

Dr. Laura Beermann, Dr. Dirk Ziegenbein, Robert Bosch GmbH

Timing-aware Software in the Loop Simulation of Automotive Software

The complexity of automotive electrical/electronic (E/E) architectures is growing rapidly. Future E/E architectures will have to integrate highly automated and autonomous driving technologies, extensive infotainment systems, connectivity solutions, and electrification. At the same time, SW and HW release cycles are being decoupled, leading to more frequent SW releases. These trends necessitate robust virtual verification and validation methods to master the complexity as well as to enable reduced development and release times.

Software in the Loop (SiL) represents the state of the art in industry for testing control functions at design time. In SiL environments, the embedded software is compiled for and runs in a pure virtual PC environment. Here, mainly the functional behavior of the software is tested, typically neglecting the complex timing effects due to the real-time embedded hardware, such as execution times, jitters, and input-output latencies. Conversely, performance simulations to ascertain efficient use of hardware resources (e.g., optimizing scheduling, distribution of tasks and memory footprint on multi-cores) are performed neglecting the functional behavior of the software. This separation results in e.g., incorrect functional simulation results, inadequate dimensioning of hardware resources, missed key functional requirements, and most importantly, prevents a credible virtual testing and validation of time-critical systems.

In this talk, we discuss and propose a modular tool architecture that couples real-time performance and functional simulation methods in a co-simulation, to enable a timing-aware functional simulation of automotive software. This method ensures that when a SiL simulation is performed in a virtual PC environment, complex real-world timing effects are also reflected in the simulation results. Our approach makes use of the novel Functional Mockup Interface (FMI) 3.0 standard, which enables an effective event driven co-simulation of functional models from different physical domains, while also promoting portability and re-usability of the software components.

A case study including standardized interfaces is presented to show an implementation of the proposed architecture integrating a commercial timing simulation tool. An AMALTHEA system model is used for modeling of the target architecture for the timing simulation combined with virtual Engine Control Unit (vECU) code wrapped in a Functional Mockup Unit (FMU).