



**Politecnico
di Torino**

POLITECNICO DI TORINO

Laurea Magistrale in Ingegneria Energetica e
Nucleare

Sessione di Laurea di Luglio 2022

**An integrated simulation environment
coupling EnergyPlus and
OpenModelica for testing the
performance of building energy control
strategies**

Supervisors:

Prof. Alfonso Capozzoli
Silvio Brandi, PhD
Davide Fop, MSc

Candidate:

Giuseppe Razzano

Alla mia Famiglia
Un giorno, spero di ripagare il vostro sacrificio

Abstract

The growing need to use renewable energy sources in buildings, especially in non-residential buildings, has led to more attention towards building energy management and the integration of renewable sources with the electrical grid. Proper control of Heating, Ventilation and Air Conditioning (HVAC) systems, along with that of thermal and electrical power generation and storage plays a key role in the integration of such systems with power grid. Renewable sources such as solar thermal or PV are not programmable and therefore their availability does not follow the energy demand of the building. As a result, the ability of a building to be flexible in adapting to the unpredictability of renewable generation and responding at the same time to the needs of the grid has become a key aspect to be addressed. This flexibility can be achieved with energy storage systems managed through appropriate controls logics. In this context, software tools capable to simulate with sufficient detail the building thermal behaviour, the energy system dynamics and their respective control systems are becoming fundamental tools. In this thesis was developed the simulation model of an office building employing EnergyPlus to model building dynamics and OPENMODELICA to model PV generation, and the HVAC plant and its relative control system. The integration of two software was achieved through Spawn of EnergyPlus. Spawn is a new tool developed by the U.S. Department of Energy capable to integrate the descriptive and simulative capabilities of Energy plus with the capabilities of OPENMODELICA to simulate energy systems in detail. Such an integrated environment has recently established itself as the state of the art in the combined dynamic thermal simulation of buildings and their energy systems, allowing at the same time a detailed description of the control systems serving the plant. The two software used in this work, namely Energy Plus and OPENMODELICA, are completely open source and their co-simulation is carried out using the Functional Mockup Interface (FMI) standard. Rule-Based controls were tested at a high level i.e., controls managing the exchange with the grid and the charging and discharging of storages, and low level control logics, i.e. controls regulating the opening and closing of valves and dampers of the thermal system. The proposed control strategies were compared through Key Performance Indicators (KPI) to a traditional control system employed as a baseline. The enhanced Rule-Based control was capable to enhance self-consumption and to lower energy cost of the analysed system. The results of the present work proved the capabilities of the implemented

simulation tool in effectively modelling an energy system with the aim of evaluating the effect of control strategies in buildings.

Contents

| | |
|---|-----------|
| Abstract | 1 |
| List of Figures | 3 |
| List of Tables | 5 |
| 1 Introduction | 7 |
| 1.1 Digital twin | 7 |
| 1.2 Introduction to co-simulation | 8 |
| 1.3 Thesis structure | 9 |
| 2 Literature review | 11 |
| 2.1 CO-simulation environments | 11 |
| 2.1.1 Energy Plus | 12 |
| 2.1.2 Modelica language | 13 |
| 2.2 Functional Mock-up Interface | 20 |
| 2.3 Spawn of Energy Plus | 30 |
| 2.3.1 Historical overview | 30 |
| 2.3.2 Spawn overview | 31 |
| 2.3.3 Simulation framework | 32 |
| 3 Control strategies overview | 35 |
| 3.1 Traditional control strategies (TCS) | 37 |
| 3.1.1 ON/OFF control | 37 |
| 3.1.2 PID control | 41 |
| 3.2 Advanced control strategies (ACS) | 43 |
| 3.2.1 Reinforcement learning | 45 |
| 3.2.2 Model predictive control (MPC) strategies | 46 |
| 4 Case study | 51 |
| 4.1 Case study description and simulation setup | 51 |

| | | |
|----------|---|-----------|
| 4.2 | Building description | 52 |
| 4.3 | Thermal and electrical system | 53 |
| 4.3.1 | Thermal system | 53 |
| 4.3.2 | Electrical system | 56 |
| 4.4 | Design | 56 |
| 4.4.1 | Building thermal load | 57 |
| 4.4.2 | Nominal mass airflow rate | 57 |
| 4.4.3 | Nominal outdoor airflow rate | 57 |
| 4.4.4 | Heating Coil | 58 |
| 4.4.5 | Heat pump | 58 |
| 4.4.6 | Humidifier | 59 |
| 4.4.7 | Temperature energy storage | 59 |
| 4.4.8 | VAV box | 60 |
| 4.4.9 | PV | 61 |
| 4.4.10 | Electrical storage | 62 |
| 4.4.11 | Schedule | 62 |
| 4.5 | Control system | 63 |
| 4.5.1 | High-level | 63 |
| 4.5.2 | Low-level | 69 |
| 5 | Results | 75 |
| 5.1 | Low-level | 75 |
| 5.2 | High-level | 76 |
| 6 | Conclusion | 89 |
| | Bibliography | 91 |

List of Figures

| | | |
|------|---|----|
| 1.1 | <i>Application framework of digital twin technology [6]</i> | 9 |
| 2.1 | <i>Energy Plus structure [10]</i> | 13 |
| 2.2 | Three different way to co-simulate Energy plus with another program | 14 |
| 2.3 | <i>Schematic for the CoolingToAmbient model</i> | 18 |
| 2.4 | <i>Structure of a zip-file of an FMU. [11]</i> | 21 |
| 2.5 | <i>Time-continuous (x) and time-discrete (m). [16]</i> | 23 |
| 2.6 | <i>Three tools are controlled by one master. [16]</i> | 24 |
| 2.7 | <i>Architecture of the FMU import. [11]</i> | 25 |
| 2.8 | FMU of the components of HVAC | 29 |
| 2.9 | <i>Architectural diagram of Spawn. [20]</i> | 32 |
| 2.10 | <i>Partitioning of the envelope, room and HVAC model. [20]</i> . . . | 32 |
| 2.11 | <i>Simulation framework using Spawn. [21]</i> | 34 |
| 3.1 | <i>Features, applications and functions of BACS in buildings and HVAC systems. [23]</i> | 36 |
| 3.2 | <i>ON-OFF control. [24]</i> | 38 |
| 3.3 | <i>Momentary control algorithm. [26]</i> | 40 |
| 3.4 | <i>Theflowchart of predictive rule-based control algorithms [27]</i> . | 41 |
| 3.5 | <i>Proportional feedback control system. [28]</i> | 42 |
| 3.6 | <i>Closed loop PI control. [28]</i> | 43 |
| 3.7 | <i>Closed loop PD control. [28]</i> | 43 |
| 3.8 | <i>nteraction between agent and environment in a Markov decision process. [30]</i> | 46 |
| 3.9 | <i>MPC receiding horizon strategy. [32]</i> | 47 |
| 4.1 | <i>Functional diagram of the system.</i> | 52 |
| 4.2 | Simple office model [34]. | 52 |
| 4.3 | <i>Building on Modelica</i> | 53 |
| 4.4 | Thermal system. | 55 |

| | | |
|------|---|----|
| 4.5 | <i>Electrical system</i> | 56 |
| 4.6 | <i>Psychrometric chart[37]</i> | 57 |
| 4.7 | <i>Element constituting a VAV box in modelica</i> | 61 |
| 4.8 | <i>Schedule.</i> | 63 |
| 4.9 | <i>Control level.</i> | 64 |
| 4.10 | <i>Control of maximum charging and discharging power.</i> | 65 |
| 4.11 | <i>SOC status</i> | 66 |
| 4.12 | <i>Control of state of charge (SOC).</i> | 67 |
| 4.13 | <i>Delivered energy for the heating system versus the maximum temperature set point [26].</i> | 68 |
| 4.14 | <i>Baseline control flowchart</i> | 69 |
| 4.15 | <i>Self-consumption control flowchart</i> | 70 |
| 4.16 | <i>Supply Air Temperature diagram [2]</i> | 71 |
| 4.17 | <i>Economizer outdoor air damper position [2]</i> | 72 |
| 4.18 | <i>VAV box control.</i> | 73 |
| 5.1 | <i>Temperature of five zone.</i> | 76 |
| 5.2 | <i>Temperature and actuators position of south and west zones</i> | 77 |
| 5.3 | <i>water and air inlet and outlet temperature of the heating coil.</i> | 78 |
| 5.4 | <i>Temperature and actuators position of the economiser.</i> | 79 |
| 5.5 | <i>Selfconsumption-capacity correlation baseline strategy.</i> | 80 |
| 5.6 | <i>Cost-capacity correlation baseline strategy.</i> | 81 |
| 5.7 | <i>Temperature of TES.</i> | 81 |
| 5.8 | <i>Electrical loads baseline strategy.</i> | 81 |
| 5.9 | <i>Selfconsumption-Volume TES correlation.</i> | 83 |
| 5.10 | <i>Load building power-Volume TES correlation</i> | 83 |
| 5.11 | <i>Energy supplied by the heat pump</i> | 84 |
| 5.12 | <i>Selfconsumption-capacity correlation for SC strategy.</i> | 84 |
| 5.13 | <i>Cost-capacity correlation for SC strategy.</i> | 85 |
| 5.14 | <i>self-consumption control with a cut off part of the flow chart.</i> | 85 |
| 5.15 | <i>Self-consumption for several SOC limit.</i> | 86 |
| 5.16 | <i>Percentage self-consumption improvement</i> | 86 |
| 5.17 | <i>SOC of TES and battery</i> | 87 |
| 5.18 | <i>Electrical loads SC strategy compared to baseline strategy</i> | 88 |

List of Tables

| | | |
|-----|--|----|
| 2.1 | Use cases with different system configurations [11] | 27 |
| 3.1 | Advantages and disadvantages of ON/OFF control techniques.[25] | 39 |
| 3.2 | Advantages and disadvantages of PID control techniques.[25] | 44 |
| 3.3 | Advantages and disadvantages of MPC control techniques.[23] | 49 |
| 4.1 | <i>Building characteristics</i> | 54 |
| 4.2 | <i>Sizing summary</i> | 58 |
| 4.3 | <i>Capacity of TES</i> | 60 |
| 4.4 | <i>inputs Modelica's PV model.</i> | 61 |
| 4.5 | <i>Capacity of electrical storage</i> | 62 |
| 5.1 | <i>Bi-hourly rate</i> | 78 |
| 5.2 | <i>Results baseline strategy</i> | 80 |
| 5.3 | <i>Results selfconsume strategy</i> | 82 |

CHAPTER 1

Introduction

The building sector is playing a main role in the energy transition and global warming mitigation process because it is responsible of 40% of the global energy consumption and of 36% CO₂ emissions [1]. Consumption is due to the fact that people spend most of their time inside buildings trying as much as possible to be in comfort situation. HVAC systems in buildings, especially in the non-residential sector, are the largest energy consuming systems; in fact there is a growing attention on the control of these systems. Needs to reduce CO₂ emissions, increase energy efficiency with the integration of renewable energy sources and the consequent adaptation of electric and thermal grids, have led to the improvement of the existing control systems. For example the publication of Guideline 36 by ASHRAE [2] and the increasingly large number of publications concerning Model Predictive Control (MPC) applications are evidence of a growing interest in novel control techniques [3]. While these new control algorithms are promising in terms of energy savings, their large-scale use remains difficult. According to Blum et al. in [4] the first challenge is « *benchmarking the performance of these algorithms against state-of-the-art algorithms and each other* » and a second challenge is « *the effort and expertise required for setting up and executing such individualized tests* ».

As mentioned in [5], the HVAC system is responsible for about 50% of the building's energy use; the Digital twin (DT) can be the key to energy savings.

1.1 Digital twin

DT can be defined by Zheng et al in [6] explains: « *the DT is a set of virtual information that fully describes a potential or actual physical production from the micro atomic level to the macro geometrical level. At its optimum, any*

information that could be inspected from a physical manufactured product can be obtained from its DT». In other words, DT is an integrated system capable of duplicating the physical behaviour of a system in its digital dual.

There are three main components in a DT system as shown in figure 1.1:

- the physical space consists of all the intelligent sensors and equipment used to collect and transmit data. The mapping (direct and indirect) between data and device attributes makes the interaction between physical and digital space possible.
- the digital space consists of two parts, the virtual environment platform (VMP) and the DT application subsystem (DTs). The VMP builds a virtual model and provides an operating environment that contains the algorithm libraries. The DTs accumulate various models and historical data created during operations and whose attributes can be obtained in the database; feedback from the virtual model is also accumulated in the database. In the DT system, historical data from the virtual model, real-time data and physical products are combined to drive the DTs synchronously.
- the information processing layer is the bi-directional link between physical space and virtual space;

The virtual model must realistically describe the physical model, this is the context of co-simulation; co-simulation environments and methods will be the main topics of this thesis work.

1.2 Introduction to co-simulation

Building energy performance simulation is a technology used to evaluate the performance of buildings and systems, or the use of different types of control on existing buildings. The complex nature of these system makes it difficult to simulate these configuration but **co-simulation** could be an efficient and flexible approach capable of combining the positive sides of the various simulation environments. The different simulators are «*interfaced with each other and exchange data in order to work even at heterogeneous runtimes*» and «*increase the transferability of solutions onto different application cases*» [7]. Although there are many advantages, many difficulties still need to be overcome: a process requires different experts from different domains which often implies a certain amount of simplification.

There is a need to find tools using unified standards to simplify the way data is exchanged.

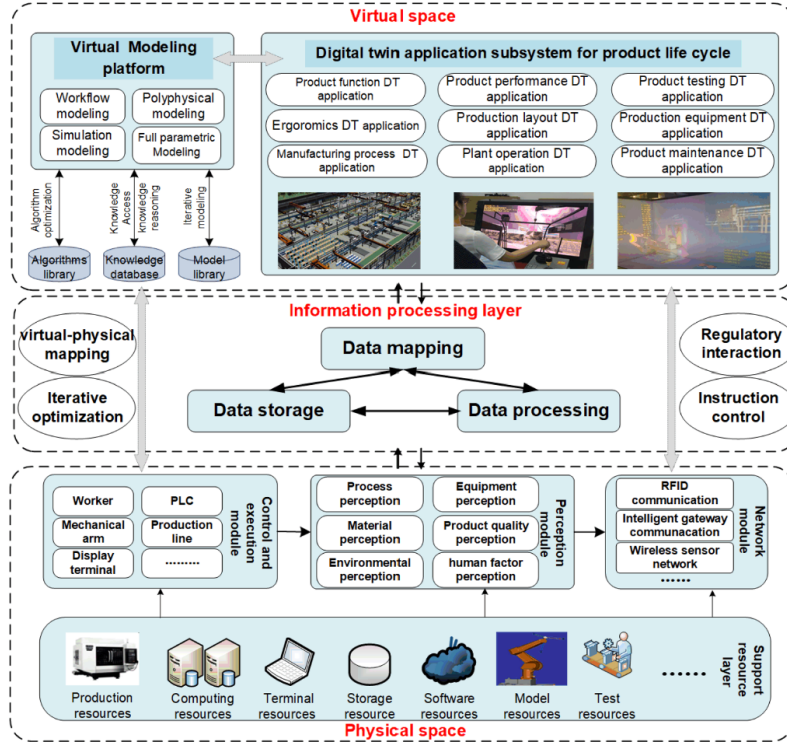


Figure 1.1: Application framework of digital twin technology [6]

A popular standard for co-simulation is the **Functional Mock-up Interface** (FMI), which contains the data and functions necessary for the simulation.

1.3 Thesis structure

The aim of this thesis work is to illustrate the tool of a co-simulation environment in which models of the building, HVAC system, thermal and electrical generation systems and their control logic are simulated simultaneously.

Chapter 2 shows the simulation environments EnergyPlus and Modelica, and the open standard that allows communication between them, i.e. FMU. Finally, the tool that unites these simulation environments is shown, namely Spawn of EnergyPlus.

Chapter 3 shows an overview of traditional and advanced control strategies. Chapter 4 shows a case study i.e. an application of a co-simulation environment through the Spawn of EnergyPlus tool in which the relevant systems are sized and traditional control logics are used.

Finally, in Chapter 5, the results of the case study are shown and analysed

to highlight the potential of a co-simulation environment.

CHAPTER 2

Literature review

2.1 CO-simulation environments

Co-simulation is defined in [8] as: *«[application] in which at least two simulators, each solving an initial-value differential or difference equation, are coupled to exchange data that depend on state variables»*. In according to [9], co-simulation can be advantageous in that it allows to:

- divide one model in sub-systems and each system can be analyzed by a suitable simulation
- study each sub-system with an appropriate level of detail and allow interaction with heterogeneous model
- facilitate teams working in parallel mode and preserve intellectual property (IP) rights

Co-simulation is used in different fields like mechanical, electrical and aerospace engineering but in the last few years the applications in the field of energetics have increased; in particular buildings systems need to be simulated in different physical domains that are related to each other. The variables of different domains will be used for the data exchange during the simulation run period; the simulation users have to chose the variables to exchange so that they represent the physical reality of the problem.

Simulation of modern buildings requires a model that represents the dynamics of the envelope and a model that describes a flexible thermal plant along with its control systems.

2.1.1 Energy Plus

Energy Plus is a building energy simulation program that is a combination of two old program: BLAST and DOE-2 [10]. It is written completely in Fortran and it does not have an interface but rather input and output text files.

An heat balance engine calculates loads each specified time step (default setting is 15 minutes), then at same time step, the building systems simulation module receives the value of the loads and calculates heating, cooling and electrical system response with a variable time step (down to seconds); through a feedback system, loads that are not met are inserted in the next phase of the load calculation. As is shown in 2.1 Energy plus has three basic components:

1. *The simulation manager* that controls the entire simulation process.
2. *The heat and mass balance simulation module* based on IBLAST that integrates HVAC system.
3. *The building systems simulation module* handles communication between the heat balance engine and various HVAC modules.

The possibility of extending Energy Plus to other domains through co-simulation, makes this software optimal for the description of buildings in their entirety.

There are three different way to co-simulate Energy plus with another program as is described in [11]:

1. one-to-one coupling (figure 2.2a): a specific interface is necessary to enable the communication between EnergyPlus and another program. The advantage of these approach is a fast communication while a disadvantage is the necessity to change the code every time Energy Plus has to be linked to another program. With this approach the main use is the connection between Energy plus and CFD tool to calculate accurately heat transfer between room air and walls for computation of load in Energy plus
2. middleware coupling (figure 2.2b) : it's a co-simulation master that manages and organizes the data exchange between Energy Plus and program B; with this approach each program needs a specific interface to communicate with the middleware software. The Building Controls Virtual Test Bed (BCVTB) [8] is an example of middleware coupling that can co-simulate Energy plus with programs like Matlab, Simulink or Dymola

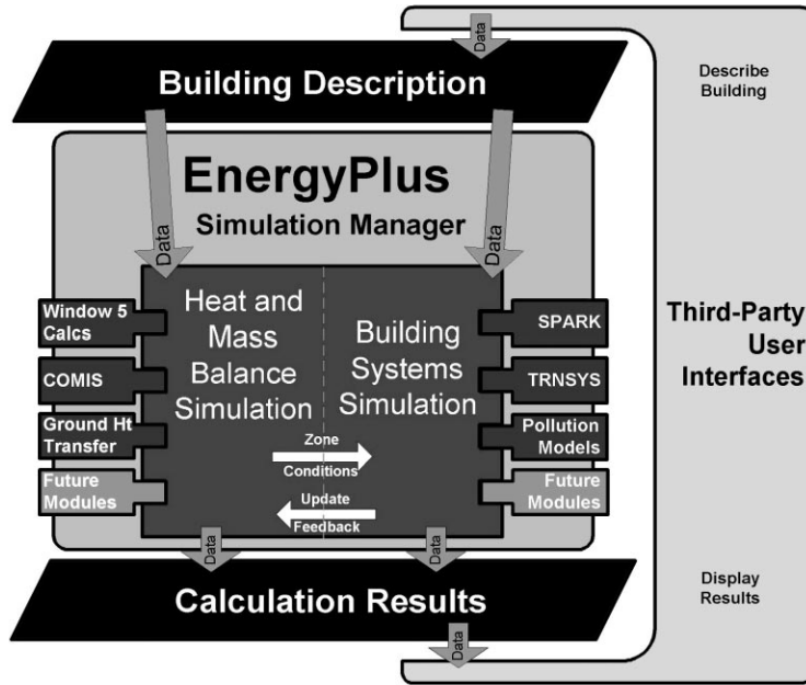


Figure 2.1: *Energy Plus* structure [10]

3. standardize the co-simulation interface (figure 2.2c): all programs, including Energy Plus, can use a standard interface in a way to combine all advantages of first and second approach. An increasingly adopted standard interface is **Functional Mock-up Interface (FMI)** that will be discussed in depth in 2.2.

2.1.2 Modelica language

The Modelica language is now widely used in industry and research for the modeling and simulation of physical systems of various types. The term model refers to a mathematical description which can lead to conclusions about the physical behavior of a complex system, provided that such model is sufficiently accurate. As reported in [12] «*Modelica is a non-proprietary modeling language and the name is a trademark of the Modelica Association which is responsible of the publication of the Modelica language specification*». As shown in [13] Modelica was written starting from other languages like Dymola and Omola which were unified for the purpose of defining a modern and uniform modeling language. An important moment in the history of this

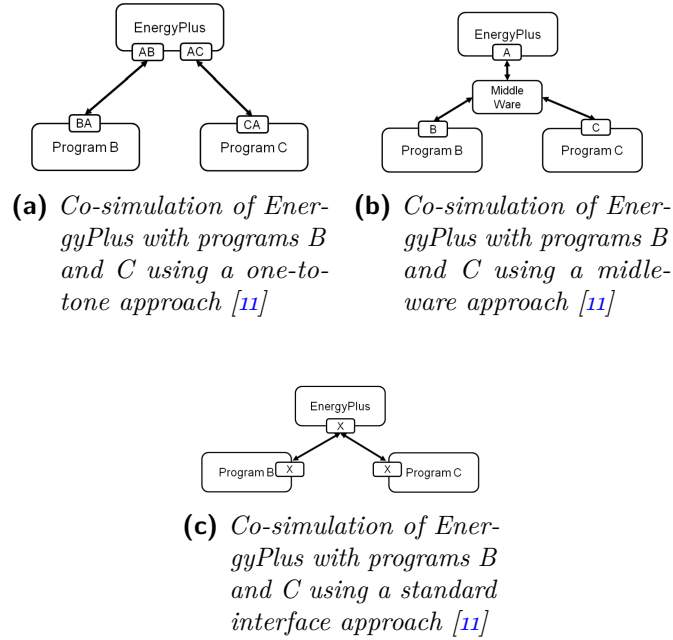


Figure 2.2: Three different way to co-simulate Energy plus with another program

language came in 1988 with the development of the *Pantelides* algorithm for differential algebraic equation index reduction. The first version of Modelica was finished in September 1997, when Hilding Elmqvist unified the various and fragmented modeling languages. This modeling language is important for the study of several engineering domains because it can be used to solve problems expressed through differential algebraic equations, describing the behavior of continuous variables and it can also handle discrete variables useful for describing control signals.

The Modelica language is described using a precise vocabulary; the following terminology will be adopted in throughout this section:

model. A model is analogous to a class, that is a behavioral description of physical phenomena.

component. A component is a an instance of a model.

subcomponent. A subcomponent is a component that is contained within other components.

system model. A system model is a model that does not have any external connection and it contains the same number of equations and variables.

quantity. In Modelica each entity has a value that can be either real, integer, string or boolean.

definition. The model definition is a description of all the equations associated to the model, along with its parameters and variables.

declaration. We have a declaration when a component, variable, parameter or constant is instantiated.

package. A collection of Modelica models is called a package; multiple packages can be used together.

keyword. A keywords are words such as *component*, that has a specific meaning in Modelica and so such words cannot be used in declarations.

Modelica model

One of the strengths of the Modelica programming language is the description of dynamic systems through mathematical equations. Below in 2.1 it is shown how is simple it is to characterize the physical behavior of a pendulum. The model name is given using the keyword *model* followed by the actual name which is *SimplePendulum* in this case. Next the parameters and constants are defined by variables that are in the equations that characterize the model. The equations are written preceded by the keyword *equation* with the operator *der* which represents the time derivative of the variables.

Code 2.1: Model of a simple pendulum [14].

```
1 model SimplePendulum
2   parameter Real L=2;
3   constant Real g=9.81;
4   Real theta;
5   Real omega;
6 equation
7   der(theta)=omega;
8   der(omega)=-(g/L)*theta;
9 end SimplePendulum;
```

As shown in [12], Modelica includes features which allow to specify physical types such as mass, pressure, temperature etc. Code 2.2 describes an

example of Newton's law using physical types that are contained in package *Modelica.Units.SI*. This package provides predefined types based on the International standard on units.

Code 2.2: Model of a Newton's law [14].

```

1  model NewtonCooling "An example of Newton's law of cooling"
2    parameter Modelica.Units.SI.Temperature T_inf=25+273.15 ...
   "Ambient temperature";
3    parameter Modelica.Units.SI.Temperature To=90+273.15 ...
   "Initial temperature";
4    parameter Modelica.Units.SI.CoefficientOfHeatTransfer h=0.7 ...
   "Convective cooling coefficient";
5    parameter Modelica.Units.SI.Area A=1.0 "Surface area";
6    parameter Modelica.Units.SI.Mass m=0.1 "Mass of thermal ...
   capacitance";
7    parameter ...
   Modelica.Units.SI.SpecificHeatCapacityAtConstantPressure ...
   c_p=1.2 "Specific heat";
8    Modelica.Units.SI.Temperature T "Temperature";
9    initial equation
10     T = To "Specify initial value for T";
11    equation
12     m*c_p*der(T) = h*A*(T_inf-T) "Newton's law of cooling";
13  end NewtonCooling;

```

It is possible to reuse constitutive equations like Newton's law creating reusable component models and then connecting this model together to form complex networks. This new type of definition is called *connector*. A *connector* is a connection between two models that exchange information.

Modelica distinguishes two types of connections:

- Block Diagrams
- Acausal connections

Any given block in a block diagram has the form described in 2.1 and 2.2 where u represents the input signals, x the internal states and y the output signals as explained in [12]. Block diagrams are useful to understand the mathematical behavior behind dynamic systems and are used to describe control systems structures; in this model, what is known and what is not must be declared and this limits the re-usability of component models.

$$\dot{x} = f(t, x, u) \quad (2.1)$$

$$y = g(t, x, u) \quad (2.2)$$

To understand acausal connections, it is necessary to distinguish two types of variables:

1. across variables (also called potential or effort variables): an example of these variables are temperature, pressure and voltage. The differences between these variables lead to a reaction of the component.
2. through variables (also called flow variables): these normally represent the flow of some conserved quantity like mass, energy, momentum etc.

With this approach it is not necessary to make any *a priori* assumptions, so it is possible to change the physical configuration of the model without deriving the new problem. In Modelica is possible to automatically transform the acausal representation into a block diagram.

The following code 2.3 is a *connector* definition from the Modelica Standard Library:

Code 2.3: Model of a simple *connector* [14].

```

1  within Modelica.Thermal.HeatTransfer;
2  package Interfaces "Connectors and partial models"
3    partial connector HeatPort "Thermal port for 1-dim. heat ...
      transfer"
4      Modelica.SIunits.Temperature T "Port temperature";
5      flow Modelica.SIunits.HeatFlowRate Q_flow
6        "Heat flow rate (positive if flowing from outside into ...
          the component)";
7    end HeatPort;
8
9    connector HeatPort_a "Thermal port for 1-dim. heat transfer ...
      (filled rectangular icon)"
10     extends HeatPort;
11
12     annotation ...
13   end HeatPort_a;
14
15   connector HeatPort_b "Thermal port for 1-dim. heat transfer ...
      (unfilled rectangular icon)"
16     extends HeatPort;
17
18     annotation ...
19   end HeatPort_b;
20 end Interfaces;

```

This connector definition shows the definition of *HeatPort_a* and *HeatPort_b*. Using these connectors allows to model two components that describe the convective heat transfer as is shown in figure 2.3, where *Cap* is a thermal capacitance and *con* is a model for convection. This model includes parameters for the heat transfer coefficient *h*, the surface area *A* and the ambient temperature *T_amb*. The following is Modelica code that refers to figure 2.3:

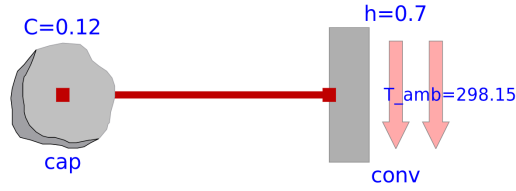


Figure 2.3: Schematic for the *CoolingToAmbient* model

Code 2.4: Model of a simple *connector*[\[14\]](#).

```

1  within ModelicaByExample.Components.HeatTransfer.Examples;
2  model CoolingToAmbient "A model using convection to an ambient ...
   condition"
3
4    ThermalCapacitance cap(C=0.12, To(displayUnit="K") = 363.15)
5    "Thermal capacitance component"
6    annotation ...
7    ConvectionToAmbient conv(h=0.7, A=1.0, T_amb=298.15)
8    "Convection to an ambient temperature"
9    annotation ...
10   equation
11     connect(cap.node, conv.port_a) annotation ...
12   end CoolingToAmbient;

```

Modelica Buildings Library

Modelica allows to use packages which contain models of components belonging to specific physical domains; such packages are called libraries. A library should be reusable and robust, for users it should be difficult to use these components incorrectly.

The library that will be used for the case study [4](#) presented in this work, is Modelica Buildings Library [\[15\]](#), an open source library for modeling building energy and control system. Modelica Buildings library was developed at the lbln of University of California for the following use cases:

- *Support for rapid prototyping:* it is easy to add new components and models or modify the existing ones.
- *Modeling of arbitrary HVAC system topologies:* it is possible to model HVAC systems with non-conventional piping or ducting layout, and compare them with conventional ones.
- *Development, verification and hardware-in-the-loop testing of control algorithms:* users can model and test model-based control algorithms and then verify the control sequences in a hardware-in-the-loop configuration.

- *Emulation of faults at the whole building-system level*: can be utilized with software like LearnHVAC and Learn High Performance Buildings to test fault in equipment or in controls.

The models of Buildings Library are composed by differential, algebraic and discrete equations, as is shown in [15]. The differential equations have one solution and it is unique because the derivatives are bounded on compact sets and the equations are once continuously differentiable with respect to all variables. The equations are differentiated during the translation in order to reduce the index of DAE system if it is higher than one and to compute the Jacobian matrix. Furthermore, the derivative cannot be too high so that the step size of the Newton method is not zero. For these reasons discontinuous functions, continuous but non-differentiable functions and functions with infinite derivatives must be approximated by differentiable functions that have bounded derivatives on compact sets. In order to avoid singularities, in the neighborhood of zero, mass and stationary energy balances are replaced by derivable equations especially for the variables that result from iterative solvers. However, it is impossible to implement this special treatment individually; for this reason the Buildings library is organized in a hierarchical way and at the base there are base classes. The primitive blocks that make up these base classes are the following:

- *ConservationEquation*: This model provides conservation equations for energy, mass fractions and trace substances for components with a varied number of input called ports. There isn't pressure drop between the fluid ports. The conservation equation can be steady state or dynamic.
- *StaticTwoPortConservationEquations*: this block must have two fluid ports with steady-state conservation equations, and it has a small flow rate used in the neighborhood of zero to avoid numerical problems.
- *FixedResistanceDpM*: this model calculates the pressure drop between two ports due to flow friction. There is a parameter that indicates where the turbulent flow begins, by default it is greater than zero and set to 30% of the nominal flow rate; however it can be changed by the user. The mass flow rate is computed for a turbulent flow as $\dot{m} = k\sqrt{\Delta p}$. For the calculation of the pressure drop the density variation is negligible for building HVAC applications, which allows to decouple the mass and energy balance.

Some applications require Modelica models to connect to other solvers or hardware to enable co-simulation or hardware-in-the-loop simulations. An interface that can be used for this purpose is the FMI that it allows to

export simulators or models called Functional Mockup Unit (FMU). The co-simulation in the Modelica environment and in particular through the Buildings library with the Spawn of Energy Plus tool will be discussed later in 2.3.

2.2 Functional Mock-up Interface

The Functional Mock-up Interface is a result of the Information Technology for European Advancement (ITEA2) project MODELISAR; it is an open standard with the purpose of enabling data exchange and co-simulation of models using a combination of XML-files, C-header files, and C-codes in source or binary form. This standard could be divided into two different part [11]:

1. The first part concerns the methods of data exchange, particularly as a simulation environment can generate a C-Code of a dynamic system in form of input/output blocks. This model can be utilized by other simulator independently because it does not use a specific header file. The equations that make up the models are differential, algebraic and discrete equations with time, state and step-events.
2. The second part concerns co-simulation of two or more models in a single simulation environment. The data exchange between sub-systems is done by pairing multiple points in discrete time. In the time between two communication points each solver resolves its own subsystem independently. There is a Master algorithm that controls and synchronizes the data exchange; the information between the data exchange is contained in a XML-file.

Architecture and workflow of FMUs

A simulation model which implements the FMI standard is called FMU (FunctionalMock-up Unit), it is a file zip with extension .fmu that contains the information needed to be used as is shown in [16]:

1. An XML-file contains all variable definition of FMU that are shown in the environment in which the FMU shall be used. For FMI for Co-simulation all information about the "slaves" is located in a specific XML-file.
2. For the FMI for Model Exchange case the model equations are given in form of C-equations in source and/or binary form. Also for the

| | |
|---|---|
| modelDescription.xml | // Description of model (required file) |
| model.png | // Optional image file of model icon |
| documentation | // Optional directory for the model documentation |
| _main.html | // Entry point of the documentation |
| <other documentation files> | |
| sources | // Optional directory containing all C-sources |
| binaries | // Optional directory containing the binaries |
| win32 | // Optional binaries for 32-bit Windows |
| <modelIdentifier>.dll | // DLL of the model interface implementation |
| win64 | // Optional binaries for 64-bit Windows |
| ... | |
| linux32 | // Optional binaries for 32-bit Linux |
| <modelIdentifier>.so | // DLL of the model interface implementation |
| linux64 | // Optional binaries for 64-bit Linux |
| ... | |
| resources | // Optional resources needed by the model |
| < data in model specific files which will be read during initialization > | |

Figure 2.4: Structure of a zip-file of an FMU. [11]

Co-simulation case, the equations are given in source and/or binary form in order to initiate a communication with a simulation tool and to perform the data exchange at communication points.

3. In addition to the functions and the XML-file, there can also be files such as images, tables or documentation that might be relevant for the model

The figure 2.4 describes the structure of an FMU zip file.

Every tool can use its own language to read the XML-file like C, C++, C#, Java, Python etc. and the information contained in this file is the same for both FMI cases with exception of the element “*Implementation*”: the tool importation must chose the C-functions for co-simulation or for data exchange.

Two header files are contained in the executive part of FMU that is called “fmiPlatformTypes.h”. In this file there are the definitions that depend on the target platform as is shown in 2.5.

Code 2.5: Header file. [16].

```

1  #define fmiPlatform "standard32"
2  #define fmiTrue 1
3  #define fmiFalse 0
4  #define fmiUndefinedValueReference
5      (fmiValueReference)(-1)
6  typedef void* fmiComponent;
7  typedef unsigned int fmiValueReference;
8  typedef double fmiReal ;

```

```

9  typedef int          fmiInteger;
10 typedef char         fmiBoolean;
11 typedef const char*   fmiString ;

```

This header file must be used also by the target simulator, so if the target simulator has a different definition, then the FMU must be recompiled with the definitions of target simulator.

An handle for the value of a variable is defined by type *fmiValueReference* while the data exchange is executed by the function *fmiSetXXX(...)* and *fmiGetXXX(...)* where XXX stands for one of the types of variable: Real, Integer, and Boolean. An array of *fmiValueReference* defines which variable is accessed and in the model description XML file the mapping between the FMU variables and this array is stored.

In the header file “fmiFunctions.h” are the functions that can be called from the simulation environment; they are supported in textual and binary representation. The functions present in the model could be the same so the function name is prefixed with the model or slave name.

FMI for Model-Exchange

The objective of the Model-Exchange interface is to numerically solve a system of differential, algebraic and discrete equations. The ordinary differential equations are in state space form and will be abbreviated as "hybrid" ODE. The type of system is piecewise continuous system and the state of "hybrid" ODE is defined by two functions: continuous state $x(t)$ and time-discrete state $m(t)$ as is shown in figure 2.5. This two functions have the following properties:

- $x(t)$ is a vector of real numbers and it is continuous inside each interval, between t_i and t_{i+1}
- $m(t)$ is a set of real, integer, string and logical variables that are constant inside each interval between t_i and t_{i+1} .

As is reported in [16], an event instant t_i occurs as a result of at least one of the following conditions:

1. t_i is defined at the previous event instant t_{i-1} due to a discontinuous change of an input signal. This type of event is called *time event*.
2. an event occurs when the indicator changes its state, for example from $z > 0$ to $z < 0$. Such an event is called *state event*.
3. when the step of an integrator is completed, *fmiCompletedIntegratorStep* must be called. This event is called *step event*.

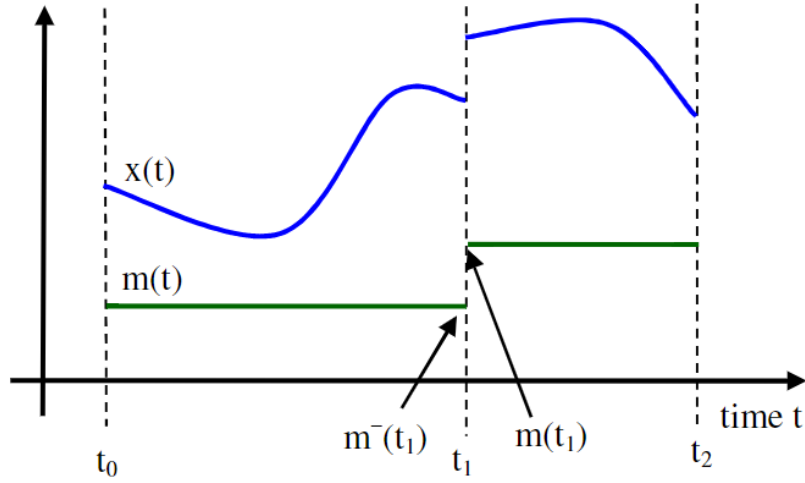


Figure 2.5: Time-continuous (x) and time-discrete (m). [16]

An event is not activated from the inside of FMU but from the environment in which the FMU is called. An FMU model can have different variables:

- \mathbf{p} are parameters that are constant during the whole simulation.
- \mathbf{u} are inputs, signals that are provided from the environment
- \mathbf{y} are outputs that could be treated as inputs by other submodels
- \mathbf{v} are internal variables that are not exchanged between models

As it is explained in [16], the non-real variables change their values only at events.

In order to be efficient, FMI calculates only the variables that are needed in present context, this feature is called "caching of variables"; for this model evaluation can notice when the inputs arguments have changed. This is obtained by setting them with an explicit function that signals every changing variables.

FMI for Co-Simulation

Co-Simulation is a simulation technique in which data exchange takes place only at discrete communication points. Two or more subsystems are solved independently by individual solver and a master algorithm checks the data exchange between subsystems and synchronization of all "slaves". The management carried out by the master is shown in the figure 2.6. Also for FMI for Co-simulation there are functions like in FMI for Model-Exchange, these

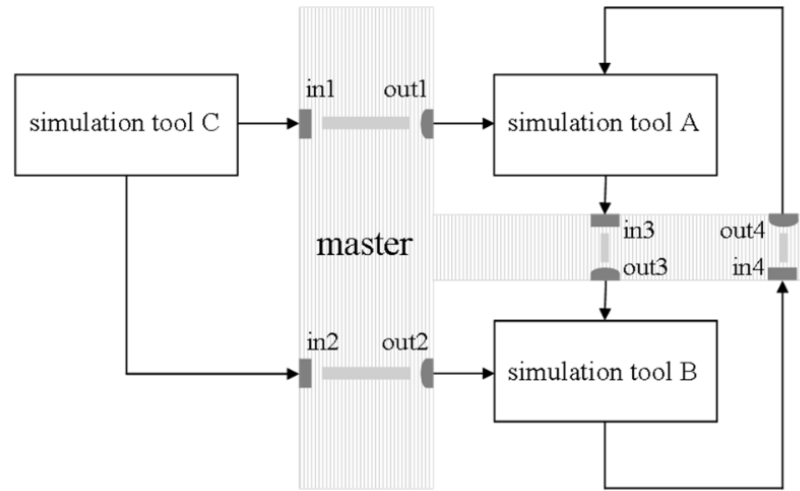


Figure 2.6: Three tools are controlled by one master. [16]

functions are useful for creation, initialization, termination, and destruction of the slaves. To do this there are special functions such as *fmiInstantiateSlave* that provides the string argument *mimeType* that needs to compute an FMU in a tool based co-simulation scenario. The function *fmiDoStep* performs a simulation for a given time step; as reported in [16] the arguments of this function are «*the instance of the current communication time, the size of the communication phase, and a flag indicating whether the previous communication phase was accepted by the master and whether a new communication phase is initiated*».

There are different co-simulation algorithms for step sizes:

- Constant and variable communication step sizes.
- Simple method without a repetition of communication step .
- complex method with with iterative repetition of communication step.

An example of Co-simulation is a connection of Energy plus and an FMU as reported in [11].

The connection between Energy plus and one or more FMUs is done through EnergyPlus module *ExternalInterface* that exchanges data during run-time. The synchronization of the data exchange is checked by EnergyPlus and there is no iteration between Energy Plus and FMU. FMUs generated by external environments are imported into Energy plus and connected to external interfaces; as it is described in figure 2.7 the FMU's output is exchanged with four object and the last one is linked to the FMU. The four object are:

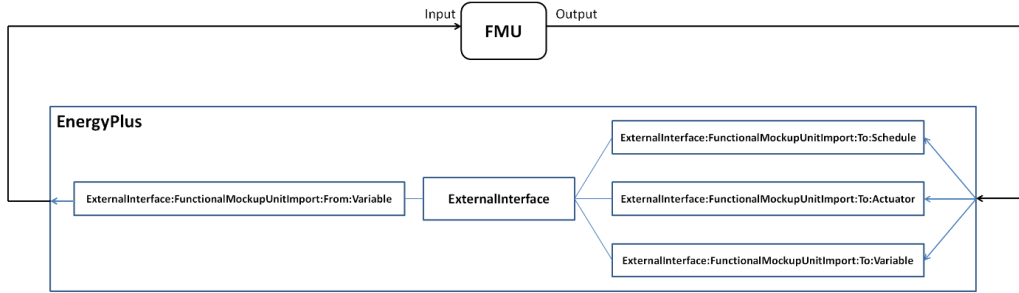


Figure 2.7: Architecture of the FMU import. [11]

- ExternalInterface:FunctionalMockupUnitImport:To:Schedule that is used to over-write schedules
- ExternalInterface:FunctionalMockupUnitImport:To:Actuator that is used in place of Energy Management System (EMS) actuators
- ExternalInterface:FunctionalMockupUnitImport:To:Variable that is used in place of Energy Management System (EMS) variables
- ExternalInterface:FunctionalMockupUnitImport:From:Variable that is used to read Energy Plus objectsOutput

When an FMU is imported there are two steps that will be described below:

- pre-processing
- co-simulation

As reported in [11], there are two main steps for implementing co-simulation between FMUs in Energy plus:

- The *pre-processing step* generates a section in the input file which is used to configure the import of the FMU. The input and output variables of Energy Plus and FMUs are defined in this input file. The information in the XML file of FMUs is extracted and written to a temporary Energy plus input file.
- The *co-simulation step* is used by users to start Energy Plus simulation with the input file. Energy Plus first detects whether the FMU has been imported, then the FMU will be decompressed and executed, finally Energy plus starts co-simulation with FMUs.

Mathematical definitions and terminologies of data exchange for the FMU import

Before the mathematical description, the nomenclature used is illustrated below as described in [11]: differential variables are distinguished from algebraic ones. A variable is differential if its derivatives are in the differential algebraic equations while in the other case we are dealing with algebraic variables. Let $n, m, q, \in \mathbb{N}$ and let $x^1 \in \mathbb{R}^n, x^2 \in \mathbb{R}^m$ and $u \in \mathbb{R}^q$.

- If x^1 and x^2 are differential variables, then the system is $F(\dot{x}^1, x^1, \dot{x}^2, x^2, u, t) = 0$ where function $F: \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^q \times \mathbb{R} \rightarrow \mathbb{R}^{n+m}$
- If x^1 is a differential variable and x^2 is an algebraic variable, then the system is $G(\dot{x}^1, x^1, x^2, u, t) = 0$ where the function $G: \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \times \mathbb{R} \rightarrow \mathbb{R}^{n+m}$
- If x^1 is an algebraic variable and x^2 is a differential variable, then the system is $H(x^1, \dot{x}^2, x^2, u, t) = 0$ where the function $H: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^q \times \mathbb{R} \rightarrow \mathbb{R}^{n+m}$
- If x^1 is an algebraic variable and x^2 is an algebraic variable, then the system is $I(x^1, x^2, u, t) = 0$ where the function $I: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \times \mathbb{R} \rightarrow \mathbb{R}^{n+m}$

The FMU and Energy plus can be linked in four different way as explained below and as summarized in the table 2.1:

- In the first case in both Energy plus and FMU the variables x^1 and x^2 are algebraic.
- In the second case the variable x^1 in Energy plus is differential while the variables x^2 in FMU is algebraic.
- In the third case the variable x^1 in Energy plus is algebraic while the variables x^2 in FMU is differential.
- In the fourth case in both Energy plus and FMU the variables x^1 and x^2 are differential.

An example of the first case can be a co-simulation system in which a wall with a phase changing material (PCM) is simulated with FMU model and a room is modeled on Energy Plus, the two differential variables are the dry bulb ambient temperature in Energy Plus and the wall temperature with PCM in the FMU. Each system solves a differential equation that is connected

Table 2.1: Use cases with different system configurations [11]

| Cases | Energy Plus | FMU |
|-------|-----------------------|-----------------------|
| 1 | Differential variable | Differential variable |
| 2 | Differential variable | Algebraic variable |
| 3 | Algebraic variable | Differential variable |
| 4 | Algebraic variable | Algebraic variable |

to the differential equation of the other system. The equations that describe the systems are 2.3, 2.4.

$$0 = f^1(\dot{x}^1, x^1, x^2), \quad \text{with} \quad x^1(0) = x^{1,0}, \quad (2.3)$$

$$0 = f^2(\dot{x}^2, x^2, x^1), \quad \text{with} \quad x^2(0) = x^{2,0}. \quad (2.4)$$

Where $x^1(0) = x^{1,0}$ and $x^2(0) = x^{2,0}$ are the initial conditions. As explained in [11], the first system computes the sequence in 2.5, for $k \in \{0, \dots, N\}$, where N is the number of time steps, and $\tilde{F}^1 : \mathbb{X}_k^n \times \mathbb{X}_k^m \times \mathbb{R}_{k+1} \rightarrow \mathbb{R}^n$.

$$x^1(t_{k+1}) = \tilde{F}^1(\underline{x}_k^1, \underline{x}_k^2, t_{k+1}) \quad (2.5)$$

and, similarly, the simulator 2 computes for some $\tilde{F}^2 : \mathbb{X}_k^m \times \mathbb{X}_k^n \times \mathbb{R}_{k+1} \rightarrow \mathbb{R}^m$,

$$x^2(t_{k+1}) = \tilde{F}^2(\underline{x}_k^2, \underline{x}_k^1, t_{k+1}) \quad (2.6)$$

The function \tilde{F}^1 and \tilde{F}^2 are used to compute the values of the state variables at the new time instant but each system uses its own time integration algorithm to advance to the next time step that is t_k to t_{k+1} . At the end of the time step, the FMU sends and receives from Energy Plus the new state of variable $x^1(t_{k+1})$. The same is done by Energy plus.

An example of the second case could be a fan modeled in an FMU that is linked to a room model of Energy Plus. Also in this example the differential variable in Energy plus is the room temperature while the algebraic variable in the FMU is the pressure difference of the fan. The equations that describe the system are the 2.7 and 2.8.

$$0 = g^1(\dot{x}^1, x^1, x^2), \quad \text{with} \quad x^1(0) = x^{1,0}, \quad (2.7)$$

$$0 = g^2(x^2, x^1). \quad (2.8)$$

As in the previously discussed example,, the system computes for $k \in \{0, \dots, N\}$, and some $\tilde{G}^1 : \mathbb{X}_k^n \times \mathbb{X}_k^m \times \mathbb{R}_{k+1} \rightarrow \mathbb{R}^n$ the sequence

$$x^1(t_{k+1}) = \tilde{G}^1(\underline{x}_k^1, \underline{x}_k^2, t_{k+1}) \quad (2.9)$$

and the simulator 2 computes for $\tilde{G}^2 : \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^m$ the sequence

$$x^2(t_{k+1}) = \tilde{G}^2(x_k^2, x_k^1, t_{k+1}) \quad (2.10)$$

As in the first case, the function \tilde{G}^1 and \tilde{G}^2 are used to compute the values of the state variables at the new time instant but each system uses its own time integration algorithm to advance to the next time step. At the end of time step Energy plus sends and receives the the new state of variable from FMU.

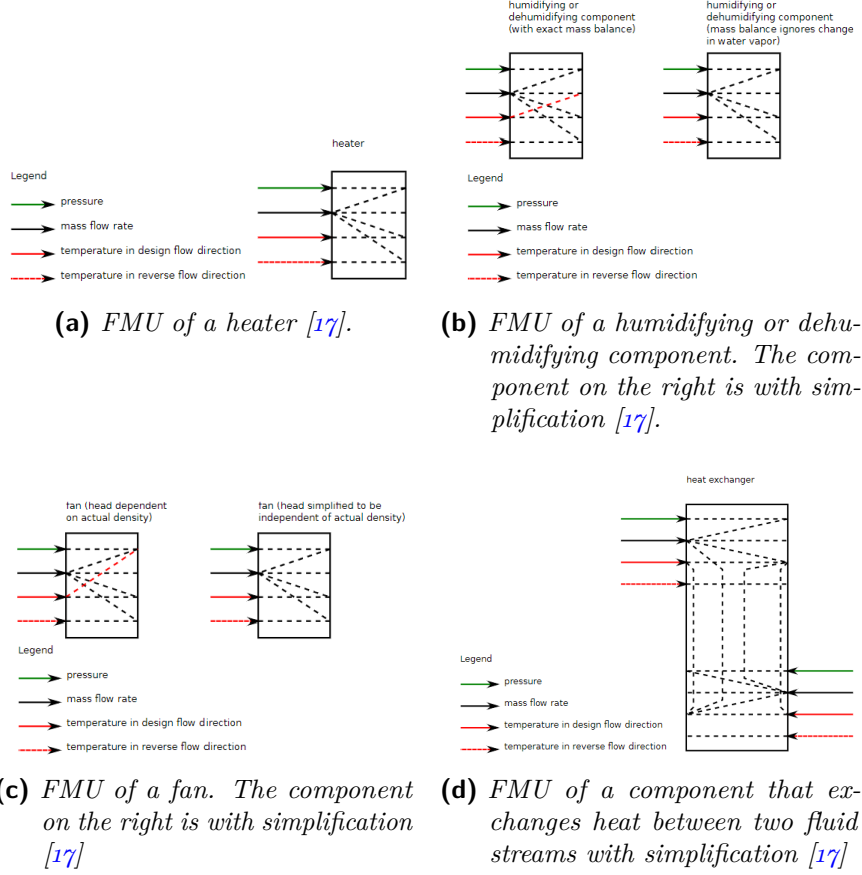
Design choices for thermofluid flow in systems that are exported as an FMU

In this sub-section input and output variables and their mutual dependencies will be described , for HVAC components modeled in state state condition;each component is described graphically as shown in the figure 2.8, where the arrows indicate inputs and the dashed arrows inside indicate the dependence between the various inputs . The HVAC component that will be discussed, as shown in [17], are:

- Heater
- Dehumidifying or humidifying components
- Fan
- Heat exchanger between supply and return air

Heater. The first is a component that adds a prescribed heat power into the fluid stream. The outlet pressure and temperature are respectively $p_b = p_a + f(\dot{m}_a)$ and $T_b = g(\dot{m}_a, \iota(T_a))$ for some function $f : \mathbb{R} \rightarrow \mathbb{R}$ and $g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. The component is shown in 2.8a

Dehumidifying or humidifying components. This component is used to humidify or dehumidify air, for example by using a humidifier or a cooling coil that cools the air below the dew point. The outlet mass flow rate depends on the rate of heat and inlet mass, so the pressure drop equations are coupled to the heat transfer equations. As is shown in figure 2.8b the output also depends on the reverse fluid but there is no need to specify the thermodynamic behavior of the reverse fluid because the component is activated only when the fan is on. When the fan is off it is possible that there is a small reverse flow; if this happens, a default temperature and mass fraction must be set: $T_a = \iota(T_b)$ and $X_a = \iota(X_b)$. The outlet flow rate is $\dot{m}_b = \dot{m}_a(1 + \Delta X_w)$

**Figure 2.8:** FMU of the components of HVAC

where ΔX_w is the fraction of water that changes state into water vapor, Since in HVAC systems ΔX_w is typically less than 0.005, tolerating an error of the same order of magnitude it is possible to decouple the equations of the energy balance and pressure drop; with this simplification it is possible to avoid calculation the upstream energy balance when resolving the pressure drop of the downstream components.

Fan. this component models the pressure rise over a fan related to the volume flow rate: $p_b - p_a = f(\dot{m}_a, \iota(T_a) \iota(X_a))$. However, by using a constant mass density for the inlet and outlet it is possible to eliminate the dependence of the thermodynamic properties between inlet and outlet as it is shown in figure 2.8c.

Heat exchanger between supply and return air. figure 2.8d shows the heat exchanger along with the inlets of the two air streams, one at the top left and the other one at the bottom right. This component is typically modeled in two ways:

1. The former is the simplest way: a constant efficiency heat exchange is calculated, so that the rate of heat transfer is $\dot{Q} = \varepsilon \dot{C}_{\min} (T_{\text{in},1} - T_{\text{in},2})$ where $\varepsilon \in (0, 1)$ is a constant, $\dot{C}_{\min} = \min(|\dot{m}_{a,1}c_{p,1}|, |\dot{m}_{a,2}c_{p,2}|)$, $T_{\text{in},1}$ is the temperature of the first inlet fluid and $T_{\text{in},2}$ is the temperature of second inlet fluid. The thermodynamic properties of the two incoming flows must be known to calculate the outgoing flows in both directions.
2. The second is a more complex method that uses the $\varepsilon - \text{NTU}$ method; still the dependence between input and output is obtained in this situation in the same way as in the first method.

Also in this case the component can be simplified by assuming that the flow direction is the one defined in design conditions as shown in the figure 2.8. This simplification makes physical sense because the flow could only be reversed when the HVAC system is off.

2.3 Spawn of Energy Plus

2.3.1 Historical overview

Co-simulation has been identified as an excellent and above all flexible approach, suitable for the description of dynamic systems with different physical domains. Each domain is simulated with a specific solver: this approach leads to high efficiency and accuracy of even a very complex dynamic system.

The idea of using simulation for HVAC systems dates back to 1997 when the participants of the IEA-ECBCS Annex 17 (IEA 1997) built emulators using TRNSYS and HVACSIM Plus as reported in [18]. Later, new dynamic component model libraries and emulators were born, the first being SIMBAD (SIMulator of Building And Devices) [19].

SIMBAD was developed based on TRNSYS and HVACSIM Plus by the Centre Scientifique et Technique du Bâtiment to test Building Energy Management Systems (BEMS); SIMBAD was a toolbox of Matlab/SIMULINK and was implemented in the bloc diagram environment and contained a library of VAV component. It had a graphical interface from which the users can drag the component of HVAC in the model.

A second component library was Virtual Cybernetic Building Testbed (VCBT),

then the development of EnergyPlus and Modelica Buildings Library led to Building Controls Virtual Testbed (BCVTB) [8], an open source software available from Lawrence Berkeley National Laboratory; with this program, users can simulate different energy systems and exchange data between them. A recent simulation framework is a software platform for the testing of advanced control strategies called BOPTEST (Building Optimization Performance Test) [18]. The goal of BOPTEST is simulating buildings and benchmarking control strategies and since the critical aspect is to obtain a realistic simulation of the building. As is written in [4] «the underlying simulation technology of the building emulator is critical to enabling evaluation with realistic test cases» and the aspects that characterize a good simulation are «the fidelity of the model, time-resolution of the simulation, and representation of both current and future system designs». Energy plus and TRANSYS can model the building envelope in a very detailed and precise way but the HVAC model, despite having a wide variety of components, has a calculation procedure that is based on the control of inputs and outputs that differ from the actual control signals suitable for actuators commonly found in building energy systems. Moreover, in Energy Plus there is no relationship between the pressure and the air flow inside the HVAC ducts; these modeling approach limits the possibility of having a timestep with a lower limit of only one minute and so it does not allow to represent the real control logic that exists in a building.

A solution to these problems could be using Modelica which has the ability to symbolically manipulate the equations of the model in order to generate computationally efficient code for the energy systems of buildings. Furthermore, Modelica has variable timesteps capable of handling nonlinear differential algebraic equation systems, which can be continuous and discrete, this happens when building, HVAC and control models are coupled.

The final solution, which will be explored in this section, is the union of Modelica, with the Buildigs library, and Energy Plus through FMI: **Spawn of Energy Plus** [20].

2.3.2 **Spawn overview**

Spawn is developed by the US Department of Energy, National Labs and industry and it is the new whole-building energy simulation engine. The purpose of spawn is not to replace Energy Plus but to integrate it with HVAC and controls models in order to obtain a realistic behavior of buildings and the possibility to prove innovative control strategies and new use cases in HVAC application.

One of the advantages of Spawn is the use of open standards: the Modelica

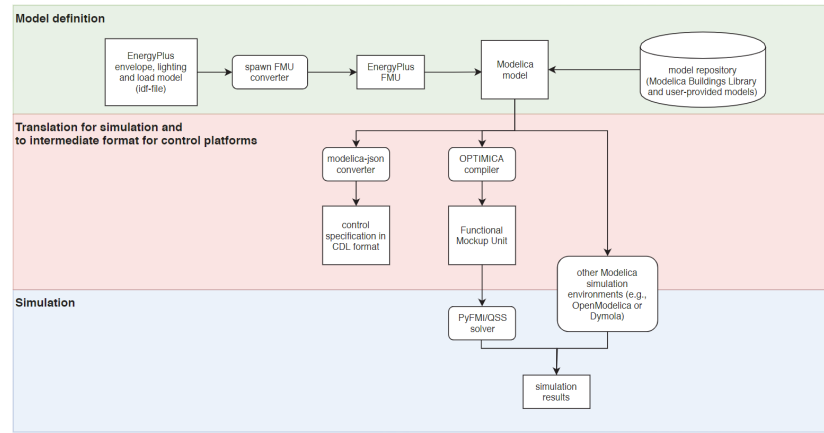


Figure 2.9: Architectural diagram of Spawn. [20]

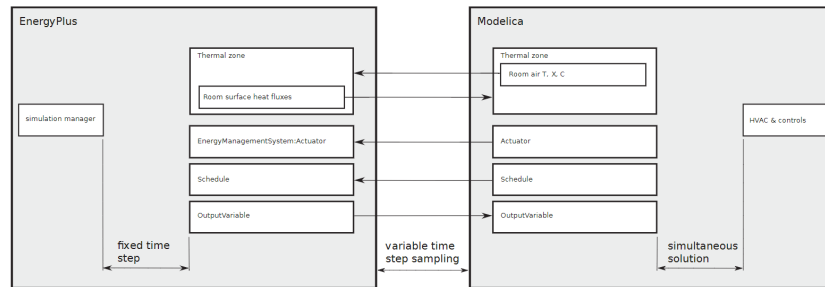


Figure 2.10: Partitioning of the envelope, room and HVAC model. [20]

language [12] and the Functional Mockup Interface (FMI) for co-simulation and model exchange [16]. Spawn pairs the envelope and internal loads models with the HVAC and control models in Modelica using an FMI interface; the models in Modelica are compiled by Spawn using Modelon’s OPTIMICA Compiler Toolkit. Figure 2.9 shows the Spawn software architecture.

2.3.3 Simulation framework

Spawn, as is written in [20], utilizes Energy Plus to model «*both internal loads and envelope heat and light transfer, including electric lighting, shading, daylight transport, solar heat gains, groundheat transfer, surface conduction, surface-to-surface radiation, and convective heat flow from interior surfaces to room air*». HVAC operation and control, room-air mass and heat balance are simulated in Modelica and the exchange data is shown in figure 2.10. The partitioning of envelope, room and HVAC models is due to three reasons:

1. the dynamics of the air temperature is faster than the conductive heat in

the building envelope, this allows Spawn to use Modelica with variable timestep solvers in order to reduce the number of synchronization steps between Energy Plus and Modelica.

2. because the airflow in HVAC and supply in the room and the infiltration are closely related by simulating these together it is possible to obtain better results.
3. this separation allows to substitute in the future the present 1D model of room air with a 3D model already present in the Buildings library.

To couple Energy Plus and Modelica, some modifications must be made to Energy Plus. Data exchange could be done at each timestep on Energy plus with External Interface but Spawn requires a more granular access; to enable this granular access, Energy plus was refactored and C++ API was added that provides functions to advance simulation in time to set input and read output, bypassing HVAC on Energy Plus in order to use it as a heat balance and daylight calculation engine. At startup Modelica calls a script that acts as an interface that reads the IDF file and creates an FMU that can dynamically link to and execute Energy Plus.

Modelica Spawn models are translated and simulated with OPTIMICA Compiler Toolkit, as Modelica/FMI based simulation platform. OPTIMICA can translate differential algebraic equation systems into an ordinary differential equation system in order to fit better to FMI.

One of the objectives of spawn is to quantify the energy savings achieved through control logic such as those described in Guideline 36 as explained in [21]; to do this, there is a challenge that must be addressed.

The challenge is to jointly simulate the pressure-flow relationship in VAV distribution systems, HVAC equipment and control sequences. The importance of modelling the ventilation duct and the pressure-flow relationship stems from the relationship between the supply fan speed and the damper position in the zone terminal; this relationship is crucial for VAV systems and the associated energy savings. However, traditional modelling tools such as EnergyPlus have major limitations for the simulation of the ducts of an HVAC system and the control logic sequences associated with them.

Figure 2.11 shows a simulation framework in which the building envelope is modelled on EnergyPlus, the HVAC system and control logic are modelled on Modelica, and the Internal Gain can be calculated both on EnergyPlus and Modelica; Spawn includes the merging of Modelica and EnergyPlus done by FMI. Climate files are passed to EnergyPlus and Modelica in the .epw and .mos format respectively. The .csv files contain the schedules required for calculating internal gain.

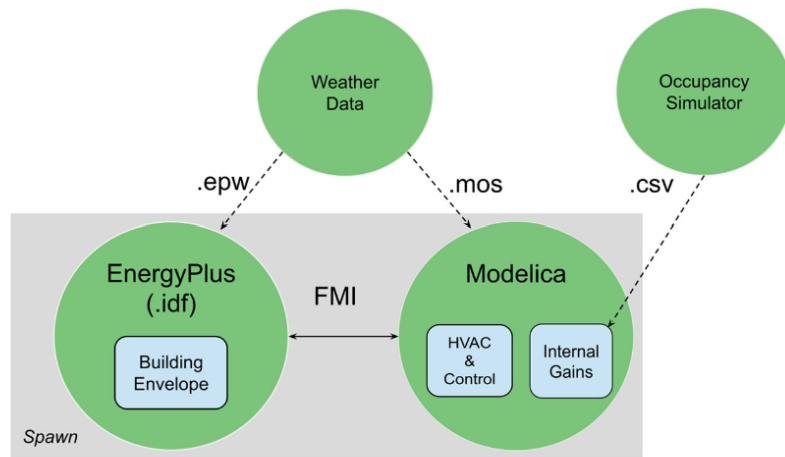


Figure 2.11: *Simulation framework using Spawn.* [21]

CHAPTER 3

Control strategies overview

Buildings are key elements for the Union Europe's energy efficiency policies, accounting for around 40% of final energy consumption [22]. The challenge of the future is to try to couple the non-programmable energy of renewable sources and the energy demand of buildings. To do this buildings will have to have an higher flexibility and sustainability and will have to be able to manage and control the energy generation/consumption/storage systems in the buildings.

A BACS is defined as a Building Automation and Control System, it is a system that ensures the thermal/visual comfort of users with energy efficient and cost saving solutions, controlling mechanical and electrical equipment present in the building.

As is reported in [23] BACS tries to achieve the following objectives:

- thermal/visual comfort: it is the feeling of well-being that the human body feels inside the building in relation to the air temperature, air quality, air speed, humidity etc.
- energy savings, that is the reducing of energy consumption without compromising thermal and visual comfort.
- cost saving, it's obtained with increasing of energy efficient and managing of renewable sources and energy exchanges with the electricity grid without compromising thermal/visual comfort.
- optimal interaction with the external environment, the environment could be national or district grid and the interaction could avoid power pick.

A BACS is based on several technological elements that are described below:

- sensor are pieces of equipment used to measure physical quantities such as temperature, humidity, mass flow rate, CO₂ etc. The measurement is then converted into a digital signal.
- CPUs receive digital measures as input and through the control strategy it outputs instructions.
- actuators are used to activate the control strategies given by CPU.
- information and communication technologies (ICT) are tools that connect the technological elements described above.

A suggested classification of control strategies can be seen in figure 3.1 in which the strategies are divided in two major groups: traditional control strategies (TCS) and advanced control strategies (ACS). The first group can

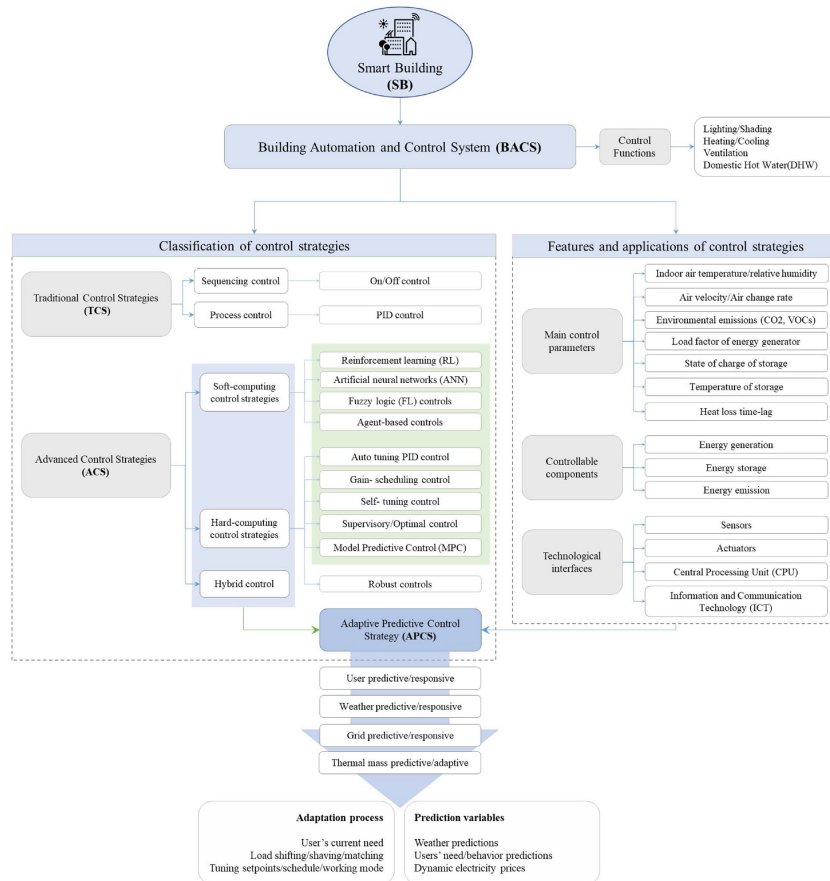


Figure 3.1: Features, applications and functions of BACS in buildings and HVAC systems. [23]

be in turn classified into two different way:

- sequencing control, also called rule based control because the control strategies are based on *if-then* conditions sequentially.
- process control, measuring disturbance variables, so that the control variables are adjusted to achieve process object.

The second one can be in turn classified into three different way:

- soft-computing control strategies, which include fuzzy logic (FL), reinforcement learning (RL), deep learning based on artificial neural network (ANN) controls. They do well with inaccurate inputs and provide approximate/statistical responses.
- hard-computing control strategies, which include auto-tuning PID control, self- tuning control, gain-scheduling control and model predictive control (MPC). They use an analytical/mathematical model so they need real and precise input data to give an accurate response.
- hybrid control strategies that combine hard and soft computing strategies

3.1 Traditional control strategies (TCS)

The main logic used in sequencing control is ON/OFF control, while in process control proportional, integral, and derivative (PID) controls is prevalent.

3.1.1 ON/OFF control

The ON/OFF controllers are also known as *discontinuous controllers*. These controllers have a discrete output and because of this simplification there will be a certain range in which the oscillation of system output will not influence the output signal. This results into a less precise of the controlled variable. The main discontinuous controller is the *two position controller* that has only two signal outputs: ON corresponds to the maximum value and OFF corresponds to the minimum one. The analytic representation of *two position controller* is shown in 3.1.

$$y(t) = \begin{cases} y_{\max} & e < 0 \\ y_{\min} & e \geq 0 \end{cases} \quad (3.1)$$

Where $y(t)$ represents the control signal and e is the control error. In figure 3.2 there is a representation of ON/OFF control, where y is the action

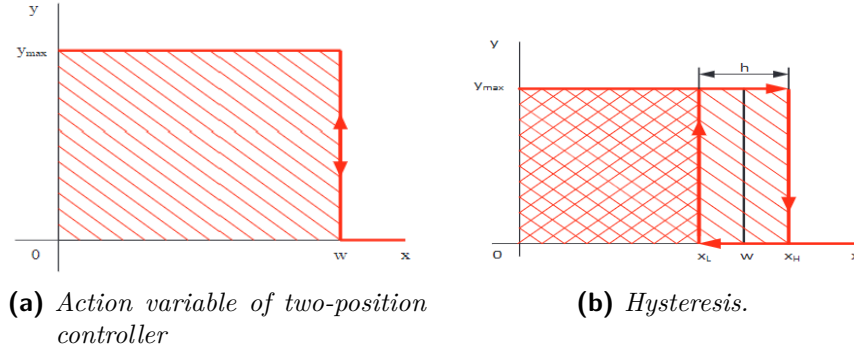


Figure 3.2: ON-OFF control. [24]

variable and x is the process variable and w is the setpoint. Error e is nonzero whenever the value of x is different from setpoint w , in this case the action variable is activated: when $x > w$ then $y = y_{\min}$ so the status is OFF, when $x \leq w$ then $y = y_{\max}$ so the status is ON. In the neighborhood of the setpoint, this control could lead to a high frequency oscillation of the output signal so to prevent this behavior, *hysteresis* is introduced. The output signal changes from y_{\max} to y_{\min} when $x = x_H = w + \frac{h}{2}$, where h is the hysteresis value, and switches back to y_{\max} when $x = x_L = w - \frac{h}{2}$.

In table 3.1 the advantages and disadvantages of On/OFF control strategies are summarized.

Rule based control

Rule-based control is a type of sequencing control based on if-then rules; the if-then rules can be represented with flow charts in which each condition is dichotomous and at the end of all the conditions lead to an action to execute. The use of if-then rules in the field of building control can lead to an action of switching on and off of a generation system or opening and closing of a valve or a damper. Simple operation, ease of implementation and the possibility of implementing rules based on experience allow this type of control to still be used on a large scale today.

According to [44]: «about 3.9 million Heat Pumps were sold in 2019, which amounts to 12.6% annual growth». In a scenario where the European energy system relies 100% on renewables as described in [43], good management of heat pumps can reduce peak electrical and thermal demand.

Many buildings are equipped with a heat pump to meet heating and cooling needs with the help of a thermal storage tank; in addition, the photovoltaic heat pump coupling allows the electrical load of the heat pump to be met

Table 3.1: Advantages and disadvantages of ON/OFF control techniques.[25]

| Control Method | Advantage | Disadvantage |
|----------------|---|--|
| ON/OFF control | | (1) It is unable to control moving process with time delays. |
| | | (2) It is not accurate enough. |
| | (1) It is the most easiest and intuitive method to implement. | (3) It does not have enough quality. |
| | (2) It is a basic method and very simple. | (4) PID controllers should be used for reducing the fluctuation produced by on/off controller. But parameter's setting for the PID controllers is difficult. |
| | (3) Low initial cost. | (5) Accepts only binary inputs. |
| | (4) Plain structure. | (6) Due to inability to set point tracking accurately, this method is inefficient. |
| | (5) Quick response. | (7) Not useful and effective in the long run. |
| | (6) Feedback controller. | |

with renewable energy. The system just described allows the building to be more flexible because the stored energy can be used when needed in order to lower the peak energy demand; the decision on when to store electrical or thermal energy can be made on the basis of experience with a flow chart.

In [26] an example is shown of a flowchart of a control algorithm that aims at switching on the heat pump only when the hourly price is lower than a price chosen as a limit.

If the hourly energy price (HEP) is lower than or equal to the limit price (LP), then the heat pump is on and the zone set point is normal and the tank set point is maximum; if HEP is higher than the LP, the heating system is turned off if the indoor and storage tank temperatures are at the acceptable level; if, on the other hand, the temperature levels are not acceptable, the heat pump is switched on and the temperatures raised to an acceptable level. The flow chart is shown in figure 3.3. In a predictive rule-based control, rules

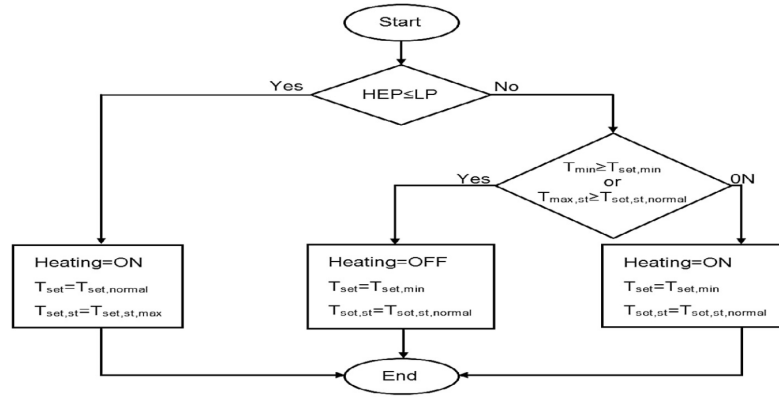


Figure 3.3: *Momentary control algorithm.* [26]

are designed on the basis of future trends of a variable. In [27], an example of a predictive rule based control is reported. The control monitors current indoor and storage tank temperatures and defines indoor temperature set points, domestic hot water tank and heating tank temperature set points based on future HEP trends:

- $CS = +1$ means that the trend of future HEPs is rising.
- $CS = -1$ means that the trend of future HEPs is falling.
- $CS = +1$ means that the trend of future HEPs is constant.

If $CS = +1$ the indoor zone temperature or the average outdoor temperature of the previous 24 hours are checked and if they are respectively lower than

the maximum set-point or a limit value then the heat pump is switched on and the zone, tank and storage tank temperatures are set to maximum values; If the zone temperature check and the average outside temperature over the previous 24 hours are not met, the zone set point is normal and the two tank set points are maximum.

If $CS = -1$ a check is made on the zone temperature and both tank temperatures; if one of these temperatures is greater than the minimum value then the heat pump is switched off. On the other hand, if all temperatures are lower than the minimum value, the heat pump is switched on and the temperatures of the internal set point and both tanks are set to the minimum value.

If $CS = 0$ the heat pump is switched on and the set points of zone and tank temperatures are normal. The control just described is shown in the figure 3.4.

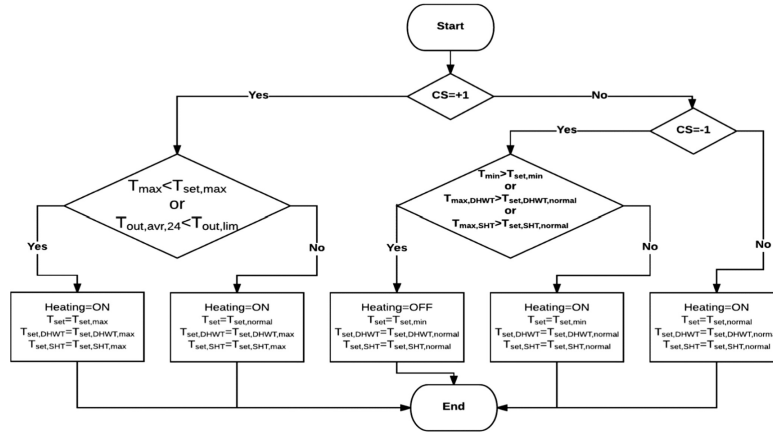


Figure 3.4: The flowchart of predictive rule-based control algorithms [27]

3.1.2 PID control

The proportional, integrative and derivative controller (also called PID) is a *continuous controller*, meaning that it can output any value within a given interval as a control signal. In a feedback control system, the controller reacts on the basis of the response it receives from the system. The simplest feedback control is *proportional control* in which the feedback control signal $u(t)$ is proportional to the error as is shown in formula 3.2, where K_P is proportional gain and $e(t)$ is the feedback error calculated as the difference between the reference signal $r(t)$ and the output signal $y(t)$.

$$u(t) = K_P e(t) \quad (3.2)$$

In figure 3.5 the block diagram for the closed-loop feedback control configuration is shown and it is possible to see Reference signal $R(s)$, feedback error $E(s)$, control signal $U(s)$, output signal $Y(s)$ and the Laplace transfer function of the plant $G(s)$. Using only proportional control it is not possible

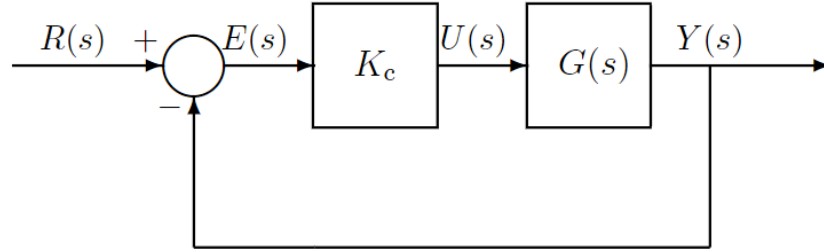


Figure 3.5: Proportional feedback control system. [28]

to eliminate the error completely and the system will settle on a value that will present an error with respect to the reference value. To try to eliminate the error, an additional action is introduced and such a logic control is called proportional-integral control (PI). The integral term is proportional to the integral of the error over time and the control signal of a PI controller is computed as shown in figure 3.3.

$$U(s) = K_P e(t) + K_I \int_0^t e(\tau) d\tau \quad (3.3)$$

The constant K_I is integral gain and can be expressed as $K_I = \frac{K_P}{\tau_I}$ where τ_I is called *integral time constant*.

The resulting Laplace transform of the PI controller is shown in the formula 3.4 with associated figure 3.6.

$$u(t) = K_P E(s) + K_I E(s) \quad (3.4)$$

Proportional-integral control has the limitation that the control signal profile can be at times very steep, which could be dangerous for the equipment of the controlled plant. This problem can be solved introducing a derivative term, which is proportional to the time derivative of the feedback error and the control signal of a PD controller is computed as shown in figure 3.5.

$$u(t) = K_P e(t) + K_D \frac{de(t)}{dt} \quad (3.5)$$

Constant K_D is called derivative gain and is often represented as $K_D = K_P \tau_D$ where τ_D is called *derivative time constant*.

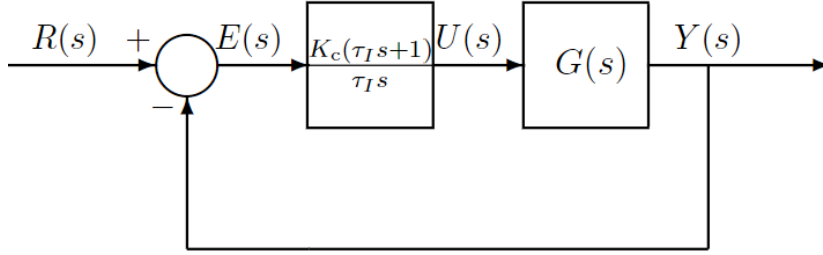


Figure 3.6: Closed loop PI control. [28]

The resulting Laplace transform of the PD controller is in formula 3.6 with associated figure 3.7.

$$U(s) = K_P E(s) + K_D sE(s) \quad (3.6)$$

All three terms can be summed up to obtain the proportional-integral-

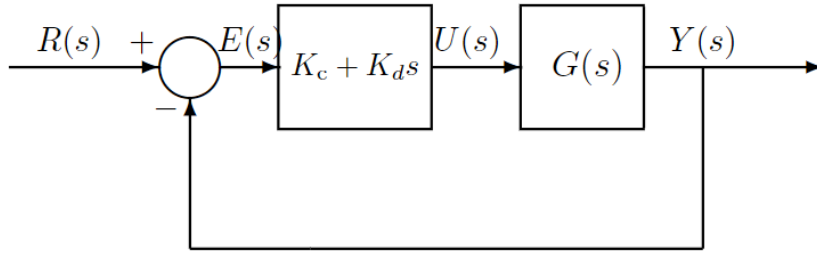


Figure 3.7: Closed loop PD control. [28]

derivative control (PID); in table 3.2 the advantages and disadvantages of PID control strategies are summarized.

3.2 Advanced control strategies (ACS)

Advanced control strategies represent the evolution of traditional control; they have the ability to predict both the behavior of the building envelope and the behavior of the system components; this capability makes it possible to calculate actions suitable for the forecast boundary conditions that respect the control objectives defined previously. This control uses the measured data of energy consumption and occupant behavior as input to bring the model closer to the needs of the occupants. ACSs are mainly used for buildings in

Table 3.2: Advantages and disadvantages of PID control techniques.[25]

| Control Method | Advantage | Disadvantage |
|----------------|--|---|
| PID control | (1) The most commonly applied controller in the industry. | (1) Producing undershoot and overshoot which makes the sudden force to the actuator and causes the depreciation of the actuators in long time. (2) Involve with the system's mathematical model. |
| | (2) This control techniques have been a dominant area of research and applications to the field of HVAC. | (3) Not proper for non-linear and complex systems or uncertain information systems. |
| | (3) Feedback controller. | (4) The long testing time is weak point of this method. |
| | (4) Derivative term combat with sudden load changes in the system. | (5) Performance is limited. (6) Tuning the controller parameters is awkward. (7) Inadequate gains selection for p PID controllers can result to the instability of the whole system. |

the tertiary sector because they have a predefined occupancy schedule and therefore less uncertainty for the control system.

Although the usefulness of this type of control is undoubted, as is reported in [29] «developing a high-fidelity forecasting model for building systems is not an easy task» because «most of the building energy systems are complex-nonlinear systems, which are strongly influenced by weather conditions, building operating modes, and occupant schedules». Furthermore, the measurement of fundamental quantities for the control requires widespread sensors, which is not yet the case in the tertiary sector.

HVAC systems are dynamic systems and therefore need transient models to be described; the same consideration must be made for the forecasting models of the disturbing variables. As reported in [29], a further classification is made on the basis of the time scale of the forecast:

- long-term load forecasts for system planning.
- medium-term forecasts for system maintenance.
- short-term modelling for daily operation, scheduling and load shifting plans

Building control is done with short-term models and is applied with three different approaches:

1. white box models are based on a physical and mathematical approach.
2. black box models are data-driven, based on an empirical approach.
3. grey box models which combine the first two approaches.

3.2.1 Reinforcement learning

Reinforcement Learning is a branch of machine learning, , together with supervised and unsupervised learning but RL returns a score in addition to the output; the score refers to the output itself.

As described in [30], to understand the application of RL, it is necessary to describe the environment as Markov Decision Processes (MDPs). The MDP is a classical formalisation of sequential decision-making that aims at learning interactions to achieve objectives. The agent interacts with the environment by taking actions, then the environment provides feedback on the agent's action with new states.

An MDP can be defined as a tuple $\langle S, A, T, R, \gamma \rangle$:

- S is the collection of environmental states.

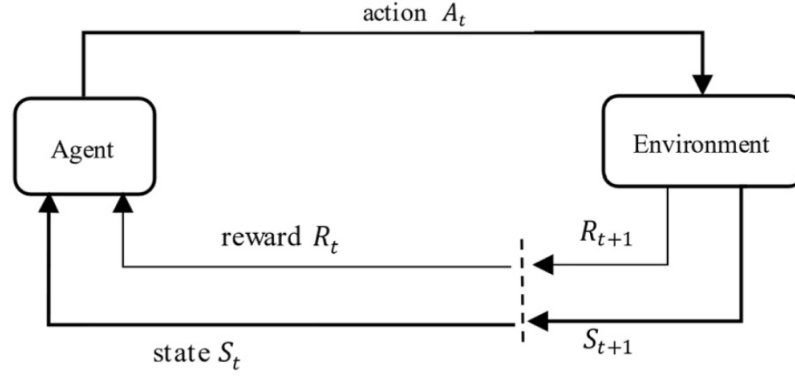


Figure 3.8: *interaction between agent and environment in a Markov decision process.* [30]

- A is the collection of reward that the agent receives from the environment.
- T is the probability that the agent take an action $a \in A$ to move the next state.
- R is the reward the agent receives when passing from one state to another.
- γ is a discount factor.

The agent performs an action at each time step and receives both the reward and information about its state from the environment.

RL, applied to the control of a building, is trained according to a trial and error approach in order to test different trajectories in a dynamic environment (e.g. changing weather conditions) and evaluate their performance with the aim of maximising a certain amount of rewards.

3.2.2 Model predictive control (MPC) strategies

Model predictive control was first used in process industry in the 1970s, to be later adopted by other industrial fields; recently, as pointed out in [23], «the application of MPC strategies in BACS has received significant attention from the research community». Because new buildings incorporate renewable energy generation, new heat generation technologies such as heat pumps and new storage systems, an advanced control system with forecasting capabilities such as an MPC is required, in order to exploit the dynamic behavior of the

building. It has been shown that better energy and cost saving results are obtained with buildings with high thermal capacity [31].

MPC is an optimal control to manage systems that have time delays of the variables that affect the system itself as HVAC systems in buildings. Indeed, MPC controls evaluate future information of predicted variables such as weather forecast, building load, occupancy and energy prices in order to implement a control action in the present aware of the effects it will have in the future. The control strategy is described as the optimal sequence of control actions; this strategy is calculated on the control horizon and of all the calculated steps only the first action will be effectively implemented until the next sample instant; all the other steps of the calculated sequence will be discarded. This strategy is called *receding horizon* strategy and is shown in 3.9.

The quality of the predicted variables depends on the quality of the input

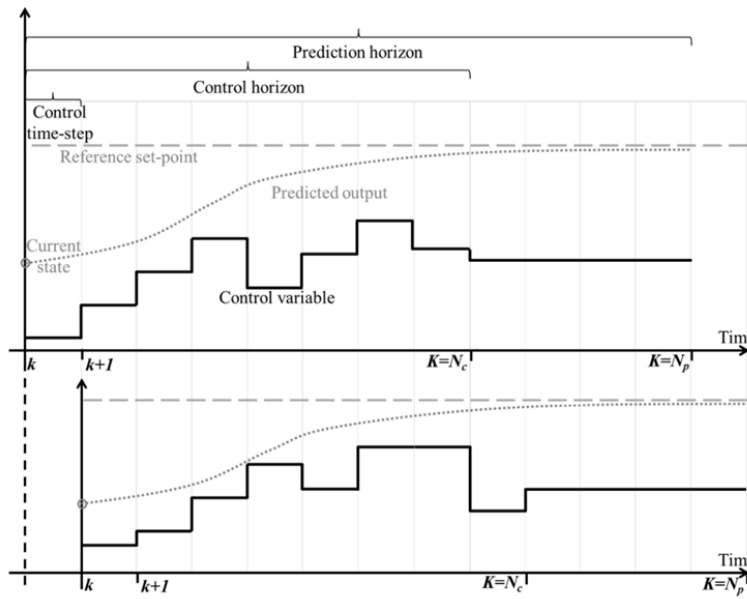


Figure 3.9: MPC receding horizon strategy. [32]

data and therefore on the quality and reliability of the sensors and on the place where they are installed.

As described in [23], the different functions of MPC in buildings can be summarized as follows:

- Weather predictive/responsive: that is the ability to predict weather conditions and implement active and passive actions with the aim

of minimizing the amount of energy injected or withdrawn from the national grid.

- User predictive/responsive, that is the ability to predict the occupation of users and their interactions with the technological equipment so that the internal loads are also foreseen, guaranteeing thermal comfort in the periods in which the zones are occupied.
- Grid predictive/responsive, that is the the ability to react to future information from the grid in order to reduce the peak of the electrical load, taking advantage of the moments when the price of energy is minimal to load the thermal storage.
- Thermal mass predictability/adaptability that is the ability to exploit the thermal mass of the building in order to modulate the energy generation, consumption and storage of the system and shift energy demands.

In table 3.3 the advantages and disadvantages of MPC control strategies are summarized.

Table 3.3: Advantages and disadvantages of MPC control techniques.[23]

| Control Method | Advantage | Disadvantage |
|----------------|--|--|
| MPC control | (1) Energy/Cost effective. | |
| | (2) Handle processes with time delays. | |
| | (3) High computational power. | |
| | (4) Disturbance robustness. | |
| | (5) Peak load shifting/shaving/ matching. | (1) Need improvement of mathematical/analytical model of the building. |
| | (6) Better regulation of input parameters. | (2) Expensive installation. |
| | (7) Control of multiple variables. | (3) Incapable of handling uncertain disturbances. |
| | (8) Future disturbances/control actions prediction. | |
| | (9) Exploits an accurate building dynamic model. | |
| | (10) Capable to integrate with thermal mass building | |

CHAPTER 4

Case study

4.1 Case study description and simulation setup

The previous chapters describe the tools and software that were used for the co-simulation of the model treated in this case study.

The case study concerns the co-simulation of an office building and related HVAC, electrical and thermal generation systems. The building is modeled on Energy Plus 9.6.0 while the HVAC and related heat generation system is modeled with the modelica language on OPENMODELICA, «*an open source Modelica-based modeling and simulation environment intended for industrial and academic usage*» [33]. Co-simulation between the Modelica environment and Energy Plus is possible thanks to Spawn contained in the latest version of the Modelica Buildings Library [15]. Data exchange between Energy Plus and OpenModelica occurs as explained in section 2.3 through the FMI standard 2.2. The goal is to simulate the building and HVAC system behavior in detail, implementing an high and low level rule based control that will be described below in 4.5.

In figure 4.1 the functional diagram of the system is shown: on the thermal side, a heat pump heats water and recharges a thermal tank; the hot water heats primary air in the air handling unit (AHU), which balances the building's internal load. On the electrical side, energy is provided by photovoltaic panels charging the electrical storage system or balancing the load directly, or by the national grid as will be explained in 4.3.2.

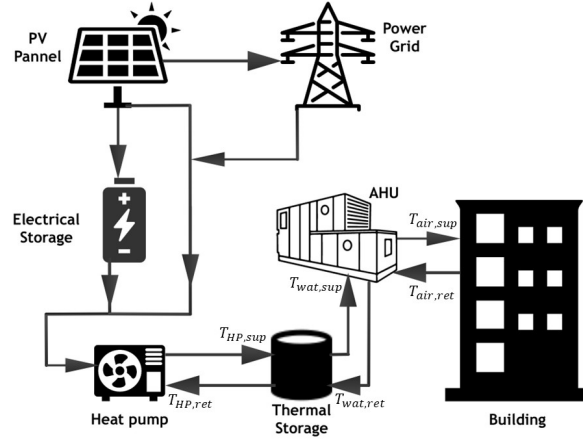


Figure 4.1: *Functional diagram of the system.*

4.2 Building description

The building chosen for the case study was taken from Commercial Reference Buildings [34] developed by the U.S. Department of Energy together with three of its national laboratories. Commercial Reference Buildings are 16 building types that represent 70% of the commercial buildings in the U.S.. Thus, the building has plausible geometric and envelope characteristics and can be used as a reference for this analysis.

The building is a simple office with five conditioned zones as shown in Energy Plus model in figure 4.2.

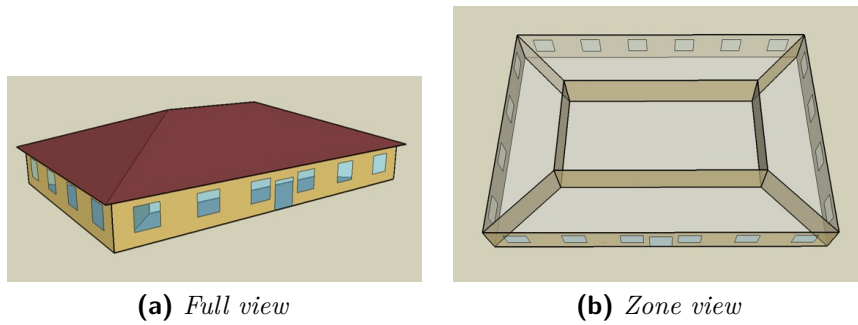


Figure 4.2: Simple office model [34].

In the model developed in Modelica, the five zones of the building are shown in the figure 4.3 (blue color blocks). These blocks, called *zon*, are the Modelica models of the thermal zones that are connected to the corresponding

thermal zones in Energy Plus FMUs; the block has air temperature T_{air} as its output and through the fluid ports connects the zone to the HVAC. Also shown in figure 4.3 is the building block in which the .IDF and climate file are specified.

The building geometry and its thermophysical properties are summarized in

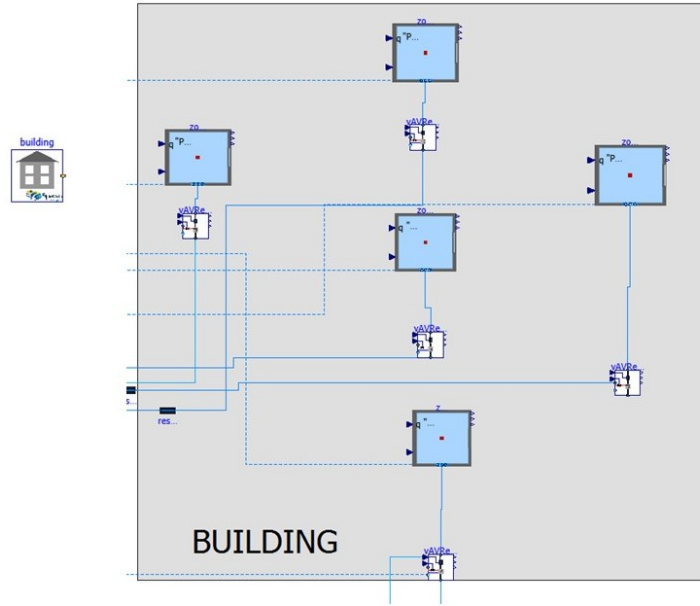


Figure 4.3: *Building on Modelica*

table 4.1.

4.3 Thermal and electrical system

4.3.1 Thermal system

The building is served by an all-air system that is responsible for ensuring thermohygrometric comfort conditions [35], i.e., an indoor set point of 20 °C and relative humidity around 50%, and adequate air quality index [36], i.e. 11 $l/(s \text{ people})$. The heat demand is covered by a heat pump that heats water in the thermal storage, and from this storage, water is sent to the AHU's heat exchange coil, which exchanges heat with air taken from outside, and to VAV boxes to regulate the temperature of the air supplied to each zone. Part of the air taken from the zones is mixed in the economizer with air

Table 4.1: *Building characteristics*

| Feature | Value |
|--|--------|
| Zone | 5 |
| Conditioned floor area [m^2] | 511.0 |
| Conditioned volume [m^3] | 1559.0 |
| Transparent/opaque envelope vertical surface ratio | 0.269 |
| Opaque envelope vertical surface [m^2] | 221.8 |
| U Value Wall W/m^2K | 0.78 |
| U Value Roof W/m^2K | 0.20 |
| U Value Foundation W/m^2K | 1.85 |
| U Value Window W/m^2K | 3.24 |

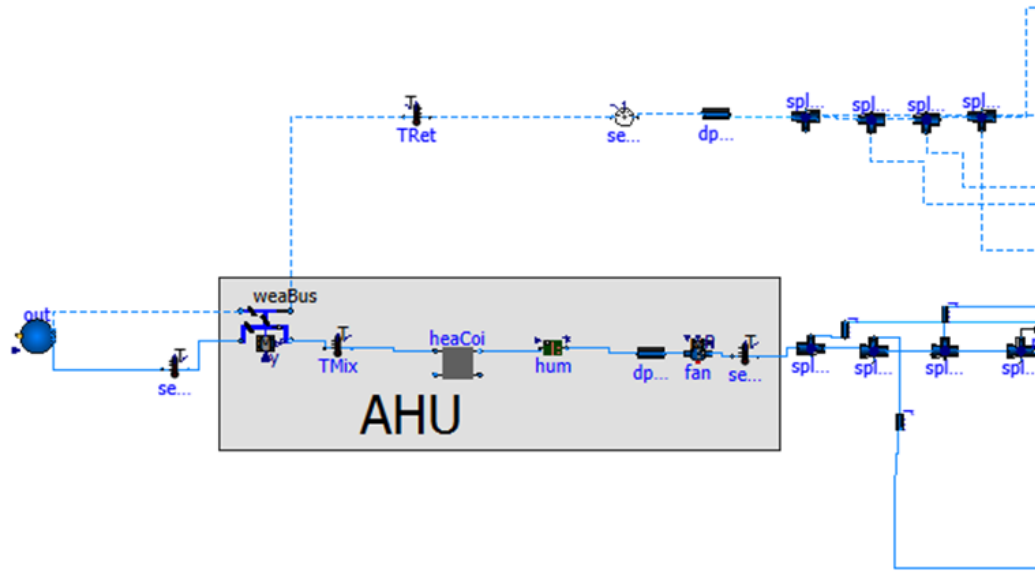
taken from outside so as to recover heat, while the remaining part is expelled outside.

In figure 4.4 the thermal system developed in Modelica is shown; figure 4.4a represents the air handling system with the following components:

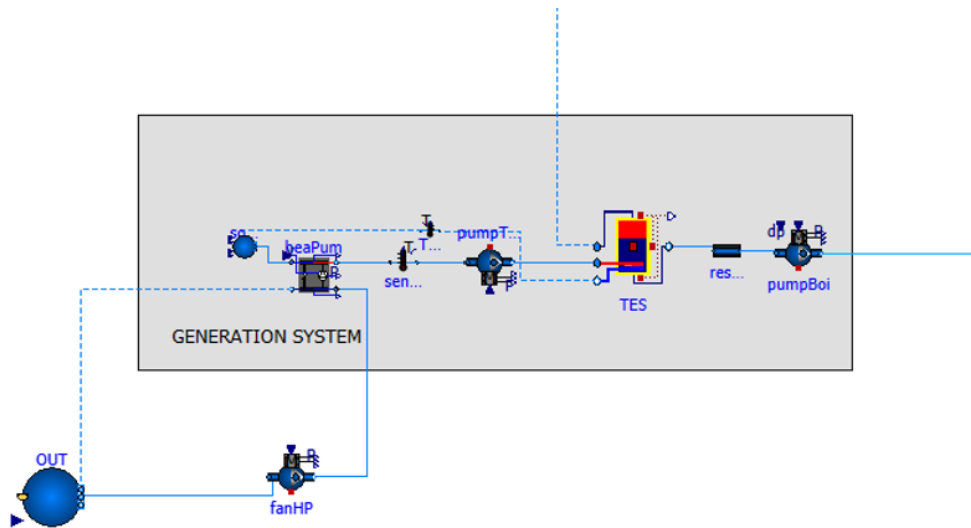
- Economizer
- Heating coil
- Humidifier
- Fan
- Components pressure drop
- Temperature sensors
- Air duct junctions

While figure 4.4b shows the generation system with the following components:

- Heat pump
- Temperature sensors
- Hydraulic pumps
- Thermal energy storage (TES) system
- Fan
- Components pressure drop



(a) AHU system



(b) Generation system

Figure 4.4: Thermal system.

4.3.2 Electrical system

Electrical loads are the sum of the internal needs of household appliances and the power required by the heat pump, fans, and pumps; appliance loads are uncontrolled because they serve the needs of the occupants. The electrical load of the building is met by the interaction of the grid and the electrical power generation of PV and stored in the electrical battery.

The interaction between the electric grid, the power produced by the PV and stored in the electric battery must be controlled in order to ensure the correct operation of the building and not to exceed the charging and discharging power of the storage system.

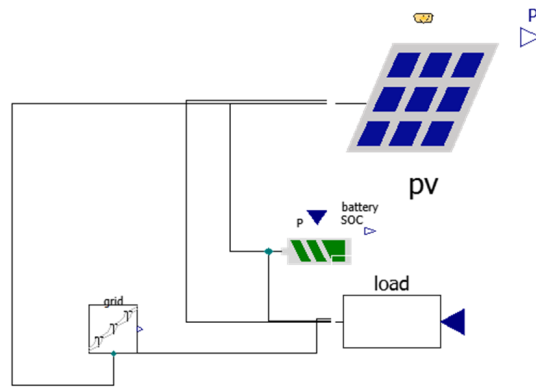


Figure 4.5: *Electrical system*

The electric system developed in Modelica is shown in figure 4.5, in which the following components are included:

- PV
- Battery
- Grid
- Load

4.4 Design

This section will show the procedure adopted to size the components used in the model for the simulation.

4.4.1 Building thermal load

The building was simulated with the Turin climate file in January. From the simulation on Energy Plus, the peak heating load of the building was extracted as equal to $\dot{Q}_{\text{loa}} \simeq 12 \text{ kW}$.

4.4.2 Nominal mass airflow rate

The nominal mass flow rate is calculated from the building heating load and indoor and supply air conditions. The internal conditions are equal to the set-point temperature $T_I = 20^\circ\text{C}$ and relative humidity $\varphi = 50\%$ while the supply conditions are assumed to be equal to a temperature $T_S = 25^\circ\text{C}$ and zero latent load as shown in the figure 4.6. The calculation of the nominal supply flow rate is as in Equation 4.1 in which the enthalpy values are taken from the psychrometric diagram also shown in Figure 4.6.

$$\dot{m}_s = \frac{\dot{Q}_{\text{loa}}}{h_S - h_I} = 2.2 \frac{\text{kg}}{\text{s}} \quad (4.1)$$

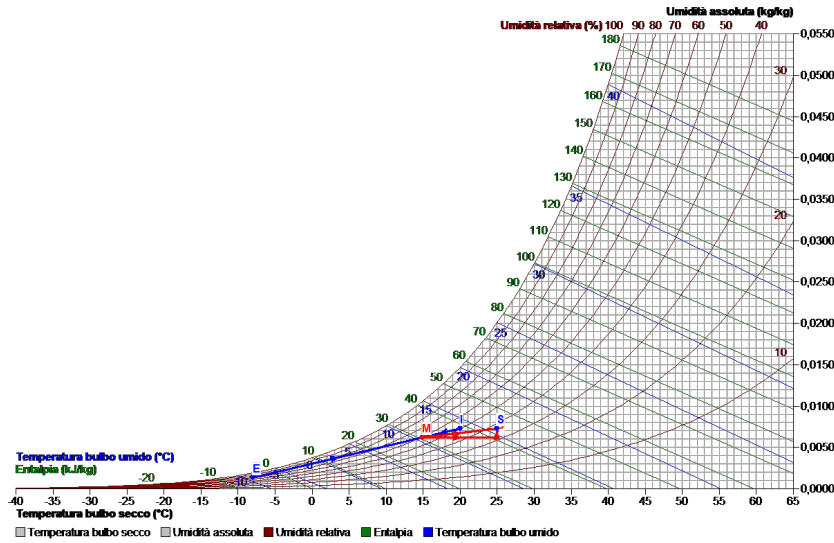


Figure 4.6: Psychrometric chart[37]

4.4.3 Nominal outdoor airflow rate

The outdoor air flow rate is calculated according to normative [36], in which the minimum outdoor air flow rate per person is equal to $\dot{V}_{\text{pers}} = 11 *$

$10^{-3} \text{ m}^3 / (\text{s} * \text{person})$, and with a maximum occupancy taken from the Energy Plus model of $n_{\text{pers}} = 28$ people, the \dot{m}_E is shown in equation ??.

$$\dot{m}_E = \dot{V}_{\text{pers}} * n_{\text{pers}} * \rho_{\text{air}} = 0.37 \frac{\text{kg}}{\text{s}} \quad (4.2)$$

From the value of \dot{m}_E , the recirculation flow rate is calculated $\dot{m}_{\text{recirc}} = \dot{m}_s - \dot{m}_E$ and thus the recirculation rate of $M = \frac{\dot{m}_{\text{recirc}}}{\dot{m}_s} = 0.83$, which corresponds to a mixing temperature $T_{\text{mix}} = 0.83 * (T_I - T_E) = 15^\circ\text{C}$ i.e. the temperature at the outlet to the economizer.

4.4.4 Heating Coil

Because it is an all-air system, the air handling unit (AHU) must also take charge of ventilation losses, so the heating coil must provide an output equal to the sum of the building heat load and ventilation losses; the nominal heating coil power is calculated in according to equation 4.3.

$$\dot{Q}_{\text{heaCoi}} = \dot{m}_s (h_s - h_M) = 29 \text{ kW} \quad (4.3)$$

Where enthalpy values are taken from the diagram in figure 4.6; equation 4.3 is confirmed by equation 4.4 in which the heating coil power is equal to the sum of the power lost through ventilation and transmission.

$$\dot{Q}_{\text{heaCoi}} = \dot{Q}_{\text{loa}} + \dot{Q}_{\text{vent}} = \dot{Q}_{\text{loa}} + \dot{m}_E (h_I - h_E) = 29 \text{ [kW]} \quad (4.4)$$

The calculated main values are summarized in the table 4.2.

Table 4.2: *Sizing summary*

| Feature | Value |
|--|--------|
| Nominal heat flow [kW] | 30 |
| Nominal volume air flow rate $\left[\frac{\text{m}^3}{\text{h}}\right]$ | 6545.5 |
| Minimal external volume air flow rate $\left[\frac{\text{m}^3}{\text{h}}\right]$ | 1100.8 |
| Total Air changes $\left[\frac{1}{\text{h}}\right]$ | 4.20 |

4.4.5 Heat pump

The heating coil delivered thermal power is provided by a heat pump with a thermal power at the condenser of about $\dot{Q}_{\text{HP,cond}} \simeq 30 \text{ [kW]}$.

The heat pump is sized by choosing the power at the compressor equal

to $P_{\text{comp}} = 7 \text{ [kW]}$ and the temperature differences at the evaporator and compressor equal to -5°C and 5°C , respectively. For the calculation of the efficiency, the Carnot efficiency method is used. Thus, the efficiency is computed as:

$$\eta_{\text{mathrmcar}} = \frac{COP}{COP_{\text{mathrmcar}}} = 0.5 \quad (4.5)$$

4.4.6 Humidifier

Relative humidity conditions are achieved with a steam humidifier; the absolute humidity of the air from outside is measured and if lower than the nominal conditions is brought to the nominal conditions. The nominal conditions of absolute humidity of the air are equal to $x = 0.0073$ which for a temperature of 20°C gives a relative humidity of 50%. This comfort condition is achieved by injecting steam into the air stream taken from outside. The outlet specific enthalpy is increased by the enthalpy of steam at 100°C for the mass of steam that was added. For this reason, the outlet temperature of the air stream from the humidifier is slightly higher than the temperature of the incoming air stream.

4.4.7 Temperature energy storage

The purpose of the TES is to store the thermal energy produced by the heat pump during the periods when it is convenient to have the heat pump switched on and provide the energy needed by the building to maintain comfort conditions. The tank is connected to the AHU and VAV boxes; it is not possible to supply energy to the AHU and VAV boxes without first passing through the TES. The sizing of the volume TES was done considering the power supplied by the heat pump for two hours whose water temperature is raised from $T_{\text{TES,min}} = 35^\circ\text{C}$ to $T_{\text{TES,max}} = 50^\circ\text{C}$, as shown in equation 4.6.

$$V_{\text{TES}} = \frac{\dot{Q}_{\text{HP,cond}} h}{\rho c \Delta T} = 3.4 \text{ m}^3 \quad (4.6)$$

In this case study, several tests were carried out with different volumes of tanks; the different volumes are summarized in the table 4.3. The chosen volumes are obtained by having the height fixed at $h = 1.8 \text{ [m]}$ and varying the area consequently. The tank presents an insulation thickness $s = 0.05 \text{ [m]}$ and a specific heat conductivity $k = 0.04 \left[\frac{\text{W}}{\text{mK}} \right]$.

The mass flow rate sent by the heat pump is shown by the equation 4.7 where

Table 4.3: *Capacity of TES*

| Case | Value [m^3] |
|------|-----------------|
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |

$\Delta T = 10^\circ\text{C}$ was chosen.

$$\dot{m}_{\text{HeaWat,flow,nom}} = \frac{\dot{Q}_{\text{coi}}}{c_w \Delta T} = 0.693 \left[\frac{kg}{s} \right] \quad (4.7)$$

4.4.8 VAV box

Despite the fact that the AHU has been sized in such a way that the air will be brought from outside conditions to an inlet condition of 25°C ; the air will be brought from outside conditions to a temperature of 18°C , as will be explained in more detail in the control chapter 4.5. This way, each zone VAV box, according to its own internal conditions, can feed air into the room at the optimal conditions; in fact, as will be illustrated in the results chapter 5, zones having different exposures to sunlight will have different requirements in terms of discharge air temperature.

As shown in figure 4.7, a VAV box contains:

- A *heat exchanger* in which primary air from the AHU passes on one side and hot water from the TES passes on the other side. The purpose of the hot water is to raise the air temperature from 18°C to 25°C .
- A *valve* that regulates water flow rate passing through the exchanger so as to vary the post-heating thermal power. The valve is fully closed when there is no need to heat and can open continuously until it is fully open.
- A *damper* that regulates the flow of air into the zone to ensure air quality in each zone. The position varies from a minimum position that injects the minimum flow rate into the zone to ensure air quality, to fully open.
- The flow and temperature *sensors* that will be used for control logic in chapter 4.5.

The post heating power of a single zone is shown in the equation 4.8.

$$\dot{Q}_{PH} = \dot{m}_s c_{p,\text{air}} (T_S - T_{\text{AHU}}) = 4 \text{ [kW]} \quad (4.8)$$

4.4.10 Electrical storage

Electrical storage stores surplus energy so that it can be used when it is needed or when photovoltaic production is zero. The electrical storage was sized considering a maximum peak electrical power of 20 kW and, accordingly, choosing a battery that would provide one hour of autonomy, a capacity $C = 20.2 kWh$ with a maximum charging and discharging power of 5 kWh as shown in [39]. The percentage of maximum charge is considered to be 96% while the percentage of minimum charge is 4%.

In this case study, several tests were carried out with battery capacities that span from 20kWh to 70kWh, as summarized in the table 4.5. The maximum

Table 4.5: *Capacity of electrical storage*

| Case | Value [kWh] |
|------|-----------------|
| 1 | 20.2 |
| 2 | 30.2 |
| 3 | 40.3 |
| 4 | 50.4 |
| 5 | 60.5 |
| 6 | 70.6 |

charging and discharging power of the battery is $|P_{\max}| = 5 [kW]$

4.4.11 Schedule

The following figure 4.8 shows the occupancy schedules, and on/off switching of the HVAC system, lights and equipment:

- *Occupation schedule*: as shown in the figure 4.8a, the occupancy schedule is given as a percentage of the maximum occupancy for all zones, that is 28 people. Office occupation begins at 07:00 and ends at 18:00 with a lunch break from 12:00 to 13:00.
- *HVAC ON/OFF*: as is shown in 4.8b, HVAC system is turned on at 06:00 and turned off at 18:00.
- *Lights and equipment ON/OFF*: as shown in figure 4.8c and 4.8d, the on/off schedule for lights and equipment is given as a percentage of the maximum power that is 5 kW for both lights and equipment.

The schedules just described apply to weekdays, whereas on holidays the HVAC is off because there is no occupancy and only 10% of the lights are on.

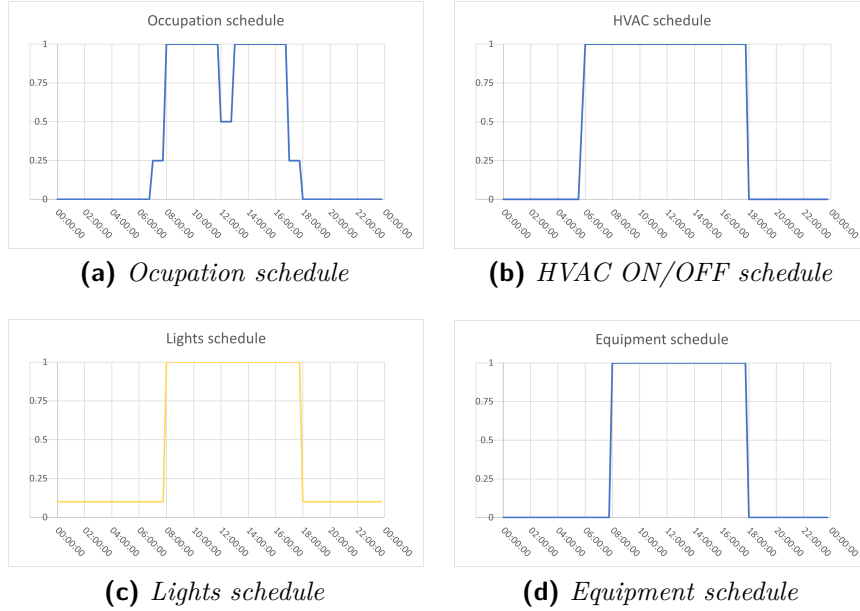


Figure 4.8: Schedule.

4.5 Control system

Plant control can be divided, as shown in the figure 4.10, into two levels:

1. *High-level*: it regulates the energy exchange between the PV, electric battery, building load and the national grid. Parallel to this, there is the control that regulates the charging and discharging of thermal and electrical storage.
2. *Low-level*: this control manages the system components of the HVAC and VAV boxes following Guideline 36 [2].

4.5.1 High-level

High-level controls are supervisory controls; in this section, controls that regulate exchange with the grid and controls that regulate storage charging and discharging will be shown.

Energy exchange with the power grid

Electrical exchange takes place between the grid, load, PV and battery following these operation modes:

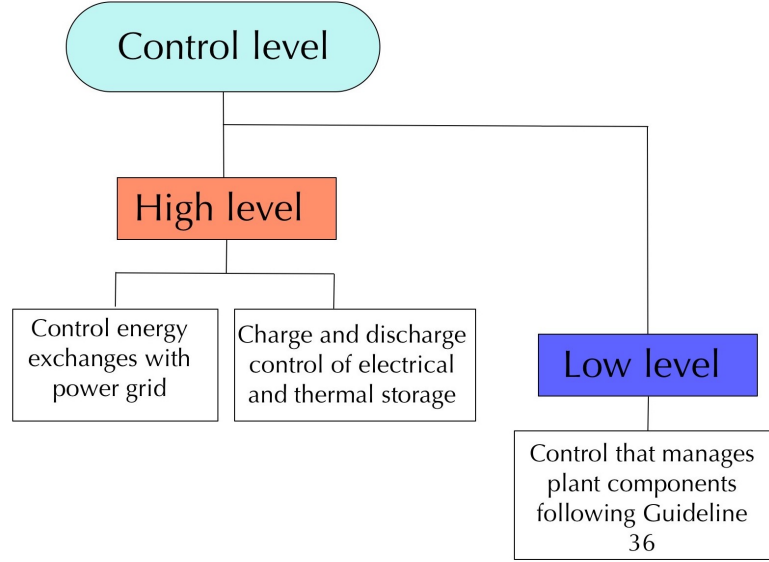


Figure 4.9: *Control level.*

1. Priority is always given to PV to meet the load.
2. Grid cannot charge the battery.
3. PV can sell to the grid (assumed no remuneration).
4. If necessary, electricity can be purchased from the grid ($C_{F_1} = 0.269 \left[\frac{\text{€}}{\text{kWh}} \right]$,
 $C_{F_{23}} = 0.249 \left[\frac{\text{€}}{\text{kWh}} \right]$).

The flow chart that regulates battery charging and discharging is shown in the figure 4.12. As explained in [40], the first check made is whether the charging or discharging power is above the imposed limit of $|P_{\max}| = 5 \text{ [kW]}$. If the charging or discharging power is higher than the limit, it is set equal to the limit. Once the charging and discharging power has been controlled and it is ensured that the limit is not exceeded, a state of charge (SOC) control of the battery is carried out, which is shown in figure 4.12.

If PV production is greater than the building's electricity needs $P_{\text{PV}} > |P_{\text{Loa}}|$, it is defined energy *surplus*; otherwise it is defined as energy *deficit*. If the battery state of charge is maximum, in case of energy surplus the power supplied to the battery is zero, as shown in 4.11a, while in case of deficit photovoltaic, battery and if necessary the power grid meet the building loads

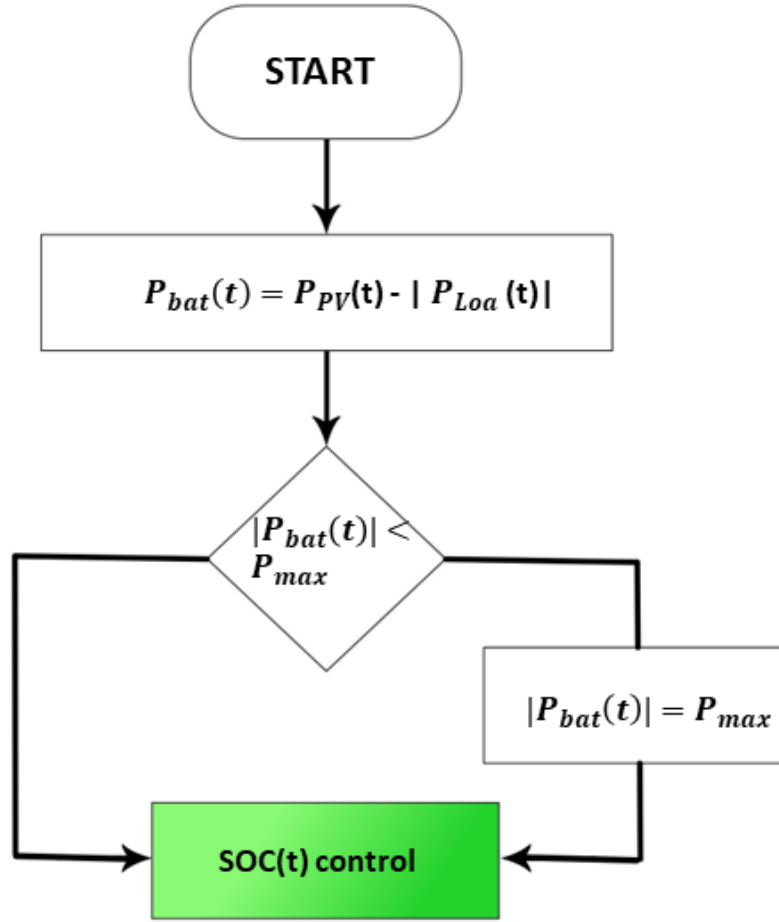


Figure 4.10: Control of maximum charging and discharging power.

is shown in 4.11b; the power balance equation is shown in 4.9.

$$P_{gri}(t) = |P_{Loa}(t)| - P_{PV}(t) - P_{bat}(t) \quad (4.9)$$

Where the grid power P_{gri} is greater than zero when supplying power to the electrical system and negative when taking power from the electrical system. If the battery state of charge is minimal in case of deficit, the battery charging power is zero and the grid provides energy to the building load as shown in 4.11d; in the case of surplus, however, PV meets the load and the surplus energy charges the battery as shown in 4.11c; in this case the electrical power balance is described in equation 4.9.

If the state of charge is neither minimum nor maximum, the balance is

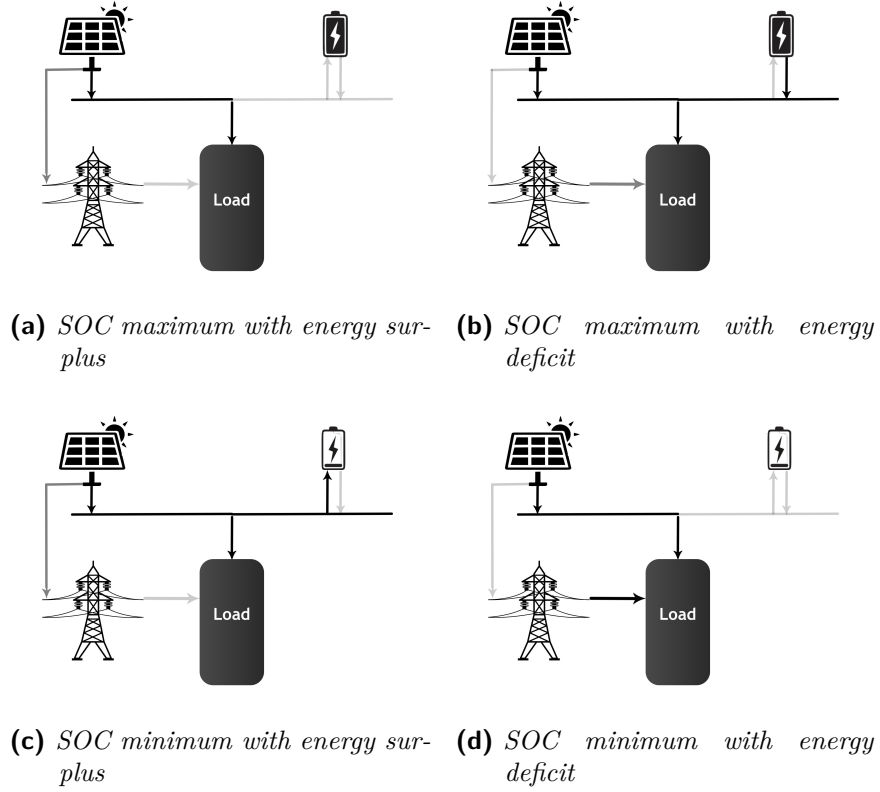


Figure 4.11: SOC status

described in equation 4.9 so the battery discharges or charges depending on whether there is a surplus of energy or a deficit.

Charge and discharge storage control

Two types of control will be shown in the following section:

- Baseline
- Self-consumption

The minimum temperature of the thermal storage was chosen to be $T_{\text{tank,min}} = 35^\circ\text{C}$ while the maximum temperature was chosen as equal to $T_{\text{tank,max}} = 50^\circ\text{C}$. As explained in [26] «*raising temperature level of the storage tank increases the delivered energy (DE) for the heating system*»; as shown in figure 4.13 a maximum tank temperature below 50°C results in a lower DE for the heating system.

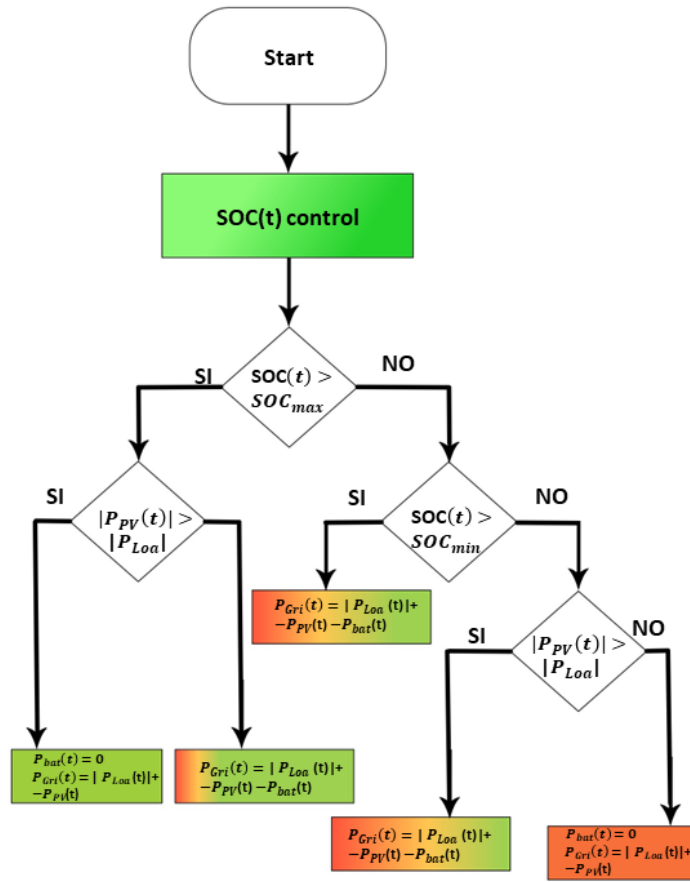


Figure 4.12: Control of state of charge (SOC).

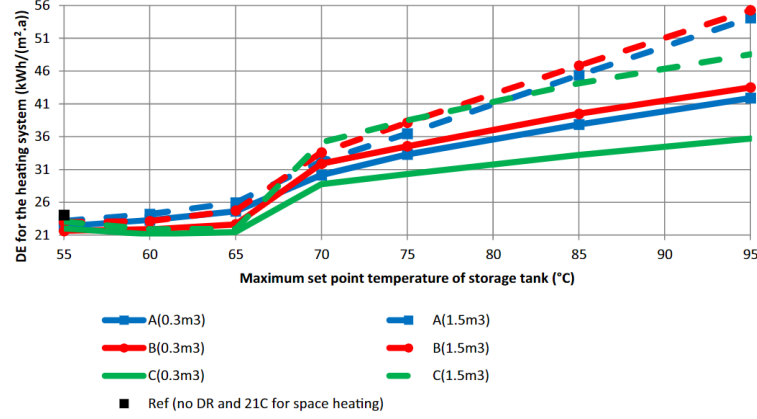


Figure 4.13: *Delivered energy for the heating system versus the maximum temperature set point [26].*

Baseline control. In baseline control, it is only ensured that the minimum tank temperature does not fall below 35°C ; whenever the tank reaches the minimum temperature, the heat pump switches on at maximum power to bring the temperature to $T_{\text{tank,max}} = 50^{\circ}\text{C}$. The flow chart of this control is shown in figure 4.14.

This control does not take into consideration the state of charge of the electric battery and thus the possibility of charging the thermal storage when there is a surplus of energy.

Self-consumption control. This control aims at maximising self-consumption; analyzing the baseline case, it can be seen the energy produced by the photovoltaic is sold to the grid when the battery is full and when the temperature $T_{\text{tank}}(t) > 35^{\circ}\text{C}$ i.e. when both storage units are full.

Therefore, the control shown in the flow chart in figure 4.15 tends to prevent the battery from being fully charged, as explained in [41]; if the SOC of the battery at time t exceeds a maximum SOC value, the heat pump is switched on in order to raise the electrical load of the building and consume the energy of the electrical battery. If the SOC is below the limit, it is ensured that the tank does not reach the minimum temperature; if the tank reaches the minimum temperature, the temperature is raised to $T_{\text{tank,med}} = 40^{\circ}\text{C}$. Once the temperature reaches 50°C , the heat pump is only switched on again when it reaches 45°C in order to avoid a continuous ON/OFF of the machine and damage it.

Whenever the heat pump is switched on, the type of control that regulates the output of the heat pump compressor is a PI control; the tank temperature

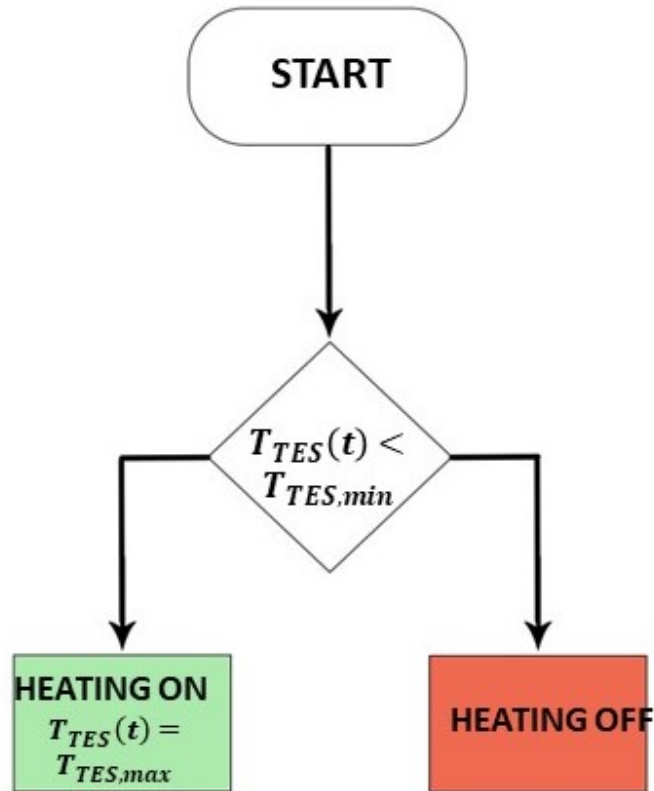


Figure 4.14: *Baseline control flowchart*

is measured and compared with the set point, which can be $T_{\text{tank,med}}$ or $T_{\text{tank,max}}$, depending on the case, then a proportional and integral response is sent.

4.5.2 Low-level

Low-level controls are field-level controls that regulate the opening and closing of valves and dampers in the HVAC system.

In 2018, the American Society of Heating, Refrigeration and Air conditioning Engineers (ASHRAE) published the Guideline 36 (G36) [2]. G36 is a guide that provides optimal control sequences to reduce design time and achieve optimum performance in terms of thermal comfort. G36 will be used for some of the plant components.

In this section, the control of the following system components will be

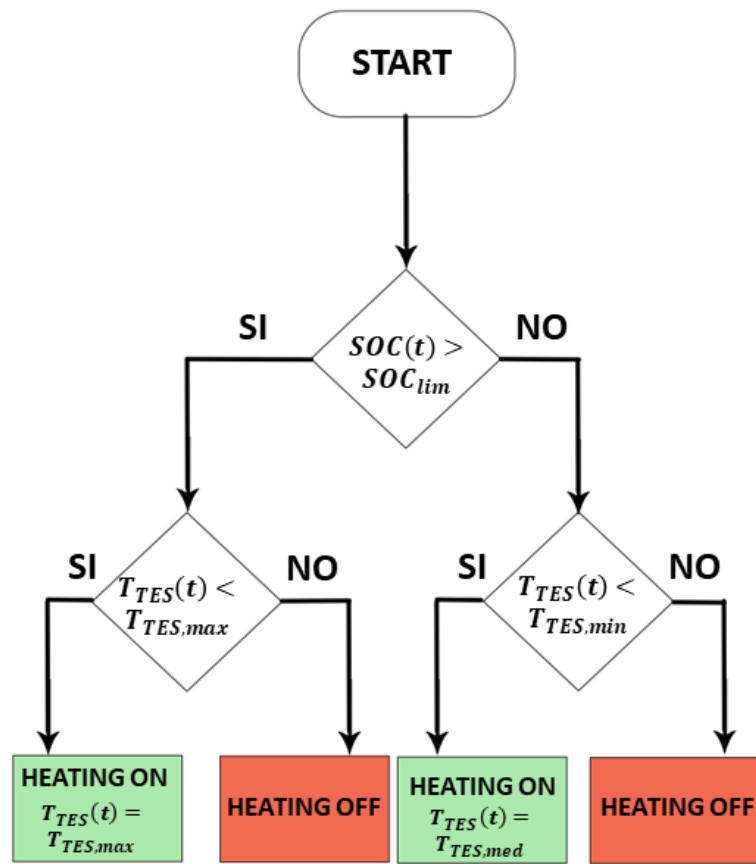


Figure 4.15: *Self-consumption control flowchart*

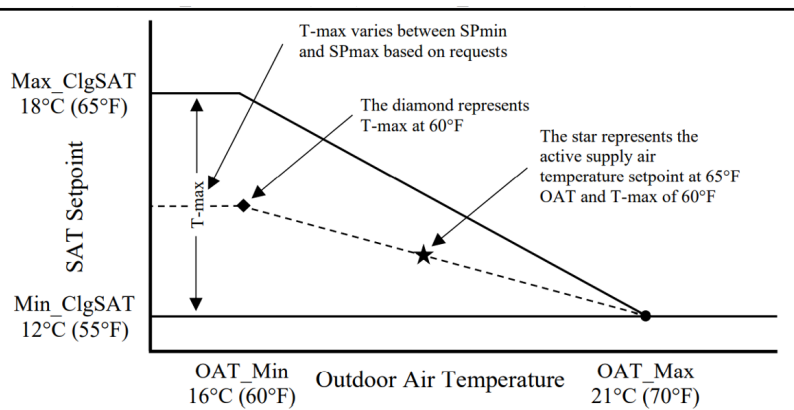
discussed in detail:

- Economizer
- Heating coil
- Fan
- VAV boxes

Economizer

Economizer control is done following G36 from the definition of the Supply Air Temperature (SAT) setpoint. The SAT is calculated on the basis of the outside temperature conditions as shown in figure 4.16. If the outside temperature is lower than $T_{\text{out}} = 16^\circ\text{C}$, the supply temperature will be $T_{\text{SAT,sp}} = 18^\circ\text{C}$; if the outside temperature is greater than $T_{\text{out}} = 21^\circ\text{C}$, the supply temperature will be $T_{\text{SAT,sp}} = 12^\circ\text{C}$.

Since in winter the outside temperature is almost always below 16°C the



Informative Figure 5.16.2.2 Example supply air temperature reset diagram.

Figure 4.16: *Supply Air Temperature diagram [2]*

AHU sends the air at 18°C and then each zone will adjust the air supply temperature according to how much it needs to heat.

The position of the economizer damper is controlled by the SAT set-point; the SAT set point is compared with the SAT measurement taken at the fan outlet. If the measured temperature is below the set point, the damper position is the minimum and only the air required to meet the air quality index is taken from outside; if the measured temperature is higher than the set-point, the damper position is opened proportionally to the difference

between measured and set-point. The line of the graph shown in figure 4.17 describing the position of the outer damper.

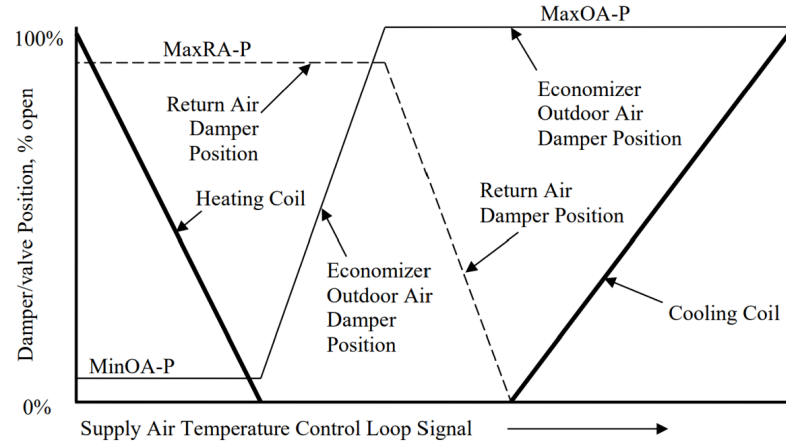


Figure 4.17: *Economizer outdoor air damper position [2]*

The percentage of outside air flow rate to total air does not correspond to the percentage of opening of the outside air damper. Several test simulations were required to calculate the minimum damper opening that guarantees the outdoor air quality conditions.

Heating coil

Heating coil control is done by monitoring the flow rate of the water passing through the heat exchanger as explained in [2]. SAT set point is calculated as explained previously in figure 4.16; SAT set point is compared to the actual measurement of the supply temperature; as the difference between set point and measured value increases, the mass flow rate sent to the exchanger increases by imposing mass flow rate on the hydraulic pump as shown in figure 4.17. When the actual supply temperature measurement exceeds the set point, the hydraulic pump is turned off.

Fan

As explained in [42] in a VAV box system, the dampers change position as the ventilation loads change; the change in damper position affects the static pressure value in a duct system. When the static pressure changes, the fan responds by changing speed in order to maintain the static pressure at a predefined set-point. For example, when the load of a zone decreases, the

corresponding VAV box damper closes; the fan then decreases its speed in order to keep the static pressure constant, which results in energy savings.

VAV boxes

The control of VAV boxes is divided into valve and damper control.

Valve. The valve regulates the mass flow rate of water that exchanges heat with the primary air for zone post-heating. The measured zone temperature is compared to the zone set point; if the difference between measured temperature and set point is less than zero, the valve opens proportionally to how large the difference is. If the difference is greater than zero, the valve is closed as shown in figure 4.18.

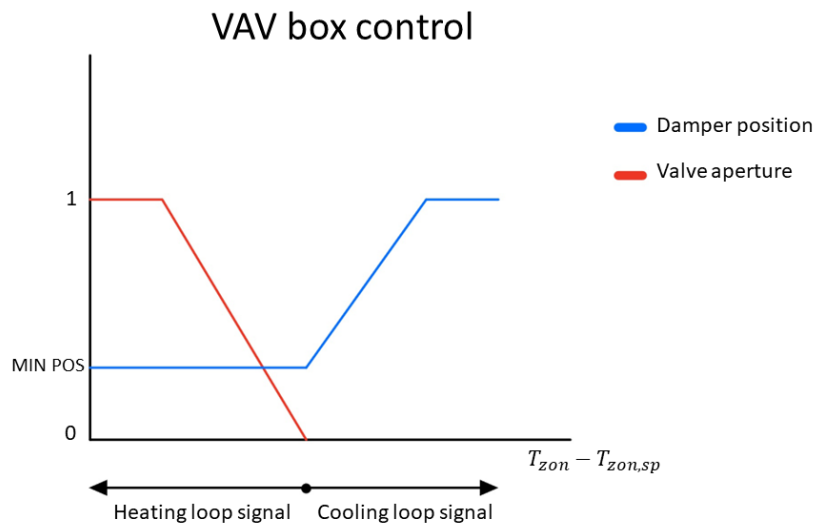


Figure 4.18: VAV box control.

Damper. The function of the damper is to regulate the mass flow to be fed into the zone. As shown in figure 4.18, if the difference between the zone temperature and the set-point is lower than zero, the damper is at the minimum position to guarantee air quality; however, if the difference is greater than zero, the damper opens.

CHAPTER 5

Results

In this chapter, the results of the high-level and low-level controls will be analysed; the analysis will start by showing the results of the low-level controls focusing on zone temperatures and the positions of the valves and dampers in VAV boxes. Then the analysis will proceed with the high-level control, testing two types of control: a simple one called *baseline* control and a control that aims at maximising the consumption produced by photovoltaics, called *self-consumption control*.

5.1 Low-level

The main objective is to have thermo-hygrometric comfort in ventilated zones, i.e. to obtain a temperature of 20 °C during the occupation hours of 07:00 to 18:00 from Monday to Friday. In figure 5.1, the five zones of the second week of January are shown and the periods of occupation are highlighted with a darker background.

The highest temperatures reached are in the southern zone due to solar exposure and in the core because it has no vertical surfaces to the outside environment; the temperature of 20 °C is reached in all areas during the occupation period.

The variables concerning the VAV boxes of the south and west zones were analysed in detail because they behave differently: the inside temperature of the south zone during the occupancy hours often exceeds the set point, while the inside temperature of the west zone reaches the proximity the set point hardly ever exceeding.

As shown in figure 5.2a, when the internal temperature in the south zone exceeds the zone set point, the valve closes and the damper opens; on the other hand, the figure 5.2b shows that in the zone to facing west, the internal

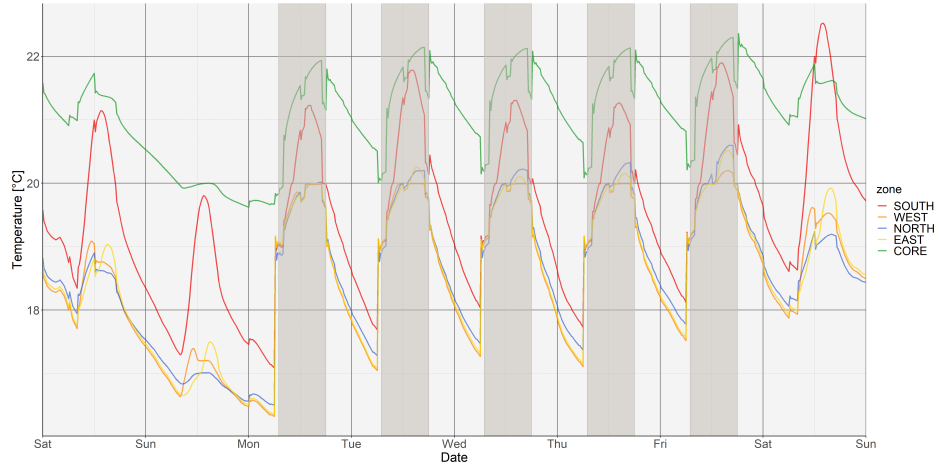


Figure 5.1: *Temperature of five zone.*

temperature is close to 20°C and therefore the valve always remains open and the damper in the minimum position as explained in 4.18.

Figure 5.3 shows the inlet and outlet temperatures of the heating coil for a January day during the hours in which the ventilation system is switched on 4.8b. The AHU supplies air to the VAV boxes at a temperature of 18°C as explained above and shown in figure 4.16. To raise the air temperature from the outside temperature to the SAT the air is mixed with a part of the return air. As shown in figure 5.4, the economizer damper is in a fixed position due to low winter temperatures as shown in figure 4.17. Then the mixed primary air is heated by exchanging heat with the hot water produced by the generation system. The figure 5.3 shows the temperature of the water supplying to the coil and the temperature of the return water.

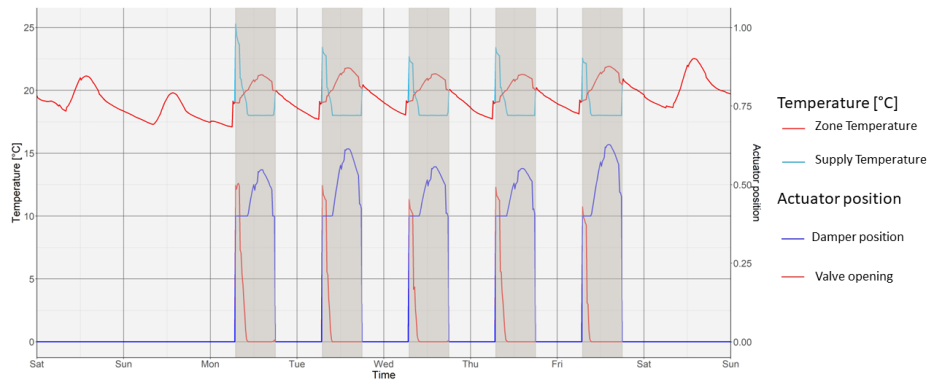
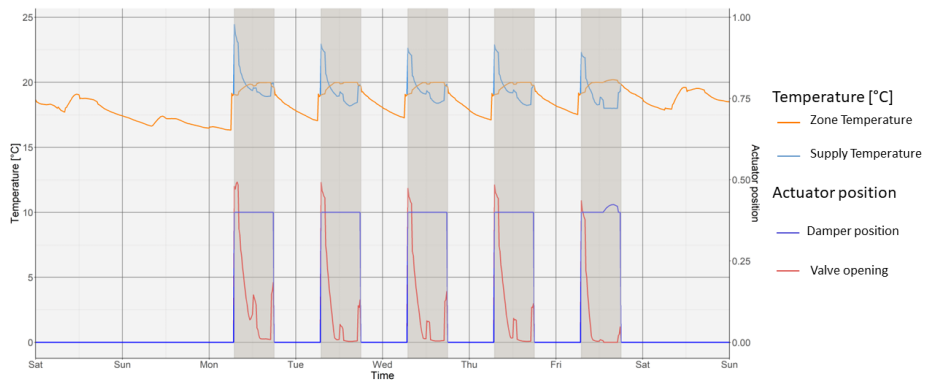
5.2 High-level

In the following section, the results of the two strategies adopted will be analysed through the following key performance indicators (KPIs):

- *Locally generated and consumed energy* (LGCE) is the energy produced by photovoltaics and self-consumed on site or stored in the electric battery, in other words it is the surplus energy that is not sold to the grid. The equation that describes this KPI is as follows

$$LGCE = E_{\text{buil,load}} - E_{\text{gri}} \text{ [kWh]} \quad (5.1)$$

- *Self-sufficiency* (SS) is the percentage of load that is met by the energy

(a) *South zone*(b) *West zone***Figure 5.2:** Temperature and actuators position of south and west zones

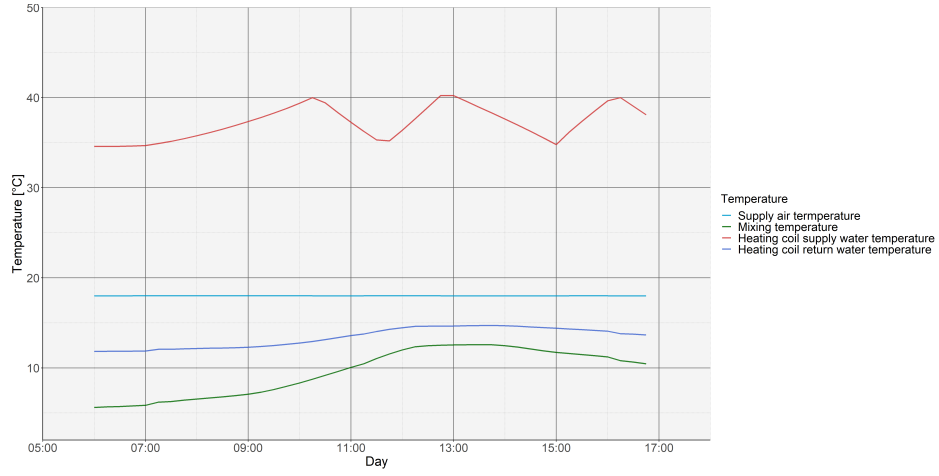


Figure 5.3: *water and air inlet and outlet temperature of the heating coil.*

produced and self-consumed; the equation that describes this KPI is

$$SS = \frac{LGCE}{E_{\text{buil,load}}} \quad (5.2)$$

- *self-consumption* (SC) is the percentage of energy produced and self-consumed of the total energy produced by photovoltaics; the equation that describes this KPI is

$$SC = \frac{LGCE}{E_{\text{PV}}} \quad (5.3)$$

- *monthly expenditure* (C) is the cost of the monthly energy bill considering a bi-hourly rate tariff summarised in the table where the minimum price is on public holidays and on weekdays from 00:00 to 08:00 and from 19:00 to 23:00. At other times the tariff with the maximum price applies.

Table 5.1: *Bi-hourly rate*

| Tariff | Value | $\left[\frac{\text{€}}{\text{kWh}} \right]$ |
|--------------|-------|--|
| C_{F_1} | 0.260 | |
| $C_{F_{23}}$ | 0.249 | |

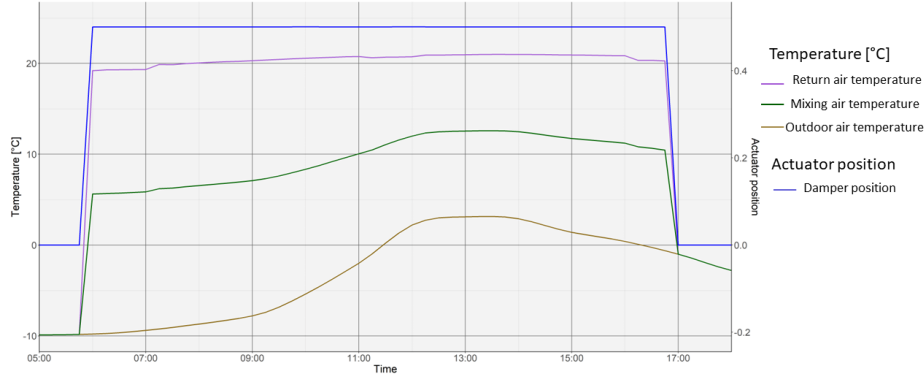


Figure 5.4: *Temperature and actuators position of the economiser.*

Baseline strategies

For baseline cases, several tests were carried out by varying the tank capacity configurations summarised in table 4.5 and the electric battery summarised in table 4.3; the table with the tests carried out and results obtained is 5.2.

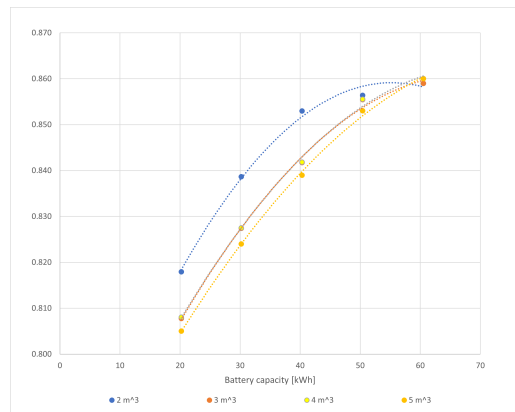
Some correlations were extracted from the tests; as shown in figure 5.5, with this strategy, as the volume of the TES increases, the self-consumption decreases. Due to a lower self-consumption, the monthly cost is higher for larger TES volumes. Figure 5.5 and 5.6 show how performances improve as battery capacity increases. It can be observed that by increasing the volume of the TES from 3 [m^3] to 4 [m^3] the KPI remains almost unchanged and by increasing the volume further it worsens significantly. It could be concluded that increasing the TES volume is not appropriate for this strategy; this behaviour will be further investigated in the next section where the self-consumption strategy will be analysed.

Case 4 in table 5.2 was chosen as representative to analyse the variables of interest for this control strategy. As shown in figure 5.7, the tank behaves exactly as explained in the flow chart in figure 4.14, every time the temperature of the TES reaches 35 °C, the heat pump is activated to raise the temperature to 35 °C.

Figure 5.8 shows the energy exchange between building load, electricity grid, photovoltaics and battery according to the rules depicted in figure 4.12.

Table 5.2: *Results baseline strategy*

| Case | Volume TES [m ³] | Battery capacity [kWh] | LGCE [kWh] | SC | SS | Cost [$\frac{\text{€}}{\text{kWh}}$] |
|------|---------------------------------|---------------------------|---------------|-------|-------|---|
| 1 | 2.0 | 20.2 | 1090.5 | 0.818 | 0.299 | 679.4 |
| 2 | 2.0 | 30.2 | 1118.1 | 0.839 | 0.307 | 672.2 |
| 3 | 2.0 | 40.3 | 1137.2 | 0.853 | 0.312 | 667.1 |
| 4 | 2.0 | 50.4 | 1141.7 | 0.856 | 0.313 | 665.9 |
| 5 | 2.0 | 60.5 | 1145.2 | 0.859 | 0.315 | 664.9 |
| 6 | 3.0 | 20.2 | 1076.9 | 0.808 | 0.294 | 691.4 |
| 7 | 3.0 | 30.2 | 1103.1 | 0.827 | 0.301 | 684.4 |
| 8 | 3.0 | 40.3 | 1122.2 | 0.842 | 0.306 | 679.3 |
| 9 | 3.0 | 50.4 | 1140.5 | 0.855 | 0.311 | 674.5 |
| 10 | 3.0 | 60.5 | 1145.5 | 0.859 | 0.314 | 673.2 |
| 11 | 4.0 | 20.2 | 1077.3 | 0.808 | 0.291 | 703.3 |
| 12 | 4.0 | 30.2 | 1103.2 | 0.827 | 0.298 | 696.4 |
| 13 | 4.0 | 40.3 | 1122.3 | 0.842 | 0.303 | 691.3 |
| 14 | 4.0 | 50.4 | 1140.6 | 0.856 | 0.308 | 686.5 |
| 15 | 4.0 | 60.5 | 1146.6 | 0.860 | 0.315 | 685.8 |
| 16 | 5.0 | 20.2 | 1073.3 | 0.805 | 0.288 | 708.6 |
| 17 | 5.0 | 30.2 | 1099.2 | 0.824 | 0.295 | 701.7 |
| 18 | 5.0 | 40.3 | 1118.4 | 0.839 | 0.300 | 696.7 |
| 19 | 5.0 | 50.4 | 1136.7 | 0.853 | 0.305 | 686.5 |
| 20 | 5.0 | 60.5 | 1146.6 | 0.860 | 0.312 | 689.7 |

**Figure 5.5:** *Selfconsumption-capacity correlation baseline strategy.*

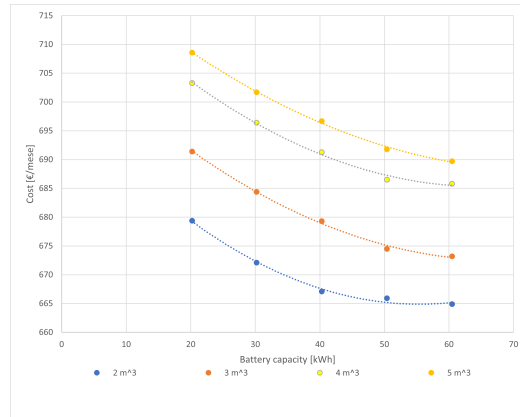


Figure 5.6: *Cost-capacity correlation baseline strategy.*

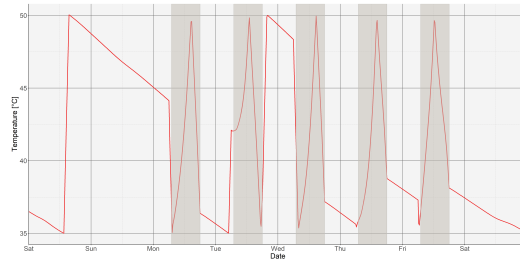


Figure 5.7: *Temperature of TES.*

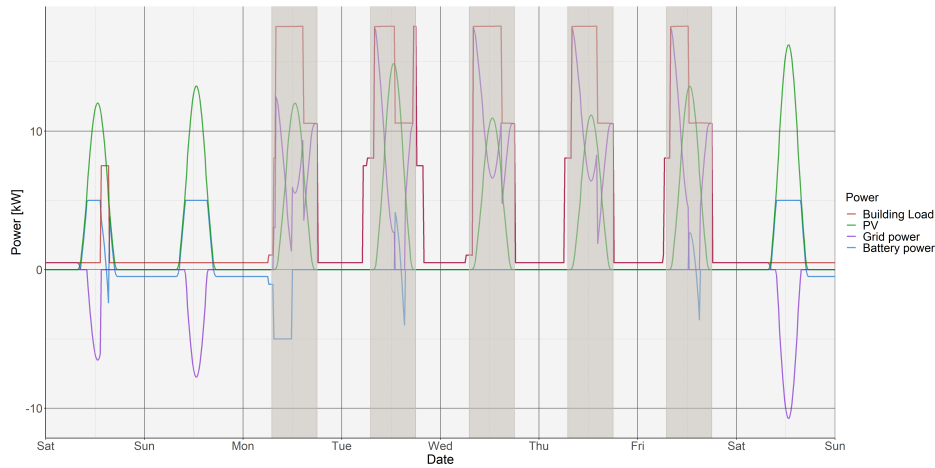


Figure 5.8: *Electrical loads baseline strategy.*

Self-consumption strategy

Various tests were also carried out for the strategy aimed at improving self-consumption and the results are summarised in table 5.3.

Table 5.3: *Results selfconsume strategy*

| Case | Volume TES [m ³] | Battery capacity [kWh] | LGCE [kWh] | SC | SS | Cost [$\frac{\text{€}}{\text{kWh}}$] |
|------|---------------------------------|---------------------------|---------------|-------|-------|---|
| 1 | 2.0 | 20.2 | 1097.0 | 0.833 | 0.312 | 642.6 |
| 2 | 2.0 | 30.2 | 1123.8 | 0.843 | 0.320 | 635.3 |
| 3 | 2.0 | 40.3 | 1130.3 | 0.858 | 0.322 | 633.0 |
| 4 | 2.0 | 50.4 | 1147.5 | 0.861 | 0.326 | 629.5 |
| 5 | 2.0 | 60.5 | 1152.5 | 0.864 | 0.328 | 626.9 |
| 6 | 2.0 | 70.6 | 1141.2 | 0.856 | 0.325 | 629.8 |
| 7 | 3.0 | 20.2 | 1111.3 | 0.834 | 0.313 | 647.3 |
| 8 | 3.0 | 30.2 | 1128.2 | 0.846 | 0.318 | 642.6 |
| 9 | 3.0 | 40.3 | 1140.1 | 0.855 | 0.322 | 639.3 |
| 10 | 3.0 | 50.4 | 1147.5 | 0.861 | 0.326 | 629.6 |
| 11 | 3.0 | 60.5 | 1158.7 | 0.869 | 0.327 | 634.2 |
| 12 | 3.0 | 70.6 | 1145.3 | 0.859 | 0.323 | 637.2 |
| 13 | 4.0 | 20.2 | 1125.4 | 0.844 | 0.315 | 651.6 |
| 14 | 4.0 | 30.2 | 1139.6 | 0.855 | 0.319 | 647.5 |
| 15 | 4.0 | 40.3 | 1143.8 | 0.858 | 0.320 | 646.0 |
| 16 | 4.0 | 50.4 | 1153.3 | 0.865 | 0.323 | 643.0 |
| 17 | 4.0 | 60.5 | 1156.1 | 0.867 | 0.324 | 642.0 |
| 18 | 4.0 | 70.6 | 1143.7 | 0.858 | 0.320 | 645.7 |

From the results obtained, an initial consideration can be made by showing the correlation between energy produced and self-consumed and the volume of TES for different battery capacities: as shown in figure 5.9 for smaller battery capacities, by increasing the volume of the TES the energy produced and self-consumed increases; on the other hand, for battery capacities above 50[kWh] by increasing the volume of the TES the energy produced and self-consumed decreases.

Considering the electrical load of the building in correlation with the volume of the TES for the different capacities, it can be seen from figure 5.10 that the electrical load of the building is independent of the capacity of the battery but strongly depends on the volume of the TES. In fact, the figure shows how the points on the graph overlap for the different TES volumes considered. The variation in electrical load is due to the heat pump consumption, in fact other electrical consumption such as water pumps and fans are switched on according to a fixed schedule and their consumption is an order of magnitude lower than that of the heat pump; lighting and equipment consumption follows

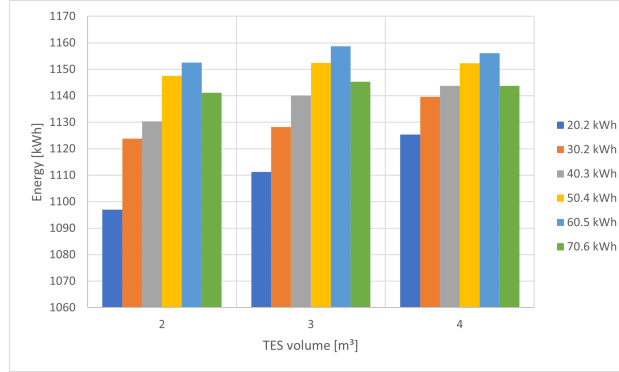


Figure 5.9: *Selfconsumption-Volume TES correlation.*

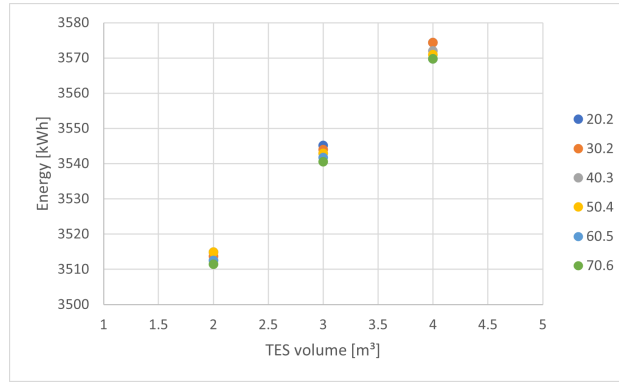


Figure 5.10: *Load building power-Volume TES correlation*

a schedule and are fixed.

The energy supplied by the heat pump to the tank is used to meet the heating load of the building according to equation 5.4.

$$E_{HP,thermal} = E_{buil,thermal} + E_{TES,thermal,loss} \quad (5.4)$$

In figure 5.11 it can be seen that the energy required by the building is the same while the TES losses increase as the volume increases. As shown in figure 5.12, self-consumption improves performance for higher battery capacities up to a maximum for 60 [kWh] and then decreases the percentage of self-consumption; the same behaviour also occurs for the monthly cost as shown for the figure 5.13.

The deterioration in performance for higher battery sizes is due to the maximum self-consumption strategy described in figure 4.15 was implemented with the same limiting SOC of $SOC_{lim} = 0.6$ for all battery sizes. As the capacity of the electric battery increases, the stored energy reaches less than the limit so the control becomes a baseline control as shown in figure 5.14

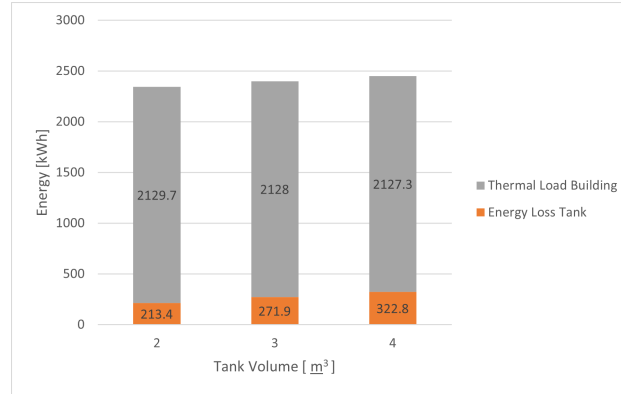


Figure 5.11: *Energy supplied by the heat pump*

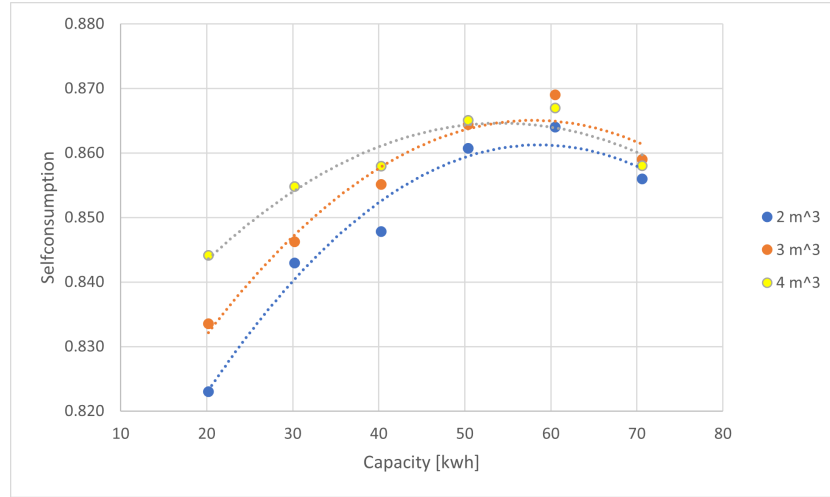


Figure 5.12: *Selfconsumption-capacity correlation for SC strategy.*

and the temperature of the TES continues to fluctuate between the minimum and intermediate temperature; the result of this behaviour is a deterioration of performance.

Several tests were carried out with various SOC_s, and it can be seen in figure 5.15 how self-consumption varies with higher battery capacity. In order to find an optimal SOC, a variable optimisation analysis must be carried out.

Comparing the baseline strategy with the maximum self-consumption strategy, it is possible to see from the graph in figure 5.16 how the improvement in terms of increased self-consumption and monthly savings occurs for smaller battery capacities. In fact, with larger capacities it is sufficient to have the control shown in figure 4.12 in which priority is given to PV consumption and the battery is only discharged when there is an energy deficit.

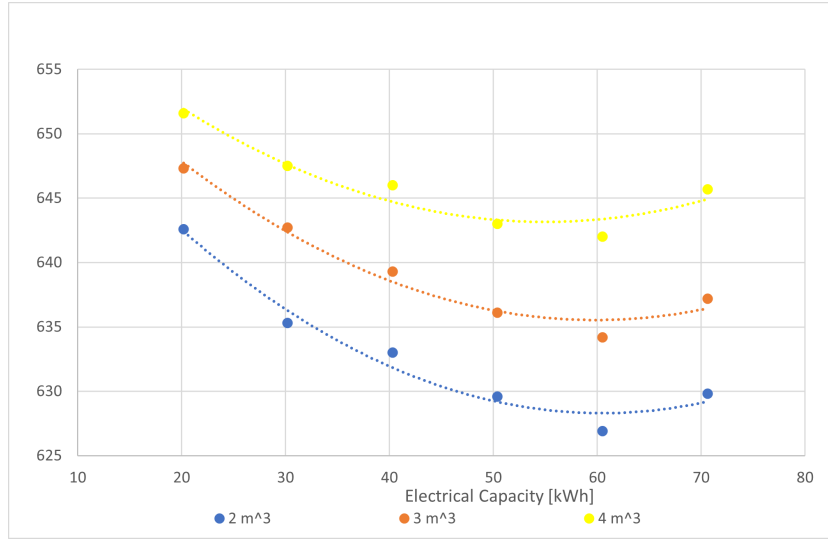


Figure 5.13: Cost-capacity correlation for SC strategy.

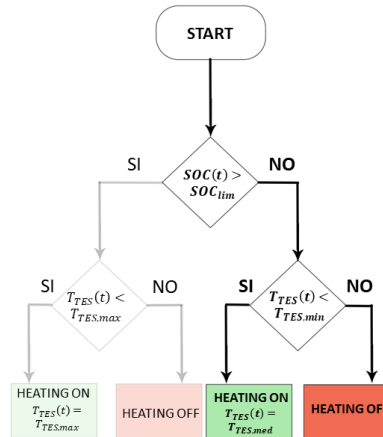


Figure 5.14: self-consumption control with a cut off part of the flow chart.

The greatest savings in terms of self-consumption are achieved for case 4 with a 20.2 [kWh] battery and a TES volume of 4 m³, for which a percentage increase in self-consumption of 3.5% results into a percentage saving on the monthly bill of 7.4%.

The following figure compares the state of charge of the TES and the battery for case 9 for baseline strategy and self-consumption strategy; in the winter period the surplus of energy from photovoltaics often occurs at the weekend when the load from lights and equipment is at its lowest. The

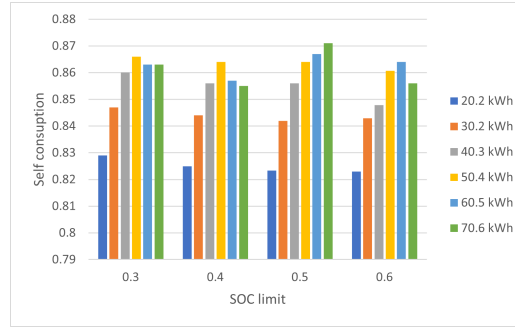


Figure 5.15: *Self-consumption for several SOC limit.*

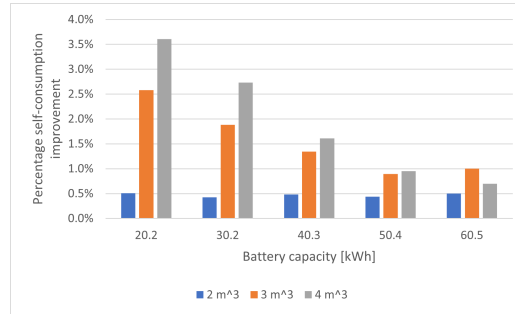
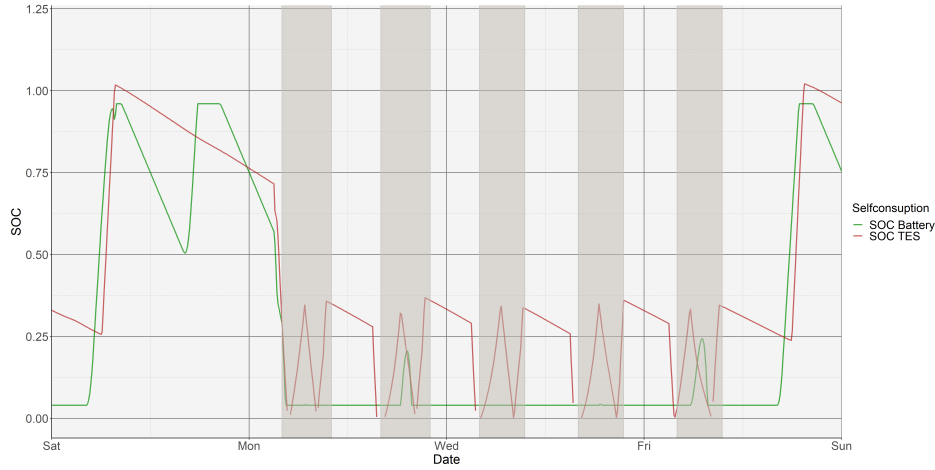


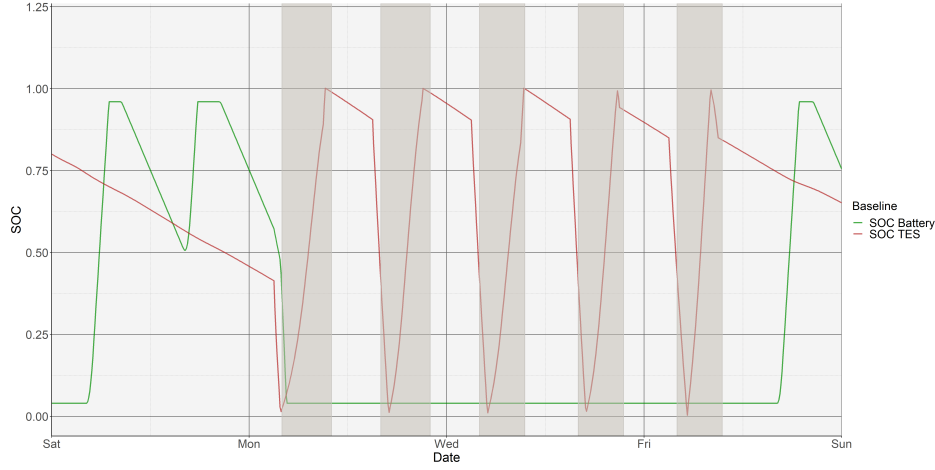
Figure 5.16: *Percentage self-consumption improvement*

highlighted background represents the period when energy is in the period with the highest price.

Figure 5.18 compares the loads of the building, the electricity grid, the power produced by the photovoltaics and the power of the electric battery; also in this figure, the highlighted background represents the period when energy is in the band with the highest price. Some qualitative considerations can be made that reflect the results found so far; from figure 5.18 it can be seen that the energy sold to the grid during the weekend has decreased, in fact as can be seen for some weekdays the heat pump is switched on and the entire load of the heat pump is met by the photovoltaic and the battery, thus avoiding both buying from the grid and selling to the grid. By increasing the energy produced and self-consumed, self-consumption also increases. Furthermore, looking at Tables 5.2 and 5.3, it can be seen that self-sufficiency is also increased because, as can be seen qualitatively from figure 5.18, the electrical load has decreased.

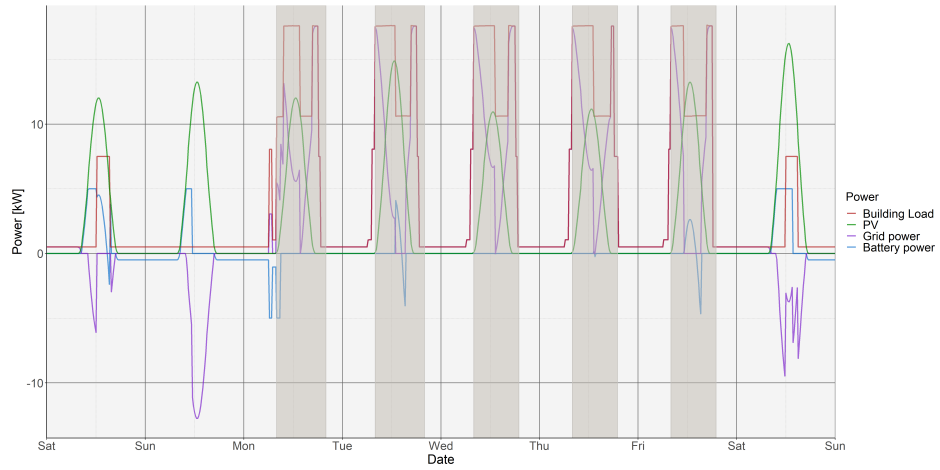


(a) SOC of TES and battery SC strategy case 4

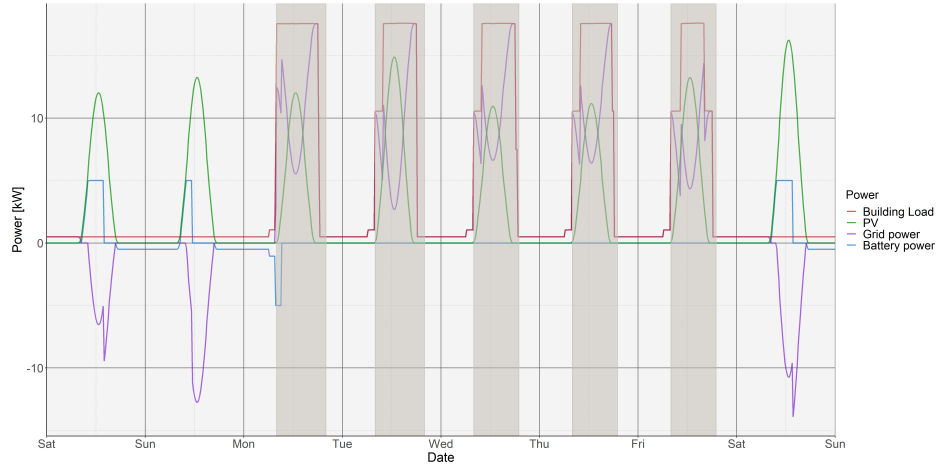


(b) SOC of TES and battery baseline strategy case 4

Figure 5.17: SOC of TES and battery



(a) *Electrical load SC strategy case 4*



(b) *Electrical load baseline strategy case 4*

Figure 5.18: Electrical loads SC strategy compared to baseline strategy

CHAPTER 6

Conclusion

The present thesis work aimed at using the new cosimulation tool called Spawn of EnergyPlus. Spawn was conceived a few years ago and became a reality in July 2021. Open standards such as Modelica and FMU enables industrial work on the integration and development of new technologies.

The possibility of coupling EnergyPlus with Modelica makes it possible to integrate an accurate simulation of the behaviour of the building envelope with the possibility of having a simulation of an HVAC system with an integrated control logic. In fact, the main advantage is to have a single control specification for both energy simulation and implementation in a single environment in order to have high-performance control sequences. Modelica also allows HVAC systems and, as shown in the chapter 4, also thermal generation systems such as heat pumps with associated thermal storage and electrical generation systems such as photovoltaic systems with associated electrical storage, to be modelled.

The tool allows to go into detail at plant level; each component has a level of complexity ranging from the most ideal model to the most realistic depending on the degree of detail desired; Spawn allows the energy losses of the thermal storage system to be evaluated on the basis of the tank volume and thus its dispersion surface area. As shown in Chapter 5, these losses are directly related to the consumption of the heat pump and thus to the total electricity consumption; so an evaluation on a single component allows an evaluation on the building consumption.

As shown in section 4.5, this instrument allows for low-level control with the aim of achieving thermal comfort and adequate air quality by taking into account internal inputs due to occupancy, artificial light and sunlight, and equipment. Also shown in section 4.5 are the high-level controls; rule-based controls (RBCs) while simple are still widely used because of their robustness and ease of implementation, and Spawn allows them to be easily integrated

into energy system simulations. The results obtained in chapter 5 demonstrate how the tool is capable of simulating the control of the exchange between the power grid, the building, the PV and the associated electrical storage, and how it is capable of solving complex systems such as interconnected thermal and electrical storages.

Despite successful experimentation with rule-based controls, Spawn's full potential is still unexplored; the next step is to test advanced control strategies such as Reinforcement Learning (RL) or Model Predictive control (MPC). Because the advanced controls are data-driven, they also incorporate information from stochastic variables such as occupancy or climate; having this kind of information allows to make building control more flexible to external influences.

The ability to test more complex control strategies with a simulated plant that comes close to reality marks the path to the future of co-simulation.

Bibliography

- [1] Lei Yang et al. “Reinforcement learning for optimal control of low exergy buildings”. In: *Applied Energy* 156 (2015), pp. 577–586. ISSN: 03062619. DOI: [10.1016/j.apenergy.2015.07.050](https://doi.org/10.1016/j.apenergy.2015.07.050).
- [2] ASHRAE. *GUIDELINE 36-2021 – HIGH-PERFORMANCE SEQUENCES OF OPERATION FOR HVAC SYSTEMS*. 2021.
- [3] Ján Drgoňa et al. *All you need to know about model predictive control for buildings*. Jan. 2020. DOI: [10.1016/j.arcontrol.2020.09.001](https://doi.org/10.1016/j.arcontrol.2020.09.001).
- [4] David Blum et al. “Building optimization testing framework (BOPTTEST) for simulation-based benchmarking of control strategies in buildings”. In: *Journal of Building Performance Simulation* 14 (5 2021), pp. 586–610. ISSN: 19401507. DOI: [10.1080/19401493.2021.1986574](https://doi.org/10.1080/19401493.2021.1986574).
- [5] Luis Pérez-Lombard, José Ortiz, and Christine Pout. “A review on buildings energy consumption information”. In: *Energy and Buildings* 40.3 (2008), pp. 394–398. ISSN: 0378-7788. DOI: <https://doi.org/10.1016/j.enbuild.2007.03.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0378778807001016>.
- [6] Yu Zheng, Sen Yang, and Huanchong Cheng. “An application framework of digital twin and its case study”. In: *Journal of Ambient Intelligence and Humanized Computing* 10 (3 Mar. 2019), pp. 1141–1153. ISSN: 18685145. DOI: [10.1007/s12652-018-0911-3](https://doi.org/10.1007/s12652-018-0911-3).
- [7] Cornelius Steinbrink et al. “CPES testing with MOSAIK: Co-Simulation planning, execution and analysis”. In: *Applied Sciences (Switzerland)* 9 (5 2019). ISSN: 20763417. DOI: [10.3390/app9050923](https://doi.org/10.3390/app9050923).
- [8] Michael Wetter. “Co-simulation of building energy and control systems with the building controls virtual test bed”. In: *Journal of Building Performance Simulation* 4 (3 2011), pp. 185–203. ISSN: 19401493. DOI: [10.1080/19401493.2010.518631](https://doi.org/10.1080/19401493.2010.518631).
- [9] Marija Trcka. *Co-simulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings*.

- [10] Drury B Crawley et al. *EnergyPlus: creating a new-generation building energy simulation program*.
- [11] Thierry Nouidui, Michael Wetter, and Wangda Zuo. *Functional mock-up unit for co-simulation import in EnergyPlus*. 2014. DOI: [10.1080/19401493.2013.808265](https://doi.org/10.1080/19401493.2013.808265).
- [12] Michael Tiller. *Introduction to Physical Modeling with Modelica*. Springer New York, NY.
- [13] Karl Johan Åström, Hilding Elmqvist, and Sven Erik Mattsson. *EVO-LUTION OF CONTINUOUS-TIME MODELING AND SIMULATION*.
- [14] Micheal M. Tiller. *Modelica University*. <https://mbe.modelica.university/>.
- [15] Michael Wetter et al. “Modelica Buildings library”. In: *Journal of Building Performance Simulation* 7 (4 July 2014), pp. 253–270. ISSN: 19401493. DOI: [10.1080/19401493.2013.765506](https://doi.org/10.1080/19401493.2013.765506).
- [16] T Blochwitz et al. *The Functional Mockup Interface for Tool independent Exchange of Simulation Models*. URL: <http://www.functional-mockup-interface.org>.
- [17] Michael Wetter, Marcus Fuchs, and Thierry S Nouidui. *Design choices for thermofluid flow components and systems that are exported as Functional Mockup Units*. 2015.
- [18] David Blum et al. *Prototyping the BOPTEST Framework for Simulation-Based Testing of Advanced Control Strategies in Buildings*. URL: <https://github.com/NREL/alfalfa>.
- [19] A Husaunndee et al. *SIMBAD: A SIMULATION TOOLBOX FOR THE DESIGN AND TEST OF HVAC CONTROL SYSTEMS*.
- [20] Michael Wetter et al. *LIFTING THE GARAGE DOOR ON SPAWN, AN OPEN-SOURCE BEM-CONTROLS ENGINE*. 2020. URL: <https://obc.lbl.gov>.
- [21] Kun Zhang et al. “Estimating ASHRAE Guideline 36 energy savings for multi-zone variable air volume systems using Spawn of EnergyPlus”. In: *Journal of Building Performance Simulation* 15 (2 Mar. 2022), pp. 215–236. ISSN: 1940-1493. DOI: [10.1080/19401493.2021.2021286](https://doi.org/10.1080/19401493.2021.2021286). URL: <https://www.tandfonline.com/doi/full/10.1080/19401493.2021.2021286>.

- [22] *RACCOMANDAZIONE (UE) 2019/ 786 DELLA COMMISSIONE - dell'8 maggio 2019 - sulla ristrutturazione degli edifici - [notificata con il numero C(2019) 3352]*.
- [23] Maryam Gholamzadehmir et al. *Adaptive-predictive control strategy for HVAC systems in smart buildings – A review*. Dec. 2020. DOI: [10.1016/j.scs.2020.102480](https://doi.org/10.1016/j.scs.2020.102480).
- [24] Tomáš Uriča and Anna Simonová. “Simulation of an on-off Controller for Systems of Second Order with the Use of LabVIEW”. In: vol. 192. Elsevier Ltd, 2017, pp. 905–910. DOI: [10.1016/j.proeng.2017.06.156](https://doi.org/10.1016/j.proeng.2017.06.156).
- [25] “Review of control techniques for HVAC systems-nonlinearity approaches based on fuzzy cognitive maps”. In: *Energies* 11 (3 Feb. 2018). ISSN: 19961073. DOI: [10.3390/en11030495](https://doi.org/10.3390/en11030495).
- [26] Behrang Alimohammadisagvand et al. “Cost-optimal thermal energy storage system for a residential building with heat pump heating and demand response control”. In: *Applied Energy* 174 (July 2016), pp. 275–287. ISSN: 03062619. DOI: [10.1016/j.apenergy.2016.04.013](https://doi.org/10.1016/j.apenergy.2016.04.013).
- [27] Behrang Alimohammadisagvand, Juha Jokisalo, and Kai Sirén. “Comparison of four rule-based demand response control algorithms in an electrically and heat pump-heated residential building”. In: *Applied Energy* 209 (Jan. 2018), pp. 167–179. ISSN: 03062619. DOI: [10.1016/j.apenergy.2017.10.088](https://doi.org/10.1016/j.apenergy.2017.10.088).
- [28] Liuping Wang. *Basics of PID Control*. 2020. URL: <https://www.researchgate.net/publication/346653954>.
- [29] Xiwang Li and Jin Wen. *Review of building energy modeling for control and operation*. 2014. DOI: [10.1016/j.rser.2014.05.056](https://doi.org/10.1016/j.rser.2014.05.056).
- [30] Qiming Fu et al. *Applications of reinforcement learning for building energy efficiency control: A review*. June 2022. DOI: [10.1016/j.jobe.2022.104165](https://doi.org/10.1016/j.jobe.2022.104165).
- [31] A. I. Dounis and C. Caraiscos. *Advanced control systems engineering for energy and comfort management in a building environment-A review*. Aug. 2009. DOI: [10.1016/j.rser.2008.09.015](https://doi.org/10.1016/j.rser.2008.09.015).
- [32] Gianluca Serale et al. “Model Predictive Control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities”. In: *Energies* 11 (3 Feb. 2018). ISSN: 19961073. DOI: [10.3390/en11030631](https://doi.org/10.3390/en11030631).
- [33] *OPENMODELICA*. <https://www.openmodelica.org/>.

- [34] *Commercial Reference Buildings*. <https://www.energy.gov/eere/buildings/commercial-reference-buildings>.
- [35] *UNI EN 12831 2006*.
- [36] *UNI EN 10339*.
- [37] *Daikin Psychrometrics Diagram*. https://www.daikin-ce.com/en_us/customers/software-downloads/daikin-psychrometrics-diagram-viewer.html.
- [38] *PVGIS*. https://joint-research-centre.ec.europa.eu/pvgis-photovoltaic-geographical-information-system_en.
- [39] *Datasheet of electrical storage*. <https://it.enfsolar.com/pv/storage-system-datasheet/5652>.
- [40] Angela Amato. *Comunità energetiche ed edifici a energia quasi zero: un tipico caso studio*.
- [41] Francesco Isaia et al. "Implementation of Different Control Strategies of Heat Pump Coupled with Thermal and Electrical Storages for a Non-Residential nZEB". In: (). DOI: [10.18462/iir.TPTPR.2021.1979](https://doi.org/10.18462/iir.TPTPR.2021.1979).
- [42] Juliusz Walaszczyk and Aleksandra Cichoń. *Impact of the duct static pressure reset control strategy on the energy consumption by the HVAC system*.
- [43] Mvan den Broek M Zappa WJunginger. "Is a 100 per cent renewable European power system feasible by 2050?" In: (2019).
- [44] *HEAT PUMPS BAROMETER*.

Riconoscimenti

Vorrei ringraziare sentitamente il Prof. Alfonso Capozzoli per aver creduto in me fin dal primo incontro che abbiamo fatto, tenendo sempre alta l'asticella e cercando di ottenere sempre il massimo da me. Inoltre vorrei ringraziare tutto il gruppo BAEDA e in particolare i miei due correlatori Silvio e Davide. Grazie per avermi accompagnato in questo percorso iniziato ad ottobre con il tirocinio e proseguito con questo lavoro di tesi; è stato molto stimolante lavorare con voi e credo di aver imparato molto da questa esperienza.

Un ringraziamento speciale alla mia Famiglia, in particolare ai miei genitori, senza di voi oggi non sarei l'uomo che sono. Spero di ripagare, un giorno, i numerosi sacrifici che avete fatto per me.

Ringrazio mio fratello Roberto, presenza fissa nella mia vita, un aiuto costante, grazie per aver sopportato i miei deliri.

Un ringraziamento a tutti gli amici di "giù", grazie a voi è stato sempre bello tornare, con la sicurezza che noi ci rivedremo sempre a Casa.

Un ringraziamento a tutti i compagni di corso avuti per tutti questi anni, questo viaggio è arrivato al traguardo anche grazie a Voi.

Grazie a tutti gli amici di "su" per avermi fatto sentire a Casa anche a Torino, ora questa città è anche un po' mia.

Infine un ringraziamento speciale a Chiara, accanto a me praticamente da sempre, per tutto il mio percorso universitario, anche e soprattutto quando le delusioni erano maggiori delle gioie. È anche grazie a te se i fallimenti si sono trasformati in traguardi.

Grazie a tutti, vi voglio bene!