Modular Multi-Rate and Multi-Method Real-time Simulation

Bernhard Thiele[†] Martin Otter[†] Sven Erik Mattsson[‡]
Bernhard.Thiele@dlr.de Martin.Otter@dlr.de SvenErik.Mattsson@3ds.com

[†] German Aerospace Center (DLR), Institute for System Dynamics and Control, 82234 Wessling, Germany
[‡] Dassault Systèmes AB, Ideon Science Park, SE-223 70 Lund, Sweden

The demand to ever increase realism and scope of models routinely exceeds the currently available computing power and thus requires thoughts on improving simulation efficiency. This is especially true for real-time simulations, where fixed timing constraints do not allow to just "wait a bit longer".

This paper presents a new approach in Modelica that allows a modeler to separate a model into different partitions for which individual solvers can be assigned. In effect, this allows to use multi-rate and multi-method time integration schemes that can contribute to improve the efficiency of a (real-time) simulation.

The first part of the paper discusses basic consideration relating to modular (real-)time integration. Afterwards, the implementation of a convenient Modelica library for the partitioning of physical models is briefly described. Finally, the presented library is used to partition a detailed six degree of freedom robot model for modular simulation. The simulation performance of that partitioned model is compared to the simulation performance achieved by using "conventional" global solvers.

Figure 1 shows the robot benchmark example which is based on the `RobotR3` example from the Modelica Standard Library. It is partitioned into three parts: a) a discrete-time (digital) *control partition*, b) a *drive partition*, solved by an implicit Euler method, and c) a *multi-body partition*, solved by an explicit midpoint method. The coupling between b) and c) is achieved by using modeling elements from the presented library.

Keywords: multi-rate / multi-method time integration; simulation; clocked discretized continuous-time partitions.

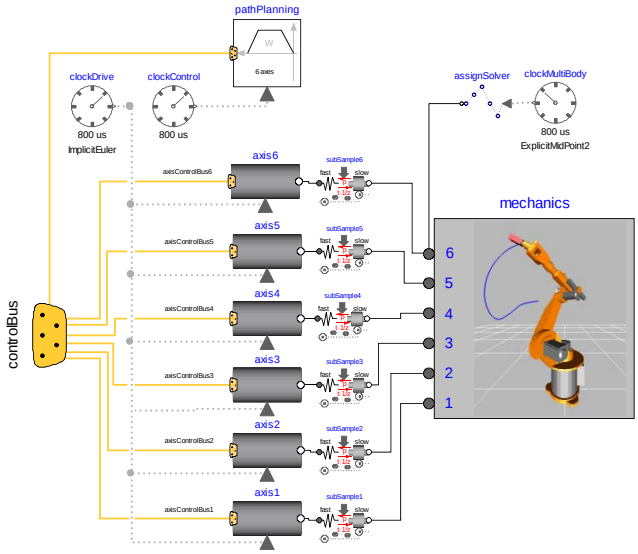

Figure 1: 6-DOF robot example partitioned into three parts. The respective solvers are assigned using the components `clockControl`, `clockDrive`, and `clockMultiBody`.