

# POLITECNICO DI TORINO

**Corso di Laurea Magistrale  
in Ingegneria Energetica e Nucleare**

**Tesi di Laurea Magistrale**

## **Development and evaluation of an RC grey-box model for an MPC system of an office building**



**Relatore**

Prof. Andrea Lanzini

**Co.Relatori**

PhD Francesco Demetrio Minuto

Dott. Daniele Salvatore Schiera

**Candidato**

Jacopo Roati

Anno Accademico 2019-2020



## Contents

List of abbreviations.....	4
List of figures.....	5
List of tables.....	6
Abstract.....	7
1 Introduction.....	8
2 Materials and Methods.....	11
2.1 Model Predictive Control.....	11
2.2 RC modeling methodology.....	12
2.3 Simulation tools.....	14
2.3.1 EnergyPlus.....	14
2.3.2 Modelica.....	15
2.3.3 ModestPy and PyFMI.....	18
2.4 Case study.....	20
2.4.1 EnergyPlus model.....	21
2.4.2 RC model and parameter learning.....	24
2.4.3 Solar module.....	29
2.4.4 Model training and validation.....	34
3 Results and Discussion.....	36
3.1 Benchmark on the whole heating season.....	39
3.2 Benchmark on weekdays.....	43
3.3 Comparison of the grey-box model with ISO standards.....	46
4 Conclusions.....	49
Appendix A.....	51
Bibliography.....	57

## List of abbreviations

ANN: artificial neural network

FMI: functional mock-up interface

FMU: functional mock-up unit

GA: genetic algorithm

GPS: generalized pattern search

HVAC: heating, ventilation and air conditioning

TPI: Time Proportional and Integrated

## List of figures

Figure 2.1 Scheme of the MPC operation .....	11
Figure 2.2 Scheme of grey-box modelling for a building .....	14
Figure 2.3 Modelica schematics example [9] .....	16
Figure 2.4 Example of a RC building model.....	18
Figure 2.5 Scheme of the method applied to the case study .....	20
Figure 2.6 Geometrical representation of the office building .....	21
Figure 2.7 Occupancy and lights schedule .....	23
Figure 2.8 Heat load from the HVAC system during a whole week of January.....	24
Figure 2.9 schematic of the 3R3C model built on OpenModelica .....	26
Figure 2.10 Sun altitude and azimuth .....	30
Figure 2.11 schematic of the sun component built on OpenModelica.....	32
Figure 2.12 schematic of the 3R3C model with the sun component on OpenModelica ..	33
Figure 2.13 Example of the heat flux destined to the external walls and to the windows calculated by the solar component .....	34
Figure 3.1 Example of a prediction of one day after a learning period of one day. ....	36
Figure 3.2 Example of a prediction of one day after a learning period of three days. ....	37
Figure 3.3 Example of a prediction of three days after a learning period of three days ..	38
Figure 3.4 Comparison between the average RMSEs of one day predictions from Basic model and from Sun model.....	40
Figure 3.5 Average RMSEs of the predictions starting on Mondays.....	41
Figure 3.6 Average RMSEs of the predictions starting on Tuesday .....	42
Figure 3.7 Comparison between the average RMSEs of one day predictions from Basic model and from Sun model.....	43
Figure 3.8 Heatmap of average RMSEs obtained from the validations with the Basic model .....	45
Figure 3.9 Validation of 12 days after a learning period of 12 days .....	48
A. 6 0.1 Heatmap of average RMSEs obtained from the validations with the Sun model. Selected period.....	56

## List of tables

Table 2.1 EnergyPlus model components and materials .....	22
Table 2.2 Descriptions of the parameters utilized in the RC model .....	27
Table 2.3 Upper and lower bound utilized during the parameters learning .....	28
Table 2.4 Lower and upper bounds for ALPHAWall and ALPHAint.....	31
Table 3.1 Computational time for parameters learning .....	46
Table 3.2 RMSE, MBS and Cv(RMSE) of two analytical RC models .....	47
Table 3.3 RMSE, MBS, Cv(RMSE) of the Basic model prediction .....	48

## Abstract

The global warming issue has stimulated the research in studying new solutions to decrease energy consumption. Energy used for building thermoregulation plays an important role in the whole energy consumption in developed countries. Model predictive control is an innovative tool utilized to optimize energy consumption and consequently reducing the production of greenhouse gases. Model Predictive Control exploits a model representing the controlled system to simulate it and use the outputs to optimize the control strategy. Cause the wide variety of building stock there is not a common methodology for modeling buildings and hence the modeling task is the most time demanding in all the process of designing and setting up a model predictive control. This thesis aims to develop and evaluate RC network models with the grey-box modeling method and investigate in some implementations. The models represent an office building that has been designed and simulated on EnergyPlus. Two RC models have been developed on Modelica with three resistors and three capacitors. Grey-box modeling implies that some parameters of the models are estimated by a data-driven algorithm. At this scope, the ModestPy python library has been utilized. An additive component has been added in one of the two RC models to improve the ability of the model to represent the reference building. The function of the implementation is to calculate the solar irradiance components hitting each wall according to the geometry of the controlled building. Each model has been used to predict the internal temperature of the office building for all the heating period of a typical year in Torino. The accuracy of the predictions has been evaluated with the root mean square error. From the results, it is deduced that the implementation in the RC model does not lead to visible improvements.

# 1 Introduction

Energy used for building thermoregulation plays an important role in the whole energy consumption in developed countries. In Europe, according to the European directive 2010/31/EU, buildings consume around 40% of total EU energy consumption and are responsible for approximately 36% of greenhouse gases production [1]. Greenhouse gases are the main cause of global warming and acting on the building energy consumptions can significantly affect their global production. Exploiting renewable energy sources rather than fossil fuels is the first choice to reduce the greenhouse gases but it is not always possible. Other strategies aim to increment the energy efficiency of buildings to reduce energy consumption without reducing the comfort of the people. Increasing the thermal insulation of the walls to reduce the thermal load required is probably the most known strategy for increasing the energy efficiency of a building. Another method is to increase the performance of the control system of the devices present in a building. The controls are responsible for the amount of energy consumed by the devices and increasing their performance means to save energy when it is not necessary. For instance, the HVAC system of a building has its own control that turns it on and off and set the amount of energy to introduce into the building during the time. Nowadays PID (Proportional, Integral and Derivative) controls are the most efficient control systems commonly used in buildings, but recent studies demonstrate that model predictive controls can surpass the performance of PIDs on building devices, such as the HVAC system[2]. Model predictive control (MPC) is a technology widely utilized in some sector like the automotive, but it still rare to find in the building sector but is particularly investigated in the last decade. Model predictive control is a control tool that predicts the state of the controlled system and runs an optimization algorithm of the control signal during the predictions. In the building sector, the input signal can be the heat load of the HVAC system. In this way, optimization of the loads can bring a reduction of energy consumption and greenhouse gas production. One of the main important components of the MPC is the model that makes possible the predictions. Unfortunately, modeling a dynamic system such as a building is not a straightforward task cause the considerable variety of exogenous and endogenous variables in the real building stock and the not existence of an established and common methodology from literature. The big efforts to configure and calibrate MPC models is the main reason this strategy is still not been implemented at any scale.

In literature there are three main ways to model a building. The first one is the white-box approach, where the models can give very accurate predictions, but their modeling needs an in-depth knowledge of the reference building composition, for example, the wall structure and materials, and its behavior. This kind of knowledge is difficult to obtain, especially in old buildings. An example of a white-box model used in an MPC control system is given in the

article [3] where the control model was developed on EnergyPlus [4]. The second big group are the black-box models that are based on time-series data analyzation. There are no physical equations, but the model equations are deduced from either statistic methods or machine learning algorithms. In [5] artificial neural network (ANN) is used to design the control model for an MPC system for a residential house. The third and last method is grey-box modeling, it is a mix between the two previous ones taking the main advantages from them. In the grey-box model there are physical equations, but the physical parameters are obtained by machine learning.

This thesis focuses on grey-box modeling where the models are formulated as a system of a variable number of thermal resistances and thermal capacities, called RC model as abbreviation. For example, in [6] the authors try to find the best level of complexity of the RC system in order to forecast the internal temperature of three real buildings, which are two residential buildings and one office building. The parameters are estimated based on historical data of internal temperature, heat consumption of the building, outdoor air temperature, and global solar radiation on the horizontal surface. Occupancy and internal gains of the buildings are not considered. Since the models represent the building as a single temperature zone, the internal temperature is the weighted average of the various room temperatures. From this study results that the best RC model to represent the buildings is a 4R2C system, however, it has to be considered that slow dynamic characterizes the buildings as the daily fluctuation of internal temperature is at most of 3 K. Article [7] analysis the factors that can affect the accuracy of an RC grey-box model and consequently the MPC performance, such as building design, model structure and model order. EnergyPlus has been used to simulate two reference buildings with one temperature zone, one with lightweight walls and one with heavyweight walls. Three different RC model are tested (1R1C, 3R3C, 5R4C) on both the EnergyPlus models. From the results, it is deduced that the thermal envelope of the building affects the model accuracy considerably. Indeed, the RC model is more precise when the reference building has heavyweight walls than lightweight walls because of the slower dynamics of the system in the first case due to the greater thermal mass.

From literature, it appears that building with fast dynamics could be more difficult to represent with an RC model. The accuracy of the predictions is crucial in MPC system because the optimization is compiled based on the ability of the model to represent the controlled system. In addition to the accuracy, also the length of a reliable prediction is a factor consider. The core of this thesis is, precisely, applied the knowledge of the grey-box modeling approach for representing an office building and investigating in some methodologies to improve the accuracy of the model. The typical strategy to improve the accuracy of an RC model is to vary the number of thermal resistances and capacitances losing sometimes the physical representation of the reference building. For this reason, a new strategy is tested in this work. The strategy tested in the thesis is to calculate the various components of solar radiation

according to the building geometry and location in order to give the model more accurate information.

The remainder of the thesis is structured as follows. Chapter 2 describes the method used in this work in order to design and test the building model, the method is followed by a description of the tool actually exploited. Chapter 3 presents the graphical results of the work lingering on the accuracy of the models prediction, after the results are described they are discussed to explain them and finally Chapter 5 summarize the main results of the thesis.

## 2 Materials and Methods

The necessity of reducing energy consumption and greenhouse gases emission has become crucial nowadays to contrast global warming. The energy consumed in developed countries for the building thermal regulation covers more than the third of all consumptions. A work of efficiency in buildings can significantly impact on the total energy consumptions and greenhouse gases production. The efficiency can be done improving the structure, for example, increasing the thermal resistance of the external walls, or avoiding wastes of energy. The two paths can be done simultaneously. This thesis focuses on the second one, the optimization of energy utilization in buildings, especially by the HVAC systems. HVAC systems are commonly controlled by temperature-gradient controls or TPI (Time Proportional and Integrated) controls. In recent years Model Predictive Controls have become more interesting in the buildings field because it has been observed that those controls can decrease the energy consumption of the HVAC system comparing the traditional control tools. This thesis studies some aspects of the development of a Model Predictive Control concerning its model.

### 2.1 Model Predictive Control

Model Predictive Control (MPC) is a control algorithm widely used to control dynamics systems. The MPC is composed of two main components: the model representing the controlled system, and an optimization tool. Figure 2.1 shows the scheme of the operation of the MPC.

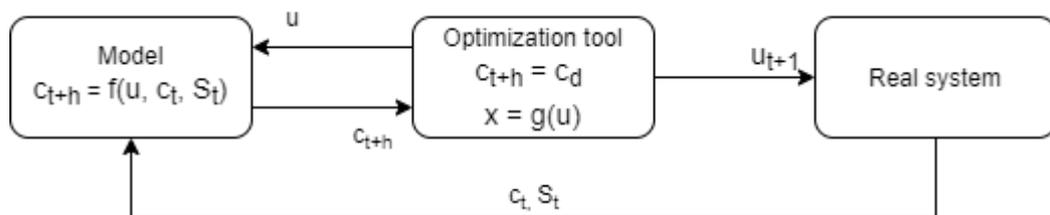


Figure 2.1 Scheme of the MPC operation

The model predicts a variable  $c$  of the reference system after a short period  $h$ , that variable depends on the input variable  $u$ , the initial state of the reference system, which includes the variable  $u$  and all other variables  $S$ . The job of the optimization tool is to determine the best value of the input variable  $u$  in order to obtain the desired value of the variable  $c$  in the model, minimizing the variable  $x$ . At a certain time sample  $t$  the model is initialized with  $c_t$  and  $S_t$  obtained from the reference system, then the optimization tool simulates the model with various input variables  $u$  with the aim to satisfy  $c_{t+h} = c_d$ ,  $c_d$  is the desired value of the variable  $c$ . The variable  $u$  is not a single value but it is the array containing a value for each time sample from  $t$  to  $h$ . The optimization tool transmits the controlled signal

for the time sample  $t+1$  to the reference system after having found the best signal  $u$  which minimize the variable  $x$ . The process is repeated at every timestep, in this way the model is always updated with the real variables of the controlled system and the optimization tool can correct its control signal according to the reference system feedbacks.

In the building sector model, the predictive controls have been finding their application for regulating the HVAC systems and for managing the energy and heat production and storage in multi-energy systems. For example, referring to Figure 2.1, the reference system is the building with its HVAC plant,  $c$  stands for the internal temperature of the building and  $u$  for the signal to control the HVAC system. The optimization tool with the model finds the value of  $u$  in order to satisfy the thermal comfort of the people in the building and minimize  $x$ , that can stand for the energy consumption or energy cost of the building. It is deducible that the ability of an MPC to provide good controls depends on the accuracy of the predictions of the model. Modeling a building and its components is still very demanding and for this reason the penetration of MPC in buildings is slower than in other fields [8]. The model itself is not the only issue because buildings dynamic is strongly affected by ambient temperature and sun irradiation, meaning also these predictions shall be accurate. The building model uses the forecast of the atmosphere condition of the whole time horizon ( $h$ ) to predict the controlled variable. The length of the time horizon depends on the scope of the MPC, for example, for an MPC designed for daily thermal storage in a building the time horizon will be of a day. The model has to be designed to satisfy a certain accuracy during the whole prediction on the time horizon. Greater is the accuracy greater is the benefit of using the MPC.

## 2.2 RC modeling methodology

The model intended for a MPC can be of various kinds, both physical or empirical, but it is preferred as simple as possible in order to have a low computational cost and be able to be exploited by the optimization tool. For buildings, grey-box modeling is very suitable for designing models for MPC systems because it consents to keep the model simple guaranteeing a good accuracy of the predictions. Moreover, with grey-box modeling, there is no need to know all the widely amount of details of the building structure, which is very difficult to obtain from an already existing building. In fact, the physical properties of the reference building are lumped in some parameters, which are estimated by a machine learning algorithm. The only requirement to model a building is to provide the building with some sensors, if not already present, in order to produce a historical data series of some variables, usually internal temperature and heating load from the HVAC system are enough. Summing up grey-box modeling is composed of two steps. The first one is the physical modeling of the building; the physics of the building is kept as simple as possible and its equations are characterized by a finite number of parameters. The second step is to determine the values of the parameters

from the historic data series of the modeled building and other data relative to the external ambient thanks to a data-driven algorithm.

The work of this thesis follows the two previous steps of grey-box modeling. The RC network analogy has been adopted to model buildings. It derives from the electrical circuit theory using the knowledge of resistors and capacitors. In the analogy, the electrical units are exchanged with thermal units. The four electrical physical quantities that characterize an RC circuit are electrical resistance, capacitance, voltage and electric current and become, respectively, thermal resistance, heat capacitance, temperature and heat flux. This modeling methodology simplifies the description of a spatially distributed system, a building, into a discrete system of elements which are the thermal resistors and the heat capacitors; each of those characterized by an equation. Equation 1 and Equation 2 describe the thermal resistor element and the heat capacitor element, respectively.

$$\Delta T = R * q \quad (1)$$

$$\frac{dT}{dt} * C = q \quad (2)$$

where  $\Delta T$  is the difference of temperature at the ends of the resistor,  $R$  is the thermal resistance,  $q$  is the heat flux,  $\frac{dT}{dt}$  is the temporal derivative of the temperature in the heat capacitor and  $C$  is the heat capacity.

The complete model is presented as a system of the two previous equations plus the boundaries conditions.  $R$  and  $C$  are the parameters that characterize the components of the model and represent the physical property of the elements in the reference building. For example, a wall and its distributed heat capacity can be represented by a heat capacitor with the proper heat capacity. The proper value to assign to a parameter can be identified with a data-driven algorithm.

The procedure of building and setting an RC model is an iterative process. Figure 2.2 shows the iterative process. It starts with collecting the data from the building sensors and external environment; the most used data are the ambient temperature and the internal zone temperature, the horizontal solar irradiance, and thermal loads of the HVAC and the internal heat gains. A first model is designed with a certain number of resistors and capacitors, each of them characterized by a parameter. Then the machine learning algorithm use the collected data in a certain period to estimate the RC model's parameters. Subsequently, the model is set with the new estimated parameters and it is validated and tested on a different period than the learning one. The validation consists of comparing the outputs obtained from the model with the relative same variables from the building data collected in the validation period and corresponding to the same time sample. Then if the accuracy of the predictions of the model satisfies the control system requirements the process stops, otherwise the process restarts at the model designing. The model structure gets changed, for example adding new

resistors or capacitors and the new RC model is subjected again to the machine learning algorithm to estimate the parameters. Then a new validation follows and the goodness of the model is questioned and the loop can restart another time or end.

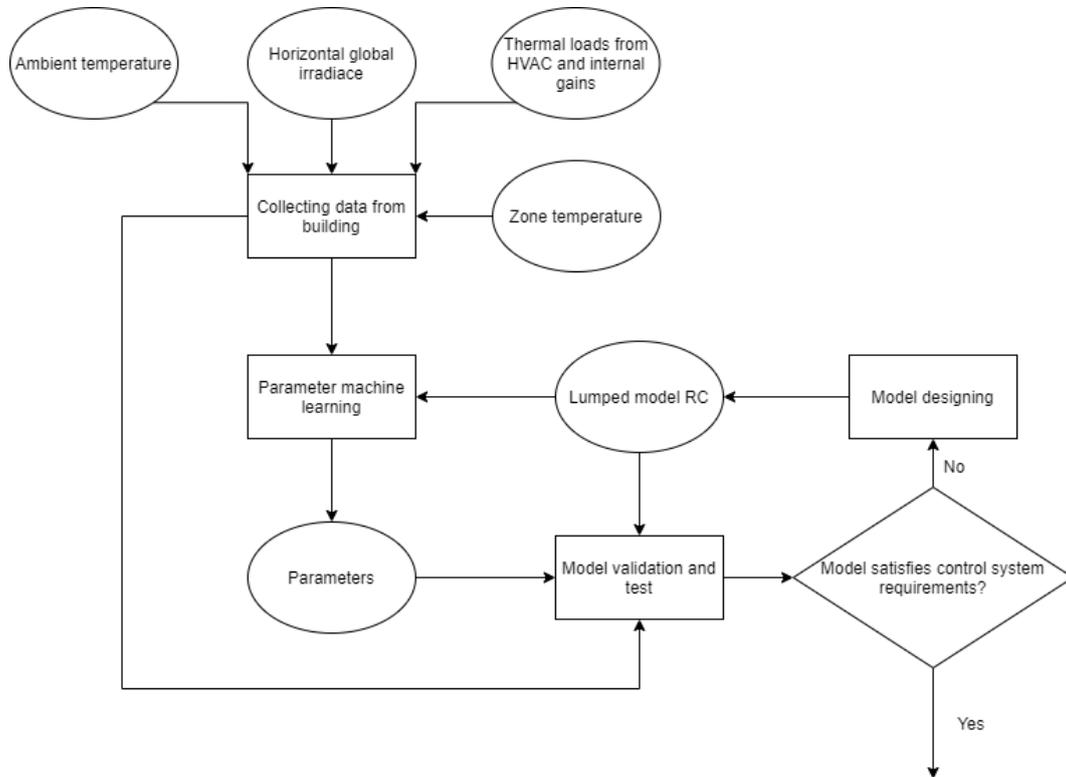


Figure 2.2 Scheme of grey-box modelling for a building

## 2.3 Simulation tools

This paragraph presents the tools used in this thesis in order to simulate a reference building, designing an RC model and estimate its parameters. The tools here described are EnergyPlus for simulating the building, Modelica for modeling the lumped model, and ModestPy with PyFMI to training and simulate the lumped model.

### 2.3.1 EnergyPlus

EnergyPlus can be used to simulate a realistic building and collect the many variables from simulations. EnergyPlus is an energy analysis and thermal load simulation program for the dynamic of buildings and it can be used through its user interface EP-Launch on Windows or any other third-party interfaces. Its development is funded by the U.S. Department of Energy's (DOE) Building Technologies Office (BTO). EnergyPlus was initially written in FORTRAN90, but

after 2014 it is all converted to C++. Its structure aims to be object-oriented and modular for facilitating the use [9].

The starting point in an EnergyPlus modeling process is the building description that begins with the geometrical structure. It is proven that the geometrical structure does not have to be very precise to have a valid final result in the simulation. The EnergyPlus guide suggests which assumptions can be made regarding, for example, windows and internal walls[9]. What has to be well defined is if a building structure is a heat transfer surface which divides two ambient with different temperature, like the roof, the floor and the external walls, or a heat storage like the internal walls if they do not separate two thermal zones. Every construction component must be characterized by a sequence of materials with their physical properties, EnergyPlus has many datasets that can be useful for this task. Moreover, there are also databases for setting the time schedule of utilization of some device such as the HVAC system or the building lights.

An important aspect to consider in the building model is the thermal zoning since EnergyPlus performs every calculation in relation to a thermal zone. A thermal zone is defined as a volume of air at a uniform temperature, a good practice is to zoning the area according to the HVAC systems, a whole area served by the same system is a thermal zone, and temperature setpoints, if two rooms are served by the same HVAC system but they have two different temperature setpoints then they are different temperature zones. Another factor that can affect the zoning is sun exposure, e.g., a region of the building is reached by more sun radiations it can lead to an increase of temperature.

Once the building description is completed, any other module can be implemented such as the zone equip module, the windows glass module and air loop module. For simulating the model, EnergyPlus requires the weather information of the building location, the proper format of data file is epw. The final result is obtained by the interaction between two main simulation loops: the heat and mass balance simulation give some conditions to the building systems simulation giving feedback to update the heat and mass balance. From the simulation, accordingly to the modules used, it is possible to obtain the temperature of each temperature zone by the time, the power used for heating and cooling, the air humidity, the occupancy comfort provided by temperature, humidity and lights, and any other information a module can calculate.

### 2.3.2 Modelica

Modelica is a free objective-oriented modeling language developed by the Modelica Association, a non-profit organization located in Sweden[10]. Modelica aims to model complex and heterogeneous systems, and the Modelica Standard Library contains about 1600 model components with about 1350 functions from many domains. The libraries can be developed by anyone and shared.

Models are visually described as schematics, components linked between each other, examples can be found in Figure 2.3. The connections are constructed from the ports of each component. Internally, a component is described by another schematic or by some equations, the ports are the inputs and outputs of the internal structure. The syntax of a Modelica model is divided into two main parts, the first one is where everything inside the model is declared such as variables, parameters and models. The second one, introduced with the word “equation”, is where the equations of the model can be placed, involving parameters and variables, or where the connections can be stated.

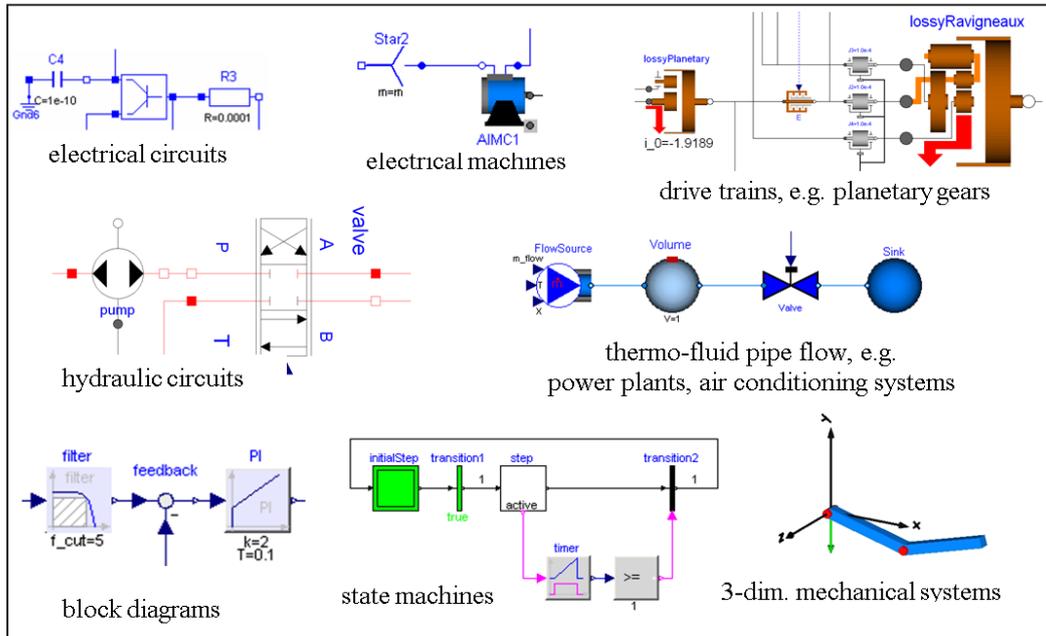


Figure 2.3 Modelica schematics example [10]

An essential characteristic of Modelica is that equations allow acausal modeling. From a practical point of view, it means that an equation can be read either from left to right or vice versa, not like in programming language where an equation is a statement which describes what the element on the left-hand is by an expression, but this element cannot be used to determine an element on the right. Indeed, on the left side of an equation on Modelica can be present even an expression. Moreover, for this reason, equations can be placed in any order and not in the order they are expected to resolve because they are solved as a system. This aspect makes a model reusable for every situation when its equations are needed.

There are many available Modelica simulation environments, both commercially and free of charge, to graphically edit and browse a Modelica model and to perform model simulations and other analysis. OpenModelica[11] is a free Modelica environment and it has been used in this thesis to design RC models of a building.

With Modelica, it is easy to represent and build the RC network thanks to the graphical schematic. Each component is a model with ports in order to connect one model to another and communicate the values of temperature and heat flux. The thermal resistor is a box with two ports and a parameter for the thermal resistance, the three equations which describe the model are the following:

$$T_a = T_{port.a} \quad (3)$$

$$T_b = T_{port.b} \quad (4)$$

$$T_a - T_b = R * q \quad (5)$$

where  $T_{port.a}$  and  $T_{port.b}$  are the temperatures of the two ports a and b,  $R$  is the thermal resistance and  $q$  is the heat flux.

The heat capacitor is characterized by only one port, the thermal capacitance parameter and three equations, which are:

$$T = T_{port} \quad (6)$$

$$derT = \frac{dT}{dt} \quad (7)$$

$$C * derT = q_{port} \quad (8)$$

where  $T_{port}$  and  $q_{port}$  are respectively the temperature and the heat flux indicated by the port.

**Error! Reference source not found.** shows a simple example of an RC model of a building composed of one thermal resistance and one thermal capacitance plus other components. The number of resistor and capacitor determine the order of the model, as shown in Figure 2.4 can be called 1R1C. The components on the picture are, in order from left to right, an input port ( $T_{amb}$ ) which is connected to a block ( $T_{ambient}$ ) that convert the scalar input values in temperature values, *ThermalResistor* express the resistance between the outdoor ambient and the building and it can be identified as the building walls, but with no heat capacity. All the thermal capacity of the building is lumped in the thermal capacitor block (*ThermalCapacitor*), *Tsensor* is a component that reads the temperature of the thermal capacitor port and transmits it to an output port ( $T_{zone}$ ). There can be any number of resistors and capacitors and other components such as heat flux block that allow a specific amount of heat flow rate to be injected in a port. Every link in the schematic represents an equality equation between the thermal quantities of the connected ports. It is important to mention that in an RC model all the heat transfers are linear because they are expressed only by the thermal resistance, while in reality, heat exchanged by radiations is a non-linear phenomenon. Despite that, a representation of a building with only linear heat transfers has been demonstrated that it is able to well approximate the reference building. The radiation heat exchanges behave similarly to the linear heat exchanges because of the low temperature difference between the building structures [12].

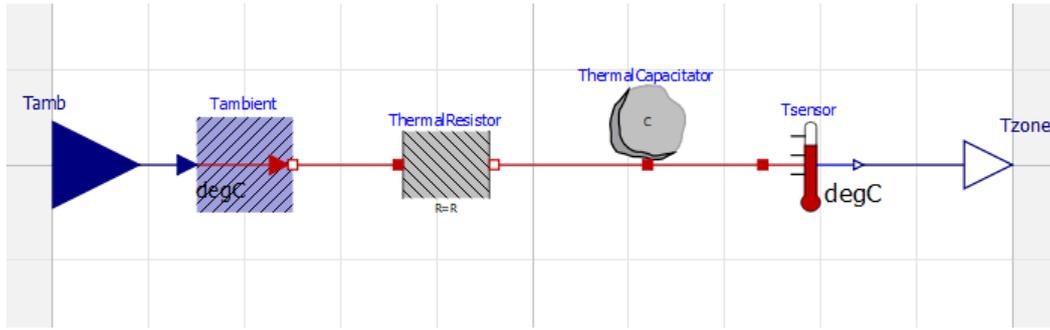


Figure 2.4 Example of a RC building model

RC models can be very complicated, but for an MPC purpose, the design is kept as simple as possible just for having a satisfying prediction of the controlled variables. The complexity of the model increases the computational time especially for the optimization of the inputs that is compiled every timestep.

### 2.3.3 ModestPy and PyFMI

ModestPy [13] is an opensource Python tool developed for estimating parameters in FMI-compliant models through an optimization problem. FMI stands for Functional Mock-up Interface and it is a standardized and unified model execution interface with the scope to allow model exchange and co-simulation of dynamic models. Models can be designed in many programming languages or simply built in different software and hence the interaction between them is not always possible. With FMI standard a modeling tool is able to generate and export a c/c++ code representing the dynamic model as a box with inputs and outputs [14]. This package of codes is called functional mock-up unit (FMU) and it can be integrated into a simulation environment and used as a proper model. With FMI 2.0, which is the latest version, an FMU set Model-Exchange always has a solver inside to compute the time equations without any specific library in the simulation tool. On Python, FMUs can be load and execute through the opensource package PyFMI [15] and it was integrated into ModestPy.

During the optimization process, ModestPy operates with a cost function that relates the measured data of the reference system and the data calculated by the model with the estimated parameters. Cost function can be calculated using either the total mean-square error ( $MSE_{tot}$ ) or the total normalized mean-square error ( $NMSE_{tot}$ ), calculated as follows:

$$MSE_{tot} = \sum_i \frac{\sum_{t=1}^N (\hat{Y}_i^t - Y_i^t)^2}{N}, \quad (9)$$

$$NMSE_{tot} = \sum_i \frac{MSE_i}{\bar{Y}_i^2}, \quad (10)$$

where  $\hat{Y}_i^t$  is the measured value of the variable  $i$  at time step  $t$ ,  $Y_i^t$  is the simulated variable  $i$  at time step  $t$ ,  $\bar{Y}_i$  is the mean measured value of variable  $i$ ,  $N$  is the number of time steps and  $MSE_i$  is the mean-square error for variable  $i$ .

MSE is suggested for models with one output or with comparable physical outputs, for the rest of cases, NMSE is the more indicated.

ModestPy has two algorithms implemented for the estimation of the parameters that are a genetic algorithm (GA) and a generalized pattern search (GPS), also it can interface with some algorithms from the ScyPy[16] eco-system. GA is a metaheuristic global search algorithm that allows ModestPy to make the estimation of the parameters with no assumptions of the model and hence make ModestPy easier to use. The genetic algorithm is an algorithm inspired by natural selection and it is based on operations of tournament-based selection and crossover. Each parameter to estimate is coded in binary as a string of 0s and 1s. In the optimization problem, in every iteration, those strings get mutated and then selected the ones which lead to a minor cost function. The algorithm stops when the maximum number of iterations is reached or when the cost function does not decrease. Since GA does not guarantee that the global optimum is found as its heuristic characteristic, it is coupled to a GPS algorithm to refine the solution. GPS is a simple gradient-free local search method. The algorithm starts from the values obtained from the GA optimization and performs a series of orthogonal explorative moves of a pre-selected step-size by changing one parameter at a time and evaluating the cost function. If the given step size does not reduce the cost function value, it is reduced by a factor of 2 and the procedure is repeated. The algorithm stops when the procedure does not improve anymore the cost function value[13]. The GPS algorithm, as the GA algorithm, does not require assumptions on the model because it does not need a gradient and hence the model equations can be even not differentiable or continuous.

ModestPy is specifically designed to facilitate the estimation of the parameters in building and HVAC models for the purpose of MPC. Thanks to its algorithms and structure ModestPy is very user-friendly and it allows estimate parameters on a very large kind of models, always coded as an FMU, without known precisely how they are designed. ModestPy, as already mentioned, has to be used on Python, both 2.7 and 3 are good, and the user needs to interact with only one class. This class is called *Estimation* and has two methods inside which are *estimate()* and *validate()*. The first one is the method related to the parameters estimation and it needs five arguments that are as follows: the path of the working directory where all the results will be saved, the FMU path, a Pandas[17] DataFrame of the model inputs, a dictionary with the known parameters and their values, a dictionary with the parameters to estimate and their bounds, and finally a Pandas DataFrame of the known values obtained from the model with the previous inputs. Some optional arguments can be added to the class like the max iterations and tolerance of both GA and GPS and the method for calculating the cost function. The parameters are given as a dictionary. With *validate()*, instead, a validation of the

parameters can be done, the starting and final time of the validation have to be inserted in the *estimate()* method as an argument. It returns the validation error as the cost function and the Pandas DataFrame of model outputs.

## 2.4 Case study

The case study presented in this thesis follows the procedure introduced in the Method paragraph. Figure 2.5 shows how the indications in the main blocks of Figure 2.2 have been applied in the case study underlining the tool used and the outputs obtained from each of them. The data of a building have been decided to be obtained from the simulation of a model in EnergyPlus. The model is simulated through the EnergyPlus interface EP-launch. Using a EnergyPlus model instead of a real building make the acquisition of the data much faster and avoids problems such as the lack of values. Moreover, from EnergyPlus is possible to know every detail of the model to make some eventual verification. The data generated by the simulation will be used during the parameter machine learning and the model validation and test. The model design of the lumped model has been done on OpenModelica utilizing the Modelica language. From this step, it has been produced the RC models used in the thesis and their relative FMUs. The machine learning of the RC model parameters have been done with the Python library ModestPy and the script has been written and compiled on PyCharm[18], which is an interface for working with Python. ModestPy works with the FMU of the RC model. From this step, the parameters of the lumped model have been estimated. The last step consists of the validation end testing of the RC model. The FMU of the RC model is set with the estimated parameters and then it is simulated using some of the data from the EnergyPlus model as inputs and some other as references to compare with the results of the RC model simulation.

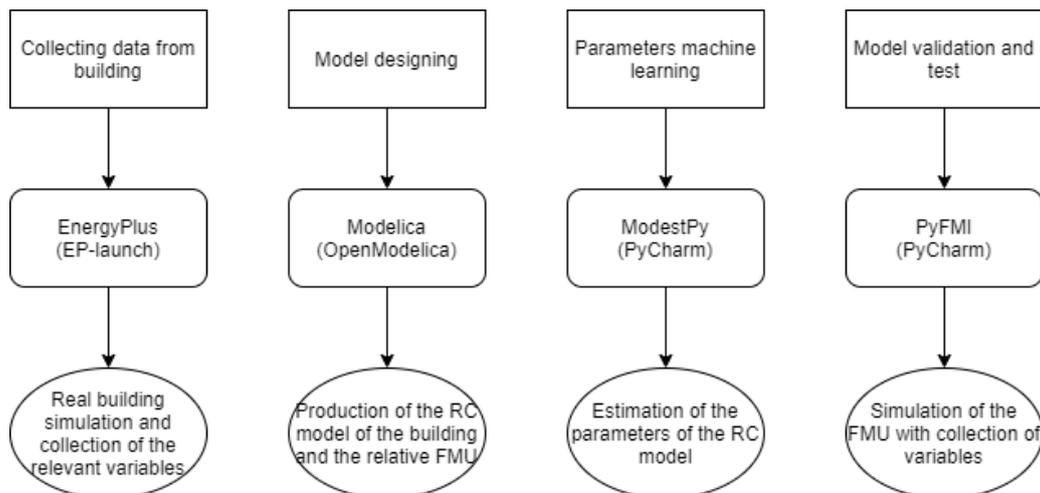


Figure 2.5 Scheme of the method applied to the case study

### 2.4.1 EnergyPlus model

This model wants to represent a single-storey office building of 1050 m<sup>2</sup> of area. The aim of the model is to simulate a real office building and to collect the variables needed to estimate the parameters of a relative RC model. The building is designed with two window facades on west and east side, it is visible in Figure 2.6. The whole area is divided into three thermal zones: the two big rooms with the façades and the hall in between. Zone 1 is the room on the east side, Zone 2 is the one in the middle and Zone 3 is the room with the window on the west side. Some internal mass has been added to the EnergyPlus script to represent the walls which divided Zone 1 and Zone 3 into two and three spaces, respectively. The roof of the building has been set as a surface not hit by sun and windows. Table 2.1 shows the constructions used in the EnergyPlus model with their material compositions and the relative physical properties.

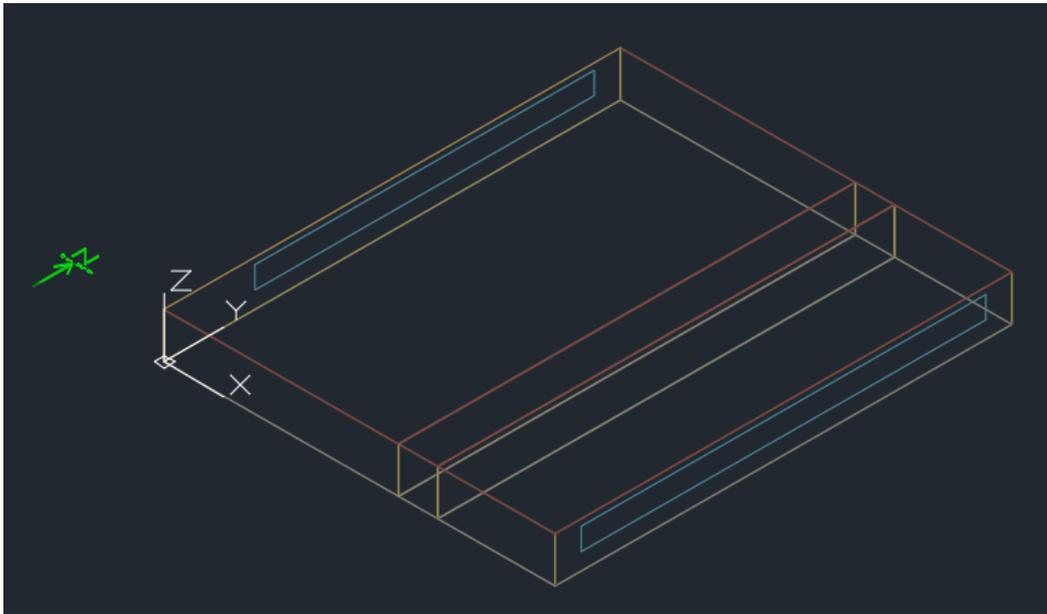


Figure 2.6 Geometrical representation of the office building

	<b>Materials</b>	<b>Thickness [m]</b>	<b>Density [kg/m<sup>3</sup>]</b>	<b>Conductivity [W/m-K]</b>	<b>Heat capacity [J/kg-K]</b>
<b>Windows</b>	Clear	0.003		0.9	
	Air space*				
<b>Exterior walls</b>	Clear	0.003		0.9	
	Brick	0.1016	1920	0.89	790
	Insulation	0.0508	43	0.03	1210
	Air space*				
<b>Floor</b>	Gypsum	0.019	800	0.16	1090
	Medium concrete	0.2032	800	1.11	920
	Acoustic tile	0.0191	368	0.06	590
	Air space*				
<b>Roof</b>	Heavy concrete	0.1016	2240	1.95	900
	Wood	0.0254	608	0.15	1630
	Heavy concrete	0.1016	2240	1.95	900
	Air space*				
<b>Partition walls</b>	Insulation	0.0508	43	0.03	1210
	Acoustic tile	0.0191	368	0.06	590
	Gypsum	0.019	800	0.16	1090
	Air space*				
	Gypsum	0.019	800	0.16	1090

\*For the air space EnergyPlus assigns only a thermal resistance [ $m^2K/W$ ] of 0.15

Table 2.1 EnergyPlus model components and materials

The model presents two internal gains: one is due to the occupancy in the building and the other one is due to the lights. These gains are set accordingly to two schedules taken from the EnergyPlus dataset which simulate the presence of people and the utilization of lights during a week in a typical office. The schedules give the percentage of a total value per every time interval. The total value is calculated by EnergyPlus in function to some parameters set. For the lights, the parameter *watt per zone floor area* has been set to 3 W/m<sup>2</sup> for Zone 1 and Zone 3, i.e. the office area, while a value of 1 W/m<sup>2</sup> has been set for Zone 2, i.e. the hall. For the occupancy, the parameter *number of people* must be set for every zone, here, in Zone 1 it has been set to 20, in Zone 2 to 5 and in Zone 3 to 40, with the number of people and their activity value, 100 W/person, EnergyPlus can calculate the total occupancy value. There are many other parameters that can be changed, but it has been decided to leave them as they were already configured. Figure 2.7 shows the values of the internal gains due to people presence and light utilization in a week accordingly to the office schedules.

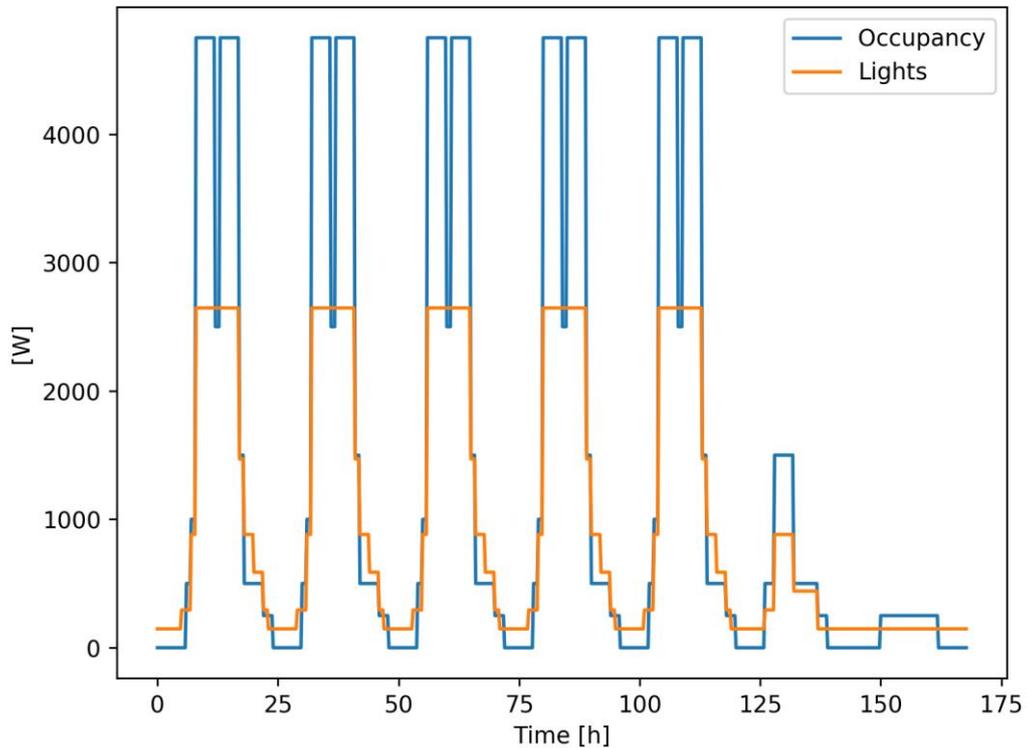


Figure 2.7 Occupancy and lights schedule

The building is also characterized by an HVAC system that has been implemented in EnergyPlus. A temperature setpoint of 21 °C has been chosen for Zone 1 and Zone 3 during the heating period, while a setpoint of 18 °C for Zone 2. The HVAC template chosen for the scope of this model is the *Ideal Load Air System* that simulates an air system able to meet the heating and cooling load of the building in order to maintain the setpoint temperature. For every timestep, the system calculates the thermal load to reach the setpoint taking into consideration all the other phenomenon of heat exchange that occurs during that timestep. The HVAC system works accordingly to another schedule that states the on/off status of the system in an office building. All the other parameters of this template have been left as default.

Finally, the simulation has been run with the weather file of Torino, used in the demo as well. The variables collected from the simulation is each zone air temperature, each zone internal gains (lights and occupancy), each zone sensible heating and cooling energy, and outside ambient air-dry bulb temperature. The variables are collected with a timestep of 900 seconds. Figure 2.8 presents the data of the sensible heat coming from the HVAC system during the first week of January. According to the office HVAC schedule of EnergyPlus, the HVAC is on from 6 am to 10 pm.

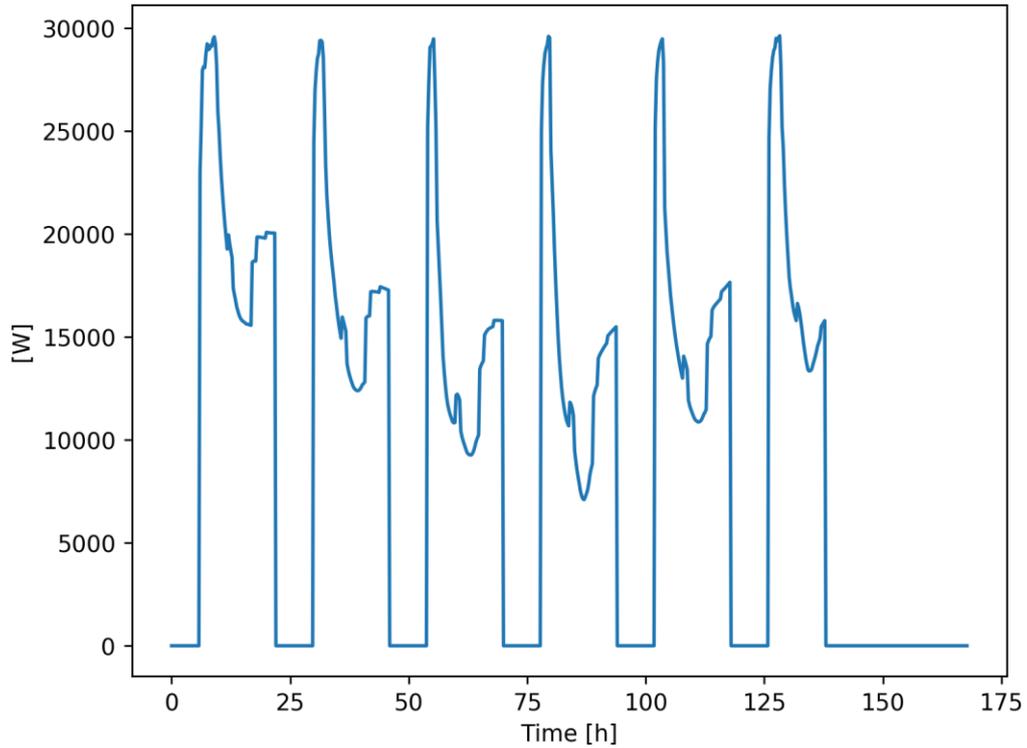


Figure 2.8 Heat load from the HVAC system during a whole week of January

#### 2.4.2 RC model and parameter learning

In literature, many authors state that a 3R3C model is often able to represent the dynamic of a building[19], for this reason, the first model chose to represent the EnergyPlus model is exactly a 3R3C. Moreover, the 3R3C representation of the building is well representative of the building structure and the exchanges of heat that involve them, as described below. The RC model is structured with three heat capacitors that represent the external walls, the internal air of the building and the mass inside the building, for example, the internal walls for dividing the rooms. The three thermal resistors stand for the heat resistance between the external ambient and the external walls, between the external walls and the internal air and between the internal air and the internal mass. The thermal resistors express the only possible heat exchanges that the model considers. Figure 2.9 shows the 3R3C model designed on OpenModelica, other than the capacitors and resistors there are other five inputs interfaces. The interfaces allow the input of the variables: ambient temperature, horizontal solar irradiance, heat gain from building occupancy, heat load of the HVAC system and heat gain from the building lights. The blocks with two red arrows are simply modules to convert a scalar value to Watt. The horizontal sun irradiance is split into two parts, one is directed to the

external walls capacitor, as in reality the part that hit the external walls. The other one is connected to the internal mass capacitor referring to the radiation passing through the windows and heating the internal mass of the building. Before the external walls capacitor and the internal mass capacitor, there are two blocks that multiply the irradiance for a certain factor. The HVAC thermal load is connected directly with the internal air capacitor as the heat is immitted in the building through an air system and so almost immediately exchanged with all the internal air. The occupancy thermal gain is connected to the internal air capacitor because it has been hypothesized that the heat exchanged is latent heat and there is not a radiative component. At the same time, the lights heat gain is connected to the internal mass capacitor because the heat is only in the form of radiations. A red thermometer is linked to the internal air capacitor, which is its temperature sensor and after that, there is an output interface. The Tzone output stands for the building variable which corresponds in the EnergyPlus model to the internal building temperature. Since the EnergyPlus model has three temperature zones and therefore three temperatures variables, a single value has been obtained from them doing a weighted average of the temperatures on the area of each zone.

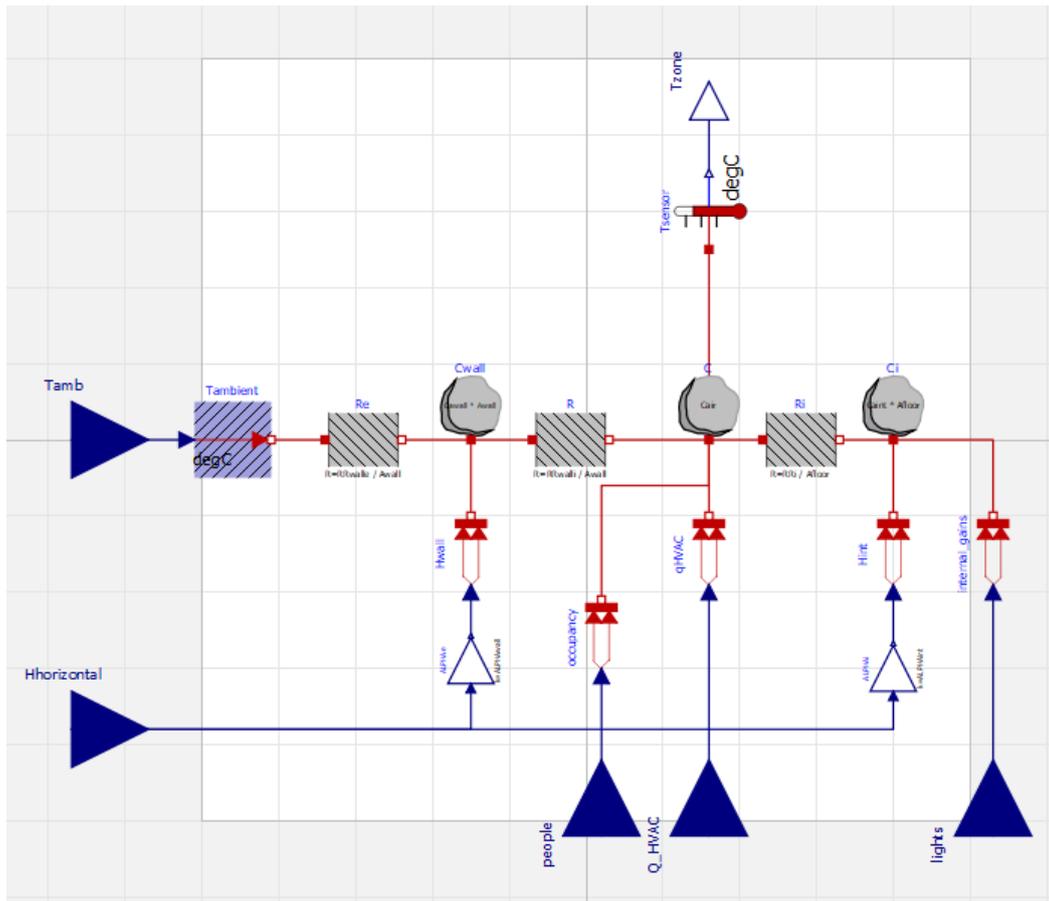


Figure 2.9 schematic of the 3R3C model built on OpenModelica

The model presents a total of eight parameters that are tuned to vary in order to let ModestPy determine them. The parameters are the two factors relative to the solar irradiance and three thermal resistances and three heat capacities. All the resistances and capacitances, excepted for the internal air heat capacitance, are related to the area of the walls or to the floor area. In this way it is easier to set the higher and lower boundaries during the estimation of the parameters and check if their values are physically possible. Table 2.2 contains all the parameters to estimate in the 3R3C model and a relative brief description.

	<i>Description</i>
<b><i>RRwalle</i></b>	Thermal resistance between the external ambient and the external walls. It is related to the area of the external walls [ $m^2K/W$ ].
<b><i>RRwalli</i></b>	Thermal resistance between the external walls and the internal air. It is related to the area of the external walls [ $m^2K/W$ ].
<b><i>RRi</i></b>	Thermal resistance between the internal mass and the internal air. It is related to the floor area [ $m^2K/W$ ].
<b><i>Cawall</i></b>	Thermal capacity of the external walls on the area of the external walls [ $J/K-m^2$ ].
<b><i>Cair</i></b>	Thermal capacity of the internal air volume [ $J/K$ ]
<b><i>Caint</i></b>	Thermal capacity of the external walls on the area of the external walls [ $J/K-m^2$ ].
<b><i>ALPHAwall</i></b>	External walls absorption factor per area [ $m^2$ ].
<b><i>ALPHAint</i></b>	Internal mass absorption factor per area [ $m^2$ ].

*Table 2.2 Descriptions of the parameters utilized in the RC model*

On ModestPy the parameters dictionary must contain a hypothetical starting value, a lower limit and an upper limit. The starting value is not important for the success of the estimation, it can also be neglected in the ModestPy script. The boundaries, on the contrary, are very useful and it is proven that the more they are narrow more the prediction of the model is accurate, always if in the boundaries there are the optimum values for the parameters[7]. Find the right boundaries is not a deterministic procedure and it can take some attempts. Thermal capacities in the RC models can be easily referred to some thermal capacity of the EnergyPlus model, or of the reference building. For example, the capacity of the external wall capacitor in RC model can be representative of the whole capacity of all external walls in EnergyPlus model and hence its value should be similar to the EnergyPlus one. For the resistances, this reasoning cannot be done because the resistances in RC models are not representative of some identified resistances in the EnergyPlus model. The approximate values of the EnergyPlus capacitances have been calculated in the following way. For every component, the total thickness has been calculated as the sum of the thickness of every material layer in that component. The density and specific heat of each component has been calculated with a weighted average of the density and specific heat of every material on the total thickness. At this point, the total area of all external walls has been multiplied for the external wall thickness, for the external wall density and for the external wall specific heat

giving the total heat capacity in Joule on Kelvin. Same has been done with the floor, the roof and the internal walls. Equations 11, Equation 12, Equation 13 and Equation 14 explain the calculation made before.

$$t_c = \sum_i t_i \quad (11)$$

$$d_c = \sum_i \frac{d_i * t_i}{t_c} \quad (12)$$

$$cp_c = \sum_i \frac{cp_i * t_i}{t_c} \quad (13)$$

$$C_c = A_c * t_c * d_c * cp_c \quad (14)$$

where  $t_i$  is the thickness of each material layer,  $t_c$  is the thickness of the component,  $d_i$  is the density of each material,  $d_c$  is the density of the component,  $cp_i$  is the specific heat of each material,  $cp_c$  is the specific heat of the component,  $C_c$  is the heat capacity of the component and  $A_c$  is the component area.

The sum of external walls and roof heat capacity divided by external walls and roof area has been assumed as the heat capacity on the area of the external walls capacitor in the 3R3C model, while internal mass and floor give the heat capacity on the area of the internal mass capacitor. The external walls area is 455 m<sup>2</sup>, the internal mass area is 560 m<sup>2</sup> and the roof or the floor are 1050 m<sup>2</sup> each. It needs to be noticed that the values just calculated are an approximation because calculated with approximative calculations. For example, the volume is calculated without considering where the walls intersect each other. Their scope is just to give a reference of the order of magnitude of the parameters to set proper limit values in the ModestPy script. Table 2.3 shows the lower and upper limits utilized for each parameter.

	<b>Lower limit</b>	<b>Upper limit</b>
<b>RRwalle</b>	0.1	8.0
<b>RRwalli</b>	0.1	8.0
<b>RRi</b>	0.01	3.0
<b>Cawall</b>	1e4	1e6
<b>Cair</b>	1e5	1e7
<b>Caint</b>	1e4	1e6
<b>ALPHAwall</b>	0	60
<b>ALPHAint</b>	0	40

Table 2.3 Upper and lower bound of each parameter utilized during the parameters learning.

Thermal resistances bounds are taken similar to the limits in the article [7] where there is a 3R3C model, same for *ALPHAwall* and *ALPHAint*. *Cawall* and *Caint* bounds are chosen to be one order of magnitude lower and greater than the values estimated previously. Also for *Cair* the range is of one order of magnitude around the approximated value, which is calculated

from the total volume of the building, the air density and the air specific heat at normal conditions.

The Pandas Dataframes of the model inputs and model known values are obtained from the EnergyPlus variables report and the weather file. The variable report, which is a csv file, has been simply coded as a Dataframe through Pandas library on Python, while ladybug.py [27] library has been utilized for the epw file to convert the file in a Pandas Dataframe and selecting the global horizontal irradiance variable. The Dataframe of the known variables contains the values assigned to the output interface of the 3R3C model (Tzone). The Dataframe of the model inputs contains ambient temperature, global horizontal irradiance, heat gains from occupancy and lights and HVAC thermal load. The algorithm of ModestPy for the estimation of the parameters has been set with a maximum number of iterations for the genetic algorithm of 15 and a tolerance of 0.001, while for the generalized pattern search the maximum number of iterations is 500 and the tolerance is  $1e^{-6}$ . From OpenModelica the model has been exported as an FMU and path of its repository has been indicated in the script for ModestPy. With all the requirements, ModestPy can run and estimate the parameters of the model.

#### 2.4.3 Solar module

Sun radiations are very effecting on the temperature of buildings. From this consideration, it has been thought to increase the accuracy of the data regarding heat gained from the sun. The solar radiation hitting a body is composed of three components: direct, diffuse and reflected. Direct radiation, called also beam radiation, is the radiation which travels a straight line from the sun to the body surface. The diffuse radiation consists of the part of sunlight that reaches the surface body after being scattered by molecules and particles in the atmosphere. The reflected radiation is the component that gets reflected by another surface before reaching the body surface. From the building geometry, it is possible to calculate the amount of radiation hitting the building at every timestep knowing the sun position.

The two information about the sun position to know are the sun altitude and azimuth. Sun altitude is the angle between the horizon plane and the sun while the azimuth is the angle on the horizontal plane of the projection of the line joining the sun and the observer. There are different conventions for the azimuth angle, the one used in this work states that the angle is zero when the projection coincides with south direction, it is positive moving clockwise and negative counterclockwise.

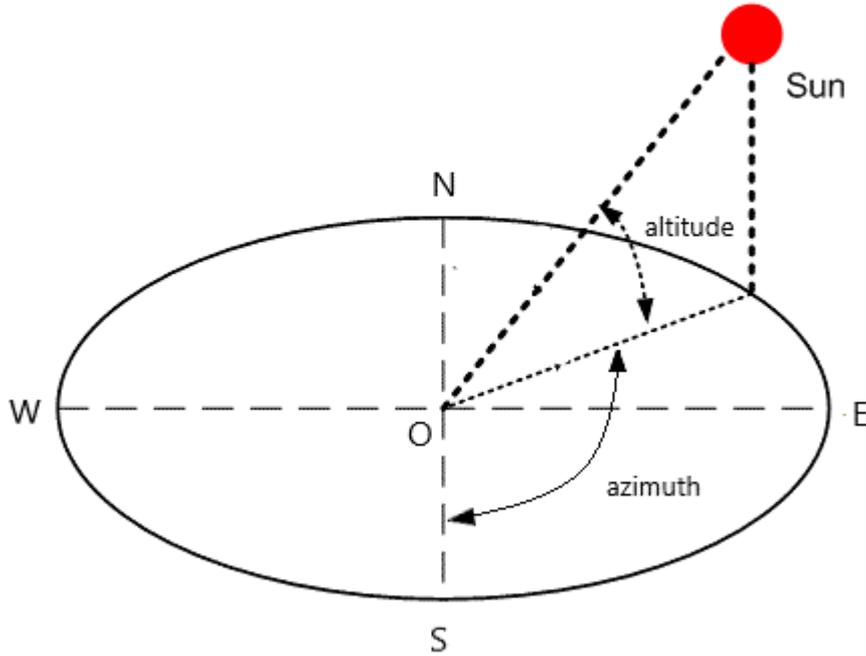


Figure 2.10 Sun altitude and azimuth

Sun altitude and azimuth are necessary to calculate the incident angle of the sun rays on a certain surface, for vertical surfaces, like the walls, the formula is the following:

$$\cos\vartheta = \cos\beta * \cos(\Psi - \Psi_s) \quad (15)$$

where  $\vartheta$  is the incident angle,  $\beta$  is the sun altitude,  $\Psi$  is the sun azimuth and  $\Psi_s$  is the azimuth of the surface which is referred to the projection of the normal to the surface.

For the RC model, it is needed the power coming from sun radiations this physical dimension is called irradiance and it is express in  $W/m^2$ . Direct irradiance, diffuse irradiance and reflected irradiance are calculated with Equation 16, Equation 17 and Equation 18, respectively.

$$E_b = E_{bn} * \cos\vartheta \quad (16)$$

$$E_d = E_{dh} * \frac{1 + \cos\varphi}{2} \quad (17)$$

$$E_r = E_{th} * \rho * \left(1 - \frac{1 + \cos\varphi}{2}\right) \quad (18)$$

Where  $E_b$  is the normal component of the direct irradiance on the surface,  $E_{bn}$  is the total direct irradiance,  $E_d$  is the diffuse irradiance,  $E_{dh}$  is the total diffuse irradiance on the horizontal plane,  $\varphi$  is the inclination of the surface,  $90^\circ$  if vertical,  $E_r$  is the reflected irradiance,  $E_{th}$  is the total irradiance on the horizontal plane and  $\rho$  is the albedo, which represents the ration of radiations reflected by a surface.

It has been decided to neglect the reflected irradiance as part of the inputs in the RC model because it depends on the surrounding environment of the building, the albedo parameter, and hence it is difficult to determine. However, reflected irradiance is usually a minor part of the total irradiance in cities, like Torino, where buildings are short and the surrounding is dark.

The two elements of sun positions in Torino have been calculated every 15 minutes by the Python library Pysolar [28] giving the city latitude, longitude and altitude.  $E_{bn}$ ,  $E_{dh}$ , and  $E_{dh}$  have been obtained from the EnergyPlus weather file of Torino. A model to calculate the irradiance hitting the building external walls and the windows has been built on Modelica. The scope of the model is to calculate the two irradiance components at every timestep, according to the geometry and position of the building. This model can then be integrated into the 3R3C model. Figure 2.11 shows the structure of the model. It calculates the solar power hitting the four external walls of a building summing the values obtained with Equation 16 and Equation 17 multiplied for the relative wall areas (windows included). Same thing with the area of the window to obtain the heat destined to the internal mass. The difference between the first and second value gives the power that hits only the external walls. In the model, there are four blocks, one for each wall, in order to calculate the direct irradiance only when the walls are oriented toward the sun. In Figure 2.12 there is the 3R3C model implemented with the preview model. It presents more inputs interfaces than the simple 3R3C model but the parameters to estimate are the same, only  $ALPHA_{wall}$  and  $ALPHA_{int}$  change their significance. Since irradiance is already multiplied by the area, these parameters represent the fraction of heat which get absorbed. Lower and upper limits change as well and they take into account lights reflection and transparency of surfaces.

	<b>Lower limit</b>	<b>Upper limit</b>
<b><math>ALPHA_{wall}</math></b>	0.0	0.25 [6]
<b><math>ALPHA_{int}</math></b>	0.0	0.95 [20]

Table 2.4 Lower and upper bounds for  $ALPHA_{wall}$  and  $ALPHA_{int}$

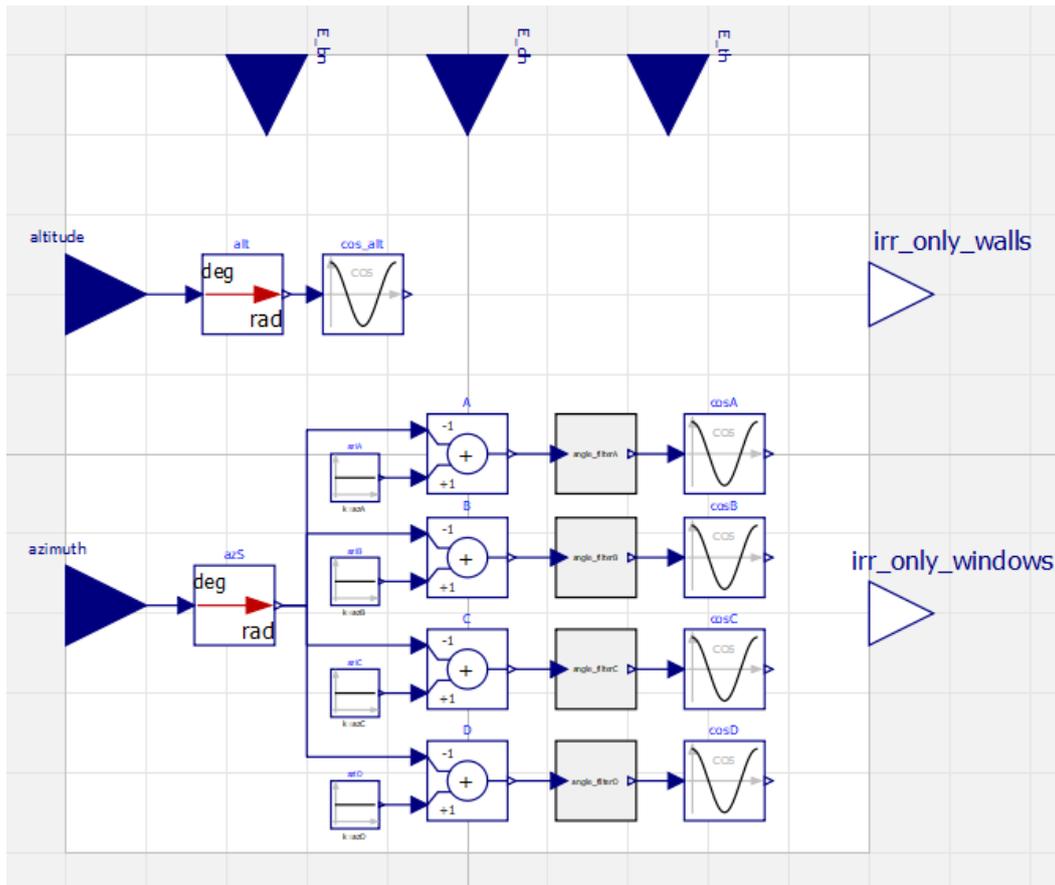


Figure 2.11 schematic of the sun component built on OpenModelica

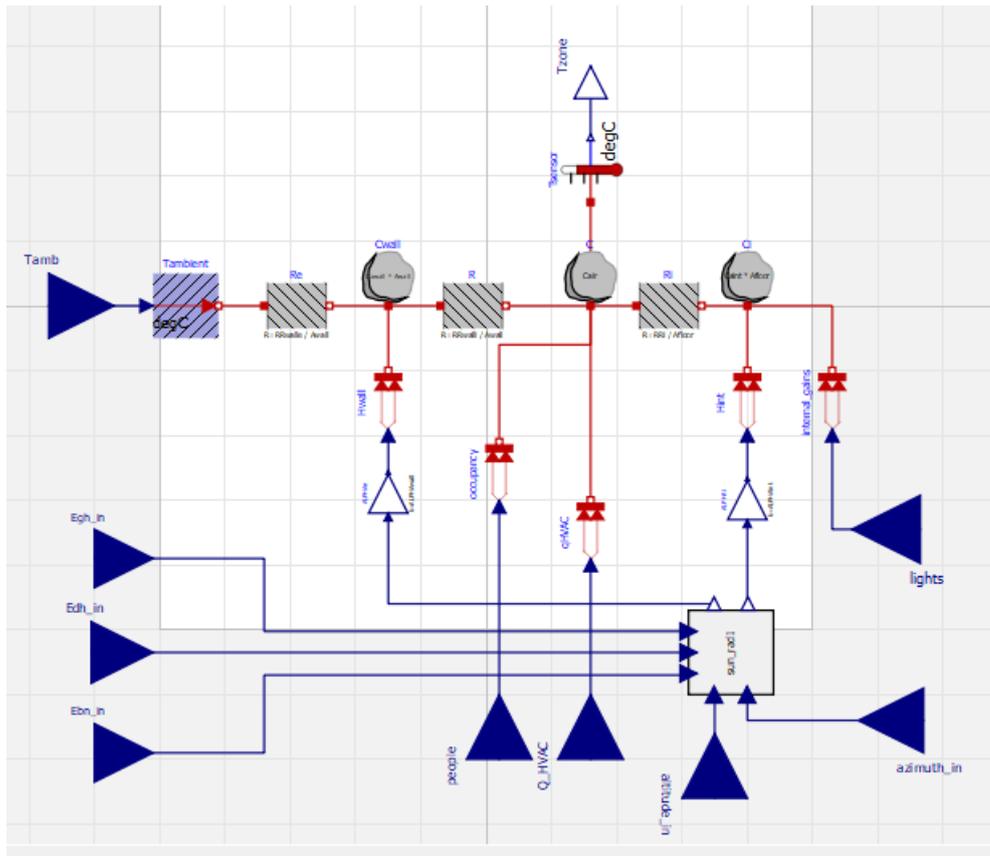


Figure 2.12 schematic of the 3R3C model with the sun component on OpenModelica

Figure 2.13 shows an example of the heat signal calculated by the sun block for four days. The blue line is the heat that is directed to the external wall capacitor, while the orange line is the heat directed to the internal mass capacitor. It is notable that the windows are on the east and west façades of the EnergyPlus model because the orange line is characterized by two peaks every day and a drop between them. The drops indicate the noon of every day because at the time the sun is oriented toward the south and the windows are not hit directly by the sun rays.

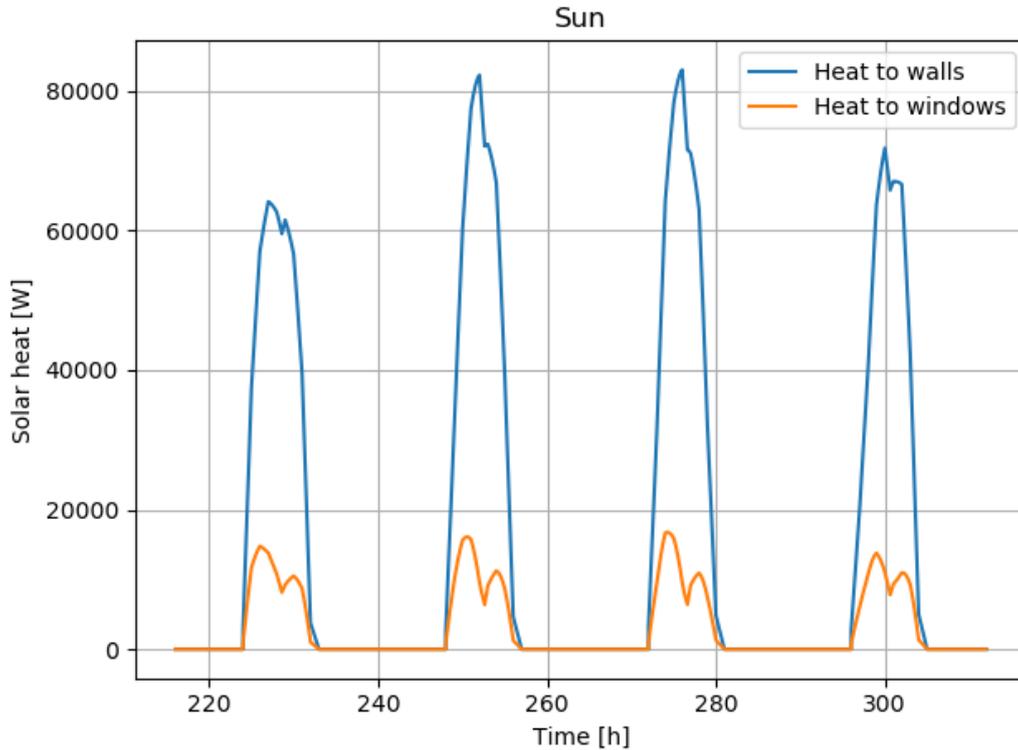


Figure 2.13 Example of the heat flux destined to the external walls and to the windows calculated by the solar component

#### 2.4.4 Model training and validation

The two RC models, from now on called Basic and Sun, have been tested on the same periods and with the same procedures in order to verify if adding the sun component to the Basic model will bring some improvements and to find the best strategy to training and exploit the models in this case study. First, the two models have been trained for six months, from October to February, where the HVAC system is used only for heating. At each day of the period, the parameters learning has been computed on one, three, five and seven previous days. In this way, for every day four sets of parameters have been found that have been used after to validate the models. The validation of the model has been done thanks to PyFMI. At every day, the FMU is loaded on PyFMI and all the variables, inputs and outputs are initialized with the data from the EnergyPlus model at the first time sample of the day. For every set of estimated parameters, the FMU parameters are set with their relative values and three validations are performed, for a prediction horizon of one, three and seven days. The root mean square error (RMSE) is utilized as a parameter to evaluate the accuracy of the predictions, it indicates the average deviation of the predicted values from the real values at a certain time sample for all the time of the prediction. As lower is the RMSE value as more the prediction is accurate. Its calculation is described in the equation below:

$$RMSE = \sqrt{\frac{\sum_i^N (\hat{Y}_i^t - Y_i^t)^2}{N}} \quad (19)$$

where, as for Equation 1,  $\hat{Y}_i^t$  is the measured value of the variable  $i$  at time step  $t$ ,  $Y_i^t$  is the simulated variable  $i$  at time step  $t$  and  $N$  is the number of time steps.

Three Pandas Dataframes have been generated during this procedure and then converted in csv file. One with columns equals the number of parameters to estimate and rows equal to the number of training procedures performed. Each column contains the value of the relative parameter estimated for every training procedure. A second DataFrame contains the RMSEs calculated comparing the EnergyPlus model data and the after simulation RC model data both on the learning period, the DataFrame has four columns, each relative to a learning length, and rows equal to the number of days in the whole testing period. The last DataFrame contains the RMSEs calculated for the validation procedures, the columns are organized as follow: first the RMSEs calculated on one, three and seven days after the learning of one day, then the three RMSEs calculated after three days of leaning, the three RMSEs calculated after five days of learning and finally the three RMSEs calculated after seven days of learning, for a total of 12 columns. The rows are equal to the number of days in the whole testing period.

The office building is characterized by a specific schedule and this brings the building thermal dynamic to have five days, from Monday to Friday, with a very similar behaviour between each other, while Saturday and Sunday different. The aim of the models is to predict the building dynamic during the working days, for this reason, a new training and validating procedure has been run on a new period where from the original period every Saturdays and Sundays have been removed. The procedure is exactly equal to the previous one and produces the same results. The two periods of training and validating, with and without the weekends, will be called respectively “Complete period” and “Selected period” from now on.

### 3 Results and Discussion

This chapter contains the graphical results of the procedures of training and validating the models described in Chapter 2. The 3R3C model with the component to calculate the solar irradiance according to the geometry will be called in this chapter “Sun model”. The 3R3C model without the implementation will be called “Basic model”.

The first three graphs show examples of predictions compared to the EnergyPlus data of the same time samples. These examples are reported to graphically demonstrate how the RC model can predict the temperature of the EnergyPlus building and make some consideration about the behaviour observed in all the tests done.

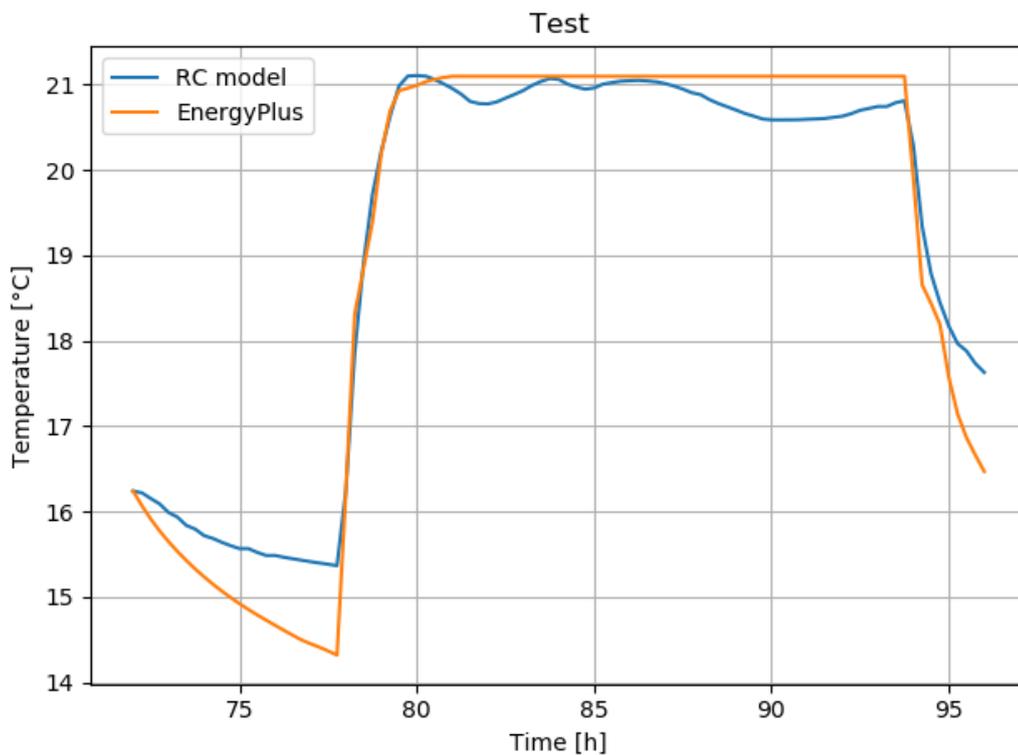


Figure 3.1 Example of a prediction of one day after a learning period of one day.

Figure 3.1 shows the comparison between the prediction of one day made from the Basic model with the set of parameters estimated on the previous day to the prediction and the EnergyPlus building temperature of the same day. The two axes of the graph indicate on the abscissa the time of the day in hours, and on the ordinate the internal temperature of the RC model and of the EnergyPlus building in Celsius degree. Both the lines in the figure start at time 72 h, which corresponds to the midnight of the previous day, and end at midnight of the analyzed day, time 96. It is notable that for the first six hours the RC mode does not follow the EnergyPlus line, leading to a difference between them of about 1 °C. At time 78 (6 am), when the HVAC system is turned on, the EnergyPlus model has a strong step rise of the duration of

two hours, and the temperature goes from below 15 °C to around 21 °C, which is the temperature setpoint. During this period, the RC model overlaps the orange line, even if it starts from a different temperatures. The period from time 80 to time 94 is the period in which the temperature on the EnergyPlus model is kept around the setpoint, in fact, the orange line is horizontal and slightly above the 21 °C. The blue line does not coincide with the EnergyPlus line anymore but it has an oscillatory behaviour remaining under that. The maximum difference in temperature in this period does not exceed 0.5 °C. The last temporal section represents the free-floating period after the HVAC system is turned off. It last two hours from 10 pm to 12 pm. Like in the initial period, the two lines diverge from each other, ending with a temperature difference greater than 1 °C.

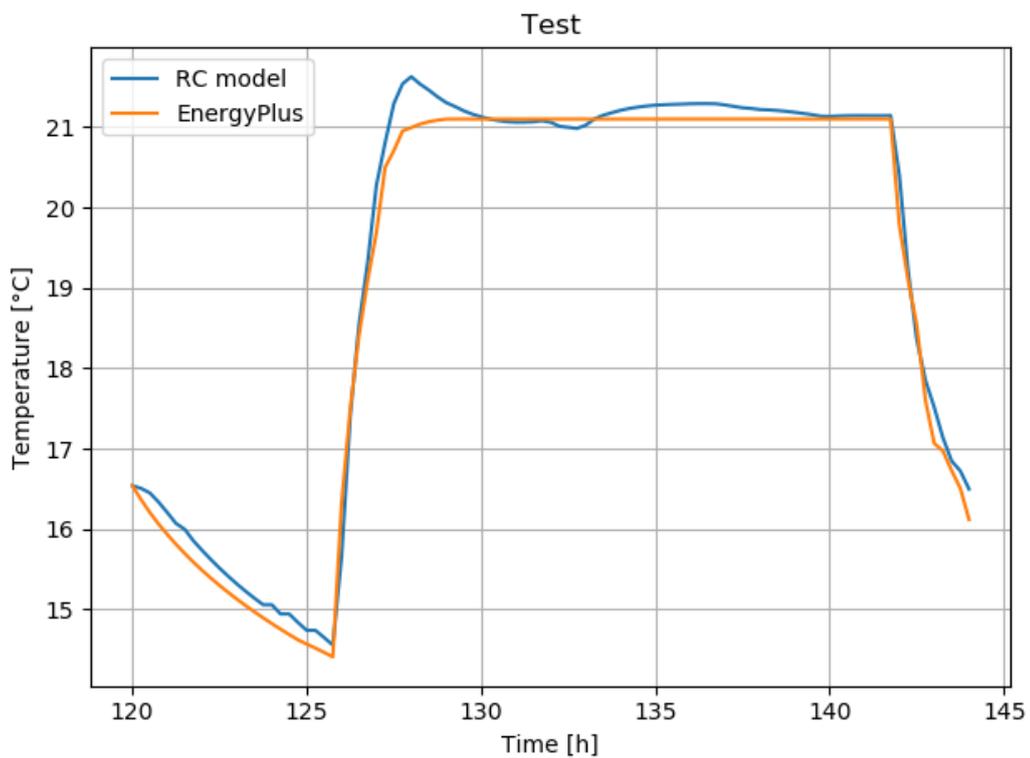


Figure 3.2 Example of a prediction of one day after a learning period of three days.

Figure 3.2 reports an example similar to Figure 3.1, but in this case, the parameters set to the FMU of the Basic model have been determined with three days of training and the predicted day is two days ahead. The two lines start at around 16.5 °C and go down maintaining a relatively constant and small difference for the 6 hours of free-floating before the HVAC system is turned on. During the ramp, the RC model line coincides with the EnergyPlus line for most of the period, but at the end, the blue line is faster and reach a temperature around 21.6 °C while the orange line stops at 21 °C. Next, the blue line rejoins the EnergyPlus line in a couple of hours and continues to approximate the orange line with little fluctuation for all the rest of the day.

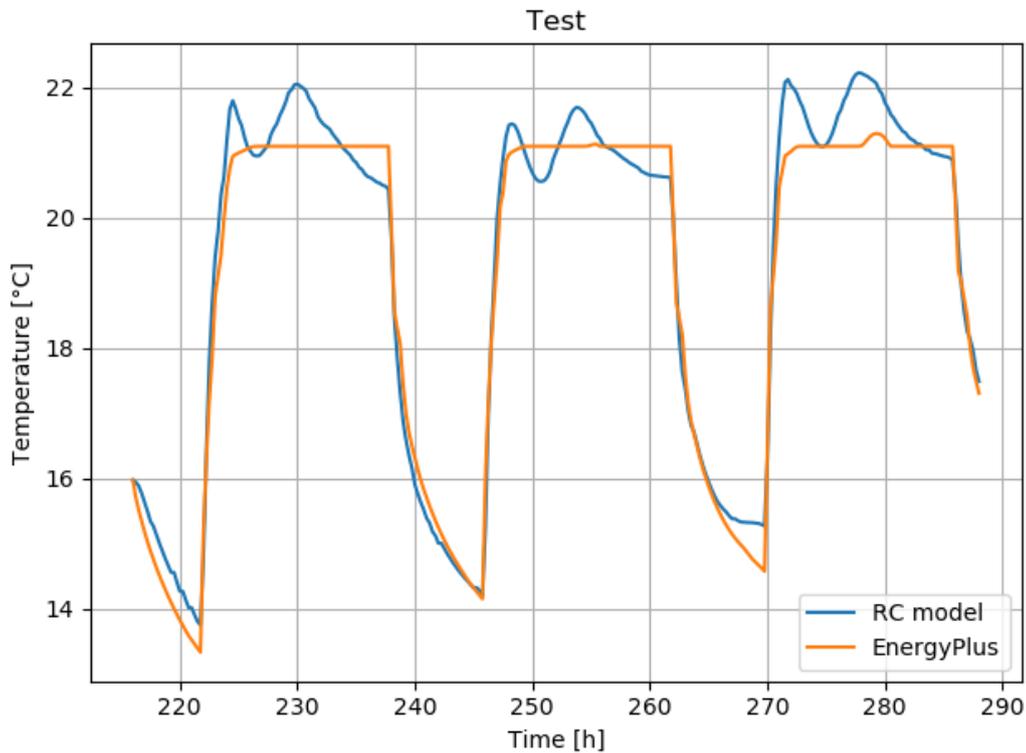


Figure 3.3 Example of a prediction of three days after a learning period of three days

Figure 3.3 shows an example of three days prediction with the parameters coming from three days learning period. In this figure, it is notable that the RC model line approximates the EnergyPlus line better on the free-floating period and on the ramps where the temperature increases rapidly after the HVAC system is just turned on. During both the three periods of setpoint temperature maintenance in the EnergyPlus model, the blue line has an oscillatory behaviour with two peaks, one at the end of the ramps and one around the center of the periods. The oscillations have an amplitude greater than 1 °C and they end when the free-floating periods start. At this instant, the RC model temperature is lower than the EnergyPlus temperature.

The three previous graphs want to show that after more than one day of parameters learning, the prediction is always more accurate in the free-floating period than with only one day of learning. This is caused by the fact that when the model simulates more than one day, the temperature at the starting of the HVAC system must be at a proper value in order to let the temperature reach the setpoint and stay around it. If this does not happen, there will be an error in the simulation that will continue to shift, up or down, the temperature line while the days go on. During the model training, this constrain followed as well. However, it has been observed that a good accuracy in representing the free-floating periods does not lead to being as much accurate in the HVAC controlled periods. During one day of training, instead,

the accuracy in the HVAC controlled time is more important to the machine learning algorithm because is the major part of one day.

### 3.1 Benchmark on the whole heating season

In this paragraph, the results of the learning and validating procedure on the whole heating season are presented for both Basic and Sun model. The “Complete period” goes from October to March, including all the days of a week, Saturdays and Sundays. Further on, “Selected period” will be called the period from October to March with no Saturdays and Sundays. The DataFrame with the RMSEs of the parameters validations has been elaborated to obtain the arithmetic average of the RMSEs characterized by the same learning length and prediction length. There have been obtained 18 values in total. Figure 3.4 compares the average RMSEs of one day predictions for each learning period length. In the bar plot, the colour blue indicates the average RMSEs calculated with the data from the Basic model training and validating process, while the orange is relative to the Sun model. The bars are grouped by the same parameters training period length, in order, 1 day, 3 days, 5 days and 7 days. With 1 day of training, the average RMSEs of both models exceed the 2 °C and Basic model average is greater than Sun model ones. With 3 days of learning the blue column is shorter than the orange ones and both do not surpass the 1 °C value. The values of the average RMSEs from 5 days learning are around 1 °C and also in this case the Basic model RMSEs is slightly minor than the Sun model RMSE. The last two bars, relative to a training period of 7 days, have practically the same height which does not exceed 1 °C.

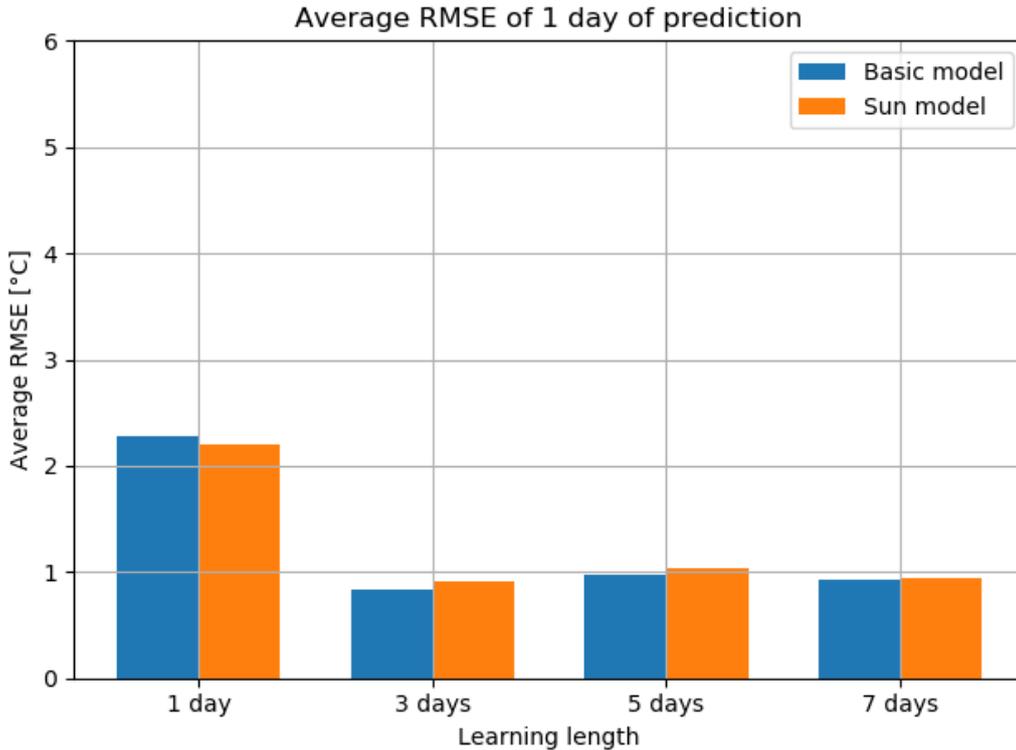


Figure 3.4 Comparison between the average RMSEs of one day predictions from Basic model and from Sun model

The same kind of plots, but built with the RMSEs of 3 days and 7 days of prediction can be found in the Appendix. They have not been shown in Results because besides the general increments of values of the average RMSEs due to the increment of the prediction length, the behaviour of the bars reflects the behaviour in Figure 3.4. One day of learning leads to the greatest RMSEs of which Basic model ones are the highest. For the other learning lengths, the RMSEs are all comparable and the predictions of the Basic model always have a slightly better accuracy.

The fact that the values of the average RMSEs coming from the prediction length of 1 day are much higher than the other values does not correspond with the results expected, especially for 1 day of prediction. In literature, for making a 1 day prediction, it is usually suggested to compile the parameters training on the previous day because it is reasonable that the two days would have very similar behaviour. Figure 3.4 seems to deny that, but the reason can be imputed to the schedules of the office building. For this reason, the data in the DataFrame of the accuracy of the predictions have been reorganized in order to group all the RMSE values relative to every day of the week. This work has been done only with the Basic model data. The results of this elaboration bring for every day twelve average RMSEs characterized by the length of the learning period and the length of the prediction, which starts on that day. The results of Mondays and Tuesdays are visible in Figure 3.5 and Figure

3.6, respectively. The graphs present for every learning length three bars indicate the average RMSEs relative to the three prediction horizons: 1, 3 and 7 days.

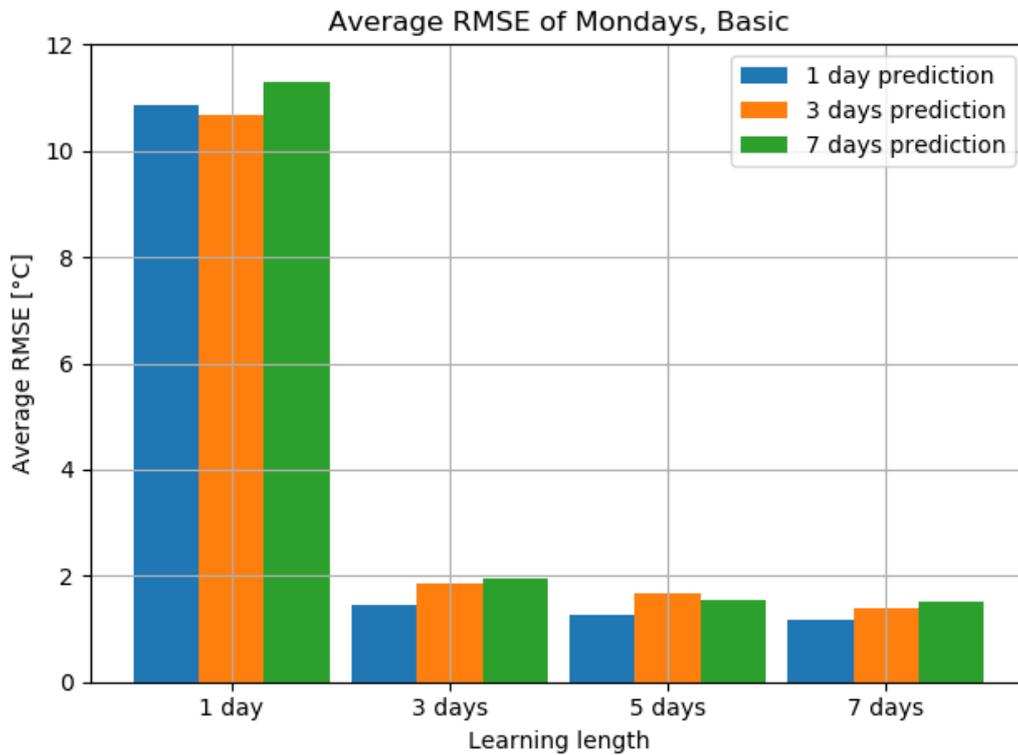


Figure 3.5 Average RMSEs of the predictions starting on Mondays

In Figure 3.5 all the values of 1 day learning length exceed the values of 10, while the other RMSEs do not and stay below 2. For the cases regarding 3 days, 5 days and 7 days of training, the most accurate prediction are always the ones of 1 day. Values greater than 10 for an RMSEs of internal temperature of a building mean that the parameters estimated are not suitable to represent that system. The predictions starting on Monday use the parameters learned on the previous days when the learning length is 1 day, the model is trained on Sunday, where the EnergyPlus building has not heat gained from the HVAC system, from the occupancy and the lights in the building. It is a day with a completely different behavior compared to Monday and that is the reason why the accuracy is so low for the predictions with 1 day of learning; when the learning period increase, the parameters are trained on days with the office schedule as on Monday and so the accuracy become better.

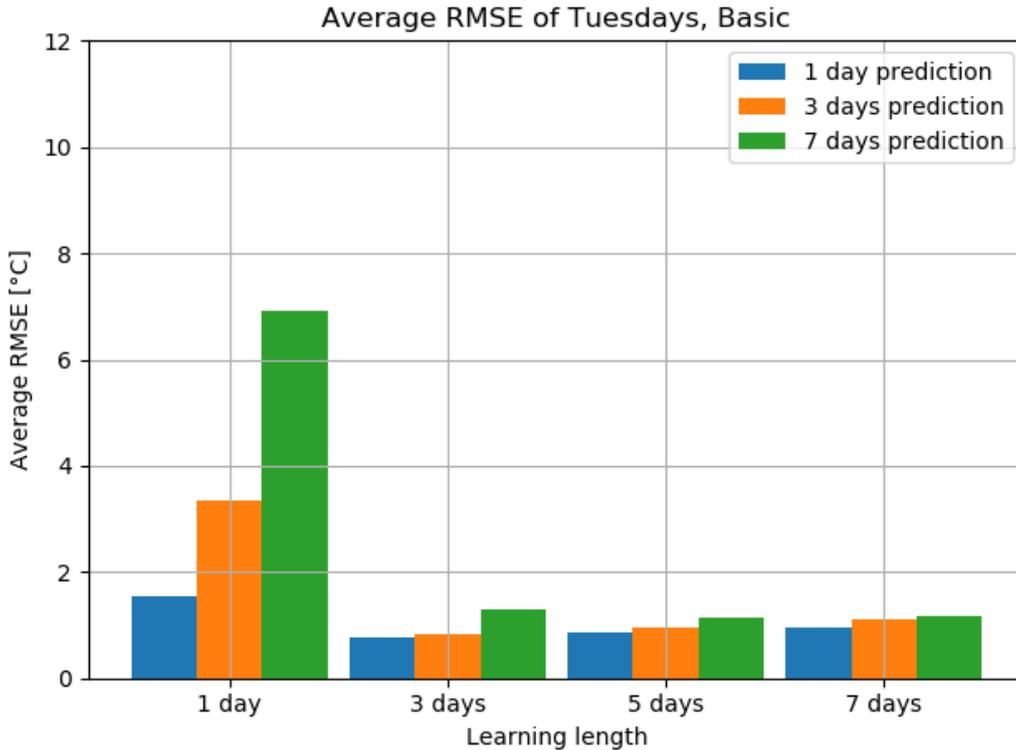


Figure 3.6 Average RMSEs of the predictions starting on Tuesday

Figure 3.6 presents the same data of Figure 3.5, but relative to Tuesday. The bars of the predictions with 1 day of learning are arranged like a ladder: the RMSE value of 1 day prediction is between 1 and 2, the value of 3 days prediction between 3 and 4 and the value of 7 days prediction around 7. In all the other learning length, the accuracy of the predictions is comparable between each other, with the RMSE values that exceed 1 in few cases, but also for these learning lengths the prediction of 7 days is the less accurate then the prediction of 3 days and the prediction of 1 day is the most accurate. The graphs of the other days of the week mimic Figure 3.6.

Comparing Figure 3.5 with Figure 3.6 shows that with one day of learning, if the day to predict is different from the day used for the training the parameters and the model will not provide an accurate prediction. The utilization of Sundays to training the model to predict Mondays explain why in Figure 3.4 the average RMSEs of 1 day learning are double the values of the other learning length. From Figure 3.6 it is deductible also that with one day of learning, the accuracy of the predictions of more than 1 day decreases in relation to the increasing of simulating days. This is explainable also by what previously said about the example of 1 day training and 1 day prediction in Figure 3.1. Since in 1 day learning it is not important that the parameters lead, at the end of the day, that the simulated temperature coincides with the real building temperature if the overall accuracy is good. Nevertheless, if those parameters are

used to simulate more days, the starting temperature of the second day will be already wrong and so all the predictions will be affected by this error.

### 3.2 Benchmark on weekdays

This paragraph contains similar results of the previous paragraphs but obtained doing the process of learning and validation on the period from October to March, considering only the weekdays, from Monday to Friday. Figure 3.7 shows the average RMSEs of the 1 day validations carried out with the Basic model and the Sun model.

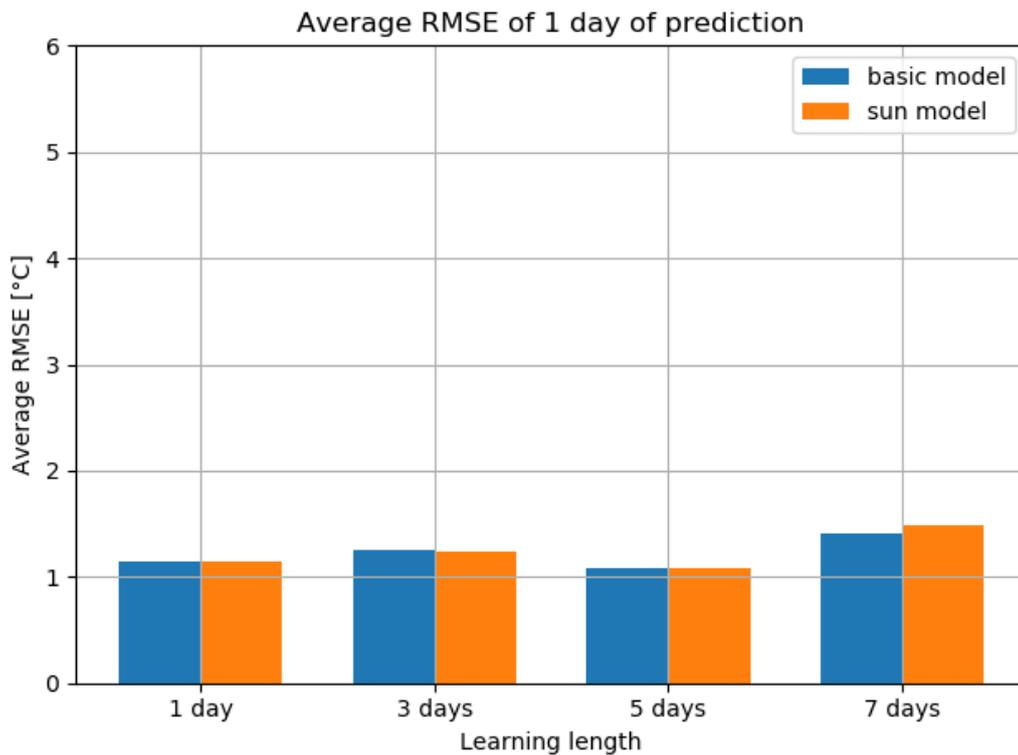


Figure 3.7 Comparison between the average RMSEs of one day predictions from Basic model and from Sun model

At 1 day of learning the two bars reach the same high, which is slightly above 1 °C, at 3 days of learning, the blue column is a bit higher than the orange one, both express values greater than the 1 day prediction values, but they are still below 1.5 °C. 5 days of learning is characterized by having the shortest columns of the graph, both Basic and Sun models, which are practically identical. Their values are still above 1 °C. The last bars present in the graph are relative to 7 days of learning and their peaks reach around 1.5 °C; the average accuracy of Sun model, in this case, is a bit worse than the average accuracy of the Basic model. Comparing Figure 3.7 with Figure 3.4, it is notable that the main difference between them is the high of

the bars at 1 day of learning. Using the Selected period, the average of the RMSEs of 1 day learning decrease of about 1 °C with respect using the Complete period, demonstrating that removing the weekends from the days of learning leads to decreasing the error of the validation after 1 day of parameter training. The other columns in the graph have a slightly increasing, this can be caused by the fact that removing the learnings on Saturdays and Sundays, also the validations on these days are not done anymore, which were characterized by the lowest RMSEs and they do not contribute any more to the RMSE average. The plot with average RMSEs of the validation on Sundays are visible in the Appendix.

Observing all the graphs produced from the validations on the Complete and Selected period there is no evidence that the 3R3C model with the component able to calculate the sun irradiance components provides better predictions than the 3R3C without that implementation. The reason is that in both Basic and Sun models, the parameters to estimate that regulate the solar heat gains are the same number. The two heat flux sources of Sun model receive two different signals compared to the respective heat flux sources of the Basic model, but all these signals are modulated by the relative parameter, which can make them very similar and cancel the more realistic sun inputs in Sun model.

In the next figure, all the average RMSEs from the Basic model are shown, the same graph relative to the Sun model is visible in the Appendix, but it looks like this one. Figure 3.8 is composed of 12 colored squares, each of them represents the value of the average RMSEs of the validations characterized by each learning length and each validation length. The learning length is reported on the abscissa axis and the prediction length on the ordinate axis. The colors of the cells indicate a value according to a scale, the scale starts with yellow and ends with black, the more the color is darker more the value is high. The darkest block of the figure is the average RMSE of 7 days of prediction and 1 day of learning, which is purple, next to it, there are two red cells, relative to 7 days of prediction and 3 days of learning, and 3 days of prediction and 1 day of learning. The other blocks have various shades of orange, the two lightest are indicated by 1 day prediction and 1 day learning and 1 day prediction 5 days learning.

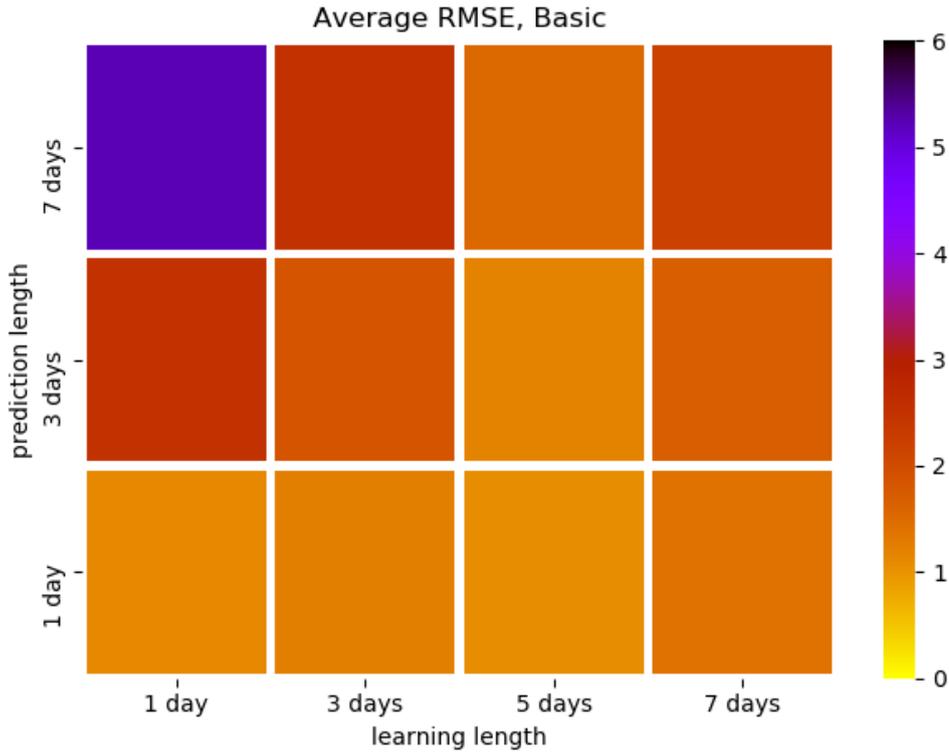


Figure 3.8 Heatmap of average RMSEs obtained from the validations with the Basic model

From Figure 3.8 it is visible that, also now that the days have all the same behaviour, a model training of one day is not suitable for doing prediction longer than one day. This deduction can also be applied to 3 days of learning length, in fact when the model predicts 7 days the accuracy decreases and also in literature, it is quite established that for predicting a certain time period the learning period shall have the same length. Another thing notable from the graph is that passing from 5 days of learning to 7 days of learning, the accuracy of all the prediction decreases. This fact is not expected, even more, if compared with the results of the Complete period, where 7 days of predictions lead always to better accuracy than 5 days. An explanation to this can be found in the building temperature data utilized both for training and validating the model. From the original file used in the Complete period procedure, the data relative to the weekend days have been cut, making Fridays be followed by Mondays; the temperature at the last time sample of Friday and the one at the first sample of Monday are not connected by any physics relation and can be quite different. This discontinuity can affect both the validation and the training which is based on a series of validation on the same period of the training. The discontinuity is present only on the building data, and not in the data from the simulation of the model, it happens that the model determines the temperature of the first sample of Monday based on the last temperature of Friday and continue the simulation from that value, but in the real data the first sample of Monday is not correlated from Friday temperature. In this way the model starts the simulation already of Monday from

the wrong temperature value. This error can be transmitted along all the simulation, decreasing the overall accuracy. This phenomenon can happen in all the prediction lengths longer than one day, but in 7 days prediction, it can happen twice.

Finally, Table 3.1 shows the seconds spent to estimate the parameters with Basic model and Sun model. The python script has been run by console by Pycharm on a computer with a processor Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz, 1992 Mhz, 4 Core(s), 8 Logical Processor(s). The four columns contain the computational time for completing the estimation of the parameters on a learning period of 1, 3, 5 and 7 days. With the same learning length, the training with the Basic model is always faster than with the Sun model. On 1 day learning the difference is very small, 7 seconds, while on 7 days is around three minutes. The difference is caused by the simulation time of the respective FMUs, during the training the FMUs are simulated 515 times. It is plausible that the Sun model requires a longer computational time because it has some components in addition to the Basic model. The duration of the learning can be increased or decreased changing some parameters in the estimate() function of ModestPy, such as the maximum number of cycles or the minimum tolerance. Moreover, the computational cost depends on the number of parameters to estimate.

	<b>1 day</b>	<b>3 days</b>	<b>5 days</b>	<b>7 days</b>
<b>Basic model</b>	132 s	321 s	346 s	708 s
<b>Sun model</b>	139 s	363 s	597 s	876 s

*Table 3.1 Computational time for parameters learning*

On an MPC system for a building, the parameters learning is usually done maximum every 24 hours in order to utilize the parameters determined on the day before to estimate the next day. The maximum value in the previous table is 876 seconds, which correspond around to 15 minutes, meaning that for setting the model with the proper parameters the learning has to be done maximum 15 minutes before the time to run the optimization algorithm. Understandably, a delay of 15 minutes on the starting of the optimization does not really affect the goodness of the optimization itself. In conclusion, the computational cost of the parameters learning is not a factor on which basing the choice of the learning length, the only factor to consider is the accuracy of the predictions.

### 3.3 Comparison of the grey-box model with ISO standards

In this section, the 3R3C basic model will be compared with the two standards: UNI EN ISO 13790[21] and UNI EN ISO 52016-1[22]. They both[22][21][21][21] introduce a building model composed by resistors and capacitors whit deterministic and analytical procedures to identify the parameters. In particular, the first standard introduces a model with 5 resistors and one capacitor representing the whole construction, while the second one presents a

model composed by modules of one resistor and one capacitor connected between each other in order to describe any component of the building. These models are validated by different works [22][23], which state that the RC analogy is a valid practice for designing dynamic building models. The comparison is made by applying the case study reported in the article [21] where an EnergyPlus model has been modeled using both the ISOs to the Basic RC model built in this thesis. The article evaluates the reliability in mimic the EnergyPlus model of two RC models, one done following the UNI EN ISO 13790 and the other one following the UNI EN ISO 52016-1. The evaluation has been made during 12 days in which the HVAC system was off and the estimation of the accuracy of the models has been done on the building temperature with three parameters: the RMSE, the Mean Bias Error (MBS) and the Coefficient of Variation of RMSE (Cv(RMSE)). MBS and Cv(RMSE) are calculated as following:

$$MBS = \frac{\sum_i^N (\hat{Y}_i^t - Y_i^t)}{\sum_i^N Y_i^t} * 100 \quad (19)$$

$$Cv(RMSE) = \frac{RMSE}{\frac{1}{N} \sum_i^N Y_i^t} * 100 \quad (20)$$

where  $\hat{Y}_i^t$  is the measured value of the variable  $i$  at time step  $t$ ,  $Y_i^t$  is the simulated variable  $i$  at time step  $t$  and  $N$  is the number of time steps. In this case, the measured variable is the variable obtained from the EnergyPlus model.

Table 3.2 shows the parameters calculated for each model in the article. The raw called Threshold reports the thresholds for MBS and Cv(RMSE) contained in the ASHRAE Guideline 14[24].

	<b>RMSE</b>	<b>MBS</b>	<b>Cv(RMSE)</b>
<b>UNI EN ISO 13790</b>	1.16 °C	-4.2%	4.7%
<b>UNI EN ISO 52016-1</b>	0.21 °C	0.2%	0.9%
<b>Threshold</b>	-	+ - 10%	30%

Table 3.2 RMSE, MBS and Cv(RMSE) of two analytical RC models

With the 3R3C Basic, the same procedure performed in the article has been tried to be simulated. The model has been trained with the free-floating temperature data of the EnergyPlus model and then with the estimated parameters, a prediction of as many days has been done. It has to be declared before comparing the results that the EnergyPlus models of the article and this thesis are different and so the ambient data used for the predictions. Figure 3.9 shows the graphical validation of the Basic model on the 12 days predictions with the building in free-floating mode. The starting date of the prediction has been chosen with no particular reason because it has no sense comparing the same period with the article since the inputs data are different.

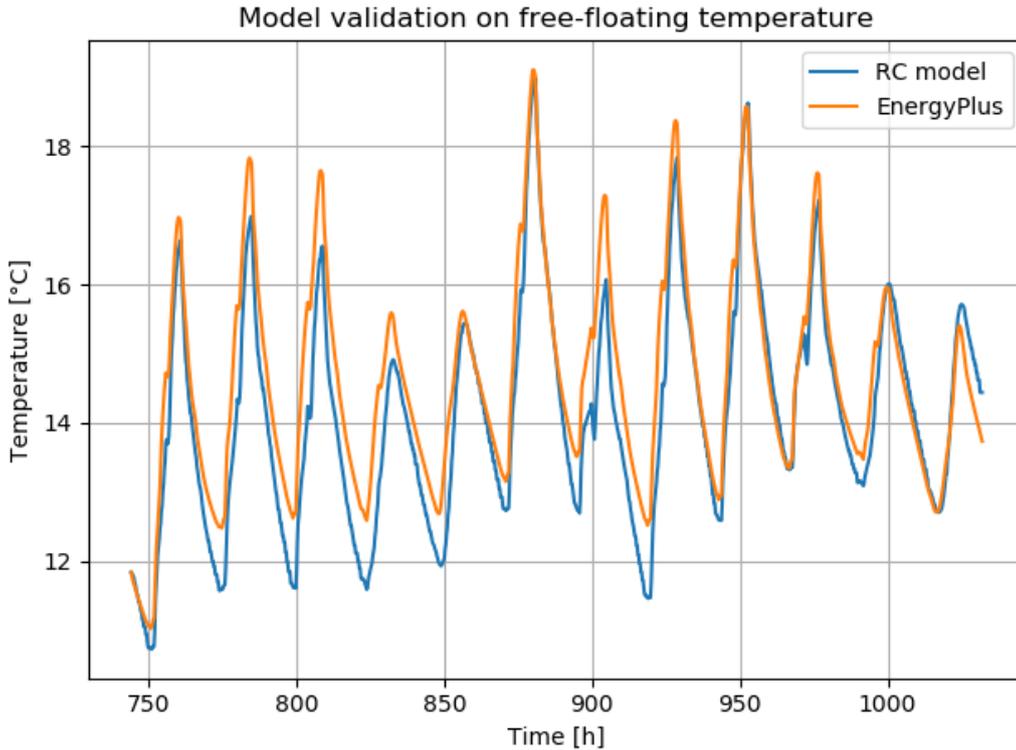


Figure 3.9 Validation of 12 days after a learning period of 12 days

The prediction starts at the 1<sup>st</sup> of February and ends at the 13<sup>th</sup>. The blue curve tracks the indoor temperature of the Basic model while the orange curve the one of the EnergyPlus model. The two curves vary in a range of temperatures between 10 °C and 20 °C. From the graph, the parts of each day are visible: the daylight when the temperature increases, and the night when the temperature decreases. The RC model line always follows the trend of the orange line, sometimes less precise than others. For example, in the first 7 days, during the night, the RC model predicts the lowest temperature almost 1 °C lower than the EnergyPlus one and sometimes the blue line does not reach the orange highest peaks. Table 3.3 reports the RMSE, MBS and Cv(RMSE) calculated with the predictions of the RC model and the EnergyPlus model data. The RMSE has a value of 0.72 °C, which is between the values of the RMSE of the two ISO models and the MBS and Cv(RMSE) which are -3.09% and 2.1%, respectively, are widely inside the range of the ASHRAE guideline thresholds. These results state that the Basic model described in this thesis and designed with the grey-box RC model is comparable, regarding the accuracy, to the RC models described in the standards.

	<b>RMSE</b>	<b>MBS</b>	<b>Cv(RMSE)</b>
<b>3R3C free-floating</b>	0.72 °C	-3.09%	2.1%

Table 3.3 RMSE, MBS, Cv(RMSE) of the Basic model prediction

## 4 Conclusions

This thesis aims to model a dynamic system such as a building with the grey-box method, designing an RC model and then determining its parameters by a data-driven algorithm, that can be used in a model predictive control (MPC). Two RC models have been investigated, one composed by three thermal resistors and three heat capacitors and the other one with the same components plus a module calculating the solar irradiance on the building according to the clear-sky equations and the building's geometry. The model has been built in Modelica and then coded in functional mock-up units (FMUs). The parameters of the models have been determined by using the Python library ModestPy, which employs a genetic algorithm and a pattern search algorithm. The reference building has been chosen an EnergyPlus model of an office building. A selection of variables from the EnergyPlus model simulation has been used to determine the RC models parameters and to validate them. The RC models have been tested along the whole heating period of the office building: at every day multiple parameters estimations and validations have been performed. The root mean square error (RMSE) has been calculated for each set of parameters and elaborated in order to compare the two different RC models and evaluate their accuracy.

The EnergyPlus model used in this thesis has a dissimilarity compared the reference buildings or building models used in literature as it has a relatively fast dynamic. In fact, the internal temperature of the building can go from about 10 °C to 21°C in two hours. All the other works in literature account for buildings characterized by a slow dynamic and a variation of internal temperature, during a whole day, lower than 5 °C. With such a fast dynamic, it is more difficult designing a lumped model and its accuracy tends to be lower than with a slow dynamic. For this reason, the benefit of adding an analytical model to account the sun irradiation has been investigated.

The results show that the two RC models are almost identical; their predictions are characterized by very similar values of RMSE, meaning that there is not an indication to utilize one model respect the other in any situation. Consequently, the choice of the model can be made according to other factors. The RC model, called Basic model, is simpler to design and understand and faster in compiling. Nevertheless, it has been shown that the computation time is not a relevant parameter for the choosing of the model. Considering that these two models should be implemented in a model predictive control for hypothetical real buildings, the data for the predictions can not be as much easy to obtain as from the EnergyPlus model. The MPC uses predictions of the inputs variable to determine the dynamic of the controlled system and thus optimize the controlled variable. An MPC with the Basic model needs as input the external ambient temperature, the heat gain from the occupancy and from the building lights, and the total horizontal solar irradiance. Instead, with the Sun model, the inputs would be the same plus the total direct irradiance and the total diffuse irradiance. Predictions of the total horizontal irradiance are available in some weather forecast but the total direct

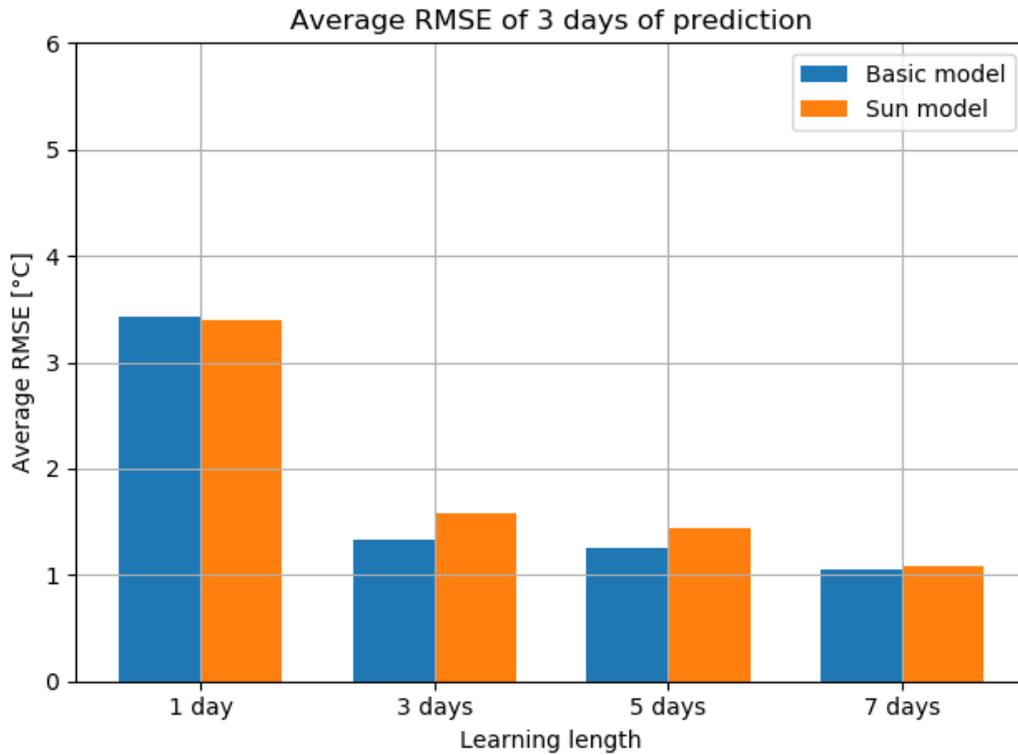
irradiance and the total diffuse irradiance are more difficult to find. This fact makes the Basic model the first choice because it is able to give the same predictions accuracy of the Sun model using fewer inputs and mostly, the inputs more available.

The values of the average RMSEs are comparable with the values reported in literature[7], where the average RMSEs of the predictions of a 3R3C model for a building with fast dynamic are all included between 2 °C and 3°C, while for a building with low dynamic are all below 1.5 °C. However, the values of RMSEs do not confirm that the lumped model is good to be used in an MPC, they can only be compared with other values in the literature for a general comparison. The reliability of a model can be established only by the MPC exploiting the model. In fact, for an MPC with the task of reducing the energy consumption for the heating, a good model is the one that will lead to an effective reduction of the energy consumption of the controlled system with respect to the building without the MPC.

The future work of this thesis is to utilize the RC models to set up an MPC for the HVAC system of a building and study the effects of the accuracy of the model predictions on the ability of the MPC to save energy. Nowadays, there are some options for developing an MPC based on the model of the controlled system. With a model coded in Modelica language, JModelica[25] offers tools for running optimization problems on the inputs of a model, unfortunately, it is not anymore an opensource program from January 2020. Another available and free tool is MPCPy[26] that has been developed exactly for optimizing the control signals of buildings devices. MPCPy is very young and still under development, i.e. it does not always work properly. An advantage that MPCPy has over JModelica is that it allows the user to avoid the programming of the interaction between the optimization tool and the model. For this reason, it is the first choice for proceeding forward in this work.

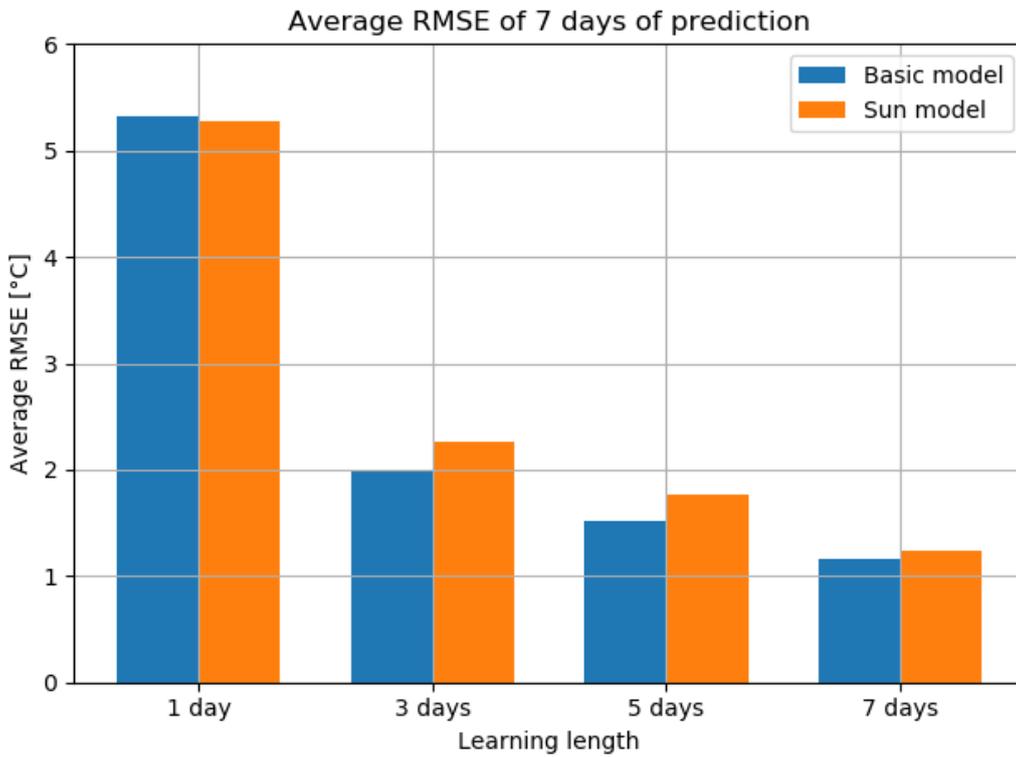
## Appendix A

Figure A. 1 shows the results obtained from the 3 days predictions made with the Basic model and the Sun model on the Complete period. The bars indicate a better performing of the Basic model rather than the Sun Model, only with 1 day learning length Sun model predictions are more accurate than Basic model predictions. All the values of the average RMSEs are under 2 °C except for the average with 1 day learning length.



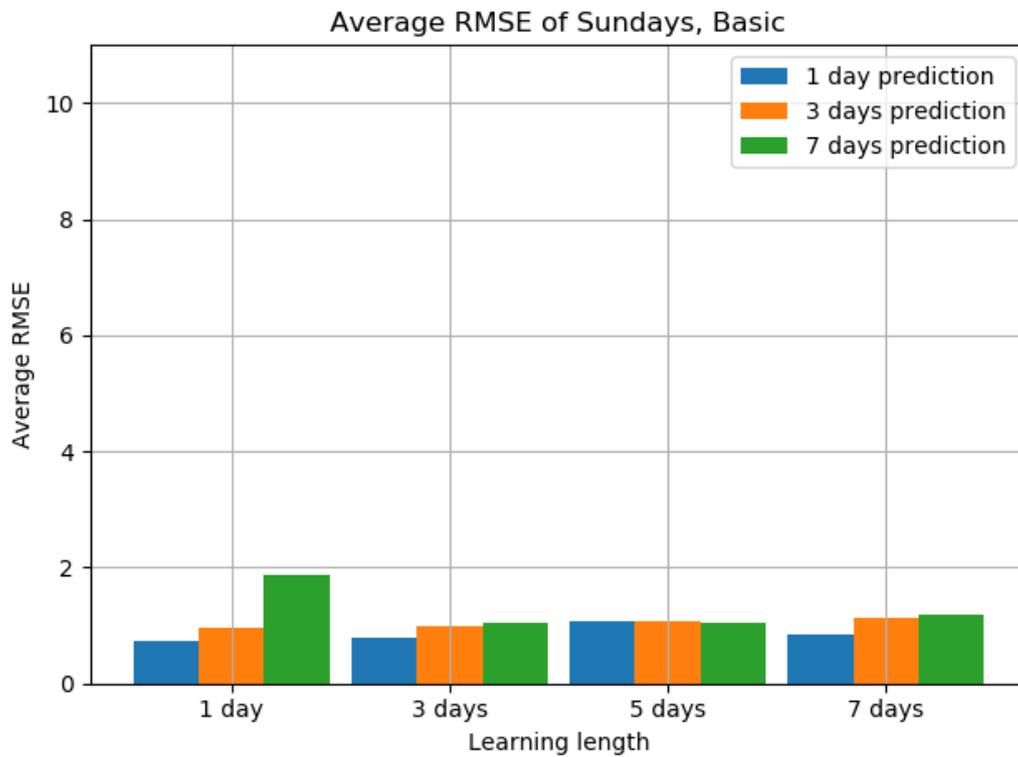
A. 1 Comparison between the average RMSEs of three days predictions from Basic model and Sun model. Complete period.

Figure A. 2A. 1 shows the results obtained from the 7 days predictions made with the Basic model and the Sun model on the Complete period. Similar to Figure A. 1, the average values of the RMSEs of the Basic sun predictions are lower than the values of the Sun model, except for 1 day learning length. The average RMSEs are below 2 °C for 5 and 7 days of learning length, around 2 °C for 3 days of learning length and are over 5 °C for 1 day of learning length.



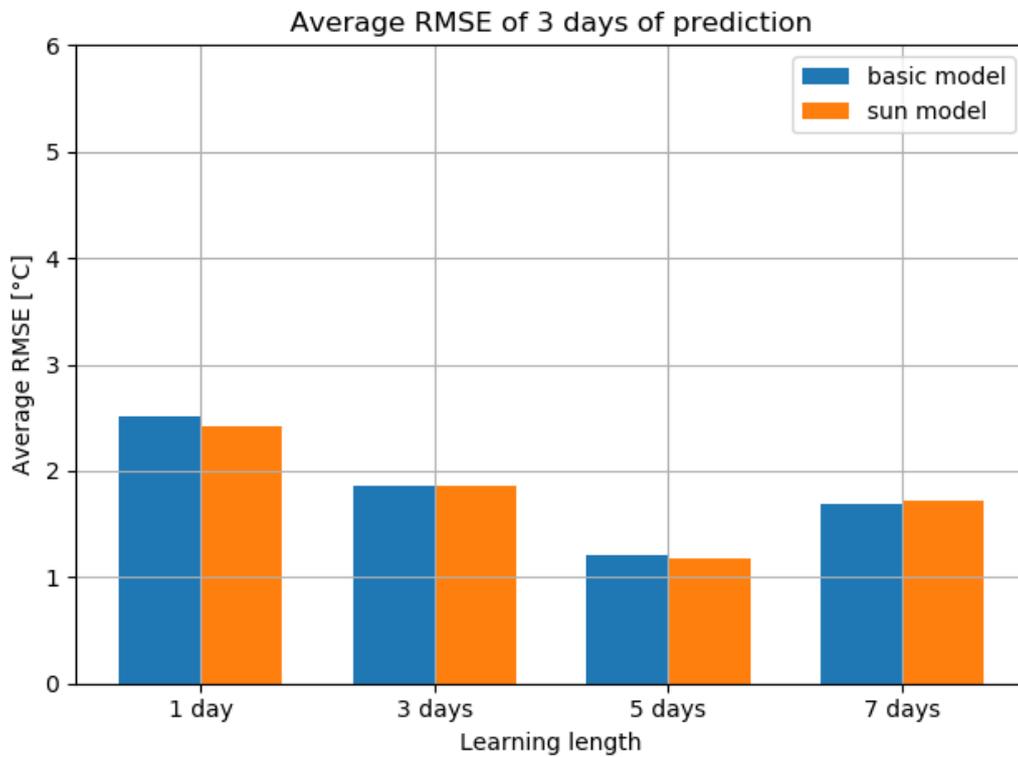
A. 2 Comparison between the average RMSEs of seven days predictions from Basic model and Sun model. Complete period.

Figure A. 3 shows the average values of the RMSEs of the predictions starting from Sunday on the Complete period. All the bars stay around the value of 1 °C, only the bar indicated by 1 day of learning length and 7 days of prediction reaches almost 2 °C. On 5 days of predictions, the three bars have the same high, on the other leaning lengths the blue is the shortest and the green is the highest.



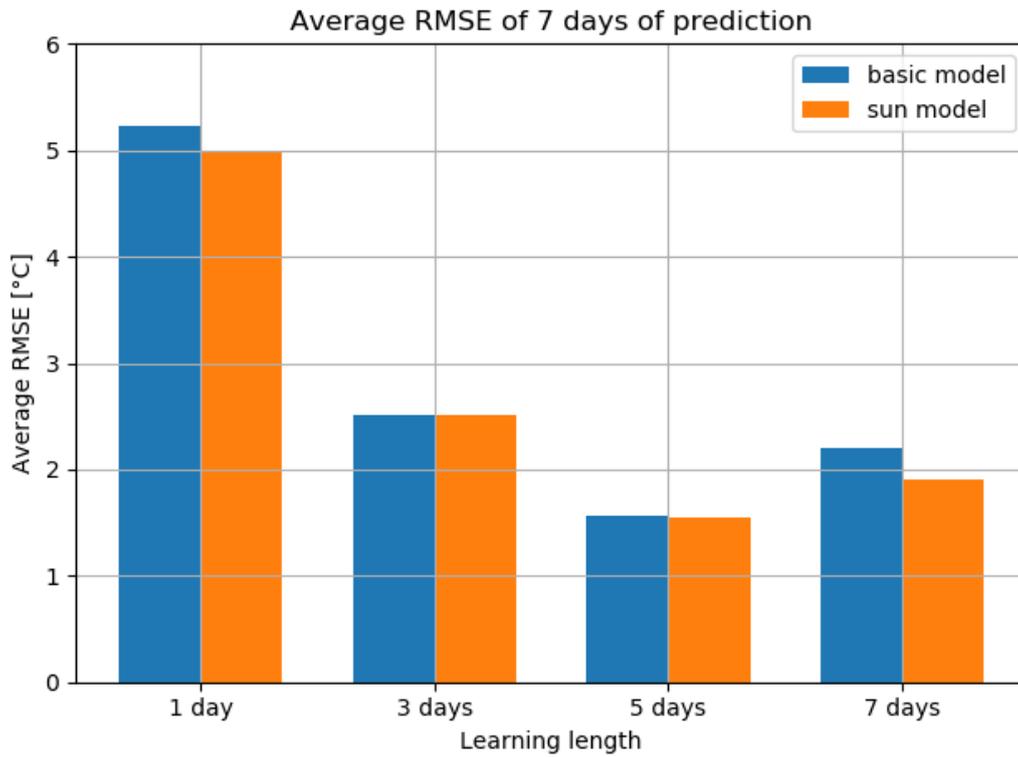
A. 3 Average RMSEs of the predictions starting on Sundays. Complete period.

Figure A. 4 shows the results obtained from the 3 days predictions made with the Basic model and the Sun model on the Selected period. The bars indicate the Sun model be slightly more accurate than the Basic Model on 1 day and 5 days of learning length. 5 days of learning length provides the lowest average RMSE, which are between 1 °C and 1.5 °C.



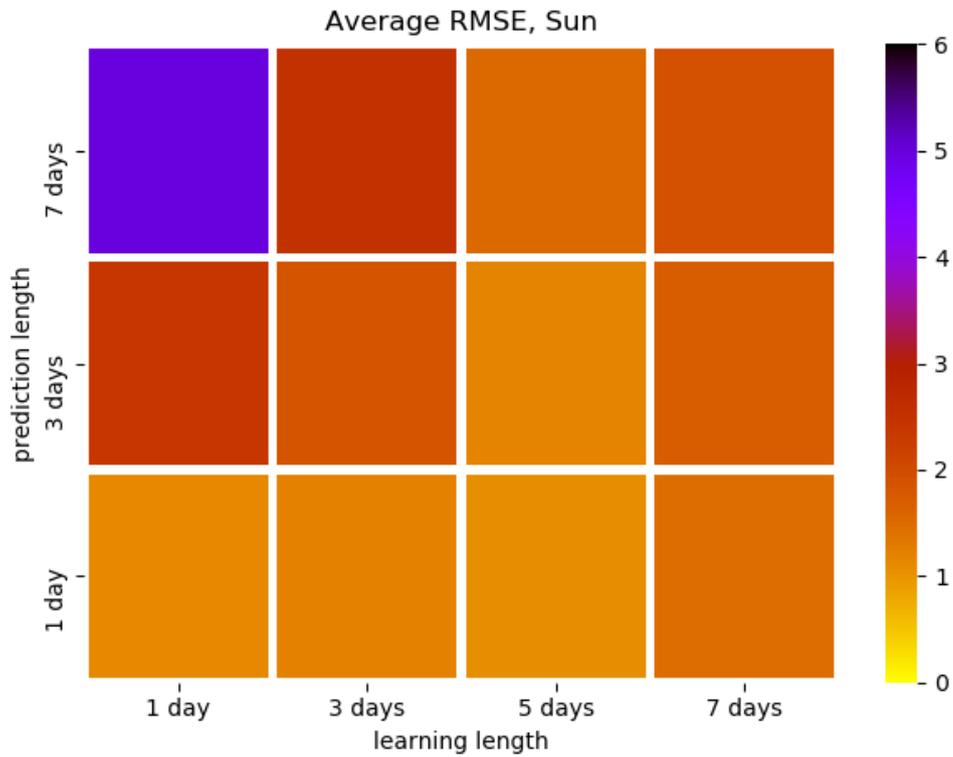
A. 4 Comparison between the average RMSEs of three days predictions from Basic model and from Sun model. Selected period.

Figure A. 5 shows the results obtained from the 7 days predictions made with the Basic model and the Sun model on the Selected period. On 1 day and 7 days of learning length, the Basic model column is higher than the Sun model ones, on the other learning lengths, the columns reach the same value. The average RMSE of the Basic model and the Sun model start from 1 day of learning length at about 5 °C and constantly decrease on 3 days and 5 days of learning length going down 2 °C. On 7 days of learning length, the values increase going back around 2 °C.



A. 5 Comparison between the average RMSEs of seven days predictions from Basic model and from Sun model. Selected period.

Figure A. 6 shows the average RMSEs of the prediction characterized by prediction length and learning length. The predictions are made with the Sun model on the Selected period. The value of each average RMSE is indicated by color according to the color scale on the right of the figure. The most accurate predictions are of 1 day and with the parameters of the model estimated on 1, 3 and 5 days. With 5 days of learning length also the predictions of 3 days have the average RMSE comparable with the average RMSE of 1 day prediction length.



A. 6 Heatmap of average RMSEs obtained from the validations with the Sun model. Selected period

## Bibliography

- [1] European Commission, "In focus: Energy efficiency in buildings | European Commission," *Focus Energy Effic. Build.*, no. February, 2020, [Online]. Available: [https://ec.europa.eu/info/news/focus-energy-efficiency-buildings-2020-feb-17\\_en](https://ec.europa.eu/info/news/focus-energy-efficiency-buildings-2020-feb-17_en).
- [2] A. Afram and F. Janabi-Sharifi, "Supervisory model predictive controller (MPC) for residential HVAC systems: Implementation and experimentation on archetype sustainable house in Toronto," *Energy Build.*, vol. 154, pp. 268–282, 2017, doi: <https://doi.org/10.1016/j.enbuild.2017.08.060>.
- [3] C. Corbin, G. Henze, and P. May-Ostendorp, "A model predictive control optimization environment for real-time commercial building application," *J. Build. Perform. Simul.*, vol. 2012, Jan. 2012, doi: 10.1080/19401493.2011.648343.
- [4] X. Pang, M. Wetter, P. Bhattacharya, and P. Haves, "A framework for simulation-based real-time whole building performance assessment," *Build. Environ.*, vol. 54, pp. 100–108, 2012, doi: <https://doi.org/10.1016/j.buildenv.2012.02.003>.
- [5] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, "Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system," *Energy Build.*, vol. 141, pp. 96–113, 2017, doi: <https://doi.org/10.1016/j.enbuild.2017.02.012>.
- [6] H. Harb, N. Boyanov, L. Hernandez, R. Streblov, and D. Müller, "Development and validation of grey-box models for forecasting the thermal response of occupied buildings," *Energy Build.*, vol. 117, pp. 199–207, 2016, doi: 10.1016/j.enbuild.2016.02.021.
- [7] D. H. Blum, K. Arendt, L. Rivalin, M. A. Piette, M. Wetter, and C. T. Veje, "Practical factors of envelope model setup and their effects on the performance of model predictive control for building heating, ventilating, and air conditioning systems," *Appl. Energy*, vol. 236, no. December 2018, pp. 410–425, 2019, doi: 10.1016/j.apenergy.2018.11.093.
- [8] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, and A. Bemporad, "Model Predictive Control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities," *Energies*, vol. 11, no. 3, 2018, doi: 10.3390/en11030631.
- [9] EnergyPlus Documentation, "EnergyPlus Documentation Getting Started with EnergyPlus Basic Concepts Manual -Essential Information You Need about Running EnergyPlus," p. 78, 2015, [Online]. Available: <https://doi.org/10.1016/j.dib.2019.104316>.
- [10] H. Ramírez, J. Jiménez-Cabas, A. Bula, and Modelica Association, "Modelica Language Specification," *Data Br.*, vol. 25, no. July, p. 104316, 2017, doi: 10.1016/j.dib.2019.104316.
- [11] OSMC, "OpenModelica User 's Guide," 2018.
- [12] M. Długosz, "Aggregation of state variables in an RC model," *Build. Serv. Eng. Res. Technol.*, vol. 39, no. 1, pp. 66-80r, Jul. 2017, doi: 10.1177/0143624417723159.
- [13] K. Arendt, M. Jradi, M. Wetter, and C. T. Veje, "ModestPy: An Open-Source Python Tool for Parameter Estimation in Functional Mock-up Units," *Proc. Am. Model. Conf. 2018, Oct. 9-10, Somb. Conf. Center, Cambridge MA, USA*, vol. 154, no. October, pp. 121–130, 2019, doi: 10.3384/ecp18154121.
- [14] A. Falcone and A. Garro, "Distributed Co-Simulation of Complex Engineered Systems by Combining the High Level Architecture and Functional Mock-up Interface," *Simul. Model. Pract. Theory*, vol. 97, no. August, p. 101967, 2019, doi: 10.1016/j.simpat.2019.101967.
- [15] C. Andersson, J. Akesson, and C. Fuhrer, "PyFMI: A Python Package for Simulation of Coupled Dynamic Models with the Functional Mock-up Interface," *Tech. Rep. Math. Sci.*, no. 2, pp. 1–40, 2016.
- [16] E. Jones, T. Oliphant, and P. Peterson, "SciPy: Open Source Scientific Tools for Python," Jan. 2001.
- [17] W. McKinney, "Data Structures for Statistical Computing in Python," *Proc. 9th Python Sci. Conf.*, vol. 1697900, no. Scipy, pp. 51–56, 2010, [Online]. Available:

- <http://conference.scipy.org/proceedings/scipy2010/mckinney.html>.
- [18] "PyCharm." [Online]. Available: <http://en.wikipedia.org/wiki/PyCharm>.
- [19] Y. Kwak, J. H. Huh, and C. Jang, "Development of a model predictive control framework through real-time building energy management system data," *Appl. Energy*, vol. 155, pp. 1–13, 2015, doi: 10.1016/j.apenergy.2015.05.096.
- [20] C. Visivo, "Qualita' edilizia."
- [21] I. Ballarini, A. Costantino, E. Fabrizio, and V. Corrado, "The Dynamic Model of EN ISO 52016-1 for the Energy Assessment of Buildings Compared to Simplified and Detailed Simulation Methods," *Proc. Build. Simul. 2019 16th Conf. IBPSA*, vol. 16, pp. 3847–3854, 2020, doi: 10.26868/25222708.2019.210431.
- [22] British Standards Institution, "ISO 52016: Energy performance of buildings -- Energy needs for heating and cooling, internal temperatures and sensible and latent heat loads," p. 204, 2017, [Online]. Available: <https://www.iso.org/standard/65696.html>.
- [23] J. Vivian, A. Zarrella, G. Emmi, and M. De Carli, "An evaluation of the suitability of lumped-capacitance models in calculating energy needs and thermal behaviour of buildings," *Energy Build.*, vol. 150, pp. 447–465, 2017, doi: 10.1016/j.enbuild.2017.06.021.
- [24] *Measurement of energy, demand and water savings / American Society of Heating, Refrigerating and Air-Conditioning Engineers*. Atlanta, GA: Ashrae, 2014.
- [25] J. Åkesson, M. Gäfvert, and H. Tummescheit, "JModelica---an Open Source Platform for Optimization of Modelica Models," in *6th Vienna International Conference On Mathematical Modelling, Vienna, Austria, 2009-02-11*, 2009.
- [26] Blum D and Wetter M, "MPCPy: An Open-Source Software Platform for Model Predictive Control in Buildings," *Proc. 15th IBPSA Conf.*, pp. 1381–1390, 2017, doi: 10.26868/25222708.2017.351.
- [27] "lbt-ladybug project description," 2020. Available: <https://pypi.org/project/lbt-ladybug/>.
- [28] "pysolar repository". Available: <https://github.com/pingswept/pysolar>.