

## Co-simulation of innovative integrated HVAC systems in buildings

Marija Trčka<sup>a\*</sup>, Jan L.M. Hensen<sup>a</sup> and Michael Wetter<sup>b</sup>

<sup>a</sup>Eindhoven University of Technology, PO Box 513, 5600MB Eindhoven, The Netherlands; <sup>b</sup>Simulation Research Group, Building Technologies Department, Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720

(Received 31 March 2009; final version received 19 May 2009)

Integrated performance simulation of buildings' heating, ventilation and air-conditioning (HVAC) systems can help in reducing energy consumption and increasing occupant comfort. However, no single building performance simulation (BPS) tool offers sufficient capabilities and flexibilities to analyse integrated building systems and to enable rapid prototyping of innovative building and system technologies. One way to alleviate this problem is to use co-simulation, as an integrated approach to simulation.

This article elaborates on issues important for co-simulation realization and discusses multiple possibilities to justify the particular approach implemented in the here described co-simulation prototype. The prototype is validated with the results obtained from the traditional simulation approach. It is further used in a proof-of-concept case study to demonstrate the applicability of the method and to highlight its benefits. Stability and accuracy of different coupling strategies are analysed to give a guideline for the required coupling time step.

**Keywords:** co-simulation; innovative building system modelling and simulation; HVAC simulation; building performance simulation

### 1. Introduction

Modern buildings are required to be energy efficient while adhering to the ever increasing demand for better indoor environmental quality. It is a known fact that, in developed countries, buildings account for 30–40% of the energy consumed. Depending on the building type, heating, ventilation and air-conditioning (HVAC) systems are responsible for 10–60% of the total building energy consumption. The long life-cycle of buildings further compounds the importance of architectural and engineering design decisions.

On the one hand, challenging goals are set by new initiatives and energy policies. For example, the European Union has defined ambitious goals for reducing emission of CO<sub>2</sub> for the industrialized countries. Also, the US Department of Energy and ASHRAE have defined their vision for 2030 (ASHRAE 2008) in a form of Net Zero Energy Buildings (NZEB). On the other hand, new buildings consist of numerous dynamically interacting components that are non-linear, dynamic and complex. This requires an integrated approach that treats innovative solutions to buildings and the systems that service them as complete entities, not as separately designed subsystems.

To design energy efficient building systems in this complex setting, integrated building performance simulation (BPS) can be used. Experience shows that BPS can indeed result in a significant reduction of emission of greenhouse gases and give substantial improvements in comfort levels (Hensen *et al.* 2004).

Because of the fragmented development of BPS tools and the rapid innovations in building and system technologies, state of the art BPS tools are often not comprehensive enough to model and simulate the relevant physical phenomena and the controls of modern mechanical system. Frequently, the user requirements exceed the functionality of the BPS tools. As it has been previously argued (Hensen 1991, Hensen and Clarke 2000), in the area of system simulation, there is still an enormous amount of work to be done.

The state of the art BPS tools are difficult and costly to extend. Adding new features requires the tool developer to have an in-depth knowledge of the programming languages used, of the underlying software architecture and the tool-specific modelling strategies. Furthermore, switching to equation-based tools, while promising for rapid prototyping of new systems, is not yet feasible for whole building simulation, as comprehensive and validated model libraries

\*Corresponding author. Email: m.trcka@tue.nl

do not yet exist. More research is also needed for such tools to allow a fast and numerically robust simulation (Wetter 2009).

Since the value of a tool is often measured by the number of its users, the tool development is mostly driven towards accommodating the existing HVAC designs. This is reflected in the amount of investments put into the market segments that have many users, e.g. into tools such as eQuest (<http://www.doe2.com>), DOE 2.1 (<http://www.doe2.com>), IES VE (<http://www.iesve.com>) and VA114 (<http://www.vabi.nl>), compared with more flexible tools, such as TRNSYS (<http://www.trnsys.com>). The adaptable tools such as TRNSYS or Modelica (<http://www.modelica.org>) have strength in system modelling and simulation and are so more likely to stimulate the innovation of new technologies for NZEB.

To successfully continue the development of the BPS tools that accelerate innovation of building technologies that help in mitigating climate change, a focus should be on supporting a flexible modelling environment that allows analysing building systems that have not yet been implemented by the program developers. A way forward would be to provide a facility to combine features from different tools, sharing developments and reusing component models. A tool should be coupled with a complementary tool in such a way that the integrated result provides more value to the end user than the individual tool does itself. This can be achieved by the integration of physical process models by linking applications at run-time. The strategy is known as process model cooperation (Hensen *et al.* 2004), external coupling (Djunaedy 2005) or co-simulation (Elliott 2004, Trčka 2008, Wetter and Haves 2008). Co-simulation is a case of simulation scenario where at least two simulators (simulation tools) solve coupled differential-algebraic systems of equations and exchange data during the time integration to couple these equations.

In general, compared with the traditional, monolithic approach, co-simulation has several advantages (Hillestad and Hertzberg 1986, Boer 2005):

- it facilitates reuse of state of the art domain simulation tools by taking advantage of the existing models;
- it allows combining heterogeneous solvers (using discretization techniques and solution algorithms that are best suited for a modelled subsystem) and modelling environments of specialized tools;
- it enables fast model prototyping of new technologies;
- it facilitates collaborative model design and development process, i.e. models developed by different design teams or subcontractors can be executed concurrently;

- it makes immediate access to new model developments;
- it permits information hiding, i.e. use of proprietary tools.

However, these flexibilities can pose numerical challenges, and to scale the use of co-simulation to a large community of building designers requires more research and development.

Co-simulation has been successfully applied in different fields, such as aerospace and automotive (<http://www.adi.com>), high performance computing, defence and internet gaming (Wilcox *et al.* 2000, Fujimoto 2003), multi-body dynamics (Park 1980), hydrology (Tseng *et al.* 1995), mechatronics (Arnold *et al.* 2002), chemistry (Hillestad and Hertzberg 1988) and aerodynamics, structural mechanics, heat transfer and combustion (Follen *et al.* 2001, Sang *et al.* 2002).

In the field of BPS, considerable effort has been made in integrating coupled physical phenomena into the individual BPS tools (e.g. ESP-r (<http://www.esru.strath.ac.uk>), EnergyPlus (<http://www.energyplus.gov>), IES VE, IDA ICE (<http://www.equa.se>), TRNSYS). Some of the integrated BPS tools integrate process models by converting models available in other tools into their own subroutines. Examples of such integrations are the coupling between ESP-r and TRNSYS (Hensen 1991, Aasem 1993, Wang and Beausoleil-Morrison 2009), COMIS and EnergyPlus (Huang *et al.* 1999), COMIS and TRNSYS (Weber *et al.* 2002, McDowell *et al.* 2003), EnergyPlus and MIT-CFD (Zhai 2003), EnergyPlus and Delight (Carroll and Hitchcock 2005) and EnergyPlus and SPARK (<http://simulationresearch.lbl.gov>).

However, only a limited amount of work has been done in process model co-operation (co-simulation). Examples of such integration include the integration of high-resolution light simulation (radiance) with building energy simulation (ESP-r) (Janak 1999) and the integration of computational fluid dynamics simulation (FLUENT) with building energy simulation (ESP-r) (Djunaedy *et al.* 2003). In the 2003 domain of HVAC simulation tools examples include integration of TRNSYS with several other programs, such as MATLAB (<http://software.cstb.fr>) and EES (Keilholz 2002). These BPS tools couple two simulators directly with each other, with one tool serving as the master and the other as the client. A different architecture has been implemented in the Building Controls Virtual Test Bed (BCVTB) that uses a middleware to manage the data exchange between different simulators, with each simulator acting as a client (Wetter and Haves 2008). However, until now, there exists no general standardized framework for integration of BPS simulators, nor do

there exist guidelines for implementation of co-simulation with regards to stability and accuracy.

The co-simulation strategy in comparison with other strategies that enable sharing of developments and reusing existing component models (Hensen *et al.* 2004) is presented in Figure 1. The coupled models are independently created and the results are analysed separately, while the simulators are coupled at run-time, exchanging data in a predefined manner. In comparison with process model inter-operation, co-simulation enables immediate use of the component models developed in different tools (providing that the tools are open for communication). Once developed and implemented, the general co-simulation interface between the simulators can be used without any code adaptation, which is necessary in any other tool integration strategy.

Using co-simulation for BPS can be beneficial since:

- there is no single tool that can be used to solve all simulation analysis problems encountered by the designers;
- each tool can benefit from future simulation model developments of emerging technologies as soon as they become available;
- rapid prototyping of new technologies, which is difficult in the state of the art domain tools, could be done using an equation-based simulation tool. When used in co-operation with whole building

energy analysis programs, it would assure the integrated approach to building and systems simulation;

- multi-scale modelling and simulation can be done by combining various building and system models, developed by different parties, to simulate scenarios on the scale of a town or even regions.

In this article, we discuss principles of co-simulation, comment on our development and implementation and test the usability of co-simulation for performance prediction of innovative integrated energy systems in buildings. We also compare the numerical and computational performance of different co-simulation implementations.

## 2. Nomenclature

### 2.1. Conventions

- (1) Vectors are typeset in bold fonts.
- (2) Superscripts denote the time step number.
- (3)  $f(\cdot)$  denotes a function, where  $(\cdot)$  stands for the undesignated variables.  $f(x)$  denotes the value of  $f(\cdot)$  for the argument  $x$ .  $f: A \rightarrow B$  indicates that the domain of  $f(\cdot)$  is in the space  $A$ , and that the image of  $f(\cdot)$  is in the space  $B$ .
- (4)  $y(t)$  is a state variable, and  $\dot{y}(t)$  denotes the time derivative of the state variable.

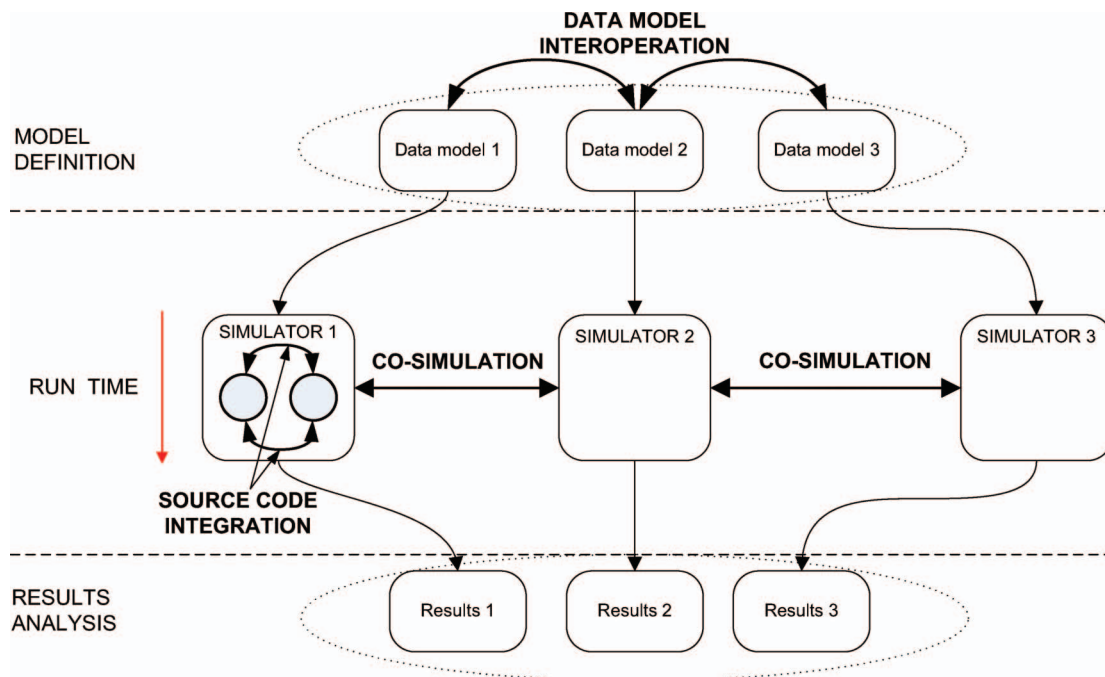


Figure 1. Co-simulation in relation with other tool integration strategies.

- (5) We say that a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is once Lipschitz continuously differentiable if  $f(\cdot)$  is defined on  $\mathbb{R}^n$  and  $f(\cdot)$  has a Lipschitz continuous derivative on  $\mathbb{R}^n$ .

## 2.2. Symbols

$a \in A$	$a$ is an element of $A$
$c_p$	specific heat capacity
$i, j, k, n$	counters
$h$	convective heat-transfer coefficient
$HG$	heat gain
$inf$	infiltration
$L$	Lipschitz constant
<b>LTE</b>	local truncation error
$\dot{m}$	mass flow
$\mathbb{N}$	set of natural numbers, $\mathbb{N} = \{0, 1, 2, \dots\}$
$\dot{Q}$	heat rate
$P$	predicted
$\mathbb{R}$	set of real numbers
$s$	surface
$sup$	supply
$sys$	system related
<b>ULTE</b>	unit local truncation error
$T$	temperature
$t$	time
$z$	zone
$\alpha, \beta, \mu$	scalar parameters
$\Delta$	difference

## 3. New functionalities enabled by co-simulation

A benefit of a co-simulation environment is that the domain-specific tools can be coupled for an integrated simulation while preserving their individual features. It therefore enables, for example, the following:

- (1) Use of disparate tools that support individual domains. For example, the EnergyPlus building model may be used as it allows modelling daylight availability in rooms, whereas TRNSYS may be used as it allows a graphical, flexible modelling of the mechanical system.
- (2) Development of control algorithms in tools such as MATLAB/Simulink or LabVIEW, which provide toolboxes for designing controllers, as well as code generation capabilities to translate a simulation model to C code that can be uploaded to control hardware.
- (3) Development of control algorithms in tools that allow a richer semantics for expressing models compared with what can be found in typical BPS tools. For example, the complexity of large control systems can be managed using a hierarchical composition in which finite-state machines define the states and their transition

at the supervisory control level, and each state may have a refinement that defines how set points are tracked within the active state (Lee and Varaiya 2002). Tools that allow such formulations include MATLAB/Simulink, LabVIEW and Ptolemy II (Brooks *et al.* 2007).

Although technically not belonging to co-simulation, a co-simulation framework also allows replacing one of the simulators with an interface to a control system, thereby enabling the use of hardware in the loop in which controls may be realized in actual hardware while the building system is emulated in simulation.

## 4. Principles and strategies

Various co-simulation realizations have different implications with regards to stability, convergence, accuracy, efficiency and ease of implementation. This section discusses different approaches for implementing co-simulation.

### 4.1. Data transfer

The communication between processes (applications) are called interprocess communications (IPC). Overviews of most commonly used IPC protocols can be found in Yahiaoui *et al.* (2004) and Trčka-Radošević and Hensen (2006). Here, a few IPC mechanisms will be mentioned.

*Remote Method Invocation (RMI)* is language specific and suitable for use with newly developed applications. Examples of co-simulation frameworks and toolkits, which are concerned with legacy applications, possibly developed in different languages, include:

- the *Common Object Request Broker Architecture (CORBA)* (Wilcox *et al.* 2000, Follen *et al.* 2001, Strassburger 2001, Duggan 2002, Sang *et al.* 2002, Taylor *et al.* 2002, Lee 2004, Wang *et al.* 2004, Boer 2005),
- *High Level Architecture (HLA)* (Wilcox *et al.* 2000, Li *et al.* 2005),
- Applied Dynamics International's (ADI) *Advantage framework* ([http://www.adi.com/products\\_sim.htm](http://www.adi.com/products_sim.htm)),
- *Model Coupling Toolkit (MCT)* ([www.mcs.anl.gov/mct](http://www.mcs.anl.gov/mct)),
- *Common Component Architecture (CCA)* ([www.cca-forum.org](http://www.cca-forum.org)), and
- *Building Controls Virtual Test Bed (BCVTB)* being developed by Lawrence Berkeley National Laboratory (LBNL) (Wetter and Haves 2008).

The implementation of any of the mentioned co-simulation frameworks for distributed building system simulation raises difficulties when interfacing state of the art BPS tools. A challenge is to integrate the data exchange with the internal data structures, time integration algorithms and program flow of the individual simulators.

#### 4.2. Coupling strategies

Based on the temporal data exchange and the iteration between the simulators, the following coupling strategies can be distinguished:

- *Strong coupling* (Struler *et al.* 2000), also called fully dynamic (Zhai 2003) or onion coupling (Hensen 1999), requires an iteration that involves the coupled simulators to guarantee user-defined convergence criteria. The strong coupling strategy allows the use of longer time steps for the same accuracy, compared with the loose coupling strategy, since implicit time integration algorithms can be used. In strong coupling, a ‘passive’ simulator, i.e. a simulator that does not control the iteration between the two simulators, must have a mechanism to rewind its state if requested from the co-simulation manager. In addition, since each simulator may include iterative solutions of equations, strong coupling leads to nested iteration loops, consisting of an inner iteration within the individual simulators, and an outer iteration to achieve convergence of the coupled simulators. To ensure convergence, the inner iterations need to be solved at higher accuracy than the outer iterations. This may be impractical to accomplish in BPS tools that do

not allow controlling the precision of the numerical error. Thus, to realize strong coupling, significant code modifications may be required.

- In *loose coupling* (Struler *et al.* 2000), also called quasi-dynamic (Zhai 2003) or ping-pong coupling (Hensen 1999), coupled simulators use the coupling data that is computed using only data from preceding time steps. There is no iteration between the coupled simulators. We distinguish two types of loose coupling strategy (see Figure 2):

- loose coupling with *sequential staggered* solution (Felippa *et al.* 1999), also called zigzagged coupling, where the coupled simulators are executed in sequence, and
- loose coupling with *naive modification for parallel processing* (Felippa *et al.* 1999), also called cross coupling, where the coupled simulators are executed in parallel.

#### 4.3. The role of the simulators

The co-simulation discussed in this article implements zigzagged coupling, in which the sending and the receiving sequence differs between the coupled simulators. For that reason, we call the simulators the *base* and the *external* simulator. The base simulator starts the communication by sending the coupling data to the communication interface. The external simulator starts the communication by reading those data from the communication interface.

#### 4.4. System partitioning

The system partitioning can be (Felippa *et al.* 1999): (i) *algebraic*, where the complete differential system of

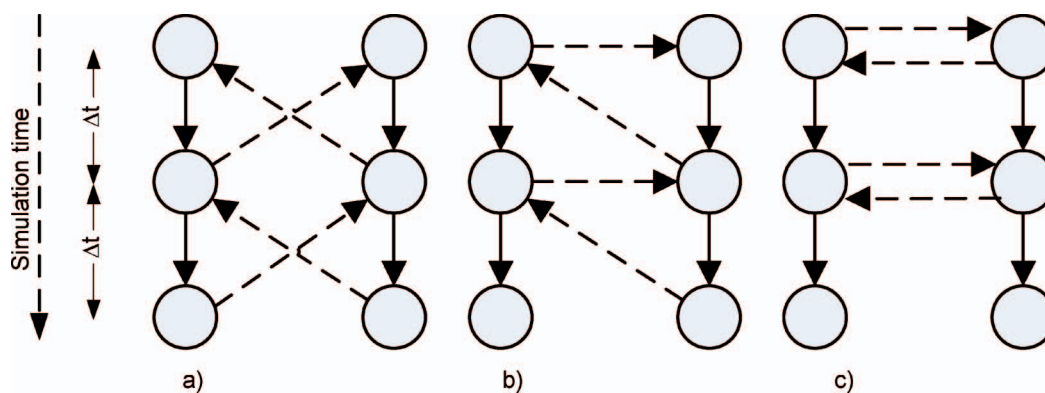


Figure 2. Sequence of coupling data exchange. (a) Time-state scheme of strong coupling; (b) time-state scheme of loose coupling with sequential simulators execution; and (c) time-state scheme of loose coupling with parallel simulators execution. The circles represent the subsystem’s state at a specific moment in simulation time. The dashed arrows indicate which coupling data (time-step wise) are available to each subsystem before the time-step calculation is performed. The full-line arrows indicate the update of state variables.

equations is discretized first, and then partitioned, or (ii) *differential*, where the sequence of discretization and partitioning is reversed. We used algebraic partitioning in our co-simulation prototype.

Depending on which data are delayed in time, Park (1980) defines two partitioning strategies: (i) *implicit–implicit* – if the coupling data depends only on the state variables of the coupled subsystem, and (ii) *implicit–explicit* – if the coupling data depends on the state variables of both subsystems.

#### 4.5. System decomposition strategies

The experiment in this article will use two different system-decomposition strategies: (i) *intra-domain* system decomposition, in which the system is decomposed within one functional domain, such as within the HVAC domain only, and (ii) *inter-domain* system decomposition, in which the system is decomposed between different functional domains, such as between the building and the HVAC system domain.

#### 4.6. Coupling data

One important design decision in implementing co-simulation is which data will be exchanged between the simulators. The data should, as much as possible, represent physical quantities as opposed to derived or abstract data because they (i) could be measured in the real world and (ii) are readily available in any domain simulator.

For intra-domain decomposition within the HVAC domain, on the basis of conservation equations, we selected a set of coupling data that includes mass flow rate, temperature and humidity ratio for exchange in both directions. Since the coupling data depends only on the state of the coupled subsystem, the intra-HVAC-domain decomposition results in an implicit–implicit partitioning.

For inter-domain decomposition, we selected as the basic set of coupling data the heat rate (convective, radiant and latent) in one direction and temperatures (air and mean radiant) and humidity ratio in the other direction. The coupling data (heat rate) depends on the states (temperatures) of both coupled subsystems, and thus, the inter-domain decomposition results in an implicit–explicit partitioning. This has a direct implication on co-simulation stability and accuracy.

The sets can be extended to include control signals if sensors and actuators are distributed among coupled simulators.

#### 4.7. Time management in co-simulation

Maybe the most important issue when discussing co-simulation is time synchronization. Many studies dealing

with distributed simulation address the issue of synchronization (Fujimoto 1998, Tacic and Fujimoto 1998, Wang *et al.* 2004, Boukerche *et al.* 2005). They all generally tackle the event driven simulations and define two main approaches for synchronization (Fujimoto 2003):

- *Conservative* approach avoids processing data out of time stamp order.
- *Optimistic* approach uses a detection mechanism and recovery approach, known as roll-back. Introducing roll-back to an existing simulator requires a major re-engineering effort (Page *et al.* 1999) to incorporate state saving mechanism. This approach is mainly used for event driven simulations.

Even though coupled BPS simulations follow their own time-management scheme, they are dependent on each other, and the execution of one will influence the execution of the other in the corresponding simulation time. It is therefore important that the simulation clocks of each BPS simulator are synchronized with each other. The conservative approach has been selected for our co-simulation implementation.

#### 4.8. Coupling time step

Accuracy and stability of co-simulation depend on the coupling time step. In general, the selection of a coupling time step depends on the time constant of the states that exchange flow variables (heat or mass flow) and the rate of change of inputs that act on the state variables, such as control actions or weather data (Trčka 2008).

#### 4.9. Multi-rate co-simulation

Multi-rate co-simulation can be used for simulation of stiff systems (systems that are comprised of subsystems with vastly different time constants). The stiff systems are decomposed by isolating the most rapidly or the most slowly responding subsystem and applying a suitable method and step length for their simulations. Larsson (2001) reports that if instantaneous values of state variables are exchanged among the simulators the coupling may generate numerical problems. One of the options to improve the coupling is to use extrapolation and/or interpolation of coupling data (Elliott 2000, Gravouil and Combescure 2001, Grott *et al.* 2003, Elliott 2004).

### 5. Stability and accuracy

The BPS tools typically contain legacy code with more than 100,000 lines of code that mixes code to implement physical equations, data exchange and numerical solution algorithms. This makes it difficult to reinitialize state variables to previous values and to control the

accuracy of iterative solutions, which is necessary for strong coupling. However, loose coupling is easier to implement, but the time delay of the coupling data causes the original numerical time integration schemes to be modified. Consequently, the stability and accuracy properties of the original time integrations scheme are no longer guaranteed.

Although the stability and accuracy of different time integration schemes are well understood (Gear 1971, Lambert 1991, Golub and Ortega 1992), the stability and accuracy of the methods resulting from partitioning are not well analysed. Kubler (2000) states that it is difficult, if not impossible, to determine these properties formally for a general class of problems.

In this section, we consider problems defined by the first order initial value ordinary differential equation

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}, t), \tag{1a}$$

$$\mathbf{y}(a) = \boldsymbol{\eta}, \tag{1b}$$

where  $\mathbf{f} : \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^m$ ,  $t \in [a, b]$  for some  $a, b \in \mathbb{R}$ , with  $a < b$ ,  $m \in \mathbb{N}$  and  $\boldsymbol{\eta} \in \mathbb{R}^m$ . We assume  $\mathbf{f}(\cdot, \cdot)$  is once Lipschitz continuously differentiable in  $\mathbf{y}$  and  $t$ . This ensures that a unique solution of Equation (1) exists.

To approximate the solution of the initial value problem (1), we will use a  $k$ -step numerical integration method. The general form of the method is

$$\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} = \Delta t \phi_{\mathbf{f}}(\mathbf{y}^{n+k}, \dots, \mathbf{y}^n, t^n; \Delta t) \tag{2}$$

with  $\mathbf{y}^\mu = \boldsymbol{\eta}^\mu$  and  $\mu \in \{0, 1, \dots, k-1\}$  (Lambert 1991). The subscript  $\mathbf{f}$  on the right-hand side indicates that the function  $\phi$ , which characterizes the particular method, depends on  $\mathbf{f}(\cdot, \cdot)$ .

For linear multi-step methods, the system (2) can be written as

$$\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} = \Delta t \sum_{j=0}^k \beta_j \mathbf{f}(\mathbf{y}^{n+j}, t^{n+j}), \tag{3}$$

where  $\{\alpha_j, \beta_j \in \mathbb{R} | j = 0, 1, \dots, k\}$  are method-specific parameters.

Let  $\alpha$  be the implicitness factor of a linear one-step numerical integration method. For members of the  $\alpha$ -family of linear one-step numerical integration methods, Equation (3) leads to

$$\mathbf{y}^{n+1} - \mathbf{y}^n = \Delta t [(1 - \alpha)\mathbf{f}(\mathbf{y}^n, t^n) + \alpha\mathbf{f}(\mathbf{y}^{n+1}, t^{n+1})], \tag{4}$$

where  $0 \leq \alpha \leq 1$ .

**Consistency** The following definitions are taken from the literature, e.g. Lambert (1991).

**Definition 5.1. Local truncation error:** The local truncation error is defined as the error produced in a single integration step starting from the exact solution.  $\square$

For the  $k$ -step numerical integration method (2), the local truncation error is

$$\text{LTE}^{n+k}(\Delta t) = \sum_{j=0}^k \alpha_j \mathbf{y}(t^{n+j}) - \Delta t \phi_{\mathbf{f}}(\mathbf{y}(t^{n+k}), \dots, \mathbf{y}(t^n), t^n; \Delta t). \tag{5}$$

**Definition 5.2. Unit local truncation error:** The unit local truncation error is defined as

$$\text{ULTE}^{n+k}(\Delta t) = \frac{\text{LTE}^{n+k}(\Delta t)}{\Delta t}. \tag{6}$$

**Definition 5.3. Consistency:** A numerical integration method is said to be consistent if for all initial value problems the unit local truncation error satisfies  $\lim_{\Delta t \rightarrow 0} \text{ULTE}^{n+k}(\Delta t) = 0$ .  $\square$

**Zero-stability** is concerned with the asymptotic behaviour in the limit as  $\Delta t \rightarrow 0$ . It is a property of the numerical integration method (3) and not of the differential equation (1).

Assuming that the initial value problem (1) satisfies the Lipschitz condition, the linear numerical integration method (3) tends to the linear constant coefficient difference system  $\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} = 0$ , as  $\Delta t \rightarrow 0$ , whose characteristic polynomial,  $\rho(r) = \sum_{j=0}^k \alpha_j r^j$ , is the first characteristic polynomial of the numerical integration method. Let the roots of  $\rho(r) = 0$  be  $\{r_i \in \mathbb{C} | i = 1, 2, \dots, k\}$ . The numerical integration method is said to be zero-stable if all the roots of the (first) characteristic polynomial satisfy  $|r_i| \leq 1$ , and any root for which  $|r_i| = 1$  is simple.

**Convergence** We will now introduce convergence (Lambert 1991).

**Definition 5.4. Convergence:** Consider problem (1) and for  $N \in \mathbb{N}$ ,  $N > 0$ , let  $\tau_N \triangleq \{t^n \in \mathbb{R} | t^n = a + n \Delta t, n \in \{0, 1, \dots, N\}, \Delta t = (b - a)/N\}$ . A numerical integration method is said to be *convergent* if for all  $t^n \in \tau_N$ ,

$$\lim_{N \rightarrow \infty} \mathbf{y}^n = \mathbf{y}(t^n). \tag{7}$$

Necessary and sufficient conditions for a numerical integration method to be convergent is that it is

both consistent and zero-stable (Gear 1971, Lambert 1991).

### 5.1. Consistency of co-simulation

In co-simulation discussed in this article, the system of Equation (2) is first partitioned algebraically, and then solved in coupled simulators. At the  $(n + k)$ -th time step, where  $n + k \leq N$ , the coupling data depends on  $\{\mathbf{y}^{n+j} \in \mathbb{R}^m \mid j \in \{0, 1, 2, \dots, k\}, n \in \{0, 1, \dots, N\}, \Delta t = (b - a)/N\}$ . If the simulators are loosely coupled, the coupling data at  $t^{n+k}$  is not available to (both) coupled simulators and needs to be predicted based on the data of the preceding time steps. In other words, the arguments of the function  $\phi_f$  are no longer  $\mathbf{y}^{n+k}$  but they are now  $\mathbf{y}_P^{n+k}$ , where  $\mathbf{y}_P^{n+k}$  represents the predicted state vector. Thus, in loosely coupled co-simulation, Equation (2) becomes

$$\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} = \Delta t \phi_f(\mathbf{y}_P^{n+k}, \mathbf{y}^{n+k-1}, \dots, \mathbf{y}^n, t^n; \Delta t),$$

and Equation (4) becomes

$$\mathbf{y}^{n+1} - \mathbf{y}^n = \Delta t [(1 - \alpha) \mathbf{f}(\mathbf{y}^n, t^n) + \alpha \mathbf{f}(\mathbf{y}_P^{n+1}, t^{n+1})]. \quad (8)$$

To determine the consistency of co-simulation, we directly use Definition 5.3. For the numerical approximation (8), the local truncation error is

$$\text{ULTE}^{n+1}(\Delta t) = \frac{1}{\Delta t} (\mathbf{y}(t^{n+1}) - \mathbf{y}(t^n) - \Delta t [(1 - \alpha) \mathbf{f}(\mathbf{y}(t^n), t^n) + \alpha \mathbf{f}(\mathbf{y}_P(t^{n+1}), t^{n+1})]), \quad (9)$$

where  $\mathbf{y}_P(t^{n+1})$  is an approximation of  $\mathbf{y}(t^{n+1})$  based on the values of  $\{\mathbf{y}(t^{n-j}) \in \mathbb{R}^m \mid j \in \{0, 1, \dots, n\}\}$ . Adding and subtracting  $\alpha \mathbf{f}(\mathbf{y}(t^{n+1}), t^{n+1})$  to the right-hand side and collecting terms that correspond to the local truncation error of the original, non-partitioned numerical scheme yields

$$\text{ULTE}^{n+1}(\Delta t) = \text{ULTE}_{\text{non-partitioned}}^{n+1} + \alpha [\mathbf{f}(\mathbf{y}(t^{n+1}), t^{n+1}) - \mathbf{f}(\mathbf{y}_P(t^{n+1}), t^{n+1})]. \quad (10)$$

Applying the norm on both sides of Equation (10) yields

$$\|\text{ULTE}^{n+1}(\Delta t)\| \leq \|\text{ULTE}_{\text{non-partitioned}}^{n+1}\| + \alpha L \|(\mathbf{y}(t^{n+1}) - \mathbf{y}_P(t^{n+1}))\|, \quad (11)$$

where  $L$  is the Lipschitz constant.

To evaluate the order of the error, the exact solutions of the state vectors in the two subsequent time steps,  $\mathbf{y}(t^{n+1})$  and  $\mathbf{y}(t^n)$ , is expressed around time  $t^n + \alpha \Delta t$ , for any  $\alpha \in [0, 1]$ , using Taylor series. When

substituted into Equation (11) for the zero-order predictor  $\mathbf{y}_P(t^{n+1}) = \mathbf{y}(t^n)$ , one obtains

$$\|\text{ULTE}^{n+1}(\Delta t)\| \leq \|\text{ULTE}_{\text{non-partitioned}}^{n+1}\| + \alpha LO(\Delta t). \quad (12)$$

It follows from Equation (12) and from  $\lim_{\Delta t \rightarrow 0} \alpha LO(\Delta t) = 0$  that if the original non-partitioned numerical scheme is consistent, i.e.  $\lim_{\Delta t \rightarrow 0} \|\text{ULTE}_{\text{non-partitioned}}^{n+1}\| = 0$ , then the partitioned numerical scheme is consistent as well.

The unit local truncation error introduced by the partitioning is of order one. The order of the error of first order accurate methods will not be changed by the partitioning. However, for the Crank–Nicholson method ( $\alpha = 1/2$ ), which is of the second order, the accuracy will be reduced by the partitioning. A more detailed analysis on a specific problem, presented in Trčka (2008), showed that the greater the capacity of the subsystem simulated in the external simulator, and the smaller the magnitude of the first derivative of the coupling data, the smaller the error. Also, if the system is defined by a differential-algebraic system of equations, the partitioning can lead to inconsistencies in the numerical approximation to the solution (Trčka 2008).

### 5.2. Zero-stability and convergence of co-simulation

By inspection of the partitioned linear numerical integration method (8), it can be seen that the partitioning changes only the right-hand side of the equation. The first characteristic polynomial that determines zero-stability of a numerical integration method depends only on the coefficients on the left-hand side of Equation (8) and thus, it does not change with the partitioning. Consequently, the partitioning does not disturb the properties of the zero-stability of the non-partitioned numerical integration method.

Since by Equation (12) it was shown that the numerical integration method (8) is consistent, and since the zero-stability of the original numerical scheme is not changed with partitioning, it follows that the numerical integration method (8) is convergent.

However, the zero-stability from the first characteristic polynomial does not completely cover the situations analysed in Kubler (2000). Kubler analysed a situation where co-simulated subsystems in a loop are modelled using algebraic equations, or where the outputs of co-simulated subsystems in a loop are not only functions of their state, but also (algebraic) functions of their inputs. The stability of the first characteristic polynomial does not ensure the zero-stability. Under assumptions that the output equations are time invariant and linear functions of the inputs, Kubler (2000) showed that



stability is maintained if in addition one of the coupled subsystems has no feed-through – meaning that the outputs depend only on the state variables but are not algebraic functions of the inputs.

The cumbersome analysis of absolute stability, i.e. stability for a finite value of  $\Delta t$ , of the partitioned numerical integration method (8) shows that the implicit–implicit partitioning of co-simulation is unconditionally stable if coupled simulators employ  $\alpha \geq 1/2$ . If  $\alpha < 1/2$ , the co-simulation is conditionally stable (Trčka 2008).

For implicit–explicit partitioning, co-simulation is conditionally stable even for  $\alpha > 1/2$ .

Further, if the subsystems are linked by a control loop, the absolute stability criterion is influenced by the control parameters.

## 6. Prototype

The co-simulation prototype is based on two state of the art BPS tools. EnergyPlus has been selected because of its advanced building model. TRNSYS has been selected because of its large library of HVAC components and its modular system modelling capability.

### 6.1. Loose coupling implementation

Figure 3 shows a flow chart of the data flow for the loose coupling. The coupling data is exchanged only in the first iteration for the current time step in both simulators (if  $j = 0$  for simulator 1 (EnergyPlus) or  $i = 0$  for simulator 2 (TRNSYS)). To understand how the data

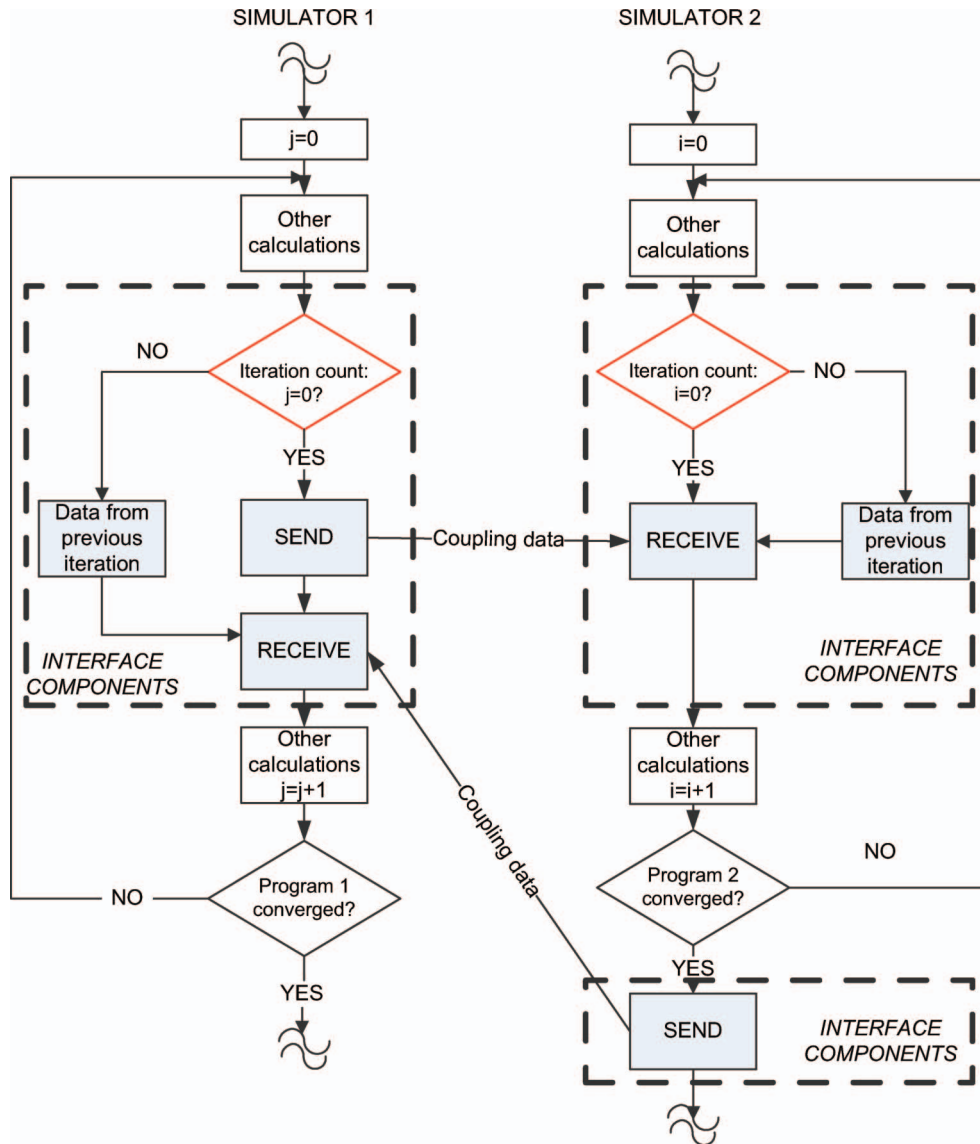


Figure 3. Flow chart of the loose coupling implementation.

sent by TRNSYS is incorporated into EnergyPlus' zone temperature correction formula, the adjustments of the energy balance equation at the  $(n + 1)$ -th time step of EnergyPlus will now be discussed.<sup>1</sup> The equation for the zone temperature integration is

$$T_z^{n+1} = \frac{\sum_{i=1}^{N_{HG}} \dot{Q}_i^{n+1} + G + (\dot{m}c_p)_{sys}^{n+1} T_{sup}^{n+1} - A}{B + F + (\dot{m}c_p)_{sys}^{n+1}}, \quad (13)$$

where  $N_{HG} \in \mathbb{N}$  is the number of different zone internal heat gain sources,  $N_s \in \mathbb{N}$  is the number of surfaces in the zone, and  $N_z \in \mathbb{N}$  is the number of adjacent zones. The terms  $A$ ,  $B$ ,  $G$  and  $F$  are defined as

$$A = \frac{C_z^n}{\Delta t} \left( 3T_z^n + \frac{3}{2}T_z^{n-1} - \frac{1}{3}T_z^{n-2} \right),$$

$$B = \frac{11}{6} \frac{C_z^n}{\Delta t},$$

$$G = \sum_{i=1}^{N_s} (h_i A_i)^n T_{si}^n + \sum_{i=1}^{N_z} (\dot{m}_i c_p)^n T_{zi}^n + (\dot{m}c_p)^n T_{inf}^n,$$

$$F = \sum_{i=1}^{N_s} (h_i A_i)^n + \sum_{i=1}^{N_z} (\dot{m}_i c_p)^n + (\dot{m}c_p)_{inf}^n.$$

For the intra-HVAC-domain decomposition, the coupling data are incorporated by the term  $(\dot{m}c_p)_{sys}^{n+1} T_{sup}^{n+1}$  in Equation (13). For the inter-domain decomposition, the coupling data are incorporated by the term  $\sum_{i=1}^{N_{HG}} \dot{Q}_i^{n+1}$  in Equation (13).

For the inter-domain decomposition, the coupling heat rate is calculated based on the known zone state in EnergyPlus, i.e.  $T_z^n$ . Because of the additional delay introduced by the type of the coupling data, the inter-domain decomposition results in lower accuracy compared with the intra-HVAC-domain decomposition. However, there are means to improve the accuracy of inter-domain decomposition by correcting Equation (13) for the time lagging, which we will now discuss.

The heat rate from the coupled (e.g. air) subsystem in the intra-domain decomposition is calculated in EnergyPlus as a function of  $T_z^{n+1}$ . In the inter-domain decomposition, the heat rate from the coupled (e.g. air) subsystem is calculated in the coupled simulator (e.g. TRNSYS) as a function of  $T_z^n$ , since at the simulation time point,  $T_z^{n+1}$  is not known. Thus, the resulting heat rates are

$$\dot{Q}_{sys,intra-dec}^{n+1} = (\dot{m}c_p)^{n+1} (T_{sup}^{n+1} - T_z^{n+1}),$$

$$\dot{Q}_{sys,inter-dec}^{n+1} = (\dot{m}c_p)^{n+1} (T_{sup}^{n+1} - T_z^n).$$

If the heat transfer rate is a linear function of the zone temperature, as in the earlier equations, a correction term calculated from the difference

$$\dot{Q}_{sys,intra-dec}^{n+1} - \dot{Q}_{sys,inter-dec}^{n+1} = C^{n+1} (T_z^n - T_z^{n+1}),$$

where  $C^{n+1}$  is a coefficient (e.g.  $(\dot{m}c_p)^{n+1}$  or  $(UA)^{n+1}$ )<sup>2</sup> can be included in the heat balance equation for the inter-domain system decomposition. By doing this, one obtains the heat balance equation for the intra-HVAC-domain system decomposition that is more accurate. This results in

$$T_z^{n+1} = \frac{\sum_{i=1}^{N_{HG}} \dot{Q}_i^{n+1} + G + (\dot{m}c_p)_{sys}^{n+1} T_{sup}^{n+1} - A + C^{n+1} T_z^n}{B + F + (\dot{m}c_p)_{sys}^{n+1} + C^{n+1}}, \quad (14)$$

which is used to update the EnergyPlus zone air temperature for the inter-domain system decomposition whenever the rate of heat transfer is linear in the zone temperature.

## 6.2. Strong coupling implementation

Strong coupling allows longer time steps than loose coupling for the same accuracy, but it requires an iteration between the simulators. In our prototype, EnergyPlus (i.e. simulator 1 in Figure 4) controls the iteration process. The iteration criterion is based on the difference between two subsequent received values of coupling data from TRNSYS. If the difference is greater than a specified value, EnergyPlus will request another iteration.

Let  $k = \{0, 1, 2, \dots, N_{max}\}$  denote the co-simulation iteration counter, where  $N_{max}$  is the maximum number of iterations. Then the EnergyPlus temperature correction formula, for co-simulation using the strong coupling strategy, can be written as

$$T_{z,k+1}^{n+1} = \frac{\sum_{i=1}^{N_{HG}} \dot{Q}_{i,k}^{n+1} + G + (\dot{m}c_p)_{sys,k}^{n+1} T_{sup,k}^{n+1} - A}{B + F + (\dot{m}c_p)_{sys,k}^{n+1}}. \quad (15)$$

The internal heat gain  $\sum_{i=1}^{N_{HG}} \dot{Q}_{i,k}^{n+1}$  contains the system heat rate calculated in TRNSYS in the  $k$ -th iteration step. In intra-domain decomposition, internal gains remain unchanged in subsequent iterations, while the coupling data from TRNSYS are implemented by the term  $(\dot{m}c_p)_{sys,k}^{n+1} T_{sup,k}^{n+1}$ .

In some cases, small change in the sensed variable can generate a large change in system output (e.g. if the system in TRNSYS is oversized, controllers are not tuned correctly or if the simulation employs a large time step). This can lead to a non-convergent solution of the coupled system of equations. To alleviate this

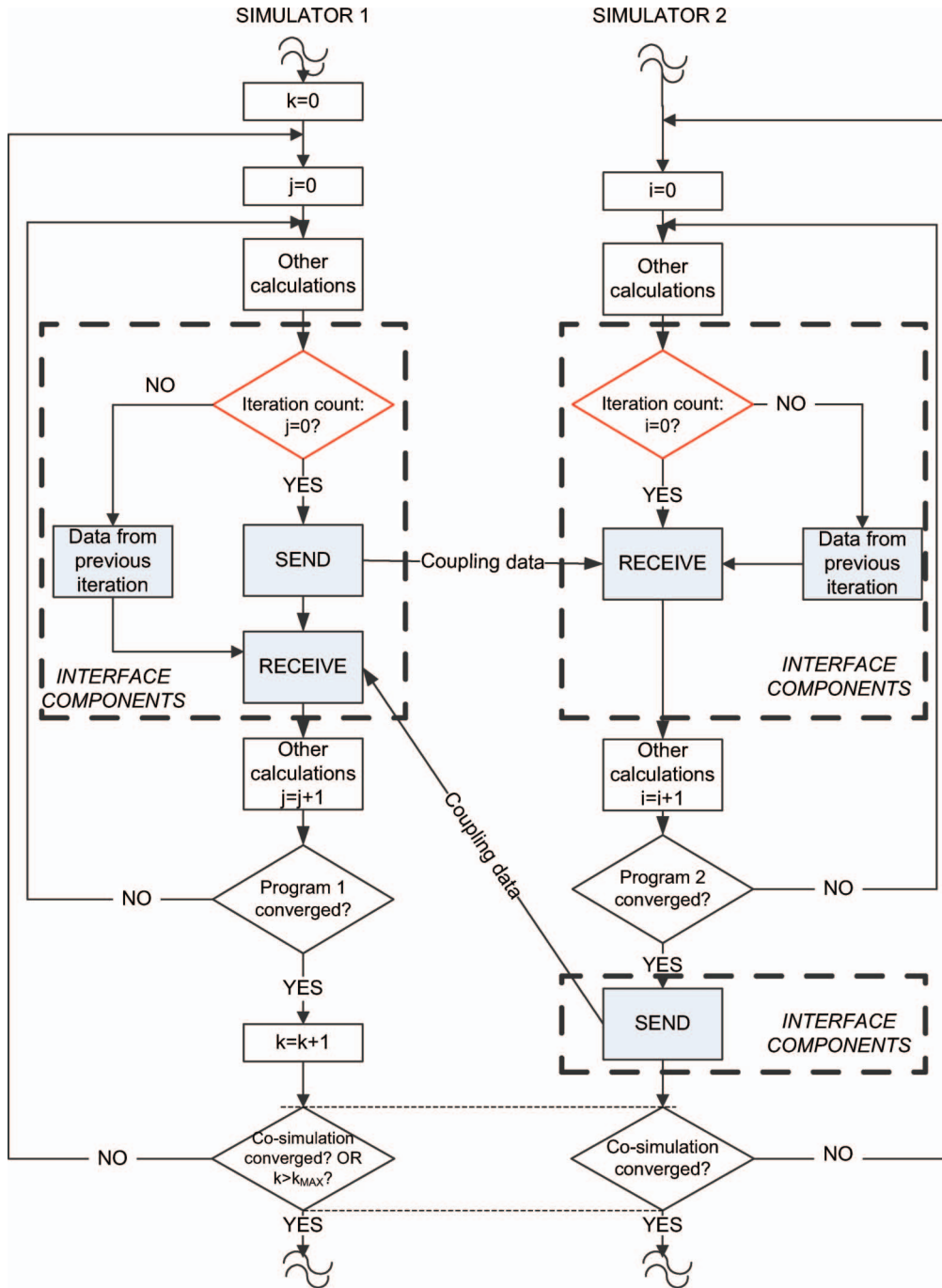


Figure 4. Flow-chart of the strong coupling implementation.

problem, the sensed variable (e.g. the zone temperature) is relaxed using  $\tilde{T}_{z,k+1,relaxed}^{n+1} = 0.25 T_{z,k+1}^{n+1} + 0.75 T_{z,k}^{n+1}$ . Using a larger coefficient for  $T_{z,k}^{n+1}$  generally resulted in faster convergence.

### 6.3. Practical implementation

The approach undertaken in this research was to implement interface components for co-simulation

within each simulator, while respecting their individual software architecture (e.g. in TRNSYS, modularization in TYPES). The interface components are used to communicate coupling data to other simulators and to control the numerical solution procedure of the coupled simulation. In TRNSYS, a proforma is created for each interface component. Each component can thus be incorporated to a TRNSYS model by a simple drag and drop action. In EnergyPlus, a

description of each interface component is added to the .idd file and thus each component can be used from EP-Launch.

Shared memory is used for interprocess communication. A portion of the shared memory is reserved for co-simulation control and synchronization. Boolean values are used to indicate the availability of new coupling data for communication in both directions.

A fixed coupling time step is set by the user before the simulation. The EnergyPlus results presented in this article were obtained with the adaptive time step feature disabled. However, in principle, it is possible to use our co-simulation implementation with an adaptive time step in an individual simulator as long as the data exchange is done at a fixed time step. In this situation, the coupling data would be held constant between the synchronization steps.

For loose coupling, multi-rate co-simulation (the case when coupled simulators are executed using different time steps) is possible provided that the coupling time step is a multiple of the time steps used in each of the simulators.

The procedure for performing a co-simulation may be as follows. First, each coupled subsystem is modelled in the corresponding tool. Interface variables that will be obtained from the other tool may be specified by a time series, or held fix, while developing the subsystem models. Next, the subsystems models are linked to the interface components for co-simulation. Then, the co-simulation is performed, and, finally, the results are analysed using the output processing facility of the individual tools.

The prototype enables general co-simulation. However, it is still to be discussed with the software developers how to add the co-simulation feature to a future release of EnergyPlus<sup>3</sup> and TRNSYS.

## 7. Verification and validation

Verification of the correctness of the implementation was performed by examining the structure of the program and by executing the computer model under different conditions. It was shown that the interfaces between the coupled simulators are implemented correctly. However, to test whether we selected the correct coupling data and communication, a validation study was done.

The main objective of this numerical validation study is to show that for a given building system, the results obtained by co-simulation agree with the results obtained by mono-simulation. In combination with analytical results discussed in the earlier sections, this should provide confidence in our co-simulation implementation. For our validation, we assume that the

individual BPS tools have been previously validated and hence, we focus our effort only on the validation of the co-simulation. Thus, the main goal is not to validate coupled state of the art simulators, but to validate the coupling itself.

The traditional BPS validation procedures are designed to test the validity of a single BPS tool. If, however, the coupled simulators were both successfully validated using the same validation technique, then they could be used to validate the coupling. However, there are two obstacles in doing so:

- Most of the traditional validation procedures for BPS tools are concerned with validating
  - (i) either only the BPS model, e.g. using inter-model comparison techniques (Judkoff and Neymark 1995, ASHRAE 2004), or
  - (ii) a single HVAC component model using empirical validation or inter-model comparison (Hensen 1991).

They are not applicable for our situation since we need a BPS tool that is validated for an integrated system that simultaneously solves a building and its HVAC system using mono-simulation.

- For the validation of a building and HVAC system that are simulated simultaneously, HVAC BESTEST (Neymark and Judkoff 2002, 2004) can be used. HVAC BESTEST Volume 1 (Neymark and Judkoff 2002) considers steady-state tests that can be solved with analytical solutions. Volume 2 (Neymark and Judkoff 2004) includes hourly simulations for dynamic effects, and other cases that cannot be solved analytically. It has been used to validate several state of the art BPS tools.

However, even though Volume 2 was used to validate EnergyPlus, the standard TRNSYS version (used in the prototypes) has been validated using only Volume 1 cases (Kummert *et al.* 2004). HVAC BESTEST Volume 2 cases were used to validate a custom version of TRNSYS that includes new code developed by the Technische Universität Dresden (TUD).

Thus, the coupling could not be validated using the traditional BPS validation procedures. However, as a preliminary validation test, the HVAC BESTEST E300 case was used with the available TRNSYS models, which do not fully correspond to the HVAC BESTEST requirements and have not been previously validated.

Further, a two-step procedure, based on the inter-model comparison technique, has been done. First, to avoid the influence of differences in different simulation models, only one simulator is used for both the mono- and the co-simulation model. Second, results obtained by

different co-simulation implementations are compared with each other. Since, the different implementations are independent of each other, this comparison can be used to further increase the confidence in the co-simulation approach.

7.1. HVAC BESTEST E300 case

For the preliminary assessment of the validity of the coupling implemented in the prototypes, the reference case (E300) from the HVAC BESTEST Volume 2 was used. The case E300 consists of a simple zone model with adiabatic envelope and without any heat capacitance and a unitary split system (Neymark and Judkoff 2004). The indoor fan is a single-speed draw-through that continuously operates. A mixed air flow (15% outside air) is blown through the evaporator. The outdoor condenser fan cycles ON and OFF together with the compressor. The controls for this system are ideal in the sense that the equipment is assumed to maintain the zone set point ( $T_{set} = 25^{\circ}\text{C}$ ) exactly when it is operating and not overloaded.

To reduce potential modelling errors, the zone model was taken from the original EnergyPlus model of E300. The system was modelled using TRNSYS Type 665. The results are compared against the results obtained by different BPS tools.

The co-simulation model of the case HVAC BESTEST E300 with EnergyPlus and TRNSYS

cannot be used with an ideal system control for the following two reasons. First, the system modelled in TRNSYS has an ON/OFF control and not a continuous control. Second, co-simulation does not allow the distributed ideal control modelling. Thus, a model of a realistic controller was used. However, using the realistic controller with the fast responding building with low-capacitance envelope and a steady-state HVAC system model results in oscillatory zone temperatures even for small time steps. Consequently, the co-simulation model does not exactly replicate the E300 model, which should be taken into account when comparing the results. Thus, for the comparison the averaged hourly data are used.

The simulations were performed using inter-domain system decomposition and loose coupling strategy. The justification of the choice for this co-simulation strategy will be given in Section 7.3, where different co-simulation strategies will be compared. The temperatures of one-day (28 June) simulation are shown in Figure 5. Similar results were obtained for loads and humidity ratios. The simulation was performed with a time step of 1 min, and the results shown are hourly averaged values. The figure shows good agreement between co-simulation result and the results obtained by other tested BPS tools.<sup>4</sup> This preliminary validation test gives confidence that the co-simulation implementation provides valid results. This confidence gives a solid ground for further validation.

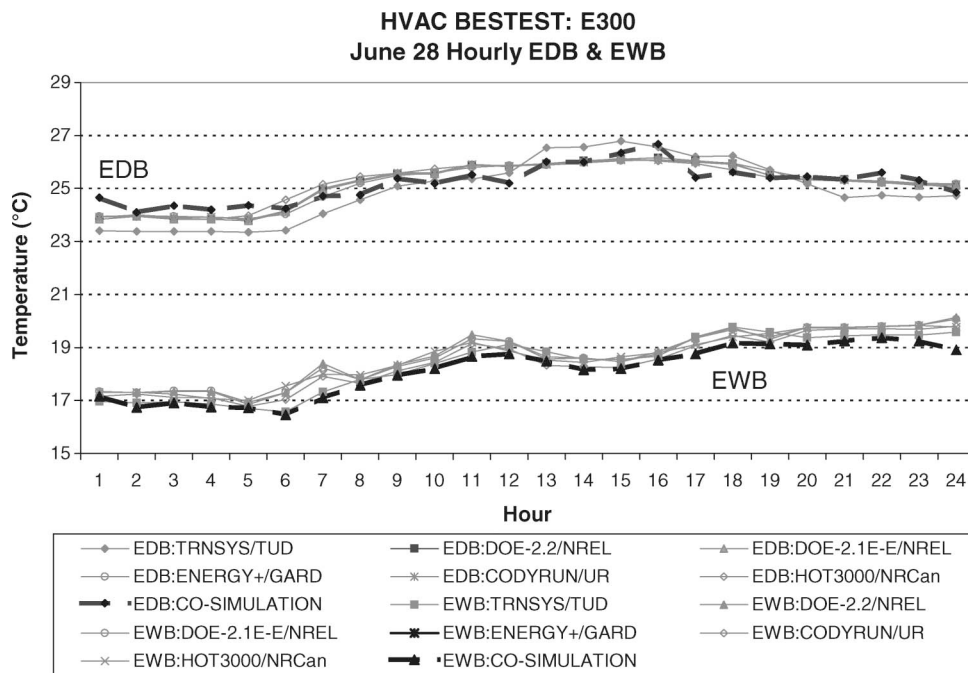


Figure 5. Zone (evaporator entering) dry-bulb (EDB) and wet-bulb (EWB) temperature for a specific simulation day.

**7.2. Comparison of mono- and co-simulation**

Another approach to validate the coupling would be to compare results obtained by co-simulation with those obtained by mono-simulation. However, the difference in results between the mono- and the co-simulation, using different tools, would be caused not only by the coupling, but also by differences in the models and solution techniques used in the coupled simulators. To avoid errors caused by different models, only one simulator was used for both the mono- and the co-simulation model.

The system presented in Figure 6 was used to validate the co-simulation prototypes that implement the loose coupling strategy. ESP-r has been modified to allow communication in both directions as shown in Figure 6.

For mono-simulation, the whole system is presented by one ESP-r model, i.e. all components are modelled and simulated in a monolithic traditional way. For co-simulation, the system is decomposed into two subsystems, presented separately by two different ESP-r models, which are then co-simulated. One model is executed using ESP-r as a base simulator, and the other using ESP-r as an external simulator. The exemplar building from ESP-r, consisting of three zones, a reception, an office and an attic, is used.

The specifications of the system in Figure 6 are (if not stated differently) as follows. The fans are identical, with the nominal air volume flow rate kept constant

throughout the simulation. The maximum cooling capacity is set according to the calculated annual maximum, obtained with the climate file for Palermo, Italy. Proportional controller, with the proportional band set to 2 K, is used to regulate the zone temperature by actuating the heat flux in the coil. The set point temperatures and the available cooling capacities for three periods of the day are shown in Table 1.

Using a time step of 1 min for both the mono- and the co-simulation, the difference between the results is negligible (e.g. see Figure 7 for temperature comparison).

With larger time steps, the influence of the time lagging of the coupling data is more noticeable. To illustrate this, the resulting zone temperatures from mono- and co-simulation obtained using a time step of 30 min are presented in Figure 8.

With an appropriately chosen coupling time step, the differences between results of a mono-simulation and results of a co-simulation as well as the differences between results of different co-simulation implementations are small. The co-simulation can produce results of the same accuracy as the mono-simulation.

**7.3. Comparison of different co-simulation implementations**

The implementations of loose and strong coupling, and of intra- and inter-domain decomposition, are independent of each other. A comparison of results

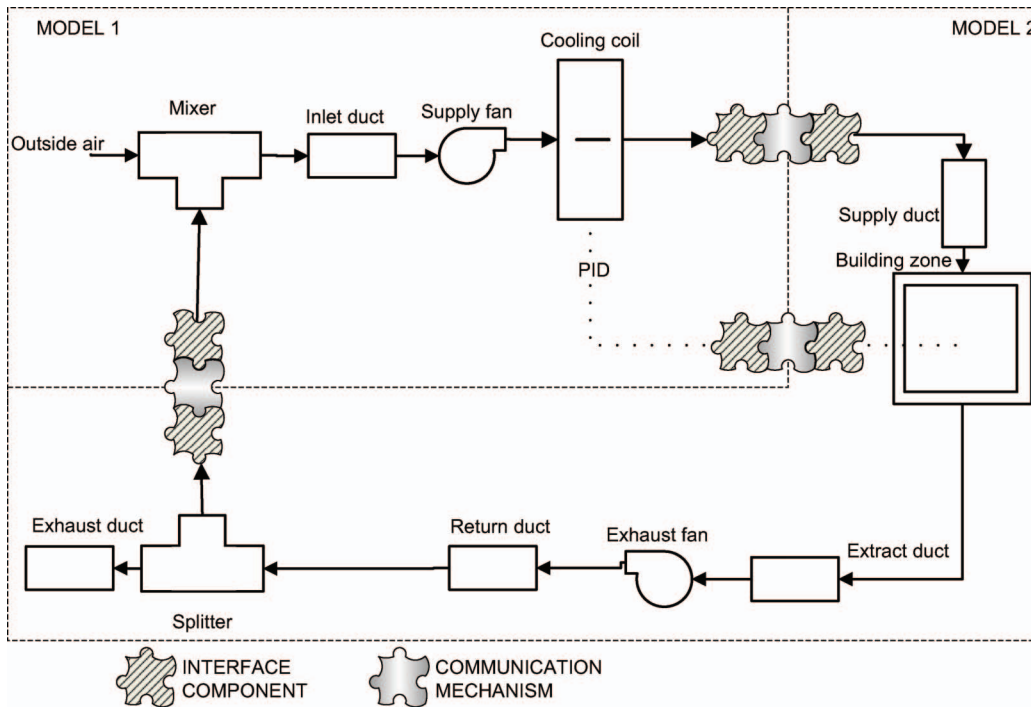


Figure 6. Sketch of the decomposed system being (co)simulated.

obtained by different implementations was done to increase the confidence in co-simulation.

The building and system presented in Figure 9 was used in the validation. The building model was built in EnergyPlus and the air system model in TRNSYS.

The system consists of (i) a cooling coil, which was sized to match the zone load by adjusting the number and dimensions of tubes, fins and rows, (ii) a constant flow fan, and (iii) a variable flow cooling water pump. The water mass flow rate was proportionally controlled to maintain the zone temperature set point,

Table 1. Zone temperature set points and available capacities.

Day period	Cooling set point temperature	Available cooling capacity
0 h–7 h	27°C	2000 W
7 h–18 h	24°C	4700 W
18 h–24 h	27°C	2000 W

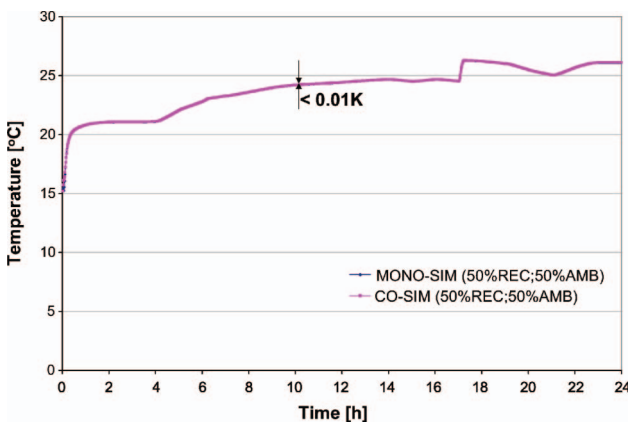


Figure 7. Zone temperature obtained by mono- and co-simulation with  $\Delta t = 1$  min.

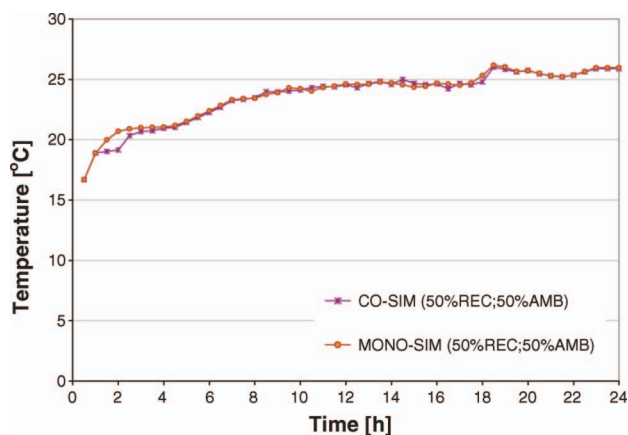


Figure 8. Zone temperature obtained by mono- and co-simulation with  $\Delta t = 30$  min.

with the lower limit of 24°C and the upper limit of 26°C. The inlet cooling water temperature was kept constant at 6°C. The system was operating from 7 h to 19 h. The nominal value of the water flow rate was either 720 kg/h or 1800 kg/h. The first nominal value corresponds to the maximal cooling demand for the simulated period, the second value was used to demonstrate the effect of an oversized system on stability of the co-simulation. Weather data for Denver, Colorado, was used and the simulation period was set to two working days from 1 to 2 August.

The simulations are done using different building and HVAC co-simulation approaches, such as

- loose coupling and intra-domain system decomposition,
- loose coupling and inter-domain system decomposition,
- strong coupling and intra-domain system decomposition, and
- strong coupling and inter-domain system decomposition.

In intra-domain system decomposition, the ducts including the return duct, mixing of the return and supply air and supply duct, were modelled and simulated in EnergyPlus. The fan, cooling coil and the proportional controller were modelled and simulated in TRNSYS.

Figure 10 shows the results of the numerical experiments, conducted with a time step of 1 min. The figure shows almost identical zone temperatures for all four combinations of the system decomposition and the coupling strategies. Similar results are obtained also for cooling water flow rates (see Figure 11). In the loosely coupled co-simulation, the oscillations of the results occur because of the use of the lagged coupling data. The difference between the inter- and the intra-domain system decompositions is due to the nature of the coupling implementation already discussed.

Co-simulation employing the strong coupling strategy with larger time steps exhibited convergence problems. However, if relaxation is used in the iterative procedure that solves for the coupling variables, good agreement with the reference results were obtained even if larger time steps were used and the system was oversized (see Figure 12).

The accuracy of co-simulation is influenced by the system decomposition approach. An intra-domain system decomposition performs better than an inter-domain system decomposition. However, the use of the suggested correction of the calculation (see Section 6.1) improves the accuracy of co-simulation of a system decomposed between the HVAC and building domains (see Figure 13).

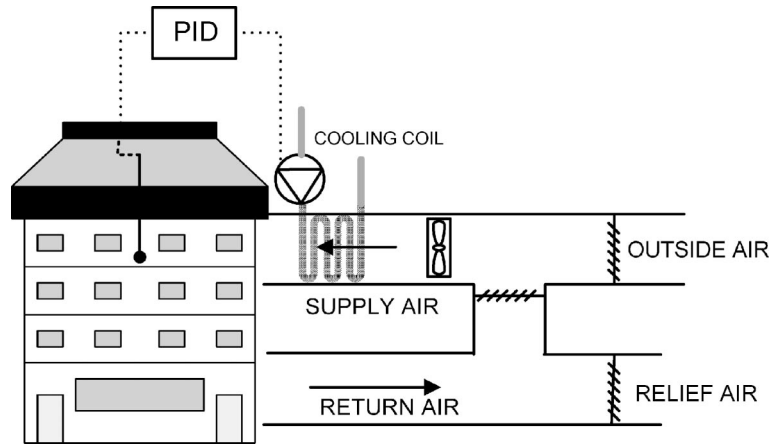


Figure 9. Sketch of the exemplar building used for the validation.

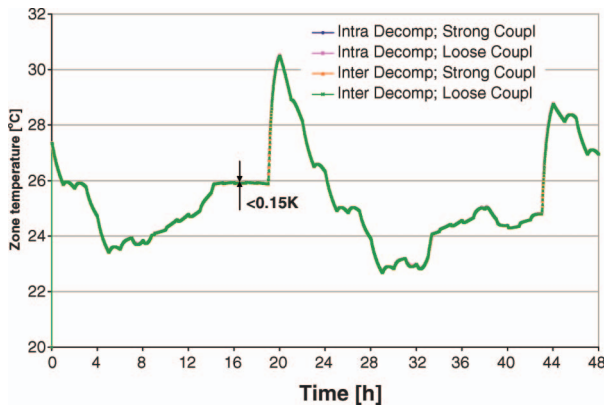


Figure 10. Zone temperature, with  $\Delta t = 1$  min and  $\dot{m}_{\text{water,nominal}} = 720$  kg/h; for all co-simulation approaches.

To improve the accuracy of the loose coupling strategy at larger time steps, a linear extrapolation of the delayed coupling data has been tested. However, because the coupling data oscillated from one time step to another, the tested first order predictor did not improve accuracy in our experiments (see Figure 14).

The shortest computation time was obtained for loosely coupled co-simulation. Using strong coupling with relaxation, stable and accurate results are achieved even with large time steps, but the computation time was high because of the iterations. In summary, with respect to computation time and ease of implementation, we recommend the loose coupling strategy with small time steps.

Next, we analysed accuracy versus computation time. To measure accuracy we computed a root mean square of the difference in the zone temperature and the reference zone temperature, which was obtained with strong coupling approach and a time step of 1 min. The simulations were performed using inter-domain decomposition and loose coupling strategy.

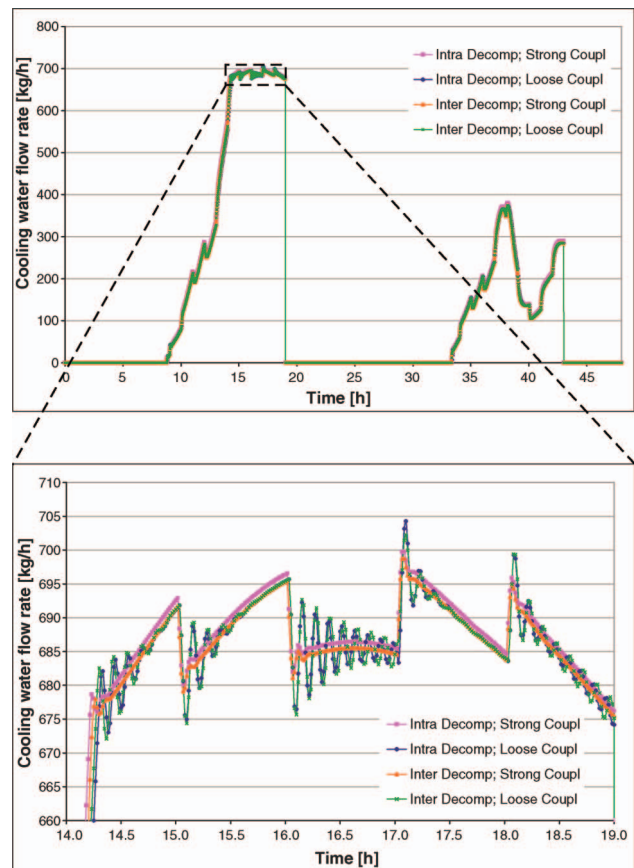


Figure 11. Cooling water flow rate, with  $\Delta t = 1$  min and  $\dot{m}_{\text{water,nominal}} = 720$  kg/h; for all co-simulation approaches.

The simulation time was 7 days. Figure 15 shows the accuracy versus computation time. The computation time was measured using the wall-clock time from the time point when shared memory has been initiated to the time point when the calculations have been finished.



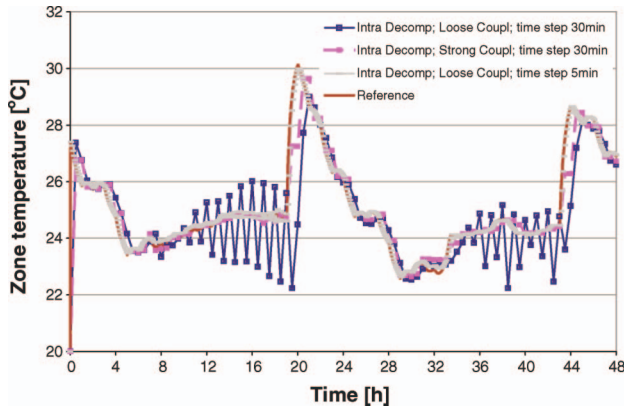


Figure 12. Zone temperature, for  $\dot{m}_{\text{water,nominal}} = 1800 \text{ kg/h}$ ; intra-domain system decomposition, using loose coupling ( $\Delta t = 5$  and  $30 \text{ min}$ ) and strong coupling (with relaxation) ( $\Delta t = 30 \text{ min}$ ).

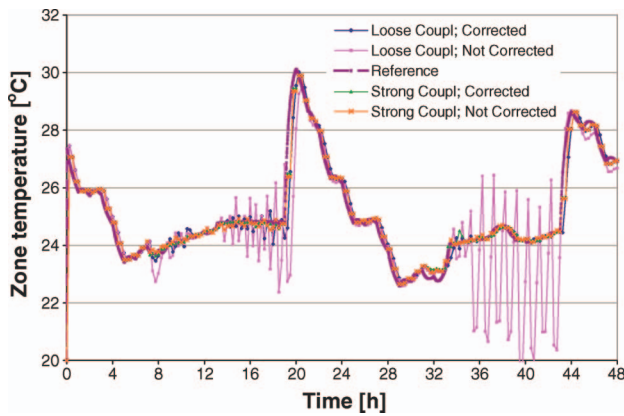


Figure 13. Zone temperature, with  $\Delta t = 15 \text{ min}$  and  $\dot{m}_{\text{water,nominal}} = 1800 \text{ kg/h}$ ; for inter-domain sys. dec., using loose and strong coupling (with relaxation), with and without correction.

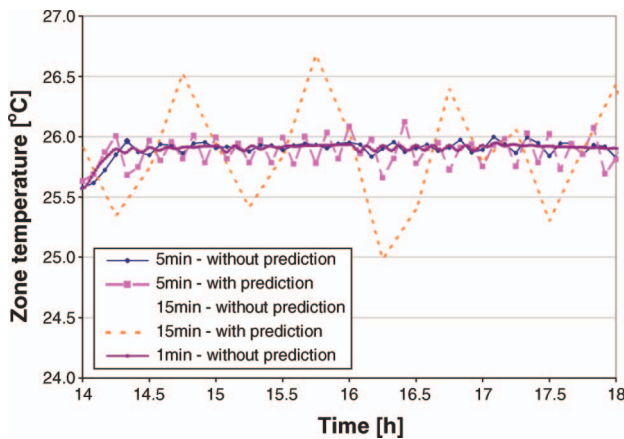


Figure 14. Zone temperature, with  $\Delta t = 5$  and  $15 \text{ min}$  and  $\dot{m}_{\text{water,nominal}} = 720 \text{ kg/h}$ ; for inter-domain sys. dec., using loose coupling, with and without prediction of/the coupling data.

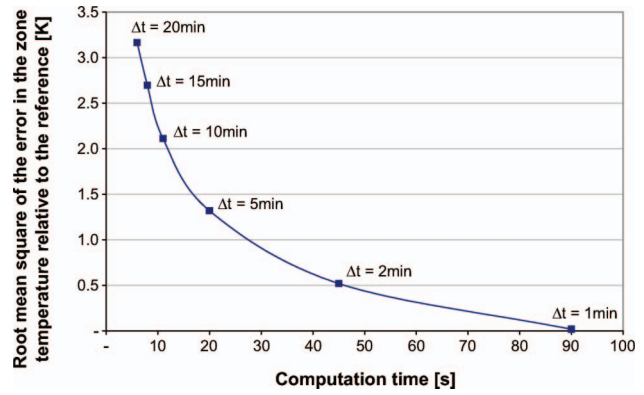


Figure 15. Accuracy versus computation time.

### 8. Case study

Several case studies were performed using the prototypes and are reported in Radošević *et al.* (2005), and Trčka-Radošević *et al.* (2006). They serve as a proof-of-concept and demonstrate the applicability and benefits of co-simulation. We will here present a case study of hybrid ventilation with evaporative cooling and run-around heat recovery.

Figure 16 shows the system that was used in this example. All cooling is done using adiabatic evaporative cooling in the exhaust air stream. The only means to provide heating is to use the heat recovery. The dynamic wind pressure is used to force air through the system when the wind pressure is sufficiently high. During periods when neither cooling nor heating is required, the pressure drop in the system is decreased by opening dampers that allow bypassing the heat exchangers. Also, the heat exchangers are bypassed when the outside temperature is sufficiently low for direct free cooling.

To achieve a sufficient fresh air supply at low wind speeds, the system was designed for low pressure drop by using large ducts and displacement ventilation which requires less kinetic energy than conventional overhead air distribution.

The building is modelled in EnergyPlus (V1-2-2). It makes use of EnergyPlus' capability to control the electric room lighting according to the daylight illumination level. EnergyPlus also has a displacement ventilation model for heat transfer and vertical temperature profile prediction. However, the system in Figure 16 cannot be modelled in EnergyPlus for the following reason: EnergyPlus models an HVAC system as a series of modules connected by fluid loops as shown in Figure 17. The fluid loops are divided into a supply and a demand side. As shown, the plant supply side cannot be connected directly to the air loop. However, such a connection is needed to implement the

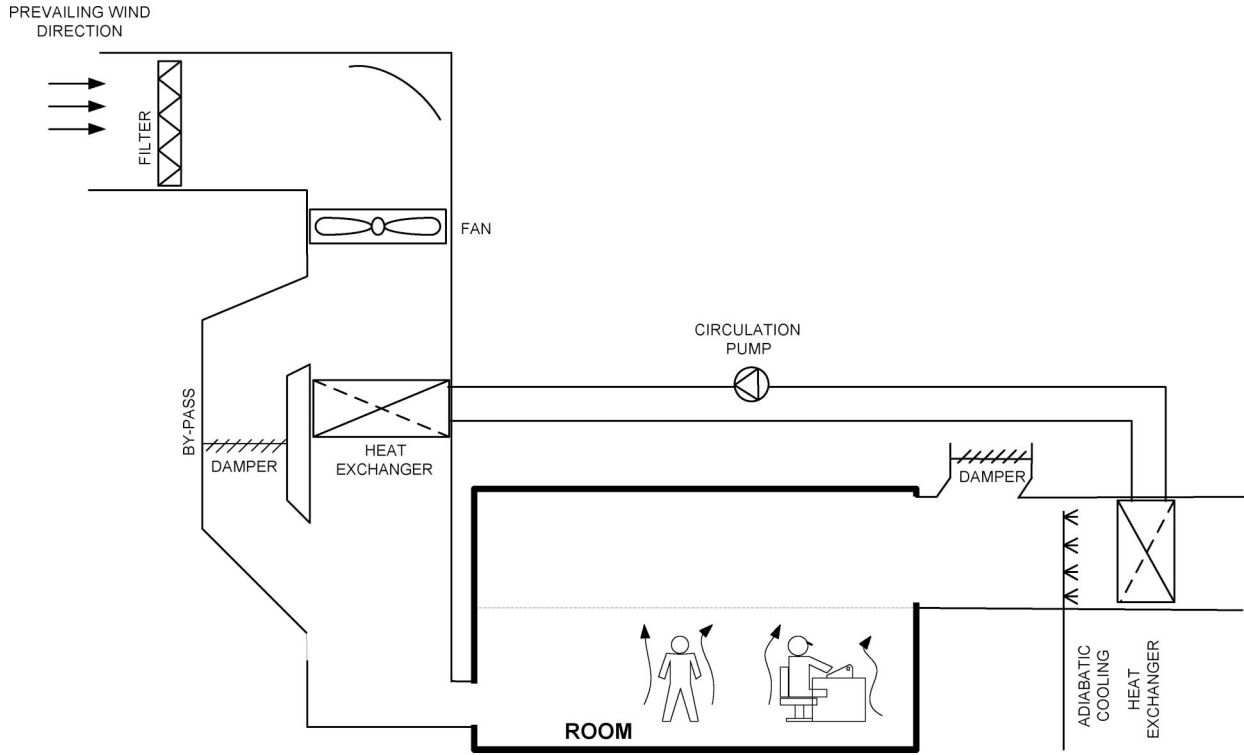


Figure 16. System schematic.

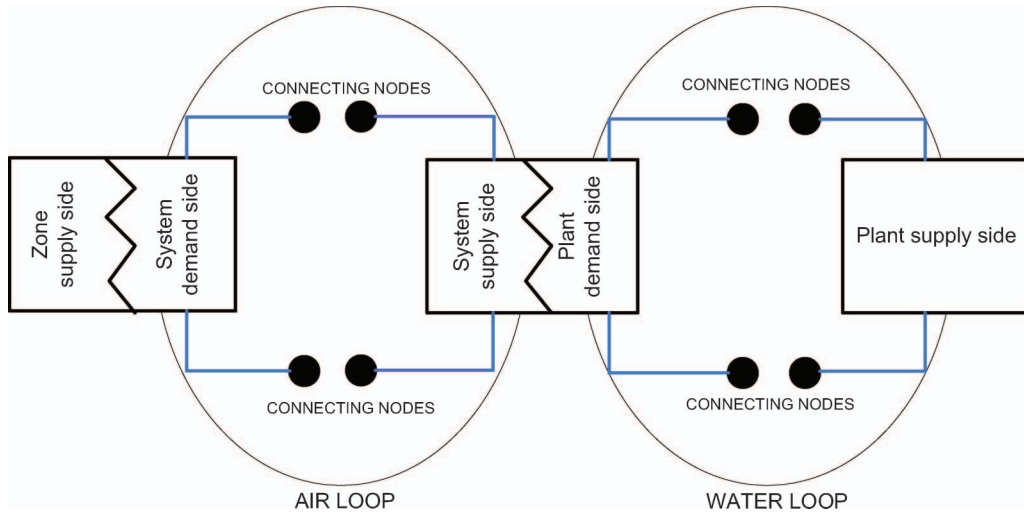


Figure 17. Fluid loops in EnergyPlus.

run-around heat recovery since the coil in the exhaust duct is, at the same time, a plant supply side component for the heat exchanger in the outside air intake.

Thus, we used TRNSYS to model the mechanical system, and coupled it to EnergyPlus to allow an integrated simulation. The simulations were performed using weather data for Palermo, Italy. The system air intake direction is in the prevailing wind direction (north-east).

### 8.1. Thermal comfort analysis

To estimate the thermal comfort, simulations are performed for a summer week (1 August – 7 August), a winter week (9 January – 15 January) and a fall week (4 October – 10 October). The resulting temperatures in the occupied zone, for the three simulation periods, are shown in Figures 18–20, respectively. The jumps in the zone temperature at the point when the system is

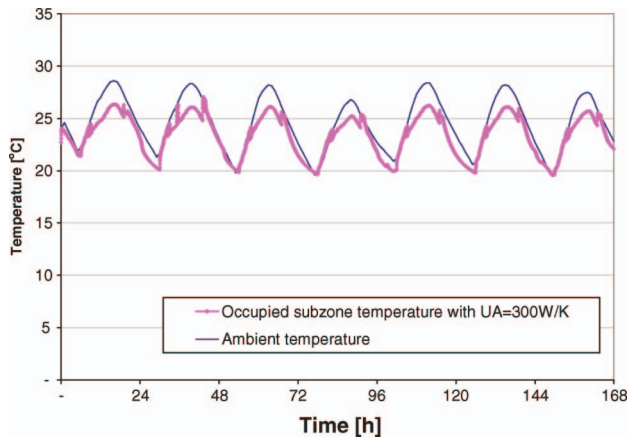


Figure 18. Simulation results for the summer week.

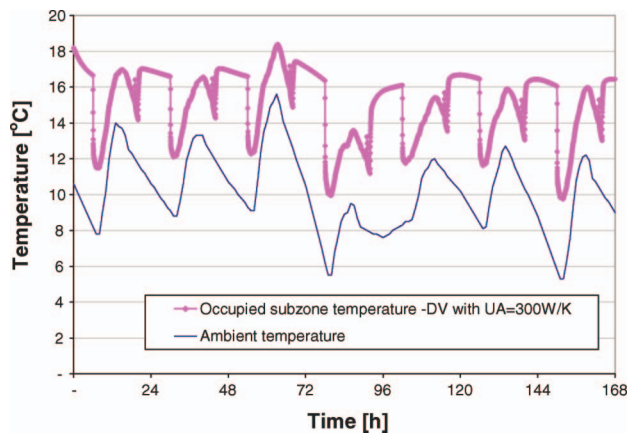


Figure 19. Simulation results for the winter week.

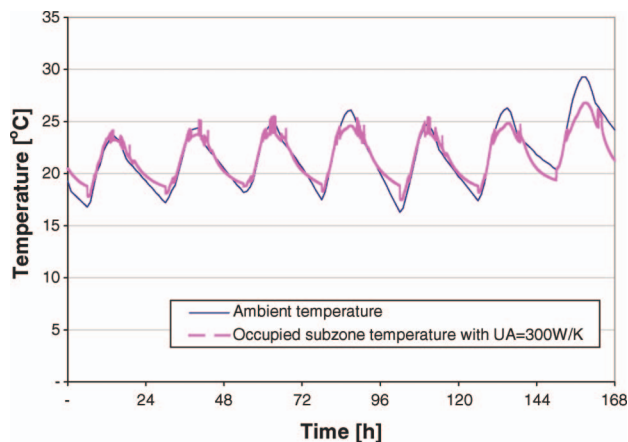


Figure 20. Simulation results for the fall week.

switched off are caused by the simplified displacement ventilation model. The temperature gradient is only calculated during hours when the system is operating. Outside of these hours, the mixed zone air temperature is reported by the simulation program.

During the occupied period in the summer week, the temperature in the occupied subzone is below 27°C.

As can be seen, the capacity of the heating coil is not sufficient during the colder periods. Therefore, the system without additional heating will fail to provide sufficient thermal comfort to the occupants in the zone.

In Figure 20, the oscillations in the occupied sub-zone temperature result from redirecting the flow through the by-pass when cooling is not required.

## 8.2. Energy consumption analysis

The fan augments the wind induced pressure if required to provide sufficient outside air. It has a constant flow rate and an efficiency of 60%. To estimate the fan energy consumption, simulations are performed for the same weeks in summer, fall and winter. The energy consumption was calculated for the system with and without the bypass in the air streams.

On a single windy day (e.g. 10 October), the energy savings were 17%. As the wind speed and direction fluctuate, the savings averaged over one week were lower: 3–8%, depending on the season. The savings are the highest for the fall period. The reason for this is that the heating requirements are lower than during the winter, which allowed for more hours where the heat exchangers were bypassed. In addition, free cooling by the ambient air can be used during more hours than in summer.

## 9. Conclusions

In this article, we discussed principles of co-simulation, reported our development and implementation, compared the numerical and computational performance of different co-simulation implementations, and tested the usability of co-simulation for performance prediction of innovative integrated energy systems in buildings.

Analysis of loosely coupled co-simulation showed that the partitioned numerical schemes used to approximate solution of a system of ordinary differential equations are zero-stable, consistent and thus convergent.

Numerical experiments have shown that strong coupling allows using longer time steps than loose coupling at the same accuracy. It was also shown that loosely coupled co-simulation with sufficiently small time steps can generate results with the same accuracy as mono-simulation.

Based on the computation time and ease of implementation, the loose coupling strategy with smaller time steps is recommended. In our comparison

of different co-simulation strategies, we observed shorter computation time for inter-domain system decomposition.

The use of predictors for the coupling data does not enhance the accuracy of loose coupling when the coupling data oscillates from one time step to another. Otherwise, the use of predictors may introduce additional numerical errors.

The developed co-simulation prototypes were used in a case study, which showed that co-simulation enables the combination of complementary features that are available in the coupled tools. Co-simulation facilitated a fully integrated design analysis, which would not have been possible if any of the BPS tools were used individually.

### Acknowledgements

This research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies of the US Department of Energy, under Contract No. DE-AC02-05CH11231.

### Notes

1. A similar discussion holds for the correction formula for the zone humidity ratio.
2. The value of the corresponding coefficient should be provided by the coupled simulator and is assumed constant from one time step to the next.
3. The co-simulation prototype is based on an old version of EnergyPlus (V1-2-2).
4. The test also revealed one error in the TRNSYS type 665. The percentage of the outside air in the supply air flow was not correctly taken into account. The error has been fixed. The code of the type 665 was also adjusted so that only indoor fan power is included in the performance data.

### References

- Aasem, E.O., 1993. *Practical simulation of buildings and air-conditioning systems in the transient domain*. Thesis (PhD). University of Strathclyde, Glasgow, Scotland.
- Arnold, M., et al., 2002. Modular dynamical simulation of mechatronic and coupled systems. In: *Proceedings of 5th World Congress on Computational Mechanics*, Vienna, Austria, Vienna University of Technology.
- ASHRAE, 2004. ANSI/ASHRAE Standard 140-2004, *Standard Method of Test for the Evaluation of Building Energy Analysis Computer Programs*. Technical report, The American Society of Heating, Refrigerating and Air-Conditioning Engineers.
- ASHRAE, 2008. *ASHRAE 2020 Vision, Position, Strategies, Actions, 2008*. Technical report, The American Society of Heating, Refrigerating and Air-Conditioning Engineers. Available at: [http://www.ashrae.org/doclib/20080226\\_ashraevision2020.pdf](http://www.ashrae.org/doclib/20080226_ashraevision2020.pdf) [Accessed April 2008].
- Boer, C.A., 2005. *Distributed simulation in industry*. Thesis (PhD). Erasmus University Rotterdam.
- Boukerche, A., Mikkler, A., and Fabri, A., 2005. Resource control for large-scale distributed simulation system over loosely coupled domains. *Journal of Parallel and Distributed Computing*, 65 (10), 1171–1189.
- Brooks, C., et al., 2007. *Ptolemy II – heterogeneous concurrent modeling and design in Java*. Technical report No. UCB/EECS-2007-7, University of California at Berkeley, Berkeley, CA.
- Carroll, W.L. and Hitchcock, R.J., 2005. DELIGHT2 daylighting analysis in EnergyPlus: Integration and preliminary user results. In: *Proceedings of 9th International IBPSA Conference*, Montréal, Canada International Building Performance Simulation Association, 139–144.
- Djunaedy, E., 2005. *External coupling between building energy simulation and computational fluid dynamics*. Thesis (PhD). Technische Universiteit Eindhoven.
- Djunaedy, E., Hensen, J.L.M., and Loomans, M.G.L.C., 2003. Towards external coupling of building energy and air flow modeling programs. *ASHRAE Transactions*, 109 (2), 771–787.
- Duggan, J., 2002. A distributed computing approach to system dynamics. *System Dynamics Review*, 18 (1), 87–98.
- Elliott, A.S., 2000. A highly efficient, general-purpose approach for co-simulation with ADAMS. In: *Proceedings of 15th European ADAMS User's Conference*, Rome, Italy.
- Elliott, M., 2004. *Guide to citing Internet sources* [online]. Poole, UK: Bournemouth University. Available from: [http://www.bournemouth.ac.uk/library/using/guide\\_to\\_citing\\_internet\\_sour.c.html](http://www.bournemouth.ac.uk/library/using/guide_to_citing_internet_sour.c.html) [Accessed September 2006].
- Felippa, C.A., Park, K.C., and Farhat, C., 1999. *Partitioned analysis of coupled mechanical systems*. Technical report CU-CAS-99-06, Center for Aerospace structures, University of Colorado.
- Follen, G., et al., 2001. A CORBA-based development environment for wrapping and coupling legacy scientific codes. In: *Proceedings of 10th IEEE International Symposium on High Performance Distributed Computing*, San Francisco, CA, USA: The Institute of Electrical and Electronics Engineers (IEEE), 22–31.
- Fujimoto, R.M., 1998. Time management in the high level architecture. *Simulation*, 71 (6), 88–400.
- Fujimoto, R.M., 2003. Distributed simulation systems. In: *Proceedings of 2003 Winter simulation conference*, New Orleans, Louisiana, USA: The Institute of Electrical and Electronics Engineers (IEEE), 124–134.
- Gear, C.W., 1971. *Numerical initial value problems in ordinary differential equations*. Englewood Cliffs, NJ: Prentice-Hall.
- Golub, G.H. and Ortega, J.M., 1992. *Scientific computing and differential equations*. London, UK: Academic Press.
- Gravouil, A. and Combescure, A., 2001. Multi-time-step explicit-implicit method for non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, 50, 199–225.
- Grott, M., Chernigovski, S., and Glatzel, W., 2003. Simulation of stellar instabilities with vastly different time-scales using domain decomposition. *Monthly Notices of the Royal Astronomical Society*, 344, 1119–1130.
- Hensen, J.L.M., 1991. *On the thermal interaction of building structure and heating and ventilating system*. Thesis (PhD). Eindhoven University of Technology.

- Hensen, J.L.M., 1999. A comparison of coupled and decoupled solutions for temperature and air flow in a building. *ASHRAE Transactions*, 105, 962–969.
- Hensen, J.L.M. and Clarke, J.A., 2000. Integrated simulation for HVAC performance prediction: state of the art illustration. In: *Proceedings of Int. ASHRAE/CIBSE Conf. '20-20 Vision'*, Dublin, Ireland.
- Hensen, J.L.M., et al., 2004. Building performance simulation for better design: Some issues and solutions. In: *Proceedings of 21st Conference on Passive and Low Energy Architecture*, Technische Universiteit Eindhoven, The Netherlands, 1185–1190.
- Hillestad, M. and Hertzberg, T., 1986. Dynamic simulation of chemical engineering systems by the sequential modular approach. *Computers and Chemical Engineering*, 10 (4), 377–388.
- Hillestad, M. and Hertzberg, T., 1988. Convergence and stability of the sequential modular approach to dynamic process simulation. *Computers and Chemical Engineering*, 12 (5), 407–414.
- Huang, J., et al., 1999. Linking the COMIS multi-zone airflow model with the EnergyPlus Building Energy Simulation Program. In: *Proceedings of 6th International IBPSA Conference*, Kyoto, Japan: International Building Performance Simulation Association.
- Janak, M., 1999. Coupling building energy and lighting simulation. In: *Proceedings of 5th International IBPSA Conference*, Vol. 2, Kyoto, Japan: International Building Performance Simulation Association, 307–312.
- Judkoff, R. and Neymark, J., 1995. *International Energy Agency Building energy simulation tests (BESTEST) and diagnostic method*. Technical report, National Renewable Energy Laboratory and national laboratory of the US Department of Energy Report of a cooperative project between IEA Solar Heating and Cooling Task 12B and IEA Energy Conservation in Buildings and Community Systems Annex 21C.
- Keilholz, W., 2002. TRNSYS World-wide. *The Journal of the International Building Performance Simulation Association, ibpsaNEWS*, 12 (1).
- Kubler, R.E.A., 2000. Two methods of simulator coupling. *Mathematical and Computer Modelling of Dynamic Systems*, 6 (2), 93–113.
- Kummert, M., Bradley, D., and Dowell, T.M., 2004. Combining different validation techniques for continuous software improvement – Implications in the development of TRNSYS 16. In: *Proceedings of ESIM 2004 Conference*, Vancouver, Canada: Canadian Chapter of the International Building Performance Simulation Association.
- Lambert, J.D., 1991. *Numerical methods for ordinary differential systems, the initial value problem*. West Sussex, UK: Wiley.
- Larsson, J., 2001. *Concepts for multi-domain simulation*. Thesis (PhD). Linköping Universitet, Sweden.
- Lee, E.A. and Varaiya, P.P., 2002. *Structure and interpretation of signals and systems*. Addison Wesley.
- Lee, J.S., 2004. High performance modeling for distributed simulation. *Lecture Notes in Computer Science*, 3397, 138–146.
- Li, N., et al., 2005. Research on construction and interoperability of complex distributed simulation system. *Lecture Notes in Computer Science*, 3398, 131–140.
- McDowell, T.P., et al., 2003. Integration of airflow and energy simulation using CONTAM and TRNSYS. *ASHRAE Transactions*, , 109 (1), 1–14.
- Neymark, J. and Judkoff, R., 2002. *International Energy Agency Building energy simulation test and diagnostic method for heating, ventilating, air-conditioning equipment models (HVAC BESTEST), Volume 1: cases E100-E200*. Technical report, National Renewable Energy Laboratory Technical report of an IEA Solar Heating and Cooling programme project.
- Neymark, J. and Judkoff, R., 2004. *International Energy Agency Building energy simulation test and diagnostic method for heating, ventilating, air-conditioning equipment models (HVAC BESTEST), Volume 2: cases E300-E545*. Technical report, National Renewable Energy Laboratory Technical report of an IEA Solar Heating and Cooling programme project.
- Page, E.H., et al., 1999. Panel: Strategic directions in simulation research. In: *Proceedings of 1999 Winter Simulation Conference*, Phoenix, Arizona, USA, The Institute of Electrical and Electronics Engineers (IEEE), 1509–1520.
- Park, K.C., 1980. Partitioned transient analysis procedures for coupled-field problems: Stability analysis. *Journal of Applied Mechanics*, 47, 370–376.
- Radošević, M., Hensen, J.L.M., and Wijsman, A., 2005. Implementation strategies for distributed modeling and simulation of building systems. In: *Proceedings of 9th International IBPSA Conference*, Montreal, Canada, International Building Performance Simulation Association, 995–1002.
- Sang, J.C., et al., 2002. Development of CORBA-based engineering applications from legacy Fortran programs. *Information and Software Technology*, 44 (3), 175–184.
- Strassburger, S., 2001. *Distributed simulation based on the high level architecture in civilian application domains*. Thesis (PhD). Otto-von-Guericke Universität, Magdeburg.
- Struler, E., et al., 2000. A new approach to software integration frameworks for multi-physics simulation codes. In: *Proceedings of IFIP TC2/WG2.5 Working Conference on Architecture of Scientific Software*, Ottawa, Canada, 87–104.
- Tacic, I. and Fujimoto, R.M., 1998. Synchronized data distribution management in distributed simulations. In: *Proceedings of the 12th Workshop on Parallel and Distributed Simulation*, Banff, Alberta, Canada: IEEE Computer Society, 108–115.
- Taylor, S.J.E., et al., 2002. Distributed simulation and industry: potentials and pitfalls. In: *Proceedings of 2002 Winter simulation conference*, San Diego, California, USA, The Institute of Electrical and Electronics Engineers (IEEE), 688–694.
- Trčka, M., 2008. *Co-simulation for performance prediction of innovative integrated mechanical energy systems in buildings*. Thesis (PhD). Eindhoven University of Technology, Eindhoven, The Netherlands.
- Trčka-Radošević, M. and Hensen, J.L.M., 2006. Distributed simulation of building systems for legacy software reuse. In: *Proceedings of 6th International Postgraduate Research Conference, International Built and Human Environment Research Week*, Technische Universiteit Delft, BuHu, University of Salford, 442–454.
- Trčka-Radošević, M., Hensen, J., and Wijsman, A., 2006. Integrated building and system simulation using run-time coupled distributed models. *ASHRAE Transactions*, 112 (1), 719–728.

- Tseng, P.H., Sciortino, A., and van Genuchten, M.T., 1995. A partitioned solution procedure for simulating water flow in a variably saturated dual-porosity medium. *Advances in Water Resources*, 18 (6), 335–343.
- Wang, W. and Beausoleil-Morrison, I., 2009. Integrated simulation through the source-code coupling of component models from a modular simulation environment into a comprehensive building performance simulation tool. *Journal of Building Performance Simulation*, 2 (2), 115–126.
- Wang, X.H., Zhang, L., and Huang, K.D., 2004. Flexible cycle synchronized algorithm in parallel and distributed simulation. *Lecture Notes in Computer Science*, 3326, 40–45.
- Weber, A., et al., 2002. TRNFLOW: Integration of COMIS into TRNSYS TYPE 56. In: *Proceedings of AIVC 23rd conference – EPIC 2002 AIVC (in conjunction with 3rd European Conference on Energy Performance and Indoor Climate in Buildings)*, Vol. 3, Lyon, France.
- Wetter, M. and Haves, P., 2008. A modular building controls virtual test bed for the integration of heterogeneous systems. In: *Proceedings of SimBuild, 3rd National Conference of IBPSA-USA*, Bekeley, CA, USA: International Building Performance Simulation Association, USA, 69–76.
- Wetter, M., 2009. Modelica-based modeling and simulation to support research and development in building energy and control systems. *Journal of Building Performance Simulation*, 2 (2), 143–161.
- Wilcox, P.A., Burger, A.G., and Hoare, P., 2000. Advanced distributed simulation: A review of developments and their implication for data collection and analysis. *Simulation Practice and Theory*, 8 (3–4), 201–231.
- Yahiaoui, A., Hensen, J.L.M., and Soethout, L., 2004. Developing CORBA-based distributed control and building performance environments by run-time coupling. In: *Proceedings of ICCCBE-X 10th International Conference on Computing in Civil and Building Engineering in Weimar*. International Society for Computing in Civil and Building Engineering, 86–93.
- Zhai, Z., 2003. *Developing an integrated building design tool by coupling building energy simulation and computational fluid dynamics programs*. Thesis (PhD). Massachusetts Institute of Technology, USA.