Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Investigating communication and step-size behaviour for co-simulation of hybrid physical systems

Irene Hafner [a,*], Matthias Rößler [b], Bernhard Heinzl [b], Andreas Körner [a], Michael Landsiedl [c], Felix Breitenecker [a]

[a] *Vienna University of Technology, Institute of Analysis and Scientific Computing, Wiedner Hauptstr. 8-10, 1040 Wien, Austria*
[b] *Vienna University of Technology, Institute of Computer Aided Automation, Favoritenstr. 9-11, 1040 Wien, Austria*
[c] *dwh Simulation Services, Neustiftgasse 57-59, 1070 Wien, Austria*

ARTICLE INFO

Article history:
Received 30 January 2013
Received in revised form 27 July 2013
Accepted 25 August 2013
Available online xxx

Keywords:
Co-simulation
Energy optimization
Thermodynamics
MATLAB
Dymola

ABSTRACT

This paper discusses the method of cooperative simulation of discrete and continuous models with the Building Controls Virtual Test Bed, a software environment that allows coupling different simulation programs. In the course of a project aiming the energy optimization in cutting factories, models of machines of differing complexity and a building containing them have to be implemented to further simulate the thermal processes. Since all partial models require individual modelling approaches, solver time steps, solvers or even simulators, the method of co-simulation is considered. The partial models will be implemented with Modelica, MATLAB, Simulink and Simscape and accessed with the co-simulation tool BCVTB. The simulation results show that this method of co-simulation can be sufficient for the needs of describing thermal systems with large time constants but has to be found insufficient for simulations requiring high accuracy and variable step solvers in the overall simulation.

1. Motivation

To optimize the energy consumption in production halls, it is necessary to model the individual machines as well as the building itself in varying detail. To further install a feedback control keeping the room temperature at comfortable values for human beings which might enter the hall, the heat emitted by the machines and its distribution in the hall have to be simulated. The simulation of the temperature progression in these machine halls is part of the INFO (Interdisziplinäre Forschung zur Energieoptimierung in Fertigungsbetrieben) project which is supported by the Austrian Research Promotion Agency (FFG). In this paper a simulation of models implemented with Modelica, MATLAB, Simulink and Simscape will be described. An overview of the intended communication between the simulators is given in Fig. 1.

2. Co-simulation

2.1. Introduction

The basic idea of co-simulation is the parallel simulation of models using different simulators via one overall simulator. This simulator executes a model accessing the individual simulators and lets them synchronize at given time-steps. This way it is possible to match the individual modelling and simulation requirements of all partial models considering the fact that in our case the machine models result in differential algebraic equations describing electrical and mechanical circuits and the room model only deals with systems describing thermal processes, which react much slower and therefore do not require comparably small time steps and can use less complex solver algorithms for the simulation. This also explains why an external simulator is used to combine Simulink and Modelica models, which could actually be interacting themselves but would use only one solver for the simulation.

The overall simulation via BCVTB leads to opening one simulator instance for every model, which means even two models using Dymola can be simulated in parallel with different solver algorithms and solver step sizes.

2.2. Methods, consistency and stability

The usage of different solvers which take individual step sizes is called multi-rate simulation. Thus, multi-rate simulation would also be possible using only one simulator but different solvers or at least step sizes for parts of the system. According to [1] multi-rate simulation is in general divided into loose coupling and strong coupling methods. By using loose coupling, the different solvers

[*] Corresponding author. Tel.: +43 15880110164.
E-mail address: irene.hafner@tuwien.ac.at (I. Hafner).

1877-7503/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jocs.2013.08.007

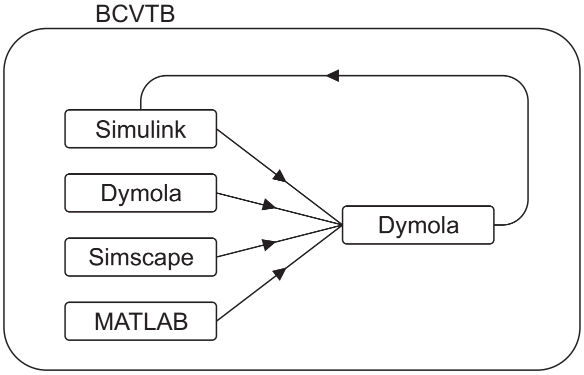

Fig. 1. Desired communication between the individual simulators.

exchange data at certain steps in time and have to extrapolate the values needed from other simulators by their prior values. Strong coupling leads to an exchange of data at every iteration until given accuracy settings are achieved. While strong coupling methods are by far more accurate than loose coupling methods, they are therefore much slower and require grave modifications in the participating solver algorithms. In this work, only loose coupling using the so-called *Jacobi*-type is considered. This means that all participating solvers extrapolate the values needed by other ones during one overall simulation time step from all previously gained values, calculate the state variables of their own systems in parallel and exchange them at the next synchronization reference to allow a new extrapolation for the next time step. Although this will lead to error accumulation, [2] and [3] show that consistency as well as stability of the coupled systems is maintained as long as the individual systems are consistent and stable, although the order of consistency might decrease. An overview of different co-simulation approaches is given in [2].

3. Software

3.1. BCVTB

The Building Controls Virtual Test Bed (BCVTB) is a co-simulation tool based on the graphical modelling interface Ptolemy. Though at first glance BCVTB and Ptolemy look exactly the same, BCVTB lacks many Ptolemy blocks (called actors in Ptolemy) and in return provides a lot of actors specifically for co-simulation. To supervise the overall simulation, so-called Directors are used. Depending on the given problem, one can choose between a Synchronous Data Flow Director which allows discrete-time synchronization at fixed time steps and a Continuous Time Director offering different variable step solver algorithms for the overall simulation.

Although this might at the first impression sound like an easy way to co-simulate models with a variable step size adapting to the demands of the given problem, it will be shown later in this paper that regarding especially these simulation directors, limits are met very soon. Other actors especially developed for co-simulation with BCVTB are simulator actors. These actors access the individual simulators using BSD-sockets, an application programming interface enabling inter-process communication which, like BCVTB itself, also has been developed at the University of California in Berkeley. In the simulator actors the name of the simulator to be accessed as well as the file to be executed has to be defined. All values needed by the respective simulator have to be transferred to the simulator actor as inputs and all values BCVTB gets from the simulator can be accessed at the output port of the actor. BCVTB basically allows co-simulation of Ptolemy, EnergyPlus, Dymola, MATLAB, Simulink, Radiance and BACnet. For further information, see [4].

3.2. Dymola/Modelica

Modelica [5] constitutes a standard for object-oriented modelling of physical systems using acausal block diagrams. Every block represents an element from one of various physical domains, e.g. electrical, mechanical or thermal. The underlying code contains a set of equations describing the behaviour of the corresponding component. By coupling an arbitrary amount of these components, systems of almost any complexity can be built quite easily without considering causalities of the signals. The solvers from a Modelica simulator like Dymola gather all equations, simplify and rearrange the system of equations and finally solve it.

3.3. MATLAB, Simulink and Simscape

MATLAB [6] is a programming environment offering its own language and various toolboxes meeting the requirements of different modelling approaches. One of these toolboxes is Simulink. Simulink models basically represent signal flow charts and can be built graphically as block diagrams of directed graphs. In block diagrams for Simscape models, acausal connections are used allowing physical modelling similar to Modelica.

4. Communication between the individual simulators

BCVTB offers several predefined functions to enable communication with the simulators mentioned above. In MATLAB the establishment of sockets, the exchange of data and the closure of the sockets is realized with predefined functions from BCVTB which have to be called in a loop over all time steps.

For the communication with Simulink BCVTB provides a preimplemented block which basically represents the functions for the communication with MATLAB. The Simulink block for the communication with BCVTB is shown in Fig. 2.

This block basically contains the same functions as are provided for the communication with MATLAB and put into the signal flow structure of Simulink. To ensure the communication at the time steps given by the overall model despite using an individual solver with variable step size, the given block has to be put in an If-Action-Subsystem which is only activated at the given synchronization references, as will be shown in Section 5.5. For Dymola, which is used as simulator for Modelica, the Modelica Buildings Library which has also been developed by the inventors of BCVTB has to be downloaded. This library contains among others a block similar to the one for the communication with Simulink. The parameters and the icon for the BCVTB block in Dymola can be seen in Fig. 3.

Apart from setting the number of values to be read from BCVTB and the number of values to be returned to BCVTB, the initial return value and the time step for synchronization can be chosen in the block parameters.

This shows one of the first weak points of BCVTB: Dymola cannot communicate with BCVTB at variable points in time with the given resources since the synchronization time step in the BCVTB block for Dymola can only be chosen fixed.

5. Model description

To focus again on the problem of simulating the thermal processes in a production hall, the individual models will be described. The machines in the production hall are modelled in Modelica, Simscape and as simple data model in MATLAB. The room itself is implemented as compartment model in Modelica. To keep the