

# Co-simulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de  
Technische Universiteit Eindhoven, op gezag van de  
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een  
commissie aangewezen door het College voor  
Promoties in het openbaar te verdedigen  
op woensdag 8 oktober 2008 om 14.00 uur

door

Marija Trčka

geboren te Belgrado, Servië

Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. J.L.M. Hensen

Copromotor:

dr. M. Wetter

# Co-simulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter

prof.dr.ir. J.L.M. Hensen, *Technische Universiteit Eindhoven*, promotor

dr. M. Wetter, *Lawrence Berkeley National Laboratory, Berkeley, USA*, copromotor

prof.dr. J. Spitler, *Oklahoma State University, Stillwater, USA*

prof.dr.ir. A.A. van Steenhoven, *Technische Universiteit Eindhoven*

prof.dr. M. Todorović, *University of Belgrade, Belgrade, Serbia*

prof.ir. P.G.S. Rutten, *Technische Universiteit Eindhoven*

A catalogue record is available from the Eindhoven University of Technology Library

ISBN: 978-90-386-1366-6

NUR: 955

Cover design by Jelena Radošević, adapted by Jac de Kok

Printed by the Eindhoven University Press, Eindhoven, The Netherlands

Published as issue 130 in the *Bouwstenen* series of the Faculty of Architecture, Building and Planning of the Eindhoven University of Technology

© Marija Trčka, 2008

All rights reserved. No part of this document may be photocopied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means whether, electronic, mechanical, or otherwise without the prior written permission of the author.

# Acknowledgements

This thesis is an outcome of my Ph.D. research at Eindhoven University of Technology that started in July 2003. During this period my professional and my personal life were meliorated by many people in different ways. Here I only mention some of them.

First and foremost I would like to thank my supervisor Jan Hensen. Jan has always believed in me, and has been encouraging me over the years. He conveyed his enthusiasm for building performance simulation to me. Without his guidance and support there would not be this thesis. On the personal side, Jan and his wife Lada, have also become friends of ours. I greatly appreciated their warm house parties which our group had a pleasure of enjoying very often. Jan and Lada, thanks for everything.

In the third year of my research, I was lucky to be given an opportunity to spend six months at the United Technologies Research Center, in Hartford, Connecticut. There I met my co-supervisor, Michael Wetter. I would like to thank Michael for bringing my work to an upper level. He carefully read every word I wrote, and from him I have learned a lot. Not only a vast amount of computer simulation, but also how to write better, how to be more precise, and many many other things. Michael and his wife Maureen have also become our friends.

I would like to thank prof.dr. Jeffrey Spitler, prof.dr.ir. Anton van Steenhoven and prof.dr. Marija Todorović who - as members of the core committee - reviewed and approved this dissertation. I would also like to express my gratitude to the other members of the committee, dr.ir. Paul Rutten and dr.ir. Jan Westra.

I thank Aad Wijsman for his listening, the many advices he has offered and in particular for reviewing this thesis. My appreciation also goes to all the curious people who attend our monthly progress meetings for their constructive feedback.

I would like to thank my office colleagues and friends: Ery, Azzedine, Mônica, Christina, Christian, Marcel, Bert, Daniel, Mohammad, Wiebe, and many others for numerous life-enriching discussions during lunches and trips to conferences. Many nationalities and cultures mixed together in a few square meters can only make one better. I am glad to have met all of them. Special thanks go to Christina and Mônica who helped us greatly in many occasions, and who will be supporting me as 'paronyms' on the day of the defence.

During our five year stay in Eindhoven we have also made many friends outside of work. They made our stay a very enjoyable experience, filled with memories. I would like to thank Ana, whose help in the beginning is greatly appreciated, Jasen and Georgi, who were always there to help, and our friends Bojana and Arjan, Perica and Emilija, and Uroš and Milica, for making us feel like at home.

I would like to thank all my colleagues at the United Technologies Research Center, in particular Scott Bortoff and Robert LaBarre, for showing their interest in my work and for giving me constant support during my stay at the company.

Several people made our stay in Hartford a pleasant experience. I thank Aleksandar and Nataša, for being such wonderful friends and for sharing their roof

---

with us. Chris and Monika, Cornel, Jose-Miguel, Stevo, and Cesare, thanks for the very many lovely evenings.

I apologize to Aleksandra and Petar, and to Gülsu and Timuçin, for canceling our planned arrangements because of the work that I had to do to finalize this thesis.

I thank my sister Jelena for being my sister, and specially for designing the cover of this thesis.

I thank my parents and my grandmother for their endless love they always unquestionably give.

At last, but certainly not the least, I thank my husband Nikola for all the love and support that he gives to me for more than eleven years. He and our beautiful son Luka make my life complete.

Eindhoven, August 2008

Marija Trčka

# Summary

## Co-simulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings

INTEGRATED performance simulation of buildings and heating, ventilation and air-conditioning (HVAC) systems can help in reducing energy consumption and increasing level of occupant comfort. However, no single building performance simulation (BPS) tool offers sufficient capabilities and flexibilities to accommodate the ever increasing complexity and rapid innovations in building and system technologies.

The existing state of the art BPS tools can be extended but this is a difficult and costly task. Adding new features requires the tool designer to have in-depth knowledge of the tool-specific modeling strategies and software architectures. Moreover, one state of the art BPS tool often provides features that are complementary with features from another state of the art BPS tool. These features, however, can not be combined since the tools are not yet opened for communication. In addition, some domain-independent and equation-based tools are well suited for rapid model prototyping and could be used to easily create models of developing technologies. However, they typically lack the vast range of state of the art domain models, and thus can not cover whole building simulation. One way to alleviate these problems is to use co-simulation, as an integrated approach to simulation.

Co-simulation approach represents a particular case of simulation scenario where at least two simulators solve coupled differential-algebraic systems of equations and exchange data that couples these equations during run-time. The main research objective underlying this thesis is concerned with investigation, theoretical analysis, development and implementation, numerical experimentation and usability testing of co-simulation of building and HVAC system simulators.

The available strategies and tools for co-simulation are first reviewed. The issues important for co-simulation realization are studied in detail, and multiple possibilities are discussed to justify the particular implementation approach taken in the thesis. Stability and accuracy of different coupling strategies are analyzed to give a guideline for the required coupling frequency. Further, co-simulation is implemented in a software prototype using the existing state of the art BPS simulators. The implementation is verified and validated against the results obtained from the traditional simulation approach. It is used in several case studies for the proof-of-concept, to demonstrate the applicability of the method, and to highlight its benefits. Finally, based on the coupling strategy analysis and the findings from the testing of the prototype, requirements and recommendations for generic co-simulation implementation are defined.

The main results of the research show that co-simulation facilitates rapid expansion of modeling capabilities of the state of the art BPS tools. It allows various aspects of buildings to be modeled and simulated in the most appropriate tools. Compared to the traditional approach it offers increased functionality and more flexibility for integrated simulation-based analysis of innovative HVAC system technologies.



# Contents

<b>Acknowledgement</b>	i
<b>Summary</b>	iii
<b>1 Introduction</b>	1
1.1 Problem statement . . . . .	2
1.2 Objectives . . . . .	4
1.3 Hypothesis . . . . .	6
1.4 Research methodology . . . . .	6
1.5 Thesis outline . . . . .	7
<b>2 State of the art in building systems simulation</b>	9
2.1 Tools for HVAC system design and analysis . . . . .	9
2.2 Modeling approaches . . . . .	11
2.2.1 Modeling approaches for HVAC components . . . . .	11
2.2.2 Modeling approaches for HVAC control . . . . .	13
2.2.3 Modeling approaches for HVAC systems . . . . .	14
2.2.4 Solution techniques for HVAC system simulation models . .	17
2.3 Selection of HVAC modeling approach . . . . .	18
<b>3 Co-simulation - Principles and strategies</b>	21
3.1 Introduction . . . . .	21
3.2 Terminology and other issues for co-stimulation implementation .	24
3.2.1 Interface classification . . . . .	24
3.2.2 Simulator's roles . . . . .	28
3.2.3 System partitioning . . . . .	28
3.2.4 System decomposition strategies . . . . .	29
3.2.5 Coupling data . . . . .	31
3.2.6 Time management in co-simulation . . . . .	32
3.2.7 Management of simulators' execution . . . . .	33
3.2.8 Coupling strategies . . . . .	33
3.2.9 Coupling frequency . . . . .	36
3.2.10 Multi-rate co-simulation . . . . .	36
3.2.11 Inter vs. intra time step data exchange . . . . .	39
<b>4 Co-simulation - Stability and accuracy</b>	41
4.1 Numerical integration schemes . . . . .	41
4.2 Consistency . . . . .	43
4.3 Stability . . . . .	43
4.3.1 Zero-stability . . . . .	44
4.3.2 Absolute stability . . . . .	44
4.4 Convergence . . . . .	45
4.5 Consistency of co-simulation . . . . .	45
4.6 Zero-stability and convergence of co-simulation . . . . .	47

## CONTENTS

---

4.7	Analysis of co-simulation for a two-body system . . . . .	49
4.7.1	Mathematical model . . . . .	50
4.7.2	Consistency . . . . .	52
4.7.3	Absolute stability . . . . .	55
4.7.4	Measures to improve accuracy of co-simulation solution . .	60
4.8	Discussion and conclusions . . . . .	66
<b>5</b>	<b>Co-simulation - Prototypes</b>	<b>69</b>
5.1	Simulation tools used in prototypes . . . . .	69
5.1.1	Control modeling in co-simulation . . . . .	72
5.1.2	Use of variable time stepping in co-simulation . . . . .	72
5.2	Communication mechanisms used in prototypes . . . . .	73
5.3	EnergyPlus/TRNSYS prototype . . . . .	74
5.3.1	Loose coupling implementation . . . . .	74
5.3.2	Strong coupling implementation . . . . .	77
5.3.3	Changes in the EnergyPlus code . . . . .	79
5.3.4	Changes in TRNSYS code . . . . .	84
5.4	ESP-r/EARTH prototype . . . . .	86
5.4.1	Changes in ESP-r code . . . . .	86
<b>6</b>	<b>Validation</b>	<b>89</b>
6.1	Literature review . . . . .	89
6.1.1	Verification and validation techniques . . . . .	91
6.2	Verification of co-simulation . . . . .	93
6.3	Validation of co-simulation . . . . .	93
6.3.1	HVAC BESTEST E300 case . . . . .	94
6.3.2	Comparison of mono- and co-simulation . . . . .	98
6.3.3	Comparison of different co-simulation implementations .	110
6.4	Discussion and conclusions . . . . .	121
<b>7</b>	<b>Case studies</b>	<b>123</b>
7.1	Hybrid ventilation with evaporative cooling and run-around heat recovery . . . . .	123
7.1.1	System description . . . . .	123
7.1.2	Model description . . . . .	125
7.1.3	Results . . . . .	127
7.1.4	Conclusions . . . . .	134
7.2	Air solar heating for desiccant regeneration . . . . .	135
7.2.1	System description . . . . .	135
7.2.2	Model description . . . . .	136
7.2.3	Results . . . . .	136
7.2.4	Conclusions . . . . .	139
7.3	Earth-to-air heat exchanger coupled to a double-skin façade . . .	139
7.3.1	System description . . . . .	139
7.3.2	Model description . . . . .	140
7.3.3	Results . . . . .	141
7.3.4	Conclusions . . . . .	143

<b>8 Conclusions</b>	<b>145</b>
8.1 Concluding remarks . . . . .	145
8.2 Directions for future work . . . . .	146
<b>References</b>	<b>149</b>
<b>Curriculum vitae</b>	<b>161</b>



# 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 side, challenging goals are set by the new initiatives and energy policies. For example, European Union has defined ambitious goals for reducing emission of CO<sub>2</sub> for the industrialized countries, which should be achieved by 2020. Also, the U.S. Department of Energy and ASHRAE have defined their vision for 2030 [ASHRAE 2008] in a form of net zero energy buildings. On the other side, new buildings consist of numerous dynamically interacting components that are nonlinear, dynamic, and complex. This requires an integrated approach that treats buildings and the systems that service them, as complete entities, not as separately designed subsystems.

To make energy efficient designs in this complex setting, the concept of integrated building performance simulation (BPS) has been developed. Its main purpose is to (i) analyze the energy consumption and comfort performance, and (ii) understand the relationship between the design parameters, the energy use, and the comfort characteristics of buildings. Experience shows that, if used properly, BPS can indeed result in a significant reduction of emission of greenhouse gasses, and give substantial improvements in fuel consumption and comfort levels.

There is an opinion that simulation has emerged as a third way of doing physics, complementing both theory and practice [Donnelly et al. 2007]. It is, of course, not realistic to think that new physics can be observed only by performing numerical computation as the underlying equations must be based on the known theory. However, simulation is still a powerful tool for:

- analysis of new system designs, retrofits to existing design, proposed changes to operating strategies;
- identification of problems like bottlenecks and design shortfalls, before building or modifying a system;
- comparing different designs and control algorithms (using repeatable boundary conditions);

- learning how different components interact as a system, etc.

Simulation facilitates problem-solving and decision making, in a fraction of time and for a fraction of the cost it would take to perform the analysis on the real system.

Fast developments in computer technology enable the use of complex simulation models even on a single PC unit. The evolution of BPS over the past few decades has encouraged building designers to apply this technology to building design, e.g., [Hui 1998; Clarke et al. 2005; Raslan and Davies 2006]. The capability of BPS in a wider scope of projects is, however, yet to be exploited.

The main barriers in the (commercial) development and wide use of BPS tools have been (i) *the high time-cost of building performance prediction* [Papamichael and Vineeta 2002; Wright et al. 1992; Thomas 2006; Sahlin 2000] and, as a consequence, (ii) *the low market interest*. With regards to the first barrier, Papamichael and Vineeta [2002] states that although potential savings resulting from the use of BPS are high (code requirements exceeded by more than 50% and the initial cost reduced), they did not seem to motivate simulation use on the level of individual buildings. Energy related performance criteria are starting to receive more attention, but are still considered only in a small fraction of buildings designed today. The cost of simulation has usually been considered too high to be routinely justified as a building design cost, and the code compliance testing has been as far as most building designers have gone with respect to considering BPS. In most cases, BPS has only been considered in the later stages of the design process, when critical decisions have already been made.

The second barrier is shown by the generally small number of simulation tool downloads. Such a small market attracts only a few commercial software developers. Thereby, the significant development efforts required for BPS tools, have been left to a fragmented research community. This has resulted in a slower development and inferior capabilities compared to simulation tools used in larger industry sectors, such as in, e.g., the automotive, electronics or aerospace sector.

It is expected that in future the BPS will play a big role in building design process [Mazria and Kershner 2008]. To meet the challenging goals set by the new initiatives and energy policies, intelligent, integrated, easy to use and yet flexible simulation tools are needed, which calls for an integration of the fragmented research within one field.

## 1.1 Problem statement

Due to the fragmented development of BPS tools, and the rapid innovations in building and system technologies, state of the art BPS tools are not equally suited for modeling and simulation of the relevant building aspects. Moreover, the user's requirements often exceed the tool's functionality. 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 from the tool developer to have in-depth knowledge of the pro-

gramming languages used, of the underlying architecture, and of the tool-specific modeling strategies. Furthermore, switching to equation-based tools is not a solution. Although they are better suited for rapid model prototyping than the BPS tools, they typically lack the vast range of state of the art models (e.g., for solar and wind processes) and domain-relevant input/output processing.

Since the value of a tool is nowadays 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 the tools like, e.g., eQuest, DOE 2.1, IES VE, and VA114, compared to the more flexible tools, like, e.g., TRNSYS. The adaptable tools like TRNSYS or Modelica have strength in system modeling and simulation, and are so more likely to drive the innovation towards net zero energy buildings. However, these tools do not have a well developed building model as present in, e.g., ESP-r, EnergyPlus, IES VE or some other similar tools.

To successfully continue with developments of the tools that will drive innovation and help reaching the ambitious goal of reducing the emission of CO<sub>2</sub>, the focus should be on enabling more analyzes of innovative designs rather than investing in "reinventing the wheel". An efficient 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.

Four main strategies that enable sharing of developments and reusing existing component models have been defined in [Hensen et al. 2004]. They are as follows:

1. **Data model interoperation** focuses on the data transfer between modeling environments by means of

- a central database (e.g., COMBINE [Augenbroe 1992]) and
- a common data format (e.g., industry foundation classes - IFC [Lockley et al. 1994], green building XML - gbXML format [<http://www.gbxml.org/>]), which requires data translators for different applications.

The strategy allows modelers to exchange product model information between the tools and thus (i) avoid the burden of creating separate data models for each tool, and (ii) alleviate possible errors introduced by unnecessary and redundant model development. Data model interoperation takes place before run time and does not facilitate integrated simulation requiring integration on the process model level;

2. **Data and process model integration** is based on providing a facility to simulate different sub-domains within the same simulation tool. Some BPS tools already integrate thermal, ventilation, air quality, electrical power and lighting calculations, enabling information exchange throughout a simulation. Examples of such integrated BPS tools are: ESP-r, TRNSYS, EnergyPlus, IDA ICE, IES VE, etc.

The advantage of this strategy is that the user needs to master only one program and does not have to switch between different tools for a holistic approach. The drawback is that the user is restricted to the features available in the particular tool used.

3. **Process model interoperation** is the way of sharing developments in models that describe the thermal, flow, and other physical processes between the simulation tools. The process model interoperation can be achieved by:

- *Exchanging component models* which, with respect to compilation and linkage time, can be done in different ways (e.g., on the level of the source code [Hensen 1991; Aasem 1993], or on the level of pre-compiled static or dynamic libraries [Curtil 2004]). The strategy facilitates code reuse, but still every new development in the building and system domain requires an effort to be included into the existing simulation tool;
- *Using generic languages* like, e.g., NMF [Bring et al. 1999] or Modelica [Tiller 2001]). Providing that translators from the generically expressed models exist, new developments can easily be included in the state of the art BPS tools. However, these translators do not exist for every BPS tool. In addition, the conversion from the existing models available in state of the art BPS tools to a generic format is currently not possible. Thus, to enable reuse of numerous existing models, first their (manual) conversion into a generic model format would be required.

4. **Process model cooperation** focuses on integration of physical process models by linking applications at run-time in order to co-operatively exchange information. It is also known as external coupling [Djunaedy 2005] or co-simulation [Elliott 2002; Wetter and Haves 2008].

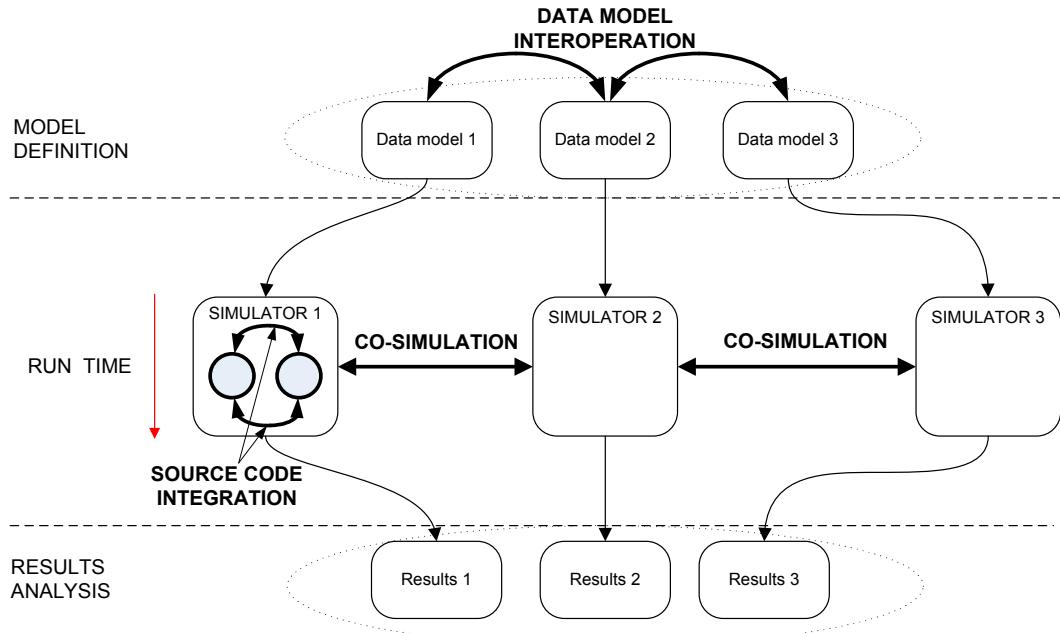
The last mentioned strategy, i.e. using the co-simulation approach, is the main focus of this thesis. This approach complements other tool integration strategies, providing even greater flexibility for modeling and simulation.

The co-simulation strategy in comparison with other strategies is presented in Figure 1.1. The coupled models are independently created and the results are analyzed separately, while the simulators (simulation tools) are coupled at run-time, exchanging data in a predefined manner.

## 1.2 Objectives

The research underlying this thesis is concerned with

- investigation,
- development and implementation,
- numerical experimentation and



**Fig. 1.1** — Co-simulation, integration in run-time in relation with other tool integration strategies.

- usability testing,

of co-simulation for performance prediction of innovative integrated mechanical energy systems in buildings. The main objective and the core issue of the research is how to properly define coupling and obtain accurate simulation results. The proposed way to achieve this objective is to:

- investigate the available strategies and tools for co-simulation;
- investigate what data need to be transferred between building energy simulator(s) and building systems simulator(s);
- investigate the stability of different coupling strategies;
- investigate the required frequency of coupling with regards to the co-simulation accuracy;
- define requirements and recommendations for co-simulation implementation;
- implement a prototype using the existing BPS tools; and
- test the prototype using case studies to show the benefits of co-simulation approach.

## 1.3 Hypothesis

The hypothesis with regards to the objectives is that

- co-simulation can help in performance prediction of innovative integrated mechanical energy systems in buildings.

## 1.4 Research methodology

The two main methods employed in this thesis are theoretical analysis and numerical simulation; the latter done on a software prototype.

Formal analysis of the partitioned numerical schemes is performed to understand how co-simulation with loose coupling influences accuracy and stability of the solution in a more general sense. To further deepen the understanding of co-simulation, the analysis of the solution characteristics is performed on a simple numerical example.

Software prototyping is used to develop, demonstrate and test various implementations of co-simulation. This thesis adopts the definition of software prototype from Djunaedy [2005]; it is *a working model that is developed from existing tools to highlight a specific function with a minimum amount of effort*.

The software prototype development is realized according to the following workflow:

1. Define requirements for the prototype using results from the literature review and the theoretical problem analysis;
2. Implement the prototype using existing state of the art simulation BPS tools to meet requirements from step 1;
3. Verify and validate the prototype;
4. Test the prototype using a case study;
5. Go back to step 1 to refine or expand the requirements based on step 4, until the final prototype is built.

Although the prototypes are implemented using some particular simulation tools, the approach is general and ensures that the obtained results and experiences are tool-independent. In fact, one of the deliverables is the formulation of the generalized requirements for implementation of co-simulation. This guarantees that the approach will have a wide and generic applicability.

Several case studies are performed on the prototypes, to serve for the proof-of-concept, and to demonstrate the applicability of co-simulation and to highlight its benefits.

## 1.5 Thesis outline

This chapter has briefly introduced the idea behind, and the motivation for, the work in the thesis.

Chapters 2 and 3 report on the results from literature review. This is to learn about state of the art in BPS in particular and co-simulation in general.

The theory on system modeling and simulation, with the special focus on HVAC system simulation, is covered in Chapter 2. The state of the art tools in the BPS are discussed in terms of their system modeling capabilities.

The results from a review of available strategies and tools for co-simulation are presented in Chapter 3. Important issues for co-simulation realization are studied, and multiple possibilities are discussed to justify the selection of the particular co-simulation implementation used in this thesis. This chapter puts the work of this thesis into the general development perspective.

Chapter 4 discusses the results from the theoretical analysis of co-simulation. The co-simulation problem is stated for a very general class of problems encountered in BPS, and the stability and accuracy characteristics are studied for a few simple linear one-step numerical schemes.

Chapter 5 describes the coupling strategies implemented in the co-simulation prototypes developed during this research. An exhaustive insight into the prototyped implementation is given.

Chapter 6 starts by giving a wider overview of verification and validation techniques. Following that, a special variation of inter-model comparison technique is defined to be used for co-simulation validation. The results of verification and the two-step validation is discussed in detail.

Chapter 7 presents a few case studies in order to illustrate the potential applications of the co-simulation approach and to demonstrate its benefits.

Chapter 8 summarizes and concludes the work, and presents some direction for future work.



# 2

## State of the art in building systems simulation

A brief historical overview of simulation tools in the BPS field is given in [Clarke 2001]. Starting from the (i) use of simplified methods found in handbooks (calculations based on analytical formulations that embody many simplifying assumptions), via (ii) simplified (still analytical) modeling of dynamics in buildings, the tools had evolved into (iii) tools that use numerical methods and provide partial integration of different aspects within the building, e.g., thermal, visual and acoustic. The next generation of simulation tools that is currently under development tends to be fully integrated, with respect to different building aspects, with new developments concerned with intelligent knowledge-based user interfaces, application quality control and user training. They match with reality much better than earlier tools, but are more complex to use.

In this chapter, the theory of system modeling and simulation, with a special focus on HVAC systems is introduced.

### 2.1 Tools for HVAC system design and analysis

The available domain computer tools [<http://www.eere.energy.gov/>], [Crawley et al. 2005] range (complexity-wise) from spread-sheet tools to more advanced special-purpose simulation tools, and (integration-wise) from tools that handle a single aspect of the building design, to tools that integrate multiple aspects of the building design [<http://www.bwk.tue.nl/bps/hensen/courseware>]. The integration of building and HVAC system models is accomplished in different levels, i.e., the models can be sequentially coupled (many duct/pipe sizing tools, BLAST, DOE-2, etc.) - without system model feedback to the building model; and the models can be fully integrated (ESP-r, EnergyPlus, IDA ICE, TRNSYS, etc.) - allowing the system deficiencies to be taken into account when calculating the building thermal condition. Levels of detail of both building and system models can vary from simple (e.g., bin method; pure conceptual representation for system model, etc.) to complex (numerical model of physical processes). In this chapter, only tools/modules for HVAC systems are discussed.

Tools for HVAC design and analysis can be categorized into several categories with respect to the problems they are meant to deal with. Although these prob-

lems are not mutually exclusive, and some tools can handle several problems, they do tend to be conducted in isolation from each other. The categories are as follows.

**Tools for pipe/duct sizing** The tools in this category are duct/pipe system design tools that consider sizing and flow distribution (AFT Fathom, DOLPHIN, Duct Calculator, DUCTSIZE, Pipe-Flo, PYTHON, etc.)

**Tools for equipment sizing and selection** The tools from this category offer HVAC equipment sizing (Carrier HAP, Trane TRACE 700, EnergyPlus, etc.). Most sizing tools are based on consensus procedures and algorithms established by ASHRAE, but many are proprietary products distributed or sold by equipment manufacturers [<http://wbdg.org/>]. Electronic catalogues that are distributed by equipment manufacturers can be used to locate a suitable component model, for the given design criteria. They can be further linked to the sizing tools, e.g., Carrier HAP tool can be linked to Carrier chiller selection tool by importing performance data for an actual chiller.

**Tools for energy performance analysis** The tools from this category are designed to predict the annual energy consumed by an HVAC system. Based on a system of equations that define thermal performance of buildings and systems, and with given boundary conditions, operation strategy and controls, the tools perform (hourly or sub-hourly) simulations (Carrier HAP, Trane TRACE 700, DOE-2, eQUEST, EnergyPlus, ESP-r, IDA ICE, TRNSYS, HVACSIM+, VA114, SIMBAD, etc.). The tools are used to calculate and analyze the full- and part-load performance, to analyze system operation strategy, to compare different design alternatives, etc.

**Tools for system optimization** The optimization tools are used in conjunction with tools for energy performance analysis. In multiple simulation runs, a set of parameters is optimized according to a given objective function. An example is the generic optimization tool GenOpt [Wetter 2001]. The optimization can also be accomplished by evolutionary programming as it is done in [Fong et al. 2006], or using adaptive neuro-fuzzy algorithms as in [Lu et al. 2005].

**Tools for control analysis and optimization** The level of HVAC system control modeling and simulation in the available tools range from (i) controllers associated with high abstraction of system modeling, through (ii) supervisory control (EnergyPlus) and (iii) implementation of simple local controllers (ESP-r, TRNSYS) to (iv) more advanced controllers, such as fuzzy logic (MATLAB based tools (SIMBAD) and Dymola, or tools coupled to MATLAB (ESP-r [Yahiaoui et al. 2003], TRNSYS [CSTB 2003])). In particular, the latter category of tools are efficient tools for design and more comprehensive testing of controllers in a simulation setting [Jreijiry et al. 2003], as well as for testing and validation of controller design in real time [Riederer 2005].

**Tools for real-time optimization of system performance** As there are benefits of expanding the use of simulation tools to early design stages [Hopfe et al. 2006], there are even more benefits of expanding it towards the operational stage in building life-cycle process. Simulation tools can be used for:

- Commissioning diagnostics (initial commissioning) - to verify the performance of the whole building and its subsystems and components [ANNEX40 2004];
- Monitoring diagnostics (continuous commissioning) and fault detection diagnostics - to detect, analyze, locate and/or predict problems with systems and equipment that are performed during everyday operations and monitoring [Hyvkinen 1996; Haves et al. 2001, 1998; Mathews and Botha 2003];
- Emulating a building and its HVAC systems - to use a simulation tool to simulate the response of a building and its HVAC systems to building energy management system (BEMS) commands. Emulators can also be used for control product development, training of BEMS operators, tuning of control equipment and imitating fault situations to see how the BEMS would cope [Clarke et al. 2002];
- Simulation assisted control - to execute the simulation model (encapsulated within the BEMS) as part of the control task in order to evaluate several possible control scenarios and make a selection in terms of some relevant criteria [Clarke et al. 2002].

The system simulation models, used in this category, need to be able to treat the departures from ideal behavior that occur in real systems and to realistically model controls and HVAC system dynamics, if they are expected to portray system performance accurately. The tools for energy performance analysis can be used as tools for real-time optimization of system performance, but models of building and its systems need to be well calibrated. In general, well calibrated first-principle models can be used, but simpler, more precise empirical (e.g., neural network models) models can be used as well [Hyvkinen 1996].

This thesis' scope includes only energy performance analysis tools.

## 2.2 Modeling approaches

### 2.2.1 Modeling approaches for HVAC components

According to Zeigler [1976], the majority of models in building and system performance simulation can be described as:

- Continuous state models, since the model's variables' ranges can be represented by real numbers, or intervals of real numbers; some models assume a discrete set of values, e.g., some controller models, and are discrete state models;

- Discrete time, since the time is specified to flow in discrete steps. If the model is continuous in state and discrete in time it is then described by a (system of) difference equation(s).
- Deterministic models. Rarely stochastic models are used as well, e.g., in predictive control applications [Clarke et al. 2002].
- Time varying, since the rules of interaction are different at different times.
- Both steady state and dynamic, since some models' response does not depend on time and some does.
- Forward, since they are used to predict the response or output variables based on a known structure and known parameters when subject to input and forcing variables. However, backward (data-driven) models (the input and the output variables are known and measured, and the objective is to determine the mathematical description and to estimate system's parameters) tend to be much simpler but are relevant only for cases when system-specific and accurate models of specific building components are required, e.g., for fault detection and diagnosis [Hyvkinen 1996].

There is a distinction between a primary and a secondary HVAC system. The former is sometimes referred to as a plant, and the latter is referred to as a system. The primary system consumes energy and delivers heating and cooling to a building (through a secondary system) and consist of chillers, boiler, cooling towers, thermal storage (on the plant level), etc. The secondary system typically includes air-handling equipment, air distribution system and liquid distribution system between the primary system and the zone.

In both primary and secondary system there are two types of components: distribution components and heat and mass balance components. The distribution components are: pumps, fans, dampers, valves, ducts and pipes. They affect the building energy consumption with [ASHRAE 2001]:

- electrical energy consumption used to drive pumps and fans, and
- thermal energy transferred to/from the working fluid in all distribution components.

The distribution components models should satisfy energy and mass balance equations. Most of the BPS tools model distribution components in a simplified way [ASHRAE 2001], which eliminates the need to calculate the pressure drop through distribution system at off-design conditions. In general, this approach is sufficiently accurate for studying temperatures in the system, but for detailed analysis, such as (i) fan/pump control loops, and (ii) answering question related to the placement of the return/exhaust fan, type and size of dampers/pipes, flow and pressure balancing between the components is necessary [Haves et al. 1998].

The heat and mass transfer components are usually described by fundamental engineering principles - first principle models, or by empirically obtained equations, i.e., by using regression analysis of design data published by a manufacturer, or by simply specifying look-up tables. The former modeling approach is

usually used for secondary system's component descriptions, while for primary system's components, due to their complexity, the latter approaches are mostly used.

### 2.2.2 Modeling approaches for HVAC control

HVAC controllers can be divided into two categories as follows.

**Local controllers** are the low level controllers that allow the HVAC systems to operate properly and provide adequate services. Local controllers can be further subdivided into two groups [Wang and Ma 2008]:

**Sequencing controllers** define the order and conditions associated with switching equipment ON or OFF. The typical sequencing controllers in HVAC systems are chiller sequencing controller, cooling tower sequencing controller, pump sequencing controller, fan sequencing controller, etc.

**Process controllers** adjust the control variables to meet required set point in spite of disturbances considering the system dynamic characteristics. The typical process controllers used in the HVAC field are P, PI, PID, ON/OFF, step controller, etc.

**Supervisory controllers** are the high level controllers that allows an overall consideration of the system level characteristics and interactions among all components and their associated variables. For example, a supervisory controller sets operation modes and set points for local controllers.

From modeling point of view, controllers are represented by equations that must be satisfied in each simulation time step. The controllers affect the interaction between building and system as well as interactions between components within the system.

In reality the closed loop local process control includes a sensor that samples a real world (measurable) variable. The controller, based on the set point value and measured value, and according to the controller-specific control algorithm, calculates the control signal that feeds the real world actuator. However, in the simulation tool the user can make use of variables that can not be sensed or actuated in reality, as well as apply a control algorithm that would not be able in reality. For example, a modeler can directly actuate the heat flux in his/her model that in reality would be only indirectly actuated by, for example, changing a valve/damper position.

Furthermore, in simulation the concept of "ideal" (local process) control becomes possible due to the accessibility to many variables not known in real world, such as the zone load. The "ideal" local process controller means that the actuated variable will be adjusted to satisfy the set point requirements for the controlled variable, without specifying the explicit control algorithm and by numerically inverting the (forward) simulation components models (from the required output calculate the input needed to satisfy this).

The possibilities to simulate different controllers are limited in the state of the art BPS tools. Nevertheless, there are differences among them. Some offer pre-defined control strategies (system-based simulation tools), some offer flexibility in specifying only supervisory controllers (EnergyPlus) and some even in specifying local controllers (TRNSYS, ESP-r). The domain-independent environments, such as MATLAB and Dymola, are efficient tools for design and testing of controllers in a simulation setting, as already stated.

### 2.2.3 Modeling approaches for HVAC systems

Hensen [1996] defines four categories of HVAC system representation in BPS tools, ranging from purely conceptual towards more explicit, as follows.

**Pure conceptual** system modeling approach represents the case where only room processes are considered, while all other processes in primary and secondary components are idealized, with a possibility to pose a capacity limitation upon them. The peak loads are then used to determine the required HVAC system size. Many BPS state of the art tools can be used to model systems using this approach. Some, e.g., ESP-r, introduce certain complexity by modeling conceptual system - thermal zone interactions through control algorithms. Thus, even though the pure conceptual system model is used, system processes are not completely idealized. Their interaction with the building is more realistically modeled since their characteristics can be included in terms of aspects such as heat injections/extraction point, flux limit values, response time and convective/radiant split. In [NRC 2004], the authors state that this method of system simulation is often misunderstood and under-rated.

**System-based** system modeling approach represents the case with preconfigured common system types, e.g., VAV system, constant-volume variable-temperature system, etc. This modeling approach is implemented in DOE-2, eQUEST, Building Energy Analyzer, BLAST, DesignBuilder, HAP, etc. The user has flexibility to specify capacities, system flow rates, efficiencies and off-design system components' characteristics, but is restricted to the system configurations and control strategies which are pre-defined in the tool.

**Component-based** system modeling approach represents the case where system model is specified by (a) network(s) of interconnected components. This approach is more flexible in terms of possible system configurations and control strategies compared to the previous approach.

**Component-based multi-domain** system modeling approach represents the case where, component representation is further partitioned into multiple interrelated concepts, e.g., fluid flow, heat and electrical power balance concepts, etc. Each balance concept is then solved simultaneously for the whole system. Thus, the overall system of equations is broken into smaller systems of equations. Different solvers, which are well adapted for the equation

types in question, can be used for different problem partitions. It is also possible to easily remove partitions as a function of the problem at hand.

As an addition to the above four categories defined by Hensen [1996], this thesis lists the fifth category: **equation-based** system modeling approach. This modeling approach represents a case where system model is represented by a basic modeling unit, which is physically “smaller” than a component and is in the form of an equation or a low-level physical process model. It has evolved as the fifth category as a need to improve the BPS tools, that had been based on technology available in the early seventies [Sahlin et al. 2003]. The equation-based simulation tools [Chow 1995; Wetter and Haugstetter 2006]:

- are input-output free (all models are declarative in nature) as opposed to the traditional procedural,
- are modular (supported by object-oriented programming languages),
- are hierarchical (enable incremental modeling, i.e., models can consist of sub-models in multiple levels), which helps in managing the complexity of large systems,
- are universal (model definition in a generic form, e.g., using NMF or Modelica),
- provide separation of modeling the physics from numerical solution algorithms,
- provide faster developments of simulation models, etc.

Examples of equation-based tools are:

**SPARK** (Simulation Problem Analysis and Research Kernel), formerly EKS/US and SPANK, is developed by Lawrence Berkeley laboratory in USA [LBNL 2003]. The primary goal of the EKS/US became improvement of the modeling and solution processes which resulted in SPARK. It is an object-oriented simulation environment, of which fundamental object is an equation.

**EKS** (Energy Kernel System) was researched in the UK [Clarke et al. 1992]. The objective of the EKS/UK was to place tool development on a task-sharing basis in order to ensure the integrity and extensibility of future systems. The primary goal of the EKS/UK became improvement of the tool development process which resulted in development of primitive parts within ESP-r simulation environment [Chow 1995].

**Neutral model format** (NMF) was designed to bring the power of DAE-based modeling to the building simulation community and yet be compatible with major BPS tools such as TRNSYS, IDA and SPARK. The basic objective of NMF is to provide a common format of model expression for a number of existing and emerging simulation tools, e.g., TRNSYS, HVACSIM+, IDA,

SPARK, etc. From a technology point of view, NMF effort has been a success but the language has never caught the sustained interest of independent BPS developers [Sahlin et al. 2004].

**IDA** is developed by Swedish Institute of Applied Mathematics. It is one of a few efforts that have been pursued beyond the stage of prototyping [Sahlin et al. 2004]. The NMF initiative continues to live with IDA, since most of the IDA models are written in NMF, besides a few written in Modelica [Sahlin et al. 2003].

**Modelica** [Tiller 2001] An ambitious modeling language, that has shown potential to bring order to the fragmented world of differential algebraic equation (DAE)-based simulation. It draws on the collective experience of a large number of first generation languages and since the first tool, Dymola, appeared in 1999, several large industries such as Toyota, Ford, United Technologies, Caterpillar, ABB, Alstom, TetraPak, etc. have adopted it [Sahlin et al. 2004]. Efforts to develop building and HVAC system simulation models resulted in various Modelica libraries, such as ATPlus [Felgner et al. 2002], UTRC Modelica library [Wetter 2006] and Building Informatics Environment [<https://gaia.lbl.gov/virBui>], which be released soon.

**SimScape** [<http://www.mathworks.com/products/simscape>] a new development by MathWorks, extends Simulink with tools for modeling and simulating multi-domain physical systems, such as those with mechanical, hydraulic, and electrical components. Simscape can be used for a variety of automotive, aerospace, defense, and industrial-equipment applications. Together with other MatLab toolboxes, Simscape allows modeling of complex interactions in multi-domain physical systems.

Based on object oriented programming language, the above projects were aiming to introduce “modern concepts from computer science and software engineering in the BPS field to make available to developers basic software modules and supporting framework that could be used to construct new BPS software” [Clarke and MacRandal 1993]. But, as Sahlin et al. [2003] concludes, nothing much has happened in recent years to change the direction of fundamental reasoning. The authors also state several factors that contributed to such a situation, as follows.

- Some exploratory projects did not deliver as expected.
- Leading research groups have reverted back to existing solutions and “organic” evolution.
- Multi-domain simulation is being attempted by coupling of existing domain specific simulators (co-simulation).
- Driven by product model research, attention has shifted from new tool development to improved integration of existing modeling and simulation tools into the design process.

Sahlin [2000] states that the primary cause of the lack of success is “unwillingness by BPS developers to learn other engineering fields”. It seems that even besides new tool development, attention has not been shifted from existing tools. Due to the difficulty in obtaining funding for work other than incremental improvements of BPS tools [Spitler 2008], many researchers/ research teams continued to improve integration of “traditional” simulation tool into design process.

The major motive for the adoption of object-oriented software engineering approaches has been its support for modularity in modeling. However, a model for the simulation of a complex system, such as a building, in object-oriented languages is not simple [Wright et al. 1992]. One of the questions is to what objects should correspond. Should they correspond to real-world entities or to equations associated with those entities. Maybe, the lack of the agreement upon the above issue has resulted on a limited presence of the object-oriented programming in the domain of BPS.

#### 2.2.4 Solution techniques for HVAC system simulation models

The differences in solution techniques, employed by different simulation tools, are based on the distinction in the way the integrator is called [Hillestad and Hertzberg 1988], as follows.

**Simultaneous modular solution** where the various components are integrated simultaneously by a common integrator.

In general, tools that employ this solution technique use model equations that are based on first principles [Hillestad and Hertzberg 1988]. Each component is described with time-averaged discretised heat and/or mass conservation statements which are combined to form a system matrix, and which are solved simultaneously in each simulation time step using either an implicit, explicit or mixed numerical scheme.

**Independent modular solution** where each module is provided with individual integrator routines.

In general, tools that employ this solution technique use model equations that can be based on first principles but can also be empirical input/output correlations [Hillestad and Hertzberg 1988]. The component’s modules encapsulate all information relevant for the component’s simulation model setting and execution. Each component is executed sequentially and the overall system solver iterates until a convergent solution has been found.

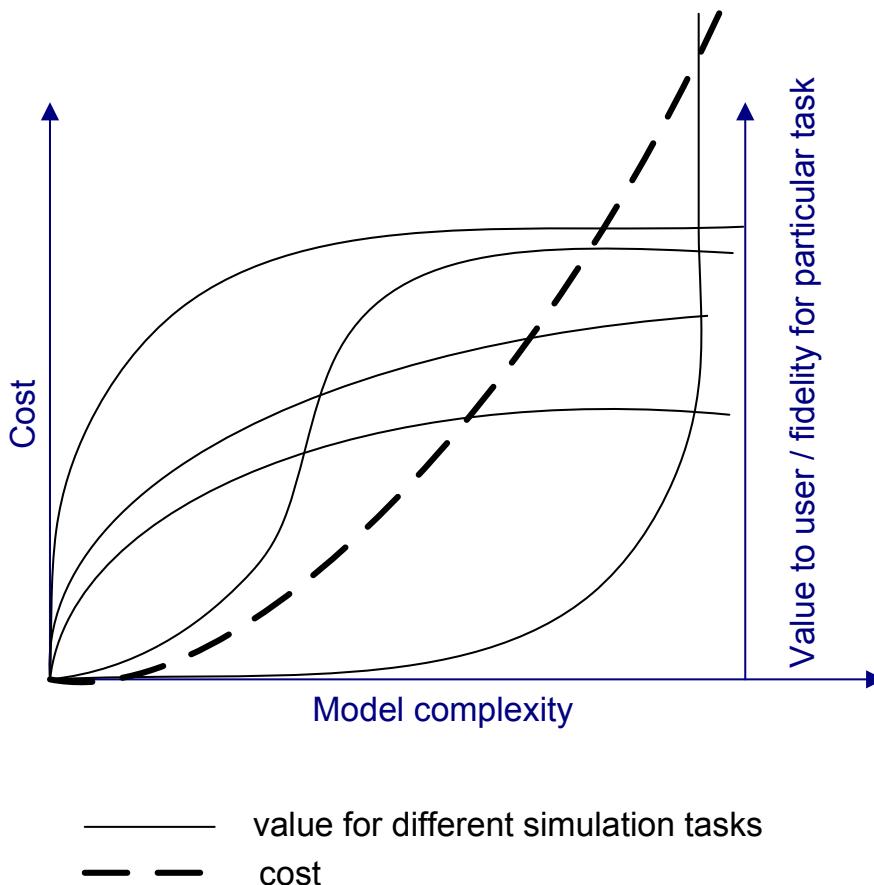
**Equation-based solution using formula manipulation** which has emerged in recent years with developments of equation-based tools. Models composed with these tools cannot be executed directly. To be executed, the model needs to be parsed to a programming language that can be compiled. Thus, tools employ different techniques to reduce the dimensionality of the linear and non-linear systems defined in the model. For example, in SPARK [Sowell et al. 2004], mathematical graph algorithms are used for problem decom-

position and reduction, greatly reducing solution time for wide classes of problems [Sowell et al. 1999].

## 2.3 Selection of HVAC modeling approach

Different HVAC system modeling approaches demand different levels of ease of use and skills required for effective use, different modeling resolution and detail, and different levels of user customization capability.

Going from the conceptual to the more explicit system representations the required knowledge about the system is increased resulting in the increased number of the parameters for the system specification, which often are difficult to obtain as they are not supplied by manufacturers. The computational requirements becomes intense and the analysis of the results more complicated.



**Fig. 2.1** — Cost and value to the user vs. complexity.

Most design analyses (to study trends and to compare systems) do not require detailed system modeling and simulation and the energy consumption can be estimated by using simpler methods. The conceptual system representation shows its advantages (lower required user expertise, lesser input data, less intense computations, easier results analysis, etc.) when only loads determinations are con-

sidered, and/or when energy reduction requirements are investigated. However, for comparing HVAC system alternatives and evaluating different control strategies [Miller 1980; Haves et al. 1998], the detailed HVAC system is required. In system based modeling approach, the speed of system alternatives evaluation is much better than in the component based modeling approach, but the investigation of innovative technologies is restricted.

Matching the applicabilities of the system modeling approaches to the design questions at hand, the user can benefit from both ease of the former categories and flexibility of the latter ones. However, building a right model for a simulation task at hand is still more an art than an engineering discipline.

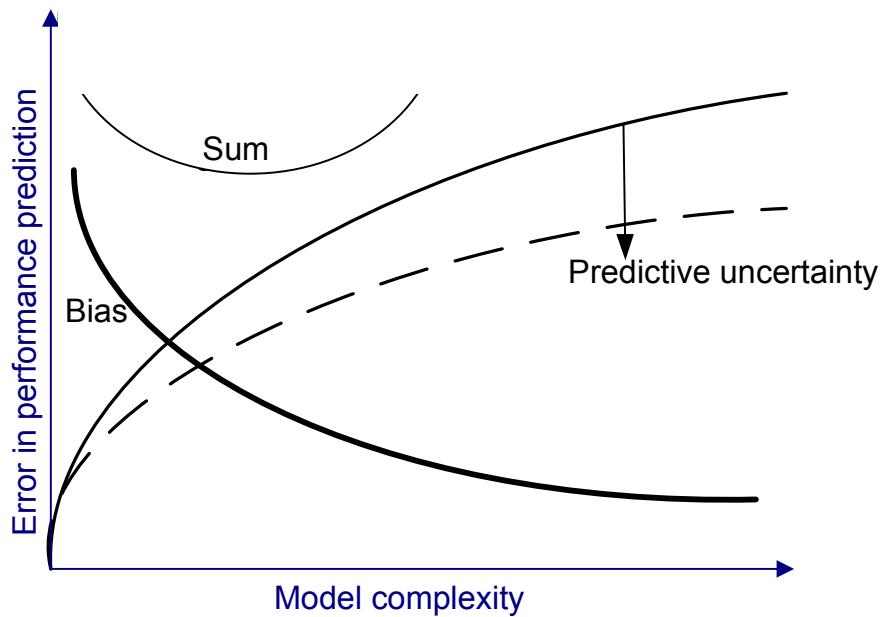
Building the right system model for a specific purpose is to require that the modeling validity and data validity of the system model match as far as possible the required validity [DMSO 2000]. The required validity is assessed only against those aspects of the real world that are of relevance for successful accomplishment of simulation objectives, represented by performance parameters.

Model complexity can be expressed in terms of scope (defined by a number of components in the model) and resolution (defined by a number of states per component in the model) of the model and interactions among components in the model. Abstraction is a general process and includes various simplification approaches that affect one or many elements from the product that defines complexity. Increase in a model complexity increases the model's cost. Thus, the model should be of the lowest complexity that preserves its validity for the intended simulation objectives. The required lowest model complexity depends on the simulation objective. Also, increasing the model complexity, for different simulation objectives, has different implications on the value of the model to the user, as represented in Figure 2.1. For different simulation objectives the model's cost exceeds the model's value to the user at different model complexities. For some objectives the model's cost will exceed its value even when the modeling complexity is low, and for some, the simulation objective can justify the use of complex models. Moreover, the rate of change in the model value can be different for different simulation objectives at different complexities. On the one hand, a simple model can have a high value at low modeling complexity for some simulation objectives. This value might not be increased by increasing the complexity. On the other hand, a model has a value only above a certain modeling complexity for some other simulation objectives.

Definition of the minimum required modeling complexity can be accomplished by using the checklist rationale from [Pace 2000]. The stakeholder defines what are the simulation objectives and thus what are the relevant performance indicators. Based on this information, the checklist framework can be used to identify the entities and variables to be used in the simulation, and thus estimate the initial modeling complexity. The initial modeling complexity should be the lowest possible complexity that satisfies the simulation objectives in terms of performance indicators. The quantification of validity of the initial/minimum required modeling complexity is achieved by specifying a range for error tolerance, as the model deviation of the real world.

The error in a verified model is a sum of: (i) modeling abstraction error, (ii) input data error, and (iii) numerical errors. Here, only former two are discussed. The first is due to the modeling abstractions, i.e., using an incomplete model of physical system, and the second is due to uncertainties in parameters themselves. Sometimes the distinction between the two is not clear.

The parameter uncertainty can be quantified and therefore the corresponding uncertainty of the model output as well. This uncertainty in output is known as predictive uncertainty.



**Fig. 2.2 —** Model uncertainty.

The modeling uncertainty is not easily quantifiable and therefore its influence can be considered as modeling bias. As shown in Figure 2.2, with increasing the modeling complexity, the predictive uncertainty rises, as there are more parameters with their uncertainties to consider. On the other hand, the models approaching reality and the bias decreases. The curve that defines predictive uncertainty depends on how much of system knowledge is available. If the modeled system is well known the input parameters are less uncertain and the rate of increasing predictive uncertainty with model complexity is lower. The modeling complexity for which the model error has its minimum will closely be related to the available system knowledge.

There is a certain modeling complexity after which the predictive uncertainty will be higher than the modeling bias. There is no sense going beyond this complexity, as the overall error in the model uncertainty will not be decreased. Hence, whether the required validity will be met by the model depends not only on the system modeling complexity, but also on the available system knowledge.

# 3

## Co-simulation - Principles and strategies

**I**N this chapter the available strategies and tools for co-simulation are reviewed. The important issues for co-simulation realization are established, and multiple possibilities are discussed to justify the selection of the co-simulation implementation discussed in this thesis.

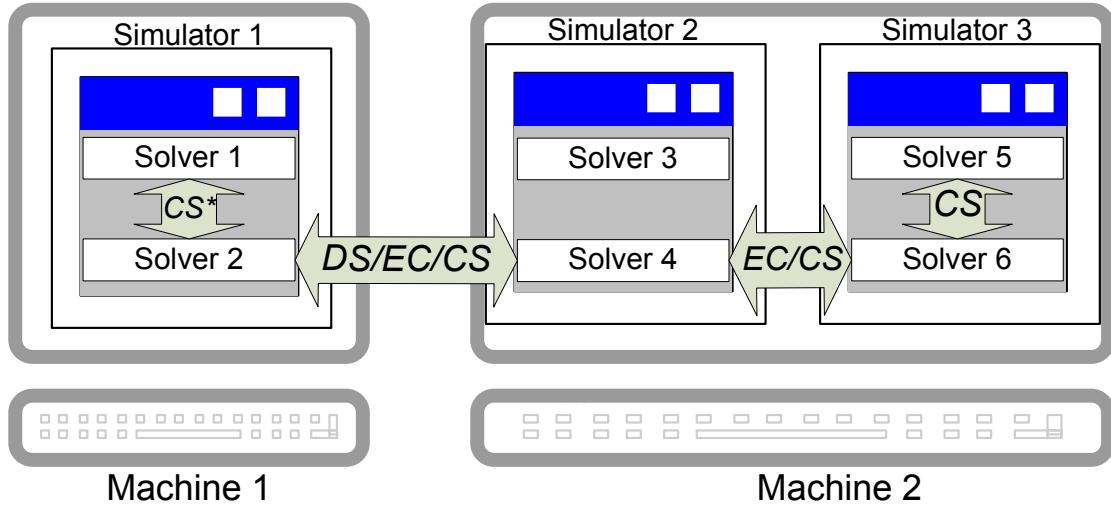
### 3.1 Introduction

The traditional way of performing integrated BPS is to model and simulate different domains/subsystems in a monolithic stand-alone simulation tool that runs on a single computer. An alternative to the traditional way is to model different domains/subsystems in different simulation tools, and then integrate these tools into a single simulation. In the literature, this alternative approach has been referred to using several terms: data and process model co-operation [Hensen et al. 2004], external coupling (EC) [Djunaedy 2005], co-simulation (CS) [Elliott 2002; Wetter and Haves 2008] and distributed simulation (DS) [Fujimoto 2005].

Data and process model co-operation and external coupling have the same meaning. The general differences between the rest of the terms are illustrated in Figure 3.1. Co-simulation in its broader meaning represents a particular case of simulation scenario where two solvers interact [Monty 2002], while in its more narrow meaning represents the same case as external coupling, where two simulators (executables) interact [Gu and Asada 2001]. Distributed simulation refers to the technology concerned with integrating various simulators over the network. Thus, external coupling and co-simulation in its narrower meaning, are less specific than distributed simulation, as the coupled simulators do not necessarily have to be distributed over the network (the models are distributed in aspects).

However, from the mathematical point of view and with regards to the general questions, e.g., consistency, stability, and accuracy of the overall simulation, there are no differences between the technologies and many of the issues that are researched and discussed in all three approaches are common.

In this thesis, the term co-simulation is adopted to address the case of simulation scenario where at least two simulators solve coupled differential-algebraic systems of equations and exchange data during run-time that couples these equations.



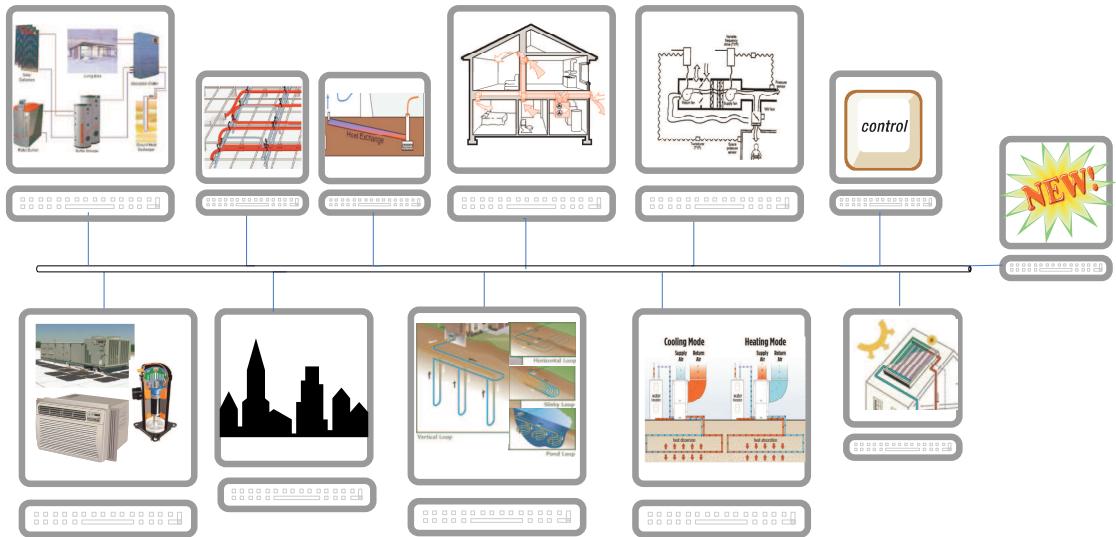
**Fig. 3.1** — Illustration of differences between distributed simulation (DS) and external coupling (EC) to achieve co-simulation (CS). CS\* refers to co-simulation in its wider meaning.

In general, compared to the traditional, monolithic approach, co-simulation has several advantages [Boer 2005; Ganse 2005; Fujimoto 2005; Hillestad and Hertzberg 1986]:

- Reusability of state of the art domain simulation tools by taking advantages of existing models;
- Combination of heterogeneous technologies (using discretization technique and solution algorithm that are best suited for a modeled subsystem) and tools (using modeling environment of specialized tools);
- Fast model prototyping of new technologies;
- Collaborative model design and development process, i.e., models developed by different design teams or subcontractors can be executed concurrently;
- Immediate availability of new model developments;
- Information hiding, i.e., use of proprietary tools, etc.

In particular, BPS can benefit from the co-simulation approach as:

- At the moment there is no a single tool that can be used to solve all simulation analysis problems encountered by designers;
- Each tool can benefit from future simulation models developments of emerging technologies, e.g., models for micro heat and power generators, fuel cell etc., as soon as they become available;



**Fig. 3.2** — Illustration of integrated BPS using co-simulation.

- Fast model prototyping of new technologies, which is difficult in the state of the art domain tools, could be done using an equation-based simulation tool, which used in co-operation with the state of the art tools would assure the integrated approach to building and systems simulation.
- Multi-scale modeling and simulation can be used by combining various building and system models, developed by different parties and simulate various scenarios on the scale of a town, or even a region (Figure 3.2).

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

On the one side, in the field of BPS, lots of effort has been put in (internal) integration of multiple interacting aspects of a building. Besides few exceptions, this resulted into integrated BPS tools (e.g., ESP-r, EnergyPlus, IES VE, IDA ICE, TRNSYS, etc.). Moreover, some of the integrated BPS tools integrated process models available in other tools, i.e., by converting the models into their own subroutines. Examples of such integrations are: the couplings between: ESP-r and TRNSYS [Hensen 1991; Aasem 1993], integration of multi-zone air flow network simulator (e.g., COMIS) with building energy simulator EnergyPlus [Huang et al. 1999] and TRNSYS [McDowell et al. 2003; Weber et al. 2002], EnergyPlus and MIT-CFD [Zhai 2003], EnergyPlus and Delight [Carroll and Hitchcock 2005], EnergyPlus and SPARK [Curtill 2004], etc.

On the other side, only a limited amount of work has been done in process model co-operation (co-simulation). Illustrious examples are: the integration of high-resolution light simulator (Radiance) with building energy simulator (ESP-r)

[Janak 1999] and the integration of computational fluid dynamics simulator (FLUENT) with building energy simulator (ESP-r) [Djunaedy et al. 2003]. In the domain of HVAC simulators examples include integration of TRNSYS with several other programs, e.g., MATLAB [CSTB 2003] and EES [Keilholz 2002]. However until now, there exists neither general standardized framework for integration of BPS simulators nor guidelines for implementation of co-simulation with regards to its stability and accuracy.

## 3.2 Terminology and other issues for co-stimulation implementation

In this section, relevant issues for co-simulation implementation are listed and the specific co-simulation terminology is clarified. Various co-simulation realizations have different implications with regards to stability, convergence and accuracy, and thus it is important to make right choices.

### 3.2.1 Interface classification

Co-simulation using available tools can require a significant investment, and thus each tool, as a candidate for co-simulation should provide an application programming interface (API) on its own side. In order to exchange relevant data between coupled tools, several interfaces have been developed and implemented in the recent years. Vaculín et al. [2004] summarizes some classification categories. This classification is summarized below, since it helps placing the co-simulation approach, discussed in this thesis, into a larger context.

**Workflow can be**

- *Uni-directional* or *sequential* workflow which means that once one simulation is finished its output will be redirected to the input of another simulation. In [Zhai 2003] this approach is also called *static* coupling. The sequentially coupled approach is only sufficient in open-loop systems. A generic framework for this workflow is provided for example by the Kepler software [<http://kepler-project.org>].
- *Bi-directional* or *run-time* workflow requires run-time exchange of coupled data between the simulations. Due to inherent feed back between subsystems, in all closed-loop configurations, there will be dynamic interactions between the components and thus run-time coupling is required. Also, bi-directional workflow is required in open-loop systems, when the coupling data is changed by the component from the downstream sub-system.

**Numerical integration can be done by**

- *Common numerical integrator* which addresses the cases where the integration is done by a single integrator (this, by definition, is not considered co-simulation), and

- *Distributed numerical integrator*, which addresses the cases where each coupled tool uses its own integrator.

**Model abstraction adaptation** addresses (i) model reduction or (ii) interpolation between the coupled models when the models of different complexities are co-simulated. The subject is not discussed in this thesis.

**Programming technique** which is often confused with numerical integration approach. It does not depend on the numerical integration approach and can be realized as a *single* or *multi* process solution. It is possible to have distributed integration as a single process, or common integration as multi process.

**Platform dependence** addresses the issue whether coupled simulators run on the same or on different platforms. Hence, there can be single platform and multi platform.

The implementation of co-simulation in this thesis considers coupling on a single platform, executed as multiple process simulation, using distributed integration and bi-directional data flow.

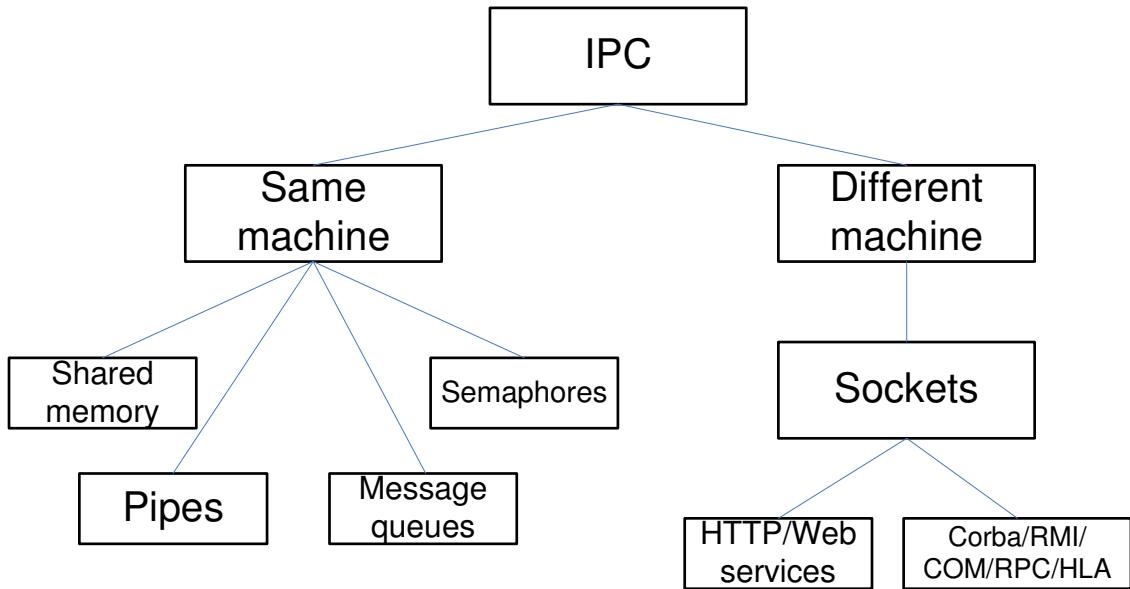
### Data transfer

API can also be distinguished by the way data is transferred between the tools. The communication and data sharing between processes (applications) are called interprocess communications (IPC). The Figure 3.3 shows the IPC taxonomy. An overview of most commonly used IPC protocols is given in [Yahiaoui et al. 2004] and [Trčka-Radošević and Hensen 2006], and a comparison of architectures for distributed computing: High Level Architecture (HLA), The Common Object Request Broker Architecture (CORBA) and Remote Method Invocation (RMI) is given in [Buss and Jackson 1998]. RMI is language-specific and suitable for use with newly developed applications. Both, CORBA and HLA are concerned with legacy applications, possibly developed in different languages.

CORBA has been used in many projects in industry. For example, NASA Glenn Research Center (GRC) program, within NASA's High Performance Computing and Communication (HPCC), has been developing large scale, detailed simulation program for design analysis of aircraft engines, called Numerical Propulsion System Simulation (NPSS) [Follen et al. 2001; Sang et al. 2002]. However, CORBA as well as the majority of the IPC mechanisms represented in Figure 3.3 (apart from HLA) is made to facilitate communication between two applications in general. They do not define the semantics required for simulations, where additional management of time and data exchange is required.

### Architecture, frameworks and toolkits for co-simulation

On the other hand, a great effort is put in developing higher-level architectures for distributed simulation in particular by US Department of Defense (DoD). These efforts resulted in a standardized protocol Aggregate Level Simulation Protocol (ALSP), a domain specific Distributed Interactive Simulation (DIS) and finally



**Fig. 3.3 —** IPC Taxonomy (partially taken from [McGregor 2005]).

merging these results in HLA that is today considered state of the art in distributed simulation and was in 2000 made IEEE standard for distributed simulation [IEEE 2000]. Today, HLA is still mostly used only within defense community for military training simulators [Li et al. 2005; Wilcox et al. 2000] and in multi-player gaming [Wilcox et al. 2000; Pollini and Innocenti 2000]. However, some initial actions have been made in order to adopt the standard in industry by DaimlerChrysler [Boer 2005], and Boeing [Wilcox et al. 2000]. Both groups extensively cooperate with military, but they extended the use of HLA to their other aspects of businesses, i.e., to model supply chain management [Wang et al. 2004; Duggan 2002; Taylor et al. 2002]. Moreover, HLA was used in [Lee 2004] to model satellite cluster management and for event based traffic simulator [Strassburger 2001, 2004].

In parallel to the initial attempts to use the HLA in the civilian domain applications, there is a delicate discussion whether or not the approach is suitable for needs outside defense community. Taylor et al. [2002] argues that the HLA complexity that suits defense community requirements, might be in excess of relatively simple data exchange requirements in major industries, and questions the appropriateness of HLA implementations away from its original domain. In addition he raises the issue of data exchange approach standard lacking, as without such a standard there cannot be universal interoperation between commercial off-the-shelf (COTS) tools. Wilcox et al. [2000] wonders whether many industrial communities either do not share, or have not yet explored the requirements to combine distributed models. Boer [2005] states that the size of projects in industry is smaller relative to the size of projects in military and that most of projects in industry will not benefit from HLA, considering costs. He argues that industry requires less complex solution. This reflects on the statement made in [Taylor et al. 2002], where it is argued that HLA in industry is only a solution that is

looking for a “fantasy” problem and not finding one yet.

ADvantage framework is developed by Applied Dynamics International (ADI) and used by major aerospace and defense industries including Boeing, Pratt and Whitney Rocketdyne, Raytheon, NASA, U.S. Navy, U.S. Air Force, U.S. Army, General Dynamics, BAE Systems, Rolls-Royce, Honeywell, Gulfstream, and many more to support system integration activities. ADvantage enables various simulation model types and embedded programming languages to be brought together to build a non-real-time co-simulation. As development progresses, ADvantage enables the non-real-time co-simulation to be quickly converted to real-time virtual prototypes, and later virtual prototypes to be swapped out with production prototypes. ADI’s ADvantage is designed to co-simulate prepared models. Each model needs to adhere to a handful of basic rules to participate in the simulation [Ellinger 2007].

In the field of Earth’s system (climate/weather/ocean) modeling and simulation, a few software for co-simulation are used, such as:

- *MCT* (Model Coupling Toolkit) [[www.mcs.anl.gov/mct](http://www.mcs.anl.gov/mct)] that is very flexible but non applicable to Windows based simulators,
- *ESMF* (Earth System Modeling Framework) [[www.esmf.ucar.edu](http://www.esmf.ucar.edu)] that is currently in development,
- a more general *CCA* (Common Component Architecture) [[www.cca-forum.org](http://www.cca-forum.org)] that uses the *Babel* [[www.llnl.gov/CASC/components/babel.html](http://www.llnl.gov/CASC/components/babel.html)] language interoperability tool and is used for multi-platform co-simulation,
- *PRISM* (Program for Integrated Earth System Modeling) [[www.prism.enes.org](http://www.prism.enes.org)],
- *OpenMI* (Open Modeling Interface) [[www.openmi.org](http://www.openmi.org)] used for co-simulation of single-processor Windows simulators in the field of hydrology, etc.

Most of these frameworks are limited to the use in the domain of earth systems and climate prediction.

At the time of writing this thesis, another framework for co-simulation is being developed by Lawrence Berkeley National Laboratory (LBNL). It is called Building Controls Virtual Test Bed BCVTB - [Wetter and Haves 2008]. The BCVTB is based on a middleware that is built using Ptolemy II, a modular environment for design and analysis of heterogeneous real-time systems. The LBNL additions to Ptolemy II allow users to couple to Ptolemy II a prototype version of EnergyPlus, MATLAB / Simulink for data exchange during run-time. The coupling data exchange, between simulators that are executed in parallel, is done using loose coupling strategy with naive modification for parallel processing (for explanation see §3.2.8) and fixed synchronization steps.

The implementation of either CORBA, HLA, or ADvantage for distributed building systems simulation mainly raises difficulties when interfacing the state of the art tools from BPS field. The BPS tools are mainly written in Fortran for which no object interface language mappings (neither for CORBA nor for HLA) have been defined nor they are prepared to be used with ADI tools. Much time and extra materials are necessary to overcome this difficulty as discussed in [Yahiaoui et al. 2004]. A more general framework, used in the field of climate/weather/ocean modeling and simulation - CCA, requires its users to make substantial structural modifications to their legacy codes and one must write wrapper code to create components from each of the system's components. The MCT deals with couplings between message passing interface based distributed memory parallel applications, which is not the case with the tools in the BPS domain. By implementing a less complex IPC and formulating the time management mechanism, it is believed that distributed simulation in the domain of BPS can expand the present technology limitations, enabling more flexible use of available state of the art tools. For example, the use of shared memory and named-pipes has advantages for co-simulation developers as it gives the ability to expose internal data to 'the outside world' without the troubles of writing complicated API's. In future, besides BCVTB, frameworks, such as ADvantage, MCT could be potentially considered to facilitate co-simulation of building systems. Still, the challenge to find a way to open the processes for this unanticipated purpose remains.

### 3.2.2 Simulator's roles

In the co-simulation prototypes discussed in this thesis, the sending and the receiving sequence differs between coupled simulators. Thus, different terms for simulators with the different sending and receiving sequences are adopted. The term *base* simulator is used to address a building energy simulator, which also functions as the overall simulation controller<sup>1</sup> and starts the communication with sending the coupling data to the interface, while the term *external* simulator is used to address a simulator that starts the communication by reading those data from the interface.

### 3.2.3 System partitioning

A system can be partitioned by [Felippa et al. 1999; Matthies and Steindorf 2002; Vaculín et al. 2004]:

- *function*, i.e., heat exchanger/heat storage, or HVAC system/building envelope,
- *physics*, i.e., fluid/structure, or heat flow/mass flow, and
- *computing requirements* to accommodate parallelism.

---

<sup>1</sup>A separate overall simulation controller may be considered in the future.

This thesis is concerned only with systems partitioned by function.

Also, the system partitioning can be [Felippa et al. 1999]:

- *algebraic*, or
- *differential*.

In algebraic partitioning, the complete differential system of equations is discretized first, and then partitioned. Therefore, the partitioning starts from a system of algebraic equations. Differential partitioning starts by partitioning of the system of differential equations. The discretization is then applied to the new (partitioned) system of differential equations. From the implementation point of view, the algebraic partitioning requires only prediction of coupling data, while the differential requires prediction of all  $p - 1$  derivatives of the coupling data, where  $p$  is the order of the differential equation being integrated. However, the approaches do not differ if the system is described by first-order differential equations, and if the predicted coupling data take the value available in the previous simulation time step.

Depending on which data are delayed in time, Park [1980] defines two partitioning strategies. He considers only algebraic partitioning, and examines a matrix that constitutes of coefficients corresponding to externally coupled terms and that multiplies the state vector. Based on diagonal values of the matrix he identifies:

- *implicit-implicit* partitioning, when all diagonal entries are zero, and
- *implicit-explicit* partitioning, when the matrix contains nonzero diagonal entries.

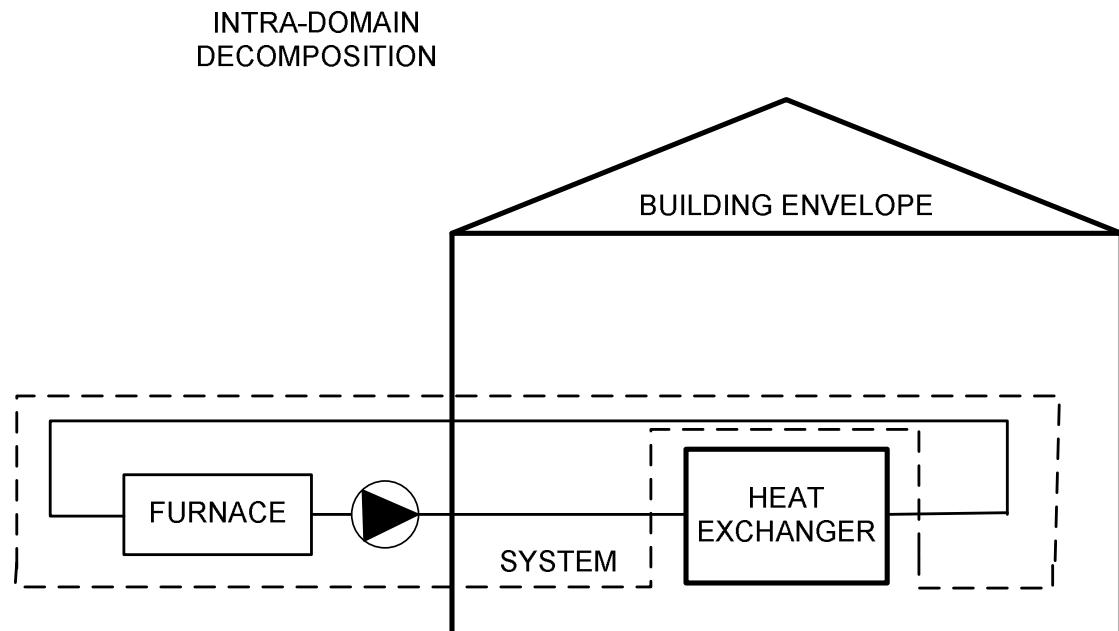
In other words, if the coupling data depends only on the state variable(s) of the coupled subsystem, the partitioning will be of the implicit-implicit type. However, if the coupling data depends on the state variable(s) of the both subsystems, the partitioning will be of the implicit-explicit type.

### 3.2.4 System decomposition strategies

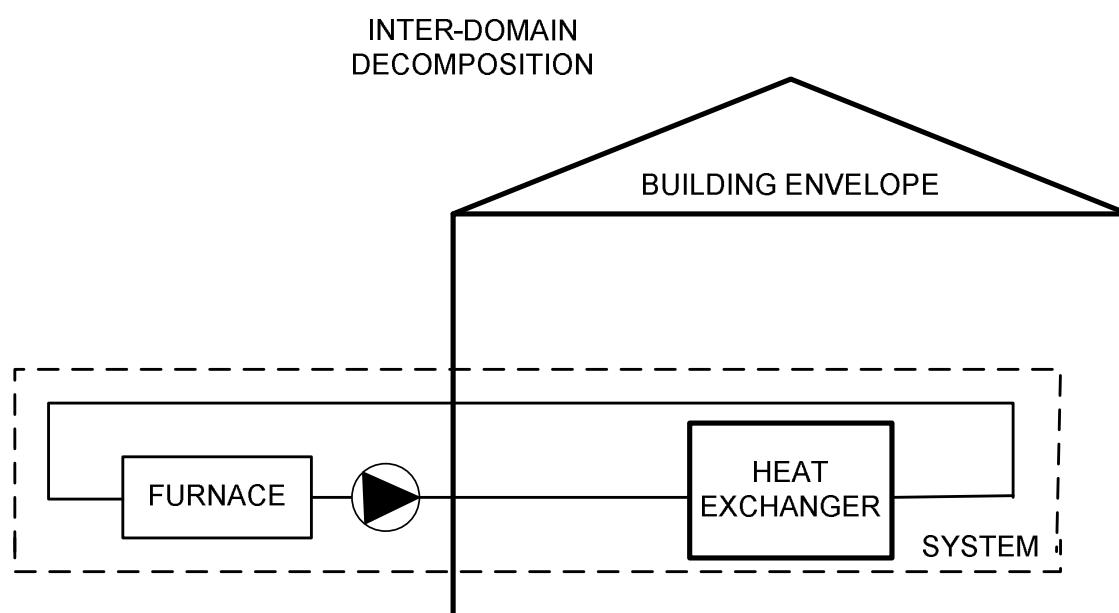
Two different system-decomposition strategies are possible. The first strategy is called *intra-domain system decomposition*. As its name reflects the system is decomposed within one domain, e.g., within HVAC domain only (Figure 3.4). This thesis discusses the decomposition only within the HVAC and not within the building domain.

The second strategy, *inter-domain decomposition*, allows system decomposition between different domains, i.e., between the building and the HVAC system domain (Figure 3.5).

A system can be decomposed so that the decomposed subsystems are either connected in series or in parallel. Since this thesis considers tools in which the flow is not pressure dependent and the flow distribution is predefined, there is no



**Fig. 3.4** — Schematic of intra-domain system decomposition.



**Fig. 3.5** — Schematic of inter-domain system decomposition.

particular difference in the implementation of series and parallel decompositions and they will not be regarded separately.

However, modeling of pressure driven flow in systems with parallel system decomposition poses coupling constraints, e.g., sum of mass flow in the parallel branches needs to be equal to the mass flow in the main branch keeping the pressure drop in the parallel branches equal. Gu and Asada [2004] developed an algorithm that deals with such coupled subsystems, connected in parallel, by using a sliding control. The algorithm minimizes the amount of model information necessary to disclose for co-simulation purposes. However, it requires some additional coupling data in the form of subsystem-dependent function that describes derivative of the output variable over time at the current state and time.

### 3.2.5 Coupling data

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

For example, in intra-HVAC-domain decomposition simulators “communicate” through working fluids (water and moist air) and through control signals. This implies that by choosing a minimum set of data that determines the thermodynamic state of the working fluid ( $p + f = c + 2$  where:  $p$  is the number of phases,  $f$  is the variance or number of degrees of freedom in the system, and  $c$  is the number of fluid components), data that quantifies the flow, and data that represents control signal(s), the complete information can be communicated between the coupled simulators. The quality of this minimum set of data has been discussed many times before. Firstly, as in many HVAC component modeling techniques, pressure drop is neglected, and mass flow is considered to be pressure independent [Hensen 1991; Bring et al. 1999], the minimum number of data decreases by one<sup>2</sup>. For two degrees of freedom and pressure-independent flow, Sahlin and Sowell [1989] defines mass flow rate and enthalpy as the data involved in interactions between HVAC simulators. If the number of degrees of freedom increases, the additional data holding the information about the humidity (moist air) has to be included. For that case, Sowell and Moshier [1995] on the bases of directness of their applicability in the conservation equations and their familiarity to practitioners defines the set of data including: mass flow rate of dry air, temperature and humidity ratio preferred among other options. Since the coupling data depends only on the state of the coupled subsystem, the intra-HVAC-domain decomposition results into implicit-implicit partitioning.

In inter-domain decomposition, the basic set of coupling data are the heat rate (convective, radiant and latent) in one direction and temperatures (air and mean radiant) and the humidity ratio in another. The sets can be extended to include control signals if a sensor and an actuator are distributed among coupled

---

<sup>2</sup>It is important to note that this thesis is restricted to the coupling of pressure-independent flows.

simulators. The coupling data (heat rate) depends on the states (temperatures) of both coupled subsystems, and thus the inter-domain decomposition results into implicit-explicit partitioning. This has a direct implication on co-simulation stability and accuracy, as it will be discussed in the following chapter.

Component models in different tools require different sets and units of coupling data. Thus, a conversion mechanism, such as an *auto-link* [Sahlin et al. 1995; Sowell and Moshier 1995; Sowell et al. 2001], is required. However, it is assumed that the unique set of coupling data expressed in SI units is communicated, while some additional but trivial transformation and mapping might be used, inside the interface, if necessary.

### 3.2.6 Time management in co-simulation

Maybe the most important issue when discussing co-simulation is time synchronization. Even though coupled simulations follow their own time management scheme, they are seamlessly 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 simulator are synchronized with each other.

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] as follows:

**Conservative** that take precautions to avoid the possibility of processing data out of time stamp order, i.e., execution mechanism avoids synchronization errors.

**Optimistic** that does not necessarily avoid synchronization errors, but rather use 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.

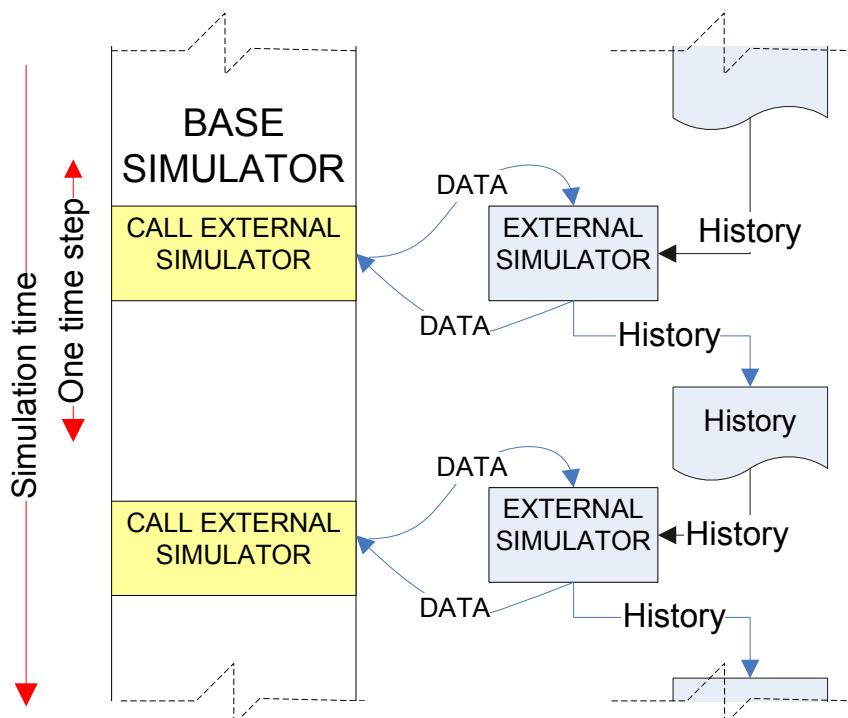
BPS is normally in the time domain, where each time advance made by a simulator is of some fixed duration of simulation time. Coupled simulators need to know whether all required information in the current simulation time step has been received. This brings couple of important points regarding synchronization realization: (i) each data to be exchanged must have a time stamp, (ii) simulators must not receive any data with time stamp less than its current simulation time, and (iii) simulators need to have mechanism of determining whether all necessary coupling data in the current simulation time step have been received.

Depending on how it is realized, time management can be *internal* and *external*. Internal time management indicates that the synchronization checking procedure is coded within the simulators themselves. It requires additional coupling data (time stamp). External time management is encompassed within the inter process communication (IPC) mechanisms, e.g., applying blocking modes. To ensure that the coupling data with a correct time stamp is exchanged, an additional procedure can be added.

### 3.2.7 Management of simulators' execution

Two distinct mechanisms are recognized:

- Mechanism with *discontinuous* external simulator run, where the base simulator invokes external simulator whenever required and coordinates the information exchange between them. External simulator is thus started, executed and ended each time step, during the overall simulation time (Figure 3.6). Djunaedy [2005] implements this mechanism. To keep the dynamic evolution of the results, if necessary, the necessary information is externalized.



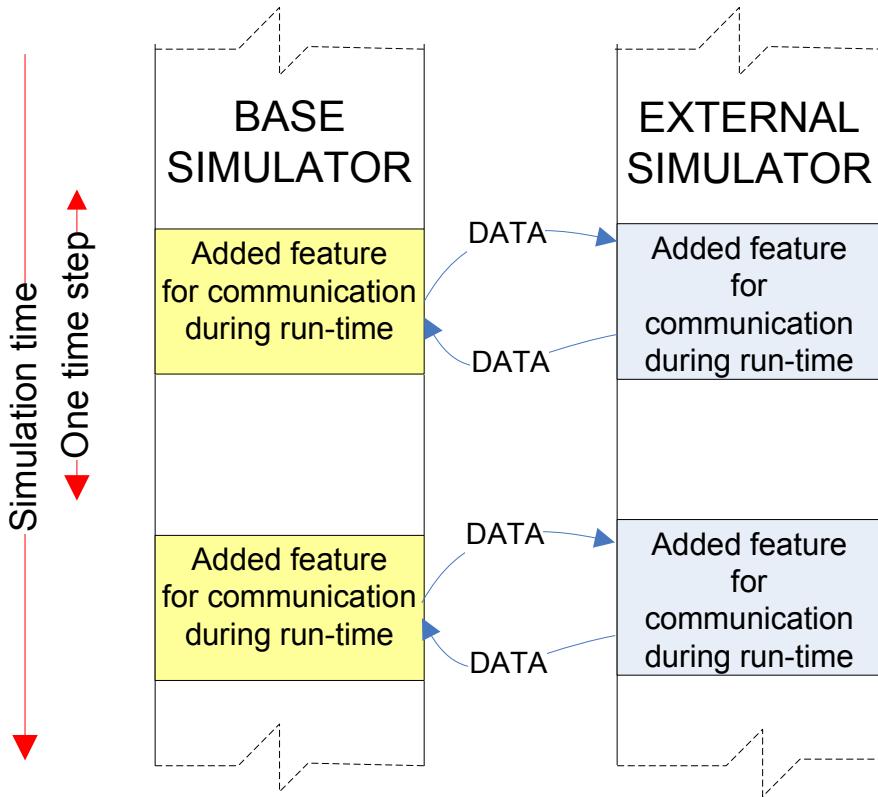
**Fig. 3.6** — Flow chart -Mechanism with discontinuous external simulator run.

- Mechanism with *continuous* external simulator run, where all coupled simulators run continuously. The coupled simulators are synchronized and exchange coupling data at (predetermined) coupling time points (Figure 3.7).

### 3.2.8 Coupling strategies

Based on how coupled simulators interact in (bi-directional) co-simulation, two different coupling strategies are distinguished, as follows.

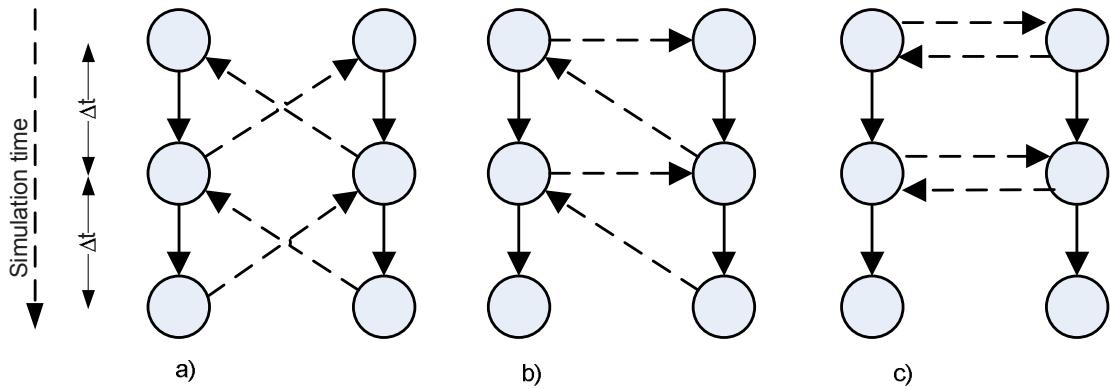
- *Fully-dynamic* [Zhai 2003], or *strong coupling* [Struler et al. 2000], or *onion coupling* [Hensen 1999], requires an iteration procedure to ascertain user-defined convergence criteria. It allows longer time steps for the same accuracy compared to the previous strategy, since iterations discard time delays.



**Fig. 3.7** — Flow chart - Mechanism with continuous external simulator run.

- *Quasi-dynamic coupling* [Zhai 2003], or *loose coupling* [Strulier et al. 2000], or *ping-pong coupling* [Hensen 1999], where coupled simulators use the predicted coupling data. The simulators can run in sequence or in parallel. If the simulations are executed in sequence (the base simulator always waits until the external simulator finishes its time step calculation), the coupling data to only one of the simulators is predicted based on the data in the previous time step(s), while the other simulation receives updated coupling data. This is referred to as *sequential staggered solution* [Felippa et al. 1999]. If the simulators are executed in parallel, the feedback between the simulators is lagged one coupling time step (or predicted based on the data in the previous time step(s)). This is referred to as *naive modification for parallel processing* [Felippa et al. 1999]. The latter implementation is faster, but the former is more accurate. In this thesis loose coupling refers to the sequential staggered solution.

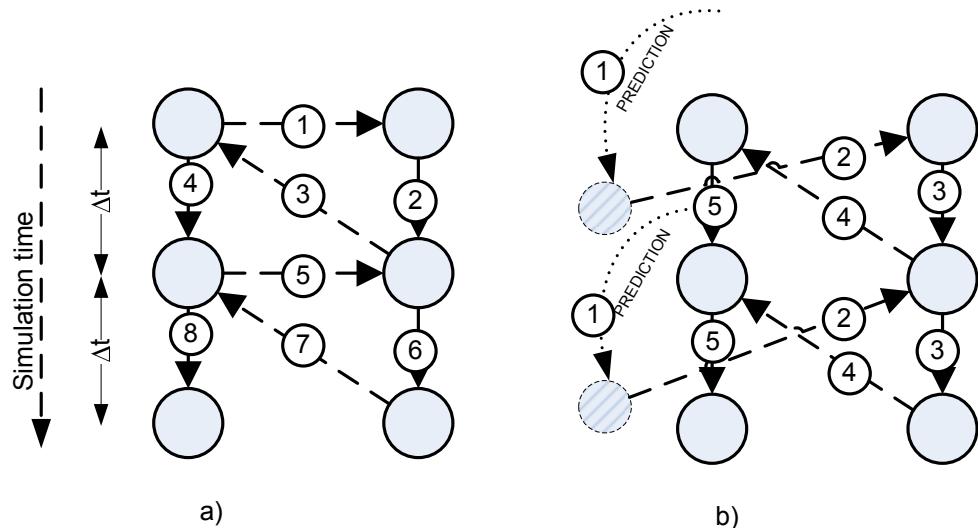
The differences in time-step information flow between the coupling strategies is shown in Figure 3.8. In the figure, the circles represent the subsystem's state at a specific moment in simulation time. The dashed arrows indicate which coupling data (time-step wise) are available to each subsystem before the time step calculation is performed. The full line arrows indicate the time step state propagations.



**Fig. 3.8** — 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.

Advantages and disadvantages of these approaches are addressed elsewhere [Hensen 1999].

The difficulty of applying iterations again differs between the coupling mechanisms employed, i.e., discontinuous or continuous and is dependant on the nature of the coupled component model (steady state or dynamic).



**Fig. 3.9** — Sequence of coupling data exchange. a) Time-state scheme without prediction; b) Time-state scheme with prediction of the coupling data sent from the base to the external simulator.

In the case where the external component is modeled in a steady state manner, there is no need for history data. Both quasi and fully dynamic coupling strategies can be used in combination with both discontinuous and continuous implementation mechanisms. For the coupling to a transient component model, history data plays an important role. A “passive” simulator, i.e., a simulator that does

not control the iteration process, must have an additional mechanism to rewind its state, based on a request from the outside, in order to ensure consistency of simulation data and synchronization of simulation time between coupled simulators. This can significantly increase the effort for the code adjustments.

To enhance the accuracy of the loose coupling strategy, the base simulator instead of sending the coupling data available in the last time step can send the predicted values of the coupling data, based on their values in several preceding time steps. The difference in state-time information flow for loose coupling with and without predictors is shown in Figure 3.9. The dotted line presents the predicted coupling data.

### 3.2.9 Coupling frequency

Coupling frequency directly influences the accuracy and stability of co-simulation results. Chapter 4 formulates explicitly how the error introduced by the partitioning of a system in co-simulation depends on the coupling frequency.

In general, to achieve a required accuracy of co-simulation results, coupling frequency can be determined by (i) the time constant of the “receiving” coupling node, as it determines the sensitivity of the “receiving” coupling node to the coupling data and (ii) the rate of change of the coupling data, which is driven by the “sending”-subsystem node’s capacitance and the rate of change of its input and the forcing function acting upon it<sup>3</sup>.

In other words, if the user can have a choice where to decompose a system being modeled, an explicit expression of the numerical error produced in a single step could be used to make decisions where to decompose a system. The error is a function of the characteristics of the receiving node and the coupling data, so choosing a node which will receive the delayed data, accuracy can be enhanced with the same time step, or a bigger time step might be used for the same accuracy.

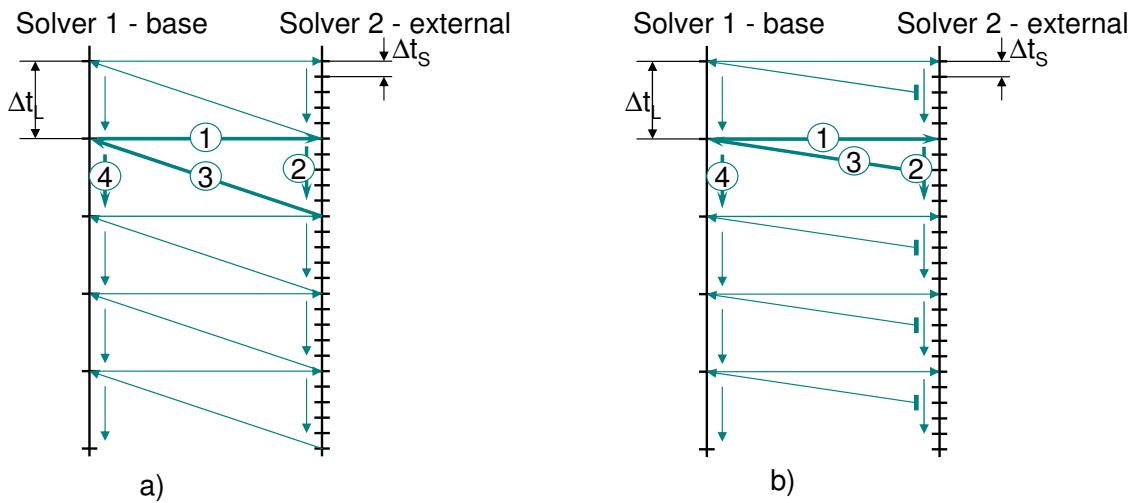
### 3.2.10 Multi-rate co-simulation

Multi-rate co-simulation can be used for simulation of stiff systems (systems that are comprised of subsystems with vastly different time constants<sup>4</sup>). The stiff systems are decomposed by isolating the most rapidly or the most slowly responding subsystem and applying the most suitable method and step length for their simulations. Different time-state schemes are possible. Larsson [2001] reports that “if only straightforward exchange of state-variables is performed among the simulators, the coupling may generate numerical problems.”. One of the options to improve the coupling is to use extrapolation and/of interpolation of coupling data [Elliott 2000; Gravouil and Combescure 2001; Elliott 2002; Grott et al. 2003]. The various time-state schemes are shown in Figures 3.10, 3.11, 3.12, 3.13, where ‘external’ refers to an external simulator and ‘base’ refers to a base simulator.

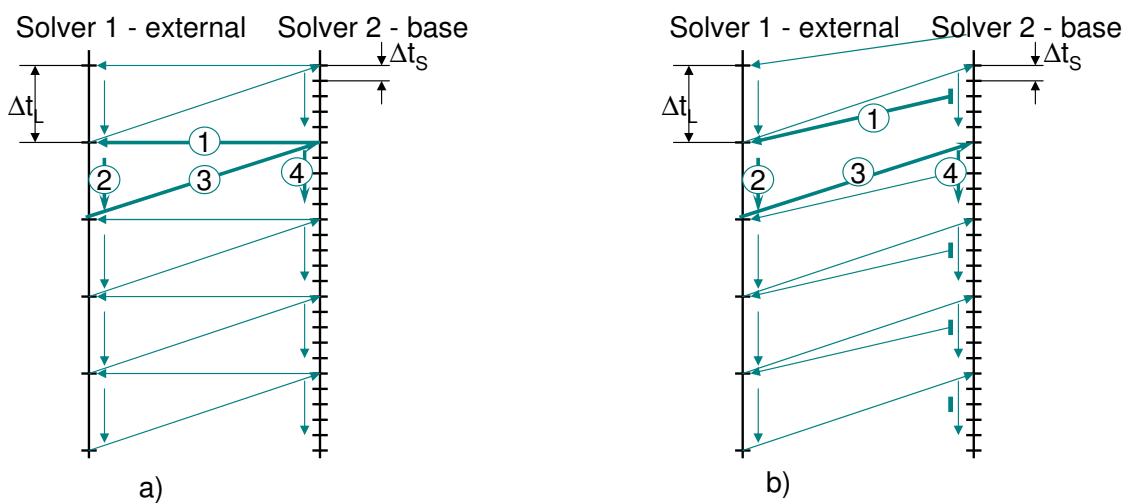
---

<sup>3</sup>The explicit dependency is discussed in Chapter 4.

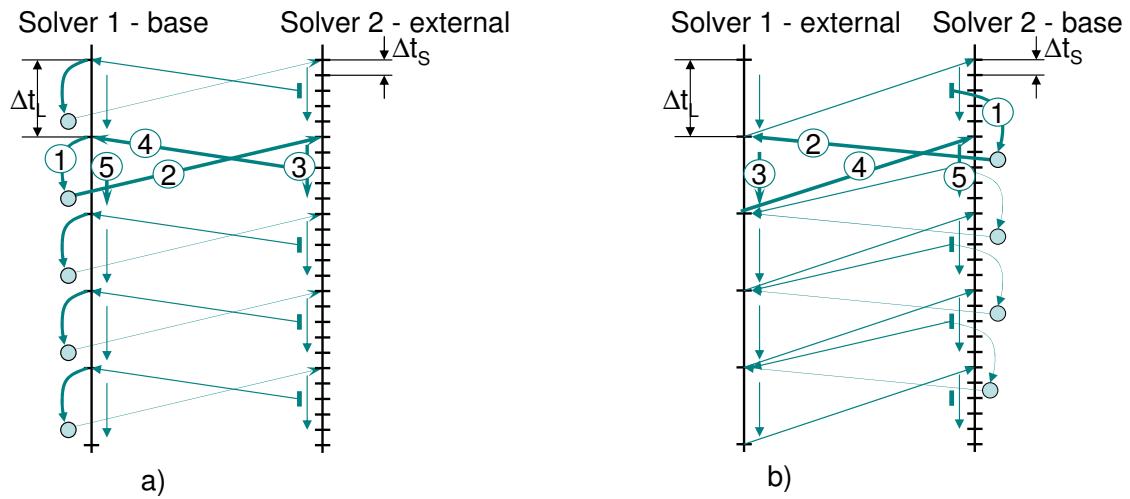
<sup>4</sup>Time constants of coupled HVAC subsystems can range from minutes (seconds), to months and even years in the case of ground-coupled heat exchangers.



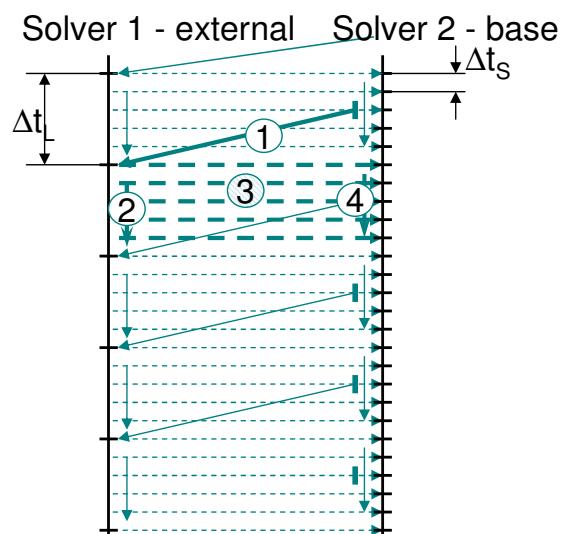
**Fig. 3.10** — Sequence of coupling data exchange. a) Time-state scheme with instantaneous coupling data; b) Time-state scheme with averaged coupling data.



**Fig. 3.11** — Sequence of coupling data exchange. a) Time-state scheme with instantaneous coupling data; b) Time-state scheme with averaged coupling data.



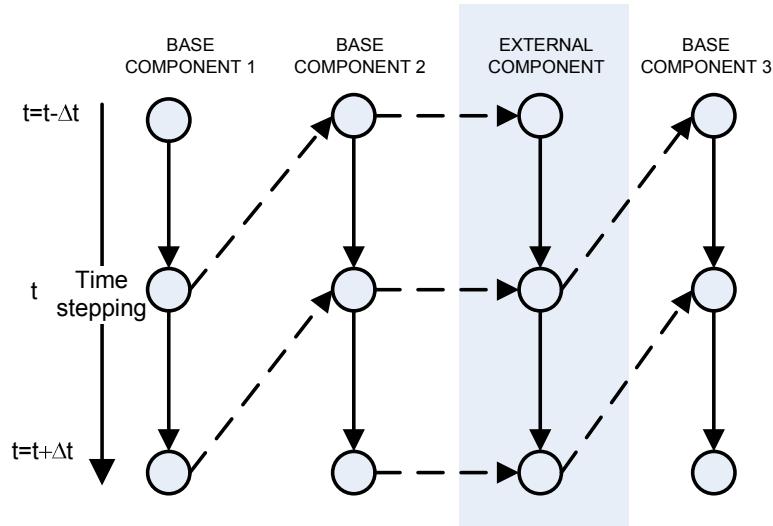
**Fig. 3.12** — Sequence of coupling data exchange. a) Time-state scheme with predicted and averaged coupling data; b) Time-state scheme with predicted averaged coupling data;



**Fig. 3.13** — Sequence of coupling data exchange. Time-state scheme with averaged coupling data and interpolation.

### 3.2.11 Inter vs. intra time step data exchange

Loosely coupled co-simulation will result in different time-step data exchange depending on which system solution technique<sup>5</sup> is used, as shown in Figures 3.14, 3.15 and 3.16.

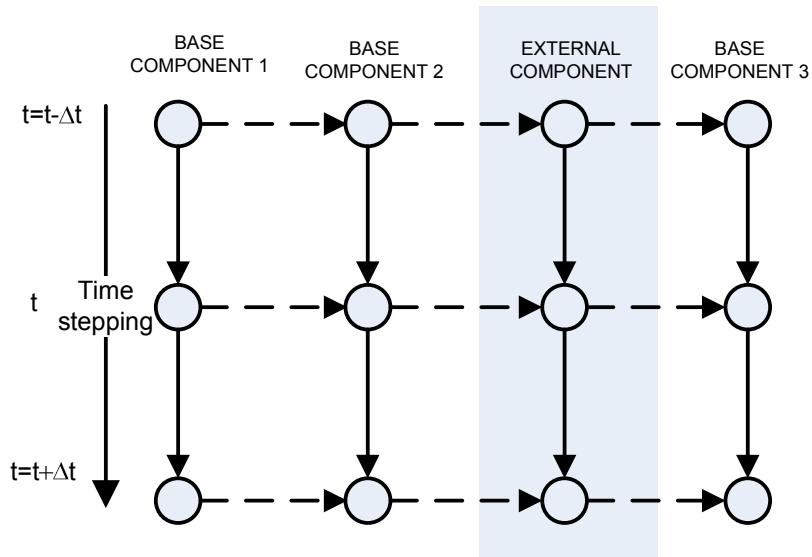


**Fig. 3.14** — State-time diagram assuming that the base simulator uses simultaneous modular solution technique with a fully implicit numerical integration scheme and is loosely coupled to the external simulator.

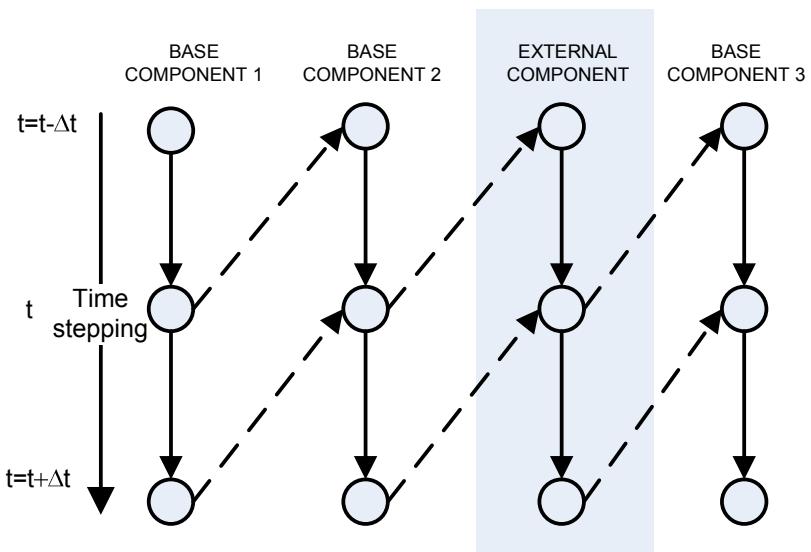
The circles represent the subsystem's state at a specific moment in simulation time. The dashed arrows indicate which coupling data are available to each subsystems' component before the time step calculation is performed. The full line arrows indicate the time step state integration. The grey boxes indicate the external simulator. The values of the coupling data, communicated from the base to the external simulator and used for external state integration from  $t$  to  $t + \Delta t$ , are only available from the simulation time  $t$ . Thus, the inter-simulation's time step data flow will disturb the original intra-time step data flow of the base simulator that uses the simultaneous modular solution technique with an implicit numerical scheme (Figure 3.14). However, if the simultaneous modular solution technique with an explicit numerical scheme is used in the base simulator (Figure 3.15) co-simulation will not disturb the data flow. The same applies for the independent modular solution technique (Figure 3.16), at least for the first iteration pass. However, if internal iteration procedure is employed (which is usually the case) and simulators are loosely coupled, the state-time diagram for the last case will look like the one in Figure 3.14.

---

<sup>5</sup>Solution techniques can be simultaneous modular or independent modular (see §2.2.4).



**Fig. 3.15** — State-time diagram assuming that the base simulator uses simultaneous modular solution technique with a fully explicit numerical integration scheme and is loosely coupled to the external simulator.



**Fig. 3.16** — State-time diagram assuming that the base simulator uses an independent modular solution technique and is loosely coupled to the external simulator. The simulators are sequentially executed.

# 4

## Co-simulation - Stability and accuracy

WHEN using co-simulation, computationally expensive systems of coupled differential and algebraic equations are partitioned and solved in separate simulators. As discussed in previous chapters the coupling between the simulators can be done by implementing either loose or strong coupling strategy. The difficulty is that the building performance simulators typically contain legacy code with more than 100 000 lines of code that mixes physical equations, data and numerical solution algorithms. On the one hand, this makes state-history rewinding, which is necessary for the strong coupling approach, a difficult task. On the other hand, the loose coupling approach is more easily implemented, but due to time-delay of the coupling data, the original numerical schemes need to be partitioned. Consequently, the stability and accuracy of the resulting numerical scheme are no longer guaranteed. The focus of this chapter is to investigate the characteristics of co-simulation implementing the loose coupling strategy.

Although the stability and accuracy of different approximation schemes are well known [Gear 1971; Lambert 1991; Golub and Ortega 1992], the stability and accuracy of the methods resulting from partitioning are rarely considered. As stated in [Kubler 2000], it is difficult, if not impossible, to determine these properties in a general sense.

In this chapter, the co-simulation problem is stated for the general class of problems that are encountered in BPS, but the stability and accuracy characteristics are obtained only for few simple linear one-step numerical schemes, applied to a first-order linear initial value problem.

### 4.1 Numerical integration schemes

This chapter considers first order initial value ordinary differential equations of the form

$$\begin{aligned}\dot{\mathbf{y}}(t) &= \mathbf{f}(\mathbf{y}, t), \\ \mathbf{y}(a) &= \boldsymbol{\eta},\end{aligned}\tag{4.1}$$

where for some  $a, b \in \mathbb{R}$ , with  $a < b$  and for some  $t \in [a, b]$  and for some  $m \in \mathbb{N}$ ,  $\boldsymbol{\eta} \in \mathbb{R}^m$  and  $\mathbf{f}(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^m$ .

**Assumption 4.1.1.** It is assumed that there exists a constant  $L \in (0, \infty)$  such that for all  $t_1, t_2 \in [a, b]$  and all  $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^m$ , the following relations hold:

$$\|\mathbf{f}(\mathbf{y}_1, t_1) - \mathbf{f}(\mathbf{y}_2, t_2)\| \leq L(\|\mathbf{y}_1 - \mathbf{y}_2\| + \|t_1 - t_2\|), \quad (4.2a)$$

$$\left\| \frac{\partial \mathbf{f}(\mathbf{y}_1, t_1)}{\partial t} - \frac{\partial \mathbf{f}(\mathbf{y}_2, t_2)}{\partial t} \right\| \leq L(\|\mathbf{y}_1 - \mathbf{y}_2\| + \|t_1 - t_2\|), \quad (4.2b)$$

$$\left\| \frac{\partial \mathbf{f}(\mathbf{y}_1, t_1)}{\partial \mathbf{y}} - \frac{\partial \mathbf{f}(\mathbf{y}_2, t_2)}{\partial \mathbf{y}} \right\| \leq L(\|\mathbf{y}_1 - \mathbf{y}_2\| + \|t_1 - t_2\|), \quad (4.2c)$$

i.e.,  $\mathbf{f}(\cdot, \cdot)$  is Lipschitz continuous in  $\mathbf{y}$  and in  $t$ , and Lipschitz continuously differentiable in  $\mathbf{y}$  and in  $t$ .  $\square$

Assumption 4.1.1 ensures that a unique solution to (4.1) exists.

To approximate the solution of the initial value problem (4.1), numerical methods typically involve discretization. In (equidistant) discretization a continuous interval  $[a, b]$  is replaced by the discrete point set  $\{t^n \in \mathbb{R} \mid t^n = a + n\Delta t; n = 0, 1, 2, \dots, N\}$ , where  $\Delta t = (b - a)/N$  is called the step length (or the time step if time is the independent variable). After discretization, a numerical method produces a sequence  $\{\mathbf{y}^n \in \mathbb{R}^m | n = 0, 1, 2, \dots, N\}$ . The value  $\mathbf{y}^n$  approximates the solution of (4.1) in the discrete point  $t^n$  (i.e.,  $\mathbf{y}(t^n)$ ). The rule for computing these values is given in form of a difference equation, in which  $\mathbf{y}^n$  is calculated in terms of the values of  $\mathbf{y}^n$  at preceding  $n-1, n-2, \dots, n-k$  discretization points. The integer  $k$  is the step-number of the numerical method. If  $k = 1$ , the numerical method is called an one-step method; if  $k > 1$ , it is called a multi-step method. A  $k$ -step general numerical method approximation to the solution of the initial value problem (4.1) can, therefore, be written as [Lambert 1991]:

$$\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} = \Delta t \phi_f(\mathbf{y}^{n+k}, \dots, \mathbf{y}^n, t^n; \Delta t) \quad (4.3a)$$

$$\mathbf{y}^\mu = \eta^\mu, \mu = 0, 1, \dots, k-1 \quad (4.3b)$$

where the subscripts  $f$  on the right hand side indicate that the function  $\phi$ , which characterizes the particular method, depends on  $f$ .

For linear multi step methods the system (4.3a) can be written as

$$\sum_{j=0}^k \alpha_j \mathbf{y}^{n+j} = \Delta t \sum_{j=0}^k \beta_j \mathbf{f}(\mathbf{y}^{n+j}, t^{n+j}), \quad (4.4)$$

where  $\{\alpha_i, \beta_i \in \mathbb{R} | i = 0, 1, \dots, k\}$  are method-specific parameters. For linear one-step methods (4.4) leads to:

$$\alpha_0 \mathbf{y}^n + \alpha_1 \mathbf{y}^{n+1} = \Delta t (\beta_0 \mathbf{f}(\mathbf{y}^n, t^n) + \beta_1 \mathbf{f}(\mathbf{y}^{n+1}, t^{n+1})).$$

The study is limited to members of the time weighting factor - parameter  $\alpha$  - family of linear one-step numerical methods:

$$\mathbf{y}^{n+1} - \mathbf{y}^n = \Delta t [(1 - \alpha) \mathbf{f}(\mathbf{y}^n, t^n) + \alpha \mathbf{f}(\mathbf{y}^{n+1}, t^{n+1})], \quad (4.5)$$

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

## 4.2 Consistency

There are two types of errors that appear during numerical approximation to solution of the differential equations:

1. round-off errors that result from inability to store numbers with innumerable decimal places;
2. discretization or truncation errors that result from the applied numerical methods.

Consistency of a numerical method is only concerned with the second type of the error.

The following definitions are taken from the literature, e.g., [Lambert 1991].

**Definition 4.2.1** (Local truncation error). *The local truncation error is the error produced in a single step starting from the exact solution:*

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

□

**Definition 4.2.2** (Unit local truncation error). *The unit local truncation error is defined as:*

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

□

**Definition 4.2.3** (Consistency). *Consider problem (4.1) and suppose Assumption 4.1.1 is satisfied. The numerical method is said to be consistent if for all initial value problems the unit local truncation error satisfies*

$$\lim_{\Delta t \rightarrow 0} \text{ULTE}^{n+k}(\Delta t) = 0.$$

□

## 4.3 Stability

As stated in [Golub and Ortega 1992], the meaning of the term stability depends on the context. In general, one can distinguish between *the problem* and *the numerical method stability* [Golub and Ortega 1992; Gear 1971]<sup>1</sup>.

The problem (4.1) is unstable if arbitrary small changes in the initial conditions can produce arbitrary large changes in the solution as  $t \rightarrow \infty$ . In the terminology of numerical analysis, problems with unstable solution are said to be *ill-conditioned*, as it would be hard to obtain the solution numerically. The round-off

---

<sup>1</sup>The definitions for problem stability and stability of numerical method are given elsewhere [Lambert 1991] and will not be stated here.

and truncation errors would cause the same effect as changes in initial conditions and the approximate solution may diverge for a finite integration interval. However, if the Lipschitz condition is satisfied (as assumed), the initial value problem (4.1) is stable.

For numerical method stability properties, the distinction is made between the zero-stability and the absolute stability.

### 4.3.1 Zero-stability

Assuming that the initial value problem (4.1) satisfies the Lipschitz condition, the linear numerical method (4.4) 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

$$\rho(r) = \sum_{j=0}^k \alpha_j r^j \tag{4.6}$$

is the first characteristic polynomial of the numerical method. Let roots of the  $\rho(r)$  be  $\{r_i \in \mathbb{C} \mid i = 1, 2, \dots, k\}$ . The location of the roots of the first characteristic polynomial defines the zero-stability of the numerical method (4.4). The numerical method is said to be zero-stable if all the roots of the (first) characteristic polynomial (4.6) satisfy  $|r_i| \leq 1$ , and any root for which  $|r_i| = 1$  is simple. The numerical method is *strongly stable* if, in addition,  $k - 1$  roots of (4.6) satisfy  $|r_i| < 1$ . The numerical method is *weakly stable* if it has more than one root on the unit circle.

The zero-stability is only concerned with what happens in the limit as  $\Delta t \rightarrow 0$ . It is a property of a numerical method (4.4) and not of the differential equation (4.1).

### 4.3.2 Absolute stability

The convergence in itself is not a guarantee that a numerical method will yield acceptable numerical results as it describes only what happens at  $\Delta t \rightarrow 0$ . For some numerical methods, there exists a value  $\Delta t_0$  of the step-length such that for fixed  $\Delta t > \Delta t_0$  the method is unstable, whereas for fixed  $\Delta t < \Delta t_0$ , the method is stable. This stability phenomenon is of the form that is distinct from the zero-stability. It considers error behavior in a numerical method using a finite value of  $\Delta t$  and is known as *absolute stability* [Lambert 1991; Hall and Watt 1976; Gear 1971].

For the analysis of the absolute stability, the second characteristic polynomial of the linear numerical method (4.4) is used. The polynomial is obtained applying

(4.4) to a linear constant coefficient homogeneous system  $\dot{\mathbf{y}} = \mathbf{A}\mathbf{y}$ :

$$\sigma(r) = \sum_{j=0}^k (\alpha_j - \Delta t \beta_j \lambda_i) r^j, \quad (4.7)$$

where  $\{\lambda_i \in \mathbb{C} | i = 1, 2, \dots, m\}$  are the eigenvalues of  $\mathbf{A}$ . The location of the roots of this stability polynomial defines the region of absolute stability of the numerical method. The linear numerical method is said to be absolutely stable for a given product  $\Delta t \lambda$ , where  $\lambda$  represents any of the eigenvalues  $\lambda_i$  of  $\mathbf{A}$ , if for that product  $(\Delta t \lambda)$  all the roots of the stability polynomial, satisfy  $|r_i| < 1, i = 1, 2, \dots, k$ , and to be unstable for that product  $(\Delta t \lambda)$  otherwise [Lambert 1991]. The numerical method (4.4) is said to have a region of absolute stability  $\mathfrak{R}_A$ , where  $\mathfrak{R}_A$  is a region of the complex  $\Delta t \lambda$ -plane, if it is absolutely stable for all  $\Delta t \lambda \in \mathfrak{R}_A$ . The intersection of  $\mathfrak{R}_A$  with the real axis is called the interval of absolute stability.

Note that the region of the absolute stability in  $\Delta t \lambda$  depends only on the used linear numerical method. However,  $\lambda$  is defined by the problem and will determine the maximum step size  $\Delta t$  to be used in order to ensure stability.

It should be noted that for a general system of the form (4.1),  $\lambda_i, i = 1, 2, \dots, m$  are the eigenvalues of a matrix defined as  $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{y}}$ . The theory propounded above is, however, restrictive to a constant matrix  $\mathbf{J}$ , which is not always the case [Lambert 1991].

## 4.4 Convergence

As  $\Delta t \rightarrow 0$ , the discrete point set converges to the continuous interval  $[a, b]$ .

**Definition 4.4.1** (Convergence [Lambert 1991]). *Consider problem (4.1) and let  $X_n \triangleq \{t^n \in \mathbb{R} \mid t^n = a + n\Delta t; n = 0, 1, \dots, (b-a)/\Delta t\}$  and suppose Assumption 4.1.1 is satisfied. The numerical method is said to be convergent if for all  $t^n \in X_n$ :*

$$\lim_{\Delta t \rightarrow 0} \mathbf{y}^n = \mathbf{y}(t^n).$$

□

The necessary and sufficient condition for a numerical method to be convergent is that it is both consistent and zero-stable [Lambert 1991; Gear 1971].

## 4.5 Consistency of co-simulation

In co-simulation, the system of equations (4.3a) is first partitioned algebraically, as described in §3.2, and then solved in coupled simulators. The coupling data depends on  $\{\mathbf{y}^{n+j} \in \mathbb{R}^m \mid j = 0, 1, 2, \dots, k; n = 0, 1, \dots, N\}$ . 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 known data in 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 the (4.3a) leads to

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

and the system of two equations, (4.5) leads to

$$\mathbf{y}^{n+1} - \mathbf{y}^n = \Delta t[(1-\alpha)\mathbf{f}(\mathbf{y}^n, t^n) + \alpha\mathbf{f}(\mathbf{y}_P^{n+1}, t^{n+1})]. \quad (4.8)$$

To determine the consistency of co-simulation, the Definition 4.2.3 is used. For the numerical approximation (4.8), the local truncation error is:

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) &= \frac{1}{\Delta t} \left( \mathbf{y}(t^{n+1}) - \mathbf{y}(t^n) \right. \\ &\quad \left. - \Delta t[(1-\alpha)\mathbf{f}(\mathbf{y}(t^n), t^n) + \alpha\mathbf{f}_P(\mathbf{y}(t^{n+1}), t^{n+1})] \right). \end{aligned} \quad (4.9)$$

Adding and subtracting the same value  $\alpha\mathbf{f}(\mathbf{y}(t^{n+1}), t^{n+1})$  to the right hand side of (4.9) yields

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) &= \frac{1}{\Delta t} \left( \mathbf{y}(t^{n+1}) - \mathbf{y}(t^n) - \Delta t[(1-\alpha)\mathbf{f}(\mathbf{y}(t^n), t^n) \right. \\ &\quad + \alpha\mathbf{f}(\mathbf{y}(t^{n+1}), t^{n+1}) - \alpha\mathbf{f}(\mathbf{y}(t^{n+1}), t^{n+1}) \\ &\quad \left. + \alpha\mathbf{f}_P(\mathbf{y}(t^{n+1}), t^{n+1})] \right) \end{aligned}$$

Collecting terms that correspond to the local truncation error of the original, non-partitioned numerical scheme

$$\begin{aligned} \text{ULTE}_{\text{non-partitioned}}^{n+1} &= \frac{1}{\Delta t} \left( \mathbf{y}(t^{n+1}) - \mathbf{y}(t^n) \right. \\ &\quad \left. - \Delta t[(1-\alpha)\mathbf{f}(\mathbf{y}(t^n), t^n) + \alpha\mathbf{f}(\mathbf{y}(t^{n+1}), t^{n+1})] \right) \end{aligned}$$

yields

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) &= \text{ULTE}_{\text{non-partitioned}}^{n+1} \\ &\quad + \alpha[\mathbf{f}(\mathbf{y}(t^{n+1}), t^{n+1}) - \mathbf{f}_P(\mathbf{y}(t^{n+1}), t^{n+1})]. \end{aligned} \quad (4.10)$$

Applying the norm on both sides of (4.10) and since  $\mathbf{f}(\cdot, \cdot)$  is by Assumption 4.1.1 Lipschitz continuous in  $\mathbf{y}$ , (4.10) yields

$$\begin{aligned} \|\text{ULTE}^{n+1}(\Delta t)\| &\leq \|\text{ULTE}_{\text{non-partitioned}}^{n+1}\| \\ &\quad + \alpha L \|(\mathbf{y}(t^{n+1}) - \mathbf{y}_P(t^{n+1}))\|, \end{aligned} \quad (4.11)$$

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  $\mathbf{y}(t^n)$ , can be expressed around time  $t^n + \alpha\Delta t$ , for any  $\alpha \in [0, 1]$ , by means of a Taylor series as follows:

$$\begin{aligned} \mathbf{y}(t^{n+1}) &= \mathbf{y}(t^n + \alpha\Delta t) + (1-\alpha)\Delta t \dot{\mathbf{y}}(t^n + \alpha\Delta t) \\ &\quad + \frac{((1-\alpha)\Delta t)^2}{2} \ddot{\mathbf{y}}(t^n + \alpha\Delta t) + \dots \end{aligned} \quad (4.12a)$$

$$\begin{aligned}\mathbf{y}(t^n) &= \mathbf{y}(t^n + \alpha\Delta t) - \alpha\Delta t \dot{\mathbf{y}}(t^n + \alpha\Delta t) \\ &\quad + \frac{(\alpha\Delta t)^2}{2} \ddot{\mathbf{y}}(t^n + \alpha\Delta t) + \dots\end{aligned}\tag{4.12b}$$

Because  $\alpha \in [0, 1]$  is arbitrary, and thereby the same  $\alpha$  as used in (4.8) can be picked. Substituting (4.12) into (4.11) and using the zero-order predictor  $\mathbf{y}_P(t^{n+1}) = \mathbf{y}(t^n)$  yields

$$\|\text{ULTE}^{n+1}(\Delta t)\| \leq \|\text{ULTE}_{\text{non-partitioned}}^{n+1}\| + \alpha L O(\Delta t).\tag{4.13}$$

It follows from Equation (4.13) and from  $\lim_{\Delta t \rightarrow 0} \alpha L O(\Delta t) = 0$  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 the first order accurate methods will not change by the partitioning. However, for the methods with the accuracy of the higher order, such as Crank-Nicholson method ( $\alpha = 1/2$ ), which is of the order two, the accuracy will be reduced by the partitioning.

In loose coupling with naive modification for parallel processing every element of  $\mathbf{y}_P^{n+1}$  needs prediction. However, in loose coupling with sequential staggered solution, only some elements of  $\mathbf{y}_P^{n+1}$  need prediction. Since this thesis is concerned with the latter, a more detailed consistency analysis is reported in §4.7 using a specific two-body problem. Two different cases of partitioning are examined. One, called implicit-implicit, where the coupling data depends only on the state of the subsystem simulated in the coupled simulator, and another, called implicit-explicit, where the coupling data is a function of both states.

## 4.6 Zero-stability and convergence of co-simulation

By inspection of the partitioned linear numerical method (4.8), it can be seen that the partitioning changes only the right hand side of the Equation (4.8). The first characteristic polynomial (4.6), that determines zero-stability of a numerical method, depends only on the coefficients on the left hand side of (4.8) 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 method.

Since by (4.13) it was shown that the numerical method (4.8) is consistent, and since the zero-stability of the original numerical scheme is not changed with partitioning, it follows that the numerical method (4.8) is convergent.

However, the zero-stability from the first characteristic polynomial does not completely cover the situations analyzed in [Kubler 2000]. Kubler analyzed a situation where co-simulated subsystems in a loop are modeled 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. In such situations, the coupled integration, illustrated in Figure 4.1 is presented by a coupled

system of equations that consists of (i) a partitioned difference equation, e.g., for linear one step methods in the form (4.8)

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \Delta t[(1 - \alpha)\mathbf{f}(\mathbf{y}^n, t^n) + \alpha\mathbf{f}(\mathbf{y}_P^{n+1}, t^{n+1})], \quad (4.14a)$$

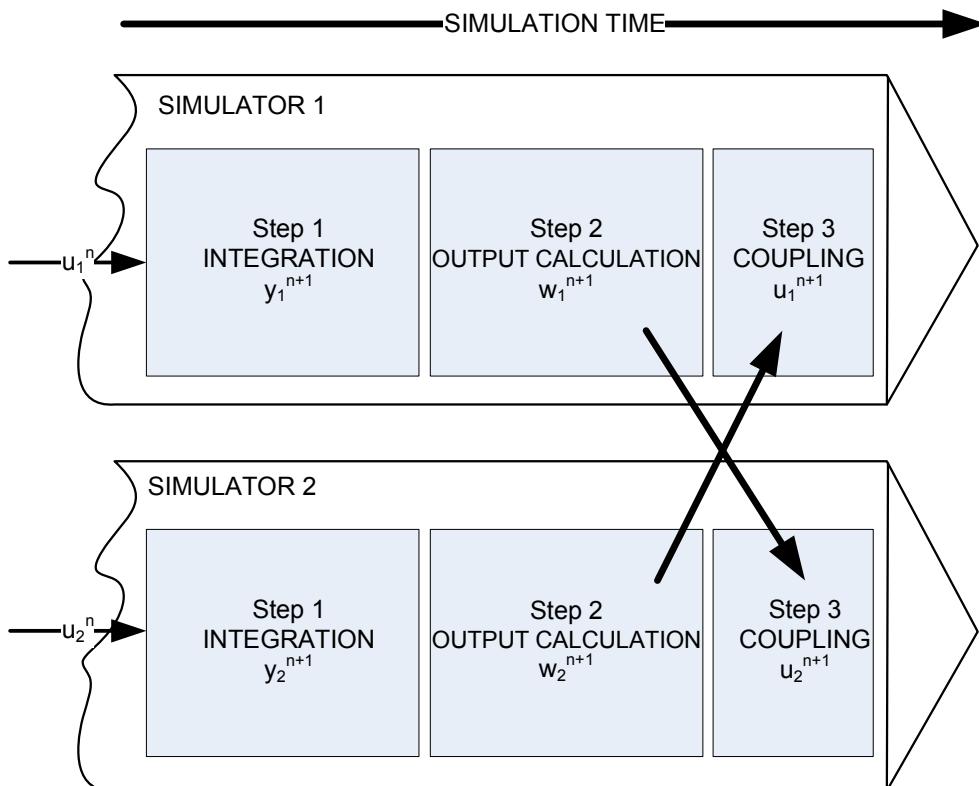
(ii) an output equation defined as

$$\mathbf{w}^{n+1} = \varphi(\mathbf{y}^{n+1}, \mathbf{u}^n, t^{n+1}), \quad (4.14b)$$

and (iii) an input vector defined as

$$\mathbf{u}^n = \mathbf{L}\mathbf{w}^n, \quad (4.14c)$$

where  $\mathbf{w}^{n+1}$  is an output vector,  $\mathbf{u}^n$  is an input vector that satisfies  $\|\mathbf{u}^{n+1} - \mathbf{u}^n\| \rightarrow 0$  as  $\Delta t \rightarrow 0$  and  $\mathbf{L}$  is the incidence matrix of which elements are zero or one representing not existing or existing interconnections of corresponding inputs and outputs.



**Fig. 4.1** — Illustration of the loose coupling as analyzed in [Kubler 2000].

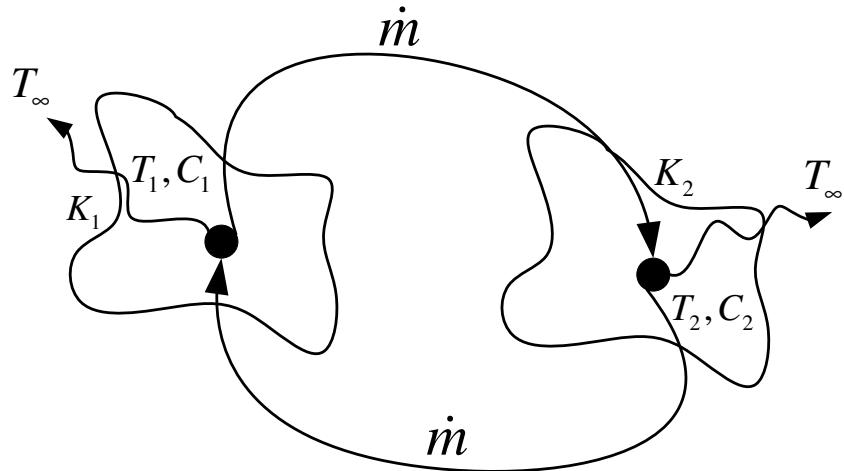
The stability of the first characteristic polynomial ensures the zero-stability only of (4.14a). Under assumptions that the output equations are time invariant and linear functions of the inputs, Kubler [2000] showed that the stability of (4.14b) and (4.14c), for two coupled subsystems, is ensured if 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. Since, this chapter

only considers co-simulation of differential equations, where the subsystems output is a function of the subsystems state, the situations analyzed by Kubler [2000] are not discussed further.

The absolute stability of the partitioned numerical method (4.8) will be examined in the following section using a specific two-body problem.

## 4.7 Analysis of co-simulation for a two-body system

To gain insight into solution characteristics of the partitioned numerical simulation of building and HVAC simulators, the study is performed on a simple system as shown in Figure 4.2. It constitutes of two subsystems with lumped capacities. Each subsystem is represented by a single state variable. Each subsystem interacts with the surrounding environment through heat transfer by conduction and convection and with each other through fluid flow. The flow is assumed to be time invariant. The consequences of a time variant flow are commented later in the chapter.



**Fig. 4.2 — Simple system.**

The subsystems are connected in a loop. In §4.7.2, a second loop is introduced through feedback control using a proportional controller that actuates the heat flux in one of the subsystems based on the temperature of the other.

### 4.7.1 Mathematical model

The system in Figure 4.2 is modeled as an initial value problem for  $t \in [a, b]$

$$\begin{aligned}\dot{T}_1(t) &= \frac{\dot{m}c_p(T_2(t) - T_1(t))}{C_1} + \frac{K_1(T_\infty(t) - T_1(t))}{C_1}, \\ \dot{T}_2(t) &= \frac{\dot{m}c_p(T_1(t) - T_2(t))}{C_2} + \frac{K_2(T_\infty(t) - T_2(t))}{C_2}, \\ T_1(a) &= T_{1,0}, \\ T_2(a) &= T_{2,0},\end{aligned}\tag{4.15}$$

where the subscripts 1 and 2 refer to the subsystems in Figure 4.2, and the subscript  $\infty$  refers to the environment,  $C$  is the overall subsystem's heat capacity,  $K \triangleq UA$ , where  $U$  is the overall heat transfer coefficient (assumed invariant) and  $A$  is the heat exchange surface area,  $\dot{m}$  is the working fluid mass flow and  $c_p$  is the specific heat capacity of the working fluid.

The matrix representation of the model, for  $t \in [a, b]$ , is

$$\begin{aligned}\dot{\mathbf{T}}(t) + \mathbf{CT}(t) &= \mathbf{g}(t), \\ \mathbf{T}(a) &= \mathbf{T}_0,\end{aligned}\tag{4.16}$$

where  $\mathbf{T}$  is the temperature vector:

$$\mathbf{T}(t) = \begin{bmatrix} T_1(t) \\ T_2(t) \end{bmatrix},$$

is a diagonal matrix for the capacity, with positive diagonal elements and thus invertible,  $\mathbf{C}$  is the interaction matrix and  $\mathbf{g}(t)$  is a forcing function. In particular,

$$\mathbf{S} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix},\tag{4.17a}$$

$$\mathbf{C} = \begin{bmatrix} \dot{m}c_p + K_1 & -\dot{m}c_p \\ -\dot{m}c_p & \dot{m}c_p + K_2 \end{bmatrix},\tag{4.17b}$$

and

$$\mathbf{g}(t) = \begin{bmatrix} g_1(t) \\ g_2(t) \end{bmatrix} = \begin{bmatrix} K_1 T_\infty(t) \\ K_2 T_\infty(t) \end{bmatrix}.\tag{4.17c}$$

In general, the flow rate and the heat transfer coefficient are functions of time and therefore the matrix  $\mathbf{C}$  is time variant. The matrices  $\mathbf{S}$  and  $\mathbf{C}$  are also temperature dependent. However, to simplify the analysis, the problem is regarded as linear with constant coefficients. The problems introduced by the time-varying coefficients will be commented after some analysis.

Applying a characteristic linear numerical method from the  $\alpha$ -family of methods (4.5) for approximating the solution of the problem (4.16), and rearranging the equations so that the known terms are placed on the right hand and the unknown terms on the left hand side, yields

$$\begin{aligned}[\mathbf{S} + \alpha \Delta t \mathbf{C}] \mathbf{T}^{n+1} &= \Delta t [\alpha \mathbf{g}(t^{n+1}) + (1 - \alpha) \mathbf{g}(t^n)] \\ &\quad + [\mathbf{S} - (1 - \alpha) \Delta t \mathbf{C}] \mathbf{T}^n\end{aligned}\tag{4.18}$$

with initial condition  $\mathbf{T}^0 = \mathbf{T}_0$ .

In loosely coupled co-simulation, the numerical scheme (4.18) changes to:

$$\begin{aligned} [\mathbf{S} + \alpha \Delta t \mathbf{C}_I] \mathbf{T}^{n+1} &= \Delta t [\alpha \mathbf{g}(t^{n+1}) + (1 - \alpha) \mathbf{g}(t^n)] \\ &\quad + [\mathbf{S} - (1 - \alpha) \Delta t \mathbf{C}] \mathbf{T}^n - \alpha \Delta t \mathbf{C}_E \mathbf{T}_P^{n+1} \end{aligned} \quad (4.19)$$

with initial condition  $\mathbf{T}^0 = \mathbf{T}_0$ , where  $\mathbf{T}_P^{n+1}$  is the predicted temperature vector at  $t^{n+1}$ , based on the known temperature vectors from the preceding steps, and  $\mathbf{C}_I$  and  $\mathbf{C}_E$  are partitioning of the matrix  $\mathbf{C}$  that will now be discussed. Different decomposition strategies lead to different partitioning of the matrix  $\mathbf{C}$ . To illustrate this, the following discussion is based on the example from Figure 4.2.

Let the second subsystem, of which specific parameters and the state variables are denoted by the subscript 2, be modeled and simulated in the external simulator. Then, if the coupling data depends on only the state variable of the coupled subsystem, e.g., coupling data sent to the external simulator is  $\dot{m} c_p T_{1,P}^{n+1}$ , the equation calculated in the external solver yields

$$\begin{aligned} [C_2 + \alpha \Delta t (\dot{m} c_p + K_2)] T_2^{n+1} &= \Delta t [\alpha g_2(t^{n+1}) + (1 - \alpha) g_2(t^n)] \\ &\quad + [C_2 - (1 - \alpha) \Delta t (\dot{m} c_p + K_2)] T_2^n \\ &\quad + (1 - \alpha) \Delta t \dot{m} c_p T_1^n \\ &\quad + \alpha \Delta t \dot{m} c_p T_{1,P}^{n+1}. \end{aligned} \quad (4.20)$$

If the coupling data depends on both state variables, e.g., coupling data sent to the external simulator is the heat flux  $\dot{Q}^{n+1} = \dot{m} c_p (T_{1,P}^{n+1} - T_{2,P}^{n+1})$ , the equation calculated in the external solver yields

$$\begin{aligned} [C_2 + \alpha \Delta t K_2] T_2^{n+1} &= \Delta t [\alpha g_2(t^{n+1}) + (1 - \alpha) g_2(t^n)] \\ &\quad + [C_2 - (1 - \alpha) \Delta t (\dot{m} c_p + K_2)] T_2^n \\ &\quad + (1 - \alpha) \Delta t \dot{m} c_p T_1^n \\ &\quad + \alpha \Delta t \dot{m} c_p T_{1,P}^{n+1} - \alpha \Delta t \dot{m} c_p T_{2,P}^{n+1}. \end{aligned} \quad (4.21)$$

If the coupling data sent to the base simulator is the heat flux  $\dot{Q}^{n+1} = \dot{m} c_p (T_2^{n+1} - T_{1,P}^{n+1})$ , the equation calculated in the base solver yields

$$\begin{aligned} [C_1 + \alpha \Delta t K_1] T_1^{n+1} - \alpha \Delta t \dot{m} c_p T_2^{n+1} &= \\ \Delta t [\alpha g_1(t^{n+1}) + (1 - \alpha) g_1(t^n)] &+ [C_1 - (1 - \alpha) \Delta t (\dot{m} c_p + K_1)] T_1^n \\ &+ (1 - \alpha) \Delta t \dot{m} c_p T_2^n \\ &- \alpha \Delta t \dot{m} c_p T_{1,P}^{n+1}. \end{aligned} \quad (4.22)$$

Thus, different system decompositions lead to different partitioning of the matrix  $\mathbf{C}$ . If the second equation from (4.15) is solved in the external simulator, the partitioning of the matrix  $\mathbf{C}$  are as follows. In case of implicit-implicit partitioning (intra-domain system decomposition) (4.20), the matrix  $\mathbf{C}$  is partitioned as:

$$\mathbf{C} = \mathbf{C}_I + \mathbf{C}_E = \begin{bmatrix} \dot{m} c_p + K_1 & -\dot{m} c_p \\ 0 & \dot{m} c_p + K_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\dot{m} c_p & 0 \end{bmatrix}. \quad (4.23)$$

For the first case of implicit-explicit partitioning (inter-domain system decomposition) (4.21), the matrix  $\mathbf{C}$  is partitioned as:

$$\mathbf{C} = \mathbf{C}_I + \mathbf{C}_E = \begin{bmatrix} K_1 & -\dot{m}c_p \\ 0 & \dot{m}c_p + K_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\dot{m}c_p & \dot{m}c_p \end{bmatrix}.$$

For the second case of implicit-explicit partitioning (inter-domain system decomposition) where the temperature is sent from the base to the external simulator (4.22), and the heat flux from the external to the base simulator (4.22), the matrix  $\mathbf{C}$  is partitioned as:

$$\mathbf{C} = \mathbf{C}_I + \mathbf{C}_E = \begin{bmatrix} K_1 & -\dot{m}c_p \\ 0 & \dot{m}c_p + K_1 \end{bmatrix} + \begin{bmatrix} \dot{m}c_p & 0 \\ -\dot{m}c_p & 0 \end{bmatrix}. \quad (4.24)$$

### 4.7.2 Consistency

To get insights on how the systems characteristics, such as the thermal capacity, influence the error introduced by the partitioning, the local truncation error  $\text{ULTE}^{n+1}(\Delta t)$  of the partitioned numerical approximation needs to be analyzed. To accomplish this, first (4.19) is written in the form (4.5), which gives

$$\begin{aligned} \mathbf{T}^{n+1} - \mathbf{T}^n &= \Delta t [(1-\alpha) \mathbf{S}^{-1}(\mathbf{g}^n - \mathbf{C} \mathbf{T}^n) \\ &\quad + \alpha \mathbf{S}^{-1}(\mathbf{g}^{n+1} - \mathbf{C}_I \mathbf{T}^{n+1} - \mathbf{C}_E \mathbf{T}_P^{n+1})]. \end{aligned} \quad (4.25)$$

Then, using the Definitions 4.2.1 and 4.2.2 the unit local truncation error of (4.25) yields

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) &= \frac{1}{\Delta t} (\mathbf{T}(t^{n+1}) - \mathbf{T}(t^n) \\ &\quad - \Delta t [(1-\alpha) \mathbf{S}^{-1}(\mathbf{g}(t^n) - \mathbf{C} \mathbf{T}(t^n)) \\ &\quad + \alpha \mathbf{S}^{-1}(\mathbf{g}(t^{n+1}) - \mathbf{C}_I \mathbf{T}(t^{n+1}) \\ &\quad - \mathbf{C}_E \mathbf{T}_P(t^{n+1}))]). \end{aligned} \quad (4.26)$$

Using the zero-order predictor  $\mathbf{T}_P(t^{n+1}) = \mathbf{T}(t^n)$ , and since  $\mathbf{g}(\cdot)$  is by Assumption 4.1.1 Lipschitz continuously differentiable in  $t$ , expressing  $\mathbf{g}(t^{n+1})$  and  $\mathbf{g}(t^n)$ , and  $\mathbf{T}(t^{n+1})$  and  $\mathbf{T}(t^n)$ , around time  $t^n + \alpha\Delta t$ , for any  $\alpha \in [0, 1]$ , by means of Taylor series (as in (4.12)), Equation (4.26) yield

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) &= \mathbf{S}^{-1}(\mathbf{S} \dot{\mathbf{T}}(t^n + \alpha\Delta t) + \mathbf{C} \mathbf{T}(t^n + \alpha\Delta t) - \mathbf{g}(t^n + \alpha\Delta t)) \\ &\quad - \frac{1}{2}(1-2\alpha) \ddot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t \\ &\quad - \alpha \mathbf{S}^{-1} \mathbf{C}_E \dot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t + O(\Delta t^2), \end{aligned} \quad (4.27)$$

where,

$$\begin{aligned} \mathbf{g}(t^{n+1}) &= \mathbf{g}(t^n + \alpha\Delta t) + (1-\alpha) \frac{\Delta t}{2} \dot{\mathbf{g}}(t^n + \alpha\Delta t) + O(\Delta t^2), \text{ and} \\ \mathbf{g}(t^n) &= \mathbf{g}(t^n + \alpha\Delta t) - \alpha \frac{\Delta t}{2} \dot{\mathbf{g}}(t^n + \alpha\Delta t) + O(\Delta t^2), \end{aligned}$$

and thus  $\alpha \mathbf{g}(t^{n+1}) + (1 - \alpha) \mathbf{g}(t^n) = \mathbf{g}(t^n + \alpha\Delta t) + O(\Delta t^2)$ .

Substituting (4.16), evaluated at time  $t^n + \alpha\Delta t$ , into (4.27) the unit local truncation error yields

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) &= \frac{1}{2}(1 - 2\alpha) \ddot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t \\ &\quad + \alpha \mathbf{S}^{-1} \mathbf{C}_E \dot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t + O(\Delta t^2). \end{aligned} \quad (4.28)$$

The term  $\frac{1}{2}(1 - 2\alpha) \ddot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t$  is caused by the original non-partitioned scheme (4.18), while the term  $\alpha \mathbf{S}^{-1} \mathbf{C}_E \dot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t$  is introduced by the partitioning, which by substituting (4.17) and (4.23) yields to  $\alpha \frac{1}{C_2} \dot{m}c_p \dot{T}_1(t^n + \alpha\Delta t) \Delta t$ . The error due to the partitioning is proportional to the factor  $\alpha$ , and thus it is zero for the explicit numerical scheme, as expected. It is also proportional to  $\dot{\mathbf{T}}(t^n + \alpha\Delta t)$  and to the reciprocal value of the capacity of the subsystem receiving the delayed data. The greater the capacity  $C_{\text{ext}}$  the smaller the error is. The lower the first derivative  $\dot{T}_{\text{base}}(t^n + \alpha\Delta t)$ , the lower  $\dot{m}c_p$  and the smaller  $\Delta t$  the smaller the error is.

The accuracy of the partitioned numerical scheme that uses (first-order) prediction for the coupling data sent by the base to the external simulator can be evaluated in the similar way, by using the exact solutions of the temperature vectors in the two subsequent time steps,  $\mathbf{T}(t^{n-1})$  and  $\mathbf{T}(t^n)$ , expressed around  $t^n + \alpha\Delta t$  by means of a Taylor series. This yields

$$\begin{aligned} \mathbf{T}_P(t^{n+1}) &= 2\mathbf{T}(t^n) - \mathbf{T}(t^{n-1}) \\ &= \mathbf{T}(t^n + \alpha\Delta t) - \alpha\Delta t \dot{\mathbf{T}}(t^n + \alpha\Delta t) + \Delta t \dot{\mathbf{T}}(t^n + \alpha\Delta t) \\ &\quad + \frac{1}{2}(\alpha^2 - 2\alpha - 1)\Delta t^2 \ddot{\mathbf{T}}(t^n + \alpha\Delta t) + \dots \end{aligned}$$

Again, substituting the above expression into (4.26), one obtains the unit local truncation error as

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) &= \frac{1}{2}(1 - 2\alpha) \ddot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t \\ &\quad - \frac{1}{2}\alpha \mathbf{S}^{-1}(\alpha \mathbf{C} - \mathbf{C}_I + \mathbf{C}_E) \ddot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t^2 + O(\Delta t^3). \end{aligned}$$

From the comparison with the error (4.28), it can be concluded that the predictor increases the order of the unit local truncation error due to the partitioning. The error introduced by the partitioning  $-\alpha \mathbf{S}^{-1} \mathbf{C}_E \ddot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t^2$  is of order two.

To comment on the influence of time variant coefficients of the matrix  $\mathbf{C}$  to the error introduced by the partitioning, the case when heat conduction system of equations (4.16) is coupled to the algebraically modeled fluid flow system of equations is discussed. In co-simulation, the algebraic system of equations is partitioned as well. Rewriting (4.26) to encounter for this yields

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) &= \frac{1}{\Delta t} (\mathbf{T}(t^{n+1}) - \mathbf{T}(t^n) \\ &\quad - \Delta t[(1 - \alpha) \mathbf{S}^{-1}(\mathbf{g}(t^n) - \mathbf{C}(t^n; \Delta t) \mathbf{T}(t^n)) \\ &\quad + \alpha \mathbf{S}^{-1}(\mathbf{g}(t^{n+1}) - \mathbf{C}_I(t^{n+1}; \Delta t) \mathbf{T}(t^{n+1}) \\ &\quad - \mathbf{C}_E(t^{n+1}; \Delta t) \mathbf{T}_P(t^{n+1})])]. \end{aligned}$$

Let  $\mathbf{C}_I(t^{n+1}; \Delta t) \triangleq \mathbf{C}_I^*(t^{n+1}; \Delta t) + \Delta\mathbf{C}_I(t_I^{n+1}; \Delta t)$  and  $\mathbf{C}_E(t^{n+1}; \Delta t) \triangleq \mathbf{C}_E^*(t^{n+1}; \Delta t) + \Delta\mathbf{C}_E(t^{n+1}; \Delta t)$ , where  $\mathbf{C}_I^*(t^{n+1}; \Delta t)$  and  $\mathbf{C}_E^*(t^{n+1}; \Delta t)$  are the unknown exact values and  $\Delta\mathbf{C}_I(t^{n+1}; \Delta t)$  and  $\Delta\mathbf{C}_E(t^{n+1}; \Delta t)$  contain the error introduced by the partitioning in one time step. Then the unit local truncation error can be written as

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) = & \frac{1}{\Delta t} (\mathbf{T}(t^{n+1}) - \mathbf{T}(t^n) \\ & - \Delta t [(1-\alpha) \mathbf{S}^{-1}(\mathbf{g}^n - [\mathbf{C}(t^n; \Delta t)] \mathbf{T}(t^n)) \\ & + \alpha \mathbf{S}^{-1}(\mathbf{g}^{n+1} - [\mathbf{C}_I^*(t^{n+1}; \Delta t) + \Delta\mathbf{C}_I(t^{n+1}; \Delta t)] \mathbf{T}(t^{n+1}) \\ & - [\mathbf{C}_E^*(t^{n+1}; \Delta t) + \Delta\mathbf{C}_E(t^{n+1}; \Delta t)] \mathbf{T}_P(t^{n+1}) )]). \end{aligned}$$

Following the same procedure as before, the error due to the partitioning is

$$\begin{aligned} \text{ULTE}^{n+1}(\Delta t) = & \alpha \mathbf{S}^{-1} [ -\Delta\mathbf{C}(t^{n+1}; \Delta t) \mathbf{T}(t^n + \alpha\Delta t) \\ & + \mathbf{C}_E \dot{\mathbf{T}}(t^n + \alpha\Delta t) \Delta t \\ & + (\alpha\Delta\mathbf{C}(t^{n+1}; \Delta t) - \Delta\mathbf{C}_I(t^{n+1}; \Delta t)) \Delta t ] + O(\Delta t^2), \end{aligned}$$

where  $\Delta\mathbf{C}(t^{n+1}; \Delta t) = \Delta\mathbf{C}_I(t^{n+1}; \Delta t) + \Delta\mathbf{C}_E(t^{n+1}; \Delta t)$ , and the order of the error depends on the order of the error introduced by the elements of matrix  $\mathbf{C}(t^{n+1}; \Delta t)$  held by  $\Delta\mathbf{C}(t^{n+1}; \Delta t)$ .

The coefficients of  $\Delta\mathbf{C}(t^{n+1}; \Delta t)$  depend on the position of the mass flow controlling device. If the mass flow controlling device is modeled in the external simulator, and the mass flow is an algebraic function of the subsystem's state which is simulated in the base simulator, then  $\Delta\mathbf{C}(t^{n+1}; \Delta t)$  is

$$\Delta\mathbf{C}(t^{n+1}; \Delta t) = \begin{pmatrix} [\dot{m}(T_1^n) - \dot{m}(T_1^{n+1})] c_p & -[\dot{m}(T_1^n) - \dot{m}(T_1^{n+1})] c_p \\ -[\dot{m}(T_1^n) - \dot{m}(T_1^{n+1})] c_p & [\dot{m}(T_1^n) - \dot{m}(T_1^{n+1})] c_p \end{pmatrix},$$

where  $\dot{m}(\cdot)$  defines the control function for the mass flow. e.g., for a proportional controller,  $\dot{m}(T) = K_p(T_{\text{set}} - T)$ , where  $K_p$  is a proportional gain and  $T_{\text{set}}$  is the temperature set point,  $\Delta\mathbf{C}(t^{n+1}; \Delta t)$  is

$$\Delta\mathbf{C}(t^{n+1}; \Delta t) = \begin{pmatrix} K_p(T_1^{n+1} - T_1^n) c_p & -K_p(T_1^{n+1} - T_1^n) c_p \\ -K_p(T_1^{n+1} - T_1^n) c_p & K_p(T_1^{n+1} - T_1^n) c_p \end{pmatrix}, \quad (4.29)$$

and  $K_p T_{\text{set}}$  is added to  $\mathbf{g}(t)$ . Substituting (4.12) into (4.29) yields

$$\begin{aligned} \Delta\mathbf{C}(t^{n+1}; \Delta t) = & \begin{pmatrix} K_p \Delta t \dot{T}(t^n + \alpha\Delta t) c_p & -K_p \Delta t \dot{T}(t^n + \alpha\Delta t) c_p \\ -K_p \Delta t \dot{T}(t^n + \alpha\Delta t) c_p & K_p \Delta t \dot{T}(t^n + \alpha\Delta t) c_p \end{pmatrix} \\ & + O(\Delta t^2) \\ = & O(\Delta t). \end{aligned}$$

Therefore, the error introduced by the partitioning, in this case, is of order one.

If the mass flow is a function of the state simulated in the base simulator, at the time  $t^{n+1}$  the external simulator will use the value of the mass flow  $\dot{m}(T_1^n)$ , while the base simulator will calculate and use the value of the mass flow  $\dot{m}(T_1^{n+1})$ , hence mass is not conserved.

A similar problem rises also for algebraically modeled humidification and dehumidification processes in a partitioned system. Both humidification and dehumidification should be simulated in the external simulator so that mass is conserved.

To conclude, the order of the error stays unchanged and is  $O(\Delta t)$  and therefore the corresponding mutated numerical scheme is consistent. However, the absolute value of the error varies for different time-state communication schemes.

### 4.7.3 Absolute stability

The second characteristic polynomial (4.7) is used to determine the absolute stability of numerical methods. However, due to the partitioning of the right hand side of the (4.4), the second characteristic polynomial can not be used to determine the absolute stability of the mutated method.

The absolute stability of the mutated numerical methods (4.8) (used with the sequential staggered coupling) is analyzed by using Routh-Hurwitz stability criterion as reported in [Park 1980] and [Tseng et al. 1995].

The stability of the partitioned numerical method (4.19) can be examined by seeking nontrivial solutions for  $\mathbf{g}(t^n) = 0, n = 1, 2, \dots, N$  and  $\mathbf{T}_P^{n+1} = \mathbf{T}^n$  in the form  $\mathbf{T}^{n+1} = \mathbf{A}\mathbf{T}^n$ , or

$$\mathbf{T}^{n+1} = \gamma \mathbf{T}^n, \quad (4.30)$$

where  $\gamma$  represents one of the eigenvalues  $\{\gamma_i \in \mathbb{C} \mid i = 1, 2, \dots, m\}$  of  $\mathbf{A}$ .  $\gamma$  is also known as the solution amplification factor. The system is stable when  $\max |\gamma_i| \leq 1$  for all  $i \in \{1, \dots, m\}$ . To analyze the Routh-Hurwitz criterion,  $\gamma_i$  is replaced by

$$\gamma_i = \frac{1 + z_i}{1 - z_i}, \quad (4.31)$$

where  $z_i$  represents one of  $\{z_i \in \mathbb{C} \mid i = 1, 2, \dots, m\}$ . (4.31) maps the unit circle defined by  $|\gamma_i| \leq 1$  into the negative real half-plane  $\text{Re}(z) \leq 0$ . Substituting (4.30) and (4.31) in (4.19) the following system is obtained:

$$\mathbf{J}(z)\mathbf{T}^n = 0, \quad (4.32)$$

where

$$\mathbf{J}(z) = [2\mathbf{S} + \alpha\Delta t(\mathbf{C}_I - \mathbf{C}_E) - (1 - \alpha)\Delta t\mathbf{C}]z + \Delta t\mathbf{C}.$$

To obtain nontrivial solution of (4.32),  $\det\mathbf{J}(z)$  needs to be zero.

If the following replacements are introduced:

$$\tau_1 = \dot{m}c_p + K_1; \quad \tau_2 = \dot{m}c_p + K_2; \quad \mu = \dot{m}c_p; \quad \zeta_1 = C_1; \quad \zeta_2 = C_2;$$

for implicit-implicit partitioning (4.23), (4.32) yields

$$\det \begin{bmatrix} [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1]z + \Delta t\tau_1 & (\Delta t\mu - 2\alpha\Delta t\mu)z - \Delta t\mu \\ \Delta t\mu z - \Delta t\mu & [2\zeta_2 + (2\alpha - 1)\Delta t\tau_2]z + \Delta t\tau_2 \end{bmatrix} = 0$$

The above equation yields:

$$a_0 z^2 + a_1 z + a_2 = 0, \quad (4.33)$$

where

$$\begin{aligned} a_0 &= [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1][2\zeta_2 + (2\alpha - 1)\Delta t\tau_2] + (2\alpha - 1)\Delta t^2\mu^2, \\ a_1 &= [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1]\Delta t\tau_2 + [2\zeta_2 + (2\alpha - 1)\Delta t\tau_2]\Delta t\tau_1 \\ &\quad + 2(1 - \alpha)\Delta t^2\mu^2, \text{ and} \\ a_2 &= \Delta t^2\tau_1\tau_2 - \Delta t^2\mu^2. \end{aligned}$$

Routh-Hurwitz criterion [Gantamacher 1974] defines the signs for the coefficients:  $a_0$ ,  $a_1$ , and  $a_2$ , in order to have all the roots,  $z$ , of the Equation (4.33) in the left half plane  $\operatorname{Re}(z) \leq 0$ , which is the mapped stability criterion. So, all the roots of the polynomial (4.33) have negative real parts iff the inequalities:

$$a_0 > 0; a_1 \geq 0; a_2 \geq 0$$

hold.

If  $\alpha \geq 1/2$  then both  $a_0$  and  $a_1$  are positive. A criterion:  $\tau_1\tau_2 \geq \mu^2$  is additionally required in order to satisfy the third inequality. It is obvious that this is satisfied, as

$$(\dot{m}c_p + K_1)(\dot{m}c_p + K_2) \geq (\dot{m}c_p)^2$$

always holds. Therefore, the co-simulation is unconditionally stable if both coupled simulators employ time weighting factor greater than or equal 1/2.

If the time weighting factor is less than 1/2, the co-simulation is conditionally stable. The stability criteria is satisfied if both inequalities  $a_0 > 0$ , and  $a_1 \geq 0$  are satisfied. These inequalities are translated into:

$$4\zeta_1\zeta_2 + (1 - 2\alpha)^2\Delta t^2\tau_1\tau_2 > (1 - 2\alpha)(2\Delta t\zeta_1\tau_2 + 2\Delta t\zeta_2\tau_1 + \Delta t^2\mu^2), \quad (4.34)$$

and

$$2\Delta t\zeta_1\tau_2 + 2\Delta t\zeta_2\tau_1 + 2(1 - \alpha)\Delta t^2\mu^2 \geq 2(1 - 2\alpha)\Delta t^2\tau_1\tau_2. \quad (4.35)$$

By deriving inequality for  $2\Delta t\zeta_1\tau_2 + 2\Delta t\zeta_2\tau_1$  from (4.35) and substituting it to (4.34), the stability criterion expressed as a restriction for the time step size is:

$$\Delta t_0^2 < \frac{4\zeta_1\zeta_2}{(1 - 2\alpha)^2(\tau_1\tau_2 - \mu^2)}. \quad (4.36)$$

It should be mentioned, that the criterion for the time step is only necessary to satisfy the inequalities for  $a_0$ ,  $a_1$  and  $a_2$ . The sufficient criteria are much cumbersome to obtain and will not be reported here.

If the error introduced by the partitioning of the algebraic system of equations is taken into account, the stability criterion is too complex to obtain. The calculation of the required criterion is cumbersome and will not be stated here.

For implicit-explicit partitioning (4.24) the stability criterion differs. The coefficient  $a_2$  stays unchanged, while even for the time weighting factor  $\alpha \geq 1/2$  the inequalities:

$$\begin{aligned} a_0 &= [\zeta_1 + (2\alpha - 1)\Delta t\tau_1][2\zeta_2 + (2\alpha - 1)\Delta t\tau_2] + (2\alpha - 1)\Delta t^2\mu^2 \\ &\quad - 2\alpha\Delta t\mu[2\zeta_2 + (2\alpha - 1)\Delta t\tau_2] > 0, \end{aligned}$$

and

$$\begin{aligned} a_1 &= [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1]\Delta t\tau_2 + [2\zeta_2 + (2\alpha - 1)\Delta t\tau_2]\Delta t\tau_1 \\ &\quad + 2(1 - \alpha)\Delta t^2\mu^2 - 2\alpha\Delta t^2\mu\tau_2 \geq 0, \end{aligned}$$

do not hold for all values of coupling term  $\dot{m}c_p$ . Due to the partitioning of the diagonal term of the  $C$  matrix, implicit-explicit partitioning (inter-domain system decomposition) has a more strict stability criterion.

It can be shown that for  $\alpha = 1/2$ , the critical time step equals  $\Delta t_0 = 2/\mu$ . More general expressions for the critical time step for the inter-domain decomposition is can be obtained by following a cumbersome calculation path and it doesn't result in elegant formulae. As it could have been expected, the critical time step is smaller than the critical time step calculated for the implicit-implicit partitioning (intra-domain decomposition approach).

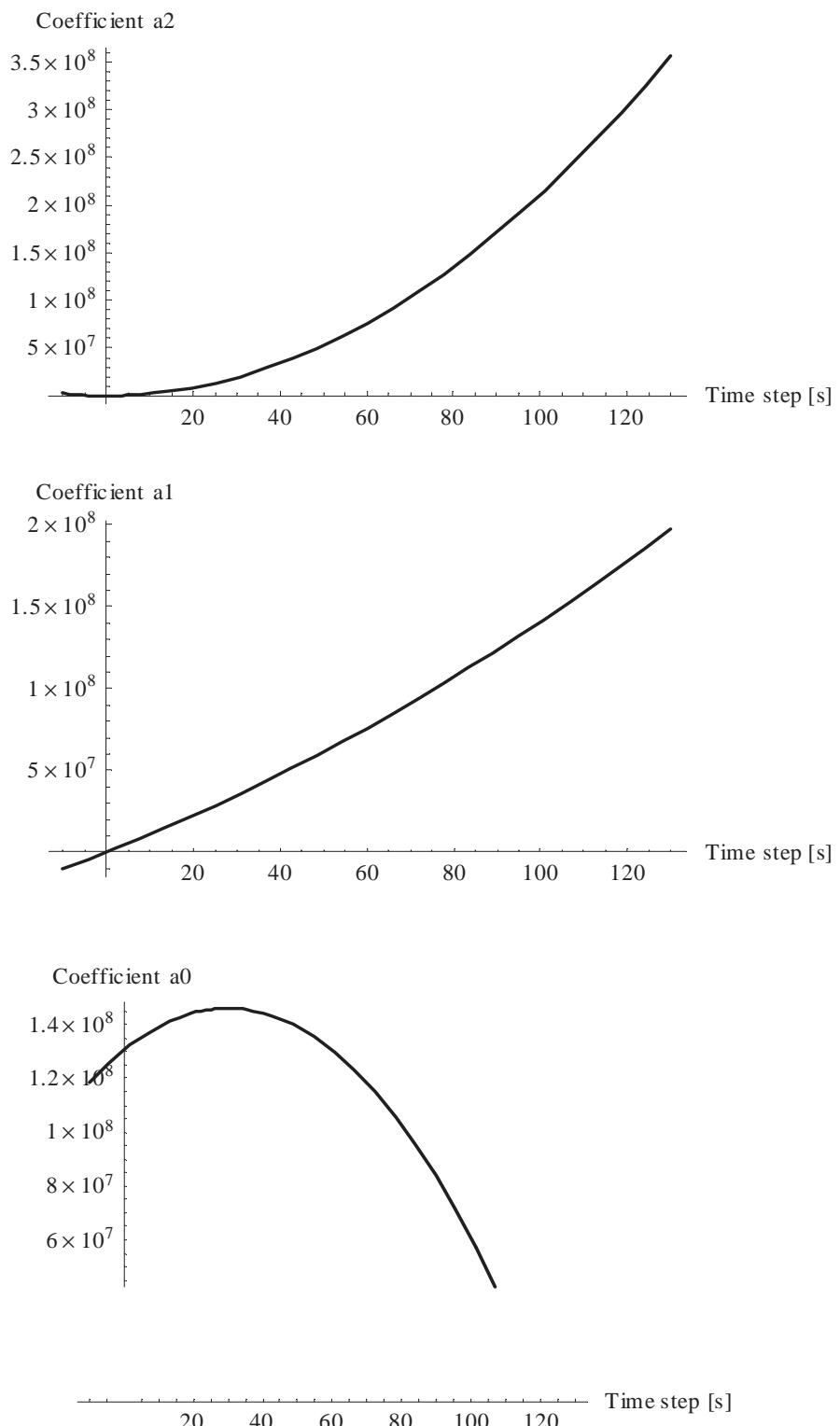
If the subsystems are linked by a control loop, the absolute stability criterion is influenced by the control parameters. The investigation even for a simple controller is difficult as the vector  $\mathbf{g}(t)$  is not zeroed and thus the requirements for the  $\det \mathbf{J} = 0$  is not sufficient. However, if the set point temperature is zeroed, a simple investigation is still possible as the actuated heat flux can be calculated in the form:  $\dot{Q}_1 = K_p \cdot T_2$ . The coefficients of the Routh-Hurwitz polynomial are thus:

$$\begin{aligned} a_0 &= [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1][2\zeta_2 + (2\alpha - 1)\Delta t\tau_2] + (2\alpha - 1)\Delta t^2\mu K_p \\ a_1 &= [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1]\Delta t\tau_2 \\ &\quad + [2\zeta_2 + (2\alpha - 1)\Delta t\tau_2]\Delta t\tau_1 + 2(1 - \alpha)\Delta t^2\mu^2 + 2\Delta t^2\mu K_p \\ a_2 &= \Delta t^2\tau_1\tau_2 - \Delta t^2\mu(\mu + K_p). \end{aligned}$$

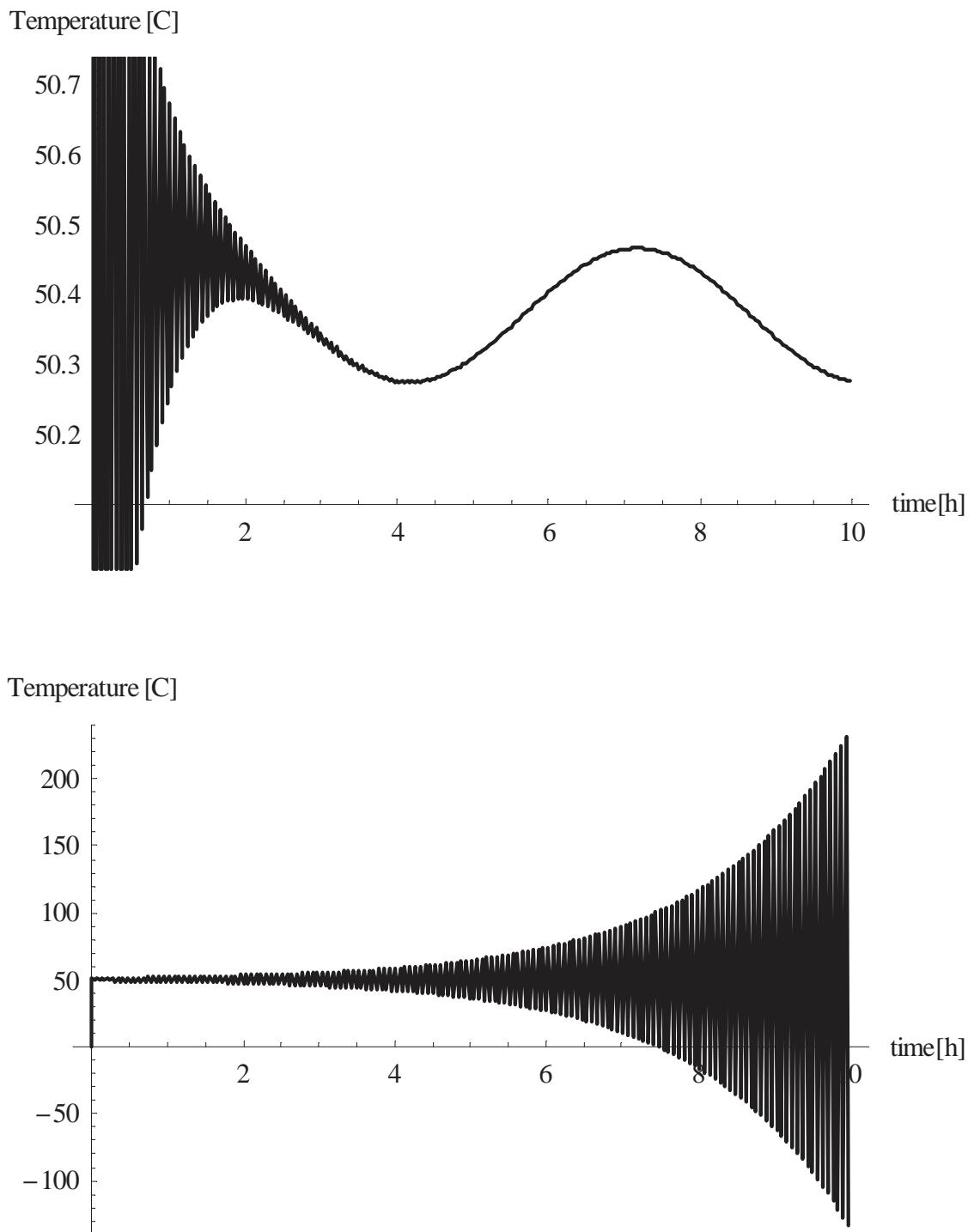
Similarly, if the temperature of the first subsystem is sensed and the actuated heat flux injected into the second subsystem, the coefficients of Routh-Hurwitz polynomial are:

$$\begin{aligned} a_0 &= [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1][2\zeta_2 + (2\alpha - 1)\Delta t\tau_2] + (2\alpha - 1)\Delta t^2\mu K_p \\ a_1 &= [2\zeta_1 + (2\alpha - 1)\Delta t\tau_1]\Delta t\tau_2 + [2\zeta_2 + (2\alpha - 1)\Delta t\tau_2]\Delta t\tau_1 \\ &\quad + 2(1 - \alpha)\Delta t^2\mu^2 + 2(1 - \alpha)\Delta t^2\mu K_p \\ a_2 &= \Delta t^2\tau_1\tau_2 - \Delta t^2\mu(\mu + K_p). \end{aligned}$$

As it can be noticed, the analysis becomes more cumbersome with adding complexity. Even, for a simple case with the control feedback loop, the stability criterion is difficult to express easily.



**Fig. 4.3** — Dependency of coefficients of Routh-Hurwitz polynomial on time step size for a specific numerical example.



**Fig. 4.4** — Illustration of introduced instabilities using a value for time step greater than the critical.

To illustrate the stability criterion in terms of  $\Delta t_0$ , the Routh-Hurwitz criterion on derived polynomials are exercised on the example from the Figure 4.2 applying the simple proportional controller. The system is modeled and simulated in Mathematica 6. Diagrams in Figure 4.3 show relationships between coefficients of Routh-Hurwitz polynomial to  $\Delta t$  for a particular set of parameters. For different sets of the parameters the curves can look differently. The value of time step where the curves cross the positive  $x$ -axes defines its critical value above which the instabilities will be introduced in the co-simulation. It can be shown that for some combinations of the inputs and parameters none of the curves ever intersects with positive  $x$ -axis and thus prove that in such cases the co-simulation stability is unconditional.

For the particular example, in order to satisfy stability criterion,  $\Delta t$  needs to be less than 121 s. To illustrate this, two co-simulations with two slightly different time step sizes are executed to show the introduced instabilities. The results are shown in Figure 4.4. Upper graph shows the results obtained with  $\Delta t = 120$  s, and the lower graph shows the results obtained with  $\Delta t = 122$  s.

#### 4.7.4 Measures to improve accuracy of co-simulation solution

##### Comparison to an analytical solution

To illustrate the behavior of the error introduced by the partitioning the analytic solution of the system (4.16):

$$\mathbf{T} = \mathbf{T}_a e^{(t-a)\mathbf{S}^{-1}\mathbf{C}}$$

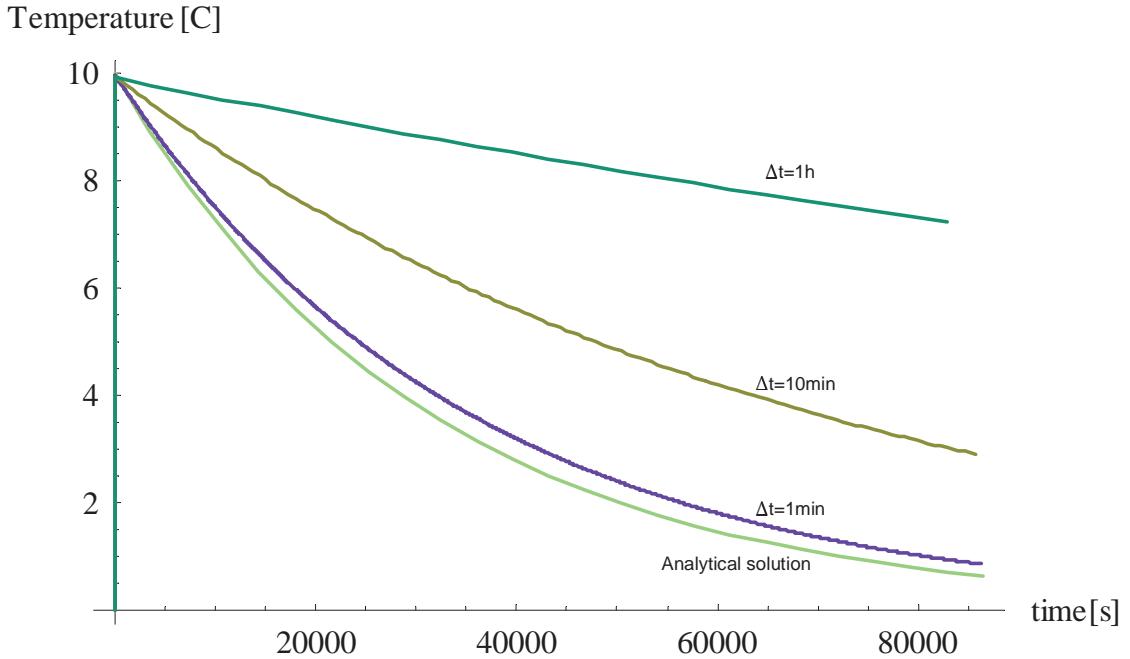
is compared to the results obtained by co-simulation (4.19) executed with several different coupling time steps. The results are shown in Figure 4.5. With increasing the coupling time step, the accuracy of co-simulation results is degraded, as expected.

To enhance accuracy, several measures can be taken:

- Co-simulation can be executed with small time steps. However, so far only heuristic approach for determination of the discretization step size has been used.
- Instead of the zero-order prediction of the coupling data, the more accurate prediction can be used.
- Instead of loose coupling, co-simulation can implement strong inter-simulator coupling.
- Co-simulation can run with variable time step size. The discretization step could be calculated based on the local truncation error estimates.

##### Strong coupling strategy

The strong coupling enhances the accuracy of co-simulation compared to the loose coupling with the same discretization step size.



**Fig. 4.5** — Comparison of free-response system's analytical solution to results of co-simulation with different coupling time steps.

In order to perform iterations successfully, the set of the coupled equations need to converge. When using inter-simulator iterations (4.19) becomes

$$[\mathbf{S} + \alpha \Delta t \mathbf{C}_I] \mathbf{T}_{k+1}^{n+1} = \Delta t [\alpha \mathbf{g}(t^{n+1}) + (1 - \alpha) \mathbf{g}(t^n)] + [\mathbf{S} - (1 - \alpha) \Delta t \mathbf{C}] \mathbf{T}^n - \alpha \Delta t \mathbf{C}_E \mathbf{T}_k^{n+1},$$

where the index  $k$  denotes the iteration step. The change of  $\mathbf{T}_{k+1}^{n+1}$  with a change with the change of  $\mathbf{T}_k^{n+1}$  can be written in the form:

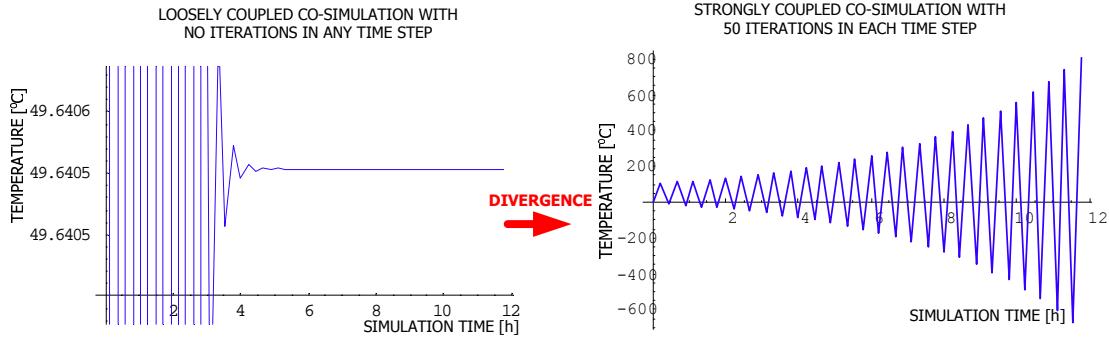
$$\frac{\partial \mathbf{T}_{k+1}^{n+1}}{\partial \mathbf{T}_k^{n+1}} = -(\mathbf{S} + \alpha \Delta t \mathbf{C}_I)^{-1} \alpha \Delta t \mathbf{C}_E. \quad (4.37)$$

To ensure the convergence of the inter-simulator successive substitution iteration process, the absolute value of the differential defined in (4.37) needs to be less than one, i.e., the eigenvalues:  $\{\lambda_i \in \mathbb{C} \mid i = 1, \dots, m\}$  of the matrix on the right hand side need to satisfy:  $|\lambda_i| \leq 1$  for all  $i \in \{1, \dots, m\}$ . Now, the eigenvalues of the matrix from (4.37) will be examined to investigate the convergence of the strong coupling.

Let a subscript  $l$  denote the receiving node in a subsystem simulated in the external simulator. Then for implicit-implicit partitioning  $\lambda_i, i \in \{1, \dots, l-1, l+1, \dots, m\}$  is zero, while

$$\lim_{\Delta t \rightarrow 0} |\lambda_l| = \lim_{\Delta t \rightarrow 0} \left| \frac{(\Delta t \dot{m} c_p)^m}{\prod_{i=1}^m (-C_i - \Delta t(K_i + \dot{m} c_p))} \right| = 0.$$

The same limit,  $\lim_{\Delta t \rightarrow 0} |\lambda_l| = 0$ , can be obtained for the implicit-explicit partitioning, although the expression for the non-zero eigenvalue is more complex. Thus, each partitioned problem converges for sufficiently small  $\Delta t$ .



**Fig. 4.6** — Iteration process that did not converge, even though the loosely-coupled simulation is stable.

To inquire about the conditionality of the convergence of the strong coupling,  $\lim_{\Delta t \rightarrow \infty} |\lambda_l|$  will now be examined. For implicit-implicit partitioning (4.23), the limit of the eigenvalues of the matrix from (4.37) when  $\Delta t \rightarrow \infty$  is

$$\lim_{\Delta t \rightarrow \infty} |\lambda_l| = \left| \frac{(\dot{m}c_p)^m}{\prod_{i=1}^m (\dot{m}c_p + K_i)} \right| \leq 1,$$

which satisfies the iteration convergence criteria. Thus, for this case, the partitioned coupled system of equations will converge for any  $\Delta t$ .

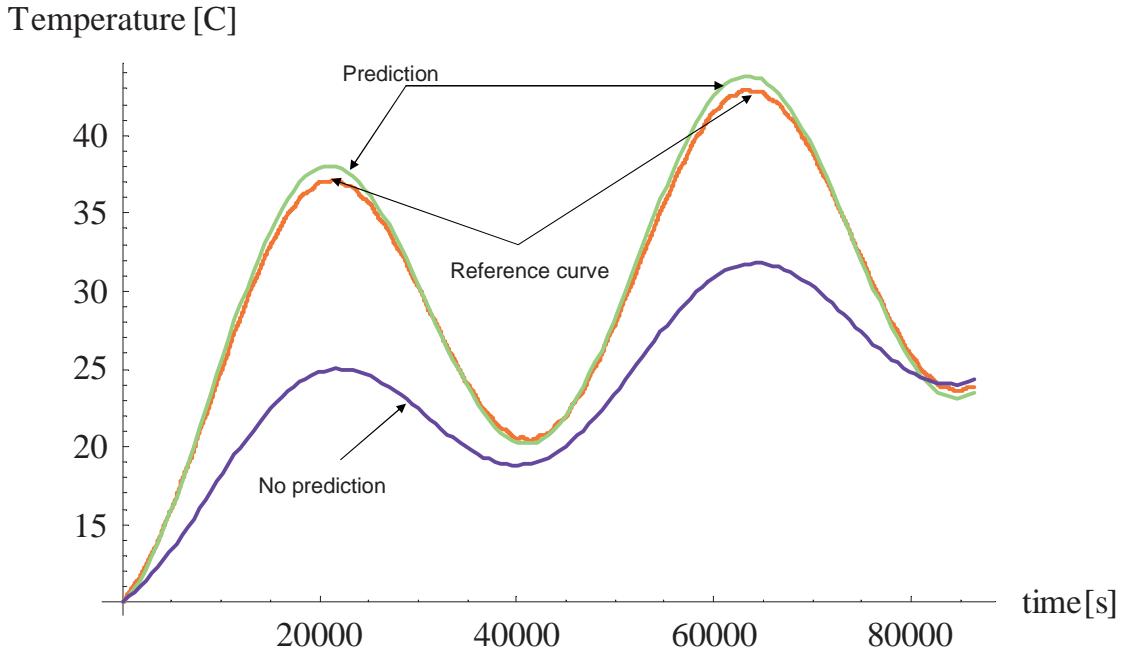
For implicit-explicit partitioning (4.24), to ensure the unconditional inter-simulator convergence, an additional criterion needs to be fulfilled. The  $K_i$  value of the subsystems that receives the heat flux as the coupling data needs to be larger than the value of  $\dot{m}c_p$ . If this condition holds, the convergence is unconditional. For any other case, where the system control is distributed and the sensed coupling data is iterated, the convergence for finite value of  $\Delta t$  depends on the value of control parameters. This dependency defines convergence criteria. It can be shown that for certain set of parameters, the strong coupled co-simulation leads to unstable solutions due to divergence of the iteration process, even though the loose co-simulation shows stable behavior (Figure 4.6).

The convergence can however, always be enforced by decreasing the time step.

## Predictors

The accuracy might be improved by using more accurate predictors for the delayed coupling data instead of the zero-order predictors (values available in the previous time step). For example, the first-order predictor:  $\mathbf{T}_P^{n+1} = \mathbf{T}^n + \Delta t \dot{\mathbf{T}}^n$ , can be estimated by using the known values for the predicted coupling data available in two preceding time steps:  $\mathbf{T}_P^{n+1} \approx 2\mathbf{T}^n - \mathbf{T}^{n-1}$ .

The Figure 4.7 shows comparison between results with and without predictions. The results of co-simulation using first-order predictor is significantly closer to the reference curve than the results of co-simulation using zero-order predictor (value available in the previous time step). The first-order predictor is more accurate for the case when the average value is communicated compared to



**Fig. 4.7** — Comparison between co-simulation results with and without predictions.

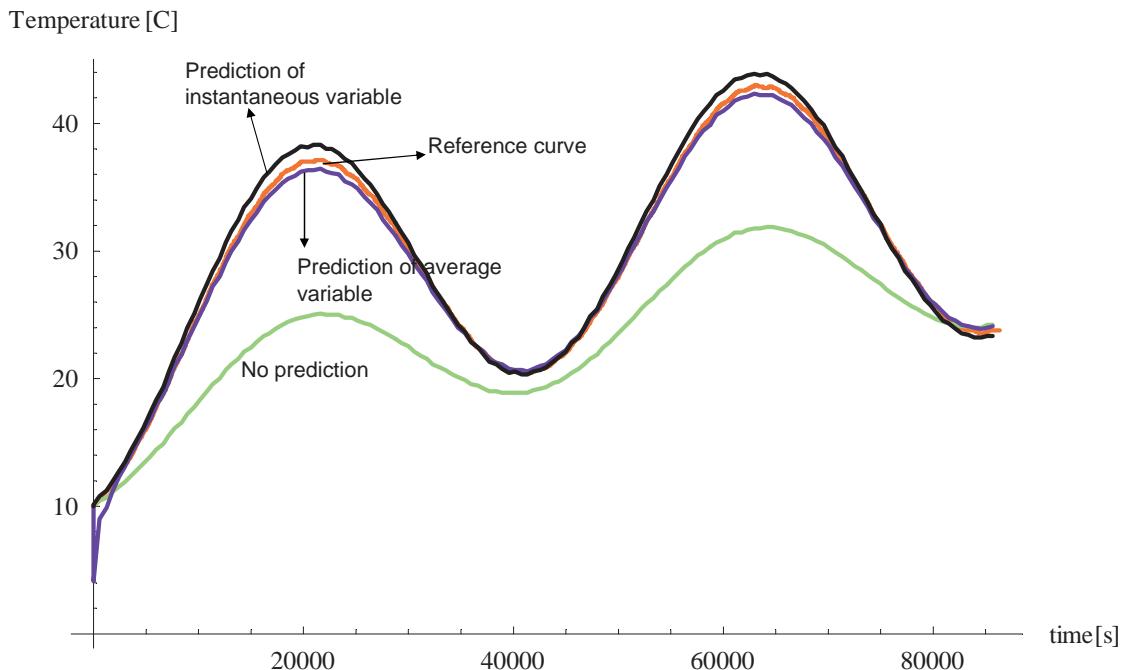
the corresponding instantaneous value. The reason for that lays in the fact that the averaging increases the order of the predictor.

If however, co-simulated subsystems have vastly different time constant and thus one of the subsystems uses a smaller time step, the co-simulation with predictor poses certain restrictions on coupling choices. If the external simulator uses a smaller time step, its input is predicted based on the state of the other subsystem in preceding coupling time step(s). The external simulator can then either use that value directly or interpolate between the preceding and predicted value for the smaller time steps.

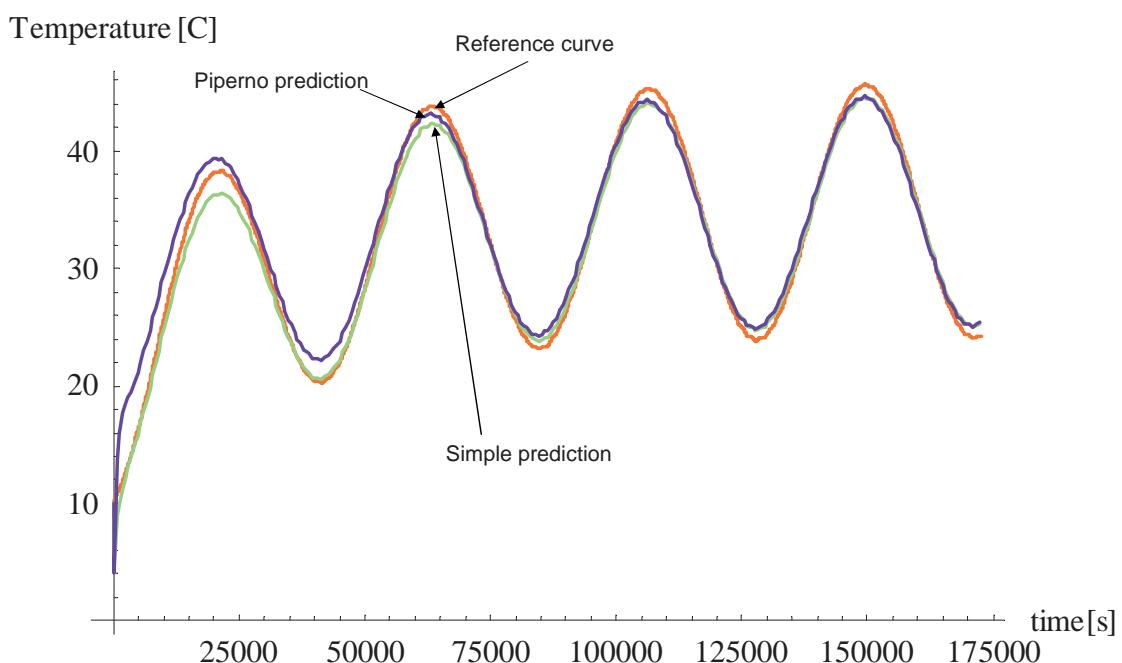
On the other hand, if the base simulator uses smaller time steps, predicting its state in the next coupling time step means predicting its state for the several smaller time steps ahead. Therefore, if the external simulator uses larger time steps the prediction of averaged coupling data (integrated coupling data over the larger time step divided by the same time step) is used (Figure 4.8).

Piperno [1997] compared the performance of several predictors (for fluid /structure coupling) and found out that the most accurate and stable predictor is the one that adapted for the case, studied here, has the form:  $T_P^{n+1} = T^n + \Delta t(1.5\dot{T}^n - 0.5\dot{T}^{n-1})$ , which numerical estimate is  $T_P^{n+1} \approx 2.5T^n - 2T^{n-1} + 0.5T^{n-2}$ . The comparison between the simple first-order predictor and predictor suggested by Piperno [1997] are shown in Figures 4.9 and 4.10. With the smaller time steps the results are almost identical, however, with larger time steps the second-order predictor, suggested by Piperno is more accurate.

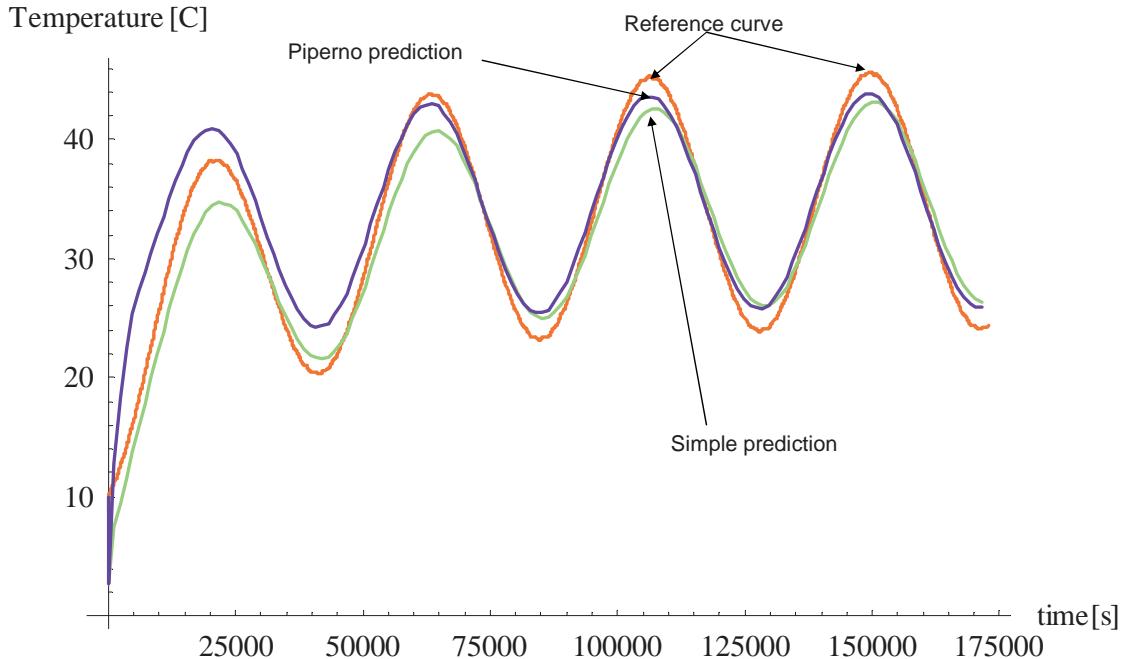
Depending on the order of the predictor, there may be several coupling data that need to be initialized. In the first time step, the values of the coupling data in one or more preceding time steps are assumed to be equal to the initial value of



**Fig. 4.8** — Multiple-time-step co-simulation results using first-order predictor for instantaneous and average variable exchange plotted against the reference curve.



**Fig. 4.9** — Co-simulation results using different predictors plotted against the reference curve.



**Fig. 4.10**—Co-simulation results using different predictors plotted against the reference curve, obtained with the larger time step.

that variable.

The above considers and is restricted to the situations where the change of coupling data per time step is small. However, in more realistic applications the changes of the coupling data may not be smooth. In such cases the predictor may overestimate the future value of the coupling data, which can lead to lowered accuracy and will be discussed in Chapter 6.

### Variable step approach

Enhancing accuracy of co-simulation can be accomplished also by using a variable-time-step approach. As opposed to the fixed-time-step approach, where the time step is assigned by the user and held constant during co-simulation, in the variable-step approach, the time step is varied during run-time. This approach can enhance simulation accuracy beyond the accuracy achieved by using the same initial (but fixed) discretization size in strongly coupled co-simulation approach.

On the one side, some BPS tools already implement some sort of simple variable time stepping procedure, e.g., in EnergyPlus. The time step size is adjusted during run-time based on the limitation of a rate of change of a particular state variable (or variables). The user specifies the desired maximal change of the chosen variable(s). If the change in one time-step calculation is exceeded, the time step is decreased.

On the other side, typical ordinary differential solvers use estimates for the local truncation error (as estimates for the global truncation error are not easy to

calculate), e.g., based on the second derivative, and compare it to the value of acceptable error tolerance. The time step is calculated so that the error estimate is smaller than the tolerance<sup>2</sup>.

However, the employment of the variable time step approach requires a time step controller. The controller can be implemented in any of the coupled simulators, or in an external tool. In either case, both coupled simulators need to have additional features in order to change the simulation time step during the execution time. As this feature is available only in few state of the art tools from the BPS domain, it will not be discussed further.

## 4.8 Discussion and conclusions

The study in this chapter was limited to linear one-step numerical methods. The investigation showed that new numerical methods, obtained by the partitioning of the original numerical methods, are zero-stable and consistent, and thus convergent. The unit local truncation error of the original numerical method is increased. The order of the error due to the partitioning is one, which means that the order of the error of the second-order accurate methods is reduced by the partitioning.

To gain insights into how HVAC system characteristics influence the error introduced by the partitioning, the error study was continued on a two-body system, described by a system of linear first-order differential equations with constant coefficients. It was shown that the error was related to the system's characteristics as follows. The greater the capacity of the subsystem simulated in the external simulator,  $C_{\text{ext}}$ , the smaller the error is. Also, the lower the first derivative  $\dot{T}_{\text{base}}(t^n + \alpha\Delta t)$  is, the lower  $\dot{m}c_p$  is and the smaller  $\Delta t$  the smaller the error is.

Further, an investigation of the error behavior under the influence of time varying coefficients of the matrix  $\mathbf{C}$  showed that the greater the error in the time varying coefficients, introduced by the partitioning in one simulation time step, i.e.,  $\Delta\mathbf{C}(t^{n+1}; \Delta t)$ , and the greater the temperature  $\mathbf{T}(t^n + \alpha\Delta t)$ , the greater the overall unit local truncation error was. Also, it was shown how the order of the error depends on  $\Delta\mathbf{C}(t^{n+1}; \Delta t)$ .

The stability analysis, using the Routh-Hurwitz stability criterion, was performed on the two-body system. The analysis resulted in a cumbersome calculation procedure already when small complexities were introduced to the problem. For the two-body system represented by a system of linear first order differential equations with constant coefficients, the partitioned numerical method is unconditionally stable for the time weighting factor  $\alpha \geq 1/2$ .

To enhance accuracy of the loose coupling, several strategies were suggested, e.g., (i) the use of the strong coupling, (ii) the use of predictors for the coupling data and (iii) the use of a variable time step. The convergence of the iteration

---

<sup>2</sup>Although the optimal step size can be estimated by minimizing the work required for numerical integration over prescribed interval [Gear 1971], generally these criteria are too complicated for practical applications.

process in the strong coupling was tested and it was shown that in some cases the iteration process was unconditionally stable, while in others the convergence could be enforced by using smaller time step. Further, several prediction functions were tested. With assumptions that the changes in the coupling data are smooth, the use of a first-order predictor showed good results. The varying time step procedure seem to be complex, implementation-wise, to be used with BPS state of the art tools.



# 5

## Co-simulation - Prototypes

CHAPTER 4 has given a simplified analysis of what types of trends can be expected in co-simulation. The results of the chapter can be considered as a qualitative analysis of the co-simulation problem, but the results do not have a direct quantitative use for a general class of large scale problems. To enable further investigation of various implementations of the co-simulation approach, a working co-simulation prototype is developed and tested.

To assure the quality of the prototype, the prototype needs to be based on well-validated simulation tools. Also, to minimize the amount of work in developing new models, the prototype needs to be based on the BPS domain specific tools, which already provide domain specific models. Thus, three state of the art BPS tools have been chosen to be included in the prototype: ESP-r, EnergyPlus, and TRNSYS.

The purpose of this chapter is to communicate in detail changes in the code of the chosen BPS tools and to give detailed insights into the prototyped implementation of co-simulation.

The approach undertaken in the thesis is to develop interface components for co-simulation within each simulator, while respecting the developer's style, that will be used to communicate coupling data to other simulators.

The implementation, discussed in this thesis, considers (but is not limited to) coupling on a single platform, executed as multiple process simulation, using distributed integration and bi-directional data flow. The prototypes employ the conservative time management with both the external and the internal synchronization mechanisms. The developed prototypes use named pipes (UNIX application) and shared memory (Windows applications). In the further text, only the continuous external simulator run mechanism is discussed.

### 5.1 Simulation tools used in prototypes

The prototype developed in this thesis is based on coupling the specific building and the HVAC system models from the three major domain-specific simulation environments (ESP-r, TRNSYS and EnergyPlus). What follows is a brief discussion of the calculation procedures implemented in the three tools used in the prototype.

**ESP-r** is a general purpose integrated simulation environment which has been under development for more than thirty years.

Building and system parts are represented by a discrete nodal scheme. A building load module generates and solves a system of equations for the heat transfer within building structure<sup>1</sup>. Clarke [2001] elaborates in detail the ESP-r calculation procedure.

ESP-r uses nodal network model to represent airflow within the building. The thermal and the airflow domains are fully integrated.

For every HVAC system component, energy and mass balance systems of equations are then derived. Each system is solved using the simultaneous modular solution technique with iterations between the domains (strong coupling).

Further, the load and the HVAC system of equations are coupled using the sequential staggered procedure. The time steps for load and system simulations can differ.

An HVAC component in ESP-r is modeled in a dynamic fashion, by encountering for the component's heat capacity. However, a limited number of the available system components limit the scope of explicit HVAC system configurations that can be modeled and simulated.

Most of the available system controllers enable only modeling of a local process control. In ESP-r, system controllers are modeled explicitly by defining:

- sensor location and sensed variable,
- actuator location and actuated variable, and
- control algorithm with its settings.

Control variable(s) for each component are predetermined by the model algorithm itself. It does not always have a real world complement. Different sensor/actuator/control algorithm combinations are possible. There are various (although limited number of) controllers in the ESP-r library.

**EnergyPlus** is a modular, structured code based on the most popular features and capabilities of BLAST and DOE-2.1E. EnergyPlus is a simulation engine for which a number of graphical interfaces are available or under development [<http://www.eere.energy.gov>].

Detailed explanation of all the EnergyPlus modules is given in EnergyPlus documentation [<http://www.eere.energy.gov>]. EnergyPlus includes multizone air flow, thermal comfort, water use, natural ventilation, photovoltaic systems, etc.

---

<sup>1</sup>It uses specially adapted solution technique where the overall matrix is partitioned into a number of discrete sub-matrices so that solution is achieved in the lowest number of computational steps. Each partitioned matrix can then be processed as far as possible by a simultaneous (direct) solution technique with the inter-component information brought together to permit the global solution stream to continue.

Building loads are calculated by a heat balance engine at a user-specified time step. The zone and the HVAC system integration is based on a method called “lagging with zone capacitance” [LBNL 2006]. The method uses information available in previous time steps to predict HVAC system response and update the zone temperature at the current time. To permit increasing the time step as much as possible while retaining stability, the method employs several measures, such as: (i) introducing the zone air capacity into the heat balance, (ii) using higher order expressions for the zone temperature first derivative, and (iii) using an adaptive time step, calculated during run-time, for HVAC system simulation and zone temperature updates.

The HVAC modules are solved using the independent modular solution technique. The HVAC system representation is based on fluid loops (air and water loops) and control managers. All system components are attached to the loops. The loops define the movement of mass and energy within the system. The difference is made between: air loop, plant loop and condenser loop. The loops are indirectly connected (through the coil model for example). Each loop has two logical simulation blocks: a supply and a demand side. The demand side places a load to the supply side. These sides are simulated independently, while the convergence between their interaction points is checked and if necessary the iteration procedure is employed. The overall iteration is required to ensure that the results among the loops are balanced. Most of the HVAC system components are modeled in a steady state fashion.

In EnergyPlus, the control implementation is motivated by the solution algorithm and it is not modeled based on actual sensors and actuators. The control is modeled using a two-level hierarchy: controllers and set point managers [LBNL 2006]. A set point manager can sense any node variables and use them to determine the set point of any other node variables. The representation of local controllers is “ideal” and based on either the (known) load to be satisfied or the set point values at their outlet to be reached, which can be established by the set point managers and varied in simulation time.

Although EnergyPlus can be used for modeling a wide variety of system configurations, the ideal local control representation and the steady state representation of HVAC component models limit the scope of the tool usage. Further, the loop based system modeling mechanism does not allow for some system configurations, as will be discussed in Chapter 7.

Also, even though the number of HVAC component models in EnergyPlus is growing, there is no graphical interface that facilitates the building of a component-based HVAC system model.

**TRNSYS** is a transient system simulation program with a modular structure that was designed to solve complex energy system problems by breaking the problem down into a series of smaller components - types. Types may be as simple as a single HVAC component or as complex as a multi-zone building

model. The simulation engine solves the system of algebraic and differential equations, associated with all HVAC-system components, building envelope thermal balance and the air network, using independent modular solution technique.

The control in TRNSYS is defined in an explicit manner, as in ESP-r, apart from a few component models that implement “ideal” local hard-wired control.

The multi-zone building model, is not as advanced as the building model of EnergyPlus or ESP-r, e.g., the view factors between surfaces are not evaluated, there is no option for light dimming control modeling, etc. However, the TRNSYS HVAC library is probably the richest among all state of the art BPS tools. It includes components for solar thermal and photovoltaic systems, low energy HVAC systems, renewable energy systems, co-generation, fuel cells, etc. Also, an equation adding mechanism facilitates to a certain extent rapid model prototyping.

ESP-r and EnergyPlus have been chosen for their advanced building model and TRNSYS for its extensive library of HVAC components. The EnergyPlus building model is coupled to TRNSYS HVAC system model(s). This prototype combines the advanced building modeling capabilities in EnergyPlus with state-of-the-art (graphical) HVAC system modeling capabilities of TRNSYS. The ESP-r HVAC system modeling capabilities are extended by coupling this simulator to a stand alone program - EARTH - for modeling and simulation of an earth-to-air heat exchanger.

### 5.1.1 Control modeling in co-simulation

Co-simulation creates restrictions upon the control modeling, as not all the variables are readily available in each distributed model. If the control model is distributed, i.e., actuator and sensor are represented in different simulators, the control variable, i.e., sensed value or actuated value should be exchanged between the simulators. The concept of the load-driven control strategy modeled across multiple simulators is implicitly excluded since the information about the required load is not available.

Depending on how a system is decomposed, in co-simulation two distributed control model configurations are possible: (i) the control algorithm and actuator are modeled and simulated in different simulators, and (ii) the control algorithm and actuator are modeled and simulated in the same simulator. Only the latter enables the use of an “ideal” control model. The sensed and actuated variables do not have to have real world counterparts, as long as the exchanged variables can be interpreted by both of the coupled simulators.

### 5.1.2 Use of variable time stepping in co-simulation

In most of the BPS tools, the user selects the simulation time step prior to the simulation run. Based on the selected time steps of the coupled simulators, the

coupling frequency can be determined (the coupling time step can not be smaller than the largest chosen simulation time step of the coupled simulators). Hence, if any of the simulators reduces its simulation time step during run time (e.g., EnergyPlus), the coupled variables will be kept constant during the reduced time steps calculations. This may lead to stability problems, due to the shorter time scales of the system and the zone temperatures responses compared to the coupling frequency, which is limited to the maximal predefined simulation time step in the coupled simulators. Therefore, even though some simulators employ variable time stepping, the coupling time step need to be assigned considering the time scale of these interactions.

## 5.2 Communication mechanisms used in prototypes

**Shared memory -** For programming the use of a shared memory, a free Fortran library *ShareBufferWin32* has been used. A portion of the shared memory is reserved for co-simulation control and (internal) synchronization. Boolean values are used to indicate the availability of new coupling data for communication in both directions. To synchronize, a simulator that requires new coupling data will wait until the coupling data becomes available.

Multiple sets of fifty shared memory elements are associated with different communication sequences<sup>2</sup> between coupled simulators (illustrated in Figure 5.1). A specific integer value is thus required for every interface component that tells through which communication sequences the relevant coupling data is communicated.

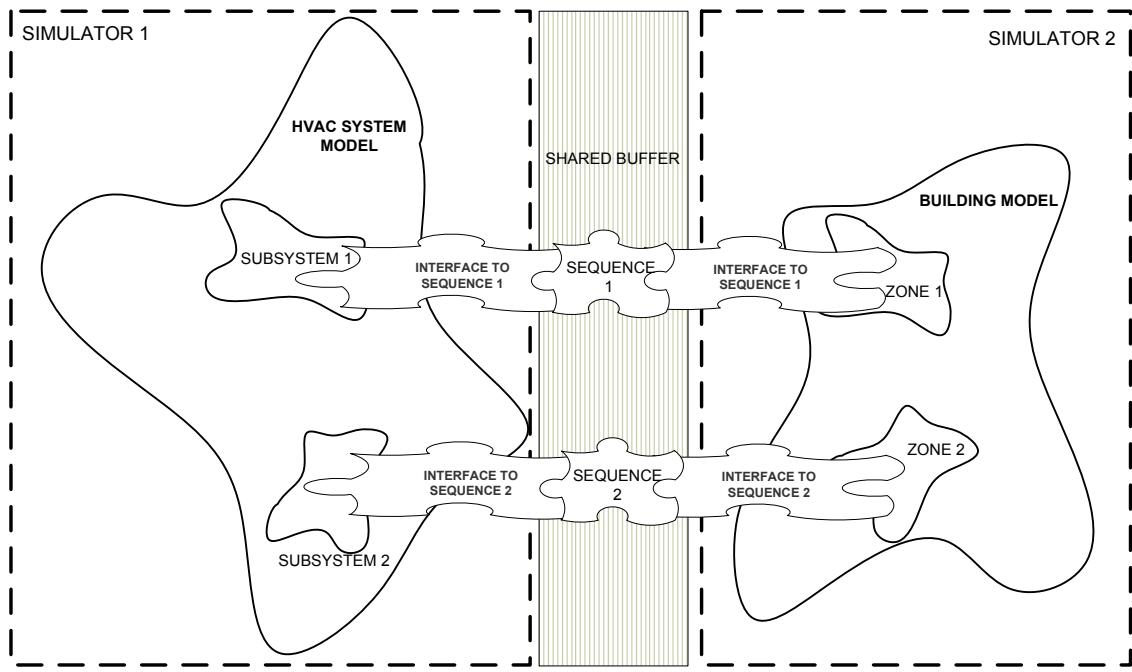
The shared memory can be initialized externally by an additional program or by any of the coupled simulators. The initialization should be done only once.

**Named pipes -** A named pipe concept (also known as FIFO - First In First Out - according to its behavior) is an extension to the traditional pipe concept on Unix and Unix-like systems<sup>3</sup>. A traditional pipe is “unnamed” because it exists anonymously and persists only for as long as the process is running. A named pipe is system-persistent and exists beyond the life of the process and must be “unlinked” or deleted once it is no longer being used. Coupled simulators attach to the named pipe via C routines to perform inter-process communication. They access the pipeline by its name as if it was a file. The data in the pipe is read sequentially, and the order of the coupling data in which they are written/read to/from the named pipe is predefined, e.g., first temperature, then mass flow, then humidity ration, etc. The pipes are used in the blocking mode for the external synchronization, e.g., if a process reads from a named pipe and if the pipe does not have data in it, the reading

---

<sup>2</sup>There can be more than one communication sequence between the coupled simulators as for example a building (modeled in one simulator) can be coupled to more than one system (modeled in another simulator).

<sup>3</sup>The concept is also found in Microsoft Windows, although the semantics differ substantially.



**Fig. 5.1** — Multiple coupling sequences through the shared memory.

process will be blocked. Similarly if a process tries to write to a named pipe that has no reader, the writing process gets blocked until another process opens the named pipe for reading.

A procedure for performing a co-simulation is as follows. Each coupled subsystem is firstly modeled in the corresponding tool. The subsystems' models should be linked to the interface components for co-simulation. The user interface to these added components is provided in each tool. Each simulation commences separately but does not evolve unless the other simulation starts as well. The simulations are synchronized in the run-time. Finally, after both coupled simulators finalize their calculation procedures, the results can be analyzed using the output processing facility of each tool.

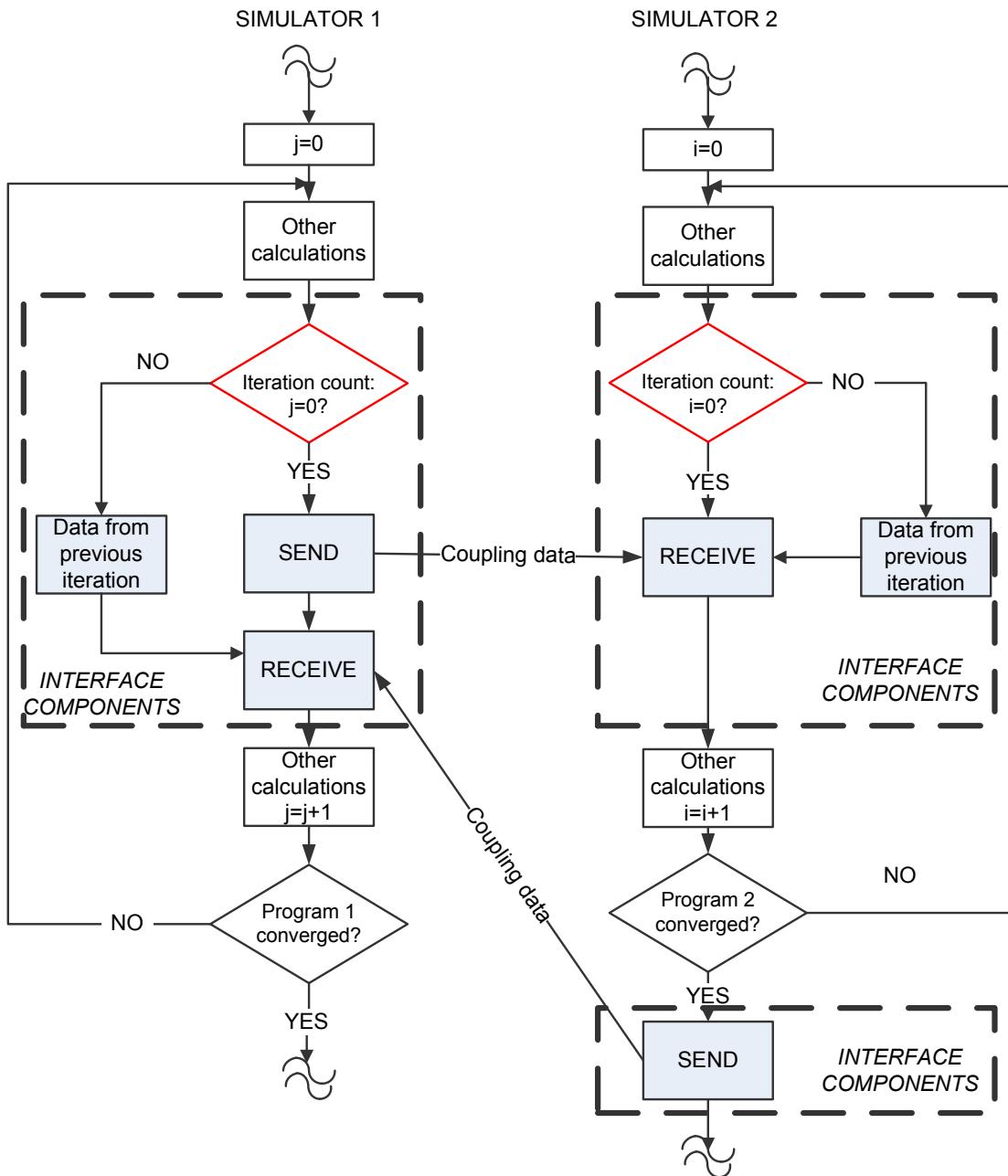
## 5.3 EnergyPlus/TRNSYS prototype

The prototype enables both loose and strong coupling between the simulators.

### 5.3.1 Loose coupling implementation

The Figure 5.2 shows the implementation flow-chart for the loose coupling.

Coupling data is communicated only in the first iteration for the current, time step in both simulators (if  $i = 0$  for TRNSYS or  $j = 0$  for EnergyPlus). EnergyPlus sends the known (e.g., available in the previous time step) coupling data to TRNSYS. Based on this data, TRNSYS performs the full time step simulation until all components converge. Then, TRNSYS sends the converged coupling



**Fig. 5.2** — Flow-chart of loosely-coupled implementation. In the co-simulation prototype, EnergyPlus takes the role of the simulator of which information flow is represented on the left side of the figure. The information flow of the second simulator corresponds to the TRNSYS role.

data to EnergyPlus. Next, based on the received data EnergyPlus performs its time step calculation involving iterations, during which the data that has been received from TRNSYS is held constant. To understand how the data communicated from TRNSYS is incorporated in EnergyPlus' zone temperature correction formula, the adjustments of the energy balance equation (5.1) at  $n + 1$  time step, used by EnergyPlus, will now be discussed<sup>4</sup>. The equation for the zone temperature integration is

$$T_z^{n+1} = \frac{\sum_{i=1}^{N_{HG}} \dot{Q}_i^{n+1} + G + (\dot{m}c_p)_{sys}^{n+1} T_{supply}^{n+1} - A}{B + F + (\dot{m}c_p)_{sys}^{n+1}}, \quad (5.1)$$

where

$$\begin{aligned} A &= \frac{C_z^n}{\Delta t} (3T_z^n + \frac{3}{2}T_z^{n-1} - \frac{1}{3}T_z^{n-2}), \\ B &= \frac{11}{6} \frac{C_z^n}{\Delta t}, \\ G &= \sum_{i=1}^{N_s} (h_i A_i)^n T_{si}^n + \sum_{i=1}^{N_z} (\dot{m}_i c_p)^n T_{zi}^n + (\dot{m}c_p)^n T_{inf}^n, \text{ and} \\ F &= \sum_{i=1}^{N_s} (h_i A_i)^n + \sum_{i=1}^{N_z} (\dot{m}_i c_p)^n + (\dot{m}c_p)_{inf}^n. \end{aligned}$$

$N_{HG}$  is the number of different zone internal heat gain sources,  $N_s$  is the number of surfaces in the zone, and  $N_z$  is the number of adjacent zones.

For the intra-HVAC-domain decomposition, the coupling data is incorporated by the term  $(\dot{m}c_p)_{sys}^{n+1} T_{supply}^{n+1}$  in (5.1). For the inter-domain decomposition, the coupling data is incorporated by the term  $\sum_{i=1}^{N_{HG}} \dot{Q}_i^{n+1}$  in (5.1).

For the inter-domain decomposition, the coupling heat rate is calculated based on the known zone state in EnergyPlus. Thereby, as explained in §3.2.5, the inter-domain decomposition is of the implicit-explicit partitioning type. Due to the additional delay introduced by the type of the coupling data, the inter-domain decomposition results in lower accuracy compared to the intra-HVAC-domain decomposition. However, there are means to improve the accuracy of inter-domain decomposition by correcting the equation (5.1) for the lagging, as follows.

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 it is calculated in the coupled simulator (e.g., TRNSYS) as a function of  $T_z^n$ , since at the simulation time point of the calculation  $T_z^{n+1}$  is not known. Thus, the resulting heat rates are

$$\begin{aligned} \dot{Q}_{sys,intra-dec}^{n+1} &= (\dot{m}c_p)^{n+1} (T_{supply}^{n+1} - T_z^{n+1}) \\ \dot{Q}_{sys,inter-dec}^{n+1} &= (\dot{m}c_p)^{n+1} (T_{supply}^{n+1} - T_z^n). \end{aligned}$$

---

<sup>4</sup>Similar discussion holds for the correction formula for the zone humidity ratio.

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

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

where  $C^{n+1}$  is a coefficient (e.g.,  $(\dot{m}c_p)^{n+1}$  or  $(UA)^{n+1}$ )<sup>5</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 into

$$T_z^{n+1} = \frac{\sum_{i=1}^{N_{HG}} \dot{Q}_i^{n+1} + G + (\dot{m}c_p)_{\text{sys}}^{n+1} T_{\text{supply}}^{n+1} - A + CT_z^n}{B + F + (\dot{m}c_p)_{\text{sys}}^{n+1} + C}, \quad (5.2)$$

which is used to update the EnergyPlus' zone air temperature for the inter-domain system decomposition whenever possible.

### 5.3.2 Strong coupling implementation

Strong coupling allows longer time steps than loose coupling for the same accuracy, but it requires iteration between the simulators. In the prototype, the iteration process is controlled by EnergyPlus (see Figure 5.3). 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

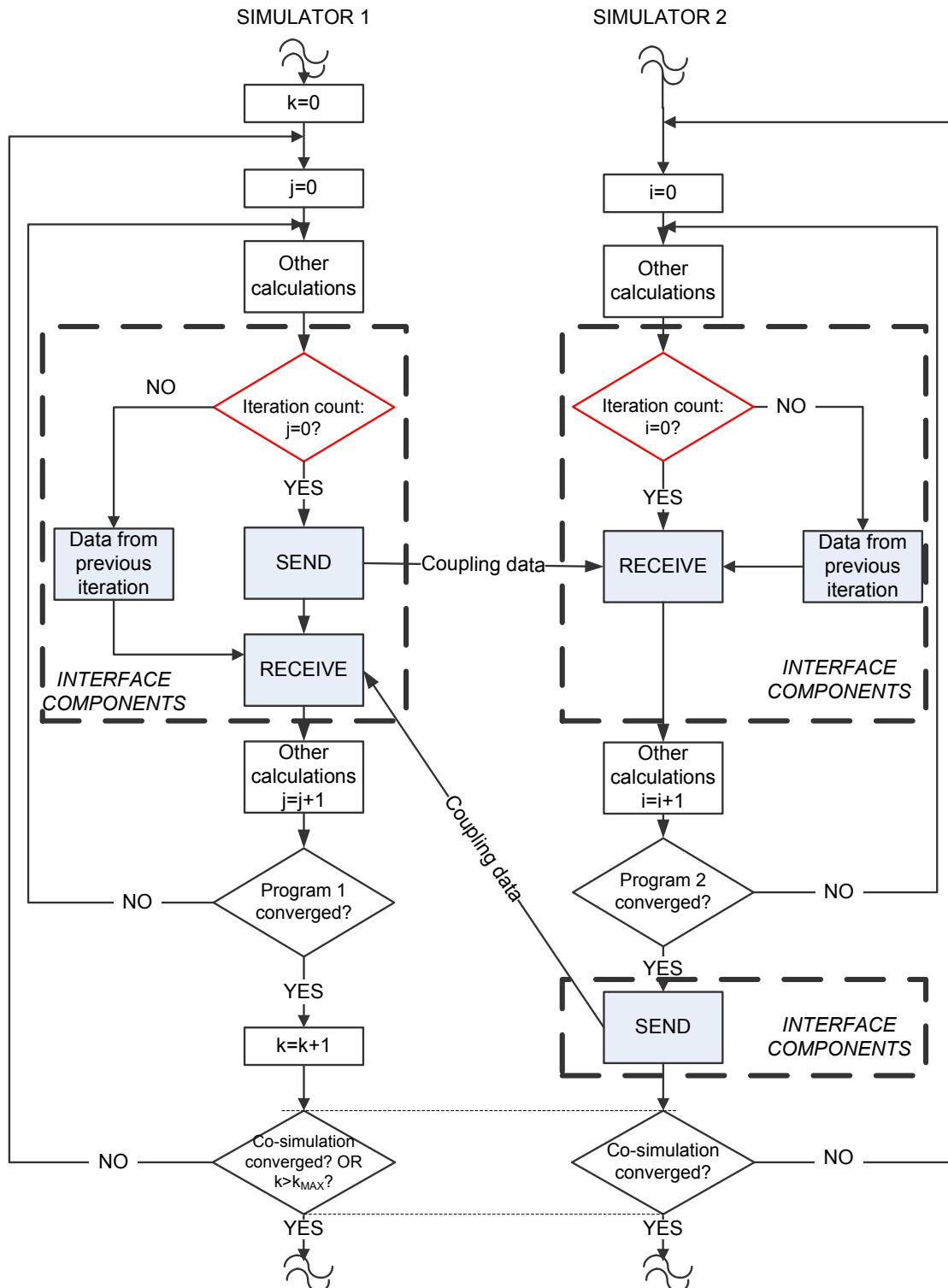
$$T_{z,k+1}^{n+1} = \frac{\sum_{i=1}^{N_{HG}} \dot{Q}_{i,k}^{n+1} + G + (\dot{m}c_p)_{\text{sys},k}^{n+1} T_{\text{supply},k}^{n+1} - A}{B + F + (\dot{m}c_p)_{\text{sys},k}^{n+1}}. \quad (5.3)$$

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 is implemented by the term  $(\dot{m}c_p)_{\text{sys},k}^{n+1} T_{\text{supply},k}^{n+1}$ . The care has to be taken so that the summand  $A$  does not change its value during the iteration process.

In some cases small change in the sensed variable can generate a large change in system output (e.g., when the system in TRNSYS is oversized and/or controllers are not tuned correctly or the simulation employs a large time step), which can lead to a non-convergent solution of the coupled system of equations. To alleviate this problem, the sensed variable (e.g., the zone temperature) is relaxed using  $\tilde{T}_{z,k+1,\text{relaxed}}^{n+1} = 0.25 T_{z,k+1}^{n+1} + 0.75 T_{z,k}^{n+1}$ . Using a larger coefficient for  $T_{z,k}^{n+1}$  generally resulted in faster convergence.

---

<sup>5</sup>The value of the corresponding coefficient should be provided by the coupled simulator and is assumed constant from one time step to the next.



**Fig. 5.3 —** Flow-chart of strongly-coupled implementation. In the prototype, Energy-Plus takes the role of the simulator of which information flow is represented on the left side of the figure. The information flow of the second simulator corresponds to the TRNSYS role.

### 5.3.3 Changes in the EnergyPlus code

The implementation on the EnergyPlus code level differs for different system decomposition strategies. Thus, the required code changes will be discussed separately for inter- and intra-domain system decompositions.

#### Inter-domain system decomposition

The building is assumed to be always modeled and simulated in EnergyPlus, while an HVAC system can be modeled and simulated in another tool, e.g., TRNSYS. The coupling heat flow is modeled as an additional internal heat gain in EnergyPlus, with negative value for cooling.

The .idd file was extended to include the data definition for the External Heat Gain component.

```
External Heat Gain,  
A1, \ field Coupled Zone Name  
    \ object-list ZoneNames  
A2, \ field Controlled Zone Name  
    \ object-list ZoneNames  
N1, \ field Number of communication sequence  
N2, \ field Actuator flag  
N3, \ field Iteration flag  
N4, \ field Predictor flag  
N5; \ field Corrector flag
```

The additional component requires the name of the coupled zone, the name of the zone of which condition is sensed if the control is modeled in a distributed manner. The coupled and the controlled zone can and most often is the same zone. The flags: (i) actuator, (ii) iteration and (iii) predictor and (iv) corrector specify (i) whether the control is distributed between the coupled simulators, (ii) whether the coupling is strong or loose, (iii) whether the prediction of the coupling data is used, and (iv) whether the correction for the lagged data is applied, respectively.

The field N1 requires an integer, used if there is more than one communication sequence. It reserves a space in the shared memory for the particular communication sequences.

Below is the part of the EnergyPlus calling tree, that is relevant for the addressed changes:

```
EnergyPlus (in Energyplus)  
    ↪ ManageSimulation (in SimulationManager)  
        ↪ ManageHeatBalance (in HeatBalanceManager)  
            ↪ ManageSurfaceHeatBalance (in HeatBalanceSurfaceManager)  
                ↪ ManageInternalHeatGains (in HeatBalanceInternalHeatGains)  
                    ↪ InitInternalHeatGains (in HeatBalanceInternalHeatGains)  
    ↪ ManageAirHeatBalance (in HeatBalanceAirManager)  
        ↪ ManageHVAC (in HVACManager)
```

```

    ↵ ManageZoneAirUpdates('PREDICT')
        (in ZoneTemperaturePredictorCorrector)
    ↵ SimHVAC (in HVACManager)
    ↵ ManageZoneAirUpdates('CORRECT')
        (in ZoneTemperaturePredictorCorrector)
    ↵ CorrectZoneAirTemp
        (in ZoneTemperaturePredictorCorrector)
    ↵ CalcZoneSums
        (in ZoneTemperaturePredictorCorrector)
    ↵ CorrectZoneHumRat
        (in ZoneTemperaturePredictorCorrector)

```

Five subroutines were changed. The changed subroutines are marked by bold letters.

### Changes in `InitInternalHeatGains`

The coupling is not performed during the warm-up period. So, the coupling between simulators is enabled only while (`.not.WarmupFlag`) is TRUE.

Since, the `InitInternalHeatGains` subroutine is only called once, in the beginning of the time step, the flags that indicate the start of the new time step are changed here, using the appropriate shared memory elements.

Further, EnergyPlus loops through all coupled heat and moisture sources for the current zone and writes the coupling data: air temperature, mean radiant temperature and humidity ratio of the coupled and controlled zones to the corresponding shared memory elements. If the prediction flag is on, predicted values for the coupling data, based on the first-order prediction, are used. It requires the knowledge of the coupling data available in the last two time steps, which are stored internally in EnergyPlus.

After the coupling data is written to the shared memory, EnergyPlus waits for the availability of the new coupling data from another simulator and reads them. The data to read are available in the form of: radiant, convective and/or latent heat rate.

If the correction flag is on, the coefficient  $C$ , used in (5.2), is also communicated back from the coupled simulator.

The radiant and latent part of the coupled heat exchange are added to the overall radiant and latent heat gain sources.

### Changes in `CorrectZoneAirTemp`

Communication with the coupled simulator in the iteration steps is performed in the subroutine `CorrectZoneAirTemp` and the general outline of the procedure is the same as in `InitInternalHeatGains` subroutine. The only difference is that the flags which indicate the new iteration step, rather than the new time step are changed here.

Depending on which coupling strategy is used and whether the correction terms are employed, the zone temperature is updated using one of the equations (5.1), (5.2) or (5.3).

### Changes in `CalcZoneSums`

The variable `SumIntGain`, is updated to encounter for the coupled convective heat power injected/extracted from the zone.

If the correction procedure for the inter-domain system decomposition is used, then the numerator and the denominator of the zone-temperature correction equation (5.2) are updated here to encounter for the correction.

The subroutine `CalcZoneSums` is called in `CorrectZoneAirTemp` after the iterative communication with the shared memory, which ensures that the last communicated values will be used in calculations.

### Changes in `CorrectZoneHumRat`

To ensure the correct moisture balance calculations, the coupled latent heat is here taken into account. The latent heat gain communicated in `InitInternalHeatGain`, in the beginning of the time step, and in `CorrectZoneAirTemp`, at the iteration steps, is used to update the overall latent gain in the corresponding zone.

Again, to improve accuracy of co-simulation of the implicit-explicit system decomposition, a correction for the time lagging of the zone humidity ratio can be used here. The same method as for the time lagging of the temperature in heat balance equation is applied here. If the latent heat gain is linear function of the zone humidity ratio, the corresponding coefficient is used to update the denominator and numerator of the humidity balance correction equation [LBNL 2006].

### Changes in `ManageHVAC`

The changes in `ManagerHVAC` subroutine are common for both decomposition approaches. To check whether the model is co-simulated, a new variable `TimeStepReduction` is introduced. Its default value is '1', and it is only changed by the additional interface routines to '0'. So the additional part of the condition is only considered in the case when `TimeStepReduction` is '0'. If it is set to '0', it means that the shortening of system time steps is forbidden, so the variable `ShortenTimeStepSys` is set to FALSE, so that no further system time step reduction is feasible.

The condition for looping over prediction-correction routine is changed and is valid also in the case where the co-simulation is employed and (i) the inter-program iteration criterion is not fulfilled or (ii) the number of inter-program iterations is less than maximum.

### Intra-domain system decomposition

In intra-domain system decomposition the coupling is done through additional components (either air or water loop components) that are built for interfacing the shared memory.

In order to enable multiple communication sequences between coupled simulators, a pair of components is always required (one component that sends the coupling data to the shared memory and one that reads the available data from

the shared memory). Also, different components are built for the strong and loose coupling. Therefore, there are eight additional components.

All eight components have their own .idd entry. However, it is generic in structure and looks as follows:

```
External Component,
A1, \ field Descriptive name
A2, \ field External coupling water inlet node
A3, \ field External coupling water outlet node
A4, \ field Controlled zone name
    \ object-list ZoneNames
N1, \ field Coupling time step
N2, \ field Actuator flag
N3, \ field Predictor flag
N4; \ field Number of communication sequence
```

The same explanations for the fields, as for the .idd inputs for the external heat gain, are valid. In addition, N1 holds the numeric value of the coupling time step. If it is zero then the coupling time step equals the assigned simulation time step. If any other value is used (the value needs to be the multiple of the simulation time step), then the communication is done at a lower frequency. The fields for inlet and outlet node name are used for linking to other components in a system model.

Two pairs of air components (for loose and strong coupling) are called from the subroutine SimAirLoopComponent, and two pairs of water components (for loose and strong coupling) are called from SimPlantEquip. The necessary changes in EnergyPlus' subroutines are as follows:

```
EnergyPlus (in Energyplus)
    ↪ ManageSimulation (in SimulationManager)
        ↪ ManageHeatBalance (in HeatBalanceManager)
            ↪ ManageAirHeatBalance (in HeatBalanceAirManager)
                ↪ ManageHVAC (in HVACManager)
                    ↪ ManageZoneAirUpdates('PREDICT')
                        (in ZoneTemperaturePredictorCorrector)
                    ↪ SimHVAC (in HVACManager)
                        ↪ SimSelectedEquipment (in HVACManager)
                            ↪ ManageAirLoops (in SimAirServingZones)
                                ↪ SimAirLoops (in SimAirServingZones)
                                    ↪ SimAirLoopComponent (in SimAirServingZones)
                                ↪ ManagePlantSupplySides
                                    (in PlantloopSupplySideManager)
                                        ↪ SimPlantEquip (in PlantloopSupplySideManager)
                                ↪ ManageZoneAirUpdates('CORRECT')
                                    (in ZoneTemperaturePredictorCorrector)
```

## Structure of additional components

Each of the new-developed interface components has the same structure. In each module there is one public subroutine (SimulateXX), which is called from

the SimAirServingZones or SimPlantEquip subroutines. The SimulateXX subroutine calls other subroutines: InitiateXX, SimXX, UpdateXX and ReportXX (respectively). InitiateXX, UpdateXX and ReportXX subroutines are the same as the corresponding subroutines written for other components. The core of the coupling is in the SimXX subroutine and it will be explained further in more detail.

As the SimXX subroutine of each sending, as well as each receiving component is similar, only the generic view of the subroutine structure in these two contexts is given.

### **SimXX subroutine for sending components**

The components that send data are called before components that read data in EnergyPlus.

In the case of loose coupling, the communication is done only if:

- the warm-up flag is off,
- it is the first pass of the first iteration in the current time step, and
- when the coupling time step differs from the simulation time step, if the simulation time is a multiple of the specified coupling time step.

For strong coupling, the communication is done only if

- the warm-up flag is off,
- it is the first EnergyPlus internal iteration in the current time step.

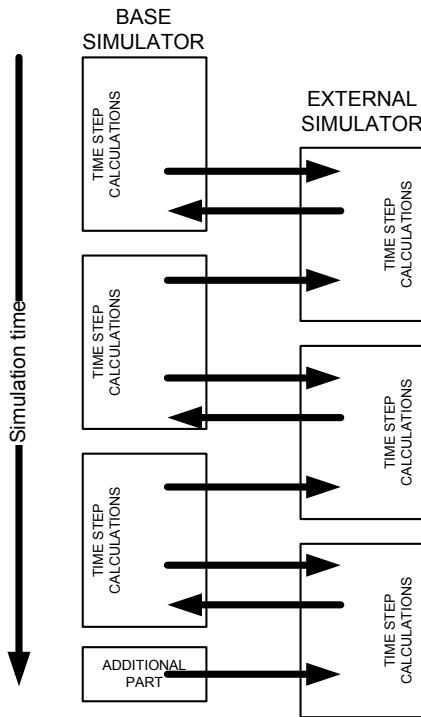
The strong coupling implementation does not allow the multi-rate co-simulation.

However, the care has to be taken to ensure that both simulators evolve equally fast in simulation time. In the strong coupling approach, the base simulator (EnergyPlus) controls the inter-simulator iteration process and needs to inform the coupled simulator whether another iteration is required, or whether the coupled simulator can proceed to the next time step. The external simulator waits for this information at the end of each inter-simulator iteration and thus the base simulator proceeds to the next inter-simulator iteration or the next time step, before the external simulator does. This is schematically shown in Figure 5.4.

The initialization of co-simulation synchronization is straightforward. However, the finalization requires additional programming effort. When the base simulator converges after the final time step, the external simulator needs to be notified. Therefore, an additional communication piece is added to the base simulator to inform the external simulator that the final step iteration process converged, so the external simulator can finalize the calculations.

### **SimXX subroutine for receiving components**

After writing to the shared memory, EnergyPlus calls receiving interface components. The conditions when the communication in SimXX takes place is the same as for sending components.



**Fig. 5.4** — Schematic representation of synchronization information flow in strongly coupled co-simulation.

If the communication is requested, EnergyPlus waits until the new coupling data is available from the coupled simulator and reads it. The received data is passed to the output nodes.

If the above condition is not met, the communication is not done and the previous available communicated data is passed to the output nodes.

### 5.3.4 Changes in TRNSYS code

Apart from the additional component developments, in order to accommodate the need for the strong coupling controlled by another simulator, some subroutines in the kernel section<sup>6</sup> needed to be changed, since TRNSYS takes a role of a passive simulator in the overall co-simulation process.

TRNSYS receives information from the coupled simulator indicating whether a new time step or a new iteration calculation is required. If the coupled simulator requires another iteration in the current time step, a call to `Exec` is done with `INTG=12`, which indicates that `INFO(15)` should be set to '1'. (`INFO(15)` is used for the interface components in strong coupling to indicate whether it is a beginning of a time step (`INFO(15)` is set to '2'), or another iterative call (`INFO(15)` is set to '1').) The calculation procedure is rewound to the beginning, setting at the same time the value of the variable `NewIteration` to '1'. If this variable is not '0', TRNSYS (in subroutine `Exec`) will not reset the values of `INFO(7)` to '-1', so

---

<sup>6</sup>TRNSYS simulation engine

the iterative calls to components will be kept counting. The value of `INFO(13)` is reset to '0', as also converged components need to be called again.

If the coupled simulator indicates that a new time step calculation is required, the call to `Exec` is done with `INTG=13`, which indicates that `INFO(15)` should be set to '2'. After this call, the calculation in TRNSYS continues as usual (call to the printers, integrators, etc.), setting the value for `NewIteration` to '0'.

TRNSYS does not distinguish between components that use different working fluids, as it's done in EnergyPlus.

### Inter-domain system decomposition

Three additional components (working fluid state and flow sending and receiving coupling components, plus component for the control coupling) are developed. The control coupling component simply reads the data from the shared memory and in general is the same as the receiving component. This three interface components can be used for both coupling strategies.

The receiving components have `INFO(9)` set to '1', which indicates that they are regarded as regular converging components, of which output depends on time and which are called at least once per time step. Additionally, the communication is done only if `INFO(15)`, originally set to '2', is higher than '0'. If higher than '0', it indicates whether EnergyPlus requires communication in the new time step or whether it requires the new iterative call in the current time step. Once the communication is done, `INFO(15)` is set to '0', so that in the following TRNSYS-inner iterative calls, the communication is skipped. The value from the last communication is kept constant.

Before the communication is done, TRNSYS checks, and if necessary waits until the corresponding indicators become set to '1', meaning that the new data is available to be read. After reading the coupling data, the same indicators are set back to '0', indicating that new data can be written to the corresponding cells in the shared memory. The components simply read from corresponding elements of the shared memory.

The sending component have `INFO(9)` set to '7', which means they are called after all other components have converged and before the setting of the storage variables is performed. The component writes coupling data to the corresponding element of the shared memory.

### Intra-domain system decomposition

For intra-domain decomposition, there are two triplets of components (working fluid state and flow sending and receiving coupling components and the component for the control coupling), since in the current implementation, a distinction between components used for different coupling strategies is made.

In the case of the strong coupling the components, apart from different data and the places in the shared memory from which the data is read, the components are identical to those used for inter-domain decomposition.

In the case of the loose coupling, receiving and control coupling components have `INFO(9)` set to '1', while the sending component has it set to '3'. This means

that the sending component is called after all components have converged, storage variables are set, but before printers and integrators are called. The communication is done only if:

- it is the beginning of the simulation, or
- the simulation time equals the coupling time, or the coupling time step is set to '0', and
- it is the first call in the current time step (only for receiving and control components).

Receiving components will not read from the shared memory as long as the new data from the coupled simulator (EnergyPlus) is not available, which is indicated through the adequate flags in the shared memory.

After writing to the shared memory, TRNSYS sets the corresponding flags to indicate to EnergyPlus that the new information is available for reading.

## 5.4 ESP-r/EARTH prototype

EARTH is stand alone program for modeling and simulation of earth-to-air heat exchangers. The changes to the EARTH simulator are minimal and only changes in ESP-r are discussed.

### 5.4.1 Changes in ESP-r code

This subsection focuses on the changes within the ESP-r code. The changes allow only intra-domain system decomposition approach and loose coupling strategy.

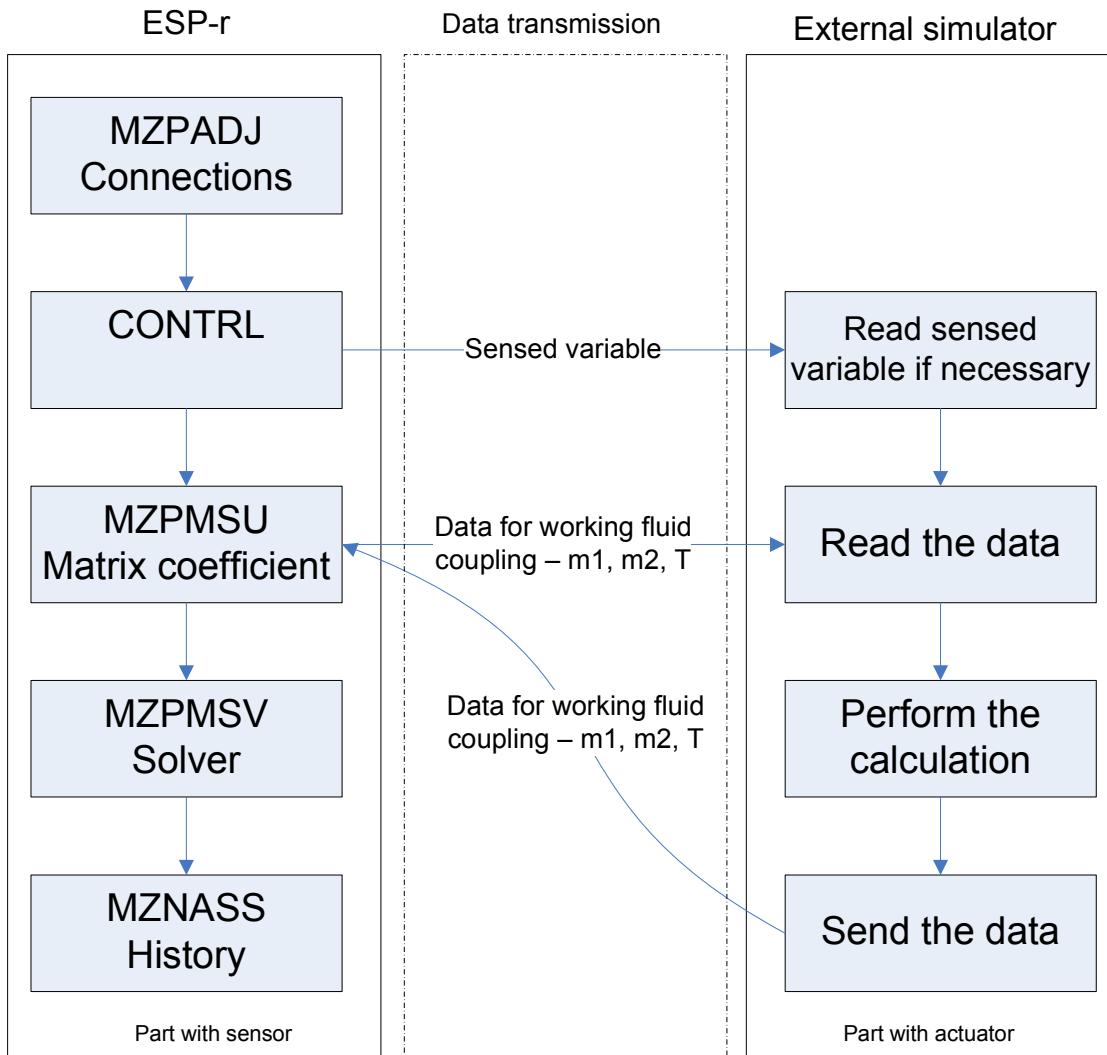
Additional components are developed for both air or water systems (Figure 5.5).

Instead of additional components, additional connections (in ESP-r terminology) could have been used as well. From the solution point of view both approaches are identical. However, in the case of connections, it would be important to take care of the order in which the connections would be defined.

The mechanism for the discontinuously running external simulator uses intermediate files. In case of the continuously running external simulator, data transfer is via named pipes.

Figure 5.5 shows the process of variable exchange and places where the original code had to be altered to accommodate the requirements for run-time exchange of coupling data. The figure shows the main subroutines involved in simulation of an HVAC system model. Only the subroutines which had to be adjusted to accommodate the coupling are discussed.

The `CONTRL` subroutine determines system control status based on the most recent calculation results. To enable the exchange of control data, "coupled" control components and an additional control law are constructed. The role of the new control law is simple. It copies relevant information from its source and transfers it to the coupled simulator. In addition, if required, the simulator reads



**Fig. 5.5** — External coupling implementation in ESP-r within one time step (MZPADJ, CONTRL, MZPMSU, and MZNASS are ESP-r-specific subroutines). External simulator refers to the EARTH simulator or to any other simulator coupled to ESP-r.

the sensed data from the named pipe. This is done in the subroutine that locates the sensor and determines the sensed condition. The “coupled” control components are only used to enable the standard system/control definitions in ESP-r., and do not have any influence on the solution.

The MZPMSU subroutine sets up the system equations in matrix form. It calls components’ subroutines that generate matrix coefficients and locates them in the system network matrix. Two new component’s subroutines were constructed (for each coupling mechanism), one as an air-loop component and the other as water-loop component. The interface components are one-node components. Each self-coupling coefficient of the energy balance matrix is set to 1, cross-coupling coefficients of the energy balance matrix is set to 0, and the right hand side coefficients of the energy balance matrix is prescribed the value of the coupling data. If the flow is driven by a flow inducer modeled in ESP-r, which is the case in this prototype, the mass flow though the coupled system is equal to the mass flow set in ESP-r, i.e., the mass flow through the heat exchanger is equal to the mass flow of the ESP-r sending component node. Otherwise, the interface component sets the mass flow in the same way the flow inducer component model in ESP-r does.

The components interface the coupled simulator and send/receive data to/from it. If the coupling time step is greater than the simulation time step employed internally, the coupling data is kept constant and equal to the last transmitted values during coupling time step calculations.

# 6

## Validation

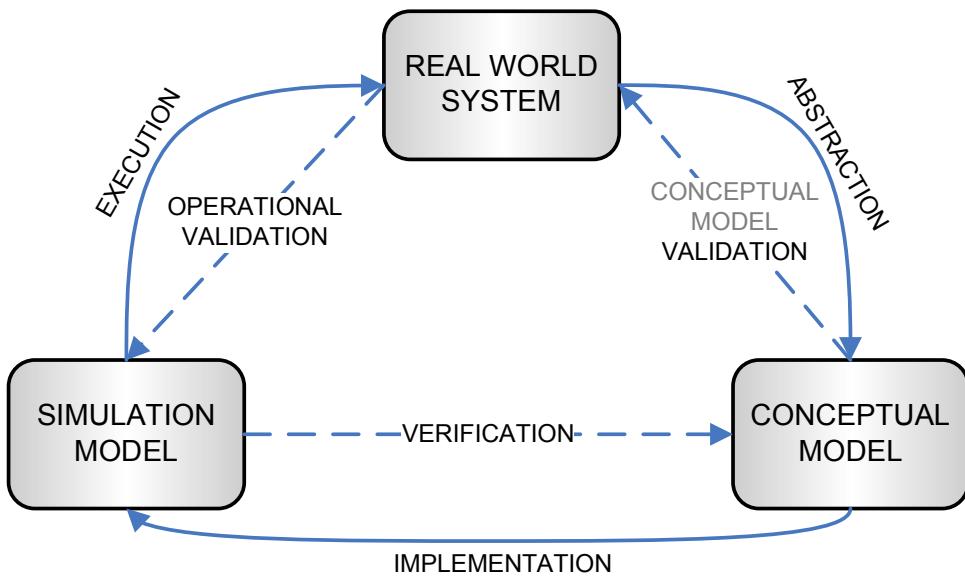
NATURE is complex, and it is not in our human ability to understand it in her full complexity. In an attempt to comprehend the parts of it, people began to examine and try to model the world we live in. Many models are developed as an attempt to explain different physical processes and give us some ability to predict future events. The models represent simplification of reality and due to both our limited knowledge and the randomness of Nature there remains always some uncertainty. Therefore, there is no such thing as an absolutely correct model of a physical system. Moreover, systems can be modeled in variety of ways, and different design analyses can require different models of the same system. The built environment, although only a part of the complex universe and much smaller in scale, is almost as complex - a universe itself.

In this chapter, an overview of verification and validation techniques is given. A special variation of inter-model comparison validation technique is defined to be used for co-simulation validation. The results of verification and two-step validation are discussed in detail.

### 6.1 Literature review

In Figure 6.1, the modeling and simulation process is presented. A conceptual model describes how a model developer intends an implementation to satisfy its requirements. It is the primary mechanism for transforming simulation requirements into specifications that will guide simulation development and implementation. In other words, it is a collection of information about assumptions, relationships and data, which describes a modeler's concept about the real world. It is the abstraction of reality that distinguishes the real world system from a conceptual model of that system for a particular task. Implementation of the conceptual model is a simulation model. The simulation model is specific to a simulation tool, while the conceptual model should be independent of the simulation tool. This also reinforces that a model should be problem- not tool-led. Defining the modeling and simulation process as in Figure 6.1, clearly defines where the validation and verification processes take place.

Conceptual model validity (or representational/behavioral accuracy [Balci 2004]) is defined [Sargent 2005] as determining that the theories and assumptions underlying the conceptual model are correct and that the model representation of the problem entity is reasonable for the intended purpose of the model. Concep-



**Fig. 6.1** — Modeling abstraction in the simulation process.

tual model validation can be also phrased as [DMSO 2004] confirming that the capabilities indicated in the conceptual model embody all the capabilities necessary to meet the requirements of simulation.

Verification (or transformational accuracy [Balci 2004]) is defined [Sargent 2005] as ensuring that the computer programming and implementation of the conceptual model is correct.

Operational validity is defined [Sargent 2005] as determining that the model's output behavior has sufficient accuracy for the model's intended purpose over the domain of the model's intended applicability. However, any deficiencies found may be due to an inadequate conceptual model, errors in the computer program and its implementation, or they may be caused by the data.

Other terms, such as accuracy or fidelity [Gross 1999; Nayak and Joskowicz 1996; Rickel and Porter 1997; Falkenhainer and Forbus 1991] are often used in place of validity. However, as stated by [Zeigler et al. 2000] fidelity has a slightly different meaning as it refers to a combination of validity and detail, while accuracy is considered to be an aspect of validity/fidelity. Precision is also an aspect of validity/fidelity. So, high fidelity corresponds to both high validity and high detail. However, most of the authors do not make such distinction between these terms.

A simulation model should be developed for a specific purpose (or application) and its validity determined with respect to that purpose. Thus the model is valid if the accuracy of variables of interest is within the acceptable range, which is required for the model's intended purpose.

### 6.1.1 Verification and validation techniques

There are three historical methods of validation and verification: (i) rationalism, (ii) empiricism, and (iii) positive economics [Sargent 2005]. Rationalism assumes that everyone knows whether the underlying assumptions of a model are true. Empiricism requires empirical validation of every used assumption and model outcome. Positive economics is not concerned with model's assumption nor its structure, but only requires that the model is able to predict the future, i.e., to match real operational data.

Sargent [2005], Law and McComas [2001] and Balci [1997] list numerous techniques for validation and verification used in practice. More than 77 different techniques are identified and classified in various ways. For example, Balci [1997] makes distinction between four categories of techniques for conventional simulation models: *informal*, *static*, *dynamic*, and *formal* techniques. The use of mathematical and logical formalism by the techniques in each category as well as complexity increase from informal to formal.

*Informal techniques* use tools and approaches that heavily rely on human reasoning without a mathematical formalism. Among others, techniques such as documentation checking, face validation, walkthroughs, etc., are considered informal.

*Static techniques* do not require a computer model execution, but they rely upon a mental mode execution. They can obtain information about structure of the model, modeling techniques, data and control flow within the model, and syntactical accuracy. This category includes: semantical analysis, structural analysis, syntax analysis, tractability assessment, data analysis etc.

*Dynamic techniques* are intended for evaluating the model based on its execution behavior. Therefore, the computer model execution is required. Some dynamic techniques use model instrumentation, i.e., addition of code into the executable model, which purpose is collecting information about model behavior during execution. This category includes: sensitivity analysis, comparison testing, compliance testing, graphical comparison, special input testing and many others.

*Formal techniques* are based on mathematical proof of correctness. The model is written using a special formal semantics. Set of required properties are also specified in the formal manner. A mathematical apparatus is then applied to prove that the model satisfies the set of properties.

In most BPS literature [Judkoff and Neymark 1995; Hensen 1991] three categories of validation techniques are recognized:

- analytical techniques,
- inter-comparison techniques, and
- empirical techniques.

Hensen [1991] and Vandaele and Wouters [1994] also cite some authors that extend the above validation methodology classification with:

- an initial examination of a model's theory and a thorough inspection of the corresponding source code,
- a parametric sensitivity analysis to ascertain whether or not the model predictions lie within the uncertainty band associated with the input data, and
- user effects, which cannot be avoided and is the influence of the model user on simulation results when converting the thermal system into the form required by the model.

To try to unify different categorizations for the purpose of building performance validation, the techniques are grouped into the three main categories:

- **Rational validation**, that besides its historical meaning includes a knowledgeable and logical results criticism. This category includes any reasoning about correctness of the model's logic. The techniques are grouped as:
  - *static*: examination of model theory and semantic analysis;
  - *passive*: by observing the simulated behavior of the system and/or face validation (performed by an expert to determine whether the input/output relationships are as expected);
  - *dynamic*: by sensitivity analysis, which is performed by changing the values of the input and internal parameters of a model to determine the effect upon the model's behavior or output, and/or by using degenerate tests, which are performed by appropriate selection of values of model input and internal parameters to determine the effect upon the model's behavior and its output. The same relationship should occur in the model as in the real system.
- **Inter-model comparison** using:
  - *analytical model* - simplified cases of a simulation model can be compared to known results of an analytic model by adjusting the values of model input and internal parameters to the cases which are easily analytically calculated.
  - *numerical model* - the simulation model can be compared to other models that have been previously validated. This group includes BESTEST [Judkoff and Neymark 1995] and HVAC BESTEST [Neymark and Judkoff 2002, 2004] validation tools, as well as the validation procedure described in the standard ASHRAE 140 [ASHRAE 2004].
  - *formal methods* for obtaining a proof of correctness.

Some dynamic and formal techniques fit into this category.

- **Empirical validation**, i.e., comparison between the system's behavior and the model forecast. It comprises only of dynamic techniques. The system data may come from:

- *experiments* performed on the system, where most of the model inputs and internal parameters are known and controlled throughout the experimentation.
- *an operational system*, by using for example available system historical data.

## 6.2 Verification of co-simulation

Two basic approaches are applied for program verification: (i) static and (ii) dynamic [Hu et al. 2001; Sargent 2005].

Static testing is performed by examining the structure of the program, while the dynamic testing is performed by executing the computerized model under different conditions and using the resulting values to determine if the implementation is done correctly.

In order to verify the co-simulation prototypes, static techniques are applied first. The code has been analyzed to determine if it is correct by examining structural properties of the program.

Further, the dynamic testing has been implemented. The prototypes were executed in several runs implementing various coupling time steps and printing coupling data either to the screen or to a file. The coupling time stamps and corresponding coupling data obtained from the coupled simulators were then compared, demonstrating that the sent coupling data are equal to the read (received) coupling data at the corresponding coupling time samples. The dynamic testing verifies exact synchronization procedure and correct data transfer.

## 6.3 Validation of co-simulation

The verification showed that the interfaces between the coupled simulators are implemented correctly. However, to test whether the approach, concerning what coupling data are chosen and at which frequency the coupling data are exchanged, is correct, is done by a validation study. The main objective of this validation study is to provide the evidence and to increase the confidence that co-simulation produces equally accurate results as mono-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 the simulation tools and do not fully satisfy the requirements for the validation of the coupling, since the results are influenced by the validity of the coupled simulators. If however, the coupled simulators were both validated using the same validation technique, they could be used to validate the coupling. But, there are two obstacles in doing so:

- Most of the traditional validation procedures, used to validate BPS tools are concerned with

- either only the BPS model, e.g., using inter-model comparison techniques [ASHRAE 2004; Judkoff and Neymark 1995], or
  - a single HVAC component model using empirical validation techniques or inter-model comparison techniques [Hensen 1991].
- So far, only HVAC BESTEST [Neymark and Judkoff 2002, 2004] procedure takes into account coupling between a building and a system model. HVAC-BESTEST Volume 1 [Neymark and Judkoff 2002] considers steady state tests, that could be solved with analytical solutions. Volume 2 [Neymark and Judkoff 2004] includes hourly 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 it 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 only a Technische Universität Dresden (TUD) TRNSYS version, with developed new source code for TUD's own calculation routines.

Thus, because of importance of dynamic effects in co-simulation, the full, exact and complete validation of the coupling could not be done using the traditional BPS validation techniques. However, as a preliminary validation test, 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 themselves.

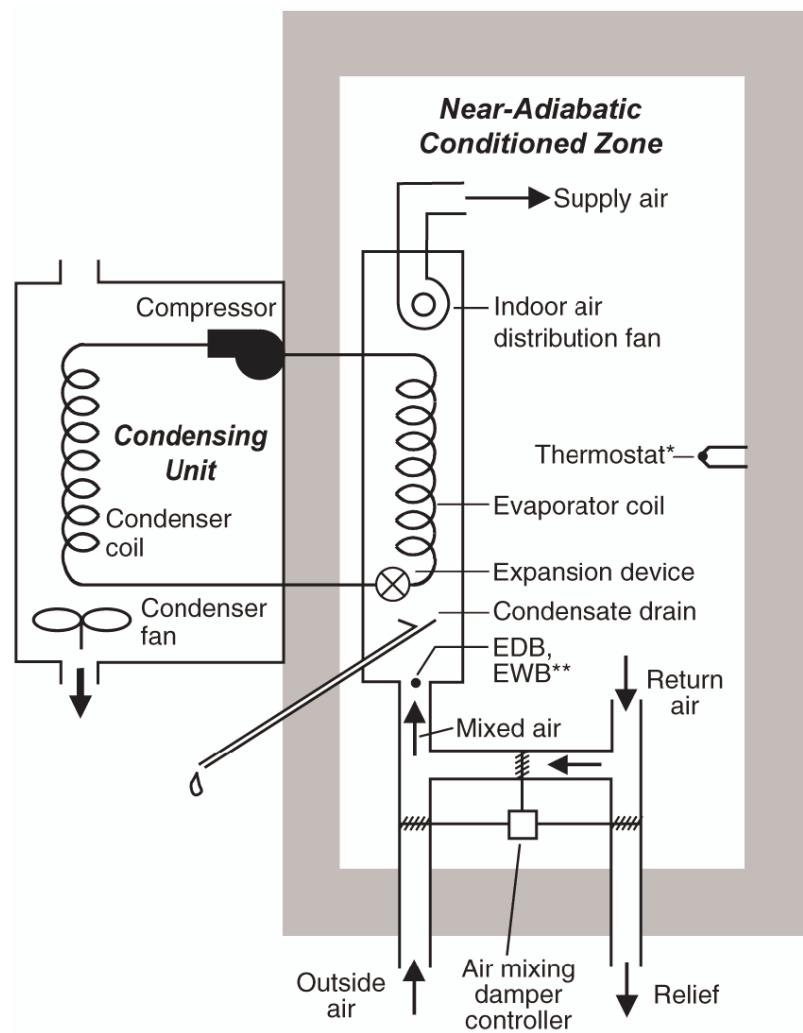
Another way to validate the coupling would be to compare results of a co-simulation to results of a mono-simulation. However, the difference in results between the mono- and the co-simulation, using different tools, would be influenced not only by the coupling but also by differences in modeling approaches and solution techniques used in the coupled simulators.

Thus, a two-step procedure, based on the inter-model comparison technique, has been acquired. Firstly, to avoid the influence of differences in different simulation models, only one simulator is used for both the mono- and the co-simulation model. Secondly, results obtained by different co-simulation implementations are compared with each other. Since, the different implementations are independent on each other, this comparison can be used to further increase the confidence in the co-simulation approach.

### 6.3.1 HVAC BESTEST E300 case

#### Methodology

"BESTEST is a comprehensive and integrated set of tests for quality assurance during development of building energy simulation computer programs. NREL originally developed the BESTEST method in IEA SHC Task 12 and Energy Conservation in Buildings and Community Systems (ECBCS) Annex 21 to test building thermal fabric (envelope) models, and to diagnose sources of predictive disagreements. This procedure was adopted with some refinements by



\* The thermostat senses only the zone air temperature (IDB).

\*\* Entering dry-bulb and wet-bulb temperatures.

**Fig. 6.2** — Unitary split-system, air-cooled condensing unit with indoor evaporator coil, and with outside-air mixing system [Neymark and Judkoff 2004].

the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) and the American National Standards Institute (ANSI), and now forms the basis for ANSI/ASHRAE Standard 140, Method of Test for the Evaluation of Building Energy Analysis Computer Programs. HVAC BESTEST extends the original BESTEST by adding the capability to test and diagnose mechanical system models. HVAC BESTEST Volume 1, cases E100–E200, are steady-state analytical verification tests that check the ability of simulation programs to apply basic performance map modeling techniques to simulation of unitary space cooling equipment on the working-fluid side of the cooling coil. These cases have been added to ANSI/ASHRAE Standard 140.8” [Neymark and Judkoff 2004].

For the preliminary assessment of the validity of the coupling implemented in the prototypes, the reference case - E300 - from the HVAC BESTEST Volume 2 is used. The case E300 consists of a simple zone model, with highly insulated - adiabatic envelope and without any heat capacitance, and a unitary split system, shown in Figure 6.2. The indoor fan is a single-speed draw-through and continuously operating. 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 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 any potential modeling error, the zone model was taken from the original EnergyPlus model of E300. The system is modeled using TRNSYS Type 665. The results are compared against the results obtained by different tools.

### Results

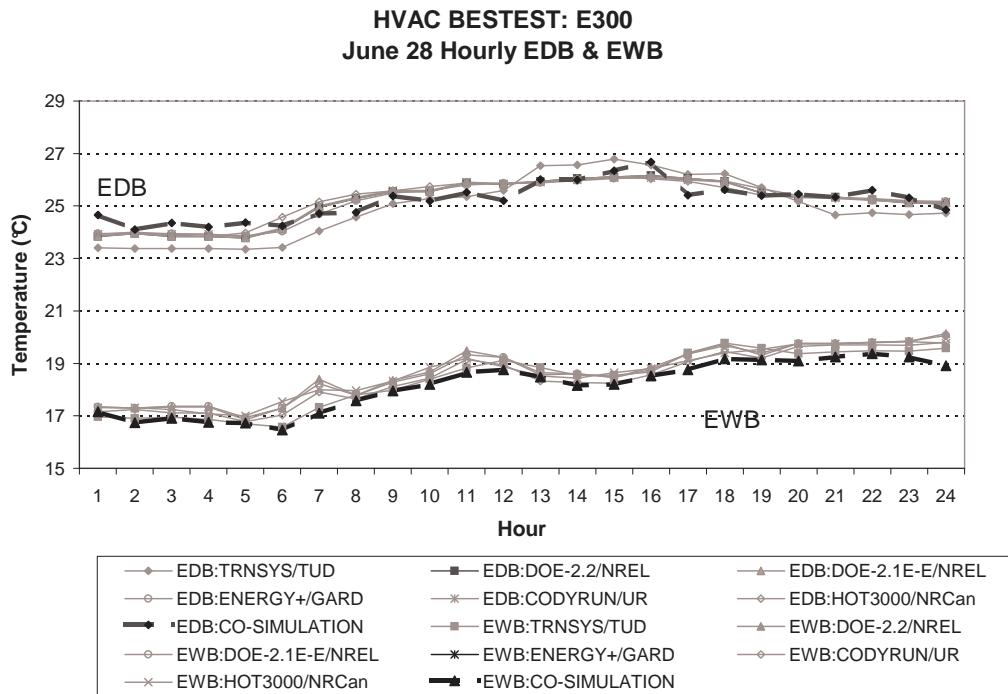
As already mentioned, the case E300 requires an ideal control model for the system. The co-simulation model of the case HVAC BESTEST E300 with EnergyPlus and TRNSYS can not be used with an ideal system control. There are two reasons for this. First, the split system model in TRNSYS can be either ON or OFF. Second, co-simulation does not allow the distributed ideal control modeling. Thus, a model of a realistic controller is used. However, using the realistic controller with the fast responding zero-capacitance building and zero-capacitance system results in oscillatory zone temperature even when a small time steps are used. Consequently, the co-simulation model does not exactly replicate the E300 model requirements, which should be taken into account when comparing the results.

The simulations are performed using inter-domain system decomposition and loose coupling strategy. The justification of the choice for this co-simulation strategy will be given in §6.3.2, where different co-simulation strategies will be compared. The results of one-day (28<sup>th</sup> of June) simulation are presented in Figures 6.3, 6.4 and 6.5. The simulation is performed with a time step of 1 minute and the shown results are hourly averaged values.

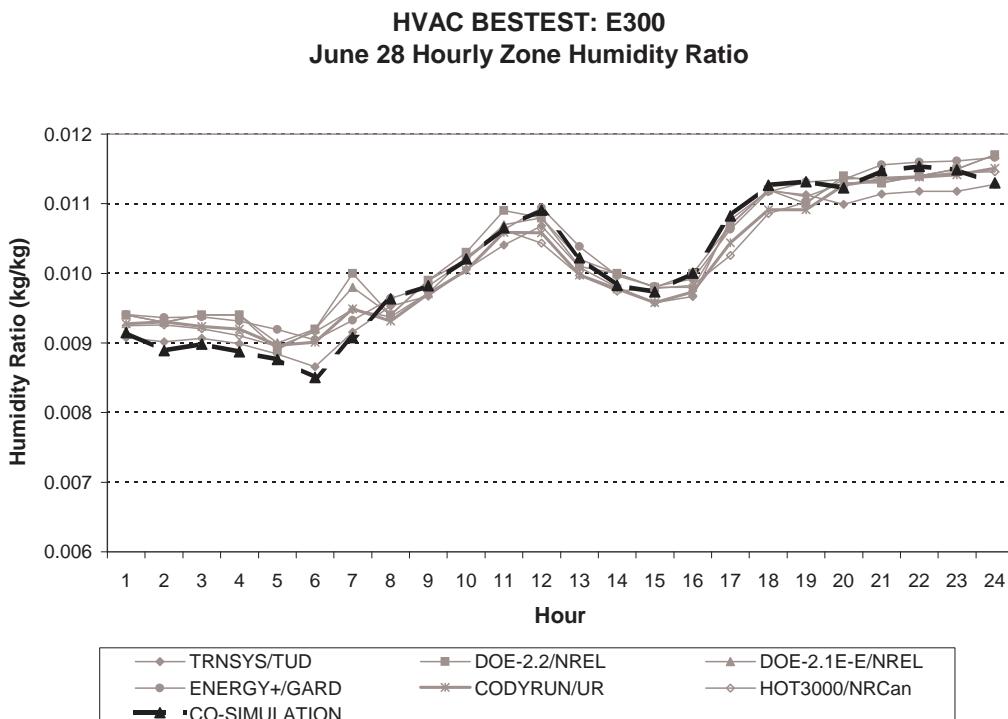
As it can be seen, the co-simulation results match well the results obtained by other tested BPS tools<sup>1</sup>. This preliminary validation test gives an evidence that

---

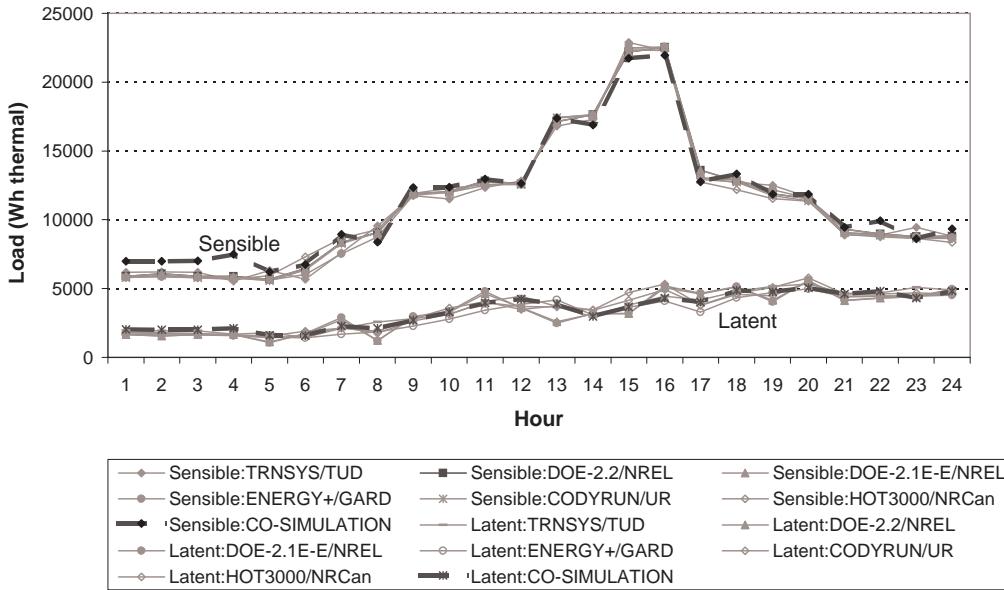
<sup>1</sup>The test also revealed one error in the TRNSYS type 665. The percentage of the outside air in the supply air flow was not correctly taken into account. The error has been fixed. The code of the



**Fig. 6.3** — Zone (evaporator entering) dry-bulb (EDB) and wet-bulb (EWB) temperature for a specific simulation day.



**Fig. 6.4** — Zone humidity ratio for a specific simulation day.



**Fig. 6.5** — Coil thermal loads for a specific simulation day.

the co-simulation implementation provides valid results. This evidence gives a solid ground for further validity investigation.

### 6.3.2 Comparison of mono- and co-simulation

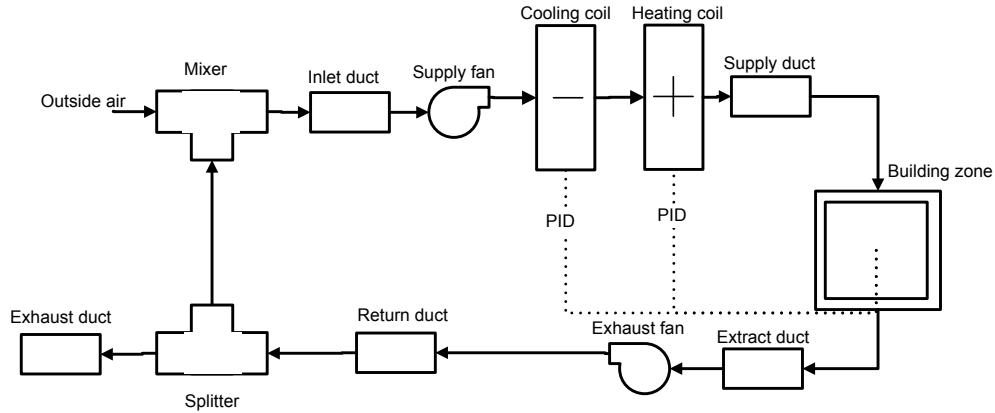
#### Methodology

The system presented in Figure 6.6 is used to validate the co-simulation prototypes implementing the loose coupling strategy. To avoid the influence of differences in simulation models, only one simulator (ESP-r) is used. ESP-r simulator is adapted for such purpose by making modifications to encounter for communication in both directions as shown in Figure 5.5.

Due to the staggered calling procedure (different sending-receiving sequence), implemented in the base and the external simulator, there is a need for two (base and external) interface components pairs. Thus, to the addition to the changes reported in Chapter 5, two new interface components were required so that ESP-r can play the role of an external simulator as well. The additional ‘receiving’ interface component sets the values of the state and the transport data to the value of the coupling data from the corresponding named pipe. However, the ‘sending’ interface component of the external simulator has a special use. It does not communicate the coupling data, but it is used to check whether coupling data need to be written to a named pipe after the current time step calculations are performed. This is done by an additional subroutine, which is added and called after the system matrix solution and before adjusting all system related history variables. The additional subroutine interfaces the corresponding named pipe and sends the coupling data to the based simulator, if required.

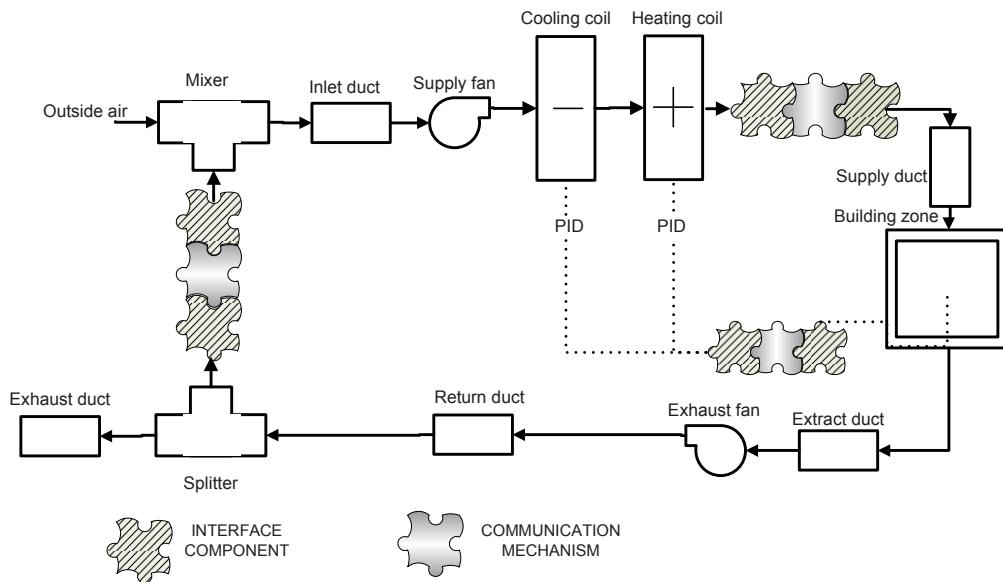
---

type 665 was also adjusted so that only indoor fan power is included in the performance data.



**Fig. 6.6** — Sketch of the system being simulated.

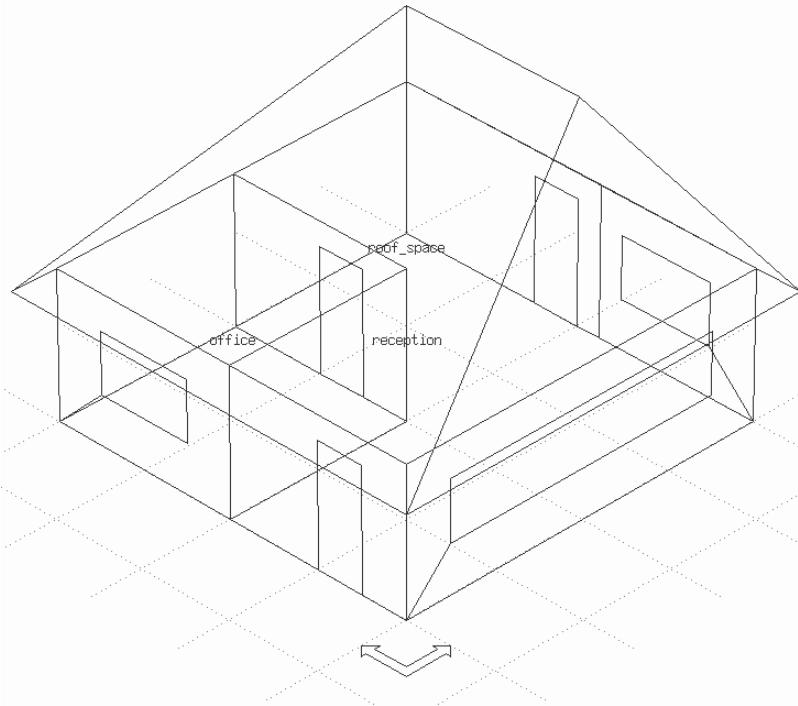
For mono-simulation purpose, the whole system (Figure 6.6) is presented by one ESP-r model, i.e., all components are modeled and simulated in a monolithic traditional way. For co-simulation, the system is decomposed into two subsystems, as shown in Figure 6.7. Each subsystem is presented separately by 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.



**Fig. 6.7** — Sketch of the decomposed system being simulated.

The exemplar building from ESP-r, consisting of three zones: a reception, an office and a roof space, shown in Figure 6.8 is used. The system from Figures 6.6 and 6.7 is coupled to the reception.

Specifications of the system in Figures 6.6 and 6.7 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 accordingly to the calculated annual maximum, obtained with the climate file for



**Fig. 6.8** — Sketch of the exemplar building used for the validation of the loose coupling.

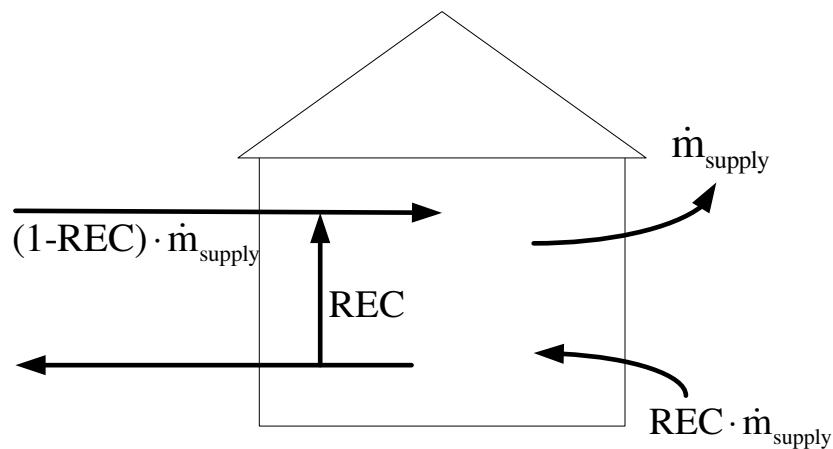
Palermo, Italy. Proportional controller, with the throttling range is set to 2 K, is used to regulate the zone temperature by actuating the heat flux in the coil. The set point temperatures and the available cooling capacities for three periods of the day are shown in the Table 6.1.

**Table 6.1** — Zone temperature set points and available capacities

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

To test the influence of the different values of the coupling data (delayed data), the amount of the returned air was varied from 0% to 100% of the overall air flow. The amount of the outside air replaced by the returned air was taken into account by increasing the infiltration rates to ensure the same energy requirements (Figure 6.9). However, this experiment was directly influenced by the solution procedure in ESP-r, which will be explained in the following sections. Thus, additional simulations using higher volume flow rates and higher capacities (oversized system) are performed to examine the matter.

The mono-simulation model and the co-simulation model are executed using various time steps and (only for demonstration purposes) for a duration of one simulation day, without any precalculation period. Zone temperature, cooling powers, moisture flow rate at the outlet of the cooling coil, and cooling energy



**Fig. 6.9 —** Air flow schematic.

use over the simulated period were compared and the results are discussed in the following section.

## Results

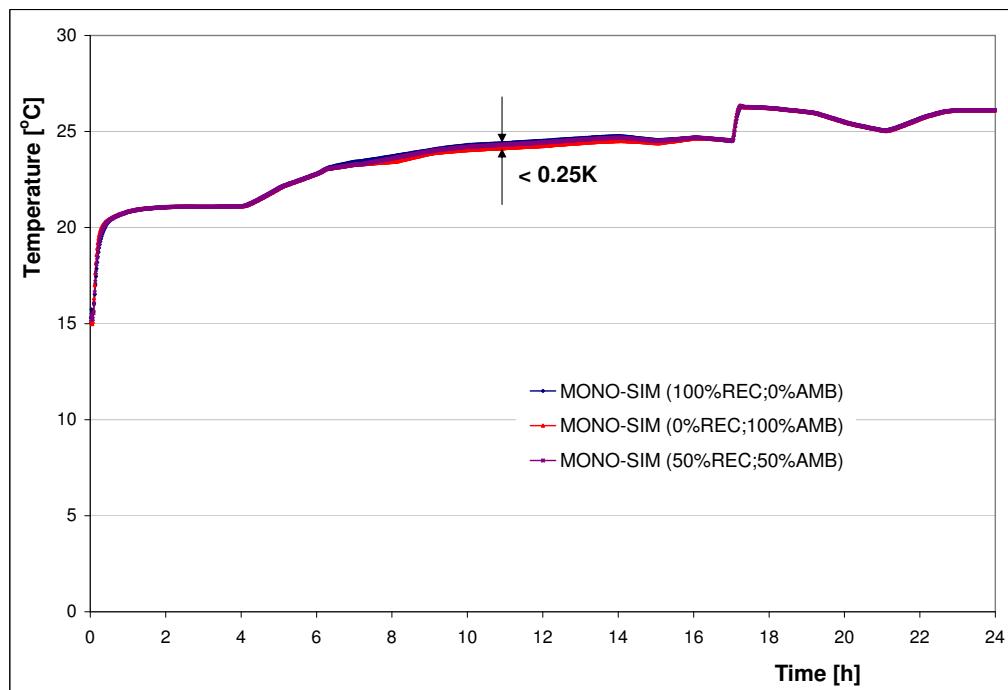
### ◊ Differences in mono-simulations

The results obtained from mono-simulations, differ for different percentages of the returned air in the overall air flow. The difference comes from the different air states entering the coil and from the order of the calculation in ESP-r. The building and the system modules are executed sequentially each time step. Infiltration is taken into account by the building module, while the air flow through the system is taken into account by the system module. As a consequence, the sensed zone temperature differs in the cases when the overall air flow from the outside is taken into account by specifying higher values for infiltration and when it is supplied to the zone through the air system, resulting in different actuated fluxes.

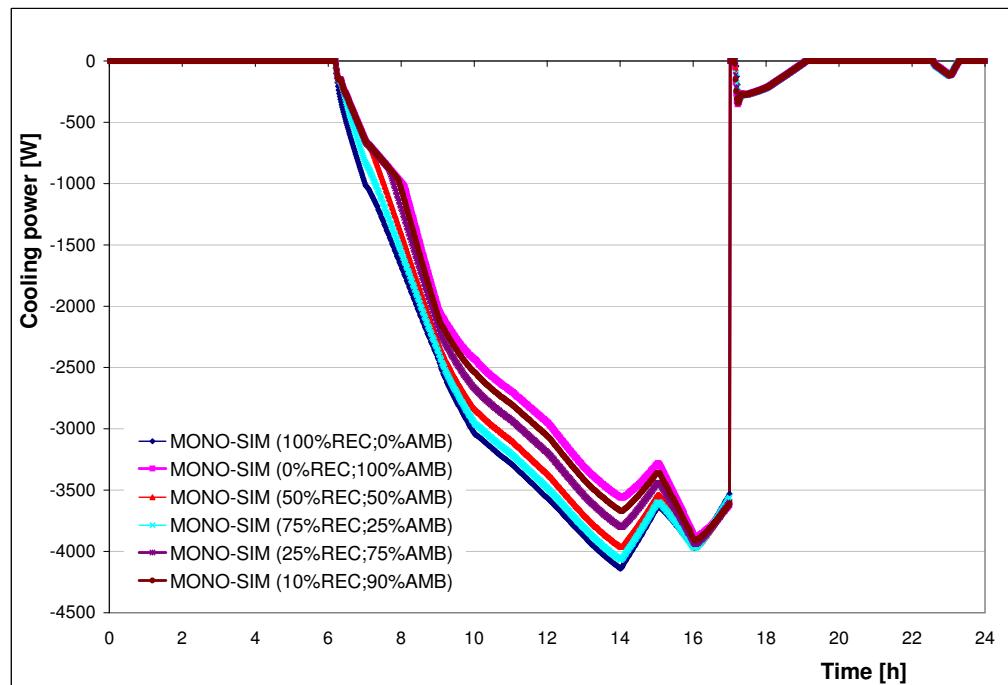
The variations of the zone temperature and the cooling power, due to the different returned air fractions in the overall air flow obtained by mono-simulations are shown in Figures 6.10 and 6.11, respectively. The zone temperature's variations are within 0.25 K range, while the resulting difference in the cooling power is within 600 W range, which corresponds to the specified maximum cooling power and the assigned proportional gain.

### ◊ Comparison of mono- and co-simulations

Applying a time step of 1 min for both the mono- and the co-simulation, the difference between results is negligible. If the difference in the first time steps, that are influenced by the specified initial conditions, is neglected, the difference in obtained zone temperatures (for the case with 50% outside and 50% recirculation air) is less than 0.01 K (Figure 6.12). The difference between the cooling power curves is less than 1% relative to the results obtained by mono-simulation



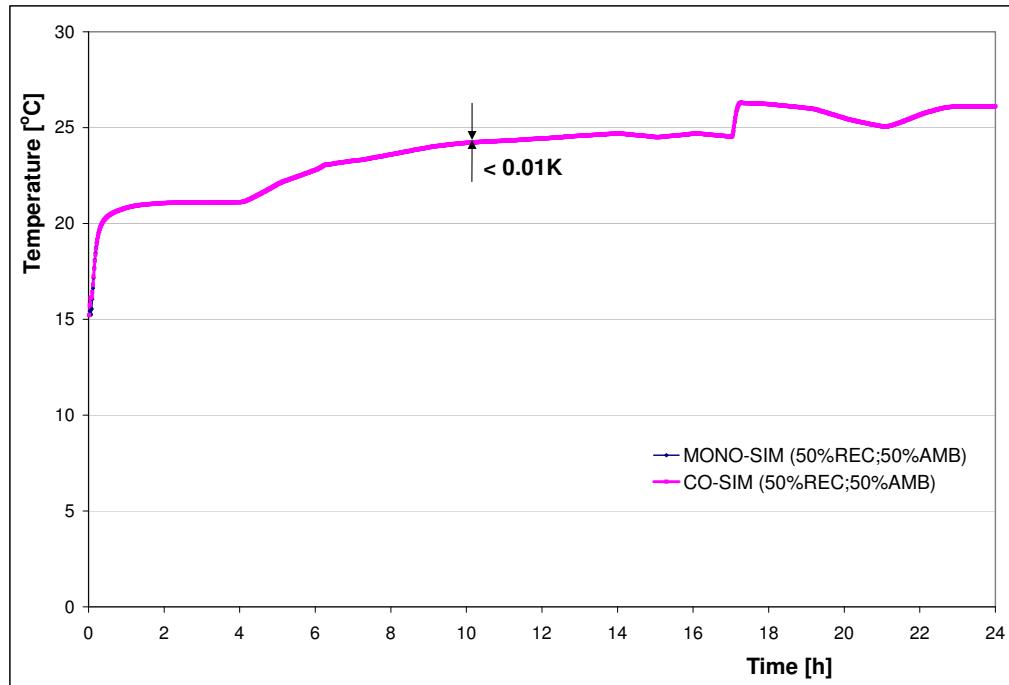
**Fig. 6.10** — Temperature curves for different percentages of the returned air that passes through the system and with  $\Delta t = 1$  min.



**Fig. 6.11** — Variations of the cooling power curves due to a different percentage of the returned air that passes through the system with  $\Delta t = 1$  min.

(see Figure 6.13). The difference between the moisture flow rate curves calculated after the cooling coil (Figure 6.14) is less than 0.00015 kg/s or 2.8% relative to the results obtained by mono-simulation.

Thereby, using a sufficiently small time step, the loosely coupled co-simulation performs as well as the mono-simulation.



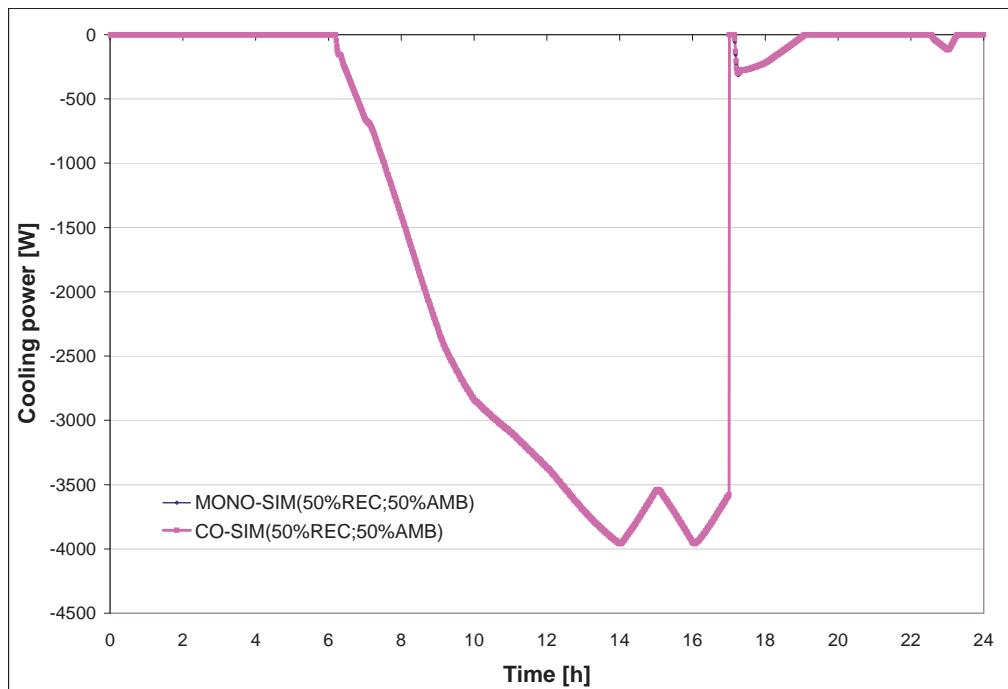
**Fig. 6.12** — Temperature curves obtained by mono- and co-simulation with  $\Delta t = 1$  min.

With larger time steps, the influence of 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 6.15, the resulting moisture flow rates after the cooling coil from mono- and co-simulation are presented in Figures 6.16, and 6.17, and the resulting cooling power in Figures 6.19 and 6.18. The biggest oscillations are obtained for the case when 100% of the outside air passes through the system, which was expected due to the non-iterative time-step sequence of building/system modules execution. Co-simulation further introduces additional inaccuracies, due to the delay of the coupling data<sup>2</sup>. However, as it can be seen, the results of co-simulation still follow the results of mono-simulations.

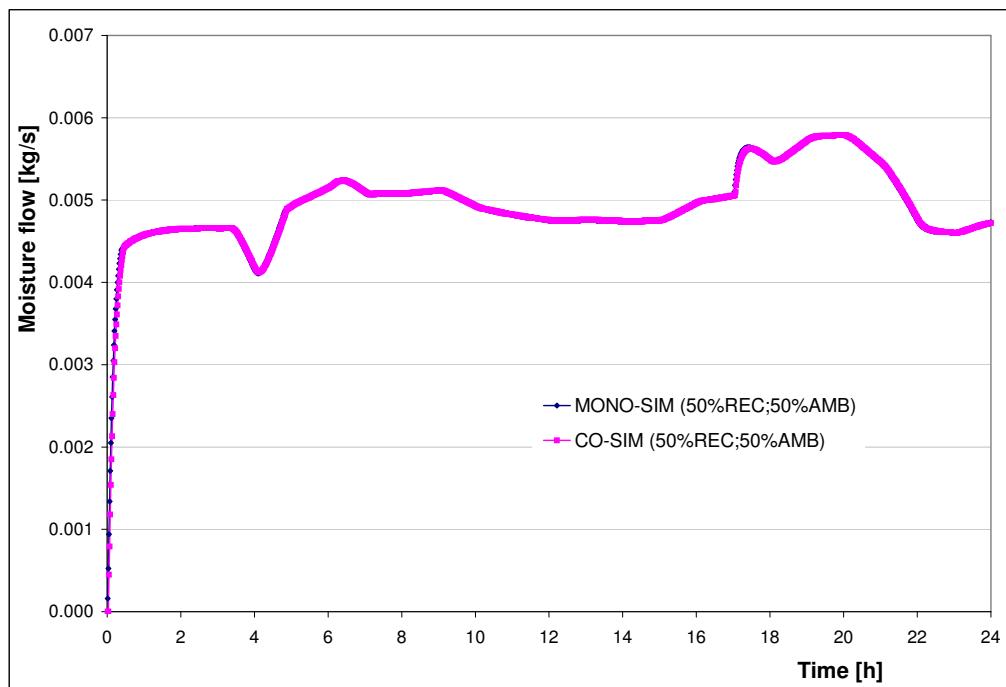
In order to further investigate the co-simulation behavior relative to the value of the coupling data, additional experiments are performed with three times

---

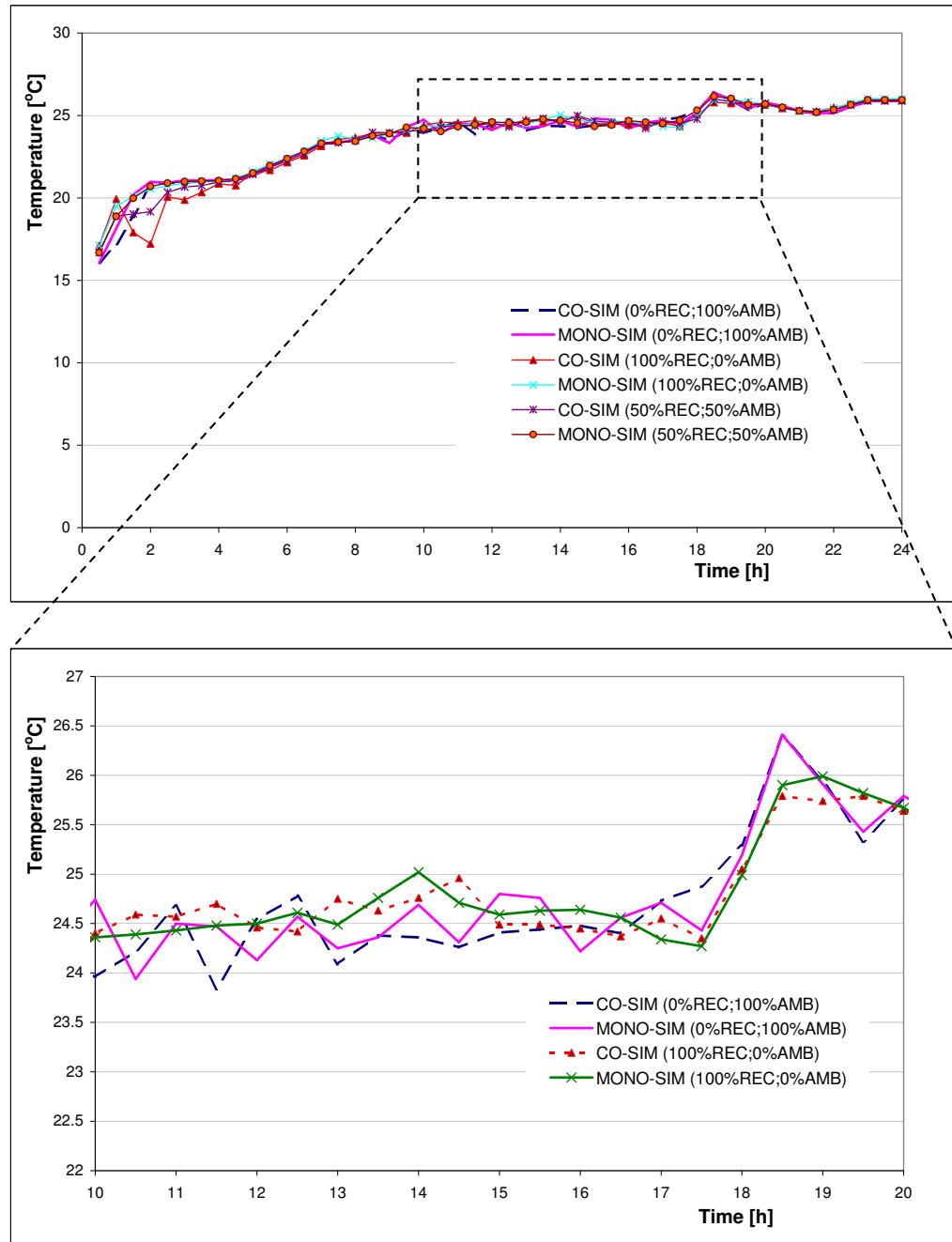
<sup>2</sup>In each iteration between HVAC energy and mass balance systems of equations, the mass balance system of equations is solved prior to the energy balance system of equations. The coupling data exchange takes place only in the first iteration during the calculation of energy balance system of equations. This means that the delayed temperature is used for converting the volume flow rate to the mass rate being used in the mass balance system of equations.



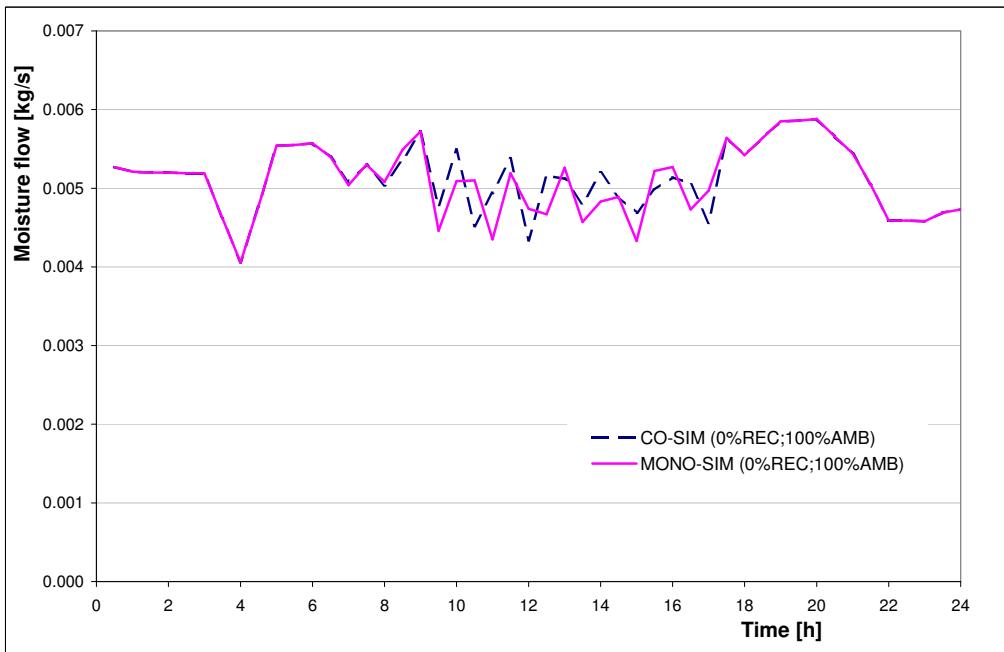
**Fig. 6.13** — Cooling power curves obtained by mono- and co-simulation with  $\Delta t = 1$  min.



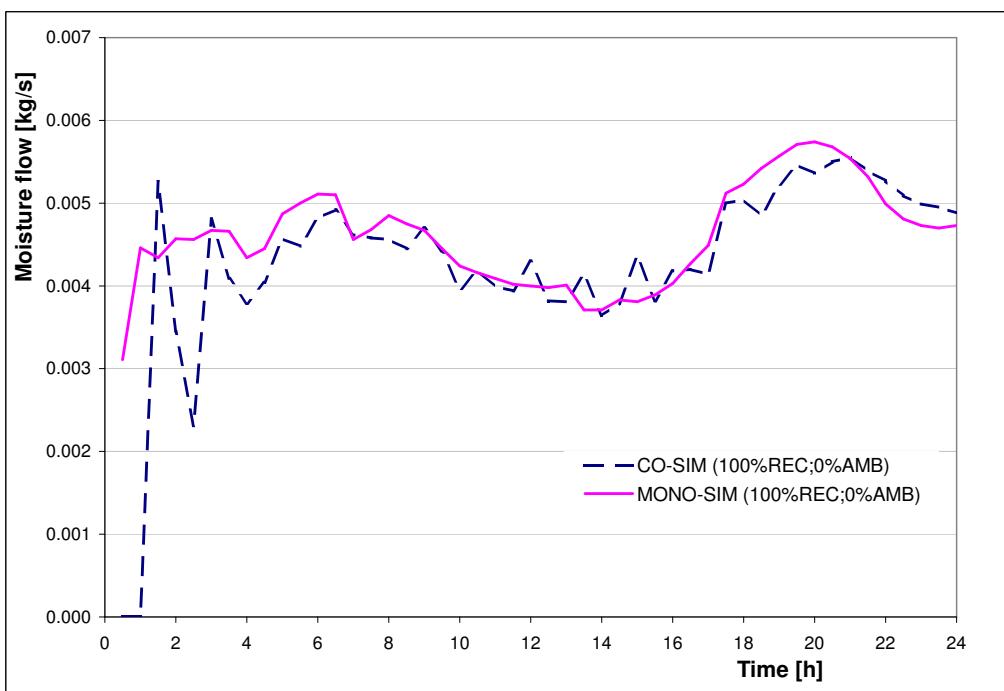
**Fig. 6.14** — Moisture flow after the cooling coil obtained by mono- and co-simulation with  $\Delta t = 1$  min.



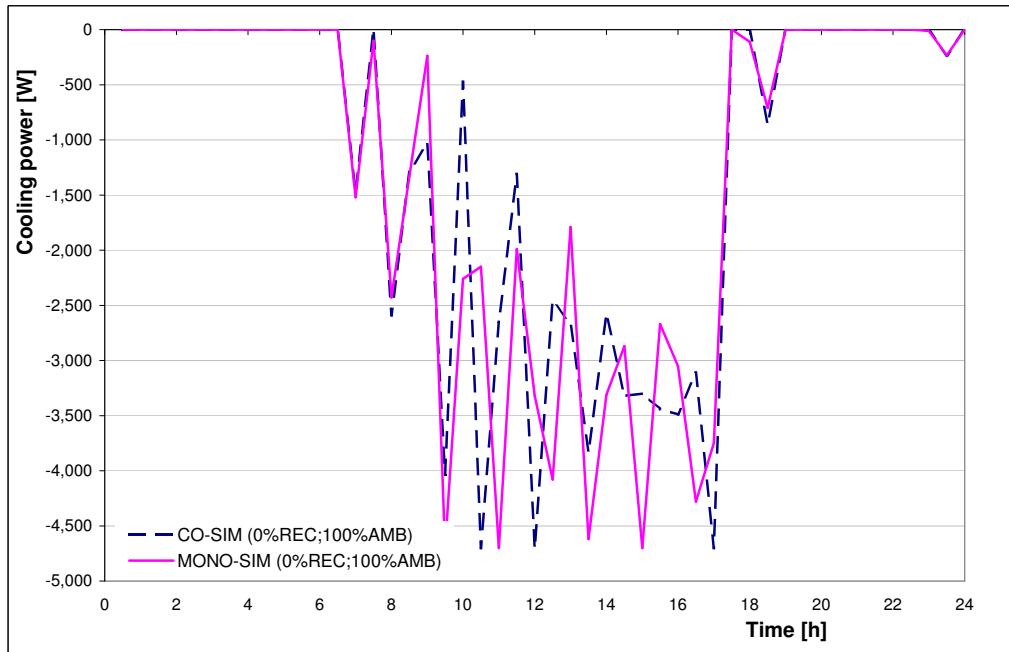
**Fig. 6.15** — Temperature curves obtained by mono- and co-simulation with  $\Delta t = 30$  min.



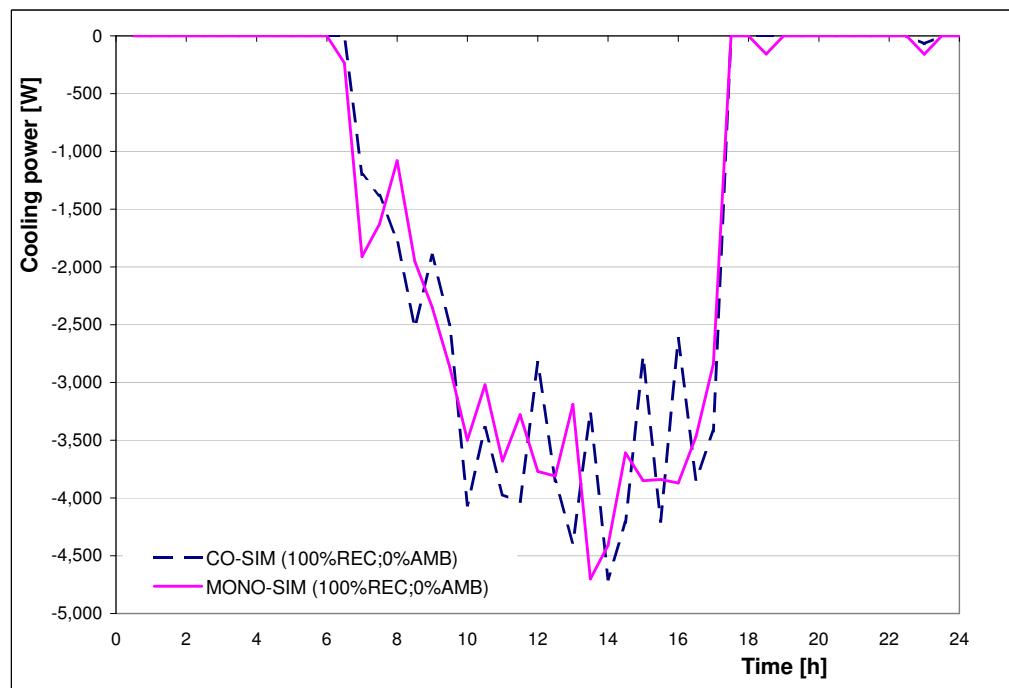
**Fig. 6.16** — Moisture flow after the cooling coil by obtained mono- and co-simulation with  $\Delta t = 30$  min and 100% outside air.



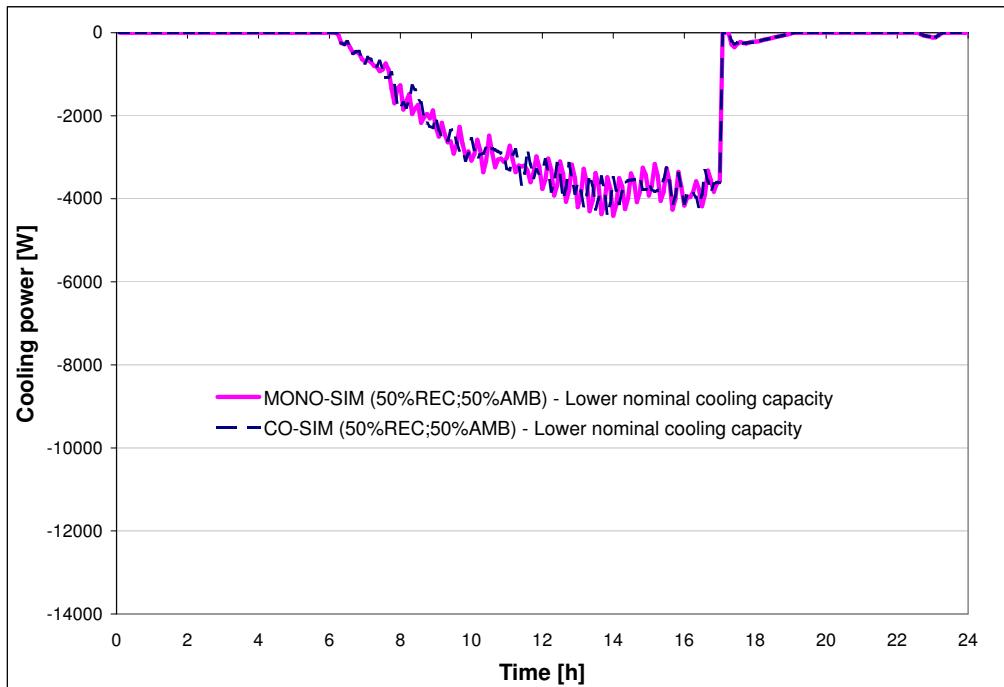
**Fig. 6.17** — Moisture flow after the cooling coil obtained by mono- and co-simulation with  $\Delta t = 30$  min and 100% recirculation air.



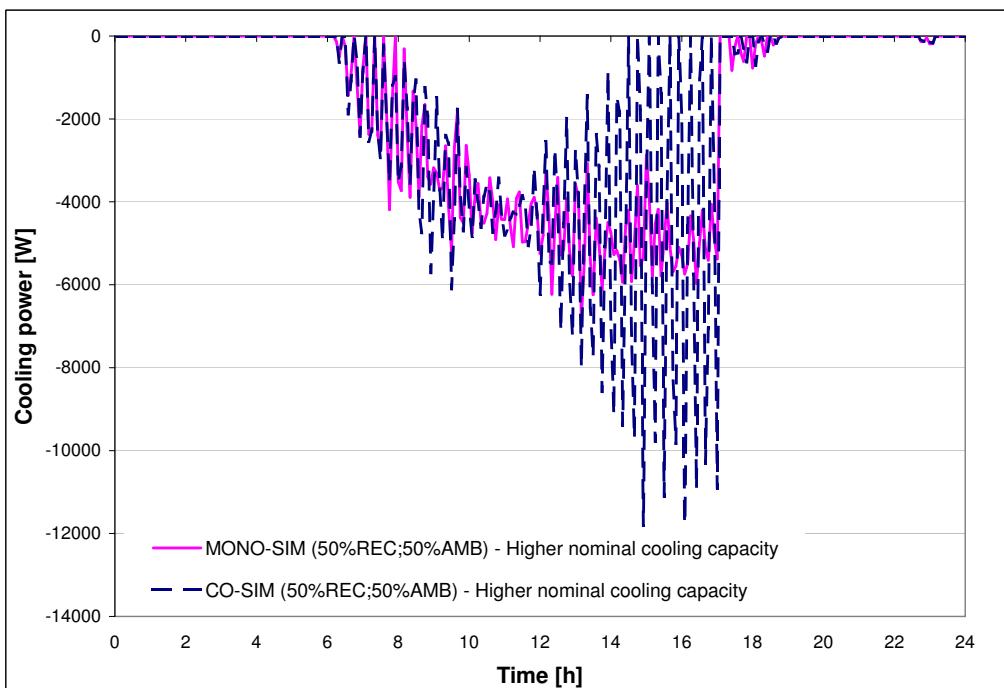
**Fig. 6.18** — Cooling power obtained by mono- and co-simulation with  $\Delta t = 30$  min and 100% outside air.



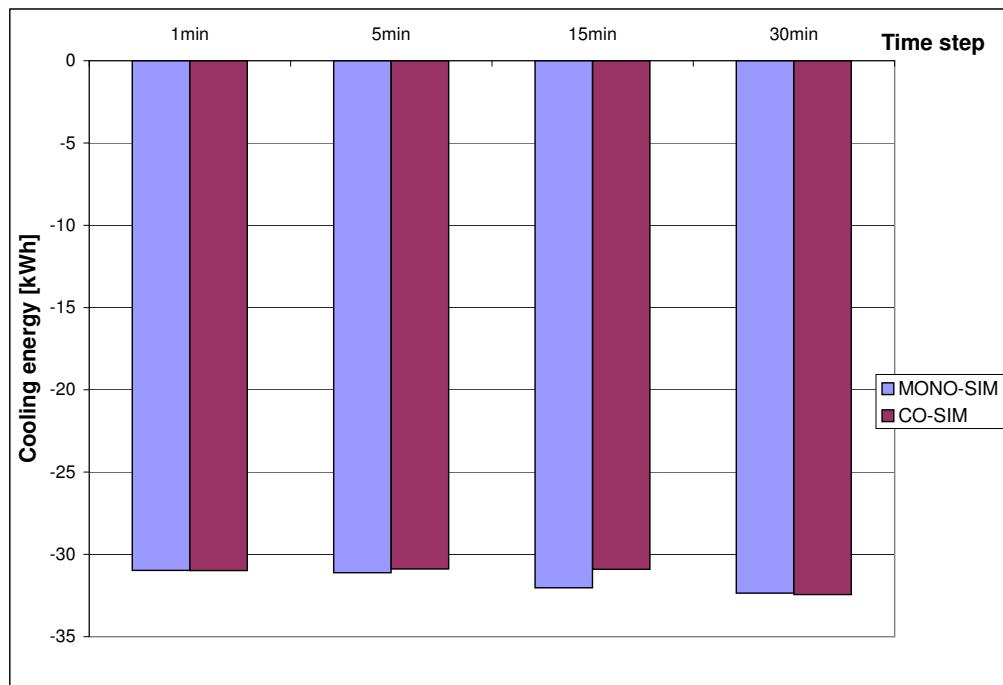
**Fig. 6.19** — Cooling power obtained by mono- and co-simulation with  $\Delta t = 30$  min and 100% recirculation air.



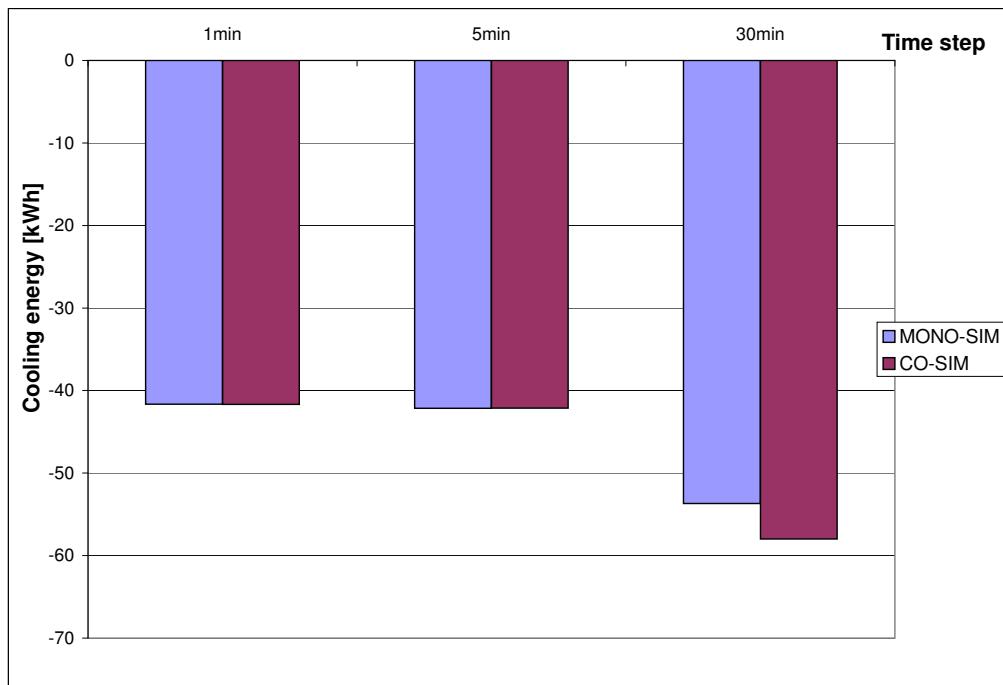
**Fig. 6.20** — Cooling power curves obtained by mono- and co-simulation with  $\Delta t = 5$  min and nominal settings.



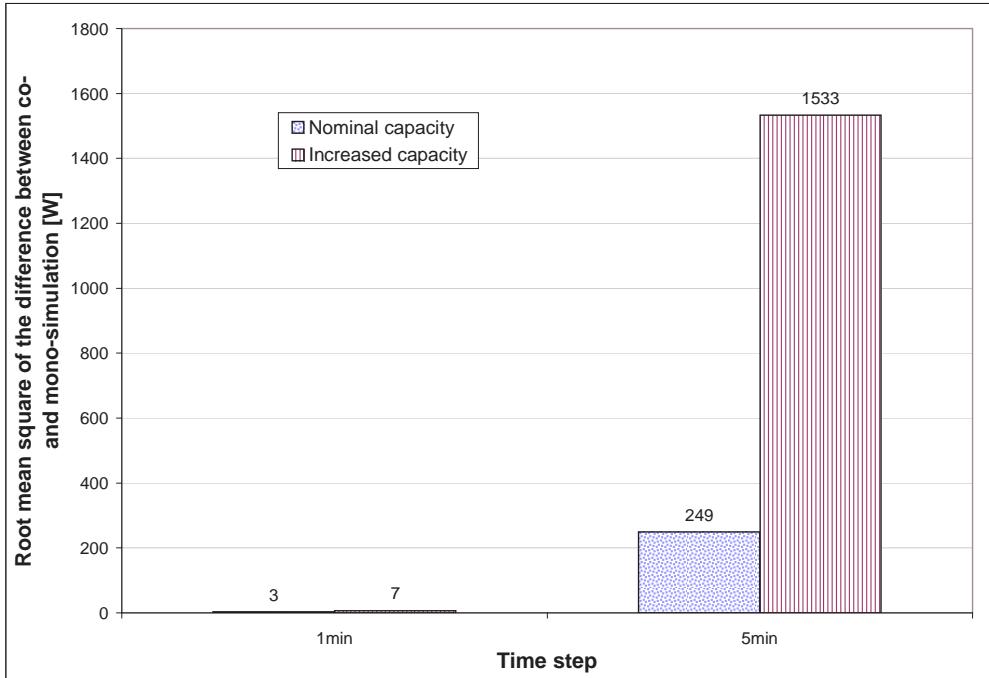
**Fig. 6.21** — Cooling power curves obtained by mono- and co-simulation with  $\Delta t = 5$  min and for the oversized system.



**Fig. 6.22** — Results for cooling energy requirements for the nominal case and mixed air flow.



**Fig. 6.23** — Results for cooling energy requirements for the oversized system.



**Fig. 6.24** — Root mean square of the difference between co- and mono-simulation with three different time steps.

greater volume flow rate. A new values for the maximum cooling capacity is set accordingly to the calculated annual maximums for the increased flow rate, obtained with the climate file for Palermo, Italy. In Figures 6.20 and 6.21 the results for the cooling power are compared. It can be seen how accuracy is altered by co-simulation. The greater values of the delayed coupling data have greater influence on accuracy, as expected.

Figures 6.22 and 6.23 show the cooling energy requirements for co-simulations using different time steps and different system settings. It can be seen that the biggest deviation of co-simulation results compared to results from mono-simulation are for simulations performed with larger time steps and higher volume flow rates.

The integration of the cooling loads resulted in almost equivalent results obtained by co- and mono-simulation performed with  $\Delta t = 5$  min and the increased volume flow rate and cooling capacity. However, a calculation of the error as a root mean square of the difference between co- and mono-simulation results, shown in Figure 6.24, gives a more accurate insights into the error behavior. The difference in mono- and co-simulation results obtained with  $\Delta t = 5$  min is not negligible for the results obtained with bigger time steps and higher volume flow rates.

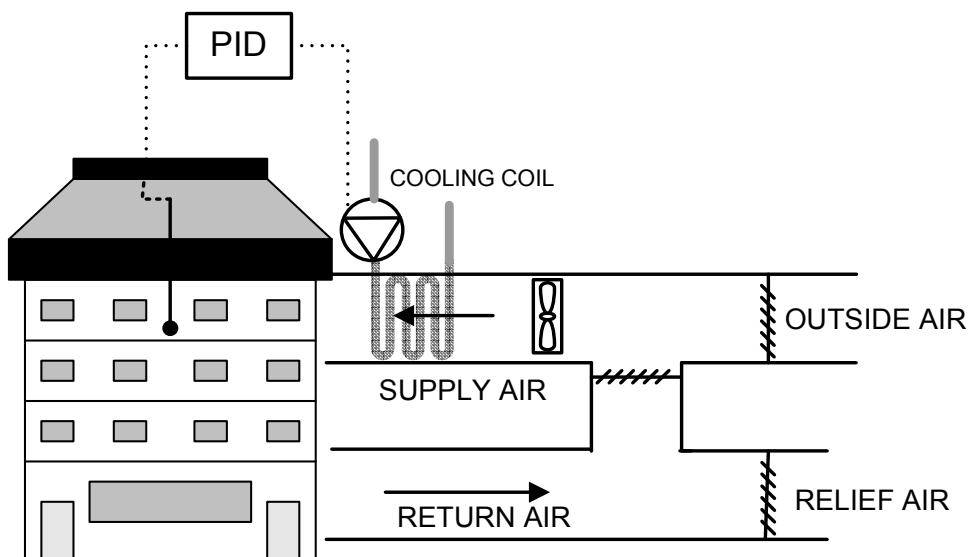
### 6.3.3 Comparison of different co-simulation implementations

EnergyPlus/TRNSYS co-simulation prototype enables the use of both loose and strong coupling strategy as well as co-simulation of intra- and inter- domain de-

composed systems. The implementations are independent and a comparison of results obtained by different implementations can be used to gain further confidence into co-simulation.

### Methodology

For validation purposes, the building and system presented in Figure 6.25 are used. The one-zone building model is built in EnergyPlus and the air system model in TRNSYS.



**Fig. 6.25** — Sketch of the exemplar building used for the validation of the strong coupling.

The system consists of (i) a cooling coil, which is sized to fit the zone load by adjusting the number and dimensions of tubes, fans and rows, (ii) a constant flow fan and (iii) a variable flow cooling water pump. The water mass flow rate is proportionally controlled to maintain the zone temperature set point, with the lower limit of 24°C and the upper limit of 26°C. The inlet cooling water temperature is kept constant at 6°C. The system is operating from 7h to 19h. The nominal value of the water flow rate was varied from 720kg/h to 1800kg/h. The first nominal value corresponds to the maximal cooling demand for the simulated period, the second value is used to demonstrate the effect of an oversized system on stability of the co-simulation. The weather data for Denver, Colorado is used and the simulation period is set to two working days from 1<sup>st</sup> to 2<sup>nd</sup> of August.

Two different models are built in EnergyPlus. For validation of co-simulation using inter-domain decomposition, only the building is modeled in EnergyPlus. For validation of co-simulation using intra-domain decomposition, the EnergyPlus model, besides the building, includes also the ducting from the zone to the cooling coil, modeled in TRNSYS, and from the cooling coil the zone is used.

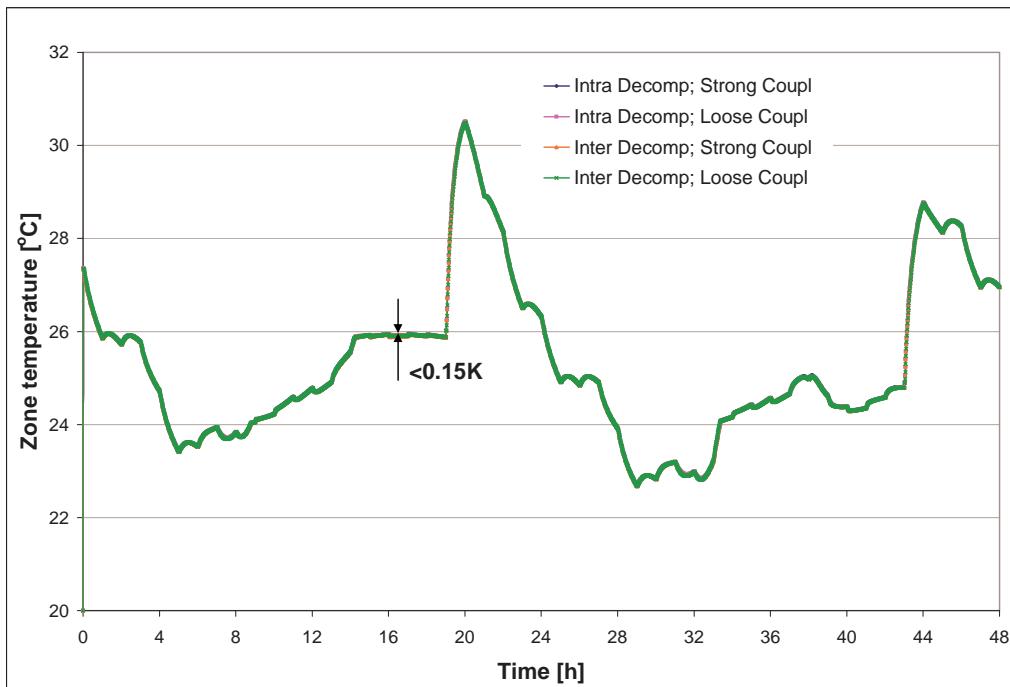
The simulations are done using different building and HVAC co-simulation approaches (in terms of system decomposition and coupling strategies). The resulting zone temperatures, and cooling water flow rates are compared.

## Results

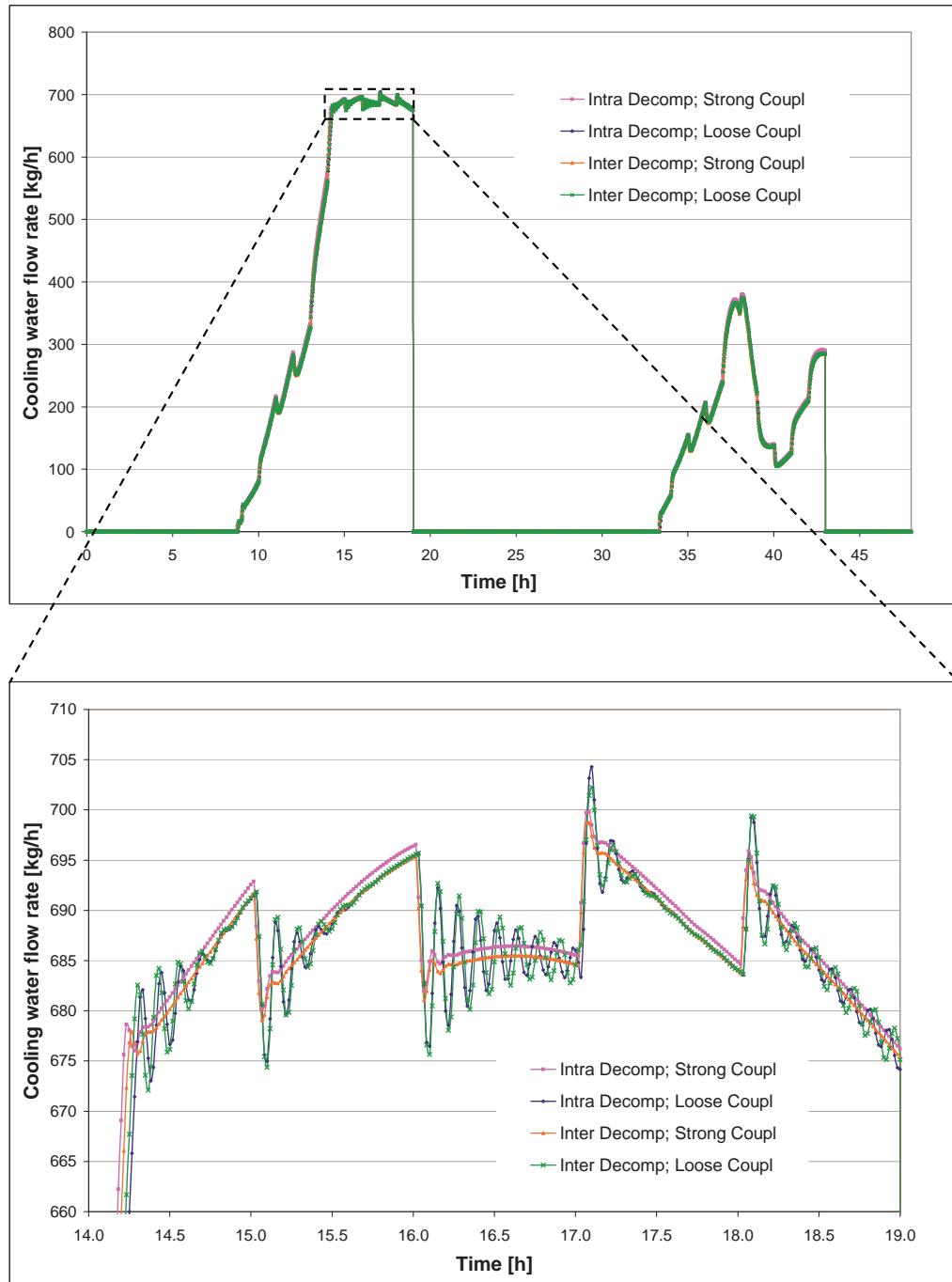
For the first comparison the time step of 1 min is used. Figures 6.26 and 6.27 show that the zone temperature and the cooling water flow rate obtained with all four combinations of the system decompositions and the coupling strategies:

- 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

are almost identical. There is a small difference in the results between the loosely-coupled and the strongly-coupled co-simulations, which can be seen from the magnified view of the graph shown in the same figure. In the loosely coupled co-simulation, the oscillations of the water flow rate (as well as the oscillations of the zone temperature) occur due to the use of the lagged coupling data. The results obtained by the strong coupling and a time step of 1 min are used as the reference for further comparisons.



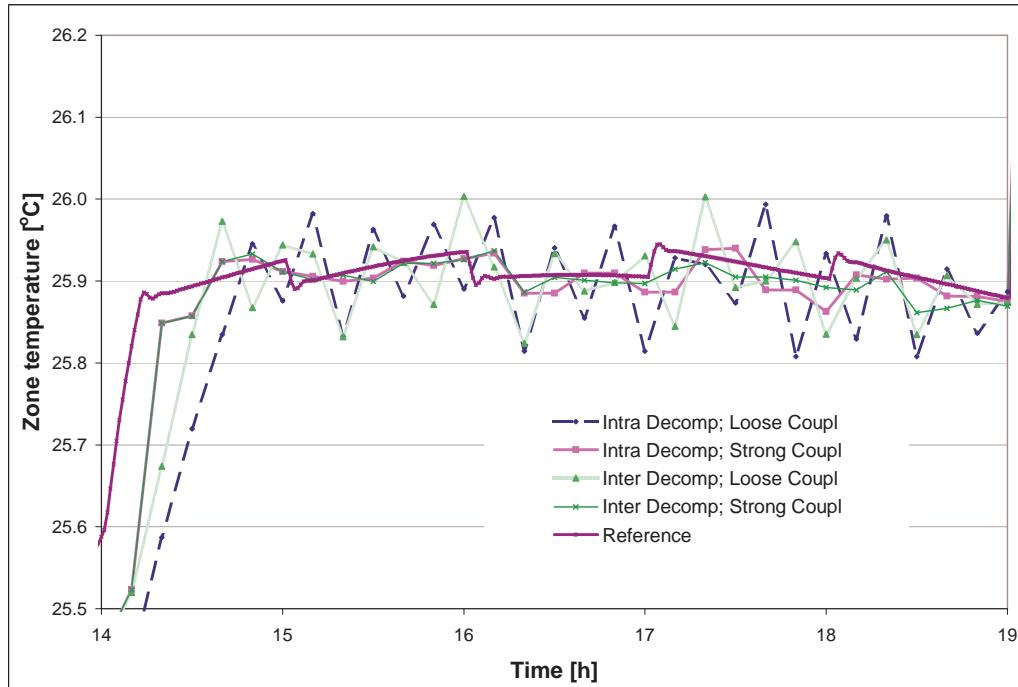
**Fig. 6.26** — Zone temperature, with  $\Delta t = 1 \text{ min}$  and  $\dot{m}_{\text{water,nominal}} = 720 \text{ kg/h}$ ; for all co-simulation approaches.



**Fig. 6.27** — Cooling water flow rate, with  $\Delta t = 1 \text{ min}$  and  $\dot{m}_{\text{water,nominal}} = 720 \text{ kg/h}$ ; for all co-simulation approaches.

With  $\Delta t = 10\text{min}$ , due to the adequately sized system, the difference between the loose and the strong coupling, although bigger than with  $\Delta t = 1\text{min}$ , is still small as shown in Figures 6.28 and 6.29. The deviations of the zone temperature are within  $\pm 0.3^\circ\text{C}$  and of the cooling water mass flow rate within  $\pm 100\text{kg/h}$ . The biggest deviations result from the loosely coupled co-simulation.

The difference between the inter- and the intra-domain system decompositions is due to the nature of the coupling implementation already discussed.

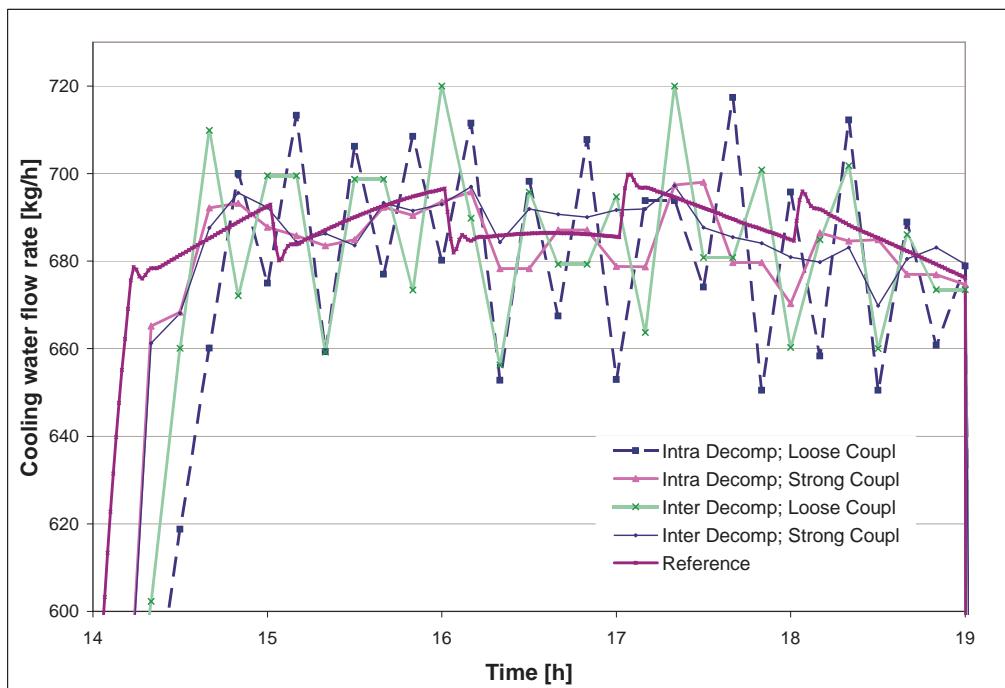


**Fig. 6.28** — Zone temperature, with  $\Delta t = 10\text{ min}$  and  $\dot{m}_{\text{water,nominal}} = 720\text{ kg/h}$ ; for all co-simulation approaches.

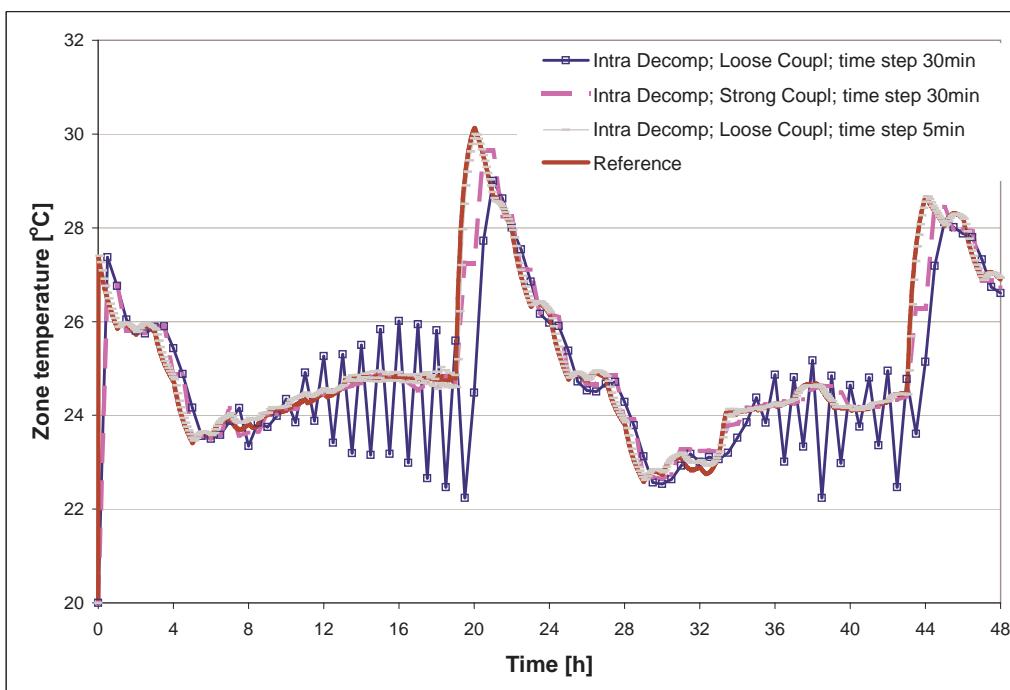
Co-simulation employing the strong coupling strategy with larger time steps showed convergence problems. The number of maximum allowed iterations (set to 20) was reached in almost every time step. To enhance the convergence of the iterations, relaxation of the zone temperature was implemented.

Results, obtained by co-simulation that implements strong coupling strategy and the relaxation of the iterating variable, show good agreement with the reference results, even if the larger time steps are implemented and/or if the system is oversized. Figure 6.30 shows how the results from the loosely-coupled co-simulation correspond to the strongly-coupled co-simulation results using the relaxation of the zone temperature. The solution of the strongly-coupled co-simulation with the relaxation follows the reference curve more closely than the solution obtained by the loosely coupled co-simulation using the same coupling time step.

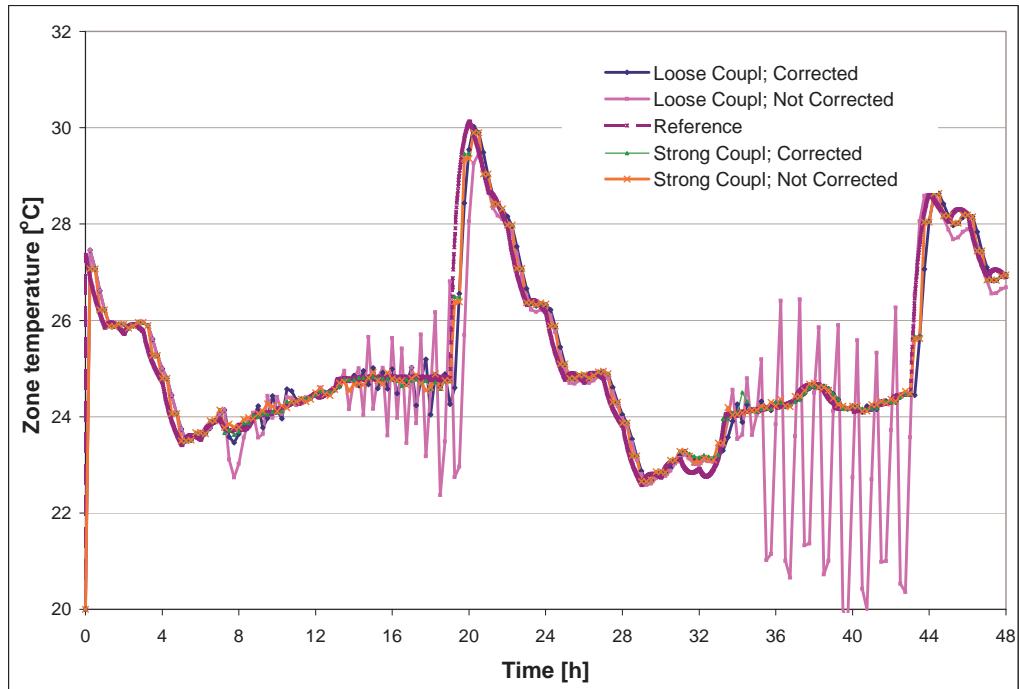
For the co-simulation implementing the loose coupling strategy, the use of correction terms (see §5.3.1) significantly affects the results as shown in Figure 6.31. The results were obtained with a coupling time step of 15 min. Of course, the



**Fig. 6.29** — Cooling water flow rate, with  $\Delta t = 10 \text{ min}$  and  $\dot{m}_{\text{water,nominal}} = 720 \text{ kg/h}$ ; for all co-simulation approaches.



**Fig. 6.30** — Zone temperature, for  $\dot{m}_{\text{water,nominal}} = 800 \text{ kg/h}$ ; intra-domain system decomposition, using loose ( $\Delta t = 5$  and 30 min) and strong (with relaxation) ( $\Delta t = 30 \text{ min}$ ) coupling.



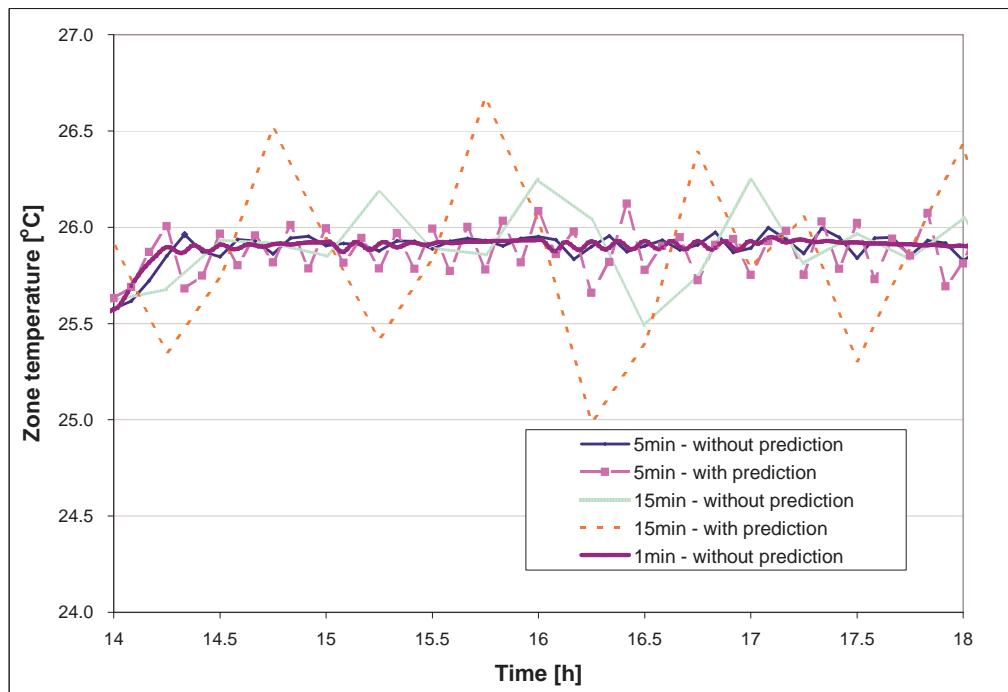
**Fig. 6.31** — Zone temperature, with  $\Delta t = 15 \text{ min}$  and  $m_{\text{water,nominal}} = 1800 \text{ kg/h}$ ; for inter-domain sys. dec., using loose and strong (with relaxation) coupling, with and without correction.

difference between corrected and not corrected results would depend on the coupling time step used in co-simulation.

The use of a first-order predictor was also investigated. In the above simulations, TRNSYS performed its calculations based on the one-time-step delayed coupling data (zero-order predictor). For co-simulation using small coupling time steps, the introduced numerical error was negligible, but the error was larger for co-simulation using larger coupling time steps. To improve the accuracy of the loosely coupled co-simulation, a first-order predictor that predicts future coupling data sent from EnergyPlus to TRNSYS is implemented. The predictor increases the accuracy in situations where the changes of the coupling data are smooth (see §4.7.4), which due to the control actions and finite value of the time step is not always realistic in the realistic applications. For the tested case, due to the delays of the coupling data, the oscillatory behavior of the zone temperature is already present in the results of co-simulation using loose coupling and small time steps. Thus, in this case, the first-order predictor overestimates the future value of the coupling data, which leads to additional inaccuracies, as shown in Figure 6.32.

Figure 6.33 compares the accuracies of co-simulation with and without predictors and with various coupling time steps against the reference results. In this case, with any time step, co-simulation performs better if no prediction of coupling data is used.

The efficiency of co-simulation was not considered when making decisions about different aspects of co-simulation implementation. However, a relative



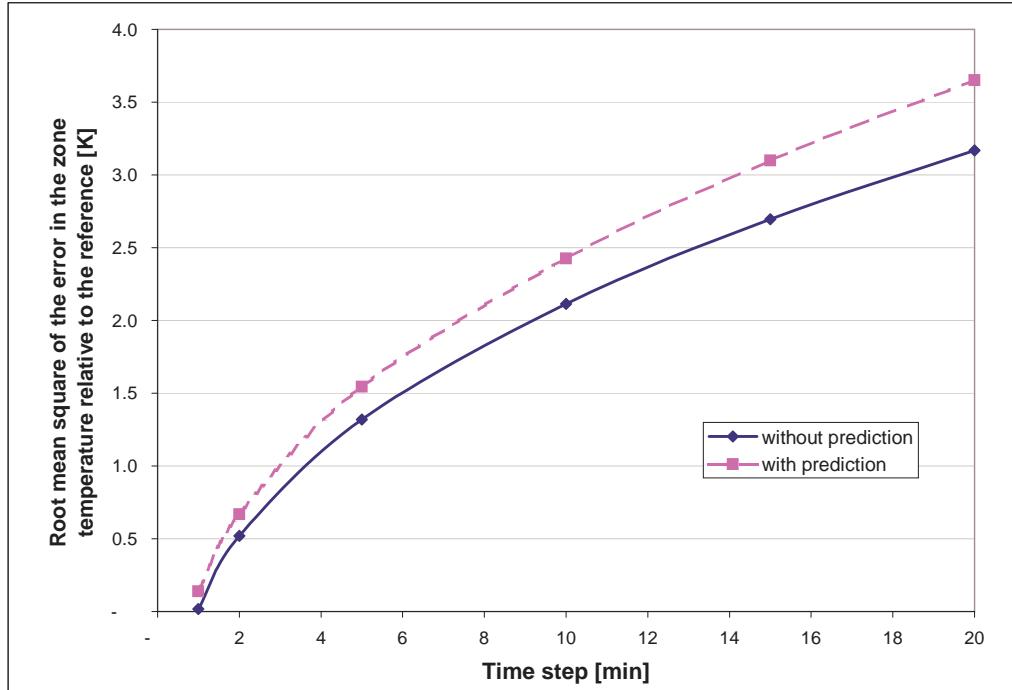
**Fig. 6.32** — Zone temperature, with  $\Delta t = 5$  and 15 min and  $\dot{m}_{\text{water,nominal}} = 720 \text{ kg/h}$ ; for inter-domain sys. dec., using loose coupling, with and without prediction of/ the coupling data.

comparison of efficiency of different co-simulation approaches will now be discussed.

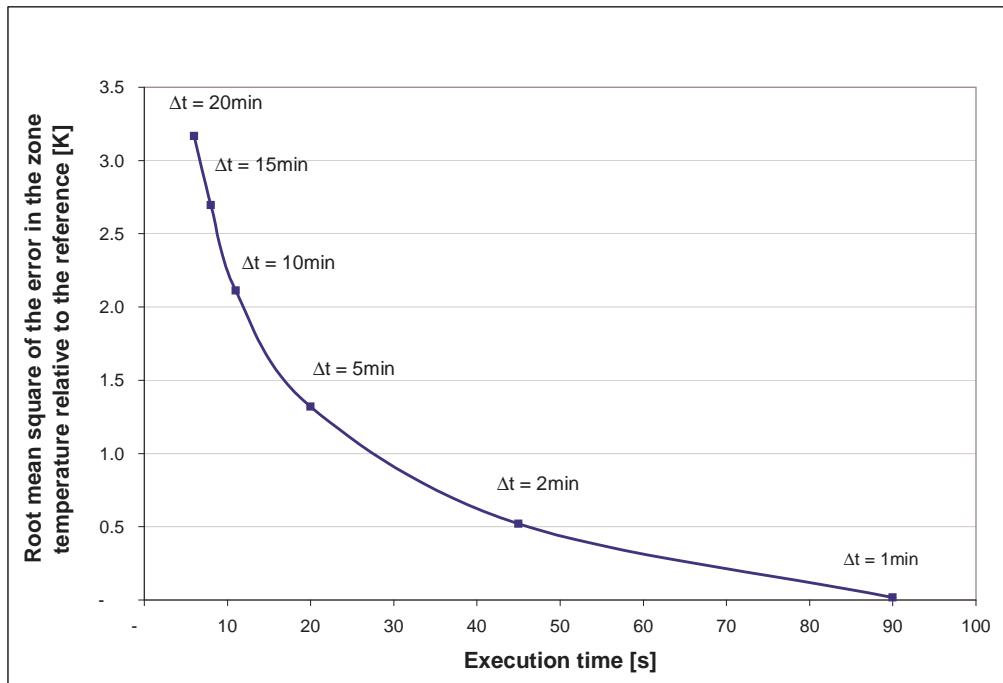
The computation time differs from one approach to another. The longest execution time was required by the strong coupling approach. Also, the intra-domain system decomposition results in a longer execution time, since it requires more extensive computations in EnergyPlus than the inter-domain decomposition does. The difference is due to the nested iteration loop in EnergyPlus (shown in Figure 5.3). The loop is only used in the case of the intra-domain decomposition approach for the iterative solution of the HVAC system related equations in EnergyPlus. As long as the `FirstHVACIteration` flag is set to `TRUE`, EnergyPlus communicates with TRNSYS in each pass through the nested loop. In the case of inter-domain system decomposition the whole HVAC system is modeled in TRNSYS and this nested loop is skipped.

The difference in the execution time between the two loosely-coupled approaches was small. The shortest execution time was required for the inter-domain decomposition using the loose coupling, as it does not solve the HVAC system in EnergyPlus. For this case study, the wall-clock time required for the execution of co-simulation using the intra-domain decomposition with the strong coupling and a simulation time step of 30 min was 13 s. This is much longer than the 7 s wall-clock time required for the co-simulation using the intra-domain decomposition with the loose coupling and a simulation time step of 5min. The use of the loose coupling with shorter time steps also provided higher accuracy, as shown in Figure 6.30. The co-simulations were run on Windows XP, using one

CPU with 2.8GHz and 3.25GB RAM.



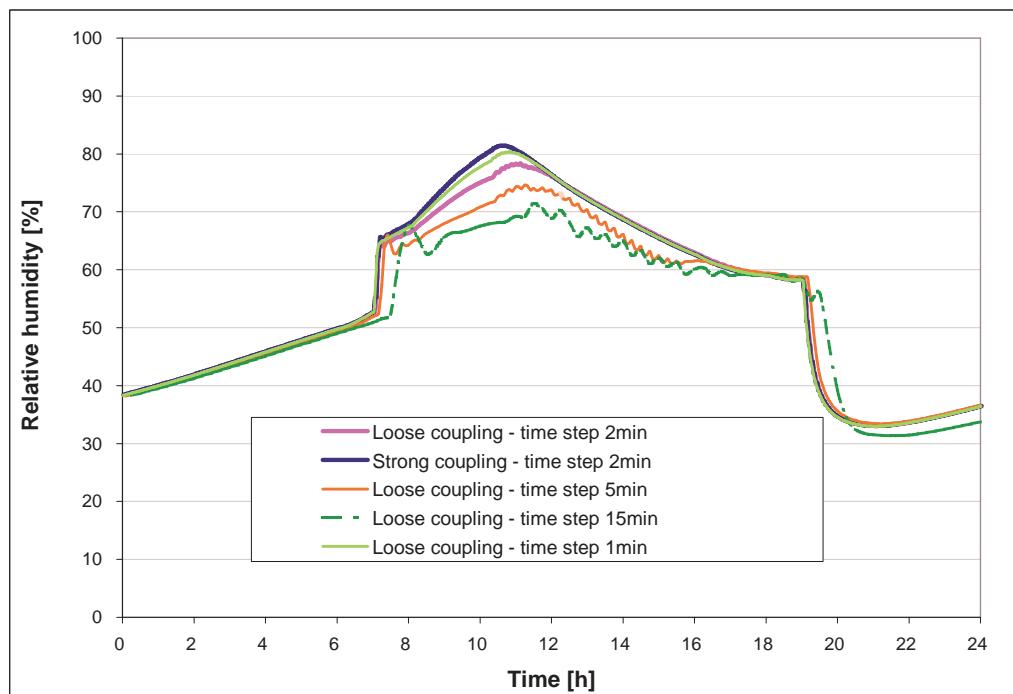
**Fig. 6.33** — Accuracy vs. simulation time step.



**Fig. 6.34** — Accuracy vs. execution efficiency.

Figure 6.33 compares the accuracies of co-simulation with and without predictors

To comment on accuracy vs. efficiency of co-simulation, the error calculated as a root mean square of the difference in the zone temperature and the reference zone temperature obtained with the strong coupling approach and a time step of 1 min, is used. The simulations are performed using inter-domain decomposition and loose coupling strategy. The error is plotted against the required execution time (wall-clock time from the time point when shared memory has been initiated to the time point when the calculations have been finished) and the graph is shown in Figure 6.34. To plot this graph, the simulations have been performed using another machine (1.83GHz and 0.99GB RAM).

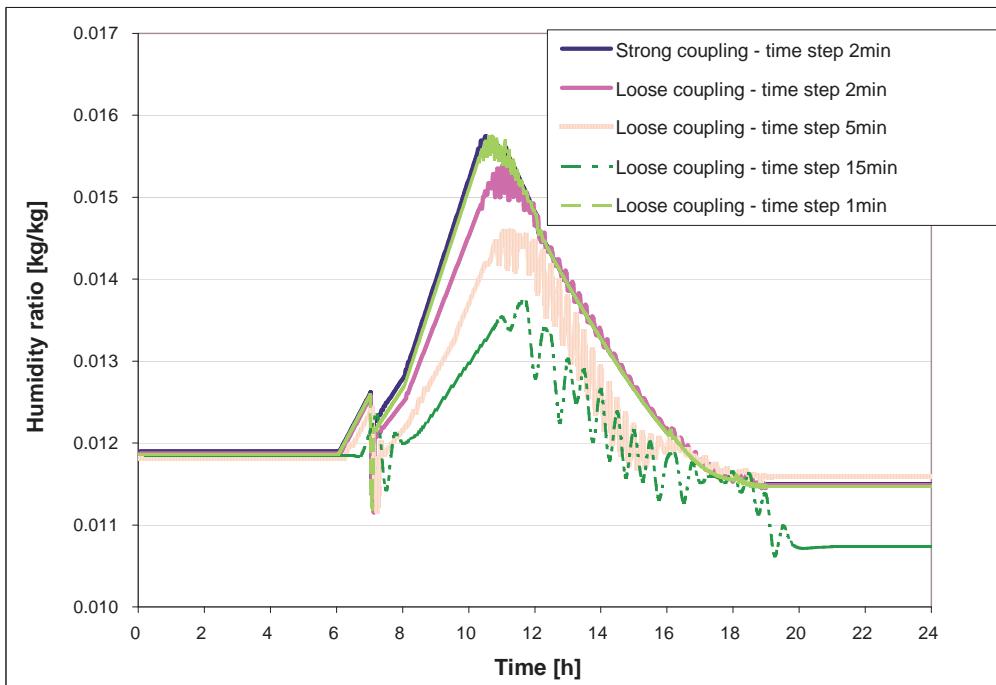


**Fig. 6.35 — Zone relative humidity.**

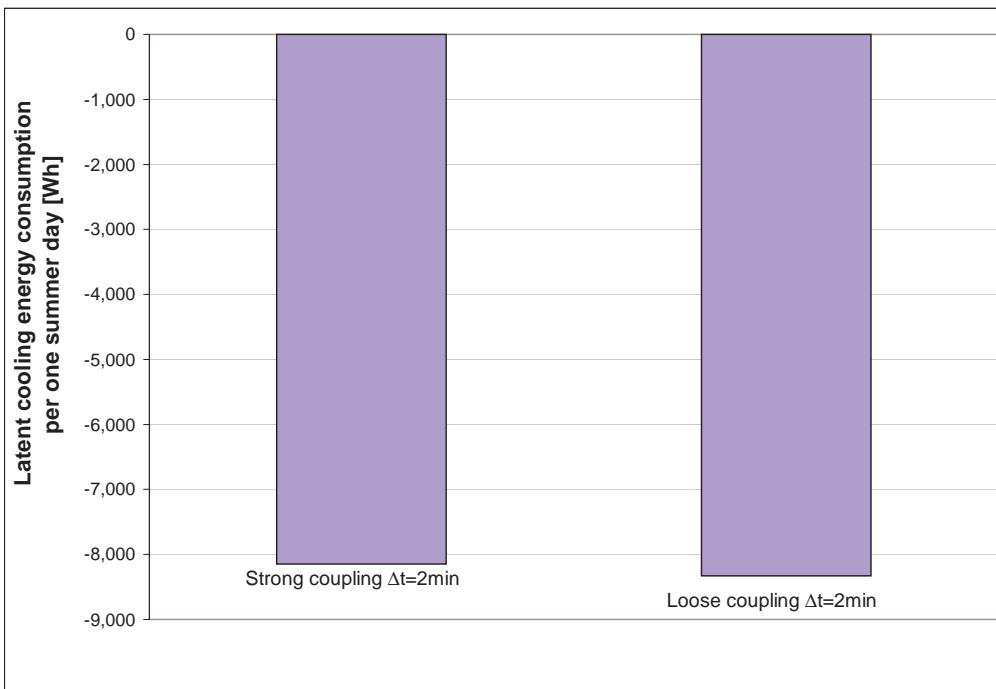
The influence of non-linearities in a co-simulation model were investigated by looking into the accuracy of prediction of vapor condensation in the cooling coil. For that purpose, the internal latent gains are rescheduled to be active only during the system operational hours. Additional simulations, using the weather data for Nashville, Tennessee, are performed for the intra-domain system decomposition using the loose coupling with different simulation time steps. A simulation using the strong coupling strategy is performed using a time step of 2min and the results are used as a reference for comparison.

The resulting zone relative humidity and the humidity ratio at the outlet of the cooling coil are shown in Figures 6.35 and 6.36, respectively.

Due to the non-linearities between mass and energy balance system of equations, involved in a condensation process, the influence of the simulation time step size on accuracy of the loosely coupled co-simulation is evident even at the high coupling frequencies. However, using a small time step, e.g., 1 min the inaccuracies caused by the delayed coupling data can be lowered.



**Fig. 6.36** — Humidity ratio at the outlet of the cooling coil.



**Fig. 6.37** — Integrated latent cooling power over a simulated summer day.

For the results obtained with  $\Delta t = 2\text{min}$  the difference in the integrated latent cooling power over the simulated day is 2.2%. This is illustrated in Figure 6.37. The absolute difference in the integrated results is  $180\text{Wh}$  and the error calculated as a root mean square of the difference in the cooling power obtained by loose and strong coupling with a time step of 2 min is 255 W.

## 6.4 Discussion and conclusions

To first establish confidence in co-simulation prototypes, an HVAC BESTEST case was used and results from co-simulation were compared against results obtained by other BPS tools in mono-simulation. The prototypes were further validated using a non-standard two-step procedure. First, the co-simulation, implementing the loose coupling strategy, was validated against the mono-simulation. In the second validation step, different co-simulation implementations were compared with each other.

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 same quality as the mono-simulation. However, in some applications, e.g., when non-linearities highly influence the solution, to preserve the same accuracy, either a loosely coupled co-simulation should be performed with small coupling time steps, or a strongly coupled co-simulation should be used.

From the execution efficiency point of view, the shortest execution time is obtained for the loosely coupled co-simulation. However, the accuracy of the results is decreased with an increase of the time step. Using strong coupling with relaxation, stable and accurate results are achieved even with the greater time steps, but the efficiency of the simulation execution is low. In summary, with respect to both (i) the execution time (defined by the specifics of implementation of strong and loose coupling), and (ii) the complexity of its implementation, the loose coupling strategy (using small time steps) is always preferred to the strong coupling strategy.

To improve accuracy of the loose coupling strategy at lower coupling frequencies, an extrapolation of the delayed coupling data was suggested, but since the changes in the coupling data were not smooth, the tested first-order predictor did not improve accuracy of the loosely coupled co-simulation in the tested case.

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 §5.3.1) improves accuracy of co-simulation of a system decomposed between the HVAC and building domains.



# 7

## Case studies

CO-SIMULATION offers great flexibility in integrating the fragmented developments by bringing together different tools. Cases that particularly benefit from the approach are those in which either:

- not all required modeling features are available in one single simulator, or
- a model of a novel (sub)system feature, not available in any of the state of the art domain tools, needs to be prototyped.

With regards to the former, co-simulation can be used to couple different-domain simulators, simulators that model the same domain in different abstraction levels, or simulators that complement each other with the available (sub)system models. When a model of a (new) technology is not available in the state of the art BPS domain tools, or its modeling abstraction does not meet a modeler's requirements, co-simulation can be used to integrate the state of the art tools with the prototype model developed in, for example, an equation-based tool.

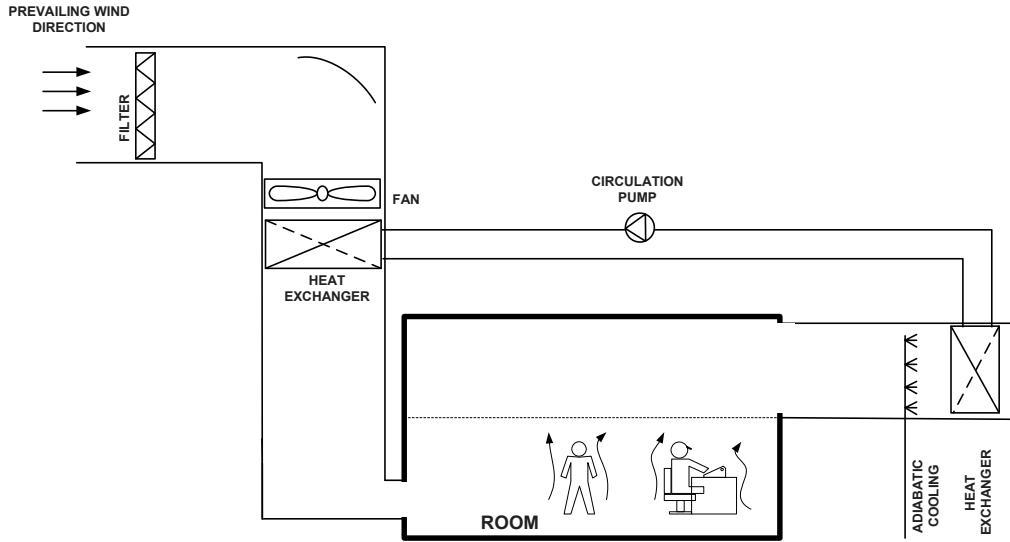
This chapter presents a few case studies in order to illustrate some potential applications of the co-simulation approach and to demonstrate its benefits.

### 7.1 Hybrid ventilation with evaporative cooling and run-around heat recovery

#### 7.1.1 System description

A system topology, as shown in Figure 7.1, is suggested [Wetter 2008] for a building in a windy moderate climate. There is only evaporative adiabatic cooling available and no heating, except what can be provided by a heat recovery system. Dynamic wind pressure is used to drive the air flow through the system, whenever the wind speed in the direction of the air intake is sufficiently high.

The control sequence is as follows. If the air speed in the duct  $v_1$  is  $v_1 < v_{\text{set}}$  for 2 min, the fan is switched on and its revolution is modulated to maintain  $v_1 = v_{\text{set}}$ . If  $\int_0^\delta v_1 dt > 1.1 v_{\text{set}} \delta$  for  $\delta = 2$  min, the fan is switched off and locked for 10 min. If there is a need for cooling (proportional controller is used with a cooling set point temperature set to  $T_{\text{zone, set}} = 26^\circ\text{C}$  and throttling range set to 2K) and the ambient temperature is not low enough to provide free cooling ( $T_{\text{ambient}} < 20^\circ\text{C}$ ),



**Fig. 7.1** — Hybrid ventilation with evaporative cooling and run-around heat recovery [Wetter 2008].

and if  $\int_0^\delta v_1 dt > 0.9 v_{\text{set}} \delta$  (i.e., there is sufficient air flow), then water is sprayed into the exhaust air stream to provide the adiabatic cooling. The water cooled by the exhaust air heat exchanger is circulated to the supply air heat exchanger by switching the pump on whenever water is sprayed into the exhaust air stream. If there is a need for heating (proportional controller is used with a heating set point temperature set to  $T_{\text{zone, set}} = 21^\circ\text{C}$  and throttling range set to 2K), then the heat recovery is activated by switching on the circulation pump. Also, to avoid the low temperatures near the floor area, the circulation pump is activated if the supply air temperature drops below  $17^\circ\text{C}$ . No additional heating is provided to the room.

To achieve a sufficient fresh air supply with reasonable wind speeds, the pressure drop in the system should be decreased as much as possible. The pressure can be lowered by: (i) decreasing the air velocities in the system, (ii) using low pressure and/or oversized components, (iii) applying a larger duct diameter, (iv) applying less fittings as possible, and (v) employing distribution system which does not require high kinetic energy, like displacement ventilation.

Low-face-velocity coils with a low row count and pressure drop of 50 Pa were successfully applied, as reported in [LBNL 2005]. The system pressure drop as low as 150 Pa could be achieved by implementing the low-face-velocity coils and a low pressure drop filter. The pressure drop of 150 Pa corresponds to the dynamic pressure introduced by the direct wind of 17 m/s, with wind pressure coefficient of 0.9. When the wind velocity is not sufficient  $v_{\text{wind}} < 17 \text{ m/s}$ , the fan increases the pressure head in the system to maintain a sufficient fresh air supply following the described control sequence.

### 7.1.2 Model description

Simulation model of the above described system is used to test the control sequence and to estimate the thermal comfort and the energy use.

#### Building model

The layout of an office space, conditioned by the hybrid ventilation system with runaround heat recovery, is given in Figure 7.2. The office space consists of a corridor and two zones, separated by a transparent light construction. The electrical lighting in the zones is controlled according to daylight illumination levels at two points in the occupied space. A minimum electrical lighting level during the office working hours is always kept above zero to ensure constant lighting of the corridor. The ventilation air is evenly distributed in all three zones.

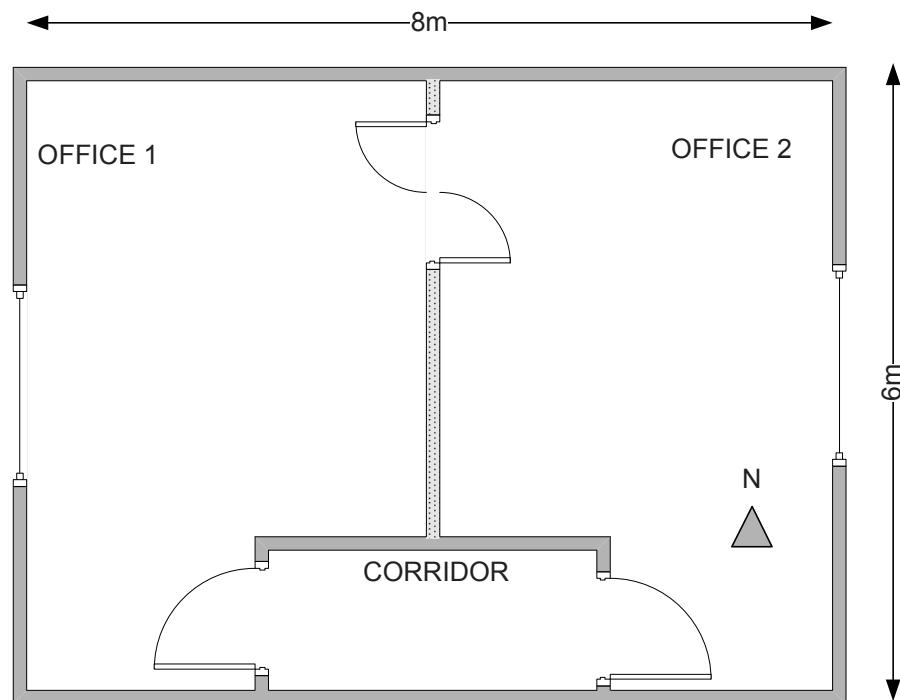


Fig. 7.2 — Office layout.

Since, the zones are separated with the transparent light construction and the ventilation air is evenly distributed in all zones, the office is modeled as a single thermal zone in EnergyPlus. Zone internal gains are assumed to be low. The design electrical lighting capacity is  $10 \text{ W/m}^2$ . The electrical lighting is controlled, using a dimming control feature available in EnergyPlus. The design levels of other internal gains, from the equipment and the people are set to  $10 \text{ W/m}^2$  and  $9 \text{ W/m}^2$ , respectively, and are scheduled according to the zone occupancy throughout a working day.

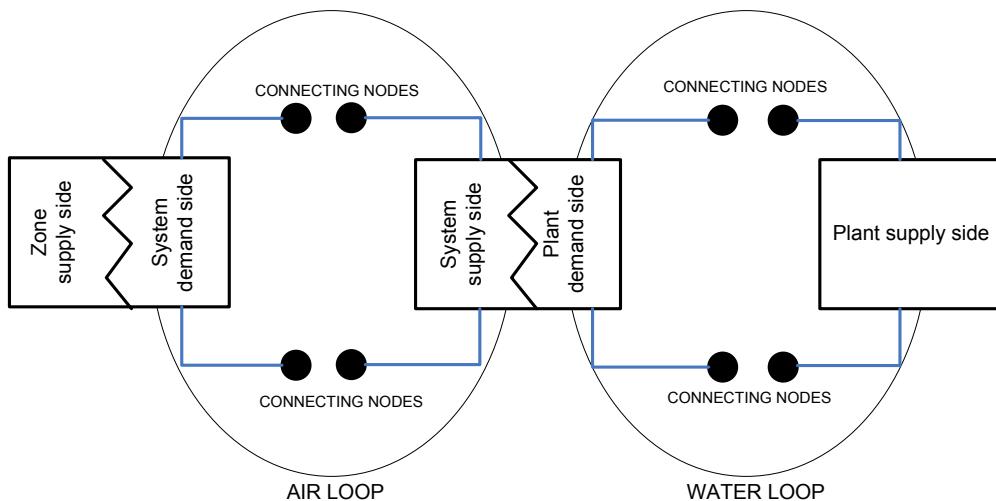
A simplified displacement ventilation model for heat transfer and vertical temperature profile prediction, available in EnergyPlus, is used. The model extends the fully-mixed room air approximation to a three node approach, with

the purpose of obtaining a first order precision model for vertical temperature profiles in displacement ventilation systems without a need to model zonal air flow model. The simplified displacement ventilation model is applicable to this specific case, since the zones are served by a low velocity floor-level displacement ventilation air distribution system and the dominant sources of heat gain are from localized sources located in the occupied part of the room.

### System model

The hybrid ventilation system with runarround heat recovery in Figure 7.1 cannot be modeled in EnergyPlus, due to the following.

EnergyPlus models an HVAC system as a series of functional elements connected by fluid loops as shown in Figure 7.3. The loops are divided into supply and demand sides. As shown, the plant supply side cannot be connected to the air loop, which is the case in the system implementing runarround heat recovery. The coil in the exhaust duct is a plant supply side component which is a part of the air loop as well.



**Fig. 7.3 — Fluid loops in EnergyPlus.**

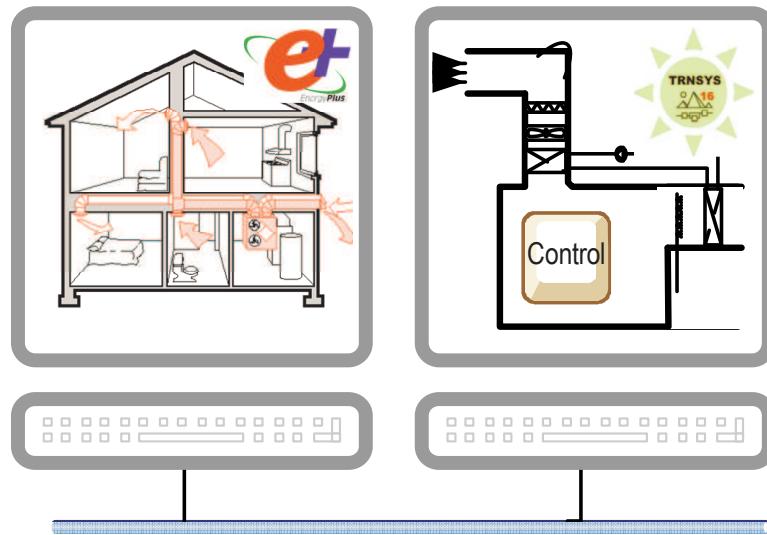
Thus, to enable the design analysis in an integrated fashion and exploitation of the building side modeling features in EnergyPlus, the system is modeled in TRNSYS and the co-simulation approach is used to integrate the distributed models.

The system control is modeled using a prototype model of a delay and a sequence of equations in TRNSYS.

### Co-simulation model

The co-simulation model is made as illustrated in Figure 7.4.

EnergyPlus program logic activates the simplified displacement ventilation model only if there is an air supply to the conditioned zone. Thereby, the intra-domain system decomposition is implemented in this case study. The whole sys-



**Fig. 7.4** — Illustration of co-simulation using EnergyPlus and TRNSYS.

tem but the air supply is modeled in TRNSYS. The loose coupling strategy is implemented with the coupling time step of 1min. Since the zone air temperature is used to control the system, the control is distributed and the sensed zone temperature<sup>1</sup> is sent from EnergyPlus to TRNSYS.

The simulations are performed using a climate file for Palermo, Italy. The prevailing wind direction (North-East) is chosen for the system air intake direction.

### 7.1.3 Results

#### Feasibility study

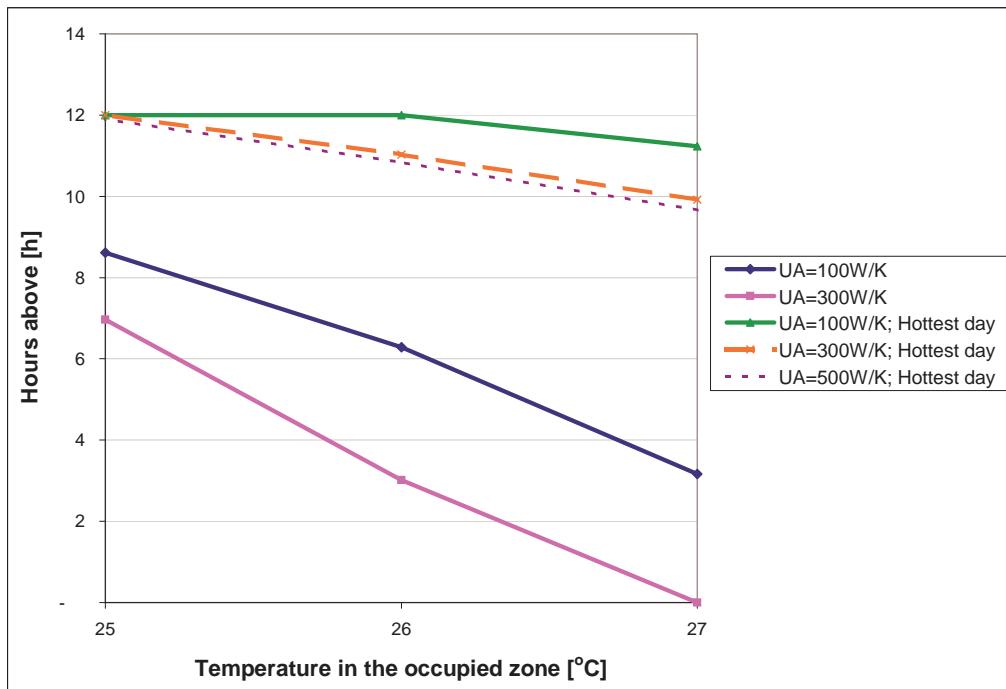
To achieve the low pressure drop in the system the air velocity through the system has to be kept low, which decreases the coil performance in terms of the heat transfer. [ASHRAE 2000] prescribes the minimum face velocity for cooling coils to be 200fpm or 1m/s. Since, the coil performance is uncertain under this working conditions, a parametric study is done to determine the value of the coil overall heat transfer coefficient under which the system would maintain the acceptable comfort levels. Simulations are performed for two different values of the overall heat transfer coefficient of the coils: 100W/K, 300W/K.

The simulations are performed for the typical summer working conditions. The simulation time is set to one working day (1<sup>st</sup> of August).

The hours during which the temperature in the occupied subzone exceeds 25°C, 26°C and 27°C are presented in Figure 7.5. As it can be seen, the lower the value of the overall coil heat transfer, the higher the temperature. For UA = 300W/K, the temperature does not exceed 27°C and the system provides sufficient thermal comfort in the occupied zone. However, for the extreme summer condition during the hottest summer day (21<sup>st</sup> of July), the occupied sub-

<sup>1</sup>The sensed zone temperature is the average zone air temperature.

zone temperature exceeds  $27^{\circ}\text{C}$  for ten hours during the occupancy period (7a.m.-7p.m.). Thus, the suggested system, using the coils with the lower values of the overall heat transfer coefficient will not perform well, in terms of the thermal comfort, for the climate of Palermo, Italy. Also, increasing the value of the overall heat transfer coefficient does not provide better results as illustrated in the figure for  $\text{UA} = 500\text{W/K}$ . For further analysis, the study assumes that  $\text{UA} = 300\text{W/K}$  is achievable for the proposed settings<sup>2</sup>.



**Fig. 7.5** — The hours during which the temperature in the occupied subzone exceeds  $25^{\circ}\text{C}$ ,  $26^{\circ}\text{C}$  and  $27^{\circ}\text{C}$  for different coil UA values.

### Recommended design changes

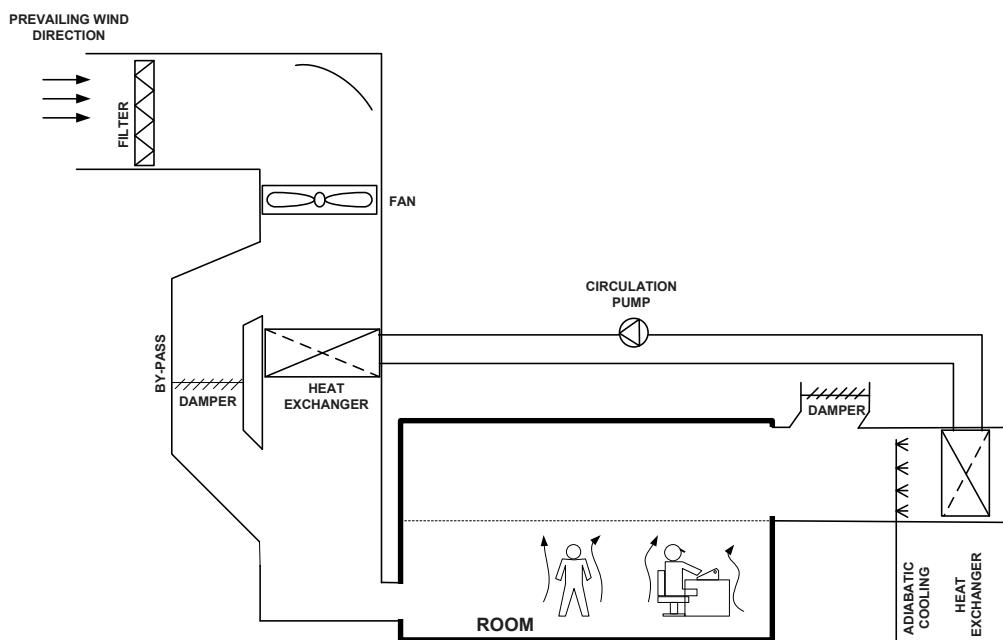
Firstly, the simulations are performed for the suggested system design, as shown in Figure 7.4. The first results revealed a potential area for an improvement of the system design and operation strategy.

An analysis of the climate file showed that the wind velocity in the direction of the air intake does not exceed the value of 16.2m/s. However, it is for 256 hours during a year above 8m/s. Thus, fan operation cost can be reduced, by using the wind dynamic pressure, but the fan would have to operate constantly to keep the supply air flow rate above the required minimum. Further, when the coils are not in use, the fan also has to overpower the pressure drop over them and thus it uses more power unnecessary.

To overcome the above weakness, a change to the original design is suggested. The new design is shown in Figure 7.6. During periods when neither cooling nor

<sup>2</sup>An oversized cooling coil will be required, but this will not be further examined since it does not add any value to the demonstration of co-simulation with this case study.

heating is required, the pressure drop in the system is decreased by redirecting the air flow through a by-pass duct. It is assumed that by-passing the coils decreases the system pressure drop characteristic from 250 to 30. A by-pass damper is used to control the air flow through the by-pass duct. The damper operation strategy is as follows.



**Fig. 7.6 — New system design.**

The damper is fully opened during periods of night ventilation. The night ventilation is enabled from the mid of April to the beginning of October. During night operation, the damper closes whenever the ambient temperature drops below  $14^{\circ}\text{C}$ , or the zone temperature drops below  $17^{\circ}\text{C}$ . The air flow rate is adjusted according to the wind velocity and the pressure drop in the by-pass.

During occupancy hours, the by-pass damper can be fully opened, fully closed or its position can be modulated to control the specified zone temperature. The operation strategy is as follows.

If either cooling or heating is required, the damper fully closes. The pressure drop in the coils requires the fan to be constantly on to keep the supply air flow on the required minimum. The resulting face velocity for this minimum is the minimum face velocity specified for the coils, and the whole air flow needs to be directed through the coils.

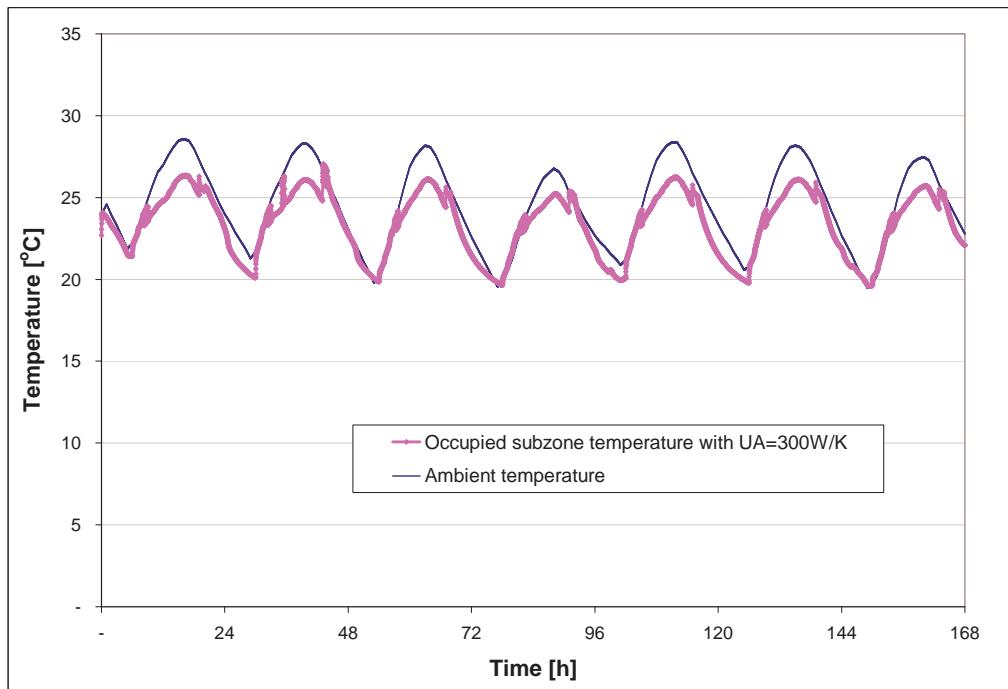
On the other hand, the damper opens if there is a need for:

- neither cooling nor heating, or
- cooling and the ambient temperature is sufficiently low ( $T_{\text{ambient}} < 20^{\circ}\text{C}$ ) so that there is no need to activate the coils and the room can be cooled by supplying the ambient air directly through the by-pass.

To avoid overcooling during the winter operation hours, the supply air flow is kept to the required minimum if there is no need for cooling, e.i.  $T_{\text{zone}} < 25^{\circ}\text{C}$ . If there is a need for cooling and the wind induced dynamic pressure on the air intake is high enough to keep the fan switched off, the damper is modulated to keep the fresh air supply flow between the required minimum and the wind induced maximum to maintain the zone air temperature  $21^{\circ}\text{C} < T_{\text{exhaust}} < 25^{\circ}\text{C}$ .

### Thermal comfort analysis

To estimate the thermal comfort, simulations are performed for a summer week (1<sup>st</sup> of August - 7<sup>th</sup> of August), a winter week (9<sup>th</sup> of January - 15<sup>th</sup> of January) and a transition-period week (4<sup>th</sup> of October - 10<sup>th</sup> of October). The resulting temperatures in the occupied zone, for the three simulation periods are shown in Figures 7.7, 7.8 and 7.9. The jumps in the zone temperature at the point the system is switched off is 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.



**Fig. 7.7** — Simulation results for the summer week using the new system design.

During the occupancy period in the summer week, the temperature in the occupied subzone is below  $27^{\circ}\text{C}$  during the occupancy period. The temperature in the occupied zone during a single summer day is shown in Figure 7.10.

Since the average zone air temperature is higher than the temperature in the occupied subzone, the heating set point temperature is set to  $24^{\circ}\text{C}$ , instead of  $21^{\circ}\text{C}$ . The temperature for a single day in the winter week is shown in Figure 7.11. As it can be seen, the capacity of the heating coil is not sufficient during the colder

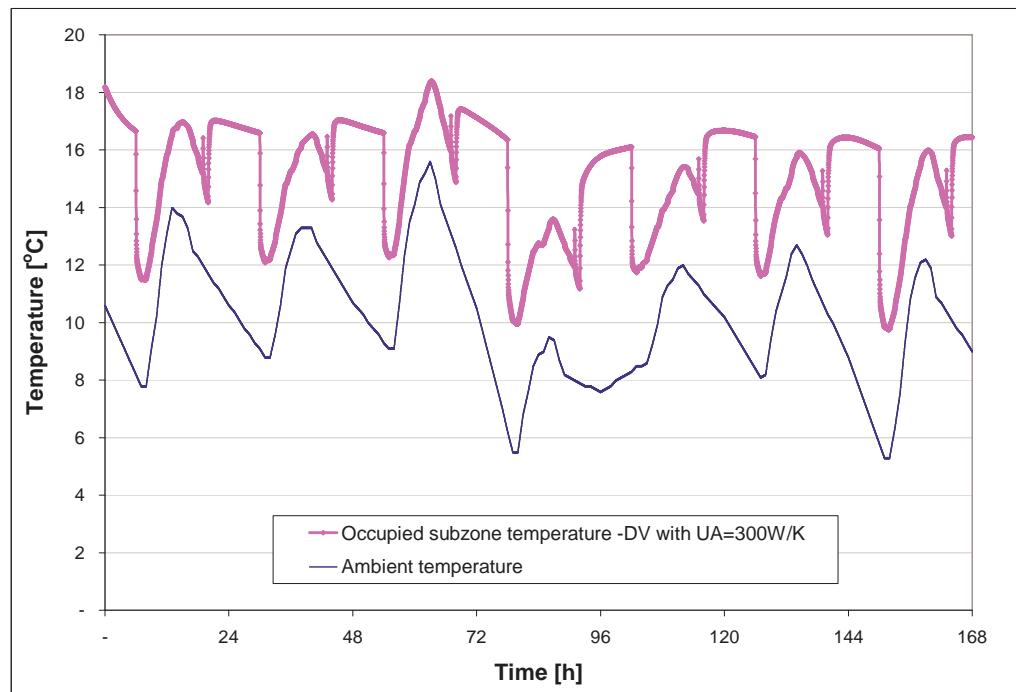


Fig. 7.8 — Simulation results for the winter week using the new system design.

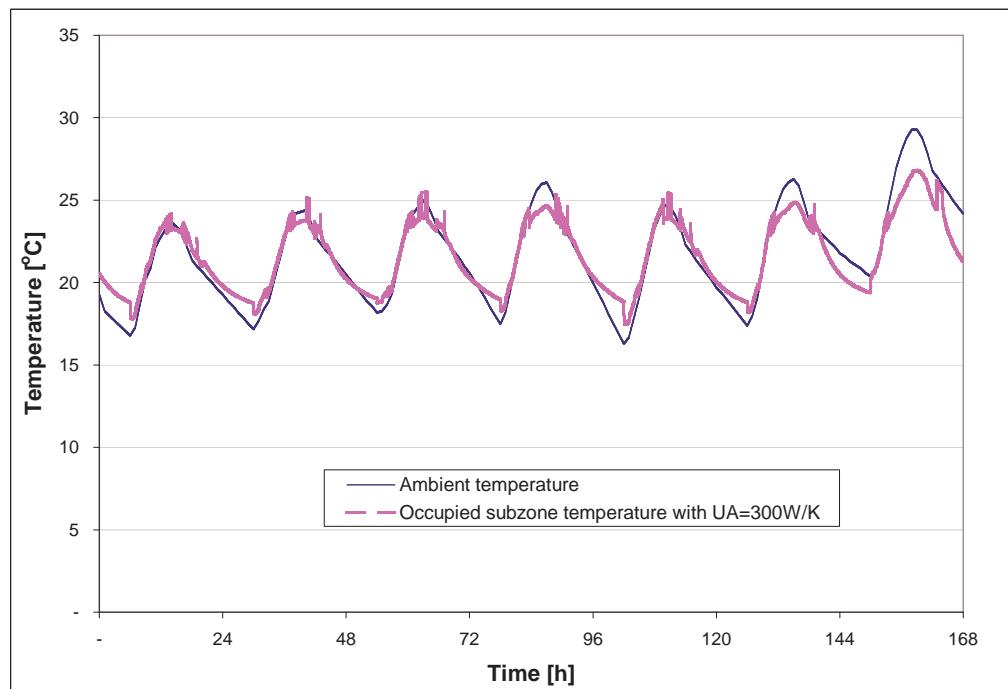
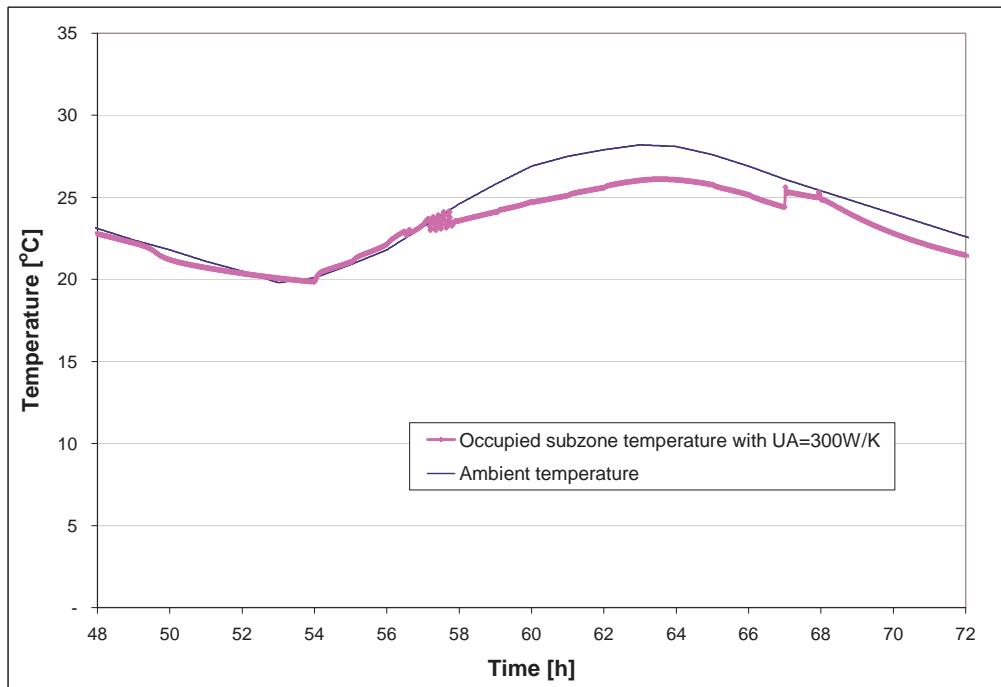
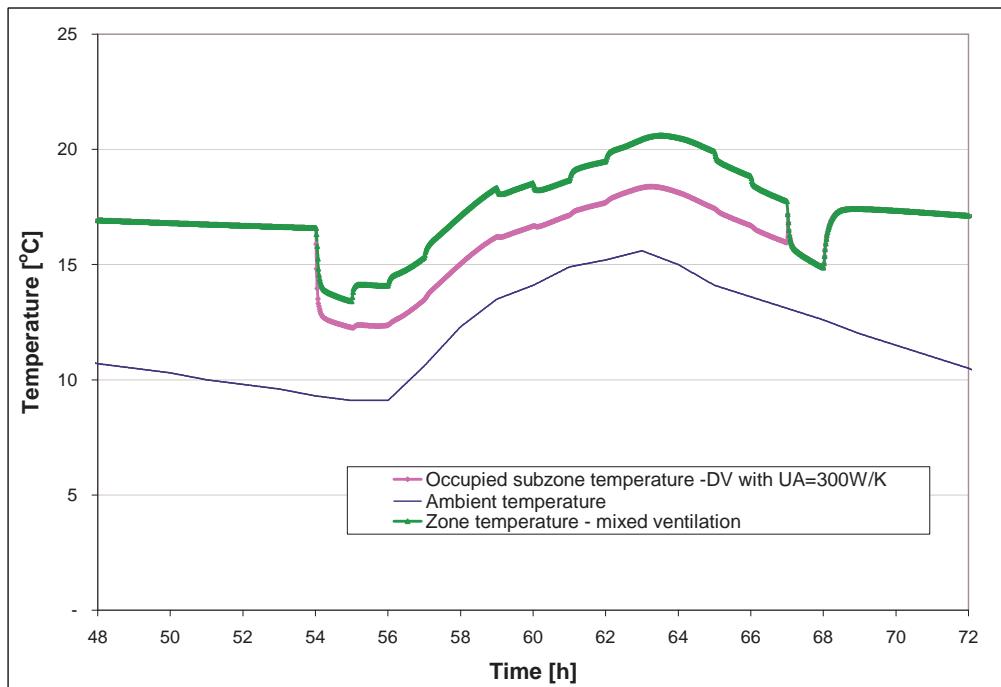


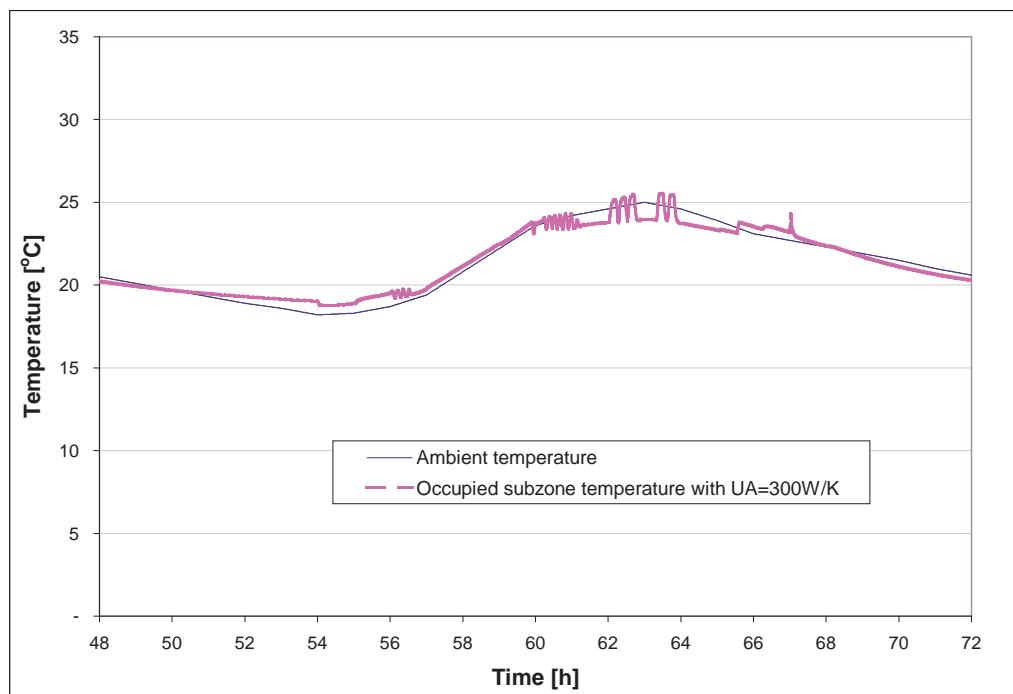
Fig. 7.9 — Simulation results for the transition-period week using the new system design.



**Fig. 7.10** — Simulation results for a single summer day (third day in the simulated summer week).



**Fig. 7.11** — Simulation results for a single winter day (third day in the simulated summer week).



**Fig. 7.12** — Simulation results for a single transition-period day (sixth day in the simulated summer week).

periods and thus, the system without additional heating will fail to provide sufficient thermal comfort to the occupants in the zone. The zone temperature obtained by assuming mixed zone condition and the same supply air flow rate is also shown in the figure. As it can be seen, in winter a mixed type of ventilation would perform better.

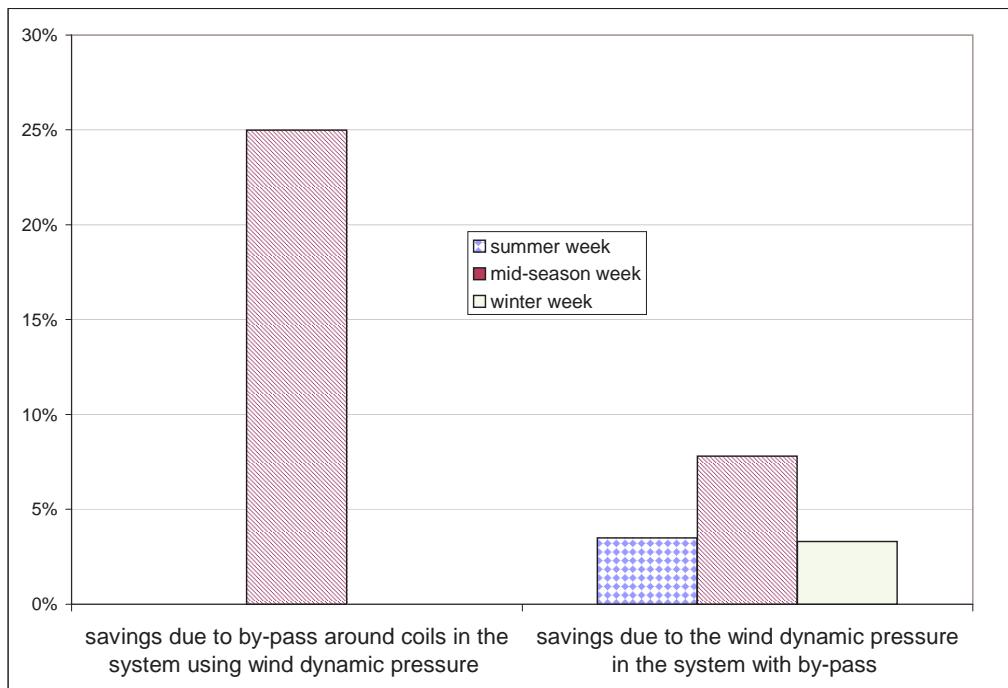
The temperature for a single day in the transition-period week is shown in Figure 7.12. The oscillations in the occupied subzone temperature result from the redirecting the flow through the by-pass when the cooling is not required.

### Energy consumption analysis

To estimate the fan energy consumption, simulations are performed for the same weeks in summer, spring and winter. The energy consumption is calculated for different cases and the comparison is made to show the potentials of energy saving (i) of the wind dynamic pressure in the system with the by-pass, and (ii) of the by-pass, if the dynamic pressure is taken into account. The comparison is made using displacement ventilation model for all the three periods.

The fan operates in parallel with the wind, with the constant flow rate. Its efficiency is assumed constant and it is set to  $\eta = 0.6$ .

The results are shown in Figure 7.13. During periods when there is constant need for either cooling or heating the energy saving potential of the by-pass is very small, since the air needs to be conditioned in the coils. However, during spring or autumn periods, when free cooling by the outside air can be exploited, the savings in the energy consumed by the fan are much bigger (25% for the



**Fig. 7.13** — Comparison of energy saving potentials.

simulated transition-period week).

For a single windy day (e.g., 10<sup>th</sup> of October), energy saving due to the wind dynamic pressure, is 17%. Due to the fact that the wind in the direction of the air intake is intermittent, the average saving is lower. Figure 7.13 shows that it ranges from 3 to 8%, depending on the season. The savings are the highest for the spring period. The reason for this is that the heating requirements are lower than during the winter and free cooling by the ambient air can be used during more hours than in the summer.

#### 7.1.4 Conclusions

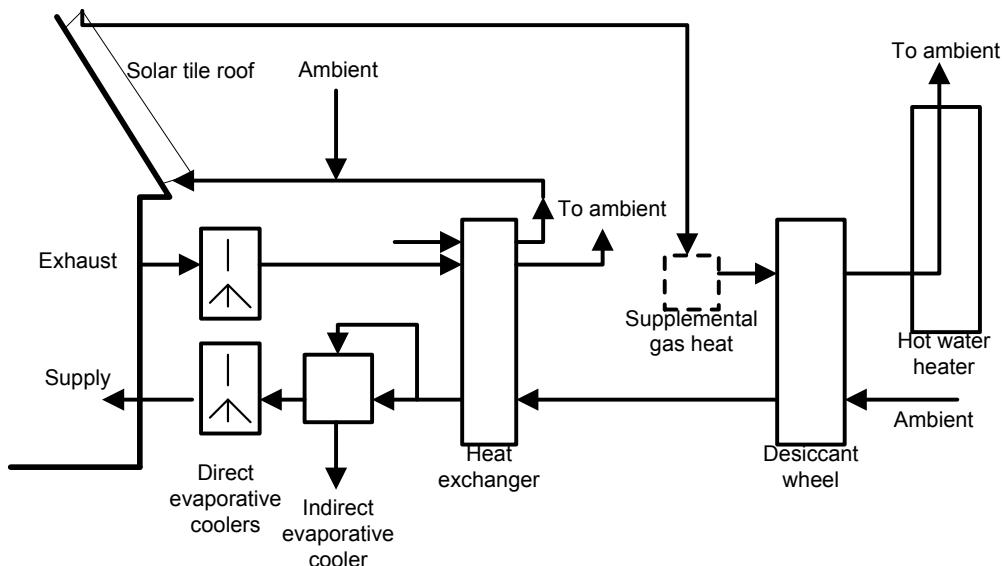
The co-simulation of the EnergyPlus building model and the TRNSYS system model enabled combination of complementary features available in these two tools. The simplified displacement ventilation model and light dimming control modeling feature, available in EnergyPlus were combined with (i) system with runaround heat recovery model in TRNSYS, which could not be modeled in EnergyPlus and (ii) the equation adding mechanism in TRNSYS, used to prototype the model of the system control strategy. Thereby the use of the co-simulation approach enabled the integrated design analysis, which would not be possible in any of the tools used individually.

## 7.2 Air solar heating for desiccant regeneration

### 7.2.1 System description

This study was inspired by a system proposed in [Archibald 2001].

The system combines three different technologies and tries to expand their individual usability to more diverse climate conditions. The system presented in Figure 7.14 consists of a desiccant wheel, a heat wheel, indirect and direct evaporative coolers and a solar tile roof. The desiccant is regenerated using the hot air from the roof solar system.



**Fig. 7.14** — Schematic of an innovative system with air solar heating for desiccant regeneration and evaporative cooling.

The supply air, after the desiccant wheel is then firstly cooled down in the heat wheel, and then split into two streams. The first stream supports an indirect evaporative cooling of the second stream without adding any humidity. Finally, the supply air is cooled in the direct evaporative cooler before it is delivered to the conditioned space.

The exhaust air is firstly cooled in the direct evaporative cooler and then passed through the first section of the heat wheel. As the exhaust volume flow rate is lower than the air flow being cooled in the wheel, an extra volume of ambient air is brought to the first section to meet the cooling requirements. The warmer ambient air can then be directed into the solar thermal tile roof system. An additional amount of the ambient air may be necessary to meet the desiccant regeneration requirements. The excess heat from the desiccant wheel can also be used for heating the water for the domestic use to improve the system performance. This is however not considered in this study. Supplemental gas heat can be added if necessary to support desiccant regeneration during cloudy days, and possibly early evening cooling hours, or other hours when solar heating is not 100% effective.

The control strategy is as follows. The system is in operation only during the office occupancy hours. The supply air flow rate is constant and set to 6.5ACH. The flow rate used in the indirect evaporative cooler is the same as the supply air flow rate. The ambient flow rate used for desiccant dehumidification is set to double supply air flow rate, since a higher air flow rate is dehumidified.

The desiccant wheel is in operation, when either the zone temperature  $T_{\text{zone}} > T_{\text{set}}$  or the zone relative humidity  $\varphi_{\text{zone}} > \varphi_{\text{set}}$ .

The indirect and direct evaporative coolers are active when the zone temperature  $T_{\text{zone}} > T_{\text{set}}$ .

## 7.2.2 Model description

A simulation model of the system in Figure 7.14, without water heater and supplemental gas heat, is used to estimate the thermal comfort and the system performance relative to a traditional system with a chiller using a compression cycle for a humid hot climate.

### Building model

To estimate the thermal comfort for different ratios of the available roof area and the conditioned space area, simulations are performed for an one-storey and a three-storey building with the same floor area. The floor layout is as shown in Figure 7.2.

### System model

The air solar collector model is not available in EnergyPlus, thus the system modeling in that tool is not feasible. Co-simulation however enables the integration of the building model in EnergyPlus with the system model in TRNSYS. Both the traditional and alternative design system options are modeled and simulated in TRNSYS.

### Co-simulation model

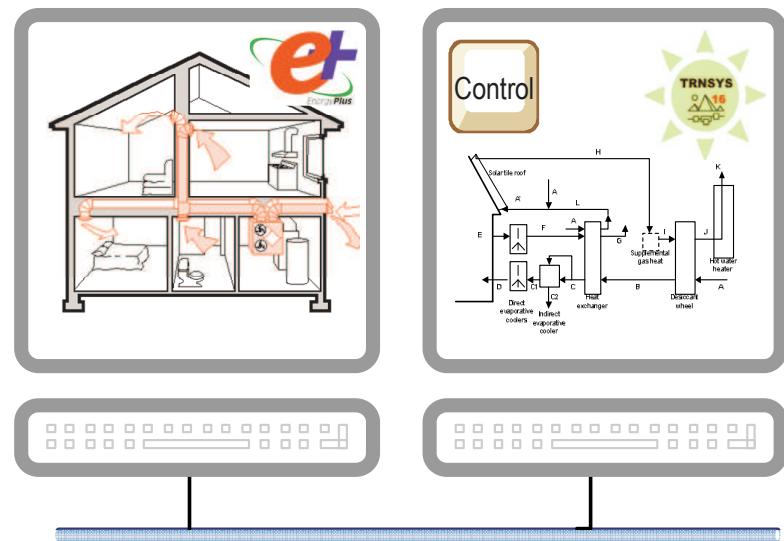
The co-simulation model is made as illustrated in 7.15. Since, the whole system is modeled and simulated in TRNSYS, inter-domain system decomposition is used. The simulators are loosely coupled.

The climate file for Nashville, Tennessee is used. The simulation time step was 1 minute.

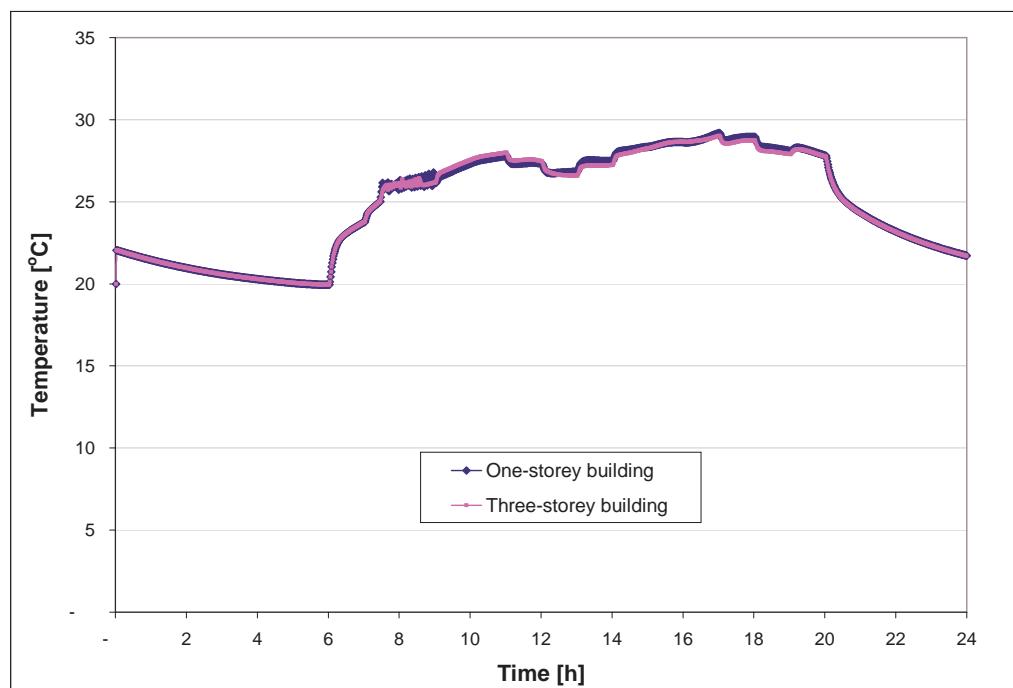
## 7.2.3 Results

### Thermal comfort analysis

The regeneration potential of the ambient air heated by the solar energy depends on the humidity ratio of the ambient air but also on the roof area. Two simulations with different roof to conditioned space area ratios are performed. The resulting zone temperatures are shown in Figure 7.16.



**Fig. 7.15** — Illustration of co-simulation model of the air solar heating for desiccant regeneration and evaporative cooling using EnergyPlus and TRNSYS.



**Fig. 7.16** — Zone temperatures obtained for a single-storey building and for a three-storey building with the same tile roof area.

**Table 7.1 — Assumed pressure drops**

Component	Pressure drop [Pa]
Tile roof	50
Desiccant wheel	190
Each heat wheel	190
Direct evaporative cooler	30
Indirect evaporative cooler	190
Filter	190
Cooling coil	250
Ducts	50

The system fails to provide the required comfort levels since the resulting zone temperatures exceed the set point temperature ( $27^{\circ}\text{C}$ ) during most of the day.

The zone temperatures for the two simulated cases are almost the same. However, the resulting zone relative humidities are significantly different. The average relative humidity for the simulated day, for the single-storey building is  $\varphi = 65\%$  and for the three-storey building is  $\varphi = 72\%$ . The reason for this is that lower air flow rates result in higher air temperatures and lower humidity ratios at the outlet of the solar air collector. Thus, for the two examined cases, the resulting air temperatures after the adiabatic cooling are almost identical, but the humidity ratios differ significantly.

### Energy consumption analysis

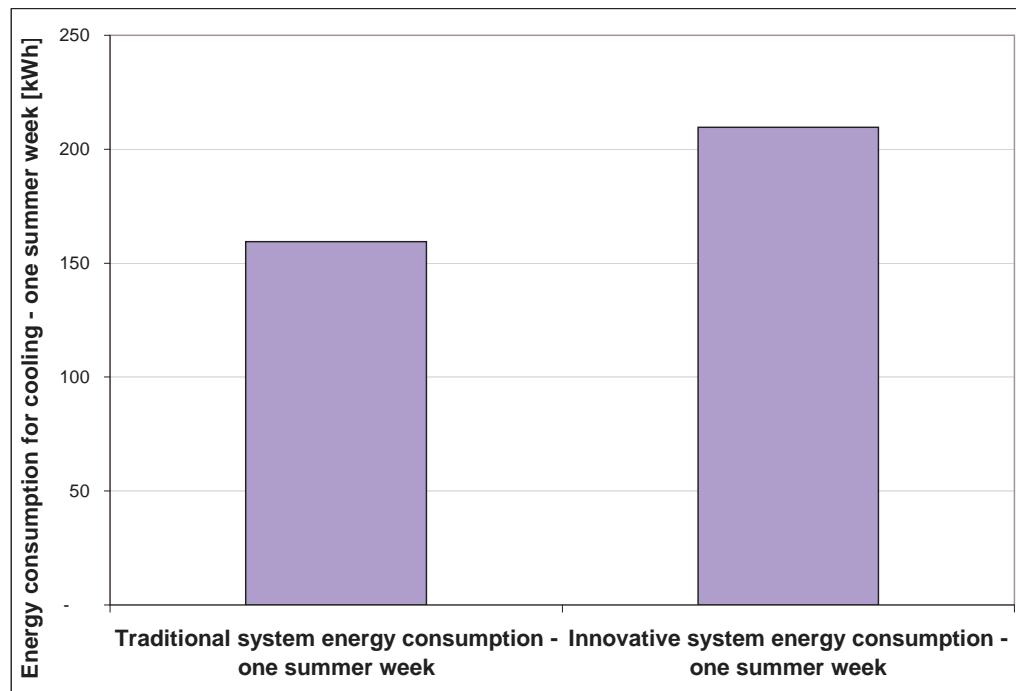
The desiccant evaporative cooling system with solar regeneration explores benefits of the indirect evaporative heat exchanger, but at the same time the energy consumption of the fans can be significant.

To assess the energy consumption relative to the traditional cooling systems, a chiller with compression cycle and a cooling coil are modeled and simulated in TRNSYS and coupled to the same building model in EnergyPlus. The traditional system is controlled by using the same temperature set points as in the alternative system.

The energy consumed by the chiller and the fan in the traditional system is compared to the energy consumed by the fans, which drive the flow of the supply and the ambient air, in the system with desiccant regeneration and evaporative cooling.

The assumed pressure drops of the components in the systems are shown in Table 7.1.

The results of the comparison are shown in Figure 7.17. As it can be seen, due to high pressure loss in the system, the energy consumption for operation of the fans in the alternative system design exceeds the energy consumption of the traditional cooling system. The traditional system does not include heat recovery.



**Fig. 7.17** — Comparison of energy consumption for cooling during a summer week, between the traditional cooling system and the system with desiccant dehumidification with evaporative cooling.

#### 7.2.4 Conclusions

The co-simulation approach enabled the integrated analysis of the complex system topology including components, not all of which are available in some state-of-the-art BPS tools.

The quick system configuration in TRNSYS made possible the comparison of different systems directly coupling them to the same building model.

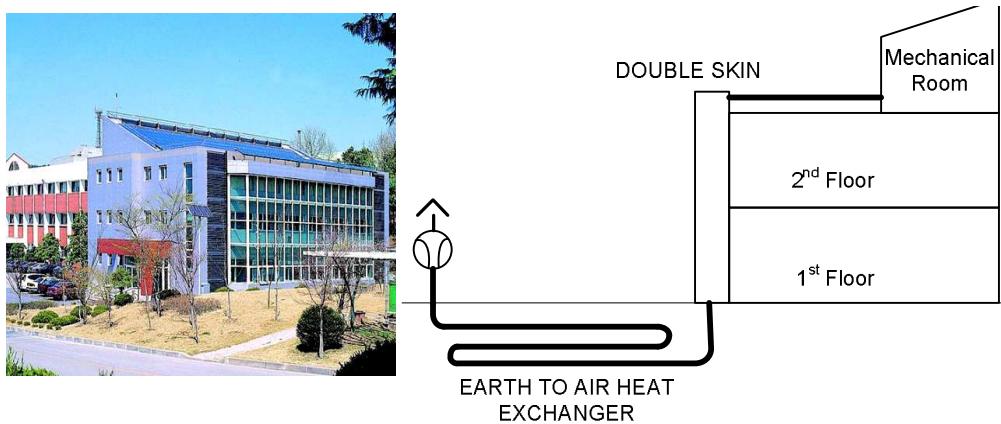
Even though the desiccant evaporative cooling system with solar regeneration explores benefits of the indirect evaporative heat exchanger, it consumes more energy for operating fans and is not applicable in the settings assumed in this study.

### 7.3 Earth-to-air heat exchanger coupled to a double-skin façade

#### 7.3.1 System description

A low energy building in Korea [Yoon et al. 1997], which schematic is given in Figure 7.18, was used to illustrate another use of the co-simulation approach.

The system includes an earth-to-air heat exchanger for preheating or pre-cooling fresh air supply, depending on the season. The building incorporates a double-skin south façade, which can act as a solar collector for additional pre-



**Fig. 7.18** — Schematic of the case study.

heating of the supply air in winter. In summer, the double skin is bypassed to avoid overheating of the air, while the double skin is naturally ventilated.

### 7.3.2 Model description

A simulation model of the above described system is used to estimate the impact of different earth-to-air designs on the energy saving potential of the overall system. The transient solar behavior and the thermal characteristics of the double-skin façade and the ground-coupled heat exchanger affect the building heating and cooling load in a dynamic way and thus the analysis required an integrated simulation approach.

#### Building model

As indicated in Figure 7.19, the building, including the double-skin façade, is modeled in ESP-r. The double-skin façade itself is modeled using the zonal air flow approach available in ESP-r. The details of the building model are reported in [Yoon et al. 1997].

#### System model

An earth-to-air heat exchanger model is at present not available in ESP-r. Therefore, the earth-to-air heat exchanger is modeled externally, in EARTH, a stand-alone tool for modeling and simulation of ground coupled heat exchangers. The co-simulation approach is used to integrate these two tools.

Based on recommended velocities in literature [Mihalakakou et al. 1996; Mihalakakou 2003; Pfafferott 2003], the pipe diameter is determined by the design volume flow rates. Different design options are simulated with two volume flow rates: (i) the lower -  $0.25 \text{ m}^3/\text{s}$  which would be sufficient for ventilation of the building, and (ii) the higher -  $1 \text{ m}^3/\text{s}$ , which could be used for free heating/cooling if the outlet temperature from the earth-to-air heat exchanger (double façade) allows. This resulted in pipe velocities between 2 and 8 m/s. The pipe depth was 1.5, 2.5 or 3 m and the pipe length 30, 70 and 150 m. A one-pipe heat

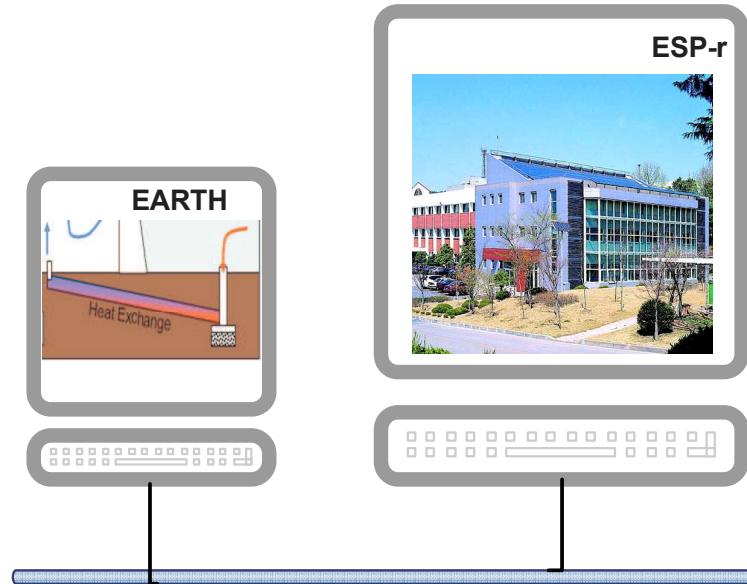
**Table 7.2 —** Tube design parameters for different design options and lower volume flow rate

	Design1	Design2	Design3
Depth [m]	1.5	2.5	3.0
Length [m]	30	70	150
Diameter [m]	0.2	0.11	0.12

exchanger was used in case of the lower volume flow rate, while four parallel tubes were considered for the higher volume flow rate (for details see Tables 7.2).

### Co-simulation model

The co-simulation model is made as illustrated in Figure 7.19.



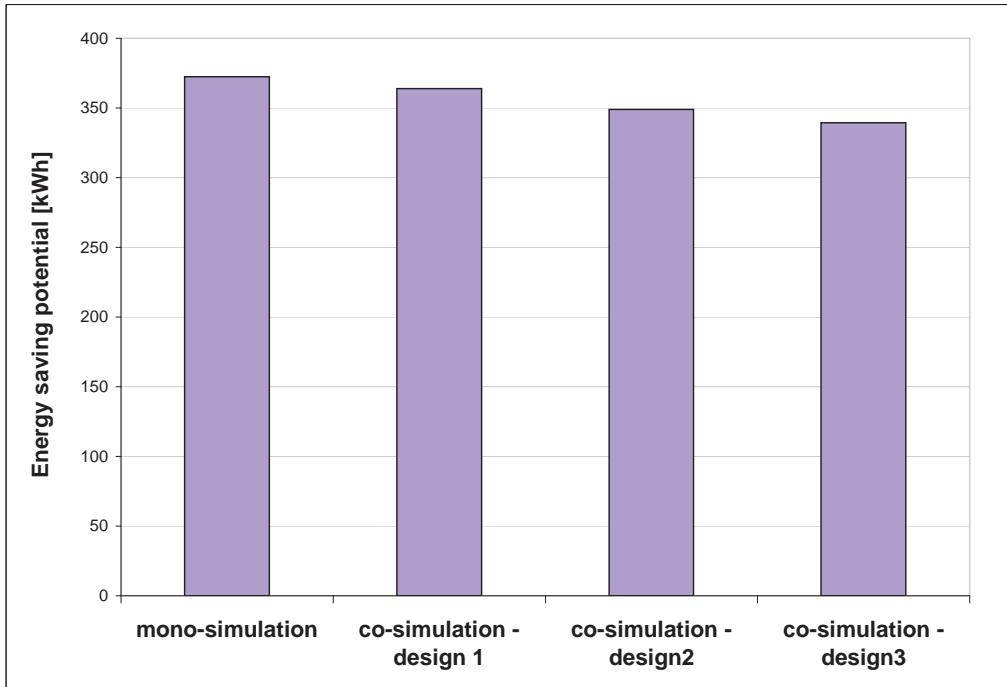
**Fig. 7.19 —** Illustration of co-simulation using ESP-r and EARTH.

Since only the earth-to-air heat exchanger is modeled outside the ESP-r environment, the intra-domain system decomposition is implemented. The loose coupling strategy is implemented with the coupling time step of 1h.

For a demonstration of the use of co-simulations, the simulations are performed for Korean climate and a period of one winter week, without a significant precalculation period. To obtain better starting condition for the soil temperature and humidity, a few seasons precalculation period could be used.

### 7.3.3 Results

The co-simulation results are compared to the results obtained by mono-simulation performed using ESP-r and an assumption that considers a monthly



**Fig. 7.20** — Estimated weekly energy saving for the lower volume flow rate obtained by (i) co-simulation of several heat exchanger designs, and (ii) mono-simulation with the assumption for the earth-to-air outlet temperature.

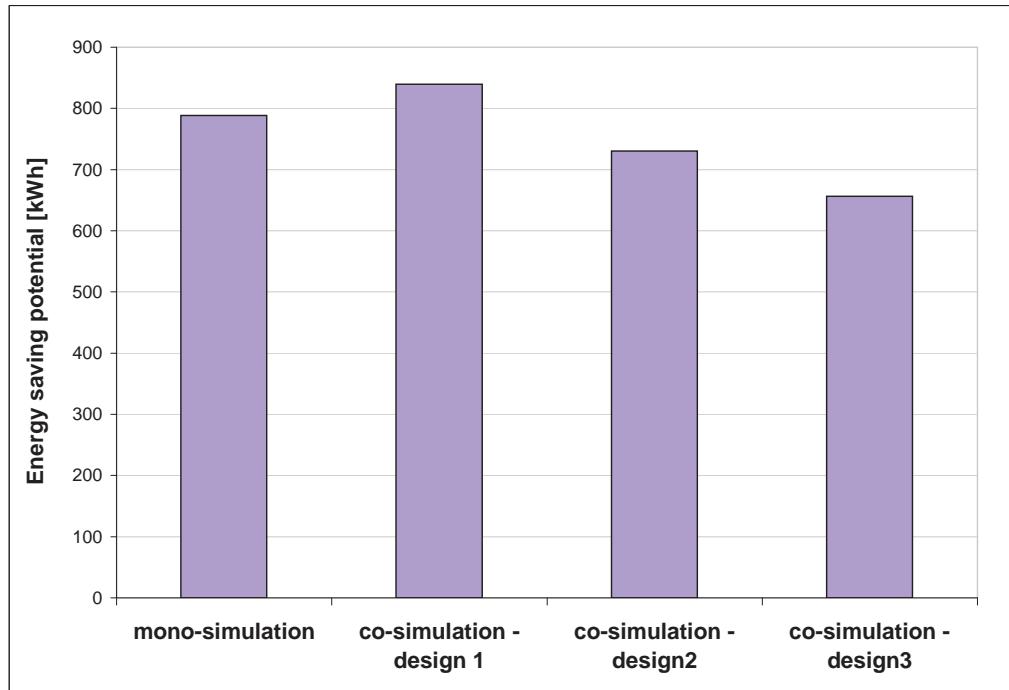
constant value for the temperature of the air entering at the bottom of the double-skin façade. Yoon et al. [1997] estimated that this temperature was equal to the ground temperature, evaluated from

$$T_{g,i} = T_{o,i} + \frac{1}{2}(T_{m,i} - T_{0,i}) \quad (7.1)$$

where  $T_{g,i}$  is the monthly mean ground temperature;  $T_{0,i}$  is the annual mean ambient dry bulb temperature and  $T_{m,i}$  is the monthly mean ambient dry bulb temperature.

The gross heat gain by both the ground-coupled heat exchanger and the double skin is evaluated from the difference between the ambient temperature and the temperature at the upper part of the double skin. It was assumed that system operates between 8a.m. and 7p.m. In some hours the heat gain is higher than the overall loss of the building itself. The minimum of these two values at each point in time were used to assess the energy-saving potential. The results are shown in Figure 7.20 and 7.21.

The temperature difference of the incoming air at the bottom of the double skin does not have significant influence on the overall result in the first case (Figure 7.20). With the lower volume flow rate, the ventilation loads have less impact on the resulting temperature of the double skin compared to solar heat gain and loads due to conduction through the construction. However, if the volume flow rate is increased, (Figure 7.21) the temperature difference of incoming air will have much more impact on the simulation results.



**Fig. 7.21** — Estimated weekly energy saving for the higher volume flow rate obtained by (i) co-simulation of several heat exchanger designs, and (ii) mono-simulation with the assumption for the earth-to-air outlet temperature.

### 7.3.4 Conclusions

It can be concluded that for low volume flow rates, the simplified model reasonably well predicts the energy-saving potential (in this specific case the difference is less than 10%). In this particular case the incoming temperature, hence, the earth-to-air heat exchanger itself, does not have a big influence on the results. However, the simplified, mono-simulation approach does not allow evaluating the influence of different design options when the volume flow rate is higher. The deviation here rises up to 25%. The co-simulation approach is necessary to accurately predict the energy-saving potential in this case.

The simulations are done only for a winter week. However, in summer, the double-skin is by-passed, and the influence of differences in earth-to-air heat exchanger design are expected to be more noticeable.



# 8

# Conclusions

THIS chapter finishes the thesis by giving a set of concluding remarks and providing directions for future work.

## 8.1 Concluding remarks

The work presented in this thesis proves the hypothesis that co-simulation can help in performance prediction of innovative integrated mechanical energy systems in buildings. This was achieved by a thorough investigation, development and implementation, numerical experimentation and usability testing of co-simulation between different BPS tools.

The research has shown that the co-simulation approach can alleviate the limitations of the available state of the art BPS tools by (i) introducing the potential to “pool resources”, and (ii) allowing simulators to complement each other without creating undue redundancy.

Numerical experimentation has shown that the strong coupling strategy allows the use of longer time steps than the loose coupling strategy to achieve the same accuracy. However, with respect to both the execution time (defined by the specifics of implementation of strong and loose coupling), and the complexity of its implementation, the loose coupling strategy (using small time steps) is always preferred to the strong coupling strategy.

Loose co-simulation requires more careful implementation to avoid degradation of accuracy. One reason for this is that there could be non-linearities in the system. A remedy for such cases would be to use small coupling time steps. Another reason is that co-simulation with loose coupling leads to a partitioning of the numerical schemes implemented in the coupled simulators.

The partitioned numerical schemes used to approximate solution of a system of ordinary differential equations are shown to be zero-stable, consistent and thus convergent. The analysis of the local truncation error of this scheme has shown that the greater the capacity of the subsystem (i.e., its receiving node) simulated in the external simulator and the lower the first derivative of the coupling data, the smaller the error is. If the system being modeled is described by a differential algebraic system of equations, the partitioning can lead to inconsistencies in the approximation of the solution.

The results of the validation study have shown that performing loosely coupled co-simulation with sufficiently small time steps could generate results with

the same accuracy as mono-simulation. From the comparison of different co-simulation strategies it can be concluded, from the execution efficiency point of view, that a system decomposed between the building and HVAC domains performs better. However, this inter-domain system decomposition results in a numerical scheme with more delayed coupling data and thus in lower accuracy. A suggested strategy that remedies this problem showed good performance for cases where the coupled heat flux is a linear function of the state variables.

The use of predictors for the coupling data enhances the accuracy of loose coupling only when the changes of the coupling data are smooth. Otherwise the use of predictors can even introduce additional inaccuracies.

The developed co-simulation prototypes were used in several case studies to confirm that the method can be practically used in performance prediction of innovative integrated mechanical energy systems in buildings. That co-simulation enables combination of complementary features available in coupled tools was demonstrated by a few use cases. The approach facilitated fully integrated design analysis, which would not be possible if any of the BPS tools were used individually.

With the co-simulation approach any new development in a single simulation tool can be made immediately available to the user of any other BPS tool, once the communication protocol between the tools has been implemented.

In perspective, this thesis showed and discussed the advantages of co-simulation in terms of simulator flexibility and capability extension, by combining features from different tools. Other aspects, such as difficulty of use and required knowledge may not favor the approach.

## 8.2 Directions for future work

### Coupling state of the art BPS tools with equation based tools

Although outside of the scope of the implementations presented in this thesis, further extensions of simulator's flexibility and capability can be achieved by linking state of the art tools to equation-based tools. The latter will allow for rapid model prototyping of emerging technologies.

### Smart system decomposition

To achieve required accuracy of co-simulation results, coupling frequency should be set according to the (i) capacity of the "receiving" node in the external simulator and (ii) rate of change of the coupling data. The former determines the sensitivity of the "receiving" coupling node to the coupling data. The latter is driven by the "sending"-subsystem node's capacitance, the rate of change of its input, and the forcing function acting upon it. Future research should focus on developing guidelines on how to decompose a system in order to minimize the coupling frequency, or to maximize the accuracy of co-simulation, with regards to the system's characteristics. Such guidelines would be a step towards intelligent co-simulation.

### **Co-simulation of pressure dependent system flows**

The thesis considers co-simulation only for tools that model fluid flow in the system independently of the pressure in the system. Future research and development of co-simulation should also incorporate pressure driven flows.

### **Decomposition in building domain**

The main focus of this thesis was put on the HVAC system modeling and simulation, and in regards to this, the studied strategies include only the HVAC intra-domain decomposition. There are, however, cases where the intra-building-domain decomposition is also required (e.g., in concrete core conditioning systems). Further research should extend the scope of co-simulation implementation in this direction.

### **Decomposition with regards to controls**

The thesis showed the analysis and implementation of different decomposition strategies and resulting partitioned numerical schemes taking into account only a simplified controller. A more general analysis with regards to controls needs to be addressed in future.

### **Variable coupling time step approach**

Variable coupling time step approach was suggested to enhance the accuracy of loose coupling. Typical ordinary differential solvers use estimates for the local truncation error, and based on this estimates they calculate the required step size for the desired accuracy. Future work needs to consider implementation algorithms with variable time step approach, and to enable its use in the co-simulation of BPS tools.

### **Moving away from the prototypes**

The prototypes discussed in this theses are, of course, not an end product. They are merely used to test different co-simulation approaches. The tools used in the prototypes are made open for co-simulation with other tools using the same communication mechanism. However, the implementation is limited to a single platform and to a single computer. Designing a more advanced coupling interface is a challenge for future work.



# References

- Aasem, E. O.: 1993, *Practical simulation of buildings and air-conditioning systems in the transient domain*, PhD thesis, University of Strathclyde, Glasgow, Scotland.
- ANNEX40, I.: 2004, Commissioning tools for improved energy performance, *Technical report*, Program of the International Energy Agency. Available at: <http://www.commissioning-HVAC.org> [Accessed: August, 2005].
- Archibald, J.: 2001, A new desiccant evaporative cooling cycle for solar air conditioning and hot water heating. Available at: <http://www.americansolar.com/forum2001-cooling.pdf> [Accessed: Jun, 2007].
- Arnold, M., Carrarini, A., Heckmann, A. and Hippmann, G.: 2002, Modular dynamical simulation of mechatronic and coupled systems, *Proceedings of 5th World Congress on Computational Mechanics*, Vienna University of Technology, Vienna, Austria.
- ASHRAE: 2000, *ASHRAE Handbook, HVAC Systems and Equipment*, American society for heating, refrigerating and air-conditioning engineers.
- ASHRAE: 2001, *ASHRAE Handbook, Fundamentals*, American society for heating, refrigerating and air-conditioning engineers.
- 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, *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].
- Augenbroe, G. L. M.: 1992, Integrated building performance evaluation in the early design stages, *Building and Environment* 27(2).
- Balci, O.: 1997, Verification, validation and accreditation of simulation models, *Proceedings of 1997 Winter simulation conference*, The Institute of Electrical and Electronics Engineers (IEEE), Atlanta, Georgia, USA.
- Balci, O.: 2004, Quality assessment, verification, and validation of modeling and simulation applications, *Proceedings of 2004 Winter simulation conference*, The Institute of Electrical and Electronics Engineers (IEEE), Washington, D.C., USA.
- Boer, C. A.: 2005, *Distributed simulation in industry*, PhD thesis, 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).

## REFERENCES

---

- Bring, A., Sahlin, P. and Vuolle, M.: 1999, Models for building indoor climate and energy simulation, *Technical report*, International Energy Agency Solar Heating & Cooling Programme. report of task 22: Building energy analysis tools.
- Buss, A. H. and Jackson, J.: 1998, Distributed simulation modeling: a comparison of HLA, CORBA, and RMI, *Proceedings of 1998 Winter Simulation Conference*, The Institute of Electrical and Electronics Engineers (IEEE), Washington, D.C., USA.
- Carroll, W. L. and Hitchcock, R. J.: 2005, DELIGHT2 daylighting analysis in EnergyPlus: Integration and preliminary user results, *Proceedings of 9th International IBPSA Conference*, International Building Performance Simulation Association, Montréal, Canada.
- Chow, T. T.: 1995, *Air-conditioning plant component taxonomy by primitive parts*, PhD thesis, University of Strathclyde, Glasgow, Scotland.
- Clarke, J. A.: 2001, *Energy simulation in building design*, Second edition, Butterworth-Heinemann, UK.
- Clarke, J. A., Cockcroft, J., Conner, S., Hand, J. W., Kelly, N. J., Moore, R., O'Brien, T. and Strachan, P.: 2002, Simulation-assisted control in building energy management systems, *Energy and buildings* **34**, 933–940.
- Clarke, J. A. and MacRandall, D. F.: 1993, The Energy Kernel System: Form and content, *Proceedings of 3rd International IBPSA Conference*, International Building Performance Simulation Association, Adelaide, Australia.
- Clarke, J. A., Tang, D., James, K. and MacRandall, D. F.: 1992, Energy kernel system, *Technical Report GR/F/07880*, Engineering and Physical Science Research Council, Swindon.
- Clarke, J., Hong, J., Kim, J., Strachan, P., Hwang, I. and Lee, H.: 2005, Simulation-based design procedure to evaluate hybrid-renewable energy systems for residential buildings in Korea, *Proceedings of 9th International IBPSA Conference*, International Building Performance Simulation Association, Montréal, Canada.
- Crawley, D. B., Hand, J. W., Kummert, M. and Griffith, B. T.: 2005, Contrasting the capabilities of building energy performance simulation programs, *Proceedings of 9th International IBPSA Conference*, International Building Performance Simulation Association, Montréal, Canada.
- CSTB: 2003, Type 155 - A new TRNSYS type for coupling TRNSYS and Matlab. Online presentation available at: <http://software.cstb.fr/articles/18.ppt> [Accessed: December, 2003].
- Curtil, D.: 2004, Lawrence Berkeley National Laboratory, personal communication.
- Djunaedy, E.: 2005, *External coupling between building energy simulation and computational fluid dynamics*, PhD thesis, 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).
- DMSO: 2000, Fidelity, validation, verification and accreditation; Recommended practices guide, *Technical report*, Defense modeling and simulation office, USA.
- DMSO: 2004, Key concepts of VV&A, *Technical report*, Defense modeling and simulation office, USA.
- Donnelly, D., Berzins, M. and Shadwick, B. A.: 2007, On the role and place of computation in science and engineering, *Computing in science and engineering* **9**(1).
- Duggan, J.: 2002, A distributed computing approach to system dynamics, *System Dynamics Review* **18**(1), 87–98.
- Ellinger, B.: 2007, Applied Dynamics International, Ann Arbor, MI, USA, personal communication.
- Elliott, A. S.: 2000, A highly efficient, general-purpose approach for co-simulation with ADAMS, *Proceedings of 15th European ADAMS User's Conference*, Rome, Italy. Available at: [http://www.mscsoftware.com/support/library/conf/adams/euro/2000/MDI\\_Cosimulation.pdf](http://www.mscsoftware.com/support/library/conf/adams/euro/2000/MDI_Cosimulation.pdf) [Accessed: July, 2006].
- Elliott, A. S.: 2002, Status update on general-purpose co-simulation with ADAMS. Online presentation available at: <http://support.mscsoftware.com/> [Accessed: Jun, 2006].
- Falkenhainer, B. and Forbus, K. D.: 1991, Compositional modeling - Finding the right model for the job, *Artificial Intelligence* **51**, 95–143.
- Felgner, F., Agustina, F. S., Bohigas, R. C., Merz, R. and Litz, L.: 2002, Simulation of thermal building behaviour in Modelica, *Proceedings of 2nd International Modelica conference*, Oberpfaffenhofen, Germany, pp. 147–154.
- 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., Kin, C., Lopez, I., Sang, J. and Townsend, S.: 2001, A CORBA-based development environment for wrapping and coupling legacy scientific codes, *Proceedings of 10th IEEE International Symposium on High Performance Distributed Computing*, The Institute of Electrical and Electronics Engineers (IEEE), San Francisco, CA, USA, pp. 22–31.
- Fong, K. F., Hanby, V. I. and Chow, T. T.: 2006, HVAC system optimization for energy management by evolutionary programming, *Energy and Buildings* **38**, 220–231.

## REFERENCES

---

- Fujimoto, R. M.: 1998, Time management in the High Level Architecture, *Simulation* **71**(6), 88–400.
- Fujimoto, R. M.: 2003, Distributed simulation systems, *Proceedings of 2003 Winter simulation conference*, The Institute of Electrical and Electronics Engineers (IEEE), New Orleans, Louisiana, USA.
- Fujimoto, R. M.: 2005, Parallel and distributed simulation - Introduction and motivation. Available at: <http://www.cc.gatech.edu/classes/AY2003> [Accessed: January, 2006].
- Ganse, A.: 2005, Distributed processing. Available at: [http://staff.washington.edu/aganse/presentations/EIS\\_Distributed\\_Systems.ppt](http://staff.washington.edu/aganse/presentations/EIS_Distributed_Systems.ppt) [Accessed: February, 2006].
- Gantamacher, F. R.: 1974, *The theory of matrices*, Chelsea Publishing Complany, New York.
- Gear, C. W.: 1971, *Numerical initial value problems in ordinary differential equations*, Prentice-Hall, Englewood Cliffs, N.J.
- Golub, G. H. and Ortega, J. M.: 1992, *Scientific computing and differential equations*, Academic Press, London, U.K.
- 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.
- Gross, D. C.: 1999, Report from the Fidelity Implementation Study Group (FDM-ISG), *Technical Report 99S-SIW-167*, Simulation Interoperability Standards Organization (SISO), Orlando, FL. Spring Simulation Interoperability Workshop (SIW).
- 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.
- Gu, B. and Asada, H.: 2004, Co-simulation of algebraically coupled dynamic subsystems without disclosure of proprietary subsystem models, *ASME Journal of Dynamic Systems, Measurement, and Control* **126**.
- Gu, B. and Asada, H. H.: 2001, Co-simulation of algebraically coupled dynamic subsystems, *Proceedings of the American control conference*, The Institute of Electrical and Electronics Engineers (IEEE), Arlington, VA, USA.
- Hall, G. and Watt, J. M.: 1976, *Modern numerical methods for ordinary differential equations*, Clarendon press, Oxford, U.K.
- Haves, P., Norford, L. K. and DeSimone, M.: 1998, A standard simulation test bed for evaluation of control algorithms and strategies, *ASHRAE transactions* **104**(1).

- Haves, P., Salsbury, T., Claridge, C. and Liu, M.: 2001, Use of whole building simulation in on-line performance assessment: Modeling and implementation issues, *Proceedings of 7th International IBPSA Conference*, International Building Performance Simulation Association, Rio de Janeiro, Brazil.
- Hensen, J. L. M.: 1991, *On the thermal interaction of building structure and heating and ventilating system*, PhD thesis, Eindhoven University of Technology.
- Hensen, J. L. M.: 1996, Application of modelling and simulation to HVAC systems, *Proceedings of 30th Int. Conf. MOSIS '96*, Krnov, Technical University of Ostrava, CZ.
- Hensen, J. L. M.: 1999, A comparison of coupled and de-coupled 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, *Proceedings of Int. ASHRAE/CIBSE Conf. "20-20 Vision"*, Dublin, Ireland.
- Hensen, J. L. M., Djunaedy, E., Radošević, M. and Yahiaoui, A.: 2004, Building performance simulation for better design: Some issues and solutions, *Proceedings of 21st Conference on Passive and Low Energy Architecture*, Technische Universiteit Eindhoven, The Netherlands.
- Hillestad, M. and Hertzberg, T.: 1986, Dynamic simulation of chemical engineering systems by the sequential modular approach, *Computers & 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 & Chemical Engineering* **12**(5), 407–414.
- Hopfe, C. J., Struck, C., Hensen, J. L. M. and de Wilde, P.: 2006, Considerations regarding decision support tools for conceptual building design, *Proceedings of 11th Int. Conf. on Computing in Civil and Building Engineering, on CD*, ISCCBE (International Society for Computing in Civil and Building Engineering), Montréal, Canada.
- Hu, A., San, Y. and Wang, Z.: 2001, Verifying and validating a simulation model, *Proceedings of 2001 Winter simulation conference*, The Institute of Electrical and Electronics Engineers (IEEE), Arlington, Virginia, USA.
- Huang, J., Winkelmann, F., Buhl, F., Pedersen, C., Fisher, D., Liesen, R., Taylor, R., Strand, R. and Lawrie, L.: 1999, Linking the COMIS multi-zone airflow model with the EnergyPlus building energy simulation program, *Proceedings of 6th International IBPSA Conference*, International Building Performance Simulation Association, Kyoto, Japan.
- Hui, S. C. M.: 1998, Simulation based design tools for energy efficient buildings in Hong Kong, *Hong Kong Papers in Design and Development*, Vol. 1, PACE publishing, Department of Architecture, University of Hong Kong.

## REFERENCES

---

- Hyviken, J.: 1996, Real time simulation of HVAC systems for building optimization, *Fault detection and diagnosis technical papers of IEA Annex 25*, Finland, Technical Research Centre of Finland, VTT Building Technology.
- IEEE: 2000, 1516-2000 - Standard for modeling and simulation High Level Architecture, *Technical report*, Institute of Electrical and Electronics Engineers.
- Janak, M.: 1999, Coupling building energy and lighting simulation, *Proceedings of 5th International IBPSA Conference*, Vol. 2, International Building Performance Simulation Association, Kyoto, Japan, pp. 307–312.
- Jreijiry, D., Husaunndee, A., Inard, C. and Villenave, J. G.: 2003, Control of ventilation in buildings using simbad building and HVAC toolbox, *Proceedings of 8th International IBPSA Conference*, International Building Performance Simulation Association, Eindhoven, The Netherlands, pp. 661–667.
- 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 U.S. 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 modeling of dynamic systems* **6**(2), 93–113.
- Kummert, M., Bradley, D. and Dowell, T. M.: 2004, Comining different validation techniques for continuous software improvement - Implications in the development of TRNSYS 16, *Proceedings of ESIM 2004 Conference*, Canadian Chapter of the International Building Performance Simulation Association, Vancouver, Canada.
- Lambert, J. D.: 1991, *Numerical methods for ordinary differential systems, the initial value problem*, John Wiley & Sons, West Sussex, U.K.
- Larsson, J.: 2001, *Concepts for Multi-Domain Simulation*, PhD thesis, Linköpings universitet, Sweden.
- Law, A. M. and McComas, M. G.: 2001, How to build valid and credible simulation models, *Proceedings of 2001 Winter simulation conference*, The Institute of Electrical and Electronics Engineers (IEEE), Arlington, Virginia, USA.
- LBLN: 2003, SPARK 2.0 Reference manual simulation problem analysis and research kernel, *Technical report*, Lawrence Berkeley National Laboratory.
- LBLN: 2005, Cleanrooms: Best-practice summaries, HVAC air systems, recirculation air system types. Available at: <http://hightech.lbl.gov/cleanrooms-bpg.html> [Accessed March, 2008.]

- LBNL: 2006, EnergyPlus engineering reference - User manual, *Technical report*, University of Illinois and The Ernest Orlando Lawrence Berkeley National Laboratory.
- Lee, J. S.: 2004, High performance modeling for distributed simulation, *Lecture Notes in Computer Science* **3397**, 138–146.
- Li, N., Wang, Z., Liu, W. H., Peng, X. Y. and Gong, G. H.: 2005, Research on construction and interoperability of complex distributed simulation system, *Lecture Notes in Computer Science* **3398**, 131–140.
- Lockley, S. R., Rombouts, W. and Plokker, W.: 1994, The COMBINE data exchange system, *Proceedings of First European Conference of Product and Process Modeling (ECPHM) Conference*, Dresden University of Technology, Dresden, Germany.
- Lu, L., Cai, W., Xie, L., Li, S. and Soh, Y. C.: 2005, HVAC system optimisation—in building section, *Energy and Buildings* **37**, 11–22.
- Mathews, E. H. and Botha, C. P.: 2003, Improved thermal building management with the aid of integrated dynamic HVAC simulation, *Building and environment* **38**, 1423–1429.
- Matthies, H. G. and Steindorf, J.: 2002, Strong coupling methods, *Technical Report 2002-06*, Institute of Scientific Computing, Technical University Braunschweig, Germany.
- Mazria, E. and Kershner, K.: 2008, Meeting the 2030 challenge through building codes. Available at: <http://www.architecture2030.org> [Accessed: August, 2008].
- McDowell, T. P., Emmerich, S., Thornton, J. W. and Walton, G.: 2003, Integration of airflow and energy simulation using CONTAM and TRNSYS, *ASHRAE Transactions* **109**(1).
- McGregor, D.: 2005, Moves 3500 - Network communication in simulation. Available at: <http://www.movesinstitute.org/~mcgredo/mv3500/lectures/Lecture1.ppt> [Accessed: July, 2006].
- Mihalakakou, G.: 2003, On the heating potential of a single buried pipe using deterministic and intelligent techniques, *Renewable Energy* **28**, 917–927.
- Mihalakakou, G., Lewis, J. O. and Santamouris, M.: 1996, The influence of different ground covers on the heating potential of earth-to-air heat exchangers, *Renewable Energy* **7**, 33–46.
- Miller, D. E.: 1980, The impact of HVAC process dynamics on energy use, *ASHRAE transactions* **86**(2), 535–553.

## REFERENCES

---

- Monty, A.: 2002, Co-simulation tools in the new VTB version. Available at: [http://vtb.ee.sc.edu/review/2002/presentations/Co-simulation\\_tools\\_in\\_the\\_new\\_VTB\\_version\(Monti\).ppt](http://vtb.ee.sc.edu/review/2002/presentations/Co-simulation_tools_in_the_new_VTB_version(Monti).ppt) [Accessed: March, 2006].
- Nayak, P. P. and Joskowicz, L.: 1996, Efficient compositional modeling for generating causal explanations, *Artificial Intelligence* **83**(2), 193–227.
- Neymark, J. and Judkoff, R.: 2002, International Energy Agency building energy simulation test and diagnostic method for heating, ventilating, and air-conditioning equipment models (HVAC BESTEST), volume 1: cases e100-e200, *Technical report*, National Renewable Energy Laboratory. technical report of a 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, and air-conditioning equipment models (HVAC BESTEST), volume 2: cases e300-e545, *Technical report*, National Renewable Energy Laboratory. technical report of a IEA Solar Heating and Cooling programme project.
- NRC: 2004, Defining the methodology for the next-generation HOT2000 simulator, *Technical report*, National Resources Canada. Available at: [http://www.buildingsgroup.nrcan.gc.ca/docs/pdf/final\\_report\\_survey.pdf](http://www.buildingsgroup.nrcan.gc.ca/docs/pdf/final_report_survey.pdf) [Accessed: April, 2005].
- Pace, D. K.: 2000, Ideas about simulation conceptual model development, *Johns Hopkins Apl Technical Digest* **21**(3), 327–336.
- Page, E. H., Nicol, D. M., Balci, O., Fujimoto, R. M., Fishwick, P. A., L'Ecuyer, P. and Smith, R.: 1999, Panel: Strategic directions in simulation research, *Proceedings of 1999 Winter simulation conference*, The Institute of Electrical and Electronics Engineers (IEEE), Phoenix, Arizona, USA.
- Papamichael, K. and Vineeta, P.: 2002, Barriers in developing and using simulation-based decision-support software, *Proceedings of ACEEE 2002 summer study on energy efficiency in buildings*, American Council for an Energy-Efficient Economy, CA, USA.
- Park, K. C.: 1980, Partitioned transient analysis procedures for coupled-field problems: Stability analysis, *Journal of Applied Mechanics* **47**, 370–376.
- Pfafferott, J.: 2003, Evaluation of earth-to-air heat exchangers with a standardised method to calculate energy efficiency, *Energy and Buildings* **35**, 971–983.
- Piperno, S.: 1997, Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations, *International journal for numerical methods in fluids* **25**, 1207–1226.
- Pollini, L. and Innocenti, M.: 2000, A synthetic environment for dynamic systems control and distributed simulation, *IEEE Control Systems Magazine* **20**(2), 49–61.

- Raslan, R. and Davies, M.: 2006, The implementation of a simulation-based compliance methodology for the UK building regulations 2006: An industry survey, *Proceedings of the European Modelling Symposium*, UK Simulation Society and the Graduate School, UCL.
- Rickel, J. and Porter, B.: 1997, Automated modeling of complex systems to answer prediction questions, *Artificial Intelligence* **93**(1-2), 201–260.
- Riederer, P.: 2005, Matlab/Simulink for building and HVAC simulation- State of the art, *Proceedings of 9th International IBPSA Conference*, International Building Performance Simulation Association, Montréal, Canada.
- Sahlin, P.: 2000, The methods of 2020 for building envelope and HVAC systems simulation - Will the present tools survive?, *CIBSE conference*, Dublin, Ireland.
- Sahlin, P., Bring, A. and Kolsaker, K.: 1995, Future trends of the neutral model format (NMF), *Proceedings of 4th International IBPSA Conference*, International Building Performance Simulation Association, Madison, WI, USA.
- Sahlin, P., Eriksson, L., Grozman, P., Johnsson, H., Shapovalov, A. and Vuolle, M.: 2003, Will equation-based building simulation make it? - Experiences from the introduction of IDA indoor climate and energy, *Proceedings of 8th International IBPSA Conference*, International Building Performance Simulation Association, Eindhoven, The Netherlands.
- Sahlin, P., Eriksson, L., Grozman, P., Johnsson, H., Shapovalov, A. and Vuolle, M.: 2004, Whole-building simulation with symbolic DAE equations and general purpose solvers, *Building and Environment* **39**, 949–958.
- Sahlin, P. and Sowell, E. F.: 1989, A neutral format for building simulation models, *Proceedings of 1st International IBPSA Conference*, International Building Performance Simulation Association, Vancouver, Canada.
- Sang, J. C., Follen, G., Kim, C. and Lopez, I.: 2002, Development of CORBA-based engineering applications from legacy Fortran programs, *Information and Software Technology* **44**(3), 175–184.
- Sargent, R. G.: 2005, Verification and validation of simulation models, *Proceedings of 2005 Winter simulation conference*, The Institute of Electrical and Electronics Engineers (IEEE), Orlando, Florida, USA.
- Sowell, E. F., and Haves, P.: 1999, Numerical performance of the SPARK graph-theoretic simulation program, *Proceedings of 6th International IBPSA Conference*, International Building Performance Simulation Association, Kyoto, Japan.
- Sowell, E. F. and Moshier, M. A.: 1995, HVAC component model libraries for equation-based solvers, *Proceedings of 4th International IBPSA Conference*, International Building Performance Simulation Association, Madison, WI, USA.

## REFERENCES

---

- Sowell, E. F., Moshier, M. A. and Erdem, E.: 2001, Automatic unit and property conversion in SPARK, *Proceedings of 7th International IBPSA Conference*, International Building Performance Simulation Association, Rio de Janeiro, Brazil.
- Sowell, E. F., Moshier, M. A., Haves, P. and Curtin, D.: 2004, Graph-theoretic methods in simulation using SPARK, *Proceedings of High performance computing symposium of the advanced simulation technologies conference*, Society for Modeling Simulation International, Arlington, VA.
- Spitler, J.: 2008, School of Mechanical Engineering, Oklahoma State University, Stillwater, OK, personal communication.
- Strassburger, S.: 2001, *Distributed simulation based on the high level architecture in civilian application domains*, PhD thesis, Otto-von-Guericke Universitat, Magdeburg.
- Strassburger, S.: 2004, Component-based simulation using HLA. Available at: [http://wwwisg.cs.uni-magdeburg.de/~strassbu/papers/Dagstuhl104\\_StrComponentBased.pdf](http://wwwisg.cs.uni-magdeburg.de/~strassbu/papers/Dagstuhl104_StrComponentBased.pdf) [Accessed: Jun, 2006].
- Struler, E., Hoefliger, J., Kale, L. V. and Bhandarkar, M.: 2000, A new approach to software integration frameworks for multi-physics simulation codes, *Proceedings of IFIP TC2/WG2.5 Working Conference on Architecture of Scientific Software*, Ottawa, Canada, pp. 87–104.
- Tacic, I. and Fujimoto, R. M.: 1998, Synchronized data distribution management in distributed simulations, *Proceedings of the 12th workshop on parallel and distributed simulation*, IEEE Computer Society, Banff, Alberta, Canada, pp. 108–115.
- Taylor, S. J. E., Bruzzone, A., Fujimoto, R. M., Gan, B. P., Strassburger, S. and Paul, R. J.: 2002, Distributed simulation and industry: Potentials and pitfalls, *Proceedings of 2002 Winter simulation conference*, The Institute of Electrical and Electronics Engineers (IEEE), San Diego, California, USA.
- Thomas, V. C.: 2006, Using M-E design programs (some reasons for the lack of progress), *Proceedings of IBPSA-USA*, International Building Performance Simulation Association, Chicago, USA.
- Tiller, M. M.: 2001, *Introduction to Physical Modeling with Modelica*, Norwell, Mass.: Kluwer Academic Publisher.
- Trčka-Radošević, M. and Hensen, J. L. M.: 2006, Distributed simulation of building systems for legacy software reuse, *Proceedings of 6th International Postgraduate Research Conference, International Built and Human Environment Research Week*, Technishce Universiteit Delft, BuHu, University of Salford, pp. 442–454.
- 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.

- Vaculín, O., Krüger, W. R. and Valášek, M.: 2004, Overview of coupling of multi-body and control engineering tools, *Vehicle System Dynamics* **41**(5), 415–429.
- Vandaele, L. and Wouters, P.: 1994, The passys services summary report, *Technical report*, Comission of the European Communities, DG XII for Science, Research and Development, Brussels, Belgium.
- Wang, S. and Ma, Z.: 2008, Supervisory and optimal control of building HVAC systems: A review, *HVAC & R Research journal* .
- 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., Koschenz, M., Holst, S., Hiller, M. and Welfonder, T.: 2002, TRNFLOW: Integration of COMIS into TRNSYS TYPE 56, *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.: 2001, GenOpt – A generic optimization program, *Proceedings of 7th International IBPSA Conference*, International Building Performance Simulation Association, Rio de Janeiro, Brazil.
- Wetter, M.: 2006, Multizone building model for thermal building simulation in Modelica, *Proceedings of Modelica Conference at arsenal research*, Vienna, Austria.
- Wetter, M.: 2008, Hybrid ventilation with evaporative cooling and run-around heat recovery. Available at: <https://gaia.lbl.gov/virBui/UseCases/RapidSystemConfiguration> [Accessed: February, 2008].
- Wetter, M. and Haugstetter, C.: 2006, Modelica versus TRNSYS - A comparison between an equation-based and a procedural modeling language for building energy simulation, *Proceedings of the SimBuild, 2nd National Conference of IBPSA-USA*, International Building Performance Simulation Association, USA chapter, Cambridge, MA, USA.
- Wetter, M. and Haves, P.: 2008, A modular building controls virtual test bed for the integration of heterogeneous systems, *Proceedings of SimBuild, 3rd National Conference of IBPSA-USA*, International Building Performance Simulation Association, USA chapter, Bekeley, CA, USA.
- 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.
- Wright, A. J., Bloomfield, D. and Wiltshire, T. J.: 1992, Building simulation and building representation: Overview of current developments, *Building Serv. Eng. Res. Technol* **13**(1), 1–11.

## REFERENCES

---

- Yahiaoui, A., Hensen, J. L. M. and Soethout, L.: 2003, Integration of control and building performance simulation software by run-time coupling, *Proceedings of 8th International IBPSA Conference*, International Building Performance Simulation Association, Eindhoven, The Netherlands.
- Yahiaoui, A., Hensen, J. L. M. and Soethout, L.: 2004, Developing CORBA-based distributed control and building performance environments by run-time coupling, *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.
- Yoon, J. H., Lee, E. J. and Hensen, J. L. M.: 1997, Integrated thermal analysis of a three story experimental building with a double-skin and a ground-coupled heat exchanger, *Proceedings of Int. Solar Energy Conference SOLAR 97 : Energy for Sustainable Prosperity*, American Soc. of Mechanical Engineers, ASME/ASES, Washington, DC.
- Zeigler, B. P.: 1976, *Theory of modeling and simulation*, John Wiley & Sons, Canada.
- Zeigler, B. P., Praehofer, H. and Kim, T. G.: 2000, *Theory of modeling and simulation, second edition*, Academic press, San Diego, USA.
- Zhai, Z.: 2003, *Developing an Integrated Building Design Tool by Coupling Building Energy Simulation and Computational Fluid Dynamics Programs*, PhD thesis, Massachusetts Institute of Technology, USA.

# Curriculum vitae

Marija Trčka (born Radošević) was born on the 19th of November 1977 in Belgrade, Serbia (then Yugoslavia). She studied Thermal Sciences at the Faculty of Mechanical Engineering, University of Belgrade, Serbia. During the studies she was fortunate to be offered two internships, at Danish Technological Institute, Århus, Denmark (2000-2001), where she was introduced to research for the first time, and at Axima Refrigeration GmbH, Lindau, Germany (2002). For her research in Denmark, Marija was awarded the ASHRAE Homer Addams Award as the best graduate student of the year participating in one of the ASHRAE projects. After graduating as a Valedictorian in 2002, Marija joined a postgraduate program developed in cooperation with the Norwegian University of Science and Technology, Trondheim, Norway (2002-2003). In July 2003, she started her Ph.D research in the Building Physics and Systems group, at the Department of Architecture, Building and Planning, Eindhoven University of Technology, The Netherlands. During her Ph.D studies, Marija spent six months at the United Technologies Research Center in East Hartford, Connecticut (2006-2007). After she obtains her Ph.D degree, Marija will stay as a postdoc in the same group in Eindhoven.