Energy Research and Development Division
FINAL PROJECT REPORT

ENERGY EFFICIENCY PROJECTS

Enabling Tools for Design of Energy Efficient Building Systems

Prepared for: California Energy Commission
Prepared by: Lawrence Berkeley National Laboratory

MARCH 2013
CEC-500-2014-055

PREPARED BY:

Primary Author(s):
Michael Wetter
Xiufeng Pang
Wangda Zuo
Thierry Nouidui
Kaustubh Phalak

Lawrence Berkeley National Laboratory
Environmental Energy Technologies Division
Building Technology and Urban Systems Department
Simulation Research Group
One Cyclotron Rd
Berkeley, CA, 94720

Contract Number: 500-10-052

Prepared for:

California Energy Commission

Heather Bird
Contract Manager

Felix Villanueva
Project Manager

Virginia Lew
Office Manager
Energy Efficiency Research Office

Laurie ten Hope
Deputy Director
ENERGY RESEARCH AND DEVELOPMENT DIVISION

Robert P. Oglesby
Executive Director

DISCLAIMER

This report was prepared as the result of work sponsored by the California Energy Commission. It does not necessarily represent the views of the Energy Commission, its employees or the State of California. The Energy Commission, the State of California, its employees, contractors and subcontractors make no warranty, express or implied, and assume no legal liability for the information in this report; nor does any party represent that the uses of this information will not infringe upon privately owned rights. This report has not been approved or disapproved by the California Energy Commission nor has the California Energy Commission passed upon the accuracy or adequacy of the information in this report.

ACKNOWLEDGEMENTS

We thank the following people for serving on the Technical Advisory Group:

Aaron Wintersmith – Capital Engineering
Alastair Robinson – LBNL
Brian Coffey – Atelier Ten
Chris Scruton – California Energy Commission
Dove Feng – UC Berkeley
Fred Baumann – UC Berkeley
Jian Sun – UC Merced
Pete Shoemaker – PGE
Shruti Kasareka – Atelier Ten
Siyu Wu - UC Merced
Spencer Dutton – LBNL

For the case study at the Kirsch Center, which is presented in Chapter 5, the authors thank Pat Cornely, Rick Andrews and Jim Ma from De Anza College for offering the facility for the case study, and thank Cole Robert from ARUP for coordination and providing drawings.

This work used and further developed Modelica models from the *Buildings* library whose development was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Building Technologies Program of the U.S. Department of Energy under Contract No. DE- AC02-05CH11231. We thank the U.S. Department of Energy for their support.

PREFACE

The California Energy Commission Energy Research and Development Division supports public interest energy research and development that will help improve the quality of life in California by bringing environmentally safe, affordable, and reliable energy services and products to the marketplace.

The Energy Research and Development Division conducts public interest research, development, and demonstration (RD&D) projects to benefit California.

The Energy Research and Development Division strives to conduct the most promising public interest energy research by partnering with RD&D entities, including individuals, businesses, utilities, and public or private research institutions.

Energy Research and Development Division funding efforts are focused on the following RD&D program areas:

- Buildings End-Use Energy Efficiency
- Energy Innovations Small Grants
- Energy-Related Environmental Research
- Energy Systems Integration
- Environmentally Preferred Advanced Generation
- Industrial/Agricultural/Water End-Use Energy Efficiency
- Renewable Energy Technologies
- Transportation

Energy Efficiency Projects - Enabling Tools for Design of Energy Efficient Building Systems is the final report for the National Lab Buildings Energy Efficiency Research Projects (contract number 500-10-052), conducted by the Lawrence Berkeley National Laboratory. The information from this project contributes to Energy Research and Development Division's Environmental Research Program.

For more information about the Energy Research and Development Division, please visit the Energy Commission's website at www.energy.ca.gov/research/ or contact the Energy Commission at 916-327-1551.

ABSTRACT

To attain California's energy efficiency goals for new and existing commercial buildings, building design and operation will need to change substantially. The lengthy time required to add new modeling capabilities to conventional whole building energy simulation programs is inconsistent with the quick turnaround necessary to meet the State's timetable of energy goals, or to influence construction project design decisions.

To support this need, we developed a free, modular, open-source library of component and system models; the Modelica *Buildings* library. A key capability allows users to rapidly prototype innovative HVAC systems, and, in particular, design and test the performance of actual supervisory and local loop control algorithms.

This project leveraged and further extended the Modelica *Buildings* library. New models have been added, models have been validated, tutorials and user guides have been developed and case studies have been conducted to validate, test the usability and demonstrate the capabilities of the library.

In case study one, a model was developed and calibrated to analyze control options for night ventilation and radiant slab cooling for a class room at the Kirsch Center in Cupertino, CA. The main finding was that with properly scheduled night-time, natural ventilation, no mechanical cooling would be needed. Radiant slab evaporative cooling during the hottest hours of the year would lead to slightly improved occupant comfort. In case study two, parametric studies were conducted using weather data for San Francisco, CA, for a chilled water plant with and without water-side economizing, and trim and respond control sequencing for a typical data center. The main findings were that total facility power consumption would be reduced by 10% through the use of a water-side economizing, and thatsavings are sensitive to the supply air temperature and chiller leaving water temperature.

All of the figures were provided by the authors from Lawrence Berkeley National Lab.

Keywords: simulation-based design, operational optimization, building energy systems, Modelica, trim and respond control sequence

Please use the following citation for this report:

Wetter, Michael, Xiufeng Pang, Wangda Zuo, Thierry Nouidui, Kaustubh Phalak. Lawrence Berkeley National Laboratory. 2013. ***Enabling Tools for Design of Energy Efficient Building Systems.*** California Energy Commission. Publication number: CEC-500-2014-055.

TABLE OF CONTENTS

Acknowledgements i

PREFACE ii

ABSTRACT iii

TABLE OF CONTENTS iv

LIST OF FIGURES v

LIST OF TABLES vii

EXECUTIVE SUMMARY 1

CHAPTER 1: Introduction 4

CHAPTER 2: Approach 6

2.1 Requirements As Prioritized By Technical Advisory Group 7

CHAPTER 3: Technical Work 8

3.1 Buildings Library 8

3.2 Component Model Development 11

CHAPTER 4: Documentation 14

4.1 User Guide 14

4.1.1 Getting Started 14

4.1.2 Best Practice 15

4.1.3 Work-Arounds 28

4.1.4 Pre- and Post-Processing 33

4.1.5 Development 33

4.2 Tutorials 35

4.2.1 Hydronic Heating System Tutorial 36

4.2.2 Air-based HVAC System Tutorial 38

CHAPTER 5: Case studies 41

5.1 Kirsch Center 41

5.1.1 Facility Information 41

5.1.2 Model Development 47

5.1.3 Model Calibration & Validation 50
5.1.4 Analysis 53
5.1.5 Conclusion 59
5.2 Chilled Water Plant for a Data Center 59
5.2.1 Problem Description 59
5.2.2 Model of the Plant 59
5.2.3 Model of the Control 60
5.2.4 System Performance 64
GLOSSARY 70
REFERENCES 71

LIST OF FIGURES

Figure 1: Schematic Diagram and Modelica System Model of the Chilled Water Plant with Wse 3
Figure 2: Pues of Chilled Water Plant Systems with and without Wse 3
Figure 3: Resistor Capacitor Network for the Radiant Slab Model. Along the Flow Path, Multiple Instances of This Model Are Used to Take Into Account the Change In Fluid Temperature 11
Figure 4: Schematic View of Overhang and Side Fin Models 12
Figure 5: Room Air Temperature and Absolute Humidity for Different Direct Evaporating Coils 12
Figure 6: Modelica Model of Bestest Case 600 13
Figure 7: Schematic Diagram of a Flow Source, a Fluid Volume, and a Pressure Source 17
Figure 8: Connection of Three Components without Explicitly Introducing a Mixer or Splitter Model 20
Figure 9: Correct (Left) and Wrong (Right) Connection of Components with Use of a Mixer or Splitter Model 21
Figure 10: Schematic Diagram of a Flow Circuit without Means to Account for the Thermal Expansion 22
Figure 11: Schematic Diagram of a Flow Circuit with Expansion Vessel that Adds a Pressure Source and Accounts for the Thermal Expansion of the Medium 23

Figure 12: Schematic Diagram of a Flow Circuit with a Boundary Model that Adds a Fixed Pressure Source and Accounts for Any Thermal Expansion of the Medium 24

Figure 13: Schematic Diagram of a Controller That Switches a Coil On And Off. 27

Figure 14: Schematic Diagram of Fans That are Configured with Filterspeed=False (Fans) and Filterspeed=True (Fanc) 29

Figure 15: Mass Flow Rate of the Two Fans For a Step Input Signal at 0 Seconds 29

Figure 16: Schematic Diagram of Two Flow Resistances in Series That Connect a Source and a Volume 30

Figure 17: Schematic Diagram of a Fixed Flow Resistance and a Valve in Series That Connect a Source and a Volume 31

Figure 18: Schematic Diagram of a Valve That Connects a Source and a Volume 32

Figure 19: Schematic Diagram of Hydronic Heating System Used in the Tutorial 36

Figure 20: Modelica Implementation of the Hydronic Heating Model Used in the Tutorial 37

Figure 21: Temperature Trajectories for the Hydronic Heating System Model 38

Figure 22: Schematic Diagram of HVAC System that is Used in the Second Tutorial 38

Figure 23: Modelica Implementation of the Hydronic Heating Model used in the Tutorial 39

Figure 24: Temperature Trajectories for the Hydronic Heating System Model 40

Figure 25: The Kirsch Center 41

Figure 26: The Biodiversity Lab 42

Figure 27: Schematic Diagram of the Hydronic System 44

Figure 28: Hot Water Supply Temperature Reset Schedule 45

Figure 29: Radiant Slab Water System 46

Figure 30: Radiant Slab Supply Water Temperature Reset Schedule 47

Figure 31: Modelica System Model with Thermal Zone, Radiant Slab and Air Exchange 48

Figure 32: Hydronic System Model 48

Figure 33: Measured Room Temperature, Slab Temperature, Outside Air Temperature and Indoor CO2 Concentration 51

Figure 34: Comparison of Measured and Simulated CO2 52

Figure 35: Comparison of Measured and simulated Room Air and Slab Temperature 52

Figure 36: The Outside Air Temperature Profile during Summer and September in San Jose TMY3 54

Figure 37: Predicted CO_2 and Room Air Temperature during the Coldest Period in Summer for Proposal #1 55

Figure 38: Predicted CO_2 Concentration during the Hottest Period in Summer for Proposal #1. 56

Figure 39: Predicted CO2 and Room Air Temperature during the Coldest Period in Summer for Proposal #2 57

Figure 40: Predicted CO2 and Room Air Temperature during the Hottest Period in Summer for Proposal #2 57

Figure 41: Predicted Room Air and Radiant Slab Temperature during Hottest Period in Summer for Proposal #3 58

Figure 42: Schematic Diagram and Modelica System Model of the Chilled Water Plant with WSE 61

Figure 43: State Graphs of the Chiller and WSE 62

Figure 44: Chilled Water Loop Differential Pressure and Chiller Setpoint Temperature Reset ... 64

Figure 45: Power Usage Effectiveness with and without Water-side Economizer 64

Figure 46: PUEs of Chilled Water Plant Systems with and without WSE 66

Figure 47: Annual Frequency Distributions of the chiller status as a Function of the Wet Bulb Temperature for Different Control Setpoints 66

Figure 48: Trajectories for the System with WSE 67

LIST OF TABLES

Table 1: Building Envelope Constructions 49

Table 2: Radiant Slab System Schedule Serving the Biodiversity Lab 50

Table 3: Hydronic System Equipment Schedule 50

EXECUTIVE SUMMARY

Through two case studies, a modular simulation approach was developed and demonstrated. These simulations were built on user-extensible models that can analyze the performance of actual building control sequences for non-conventional building energy and control systems.

This technology has been developed in anticipatiion of the need for building design and operation to change substantially to meet California's energy efficiency goals for new and existing commercial buildings. The lengthy time required to add new modeling capabilities to conventional whole building energy simulation programs is inconsistent with the quick turnaround necessary to meet the timetable of the State's energy goals, or to influence construction project design decisions. Moreover, conventional building simulation programs are difficult to extend and primarily contain models of established technologies that have been on the market long enough to be implemented by a software developer in these programs. This approach does not meet the needs of design firms that want to test the performance of innovative emerging technologies, test new arrangements for heat recovery and energy harvesting, and develop and test innovative strategies for controlling buildings as a system at a pace that is compatible with the building delivery cycle.

The approach is to transfer to the buildings industry a new generation of modeling and simulation technology that has experienced significant growth in other industries and that also faces the need for rapid technology development and increased system integration by means of energy recovery, energy conversion and integrated controls, e.g. the hybrid automotive industry or the aerospace industry. This modeling and simulation technology is based on open, non-proprietary modeling language Modelica that has been developed to support the design and operation of complex engineered systems. It allows users to quickly add new component models and combine existing components in an arbitrary way as long as the connections are physically consistent.

In this project, the Modelica Buildings library has been developed, validated and demonstrated the usability, as a free open-source modeling library for building energy and control systems. New component models were also developed and validated them through comparative model validation and through measurements in a building. The documentation of a variety of models was substantially improved and extended. To aid users in learning this new approach, a user guide was developed that helps users getting started, familiarizes them with best practices, provides work around to common problems, and provides instructions for how to add new component models. To explain to users how to best build new system models based on existing component models, tutorials were developed that provide step-by-step instructions for how to construct a model.

Lastly, two cases studies were conducted. One case study was conducted in the Kirsch Center at De Anza College in Cupertino, CA, which is a LEED Platinum rated building. The objective was to optimize the operation of the night ventilation and radiant cooling slab for a class room. The second case study was for a hypothetical data center in San Francisco, CA. The objective was to

optimize the set points for the chilled water plant, and to compare the system-level performance with and without water-side economizer.

For the case study at the Kirsch Center, a model of a class room with natural ventilation and radiant slab was calibrated and used to deduce a schedule for the window operation for night venting. The lack of detailed measurements of lighting and equipment loads, the uncertainty introduced by opening the door to the hallway, which affected the CO_2 levels that had to be used to infer occupancy, and the lack of local weather data provided challenges for the model validation. The actual implemented control sequences deviated from the design intent. The measured data indicate that the operable windows were not operated to utilize the cross ventilation capability. This may be due to the situation that the owner rejected the concept of automatically opening windows to allow active participation of students, staff and faculty in the operation of the building in order to learn how it behaves. The operation of the radiant slab system can lead to a waste of pumping energy since the operation of the zone pump is disconnected from the operation of the hot water and chilled water switch valves. The major finding was that with properly scheduled night-time natural ventilation the room temperatures can be maintained within the comfort zone without the use of mechanical cooling. The use of the radiant cooling slab, cooled using evaporative cooling, further improves comfort during the hottest hours of the year. Recommendations have been provided to the operator in terms of window openings as a function of the forecasted maximum outdoor temperature of the next day to allow manual operation of the windows.

Figure 1 shows the schematic diagram for the case study that analyzes the system-level performance of a chilled water plant with and without economizer. The independent variables were the setpoint, $T_{air,set}$, for the air supply temperature, $T_{air,\,sup}$, and the maximum temperature of the water that leaves the evaporator, $T_{chi,max}$. The actual temperature was determined by a trim and respond logic that cascades the set points for the water flow rate and temperature. The performance metric was the Power Usage Effectiveness (PUE), which is the ratio of the total facility power usage, including power usage for the Information Technology (IT) and cooling system, divided by the power usage for the IT. A lower PUE indicates a more energy efficient cooling system. The major finding was that a water-side economizer reduces the PUE by about 10% and that the PUE is sensitive to the above set points. For a *500* kW data center, these savings would yield around *$90,000* annual reduction in electricity costs. Figure 2 shows that the system level performance is sensitive to chilled water temperatures and supply air temperatures in the upper right quadrant where both are high. Conversely, raising the supply air temperature has little benefit unless the chilled water temperature is also high, thus increasing the performance of the chiller and the hours when free cooling can be provided. The variability in the sensitivity is a complex combination of the frequency distribution of the hours where free cooling can be provided, and the set point adjustments of the trim and repond logic. The contour plots demonstrate that with simulations, a large trade-space can be analyzed. Were the analysis done at a real plant that operates at or below a maximum chilled water leaving temperature of about 10°C, one may have drawn wrong conclusions as to the benefit of an increased supply air temperature.

Figure 1: Schematic Diagram and Modelica System Model of the Chilled Water Plant with Wse

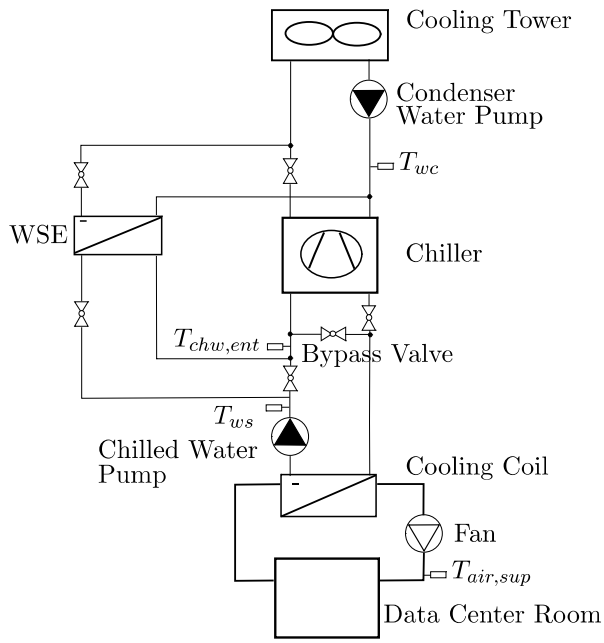
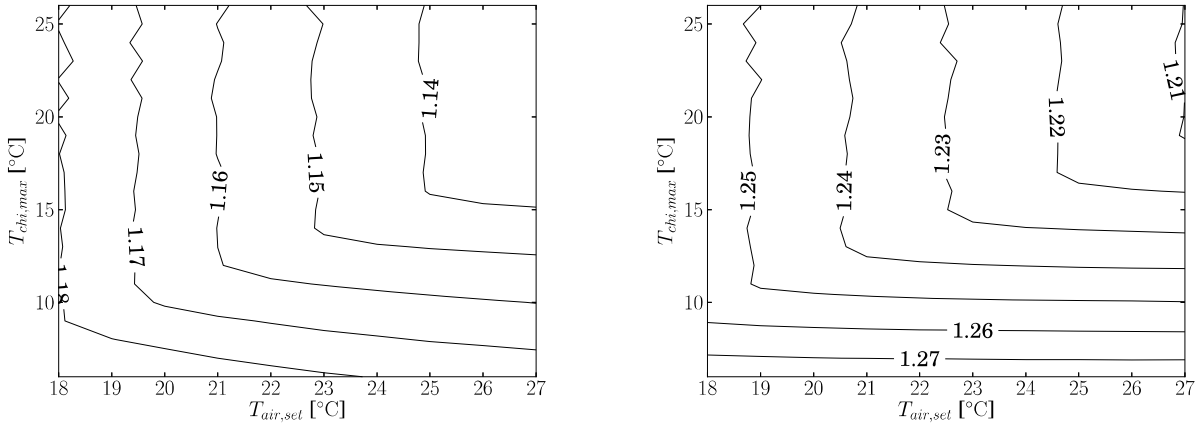


Figure 2: Pues of Chilled Water Plant Systems with and without Wse



(a) With WSE (b) Without WSE

CHAPTER 1: Introduction

To attain California's energy efficiency goals for new and existing commercial buildings, building design and operation will need to change substantially. The lengthy time required to add new modeling capabilities to conventional whole building energy simulation programs is inconsistent with the quick turnaround necessary to meet the State's timetable of energy goals, or to influence construction project design decisions. Better thermal integration and building controls offer a high return on investment; however, design firms lack the tools necessary to test reductions in energy use and peak power demand. Without such tools, these system-level solutions cannot be rapidly employed to reduce demand and increase supply efficiency.

Today's building simulation programs contain models of established technologies that have been on the market long enough to be implemented by a software developer in these programs. However, innovative systems focus on new ways to recover heat, harvest energy, and control equipment. Such systems are not available in today's building simulation programs. Design firms need a way to add such technologies, test new arrangements for heat recovery and energy harvesting, and develop and test new strategies for controlling buildings as a system, at a pace that is compatible with the building delivery cycle.

In addition, energy simulation programs are designed for whole-building energy analysis and are rather monolithic, and operational-tools for measurement and verification almost exclusively rely on statistical methods. Users would benefit from flexible energy modeling tools that can be used in combination with these statistical tools to identify expected performance, and to run automated scenarios at the system and subsystem-level, to learn why more energy is used than anticipated.

The private market has not created tools that adequately allow designers to test the performance of new systems and reduce the risk of adopting new systems for a variety of reasons:

- The few HVAC consulting firms that have invested in the development of customized in-house tools are generally not interested in offering these tools to competing companies.
- Software development companies are targeting their development to more standard systems that represent a larger part of the building segments. Furthermore, modeling of control algorithms requires fundamental changes in today's building simulation programs that were not recognized when conventional building simulation programs such as DOE-2 and EnergyPlus were designed.
- Private companies have neither the incentive nor the networks needed to coordinate and leverage the global development of open-source software that can be used for building energy and control system modeling and simulation.

The goal of this project was to accelerate the invention, development, design, and rate of adoption of very-low-energy building systems. The objective was to address the above needs through the further development, documentation, and validation of the free open-source Modelica *Buildings* library, a modular library for building energy and control systems that is implemented in the Modelica language. Modelica is an open standardized language that allows graphical and textual modeling. The Modelica language is equation-based (input/output free) and object-oriented and can represent differential equations, algebraic equations and discrete equations. It has been developed for modeling, simulation, and optimization of complex engineered systems and their feedback controls.

Within this project, new models were added based on feedback from a Technical Advisory Group. Existing models, which were implemented as a proof-of-concept in an LBNL-internal funded project, have been validated and their documentation has been expanded. Furthermore, user guides and tutorials have been written to show new users how to use the *Buildings* library to implement models that can be used to optimize HVAC systems and their control sequences.

CHAPTER 2: Approach

This project built upon and extended previous developments that led to an open-source modeling and analysis infrastructure with components for, and systems of, very-low-energy building technologies.[1] This model infrastructure is built on Modelica, an open-source modeling language that is used for the design and operation of various industrial applications. These models can be deployed on central computers for large buildings, as well as on embedded systems with limited processing and storage for small commercial and residential buildings that are more sensitive to hardware costs. Our extension of this infrastructure was aimed at the following:

- Enabling design firms to rapidly test new ways of integrating heat recovery into systems and integrating ambient sources and sinks for heating and cooling through thermal integration and advanced controls.
- Enabling design firms to develop and test the performance of control sequences such as those for the integration of renewable energy systems, chiller sequencing, and fan static pressure reset.
- Enabling design firms to rapidly add models of new components and systems without a software developer. This will allow design firms to expand the tools themselves, so they can test new system-integration ideas more quickly—even for little-known or not-well-understood systems.
- Enabling measurement and verification tool developers and providers to add physics-based models of products, subsystems, and whole-building systems to augment their statistical methods with physics-based models that compute the expected performance under various operational scenarios.

The modeling infrastructure was also further developed for the analysis of control sequences for standard heating, ventilation and air conditioning (HVAC) systems, either linked to EnergyPlus or as a stand-alone tool.

A Technical Advisory Group (TAG), was established to ensure that the models developed meet the needs of design firms and support future Title 24 standards. Input from design firms was obtained to prioritize the addition of models to the existing open-source *Buildings* library and to expand requirements used to develop this library. Additional documentation was written to move the research version of the *Buildings* library from research groups to design firms. Usability testing on actual building projects provided input to the development of models and documentation.

[1] See http://simulationresearch.lbl.gov/modelica

2.1 Requirements As Prioritized By Technical Advisory Group

Two TAG meetings were held in the first and third quarter of the project. During these meetings previous work was reviewed, use cases and requirements discussed and future work prioritized. A high priority item that was identified in these meetings was the expansion of the documentation of the library. A specific request was made for developing a tutorial with step-by-step instructions for how to build system models using exising components. Also, development of a user guide with best practices has been recommended. The tutorial and the user guide are described in Chapter 4 of this report and are available online[2].

The TAG also recommended focusing the project on controls design, in particular of building systems that use thermal mass, night ventilation, and/or radiant slabs. The case studies in Chapter 5 demonstrate such applications. The first case study uses a class room with night ventilation and a radiant slab for cooling. The second case study demonstrates the optimization of a control sequence that exploits the free-cooling potential of the ambient air through the use of a water-side economizer for a chilled water plant.

[2] See http://simulationresearch.lbl.gov/modelica/releases/v1.2_build1/help/Buildings_Examples_Tutorial.html and http://simulationresearch.lbl.gov/modelica/userGuide/

CHAPTER 3: Technical Work

3.1 Buildings Library

The Modelica *Buildings* library has been further developed by this project and used for the case studies in this project.

The *Buildings* library is an open-source, free library with component and system models for building energy and control systems. The library is accompanied by Python modules that can be used to automate simulations and post-processing results, and by a user guide.

The library has been developed to support the following use cases in addition to conventional energy analysis:

Support for rapid prototyping: Users can rapidly add new component and system models by modifying existing models or extending basic models that implement, for example, steady-state and dynamic energy and mass balances of a heat and mass exchanger.

Modeling of arbitrary HVAC system topologies: Users can model HVAC systems with non-conventional piping or ducting layout, and user-defined control algorithms. For example, the application in Wetter (2009a) shows the implementation of a novel hydronic heating system where pumps located at each radiator replace the central pump, and a hierarchical controller adjusts the furnace temperature based on the pump speed.

Development, verification, and hardware-in-the-loop testing of control algorithms: Users can use models from the library to support development of model-based control algorithms, and then verify the control sequences in a hardware-in-the-loop configuration. This process is described in Nouidui et al. (2012), where a thermal model of a room has been linked to Radiance and to a test cell in order to optimize the control of the blinds. For an example where a hierarchical supervisory and local loop control algorithm has been verified in simulation, see the implementation of the variable air volume flow control sequence from the American Society of Heating, Refrigerating and Air-Conditioning Engineers. (ASHRAE) Sequences of Operation for Common HVAC Systems (ASHRAE 2006) described in Wetter (2011). In this example, the supervisory control algorithm is implemented using a state graph and the local loop control is implemented using block diagrams.

Stabilization of feedback control: Users can start with a process model that may be non-linear and based on first principles. Next, they can linearize the model around a set of steady-state operating points in order to analyze in the frequency domain the gain and phase margins, such as in Eisenhower et al (2012). Alternatively, the linearized models may be used as input to a model order reduction algorithm. Model order reduction algorithms compute a low order model that is suitable for gain scheduling in the frequency domain. See Wetter (2009b) for such an application.

Emulation of faults at the whole building-system level: For example, the LearnHVAC[3] and Learn High Performance Buildings, Deringer et al (2012) interactive eLearning software, allows testing a variety of fault scenarios and examining the impact of faults on whole building performance. In these tools, models of faulty equipment or faulty controllers are part of a dynamic whole building model that includes HVAC, building envelope, and detailed control algorithms whose feedback loops may compensate for some of the faults.

The above list represents applications in which the *Buildings* library has already been used. Work is currently in progress to use models from the Buildings library in fault detection and diagnostics algorithms, and to build models for model-based design of full-scale experiments at the Facility for Low Energy Experiments in Buildings[4] that is currently under construction at the Lawrence Berkeley National Laboratory (LBNL).

The Buildings library version 1.2 build1 contains dynamic and steady-state models for building energy and control systems. The library contains models for air-based HVAC systems, water-based heating systems, controls, heat transfer among rooms and the outside, and multizone airflow, including natural ventilation and contaminant transport.

A detailed description of all models can be found online at http://simulationresearch.lbl.gov/modelica.

The models are organized in the following main packages:

User's Guide: This package contains a user's guide for the library. In addition, some individual packages contain additional user's guides that are applicable to components of the respective packages.

Airflow: This package contains models for multizone airflow and contaminant transport. The models compute air flow between adjacent rooms and between rooms and the outside. Each room volume is assumed to be completely mixed. The driving forces for the air flow are pressure differences that can be induced, for example, by a flow imbalance of the HVAC system, density differences across large openings such as doors or open windows, stack effects in high rise buildings, and wind pressure on the building facade.

BoundaryConditions: This package contains models to read TMY3 weather data and to compute boundary conditions, such as solar irradiations and sky temperatures.

Controls: This package contains blocks that model continuous time and discrete time controllers. There are also blocks that can be used to schedule setpoints.

Fluid: This is the largest package of the library. It contains component models for air-based and water-based HVAC systems. The package contains models such as chillers, cooling towers, heat

[3] See http://www.learnhvac.org/.

[4] See http://utbf.lbl.gov/.

exchangers, flow resistances, valves and air dampers, mass flow boundary conditions, pressure boundary conditions, pumps and fans, sensors, and energy storage.

HeatTransfer: This package contains models for steady-state and dynamic heat transfer through opaque constructions such as multi-layered walls. It also contains models for layer-by-layer heat transfer through fenestration systems that are similar to the algorithms of the WINDOW 6 program.

Media: This package contains different implementations for various media. The media models in this package are compatible with models of the Modelica.Media package, but the implementation is in general simpler, which often leads to fewer nonlinear equations and system models that are more robust. Due to the simplifications, the media model of this package is accurate for a smaller temperature range than the models in Modelica.Media, as the smaller temperature range generally suffices for building HVAC applications.

Rooms: This package contains models for heat transfer in rooms and through the building envelope. Multiple instances of these models can be connected to create a multi-zone building model. To compute the air exchange between rooms and between a room and the exterior, the room models can be connected to multi-zone air exchange models from the package Buildings.Airflow. The room models can also be linked to models of HVAC systems that are composed of the components in the package Buildings.Fluid.

Utilities: This package contains utility models, such as for computing thermal comfort, for run-time data exchange with the Building Controls Virtual Test Bed[5] or for computing psychometric properties. The package also contains various mathematical functions that are used throughout the library.

The above packages also contain user's guides, tutorials, and examples. The user's guides explain, for models of the respective package, what their underlying assumptions, limitations, and proper use are.

In total, the Buildings library contains 236 models and blocks, 94 functions and 279 example models.

Component and system models have been validated using empirical validation, comparative testing, and analytical verification.

Empirical validation, in which models are compared against measurements, has been used to test the window heat transfer and transmitted solar radiation (Nouidui et al, 2012).

Comparative testing, in which results are compared with other simulators, has been used for the following models: In Nouidui et al. (2012), we compared the optical properties and the surface temperatures of window systems with results obtained by the WINDOW 6 program. In Nouidui et al (2012), we present results from the BESTEST validation of the building envelope model. In Wetter (2006), we compare air flow rates for the multizone airflow models with

[5] See http://simulationresearch.lbl.gov/bcvtb.

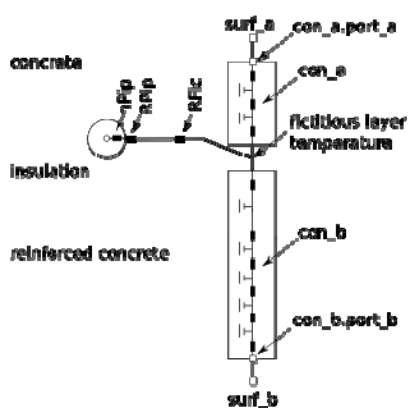
results obtained by the CONTAM program. Analytical verification, in which results are compared with exact solutions, has been used to test individual component models, such as for heat and mass balances, heat and mass exchangers, flow resistance elements, fan models, and radiosity transport models.

3.2 Component Model Development

Within this project, new component models were developed for radiant slabs, shading devices and direct evaporating cooling coils.

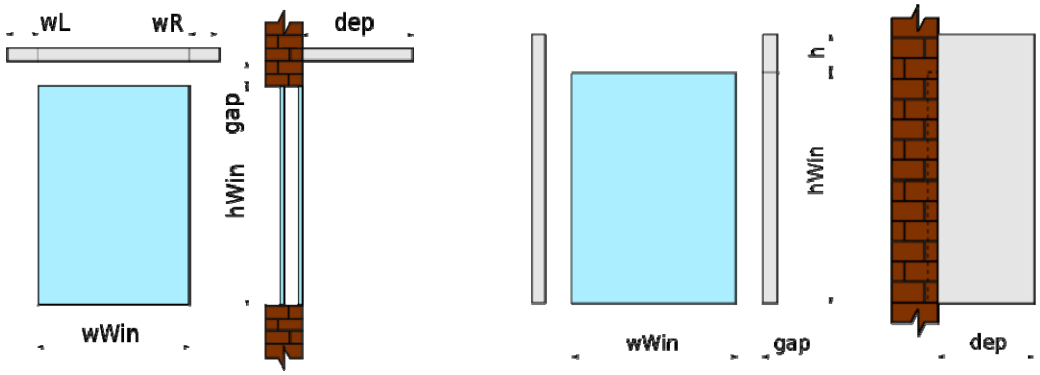
The radiant slab model is a finite difference model that computes transient heat flow between the fluid inside the pipe and the surfaces of the construction. The heat conduction is computed using a finite difference model. The model computes a fictitious temperature, which is an average temperature of the plane that contains the pipes. Figure 3 shows the resistor-capacitor network. The model allows an arbitrary number of these heat transfer computations to be placed along the water flow direction, and the model allows multiple parallel circuits. A detailed description of the model can be found at: http://simulationresearch.lbl.gov/modelica/releases/v1.2_build1/help/Buildings_Fluid_HeatExchangers_RadiantSlabs.html.

Figure 3: Resistor Capacitor Network for the Radiant Slab Model. Along the Flow Path, Multiple Instances of This Model Are Used to Take Into Account the Change In Fluid Temperature



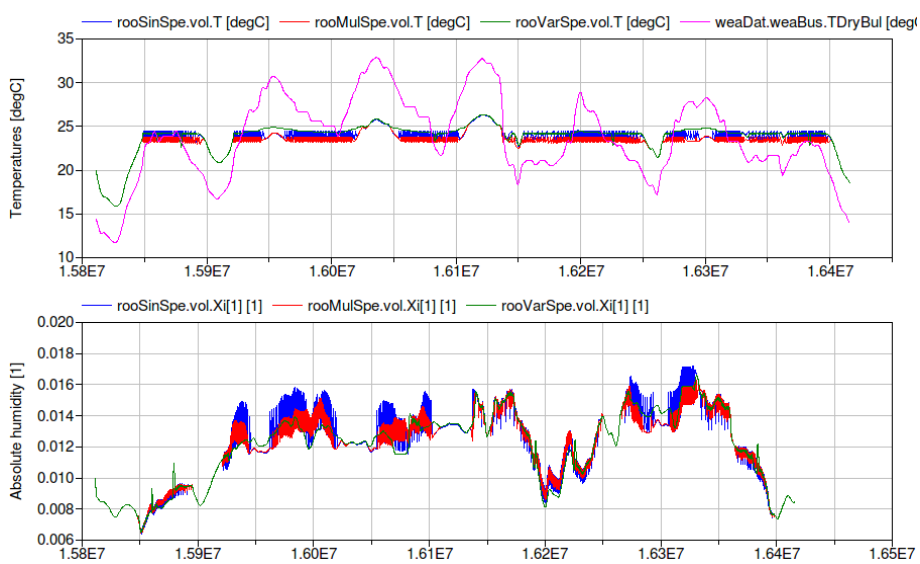
The shading model outputs the fraction of the window area that is sun exposed for a window that may have an overhang and sidefins. Figure 4 shows the configuration of the overhang and the side fins that can be modeled with this component. The model also has been integrated into the room model and is used as part of multizone building models, and it has been validated using the ANSI/ASHRAE BESTEST validation. The detailed model description can be found at http://simulationresearch.lbl.gov/modelica/releases/v1.2_build1/help/Buildings_HeatTransfer_Windows.html

Figure 4: Schematic View of Overhang and Side Fin Models



The direct evaporating cooling coil model can be used to model a coil with variable speed compressor or with a compressor with multiple stages. The model is based on steady-state performance curves. A library of performance curves is available. The model can be configured as a steady-state or dynamic model. The latent degradation due to reevaporation of water from the coil when the coil is off, is computed using a dynamic mass transfer model. The model has been validated using comparative model validation with EnergyPlus. Figure 5 shows, for three models of a simple room with time-varying heat load, the room air temperature and humidity for a coil with single speed (blue), dual-speed (red) and variable speed (green). For the case of a coil with single speed, the room has the highest absolute humidity because of the latent degradation of the coil. The dual-speed and variable speed coil have similar room air because they have similar cycling ratios.

Figure 5: Room Air Temperature and Absolute Humidity for Different Direct Evaporating Coils

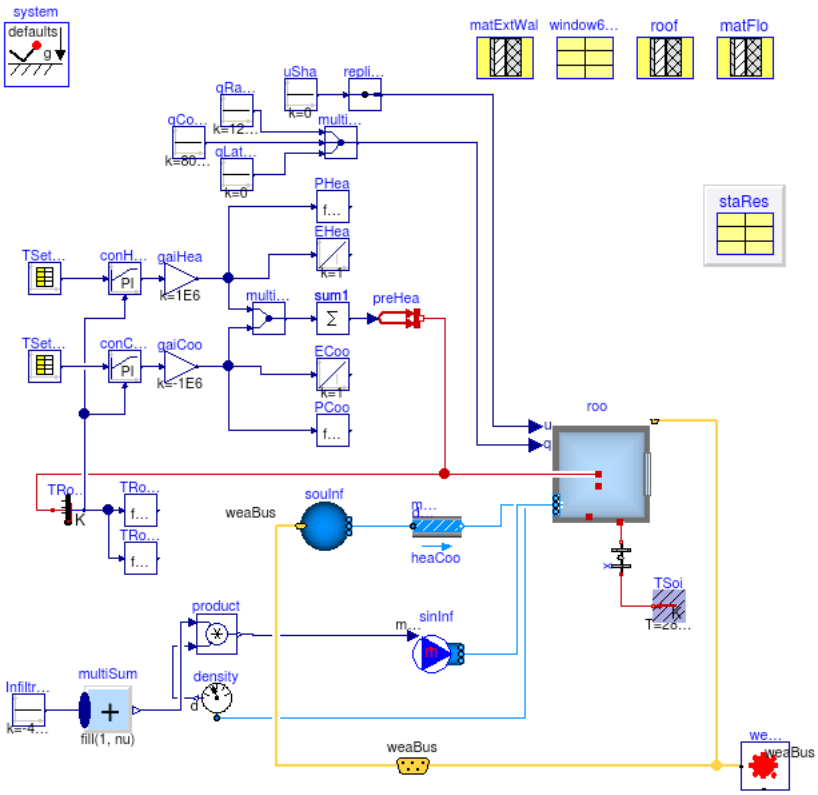


In the course of this project improvements were made to several other models. For example, the fan and pump models (http://simulationresearch.lbl.gov/modelica/releases/v1.2_build1/help/Buildings_Fluid_Movers.html) have been improved to better handle very small mass flow rates. Furthermore, the documentation of models for controllers, multizone air exchange and contaminant transport, fans and pumps, and valves has been expanded.

The room model using the ANSI/ASHRAE BESTEST 140-2007. All test models are available in: http://simulationresearch.lbl.gov/modelica/releases/v1.2_build1/help/Buildings_Rooms_Examples_BESTEST.htm. The results are in the range of other building simulation programs.

Figure 6 shows the implementation of case 600. On the top right are the definitions of the material constructions. On the top left are the internal heat gains. Below are the feedback control loops for the heating and cooling. At the bottom left is the model for the air infiltration and at the bottom right is the model for the heat transfer with the ground and the model that reads weather data.

Figure 6: Modelica Model of Bestest Case 600



CHAPTER 4: Documentation

This chapter describes the user guide and the tutorials that have been developed within this project to assist users in getting started and in using best practices when developing models.

4.1 User Guide

This section describes the user guide that has been developed to assist users in the effective use of the Buildings library. The user guide covers how to

- get started,
- follow best practices when createing models and running simulations,
- fix common problems in translating or simulating models,
- automate pre- and post-processing of model inputs and outputs,
- develop and contribute new models to the library, and
- get help.

The following information can be found in the Modelica files rather than in the user guide:

- The physical assumptions and the equations that the models are based on.
- Advice as to which models may be used if multiple models may be available for similar equipment or physical phenomena.

An online version of the user guide is available at http://simulationresearch.lbl.gov/modelica/userGuide/, and the Modelica files are available at http://simulationresearch.lbl.gov/modelica/releases/latest/help/Buildings.html.

4.1.1 Getting Started

4.1.1.1 Literature for Users

Recommending new users of Modelica, to study the books by Michael Tiller (Tiller, 2001) and Peter Fritzson [(Fritzson, 2011) and (Fritzson, 2004)], and the tutorials that are listed at https://www.modelica.org/publications.

Although the Modelica Language Tutorial is for an older version (Modelica 1.4), it is still instructive and relevant to understanding the concepts of the language.

Links to papers that describe or used the *Buildings* library are available at: http://simulationresearch.lbl.gov/modelica/Publications. The model documentation from the download page contains user guides that describe the individual packages of the *Buildings* library.

4.1.1.2 Literature for Developers

It is essential that users who develop new thermo-fluid models understand the concept of stream connectors. Stream connectors are explained in the Modelica language definition, available at https://www.modelica.org/documents, and in the paper (Franke, et al., 2009a). The *Buildings* library uses similar modeling principles, and the same base classes, as the `Modelica.Fluid` library. Also recommend reading the paper about the standardization of thermo-fluid models in `Modelica.Fluid` (Franke, et al., 2009b).

4.1.1.3 Software Requirements

Check the software requirements for the different versions of the Buildings library at http://simulationresearch.lbl.gov/modelica/Download

4.1.1.4 Running the First Simulations

To start using Modelica, run the example models of the *Buildings* library. Make variations in these examples by changing values of model parameters or by replacing existing component models by new ones. The example models can be found in the package, `Examples`, and detailed tutorials with step-by-step instructions for how to build system models can be found in the Tutorial package.

Note that heat transfer models, which can be found in `Buildings.HeatTransfer.*.Examples` are easier to understand than fluid flow models because

- Fluid flow handling adds more complexity due to flow reversal (i.e., if the mass flow rate changes its direction).
- Fluid flow models may need to handle multiple species such as air and water vapor, as well as trace substances such as CO_2.
- Fluid flow models use packages that define medium models, such as dry air, moist air, water or other fluids.

4.1.2 Best Practice

This section explains to library users best practices in creating new system models. The selected topics are based on problems that are often observed with new users of Modelica. Experienced users of Modelica may skip this section.

4.1.2.1 Organization of Packages

When developing models, one should distinguish between a library which contains widely applicable models, such as the *Buildings* library, and an application-specific model which may be created for a specific building and is of limited use for other applications. Recommend storing application-specific models outside of the *Buildings* library. This will allow replacing the *Buildings* library with a new version without having to change the application-specific model. If during the course of the development of application-specific models, some models turn out to be of interest for other applications, then they can be contributed to the development of the *Buildings* library.

4.1.2.2 Building Large System Models

When creating a large system model, it is typically easier to build the system model through the composition of subsystem models that can be tested in isolation. For example, the package `Buildings.Examples.ChillerPlant.BaseClasses.Controls.Examples` contains small test models that are used to test individual components in the large system model `Buildings.Examples.ChillerPlant`. Creating small test models typically saves time as the proper response of controls, and the proper operation of subsystems, can be tested in isolation of complex system-interactions that are often present in large models.

4.1.2.3 Propagating Parameters and Media Packages

Consider a model with a pump, `pum` and a mass flow sensor, `sen`. Suppose that both models have a parameter, `m_flow_nominal` for the nominal mass flow rate that needs to be set to the same value. Rather than setting these parameters individually to a numeric value, it is recommended to propagate the parameter to the top-level of the model. Thus, instead of using the declaration,

```
Pump pum(m_flow_nominal=0.1) "Pump";
TemperatureSensor sen(m_flow_nominal=0.1) "Sensor";
```

We recommend using

```
Modelica.SIunits.MassFlowRate m_flow_nominal = 0.1
                                "Nominal mass flow rate";
Pump pum(m_flow_nominal=m_flow_nominal) "Pump";
TemperatureSensor sen(m_flow_nominal=m_flow_nominal) "Sensor";
```

This allows changing the value of, `m_flow_nominal` at one location, and then having the value be propagated to all models that reference it. The effort for the additional declaration typically pays off as changes to the model are easier and more robust.

Propagating parameters and packages is particularly important for medium definitions. This allows the user to change the medium declaration at one location and then have it propagated to all models that reference it. This can be done by using the declaration

```
replaceable package Medium = Modelica.Media.Interfaces.PartialMedium
"Medium model for air" annotation (choicesAllMatching=true);
```

Here, the optional annotation, `annotation (choicesAllMatching=true)` is added which causes a graphical user interface to show a drop-down menu with all medium models that extend from `Modelica.Media.Interfaces.PartialMedium`.

If the above sensor requires a medium model, which is likely the case, its declaration would be

```
TemperatureSensor sen(redeclare package Medium = Medium,
                      m_flow_nominal=m_flow_nominal) "Sensor";
```

At the top-level of a system-model, one would set the `Medium` package to an actual media, such as by using

```
package Medium = Buildings.Media.PerfectGases.MoistAir   "Medium
model";
TemperatureSensor sen(redeclare package Medium = Medium,
m_flow_nominal=m_flow_nominal) "Sensor";
```

4.1.2.4 Thermo-Fluid Systems

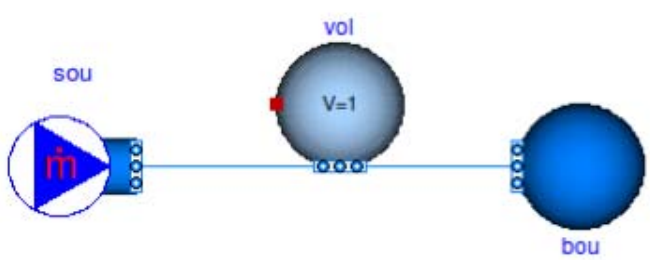
In this section, we describe best practices that are specific to the modeling of thermo-fluid systems.

Overdetermined Initialization Problem and Inconsistent Equations

This next section explains how state variables, such as temperature and pressure, can be initialized.

Consider a model consisting of a mass flow source `Modelica.Fluid.Sources.MassFlowSource_T`, a fluid volume `Buildings.Fluid.MixingVolumes.MixingVolume` and a fixed boundary condition `Buildings.Fluid.Sources.FixedBoundary`, connected in series as shown in the figure below. Note that the instance `bou` implements an equation that sets the medium pressure at its port, i.e., the port pressure `bou.ports.p` is fixed.

Figure 7: Schematic Diagram of a Flow Source, a Fluid Volume, and a Pressure Source



The volume allows configuring balance equations for energy and mass in four different ways. Let $p(\cdot)$ be the pressure of the volume, p_0 be the parameter for the initial pressure, $m(\cdot)$ be the mass contained in the volume, $m_i(\cdot)$ be the mass flow rate across the i-th fluid port of the volume and t_0 be the initial time. Then, the equations for the mass balance of the fluid volume can be configured as shown in the table below.

Parameter `massDynamics`	**Initialization problem** if $\rho = \rho(p)$	**Initialization problem** if $\rho \neq \rho(p)$	**Equation used during time stepping**
`DynamicsFreeInitial`	Unspecified	Unspecified	$dm(t)/dt = \sum m_i(t)$
`FixedInitial`	$p(t_0) = p_0$	Unspecified	$dm(t)/dt = \sum m_i(t)$
`SteadyStateInitial`	$dp(t_0)/dt = 0$	Unspecified	$dm(t)/dt = \sum m_i(t)$
`SteadyState`	Unspecified	Unspecified	$0 = \sum m_i(t)$

Unspecified means that no equation is declared for the initial value $p(t_0)$. In this situation, there can be two cases:

- If a system model sets the pressure in the above model `vol.p=vol.ports.p=bou.ports.p` due to the connection between them, then $p(t_0)$ of the volume is equal to `bou.ports.p`.
- If a system model does not set the pressure (i.e., if `vol` and `bou` are not connected to each other), then the pressure starts at the value `p(start=Medium.p_default)`, where `Medium` is the name of the instance of the medium model.

Since the model `Buildings.Fluid.Sources.FixedBoundary` fixes the pressure at its port, the initial conditions $p(t_0)= p_0$ and $dp(t_0)/dt=0$ lead to an overspecified system for the model shown above. To avoid such situation, use different initial conditions, or add a flow resistance between the mixing volume and the pressure source. The flow resistance introduces an equation that relates the pressure of the mixing volume and the pressure source as a function of the mass flow rate, thereby removing the inconsistency.

> Warning: The setting `FixedInitial` should be used with caution: Since the pressure dynamics is fast, this setting can lead to very fast transients when the simulation starts. Such transients can cause numerical problems for differential equation solvers.

Similarly, for the energy balance, let $U(\cdot)$ be the energy stored in the volume, $T(\cdot)$ be the temperature of the volume, $m_i(\cdot)$ be the mass flow rate that carries the specific enthalpy per unit mass $h_i(\cdot)$ across the i-th fluid connector of the volume, and let $Q(\cdot)$ be the heat flow at the heat port of the volume. Then, the energy balance can be configured as shown in the table below.

Parameter `energyDynamics`	**Initialization problem**	**Equation used during time stepping**
`DynamicsFreeInitial`	Unspecified	$dU(t)/dt = \sum m_i(t)h_i(t) + Q(t)$
`FixedInitial`	$T(t_0)= T_0$	$dU(t)/dt = \sum m_i(t)h_i(t) + Q(t)$
`SteadyStateInitial`	$dT(t_0)/dt = 0$	$dU(t)/dt = \sum m_i(t)h_i(t) + Q(t)$
`SteadyState`	Unspecified	$0 = \sum m_i(t)h_i(t) + Q(t)$

Unspecified means that no equation is declared for $T(t_0)$. In this situation, there can be two cases:

- If a system model sets the temperature (i.e. if in the model the heat port of `vol` is connected to a fixed temperature), then $T(t_0)$ of the volume would be equal to the temperature connected to this port.
- If a system model does not set the temperature, then the temperature starts at the value `T(start=Medium.T_default)`, where `Medium` is the medium model.

Notes:

- Selecting `SteadyState` for the energy balance and not `SteadyState` for the mass balance can lead to inconsistent equations. The model will check for this situation and stop the translation with an error message. To see why the equations are inconsistent, consider a volume with two fluid ports and no heat port. Then, it is possible that $m_1(t) \neq 0$ and $m_2(t) = 0$, since $dm(t)/dt = m_1(t) + m_2(t)$. However, the energy balance equation is $0 = m_1(t)h_1(t) + m_2(t)h_2(t) + Q(t)$, with $Q(t) = 0$ because there is no heat port. Therefore, we obtain $0 = m_1(t)h_1(t)$, which is inconsistent.

- Unlike the case with the pressure initialization, the temperature in the model `bou` does not lead to `vol.T = bou.T` at initial time, because physics allows the temperatures in `bou` and `vol` to be different.

The equations for the mass fraction dynamics (such as the water vapor concentration), and the trace substance dynamics (such as carbon dioxide concentration) are similar to the energy equations.

Let $X(\cdot)$ be the mass of the species in the volume, $m(t_0)$ be the initial mass of the volume, x_0 be the user-selected species concentration in the volume, $x_i(\cdot)$ be the species concentration at the i-th fluid port, and $(\cdot)$ be the species added from the outside, for example the water vapor added by a humidifier. Then, the substance dynamics can be configured as shown in the table below.

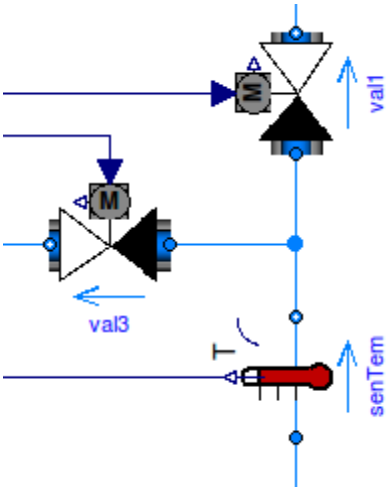
Parameter `energyDynamics`	**Initialization problem**	**Equation used during time stepping**
`DynamicsFreeInitial`	Unspecified	$dX(t)/dt = \sum_i(t)x_i(t) + (t)$
`FixedInitial`	$X(t_0)=m(t_0)x_0$	$dX(t)/dt = \sum_i(t)x_i(t) + (t)$
`SteadyStateInitial`	$dX(t_0)/dt=0$	$dX(t)/dt = \sum_i(t)x_i(t) + (t)$
`SteadyState`	Unspecified	$0 = \sum_i(t)x_i(t) + (t)$

The equations for the trace substance dynamics are identical to the equations for the substance dynamics, if $X(\cdot)$, $(\cdot)$and $x_i(\cdot)$ are replaced with $C(\cdot)$, $\dot{C}(\cdot)$ and $c_i(\cdot)$, where $C(\cdot)$ is the mass of the trace substances in the volume, $c_i(\cdot)$ is the trace substance concentration at the i-th fluid port and $\dot{C}(\cdot)$ is the trace substance mass flow rate added from the outside. Therefore, energy, mass fraction and trace substances have identical equations and configurations.

Modeling of Fluid Junctions

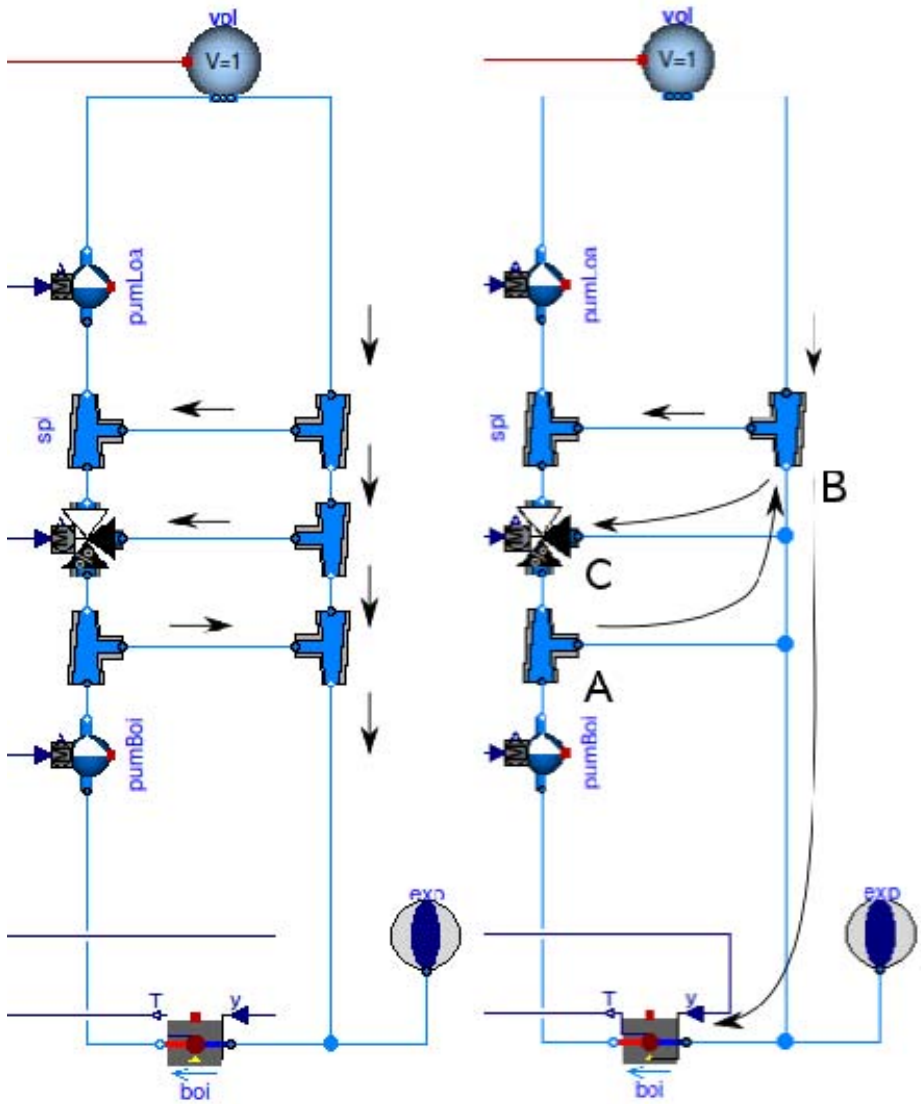
In Modelica, connecting fluid ports as shown below leads to ideal mixing at the junction. In some situations, such as the configuration below, connecting multiple connectors to a fluid port represents the physical phenomenon that was intended to model.

Figure 8: Connection of Three Components without Explicitly Introducing a Mixer or Splitter Model



However, in more complex flow configurations, one may want to explicitly control what branches of a piping or duct network mix. This may be achieved by using an instance of the model `SplitterFixedResistanceDpM` as shown in the left figure below, which is the test model `BoilerPolynomialClosedLoop`.

Figure 9: Correct (Left) and Wrong (Right) Connection of Components with Use of a Mixer or Splitter Model



In the figure on the left, the mixing points have been correctly defined by use of the three-way model that mixes or splits flow. By setting the nominal pressure drop of the mixer or splitter model to zero, the mixer or splitter model can be simplified so that no equation for the flow resistance is introduced. In addition, in the branch of the splitter that connects to the valve, a pressure drop can be modelled, which then affects the valve authority. However, in the figure on the right, the flow that leaves port A is mixing at port B with the return from the volume `vol`, and then it flows to port C. Thus, the valve is exposed to the wrong temperature.

Use of Sensors in Fluid Flow Systems

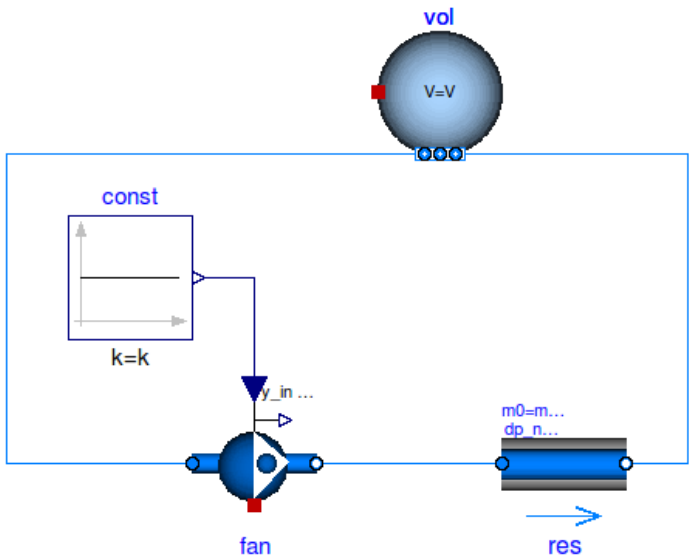
When selecting a sensor model, a distinction needs to be made whether the measured quantity depends on the direction of the flow or not. If the quantity depends on the flow direction, such

as temperature or relative humidity, then sensors with two ports from the `Buildings.Fluid.Sensors` library should be used. These sensors have a more efficient implementation than sensors with one port for situations where the flow reverses its direction. The proper use sensor is described in the User's Guide of the `Buildings.Fluid.Sensors` package.

Thermal Expansion of Water

This section explains how to account for the thermal expansion of water. Consider the flow circuit shown below that consists of a pump or fan, a flow resistance and a volume.

Figure 10: Schematic Diagram of a Flow Circuit without Means to Account for the Thermal Expansion



When this model is used with a medium model that models compressible flow, such as the medium model `Buildings.Media.IdealGases.SimpleAir`, then the model is well defined because the gas medium implements the equation $p = \rho RT$, where p is the static pressure, ρ is the mass density, R is the gas constant and T is the absolute temperature.

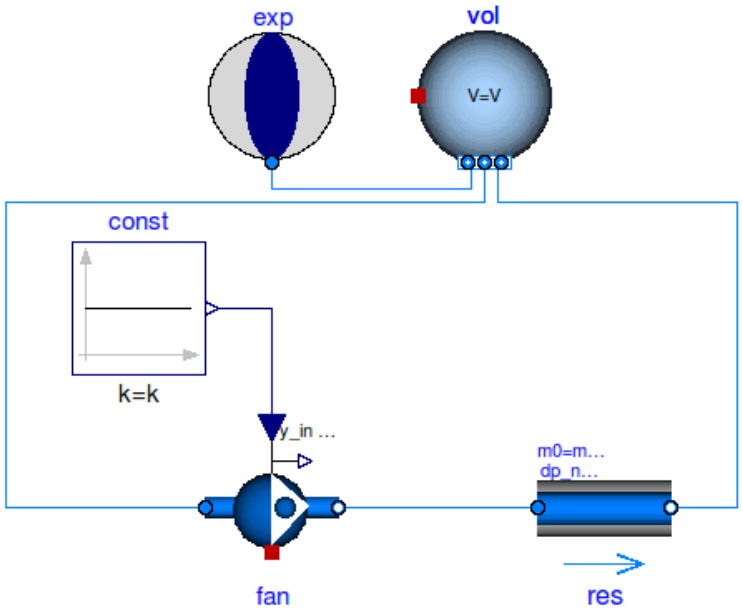
However, when the medium model is changed to a model that models incompressible flow, such as `Buildings.Media.GasesConstantDensity.SimpleAir` or `Buildings.Media.ConstantPropertyLiquidWater`, then the density is constant. Consequently, there is no equation that can be used to compute the pressure based on the volume. In this situation, trying to translate the model leads, in Dymola, to this error message:

```
The DAE has 151 scalar unknowns and 151 scalar equations.
Error: The model FlowCircuit is structurally singular.
The problem is structurally singular for the element type Real.
The number of scalar Real unknown elements are 58.
The number of scalar Real equation elements are 58.
```

Similarly, if the medium model `Modelica.Media.Water.WaterIF97OnePhase_ph` is used, which models density as a function of pressure and enthalpy, then the model is well-defined, but the pressure increases the longer the pump runs. The reason is that the pump adds heat to the water. As the water temperature increases from 20°C to 40°C, the pressure increases from 1 bar to 150 bars.

To avoid this singularity or increase in pressure, a model that imposes a pressure source and that can account for the expansion of the fluid needs to be used. For example, you may use `Buildings.Fluid.Storage.ExpansionVessel` to form the system model shown below.

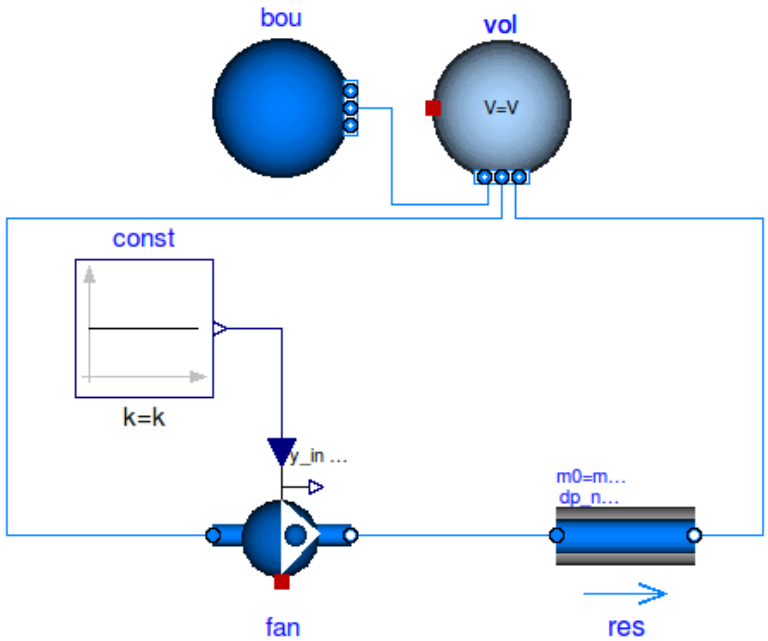
Figure 11: Schematic Diagram of a Flow Circuit with Expansion Vessel that Adds a Pressure Source and Accounts for the Thermal Expansion of the Medium



Alternatively, you may use `Buildings.Fluid.Sources.FixedBoundary`, which sets the pressure to a constant value and adds or removes fluid as needed to maintain the pressure. The model `Buildings.Fluid.Sources.FixedBoundary` usually leads to simpler equations than `Buildings.Fluid.Storage.ExpansionVessel`. Note that the medium that flows out of the fluid port of `Buildings.Fluid.Sources.FixedBoundary` is at a fixed temperature, while the model `Buildings.Fluid.Storage.ExpansionVessel` conserves energy.

However, since the thermal expansion of the fluid is usually small, this effect can be neglected in most building HVAC applications.

Figure 12: Schematic Diagram of a Flow Circuit with a Boundary Model that Adds a Fixed Pressure Source and Accounts for Any Thermal Expansion of the Medium



Nominal Values

Most components have parameters for the nominal operating conditions. These parameters have names that end in `_nominal` and they should be set to the values that the component typically have if they are run at full load or design conditions. Depending on the model, these parameters are used differently, and the respective model documentation or code should be consulted for details. However, the table below shows typical use of parameters in various models to help the user understand how they are used.

Parameter	Model	Functionality
`m_flow_nominal` `dp_nominal`	Flow resistance models.	These parameters may be used to define a point on the flow rate versus pressure drop curve. For other mass flow rates, the pressure drop is typically adjusted using similarity laws. See `FixedResistanceDpM`.
`m_flow_nominal` `m_flow_small`	Sensors. Volumes. Heat exchangers.	Some of these models set `m_flow_small=1E-4*abs(m_flow_nominal)` as the default value. Then, m_flow_small is used to regularize, or replace, equations when the mass flow rate is smaller than `m_flow_small` in magnitude. This is needed to improve the numerical properties of the model. The error in the results is negligible for typical applications, because at flow rates below 0.01% from the design flow rate, most model assumptions are not applicable anyways, and the HVAC system is not operated in this region. Modelica simulates in the continuous-time domain, thus such small flow rates can occur, and therefore models are implemented in such a way that they are numerically well-behaved for zero or near-zero flow rates.
`tau` `m_flow_nominal`	Sensors. Volumes. Heat exchangers. Chillers.	Because Modelica simulates in the continuous-time domain, dynamic models are in general numerically more efficient than steady-state models. However, dynamic models require product data that are generally not published by manufacturers. Examples include the volume of fluid that is contained in a device, and the weight of heat exchangers. In addition, other effects such as transport delays in pipes and heat exchangers of a chiller are generally unknown and require detailed geometry that is typically not available during the design stage. To circumvent this problem, many models take, as a parameter, the time constant, `tau` and lump all its thermal mass into a fluid volume. The time constant `tau` can be understood as the time constant that one would observe if the input to the component has a step change, and the mass flow rate of the component is equal to `m_flow_nominal`. Using these two values and the fluid density `rho`, components adjust their fluid volume `V=m_flow_nominal tau/rho` because having such a volume gives the specified time response. For most components, engineering experience can be used to estimate a reasonable value for `tau`, and where generally applicable values can be used,, components already set a default value for `tau`. See for example `WetCoilDiscretized`.

4.1.2.5 Start Values of Iteration Variables

When computing numerical solutions to systems of nonlinear equations, a Newton-based solver is typically used. Such solvers have a higher success of convergence if good start values are provided for the iteration variables. In Dymola, to see what start values are used, one can enter on the simulation tab the command

```
Advanced.LogStartValuesForIterationVariables = true;
```

Then, when a model is translated, for example using

```
translateModel("Buildings.Fluid.Boilers.Examples.BoilerPolynomialClo
sedLoop");
```

an output of the form

```
Start values for iteration variables:
  val.res1.dp(start = 3000.0)
  val.res3.dp(start = 3000.0)
```

is produced. This shows the iteration variables and their start values. These start values can be overwritten in the model.

4.1.2.6 Avoiding Events

In Modelica, the time integration is halted whenever a real elementary operation such as $x > y$, where x and y are variables of type `Real`, changes its value. In this situation, an event occurs and the solver determines a small interval in time in which the relation changes its value. Determining this time interval often requires an iterative solution, which can significantly increase the computing time if the iteration requires the evaluation of a large system of equations. An example where such an event occurs is the relation

```
if port_a.m_flow > 0 then
  T_in = port_a.T;
else
  T_in = port_b.T;
end if;
```

or, equivalently,

```
T_in = if port_a.m_flow > 0 then port_a.T else port_b.T;
```

When simulating a model that contains such code, a time integrator will iterate to find the time instant where `port_a.m_flow` crosses zero. If the modeling assumptions allow approximating this equation in a neighborhood around `port_a.m_flow=0`, then replacing this equation with an approximation that does not require an event iteration can reduce computing time. For example, the above equation could be approximated as

```
T = Modelica.Fluid.Utilities.regStep(
  port_a.m_flow, T_a_inflow, T_b_inflow,
  m_flow_nominal*1E-4);
```

where `m_flow_nominal` is a parameter that is set to a value that is close to the mass flow rate that the model has at full load. If the magnitude of the flow rate is larger than 1E-4 times the

typical flow rate, the approximate equation is the same as the exact equation, and below that value, an approximation is used. However, for such small flow rates, not much energy is transported and hence the error introduced by the approximation is generally negligible.

In some cases, adding dynamics to the model can further improve the computing time, because the return value of the function `Modelica.Fluid.Utilities.regStep()` above can change abruptly if its argument `port_a.m_flow` oscillates in the range of `±1E-4*m_flow_nominal`, for example due to numerical noise. Adding dynamics may be achieved using a formulation such as

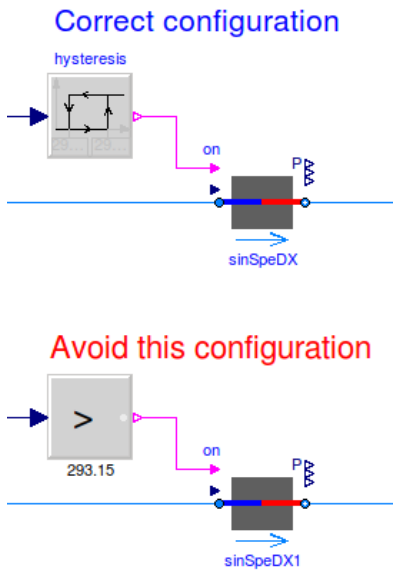
```
TMed = Modelica.Fluid.Utilities.regStep(
  port_a.m_flow, T_a_inflow, T_b_inflow,
  m_flow_nominal*1E-4);
  der(T)=(TMed-T)/tau;
```

where `tau` > 0 is a time constant. See for example `Buildings.Fluid.Sensors.TemperatureTwoPort` for a robust implementation.

> Note: In the package `Buildings.Utilities.Math` the functions and blocks whose names start with smooth can be used to avoid events.

4.1.2.7 Controls

Figure 13: Schematic Diagram of a Controller That Switches a Coil On And Off.



In the top configuration of Figure 13, the hysteresis avoids numerical problems (and short-cycling) if the control input remains close to the set point. The bottom configuration can cause the integration to stall if the input signal to the threshold block is the solution of an iterative solver and remains around 293.15 Kelvin.

When implementing an on/off controller, always use a controller with hysteresis such as shown in the top configuration of Figure 13. If no hysteresis is used, then numerical problems can occur if the variable that is input to the controller depends on a variable that is computed by an iterative algorithm. Examples of iterative algorithms are nonlinear equation solvers or time integration algorithms with variable step size (such as the radau and dassl solver in Dymola). The problem is caused as follows: Let $T(t) \in R$ be the input into a controller, such as a room air temperature. If $T(t)$ is the state variable computed by solving a differential equation, or if $T(t)$ depends on a variable that needs to be solved for iteratively, then $T(t)$ can only be approximated by some approximation $T^*(\square,t)$, where $\square$ is the solver tolerance. Even if the system is at equilibrium, the solver can cause the value of $T^*(\square,t)$ to slightly change from one iteration to another. Hence, $T^*(\square,t)$ can exhibit what is called numerical noise. Now, if $T^*(\square,t)$ is used to switch a heater on and off whenever it crosses at set point temperature, and if $T(t)$ happens to be at an equilibrium near the set point temperature, then the heater can switch on and off rapidly due to the numerical noise. This can cause the time integration to stall.

To illustrate this problem, try to simulate

```
model Unstable
  Real x(start=0.1);
equation
  der(x) = if x > 0 then -1 else 1;
end Unstable;
```

In Dymola 2013, as expected the model stalls at t = 0.1 because the `if-then-else` construct triggers an event iteration whenever x crosses zero.

4.1.2.8 Numerical Solvers

Dymola 2012 FD01, is configured to use dassl as a default solver with a tolerance of 1E-4. Recommend changing this setting to radau with a tolerance of around 1E-6, as this generally leads to faster and more robust simulation for thermo-fluid flow systems.

Note that this is the error tolerance of the local integration time step. Most ordinary differential equation solvers only control the local integration error and not the global integration error. As a rule of thumb, the global integration error is one order of magnitude larger than the local integration error. However, the actual magnitude of the global integration error depends on the stability of the differential equation. As an extreme case, if a system is chaotic and uncontrolled, then the global integration error will grow rapidly.

4.1.3 Work-Arounds

This section describes work-arounds that often help if there are problems during the model translation, initialization, or simulation.

4.1.3.1 Avoiding Step Changes

All flow machines (fans and pumps) have a boolean parameter, `filteredSpeed`, and all actuators have a boolean parameter, `filteredOpening`. If set to `true`, which is the default setting, then the control input signal is sent to a 2nd order low pass filter that changes a step signal to a smooth signal. This typically improves the robustness of the simulation.

To see the effect of the filter, consider the model below in which `fanS` is configured with `filteredSpeed=false`, and `fanC` is configured with `filteredSpeed=true`. Both fans are connected to a step input signal. The configuration of `fanS` causes the fan speed to instantly change from 0 to 1. In large system models, this can lead to high computing time or to convergence problems. The `fanC` avoids this problem because the speed of the fan varies continuously, thereby making it easier for the solver to compute a solution. In this model, we set the parameter `raiseTime=30` seconds.

Figure 14: Schematic Diagram of Fans That are Configured with Filterspeed=False (Fans) and Filterspeed=True (Fanc)

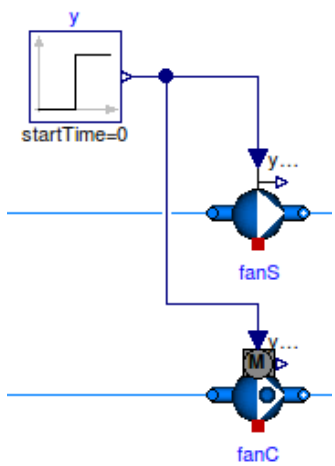
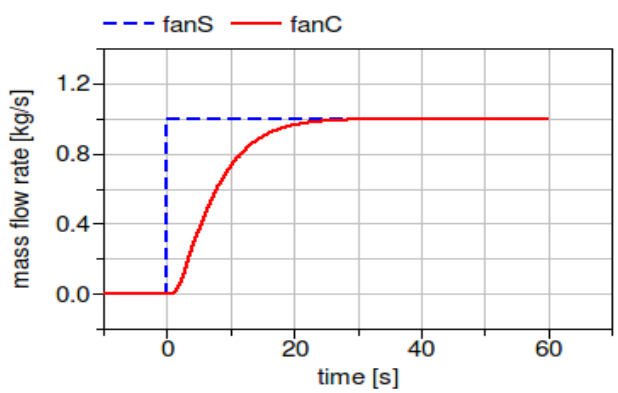


Figure 15: Mass Flow Rate of the Two Fans For a Step Input Signal at 0 Seconds



For fans and pumps, the dynamics introduced by the filter can be thought of as approximating the rotational inertia of the fan rotor and the inertia of the fluid in the duct or piping network. The default value is `raiseTime=30` seconds.

For actuators, the rise time approximates the travel time of the valve lift. The default value is `raiseTime=120` seconds.

Note: When changing `filteredSpeed` (or `filteredOpening`), or when changing the value of `riseTime`, the dynamic response of the closed loop control changes. Therefore, control gains may need to be retuned to ensure satisfactory closed loop control performance.

For further information, see the User's Guide of the flow machine package, and the User's Guide of the actuator package.

4.1.3.2 Breaking Algebraic Loops

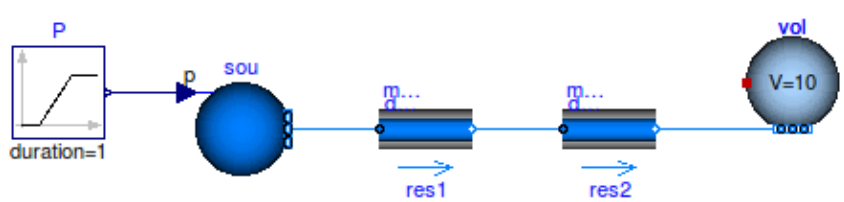
In fluid flow systems, flow junctions where mass flow rates separate and mix can couple non-linear systems of equations. This leads to larger systems of coupled equations that need to be solved, which often causes larger computing time and can sometimes cause convergence problems. To decouple these systems of equations, in the model of a flow splitter or mixer (model `Buildings.Fluid.FixedResistances.SplitterFixedResistanceDpM`), or in models for fans or pumps (such as the model `Buildings.Fluid.Movers.FlowMachine_y`), the parameter `dynamicBalance` can be set to `true`. This adds a control volume at the fluid junction that can decouple the system of equations.

4.1.3.3 Reducing Nonlinear Equations of Serially Connected Flow Resistances

In fluid flow systems, if multiple components are connected in series, then computing the pressure drop due to flow friction in the individual components can lead to coupled nonlinear systems of equations. While this is no problem for small models, the iterative solution can lead to higher computing time, particularly in large models where other equations may be part of the residual function.

For illustration, consider the simple system shown below in which the flow resistances `res1` and `res2` compute the mass flow rate as $m = k\,(\Delta p)^{0.5}$ if the parameter `from_dp` is set to `true`, or otherwise the pressure drop between their inlet and outlet as $\Delta p = (m/k)^2$. (Both formulations are implemented using regularization near zero.)

Figure 16: Schematic Diagram of Two Flow Resistances in Series That Connect a Source and a Volume

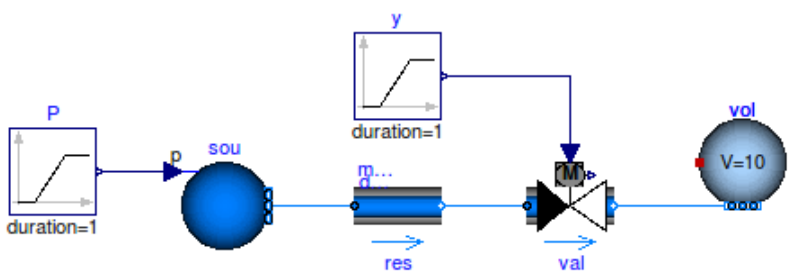


Depending on the configuration of the individual component models, simulating this system model may require the iterative solution of a nonlinear equation to compute the mass flow rate or the pressure drop. To avoid a nonlinear equation, you could do any of the below measures.

- Set the parameter `res2(dp_nominal=0)`, and add the pressure drop to the parameter `dp_nominal` of the model `res1`. This will eliminate the equation that computes the flow friction in `res2`, thereby avoiding a nonlinear equation. The same applies if there are multiple components in series, such as a pre-heat coil, a heating coil, and a cooling coil.
- Set `from_dp=false` in all components, which is the default setting. This will cause Modelica to use a function that computes the pressure drop as a function of the mass flow rate. Therefore, a code translator is likely to generate an equation that solves for the mass flow rate, and it then uses the mass flow rate to compute the pressure drop of the components that are connected in series.

Control valves also allow lumping the pressure drop into the model of the valve. Consider the situation where a fixed flow resistance is in series with a control valve as shown below.

Figure 17: Schematic Diagram of a Fixed Flow Resistance and a Valve in Series That Connect a Source and a Volume



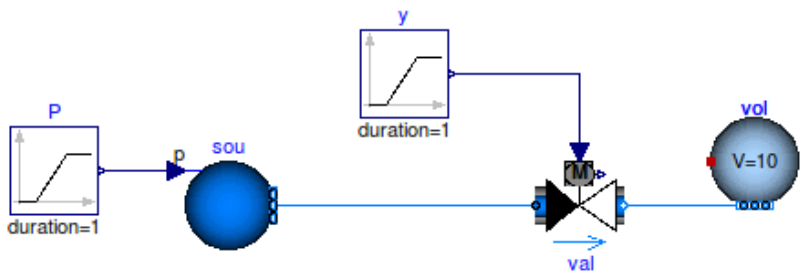
Suppose the parameters are

```
Buildings.Fluid.FixedResistances.FixedResistanceDpM res(
  redeclare package Medium = Medium,
  m_flow_nominal=0.2,
  dp_nominal=10000);

Buildings.Fluid.Actuators.Valves.TwoWayLinear val(
  redeclare package Medium = Medium,
  m_flow_nominal=0.2,
  dpValve_nominal=5000);
```

To avoid a nonlinear equation, the flow resistance could be deleted as shown below

Figure 18: Schematic Diagram of a Valve That Connects a Source and a Volume



If the valve is configured as

```
Buildings.Fluid.Actuators.Valves.TwoWayLinear val(
  redeclare package Medium = Medium,
  m_flow_nominal=0.2,
  dpValve_nominal=5000,
  dpFixed_nominal=10000);
```

then the valve will compute the composite flow coefficient $k^{\bar{}}$ as

$$k^{\bar{}} = (1/k_v(y) + 1/k_f)^{-1/2}$$

where $k_v(y) = m\square(y)\,(\Delta p)^{-0.5}$ is the flow coefficient of the valve at the lift y, and k_f is equal to the ratio `m_flow_nominal/sqrt(dpFixed_nominal)`. The valve model then computes the pressure drop using $k^{\bar{}}$ and the same equations as described above for the fixed resistances. Thus, the composite model has the same valve authority and mass flow rate, but a nonlinear equation can be avoided.

For more details, see the User's Guide of the actuator package.

4.1.3.4 Prescribed Mass Flow Rate

For some system models, the mass flow rate can be prescribed by using an idealized pump or fan (model `Buildings.Fluid.Movers.FlowMachine_m_flow`) or a source element that outputs the required mass flow rate (such as the model `Buildings.Fluid.Sources.MassFlowSource_T`). Using these models avoids having to compute the intersection of the fan curve and the flow resistance. In some situations, this can lead to faster and more robust simulation.

4.1.3.5 Avoiding Overspecified Initialization Problems

If in thermofluid flow systems, Dymola fails to translate a model with the error message:

```
Error: The initialization problem is overspecified for variables
of element type Real
The initial equation
...
refers to variables, which are all knowns.
To correct it you can remove this equation.
```

then the initialization problem is overspecified. To avoid this, set

```
energyDynamics =
Modelica.Fluid.Types.Dynamics.DynamicsFreeInitial;
massDynamics = Modelica.Fluid.Types.Dynamics.DynamicsFreeInitial;
```

in the instances of the components that contain fluid volumes.

4.1.4 Pre- and Post-Processing

The Modelica language specification does not specify a scripting language for simulating models, and it does not standardize the file format of the result file. To provide scripts that automate simulation of models and post-processing of results, LBNL created the BuildingsPy Python library. This library can be used

- To run Modelica simulations using Dymola.
- To process `*.mat` output files that were generated by Dymola or OpenModelica.
- To run unit tests as part of the library development.

See the separate documentation at http://simulationresearch.lbl.gov/modelica/buildingspy/ for how to use this library.

In addition, Dymola provides MATLAB scripts that can be used to process `*.mat` output files that were generated by Dymola. See the Dymola documentation for how to use these scripts.

To optimize cost functions that are computed by Dymola, the GenOpt optimization program can be used. See GenOpt's directory `example/dymola` for examples that use Dymola to compute the cost function in an optimal control problem.

4.1.5 Development

This section describes the development of the *Buildings* library.

4.1.5.1 Contributing

We welcome contributions of new models and suggestions for how to improve the library. Models that are contributed need to follow the following guidelines, as this is needed to integrate them in the library, make them accessible to users and further maintain them:

- They should be of general interest to other users and well documented and tested.
- They need to follow the coding conventions described in
    - The Buildings library user guide.
    - The Modelica Standard Library user guide.

    - The Buildings library development checklist.

- They need to be made available under the Modelica license.
- For models of fluid flow components, they need to be based on the base classes in `Buildings.Fluid.Interfaces`, which are described in the user guide of this package. Otherwise, it becomes difficult to ensure that the implementation is numerically robust.

The website for the development of the library is https://corbu.lbl.gov/trac/bie

4.1.5.2 Adding a New Class

Adding a new class, such as a model or a function, is usually easiest by extending, or copying and modifying, an existing class. In many cases, the similar component already exists. In this situation, it is recommended to copy and modify a similar component. If both components share a significant amount of similar code, then a base class should be introduced that implements the common code. See for example `Buildings.Fluid.Sensors.BaseClasses.PartialAbsoluteSensor` which is shared by all sensors with one fluid port in the package `Buildings.Fluid.Sensors`.

The next sections give guidance that is specific to the implementation of thermofluid flow devices and to pressure drop models.

Thermofluid Flow Device

To add a component of a thermofluid flow device, the package `Buildings.Fluid.Interface` contains basic classes that can be extended. See `Buildings.Fluid.Interface.UsersGuide` for a description of these classes. Alternatively, simple models such as the models below may be used as a starting point for implementing new models for thermofluid flow devices:

- `Buildings.Fluid.HeatExchangers.HeaterCoolerPrescribed`

  For a device that adds heat to a fluid stream.

- `Buildings.Fluid.MassExchangers.HumidifierPrescribed`

  For a device that adds humidity to a fluid stream.

- `Buildings.Fluid.Chillers.Carnot`

  For a device that exchanges heat between two fluid streams.

- `Buildings.Fluid.MassExchangers.ConstantEffectiveness`

  For a device that exchanges heat and humidity between two fluid streams.

Pressure Drop

When implementing equations for pressure drop, it is recommended to expand the base class `Buildings.Fluid.BaseClasses.PartialResistance`. Models should allow computing the flow resistance as a quadratic function with regularization near zero as

implemented
in `Buildings.Fluid.BaseClasses.FlowModels.basicFlowFunction_dp` and
in `Buildings.Fluid.BaseClasses.FlowModels.basicFlowFunction_m_flow`. The governing equation is

$$k = m\square\ (\Delta p)^{-0.5}$$

with regularization near zero to avoid that the limit $dm\square\ /d\Delta p$ tends to infinity as $m\square\ \rightarrow 0$, as this can cause Newton-based solvers to stall. For fixed flow resistances, k is typically computed based on nominal conditions such as $k = m\square_0(\Delta p_0)^{-0.5}$, where $m\square$ is equal to the parameter `m_flow_nominal` and Δp is equal to the parameter `dp_nominal`.

All pressure drop models should also provide a parameter that allows replacing the equation by a linear model of the form

$$m\square\ m\square_0 = (k^{-})^2\Delta p$$

> Note: Equations for pressure drop are implemented as a function of mass flow rate and not volume flow rate. For some models this allows decoupling the mass flow balance from the energy balance. Otherwise, computing the mass flow distribution would require knowledge of the density, which may depend on temperature, and temperature is only known after solving the energy balance.

When implementing the pressure drop model, also provide means to

- use homotopy, this should be used by default.
- disable the pressure drop model.

Disabling the pressure drop model allows, for example, a user to set in a series connection of a heating coil and a cooling coil the pressure drop of the heating coil to zero, and to lump the pressure drop of the heating coil into the pressure drop model of the cooling coil. This often reduces the size of the system of nonlinear equations.

4.2 Tutorials

To address the feedback of the Technical Advisory Group, two tutorials that explain step-by-step how to build system model have been implemented. One tutorial is for a hydronic heating system, and the other is for an air-based HVAC system.

4.2.1 Hydronic Heating System Tutorial

Figure 19: Schematic Diagram of Hydronic Heating System Used in the Tutorial

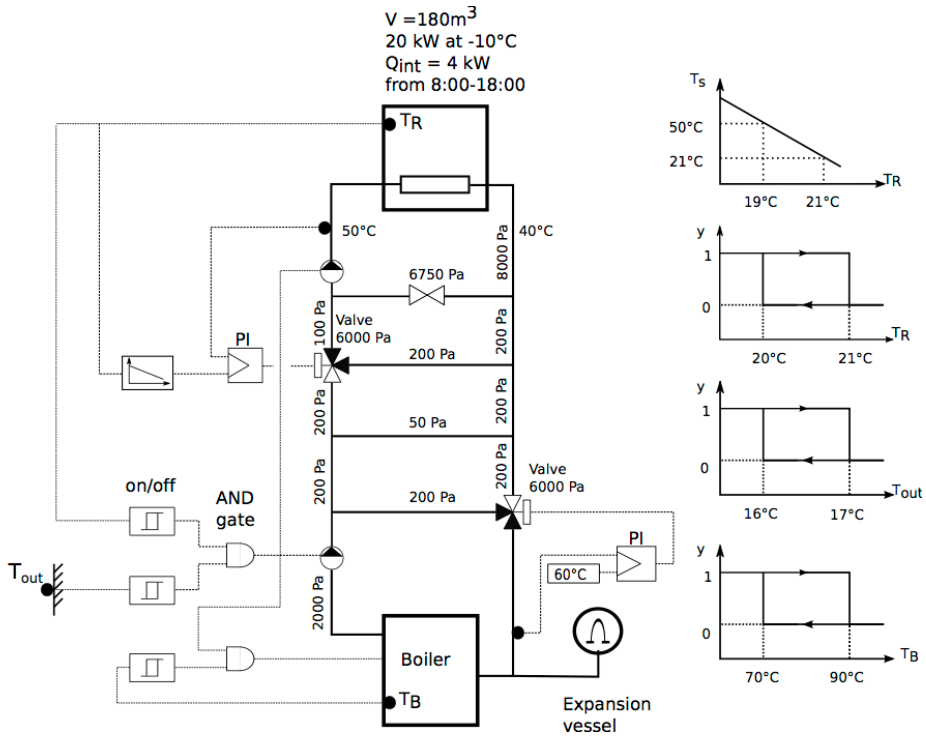


Figure 19 shows the schematics and the control for the hydronic heating system.

The model consists of

- a room with a heating load, approximated as steady-state heat transfer with the environment,
- a heating loop with a constant bypass and a three-way valve, which is modulated to track the room temperature set point, and
- a boiler loop with constant mass flow rate, boiler on/off control and control valve to ensure a minimum return water temperature.

To explain the implementation of this model, the model has been created in the following stages:

1. Buildings.Examples.Tutorial.Boiler.System1 implements the room model without any heating.
2. Buildings.Examples.Tutorial.Boiler.System2 adds a radiator that is fed with water at a constant temperature and flow rate. The pump is switched on and off depending on the room temperature.
3. Buildings.Examples.Tutorial.Boiler.System3 adds the boiler circuit with open loop control for the boiler and the mixing valves.

4. Buildings.Examples.Tutorial.Boiler.System4 adds closed loop control for the boiler and the pumps.
5. Buildings.Examples.Tutorial.Boiler.System5 adds closed loop control for the two valves.
6. Buildings.Examples.Tutorial.Boiler.System6 replaces the constant outdoor temperature with weather data from a file, and changes the valve control from P-control to PI-control.
7. Buildings.Examples.Tutorial.Boiler.System7 replaces the boiler and pump control using a state machine.

Figure 20 shows the implementation of the model, *System7* in Modelica and Figure 21 shows the trajectories for the temperatures for a simulation period of two days. The top figure shows the boiler temperature around 80°C. The room temperature first drops due to the thermal lag of the system and then stays at 20°C. Shown in green is the outside air temperature. The bottom figure shows in green the return water temperature that enters the furnace. This is controlled to be at 60°C or higher as indicated by the magenta line. The red line is the setpoint temperature for the supply water to the radiators, and the blue line is the actual supply water temperature. The setpoint is tracked well except for the first few hours during which the capacity of the furnace is too low to heat up the room and the water in the hydronic loop. Around *t=15,000* seconds (4 hours) the room air setpoint is reached, at which time the capacity is sufficient and therefore the supply water temperature setpoint is tracked and the minimum return water temperature of 60°C is exceeded.

Figure 20: Modelica Implementation of the Hydronic Heating Model Used in the Tutorial

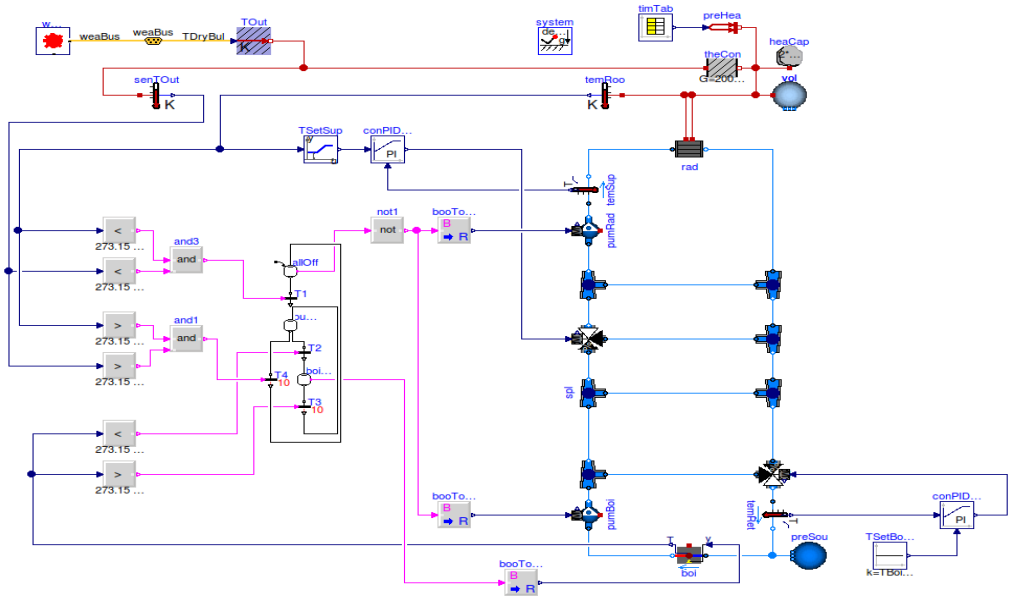
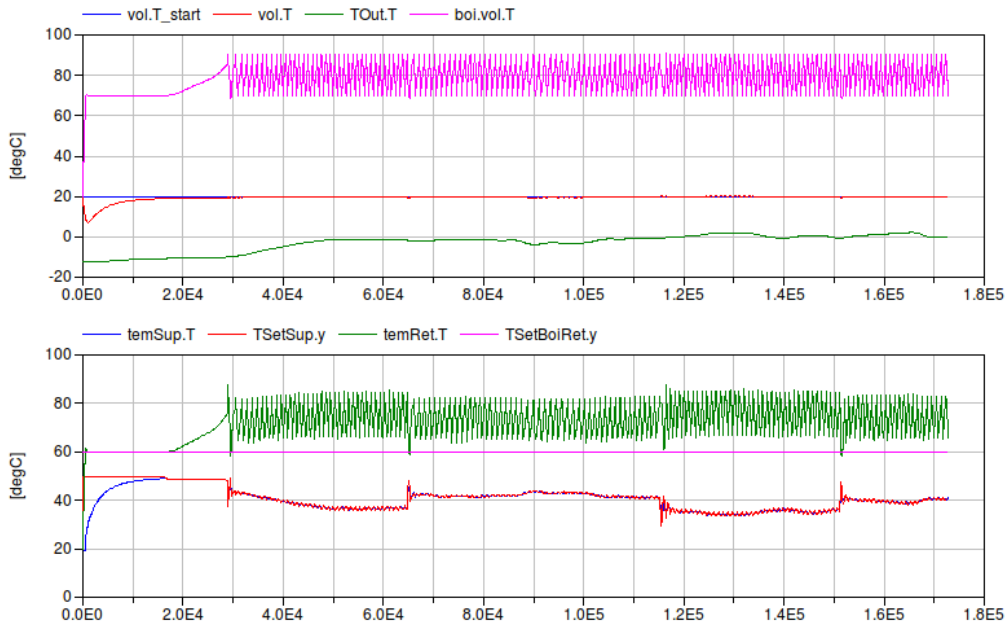


Figure 21: Temperature Trajectories for the Hydronic Heating System Model



4.2.2 Air-based HVAC System Tutorial

Figure 22: Schematic Diagram of HVAC System that is Used in the Second Tutorial

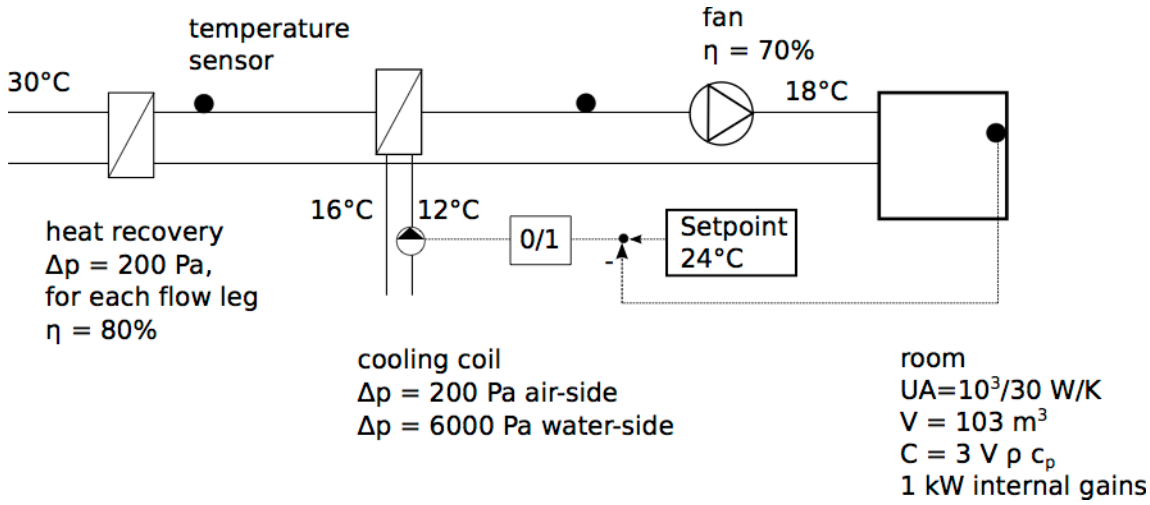


Figure 22 shows the schematic diagram of the air-based cooling system that is used in the second tutorial. The model consists of

- a room with a cooling load due to internal heat gains and due to conductive heat gains from the environment,

- a fresh air supply with a heat recovery, a cooling coil and a fan. The fan is operating continuously at full speed. The room air temperature is controlled by a controller that switches the water flow rate through the coil on and off with a dead-band of *1* Kelvin.

To explain the implementation of this model, the model has been created in the following three stages:

1. Buildings.Examples.Tutorial.SpaceCooling.System1 implements the room model without air supply.
2. Buildings.Examples.Tutorial.SpaceCooling.System2 implements the air supply with open-loop control.
3. Buildings.Examples.Tutorial.SpaceCooling.System3 adds closed-loop control.

Figure 23: Modelica Implementation of the Hydronic Heating Model used in the Tutorial

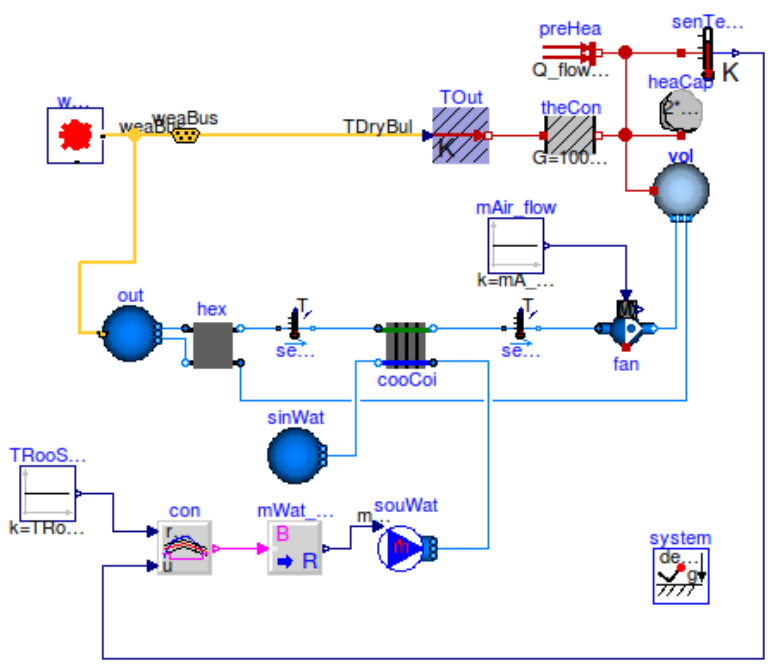


Figure 24: Temperature Trajectories for the Hydronic Heating System Model

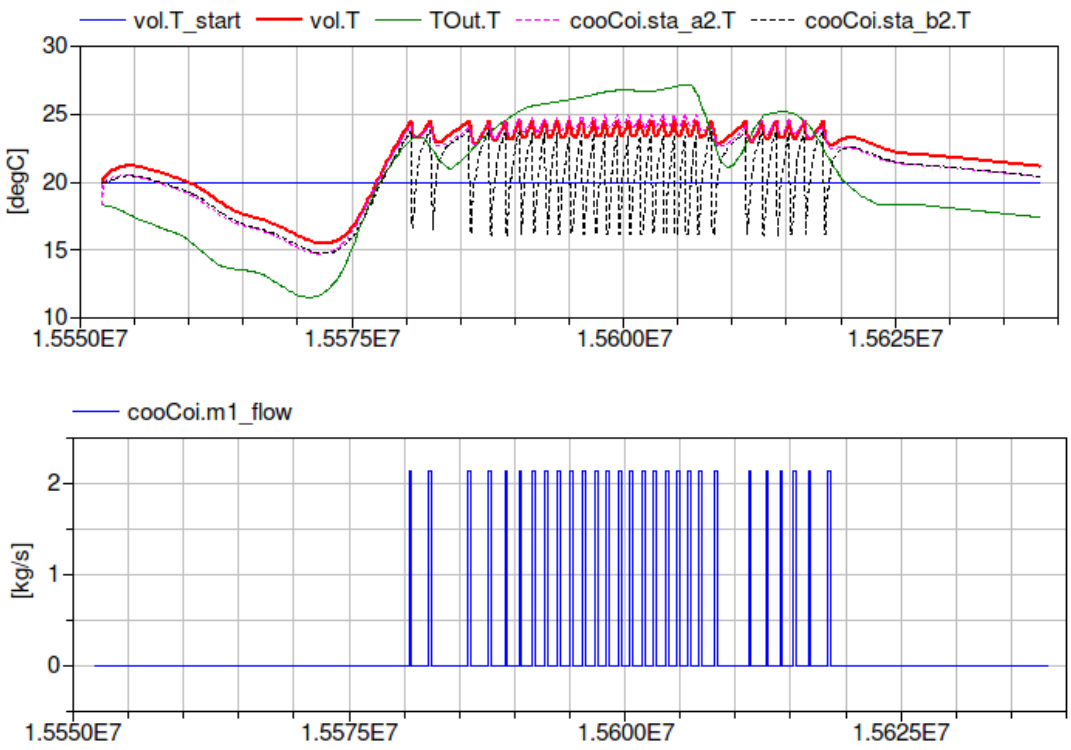


Figure 23 shows the Modelica implementation of *System3* and Figure 24 shows the trajectories for the temperature and the cooling water mass flow rate. In the top of the figure, the blue line at 20°C is the initial room temperature and the setpoint temperature. Green is the outside air temperature and red is the actual room air temperature. The dashed magenta line is the leaving air temperature of the coil and the dashed black line is the leaving water temperature of the coil. The bottom figure shows the water mass flow rate of the coil. For this implementation, the water valve is either open or closed because we have used an on/off controller. Alternatively, a modulating control could have been used with a P, PI or PID controller.

CHAPTER 5:
Case studies

This chapter presents two case studies that illustrate the use of a model to improve the control of building HVAC systems. The first case study was done at the Kirsch Center in Cupertino, CA. The objective was to maintain thermal comfort while reducing energy use for the space conditioning. In this study, models developed within this project have been validated and calibrated, and subsequently used to test different operational scenarios. The objective of the second case study is to evaluate the energy savings for a data center that can be achieved through the use of a water-side economizer. As the system-level performance is sensitive to temperature setpoints, parameteric studies are presented for the system with and without water side economizer.

5.1 Kirsch Center

For this study, we have developed a radiant slab model. We used this model in combination with other models in the Modelica Buildings library to develop a zone model for a real classroom. The classroom is primarily naturally ventilated with a radiant floor slab system for supplementary cooling or heating. The design-intent controls of the system for this classroom are identified. The zone model was calibrated and validated using measured data. The usability of this model to improve system performance is discussed. Different control options for a class room have been analyzed. The major finding is that with properly scheduled night-time natural ventilation, the room temperatures can be maintained within the comfort zone without the use of mechanical cooling. The use of the radiant cooling slab, cooled by evaporative cooling, improves comfort during the hottest hours of the year.

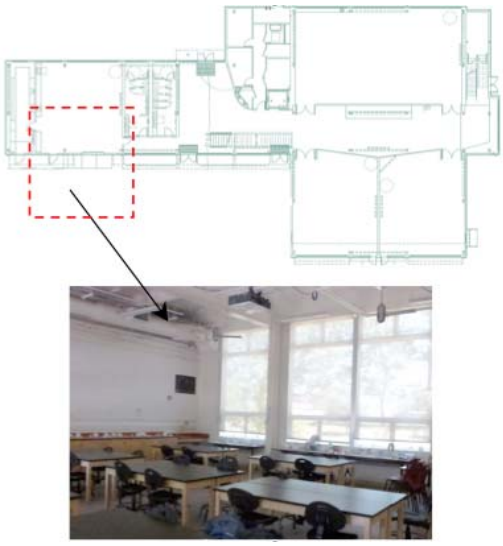
5.1.1 Facility Information

Figure 25: The Kirsch Center

Photo Credit: http://www.deanza.edu/kirschcenter/

The Kirsch Center for Environmental Studies is a 22,000ft^2 (2,044m^2) building of classrooms, offices, and laboratories on the campus of De Anza Community College in Cupertino, California, as shown in Figure 25. It is a LEED Platinum Certified building. The Kirsch Center is designed to be a teaching building in order for all students to learn about stewardship and resource conservation. The owner therefore rejected the concept of automatically opening windows because the intent is that students, staff and faculty would need to be active participants in the operation of the building in order to learn about its operation and impact on efficiency.

Figure 26: The Biodiversity Lab



The selected zone for the study is the biodiversity lab located on the first floor west wing of the building as shown in Figure 26. Comfort conditioning in this zone is primarily provided by natural means. A high performance envelope, supplemented by passive solar gain in the south-side entryway and adjacent hallway, along with interior gains, lead to little required mechanical heating. For cooling, it is first to be naturally ventilated, with manually operable windows on both sides of the room to provide cross-ventilation, opened or closed by classroom instructors as part of the "teaching by doing" features of the building. The need for fresh-air is monitored by measuring the CO_2 concentration of the classroom and provided primarily by natural ventilation. Additional heating and cooling to this zone are provided by a radiant floor slab. In the winter, the water for the slab is heated by a high efficiency condensing gas-fired boiler. The cool water for summer radiant cooling is provided by an evaporatively cooled chiller. More information about the biodiversity lab is presented in the following sections.

Heat gains from equipment were minimized as much as possible with a peak of 1.0 W/ft^2 (10.8 W/m^2). Generally, light colored wall finishes and efficient light fixtures minimize the lighting energy consumption. The peak installed lighting power density is 1.0 W/ft^2 (10.8 W/m^2).

5.1.1.1 Design-Intent Controls

A Building Automation System (BAS) is used for the HVAC system. The central station for the control system permits monitoring, trending, setpoint and control sequence modification. According to the design documents, the design-intent controls were identified, and described below.

Air System & Natural Ventilation

A display of carbon dioxide levels is installed in the biodiversity lab. A lamp integrated in the same display is intended to assist the instructors and building operators to know when to open and close the operable window for natural ventilation. The green lamp indicates that the occupants should open windows while the red lamp indicates that they should close windows. The lamp will be illuminated green when any of the following conditions are true:

- During unoccupied hours, the indoor air temperature is higher than the outside air temperature and the outside air temperature is 50°F (10°C) or greater.
- During occupied hours,
  - The indoor air temperature is above the maximum adaptive comfort temperature and the outside air temperature is below indoor air temperature.
  - The indoor air temperature is below the minimum adaptive comfort temperature and the outside air temperature is above indoor air temperature.
  - The indoor CO2 concentration exceeds 800 PPM.

The lamp is illuminated red whenever the green lamp is not illuminated. An exhaust fan is available for the instructors and building operators to provide additional ventilation air. The exhaust fan is also scheduled to provide room air purge at night.

Hydronic Systems

Figure 27 shows the schematic diagram of the hydronic system for the entire building. The chilled water is provided to the radiant floor systems and AHU cooling coils (serving the east wing of the building) from an evaporatively cooled packaged water chiller. The chilled water loop is configured as primary and secondary systems, and a variable speed secondary pump distributes the chilled water to the AHU cooling coils and radiant floor systems. The chilled water system is enabled whenever all of the following conditions apply:

- The date is between April 1 and October 31.
- Any building chilled water control valve has been at least 20% open for a continuous period of 15 minutes.
- The outside air temperature is above 70°F (18.3°C).

Figure 27: Schematic Diagram of the Hydronic System

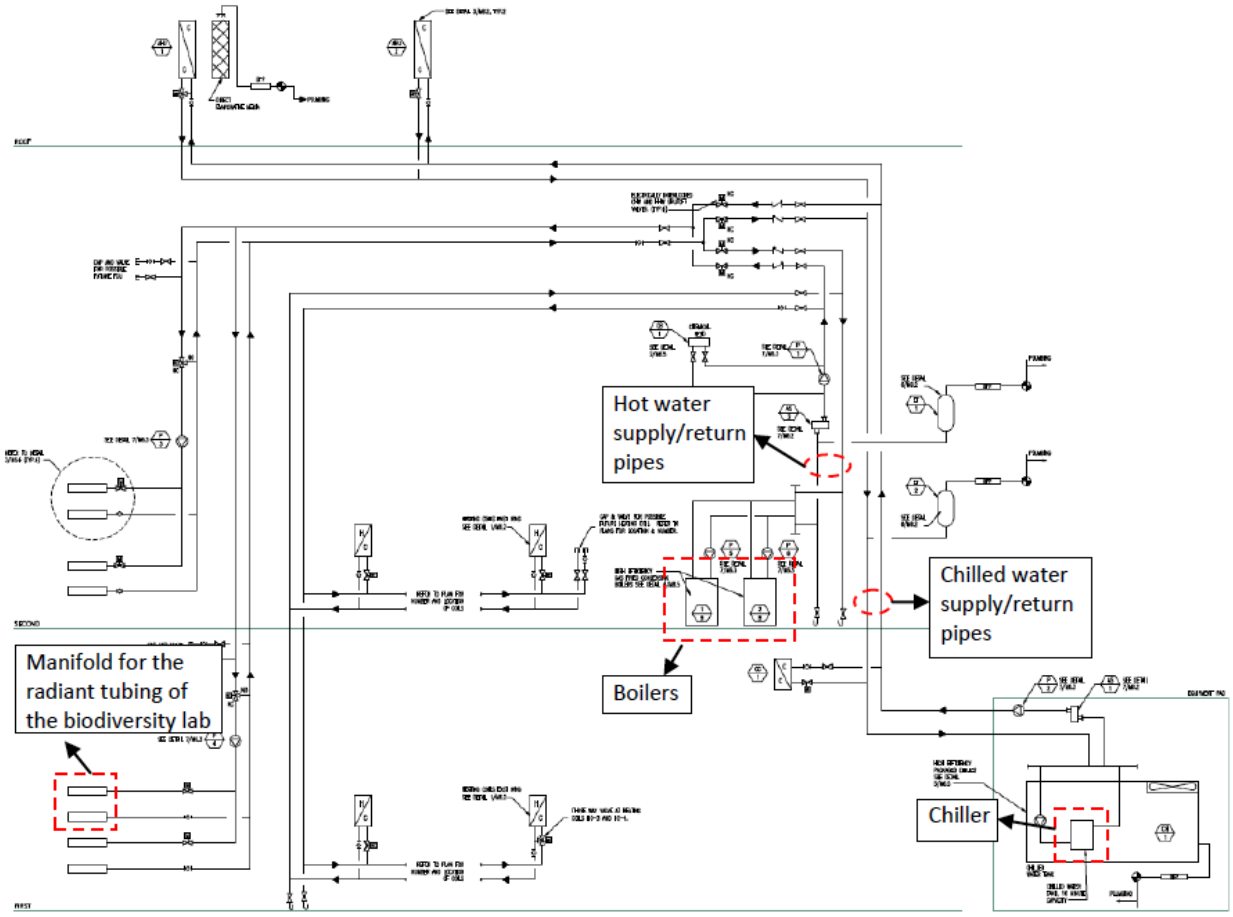


Figure 28: Hot Water Supply Temperature Reset Schedule

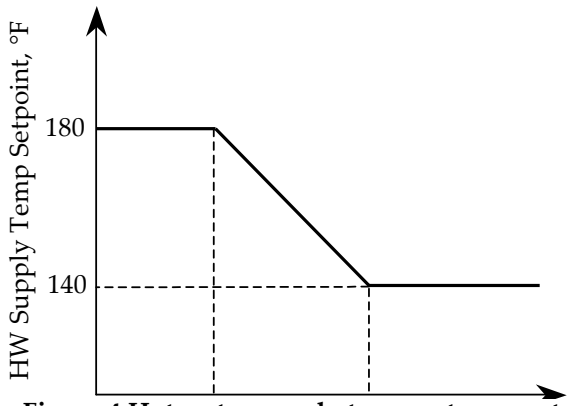


Figure 4 Hot water supply temperature reset schedule

Once enabled, the chiller will operate to maintain a primary chilled water supply temperature of 45°F (7.2°C). The secondary chilled water pump will be modulated to maintain the chilled water secondary differential pressure set point at 5 psi (34.5 kPa).

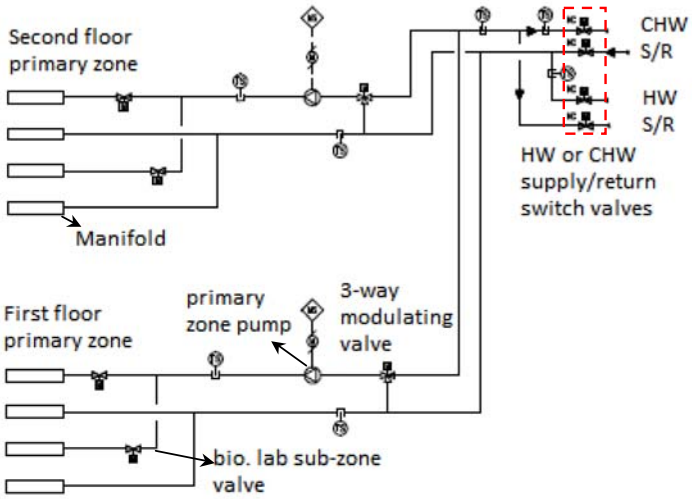
Two condensing boilers provide all building heating requirements. The hot water loop is configured as a primary and secondary system. Hot water is supplied to the heating coils and the radiant systems through a variable speed secondary pump. The hot water system is enabled whenever both of the following occur:

- The outside air temperature is below 65°F (18.3°C).
- At least one hot water control valve has been at 20% open for a continuous period of 20 minutes.

The hot water supply temperature from the boiler will be reset based on the outside air temperature as shown in Figure 28. The hot water pump variable frequency drive is modulated to maintain the hot water loop differential pressure set point at 5 psi (34.5 kPa).

Radiant Slab Control

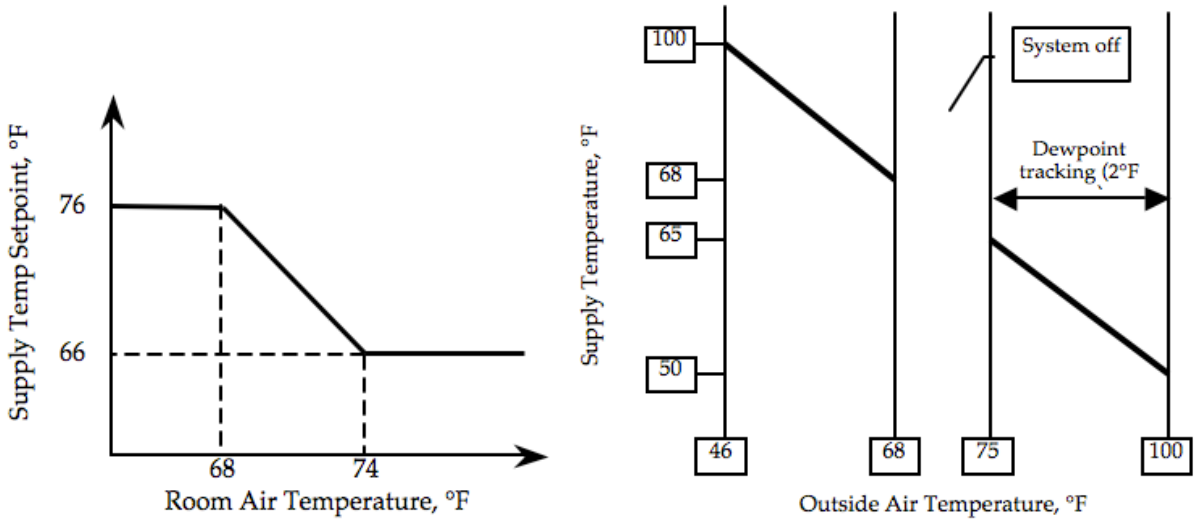
Figure 29: Radiant Slab Water System



The schematic diagram of the radiant floor slab system is shown in Figure 29. The radiant system is divided into two primary zones, the first floor zone and the second floor zone. Each floor zone is further divided into two sub-zones. The biodiversity lab is one of the sub-zones of the first floor primary zone. The indoor comfort of the biodiversity lab is first provided by natural ventilation and supplemented with the radiant floor slab for additional heating and cooling needs. The room air temperature fluctuates throughout the day in relation to the outside air temperature. During occupied hours, when the outside air temperature is between 68°F (20°C) and 74°F (23.3°C), the lamp recommending that windows should be fully opened will be on. The instructor is responsible for opening the operable window according to the lamp signal.

The radiant slab system is fed by chilled water when in cooling mode and by hot water when in heating mode, which is achieved by shutting off or opening the four hot water (HW) or chilled water (CHW) switch valves. Whenever the average zone temperature is below its heating set point 68°F (20°C), the radiant slab system is in heating mode and the HW supply and return valves will be commanded to open. Whenever the average temperature is above the cooling set point 74°F (23.3°C), the radiant slab system is in cooling mode and the CHW supply and return valves will be commanded to open. A hardwired interlock on HW or CHW switch valves is included to ensure that the two systems are not cross connected.

Figure 30: Radiant Slab Supply Water Temperature Reset Schedule



The three-way valve for each primary zone is modulated to maintain the radiant slab HW or CHW supply temperature set point. The supply water temperature set point is determined by two alternate modes: 1) controlling room air temperature, and 2) controlling radiant slab temperature. The building operator will choose the mode to calculate the supply water temperature set point for the radiant slab. The supply water temperature set point is reset based on the diagram shown in Figure 30. The controlling radiant slab temperature mode is currently in use. In either mode, the supply water temperature is never allowed to drop more than 2°F (0.9 K) from the room dew point. In that case, the three-way valve will be modulated to maintain the supply water temperature 2°F (0.9 K) above the room dew point to prevent condensation.

The sub-zone valve is commanded to open and close whenever both of the following apply:

- The outside air temperature is either below 68°F (20°C) or above 74°F (23.3°C).
- The sub-zone is in an occupied period with optimum start/stop for cooling and heating.

5.1.2 Model Development

The biodiversity lab with its radiant slab and ventilation system, and part of the cooling plant and the hydronic system was modeled. Since the case study took place during the summer only the chilled water system was modeled.

5.1.2.1 Radiant Slab Model Description

As one sub-task, a radiant slab model was developed in the Modelica building library. The model is applicable for a radiant slab with pipes or a capillary heat exchanger embedded in the construction. The model is described in Chapter 3.

5.1.2.2 Zone Model

Figure 31: Modelica System Model with Thermal Zone, Radiant Slab and Air Exchange

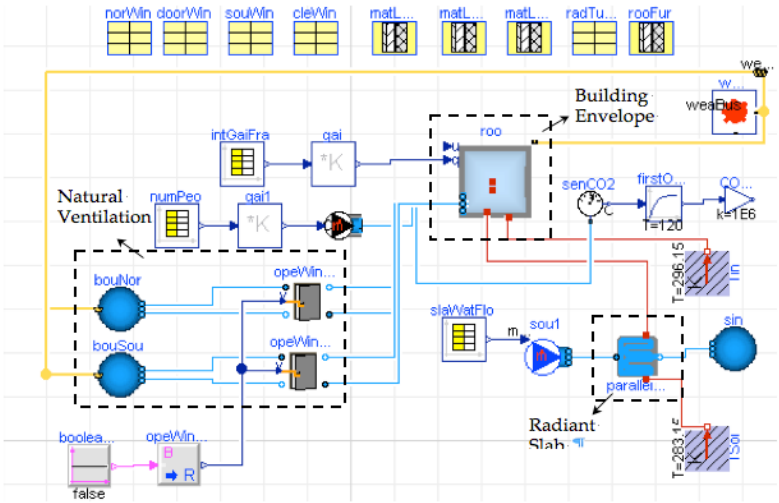
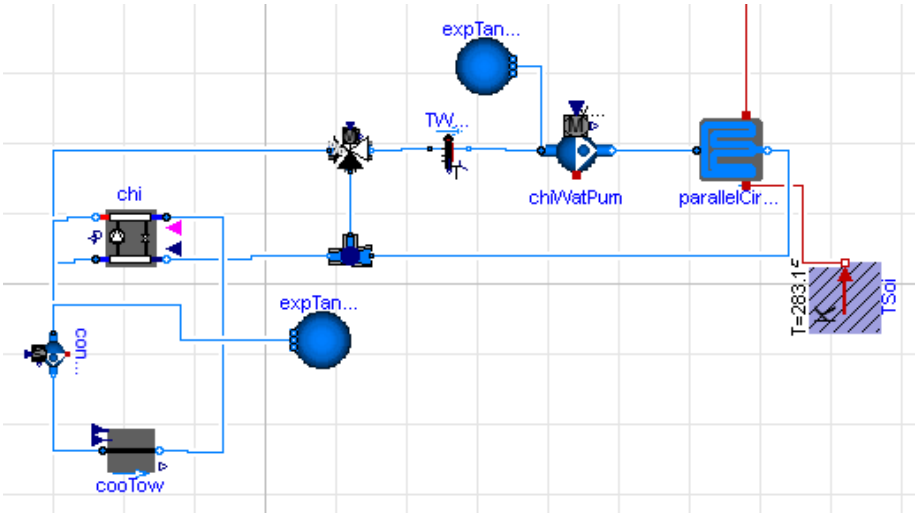


Figure 32: Hydronic System Model



The zone model includes four major parts: the building envelope, the natural ventilation, the radiant slab, and the hydronic system as shown in Figure 31 and in Figure 32. Each part is described in detail in the following sections:

Building Envelope

The building envelope was modeled using the model, *Buildings.Rooms.MixedAir*. This model allows any number of constructions and surfaces that participate in the heat exchange through

convection, conduction, infrared radiation, and solar radiation. A description of the model assumptions and the implementation and validation can be found in Wetter et al. (2011).

The dimension and the constructions were primarily obtained from the eQUEST model used for the LEED energy analysis in combination with other documents. The building envelope constructions are listed in Table 1. The biodiversity lab has two exterior walls facing north and south. The east wall, west wall, and ceiling are modeled as interior constructions with a fixed temperature boundary condition on the surface outside of the room. The south façade has overhangs along with the window with an average depth of 2 ft (0.6 m).

Table 1: Building Envelope Constructions

Exterior Wall Construction	• Exterior Wall Construction (outside to inside): 1″ stucco, 5/8″ plywood, 1″ polyisocyanurate (R-6) rigid insulation, 2 x 4 metal stud with R-13 batt insulation (R-6 assembly), 5/8″ gypsum board. Calculated equivalent U-value of 0.059 (R-16 cont. equivalent) • Interior Wall Construction: 1/2″ gypsum board, 4″ insulation board, 1/2″ gypsum board
Slab Construction	• Most: Slab on Grade, Uninsulated • Under Radiant Floor: 2″ of rigid insulation (R-10)
Windows	• North-facing windows & North-facing clerestory: Viracon VE 2-55 (COG: SC = 0.30, COND = 0.29, VT = 0.40) • South, East, and West-facing windows: Viracon VE 2-40 (COG: SC = 0.26, COND = 0.29, VT = 0.32) • Non-North Clerestory windows: Viracon VE 2-85 (COG: SC = 0.45, COND = 0.29, VT = 0.65)
Doors	• U = 0.5 Btu/h·ft^2·°F (2.8 W/m^2·K)

The occupancy schedule is 8am to 9pm, Monday to Friday.

Natural Ventilation

The natural ventilation through the operable windows is modeled using a model that simulates bi-directional air flow through a large opening. The cross ventilation effect was achieved by applying different boundary conditions for the north facing and south facing windows. The operable window can be commanded to be open or close through the model input control signal.

Radiant Slab

The radiant slab system used in the biodiversity lab has eight parallel circuits. Therefore, a multiple parallel-circuits radiant slab model was developed to reduce computing effort by using one single circuit radiant slab model instance instead of eight. The radiant slab system schedule is listed in Table 2.

Table 2: Radiant Slab System Schedule Serving the Biodiversity Lab

Tubing C.C. distance	Tubing type	Tubing size	Max. loop length	Max. loop DP	Flow
6" (0.15m)	PEX	5/8" (0.016m)	400' (122m)	10' (3m)	7.7 GPM (1.7m³/h)

Hydronic System

Since the study took place in summer and the heating system is disabled, only the chilled water system is modeled as shown in Figure 32. The equipment schedules for the chilled water system are presented in Table 3.

Table 3: Hydronic System Equipment Schedule

Item	Flow rate		Total head			Motor nominal power		
Zone pump	9.7 GPM (2.2 m³/h)		19 ft. wg. (56.8 kPa)			1/6HP (124W)		
Chiller	Nominal capacity	Min. COP	Evaporator				Condenser	
			Flow	EWT	LWT	DP	Flow	EAT
	42 ton (148 kW)	3.8	77 GPM (18 m³/h)	58°F (14°C)	44°F (7°C)	10 ft.wg. (30 kPa)	90 GPM (20 m³/h)	68°F (20°C)

The hydronic system model includes the evaporatively cooled chiller and the associated pumps and control valves. The evaporatively cooled chiller is implemented using a chiller with a cooling tower. Since only a small portion of the building is modeled, the equipment capacity is sized according to the biodiversity lab. However, the original equipment efficiencies are kept equal. For example, the chiller has a nominal capacity of 42 ton (148 kW) with a coefficient of performance (COP) of 3.8 at the nominal condition. While in the model, the chiller nominal capacity is reduced to 1 ton (3.5 kW) but still has a COP of 3.8 at the nominal condition.

5.1.3 Model Calibration & Validation

The operational data on June 20th, 21st, 25th, and 26th, 2012 have been trended at 15-minute intervals. The trended data included the room air temperature, the slab temperature (embedded 3/4" (0.02m) below the surface), the room CO_2 concentration, the outside air temperature, and the chilled water supply temperature. Due to the moderate weather on June 25th and 26th, the radiant slab and the chilled water system were off. Therefore, only the data on June 20th and 21st were used for the model validation as shown in Figure 12 (the data from 2am to 6am on June 21st was missing).

In the calibration, the main free parameters were the occupancy schedule and density, which was varied to match the measured CO_2 concentration profile. We assume a 10-minute time constant of the CO_2 concentration sensor. Figure 13 compares the measured CO_2 concentration with the simulated results. As shown in the figure, the simulated results track the measurement pretty well. With the calibrated occupancy schedule and occupant density, the lighting and plug load can be determined accordingly so that the total internal heat gain is obtained.

To validate the model, the Typical Meteorological Year3 (TMY3) weather data for San Jose, CA was used to run the simulation with the outdoor air temperature replaced by the measured data. The comparison between the measured and simulated room air temperature and slab temperature is shown in Figure 33.

Figure 33: Measured Room Temperature, Slab Temperature, Outside Air Temperature and Indoor CO2 Concentration

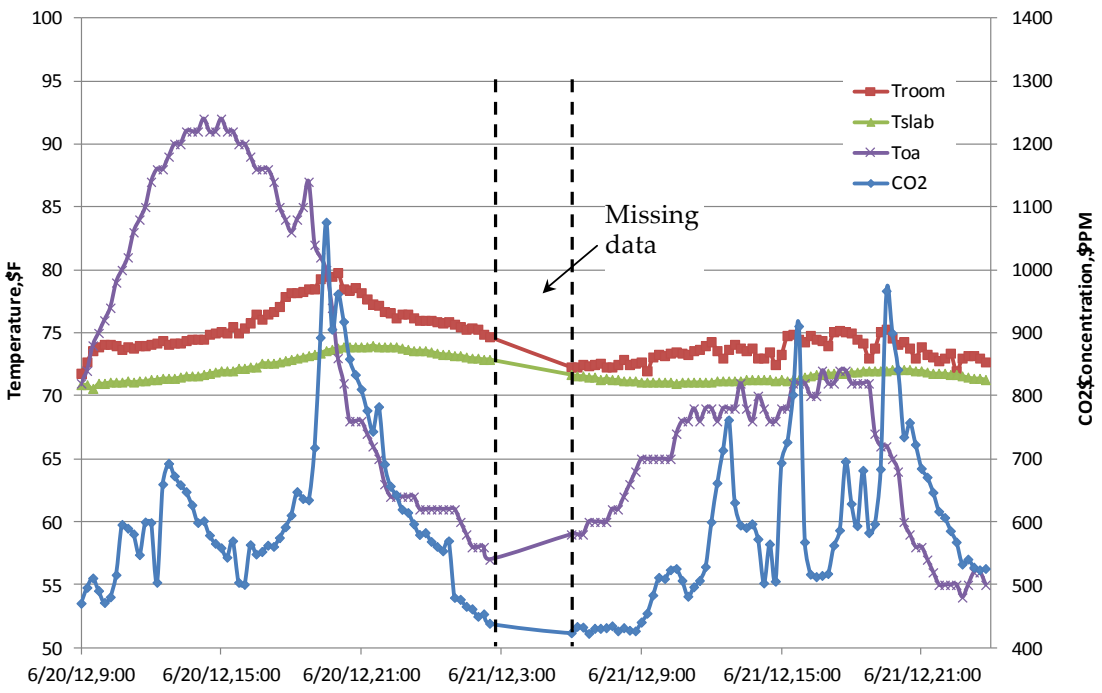


Figure 34: Comparison of Measured and Simulated CO2

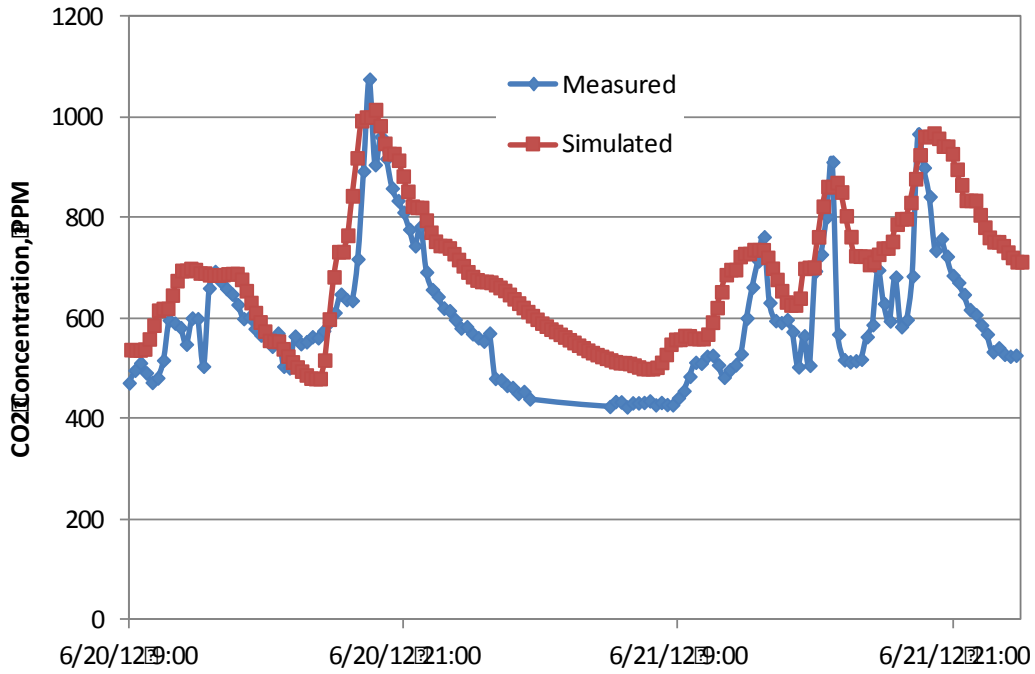
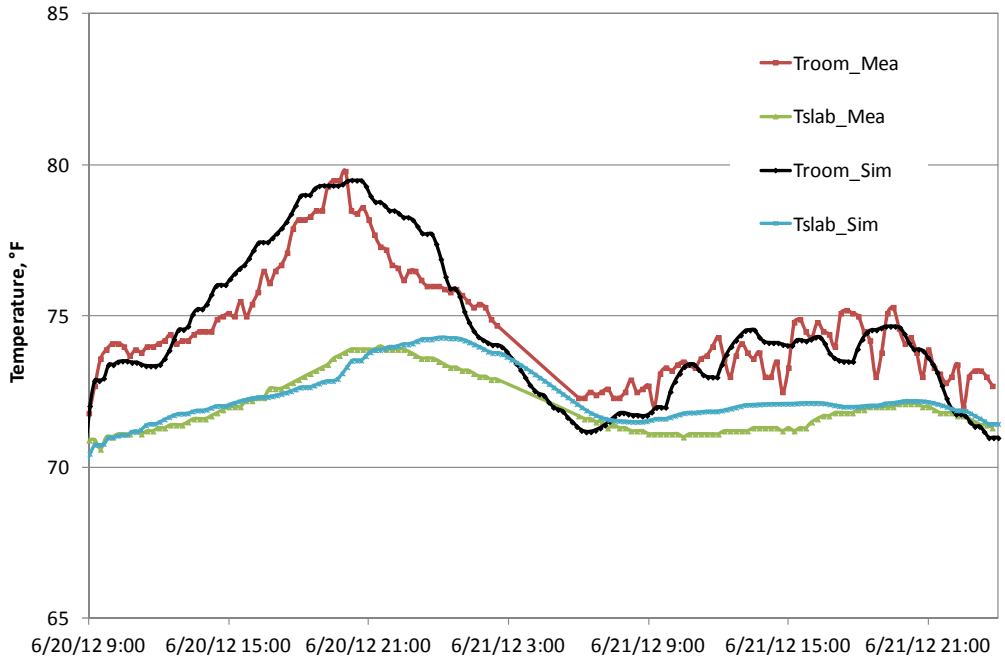


Figure 35: Comparison of Measured and simulated Room Air and Slab Temperature



As shown in Figure 35, the simulated room air and slab temperature can track the changes of the measured room air and slab temperature. The observed discrepancies may arise from the following:

- Though the outside air temperature in San Jose TMY3 weather data was replaced by the real measurement, the other key parameters, for which no measurements exist, such as the solar radiation and the wind speed and direction, are different from the local weather conditions.
- The constructions used in the model are obtained from the design documents while the actual construction could be different from the design. For example, the mass of the construction and the furniture could be the reason for the shift of the peak room air and slab temperature.
- Though the occupancy schedule and density can be derived from the CO2 concentration measurement, the lighting and plug load may vary due to the way the equipment and lighting is used. For example, the instructor may turn off the lights to use the projector, and students may or may not use their personal laptops.
- The door connecting the biodiversity lab with the rest of the building was opened during our site visit. Whether the door is open or closed could have a big impact on the room air temperature and CO2 concentration since the sensor measuring the two parameters is located right beside this door. The east wing of this building uses air handling units for air conditioning which could cause air flow from the east wing to the west wing due to the imbalance of the outside air intake.
- The radiant slab primary zone pump operation and supply water temperature were not trended. According to the building operator, the radiant slab pump is switched on two hours earlier than the class starting time and switched off one hour before the class ends. The simulation was performed using this schedule, which might not be accurate. Since the radiant slab supply water temperature is missing, we do not know if the chilled water is indeed supplied to the radiant slab water system (the CHW switch valves might be closed even though the radiant slab primary zone pump is on). Therefore, a constant radiant slab water temperature at 71.6°F (22°C) was assumed.

Taking into account these factors, the model predictions are within the expected range. Furthermore, other components of the radiant slab model, such as the heat conduction in the slab, and the convection and radiation of the slab surface, have been validated previously (Nouidui et al. 2012). Therefore, we consider the model sufficiently validated for this study.

5.1.4 Analysis

This section discusses methods of improving the building operation using the developed model.

5.1.4.1 Existing Control

A site visit showed that the actual control sequences differ from the design intent. According to the building operator and on-site observation, the controls related to the biodiversity lab that are implemented differently from the design intent are:

- The primary zone pump is controlled only by a schedule. It starts two hours earlier than the occupied starting hour and stops one hour earlier than the occupied ending hour.
- The exhaust fan is turned on from 3am to 4am, Monday to Friday, through the Building Automation System (BAS).

5.1.4.2 Recommendations

Using manually operated windows to cool the building thermal mass is challenging due to the dynamics of heat storage, the diurnal temperature swings and the sensitivity of natural ventilation to wind speed, direction and building site. To suggest how to best operate the cross ventilation, we will analyze several proposals to improve the system performance during summer using the developed model. To allow manual operation of the window, our recommended window operation will only open and close windows at 7 pm and at 8 am, and the amount of the window opening is dependent on the forecast of the next day's maximum outdoor temperature.

Figure 36: The Outside Air Temperature Profile during Summer and September in San Jose TMY3

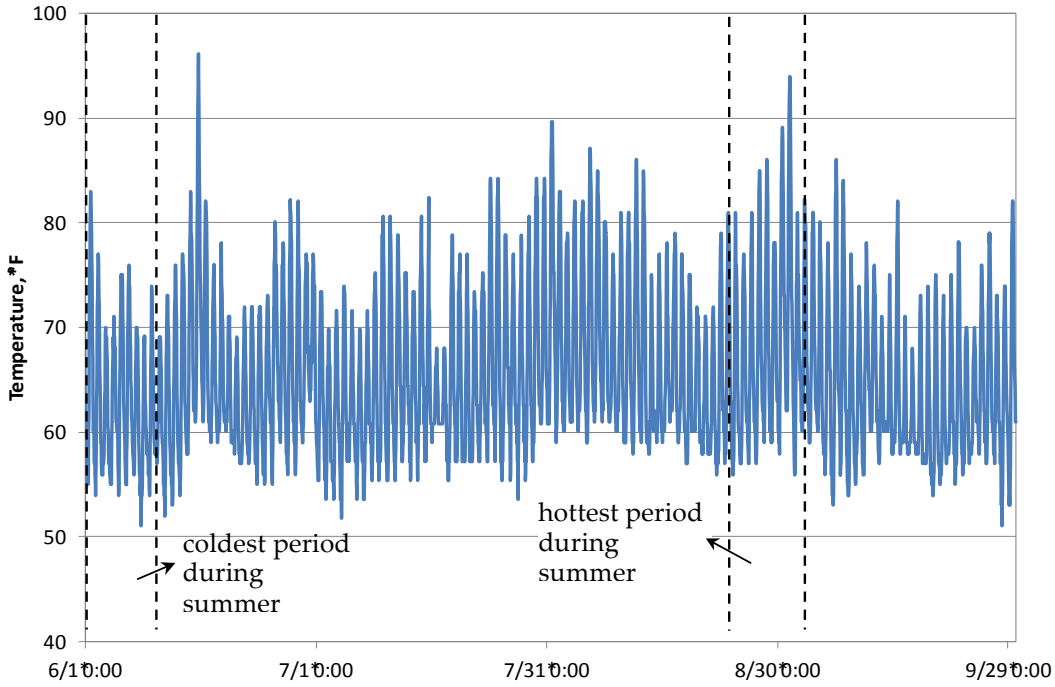


Figure 36 shows the outside air temperature profile in summer including September in the San Jose TMY3 weather data. To assess the performance of the proposed sequences of operation the coldest period (June 1st to June 10th) and the hottest period (Aug 20th to Sep 1st) was choosen. Proposal #1, the most aggressive proposal:

Open both sides of the windows at 7pm and close them at 8am every day and keep the radiant slab system off during the summer including September.

Figure 37: Predicted CO_2 and Room Air Temperature during the Coldest Period in Summer for Proposal #1

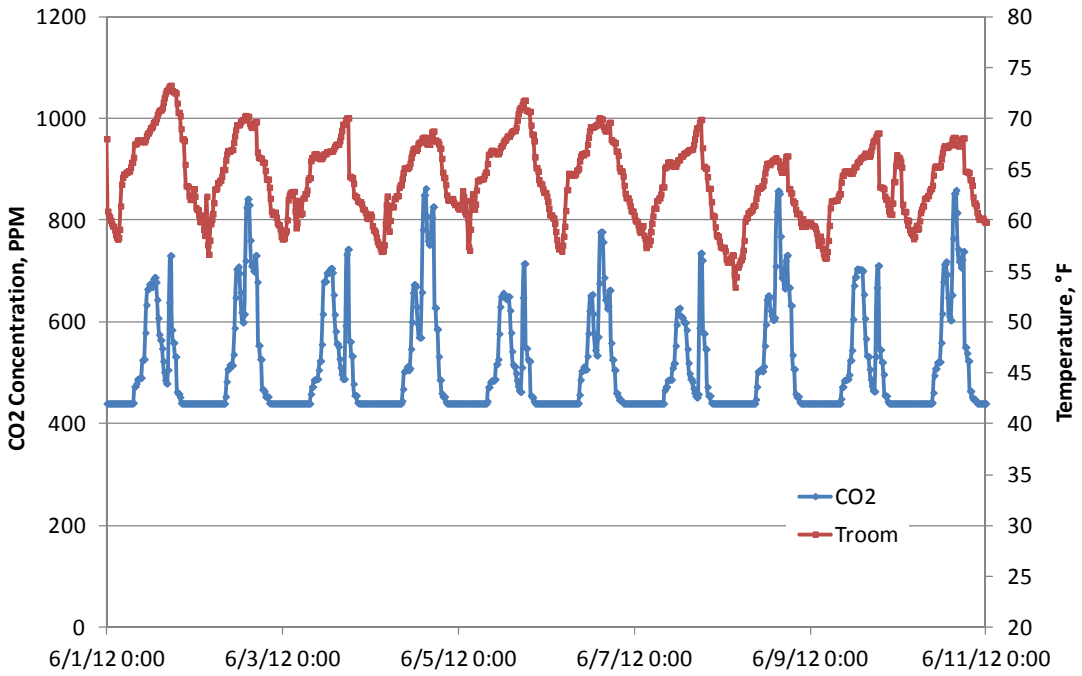
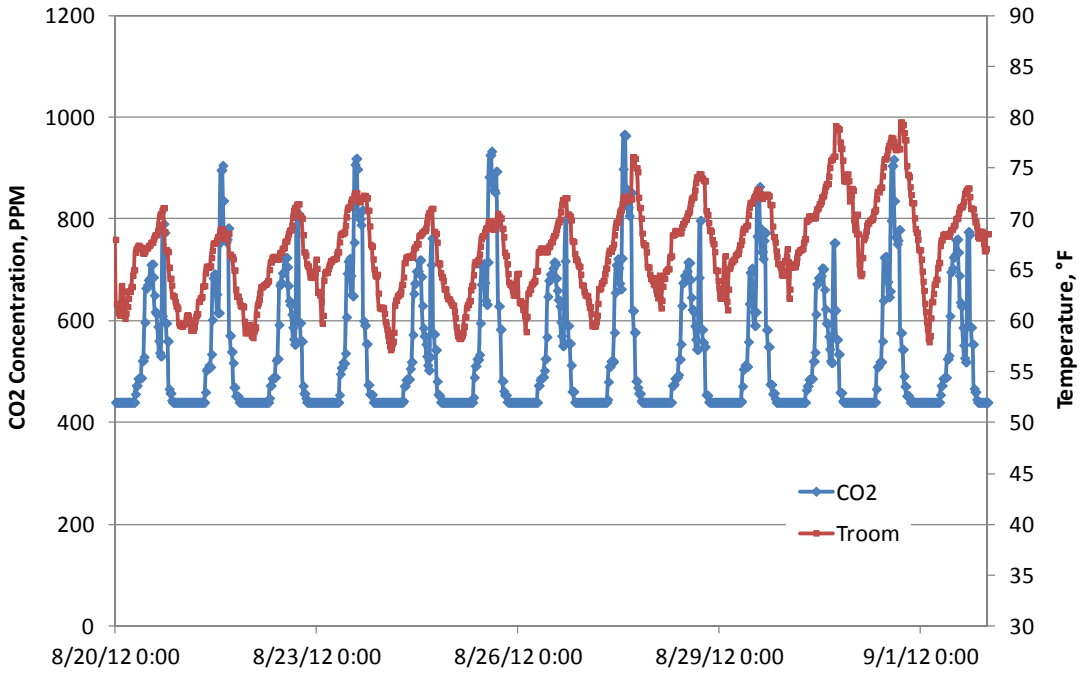


Figure 38: Predicted CO_2 Concentration during the Hottest Period in Summer for Proposal #1



As can be seen from Figure 37 and Figure 38, the predicted CO_2 concentration is less than 1,000 PPM and the highest room air temperature is 80°F (26.7°C). However, the biggest problem is the overcooling of the room during the night time air purge. For example, on June 9th, at 9am, the room air temperature is around 63°F (17.2°C). To address this problem, we developed Proposal #2:

The radiant slab remains off during summer and the operation of the windows will be:

- *Keep one window on north side open at 10% of full open area all the time to provide minimum ventilation air.*
- *If the next day forecast highest outside air temperature is between 80°F (26.7°C) and 86°F (30°C), open 40% of windows on both sides at 7pm and close them at 8am.*
- *If the next day forecast highest outside air temperature is beyond 86°F (30°C), fully open the windows on both sides at 7pm and close them at 8am.*

Figure 39: Predicted CO2 and Room Air Temperature during the Coldest Period in Summer for Proposal #2

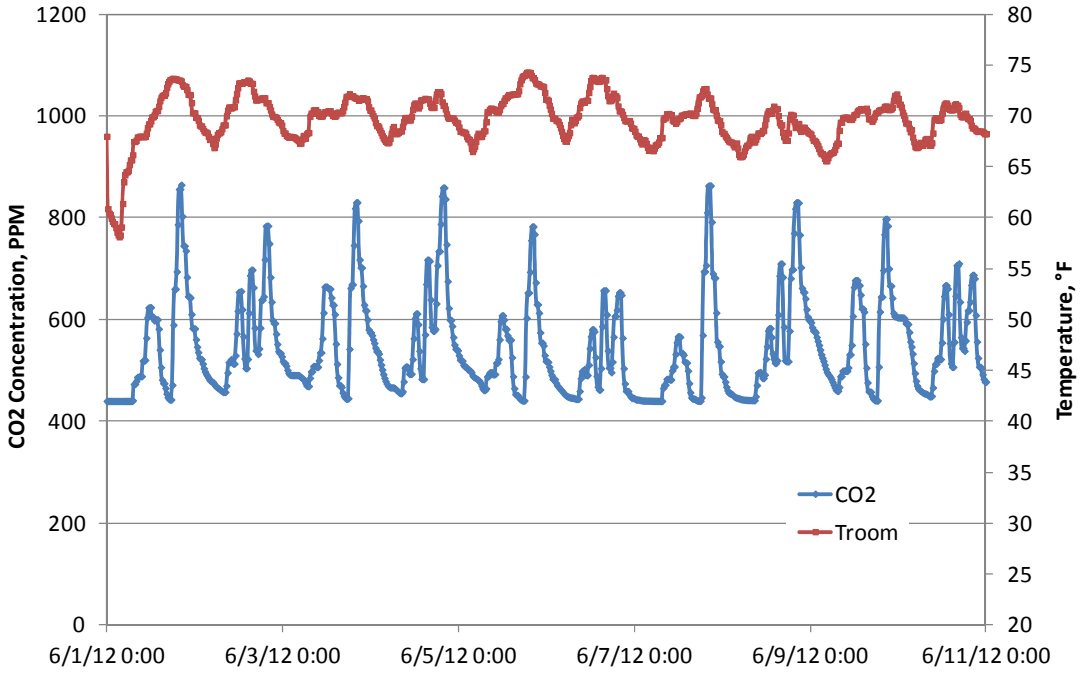
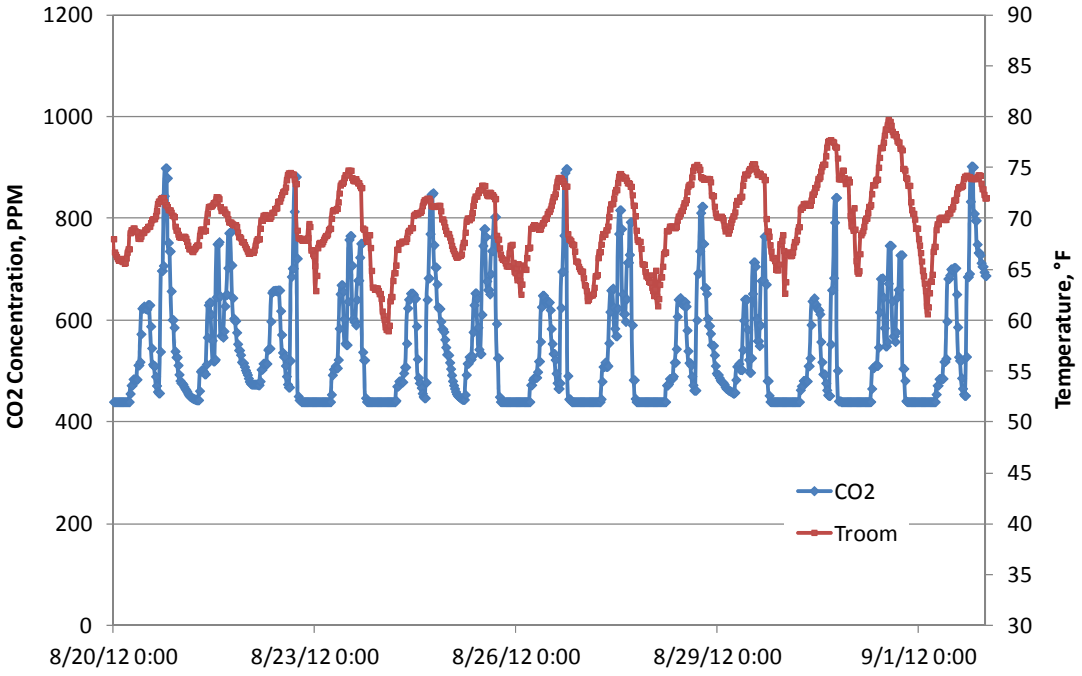


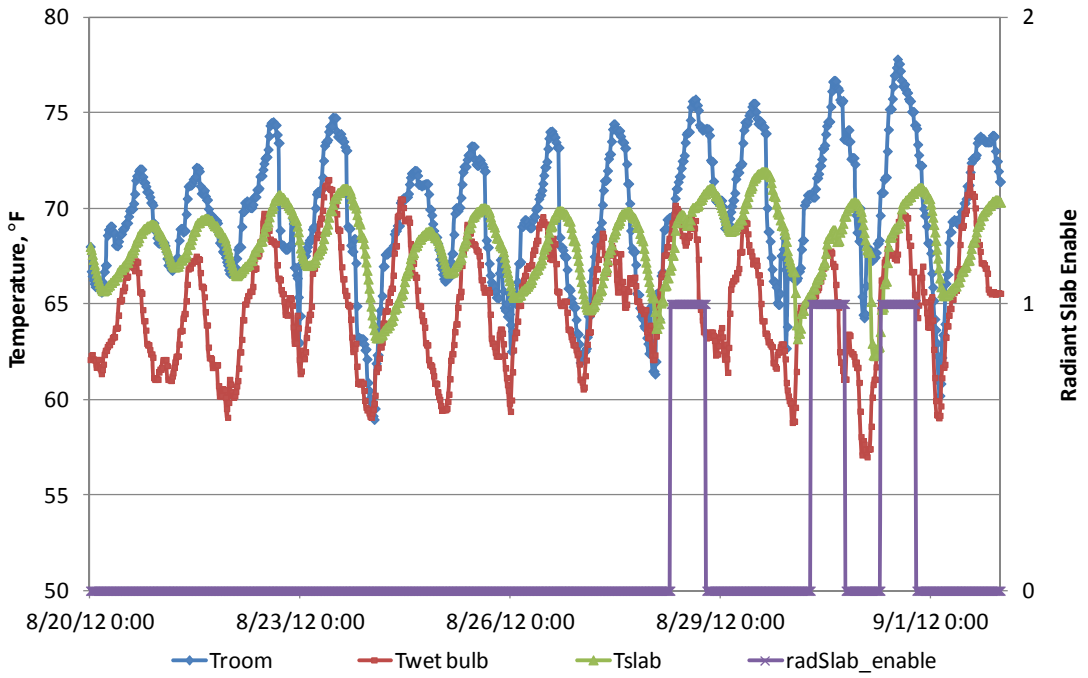
Figure 40: Predicted CO2 and Room Air Temperature during the Hottest Period in Summer for Proposal #2



As shown in Figure 39 and Figure 40, the room air temperature during the coldest period in summer and the CO_2 concentration are both improved considerably. In this case, during the coldest period, the lowest room air temperature at 9am (the class starting time) is around 68°F (20°C). With Proposal #2, the only concern left would be the slightly high room air temperature, i.e. 80°F (26.7°C), during hot days. When room air temperature rises above 78°F (25.6°C), occupants may start to complain. The radiant slab system could be used during the unusual hot days. This yields Proposal #3:

The window operation is the same as Proposal #2. If the next day forecasted highest outside air temperature is above 86°F (30°C), turn on the radiant slab system two hours earlier than the occupied starting hour (9am) and turn it off 2 hours earlier than the occupied ending hour (9pm). The supply cold water temperature should be 5.4°F (3 Kelvin) higher than the outside air wet bulb temperature. This can be provided by evaporative cooling, without requiring a compressor to run.

Figure 41: Predicted Room Air and Radiant Slab Temperature during Hottest Period in Summer for Proposal #3



The system performance during the coldest period in summer is the same asProposal #2 since the radiant slab system does not switch on during that period. Figure 41 shows the improved room air temperature by implementing Proposal #3, where the highest room air temperature drops from 80°F (26.7°C) to 78°F (25.6°C). In Proposal #3, we recommend the use of evaporative cooling equipment to take full advantage of the radiant slab system and the moderate outside air.

With the above analyses, we recommend two options to the building owner:

1 Proposal #2 to achieve the best energy performance with occupants experiencing a bit of high room air temperature, up to 80°F (26.7°C).

2 Proposal #3 to achieve better indoor comfort, by switching on the radiant slab system with modest energy usage if all cooling is provided by evaporative cooling without the use of a compressor.

5.1.5 Conclusion

A classroom in a real building is modeled using the Modelica Buildings library and calibrated with measured data. This classroom is primarily designed as a naturally ventilated space with supplementary radiant floor slab heating and cooling. The implemented control sequences were found to be different from the design-intent. The measured data indicates that the operable windows were not operated to utilize the cross ventilation capability. Three control options were analyzed and their performances were communicated to the building owner. All control options allow manual operation of the windows. The simulation results show that use of evaporative cooling is sufficient to maintain thermal comfort during the hottest period.

5.2 Chilled Water Plant for a Data Center

5.2.1 Problem Description

The next example will illustrate how Modelica can be used to implement, test, and improve the performance of a user-defined control sequence. The control sequence is for a cooling plant with a water-side economizer. It contains a state graph with delay on the state transitions that is used to enable and disable the water side economizer, a continuous time on/off control with event detection and hysteresis to enable and disable the chiller, and discrete time control for the setpoint scheduling. The control sequence is implemented by combining block- diagrams and a state graph with acausal models of the physical components of the chilled water plant. The objective is to improve the energy efficiency of a chilled water plant by optimizing the control setpoints. The chilled water plant is used to cool a data center in San Francisco, CA, with a constant cooling load of 500 kW.

5.2.2 Model of the Plant

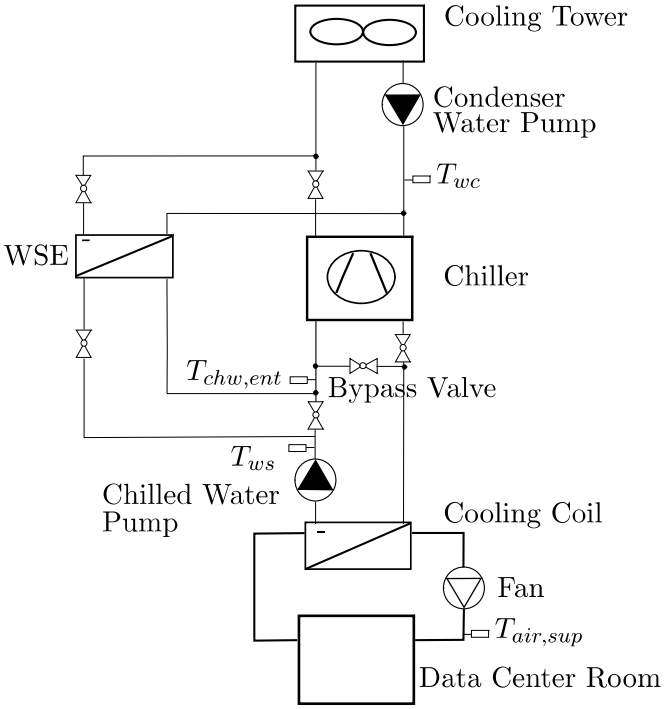
Two models were analyzed, one with and one without a water-side economizer (WSE). The WSE is a heat exchanger added to the water loop of the chilled water plant to provide free cooling. Figure 42 shows the schematic model and the Modelica system model of the chiller plant with WSE. The room of the data center was modeled using a mixed air volume with a heat source. Heat conduction and air infiltration through the building envelope were neglected since the heat exchange between the room and the ambient environment was small compared to the heat released by the computers. The chiller model is based on performance curves and has a variable speed compressor. The cooling tower has a variable speed fan and uses a performance curve to compute the approach temperature at off-design conditions. The water-side economizer is a steady-state heat exchanger model with constant effectiveness of 0.8. The chilled water flow rate is variable and controlled as described below. The supply air mass flow

rate to the data center is constant, and the supply air is cooled using a finite element cooling coil model.

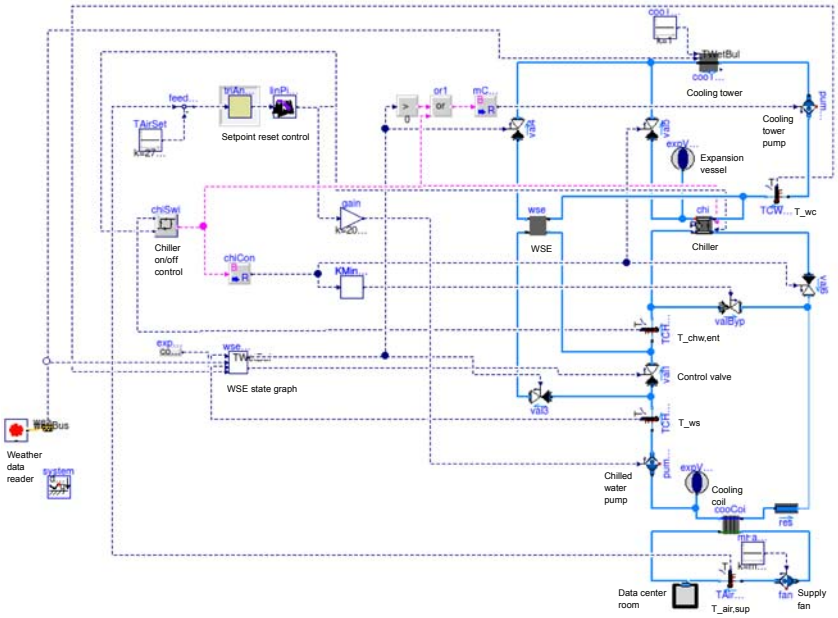
5.2.3 Model of the Control

The control objective was to maintain the temperature of the supply air to the room, while reducing energy consumption of the chilled water plant. The control was based on the control sequence proposed by Stein (2009). To simplify the implementation, we only applied the controls for the differential pressure of the chilled water loop, the setpoint temperature of the chilled water leaving the chiller, and the chiller on/off control. For the system with WSE, there was an additional enable/disable control for the WSE recommended by Taylor (2010). This control sequence was independent from the other control sequences. The pump for the condenser water loop, the cooling tower fan and the supply air fan were running at constant speeds. The detailed control sequence for the chilled water plant with WSE is described in Stein (2009).

Figure 42: Schematic Diagram and Modelica System Model of the Chilled Water Plant with WSE



(a) Schematic diagram

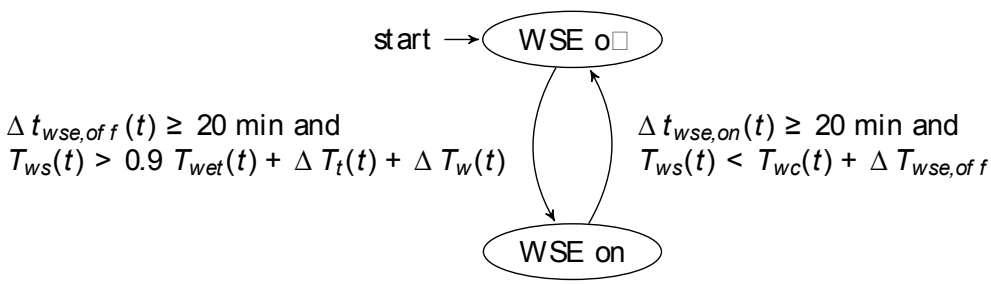


(b) Modelica model

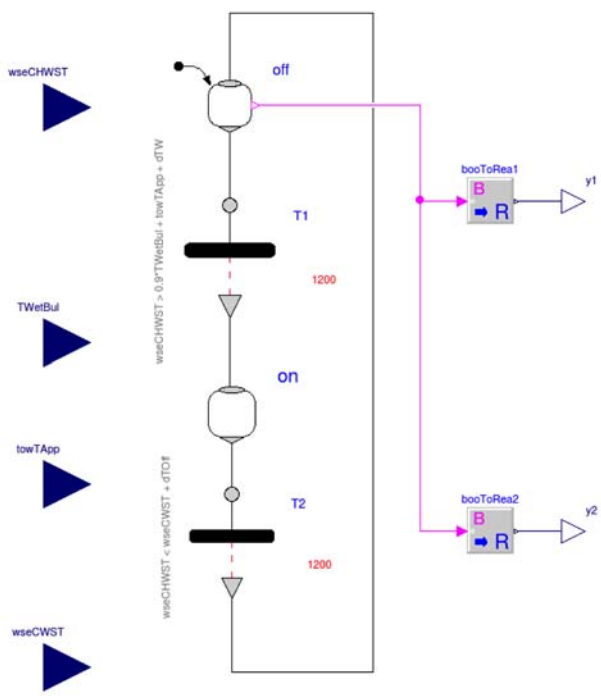
The chiller is switched on if $T_{chw,ent}(t) > T_{chi,set}(t) + \Delta T_{chi,ban}$ and switched off if $T_{chw,ent}(t) < T_{chi,set}(t)$, where $T_{chw,ent}(t)$ is the temperature of the chilled water entering the chiller, $T_{chi,set}(t)$ is the setpoint temperature of the chilled water leaving the chiller, and $\Delta T_{chi,ban} = 2.2$ Kelvin is the dead-band temperature to prevent chiller short cycling.

Figure 43 shows the state graph of the WSE control. The WSE is enabled if $\Delta t_{wse,off}(t) >=$ 20 minutes, and $T_{ws}(t) > 0.9\ T_{wet}(t) + \Delta T_t(t) + \Delta T_w(t)$, with temperatures in Kelvin, where $\Delta t_{wse,off}(t)$ is the time of the WSE in the off status, $T_{ws}(t)$ is the temperature of the chilled water leaving the cooling coil, $T_{wet}(t)$ is the wet bulb temperature, $\Delta T_t(t)$ is the temperature difference between the water leaving the cooling tower and the air entering the cooling tower, $\Delta T_w(t)$ is the temperature difference between the chilled water leaving the WSE and the condenser water entering the WSE.

Figure 43: State Graphs of the Chiller and WSE



(a) State graph of the WSE control.



(b) Implementation of the state graph of the WSE control.

The WSE is disabled if $\Delta t_{wse,on}(t) >= 20$ minutes, and $T_{ws}(t) < T_{wc}(t) + \Delta T_{wse,off}$, where $\Delta t_{wse,on}(t)$ is the time elapsed since the WSE was switched on, $T_{wc}(t)$ is the return water temperature of the cooling tower and $\Delta T_{wse,off} = 0.6$ Kelvin is an offset temperature.

The state graph is implemented in the Modelica model graphically as shown in Figure 43. On the left are the connectors for the control input signals, in the middle is the state graph and on the right are signal conversions from boolean to real signals, and the connectors for the output signals. In the state graph, the ovals are the states which are either "off" or "on". The arrow indicates the initial state. The horizontal bars are guards for the transitions.

The setpoint reset strategy is to first increase the differential pressure $\Delta p(t)$ of the chilled water loop to increase the chilled water flow rate. If $\Delta p(t)$ reaches the maximum value and further cooling is needed, then the chiller temperature setpoint $T_{chi,set}(t)$ is reduced. If there is too much cooling, then $T_{chi,set}(t)$ and $\Delta p(t)$ will be changed in the reverse direction.

This control strategy is realized using the trim and respond logic (Taylor 2007), which is as follows:

A cooling request is triggered if the input signal $y(t)$ is larger than a threshold y_δ. We used y_δ=0 and $y(t)=T_{air,sup}(t)-T_{air,set}$, where $T_{air,sup}(t)$ and $T_{air,set}$ are the actual and the set point temperatures of the supply air to the room. The sampling period was $\Delta t=120$ seconds.

The control signal $u(t)$ is computed as

$$u(t) = \begin{cases} \min(1, \max(0, u(t-\Delta t) + \Delta u_i)), & \text{if } y(t) \geq y_\delta, \\ \min(1, \max(0, u(t-\Delta t) - \Delta u_d)), & \text{otherwise,} \end{cases}$$

where $\Delta u_i = 0.03$ and $\Delta u_d = 0.03$ are parameters for increasing and decreasing $u(t)$.

As shown in Figure 44, the control signal $u(t)$ is converted to control setpoints as

$$\Delta p(t) = \begin{cases} \Delta p_{min} + (\Delta p_{max} - \Delta p_{min})\frac{u(t)}{x}, & \text{if } u(t) \in [0, x], \\ \Delta p_{max}, & \text{otherwise,} \end{cases}$$

and

$$T_{chi,set}(t) = \begin{cases} T_{chi,max}, & \text{if } u(t) \in [0, x], \\ T_{chi,max} - (T_{chi,max} - T_{chi,min})\frac{u(t)-x}{1-x}, & \text{otherwise,} \end{cases}$$

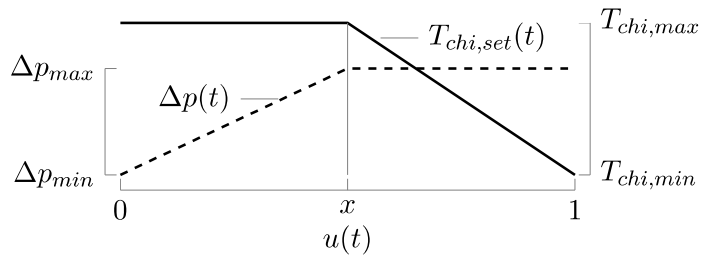
where $0 < x < 1$ is a user-selected parameter, Δp_{min} and Δp_{max} are the minimum and maximum differential pressure of the chilled water loop, and $T_{chi,min}$ and $T_{chi,max}$ are the minimum and maximum values of $T_{chi,set}(t)$.

For our experiments, we used $x=0.5$, Δp_{min} = 25.94 kPa and Δp_{max} = 129.7 kPa and $T_{chi,min}=5.56$°C, which is the minimum value in the range of performance curve of the selected chiller. We

varied the value of $T_{chi,max}$ for a parametric study to find the optimal value to reduce the system energy consumption.

The details will be discussed in the following paragraph.

Figure 44: Chilled Water Loop Differential Pressure and Chiller Setpoint Temperature Reset



To optimize the control setpoints for system energy efficiency, a parametric study was conducted in which we varied two control setpoints for the chilled water plants with and without WSE.

One parameter was the setpoint temperature for the supply air, $T_{air,set}$. The $T_{air,set}$ was varied from 18°C to 27°C, as these are the lower and upper bounds for the air supply temperatures for data centers recommended by ASHRAE (2011). The other independent variable was the maximum value of the chiller setpoint $T_{chi,max}$, which was varied from 6°C to 26°C. The `ElectricEIR` chiller model with a nominal cooling capacity of *742 kW* was used.

5.2.4 System Performance

The energy efficiency of the chilled water plant system was evaluated using the power usage effectiveness (PUE) as described in Belady et al. (2008). The PUE is the ratio of the total facility power usage, including power usage for the Information Technology (IT) and cooling system, divided by the power usage for the IT. A lower PUE indicates a more energy efficient cooling system.

Figure 45: Power Usage Effectiveness with and without Water-side Economizer

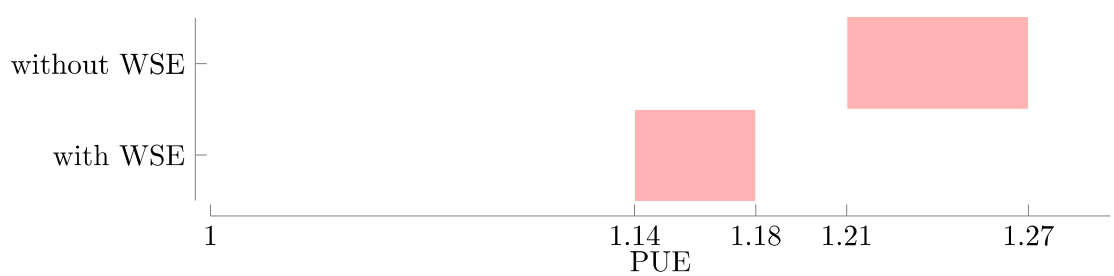


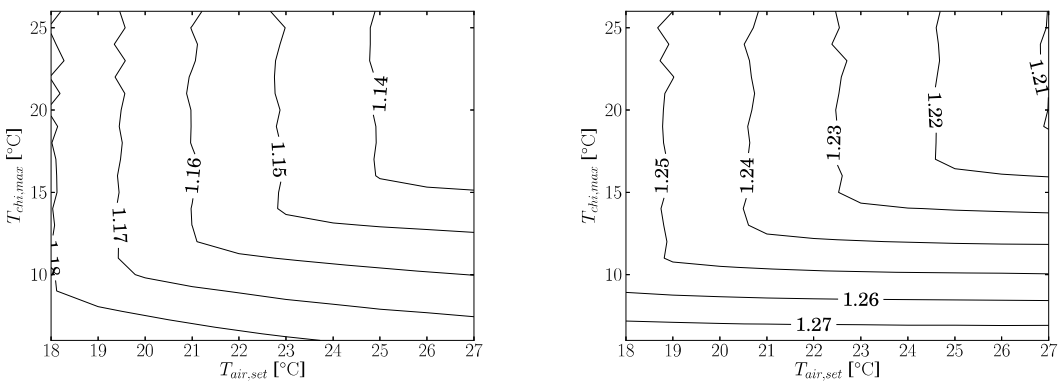
Figure 45 and Figure 46 shows the PUEs of the chilled water plant systems with and without WSE for different values of $T_{air,set}$ and $T_{chi,max}$. Figure 45, Figure 46(a), and Figure 46(b) show, for systems with and without a WSE, a range of PUEs between 1.14 to 1.18 and between 1.21 to 1.27, respectively. Compared to a cooling plant without WSE and $T_{chi,max}$ = 10 Celsius, an optimized WSE reduces the facility electrical energy use by 10%. For a *500* kW data center, at *0.2* $/kWh,◎reducing the PUE by 10% yields a *$90,000* annual reduction in electricity costs.

For both systems, increasing $T_{chi,max}$ can significantly reduce the PUE. There are two reasons for this. First, for both systems, the chiller is more efficient when working at higher evaporator temperatures. However, for a given air setpoint temperature, $T_{air,set}$, there is a maximum chilled water leaving temperature above which the temperature difference across the cooling coil is insufficient to provide the required cooling. If the cooling provided by the cooling coil is insufficient, then $y(t)$, the difference between the actual and the set point temperature of the supply air, is larger than zero. Then, it follows from the above equations that $u(t)$ increases, and hence, that the setpoint of the chilled water leaving temperature $T_{chi,set}(t)$ decreases. Therefore, there is an upper bound for $T_{chi,max}$ above which a further increase does not yield higher system-level efficiency because the chilled water is too warm, which will be compensated for by the trim and response logic. Second, for the system with WSE, increasing $T_{chi,max}$, and hence $T_{chi,set}(t)$, reduces the PUE because this increases the hours when the chiller is off.

For both systems, increasing $T_{air,set}$ also reduces the PUE if Tchi,max is larger than a certain value. The reason is that a higher air supply temperature leads to an increase in chilled water temperature, thereby increasing the coefficient of performance of the chiller and the number of hours during which the economizer can be used.

There are some oscillations on the vertical parts of the contour lines, for example at *PUE = 1.17* and *1.18* in Figure 46(a) and at *PUE = 1.23* to *1.25* in Figure 46(b). This is caused by the large values of $\Delta u_i=0.03$ and $\Delta u_d=0.03$ used by the trim and response logic. Further numerical experiments showed that reducing the values of Δu_i and Δu_d led to smoother PUE contour lines in Figure 46, but they caused over-heating and over-cooling of the supply air to the room. Since the control objective was that the temperature of the supply air $T_{air,sup}(\cdot)$ tracks the setpoint $T_{air,set}$, we used $\Delta u_i=0.03$ and $\Delta u_d=0.03$.

Figure 46: PUEs of Chilled Water Plant Systems with and without WSE



(a) With WSE (b) Without WSE

Figure 47: Annual Frequency Distributions of the chiller status as a Function of the Wet Bulb Temperature for Different Control Setpoints

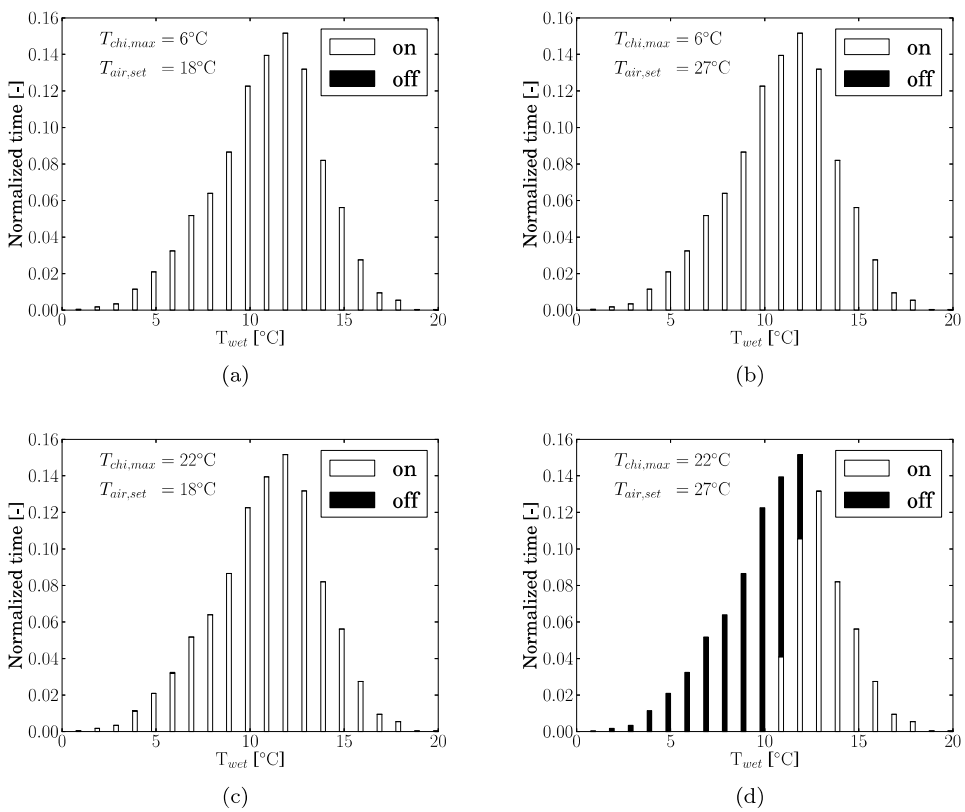
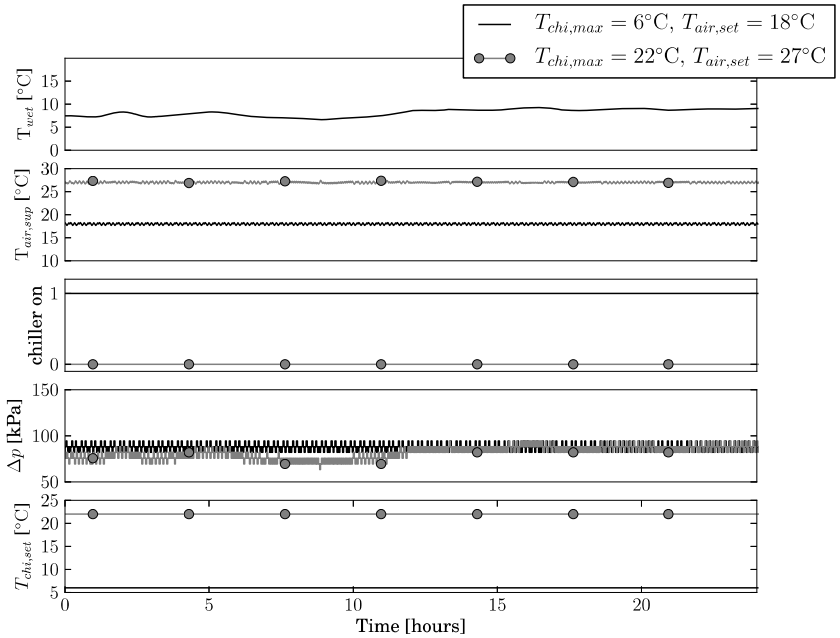


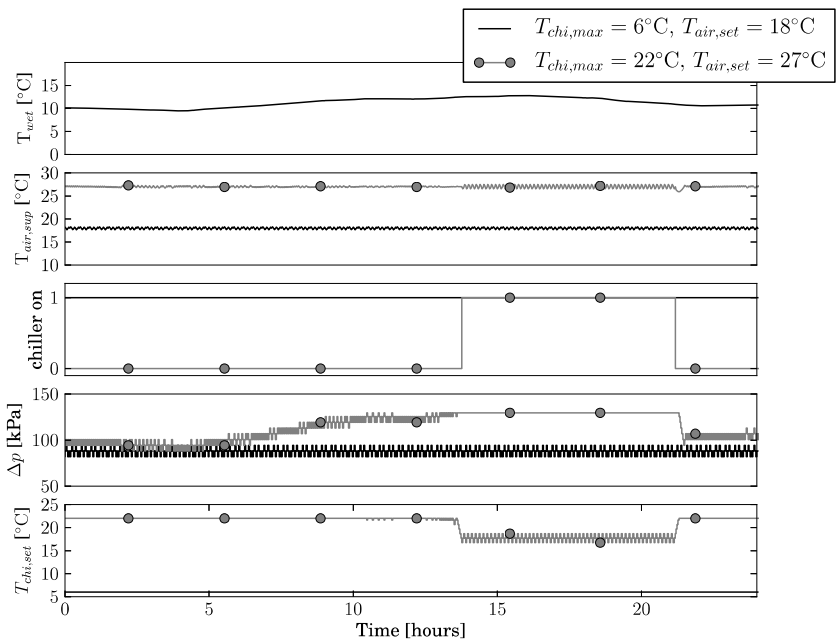
Figure 47 shows the outside wet bulb temperature $T_{wet}(t)$ and hours of corresponding chiller status for a whole year, for the system with WSE. When $T_{chi,max} = 6$ Celsius, the chiller was running for the whole year to meet the low setpoint temperature (Figure 47(a) and (b)). That

explains why the PUEs for $T_{chi,max} = 6$ Celsius are the highest among the system with WSE in Figure 46. Comparing Figure 47(a) with Figure 47(c) shows that increasing $T_{chi,max}$ from 6 Celsius to 22 Celsius or increasing $T_{air,set}$ from 18 Celsius to 27 Celsius alone does not increase the hours when the chiller is off. Both setpoints need to be increased to achieve a reduction in chiller operating hours.

Figure 48: Trajectories for the System with WSE



(a) Cold day



(b) Hot day

The Modelica models can also provide detailed transient information of the chilled water plant control. Figure 48 shows the control signals of the system with WSE under two different control setpoints on a cold day (January 14). On the cold day, the WSE can provide all the cooling needed and the chiller can be off for the whole day if the setpoint temperatures are high. With low setpoint temperatures, the chiller has to be on to cool the chilled water to the low value of $T_{chi,set}(t)$. The small oscillations of $\Delta p(t)$ are caused by the trim and respond logic which samples every 120 seconds.

Figure 48(b) shows the results for June 30, a hot day. With low setpoint temperatures, the chiller was on all the time. With high setpoint temperatures, the chiller was off in the morning. As the outside wet bulb temperature raises during the morning, the rate of heat transfer of the economizer is reduced. This is compensated for by an increase of the pressure setpoint $\Delta p(t)$ for the chilled water loop, thereby increasing the chilled water flow rate. In the early afternoon, this is insufficient, leading to a reduction of the chilled water set point $T_{chi,set}(t)$ after which the chiller switches on and remains on until the early evening.

GLOSSARY

ASHRAE- American Society of Heating, Refrigerating and Air-Conditioning Engineers

BAC- Building Automation System

CHW- Chilled Water

CO_2- Carbon dioxide

COP- Coefficient of Performance

HVAC- Heating, Ventilating, Air Conditioning

HW- Hot Water

IT- Infromation Technology

kW- Kilowatt

LBNL- Lawrence Berkeley National Laboratory

LEED- Leadership in Energy & Environmental Design

PUE- Power Usage Effectiveness

TAG- Technical Advisory Group

TMY3- Typical Meteorological Year3

WSE- water-side economizer

REFERENCES

ASHRAE 2011. American Society of Heating, Refrigerating and Air-Conditioning Engineers. Thermal Guidelines for Data Processing Environments. Expanded Data Center Classes and Usage Guidance.

ASHRAE, 2006. Sequences of Operation for Common HVAC Systems.

Belady, C. Rawson, A., Pflueger, J. and Cader, T., 2008. Green Grid Data Center Power Efficiency Metrics: PUE and DCIE, The Green Grid. White Paper #6. http://www.thegreengrid.org/~/media/WhitePapers/White_Paper_6_-_PUE_and_DCiE_Eff_Metrics_30_December_2008.pdf?lang=en [Last accessed 14-September-2012].

Deringer J.J., Nahman, J.E., Heming K., Wetter M., Pang X., and Konstantoglou M., 2012. LearnHPB and eLAD - Two Related Online eLearning Platforms for High Per- formance Buildings. In: ACEEE Summer Study on Energy Efficiency in Buildings, Aug., Pacific Grove, CA. http://www.aceee.org/files/proceedings/2012/data/papers/ 0193-000334.pdf [Last accessed 14-September-2012].

Eisenhower, B., Gasljevic, K., and Mezi´c, I., 2012. Control-Oriented Modeling and Calibration of Building Energy Models Using Modelica. In: Proc. of SimBuild, Aug., Madison, WI.

Franke, F. Casella, M. Otter, K. Proelss, M. Sielemann, and M. Wetter, 2009b. Standardization of thermo-fluid modeling in Modelica.Fluid. In F. Casella, editor, Proc. of the 7-th International Modelica Conference, Como, Italy.

Franke, F. Casella, M. Otter, M. Sielemann, H. Elmqvist, S. E. Mattsson, and H. Olsson, 2009a. Stream connectors – an extension of modelica for device-oriented modeling of convective transport phenomena. In F. Casella, editor, Proc. of the 7-th International Modelica Conference, Como, Italy.

Fritzson, Peter, 2004. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. John Wiley & Sons.

Fritzson, Peter, 2011. Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica. Wiley-IEEE Press, ISBN 978-1-1180-1068-6.

Nouidui, T.S., Phalak, K., Zuo, W., and Wetter, M., 2012. Validation and Application of the Room Model of the Modelica Buildings Library. In: Proc. of the 9-th International Modelica Conference, Sep., Munich, Germany.

Stein, J., 2009. Waterside Economizing in Data Centers: Design and Control Considerations. ASHRAE Transactions, 115 (2), 192–200.

Taylor, S.T., 2007. Increasing efficiency with VAV System Static Pressure Setpoint Reset. ASHRAE Journal, 24–32.

Taylor, S.T., 2010. Personal communication on October 22, 2010.

Tiller, Michael M., 2001. Introduction to Physical Modeling with Modelica. Kluwer Academic Publisher.

Wetter, M., 2006. Multizone Airflow Model in Modelica. In: C. Kral and A. Haumer, eds. Proc. of the 5-th International Modelica Conference, Vol. 2, Sep., Vienna, Austria, 431–440.

Wetter, M., 2009a. A Modelica-based model library for building energy and control systems. In: P.A. Strachan, N.J. Kelly and M. Kummert, eds. Proc. of the 11-th IBPSA Conference, Glasgow, Scotland, 652–659.

Wetter, M., 2009b. Modelica-based Modeling and Simulation to Support Research and Development in Building Energy and Control Systems. Journal of Building Performance Simulation, 2 (2), 143–161.

Wetter, M., 2011. Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed. Journal of Building Performance Simulation, 4, 185–203.