Suitability of Different Real-Time Solvers for a Model-Based Engineering Toolchain using Industrial Rexroth Controllers

Nils Menager[1] Rüdiger Kampfmann[1] Niklas Worschech[1] Lars Mikelsons[1]

[1]Bosch Rexroth AG, Lohr a. Main, Germany {nils.menager, fixed-term.ruediger.kampfmann, niklas.worschech, lars.mikelsons}@boschrexroth.de

Due to the increasing complexity of technical systems, model-based engineering is getting more and more important during the development process of new products. The code generation from models and the usage of this code on hardware targets is one important feature of model-based development. To execute this code on the hardware device, a simulation runtime is additionally required, which offers numerical methods to solve the model equations. To use generated code on a controller, the simulation has to be executed in real-time, which is a huge requirement for the solver. A real-time solver has to guarantee that one time step is finished in a limited time, i.e. the real time cycle. In this context, linear-implicit Runge-Kutta methods, also known as Rosenbrock methods, are introduced.

In this work, a Modelica-based open source toolchain for model-based engineering with Rexroth controller is presented. This toolchain allows to generate C++-Code from Modelica models, which can be executed directly on industrial PLCs. Using this toolchain, a virtual commissioning can be easily performed. Therefore, the former described toolchain is used to execute both the plant model and the controller model on the hardware controller. In this contribution, a virtual commissioning of a single hydraulic axis, a typical hydro-mechanical system, on a standard Rexroth PLC is shown. Therefore, instead of parameterizing the controller directly on the real system, the controller model running on the PLC is connected to the system model, which is additionally executed on the PLC in parallel to the controller model. By doing so, the commissioning times can be reduced significantly, as the commissioning process can already be started during the build-up of the system using a simulation model of the system.

As hydro-mechanical systems are in general mathematically stiff, the choice of the solver for the system model equations is not arbitrary. In this work, five different real-time solvers, beginning with a simple explicit Euler through to more complex linearly implicit methods, are tested the hydro-mechanical example system. Furthermore, typical issues like time and state events as well as algebraic loops are discussed in context of real-time simulation requirements.