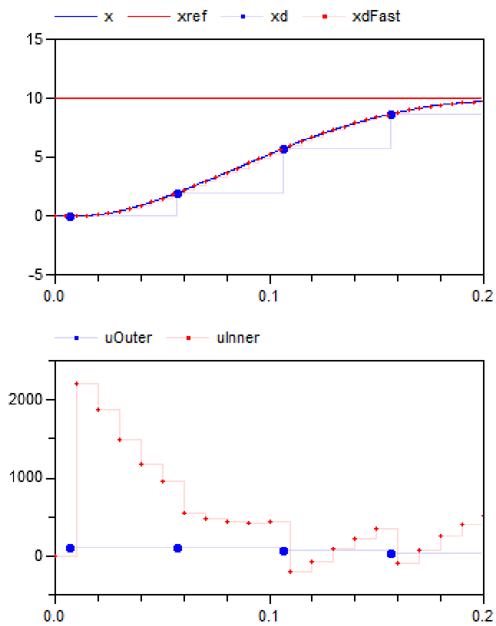
Fundamentals of Synchronous Control in Modelica

Hilding Elmqvist[1] Martin Otter[2] Sven Erik Mattsson[1]
[1]Dassault Systèmes AB, Ideon Science Park, SE-223 70 Lund, Sweden
[2]DLR, Institute of System Dynamics and Control, Wessling, Germany
Hilding.Elmqvist@3ds.com, Martin.Otter@dlr.de, SvenErik.Mattsson@3ds.com

The scope of Modelica has been extended from a language primarily intended for physical systems modeling to modeling of complete systems by allowing the modeling of control systems and by enabling automatic code generation for embedded systems.



This paper describes the fundamental synchronous language primitives introduced for increased correctness of control systems implementation. The approach is based on associating clocks to the variable types. Special operators are needed when accessing variables of another clock. This enables clock inference and increased correctness of the code since many more checks can be done during translation. Furthermore, the sampling period of a clocked partition needs to be defined only at one place (either in absolute time or relatively to other clocked partitions), general equations can be used in a clocked partition, and in particular continuous-time equations that are automatically discretized.

In the paper a rational is given why the new language elements have been introduced by comparing the new possibilities with the features of Modelica 3.2 to model sampled data systems. A companion paper (*Elmqvist, et.al, 2012*) describes the state machine features of Modelica 3.3. Yet another companion paper (*Otter*, *et.al*, *2012*) describes a Modelica library, Modelica_Synchronous, which supports a graphically oriented approach to synchronous control systems implementation.

The new language elements follow the synchronous approach. They are based on the clock calculus and inference system of Lucid Synchrone version 2 and 3 (*Pouzet 2006*). However, the Modelica approach also uses multi-rate periodic clocks based on rational arithmetic and also non-periodic and event based clocks are supported.

In the text box to the right, a very simple system consisting of a continuous-time plant and a sampled data P controller is shown using the new language elements. In the figure above, simulation results of a system with different sampling periods are shown.

```
// Continuous system
der(x) = v;
m*der(v) = f - k*x - d*v;

// Controller with period of 0.01 s
vd = sample(v, Clock(0.01));
u = K*(vref-vd);
f = hold(u);
```

References

Elmqvist H., Gaucher F., Mattsson S.E, and Dupont F. (2012): **State Machines in Modelica**. Proceedings of 9th Int. Modelica Conference, Munich, Germany, Sept. 3-5.

Otter M., Thiele B., and Elmqvist H. (2012): **A Library for Synchronous Control Systems in Modelica**. Proceedings of 9th Int. Modelica Conference, Munich, Germany, Sept. 3-5.

Pouzet M. (2006): **Lucid Synchrone, Version 3.0, Tutorial and Reference Manual**. http://www.di.ens.fr/~pouzet/lucid-synchrone/