Highly automated and, even more so, autonomous driving significantly increase the requirements for an early and automatable validation process, because the test kilometers required for these validations far exceed the possibilities of real-time tests. The solution is software-in-the-loop (SIL) simulation. In contrast to testing real ECU components, this approach tests only the software part of the ECU, an option that is already available in the function development phase. However, to do this, function code must be compiled and executed.

Working with Virtual ECUs
The code is executed on the VEOS simulation platform. Among other things, VEOS simulates virtual ECUs (V-ECUS): Their code corresponds as closely as possible to production code, provided it is independent of the target platform hardware. If code is developed in accordance with the AUTOSAR standard, it becomes platform-independent. For example, AUTOSAR defines a layer model for modularizing the software components of an ECU, which also includes standardized interfaces for basic functionalities. This way, ECU code can be implemented independently of the hardware.

Generating AUTOSAR-Compliant Software
With SystemDesk, dSPACE offers an authoring and system generation tool that facilitates the integration of ECU software components based on AUTOSAR descriptions. Everything can be integrated, whether it is individual components of the application software with the function code or the complete code including all required AUTOSAR basic software modules. Additionally, SystemDesk configures and generates an operating system specifically for simulation purposes. This makes it possible to accurately simulate the behavior of the ECU up to the configuration of different operating system tasks. Generating potentially missing basic software (or integrating externally supplied modules) also enables connecting to simulated bus systems, such as automotive Ethernet.

Service-Based Communication
Today, the AUTOSAR Classic Platform is the tool of choice when developing software for highly efficient control units. However, highly automated and autonomous driving require different framework conditions. In this specific use case, the communication between functions is no longer technically defined in advance and integrated into the generated code of a run-time environment (RTE). Instead, the definition specifies only who communicates with whom. The actual connec-
a validation challenge if some of these ECU
can be updated via wireless con-
nections after the vehicle has been del-
ered. New functions can also be added later (over-the-air update). The required software architectures are
described by using the AUTOSAR Adaptive Platform, which notably differs from the Classic Platform. How-
ever, some of the structure still remains. For example, the platform is still divided into an application layer and basic services, which each ECU must provide. It is also still independent of hardware interfaces thanks to an operating system that provides standardized interfaces (Portable Operating System Interface, POSIX for short).

Efficient software development for the AUTOSAR Adaptive Platform with the dSPACE tool chain.