Editorial

ModelBasedDesignofMechatronicSystems

prof. Ing. Michal Kelemen, PhD.
Technical University of Košice, Faculty of Mechanical Engineering, Institute of Automation, Mechatronics and Robotics, Department of Mechatronics

Michal Kelemen, prof. Ing. PhD. (born in 1974) is a professor of the Department of Mechatronics at the Faculty of Mechanical Engineering at the Technical University of Košice, Slovakia. He received M.S. degree in mechanical engineering from Technical University of Košice, Slovakia in 1998 and Ph.D. degree in Mechatronics from Technical University of Košice, Slovakia in 2002. He has been awarded in the 1998 "Price of the VolksBank" for the best M.S. graduate and 2007 Price "Scientist of the year". His research interests include mechatronic systems, intelligent robotic systems, measurement of non-electric quantities, and microcomputer systems. He has authored more than 200 journal and conference papers on these topics.

Model-Based Design is a process that enables faster and more cost-effective development of dynamic systems, including control systems, signal processing, and communications systems. In Model-Based Design, a system model is at the centre of the development process, from requirements development, through design, implementation, and testing. The model is an executable specification that you continually refine throughout the development process. After model development, simulation shows whether the model works correctly.

Model-Based Design helps engineers achieve certification to safety standards by supporting requirements traceability, verification, and documentation. These capabilities span multiple design stages. For example, requirements linked to model are inserted as comments in generated code. Qualification kits, available for several verification tools, can reduce the amount of manual review needed.

This method avoids work-intensive iterations in later development phases, and save time and money by:

■ ***Verifying at an early stage, by means of model simulation, that the model and requirement are correct.***

■ ***Verifying that the code and the mode are consistent, and that the code correctly represents the model's functionality, by simulating the generated code on the host PC.***

■ ***Verifying seamless traceability for documenting the software development.***

■ ***Allowing resource requirements to be estimated at an early stage by simulating the code on the appropriate evaluation hardware.***

Not only the number of electronic control units (ECUs) in modern vehicles constantly is increasing, the software of the ECUs is also becoming more complex. Both make testing a central task within the development of automotive electronics. Testing ECUs in real vehicles is

time-consuming and costly, and comes very late in the automotive development process. It is therefore increasingly being replaced by laboratory tests using hardware-in-the-loop (HIL) simulation.

Time to market is speeding up, especially in automotive electronics. Almost 90% of automotive innovations are currently connected with new electronics. Test drives can scarcely cope with the volume of systematic testing needed, especially just before start of production. The growing number of recall campaigns is a clear indication of this. It is little wonder that testing and error finding have become key tasks in the development process.

ECU testing typically is done using hardware-in-the-loop simulation. The ECU (prototype) is connected to a real-time simulation system simulating the plant (engine, vehicle dynamics, transmission, etc.) or even the whole vehicle.

As HIL has become a standard method for testing ECUs and control strategies during the whole development cycle (i.e., not only after availability of the final ECUs), different needs of different users have to be addressed by the various test systems.