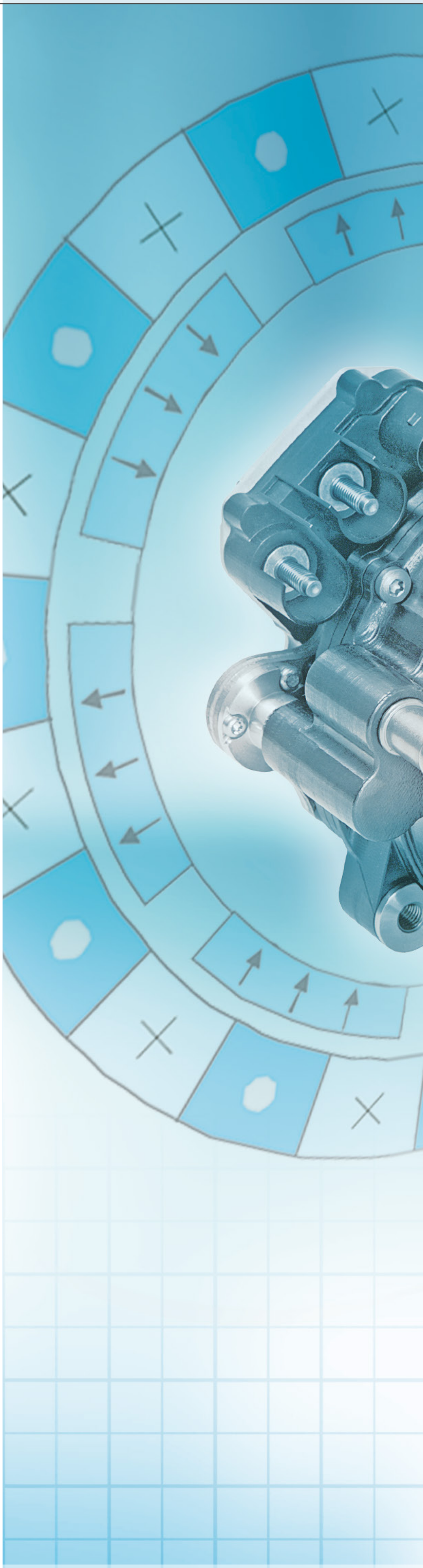
Dr. Günter Hetzel
Dr. Günter Hetzel is an expert for test tools and methods at Dr. Ing. h.c. F. Porsche AG in Weissach, Germany.

Florian Strecker
Florian Strecker is an expert for vehicle dynamics computing and system dynamics at Dr. Ing. h.c. F. Porsche AG in Weissach, Germany.

Seeing is knowing: The animation software MotionDesk is the perfect tool for visualizing simulated driving maneuvers.

ASM Vehicle Dynamics

ASM Vehicle Dynamics is a simulation model for simulating the vehicle dynamics behavior of a vehicle realistically in real time. The physical vehicle characteristics are represented by a multibody system with 26 degrees of freedom. ASM Vehicle Dynamics includes a configurable drivetrain with elastic shafts, a brake circuit and motor model, different wheel models, non-linear kinematics and elastokinematics properties of the chassis, a three-dimensional description of aerodynamics and a complex steering model with several degrees of freedom. An environment with a road, maneuvers, and an open- and closed-loop driver is included as well. All parameters can be altered during run time. The modular components of the additional ASM can be connected to form a virtual vehicle, for example, for testing ECU networks in a HIL environment.



Continental's new belt-driven starter generator uses not only the 48 V vehicle electrical system but also an asynchronous machine as the electric motor and generator – a novelty. dSPACE's flexible HIL test systems spur the development of the mild hybrid ECU.

The ancient Greek word 'synchronos' means 'together with time'. In the field of electric motors and generators, the term 'synchronous machine' is therefore often used to refer to a permanent magnet synchronous machine (PMSM) or a separately excited synchronous machine (SSM) with sliding contacts, where the rotor revolves around the stator with exactly the same speed as the stator's rotating fields, in other words: synchronously.
Asynchronous machines (induction machines) are different. In generators, the rotor runs faster than the stator rotating field; in electric motors, it runs more slowly. While the power density of asynchronous machines is not as high as that of a PMSM, asynchronous machines are more affordable because they do not use an expensive permanent magnet.
In addition, an asynchronous machine does not require a direct connection to the rotor and therefore works without sliding contacts. This simple setup makes them very robust. For this very reason, the industrial sector has been using asynchronous machines for decades, and the technology prevails even in the tough environment of combustion engines, making the system more reliable.

Continental is the first automotive supplier to use an induction motor in a 48 V belt-driven starter generator (BSG) in order to produce a high number of units at low costs while also offering new ways for reducing CO_2 emissions in an affordable mild hybrid.

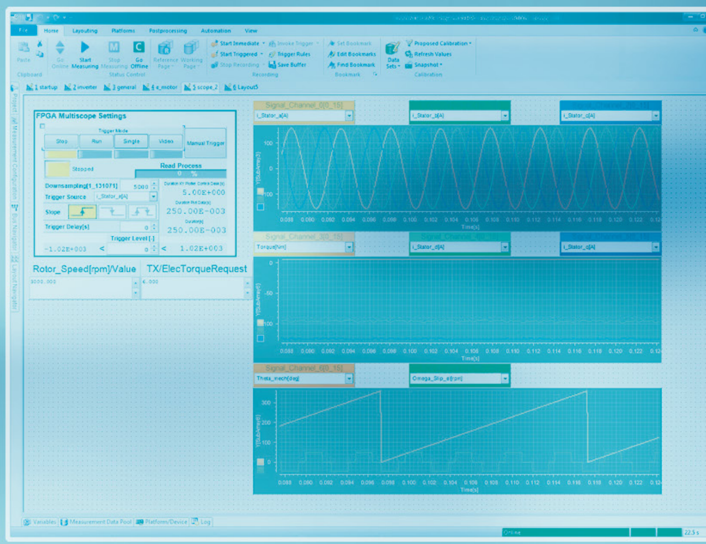
Proven Test Platforms

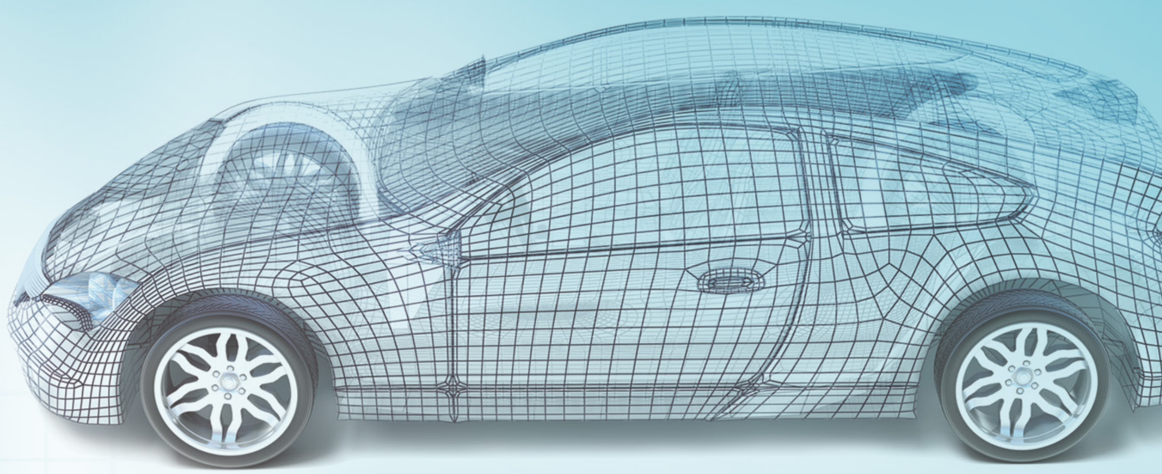
The 48 V vehicle electrical system consists not only of the electric motor with the drive belt and an integrated inverter, but also includes a lithium-ion battery and a 12 V DC/DC converter with the associated control software. The developers focus on designing, implementing, and validating these controls, and also on the software for functional safety. Building on its long-standing experience in developing high-voltage power electronics, Continental uses only tried and tested concepts, platforms and tools for the 48 V BSG. This considerably shortens development time, increases robustness, and reduces development costs. This is where the high flexibility and realism of the dSPACE hardware-in-the-loop (HIL) test systems come into play. Continental has already worked with these systems for many years to validate control functions in the development phase. >>

Synchronously

Asynchronous

New ways for cost-effective mild hybrids





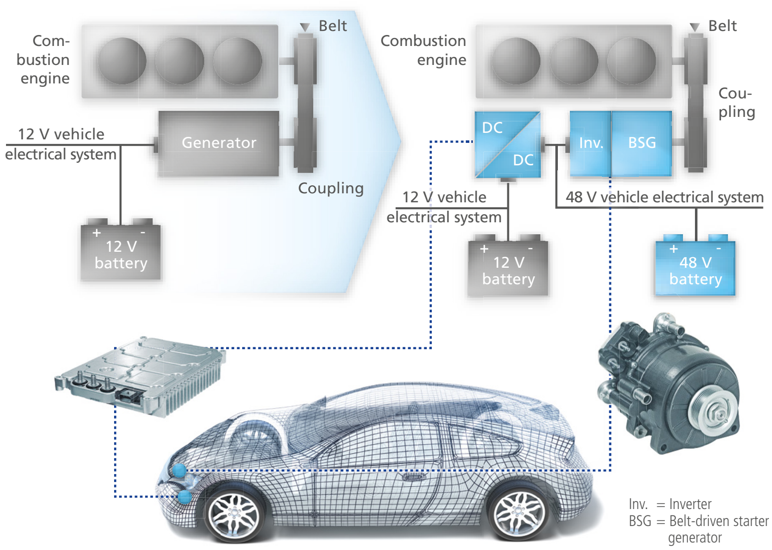


Figure 1: For the mild hybrid components (light blue) in the 48V Eco Drive, Continental included an additional 48 V vehicle electrical system in addition to the conventional 12 V system.

The systems link the pure, PC-based simulation with the much more expensive and time-consuming test rigs that include real electric motors. The developers can use the measured motor parameters and the control algorithms that were developed from MATLAB® models during simulation to further optimize the electric motors.

Early Optimization and Tests

The new functions can be tested and analyzed on the inverter hardware at a very early stage. For this, the HIL simulator emulates the electric motor. This approach lets developers detect potential errors in the interplay of the hardware and functions early on, eliminate them, and begin to optimize the two components even before the control software is integrated into the 48 V BSG and taken to the test bench. Test automation on a HIL simulator also provides a quick and comprehensive way to cover a broad range of tests and validate the software against customer requirements at an early stage. The requirements and tests are specified in IBM® Rational® DOORS®. Continental performs, analyzes, and summarizes the resulting automated regression tests on the HIL simulator by means of automated test coverage metrics before shipping the software to the customer.

No Red Tape for the Long-Standing Partner

For the 48 V BSG, Continental used dSPACE's highly dynamic DS5203 FPGA Board with the Xilinx® System Generator (XSG) library. This library contains quasi-continuous models for the inverter, the mechanics, and the electric motor. In order to correctly simulate the electric motor for the development of the 48 V BSG, an asynchronous machine first had to be integrated into the XSG Electric Components Library. Although the library was planned for a future dSPACE Release at the time, dSPACE granted Continental access to a beta version of the latest XSG library. This made it possible to gain joint practical experience with the new models, which have since been published in a regular Release.

Exact Simulation of the Induction Machine

Within a few days, the first FPGA-based asynchronous machine that can interact with the inverter ECU was commissioned at the HIL laboratory at Continental Regensburg in collaboration with dSPACE. To make the simulation results as realistic as possible, not only current- and temperature-dependent effects were considered, but also frequency-dependent effects. The required FPGA model extensions were also implemented fast and easily with the dSPACE XSG Utils Library, which is also FPGA-based. By using this new model approach, Continental was able to consider many relevant effects and use the 2-D look-up tables that were recorded at the test rig for the HIL simulator as they were. This made the simulation of the real behavior of the machine as exact as possible.

> "Because of our very positive experience during the development of the 48 V belt-driven starter generator, the first follow-up projects at Continental also use FPGA- and machine-model-based HIL simulation."
>
> *Anja Poppe, Continental*

High Flexibility and Simulation Quality

The high flexibility of the dSPACE modeling interface also makes it possible to simulate recorded look-up tables on the FPGA without creating a new FPGA version. The look-up tables can be created in MATLAB®/Simulink® or adjusted during run time in dSPACE ControlDesk® Next Generation. This ensures flexibility for future applications, such as validating a new motor variant. By integrating the multiscope instrument of the XSG Utils Library in ControlDesk, Continental is able to visualize FPGA-internal variables (such as currents, voltages, inductivity, or magnetic saturation) at the FPGA's rate. Continental can therefore optimize control strategies and continuously improve control quality. dSPACE immediately provided optimizations for the asynchronous machines and solutions for problems that Continental discovered during the test phase, spurring the development of the 48 V BSG.

On the Path to Production Quality

Due to the positive experience during the development of the 48 V BSG, the first follow-up projects at Continental already use the tried and tested HIL simulation based on the FPGA and the machine model. The starter generator decreases the fuel consumption of a compact car by up to 20%, partly due to energy recuperation. But the ancient Greek definition holds true not only for recuperation, i.e., in the generator mode of the asynchronous machine. Because just like the rotor that moves faster than the stator rotation field, the entire 48 V belt-driven starter generator will be ahead of its time when the series production starts in 2016. ■

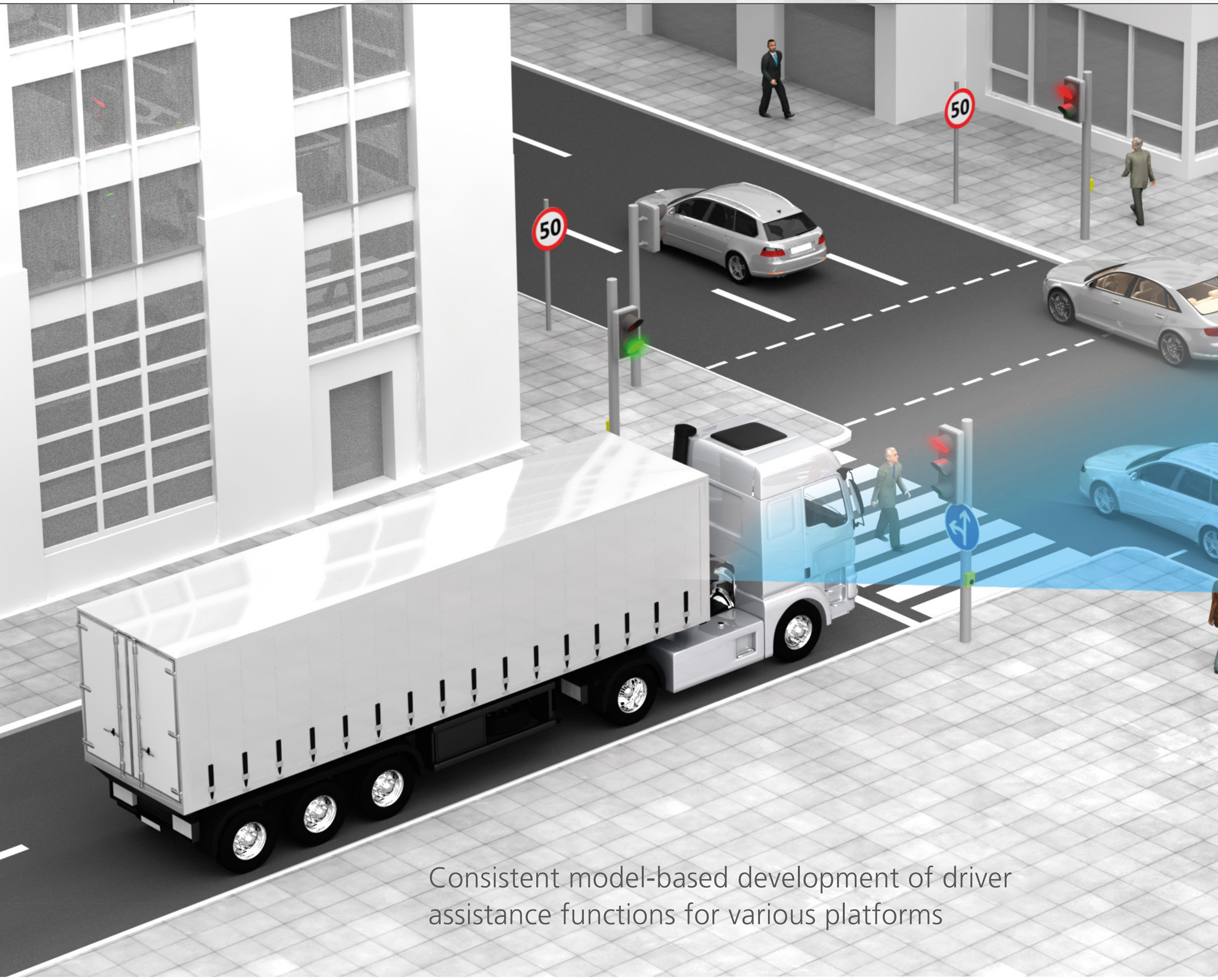
Anja Poppe, Josef Laumer, Continental

Figure 2: One of the HIL test rigs at Continental in Regensburg.

Figure 3: Multiscope instrumentation in ControlDesk Next Generation.

Anja Poppe
Anja Poppe is Software Test Manager and responsible for test strategies and test equipment for Software & Systems Engineering Hybrid & Electric Vehicle at Continental in Regensburg, Germany.

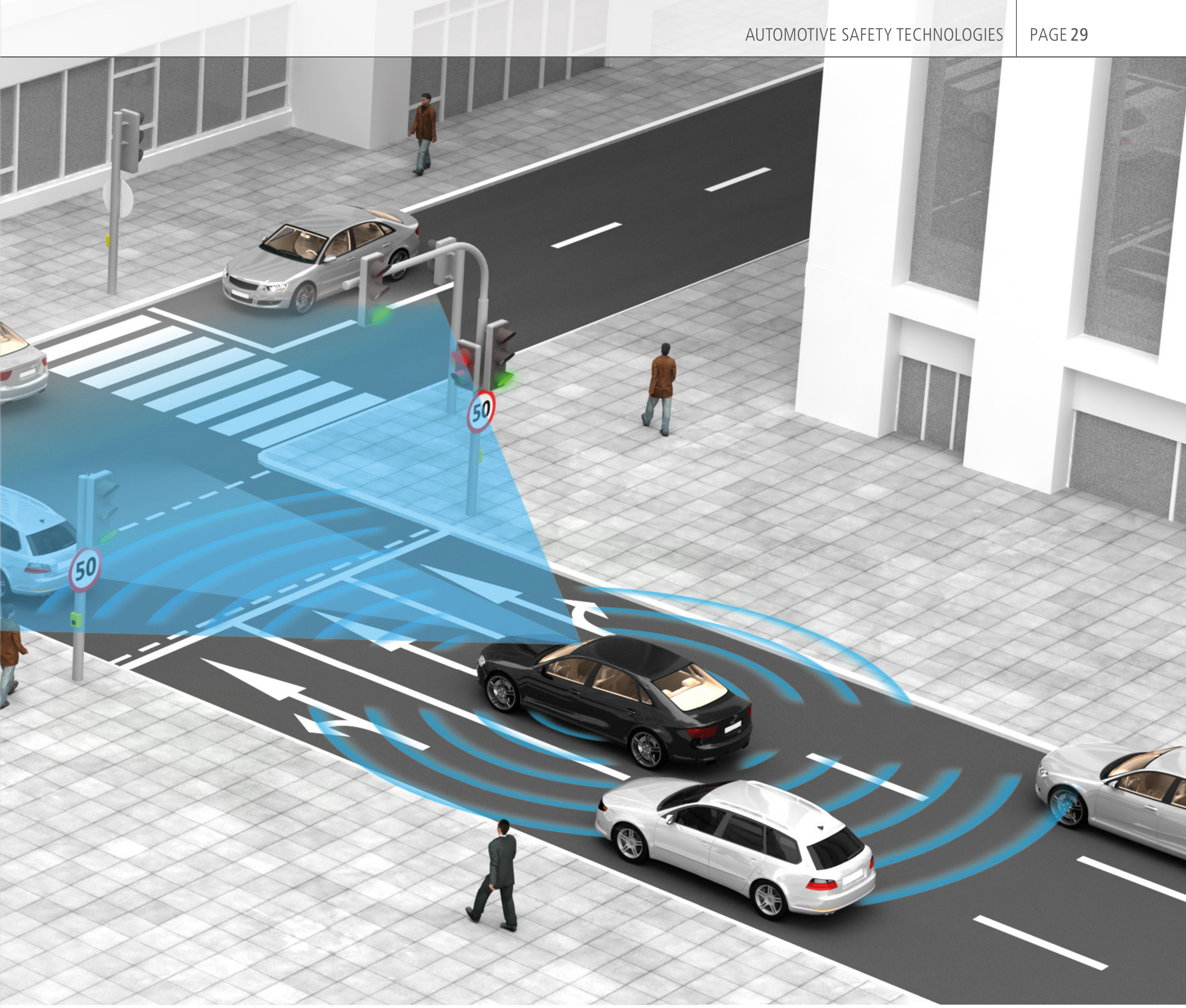
Josef Laumer
Josef Laumer is Function Developer for electrical machine control for Software & Systems Engineering Hybrid & Electric Vehicle at Continental in Regensburg, Germany.



Consistent model-based development of driver assistance functions for various platforms

Developing Intelligent Assistants

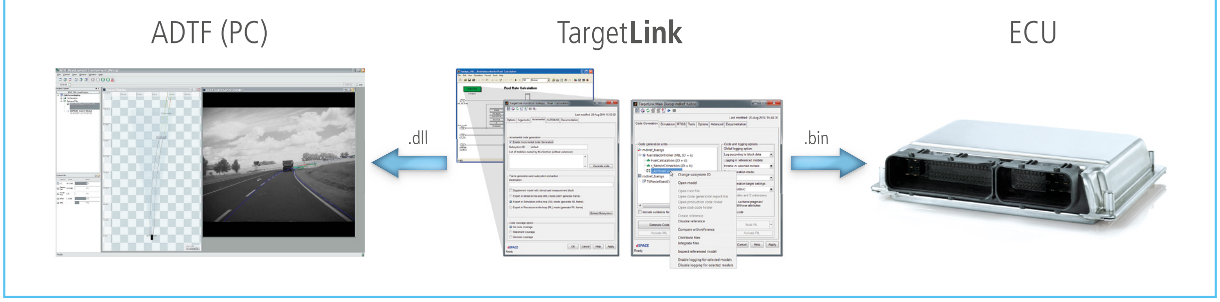
In an advanced development project for sensor-based driver assistance systems at Automotive Safety Technologies GmbH, the goal is to handle and analyze complex data reliably. The production code generator TargetLink supports this task and enables a seamless, efficient workflow.



The field of sensor-based driver assistance, with its multitude of data fusions from many different sensors, generates complex algorithms and data structures, as well as large amounts of data. Advanced development projects that focus on cross traffic and intersection traffic are all about the efficient processing of:

- Data fusions from many different sensors (camera, radar, laser)
- Building objects on the basis of raw laser data
- Camera-based multi-hypothesis object tracking (of vehicles, pedestrians, etc.)

To ensure that diagnoses of the complex functions can be performed during model-based development, a high volume of function-internal information, i.e., information that is not directly accessible, must be made available for validation and testing. These bits of detailed information require additional memory and an increased run time, which is only acceptable during the development phase. After the function has been delivered, this information is not needed any longer. Instead, the focus is now on minimal memory requirements and an optimal run time. This culminates in the need for a software component (SWC) that can be created scalably in terms of its non-functional scope and compiled flexibly to match the specific application.
During the development phase, suitable target platforms are often >>



Simplified representation of integrating a software component (SWC) generated from a function model: While in the control unit the SWC is integrated as a binary file (.bin), the ADTF includes the SWC as an executable library (.dll, Windows).

not available yet and it is important to implement prototype functions quickly, without any relevance to specific platforms, especially during preliminary development projects The development environment ADTF (Automotive Data and Time-Triggered Framework) is a fitting tool for this, making it possible to run functions on a PC and connect them with other components.

Development Environment for Sensor Data

ADTF is a development environment used for synchronously recording sensor data online in the vehicle and supplying the relevant function.

ADTF is also used for playing the recorded data offline in order to let the functions run independently of the recording time. In this environment, components can be self-programmed, or existing components can be controlled on the ECU. Thus, in principle it is possible to use the same model for generating the ECU code and to create a Windows®/Linux library. Another requirement, as mentioned above, is that during the development phase a maximum amount of development-relevant, function-internal information must be accessible so that the functionality can be analyzed and understood, especially if the functionality behaves incorrectly. To achieve accessibility, though, this internal data must already be provided by the model. The component that calls the compiled model within the development environment can then analyze and visualize this data or forward it to other components.

Requirements for Technical Implementation

To implement the functions in executable ECU software, the production code generator TargetLink® by dSPACE comes into play. With this tool, code for a SWC is generated from the function model and integrated into both the controller and the development environment. It must be possible to trace and visualize a behavioral analysis of the SWC at run time for these two platforms. This happens in both run time environments (the ECU and the development environment) but differently. On the ECU, the SWC makes the predefined runtime variables accessible to an external tool via a measurement and calibration protocol (XCP). In comparison, in the development environment any SWC variables can be transferred to other programs and visualized if desired.
Since a PC-based environment does not require any memory from the SWC, unlike an ECU, the SWC can be extended by any number of further debug variables to fully exploit the debugging capabilities. The chal-

Specifications in the TargetLink Data Dictionary to provide the desired flexibility in code generation for debug variables.

Property	Values
Description	"Variable class"
Storage	default
Scope	global
ArgClass	< >
Volatile	off
Const	off
Macro	off
Alias	off
InitAtDefinition	off
RestartFunctionName	"InitPredictionVariables" Initialization function
SectionName	"
TypePrefix	"UAS_API" Compiler switch
DeclarationStatements	{ }
UseName	on

> "With TargetLink, we are implementing an automated, seamless workflow for the efficient development of high-performance driver assistance functions that supports a switchable code generation for both the development platform and the target processor."
>
> *Matthias Issbruecker, Automotive Safety Technologies GmbH*

lenge lies in leaving the ECU code largely unchanged on the one hand, and taking account of the requirements for integrating the SWC on the other, such as creating as many variables for debugging as possible and making these variables available. Another challenge is the high level of automation for integration in each target platform (Linux or Windows) to reduce the level of manual interventions during the workflow. The solution is based on modifications made in the SWC model, in the SWC database, and in appropriate scripts.

Automated Implementation with TargetLink

To achieve this automation and support the integration of TargetLink in the development process, TargetLink's callback mechanism was used. TargetLink-specific hook functions support customer-specific modifications during the compilation process. Customer-defined instructions are executed automatically when the hook functions are called, making additional manual adjustments unnecessary.
With TargetLink, it was possible to implement the additional debug variables as arrays that were either created (development environment) or omitted (ECU) during the compilation process via a compiler switch. Compilation variants can be used to develop the same code for the different development platforms and to reduce the code to the desired form during the compilation process. Since these debug variable arrays can have different widths and data types, a reusable generic solution was implemented in the form of a library. In addition, with preprocessor instructions and automatically inserted prefixes, it was possible to generate identical C code for different run-time environments (ECU, development environment for Windows or Linux), depending on the compiler switch. ■

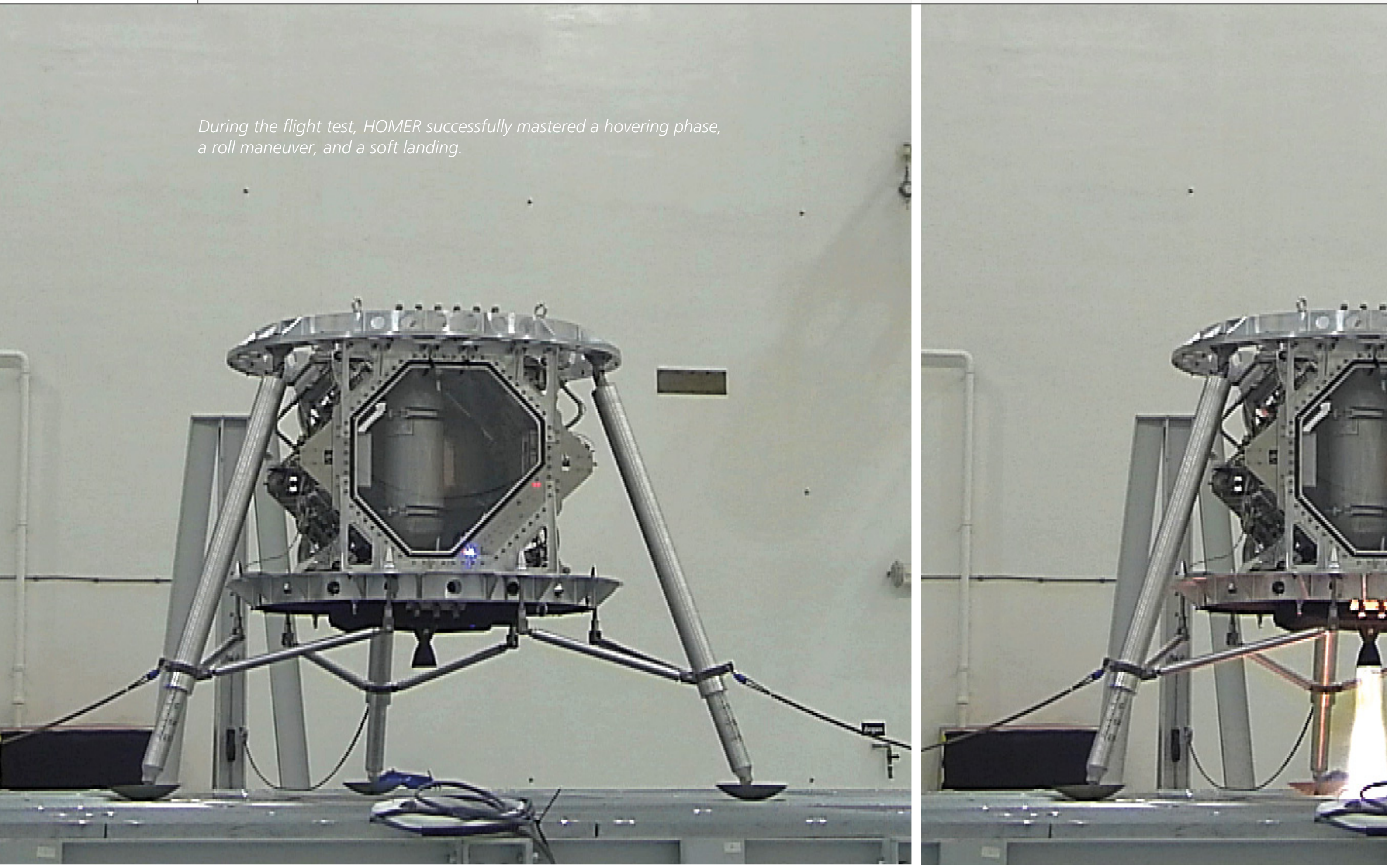
Matthias Issbruecker, Automotive Safety Technologies GmbH,
Mohinder Pandey

Conclusion

To meet the requirements for efficient integrated development, Automotive Safety Technologies uses the production code generator TargetLink to generate target-platform-oriented code. A compiler switch is used to generate platform-specific code automatically from a function model. The code is either equipped with additional debugging variables or optimized to run on ECUs. The advantage is that the development environment ADTF is fed from the exact same function model in order to develop driver assistance functions comfortably, to generate the code for the ECUs, and to test the code under real run-time conditions.

Matthias Issbruecker
Matthias Issbruecker works on the development of intersection assistants at Automotive Safety Technologies GmbH in Gaimersheim, Germany.

Mohinder Pandey
Mohinder Pandey worked on the preliminary development of intersection assistants at Automotive Safety Technologies GmbH in Gaimersheim, Germany.

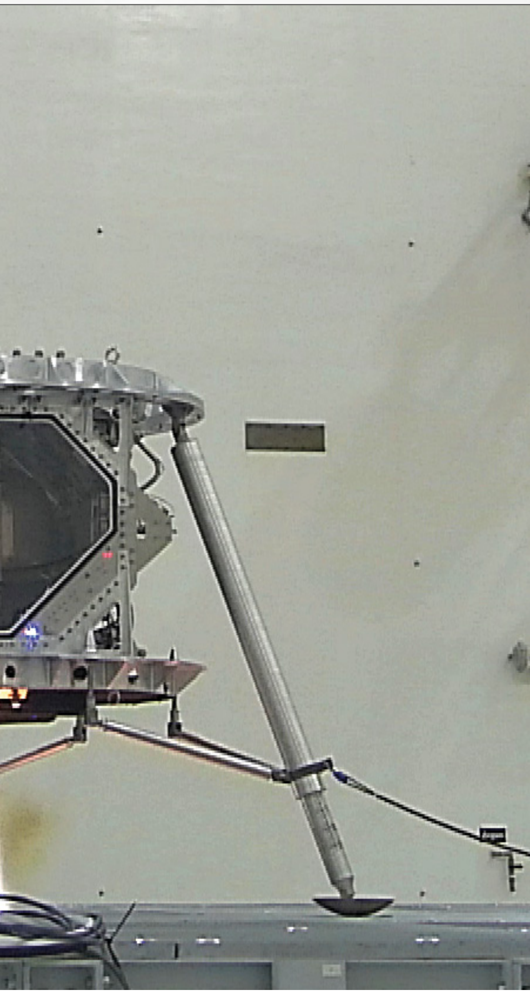


During the flight test, HOMER successfully mastered a hovering phase, a roll maneuver, and a soft landing.

HOMER Take-off: A Review

With HOMER (HOver ManoEuvRe), Airbus Defence and Space had developed an innovative two-in-one prototype for future space vehicles that masters both landing and hovering maneuvers. Two dSPACE MicroAutoBoxes were on board to control the test flights.

At Airbus Defence and Space, the HOMER project was a novelty when it comes to the products it used. For the very first time, commercial off-the-shelf (COTS) products were used for such a complex development task. HOMER was one of Airbus Defence and Space's five most distinguished projects. Its goal was to assess the maturity of new technologies and the associated know-how, and to develop new key competencies. HOMER can be thought of as a kind of incubator for new technologies. One of the main challenges of its development was the limit on weight and volume (max. 300 kg for a volume of 1 m^3). The requirements for other projects at Airbus Defence and Space are less strict because they develop far



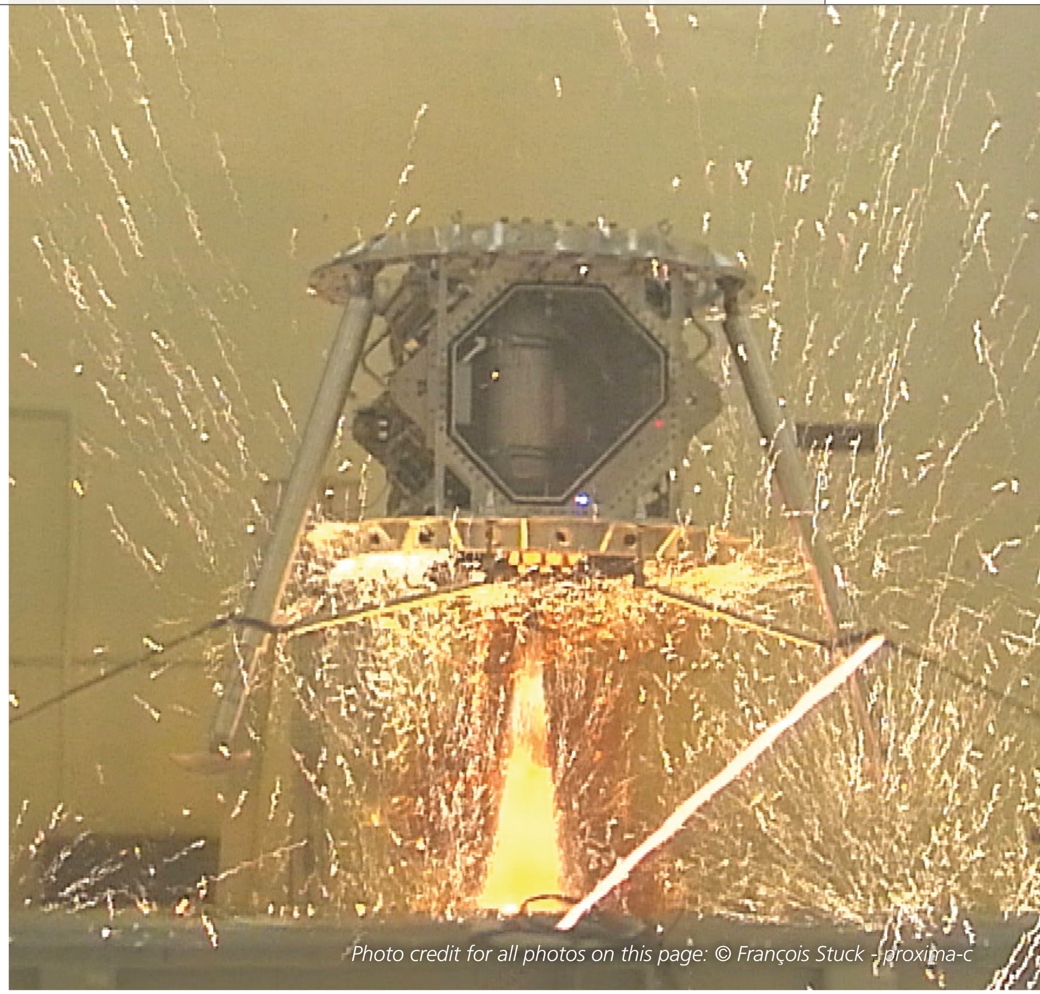


Photo credit for all photos on this page: © François Stuck - proxima-c

> "Our successful utilization of dSPACE products for the HOMER project allows us to strongly and confidently consider dSPACE solutions for further R&T projects."
>
> *Stéphane Heynen, responsible for Ground Control Systems, Airbus Defence and Space*

larger space vehicles of twenty tons or more.

Space Vehicle in Two Variants

HOMER can be configured for two use cases. One version was optimized for landing maneuvers, the other for hovering maneuvers (figure 2). The landing version of HOMER (Lander, for the ODYSSEY mission) has shock-absorbent landing legs and an engine for vertical movement. The hovering version (Impactor, for the ILIAD mission) does not have a landing system but rather two additional engines for lateral movement. The first tests focused on the Lander version.

Interdisciplinary Development Across Countries

Six Airbus Defence and Space offices were involved in the HOMER project: two from France, four from Germany. A new organizational structure had to be created from scratch to guarantee seamless cooperation. A typical example for this is the collaboration of teams from simulation and flight control, who were merged at Airbus Defence and Space. This project team counted approx. 25 employees who were divided into several groups of both French and German engineers. Each group focused on one subtask, such as the attitude control system or the main propulsion system.

Rapid Control Prototyping with dSPACE Tools

The COTS system for developing HOMER's hardware and software had to meet strict requirements for I/O capability, weight, and configurability, because a lean and cost- >>

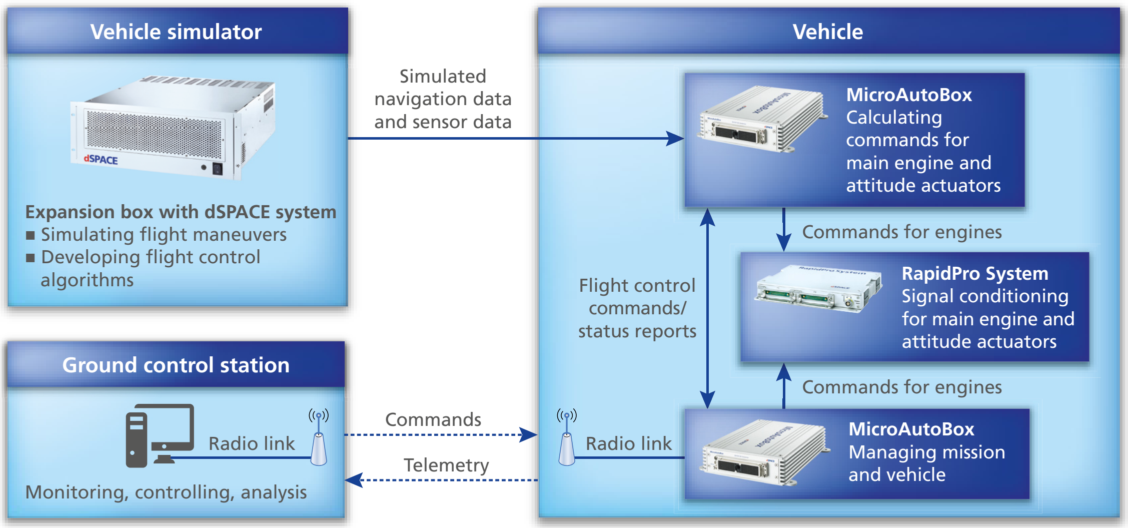


Figure 1: To develop the flight control algorithms, HOMER was connected to a modular dSPACE system in a dSPACE Expansion Box. This made it possible to simulate the flight maneuvers and associated sensor data before the actual test flight.

effective development system must not compromise safety. The choice eventually landed on dSPACE tools because they had been tried and tested in previous projects at Airbus Defence and Space. The flexible I/O of dSPACE MicroAutoBox made it easy to connect existing hardware. This hardware included an inertial navigation system, a camera, a radar altimeter, as well as different sensors and actuators. Function design was made easy by the model-based approach, i.e., model design in MATLAB®/Simulink® and automatic code implementation on the dSPACE hardware (figure 1) via Real-Time Interface (RTI). "The most striking advantages of the dSPACE tools are their easy configuration directly from the block diagram, the many possibilities for instrumentalization in the experiment software ControlDesk, and their real-time capability," says Thierry Poirrier, who was in charge of developing HOMER's electrical subsystems. During the development process, a total of five MicroAutoBox configurations and RapidPro power stage configurations, including the associated implementation and experiment software, were used.

Test Flight with Two MicroAutoBoxes

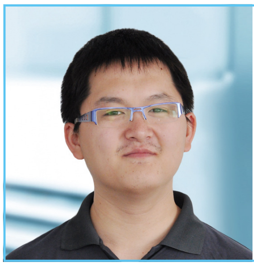
HOMER was assembled at the Airbus Defence and Space facilities in Bremen and then shipped to the test center in Aquitaine, France. On October 23, 2012, HOMER passed the validation test, which consisted of a hovering phase, a roll maneuver, and a soft landing from 1 m above ground (see video). "HOMER has been approved at system level," reports Stéphane Heynen, who is responsible for ground control systems. Two dSPACE MicroAutoBoxes were installed in HOMER: one dedicated to the mission and onboard system management, the other to concrete flight control. Because of the extreme vibrations that HOMER is subjected to, especially during takeoff and landing, an environment expert recommended putting additional absorbent foam inside the MicroAutoBoxes to ensure that they work properly during the test flight.

Accelerating Development by 9 Years

"For this prototype, it took us only 4 years to get from our first work

"We used dSPACE products all along the development and validation process. The cost-effective dSPACE tools were used on all tests systems and ensured the representativeness of the test results."

Thierry Poirrier, responsible for Electrical Subsystems, Airbus Defence and Space



"The dSPACE products provide fast, reliable, and robust real-time prototyping tools, letting us focus on our core activities."

Clément Gu, responsible for Simulation and Flight Control Software, Airbus Defence and Space

steps to finishing the Lander version. For an operational vehicle, this process usually takes around 15 years in space flight projects," states Clément Gu, who developed the simulation and flight control software. Airbus Defence and Space is the first space flight company in Europe with such test flight competence. Because the new technologies and work methods proved successful, Airbus Defence and Space will continue to use them in future space projects. One such project could be a space vehicle for removing space debris, which requires high-precision propulsion, control, and docking abilities. ■

By kind permission of Airbus Defence and Space.

Figure 2: Top: the Lander version (one vertical engine, three landing legs).
Bottom: the impactor version (one vertical engine, two horizontal engines instead of landing legs).

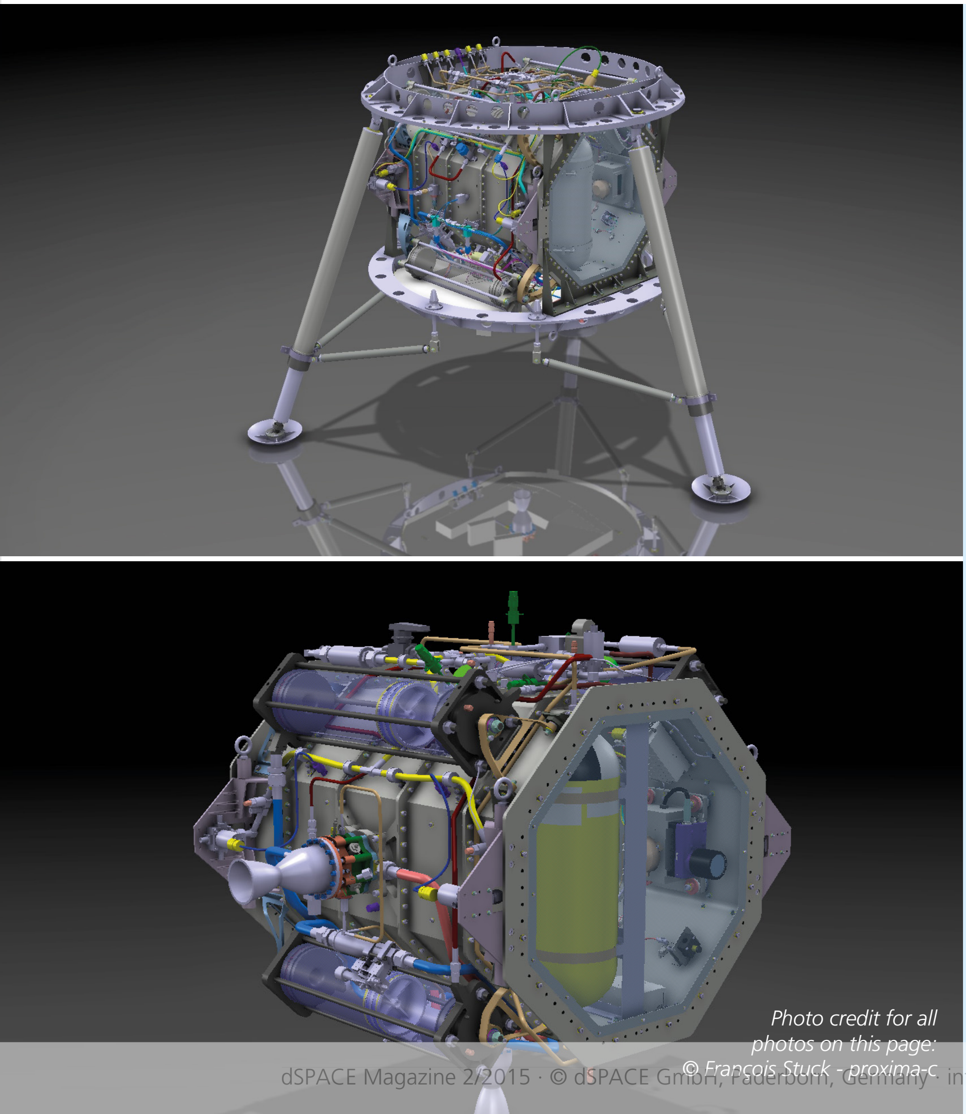


Photo credit for all photos on this page: © Francois Stuck - proxima-c

Conclusion

With HOMER (HOver ManoEuvRe), Airbus Defence and Space had developed an innovative multi-functional prototype for future space vehicles that mastered both landing and hovering maneuvers. Various dSPACE products were involved in HOMER's development test flights: e.g., two MicroAutoBoxes for onboard flight control. HOMER took only 4 years to develop instead of the usual 15 years for such a space flight project. The HOMER project has now been successfully completed.

This video shows HOMER's first test flight: www.dspace.com/go/dMag_20152_HOMER

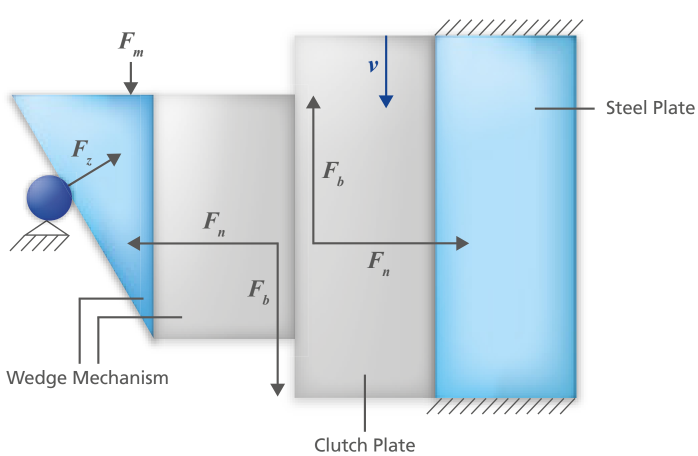
Previous attempts at wedge clutches in automatic transmissions often failed because of unpleasant shift jerking. Researchers from Shanghai Jiao Tong University are working on solving this problem through a precisely controlled electric motor. The validation is based on hardware and software tools by dSPACE.

The current trend in the automotive industry of electrifying the drivetrain sometimes also touches the transmission system. For example, in automatic transmissions, electric actuators could be capable of replacing conventional hydraulic clutches. One reason is that electric actuators are usually more compact and lighter than hydraulic actuators, another is that they do not need a permanently-running combustion engine to sustain the built-up forces. Because this can save a significant amount of fuel, researchers at the Shanghai Jiao Tong University (SJTU) recently evaluated an electrically driven wedge clutch.

A Challenging Transition

The wedge mechanism is driven between a rotating clutch plate and a fixed counter bearing (figure 1). The further the electric motor drives the wedge, the tighter the clutch plate is pressed against its counterpart. With a constant wedge angle, however, a critical point is eventually reached at which the ratio between the drive force of the electric servomotor and the friction force on the clutch plate is enormous. This means that even a small actuating force can suddenly cause a very large normal force on the clutch plates. In the transmission output torque, this sudden transition between "sliding" and full traction translates into a distinct jerk that >>

Figure 1: Force analysis on a simplified depiction of the wedge clutch mechanism. Even a small actuating force F_m can cause a very large normal force F_n on the clutch plates, translating into a distinct jerk that would seriously affect ride comfort.



Smooth and e-fficient

Electric wedge clutch for smoother shifts

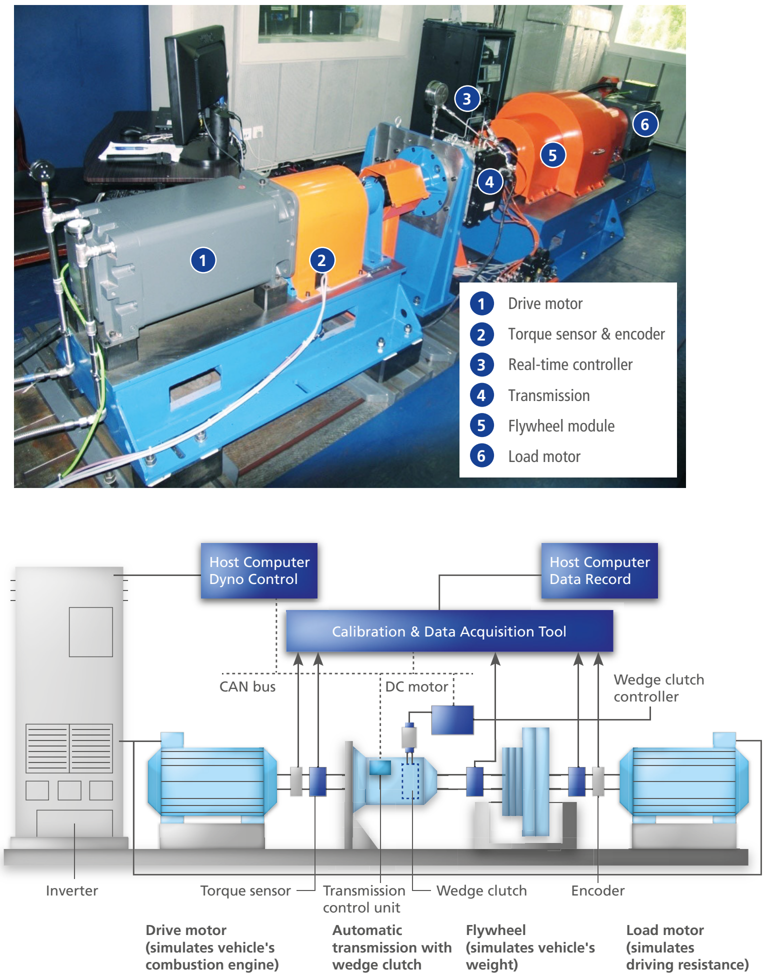


Figure 2: The dynamometer test rig emulates a mid-class passenger car for the effective validation of the wedge clutch system.

would seriously affect ride comfort and thus destroy the chances of the wedge clutch going into series.

Validation on a Test Bench

By using an optimal control of the electric actuator, researchers at SJTU are working on preventing exactly that problem, making it possible for the wedge clutch to smoothly engage and disengage at lower torques and with low power losses. To quickly, accurately and precisely validate the feasibility of this kind of control, and thus the wedge clutch, on their dynamometer test bench, the engineers used MicroAutoBox II from dSPACE together with a PID (proportional-integral-derivative) controller. The controller's task is to control the fluctuations of the normal force on the clutch plates within a few milliseconds, thereby optimizing the switching behavior.

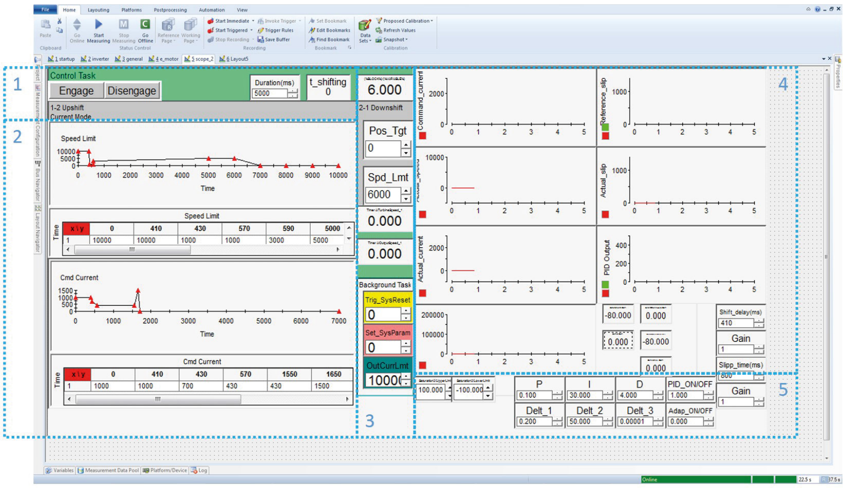
Simulating a Mid-Class Car

For the input value, the PID controller uses the deviation between the predetermined desired and actual normal force on the clutch plate. This deviation is the basis for controlling the voltage of the DC-powered servomotor as an output variable. Together with the rotation angle, the voltage is implemented as a corresponding load torque on the wedge clutch. The wedge clutch is built into the transmission bell housing in the center of SJTU's dynamometer test bench (figure 2). On the drive side, a fast-response electric motor emulates car combustion engines with cylinder capacities of up to 1.6 liters and torques of up to 297 Nm at about 6500 rpm. On the output side, there is a flywheel, which simulates the inertial mass of a mid-class car, and a load motor for reproducing different driving resistances.

Adaptive PID Controllers in Action

By using the PID controller in a closed loop, slip differences between the clutch plates of the drive side and the output side were reduced to a minimum in a short time so the gears could switch smoothly. Because conventional PID controllers produce unstable results under complex driving conditions and varying system parameters, the researchers at SJTU used adaptive PID controllers with operating-point-dependent parameter control

> "Our researchers were impressed by the seamless integration of dSPACE's tools. This gave them the chance to concentrate completely on developing the algorithms."
>
> *Jian Yao, Shanghai Jiao Tong University*



1) Switching control signals for clutch engagement and disengagement

2) Calibrating the DC motor's actuation signals for
- Actuating clutch travel
- Providing the normal force

3) Setting the initial values for
- DC motor control limits
- Communication parameters
- Sampling frequency
- System initialization

4) Displaying
- Slipping errors between clutch plates
- DC motor current
- Rotary speed and angle
- Control outputs

5) Setting initial values for
- PID control parameters
- Adaptive gain

Figure 3: ControlDesk Next Generation helped the SJTU researchers create a rich user interface for monitoring and managing signals in their test environment.

in order to control the real servo-motor on the test bench.

Tests Like Under Real Driving Conditions

The researchers develop the control concepts in MATLAB®/Simulink® and compile and execute them on a dSPACE MicroAutoBox II during the tests. The results showed that the PID controller exhibits a robust performance and complies with all of the reference characteristics when the system parameters and driving conditions are changed, making it possible to control the transmission as under real driving conditions. In addition to benefitting from MicroAutoBox, whose numerous I/O interfaces were easy to integrate, the SJTU test bench also profited from dSPACE's experimentation and visualization software, ControlDesk® Next Generation.

Comprehensive Test Instrumentation

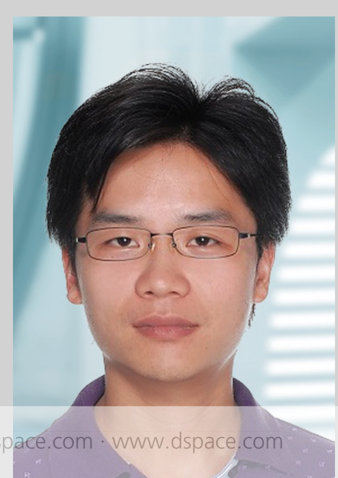
With the numerous instruments in ControlDesk, the researchers at SJTU created a rich user interface for monitoring and managing the signals in their test environment (figure 3). They were therefore able to perform tasks such as calibrating the control signals for the DC motor or setting the important initial values and parameters of the PID controller on the computer screen, to name just a few examples. In addition, they reproduced and analyzed all the important signals in detail, from the slip errors between the clutch plates, to the current, speed and angle of the DC motor, to various control outputs.

Paving the Way for Future Tests

Above all, the researchers at SJTU were particularly impressed by the seamless connection between the dSPACE tools and MATLAB/Simulink. This coupling gave them complete trust in the compact and robust hardware system, letting them concentrate on algorithm development. Because of their positive experiences, they are planning further tests to complete the developed control algorithms and develop the actual controller for their promising wedge clutch. Their future development activities plan to include further dSPACE tools such as the production code generator TargetLink® or a hardware-in-the-loop (HIL) simulator. ■

Jian Yao,
Shanghai Jiao Tong University

Jian Yao
Jian Yao is a PhD student at Shanghai Jiao Tong University (SJTU) majoring in automatic transmission control and research, with special expertise in the electrification of actuators and the transmission system.



The ISO 26262 standard ("Road vehicles – Functional safety") became effective in November 2011. It describes processes and methods for ensuring functional safety during the development and production of safety-relevant vehicle electrics/electronics systems. The standard defines the state of technology and is therefore mandatory for car manufacturers. But how can manufacturers translate the generic requirements into concrete development measures? One approach is using reference workflows and tools that were qualified or certified by accredited certification authorities such as TÜV. In the end, this approach also reduces product liability risks.

ISO 26262 Requirements for Using Software Tools

ISO 26262 stipulates that a software tool used in safety-relevant projects must first be classified for the specific field of application. The classification includes analyzing the impact of the tool on fulfilling or violating functional safety requirements. The classification result is called the tool confidence level (TCL). The tool might then have to be qualified based on the TCL (see info box page 42). Usually, the qualification is performed by using a combination of suitable qualifiation methods, which depends on the Automotive Safety Integrity Level (ASIL) of the system that is being developed and tested with the software tool. For the users, however, implementing these qualification methods is very difficult and effort-intensive, because it requires sound knowledge of the processes and methods involved in the tool's development. Therefore, implementing the methods requires expert knowledge on >>

In 2014, TÜV SÜD certified dSPACE's test automation software, AutomationDesk, for testing safety-relevant systems according to ISO 26262 and IEC 61508 – making AutomationDesk the first-ever commercial test automation software for hardware-in-the-loop (HIL) simulation to receive this certificate. But what exactly does this mean for the users?

Certifiably Safe

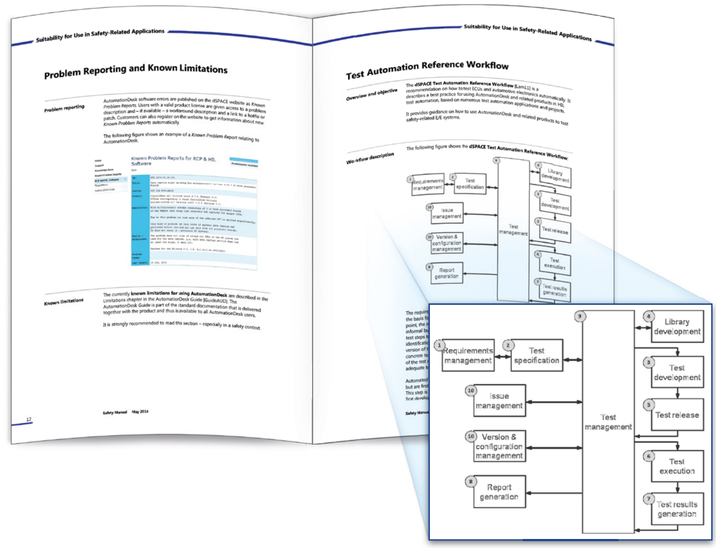
The TÜV SÜD certificate for AutomationDesk makes the classification, qualification and validation according to the ISO 26262 and IEC 61508 standards much easier

What Do ISO 26262 and IEC 61508 Define?

IEC 61508 is the internationally recognized generic standard for the development of safety-related electronic systems. In the automotive industry, ISO 26262 replaced IEC 61508 in late 2011 as the standard for the functional safety of passenger cars. ISO 26262 is a modification of IEC 61508, adjusted to the specific conditions in the automotive sector. Implementing these standards increases the functional safety of a vehicle system with electrical/electronic components. In order to adhere to these standards, the tools that are used for developing the electrical/electronic components have to be classified, qualified and/or validated.

ISO 26262 defines four methods for tool qualification. Depending on the respective ASIL, the users select a suitable combination of these methods:

a. Increased confidence from use (ISO 26262-8, 11.4.7 to 11.4.10)
b. Evaluation of the tool development process (ISO 26262-8, 11.4.8)
c. Validation of the software tool (ISO 26262-8, 11.4.9)
d. Development in accordance with a safety standard (ISO 26262-8, 11.4.10)



The AutomationDesk Safety Manual with the Test Automation Reference Workflow.

assessing tool development according to ISO 26262.

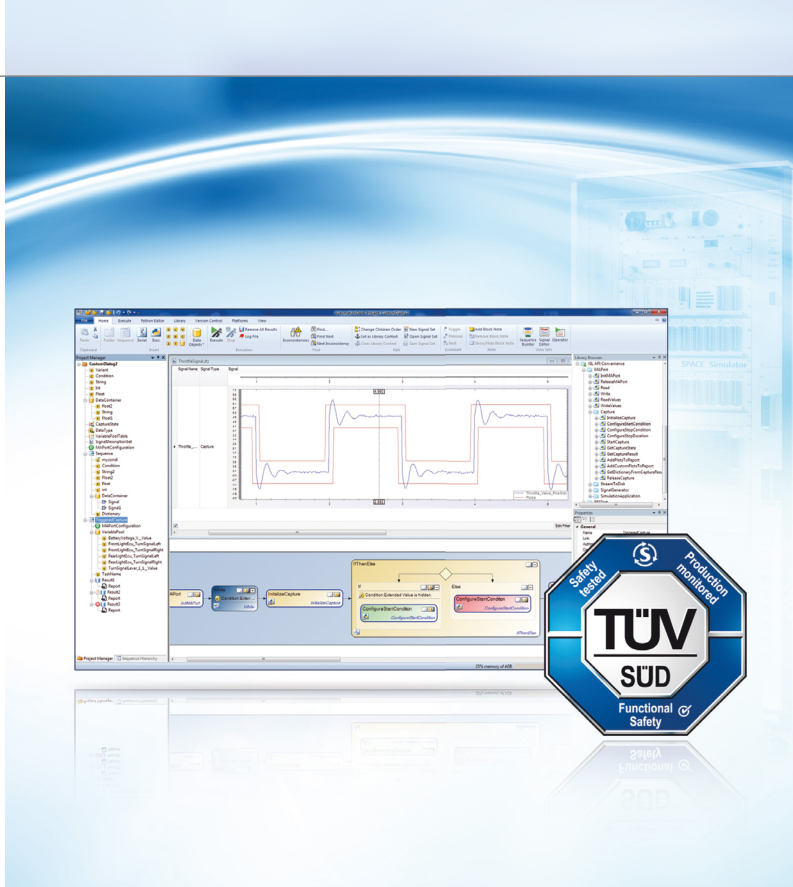
TÜV SÜD Certificate

In close cooperation with dSPACE GmbH in Paderborn, TÜV SÜD particularly analyzed dSPACE-internal processes and documentation to assess the development and quality assurance of AutomationDesk (see info box page 43). TÜV SÜD also analyzed the customer communication process and bug reporting as well as the new AutomationDesk Safety Manual. As a result, TÜV awarded AutomationDesk the "Software Tool for Safety-Related Development" certificate, which underlines that AutomationDesk is suitable for testing safety-relevant systems in the automotive, commercial vehicle, and aviation sectors, and many other areas of industry. This officially confirms that AutomationDesk is "fit for purpose" for testing safety-relevant systems according to ISO 26262 and IEC 61508, for all Automotive Safety Integrity Levels, from ASIL A to ASIL D and SIL1 to SIL3, respectively.

The AutomationDesk Safety Manual

One key component of the certification is the AutomationDesk Safety Manual, a supplemental user documentation for using AutomationDesk in safety-relevant projects. In particular, it provides

- A general overview of AutomationDesk and all available product and user documentation
- Recommendations and best practices for commissioning and using AutomationDesk, including the AutomationDesk Test Automation Reference Workflow
- An example of the Test Automation Reference Workflow in the test process defined by ISO 26262
- A tool classification for AutomationDesk
- The "fit for purpose" certificate for AutomationDesk according to ISO 26262 and IEC 61508



AutomationDesk: First certified commercial software for HIL test automation.

Valuable Support for the Development of Safety-Relevant Systems

With this certificate, TÜV SÜD, as an independent organization, confirms the high quality of the development process and the comprehensive quality assurance for AutomationDesk. And if AutomationDesk is always used as described in the AutomationDesk Safety Manual, users do not even have to go to the lengths of creating their own tool classification. The certificate also makes qualifying AutomationDesk much easier, because it attests that the software fulfills the relevant criteria of the qualification methods "Evaluation of the tool development process" and "Validation of the software tool" and most of the criteria of "Increased confidence from use".
AutomationDesk was the first commercial software product for test automation in a hardware-in-the-loop context to be awarded such a certificate. Together with the AutomationDesk Safety Manual, the certificate gives users invaluable support for using AutomationDesk to develop and test safety-related systems according to ISO 26262 and IEC 61508. The TÜV SÜD certificate for AutomationDesk facilitates the classification, qualification, and validation according to the standards.

Outlook

The certified software version is AutomationDesk 4.1. Another certification process for the new AutomationDesk version 5.0, which will be available as of summer 2015, is close at hand. Future versions will be certified as needed. For more information on the TÜV SÜD certificate for AutomationDesk, please contact your sales representative. ■

Basis of the TÜV SÜD Check

In order to certify AutomationDesk, TÜV SÜD audited the dSPACE departments responsible for AutomationDesk.
TÜV SÜD checked

- The AutomationDesk development process, e.g., requirements management, change management, release management
- The validation of AutomationDesk, e.g., traceability of test results according to the requirements
- The customer communication process
- The AutomationDesk Safety Manual

These factors were assessed on-site based on all relevant development documents for AutomationDesk, such as the Specifications Document, detailed functional and component-oriented specifications, design documents, test catalogs, test results, process documentation, company-wide development regulations and their project-specific modifications.
The cooperation with the different departments was key to the fast and positive certification result. TÜV SÜD found that the AutomationDesk processes were already very well suited. dSPACE quickly implemented the few suggestions for improvement for the audit and was consequently awarded the certificate.

Rapid AUTOSAR

MicroAutoBox II as a powerful, AUTOSAR-capable prototyping and development platform

With its new RTI AUTOSAR Blockset 2.0, dSPACE now supports the execution of entire ECU applications on MicroAutoBox II. The result: many more possiblities and higher productivity than in ECU-based approaches.

Efficient processes are essential for the success of production-oriented development projects. One important component is the seamless transition between rapid control prototyping (RCP) and AUTOSAR-based production software development. For example, developers want to reuse existing software components (SWCs) to design new controllers in MATLAB®/Simulink®, test new control algorithms in the ECU software, or validate the complete application software of an ECU as early as possible. This makes AUTOSAR support in the early development stages more important every day. But the technical complexity of AUTOSAR makes it difficult for function developers to use the standard. Furthermore, the common AUTOSAR tools are especially geared towards software experts. The RTI AUTOSAR Blockset 2.0 and MicroAutoBox II, dSPACE's compact rapid control prototyping system, bridge the gap between model-based control design and AUTOSAR production software, making it easier for function developers to master the complex matter.

Easy Software Reuse

Often, new control algorithms build on existing functions. In the past, if these functions did not exist as Simulink or TargetLink® models but as production-ready C code, reusing them for model-based development often entailed a great deal of work. Either the models had to be developed afterwards, which is time-consuming, or the C code was adapted for Simulink with customer-specific, and therefore cost-intensive, solutions. The RTI AUTOSAR Blockset 2.0 makes it possible to import software components that were developed according to the AUTOSAR standard into Simulink without additional effort. The software components can also be integrated in new controller models.

Early Start for Prototyping and Testing

Testing new control strategies together with the ECU software, and testing and evaluating the controller behavior in combination with the real plant model are important components of the development process and often take place in the real vehicle. However, the first prototypes of new ECUs often become available only months after the start of development, and at first only in limited numbers. Moreover, not all basic software (BSW) modules are completely implemented and tested from the beginning, such as I/O drivers for new sensors and actuators. These factors can significantly delay the tests of new control strategies. The RTI AUTOSAR Blockset 2.0 lets developers use MicroAutoBox II as a universal AUTOSAR-based development ECU, making prototyping >>

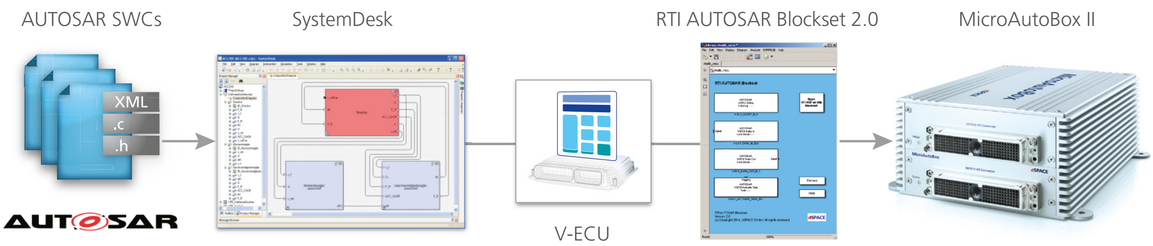


Figure 1: AUTOSAR-based workflow for rapid control prototyping.

MicroAutoBox II and AUTOSAR

MicroAutoBox II provides a comprehensive AUTOSAR real-time operating system with low I/O latencies, similar to those of production ECUs. It supports AUTOSAR versions 3.x and 4.x. Compliance with AUTOSAR OS Scalability Class 1 ensures a high degree of compatibility with the AUTOSAR standard. In addition, the most important basic software services, such as the ECU State Manager (EcuM), NVRAM Manager (NvM), and AUTOSAR CAN Stack, are supported. AUTOSAR Software Components can therefore be used in a realistic AUTOSAR environment for rapid control prototyping, benchmarking, and tests, long before the first prototypes of a production ECU become available.

and testing possible long before the first prototypes of the production ECU are available.

Flexible AUTOSAR Development Platform

One main goal of AUTOSAR is the easy integration of software components from different suppliers into one ECU software. Suppliers are facing the challenge of extending their functionalities by customer-specific functions and testing them under customer-specific operating conditions. With MicroAutoBox II as an AUTOSAR-capable development system, suppliers can modify their software components very early and test them efficiently. They do not have to wait for prototypes of the production ECU, nor do they have to know the project-specific development environment. The many parameters and their complex dependencies make configuring the BSW in the traditional production tools cumbersome and susceptible to errors. The model-based approach, in contrast, makes configuring the operating system and I/O of MicroAutoBox II intuitive, and customer-specific actuators and sensors can be connected fast and flexibly. Customer-specific test environments can therefore be set up with minimum effort. This lets suppliers develop prototype solutions for a customer independently of the customer's tools.

Seamless Tool Chain

In addition to the RTI AUTOSAR Blockset 2.0 and MicroAutoBox II, dSPACE provides powerful tools for model-based software development, software integration, and offline validation: TargetLink, SystemDesk®, and VEOS®. Together, these tools cover all phases of function and software development, from designing new control functions to the software architecture and software integration, from PC-based simulation to rapid control prototyping and in-vehicle testing. The tools also allow for a seamless development process (figures 1 and 2). The experiment software ControlDesk® Next Generation completes the tool chain. It lets developers use the same measurement and calibration data and experiment layouts for all the platforms up to the production ECU, reducing the adaptation effort. The link between software development and subsequent prototyping and testing activities is dSPACE SystemDesk. With SystemDesk, developers can import AUTOSAR SWCs and connect them to build application software. SWCs can come from the production code generator TargetLink or from other AUTOSAR tools. Engineers can also

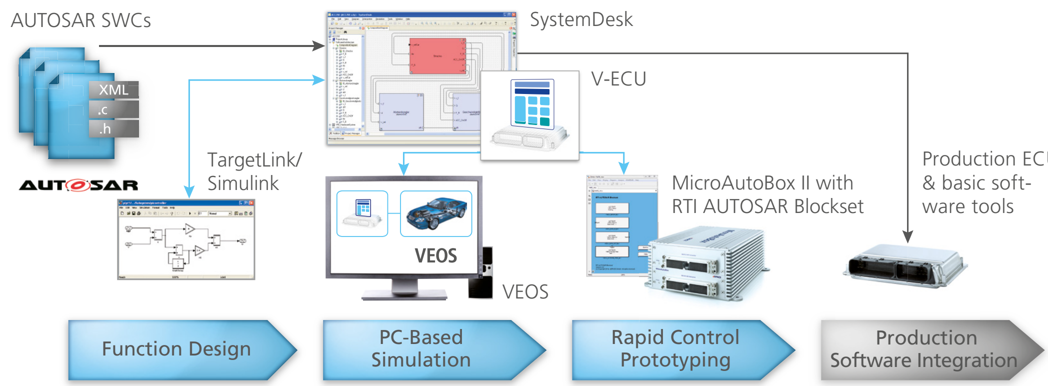


Figure 2: Seamless dSPACE tool chain for function design, rapid control prototyping, and testing.

dSPACE's seamless AUTOSAR tool chain improves productivity in the development process.

use SystemDesk to configure the AUTOSAR operating system, basic software, and run-time environment (RTE) of the target system or use existing configurations from other AUTOSAR tools. Based on this information, the SystemDesk V-ECU Generation Module can be used to generate virtual ECU (V-ECU) software that can be used with the PC-based simulation platform VEOS for offline validation and the RTI AUTOSAR Blockset 2.0 for rapid control prototyping. In both cases, the V-ECU can be enhanced with additional functions based on Simulink models.

Maximum Test Coverage

While the functional characteristics of a V-ECU can be validated efficiently via offline simulation with VEOS, the same V-ECU can be executed on MicroAutoBox II together with the physical plant by using the RTI AUTOSAR Blockset 2.0. Developers can thus validate real-time-specific parts of the software behavior, and evaluate and test the V-ECU under realistic operating conditions, even in the vehicle. Functional and non-functional characteristics can therefore be thoroughly validated and a high test coverage can be achieved.

Conclusion

The new RTI AUTOSAR Blockset 2.0 makes MicroAutoBox II a universal, real-time-capable AUTOSAR development system for production-oriented, model-based development and testing. Its possibilities and productivity by far surpass those of ECU-based development processes.

And Without AUTOSAR?

What if your ECU software is not based on AUTOSAR? You can also set up seamless tool chains and processes for non-AUTOSAR software. For more information please contact our experts by e-mail to *rcp@dspace.de*.

MicroAutoBox II: Universal, real-time-capable development system with the extensions Embedded PC (center) and the RapidPro SC Unit, which provides signal conditioning (right).

Published at the end of 2011, the DO-178C standard mainly differs from its predecessor DO-178B in that it has standard supplements to provide greater scope for using new software development methods. The most important supplements are those on methods for model-based design and model-based verification, which are described in supplement DO-331. These key software design techniques offer great potential for achieving highly efficient software development in the aerospace sector while not only maintaining the high quality and safety requirements for software but actually improving them. This article shows how to use TargetLink together with DO-178C and DO-331 and which aspects to take into account.

Models: Opening the Door to Innovative Methods

The ability to represent requirements by models in accordance with DO-331 is a decisive advance in efficient and quality-oriented software development. The shift from purely textual requirements to formalized requirements expressed as models opens up many new options for automated analysis, source code generation, and verification. There are two different types of software requirements according to DO-178B/C and DO-331:

- *High-level requirements (HLR)*
  These describe what the software has to do, but not how it has to do it (i.e., the software is treated as a black box). HLRs are derived from the requirements for the actual system that are defined in the system process, e.g., according to

>>

Code generator TargetLink for aerospace applications

Safe Code According to DO-178C

dSPACE's production code generator, TargetLink, is suitable not only for automotive production projects, but also for projects in civil and military aviation. dSPACE offers a comprehensive workflow description particularly for using TargetLink in DO-178-compliant aerospace projects. It describes how to use a TargetLink-based tool chain to make software certification easier.

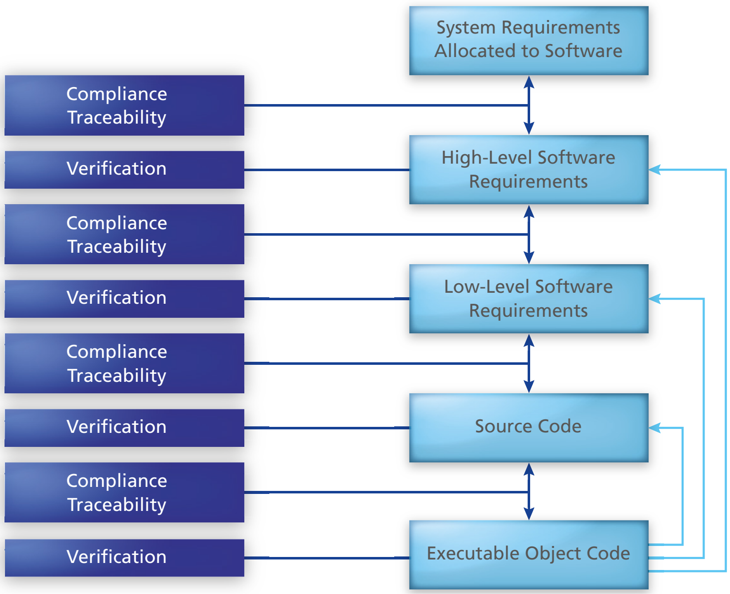


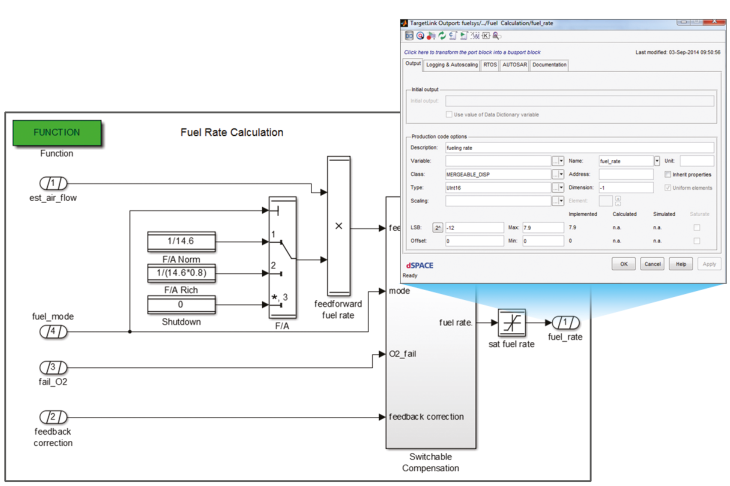
Figure 1: Important development phases according to DO-178C, including the necessary verification steps.

ARP4754 (Aerospace Recommended Practice).

- *Low-level requirements (LLR)*
  These describe the internal workings of the software (viewed as a white box), i.e., how it has to do what it does. LLRs are naturally derived from the HLRs. It must be possible to generate the actual source code directly from the LLRs.

Models can now be used to represent requirements on these two levels (figure 1). Simulink®/TargetLink® models are used particularly often for representing the LLRs from which the actual source code is then generated by means of automated code generation. According to DO-331, such models representing LLRs are called design models. They contain the description of the actual functionality and also all necessary detailed information on the software, such as internal data structures, control flow information, and potential fixed-point representations (figure 2).

Figure 2: Simulink/TargetLink design models are used for direct automatic source code generation with TargetLink.



From Design Model to Source Code at a Click

Design models representing requirements (according to DO-331) offer a direct route to creating the software source code – by using automatic code generation instead of manual coding. In terms of quality and reliability, TargetLink by far surpasses human programmers and produces source code deterministically at the click of a button:

- The source code generated by TargetLink is very readable and suitable for reviews. This is ensured by extensive source code commenting and easy-to-understand symbol names, and by using a subset of the C language.
- The code can be traced straight back to the design model. This provides direct traceability between the source code and the associated model from which it was generated.
- The code to be generated with TargetLink is also highly configurable in order to fulfill the coding guidelines, combine the code generated with TargetLink with existing legacy code, and optimally integrate the generated code into the software architecture.

In general, the quality, configurability, and efficiency of the generated code are outstanding TargetLink features that are visible in all application areas.

Model-Based Verification: the Key to Easier Certification

The great advantages of using models to specify requirements (HLRs and LLRs) are evident not only in automatic production code generation but also in other areas such as verification steps. These have to be performed in parallel to the development process to test the resulting artifacts, such as models, source code, and object code, in each individual development step (figure 1). To prove that the models fulfill the requirements they were derived from (figure 1), a combination of model simulation, coverage analysis, and test case generation can be used. According to DO-178B/C, test cases must be created solely on the basis of requirements. If a requirement is itself expressed as a model, e.g., a Simulink/TargetLink model, techniques for automatic test vector generation such as those provided by BTC EmbeddedTester® can be used. A typical way of verifying that the executable object code is consistent with the HLRs and LLRs (figure 1) is to execute it on the target platform. TargetLink provides extremely powerful mechanisms for this in the form of processor-in-the-loop simulation, in which the automatically generated code is translated directly by the target compiler and executed on an evaluation board with the target processor (figure 3).

DO-178C/DO-331 Workflow Document for TargetLink

dSPACE provides the workflow document "TargetLink – Model-Based Development and Verification of Airborne Software" for using TargetLink in DO-178C/DO-331-compliant projects. The document describes how to meet the individual requirements, called objectives, of DO-178C/DO-331. It focuses not just on TagetLink itself but also the entire TargetLink ecosystem consisting of a complete model-based tool chain that can contain third-party tools. These include tools from TargetLink cooperation partners, such as BTC Embedded Systems, Model Engineering Solutions, and AbsInt.The document can be requested by an e-mail to *TargetLink.Info@dspace.de*. ▪

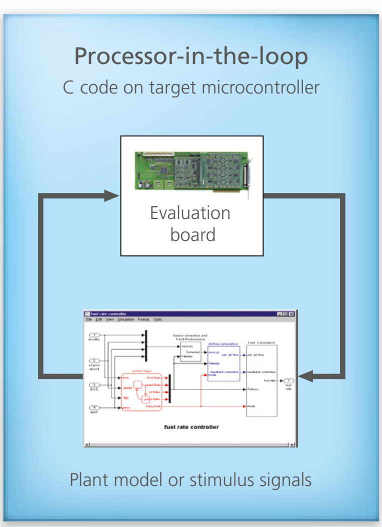
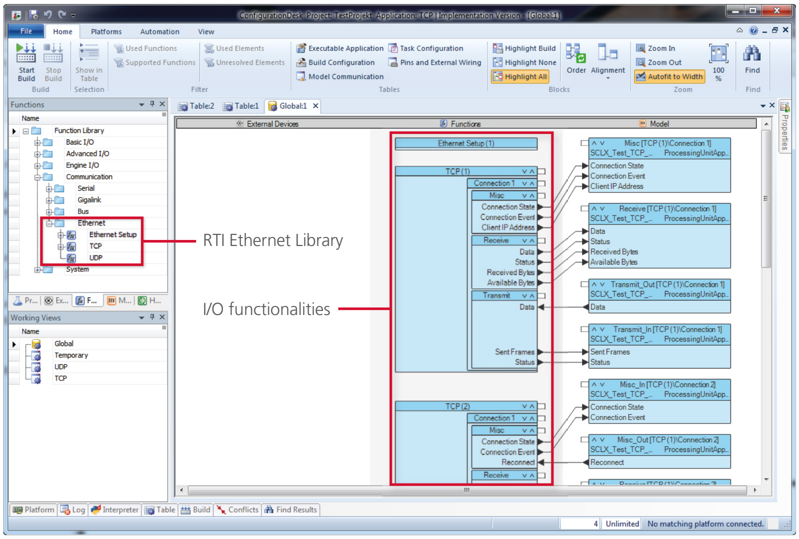


Figure 3: Running the executable object code in processor-in-the-loop simulation to verify that the object code fulfills the requirements.

Conclusion

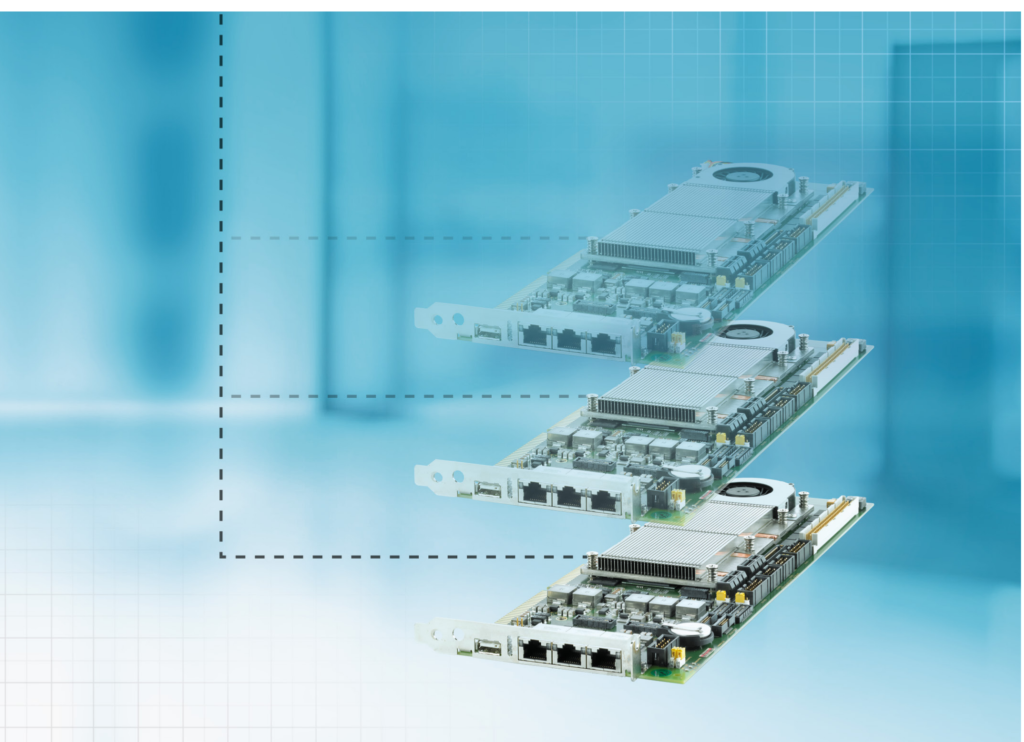
TargetLink is well suited for DO-178C-compliant aerospace projects, making it possible to generate high-quality source code at the click of a button. Due to its layout commenting and symbol names, the generated code is highly readable, provides seamless traceability to the requirements and is easy to configure, e.g., to fulfill coding guidelines. TargetLink and its integration with third-party tools offer an ideal environment for verification, simulation, analysis, and tests. From the requirements to the final source code – with TargetLink, users have their DO-178C-compliant development projects under control.

SCALEXIO Ethernet Solution



For SCALEXIO®, dSPACE offers the new SCALEXIO Ethernet Solution. It adds Ethernet-I/O functionalities to ConfigurationDesk® the configuration and implementation software, making it possible to send and receive data based on TCP/IP and UDP/IP during hardware-in-the-loop tests with SCALEXIO. The SCALEXIO Ethernet Solution can be configured completely in ConfigurationDesk. Customers can use the new solution to connect third-party systems, such as test benches and measurement systems, or they can couple co-simulation applications to perform on two simulation systems at the same time.

Multiprocessor Systems with the DS1007 PPC Processor Board



As of dSPACE Release 2015-A, the DS1007 PPC Processor Board provides multiprocessor support for modular real-time hardware. Several DS1007 boards can now be connected to increase not only the I/O range but also computing power, adjusting it to the complexity of each application. Users can couple the processor boards synchronously as usual via a piggyback (Gigalink) module that was designed for this particular task. The module connects the boards via low-latency fiber-optic cables.
The Real-Time Interface for Multiprocessor Systems (RTI-MP) makes distributing the model parts across the processor cores easy.

Training the Developers of Formula Student Race Cars

In the course of February and March, dSPACE again welcomed promising young professionals from motor racing and engineering in Paderborn. A total of 21 Formula Student developers visited dSPACE, their team sponsor, to learn how to use the dSPACE development tools for open and closed loop control of their race cars.

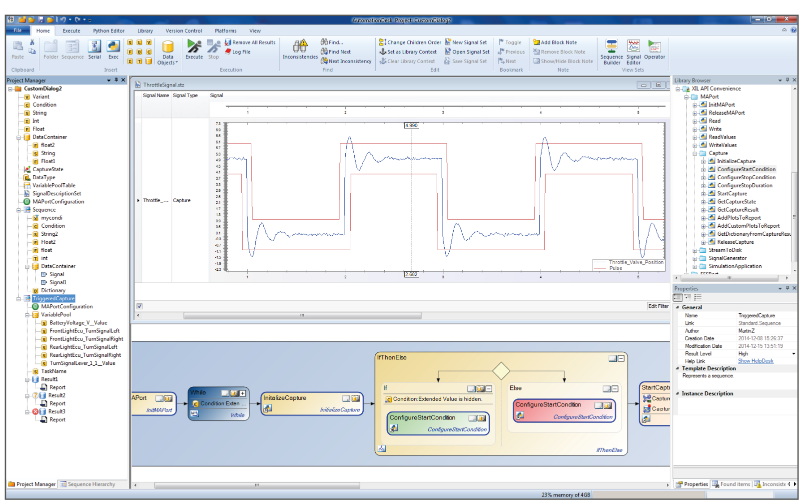
Since 2007, dSPACE has supported select Formula Student teams with development tools and technical know-how. In 2015, 22 academic racing teams benefit from this co-operation, which traditionally includes extensive training sessions. By getting involved in Formula Student, dSPACE is already thinking about tomorrow's need for skilled workers, helping young researchers to get their ideas on the road early on. ■

AutomationDesk 5.0: Optimized Test Automation and Test Development

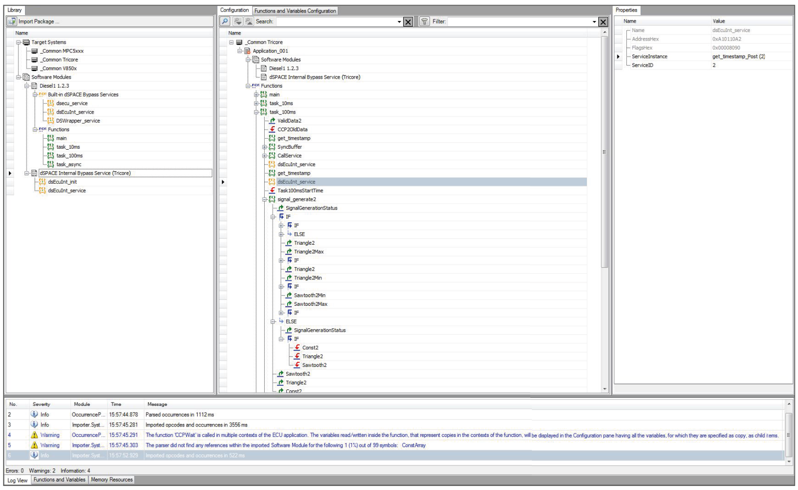
The version 5.0 of the test automation software AutomationDesk provides even better usability. The latest feature is signal-based testing, an intuitive graphical description of signals and stimuli for evaluation. This makes it easy to develop tests graphically, almost as if on paper.

AutomationDesk also complies with the ASAM XIL 2.0 standard by offering an interface to test benches that is manufacturer-independent. The high-quality implementation of the standard in AutomationDesk lets users conveniently access simulation variables and add failure potentials to ECU pins.

AutomationDesk now also contains a library for a direct connection to MotionDesk, dSPACE's visualization software for ECU testing. This makes the tool chain for systematic tests of camera-based driver assistance systems even more seamless. ■

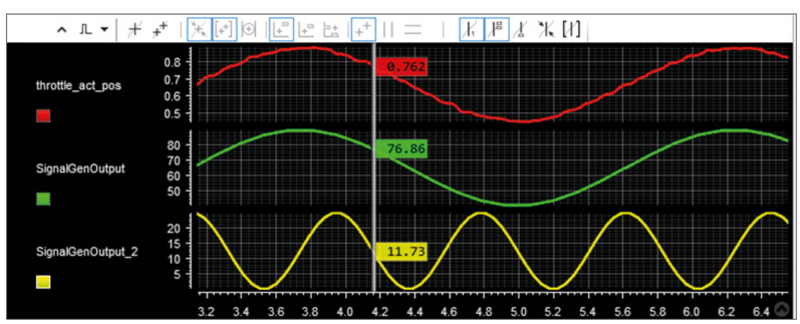


ECU Interface Manager 1.6: Support for Multicore ECUs



Version 1.6 of the dSPACE ECU Interface Manager now lets you prepare the software of powerful multicore ECUs for bypassing, while offering all the advantages of Hex-code-based bypass integration: Function developers can implement a bypass fast and largely autonomously, without having to access the build environment or the source code of the ECU software. They can then use fast iteration cycles to test new functions on production ECUs. The ECU Interface Manager supports microcontroller families for automotive applications from Infineon, Freescale, Renesas, and ST Microelectronics. During bypass integration, the ECU Interface Manager automatically ensures data consistency even across core boundaries, keeping the complexity of multicore software away from the user.

ControlDesk 5.4: More Hardware Support, More Convenient Handling



ControlDesk Next Generation 5.4, the latest version of dSPACE's experiment and visualization software software, opens up many enhanced possibilities for new projects. The software now also supports multiprocessor systems based on the DS1007 PPC Processor Boards and Vector's VN5610 interface for CAN bus integration. ControlDesk 5.4 also offers more recording features: You can now record maps, curves, and shared axes in the ASAM MDF 4.1 format. The Signal Editor lets you import and export signal descriptions and signal generators according to ASAM AE XIL API 2.0.1. A new Data File Segment makes playing recorded measurement data more flexible by letting you choose a start time and a playback duration. It has also become easier to use simulation systems consisting of ECUs and plant models for virtual validation, a field that is constantly growing in importance. You can use the XIL API MAPort Platform Module to connect ControlDesk to third-party simulation platforms according to the ASAM XIL API standard to read, write, and measure simulation variables on these platforms.

dSPACE on Board

Discover intriguing and innovative applications, achieved with dSPACE development tools.

Eyes for the Car

AdasWorks is a software tool for developing driver assistance systems and autonomous vehicles. When integrated in a vehicle with a front camera, AdasWorks uses the camera data to navigate the car completely autonomously. A dSPACE MicroAutoBox calculates the digital output signals for the electromechanical actuators, effectively steering, decelerating, and accelerating the vehicle.

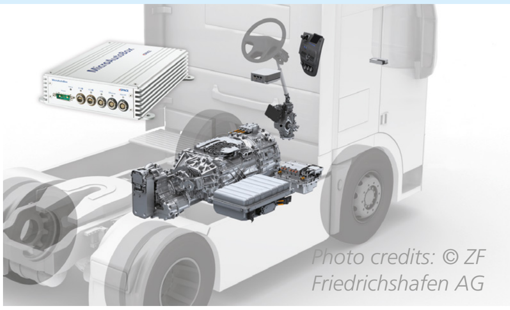
AdasWorks and ThyssenKrupp Presta Hungary demonstrate the autonomous vehicle.
www.dspace.com/go/dMag_20152_ADASW

The following media report explains the application and the use of the MicroAutoBox:
www.dspace.com/go/dMag_20152_VISYS

Maneuvering by App

ZF's new maneuvering and coupling assistant turns maneuvering trucks with several trailers into a finger exercise. A tablet app lets you maneuver the truck to the desired position – easily and precisely. Two dSPACE MicroAutoBoxes in the truck convert the maneuver commands into control signals for the engine, brakes, and steering.

Precisely maneuvering a 25 m truck at the touch of a finger.
www.dspace.com/go/dMag_20152_ZF



Among other systems, the ZF Innovation Truck makes use of two dSPACE MicroAutoBoxes.

Drift Against the Computer

Is a computer a better driver than a person? An automobile magazine's test team sets out to find the answer on the race track. Their challenger: the prototype of a BMW M235i that has all the electronic components needed for controlled drifting in a circle without any driver intervention. A dSPACE AutoBox in the trunk is part of the computer system.

The autonomous BMW M235i drifting in a circle.
www.dspace.com/go/dMag_20152_ABTV

A dSPACE AutoBox contributes to the controlled drifting experience.

Learn more about these applications online, via videos, photos, and reports:
www.dspace.com/go/dMag_20152_REF_E