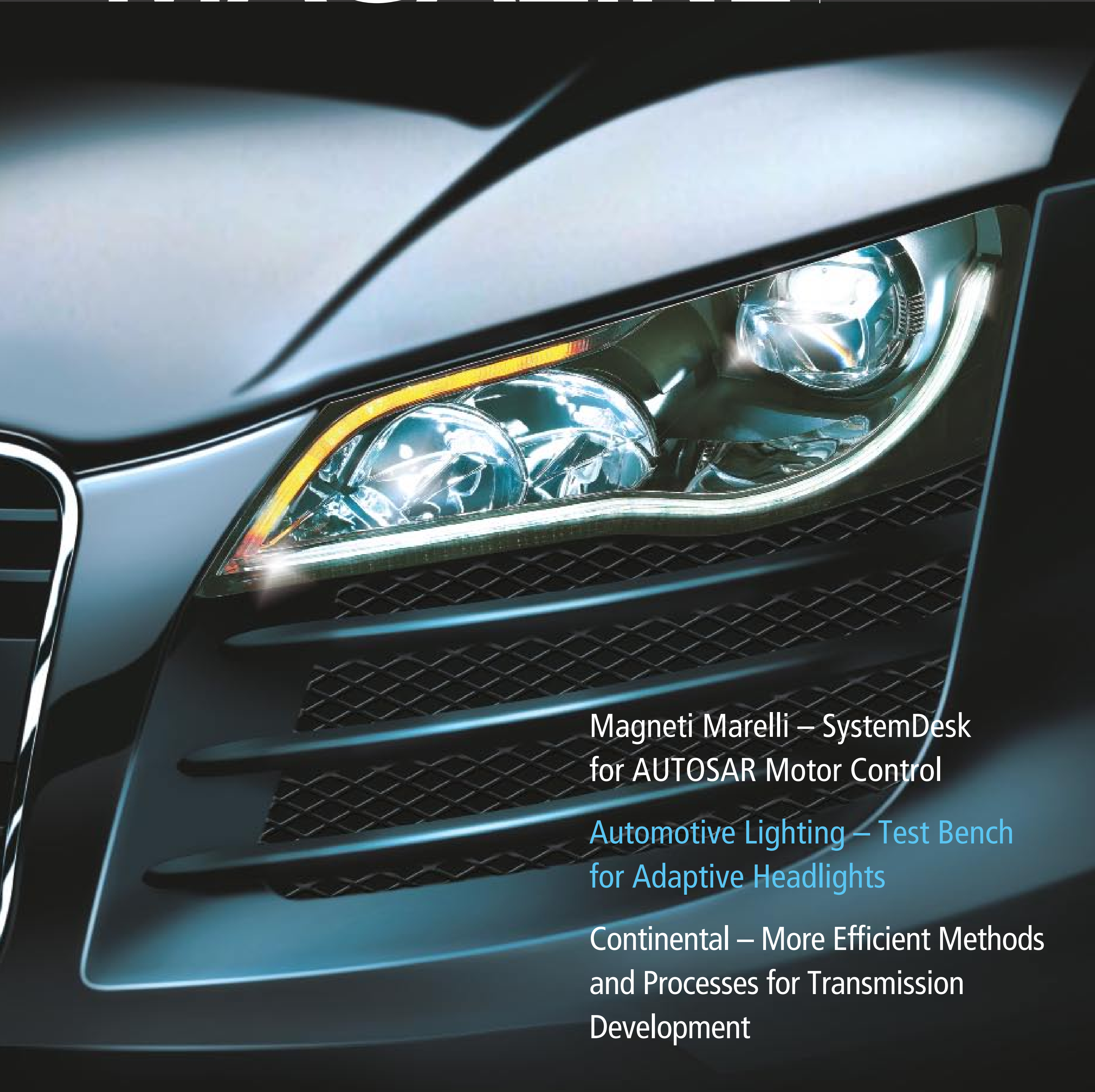
www.dspace.com

dSPACE MAGAZINE

2/2008



Magneti Marelli – SystemDesk for AUTOSAR Motor Control

Automotive Lighting – Test Bench for Adaptive Headlights

Continental – More Efficient Methods and Processes for Transmission Development

The more efficient Path

In the world of software development for embedded systems, high pressure from competition calls for a permanent increase in the efficiency of the methods, processes and tools. In many cases, the decisive lead is won by model-based development and automatic production code generation.

Our practical example from the development department at Continental AG illustrates how this novel method is being used: in the initial setup of the tool chains and processes, then the first production projects and, finally, in the software development for an innovative double clutch transmission.

TargetLink, the production code generator from dSPACE, is easy to integrate into existing development processes at Continental. It guarantees a seamless process from the initial idea to the final implementation of the generated code on the electronic control unit. Executable specifications are available as models. They are easy to comprehend, can be understood by all teams involved, and can be checked by simulation right from the very start.

The final production code for the target processors is generated by just a click of a button, so to speak – yet fulfils the highest possible requirements regarding execution time, memory usage and software quality. Simply the fact that error-prone handcoding is redundant, achieves the necessary software quality. The result speaks for itself: In direct comparison to commonly used methods, there is a definite quality improvement within a noticeably reduced development time.

Read more on page 12.

Dr. Herbert Hanselmann
President

There was a time when dSPACE was viewed almost exclusively as a hardware company. Even now, I get the impression that hardware, being more tangible and easier to grasp, is still what mainly shapes our image. Beyond a doubt, sitting in front of a screen with colored images is just not the same as standing in front of a 10-cabinet HIL simulator. And reliable hardware simply seems to command more respect than software.

But as a look at our development investment shows, the biggest piece of the pie goes to software. The situation was no different all those years ago when we founded the company: out of 4 founding members, only one had a purely hardware background.

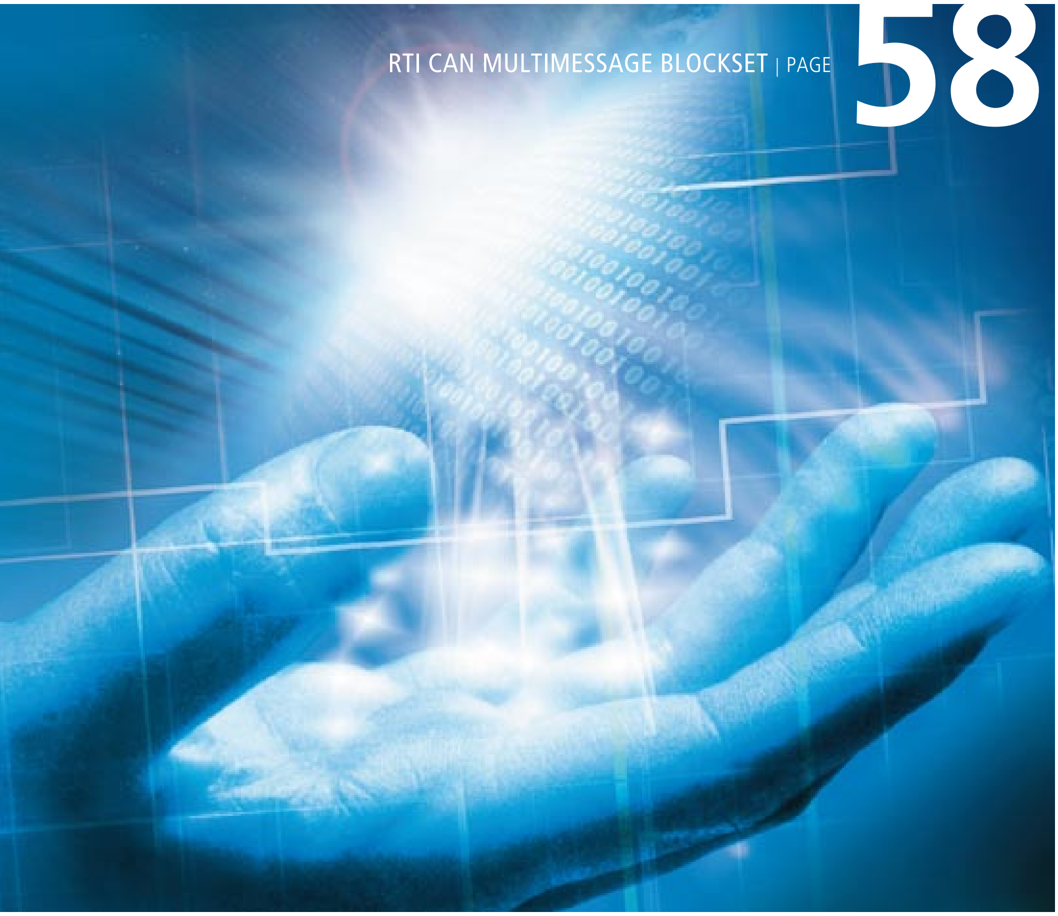
The enormous investments we have made on the software side were, and are, vital to our success. Professional customers do not need just a piece of hardware, they need a solution for tasks and problems. And this solution cannot exist without sophisticated, usable, application-related software to get the boards and boxes running efficiently.

As many may have noticed, we now also provide software tools that do not need dSPACE hardware. In effect, because they are useful extensions to our product portfolio. This all began with TargetLink in 1999, and continued with SystemDesk in 2007. TargetLink is a success story both technically and commercially – and we expect the same of SystemDesk.

Other products that are independent of dSPACE hardware are the Automotive Simulation Models (ASM). Even though they were mainly used with dSPACE simulators so far, they do not have to be. They are also being used in purely offline simulation, after which they can easily be reused in HIL simulation. This provides another major benefit: an integrated development process. Reports say that the ASMs match or even outdo the best models on the market in terms of quality and usability. So this looks like the beginning of a success story for yet another dSPACE software product. Nearly 40 satisfied customers have already opted to use the ASM engine models, and the vehicle dynamics model is also in productive use in more than 30 companies. Even though we now make hardware-independent software tools, the fact remains that we are also a hardware company and aim to stay one step ahead in that field too. Performance, latencies, reliability, in-vehicle suitability, packaging, special I/O features – our developments score high here and elsewhere. The proportion of software may increase, but hardware will always be one of dSPACE's trademark features.

Dr. Herbert Hanselmann
President

Embedded Success dSPACE



IMPRINT

dSPACE MAGAZINE is published periodically by:

dSPACE GmbH · Technologiepark 25
33100 Paderborn · Germany
Tel.: +49 52 51 16 38-0
Fax: +49 52 51 6 65 29
dspace-magazine@dspace.com
www.dspace.com

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

Technical Authors: Ralf Lieberwirth, Sonja Lillwitz, Julia Reinbach, Dr. Gerhard Reiß
Co-Workers on This Issue: Dr. Ulrich Eisemann, Björn Müller, Karsten Theissen

Editors and Translators:
Robert Bevington, Stefanie Bock,
Dr. Michelle Kloppenburg, Christine Smith

Design: Krall & Partner, Düsseldorf
Media Design (Layout) dSPACE: Tanja Raeisi
Print: Merkur Druck, Gütersloh, Germany

Contents

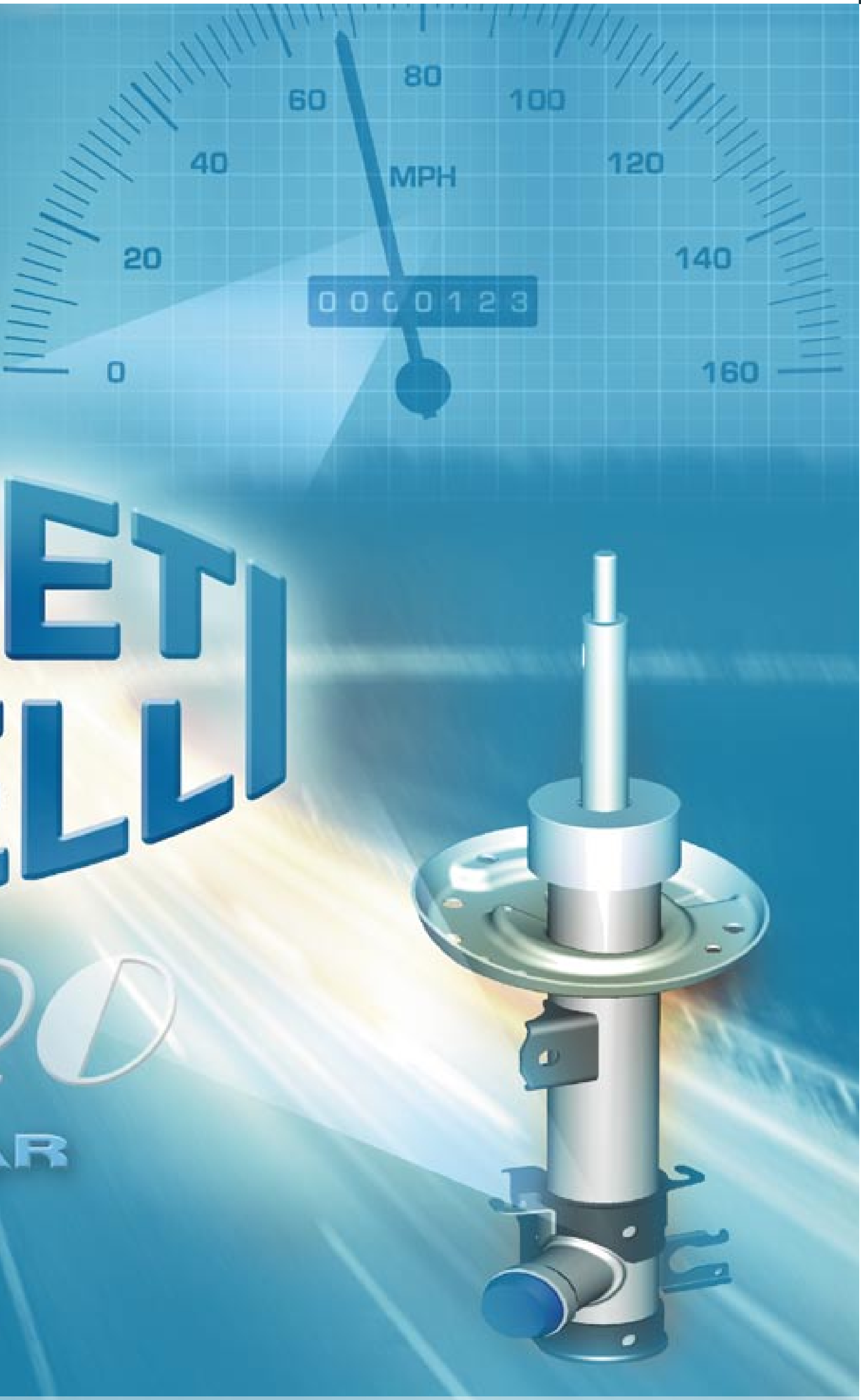
Customer

Products

Engine Management AUTOSAR Way

the

Magneti Marelli migrates ECU software to the AUTOSAR standard



AUTOSAR is a hot topic in the development departments of automotive suppliers and OEMs. The standard is frequently accused of having too much overhead in implementing complex systems. In an engine ECU project, Magneti Marelli has set out to prove otherwise.

„Magneti Marelli AUTOSAR Cross-Project X-PRO“ is a special project in which we are developing a purpose-built demonstrator to verify the feasibility of AUTOSAR-compliant software architectures – even for complex engine management systems. In concrete terms, our task is to migrate the entire software of an existing engine ECU to AUTOSAR and then reimplement it on that ECU. Engine control software is usually enormous in size, making it vital to be able to reuse modules simply. That is reason enough for our interest in software development the AUTOSAR way. Working together with dSPACE, we chose an approach in which the new automation feature of SystemDesk, the system design software, plays a major part. Because this demonstrator project is concerned with feasibility and with obtaining information on performance and resource consumption, we did not design a new architecture top-down, but largely retained the existing ECU software design. To do so, from the existing ECU data we extracted all the information needed for reconstructing the software architecture and scheduling, and transferred it to SystemDesk by means of a script (fig. 1). This procedure quasi-automatically produces an AUTOSAR-compliant architecture that requires only a few manual adjustments in SystemDesk.

Preparing Data and Information

Before AUTOSAR, there was no standard available for describing all the implementation information, so it is not possible to reuse the individual software modules systematically. Thus, our first task was to collect data on the ECU’s configuration, parameterization and implementation from a diversity of data sources and bring it together at a

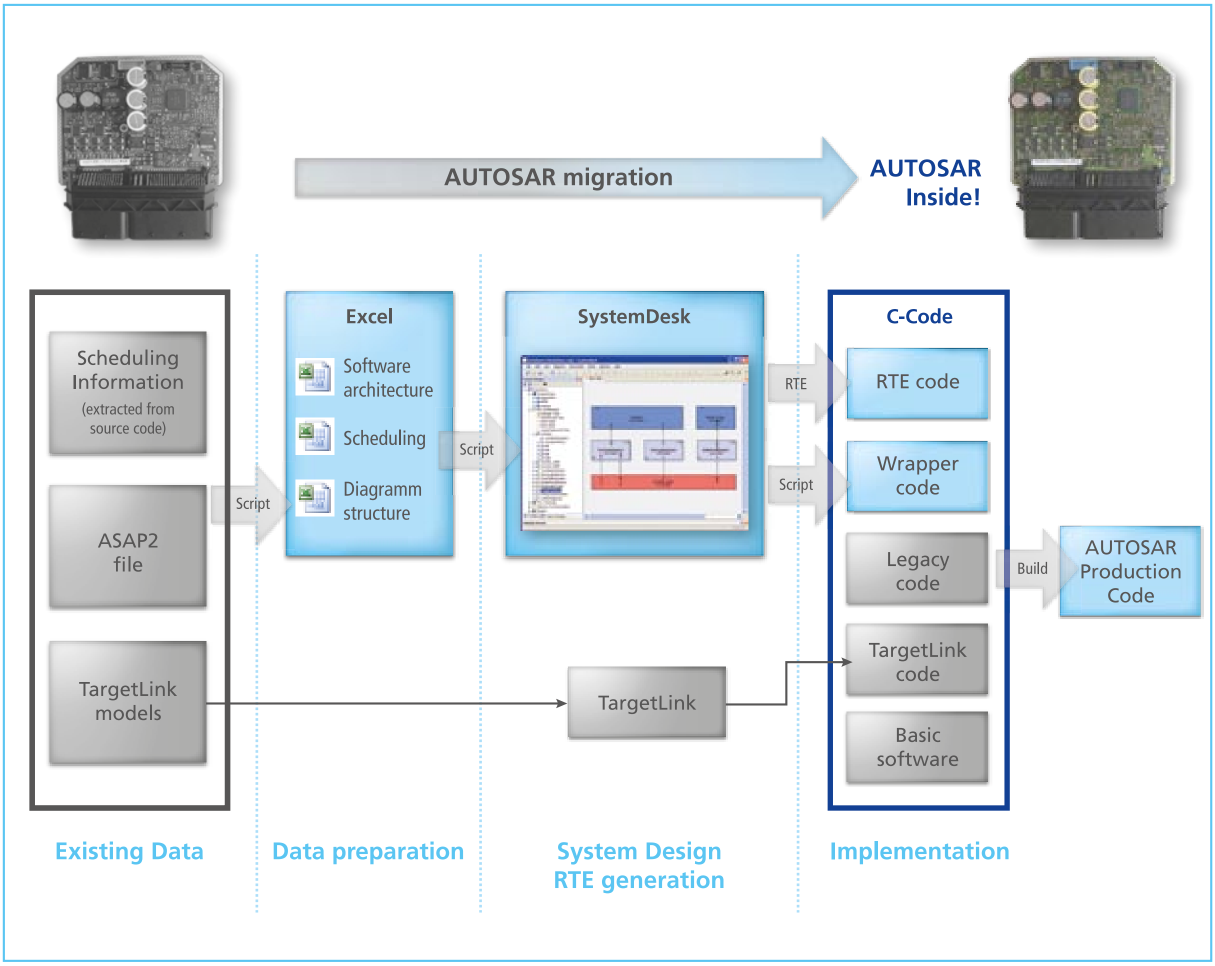


Fig. 1: Overall workflow for AUTOSAR migration.

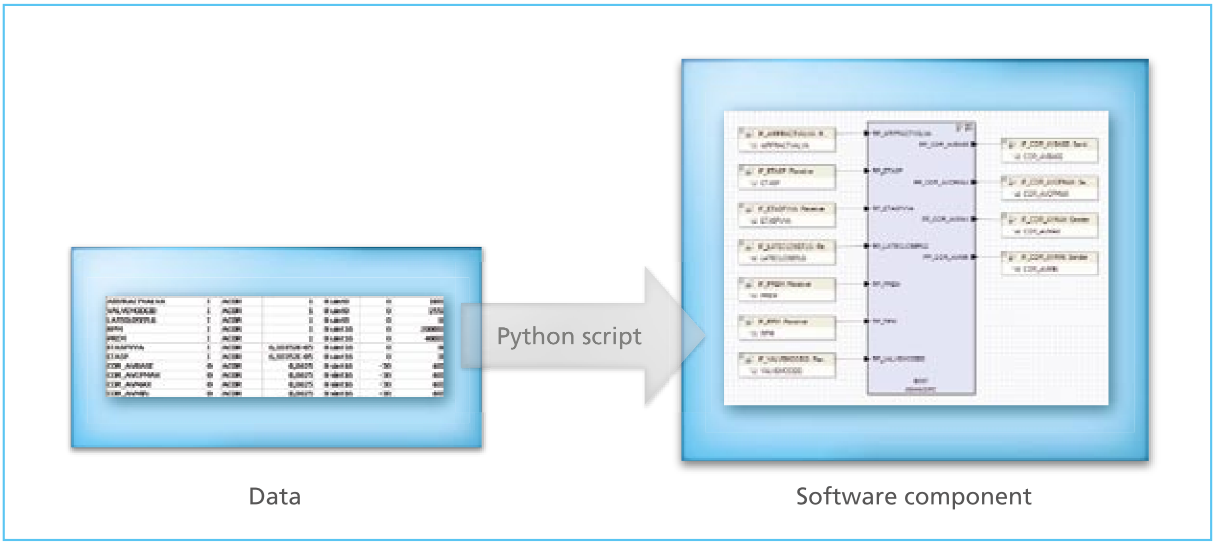


Fig. 2: All the variables of a software component (SWC) in Excel on the left, with corresponding SWCs in SystemDesk on the right.

> "Variables are accessed via the SystemDesk-generated RTE with the same performance as in classic implementation."
>
> *Luigi Romagnoli, Magneti Marelli*

central point for further processing. For example, we used a script to extract the architecture information from existing TargetLink function models. Description files such as the ASAP2 file were also used to obtain the data types, min/max values and scalings of the implemented variables. Since the configuration of the old ECU's operating system also had to be reused, the entire function schedule was extracted from the old source files by a script and transferred to Excel. Standards like AUTOSAR will obviously make it much easier to reutilize existing software modules and implementation data in the future.

Building the Software Architecture

Data from an Excel sheet now had to be migrated to the system world to produce an AUTOSAR-compliant system architecture. Via SystemDesk's automation interface, a Python script was used to create all the required software components fully automatically, including sender/receiver interfaces and data elements (fig. 2). With an import volume of over 20,000 elements, the occasional error is obviously unavoidable. However, SystemDesk is able to detect and display all the inconsistencies in the input data. For example, if a data type did not match the specified scaling information and min/max values, this error was indicated while the software architecture was still being created.

To import the variables, we used a naming scheme that allowed all the connections in the software architecture to be built automatically. With more than 170 software components and several thousand input and output signals, this saves an inestimable amount of time. With such enormous quantities of data, it hardly makes sense to represent the entire architecture in a single diagram. SystemDesk therefore allows subfunctionalities to be brought together in composition diagrams, each providing a different view of the software architecture. This means that even in large-scale projects everyone has a clear view of what they are doing, and the relevant information is always available for discussing aspects that concern several teams.

Importing the OS Schedule

The old operating system schedule also had to be used for the AUTOSAR ECU. First an Excel sheet was used to assign runnables to software components, and then all that was left to do was to specify the execution sequence within the operating system tasks. After that the schedule was transferred to SystemDesk completely automatically.

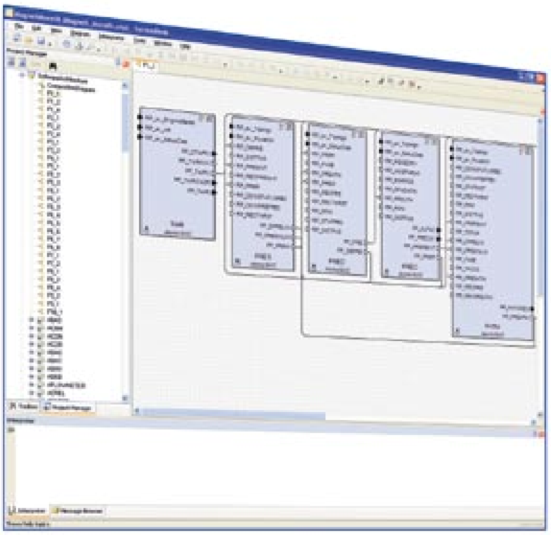


Fig. 3: Subset of the overall model shown in an individual composition diagram.

> “SystemDesk’s automation feature not only speeds up processes, it also ensures data integrity.”
>
> *Alessandro Palma, Magneti Marelli*

Reducing Graphical Complexity

Engine ECUs are among the most complex ECUs in a vehicle. Around 10,000 variables, 600 runnables, and 16 tasks have to be defined and optimally managed. Any means of reducing complexity helps avoid errors and speeds up the development process. Composition diagrams showing individual aspects of the software, as described above, are not the only way in which SystemDesk reduces complexity. There are numerous other methods (see also Fig. 3). For example, it takes only three clicks to hide all the (connected) ports of a software component or to display a dialog showing all the as yet unconnected ports of an entire software architecture. What we particularly liked about working with SystemDesk’s graphical interface was the ability to represent complex systems in a simple, reduced form. This is the only way to run successful reviews and explain new ideas quickly on screen.

RTE Generation

In migrating legacy code to the AUTOSAR standard, an important intermediate goal is to perform RTE generation for the entire software architecture, including the OS schedule. The run-time environment (RTE) is slim middleware that embodies the communication between AUTOSAR software components in software form. In some circumstances, this intermediate layer might require extra resources compared with our previous software implementation. It was therefore important for us to estimate the memory and execution time requirement of the RTE code generated by SystemDesk. To obtain useful information on this, no AUTOSAR-compliant application code or basic software is needed at this point in time.
The analysis of the RTE code generated by SystemDesk revealed that memory consumption was virtually negligible. Access to the variables was implemented almost completely by define statements, so there was hardly any impairment to performance as compared to classic implementation.

Integrating the Application Code

Having established that RTE generation delivers sufficiently efficient code, we are currently working on developing the AUTOSAR application layer. Not only the handcoded legacy code will be reused for this, but also the entire C code that was formerly generated directly from TargetLink models. To make the code AUTOSAR-compliant, the approach we are using in this project phase is to embed both legacy and TargetLink code in AUTOSAR-compliant wrappers, using the SystemDesk automation feature, among others. Another reason for choosing this procedure is that the legacy code and the TargetLink code both have the same interface, so the same kind of

> "The numerous options provided for graphically reducing complexity make working with large-scale system models easier and faster."
>
> *Luigi Romagnoli, Magneti Marelli*

wrapper can be used. However, in the future we plan to generate both the AUTOSAR-compliant code and the software component descriptions directly from TargetLink models using TargetLink's special AUTOSAR support. We expect that this will improve resource consumption by eliminating the need for AUTOSAR-compliant wrappers, and also simplify the workflow in the interaction with SystemDesk, making the overall development process more efficient.

Conclusions and Outlook

The development tasks we have performed show that the specifications and description files for an existing engine ECU can be migrated to AUTOSAR with a reasonable amount of effort. With support from automatable tools, even complex systems and large data volumes can be handled reliably.

Initial performance measurements show that the AUTOSAR standard does not necessarily lead to increased execution time and memory consumption if the development tools are designed for the best possible efficiency. Work on the AUTOSAR-compliant architecture and the wrapper software has been completed. We are now in the implementation phase and are integrating the existing application code and the operating system. We will be presenting the completely migrated ECU in the last quarter of the year.

Alessandro Palma
Luigi Romagnoli
Walter Nesci
Manager AUTOSAR Cross-Project X-PRO
walter.nesci@magnetimarelli.com
Magneti Marelli
Italy

Glossary

ASAP2 – An ASAP2 description file containing all the information on the relevant data objects in the ECU, such as characteristics like parameters, maps, and look-up tables.

Python – Script language defined for maximum simplicity and usability.

Runnables – An executable element in an AUTOSAR SWC, comparable to a function.

Runtime Environment (RTE) – Intermediate layer that connects the software components in the application code and the basic software in AUTOSAR designs.

Scheduling – Definition of the timing of process and task execution.

Sender/receiver interfaces – Data interfaces for AUTOSAR SWCs.

Software component (SWC) – Structural element of AUTOSAR used for creating reusable software modules.

Task – A process that runs in a system.

Current Model Data

- **Software components: 172**
- **Data elements: 2650**
- **Data accesses: about 5000**
- **Tasks: 16**
- **Runnables: 624**

Summary

- An existing ECU successfully migrated to AUTOSAR
- Efficient RTE code straight from SystemDesk
- The complexity of engine management models can be handled with SystemDesk

Shift Gears Quickly

Processes and methods for using TargetLink in model-based development of transmission function software

The goal is to noticeably reduce the development time of prototypes and end products. To meet this, innovative methods for developing embedded software are necessary. That is why years ago, Continental AG decided to introduce model-based development and production code generation for developing transmission electronic control units (ECUs). This article shows the big picture, from working out the processes and methods, to the successful use of the code generator TargetLink in two production projects.

During the past decade, enormous demands were placed on the development of embedded software. On top of the traditional process of manual coding and software tests, more stringent requirements regarding safety standards and development methods have now entered the scene. The continually increasing complexity makes these requirements even more demanding, especially with regard to the control of the new transmission generation with double clutch transmission (DCT) and continuously variable

The double-clutch transmission combines the energy efficiency of a manual transmission with the comfort of torque converter transmission.

transmission (CVT). In addition, automotive industry suppliers are constantly looking for new ways to shorten development times – as the market demands. As a result, the Business Unit Transmission at Continental AG performed a study a few years ago, which analyzed the application of model-based design and production code generation. The potential productivity increase resulting from this innovative, model-based method was initially assessed by using the method as an internal project. At the same time, the process, methods and tools for model-based design, including automatic code generation, were defined and adapted to the specific needs of the Business Unit Transmission.

After being used successfully in the first production project, the entire application software of a double clutch transimission was developed with this tool chain. This article describes the individual development steps and experience gained from the initial tests, the first production projects, and the double clutch transmission, which was autocoded entirely with TargetLink.

Designing the Processes and Methods

As part of an internal project, first the necessary process steps were drawn up, in order to start off the model-based design and automatic production code generation. The following requirements were placed on the model-based design to cover the complete V-cycle for the function and software development, with regard to the interaction between the OEMs, transmission and electronics suppliers (see fig. 1):

- Model the physical requirements of the ECU
- Simulate the actual ECU behavior (tasks, execution order, operating system)
- Support distributed development by setting up a multi-user environment
- Verify, validate and archive the models, code and scripts by adapting them to the traditional process

First, the model-based development method was applied to functionalities in demo vehicles. The preferred solution that arose was to use established tools based on the latest technology. Thus, MATLAB®/Simulink® were selected for the function design. The success of the project depended greatly on whether the demonstration of new functionalities in demo vehicles could be completed within just a short time frame. The simulation of concepts, and the subsequent test and verification were performed by model-in-the-loop (MIL) simulation on the PC. After successful verification, the function was ready for tests in the demo vehicle and on the test bench with the rapid control prototyping (RCP) environment. For a fast validation and fine-tuning of the functions, Real-Time Workshop® and dSPACE AutoBox/MicroAutoBox were used. Front-loaded tests made it possible to assess the customer's requirements quite early. After model-based development was used successfully in the rapid prototyping phase, the V-cycle phase was to switch from manual C coding to automated production code generation. dSPACE TargetLink was selected as the code generator for this step.

> "For the project performed with TargetLink, the time-to-market was even shortened, which our customer applauded."
>
> *Georg Grassl, Continental AG*

To prove the feasibility of automatic, optimized code generation for a 16-bit microcontroller, a position control algorithm for a transfer case transmission was used. First, the already existing function was transferred into a model-based design and then expanded with implementation information for production code generation. Then the automatically generated code was integrated and validated on the production ECU. To prove the efficiency of the generated code, its resource consumption was compared with that of manually coded software already used in series production.
Finally, on the basis of these preliminary tests, the important process elements for generating production-intent software with the new development method were identified. The following section illustrates these process elements in closer detail, including a few supportive methods:

Model-Based Function Design

By simulating Float models it is possible to quickly evaluate whether the functional requirements can be met. This step is then followed by rapid prototyping to validate the functional requirements in a demo vehicle. It is necessary to define the function architecture at an early state, together with the resulting generation of model libraries, so that partitioned development can be achieved and the configuration elements created. In addition, the modeling guidelines for the function and software also contribute towards making the compatible, model-based design possible. Industrial standards regarding adherence to in-house guidelines are evaluated and, if necessary, project- and application-specific agreements are added.

Model-Based Software Design

Model-based design templates to generate optimized target code are introduced and must already be adhered to during the function design stage. This is in close connection to the standards established by MISRA (Motor Industry Software Reliability Association) and the tool manufacturers: for example, the MISRA AC TL guidelines. Designing software means scaling, defining block properties, adapting the configuration of the code generator and thereby supporting the automatic code generation. The so-called implementation model is made when the implementation information (memory assignment, etc.) is added.

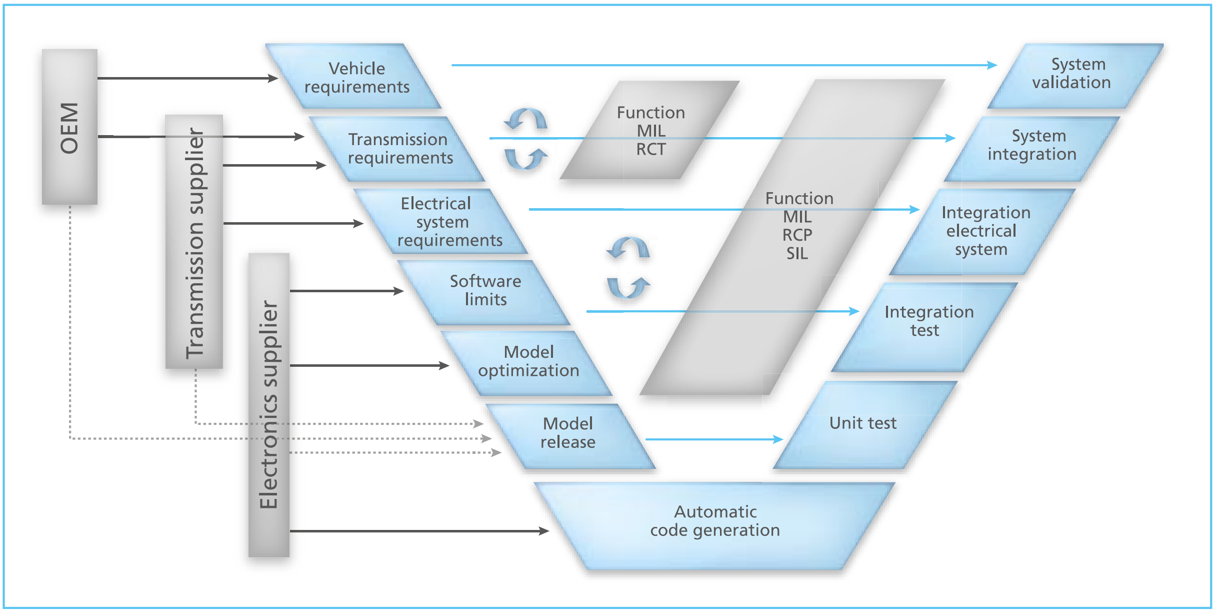


Fig. 1: The development cycle in the Business Unit Transmission.

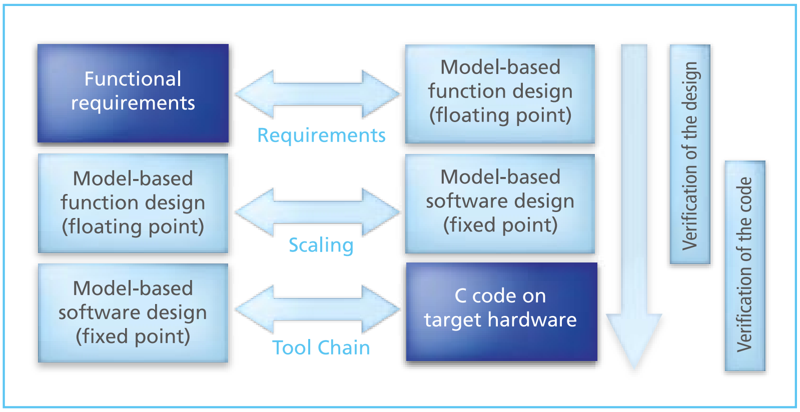

Functional requirements
Requirements
Model-based function design (floating point)
Model-based function design (floating point)
Scaling
Model-based software design (fixed point)
Model-based software design (fixed point)
Tool Chain
C code on target hardware
Verification of the design
Verification of the code

Fig. 2: Model-based testing – design and code verification.

At this the point the software architecture needs to be defined early, in congruence with the function architecture. The result is the partitioning into C modules and C functions, which enables the testability, configuration (archiving) and maintenance.

Model-Based Collaboration

Tight collaboration between all involved parties at the model level is necessary to ensure that the models are designed and exchanged consistently. Complex functionalities are partitioned into smaller units and placed in model libraries. Documentation that is based on the models replaces the traditional, hand-written software requirements specifications.

> "Within the framework of an in-house study by the OEM, the result was that the handwritten code cannot measure up to the code generated by TargetLink in terms of freedom from error."
>
> *Georg Grassl, Continental AG*

Model-Based Testing

To optimize the process, especially via automation, model-based testing and the corresponding quality measures are essential. The standard software quality measures are modified. This makes quick, micro-V-cycles possible, resulting in fast progress towards model maturity. Model-based testing offers the possibility of a step-by-step approach (see fig. 2).

First Production Application of the New Process and Methods

Based on the results of the preliminary studies mentioned above, the first high-volume project was started with a defined, well-coordinated cooperation between the partners in all areas concerning model-based design and automatic code generation. The modeled algorithm – a subfunctionality of the application software – is a gear-shift strategy for a six-stage automatic transmission (see fig. 3) that should be used for a 16-bit platform. Due to their success in the preliminary studies, MATLAB/ Simulink and dSPACE TargetLink were used in accordance with the previously developed processes and methods.
The test of the function software was performed according to conventional quality assurance procedures, which were already defined for the written C-code and adjusted to fit the conditions of automatic code generation.
To ensure compatibility with in-house code rules, a static code analysis was performed. At model level, structural coverage tests were performed and supported by tools for test vector generation. Thus, the quality goal of reaching 100% coverage of the fixed-point code (highest test coverage level: modified condition/decision coverage, MC/DC) was achieved. These module tests were performed with a standard tool that can simulate the microcontroller platform. Thus, the complete tool chain (code generator, compiler and linker) was verified successfully.

Results and Insights from the First Production Projects

The method we chose ensures that the tasks were performed at the required time, thereby supporting the stringent time frame. As a result, the time-to-market for this project was even shortened, which our customer applauded. Furthermore, the developed transmission software had to adhere to tight restrictions regarding resource consumption:

- a maximum CPU utilization of 15%

- a maximum ROM assignment of 100 kByte

With TargetLink, the values stayed far below these levels, whereas the size of the automatically generated function software was:

- maximum 10 % CPU assignment
- 60 kByte compiled code
- 50 modules
- 20 % ROM assignment of the entire application software

The electronic control unit for double clutch transmission has to pass specific qualification tests in the Continental Automotive systems lab.

It became clear that the software freeze of the applied code generator version had to be performed, at the latest, when the software quality measures and code verification began. Any update to the code generator can lead to altered source code, which would make it necessary to repeat all software tests and quality measures.
Within the framework of an in-house study by the OEM, the result during validation was that the handwritten code cannot measure up to the code generated by TargetLink in terms of freedom from error. By using model-based development and automatic code generation we were able to make an exceptionally high number of software deliveries precisely on schedule, thanks to the structured process for this new method. The method was sent to other users at different locations to be used as the standard method for transmission projects.

Development of the Function Software for Dual Clutch Transmission

In 2006, due to the positive experience when using the new methods and processes, we started the production development of hardware and software for a DCT transmission ECU (see fig. 4). The entire function software for this was created by using model-based development and automatic code generation with TargetLink.
The tasks were divided as follows:

- The OEM makes the function requirements.
- Continental makes the model-based function design available, validated by using rapid control prototyping.

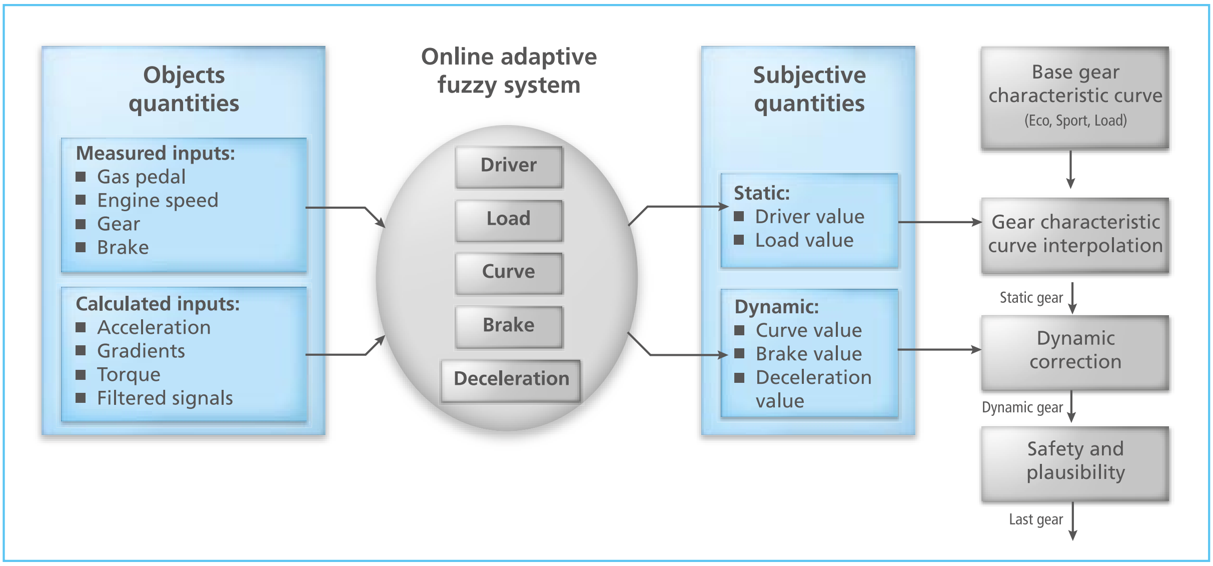


Fig. 3: Gearshift strategy – structure and signal flow.

- Continental, supplier of the transmission ECU, makes the hardware and software available (incl. the model-based software design, automatic code generation, model-based tests, and quality assurance).
- The OEM and Continental perform the system integration and validation of the function in the vehicle via hardware-in-the-loop simulation.

The handling of the process elements, the corresponding methods for model-based development, and the automatic code generation for this DCT project were similar to the handling of the first production project. During the project's run time, additional challenges arose due to the complex DCT functions:

- Function reuse
- Adapted code generation for device-specific calibration methods
- Partitioned design in a multi-user environment

The following three sections briefly describe how these challenges were met.

> "Using model-based development and automatic code generation led to an exceptionally high number of software deliveries precisely on schedule."
>
> *Georg Grassl, Continental AG*

Function Reuse

When looking at a DCT system, it becomes clear that two clutches and four gear components call for the reuse of functions (see fig. 4). The limited hardware resources make reuse an even greater necessity. Actuator functions for two clutches, two shafts and four gear actuators must use a common algorithm.
Simulink libraries are also used in this project for low-level functions (i.e. filter routines), resulting in "nested reuse". The requirements are fulfilled by TargetLink's Function Reuse feature, which made it possible to reduce the resource consumption. Since developers are aware that the functions can be reused, they can take this into account when designing the model architecture.

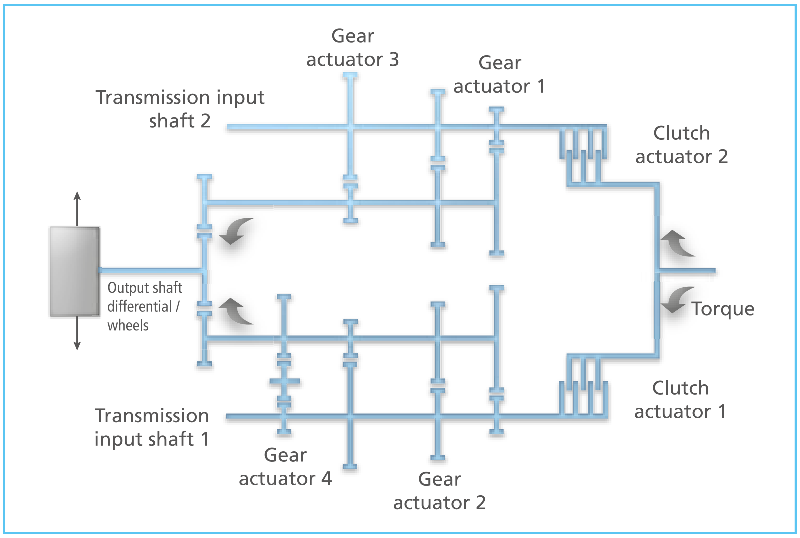


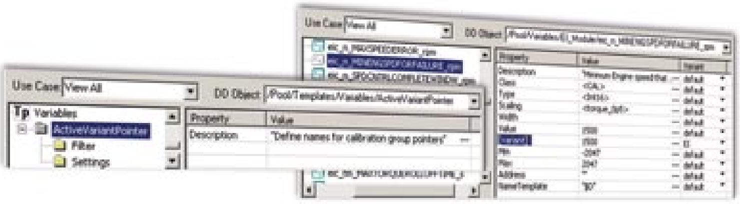
Fig. 4: DCT schematics.

Adapted Code Generation for Device-Specific Calibration Methods

In order to cut development costs in this project, a specific calibration method for electronic control units was to be implemented without calibration devices and/or memory expansion. As a result, a set of calibration data is organized as a structure, and access to this during run time is carried out by rerouting the pointer to the structures from the ROM to the RAM. To implement the necessary code pattern, TargetLink features such as variant coding and templates were used on the one hand (see fig. 5). On the other, we worked together with dSPACE to make changes in the code generator. The conversion of the entire DCT model to these specific calibration methods was completed within just one week, which proves how high the performance of the open program interfaces delivered by the code generator truly is.

Partitioned Design in a Multi-User Environment

In this project, the function design was performed by Continental as well as by the OEM involved. The high number of involved developers was much too large for the previously used single-user method, so the developed processes had to be expanded to meet multi-user capability. To be specific, large functionalities were partitioned into smaller model fragments and



Use Case: View All

DD Object: /Pool/Templates/Variables/ActiveVariantPointer

Property	Value
Description	"Define names for calibration group pointers"

Fig. 5: Definition of a pointer to data variants by using a variable template and assigning variants to parameters.

managed in Simulink model libraries. To keep the data within the environment consistent, the multi-user support offered by the include files of the dSPACE Data Dictionary was used.
The Transmission Team decided to keep the model, the data description, the test vectors (functional and structural), the test areas, and the fine-tuning guidelines as a package (see fig. 6) in the configuration management. This smoothes the way for reuse in other projects and for consistent development.

Results and Important Findings

Thanks to the previously developed processes and methods, 100 % of the DCT project's function software could be developed on the basis of models and autocoded with TargetLink. The function software that arose in this high volume project amounted to:

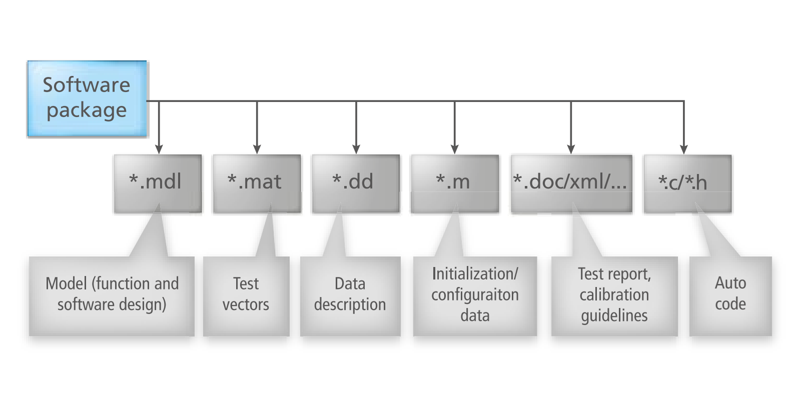
- 250 kByte compiled code
- 120 model libraries

Considering the challenges and corresponding solutions described above, it was important that we could place our trust in the cooperation with those who delivered the code generator. This was especially true for the specific support inquiries the project team had regarding memory assignment, data variants, and how robust the code generator could be. For model-based development, including automatic code generation, it was also helpful to have the updates and patches for the code generator synchronized with the project's software tasks.

Looking Ahead

With the know-how gathered by using the project-specific approach for a model-based design and automatic code generation, the Transmission Team is now ready to use this method for low-level software (actuator control). In addition, we are working on moving from a project-specific approach towards a platform approach, while still maintaining quick methods. This way, we can support the worldwide, in-house introduction of the development method, and still maintain consistency. ■

Georg Grassl
Business Unit Transmission
Gerd Winkler
Business Unit Engine Systems
Continental AG
Germany



Software package
*.mdl
*.mat
*.dd
*.m
*.doc/xml/...
*c/*h
Model (function and software design)
Test vectors
Data description
Initialization/ configuraiton data
Test report, calibration guidelines
Auto code

Fig. 6: Software package for multi-user environments.

Summary

The Business Unit Transmission implements a seamless method for model-based development and automatic code generation. The development phases are closely connected with one another by an executable specification (a model). Meanwhile, the entire function software for the DCT is implemented by using model-based design and automatic code generation. By working together with the OEMs, the Business Unit Transmission is able to set up an efficient process for developing function software.

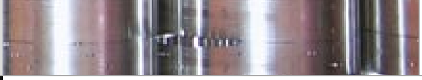
The tool chain used for this method already proved its worth for high volume projects. The process is based on reliable elements from projects with conventional development methods, which were then adapted to the meet the requirements of model-based development and automatic code generation. This made it possible to introduce the new methods efficiently.

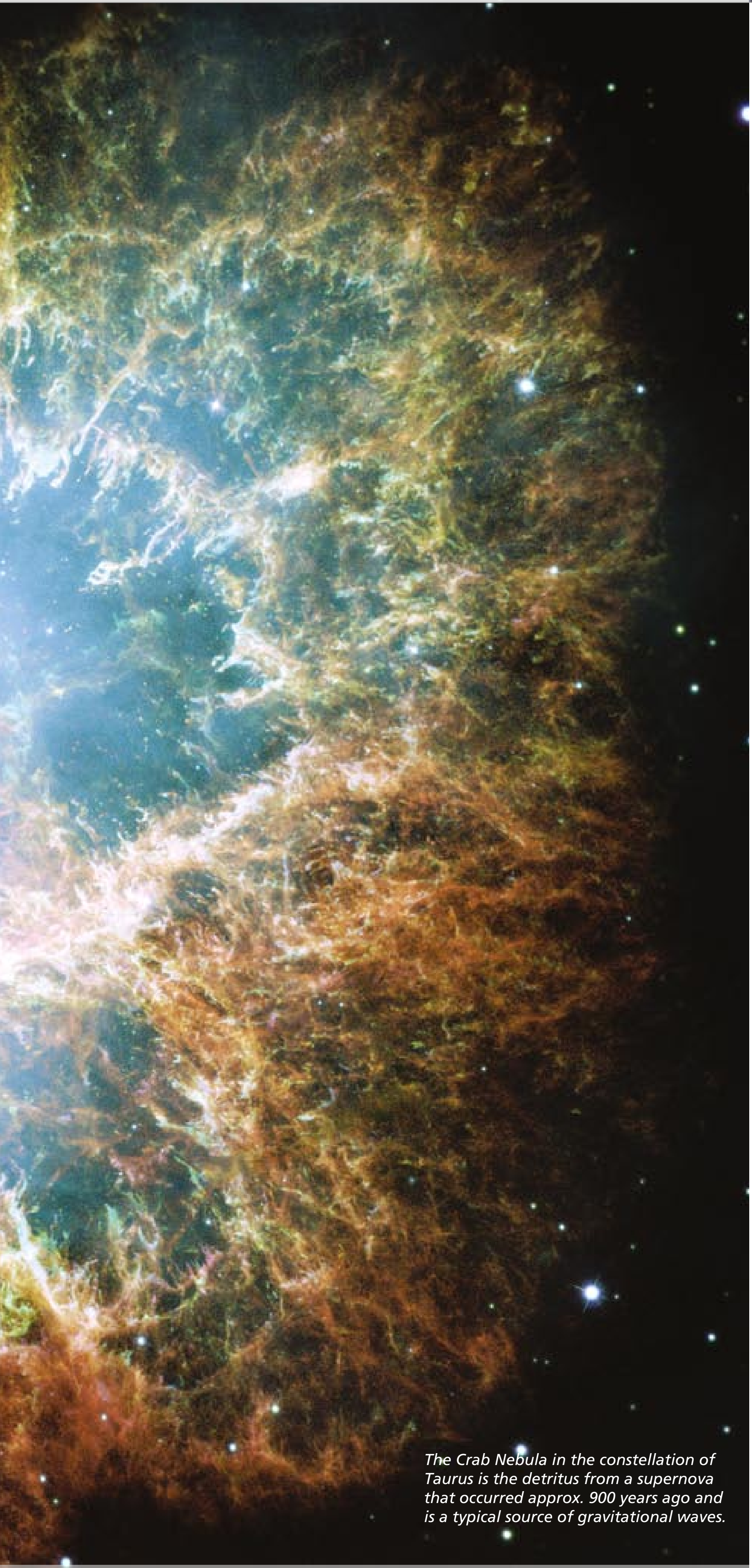
The process was done step-by-step: first, an internal project to implement the new methods; then the initial production project with a 20% share of the autocoded application software; finally, the highly complex DCT transmission with 100% automatically generated application code.

Using this new method resulted in immediate quality improvement. When comparing identically complex projects, one using traditional development methods and the other using model-based development with the resulting options (rapid prototyping, automatic production code generation, model-based testing), it becomes quite evident that model-based development is clearly superior to traditional methods.

Hunting Waves in Space

Gravitational Wave Research with dSPACE Equipment





The Crab Nebula in the constellation of Taurus is the detritus from a supernova that occurred approx. 900 years ago and is a typical source of gravitational waves.

Detecting gravitational waves is one of the unsolved challenges in physics. Albert Einstein predicted their existence in 1915, but he doubted it would ever be possible to detect them. Now modern observation technology has brought this within reach. With the help of dSPACE equipment, researchers at the Laser Interferometer Gravitational-Wave Observatory (LIGO) in the USA have set up an observatory dedicated to detecting gravitational waves.

If all goes as planned, LIGO, the laser interferometer, will detect gravitational waves for the very first time (seen here is the installation in Hanford, Washington). Each of the interferometer's two arms is 4 km long. Pictures courtesy of the LIGO Project

Ripples in Space

Any suitably asymmetric system of accelerated masses generates gravitational waves. These propagate at the speed of light and briefly deform space and all the objects in it as they pass through. These deformations are minute, however. Even if a star in our galaxy exploded – a supernova – the resulting gravitational wave would distort the 150-million-kilometer distance between the Sun and the Earth by just the diameter of a hydrogen atom. And for only a thousandth of a second. The idea that a gravitational wave could shake us like an earthquake is pure science fiction. That's what makes detecting them so incredibly difficult. But despite the fact that they are so weak, gravitational waves would provide valuable information on the Universe – which is why researchers are so interested in them.

> "The dSPACE equipment met our requirements and enabled us to complete the work successfully in the time and budget we had allocated."
>
> *Dr. Mark Barton, California Institute of Technology*

How to Detect Gravitational Waves

Because gravitational waves briefly deform everything in their path, there is one obvious method of detecting them: Monitor the length of a defined distance very accurately. If you observe a sudden variation in length, this might be because a gravitational wave just passed through. The distance needs to be as long as possible, because the longer it is, the more it is deformed. We made this idea for a gravitational-wave observatory a reality by using a laser interferometer. Laser interferometers exploit the wave properties of light to measure lengths precisely and already proved useful as optical "precision rulers" in numerous fields.

Unlike a classic observatory, a gravitational-wave observatory does not need a clear view of a starry sky, since gravitational waves can penetrate any material, even the Earth, without obstacle. So it makes no difference where the observatory is located and which way it is facing – giving it an obvious advantage over conventional observatories.

LIGO: The Gravitational-Wave Observatory

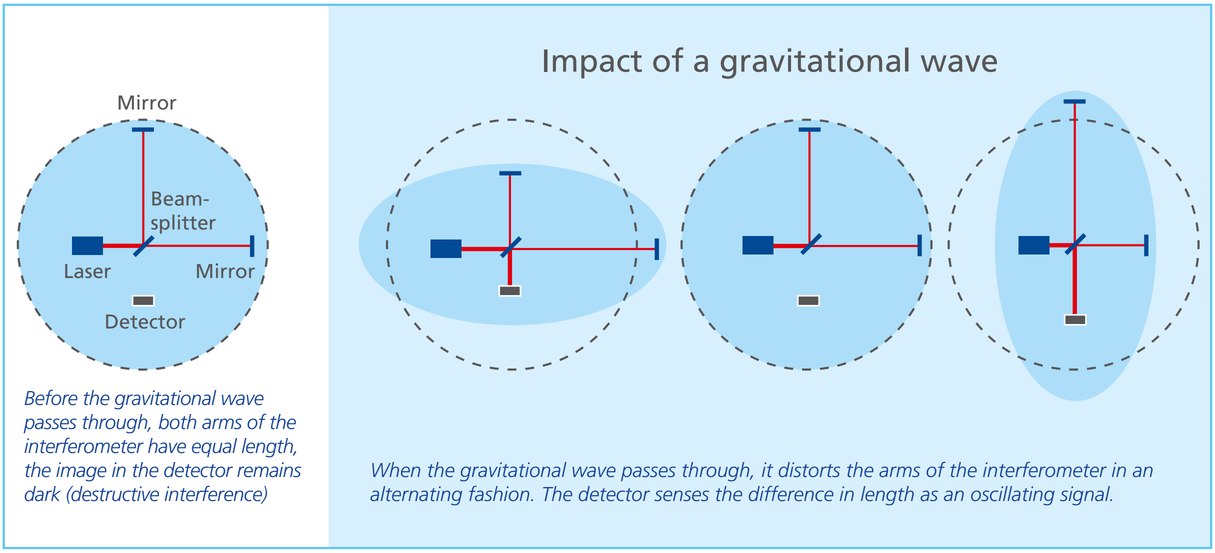
In LIGO (Laser Interferometer Gravitational Wave Observatory), a semi-transparent mirror splits a laser beam. The two half-beams then run along two 4-km beam tubes at right angles to each other. Because each half-beam makes about 50 roundtrips between a pair of mirrors in the interferometer arm, the 4-km arm has the same sensitivity as if it were 200 km long. The half-beams are finally recombined in a photodetector in such a way that they cancel each other out precisely (destructive interference), i.e., the

image on the photodetector remains dark. When a gravitational wave arrives, the lengths of the two arms are deformed in opposite senses, so the half-beams get out of sync and do not cancel each other out completely: a signal appears in the photodetector.
To detect gravitational waves, LIGO has to be able to measure changes in the length of the 4-km interferometer arm that are only a billionth of the diameter of an atom. LIGO is so sensitive that it also registers numerous other effects as disruptive vibrations, for example, sea waves on the coast several kilometers away or a tractor working in a distant field. The main challenge is therefore to isolate the entire facility from this background noise. Because it is the distances between the pairs of arm mirrors that are sensitive to the gravity wave, the suspensions for these “test mass” optics require particular care in design. We implemented active damping for them with the aid of a dSPACE prototyping system.

dSPACE Equipment Stops Seismic Noise

All of LIGO’s main mirrors are suspended from thin wires as a series of pendulums. This alone reduces vibrations above the pendulums’ eigenfrequencies without any further intervention. In Initial LIGO, all the hanging masses were supported by simple single-stage wire loops, but for the forthcoming Advanced LIGO upgrade, we are planning a much more ambitious set of suspensions. The most elaborate of these will be a pair of quadruple pendulums, each with one glass and two metal auxiliary masses. To prove that we could model, control and damp such a complicated system we instrumented the prototype with 20 sensors and actuators distributed around the suspension system to absorb any shocks as fast as

Schematic of the effects of being hit by a gravitational wave (greatly exaggerated for clarity).



Before the gravitational wave passes through, both arms of the interferometer have equal length, the image in the detector remains dark (destructive interference)

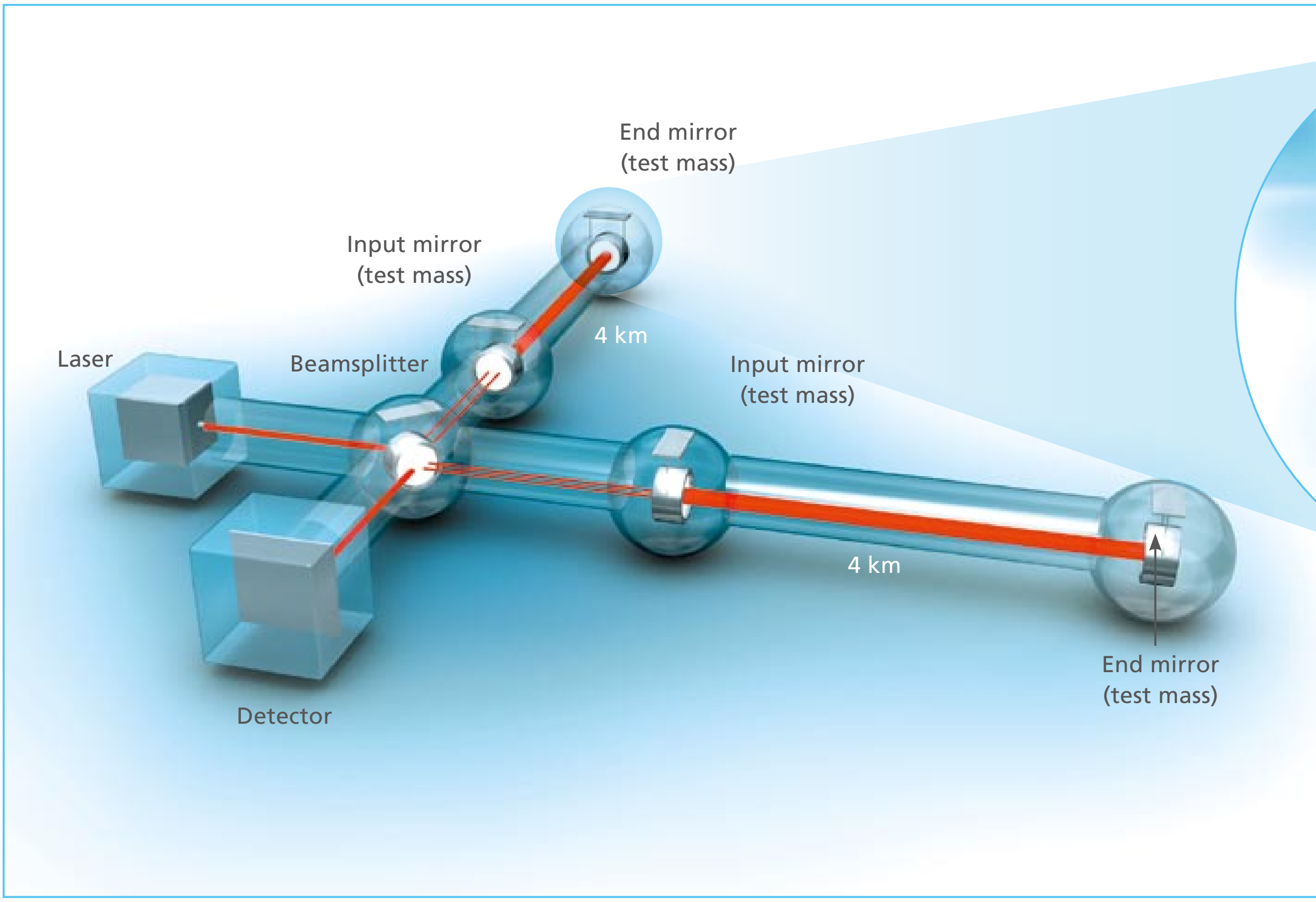
When the gravitational wave passes through, it distorts the arms of the interferometer in an alternating fashion. The detector senses the difference in length as an oscillating signal.

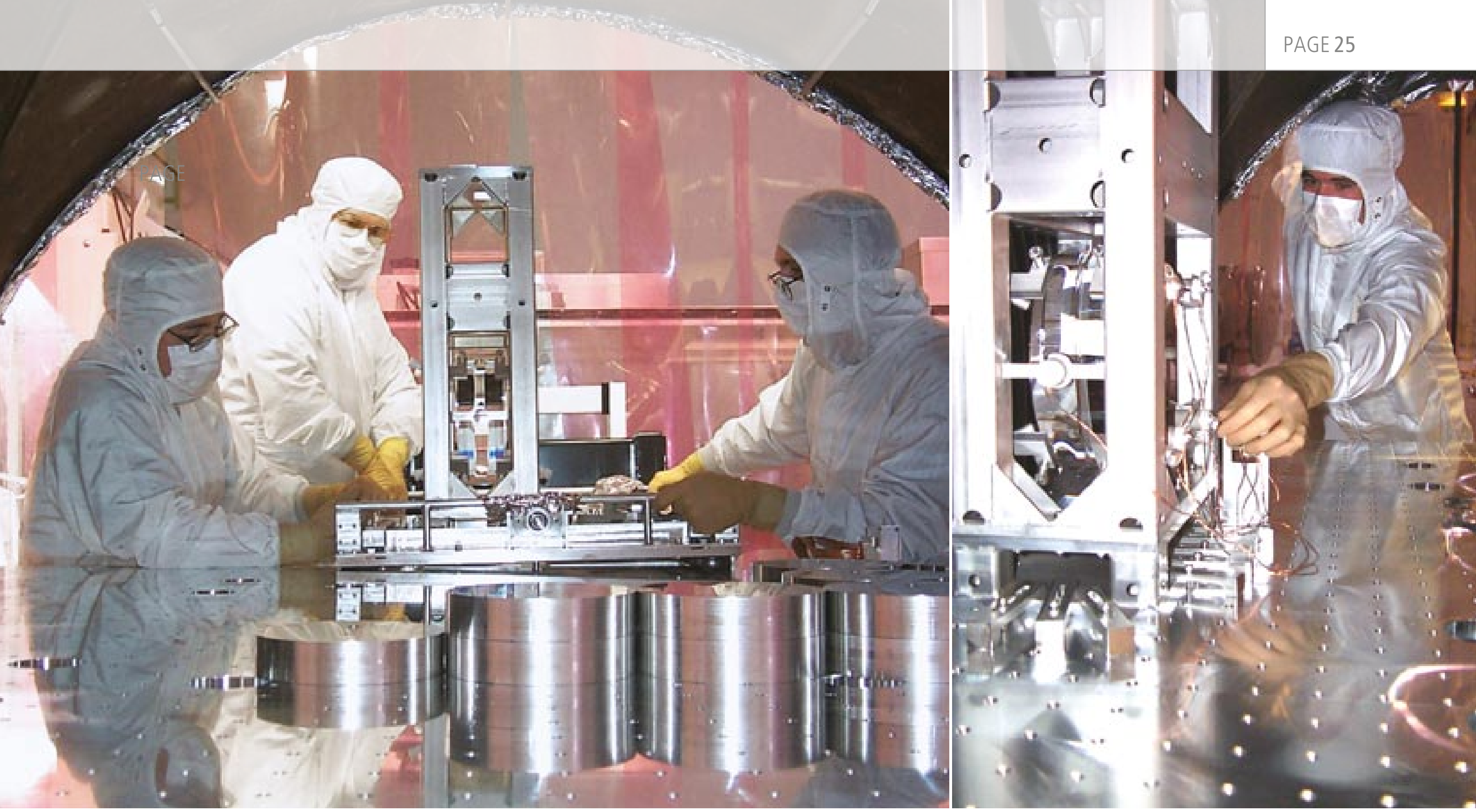
possible. Each sensor is a combination of an LED and a photodiode that registers mirror deflection when the light beam between the two is interrupted by a flag. The actuators consist of voice coils that use the sensor signals to keep the mirrors motionless by acting on magnets. The sensor signals are registered via a dSPACE DS2003 Multi-Channel A/D Board, after which a dSPACE DS1005 PPC Board calculates the appropriate output values, which it transmits to the actuators via several DS2102 High-Resolution D/A Boards. We developed the model for the control with MATLAB®/Simulink®, and the entire experiment is monitored via dSPACE ControlDesk. This easy-to-use combination of MATLAB/Simulink with dSPACE hardware and software enabled us to concentrate completely on the actual experiment and easily stay on schedule. Several test runs with artificially generated disturbance signals verified that the control system was performing correctly.

A Second Observatory – Just to Make Sure

LIGO consists of two identical observatories in different locations in the USA (Hanford, Washington and Livingston, Louisiana). Two observatories are needed to ensure that a presumed gravitational wave is not actually just a local vibration. A real gravitational wave would cause the same signal at both locations, but a local vibration would only occur in one of the installations. Even so, if a presumed gravitational wave signal occurred, it would always be compared with other observatories as well (including ones in Europe and Japan). If the same signal were measured everywhere, that would constitute proof that a gravitational wave had occurred.

Simplified layout of the LIGO gravitational-wave observatory. In the forthcoming Advanced LIGO upgrade, the end mirror suspension will be a quadruple pendulum with a mix of glass and metal upper masses plus a reaction chain of similar design to serve as a quiet reference for applying control forces. The dSPACE equipment made it possible to smoothly design and implement a control system for the prototype. Diagram of suspension courtesy of the University of Glasgow.





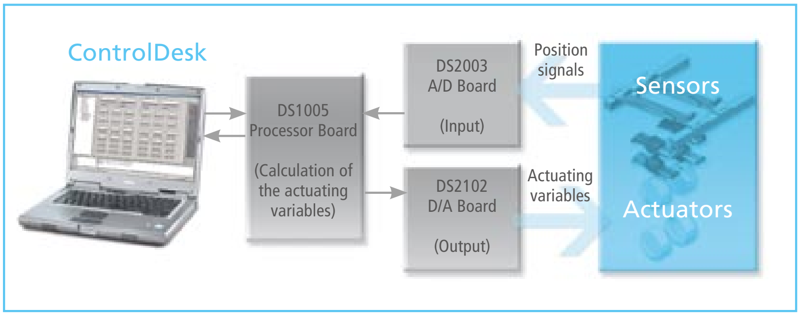
Adjusting the mirrors in the LIGO. Picture courtesy of the LIGO Project.

A New Window on the Universe

Up to now, astronomers have used light and radio telescopes to observe the Universe. Gravitational-wave astronomy with LIGO has given them a new window on the Universe. Large parts of the Universe are concealed behind dark clouds that are impenetrable to waves in the visible and radio ranges. Gravitational waves, on the other hand, can penetrate these clouds unhindered to bring us fresh information on unknown areas of the Universe. Moreover, gravitational-wave astronomy will supply additional information to light and radio wave astronomy, and will provide answers to many open questions concerning black holes, neutron stars, colliding galaxies and other phenomena.

Dr. Mark Barton
LIGO Project
Caltech
USA

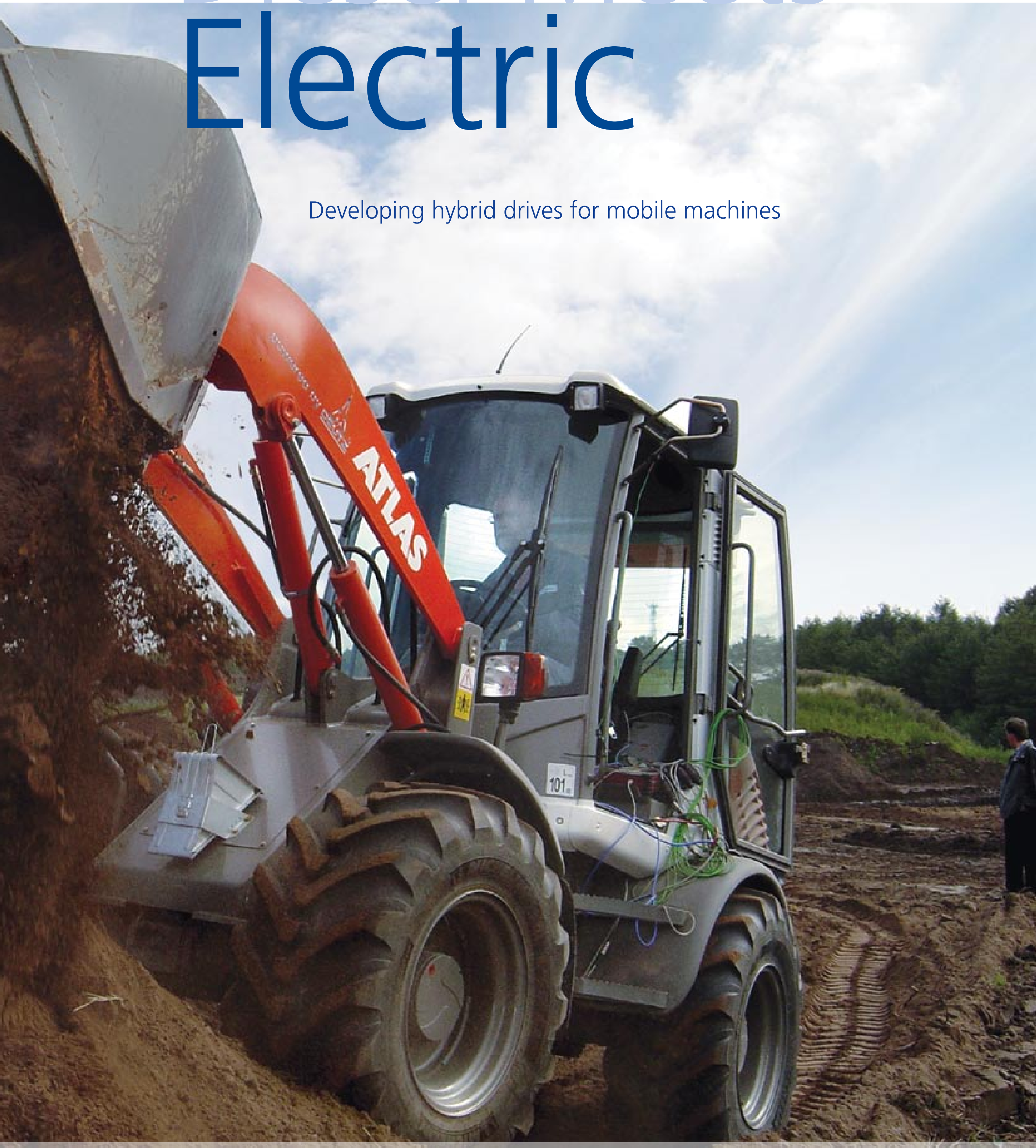
A combination of dSPACE hardware and software is used to actively dampen the mirrors.



For years now, hybrid drives have been gaining ground in passenger cars. But mobile work machines, with their typical stop-and-go operation, are also ideal candidates for hybrid drives. Engine specialist Deutz and construction equipment manufacturer Atlas Weyhausen used dSPACE tools to develop a wheel loader with a hybrid drive. This carries a lot of advantages over a pure diesel engine.

Diesel Meets Electric

Developing hybrid drives for mobile machines



Why Hybrid?

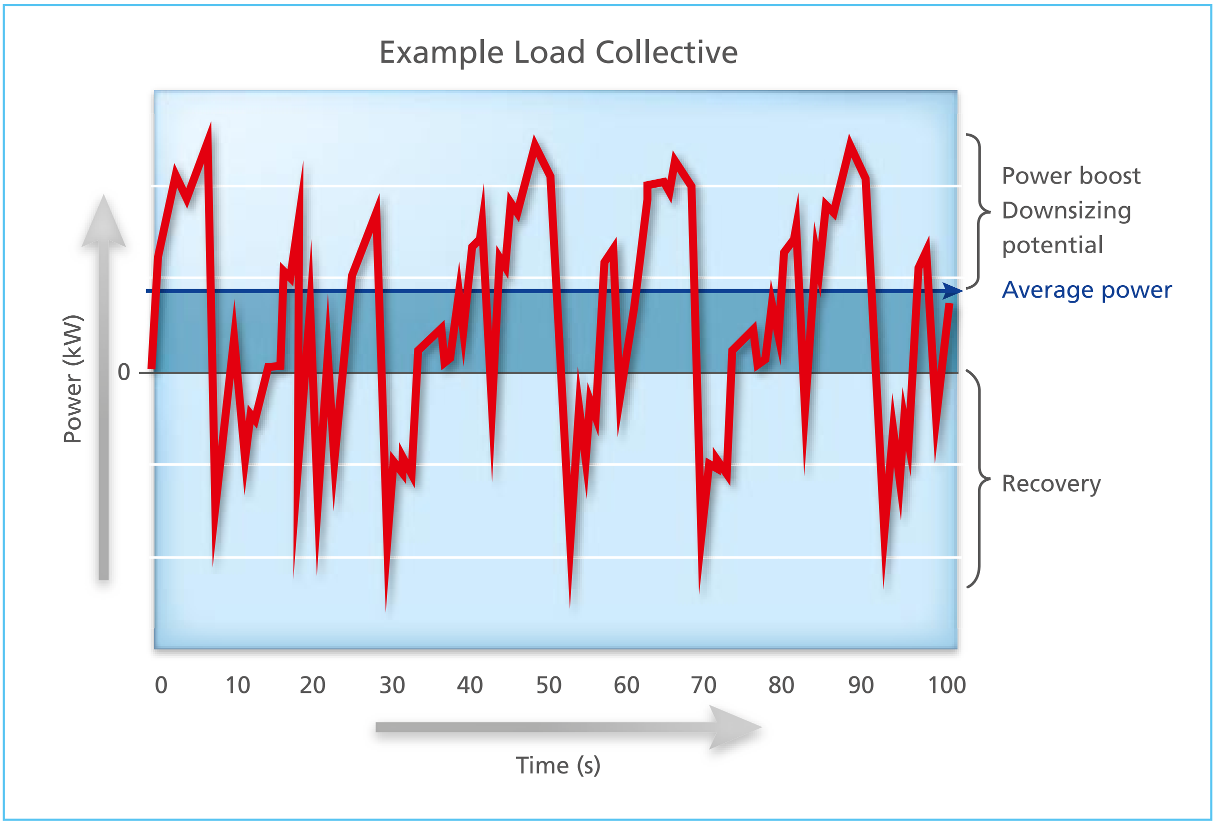
Most of today's hybrid drives are a combination of an engine and an electric motor. The electric motor plays a dual role: On the one hand, it is a generator that stores the energy that would otherwise be dissipated as heat when the vehicle brakes. On the other, it supports the engine at inefficient operating points, for example, when the vehicle starts up or when engine speeds are low. Unlike combustion engines, electric motors can reach high torques in these situations. These advantages mean that when used with suitable applications, hybrid drives have greater system efficiency, which in turn means lower CO_2 emissions and fuel consumption. In the face of rising fuel prices and stricter emission laws, hybrid drives are becoming more popular for this very reason.

Off-Road Benefits

Hybrid drives can really show their strength in any vehicle with frequent acceleration and braking, because this produces a lot of braking energy that can be recovered. These do not have to be on-road vehicles: excavators, loaders, and forklift trucks are all obvious candidates. What these off-roaders have in common is that they brake and accelerate frequently, travel very short distances, and have high load peaks – ideal conditions for using hybrid drives. They also have idling phases in which the motor can be switched on and off automatically to save even more fuel.

Wheel Loader with Hybrid Drive

In a joint project with wheel loader specialist Atlas Weyhausen, we used dSPACE tools to develop what is called a "mild" hybrid system for their AR-65 Super wheel loader. "Mild" means that the electric motor is rigidly coupled to the diesel engine and supports frequent braking and acceleration. This distinguishes mild hybrids from micro hybrids, which have only an automatic start-stop function, and from full hybrids, which allow pure electric driving. Hybrids have great potential for saving fuel by:



Load curve of a mobile machine. A fast sequence of braking, accelerating, and idling phases is typical – and ideal for hybrid drives.

Overview of the Implemented Hybrid Functions

- **Power Boost**
  Electric motor switched in when power peaks occur
- **Raising/Shifting the Load Point**
  Diesel engine operating point shifted to a range with higher efficiency
- **Start/Stop**
  Engine switched on/off during idle phases
- **Energy Recovery**
  Excess energy stored in Li-ion battery

The implemented functions provide a total fuel saving of 20 %, and CO_2 emissions are also reduced.

> "The RTI CAN MultiMessage Blockset meant we could set up the entire CAN communication system quickly and conveniently."
>
> *Marco Brun, Deutz AG*

- Downsizing the diesel engine (using a diesel engine with less power, made possible by the added electric motor)
- Raising and shifting the load point (running the diesel engine in ranges with optimum efficiency)
- Energy recovery (storing braking energy in the battery)
- Start-stop (automatically switching off the engine during idle phases)

System Design

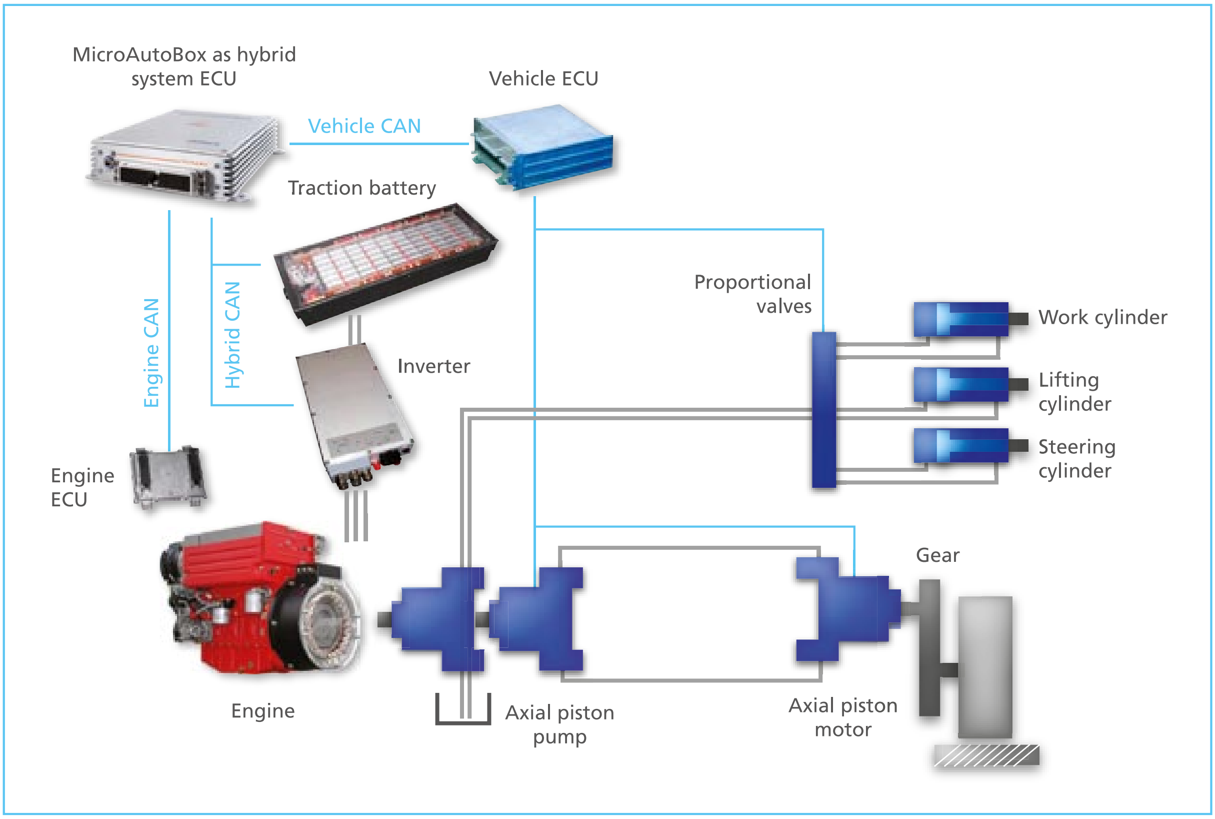
First of all, the wheel loader is driven by a DEUTZ three-cylinder diesel engine that delivers 36.9 kW at an engine speed of 2100 min^{-1}. Secondly, the powertrain is equipped with a constantly excited synchronous machine that has a rated and a peak output of 15 and 30 kW respectively and is integrated into the flywheel bell housing of the diesel engine. The rotor is directly coupled to the crankshaft. This form of integration meant that it was no problem to install the hybrid drive mechanically, as the electric machine requires hardly any additional space.

The hybrid system's traction battery is a lithium ion battery with a rated voltage of 400 V that is connected to the electric machine via a inverter. The inverter is responsible for commutating the synchronous machine and regulating torque by means of field-oriented control.

The hybrid drive is connected to two hydraulic pumps, the traction pump and the work pump. Like the hydraulic traction motor, the traction pump is designed as an axial piston machine. It is responsible for generating the hydraulic volume flow for the traction motor. This is called a hydrostatic traction drive – there is no mechanical gearbox anywhere in sight. The work pump sends the hydraulic oil through proportional valves to the hydraulic cylinders used to raise the bucket and steer the wheel loader.

Development with dSPACE Hardware and Software

We used several dSPACE tools to develop the software functions for the hybrid system's ECU:



Schematic of the mild hybrid system in the wheel loader. The MicroAutoBox is used as a superordinate hybrid system ECU.

- MicroAutoBox (as hybrid system ECU)
- Real-Time Interface (for setting up the I/O interfaces for the MicroAutoBox)
- RTI CAN MultiMessage Blockset (for setting up CAN communication)
- ControlDesk and CalDesk (for calibrating the hybrid functions)

By using RTI and the RTI CAN MultiMessage Blockset, we were able to implement fully functioning system software on the MicroAutoBox in only 3 months. The RTI CAN MultiMessage Blockset proved to be a very easy-to-use tool, and its support for linking CAN configuration files (DBC files) enabled us to set up the CAN communication very quickly.

Three CAN channels were set up in the wheel loader: engine CAN, hybrid CAN, and vehicle CAN. Because we programmed the system software directly in Simulink, we were able to try out the software functions immediately on a plant model (MIL) containing the engine, electric machine, inverter, battery, work hydraulics and traction hydraulics components. We were therefore able to test the software functions long before the first prototype components became available. This was absolutely essential in view of the very short development time assigned to this project.

Using the pretested software functions and the inputs and outputs configured with RTI (digital, analog, PWM, CAN), we produced a software version that would run on the MicroAutoBox and tested it on the test bench. Functions such as start/stop were tested and calibrated by means of ControlDesk and CalDesk.
Finally, we put the wheel loader into operation with the MicroAutoBox as a superordinate hybrid system ECU and implemented the functions for boosting power and raising/shifting the load point.

Cutting Emissions and Costs

By reducing fuel consumption, the hybrid drive not only lowers CO_2

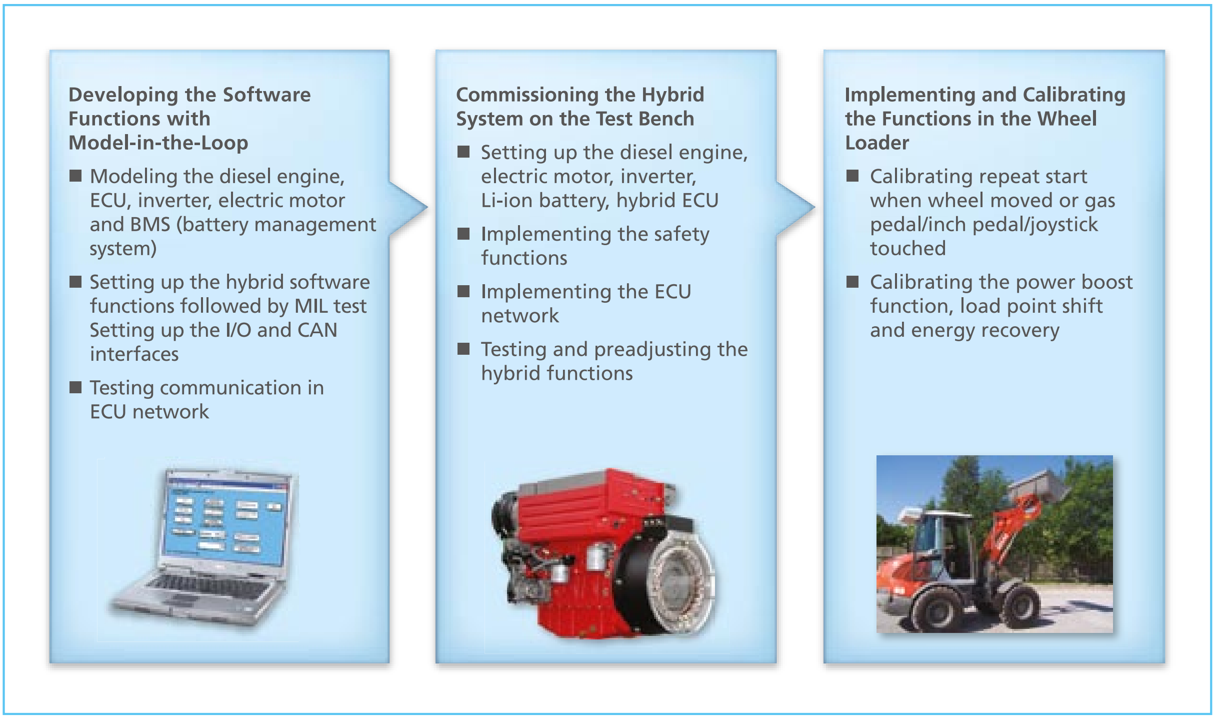
emissions, it also results in considerable cost savings. An example calculation illustrates this clearly: Starting with a 20 % reduction in fuel consumption (and assuming that fuel consumption was previously 6.5 l / hour and that diesel costs € 1.30), € 1.70 per hour will be saved on a typical working day, or € 1500 in a year. The total savings for the entire working life of a wheel loader would more than offset the higher price of acquisition for the hybrid drive.

Production Status in 2010

With these obvious benefits in terms of emission and cost reduction, we see great potential for hybrid drives in the field of mobile work machines. So as our next step, we will employ further wheel loaders and construction equipment wheel loaders – each equipped with a dSPACE MicroAutoBox – at various customers to gather experience in everyday use. Our aim is to bring this hybrid system up to production status by mid-2010.

Marco Brun
Deutz AG
Germany

Externally, the only part that differs from the standard wheel loader is the roof-mounted lithium ion battery.



Developing the Software Functions with Model-in-the-Loop

- Modeling the diesel engine, ECU, inverter, electric motor and BMS (battery management system)
- Setting up the hybrid software functions followed by MIL test Setting up the I/O and CAN interfaces
- Testing communication in ECU network

Commissioning the Hybrid System on the Test Bench

- Setting up the diesel engine, electric motor, inverter, Li-ion battery, hybrid ECU
- Implementing the safety functions
- Implementing the ECU network
- Testing and preadjusting the hybrid functions

Implementing and Calibrating the Functions in the Wheel Loader

- Calibrating repeat start when wheel moved or gas pedal/inch pedal/joystick touched
- Calibrating the power boost function, load point shift and energy recovery

The workflow for developing the hybrid software functions. Fully functioning system software was implemented on the dSPACE MicroAutoBox in the wheel loader after only 3 months.

ACC in the Loop

Hitachi, Ltd.: Effective ACC Development Using dSPACE Simulator

The Automotive Systems group at Hitachi, Ltd. uses dSPACE hardware-in-the-loop simulators to efficiently develop an Adaptive Cruise Control (ACC) system that controls the distance to the vehicles and traffic ahead. The test environment setup uses a dSPACE Simulator to perform functional verification tests of the ACC system, and significantly reduces the development time and required man-hours.

Mr. Takaki in the Hitachi test laboratory.

Hitachi's Approach to Safety

Hitachi always emphasizes safety. This is why we make a continuous effort to develop various systems that support safe driving.

We have already put the following techniques into practical use: image processing cameras that supplement human vision and are capable of stably recognizing traffic; ACC systems that control the distance to the vehicle ahead by using laser radar or millimeter wave radar; Lane Keep Support (LKS) systems; and pre-crash braking systems.

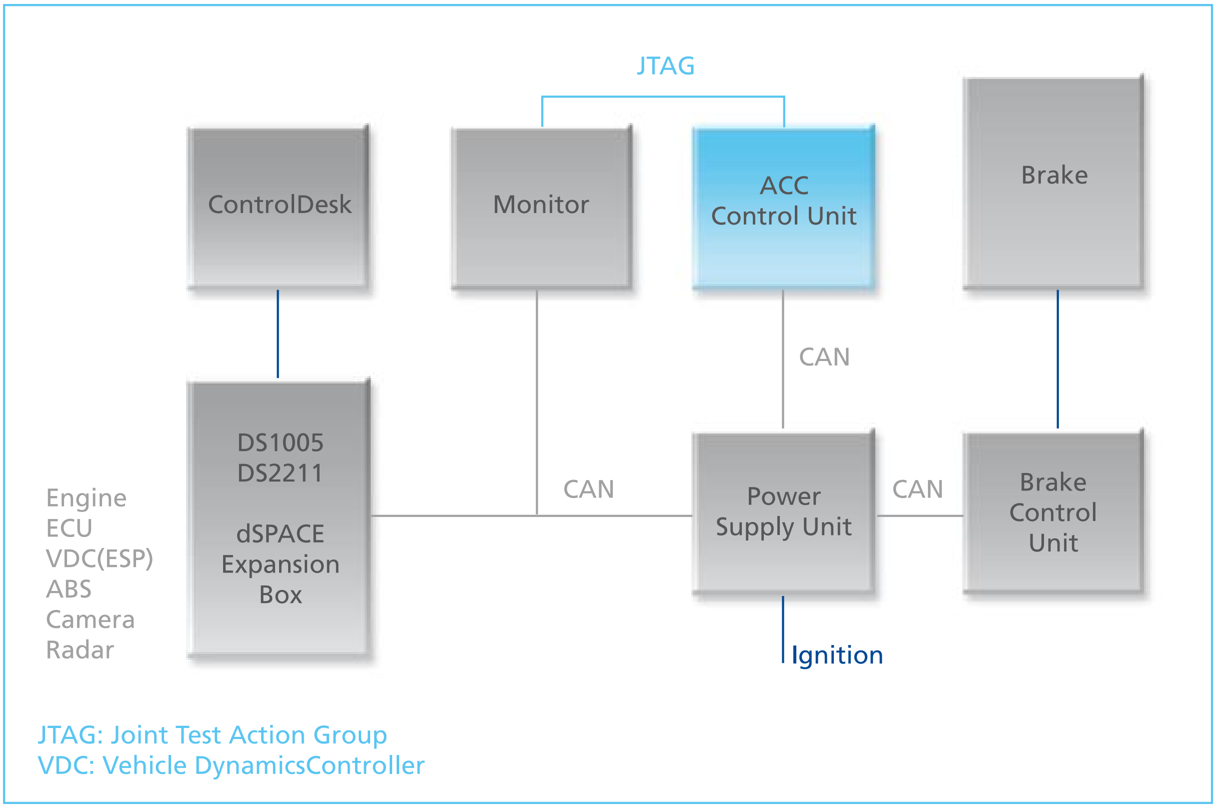
Role of Hardware-in-the-Loop (HIL) Simulation in ACC Development

With the conventional ACC test environment, it took a great amount of time to establish and prepare a development environment because switch boxes and control programs were all made in-house. In addition, every time a change was made to the ACC specifications, numerous efforts were required to update the environment. In a test environment using a hardware-in-the-loop (HIL) simulator, however, it is possible to respond flexibly to changes in test conditions by simulating everything in the software (e.g. camera and radar outputs, vehicles, and surrounding environments). When the conventional test environment was used, the time spent for developing the first generation ACC was about 12 months, whereas the development period of the second generation ACC was about 6 months when the HIL simulator was used. This is a significant time reduction. In addition, the development period for the third-generation ACC based on the know-how from the second generation HIL simulator environment, was reduced to about 3 months. We attained much higher development work efficiency.

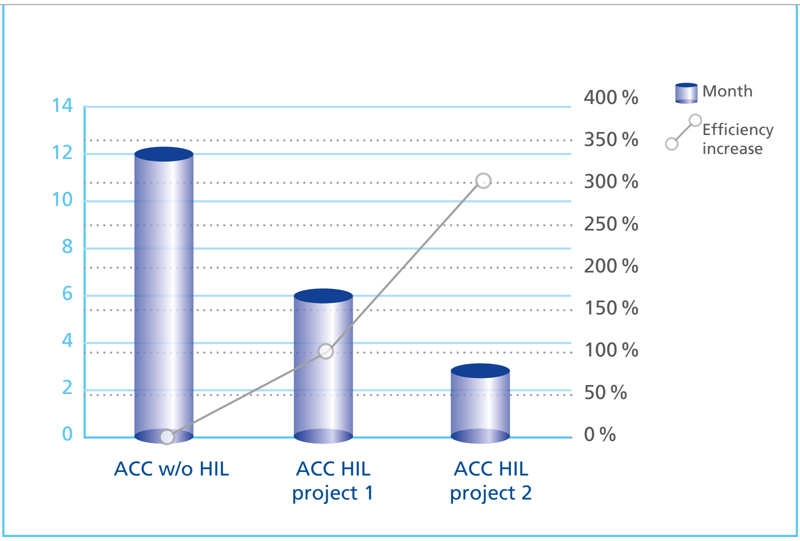
> "I've been using dSPACE products for seven years, and haven't encountered any problems with them."
>
> *Mr. Takaki, Hitachi, Ltd.*

Evaluation of dSPACE Products

The Automotive Systems group has been using dSPACE products, without any problems, for over 7 years



Overview of the test system where the dSPACE HIL simulator, an ACC control unit, and a brake control unit are connected via a CAN network.



Development time reduced from 12 to 3 months by using HIL tests.

> "Test environments using dSPACE Simulator reduce the development time significantly, resulting in efficient development."
>
> *Mr. Kawakami, Hitachi, Ltd.*

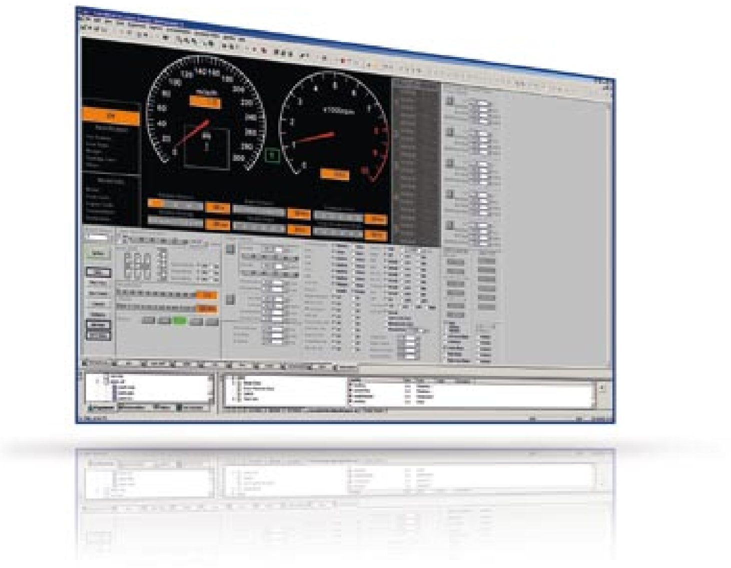
now, and highly values the reliability and durability of the products. As usability of the software is high, it is now possible to easily grasp the vehicle movements with the test parameters (inter-vehicular distance, vehicle speeds, etc.).

The Future of ACC Development

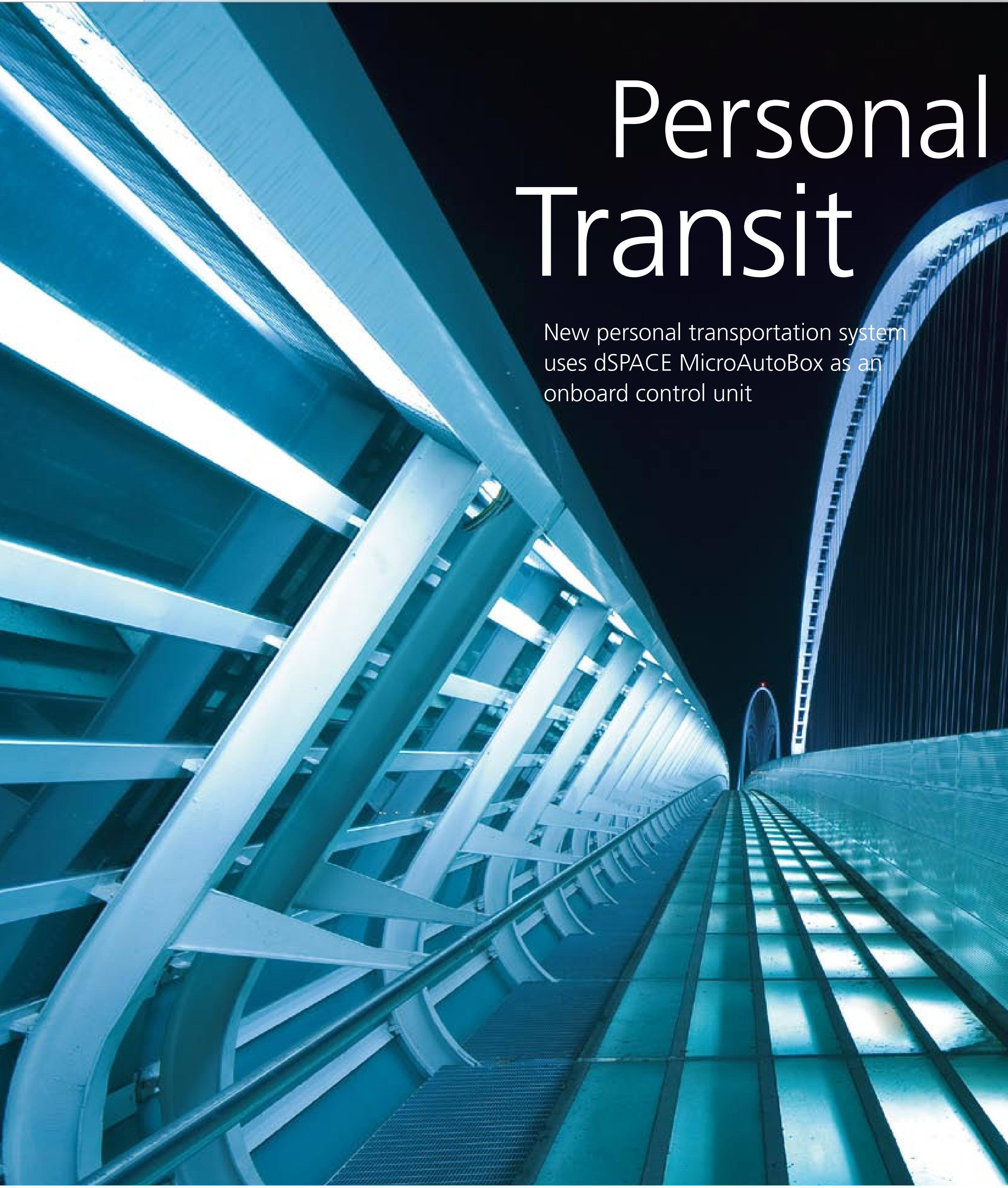
ACC systems used to be installed in only a few luxury vehicles. Since then, the number of vehicle models/types equipped with an ACC system has increased. If all the variants of the models/types, which vary according to destination and hardware, are included, testing will have to be performed for all of the several hundreds of ACC variants. With HIL simulation, it is possible to flexibly test such a wide variety of variants, just by changing software models.

ACCs developed at Hitachi, Ltd., which were implemented by many automobile manufacturers, have made a significant contribution to increasing the safety of automobile traffic systems.

Mr. Takaki
Mr. Kawakami
Hitachi, Ltd.
Japan



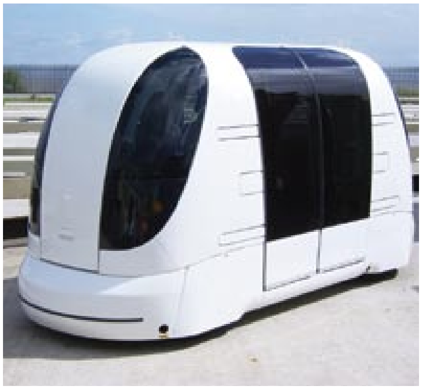
Direct control and visualization of ACC signals in ControlDesk.



Personal Transit

New personal transportation system uses dSPACE MicroAutoBox as an onboard control unit

Rapid



Welcome to the future! From 2009 onward, passengers will be able to travel non-stop across London Heathrow Airport in the ULTra Personal Rapid Transit (PRT) system. This new, environmentally friendly, individual passenger transportation system has been developed by ATS Ltd. The airport operator is using 18 vehicles in the pilot phase – each controlled by its own dSPACE MicroAutoBox as it travels along a 3.8 km guideway network.

Passengers request their destination at the station and then board their ULTra PRT vehicle. The vehicle travels to the destination non-stop in the slot allocated to it by the synchronous control system.

Intensive Trial Phase

The ULTra PRT system due to go into operation at Heathrow has been under development for over 5 years. Mule and pre-production prototype vehicles have been tested extensively at ATS's Test Track in Cardiff (Wales). Testing has included passenger trials encompassing a wide range of users, including those with disabilities. The test program has demonstrated the reliability of the vehicles including their batteries and opportunity-based battery charging strategy. The use of onboard batteries for power storage is considered to be a key feature of the system design since it avoids the high infrastructure cost and single-point-of-failure issues associated with continuous power pick-up systems.

> "Reliability and flexibility were our main concerns in choosing the onboard control units for ULTra PRT, and the MicroAutoBox gives us both."
>
> *Dr. Torquil Ross-Martin, ATS Ltd.*

The World's First PRT System

The ULTra PRT developed by ATS Ltd. from Bristol (UK) is the first system ever of this type in the world. ULTra's synchronous control works on the slot principle used in air traffic control, in which a vehicle sets off as soon as a route is guaranteed to be free from the starting point to the selected destination. Delays or jams never occur – each vehicle travels non-stop to its destination in a specially reserved slot and by the fastest route. The vehicles roll on tires along a special guideway made of concrete and steel. The four-seater vehicles are around three and a half meters long, barrier-free, and can carry a 500 kg payload. Each has its own autonomous, battery-powered electric drive and is controlled by its own dSPACE MicroAutoBox. Passengers select their destinations at the station, and they can also choose who they travel with – just like traveling by car. The environment also benefits, since ULTra PRT's battery-powered vehicles require only about a third of the energy consumed by a car per passenger kilometer. This dramatically reduces the volume of CO_2 emitted by the overall system per passenger kilometer.

The Control Unit: The Heart of the PRT System

ATS Ltd. is using dSPACE MicroAutoBox as the ULTRa PRT's onboard vehicle control unit throughout development and is using them for the pilot system vehicles. The MicroAutoBox executes the main vehicle supervisory control functions, the guidance and navigation functions, and autonomous driving (communicating with smart actuators for steering, drive and braking functions). The MicroAutoBox is also in charge of controlling ancillary systems such as HVAC (heating, ventilation, air conditioning), lighting, and battery charging, and performs fault detection and health monitoring functions. To ensure safety, a guideway-based Automatic Vehicle Protection (AVP) system is used to ensure that safe separation is maintained between vehicles at all times. The MicroAutoBox plays an important role in this system, initiating emergency braking when commanded to do so by the AVP system. The PRT system is subject to Her Majesty's Rail Inspectorate (HMRT) regulation, and throughout development ATS have maintained a dialog with HMRI so as to ensure a smooth passage to safety approval.

Looking Forward

From 2009, the ULTra Personal Rapid Transit system will go into pilot operation on a 3.8 km guideway network at London Heathrow Airport. In the first phase, it will connect Terminal 5 and a business parking facility. After successful completion of the pilot project, ULTra PRT will be extended to other parts of the airport, connecting other terminals, hotels, car rentals, and multi-story parking facilities. ATS Ltd. and its business partners are already planning other projects in Europe, the Middle East, and the USA. ATS plans to continue adding functionality to its vehicles and expects to continue utilizing dSPACE systems in their development.

Interview

with Dr. Torquil Ross-Martin, Head of Research and Development, ATS Ltd.

Dr. Ross-Martin, why did ATS Ltd. decide to equip each vehicle in the ULTra PRT system with its own dSPACE MicroAutoBox?
Reliability and flexibility were our main concerns in choosing the onboard control units for ULTra PRT, and the MicroAutoBox gives us both. The powerful processor and large memory and I/O count mean that we have the reserves we need, especially in prototype operation. It gives us the flexibility to meet new customer requirements or make extensions, and we were able to go into production early in the knowledge that there were plenty of resources left for later requirements. It's an established, proven product with an effective tool chain, and has completely met our expectations.

What was the most critical part of control development for the ULTra PRT?
For everyday operations, we had to make the system robust against faults and errors and also develop suitable diagnostic and corrective mechanisms. This was a greater challenge, and we met it – as approval from Her Majesty's Rail Inspectorate shows.

Dr. Ross-Martin, what makes your PRT system so outstanding, compared with bus and rail transport?
Buses and trains are usually only available according to a fixed schedule. They stop at numerous stations, which are irrelevant for the passengers not getting off. This makes journeys unnecessarily long. The ULTra PRT from ATS Ltd. gives passengers independence. A separate vehicle is allocated to each passenger or each group of passengers and their baggage, and it takes them straight to their destination without stopping. Passengers will no longer have to bother with schedules or share vehicles with strangers.

How do you see the future of ULTra PRT?
Our system has been designed primarily to provide transport in urban areas – not just the airports and business campuses where it is first being applied. It has great advantages in comfort and convenience compared with conventional public transport and in safety, emissions, and reliability compared with cars. Moreover it is designed as a complementary system – it can be added into existing cities without disrupting established road and rail systems. Thus I expect to see a very strong future with systems appearing in cities within the next 10 years.

Thank you, Dr. Ross-Martin.

Each individual vehicle in the ULTra PRT system has its own MicroAutoBox as an onboard control unit.

Summary

- ULTra PRT – New modular personal rapid transport system from ATS Ltd.
- Control functions developed and tested with dSPACE prototyping systems.
- dSPACE MicroAutoBox has a new job as a regular onboard control unit.

Smoothing the Tension

Optimal web tension control with a dSPACE prototyping system

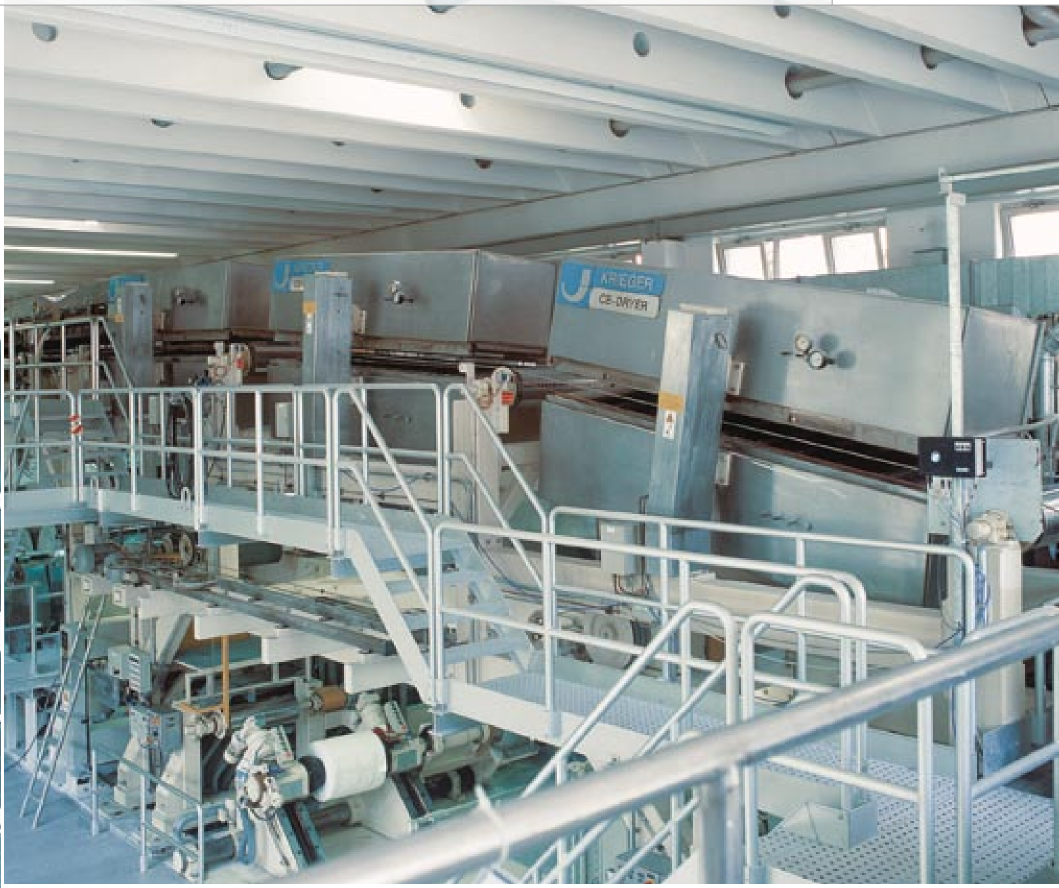


Fig. 1: The VESTRA paper coating system – the pilot coater at PTS, Munich.

When paper is finished in a coating system, it can be given either a glossy surface or a smooth, matte or closed one. Coated and processed paper meets the highest standards of printing quality. However, the production of coated paper is prone to web breaks, which causes productivity losses. Current coaters either cannot compensate for huge variations in web tension, or do not compensate sufficiently.

Modern coating systems are continuous production plants where the paper webs are processed at different production sections. The paper runs through several processing steps and must endure elastic or plastic deformations. The individual subsystems are coupled with each other by the running paper web itself. This significantly affects web stability throughout the entire coater and can even cause web breaks, resulting in idle times and plant standstills. Fluctuations in web tension arise throughout the system from changes in the physical parameters of the paper web during coating or in speed variations of the plant, from malfunctions or from applying/removing the blade and resulting coupling effects that are transmitted throughout the whole coater by the web. Conventional tension controllers are unable to compensate for these fluctuations sufficiently. If the web tension is not held within certain bounds (especially during coater start-up and shutdown), the attainable production quality may suffer badly or even be useless. Our objective was therefore to develop an improved control system for the web run by stabilizing web tension, taking into account the electrical and mechanical behavior of plant and web.

System Model of the Pilot Coater

To design an optimal controller and implement it on the pilot coater, we needed to acquire a thorough understanding of the entire system's physical and technical properties with regard to the web run to be optimized. We were able to model, simulate, and analyze the entire system using the collected plant data and the known, nonlinear physical system description (fluid mechanics and elasticity theory). We identified the following subsystems, coupled by the paper web, as the most important components of the system:

- Drive system (motor, gear, shaft, and clutch)
- Nip section (guide roller for the web)
- Paper web (textile web)

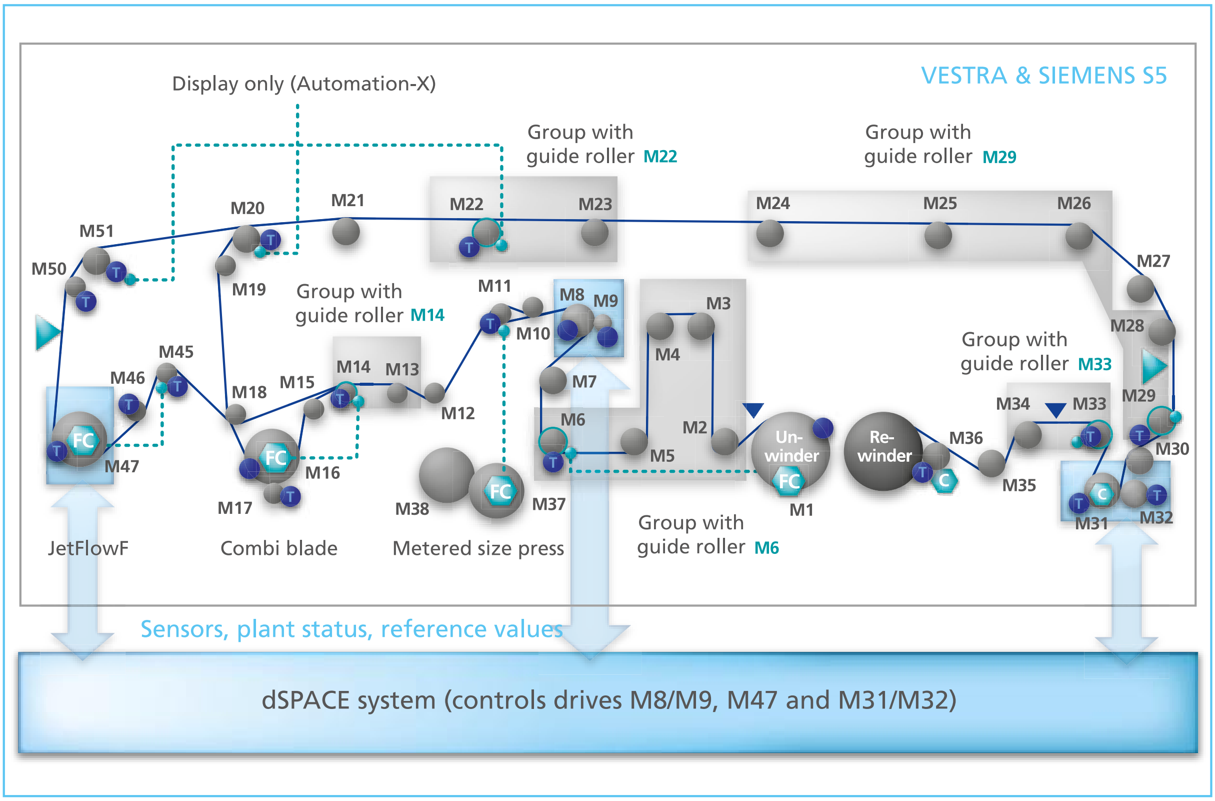
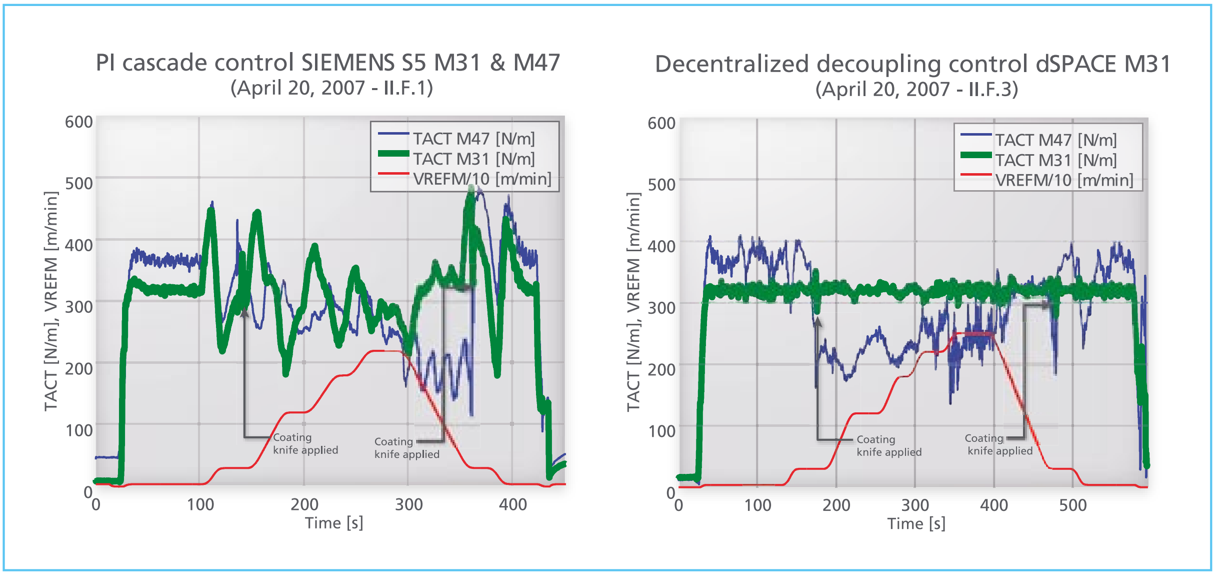

Display only (Automation-X)
VESTRA & SIEMENS S5
Group with guide roller M22
Group with guide roller M29
M51
M50
M20
M21
M22
M23
M24
M25
M26
M27
M19
Group with guide roller M14
M11
M8
M9
M10
M3
M28
Group with guide roller M33
M45
M46
M18
M15
M14
M13
M12
M7
M4
M6
M5
M2
M34
M33
M29
M36
M30
FC
T
C
M47
M17
M16
M38
M37
Un-winder
Re-winder
M1
M35
M31
M32
JetFlowF
Combi blade
Metered size press
Group with guide roller M6
Sensors, plant status, reference values
dSPACE system (controls drives M8/M9, M47 and M31/M32)

Fig. 2: The VESTRA coater with an interface to the dSPACE system for testing and verifying the new control structures.


PI cascade control SIEMENS S5 M31 & M47
(April 20, 2007 - II.F.1)

Decentralized decoupling control dSPACE M31
(April 20, 2007 - II.F.3)

Fig. 3: Comparison between conventional PI cascade control (SIMATIC S5) and Decentralized Decoupling Control (dSPACE) at gripping press M31/M32 at different speeds.

These components describe the behavior of the paper and the web tension as a function of the drive units (motor and guide roller) and the possible load effects from friction or from applying the coating knife at the coating device. Our system model allows us to study coater processes offline and provides an easy way to test and analyze new strategies.

Properties of the Improved Control Structures

In all continuous production processes in which a paper web is conveyed through various subsystems, the overall plant behavior in relation to the paper web is complicated. The individual processing sections and/or nip sections (gripping presses or rollers) are coupled with one another via the paper web. Any undesired variations in web tension first affect the source subsystem and are then passed to succeeding and preceding subsystems. The control structures must therefore have the following properties:

- Good reference tracking (short transient time and small overshoot) and disturbance rejection of the control loop
- Robustness (for handling parameter uncertainties or variations)
- Good decoupling of the subsystems (optimal suppression of coupling effects)

To facilitate the implementation of the new control structures on the VESTRA coater, we decided to use rapid control prototyping with a dSPACE system. The dSPACE system was connected to the VESTRA via a specially developed electronic interface (see fig. 2) and allowed direct control of drives M8/M9 (gripping press 1), M47 (applicator device JetFlow-F) and M31/M32 (gripping press 2).

Decentralized Decoupling Control

Our first approach to stabilizing the web run is Decentralized Decoupling Control. Simulations allowed us to efficiently study and compare conventional PI cascade control (the control structure in the SIMATIC S5) and Decentralized Decoupling Control with respect to control performance, robustness and decoupling properties. The simulated plant behavior with PI cascade control reflected the qualitative and almost the quantitative behavior of real plant if delays due to slow cycle times and communication delays (due to the ProfiBus) are taken into account. First the standard PI cascade control was tested with the Siemens SIMATIC S5 and with the dSPACE system. Both had an identical controller structure and design. However, the dSPACE system provided clearly improved control performance simply because of its fast cycle time (the time the microprocessor takes to execute the programmed algorithms) and its negligible communication delays (unlike the SIMATIC S5, the dSPACE system is connected to the inverters directly).

On the other hand, the coupling effects due to the paper web are approximately the same for both systems. The Decentralized Decoupling Control at gripping press M31/M32 implemented on the dSPACE system reacts noticeably faster and more effectively to events such as the propagation of tension fluctuations arising from applied or removed coating blades or from changing reference machine speeds (VREFM). The web tension TACT M31 at gripping press M31/M32 is considerably smoother (see fig. 3).

In comparison with conventional PI cascade control, Decentralized Decoupling Control achieves high to very high control performance quality: good reference tracking, good disturbance rejection and broad robustness under varying production conditions.

Feedforward Torque Control of Applicator Roller

One of the most critical situations during paper coating arises when the coating knife (blade) is applied to wipe off excess coating color. The blade imposes a load on the coating device's impression roller and generates a load torque $M^*_{L,i}$ with pulse-shaped behavior at the time of blade application. Adding the coating color results in a change in paper properties (Young's modulus $\Delta E_{i-1,i}$) and in a certain amount of "lubrication" between blade and paper. This reduces the effective torque to an approximately constant value that the impression roller drive has to compensate for continuously during the coating action in order to maintain the desired speed and/or tensile force (web tension). Without feedforward torque control of the applicator roller, this dynamic load torque on the impression roller causes a noticeable drop in speed, which inevitably leads to decreasing web tension at the nip section and to increasing web tension at the succeeding subsystem. These tremendous tensile force fluctuations in TACT M47 may lead to undesired and dangerous web breaks. Our second approach is therefore a feedforward torque control that can be implemented modularly at the specified applicator roller and independently of any preset control algorithm used at the coating device. If the respective moments at which the blade is applied or removed, (i.e., the start and end of coating) are known, the feedforward torque control we have developed provides almost complete compensation of the load torque. It does this by means of the feedforward torque MFFC_Blade, whose characteristics can be determined,

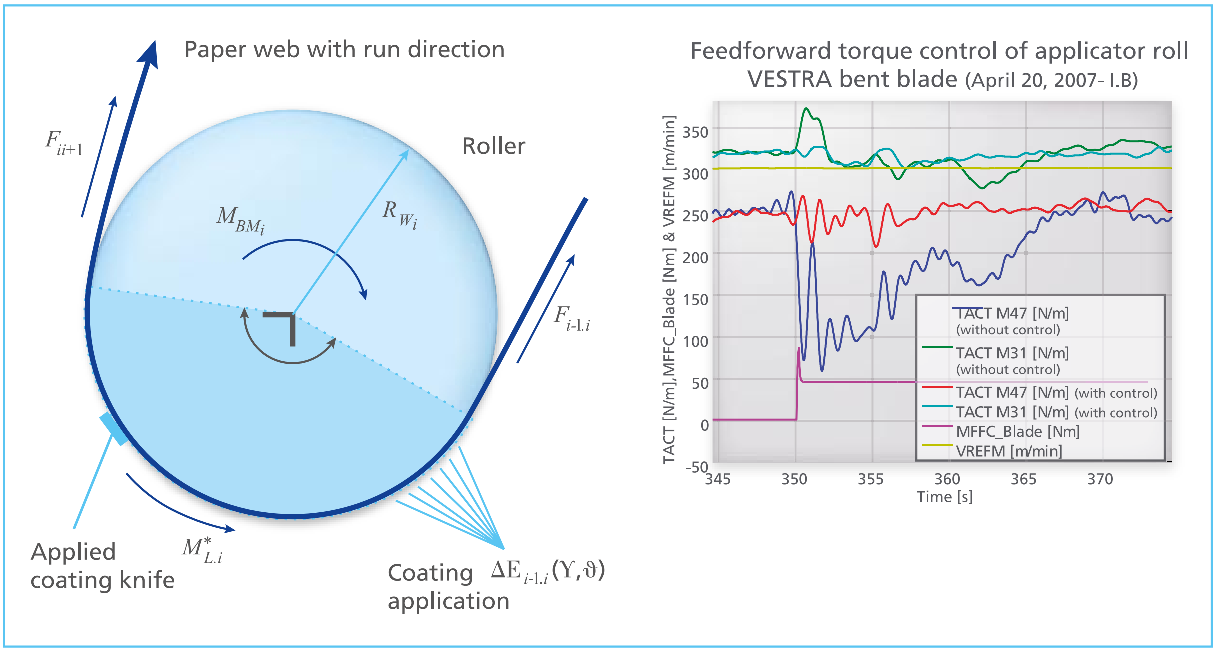


Fig. 4: Coating device M47 (JetFlow F) with coating knife (here: bent blade) – Comparison with and without feedforward torque control of applicator roller.

Glossary

Cascade control –
Nested control loops. The overall plant is divided into smaller controlled systems to simplify controller design and implementation for each.

Decentralized decoupling –
Specially designed state space control that minimizes the coupling effects of the remaining system on a specific subsystem.

Nip section –
According to an idealized model, each subsystem at which the web does not slip through or is subject to lateral offset.

> "The fast cycle times of the dSPACE system alone resulted in considerably improved control performance."
>
> *Christoph Hackl, Technische Universität München*

e.g. empirically, and stored in look-up tables. The feedforward torque control of the applicator roller was implemented and its applicability was verified at the VESTRA by simulation and experiment. Almost precise compensation was achieved (see fig. 4).

Result

We developed a simulation toolbox consisting of several modules for individual nip sections (drive, roller and paper web) and for the entire VESTRA. This toolbox facilitated the efficient testing and offline verification of both new control structures: the Decentralized Decoupling Control and the feedforward torque control. The new strategies were implemented on the actual plant via the extended electronic interface between the dSPACE system and the VESTRA.

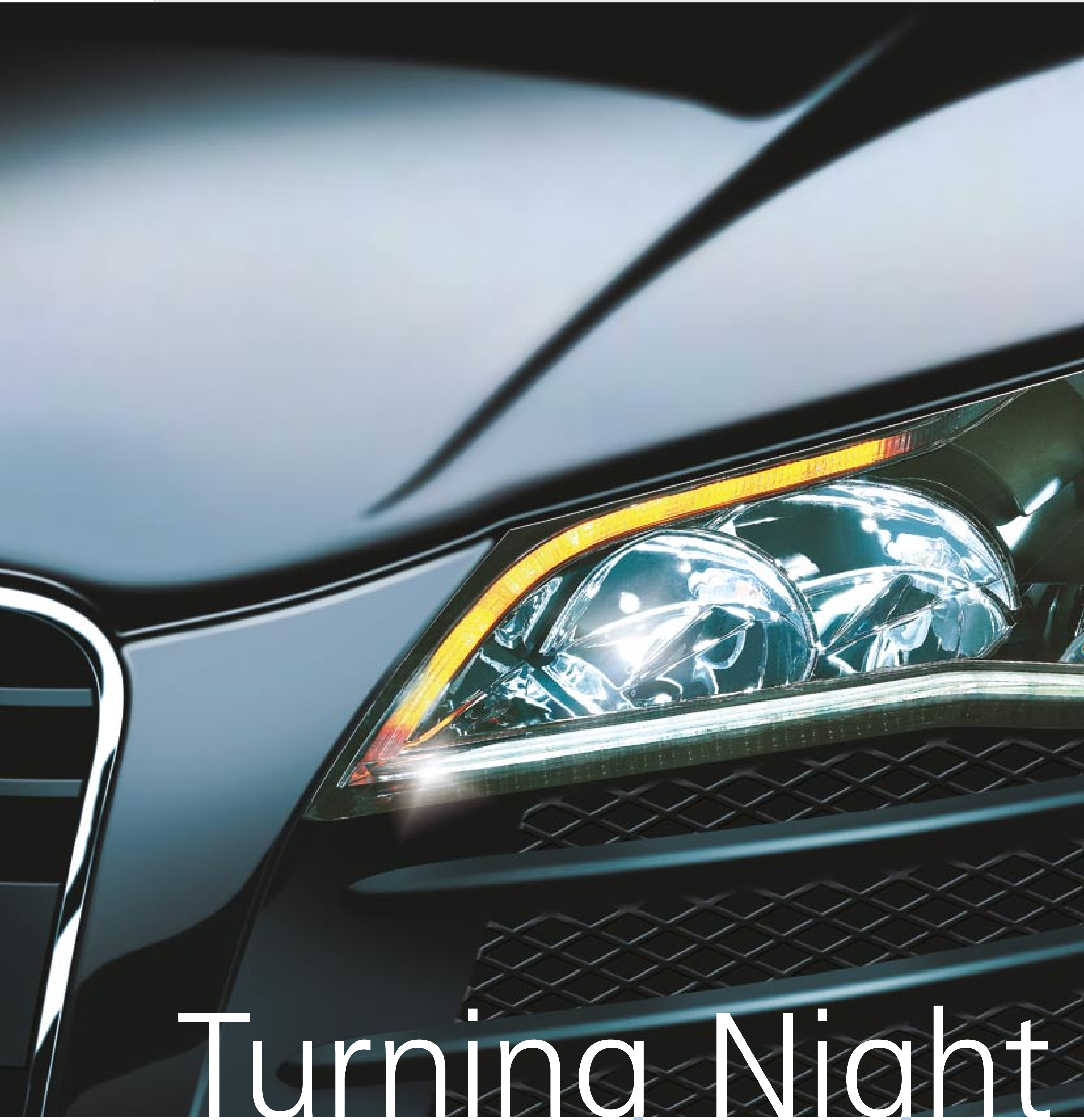
Decentralized Decoupling Control provides a simple, efficient, and automatable tool for controller design for coupled subsystems in continuous production. It achieves high to very high control performance quality: good reference tracking, good disturbance rejection, and great robustness for use even under widely varying production conditions. For a particularly smooth web run, Decentralized Decoupling Control should be installed at every defined nip section.

The feedforward torque control of the applicator roller greatly reduces the risk of a web break. The effect of applying and removing the coating knife (blade) can be compensated for almost completely.

Both control concepts and the simulation toolbox are generally suitable for use in continuous production processes (paper, foil industries, etc.). Our results show a considerably smoother web run due to stabilized web tensions that overcome the disadvantages of conventional control and minimize the risk of web breaks and associated idle times.

Christoph M. Hackl
Lehrstuhl für Elektrische Antriebssysteme
(EAT) (Institute for Electrical Drive Systems)
Technische Universität München
Germany

Beatrix Mair
Papiertechnische Stiftung
(The Paper Technology Specialists, PTS)
Munich
Germany



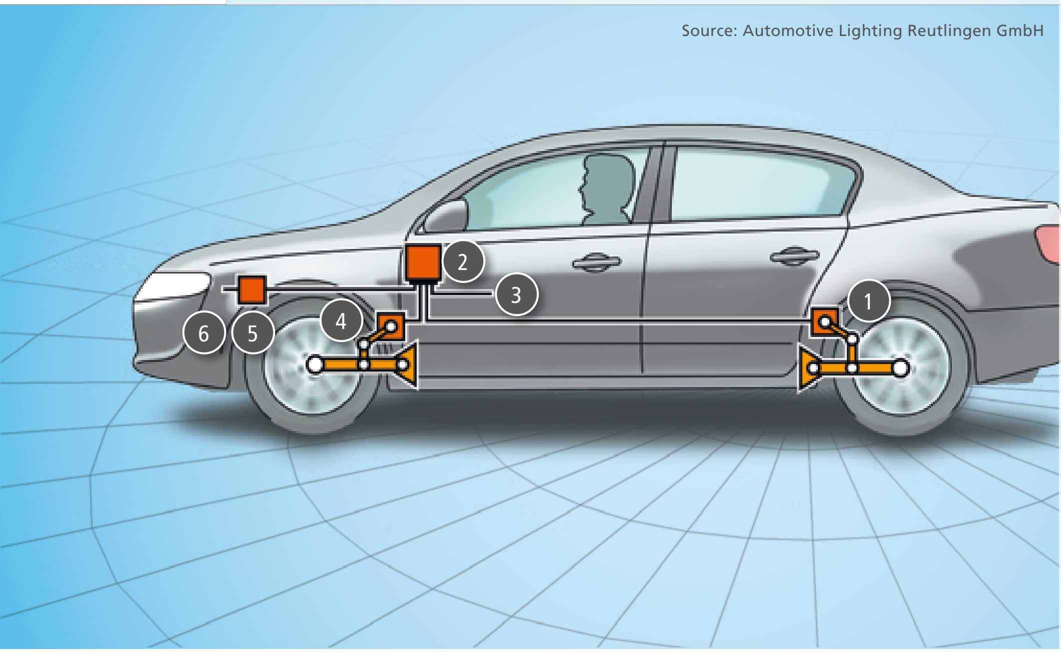
Turning Night into Day

Xenon and LED vehicle lights

Source: Automotive Lighting Reutlingen GmbH

We’ve come a long way from the gas lamps on the first motor vehicles to the xenon and LED headlights of today. But the goal is the still same: shine as much light as possible on the road, not at the sky, and never glare oncoming traffic.

Source: Automotive Lighting Reutlingen GmbH



Source: Automotive Lighting Reutlingen GmbH

1. Inter-axle sensor
2. ECU
3. Speed signal
4. Front axle sensor
5. Motor
6. Headlight

The components of the automatic dynamic automatic vertical aim control.

Light up the Road, not the Oncoming Traffic

Heavy vehicle loads, sudden acceleration and abrupt braking maneuvers cause vehicles to tilt longitudinally. To keep the maximum amount of light on the road, the angle of the headlights has to change to compensate for vehicle movements. This is what automatic headlight adjustment does.

Automatic Vertical Aim Control

At Automotive Lighting, we develop automatic vertical aim control systems (AVA) and also advanced frontlighting systems (AFS) for halogen and xenon headlights. Halogen headlights can be adjusted manually by means of a handwheel, but automatic adjustment is mandatory for xenon (high-intensity discharge) and LED headlights by law. This is because of their higher luminosity: If they are not adjusted properly, glare can be a hazard to oncoming traffic. Automatic AVA control uses either one or two axle sensors: one fitted to the rear axle and an optional one on the front axle. These capture the angle of the car body and compare it with the zero position defined by the OEM. If the two values differ, the sensors send data for corrective action to the headlight adjustment ECU. The ECU uses this data to calculate a new reference value so that the stepper motors can regulate the headlight level accordingly. Automatic vertical aim control can be static or dynamic.

- In static AVA, the vehicle angle caused by the vehicle load is measured and adjustment is performed before the vehicle moves off. Any changes that occur while the vehicle is moving, due to intense acceleration, braking, etc., are ignored.
- Dynamic AVA also compensates vehicle angles caused by braking and accelerating actions. For example, emergency braking can cause the vehicle to tilt forward

> “The DS1006 Processor Board from dSPACE has sufficient computing power to simulate complex and real-time-close models.”
>
> *Mathias Bako, Automotive Lighting Reutlingen GmbH*

The evaluation of measurement results was handled in ControlDesk.

> "The dSPACE hardware is exactly right for our purposes and was integrated very successfully into our test environment."
>
> *Mathias Bako, Automotive Lighting Reutlingen GmbH*

by up to 1.5°, shortening the range of the light cone from 100 m to 40 m. Stepper motors are used to control the headlights dynamically.

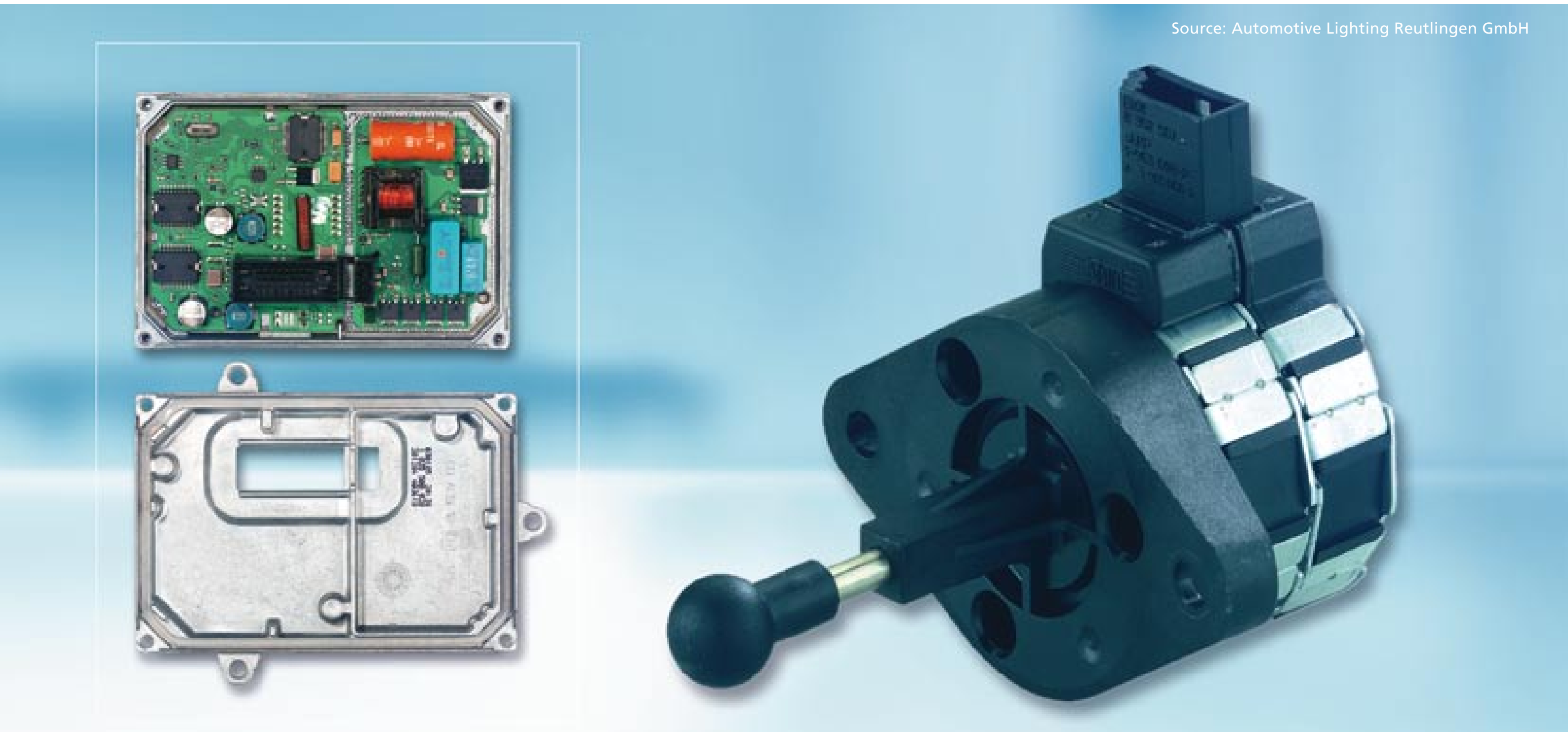
Advanced Frontlighting

Especially on winding country roads, improved visibility ensures greater safety, comfort, and driving pleasure. This motivated Automotive Lighting to develop advanced frontlighting systems, a new kind of dynamic directional light. An electronic control unit continuously evaluates various vehicle dynamics parameters such as steering angle, speed, and yaw rate. The system uses the evaluation results to move the low-beam headlights horizontally according to the radius of each road bend, ensuring that the road is optimally illuminated. This method provides up to 70 % more visibility compared with today's headlight systems.

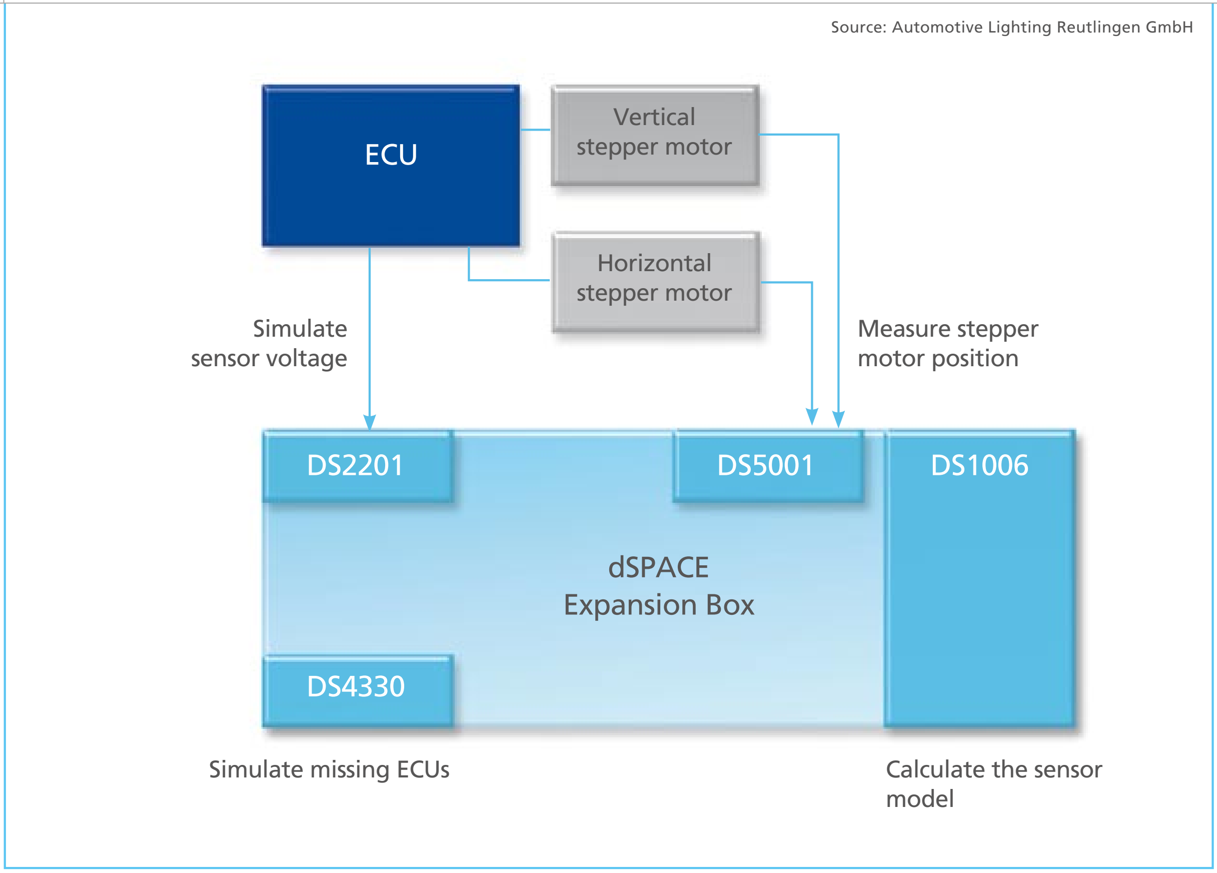
Test Bench for AVA and AFS

We use hardware-in-the-loop (HIL) simulation on our test bench to test whether the ECU and the stepper motors for the AVA and AFS functions are working correctly. Our ECU is connected to a dSPACE Extension Box. The hardware in this simulates the sensor system and checks the steps of the stepper motors in real time. To simulate the sensors, we created a MATLAB®/ Simulink® model that runs on a DS1006 Processor Board. The resulting sensor voltage, which indicates the vehicle's angle of tilt, is forwarded to the ECU via the DS2201 Multi-I/O Board. The ECU uses it to calculate the new reference actuating variable, which it then passes to the stepper motors. During a critical driving situation such as an abrupt braking maneuver, highly dynamic headlight control is vital. We therefore had strict requirements regarding the motor's testing precision. We can test the individual steps of the stepper motor in real time with the DS5001 Digital Waveform Capture Board. A LIN bus system is used for communication between the ECUs involved in automatic vertical aim control and for controlling the headlights via actuators. To achieve

The headlight positions are regulated via a stepper motor and our ECU AL box.



Source: Automotive Lighting Reutlingen GmbH


Source: Automotive Lighting Reutlingen GmbH

The dSPACE simulation and testing hardware is contained in a compact extension box.

Source: Automotive Lighting Reutlingen GmbH

greater test depth, one or more ECUs are simulated as LIN nodes with the DS4330 LIN Interface Board in another test stage (according to the V-cycle). The simulated LIN nodes behaved in exactly the same way as real bus nodes.

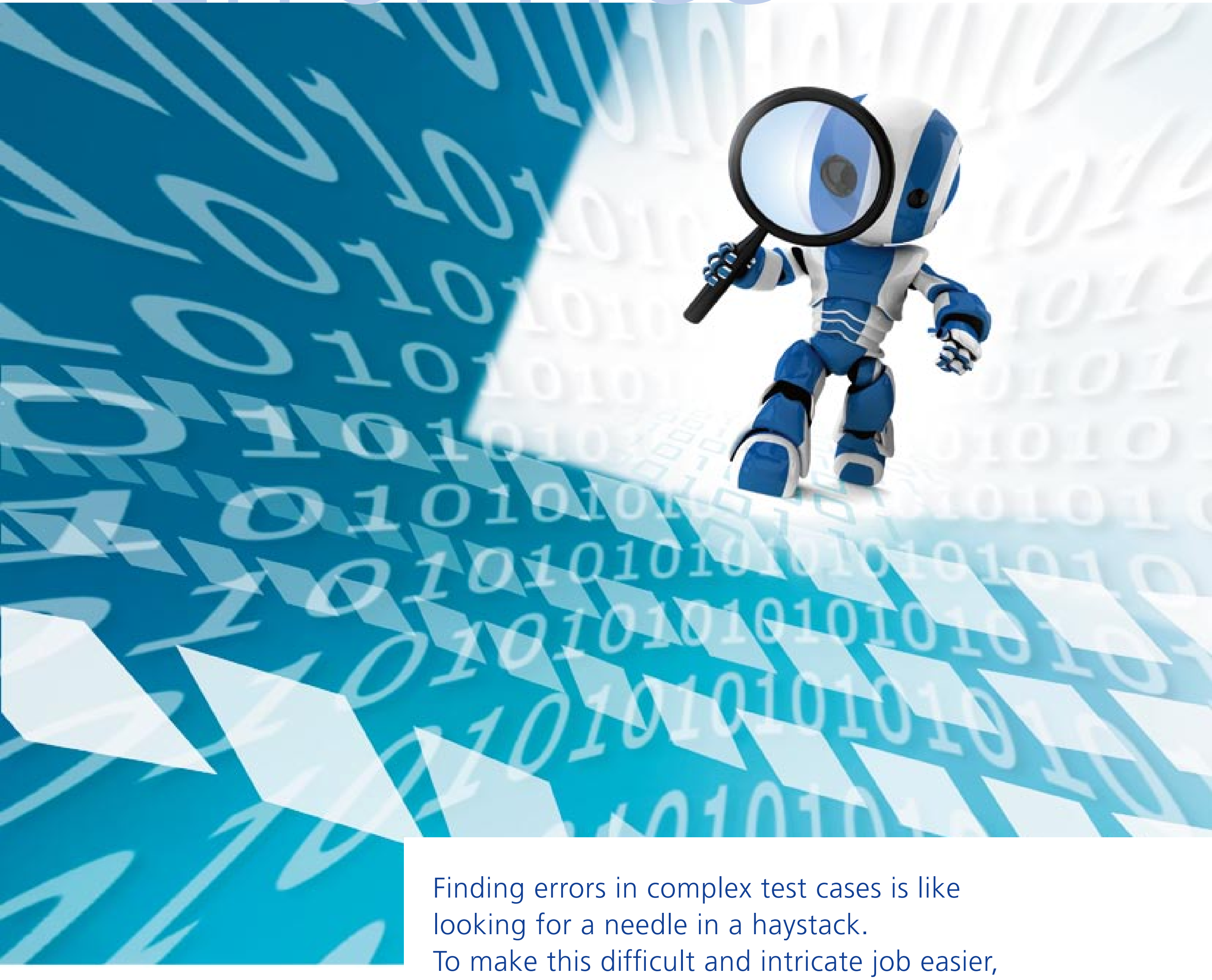
We managed and controlled the evaluation of measurement results with dSPACE ControlDesk. We used Python scripts to automate project-specific procedures and adjust them precisely to our requirements. This included actions such as starting ControlDesk, loading the application to the DS1006, running the simulation and reading out the measurement values.

Mathias Bako
Automotive Lighting Reutlingen GmbH
Germany

Outlook

Our test bench still has free capacity for extra computing power and extensions, so we aim to add further innovations in the future. For example, communication can be performed via FlexRay as well as via LIN and CAN. Another conceivable test setup would be to test the intensity of the light via optical feedback.

Completely Error-Free



Finding errors in complex test cases is like looking for a needle in a haystack. To make this difficult and intricate job easier, AutomationDesk 2.2 has a brand-new feature: the Debugger. This feature lets users execute and check their test sequences step by step, locating error sources and eliminating them.

> The Debugger makes trouble-shooting fast and efficient, allowing the targeted observation of potential error sources.

Test automation is a standard part of hardware-in-the-loop (HIL) simulation, and as ECU functions become more complex, so do test cases.

AutomationDesk Version 2.2, dSPACE's test automation software, facilitates the checking of a large volume of test cases. The Find Inconsistencies function performs static analyses before test execution begins. And with the new Debugger, test sequences can be checked and corrected step by step while they are still being implemented.

Test the Test

Test sequences are described graphically in AutomationDesk. They consist of a large number of single test steps called blocks. Before the tests are even executed, users can run the Find Inconsistencies function to find and correct errors. The function analyzes the test structure and all the defined data objects, references, and code fragments so that problems such as open references or syntax errors are detected automatically.

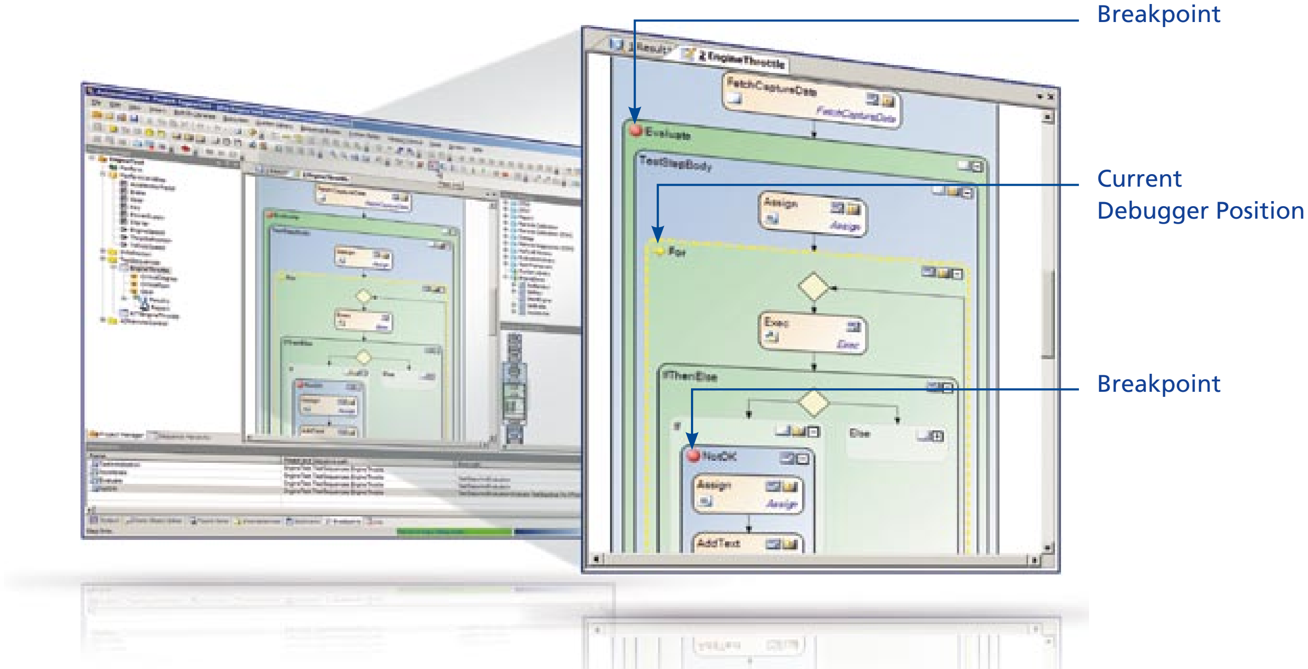
The next step is to run the test in offline mode to find errors in the test sequence without having to have an entire test environment available, such as an HIL simulator or calibration tools. This is done by defining default values to be returned when unavailable hardware or software is accessed. This procedure finds errors in the sequence or control structure such as erroneous conditions for If-Else branches or errors in loops, incorrect parameterizations, and run-time errors. When all these have been corrected, the test is executed on the HIL simulator. The offline mode means the HIL simulator is available for executing other tests during test development.

Debugging on the PC

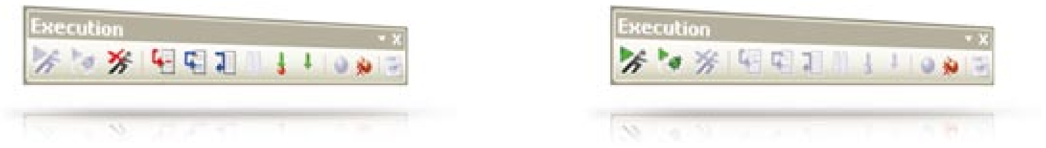
When unexpected test results or errors occur during test execution – offline on a PC or online on a HIL simulator – the Debugger's stepwise procedure helps pinpoint the cause.

Developers can use the Debugger to insert break-points at any desired positions in the test sequence. When the sequence is run in debug mode, execution automatically stops when it reaches a breakpoint. The Debugger's toolbar then provides the following options:

- Continue running the test sequence up to the next breakpoint.



The Debugger stops the test execution at each breakpoint.



The Debugger's toolbar lets users check their test sequences step by step.

- Execute the test sequence stepwise.
- Either go down one level in complex, hierarchical test blocks to execute that level, or run the whole current hierarchy level.

The current variable values can be viewed throughout the debugging process. If they deviate, they can be changed to allow sequence testing to continue with correct values. Thus, users can check a test step by step and correct it if necessary as they go along.

Efficient Processes

When debugging has finished, users return to the normal execution mode to restart the test sequence. Step-by-step execution supports fast, efficient error-finding by allowing specific potential error sources to be investigated. Using the Find Inconsistencies function, the offline mode and the Debugger speeds up test development, enhances the quality of tests, and improves the overall test process.

A Successful Combination

TargetLink 3.0: Redesigned Blockset and Model Referencing

The best of two worlds: The redesigned blockset in TargetLink 3.0, our production code generator, provides even closer MATLAB®/Simulink® integration in addition to TargetLink's reliable features. A wide range of third-party Simulink tools can be used for TargetLink models. TargetLink 3.0 also supports model referencing to facilitate modular, distributed development processes.

Interview

with Dr. Ulrich Eisemann, Product Manager TargetLink (dSPACE)

Mr. Eisemann, what innovations does TargetLink Version 3.0 offer?
Well, for one thing, we redesigned the TargetLink blockset. And we added support for model referencing. These are the two most important innovations.

How do they benefit users?
The main change in the blockset is the way it is implemented, giving users considerably improved tool integration and also enhanced process integration as well. From the outside, TargetLink still has the same look & feel, and familiar processes can still be performed with TargetLink 3.0, of course. Model referencing plays an important role wherever extensive model designs are developed by large teams. It substantially improves modular, distributed work in large work groups.

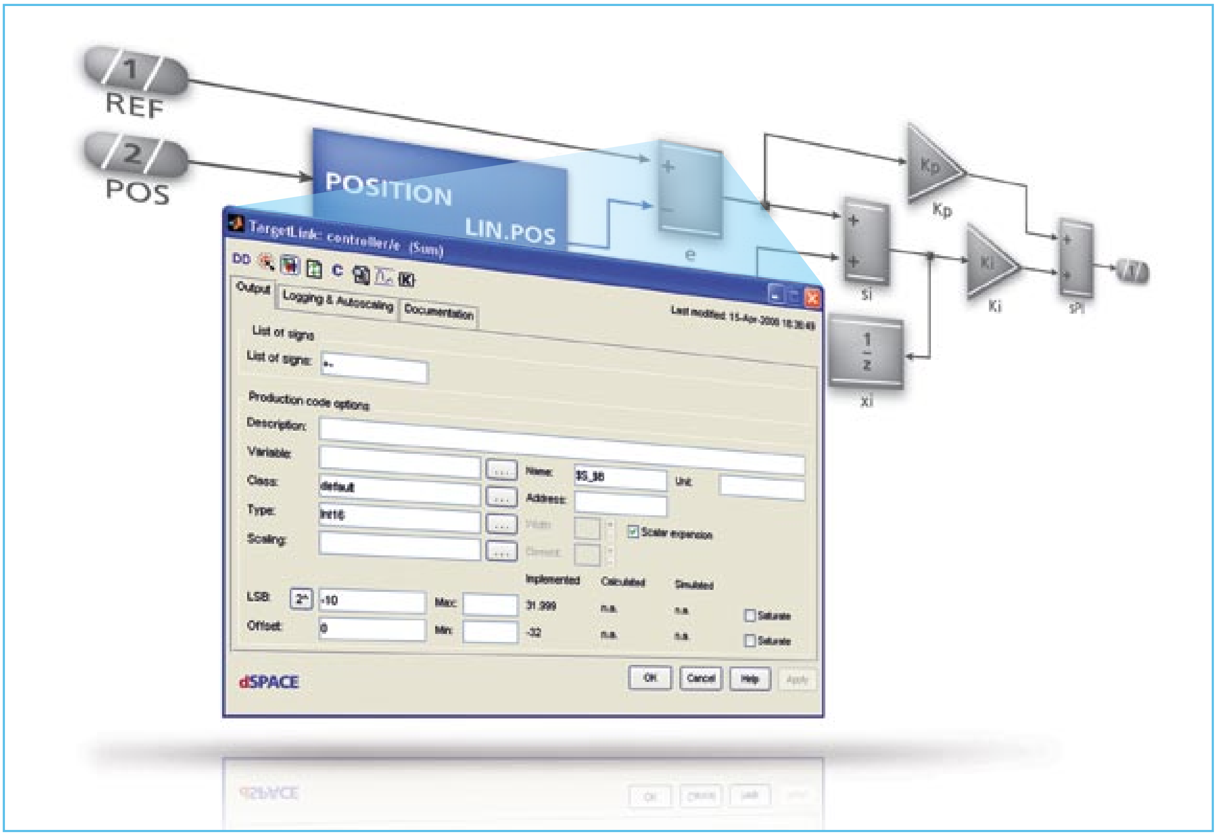
Can existing TargetLink models still be used with TargetLink 3.0?
Yes. Automatic upgrades of older TargetLink models and libraries were an important goal in developing the new version. Our customers have large collections of models, and obviously they want to continue using them under TargetLink 3.0. So there is an automatic upgrade routine for migrating older models. We kept all changes to the TargetLink API to an absolute minimum. This makes it easier for our customers to migrate their tool chains.

Mr. Eisemann, thank you for talking to us.

Tighter MATLAB/Simulink Integration

The new blockset in TargetLink 3.0 is a symbiosis between the proven features of previous TargetLink blocks and those of Simulink blocks (fig. 1). This gives TargetLink users the best of both worlds: On the one hand, they benefit from TargetLink's powerful, much appreciated features such as easy-to-use block dialogs, integrated signal logging and plotting for all simulation modes (model-in-the-loop, software-in-the-loop, processor-in-the loop), and overflow warnings. And on the other hand, they can use a wide range of third-party Simulink tools directly on TargetLink models. From a technical point of view, the TargetLink blocks in the new design are normal Simulink blocks but with a mask for storing TargetLink data and the proven TargetLink dialogs. Benefits of the new TargetLink 3.0 blockset are:

- Greatly increased compatibility with Simulink toolboxes and other vendors' tools. For instance, test vector generation with a tool like Reactis® or model coverage can be directly applied to TargetLink models.



TargetLink: controller/e (Sum)

Output | Logging & Autoscaling | Documentation

Last modified: 15-Apr-2008 18:28:49

List of signs

List of signs: +-

Production code options

Description:

Variable:

Name: $S_$B

Unit:

Class: default

Address:

Type: Int16

Scaling:

☑ Scalar expansion

			Implemented	Calculated	Simulated	
LSB:	2^ -10	Max:	31.999	n.a.	n.a.	□ Saturate
Offset:	0	Min:	-32	n.a.	n.a.	□ Saturate

Fig. 1: The TargetLink blocks in the redesigned blockset still look the same and are still handled in the same way, even though TargetLink and Simulink ports have been merged.

- Easier prototyping of TargetLink models using RTI and ControlDesk
- Greater efficiency: Shorter model loading and initialization times, and faster model-in-the-loop (MIL) simulations
- Simpler transition from Simulink to TargetLink with the new model preparation feature
- Identical MIL simulation behavior in Simulink and TargetLink

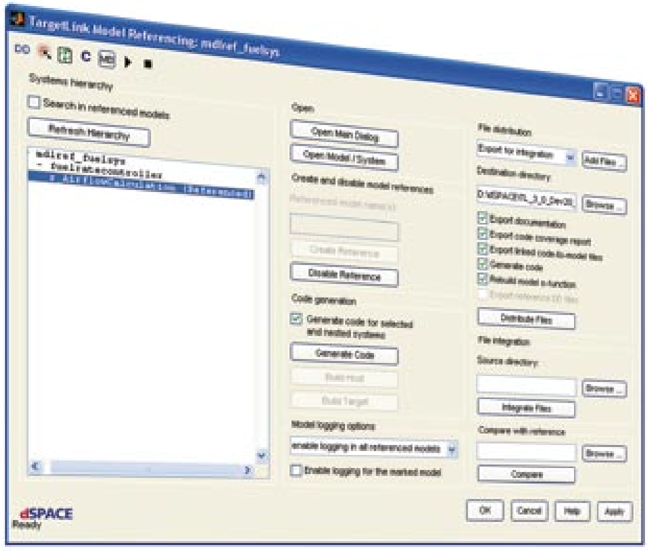
> The redesigned blockset in the TargetLink 3.0 production code generator provides even closer MATLAB/Simulink integration in addition to the proven TargetLink features.

Distributed Development: Model Referencing

TargetLink 3.0 supports model referencing, a feature in MATLAB/Simulink that decisively simplifies the modular development of model-based designs (fig. 3). A large TargetLink model can be partitioned into submodels that can be developed, versioned, and simulated separately. Incremental code can also be generated for the individual referenced models. Modules that are developed separately can then be brought together in one integration model. Benefits of model referencing are:

- Enhanced handling of very large models
- Modular development of subfunctionalities
- Better teamwork in large groups: Each developer can be assigned a clearly defined subfunctionality.

The TargetLink Model Referencing Control Center is a new, intuitive user interface that supports users in developing individual models and also in integrating them to make one large model. A particularly useful feature is that references to models can be dissolved temporarily and then restored again afterwards.



TargetLink Model Referencing: mdlref_fuelsys

Systems hierarchy

☐ Search in referenced models

Refresh Hierarchy

```
mdlref_fuelsys
- fuelratecontroller
  + AirflowCalculation (Referenced)
```

Open

Open Main Dialog

Open Model / System

Create and disable model references

Referenced model name(s):

Create Reference

Disable Reference

Code generation

☑ Generate code for selected and nested systems

Generate Code

Build Host

Build Target

Model logging options

enable logging in all referenced models

☐ Enable logging for the marked model

File distribution

Export for integration

Add Files ...

Destination directory

D:\dSPACE\TL_3_0_Dev20

Browse ...

☑ Export documentation

☑ Export code coverage report

☑ Export linked code-to-model files

☑ Generate code

☑ Rebuild model s-function

☐ Export reference DD files

Distribute Files

File integration

Source directory:

Browse ...

Integrate Files

Compare with reference

Browse ...

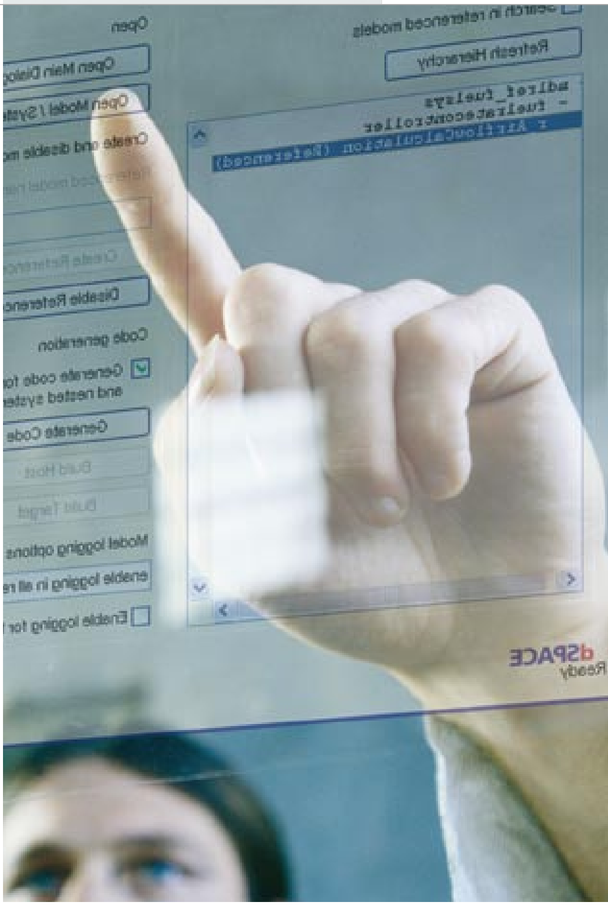
Compare

OK Cancel Help Apply

dSPACE

Ready

Fig. 2: TargetLink 3.0 provides a special graphical interface for distributed work on referenced models.



MATLAB Version Support and Other Features

TargetLink 3.0 supports five MATLAB releases altogether: MATLAB R2006a+, R2006b, R2007a+, R2007b+ and R2008a. This gives users maximum freedom.

Other new features such as the ability to inherit vector widths, and flexible code generation at the Simulink-Stateflow interface round off this newest version of TargetLink. ■

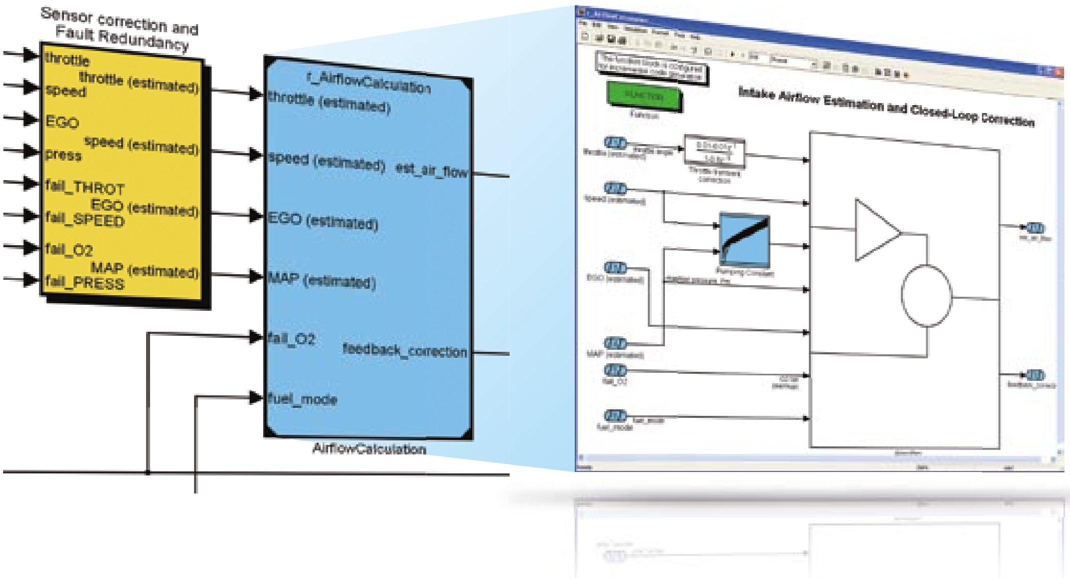
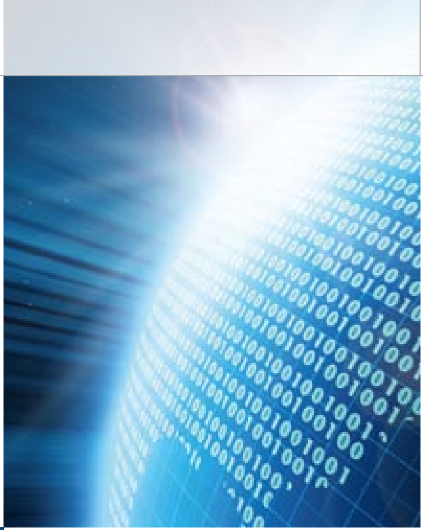


Fig. 3: Using model referencing to design TargetLink models in modular form.

Communication under Control

RTI CAN MultiMessage Blockset tests large CAN setups



In our last issue, we looked at multi-tool workflows for the CAN bus in the dSPACE tool chain. This second article describes in detail how the RTI CAN MultiMessage Blockset supports the testing of large-scale CAN setups. With its easy-to-use graphical interface and a wide range of configuration options, the blockset proves its practical value time and time again.

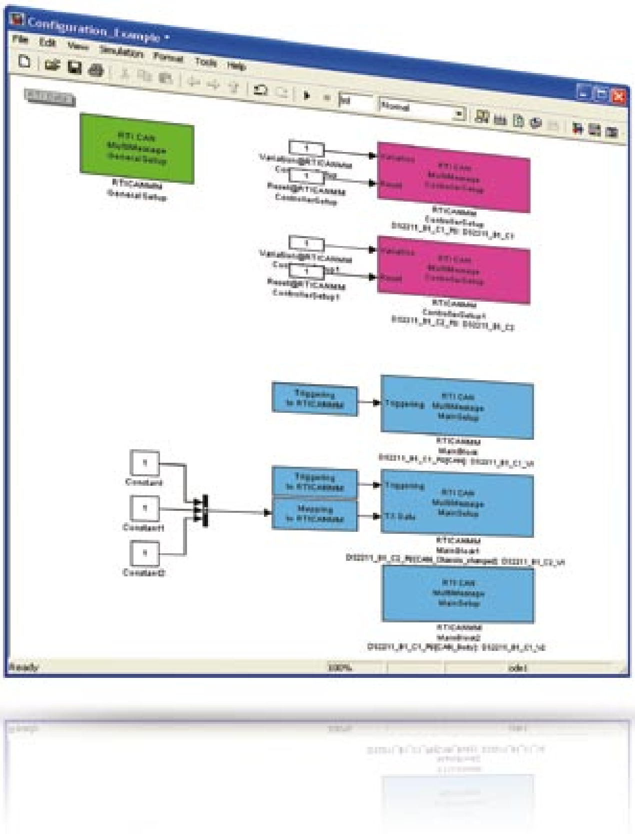

Configuration_Example *
File Edit View Simulation Format Tools Help
Normal
RTI CAN MultiMessage GeneralSetup
RTICANMM GeneralSetup
Variation@RTICANMM ControllerSetup
Reset@RTICANMM ControllerSetup
Variation
Reset
RTI CAN MultiMessage ControllerSetup
RTICANMM ControllerSetup
DS2211_B1_C1_P8: DS2211_B1_C1
RTICANMM ControllerSetup1
DS2211_B1_C2_P8: DS2211_B1_C2
Triggering to RTICANMM
Triggering
RTI CAN MultiMessage MainSetup
RTICANMM MainBlock
Mapping to RTICANMM
TX Data
Constant
Constant1
Constant2
RTICANMM MainBlock1
RTICANMM MainBlock2
Ready
100%
ode1

Fig. 1: Bus configurations performed with the RTI CAN MM Blockset.

Typical use cases for the RTI CAN MultiMessage Blockset range from setting up simple restbus simulations to defining extensive tests on CAN network communication. Using a single Simulink block in the real-time model, developers can control, configure, and edit large numbers of CAN messages quickly and individually. Once these configurations have been made, they can be reused in downstream process steps, for example with experiment and automation tools.

Starting with a Communication Matrix

A communication matrix (usually a DBC file) is the basis for configuring communication tests with the RTI CAN MultiMessage Blockset. The information in the matrix – cycle times, assignments between messages and ECUs, signal thresholds, defaults, etc. – can be used directly for configuring a restbus simulation. Whenever the DBC file is updated during the course of a project, only a "delta" configuration is necessary. This is especially useful if the databases or integration stages change frequently during vehicle development.

Another useful feature is the ability to use different restbus configurations for one and the same CAN bus. The blockset can be used to assign up to 20 versions of a test configuration to a single CAN controller. The different database or configuration versions can be changed during run time by an assigned model variable or an automatable trace variable.

Simple Restbus Simulations

When the data has been imported, a basic restbus simulation can be created by simply choosing the Rx (receive) and Tx (transmit) messages for specific selected ECUs.

This static simulation does not yet allow exchange data with the plant model, but works with defaults instead. Even so, such simulations, where fixed values are sent, are often sufficient for running tests when ECUs are first put into operation.

If more comprehensive simulations are required, there is a wide range of test functions available. With versatile settings, these support not only the error-free simulation of CAN communication (good tests), but also the insertion of specific errors into the communication process (bad tests). The blockset's functions are subdivided into message-level and signal-level functions for easy use.

Tests at Message Level

The message-level test functions control aspects such as message transmission and the failure of specific messages or even whole ECUs. One example of message transmission manipulation is the function for changing cycle times.

The "Message cycle time defaults" functionality is used to set cycle times and delay times for the messages to be transmitted. Users can either take the information contained in the database or define it themselves in the graphical view.

If they choose to define data, they can specify the cycle time for the messages and also a delay time that has to expire before a specific

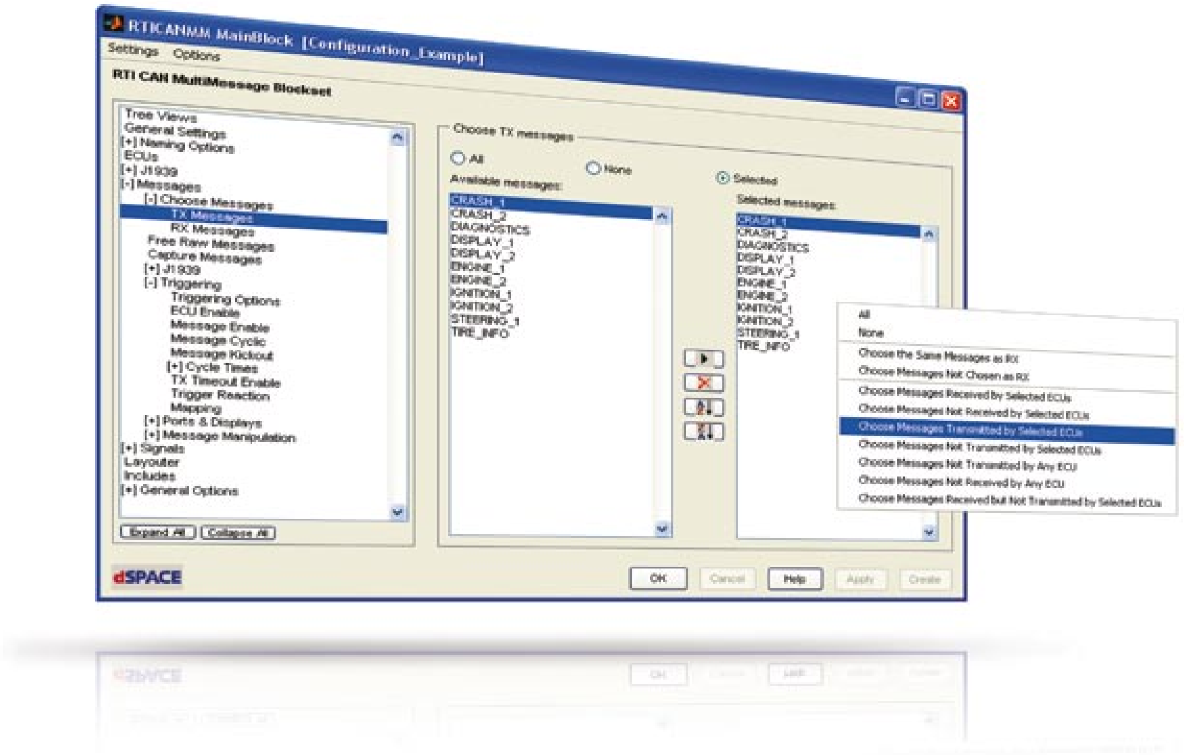


Fig. 2: Selecting Rx (receive) and Tx (transmit) messages for a basic restbus simulation.

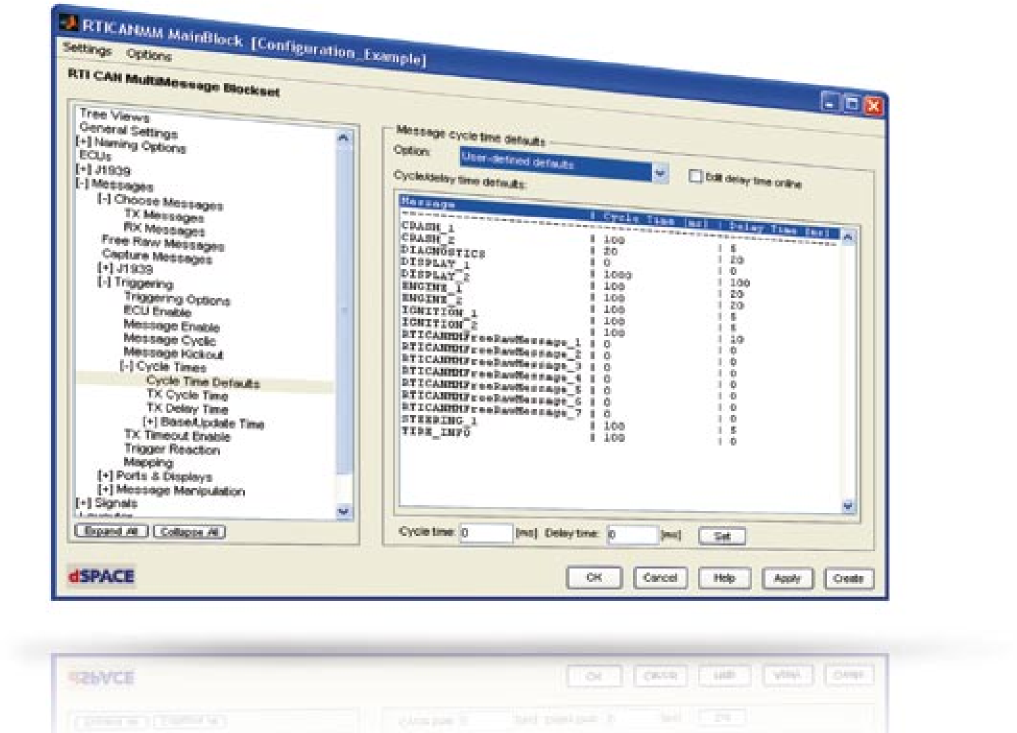
message is transmitted during the test. What is particularly useful is that this value can be changed during test execution, either by a model variable or by a trace variable.

Not only the transmission itself is modifiable. Users can also manipulate specific aspects of the messages such as their IDs, lengths, or contents. One frequently used feature is the option to define or modify the checksums for messages and the signals they contain in order to check data integrity. This is done in two steps. Step One: Define the messages classes or cases that require different checksum algorithms. For example, one class can be defined to comprise the engine ECU's messages to the transmission ECU, while a second class contains messages to be received by the ESP ECU. The blockset now generates header files for these classes as frameworks for users to integrate their own checksum algorithms – simple look-up-tables – to be used for the tests. Step Two: Assign messages or even individual signals to the defined classes. The easiest way of doing this is to use settable filters, which concludes the configuration of the checksum calculation.

Tests at Signal Level

The blockset and its powerful test function also support testing at signal level. There is the usual option for transmitting static values, plus an option for transmitting signals from the real-time model. Another very useful feature is the possibility to assign signal names from the database and signal names from the real-time model to each other.

All the typical functions for defining contents and manipulating signals are available: counters, error values, parity signals, etc. One such function is "dynamic signal values", which fixes the value to be transmitted for a predefined number of transmissions. When these transmissions have been completed, the previous signal source or manipulation becomes effective again.



Message cycle time defaults

Option: User-defined defaults

☐ Edit delay time online

Cycle/delay time defaults:

```
Message                      | Cycle Time [ms] | Delay Time [ms]
-------------------------------------------------------------------
CRASH_1                      | 100             | 5
CRASH_2                      | 20              | 20
DIAGNOSTICS                  | 0               | 0
DISPLAY_1                    | 1000            | 100
DISPLAY_2                    | 100             | 20
ENGINE_1                     | 100             | 20
ENGINE_2                     | 100             | 5
IGNITION_1                   | 100             | 5
IGNITION_2                   | 100             | 10
RTICANMMFreeRawMessage_1     | 0               | 0
RTICANMMFreeRawMessage_2     | 0               | 0
RTICANMMFreeRawMessage_3     | 0               | 0
RTICANMMFreeRawMessage_4     | 0               | 0
RTICANMMFreeRawMessage_5     | 0               | 0
RTICANMMFreeRawMessage_6     | 0               | 0
RTICANMMFreeRawMessage_7     | 0               | 0
STEERING_1                   | 100             | 5
TIRE_INFO                    | 100             | 0
```

Cycle time: 0 [ms] Delay time: 0 [ms]

Fig. 3: Defining message cycle times and transmission delays.

Glossary

DBC file – Database container; file format to describe a CAN communication network.

Restbus simulation – Reproduces messages from ECUs that are not present as real devices in the network.

Communication matrix – Describes all the nodes (ECUs) involved in communication and the data exchange between them.

Trace variable – Reference to a model variable that can be used for experimentation and automation.

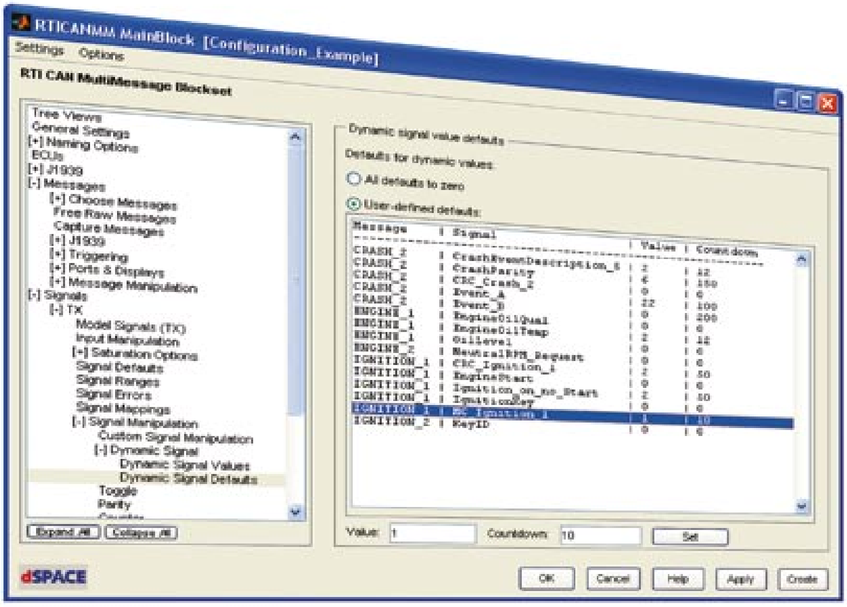
Extra Options

The messages and signals that are configured are not only ones defined in the database. The RTI CAN MultiMessage Blockset also provides an option for creating messages outside the database. These special messages can be changed during run time, so they are useful for sending experimental messages during diagnostic tests, ECU commissioning, etc.

Multiple Bus Support

With its versatile functionality and multi-tool handling (see dSPACE Magazin 1/2008) , the RTI CAN MultiMessage Blockset makes it easy to handle and test even complex CAN setups. Once a system has been configured, it is available for comprehensive experiments in ControlDesk or AutomationDesk and can easily be adapted to new data versions. ■

For details on using the RTI CAN MultiMessage Blockset in a development project, turn to pages 26 – 31.



RTICANMM MainBlock [Configuration_Example]
Settings
Options
RTI CAN MultiMessage Blockset

```
Tree Views
General Settings
[+] Naming Options
ECUs
[+] J1939
[-] Messages
    [+] Choose Messages
    Free Raw Messages
    Capture Messages
    [+] J1939
    [+] Triggering
    [+] Ports & Displays
    [+] Message Manipulation
[-] Signals
    [-] TX
        Model Signals (TX)
        Input Manipulation
        [+] Saturation Options
        Signal Defaults
        Signal Ranges
        Signal Errors
        Signal Mappings
        [-] Signal Manipulation
            Custom Signal Manipulation
            [-] Dynamic Signal
                Dynamic Signal Values
                Dynamic Signal Defaults
            Toggle
            Parity
```

Dynamic signal value defaults

Defaults for dynamic values:

○ All defaults to zero

● User-defined defaults:

```
Message      | Signal                       | Value | Count down
-----------------------------------------------------------------
CRASH_2      | CrashEventDescription_5      | 2     | 42
CRASH_2      | CrashParity                  | 6     | 150
CRASH_2      | CRC_Crash_2                  | 0     | 0
CRASH_2      | Event_A                      | 22    | 100
CRASH_2      | Event_B                      | 0     | 200
ENGINE_1     | EngineOilQual                | 0     | 0
ENGINE_1     | EngineOilTemp                | 2     | 42
ENGINE_1     | Oillevel                     | 0     | 0
ENGINE_2     | NeutralRPM_Request           | 0     | 0
IGNITION_1   | CRC_Ignition_1               | 2     | 50
IGNITION_1   | EngineStart                  | 0     | 0
IGNITION_1   | Ignition_on_no_Start         | 2     | 50
IGNITION_1   | IgnitionKey                  | 0     | 0
IGNITION_1   | MC_Ignition_1                | 1     | 10
IGNITION_2   | KeyID                        | 0     | 0
```

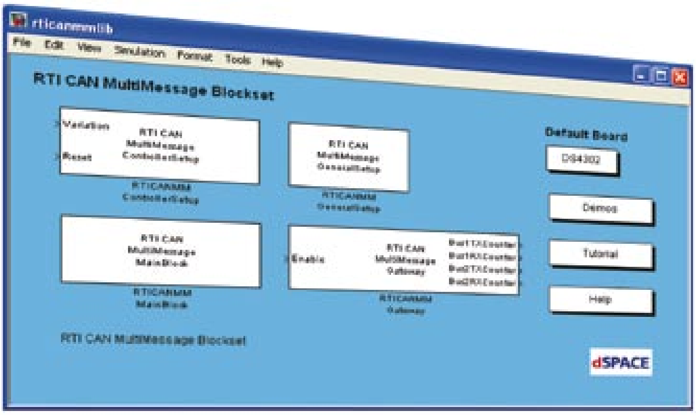
Value: 1 Countdown: 10

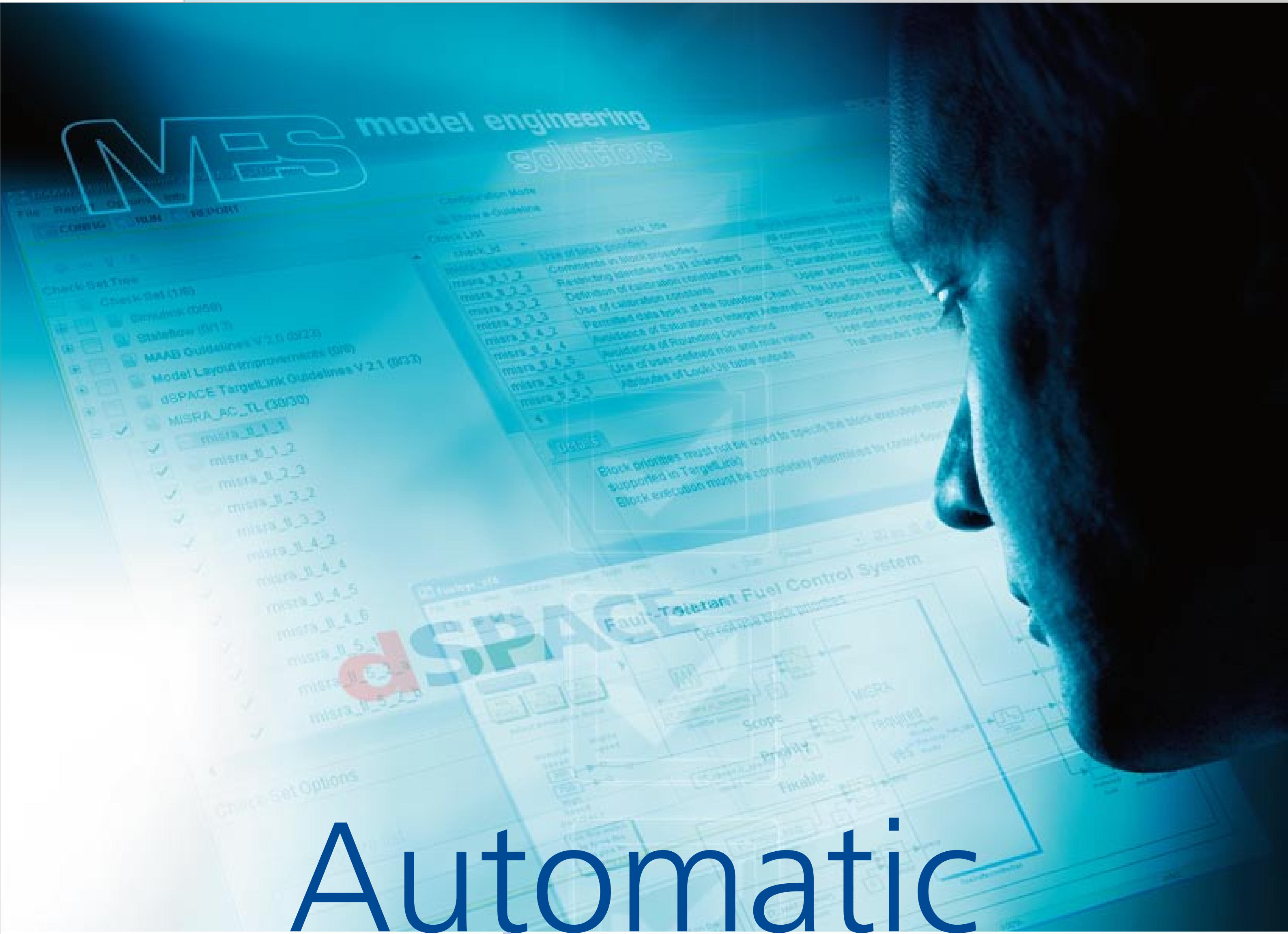
The transmission controls and signal manipulations can be changed during test execution.

Fig. 4: Graphical page for specifying dynamic transmit signals.

The RTI CAN MultiMessage Blockset:

The RTI CAN MultiMessage Blockset is used for combining dSPACE systems with CAN communication networks and configuring the networks. It is an extension to dSPACE's implementation software Real-Time Interface (RTI). The blockset can be used in rapid control prototyping applications and also in hardware-in-the-loop applications. Users have an intuitive user interface with extensive configuration options for CAN communication. Even complex CAN setups can be handled easily with the RTI CAN MultiMessage Blockset, since a very large number of CAN messages can be controlled via a single Simulink block. This has a positive effect on the model size and shortens code generation and build times. The blockset also provides import options for DBC, MAT and FIBEX files.

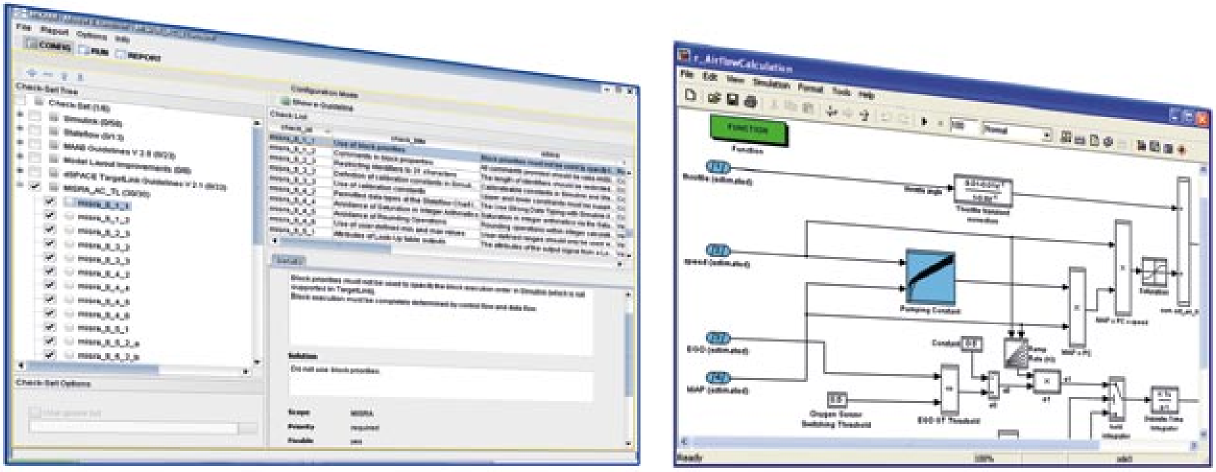



MES model engineering solutions
Check Set Tree
MAAB Guidelines V 2.0
dSPACE TargetLink Guidelines V 2.1
MISRA_AC_TL (30/30)
dSPACE
Fault-Tolerant Fuel Control System
Check Set Options

Automatic Compliance Checks

Fully automated checks for TargetLink models

In model-based design, modeling guidelines play a vital role in ensuring design quality. The Model Examiner from Model Engineering Solutions (MES) automatically checks whether TargetLink/Simulink® models comply with modeling guidelines. Now it also includes special automatic model checks for the MISRA TargetLink guidelines (MISRA-AC-TL).



The Model Examiner (left) automatically checks models against the TargetLink MISRA guidelines.

Functional Safety in the Model

Modeling guidelines are extremely valuable in efficient, model-based development. They are used to ensure that models meet all the requirements regarding transparency, functional safety, process integration, and code efficiency. Compliance with modeling guidelines and, in connection with that restriction to a reliable language subset is essential, particularly where safety-critical functions are developed according to standards such as IEC 61508 or the future ISO 26262. One answer to this is the MISRA TargetLink Guidelines (MISRA-AC-TL), which define modeling requirements with respect to functional safety (see dSPACE Magazine 1/2008). dSPACE and many of its customers also have company-specific guidelines for TargetLink. And in view of the sheer size of today's models and the large number of guidelines, it can be very difficult to ensure compliance without tool support. Moreover, not every function or software developer is an expert on model-based design – another good reason for using a tool.

MES Model Examiner 1.2

The Model Examiner from MES not only lets TargetLink users check for compliance with MAAB guidelines (The MathWorks Automotive Advisory Board); it also provides dedicated TargetLink support. The new version 1.2 of the Model Examiner incorporates the dSPACE Target Link guidelines as well as the official TargetLink guidelines issued by MISRA (Motor Industry Software Reliability Association). The Model Examiner checks whether a model complies with guidelines and issues a warning if a rule has been violated. It can also perform automatic correction and generate documentation. Project-specific rules can also be defined and applied to TargetLink/Simulink models. Developers can run a model analysis from the Model Examiner's own user interface or by calling scripts. They can therefore easily integrate the Model Examiner into an existing tool chain. Checks that are included with the Model Examiner can also be loaded into the Model Advisor. The Model Examiner's API allows a company-specific infrastructure to be integrated. If the Model Examiner does not yet support a particular guideline, users can add tool-chain-specific extensions to it in the form of additional scripts. The Model Examiner is available from Model Engineering Solutions.

> The Model Examiner checks whether a model complies with guidelines and issues a warning if a rule has been violated.

Looking Ahead

dSPACE and Model Engineering Solutions are cooperating on the further development of the Model Examiner to provide optimum support for TargetLink users. The main focus is on supporting the greatest possible number of available rules, and on enhancing the Model Examiner whenever new versions of TargetLink become available. ■

www.model-engineers.com
info@model-engineers.com

Demanding Student Competitions on Vibration Control

As part of the Earth & Space 2008 conference from March 3-5, 2008 in Long Beach, California, USA, the Smart Materials and Structures Laboratory at the University of Houston hosted two parallel competitions on vibration control: one for undergraduates, another for graduates. The objective was to design a controller which would dampen the oscillations of a flexible beam. In addition to the flexible beam with piezoceramic sensors and actuators (made possible by an National Science Foundation award, DUE 0442991), a TREK amplifier, a PC with MATLAB®/Simulink® and a dSPACE system were also available for the participants to use. With these devices and development tools, the algorithms could be created by using models as a base, and performed in real-time on the dSPACE system. The control signal of the implemented algorithms was sent via the TREK amplifier to the piezoceramic actuators in order to counteract the beam's vibrations. Altogether, the competition lasted three days. The first day gave each team one hour to record the sensor data for designing the control. On the second day, the control could be tested for up to one hour. The third day allowed 30 minutes for final optimizations on the control. The competition for the doctoral candidates was more difficult because sensor noise, external excitation and mass uncertainty were also involved. A panel of experts graded the projects on the settling time, chattering effects and overshoot.Students from the University of Akron, UC Berkley, University of Houston, Prairie View A&M, Rice University and Washington University took the top three places in the undergraduate competition; the top three teams in the graduate competition came from Rice University and the University of Houston. dSPACE Inc. and TREK Inc. donated the student awards:
First Prize $ 500, Second Prize $ 200, Third Prize $ 100.
http://egr.uh.edu/smsl/conference.html

Crankshaft Sensors That Detect Rotation Direction

The DS2211 HIL I/O Board has a new feature for simulating crankshaft sensors that detect the direction of rotation. The configuration for this is performed in Real-Time Interface, the software for implementing MATLAB®/Simulink®/Stateflow® models on dSPACE hardware. Support for the SAE standard J2716 (SENT) will also be available soon. This is an alternative to LIN for transmitting high-resolution sensor data to the ECU.

The DS2211 HIL I/O Board is the central I/O board for hardware-in-the-loop simulation. It is tailored to the simulation and measurement of automotive signals – particularly in engine and vehicle dynamics applications – and combines a variety of typical HIL I/O signals on one board.

Correction

In the article "Safe Modeling" shown in the 1/2008 issue, the logo of the Motor Industry Software Reliability Association (MISRA) was used. This should not lead to the impression that MISRA expresses recommendations for products and services that dSPACE offers. We apologize if this impression arose.
The Editors

dSPACE direct

Would you like up-to-the-minute information on products, releases and events? Then why not subscribe to our monthly e-mail newsletter, dSPACE direct.
www.dspace.com/dspace_direct

Request information

Please check the corresponding field on your response card and return it

- By mail
- By fax to +49 52 51 6 65 29
- or request information via our Website: *www.dspace.com/magazin*

Your opinion is important. Please send your criticism, praise, or comments to *dspace-magazine@dspace.com* – thank you!

Jobs

Are you an engineer who is just graduating?
Or are you looking for new professional challenges?
Then come and join our team in Paderborn, Munich or Stuttgart, Germany; Paris, France; Cambridge, United Kingdom; Tokyo, Nagoya, Japan or Detroit, MI, USA!
Due to our continuous growth, dSPACE is looking for engineers in

- Software development
- Hardware development
- Applications
- Technical sales
- Product management

Current offers at
www.dspace.com/jobs

www.dspace.com
dSPACE Simulator
Automatic Report Generation
AutomationDesk can automatically generate report documentation
based on the test results. You can select from a library of report test steps
(such as AddText, AddTable, or AddPicture) to specify the report content

System Architecture | Rapid Prototyping | ECU Autocoding | **HIL Testing** | ECU Calibration

Hardware-in-the-Loop Simulation
The best for every test

To err is human. But errors don't belong on the road. Our dSPACE hardware-in-the-loop (HIL) simulators offer the ultimate in reliability and precision. Whether it's function testing, system testing, or network testing, no ECU error goes undetected. Ever. 24 hours a day, 7 days a week.

HIL simulators shorten the testing process even when testing depth increases. They ensure quality. Save money. And help you stay ahead.

Why do almost all major automobile manufacturers use dSPACE for testing? Lots of reasons. We have the most experience. We offer turnkey systems including mathematical models for any size and domain. We support automated testing, and provide unparalleled engineering support.

When it comes to accelerating your success, dSPACE HIL passes every test.

Embedded Success dSPACE