Data-Driven Models Applied to Heating System Control Design
Based on Weather Forecast

BY

FEDERICO PICCA
Laurea, Politecnico di Torino, Turin, Italy, 2009

THESIS

Submitted as partial fulfilment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois at Chicago, 2012

Chicago, Illinois

Defense Committee:

William Worek, Chair and Advisor
W. J. Minkowycz
Marco Carlo Masoero, Politecnico di Torino

This thesis is dedicated to my grandparents
who are still looking out for me.

ACKNOWLEDGEMENTS

It is a great pleasure for me to thank all the people who helped me, during these last years, to complete successfully the joint honors degree program in Mechanical Engineering at the UIC of Chicago and at the Politecnico di Torino. My first thank goes to my Italian advisor Prof. Marco Masoero (Politecnico di Torino) who helped me, with kindness and professionalism, to complete this project on time despite his many commitments.
At the same time, I would like to thank Prof. William Worek of UIC, for his interest toward my work and his constant involvement along this last year.
A special thank goes to Ing. Jacopo Toniolo who helped to solve all the issues we found along the writing of this thesis. He helped me with friendliness and handiness, never leaving me alone in the difficult moments this project provided us.
Also, I would like to show my gratitude to the organizer and the coordinators of the TOP-UIC project: Prof. Muzio Gola, Alice Gili and Lynn Thomas.
A big thanks goes also to my family (especially to my sister) who was able to support me when I needed and to be strict with me when I deserved it.
I would like to thank also:
Matt, for having helped me to improve my English;
Dario, for having helped me to find the specific material I needed and with Matlab and for being always available for me;
Alberto, for having been next to me for all this time;
Evgeniya, for having been next to me in the last period of this journey and to support me and bear me day by day;
Francesca, for having been what she was to me;
Sara, for having been next to me at the beginning of this course;
Stefano and Francesco, for having shared this wonderful experience with me;
Laura, for having been the best classmate ever to me;
Ermand, Sonia, Enrico, Silvia, Luca, Francesco and all my friends.

FP

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

LIST OF FIGURES (CONTINUED)

SUMMARY

This project was promoted by a control unit producer. The target of the company is to implement a boiler control system, regulating with respect to the outdoor temperature (and not to the indoor one), which is cheap, efficient and easy to use. The first part of this work aimed to acquire the necessary knowledge about the System Identification methods and about the building. Using, then, the information collected in the first part, we related the heating system of a building with the Black-Box regression methods and, without considering any physical law but as comparison, a control system was developed. This control system is currently (winter 2011-2012) working in the building used for the study and, at the end of the heating season, (April 2012) we will collect the season data to see if our goals were reached or not.

CHAPTER 1

MOTIVATIONS AND AIMS

This section outlines the organization of this work as well as its motivations and aims. Nowadays we live in a critical context from an environmental perspective. Therefore it has become more and more important to implement policies inspired by the protection of the environment and the development of a sustainable economy, and focused on improving quality of life. Even architecture can help to achieve these goals by promoting the construction or rehabilitation of buildings so as to reduce their environmental impact and energy consumption.

Primary energy consumption for domestic purposes represents the 46% of overall energy consumption in Italy[1] and thus reducing inefficiencies and waste in this field may have important and beneficial effects on a global scale. Since 1997 when the Kyoto protocol was approved by 169 nations committing to reduce greenhouse gas emissions to address climate change, many steps have been made to promote sustainable architecture. The ISO 9004:2009[2] is one example of this. This regulation ("Managing for sustainability") not only provides a guide to improving building performance but also a guide to achieving sustainable success.

[1] Based on ENEA data for 2009, http://www.enea.it/it

[2] ISO 9004:2009, Geneva:2009

Italy, similarly, decided to cut CO_2 emissions by enforcing a regulation, which provides specific instructions on how to construct a building with a minor environmental impact by reducing energy consumption. The regulation we are speaking about is the '*Decreto Legislativo19/08/05 n°192*'[3] modified and completed by the '*Decreto Legislativo 29/12/06 n°311*'[4].

The aim of this project is to present a methodology for creating a control system that allows boilers to decide when to come on and off and how long for depending on outdoor temperature and considering the thermal capacity of the building. This means creating a cheap control program capable of predicting, using weather forecasts, the outlet temperature profile of the water taking into consideration the current temperature of the building (and so that the building needs to be heated just for the difference between the indoor temperature and the comfort temperature we want to obtain).

In order to present this methodology we wrote a program for the specific case of an existing building (built in 1939) in Corso Einaudi, Turin. Obviously are of that time also the criteria the heating system was built according to, and so they were not studied for energy savings. In order to improve the efficiency of the heating system the boiler was changed in 2010 before the heating season, but we wanted to see if it was possible to save even more gas by implementing the control system described before.

[3] Decreto legislativo 19/08/05 n°192 is a law decree about the energy efficiency of buildings and architecture (http://efficienzaenergetica.acs.enea.it/doc/dlgs_192-05.pdf).

[4] Decreto legislativo 29/12/06 n°311 is a law decree that corrects and completes the 19/08/05 n°192 (http://efficienzaenergetica.acs.enea.it/doc/dlgs_311-06.pdf).

There are two reasons why we didn't directly regulate the indoor temperature using thermostatic valves capable of turning the boiler off when the temperature of the building is within a certain comfort range:

1. The heating system we will consider actually heats two buildings, one next to the other, for a total of 36 apartments; therefore it would be quite difficult for only one boiler to heat all the apartments at the same temperature. NB: In Italy is very common that one heating system serves many apartments all together[5].
2. Even if we were able to install thermostatic valves in some of the apartments it could create problems with other people living in the building who might find the heating system turned off when they feel cold. In fact if the rooms where the thermostatic valves are installed were suddenly reaching the comfort temperature before the others, the control program would turn off the boiler, even if many rooms didn't reach the set point. This could cause the necessity of a reset of the comfort temperature to a higher level, with the risk of wasting gas, or it could create problems among neighbors.

To implement this control system it was decided not to use any thermodynamic equations, but black box correlations only. We thought that this approach would be more effective, albeit less general, for the specific building.

In the first chapter we will describe the building we studied. In the second the data collection we made and the various regulation strategies we applied during the collection period. In the third we will briefly describe some theories about System Identification.

---

[5] Based on PRIMO MASTRANTONI, *Riscaldamento Autonomo o Centralizzato?*, http://consumatori.myblog.it/archive/2011/11/10/riscaldamento-autonomo-o-centralizzato.html.

In the fourth and fifth chapters we will apply the theories explained in the third chapter implementing the models needed to create our control system. In the sixth chapter we will describe a methodology to find by trial and error the best regulation strategy with respect to the outdoor temperature. In the seventh chapter we will present the control program script we wrote on MatLab and in the eighth and final chapter we will discuss the conclusions of this project.

CHAPTER 2

THE BUILDING

The Building we are considering for our analysis is a residential condominium in Turin, located at the corner of Corso Luigi Einaudi and Corso Mediterraneo. This building was built during the Fascist period; to be more specific, building permission was given in 1939. The structure is divided in two parts (number 63 and number 65 of Corso Luigi Einaudi): each of them constitutes an independent apartment block. The two blocks have separate entrances, but the internal courtyard and the heating system are shared.

Figure 1: The building

The building is composed of eight floors and a basement where we the central heating system is installed. It occupies the northern corner of a square block. The Building is L-shaped, with the main facades facing Northeast and Northwest and with the internal courtyard facing South. Because of recent renovations the number of apartments differs from floor to floor, but the heated floor area is almost always the same.

The total height of the building is 28.7 meters for civic 65 and 26.1 meters for civic 63. Each floor is 3.15 meters high with the exception of the lower level, which is 3.9 meters high, and the last floor, which is 3.3 meters high.

The total number of apartments in the two blocks is 36, 13 in the first and 23 in the second. The penthouse is currently uninhabited, but it is being considered renewal together with the roof.

2.1 Construction Data

The information required to determine the building specifications were obtained using:

- Inspections to directly measure the dimensions and to observe the building structure;
- Apartments plans;
- Research in the Archivio Edilizio[6] of the city of Turin, where we were able to find the original projects, the building permission, etc.

The transmittance of the walls was evaluated according to the stratigraphy of the materials used, outlined – even if not in detail – in the original project. In order to verify the data we measured the infill thickness and we correlated the actual figures and the project

[6] "Archivio Edilizio" are the Consturction Records of a certain City.

data. The walls are made of full bricks of a total thickness of 55 cm. On the first two floors of the main facade an external coating of travertine was applied, while the last two floors present a plaster coating. The whole internal facade is covered with a plaster layer.

Thanks to a hole already present in the wall, we were able to determine the thickness of the plaster layer. The bearing walls are 50cm thick, while the internal walls are 30cm thick.

To evaluate the transmittance of the walls we didn't use a direct measurement, instead using a theoretical calculation based on the data obtained from the Abaco Edilizio[7].

Inspecting the penthouse we noticed that the roof has no insulation and is made of tiles laid on a wooden structure. There are some skylights and many cracks in the roof. The gradient of the roof is around 30 degrees and the apex is 3 meters high.

The concrete slab is 22cm thick and has an additional layer of bricks 8cm thick.

The floors are covered with a layer of ceramic or marble tiles 1cm thick; the ceilings present a plaster layer 1cm thick. The total thickness of the stratum between the two levels is 32cm.

The survey of the windows was made by a glazier who is responsible for the replacement of the existing single-glazed windows with a more efficient type of double-glazing. We measured the proportions of the glass, the thickness and the size of the frames, the characteristics of the wall in the sub window space and the box for the blinds.

The transmittance of the windows was determined by analyzing the type of windows installed. We measured the dimensions of the fixtures, of the glass surface, of the frame, and of the wooden vertical dividers. The glass is single-glazed, with a thickness of 4mm.

---

[7] "Abaco Edilizio" is the Consturction Abacus of a certain building.

The doors to the rooms are simple wooden frames, with a honeycomb internal structure. The transmittance is about 2.3 W/m^{2}K.

Below are some useful numerical data for the analysis of the building.

TABLE I: BUILDING DIMENSIONS

Building Dimensions		
Total Volume	9416.1	m^3
External Area	3612.8	m^2
Living Space	2227.0	m^2
Floors Height:		
Lower Level	3.6	m
Intermediates	3.1	m
Top Level	3.0	m

The configuration of the apartments and the living space is shown in the picture below:

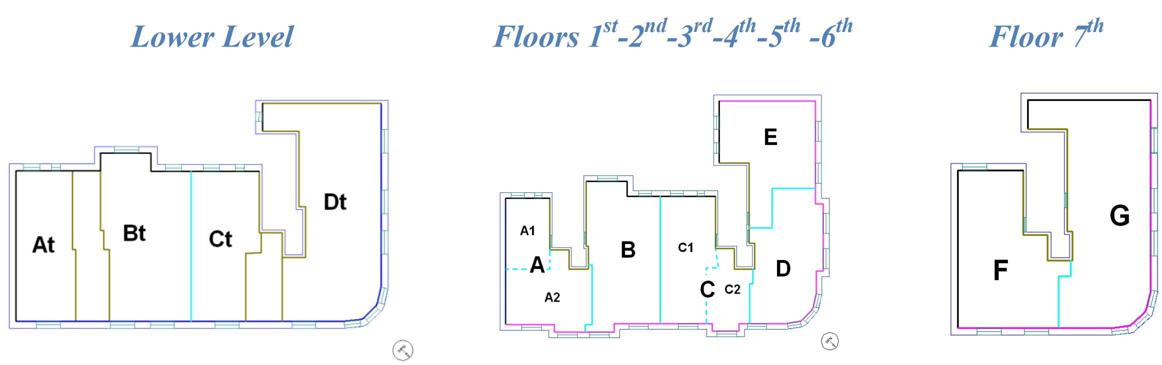


Figure 2: Apartment Configuration

TABLE II: STRATIGRAPHY OF THE WALLS

Stratigraphy of the Walls			
N.	**Layer Description (inside to out)**	**s [mm]**	**R [m2K/W]**
	External Wall with Marble Coating		
1	Plaster and Sand lime	15	
2	Solid Masonry	460	
3	Marble	30	
	Total Transmittance U	**[W/m²K]**	**1,222**
	External wall		
1	Plaster and Sand lime	2	
2	Solid Masonry	452	
3	Lime Mortar and Cement	5	
4	Clinker Coating	8	
	Total Transmittance U	**[W/m2K]**	**1,234**
	External Wall facing Internal Courtyard		
1	Plaster and Sand lime	15	
2	Brick Masonry	450	
3	Plaster and Sand lime	15	
	Total Transmittance U	**[W/m2K]**	**1,229**
	Internal Wall on the Stairs		
1	Plaster and Sand lime	10	
2	Solid Masonry	460	
3	Plaster and Sand lime	10	
	Total Transmittance U	**[W/m2K]**	**1,969**
	Walls under Window		
1	Plaster and Sand lime	15	
2	Brick Masonry	250	
3	Lime Mortar and Cement	15	
	Total Transmittance U	**[W/m2K]**	**1,774**
	Window box		
1	Fir	15	
2	Air Weakly Ventilated	300	
3	External Brick Masonry	120	
	Total Transmittance U	**[W/m2K]**	**2,020**
	Lower Level Floor		
1	Ceramic Tiles	10	
4	Concrete Subfloor	50	
5	Sand and Gravel Substrate	40	
6	Slab Brick	200	
	Total Transmittance U	**[W/m2K]**	**1,325**
	Slab Between Levels		
1	Ceramic Tiles	10	
2	Concrete Subfloor	35	
3	Sand and Gravel Substrate	40	
4	Slab Brick	200	
5	Plaster and Sand lime	15	
	Total Transmittance U	**[W/m2K]**	**1,690**
	Slab Under-Roof roof space		
1	Stiferite	100	
2	Concrete Subfloor	50	
3	Sand and Gravel Substrate	40	
4	Slab Brick	200	
5	Plaster and Sand lime	10	
	Total Transmittance U	**[W/m2K]**	**0,239**

2.2 Infrared thermography of the building

The building was built at a time when energy saving was not of much concern. Therefore it is logical that no special precautions were taken to prevent the formation of thermal bridges. In the absence of other means of investigation, it was necessary to make an infrared thermography of the building to see if any thermal bridges had been created, for example by balconies or floors between two levels. The instrument used is an infrared camera, ThermoTracer TH9100MV/WV. The emissivity was set to a value of 0.9.

Figure 3: NEC ThermoTracer Camera TH9100MV/WV

Unfortunately it was not possible to access the individual units or the internal courtyard when the measurements were taken. However, there were no signs of heat dispersion from an outside view of the facades facing the street. This suggests that the building is massive enough to ensure good thermal behavior throughout the structure. Below the thermal images taken during the final period of the daily heating cycle, at around 7pm, are shown.

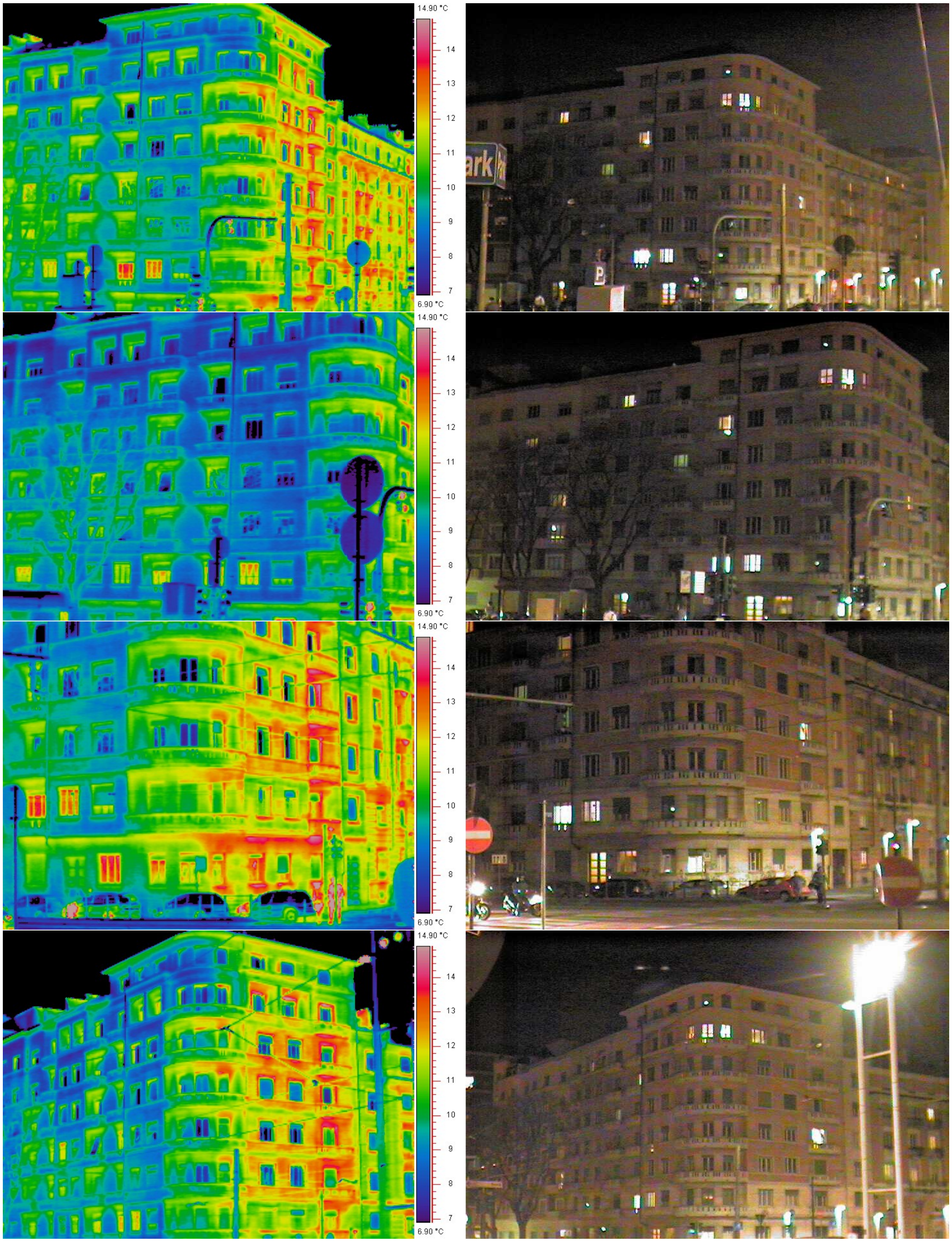


Figure 4: Thermal images of the building

In contrast to the transmittance measurements, which for reasons of time were excluded, the infrared thermography is a technique that allows rapid and non-invasive analysis.

We need to make some clarifications in order to avoid misinterpretation of the data. Firstly, no measures have been taken to reduce reflection from the glass surfaces. For this reason it makes no sense to consider the color associated to the frames, because the surface has an emissivity completely different from that of the rest of the building. In addition, it makes no sense to make a quantitative comparison between the Northwest surface, which is warmer, and the Northeast one, which is colder. The temperature of the two walls is different because of the thermal capacity and of the higher solar irradiation on the Northwest wall at the end of the day. No useful considerations can be made on the facade, because it had been heated to a temperature higher than that produced by the heating system.

Focusing on the Northeast facade we can observe hot spots due to localized scattering, corresponding to the window boxes and parapets, which are thinner than the rest of the wall. The temperature differences do not exceed 2-3°C with respect to the solid wall. The balconies do not constitute a problem: the temperature is about the same as the surrounding walls. The last image shows how the first floor slab comprises localized losses of a certain amount; however they are sufficiently small as not to require specific measures to reduce them. We assumed that this temperature difference was due to the location and the characteristics of the first floor, which is likely to be heated more than the others, being higher, to ensure a comfortable temperature for the occupants. It is possible that the heat, moving upwards, makes the local temperatures of the ceiling higher than that of the other floors. Moreover it is unlikely that the first floor slab was designed with different criteria to the rest of the building.

A second point of interest is the roof slab. This confirms the absence of insulation observed during the survey and provides a starting point for possible renovation.

Overall, this analysis shows how the structure presents a reasonably uniform thermal behavior. Furthermore the installation of additional insulation in the facades turned out not to be necessary.

2.3 The Heating System

The heating system consists, for both buildings (civic 63 and 65), of a heat generation system shared between the two structures and with two independent distribution systems, one for each block. The plant was designed for winter heating only. Hot water is produced in each apartment by an individual boiler.

2.3.1 The thermal power plant

The heating plant is located in the basement of civic 65. In 2008 the heating system was entirely renovated in order to increase its efficiency. In particular the old traditional boiler was substituted with a condensing boiler and a new control system was installed with a thermostatic valve, which regulates the amount of hot water flowing in the radiators, for each apartment

The saving due to these improvements was estimated to be around 15-20% for the production of hot water at 80°C and 20-30% for the production of hot water at 60°C.

The installed condensing boiler is a PYROGAS VARINO 65-300

Figure 5: Condensing boiler Pyrogas Varino 200

It presents the following main characteristics:

- *Operating pressure 4.0 bar*
- *Test pressure 6.0 bar*
- *Boiler flow and return flanges PN 6*
- *Max. Operating Temperature 90 ° C*
- *Minimum return Temperature no limitation*
- *Maximum content of CO_2 with CH_4 11.7%*
- *(Dry flue gas) and LPG 13.7%*

The qualities required to the water are:

First filling: *Total hardness: <10° F (100 mg eq. $CaCO_3$ / l, 84 mg $MgCO_3$ / l)*

Aqua integration: *Total hardness: <1° F(10 mg eq. $CaCO_3$ / l, 8.4 mg $MgCO_3$ / l)*

Water circulation:

- *Total hardness: <1 ° F*
- *PH value (20 ° C): 8.3 - 9.5*
- *Phosphates (PO4): <30 mg / l*
- *Chloride (Cl): <50 mg / l*
- *Oxygen (O2): <0.1 mg / l*
- Chloride (Cl): <50 mg / l
- Oxygen (O2): <0.1 mg / l

2.3.2 The Distribution Circuit

The boiler is placed in a double distribution circuit: in both circuits there is a four-way mixing valve connected to the climate control unit, which mixes the supply and return of the boiler distribution plant.

Figure 6: Four-ways mixing valve

On the outlet branches, upstream and downstream of the mixing valve, some ball valves are installed with a manual tap. Circulation from the boiler to the distribution is guaranteed by a pair of 600 W twin "inverter-controlled variable flow pump". The pump for civic 65 has been recently replaced, model D40-120F Grundfos Magna. A dedicated third pump allows circulation in the condensation circuit.

Figure 7: Twin pump number 1

Figure 8: Twin pump number 2 (recently replaced)

There are three expansion tanks: one for the 35liter anti-condensation circuit and one for each of the two plants. The difference arises from the different water content of the two plants. The function of the expansion tank is to compensate for density variations in the water. Normally the vessels are designed to withstand the maximum thermal expansion of the water mass that would occur on the hot branch, and they are installed on the cold branch to maximize safety.

Figure 9: Expansion tank civic 63

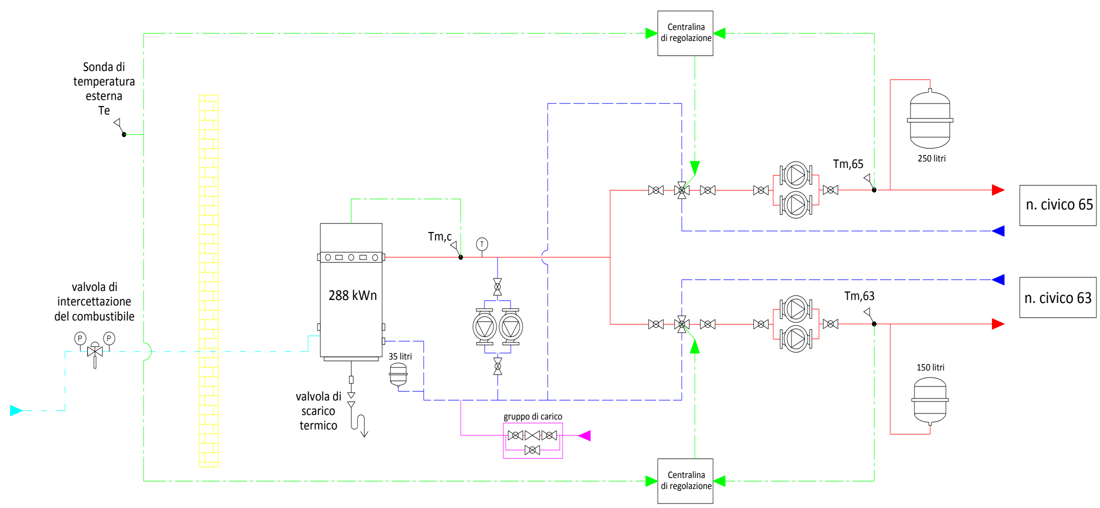
Figure 10: Expansion tank civic 65

Figure 11: Expansion tank boiler

The gas meter is placed in the internal courtyard. The chimney comes out vertically from the boiler and goes outside the building up to the roof.

There are no thermal storage systems. This means that the boiler operating times, in some periods, are highly intermittent.

The table below shows the functional diagram of the system:



Boiler and Distribution System Outlet Pipe

Boiler and Distribution System Return Pipe

Aqueduct Connection Pipe

Control and Regulation System

Gas Pipe

Figure 12: Heating system scheme

2.3.3 Heat Distribution System

The heat distribution is realized through the use of risers, which supply the hot water to each apartment and through a horizontal pipe network in each of them. This was a very common solution in the construction industry before the implementation of new laws on energy saving.

There are, respectively, four and five risers in civics 63 and 65. The diameter of the columns differs according to the capacity needed at the installation point.

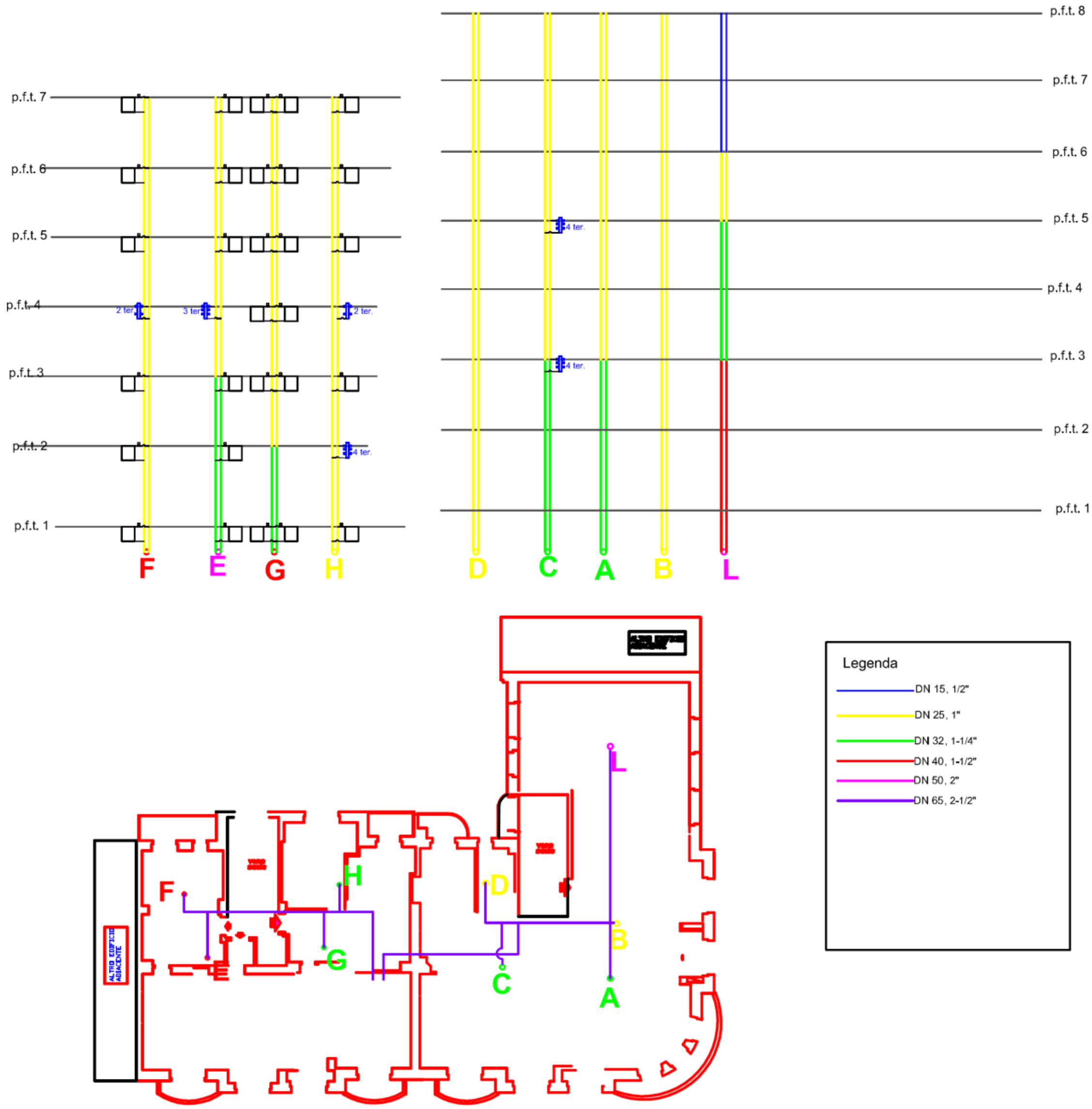


Figure 13: Risers and heat distribution scheme

The terminals are radiators made of cast iron, capable of receiving hot water at a supply temperature of about 70-80 ° C. These units are installed in each room of the two buildings.

2.4 The Control System

The old heating system regulator was entirely controlled by two units inside the boiler room. This is a compensation regulator based on the detection of the outside temperature, by a sensor, on which the supply temperature of the distribution system is based. This temperature can be obtained by turning on the boiler to its two-speed modulation. The new heating system has the following characteristics:

- Control in the boiler room
- Modern condensing boilers are equipped with an automatic adjustment system. The external temperature is directly plugged into the boiler unit, which regulates the flow temperature linearly with the outside temperature. The power of the modulating burner can be adjusted according to the return temperature in order to ensure the achievement of the predicted flow temperature. The modulation range is between 30% and 100%, with a range of temperatures between 45 and 90°C.
- Room control with thermostatic valves

The control of power supply, as required by the law, is provided by a set of thermostatic valves installed on each radiator inlet pipe.

The thermostatic valve is a control device that modulates the flow with respect to the ambient temperature. The opening is a function of the temperature difference between the ambient and the set point. When the temperature is equal to the one the valve has been set to, it remains closed, to gradually open up as the ambient temperature drops below the set level. This type of control is defined “retroactive”, since it occurs after the external and internal conditions have acted on the system, providing a response by raising or decreasing the power output coming from the radiator. The control stability mainly depends on the reaction time: the quicker it is, the lower the proportional band, in which there are oscillations of the temperature. The ideal situation would be to maintain a constant temperature.

Thermostatic valves can use different types of feedback adjustments: the most common is the proportional one with a temperature band of 0.5°C, 1°C or 2°C; but there are more sophisticated devices on the market, such as the following:

- PI, where a convergence of the temperature is obtained avoiding any "offset";
- PID, which in addition guarantees a faster response, but this often causes excessive flow fluctuations.

There are three different types of sensor, characterized by different reaction times.

- Wax Sensors: High heat conduction and thermal capacity. The reaction time of the thermostatic valves is very long, in the order of hours. They are not suitable for domestic use where thermal transients are quite fast;
- Liquid Sensors: These devices present a high thermal capacity and, as for the previous ones, they are characterized by long reaction times, in the order of several minutes. They can be used for residential purposes but are not very efficient;
- Condensing Gas Sensors: This category of sensors present a low heat capacity that, combined with conduction and convection heating, allows a rapid response to temperature changes and reduces the response times. The cost is considerably higher, but comfort conditions are easier to achieve.

The last category of sensors is the one installed in the building we are going to study, during the renewal of the heating system. These are high precision devices, which are equipped with a pre-setting system for initial calibrations.

Figure 14: Thermostatic valves

CHAPTER 3

CONTROL SYSTEM AND OPERATION MODES

As stated in the introduction, our goal is to create a control system able to provide the boiler with the temperature at which the water has to be heated and to decide on an ON-OFF strategy according to the weather forecasts. Therefore we want to create a control system which is able to regulate the amount of hours the boiler works depending on the external temperature and considering the influence of the building's thermal capacity on the indoor temperature. In order to do this we need to correlate the outdoor and indoor temperatures and the water temperature coming out from the boiler.

Commonly a "Climatic curve" correlating the outdoor temperature and that of the water is used to control the boiler and the heating system. In this project we would like to go a little further and to create a control system, which is also able to consider the current temperature of the building while regulating it. The "Climatic curve", in fact, is an empirical curve that does not consider the thermal insulation of the building, the efficiency of the radiators and many other important characteristics of the building. Therefore our first aim is to create a mathematical model that is able to correlate the outdoor temperature to the water temperature inside the heating system, considering the dynamics of the temperatures inside the building, and thus the dynamics of the entire building. To do this it was decided to use a black-box approach which, by definition, needs a large set of data to be reliable.

3.1 Data Collection

After having studied the building from the structural point of view, we had to consider which data we would have needed for our study and where to place the sensors to collect it in an efficient way.

The data we decided to collect are:

- Date
- Time
- Outdoor Temperature
- Boiler ON-OFF
- Natural gas consumption
- Water supply Temperature (boiler outlet)
- Water return Temperature (boiler inlet)
- Indoor ambient Temperature

To have a complete enough data set, the measures were taken every five minutes for a period of four months between November 2010 and March 2011.

The internal temperature was measured both in one of the apartments and in a storage room where usually people don't go. The reason why we chose to measure it in two different places is that: the apartment gives more an idea of the real temperature perceived by the people inside the building (they can open the windows, turn on the oven, etc.); while the storage room gives more the measure of the ideal temperature that there would be inside the building if the heating system only was controlling it. In other words, an estimate on the effect of occupant's behavior was obtained from the comparison.

Both the room of the apartment and the storage room were chosen to be on the Northern side of the building. This choice is due to the fact that the north one is the coldest side of the building and to the consideration that if we are able to warm up a room on that side of the building, we would, for sure, also be able to heat all the rooms on the other sides. Furthermore these rooms were chosen from the ones available at the top of the building. This is because, as we have already described, the distribution system is vertical and so the heating water decreases its temperature on its way up to the last floor because of the heat losses due to the non-sufficient insulation of the pipes. Therefore the water arriving to the higher floors is always colder than the one heating the lower ones.

In this chapter we will briefly describe the tools we used to collect data and the heating operation modes we used to evaluate and compare different control strategies. We want then to evaluate which control strategy is more convenient from the gas consumption point of view in order to have an idea of which control modes it's worth considering during the next modeling step.

3.2 Measuring Instruments

To detect the data we are interested in we used the following tools:

- A Pt1000 sensor to measure the outdoor temperature;
- A NTC sensor to measure, again, the outdoor temperature;
- A Logger Comark N2013 to detect the apartment's temperatures;
- A diaphragm meter to measure the gas consumption;
- A Central control unit.

3.2.1 Pt1000 Sensor

The Pt (Platinum Thermo-Resistance) sensors are thermal resistors with high temperature coefficient, this means that they increase their resistance with rising temperatures. They are able to provide an excellent precision for a huge range of temperatures (between -200°C and 854°C).

Figure 15: Pt1000 sensors

3.2.3 NTC Sensor

The NTC (Negative Temperature Coefficient) sensors, are thermal resistors with low temperature coefficient. The NTC usually have a negative temperature coefficient (between -6% and -2% per degree Celsius), which means that their resistance decreases with increasing temperatures. They are used to either measure the temperature directly (in electronic thermometers or boilers) or as control elements in electrical and electronic circuits (for example to increase or decrease a current or a voltage as a function of the temperature).

Figure 16: NTC sensor

3.2.3 Logger Comark N2013

It is a temperature and humidity recorder presenting a LCD display. Compact, lightweight and durable it is designed for many applications including food, pharmaceutical and chemical industries. The recorders are ideal for construction services, scientific experiments, production processes and HVAC systems.

The Logger Comark N2013 has the following characteristics:

1. LCD display for temperature and humidity readings, alarm indicator.
2. The measurement ranges are:
    - Temperature -20 ° / -4 ° F to + 60 ° C / +140 ° F
    - Humidity 0 to 97 % RH non-condensing.

Figure 17: Logger Comark N2013

3.2.4 Gas meter

To know the heating system gas consumption, we used a gas meter modified by adding an additional electronic device in order to enable the data acquisition and transmission through an electronic system. The remote reading allowed us to take a record of the user consumption every 5 minutes.

Figure 18: Gas meter

3.2.5 Central control unit

The plant control unit is based on a PLC (programmable logic controller), installed in the boiler room and connected to a UMTS modem that allows a remote management of the plant. Through this PLC system it is possible to change the boiler-operating program in time (e.g. night shutdown).

PLC is an electronic device very similar to a computer interfacing with external devices through a special programming language. The PLC operations block diagram is the following:

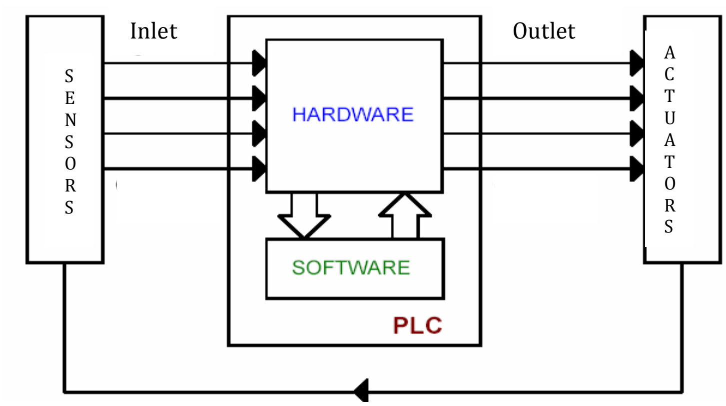


Figure 19: PLC operations block

3.3 Operation Modes

In order to make the data set more heterogeneous and complete we didn't just use one regulation strategy during the data collection period. We adopted 5 different control modes:

- *24 hours ON*;
- *18 hours ON* with overnight shut down;
- *13 hours ON* with double switch-off;
- *24 hours ON Bi-climatic* regulation;
- *Bi-climatic WF* (Weather Forecast) regulation.

In the following chart we can see how we changed the boiler control strategy over time:

TABLE III: BOILER CONTROL STRATEGY LEGEND

Legend	
Symbol	Description
A	ON-OFF, Climatic 1
Y	*18h ON* with overnight shut down
H	*13h ON* with double-switch off
I	*24h ON* with mono-climatic curve
Bi Climatic Curve	*24h ON* with 2 different regulations curve (1 for the night and 1 for the day)
Bi Climatic WF	As *Bi-climatic*, switched off between 11:30-16:00
Sample WF	*Sample ON-OFF* based on Weather Forecast with Climatic curve regulation

TABLE IV: BOILER CONTROL STRATEGY FOR THE HEATING SEASON 2010-2011

Day	Type	Tout [°C]	Day	Type	Tout [°C]	Day	Type	Tout [°C]	Day	Type	Tout [°C]
11/21/10	*A*	6.74	12/22/10	*I*	2.99	01/22/11	*Bi Climatic*	1.76	02/22/11	*Bi Climatic WF*	5.45
11/22//10	*A*	7.11	12/23/10	*Y*	5.02	01/23/11	*Bi Climatic*	0.42	02/23/11	*Bi Climatic WF*	5.89
11/23/10	*A*	6.08	12/24/10	*Y*	6.04	01/24/11	*Bi Climatic*	1.39	02/24/11	*Bi Climatic WF*	4.45
11/24/10	*A*	5.77	12/25/10	*Y*	6.29	01/25/11	*Bi Climatic*	1.13	02/25/11	*Bi Climatic WF*	4.33
11/25/10	*A*	4.60	12/26/10	*Y*	3.10	01/26/11	*Bi Climatic*	1.14	02/26/11	*Bi Climatic WF*	5.53
11/26/10	*A*	3.94	12/27/10	*Y*	-0.19	01/27/11	*Bi Climatic*	2.83	02/27/11	*Bi Climatic*	5.48
11/27/10	*A*	1.78	12/28/10	*Y*	-0.89	01/28/11	*Bi Climatic*	3.97	02/28/11	*Bi Climatic*	3.02
11/28/10	*A*	1.58	12/29/10	*Y*	-0.32	01/29/11	*Bi Climatic*	2.41	03/01/11	*Sample WF*	6.27
11/29/10	*A*	4.17	12/30/10	*I*	2.63	01/30/11	*Bi Climatic*	1.19	03/02/11	*Sample WF*	4.96
11/30/10	*A*	2.67	12/31/10	*I*	2.35	01/31/11	*Bi Climatic*	4.13	03/03/11	*Sample WF*	2.87
12/01/10	*H*	2.02	01/01/11	*I*	2.59	01/01/11	*Bi Climatic*	3.47	03/04/11	*Sample WF*	4.13
12/02/10	*H*	1.55	01/02/11	*Y*	1.52	02/02/11	*Bi Climatic*	2.77	03/05/11	*Sample WF*	6.80
12/03/10	*H*	1.33	01/03/11	*Y*	2.19	02/03/11	*Bi Climatic*	4.00	03/06/11	*Sample WF*	8.82
12/04/10	*H*	1.07	01/04/11	*Y*	1.87	02/04/11	*Bi Climatic WF*	5.16	03/07/11	*Sample WF*	5.60
12/05/10	*H*	0.97	01/05/11	*I*	1.61	02/05/11	*Bi Climatic WF*	6.73	03/08/11	*Sample WF*	5.77
12/06/10	*I*	2.07	01/06/11	*I*	1.85	02/06/11	*Bi Climatic*	7.31	03/09/11	*Sample WF*	5.36
12/07/10	*H*	4.34	01/07/11	*I*	2.75	02/07/11	*Bi Climatic WF*	7.68	03/10/11	*Sample WF*	7.36
12/08/10	*I*	6.30	01/08/11	*I*	3.67	02/08/11	*Bi Climatic WF*	6.86	03/11/11	*Sample WF*	8.26
12/09/10	*I*	6.49	01/09/11	*I*	4.46	02/09/11	*Bi Climatic WF*	6.96	03/12/11	*Sample WF*	7.49
12/10/10	*I*	4.27	01/10/11	*I*	5.00	02/10/11	*Bi Climatic WF*	6.83	03/13/11	*Sample WF*	5.38
12/11/10	*I*	3.51	01/11/11	*Y*	6.34	02/11/11	*Bi Climatic WF*	6.81	03/14/11	*Sample WF*	8.63
12/12/10	*I*	3.16	01/12/11	*I*	4.99	02/12/11	*Bi Climatic WF*	7.07	03/15/11	*Sample WF*	8.46
12/13/10	*I*	3.28	01/13/11	*I*	5.25	02/13/11	*Bi Climatic*	7.36	03/16/11	*Sample WF*	8.47
12/14/10	*I*	1.60	01/14/11	*I*	3.76	02/14/11	*Bi Climatic WF*	7.69	03/17/11	*Sample WF*	9.77
12/15/10	*I*	-0.10	01/15/11	*Sample WF*	2.90	02/15/11	*Bi Climatic*	5.74	03/18/11	*Sample WF*	12.06
12/16/10	*I*	-1.36	01/16/11	*Sample WF*	2.70	02/16/11	*Bi Climatic*	4.50	03/19/11	*Sample WF*	13.13
12/17/10	*I*	-2.72	01/17/11	*Sample WF*	3.23	02/17/11	*Bi Climatic*	5.27	03/20/11	*Sample WF*	9.76
12/18/10	*I*	-2.38	01/18/11	*Sample WF*	4.68	02/18/11	*Bi Climatic*	7.53	03/21/11	*Sample WF*	8.55
12/19/10	*I*	-0.16	01/19/11	*Sample WF*	5.69	02/19/11	*Bi Climatic*	7.69	03/22/11	*Sample WF*	10.57
12/20/10	*I*	0.96	01/20/11	*Sample WF*	0.96	02/20/11	*Bi Climatic*	6.63	03/23/11	*Sample WF*	11.44
12/21/10	*I*	1.84	01/21/11	*Bi Climatic*	2.04	02/21/11	*Bi Climatic WF*	6.47	03/24/11	*Sample WF*	13.11

From the chart above we can see two more options with respect to the ones listed: *A* and *Sample WF*. The first one (*A*) is the control strategy that the boiler is able to provide by itself following the two climatic line (which we will describe later) and turning off the system when the power of the boiler decreases under a certain limit. The second one (*Sample WF*) is a control strategy based on the manual change of the amount of hours the boiler is working. We did this knowing how the boiler worked for the previous part of the winter season and basing our control strategy on the weather forecast.

The first thing we had to do with all these data was to divide them according to the control strategy. Once we did it we were able to evaluate every single strategy from a qualitative point of view:

3.3.1 *24 hours ON Mono-Climatic* Control Straegy

In this first case the boiler adjusted the water flow temperature in function of the outside temperature. As we can see from the following graph, which considers one week's data as a significant sample, the relationship between the two temperatures can be considered linear. The equation describing the regression line is:

$$T_{outlet} = -1.5217 \cdot T_{outdoor} + 62.826$$

It can be obtained considering the outlet temperatures of all the days in which the boiler was told to work following a *24h ON mono-climatic* control strategy and plotting them with respect to the correspondent outdoor temperatures. Plotting the graph on Excel and asking it for a regression line we obtained the previous equation. The graph from which it was obtained is:

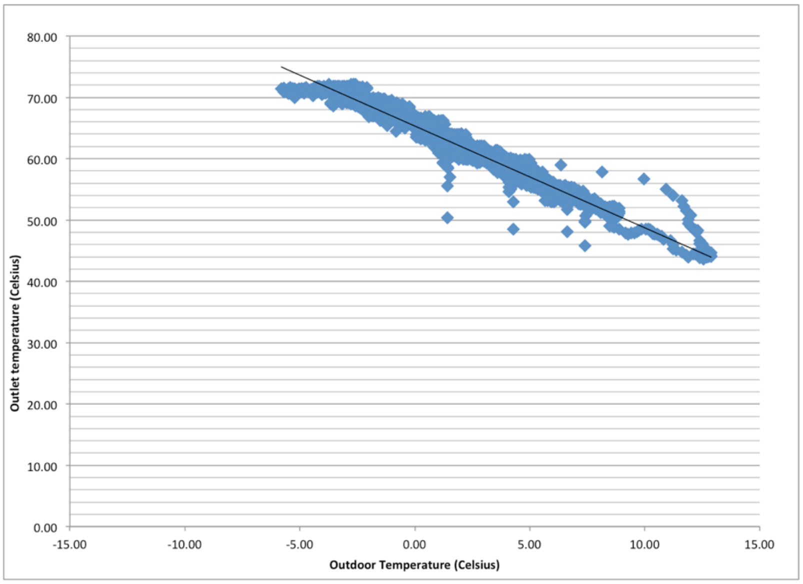


Figure 20: Mono-Climatic line from experimental data

This line correlating the outdoor temperature with the inside one is the one the boiler uses when it has to adjust by itself the water temperature. Plotting in the same graph the Outdoor Temperature, the Indoor Temperature, the water supply temperature and the Mean Power with respect to the Time we have:

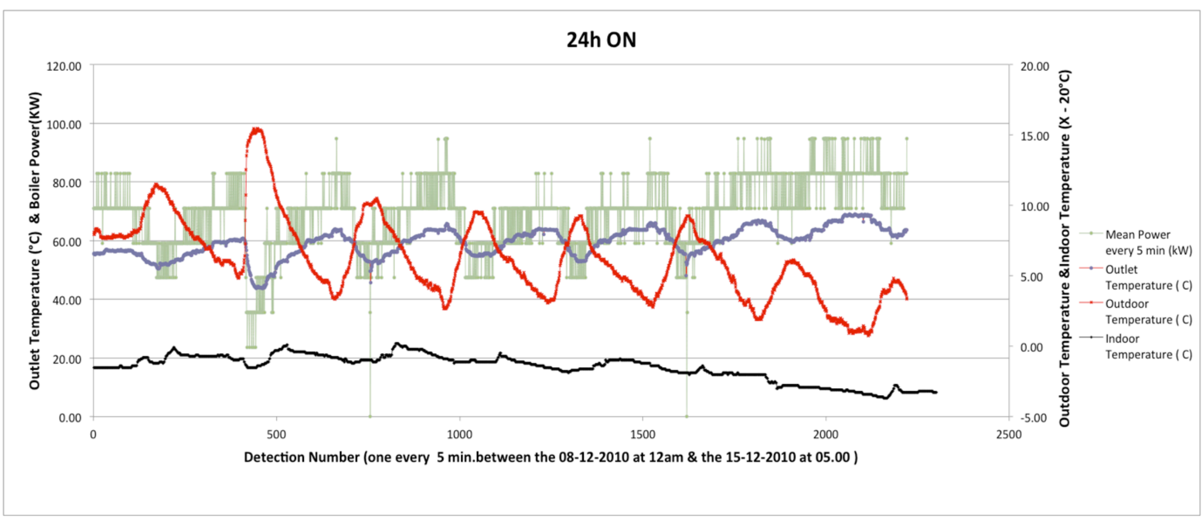


Figure 21: Outdoor temperature, indoor temperature, water supply temperature and mean power for the *24h ON* control strategy

As we can see, using this control strategy we were able to maintain an almost constant indoor temperature. Furthermore the flow temperature is perfectly identical to the outdoor temperature. This explains the linear relationship between the two. The boiler was running 24 hours a day and it was just increasing or decreasing its power according to the outdoor temperature. Looking at the power line, we can see how the boiler never delivers 100 kW: it remains at an average power of 72 kW. One more thing we can notice from the previous graph is that there are two points where the boiler power is equal to zero. These are probably due to the control system sensitivity that, under 30 kW, shuts the boiler down.

3.3.2 *18 hours ON* with Overnight shut down Control Strategy

In this case the boiler was working between 5am and 12am, and turned off during the other hours. Plotting in the same graph the Outdoor Temperature, the Indoor Temperature, the flow temperature (coming out from the boiler) and the Mean Power with respect to the Time we have:

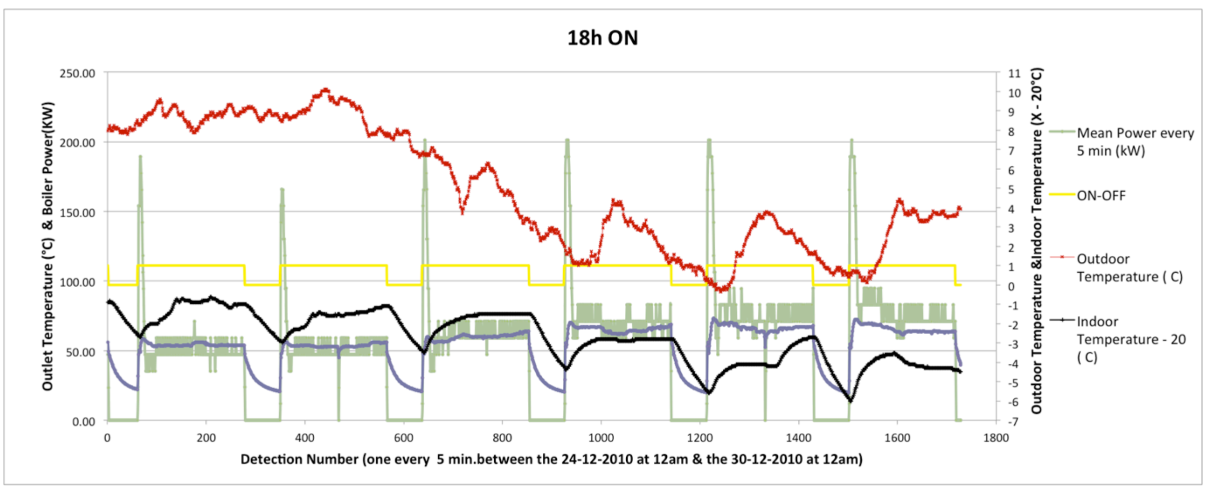


Figure 22: Outdoor temperature, indoor temperature, water supply temperature and mean power for the *18h ON* control strategy

As we can see, also using this control strategy we were able to matain the indoor temperature almost constant.

During the 18 hours a day the boiler was running, the flow temperature was linearly related to the outside temperature. For 10-15 minutes after each ignition, the boiler was working at maximum power (over 200 kW) before stabilizing to the steady state power (around 83kW)

3.3.3 *13 hours ON/OFF* with Double Switch-off Control Strategy

In this case the regulation curve was always the same as the first case, with the only difference that the boiler was working just 13 hours a day (from 6am to 2:30pm and again from 5:30pm to 10pm).

Plotting on the same graph the Outdoor Temperature, the Indoor Temperature, the flow temperature (coming out from the boiler) and the Mean Power with respect to the Time we have:

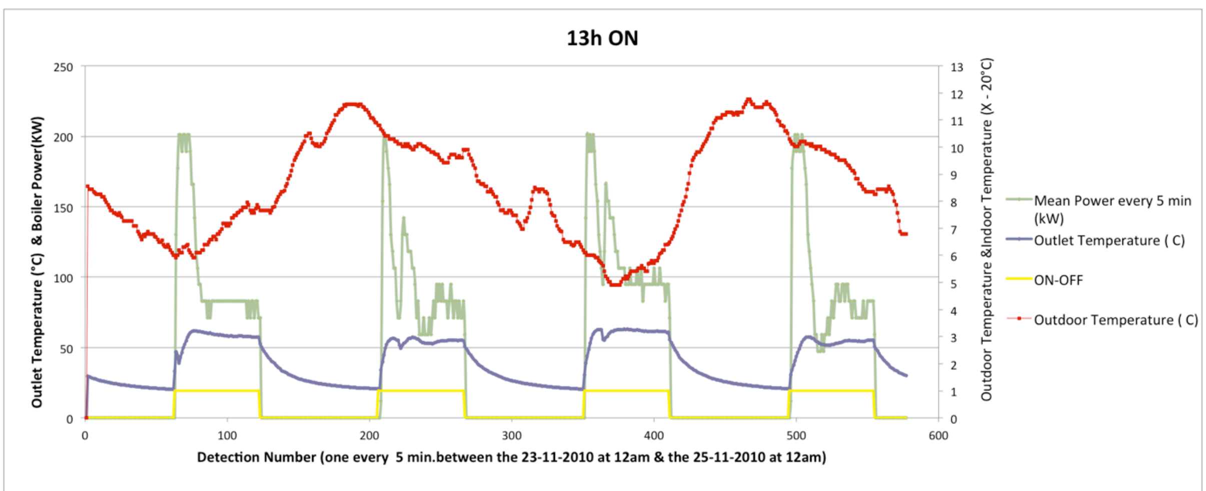


Figure 23: Outdoor temperature, indoor temperature, water supply temperature and mean power for the *13h ON* control strategy

As we can see from the graph this control strategy was not able to provide a constant temperature inside the apartments. Furthermore, as it happened in the previous case, the boiler worked at its maximum power for several minutes (even 30 min) every time it was turned on; before stabilizing at around 85 kW.

3.3.4 *24 hours ON Bi-Climatic* Control Strategy

Adopting this control strategy the regulation unit works following two different climatic curves, one for the day and one for the night. The idea is that during the night people are sleeping under their blankets and so there is no need to heat as much as during the day. Furthermore the apartments are supposed to be already warm from the day's heating.

In the following graph we represented the two regulation lines the boiler is following: the red for the day, the blue for the night.

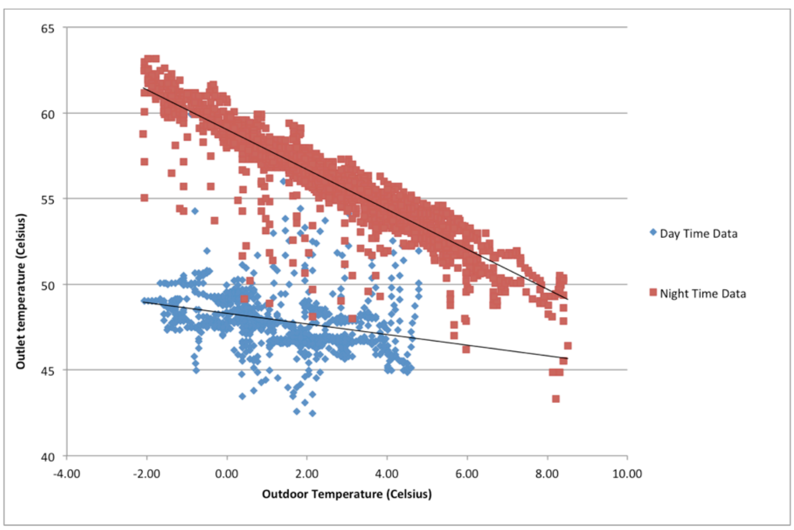


Figure 24: Bi-Climatic lines from experimental data

Plotting on the same graph the outdoor temperature, the indoor temperature, the flow temperature (coming out from the boiler) and the mean power with respect to the:

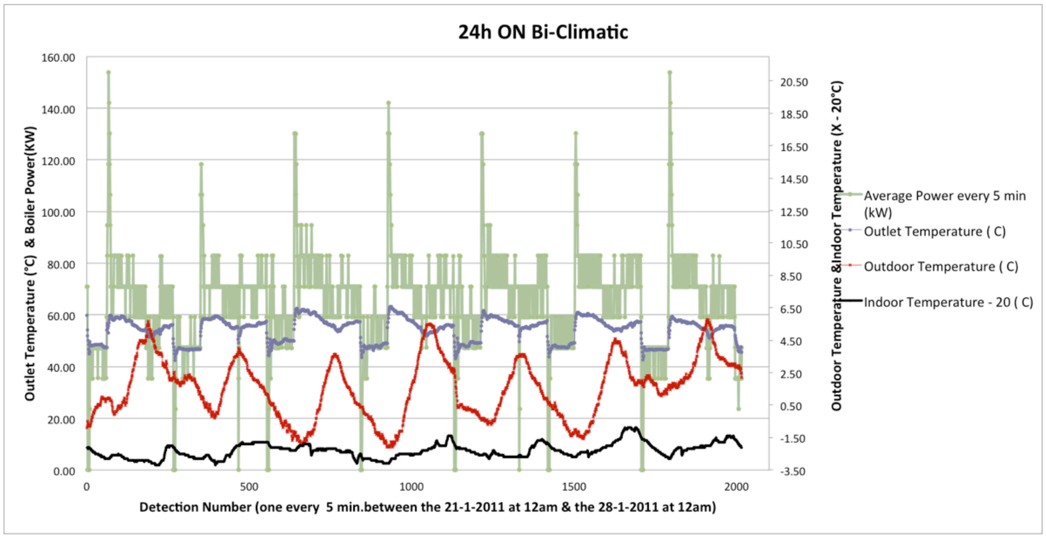


Figure 25: outdoor temperature, indoor temperature, water supply temperature and mean power for the *24h ON Bi-Climatic* control strategy

As can be seen from the graph, the indoor temperature follows the outdoor temperature trend, but the range of indoor temperatures is quite small and satisfies the comfort requirements. The boiler was working 24 hours a day modulating its power with respect to the outside temperature and to the time (as explained it changes between day and night). It was never working at its maximum power and the average power was around 75kW.

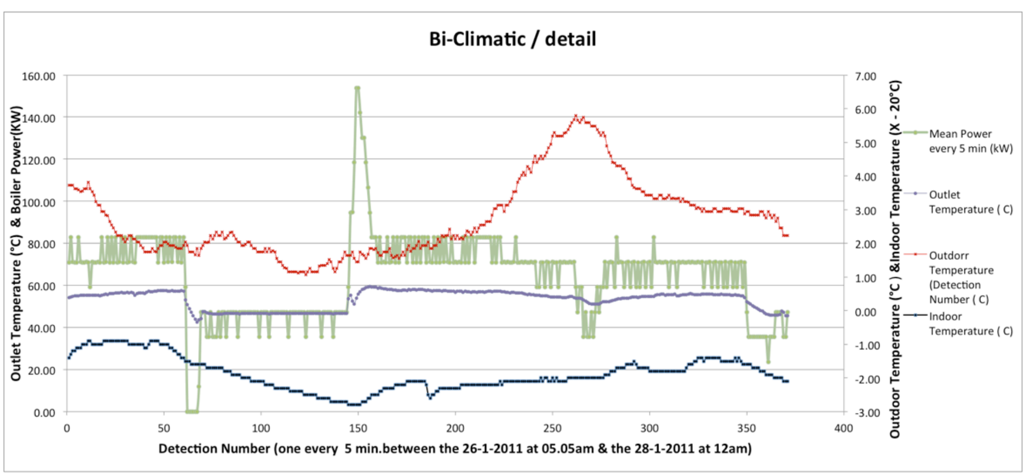


Figure 26: One day detail of the outdoor temperature, indoor temperature, water supply temperature and mean power for the *24h ON Bi-Climatic* control strategy

3.3.5 *Bi-climatic WF* (Weather Forecast) Control Strategy

The control strategy is similar to the one made for the previous case, with the only difference that the boiler is turned off between 12pm and 4pm and, if the outdoor temperature is higher than 4°C, between 12am and 5pm.

Plotting on the same graph the outdoor temperature, the indoor temperature, the flow temperature (coming out from the boiler) and the mean power with respect to the time we have:

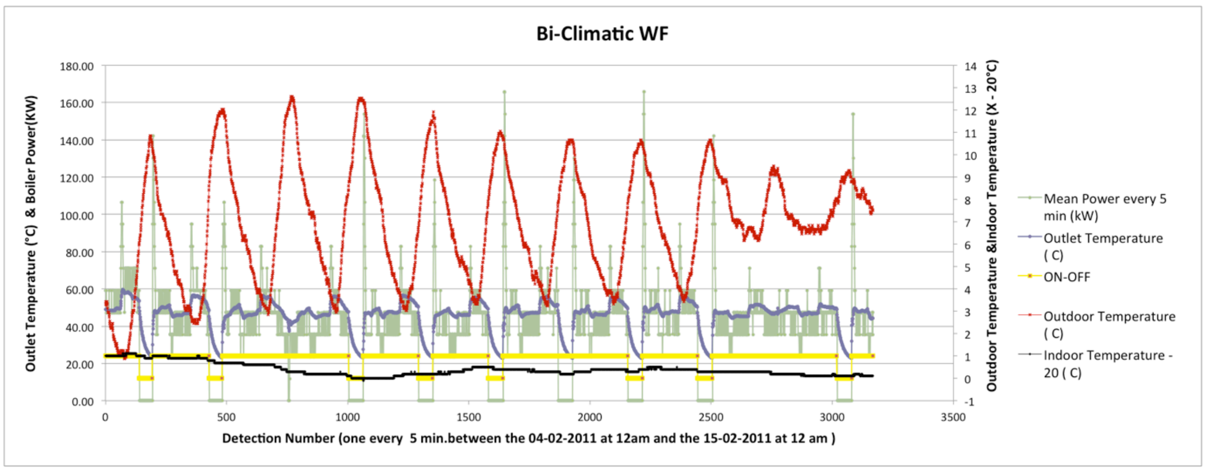


Figure 27: Outdoor temperature, indoor temperature, water supply temperature and mean power for the *Bi-Climatic WF* control strategy

As can be seen from the graph, the situation was very similar to the previous one. Without forgetting that we made this regulation when the outdoor temperature never went under 1°C, we can notice that the boiler never had to work at its maximum power (the max power released was of 168 kW, and never for more than 5 minutes). Furthermore the mean boiler power was around 52 kW.

3.3.6 Results

Plotting the gas consumption for each regulation strategy with respect to the outdoor temperature we obtained:

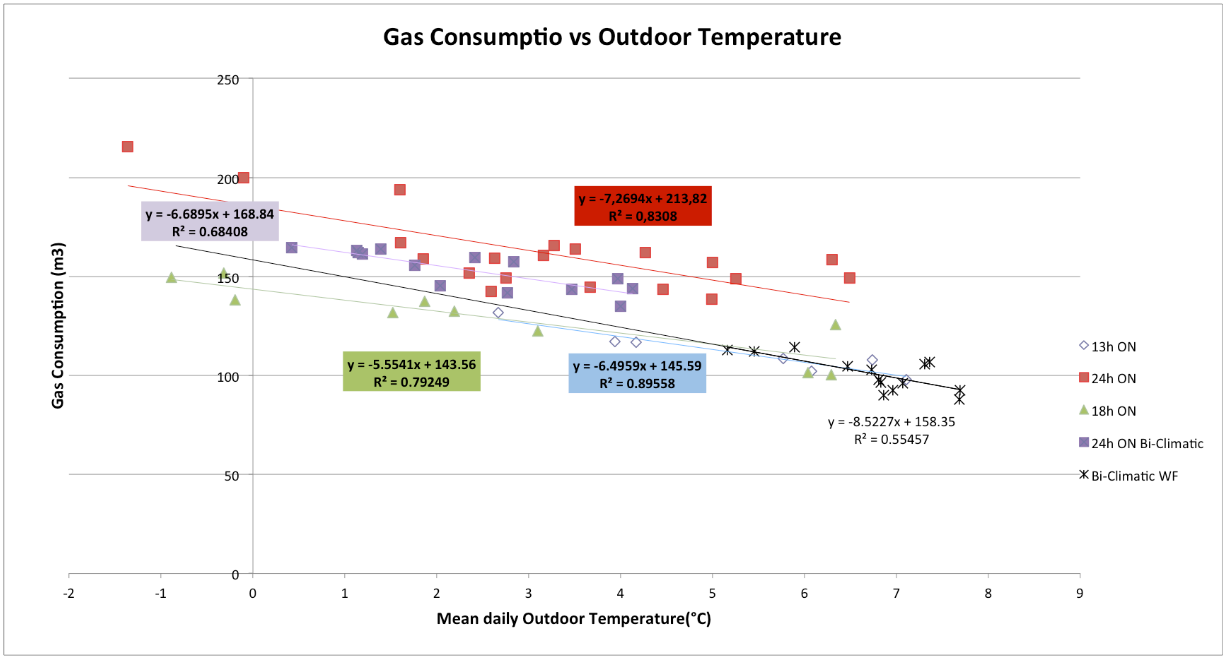


Figure 28: Gas consumption vs. outdoor temperature for every regulation strategy

This chart shows us the relationship between gas consumption and the outside temperature. Comparing the different control modes, we can immediately see that the *24 hours ON* consumes more than the others and that it never uses more than 100 kW, which is half of the boiler nominal power. For these reasons we can not consider this kind of control strategy as convenient

From the graph we can also see that the *13 hours ON* consume far less than the *24h ON* control, but we can't consider it as an option because it is not able to ensure the required comfort for people. In contrast to this it has to be said that this control strategy is the one that makes the boiler work at its nominal power for the longest period every day, thus justifying the expense of a 200kW machine. The *24h Bi-climatic* regulation doesn't differ much from the regular *24h ON* setting. The biggest difference is in the maximum power required by the boiler: 156 kW, which is enough to justify the choice made regarding the 200kW boiler. Therefore if we had to choose between the *24h ON Mono-Climatic* and the *24h ON Bi-Climatic*, we would choose the second one more for the better usage of the boiler rather than because of the lower consumption. This control strategy doesn't seem to be the best one as regards consumption.

With regard to the *18h ON control* mode we can see that the consumption is lower than in the other cases (especially at low temperatures). Furthermore the boiler works at its maximum power for a good amount of time every day, justifying the choice of the installed boiler over a less powerful one. From the point of view of consumption this one seems to be the best regulation strategy, even if we cannot make a complete comparison because the bi-climatic WF was sampled only at a time in which the weather was not cold enough.

The table below shows the historical data for the gas consumption and expenditures from 2003 to 2006 – before the traditional boiler was exchanged with a condensation one.

TABLE V: HISTORICAL GAS CONSUMPTION AND EXPENDITURES DATA

Season:	Gas consumption [m^3]	Euro [€]
2003-4	34000	€ 28.900
2004-5	35000	€ 29.750
2005-6	36000	€ 30.600
CH_4 cost	0,85	€/m^3

The next table shows the results of an evaluation for the new control methods.

TABLE VI: CONTROL STRATEGIES COMPARISON AND EVALUATION

	24h ON	18h ON	13h ON	24h ON Bi-Climatic	Bi-Climatic WF
Mean Outdoor Temperature	**3.33**	**3.33**	**3.33**	**3.33**	**3.33**
Dalily Gas Consumption (m^3)	**170.86**	**120.53**	**118.57**	**148.95**	**123.29**
Estime of the Seasonal Gas Consumption (m^3)	**19934.81**	**14082.57**	**14188.40**	**17884.23**	**14428.38**
Seasonal Money Expense (€)	**16944.59**	**11970.19**	**12060.14**	**15201.60**	**12264.13**
Expense Comparison with 24h ON Regulation Strategy (€)	**0.00**	**4974.40**	**4884.44**	**1742.99**	**4680.46**
Expense Comparison in Percentage	**0.00**	**29%**	**29%**	**10%**	**28%**
Percentage of m^3 of Gas Saved with Respect to the Historical Case	**-43.04%**	**-59.76%**	**-59.46%**	**-48.90%**	**-58.78%**
m^3 of Gas Saved with Respect to the Historical Case (m^3)	**-15065.19**	**-20917.43**	**-20811.60**	**-17115.77**	**-20571.62**
Energy Saved with Respect to the Historical Case (kWh)	**-59332.10**	**-82380.28**	**-81963.48**	**-67408.00**	**-81018.35**
Money Saved with Respect to the Historical Case (€)	**12805.41**	**17779.81**	**17689.86**	**14548.40**	**17485.87**
Yearly Saving for Each Family (€)	**400.17**	**555.62**	**552.81**	**454.64**	**546.43**

Let's start considering the *24h ON Mono-Climatic* control. As we can see from the table the mean daily consumption of this control strategy is the highest (170.8 kW) of the whole set. However it is the regulation that provides the best comfort condition. Since today energy saving is a crucial topic, it would be difficult to choose this control mode.

As regards the *18h ON* control, the system saves up to 50 m^3 of gas (daily) and it produces almost € 5,000 of savings a year. Therefore it would make each family save €156 a year. If in addition to the control system we also consider the saving from the condensation boiler we can see that the saving comes to €556 a year per family. Which means a total saving of 17.780. However, even if this system would provide high savings, the interior comfort is not totally satisfied (especially for the lowest temperatures). The *13h ON* control has a value not too far away from the previous case, but, for the same reason, we cannot accept this strategy as applicable.

Speaking now about Bi-climatic control and considering the *24h ON* control, we can see that the values are quite similar to the standard *24 hours ON Mono-Climatic* control. The only difference is that in this case the boiler is better used for its nominal power making this solution more interesting. As regards the *Bi-climatic WF* regulation we can draw a similar conclusion. Even if the outdoor temperature wasn't low enough when we applied it to the system, from the collected data we can see that this regulation mode uses the boiler better and produces high gas savings. Therefore it would probably be the best strategy to adopt as a regulating mode because, unlike the *13h ON* and the *18h ON* regulations, with this control mode the internal comfort is ensured.

Finally we can propose some economic considerations: comparing the gas consumption before and after the condensation boiler was installed we were able to compute the payback time of the boiler for the various regulation strategies. In the following table the results are show with an interest rate of 5%:

TABLE VII: PAY BACK TIME FOR THE VARIOUS CONTROL STRATEGIES

Interest Rate		5%	5%	5%	5%	5%	5%	5%
Pay Back Time (years)	0	1	2	3	4	5	6	7
18h ON (Discounted Cash Flow - €)	-74000	-57067	-40940	-25581	-10954	2977		
13h ON (Discounted Cash Flow - €)	-74000	-57153	-41107	-25826	-11273	2588		
Bi-Climatic WF (Discounted Cash Flow - €)	-74000	-57347	-41487	-26382	-11996	1705		
Bi-Climatic (Discounted Cash Flow - €)	-74000	-60144	-46949	-34381	-22412	-11013	-157	10182
24h ON (Discounted Cash Flow - €)	-74000	-61804	-50189	-39128	-28593	-18559	-9004	97

Plotting the results we just obtained on a graph we have:

Figure 29: Pay back time for the various control strategies

As can be seen the Bi-climatic WF provides the system with a payback time equal to the 18h control system (and of the 12h one). As we can see from these first observations, what we supposed before about the advantages of adopting this system is clear.

CHAPTER 4

MODEL STRUCTURES

The model-based control design process involves modeling the plant to be controlled, analyzing and synthesizing a controller for the plant, simulating the plant and controller, and deploying the controller.

For control design engineers, National Instruments provides a powerful set of mathematical algorithms in the Mathwork, MATRIXx and LabVIEW System Identification tools, that reduce the effort required to develop models for model-based design. Unlike modeling from first principles, which requires an in-depth knowledge of the system under consideration, system identification methods can handle a wide range of system dynamics without requiring knowledge of the actual system physics.

4.1 The System Identification Procedure[8]

4.1.1 The Data Record

The input-output data are sometimes recorded during a specifically designed identification experiment, where the user may determine which signals to measure and when to measure them and may also choose the input signals. The objective with *Experiment Design* is thus to make these choices so that the data become maximally informative, subject to any constraints.

[8] Based on L. LJUNG, *System Identification: Theory for the user*, 2nd Edition, Perentice Hall, 1999, Passim

4.1.2 The Set of Model or the Model Structure

A set of candidate models is obtained by specifying within which collection of models we are going to look for a suitable one. This is, without doubt, the most important and, simultaneously, the most difficult choice in the System Identification procedure. It is here that *a priori* knowledge and engineering intuition and insight have to be combined with formal properties of models. Sometimes the model set is obtained after careful modeling. Then a model with some unknown physical parameters is constructed from physical laws and other well-established relationships. In other cases standard linear models may be employed, without reference to the physical background. Such a model set, whose parameters are basically viewed as vehicles for adjusting the fit to the data and do not reflect physical considerations in the system, is called a *Black Box*. Model sets with adjustable parameters with physical interpretation may accordingly be called *Grey Boxes*. Generally speaking a model structure is a parameterized mapping from past inputs and outputs Z^{t-1} to the space of the model outputs:

$$\hat{y}(t|\theta) = g(\theta, Z^{t-1})$$

Where:

- $\hat{y}$ is the estimate of the output;
- θ is the finite dimensional vector used to parameterized the mapping.

4.1.3 Determining the "Best" Model in the set, guided by the data

This is the *identification method*. The assessment of model quality is typically based on how the models perform when they attempt to reproduce the measured data.

4.1.4 Model Validation

After having settled on the preceding three choices, we have, at least implicitly, arrived at a particular model: the one in the set that best describes the data according to the chosen criteria. It, then, remains to test whether this model is "good enough" and if it is valid for its purpose. Such tests are known as *Model Validation.* They involve various procedures to assess how the model relates to observed data, to prior knowledge and to its intended use. Deficient model behaviors in these respects make us reject the model, while good performances will develop a certain confidence in it. We have to observe that *a model can never be accepted as a final and true description of the system*. Rather, it can at best be regarded as an "adequate" description of certain aspects that are of particular interest to us.

4.1.5 The System Identification Loop

The System Identification procedure has a natural, logical flow: first collect data then choose a model set, then pick the "best" model in the set. It is quite likely, though, that the model first obtained will not pass the model validation tests. We must then go back and revise the various steps of the procedure.

The model may be deficient for a variety of reasons:

- The numerical procedure failed to find the best model according to our criteria;
- The criteria were not well chosen;
- The model set was not appropriate, in that it did not constitute an "adequate" description of the system;
- The data set was not informative enough to provide guidance in selecting good models.

The great part of an Identification application in fact consists of addressing these problems, in particular the third, in an iterative manner, guided by prior information and the outcomes of previous attempts. Interactive software is obviously an important tool for handling the iterative character of this problem.

4.2 Model Structures[9]

The selection of a suitable model structure is prerequisite before its estimation. The choice is based upon understanding of the physical systems. Three types of models are common in system identification: the black-box model, grey-box model, and the user-defined model. The black-box model assumes that systems are unknown and all model parameters are adjustable without considering the physical background. You cannot adjust all the parameters arbitrarily. The grey-box model assumes that part of the information about the underlying dynamics or some of the physical parameters are known and the model parameters might have some constraints. The user-defined model assumes that commonly used parametric models cannot represent the model you want to estimate. You can define your special system model by using a template VI with a predefined input-output interface.

Table of Contents

1. Black-box Models;
2. Grey-box Models;
3. User defined Models;
4. Conclusions.

[9] Based on L. LJUNG, *System Identification: Theory for the user*, 2nd Edition, Perentice Hall, 1999, Passim

4.2.1 Black-box Models

A variety of parametric model structures are available to assist in modeling an unknown system. Parametric models describe systems in terms of differential equations and transfer functions. These models provide insight into the system physics and compact model structures. It is often beneficial to test a number of structures to determine the best one. A parametric model structure is also known as a black-box model, which defines either a continuous-time system or a discrete-time system.

Generally, we can describe a system using the following equation, which is known as the general-linear polynomial model or the general-linear model.

$$y(t) = G(q)u(t) + H(q)e(t)$$

Where:

q is the forward shift operator $[q \cdot u(t) = u(t-1)]$

$u(t)$ and $y(t)$ are the input and output of the system respectively;

$e(t)$ is zero-mean white noise, or the disturbance of the system;

$G(q)$ is the transfer function of the deterministic part of the system;

$H(q)$ is the transfer function of the stochastic part of the system.

The general-linear model structure, shown in Figure 1, is:

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

$$\text{Where: } \begin{cases} A(q) = 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ B(q) = 1 + b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b} \\ C(q) = 1 + c_1 q^{-1} + \cdots + c_{n_c} q^{-n_c} \\ D(q) = 1 + d_1 q^{-1} + \cdots + d_{n_d} q^{-n_d} \\ F(q) = 1 + f_1 q^{-1} + \cdots + f_{n_f} q^{-n_f} \end{cases}$$

It provides flexibility for both the system dynamics and stochastic dynamics. However, a nonlinear optimization method computes the estimation of the general-linear model. This method requires intensive computation with no guarantee of global convergence.

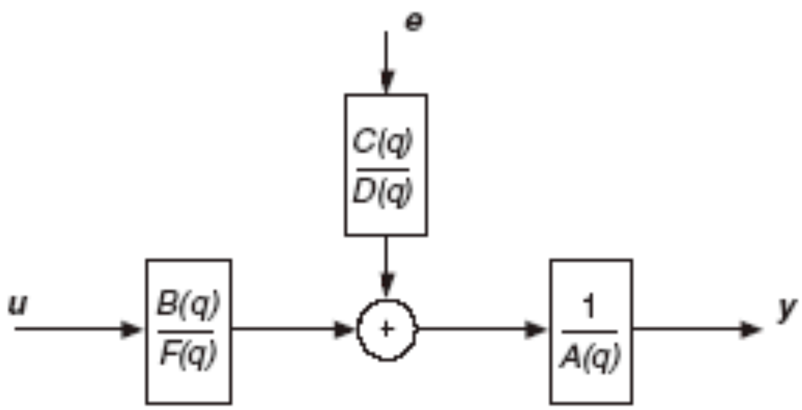


Figure 30: General linear model

Simpler models that are a subset of the General Linear model structure are possible. By setting one or more of A(q), B(q), C(q), D(q) or F(q) polynomials equal to 1 we can create simpler models such as AR, ARX, ARMAX, Box-Jenkins, and output-error structures. Each of these methods has its own advantages and disadvantages and is commonly used in real-world applications. For any particular problem the choice of the model structure depends on the dynamics and the noise characteristics of the system. Using a model with more freedom or parameters is not always better as it can result in the modeling of nonexistent dynamics and noise characteristics. This is where physical insight into a system is helpful.

4.2.1.1 AR Model Structure

The AR model structure is a process model used in the generation of models where outputs are only dependent on previous outputs. No system inputs or disturbances are used in the modeling.

$$A(q)y(t) = e(t)$$

This is a very simple model that is limited in the class of problems it can solve. Strictly speaking this means that the AR model structure is the model for a signal, not a system. Time series analyses, such as linear prediction coding commonly use the AR model.

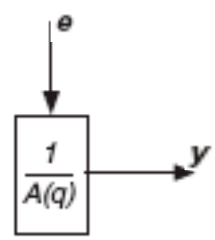


Figure 31: AR model structure

4.2.1.2 ARX Model Structure

The ARX model, shown in Figure 3, is the simplest model incorporating the stimulus signal.

$$A(q)y(t) = B(q)u(t) + e(t)$$

The estimation of the ARX model is the most efficient of the polynomial estimation methods because it is the result of solving linear regression equations in analytic form. Moreover, the solution is unique. In other words, the solution always satisfies the global minimum of the loss function. The ARX model is therefore preferable, especially when the model order is high. The disadvantage of the ARX model is that disturbances are part of the system dynamics. The transfer function of the deterministic part $G(q)$ of the system and the transfer function of the stochastic part $H(q)$ of the system have the same set of poles. This coupling can be unrealistic. The system dynamics and stochastic dynamics of the system do not always share the same set of poles. However, you can reduce this disadvantage if you have a good signal-to-noise ratio. When the disturbance e(t) of the system is not white noise, the coupling between the deterministic and stochastic dynamics can bias the estimation of the ARX model. Usually the model order is set higher than the actual model order to minimize the equation error, especially when the signal-to-noise ratio is low. However, increasing the model order can change some dynamic characteristics of the model, such as the stability of the model.

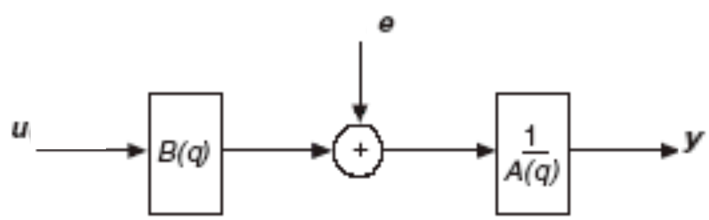


Figure 32: ARX model structure

4.2.1.3 ARMAX Model Structure

Unlike the ARX model, the ARMAX model structure includes disturbance dynamics.

$$A(q)y(t) = B(q)u(t) + C(q)e(t)$$

ARMAX models are useful when we have dominating disturbances that enter early in the process, such as at the input. For example, a wind gust affecting an aircraft is a dominating disturbance early in the process. The ARMAX model is more flexible in the handling of disturbance modeling than the ARX model.

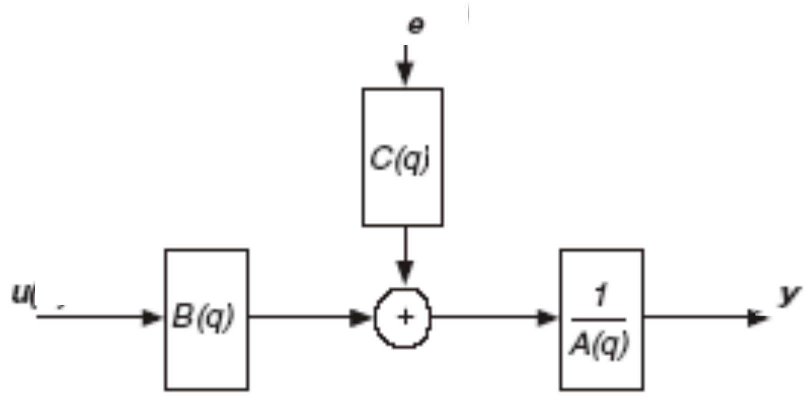


Figure 33: ARMAX model structure

4.2.1.4 Box-Jenkins Model Structure

The Box-Jenkins (BJ) structure provides a complete model with disturbance properties modeled separately from system dynamics.

$$y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

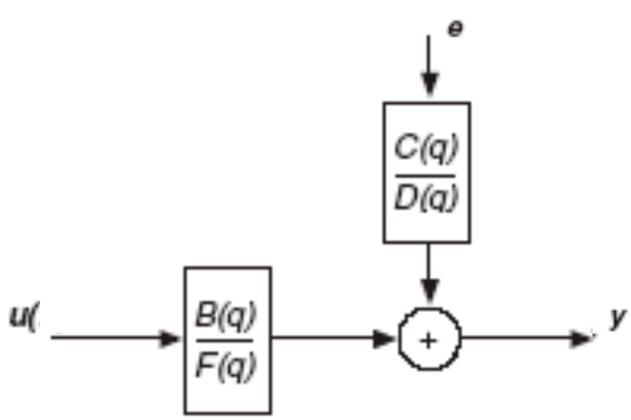


Figure 34: Box-Jenkins model structure

The Box-Jenkins model is useful when you have disturbances that enter late in the process. For example, measurement noise on the output is a disturbance late in the process.

4.2.1.5 Output-Error Model Structure

The Output-Error (OE) model structure describes the system dynamics separately. No parameters are used for modeling the disturbance characteristics.

$$y(t) = \frac{B(q)}{F(q)}u(t) + e(t)$$

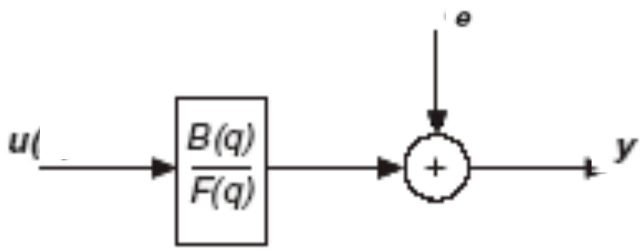


Figure 35: OE model structure

4.1.2.5 Transfer function Model

For stochastic control, the above general-linear polynomial models are commonly used because these models separately describe the deterministic and stochastic parts of a system. However, in classical control engineering, the deterministic part of the system is more important than the stochastic part. Transfer function models are commonly used to describe only the deterministic part of the system. Transfer function models can describe both continuous-time and discrete-time systems. The following equations describe a continuous-time and discrete-time transfer function model respectively.

$$y(t) \;=\; G(s)u(t)$$

$$y(k) \;=\; G(z)u(k) + e(k)$$

Where:

- y(t) and y(k) are the system outputs;
- G(s) and G(z) is the transfer function between the stimulus and the response;

- $u(t)$ and $u(k)$ are the system inputs.

Usually the transfer function model structure is used to represent single-input and single-output (SISO) physical systems or multiple-input and single-output (MISO) physical systems. For more complicated multiple-input and multiple-output (MIMO) physical systems, use the state-space model structure.

In the LabVIEW System Identification Toolkit, transfer function model structure is also used to describe physical systems in some special circumstances, including systems in closed-loop and systems with step response being output. The transfer function model can also be estimated in the frequency domain using frequency response function (FRF) data.

4.2.1.6 State-Space Model

The previous classical parametric system identification methods minimize performance function, which is based on the sum of squared errors. These methods work well in many cases. However, for complex systems of a high order (i.e. having many parameters, with several inputs and outputs, and having a large number of measurements) the classical methods can suffer from several problems. They can experience a local minimum in the performance function and thereby a lack of convergence to a global minimum. The user will need to specify complicated parameterization of system orders and delays. They may also suffer potential problems with numerical instability and excessive computation time to execute the iterative numerical minimization methods needed. In addition, modern control methods require a state-space model of the system. For cases such as these the State-Space (SS) identification method is the appropriate model structure.

The following equations describe a state-space model.

$$x(t+1) = Ax(t) + Bu(t) + Ke(t)$$

$$y(t) = Cx(t) + Dx(t) + e(t)$$

Where:

- x(t) is the state vector;
- y(t) is the system output;
- u(t) the system input;
- e(t) is the stochastic error;
- A, B, C, D, and K are the system matrices.

The dimension of the state vector $x(t)$ is the only setting we need to provide for the state-space model.

In general, the state-space model provides a more complete representation of the system than polynomial models – especially for MIMO systems – because the state-space model is similar to a first principle model. The identification procedure does not involve nonlinear optimization so the estimation reaches a solution regardless of the initial guess. Moreover, the parameter settings for the state-space model are simpler than polynomial models. We need to select only the order, or the number of states, of the model. The order can come from prior knowledge of the system. You also can determine the order by analyzing the singular values of the information matrix. For MIMO systems, when the model order is high, we use an ARX model because the algorithm involved in the ARX model estimation is fast and efficient when the number of data points is very large. The state-space model estimation with a large number of data points is slow and requires a large amount of memory. If we must use a state-space model, for example in modern control methods, we have to reduce the sampling rate of the signal in case the sampling rate is unnecessarily high. The other polynomial models, including the ARMAX, output-error, Box-Jenkins, and general-linear models, involve iterative, nonlinear optimization in the identification procedure. They require excessive computation time, and the minimization can get stuck at a false local minimum, especially when the order is high and the signal-to-noise ratio is low. However, we can use these models when the stochastic dynamics are important because they provide more flexibility.

4.2.2 Grey-box Model

In some circumstances, not all the information about the underlying dynamics or physical parameters of the system is known. In this case we can select the grey box model to specify these partially known physical parameters or constraints. Then we can estimate the remaining unknown parameters with partially known model estimation methods.

In addition to this we can use partially known state-space models to describe physical systems in a continuous-time or discrete-time form with symbolic variables rather than numerical values. The physical system can be a SISO, MISO, or MIMO system. We can set a constant value or constraints (initial guesses and upper and lower limits) on each symbolic variable with prior knowledge.

For a simpler SISO physical system, we can also use the partially known transfer function model to describe the physical system in continuous-time form. We can apply the prior knowledge we have about the system's most important physical parameters using the static gain, delay, time constant, natural frequency, and damping ratio inputs.

Grey-box model identification involves an optimization process to minimize the difference between the estimated output and the measured real output. Finding the global optimum depends on the limit range we set and the initial values. To decrease the estimation time, we can set a narrow limit range and reasonable initial parameters.

4.2.3 User defined Model

Sometimes, we may find the parametric model and grey-box model cannot represent the physical system we want to estimate, especially when the physical system contains some non-linear factors. Some programs provide a template to define more complicated SISO, MISO, and MIMO system models. In the template, we can use the following methods to build our special system model:

1. Mathematics or signal processing Vis;
2. Formula node;
3. Mathscript node;
4. Simulation nodes.

4.2.4 Conclusion

As has been discussed there are a variety of model structures available to assist in modeling a system. The choice of model structure is based upon an understanding of the system identification method and insight and understanding into the system undergoing identification. The characteristics of both system and disturbance dynamics play a role in the proper model selection.

These system identification methods can handle a wide range of system dynamics without requiring knowledge of the actual system physics, thereby reducing the engineering effort required to develop models. System Identification using MATLAB, MATRIXx or LabVIEW System Identification tools in conjunction with National Instruments hardware provides the control design engineer with a full suite of tools for developing, prototyping and deploying control algorithms.

4.3 How to Interpret the Noise Source

In many cases of system identification, the effects of the noise on the output are insignificant compared to those of the input. With good signal-to-noise ratios (SNR), it is less important to have an accurate noise model. Nevertheless it is necessary to understand the role of the noise and the noise source *e(t)*, whether it appears in the ARX model or in the general descriptions given above.

There are three aspects of the noise that should be stressed:

- understanding white noise;
- interpreting the noise source;
- using the noise source when working with the model.

These aspects are to be discussed here one by one.

How can we understand white noise? From a formal point of view, the noise source *e(t)* will normally be regarded as *white noise*. This means that it is entirely unpredictable. In other words, it is impossible to guess the value of *e(t)* no matter how accurately we have measured past data up to time *t*-1.

How can we interpret the noise source? The actual noise contribution to the output, *H e(t)*, has real significance. It contains all the influences on the measured *y*, known and unknown, that are not contained in the input *u*. It explains and expresses the fact that even if an experiment is repeated with the same input, the output signal will typically be somewhat different. However, the noise source *e(t)* need not have a physical significance. In the airplane example mentioned earlier, the noise effects are wind gusts and turbulence. Describing these as arising from a white noise source via a transfer function *H*, is just a convenient way of conveying their character.

How can we deal with the noise source when using the model? If the model is used just for simulation when the responses to various inputs are to be studied, then the noise model plays no immediate role. Since the noise source *e(t)* for new data will be unknown, it is taken as zero in the simulations, so as to study the effect of the input alone (a noise-free simulation). Making another simulation with *e(t)* being arbitrary white noise will reveal how reliable the result of the simulation is, but will not give a more accurate simulation result for the actual system's response.

The need and use of the noise model can be summarized as follows:

- It is, in most cases, required to obtain a better estimate for the dynamics, *G*.
- It indicates how reliable noise-free simulations are.

4.4 The Tool: Interactive Software[10]

The work to produce a model by identification is characterized by the following sequence:

1. Specification of a model structure;
2. Computer selection of the best model in this structure;
3. Evaluation of the properties of this model;
4. Testing of new structure (go to step 1).

The first stage that requires help is the computation of the model and the evaluation of its properties. There are now many commercially available program packages for identification, which provide such help. They typically contain the following routines:

A. *Handling of data, plotting and the like*: filtering of data, removal of drift, choice of data segment, and so on;

B. *Nonparametric identification methods*: estimation of covariance, Fourier transforms, correlation and spectral analysis, and so on;

C. *Parametric estimation methods*: Calculation of parametric estimates in different model structures;

D. *Presentation of models*: Simulation of models, estimation and plotting of poles and zeros, computation of frequency functions and plotting in Bode diagrams, and so on;

E. *Model validation*: computation and analysis of residuals ($\varepsilon(t,\theta_N)$), comparison between different models' properties, and the like.

---

[10] L. LJUNG, *System Identification Toolbox, For Use with MATLAB®, User's guide*, The MathWorks, Inc., 1997, Passim.

The existing program packages differ mainly by various user interfaces and by different options regarding the choice of model structure according to item C.

One of the most used packages is *MathWork's System Identification Toolbox* (Sitb), Ljung (1995), which is used together with Matlab. Sitb gives us the possibility to use all model structures of the black-box type described before, with an arbitrary number of inputs. ARX-models and state-space models with an arbitrary number of inputs and outputs are also covered. Moreover, the user can define arbitrary tailor-made linear state-space models in discrete and continuous time. A Graphic User Interface helps the user both to keep track of identified models and to guide him or her to available techniques.

Figure 36: Deriving models from data.

From the picture we can see that we can use a System Identification Toolbox (top) to analyze and preprocess data (left), estimate linear and nonlinear models (bottom), and validate estimated models (right).

The System Identification Toolbox provides a graphical user interface (GUI). The GUI covers most of the toolbox's functions and gives easy access to all variables that are created during a session. It can be started by typing "ident" in the MATLAB command window.

System Identification is about data and models and creating models from data. The main information and communication window "*ident*" is therefore dominated by two tables:

- A table of available data sets, each represented by an icon: The Model Board;
- A table of created models, each represented by an icon: The Data Board.

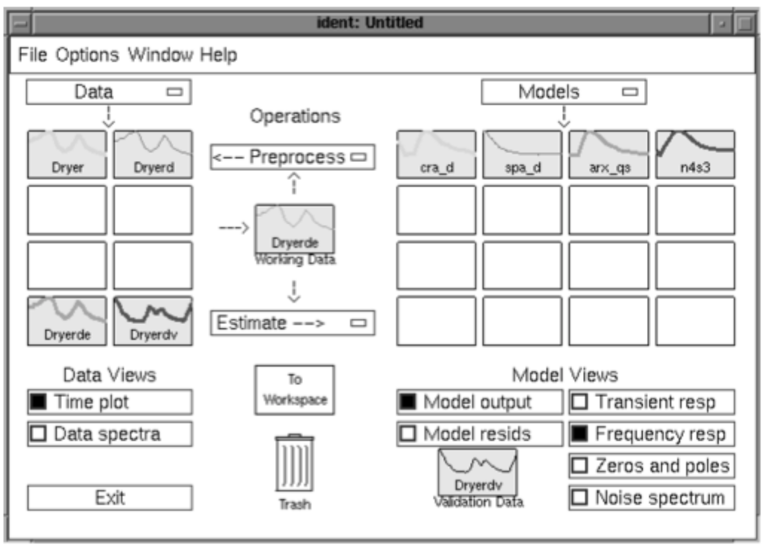


Figure 37: The System Identification Toolbox main window

It is possible to enter data sets into the Data Board by:

- Opening earlier saved sessions.
- Importing them from the MATLAB workspace.
- Creating them by de-trending, filtering, selecting subsets, etc., of another data set in the Data Board.

Imports are handled under the pop-up menu "*Data*" while creation of new data sets is handled under the pop-up menu "*Preprocess*".

The models are entered into the summary board by:

- Opening earlier saved sessions.
- Importing them from the MATLAB workspace.
- Estimating them from data.

Imports are handled under the pop-up menu "*Models*", while all the different estimation schemes are reached under the pop-up menu "*Estimate*".

4.4.1 The Working Data

All data sets and models are created from the Working Data set. This is the data that is given in the center of the "*ident*" window.

4.4.2 The Validation Data

The two model views *"Model Output"* and "*Model Residuals*" illustrate model properties when applied to the Validation Data set. This is the set marked in the box below these two views. It is good and common practice in identification to evaluate an estimated model's properties using a "fresh" data set, that is one that was not used for the estimation. It is thus advisable to allow the Validation Data to be different from the Working Data, although they should of course be compatible with these.

4.4.3 The Work Flow

The first step is to import data. The next step is usually to remove the means and/or trends from the data and to select subsets of data for estimation and validation purposes. We can continue estimating models, using the pop-up menu "*Estimate*". Then it is useful to examine the obtained models with respect to the favorite aspects using the different "*Model*

Views". The basic idea is that any checked view shows the properties of all selected models at any time. Inspired by the information we gain from the plots, we continue to try out different model structures (model orders) until we find a model we are satisfied with.

4.4.4 Workspace Variables

The models and data sets created within the GUI are normally not available in the MATLAB workspace. Indeed, the workspace is not at all littered with variables during the sessions with the GUI. The variables can however be exported at any time to the workspace.

4.4.5 Getting Data into the GUI

The information about a data set that should be supplied to the GUI are:

1. The input and output signals
2. The data set name
3. The starting time
4. The sampling interval
5. Data notes

By selecting the pop-up menu "*Data*" and choosing the item "*Import...*", a dialog box will open where we can enter information items 1-5 just listed.

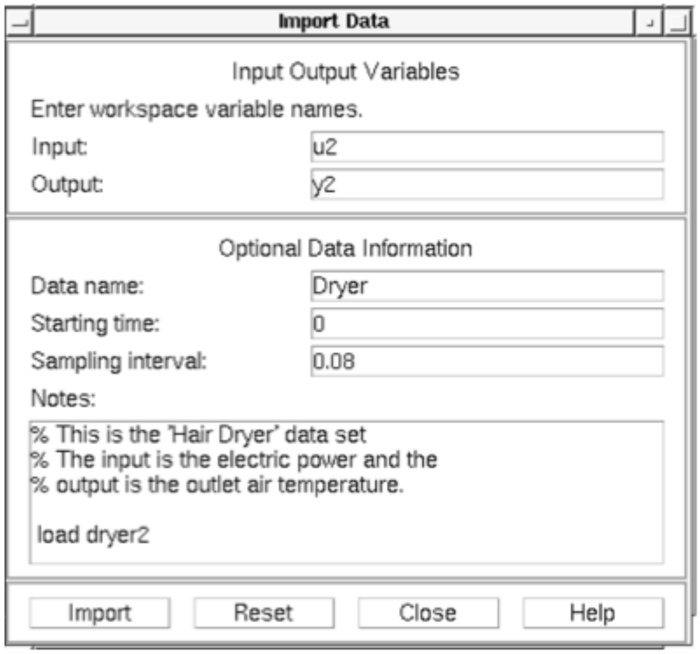


Figure 38: The dialog for importing data into the GUI

4.4.6 Taking a Look at the Data

The first thing to do after having inserted the data set into the Data Board is to examine it. By checking the "*Data View*" item "*Plot Data*", a plot of the input and output signals will be shown for the data sets that are selected. For multivariable data, the different combinations of input and output signals are chosen under menu item "*Channel*" in the plot window.

The purpose of examining the data in these ways is to find out if there are portions of the data that are not suitable for identification, if the information contents of the data is suitable in the interesting frequency regions, and if the data have to be preprocessed in some way before using them for estimation.

4.4.7 Preprocessing Data: De-trending

De-trending the data involves removing the mean values or linear trends from the signals (the means and the linear trends are then computed and removed from each signal individually). Advanced de-trending, such as removing piecewise linear trends or seasonal variations cannot be accessed within the GUI. It is generally recommended to remove at least the mean values of the data before the estimation phase, unless physical insight involving actual signal levels is built into the models.

4.4.8 Selecting Data Ranges

It is often the case that not all of the data record is not suitable for identification, due to various undesired features (missing or “bad” data, outbursts of disturbances, level changes etc.), so that only portions of the data can be used. In any case, it is advisable to select one portion of the measured data for estimation purposes and another portion for validation purposes. For multivariable data it is often advantageous to start by working with just some of the input and output signals.

4.4.9 Pre-filtering

By filtering the input and output signals through a linear filter (the same filter for all signals) we can focus the model’s fit to the system to specific frequency ranges. Pre-filtering is a good way of removing high frequency noise in the data, and also a good alternative to de-trending (by cutting out low frequencies from the pass band). Depending on the intended use for the model, we can also make sure that the model concentrates on the important frequency ranges..

4.4.10 Quick Start

The pop-up menu item "*Preprocess* > *Quickstart*" performs the following sequence of actions:

1. It opens the Time plot Data view
2. It removes the means from the signals
3. It splits these de-trended data into two halves: the first one is made Working Data and the second one becomes Validation Data.

All the three created data sets are inserted into the Data Board.

4.4.11 Checklist for Data Handling

- Insert data into the GUI's Data Board.
- Plot the data and examine it carefully.
- Typically de-trend the data by removing mean values.
- Possibly pre-filter the data to enhance and suppress various frequency bands.
- Select portions of the data for Estimation and for Validation. Drag and drop these data sets into the corresponding boxes in the GUI.

4.4.12 Simulating Data

The GUI is intended primarily for working with real data sets, and does not itself provide functions for simulating synthetic data. That has to be done in command mode, and you can use your favorite procedure in SIMULINK, the Signal Processing Toolbox, or any other toolbox for simulation and then insert the simulated data into the GUI as described above. The System Identification Toolbox also has several commands for simulation.

4.4.13 Estimating Models The Basics

Estimating models from data is the central activity in the System Identification Toolbox. It is also the one that offers the most variety of possibilities and thus is the most demanding one for the user.

All estimation routines are accessed from the pop-up menu "*Estimate*" in the "*ident*" window. The models are always estimated using the data set that is currently in the "*Working Data*" box. One can distinguish between two different types of estimation methods:

- *Direct estimation of the Impulse or the Frequency Response of the system*. These methods are often also called nonparametric estimation methods, and do not impose any structure assumptions about the system, other than that it is linear.
- *Parametric methods*. A specific model structure is assumed, and the parameters in this structure are estimated using data. This opens up a large variety of possibilities, corresponding to different ways of describing the system. Dominating ways are state-space and several variants of difference equation descriptions. We will only consider this approach.

4.4.14 Estimation of Parametric Models

The SITB supports a wide range of model structures for linear systems. They are all accessed by the menu item "*Estimate > Parametric Models...*" in the "*ident*" window. This opens up a dialog box "*Parametric Models*", which contains the basic dialog for all parametric estimation as shown in the following picture:

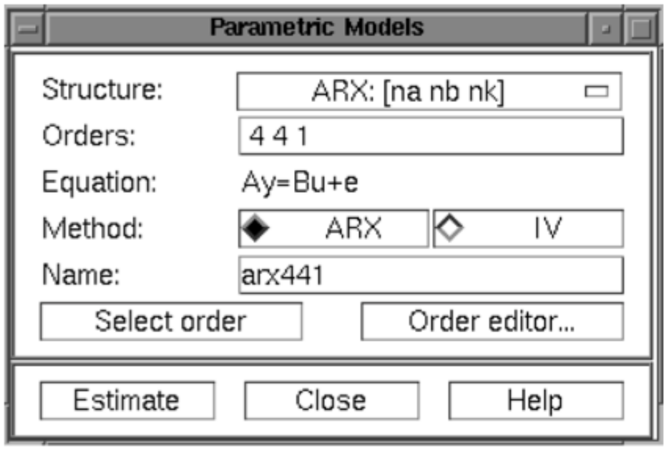


Figure 39: The dialog box for estimating parametric models

The basic function of this box is as follows:

As we select "*Estimate*", a model is estimated from the Working Data. The structure of this model is defined by the pop-up menu "*Structure*" together with the edit box "*Orders*". It is given a name, which is written in the edit box "*Name*". The GUI will always suggest a default model name in the "*Name*" box, but we can change it to any string before selecting the "*Estimate*" button. The interpretation of the model structure information (typically integers) in the "*Orders*" box, depends on the selected "*Structure*" in the pop-up menu. This covers, typically, six choices:

- ARX models
- ARMAX model
- Output-Error (OE) models
- Box-Jenkins (BJ) models
- State-space models
- Model structure defined by Initial Model (User defined structures)

It is possible to fill out the "*Orders*" box yourself at any time, but for assistance it is possible to select "*Order Editor…*". This will open up another dialog box, depending on the chosen "*Structure*", in which the desired model order and structure information can be entered in a simpler fashion.

NOTE**:** For the state-space structure and the ARX structure, several orders and combinations of orders can be entered. Then all corresponding models will be compared and displayed in a special dialog window. This could be a useful tool to select good model orders. When this option is available, a button "*Order selection*" is visible.

4.4.15 Estimation Method

A common and general method of estimating the parameters is the *prediction error approach*, where simply the parameters of the model are chosen so that the difference between the model's (predicted) output and the measured output is minimized. This method is available for all model structures. Except in the case of the ARX model, the estimation involves an iterative, numerical search for the best fit. To obtain information from and interact with this search, select "*Iteration control...*". This also gives access to a number of options that govern the search process. For some model structures (the ARX model, and black-box state-space models) methods based on correlation are also available: Instrumental Variable (IV) and Sub-space (N4SID) methods.

4.4.16 Resulting Models

The estimated model is inserted into the Model Board. It is possible to examine its various properties and to compare them with other models' properties using "*Model View*".

4.4.17 How to Know Which Structure and Method to Use

There is no simple way to find out "the best model structure"; in fact, for real data, there is no such thing as the "best" structure. It is best to be generous at this point. It often takes just a few seconds to estimate a model, and using the different validation tools described in the next section, we can quickly find out if the new model is any better than the ones you had before. There is often a significant amount of work behind the data collection, and spending a few extra minutes trying out several different structures is usually worthwhile.

4.4.17.1 ARX Models The Structure

In the system identification toolbox the ARX model correlates the current output *y(t)* to a finite number of past outputs *y(t-k)* and inputs *u(t-k)* through the following equation:

$$y(t)+a_1y(t-1)+...+a_{na}y(t-n_a) = b_1u(t-n_k)+...+b_{nb}u(t-n_k-n_b+1)$$

The structure is thus entirely defined by the three integers:

- n_a: number of poles
- n_b: number of zeros
- n_k: pure time-delay (the dead-time) in the system. For a system under sampled-data control, typically n_k is equal to 1 if there is no dead-time.

For multi-input systems n_b and n_k are row vectors, where the i^{th} element gives the order/delay associated with the i^{th} input.

4.4.17.2 Estimating Many Models Simultaneously

By entering any or all of the structure parameters as vectors, using MATLAB's colon notation, like n_a=1:10, etc., we can define many different structures that correspond to all combinations of orders. When selecting "*Estimate*", models corresponding to all of these structures are computed. A special plot window will then open that shows the fit of these models to Validation Data.

Multi-input models: For multi-input models you can of course enter each of the input orders and delays as a vector. The number of models resulting from all combinations of orders and delays can however be very large. As an alternative, it is possible to enter one vector (like nb=1:10) for all inputs and one vector for all delays. Then only such models are computed that have the same orders and delays from all inputs.

4.4.17.3 Estimation Methods

There are two methods to estimate the coefficients a and b in the ARX model structure:

Least Squares: Minimizes the sum of squares of the right-hand side minus the left-hand side of the expression above, with respect to a and b. This is obtained by selecting ARX as the "*Method*".

Instrumental Variables: Determines 'a' and 'b' so that the error between the right- and left-hand sides becomes uncorrelated with certain linear combinations of the inputs. This is obtained by selecting IV in the "*Method*" box.

4.4.17.4 ARMAX, Output-Error and Box-Jenkins Models

There are several elaborations of the basic ARX model, where different noise models are introduced. These include well known model types, such as ARMAX, Output-Error, and Box-Jenkins.

4.4.17.5 The General Structure

As we have already seen before, the general input-output linear model for a single-output system with input u and output y can be written as:

$$A(q)y(t) = \sum_{i=1}^{n_u} \left[\frac{B_i(q)}{F_i(q)}\right] u_i(t - nk_1) + \left[\frac{C(q)}{D(q)}\right] e(t)$$

Here u_i denotes input #i, and A, B_i, C, D, and F_i, are polynomials in the shift operator (z or q).

The general structure is defined by giving the time-delays n_k and the orders of these polynomials *(i.e.*, the number of poles and zeros of the dynamic model from u to y, as well as of the noise model from e to y).

The general form is just a way to express with a unique formula all the models described before:

- ARX: $A(q)\ y(t) = B(q)\ u(t\text{-}n_k) + e(t)$
- ARMAX: $A(q)\ y(t) = B(q)\ u(t\text{-}n_k) + C(q)\ e(t)$
- OE: $y(t) = [B(q)/F(q)]\ u(t\text{-}n_k) + e(t)$
- BJ: $y(t) = [B(q)/F(q)]\ u(t\text{-}n_k) + [C(q)/D(q)]\ e(t)$

These equations are called: "shift operator polynomials" because of the use of the "shift operator" *b*. These are just compact ways of writing difference equations. For example the ARMAX model in longhand would be:

$$y(t)+a_1(t\text{-}1)+\ldots+a_{na}y(t\text{-}n_a)=b_1u(t\text{-}n_k)+\ldots b_{nb}u(t\text{-}n_k\text{-}n_b\text{+}1)+e(t)+c_1e(t\text{-}1)\ +\ldots+c_{nc}e(t\text{-}n_c)$$

Note that:

- *A(q)* corresponds to poles that are common between the dynamic model and the noise model (useful if noise enters the system "close to" the input);
- *F(q)* determines the poles that are unique for the dynamics from input # I;
- *D(q)* determines the poles that are unique for the noise.

The motivation for introducing all these model variants is to provide for flexibility in the noise description and to allow for common or different poles (dynamics) for the different inputs.

4.4.18 Entering the Model Structure

Use the "*Structure*" pop-up menu in the "*Parametric Models*" dialog to choose between the ARX, ARMAX, Output-Error, and Box-Jenkins structures.

The orders of the polynomials are selected by the pop-up menus in the "*Order Editor*" dialog window, or by directly entering them in the edit box "*Orders*" in the "*Parametric Models*" window. When the order editor is open, the default orders, entered as you change the model structure, are based on previously used orders.

4.4.19 Estimation Method

The coefficients of the polynomials are estimated using a prediction error/ Maximum Likelihood method, by minimizing the size of the error term "e" in the expression above. Several options govern the minimization procedure. These are accessed by activating "*Iteration Control*" in the "*Parametric Models*" window, and selecting "*Option*".

CHAPTER 5

MODEL ESTIMATION FOR THE OUTLET TEMPERATURE

Now that we know the basics of the System Identification theory and how the System Identification Toolbox works on MatLab, we can start computing the correlations we need to create our regulation program.

In order to obtain a regulation strategy for the outlet water temperature, we adopted an inverse approach, first correlating the outlet temperature and the outdoor and the indoor temperatures and the ON-OFF profile chosen. We did this separately for each control strategy we deemed able to ensure a comfort temperature inside the building (so we excluded the *13h ON* control strategy from this study).

In the next paragraphs the computation of the *24h ON* correlation is described in detail as an example for the approach we used. For the other regulation strategies we will just discuss the results obtained for them.

5.1 *24h ON* control strategy

After having divided the data with respect to the control strategy we had to rid them of Outliers. There were few apparent anomalies where the gas consumption was equal to zero even if at that moment the boiler was meant to be on according to the ON-OFF profile given to it. However these were not wrong signals the control system was recording; in fact they were accompanied by a drastic decrease in the outlet water temperature and of the return water temperature, probably caused by a bad filter inside the boiler that was recently substituted with a more sophisticated one. This new filter is able to decrease the outlet temperature by 1°C by decreasing the water flow heated by the boiler instead of turning off the entire system. Another kind of outlier we had to confront was the transient data caused by the change of the regulation strategy during the data collection. The last type of outlier we found was the 'missing data'. These were caused by an error of the emitter that sometimes wasn't sending all the data to the collecting system that it was supposed to. In order to make up for all these outliers we used statistical methods such as regressions made out of the data right before and right after the outliers, evaluating the theoretical value they would have had if no external causes hadn't made them appear, and substituting the outliers those values just obtained.

Once we had obtained a completely clean data set, we saved it as a .csv (comma separated value) file and imported it to matlab using the 'csvimport' function (reported in Appendix A) as follows:

```
>> clear all
close all
%load the data file I
path = 'I.csv';
[Text,ONOFF,Cons,Tmand,Tint_all,Tint_uff,Trit] =
csvimport(path,'delimiter',';','column',
[3,4,5,6,7,8,9],'outputAsChar', false,'noHeader',true);
>>
```

Once the data were in the matlab workspace we opened the System Identification Toolbox using the command 'ident':

```
>> ident
Opening System Identification Tool ....... done.
>>
```

The GUI window that appeared is the following:

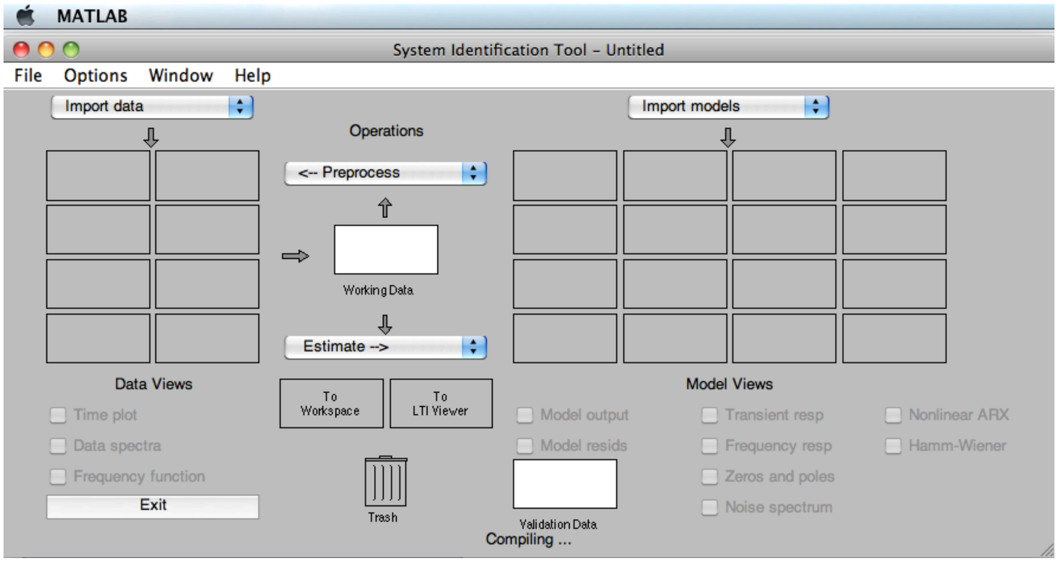


Figure 40: The System Identification Toolbox main window

Using the 'Import Data' command we were able to import the inputs and the outputs of our models and to define the starting time and the sample interval (we chose both of them as equal to 1 for simplicity):

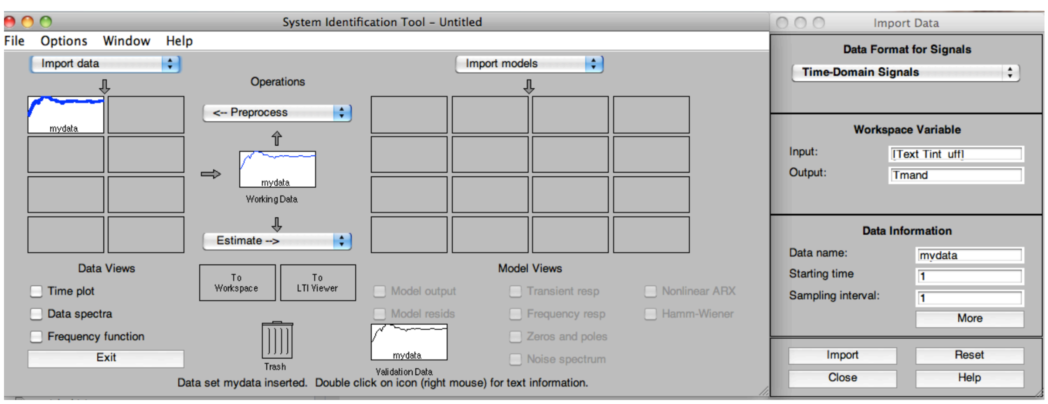


Figure 41: The System Identification Toolbox import data window

Eliminating the mean from each input and output, and choosing the ranges for evaluating the model and for validating it, we obtained the following data set (where u1 and u2 are the inputs: the outdoor and the indoor temperatures, and y1 represents the output):

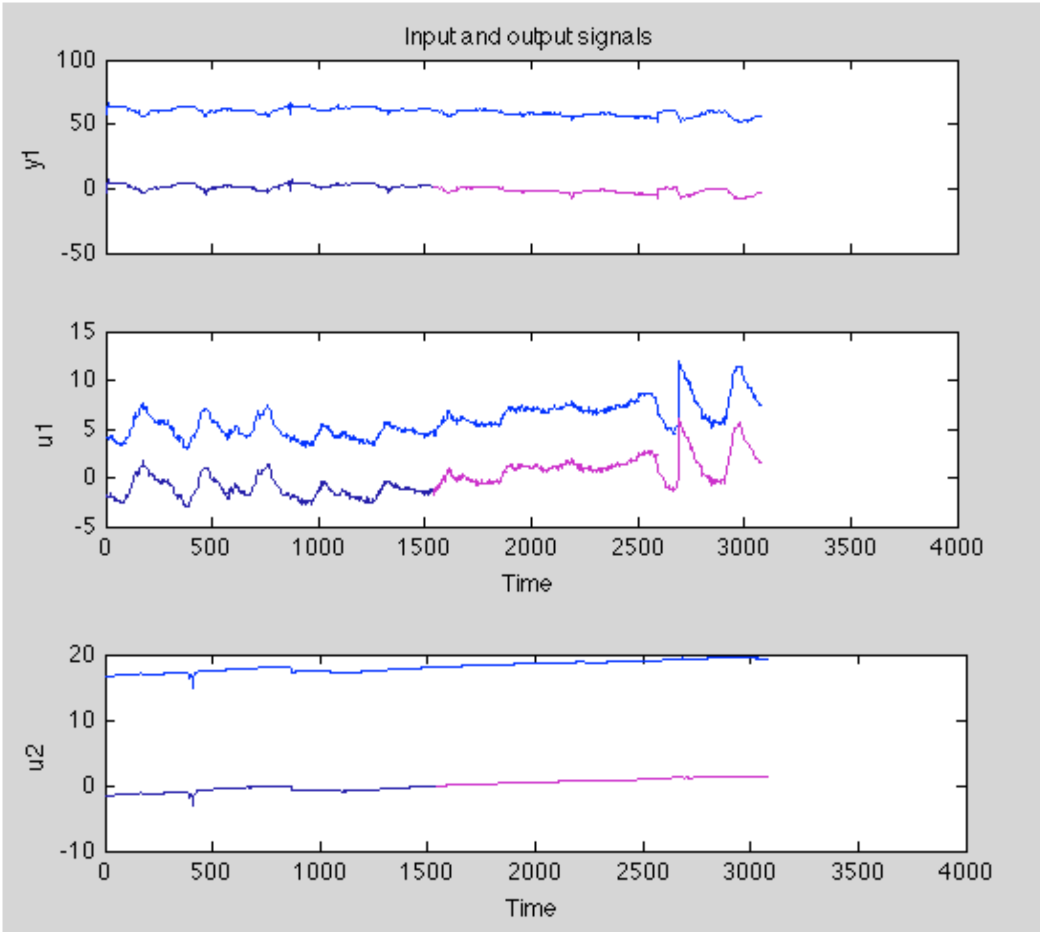


Figure 42: *24h ON* Inputs (u1, u2) and output (y1) signals vs time

As it can be seen in the graph the blue line is the real input and output values as we collected them, while the pink and dark blue lines represent the values without their means. The dark blue color represents the data used to evaluate the models while the pink one indicates the validation data.

Once all the data were ready to be used for our computations we started evaluating the models using the 'Linear Parametric Models' command under the 'Estimate' menu:

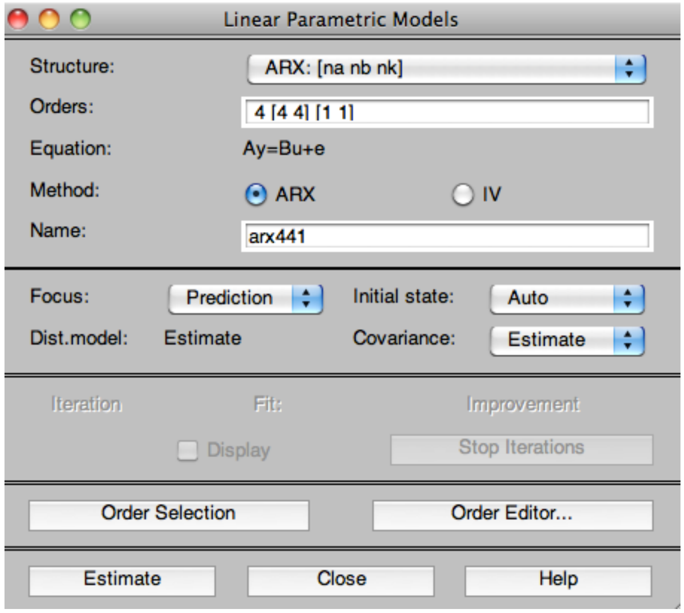


Figure 43: Model type and order selection window

We started evaluating the correlation first by considering the ARX model structure. Using the ‘Order Selection’ command, which let us compute a huge number of simulations in a small time, we obtained:

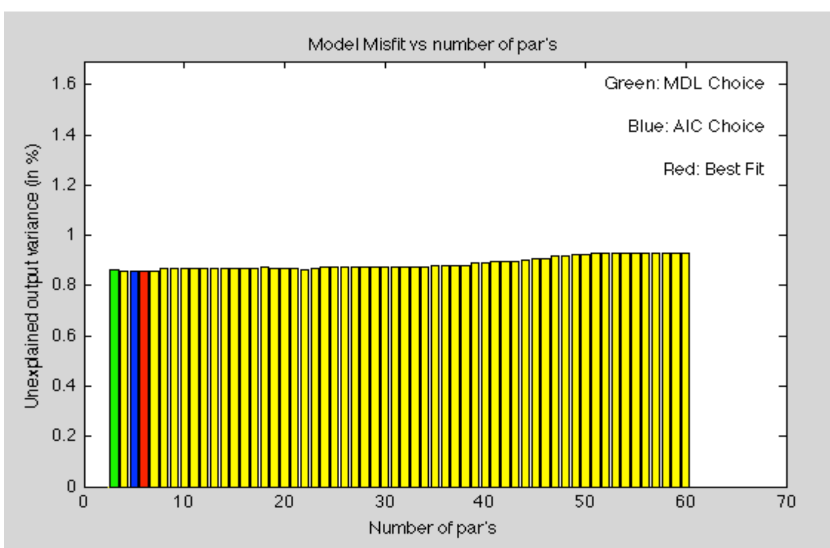


Figure 44: ARX models misfit vs number of parameters

The Order Selection command exists for the ARX structure only, and it is very important since it gives us an idea of the order of some of the coefficients: B(q) and A(q). As it can be seen, the picture already tells us which are the best models among the ones evaluated by Matlab for the ARX structure. The following graph represents a comparison between the outlet temperature estimated through the ARX models just evaluated and the real one (the data are represented without their means), plotted against time:

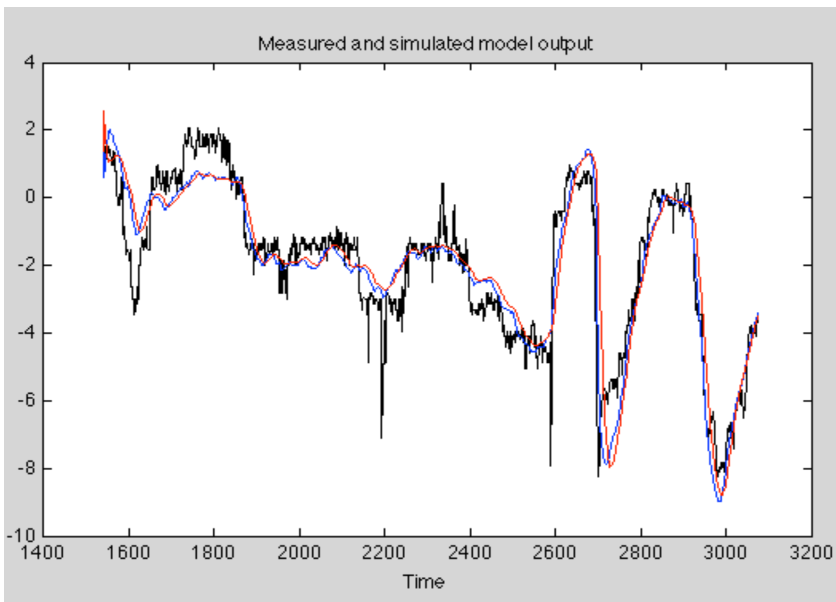


Figure 45: Measured and simulated model output

Having an idea of the order of some of the coefficients that best fit the model we were searching, we continued our analysis by evaluating the ARMAX, the OE and the BJ model structures. Unfortunately there is no fast way, as for the ARX, to compute a large number of models at the same time; they must be computed one by one. After having tried more than 200 other models we selected the best four for each type (ARX, ARMAX…) and we collected them in the following chart:

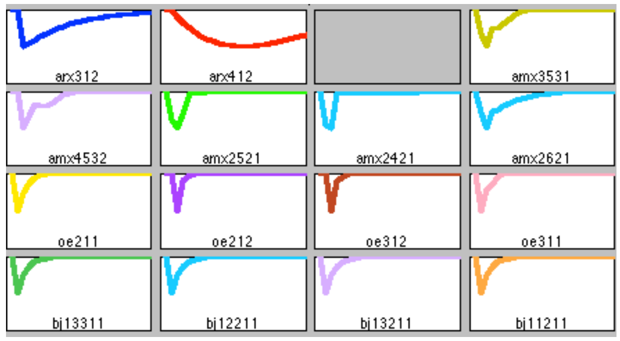


Figure 46: 24h ON best estimation models

The numbers after the model types represent the coefficient order for the corresponding model. Representing the outlet temperature obtained by the best models of the previous chart on a graph together with the real outlet temperature (each of them without its mean) we have:

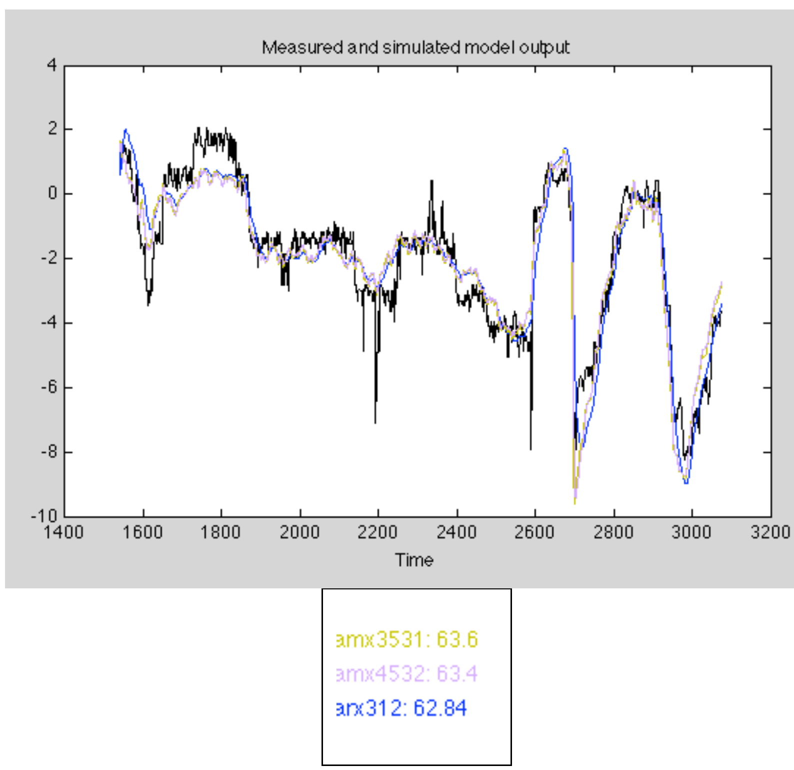


Figure 47: *24h ON* measured and simulated model outputs with their best fits

As it can be seen from the previous graph all the selected models represent the real data very well. All of them are able to evaluate more than 60% of the values correctly ("The %Fit indicates the agreement between the model response and the measured output: 100 means a perfect fit, and 0 indicates a poor fit"[11]). Considering just the 'Best Fit' numbers in the chart we would choose the ARMAX3531 model to represent the outlet temperature in a *24h ON* system (since it is the one that provides the highest Best Fit value); however, in order to reduce gas consumption, we want the line we are giving the boiler to control the outlet temperature to have as few fluctuations as possible. As it can be seen from the graph, the yellow line representing the ARMAX3531 model has many oscillations. Therefore we can say that the blue line, which represents the ARX312 model, represents the model that best evaluates the outlet temperature as we want it to be.

5.2 *18h ON* Control Strategy

Using the same procedure just outlined for the 24h ON regulation strategy we cleaned up the data set, importing it to the MatLab workspace first, and then to the System Identification Toolbox. Eliminating the means from the inputs (outdoor (u1) and indoor (u2) temperatures) and from the output (outlet temperatures (y1)) we obtained the following inputs and outputs graph:

---

[11] L. LJUNG, *System identification Toolbox*, The Mathworks: http://www.mathworks.co.uk/help/toolbox/ident/gs/brav7fy.html, 1997

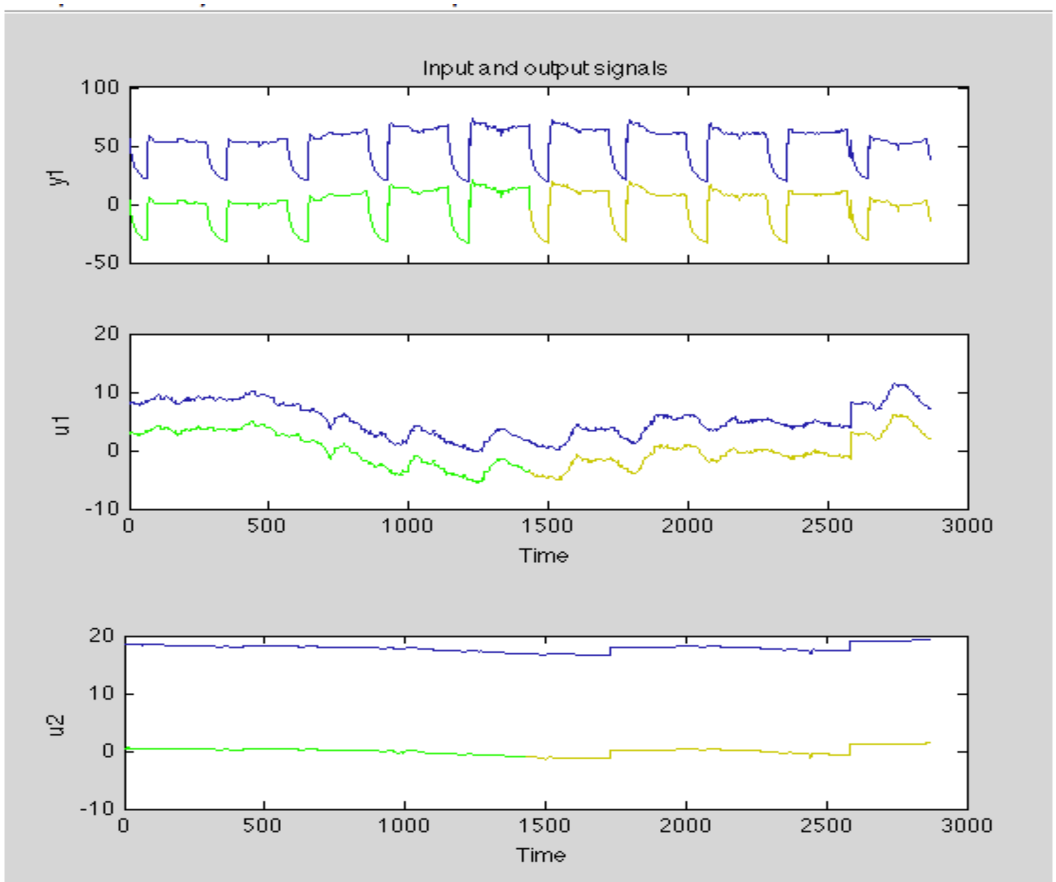


Figure 48: *18h ON* first attempt inputs (u1, u2) and output (y1) signals

In this case the evaluation data set is represented with a green line while the yellow line represents the validation data set.

Computing the ARX models as we described before for the 24h ON regulation strategy, we obtained the following graph for the 'Order Selection' command:

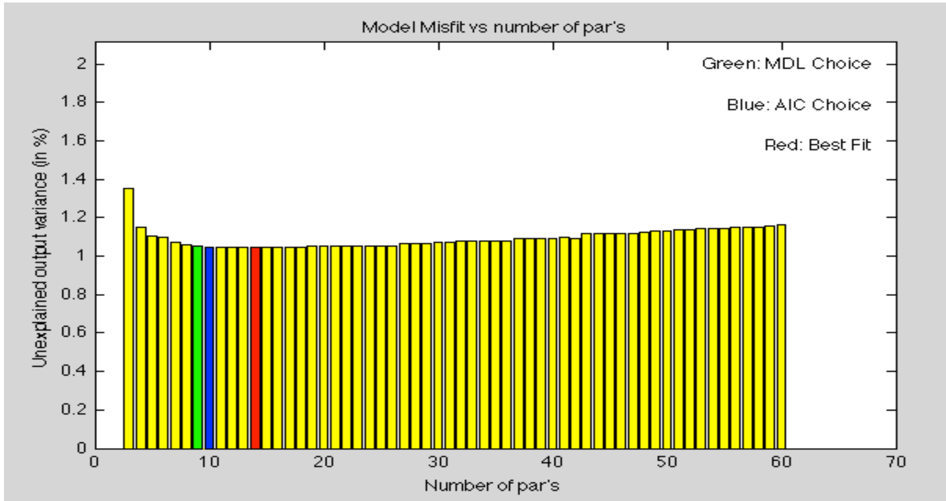


Figure 49: *18h ON* ARX models misfit vs number of parameters

What is interesting is that the number of points the model is not able to estimate has a maximum for low coefficient orders; it then decreases to a minimum and it finally goes up again. This behavior is to be expected since for low coefficient orders the model is not able to consider all the variability of the output with respect to the inputs, while for high model coefficients the model perfectly follows the estimation data, although the results are not reliable for estimation purposes.

Having an idea of the order of some of the model coefficients we continued our analysis considering the ARMAX, the OE and the BJ structures. Unfortunately there is no fast way, as for the ARX structure, to compute a large number of models all together; they must be computed one by one. After having tried more than 600 other models we selected the ones able to ensure the best fits for each structure type (ARX, ARMAX…) and we summarized them in the following chart:

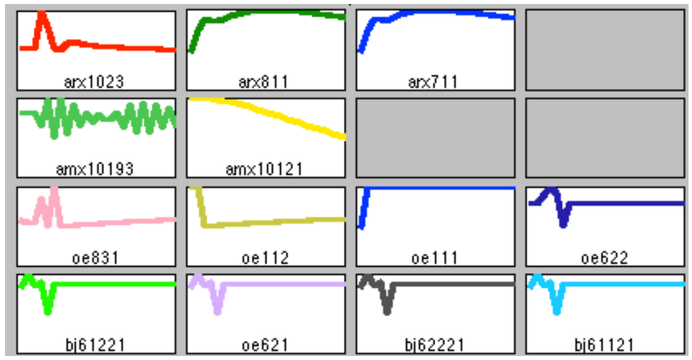


Figure 50: *18h ON* best estimation models

Representing the comparison between the outlet temperature, obtained from the best models among the ones in the chart above, and the real outlet temperature (validation data) with respect to time we have:

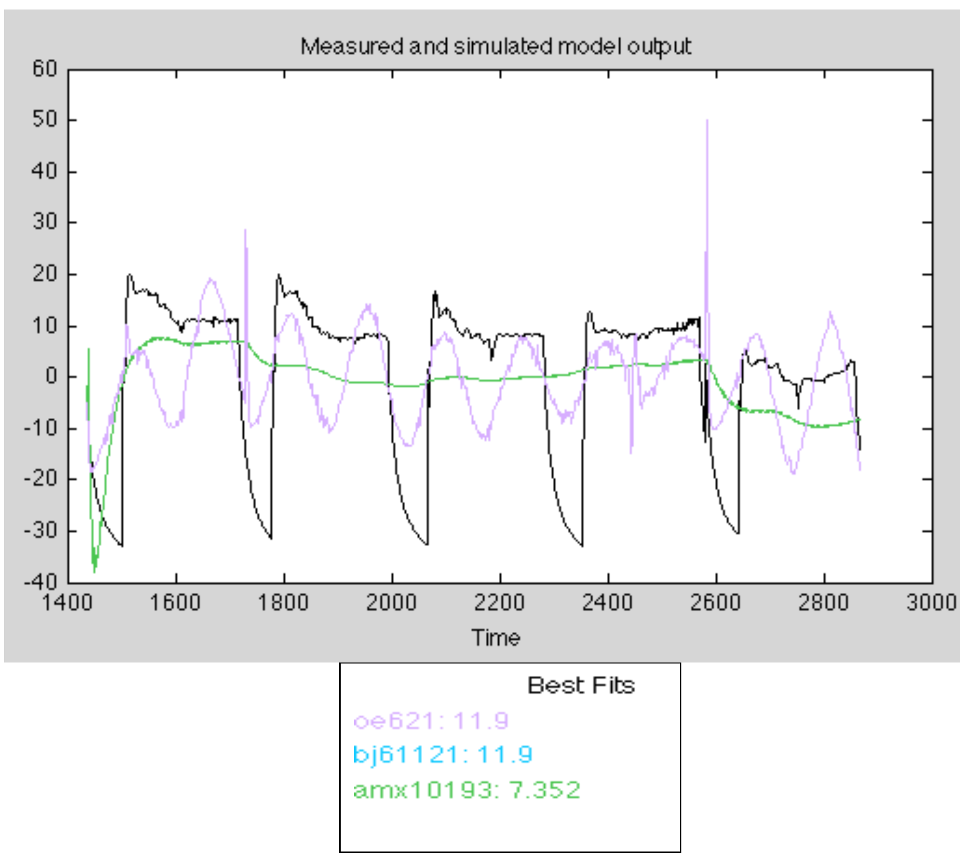


Figure 51: *18h ON* measured and simulated model outputs with their best fits

As it can be seen, none of the computed models is able to represent the outlet temperature properly (Best Fit = 11.9%). This means the inputs we chose were not sufficient or not suitable to describe the model.

In order to overcome this problem we decided to add one input: the ON-OFF profile we gave the boiler during the *18h ON* control days (u3). Doing this the model inputs and outputs resulted as follows:

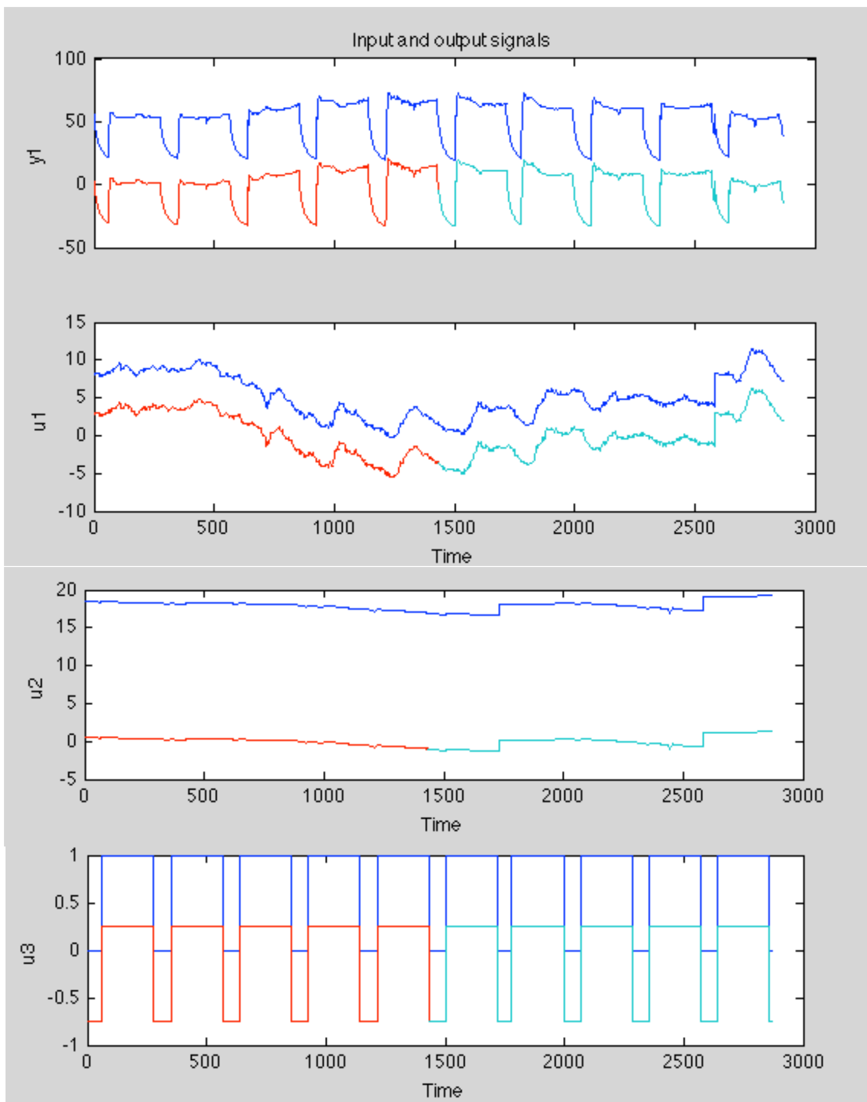


Figure 52: *18h ON* inputs (u1, u2 , u3(new)) and output (y1) signals

In this case a red line represents the evaluation data set while a cyan one represents the validation data set.

Computing the ARX models as described before for the 24h ON regulation strategy we obtained the following graph for the 'Order Selection' command:

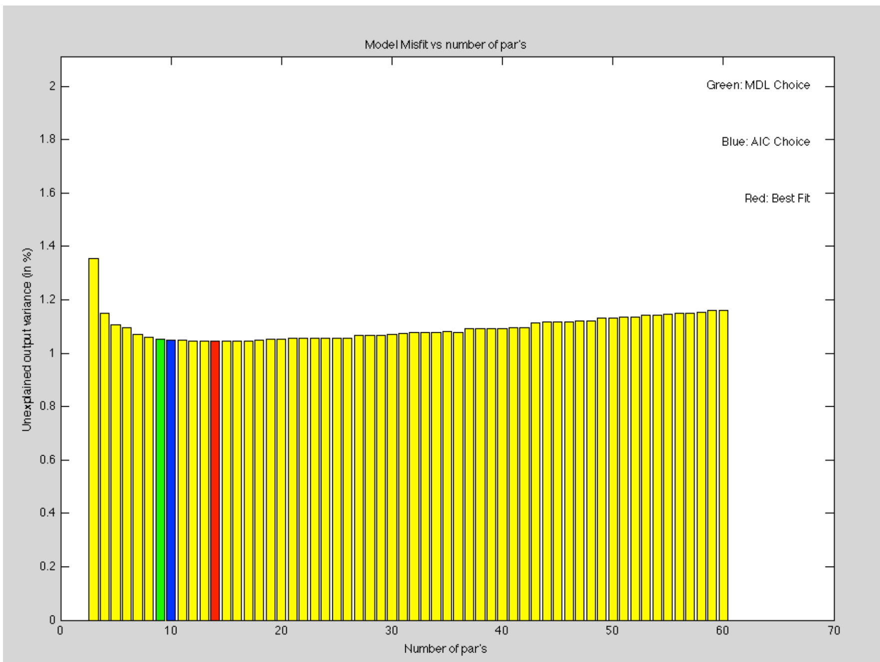


Figure 53: *18h ON* (with ON-OFF profile as additional input) ARX models misfit vs number of parameters

The graph is very similar to the previous one.

Having an idea of the order of some of the coefficients that best fit the model we were searching for, we continued our analysis considering the ARMAX, the OE and the BJ structures. Unfortunately there is no fast way, as for the ARX models, to compute a large number of models all together; they must be computed one by one. After having tried more than 400 other models we selected the best ones for each type (ARX, ARMAX…) and we collected them in the following chart:

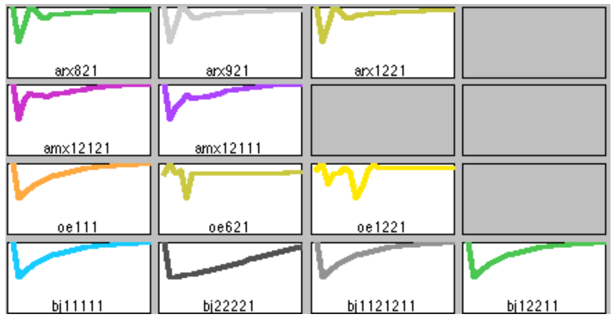


Figure 54: *18h ON* best estimation models with ON-OFF profile as additional input

Representing the outlet temperature obtained by some of these models on a graph comparing them with the real outlet temperature (validation data) with respect to time we have:

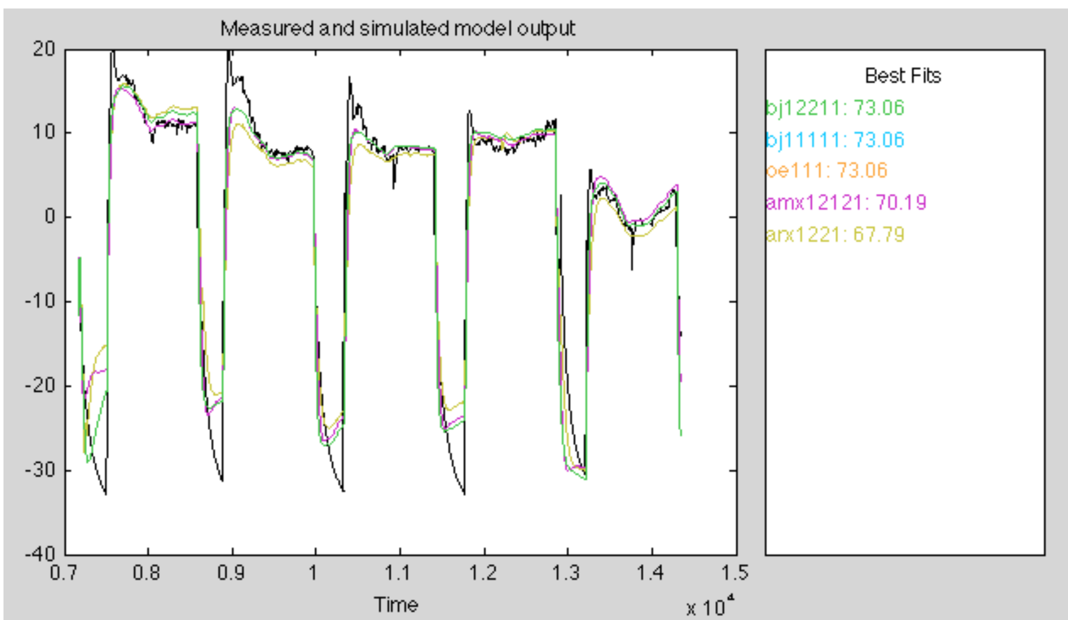


Figure 55: *18h ON* (with ON-OFF profile as additional input) measured and simulated model outputs with their best fits

As can be seen all the selected models represent the real data very well. In fact all of them are able to evaluate more than 60% of the values correctly. Therefore we can state that adding the ON-OFF profile was a good strategy to correlate the outdoor and the indoor temperatures with the outlet one. This was to be expected as not giving the ON-OFF profile as an input can confuse the program as it has to cope with different outlet temperatures for similar, if not equal, conditions.

Considering the 'Best Fit' numbers, we can see that the first two BJ models and the OE111 present the same probability of inaccurately estimating the outlet temperature. But, as stated before, while describing the various model structures, the OE has one coefficient less than the BJ and, in this particular case, its coefficient orders are also equal or lower than those of the BJ; having fewer coefficients of lower orders means having lower computational costs. Therefore we can say that the orange line, which represents the OE111 model, represents the model that best evaluates the outlet temperature with the inputs given.

N.B. This model is also able to evaluate the 13h ON outlet temperature. In fact it needs the same kind of inputs. Evaluating the 13h ON model with the OE111 just evaluated we obtain a best fit 73.06%:

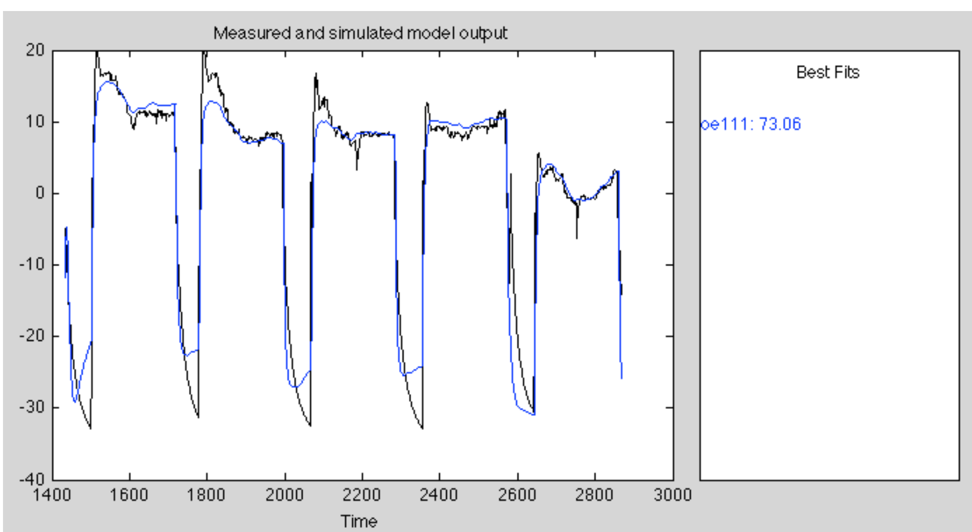


Figure 56: *13h ON* measured and simulated model outputs with their best fits

5.3 *24h Bi-Climatic* Control Strategy

Since the 24h ON Bi-Climatic doesn't have any ON-OFF profile (it is always on) we decided once again to adopt the approach we used for the 24h ON regulation strategy, giving as inputs only the indoor (u1) and the outdoor (u2) temperatures. Therefore representing the inputs and output (y1) on a graph we have:

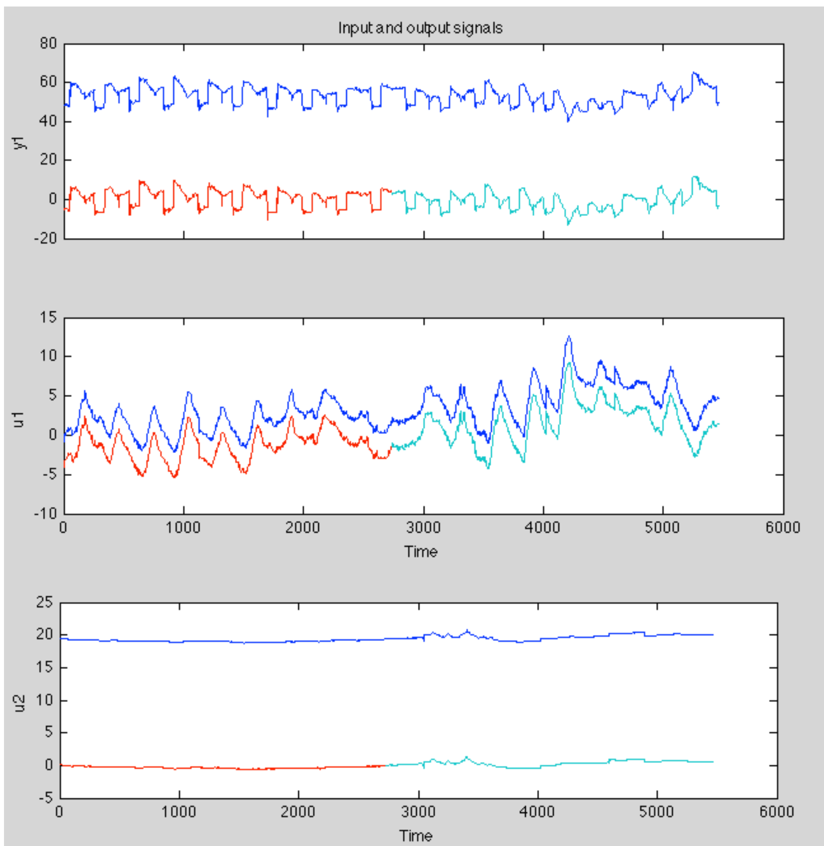


Figure 57: *24h Bi-Climatic* inputs (u1, u2) and output (y1) signals

In this case the evaluation data set is represented by a red line while the cyan line represents the validation data set.

Computing the ARX models as described before we obtained the following graph for the 'Order Selection' command:

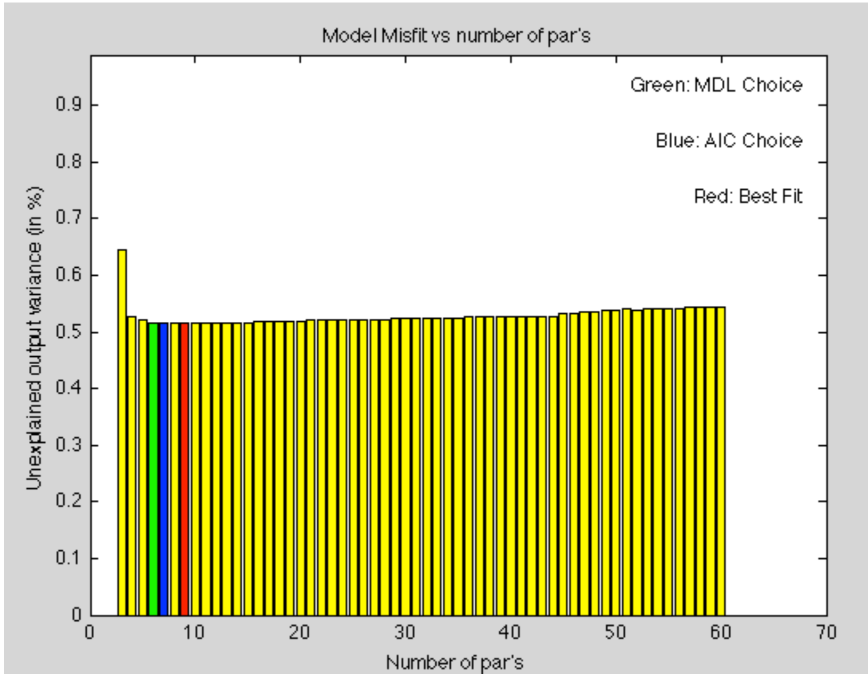


Figure 58: Figure 59: *24h ON Bi-Climatic* ARX models misfit vs number of parameters

As it can be seen the graph is very similar to the previous one. Having an idea of the order of some of the coefficients that best fit the model we were searching we continued our analysis, evaluating some ARMAX, OE and BJ models. Unfortunately there is no fast way, as for the ARX structure, to compute a large number of models in a row; they must be computed one by one. After having computed more than 900 other models we selected the best ones for each type (ARX, ARMAX…) and collected them in the following chart:

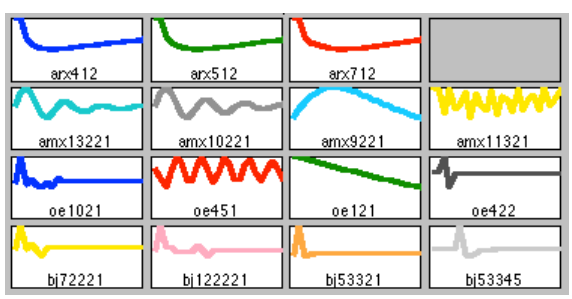


Figure 60: *24h ON Bi-Climatic* best estimation models

Representing the outlet temperature obtained by these models on a graph comparing it with the real outlet temperature (validation data) with respect to time we have:

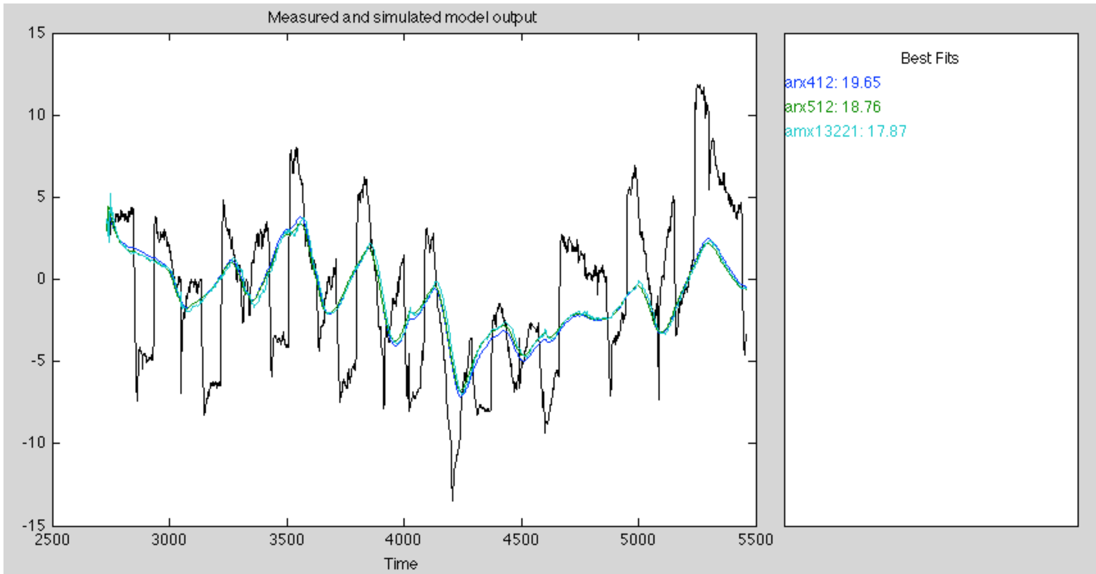


Figure 61: *24h ON* measured and simulated model outputs with their best fits

As it can be seen none of the computed models is able to represent the outlet temperature properly. In fact the Best Fit we were able to find is equal to 19.65%. Neither of the computed models is able to describe how the outlet temperature changes between the day and the night (with the climatic curve switch) and to model the moments when the boiler is turned off. This means the inputs we chose were not sufficient or not suitable for the model we wanted to obtain. To overcome this problem we decided to add one input: the C1C2 profile that tells us when the system changes the climatic curve it uses to adjust the outlet temperature (the new u2, while the indoor temperature becomes u3). In order to do this we first had to create this input, since it wasn't one of the data we had collected. Therefore we added one column to the .csv files related to the Bi-Climatic regulation.

Importing the inputs and the outputs to the System Identification Toolbox and plotting them against time we have:

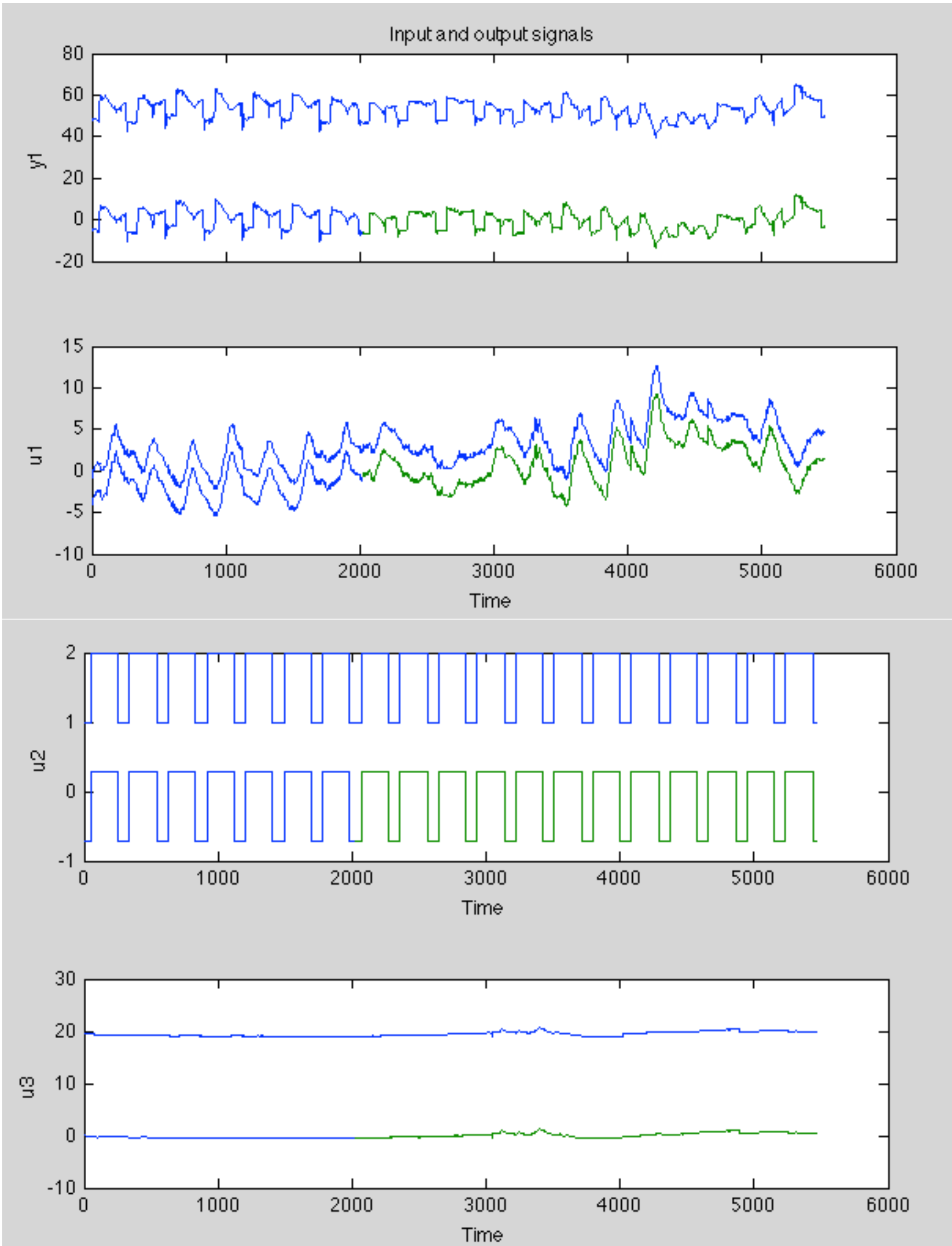


Figure 62: *24h ON Bi-Climatic* inputs (u1, u2 (new) , u3) and output (y1) signals

To import the new column in the Workspace we had to rewrite the script to import this with the rest of the file as follows:

```
>>%for Bi-Climatic curves (there is one more column in the .csv file)
clear all
close all
%load the data file
path = 'Biclimatica.csv';
[Text,ONOFF,Consumo,Tmand,Tint_all,Tint_uff,Trit,C1C2] = csvimport(path,'delimiter',
';','column',[3,4,5,6,7,8,9,10],'outputAsChar',false,'noHeader',true);
```

In this case the evaluation data set is represented by a green line while the validation data line is blue. As we can see, in this case we used as estimation data the ones that chronologically came after the data we used as validation set. The reason for this choice is that the second part of the data is much more heterogeneous and gives us the possibility of estimating a model which better knows how to behave in different situations.

Computing the ARX models as described before we obtained the following graph for the 'Order Selection' command:

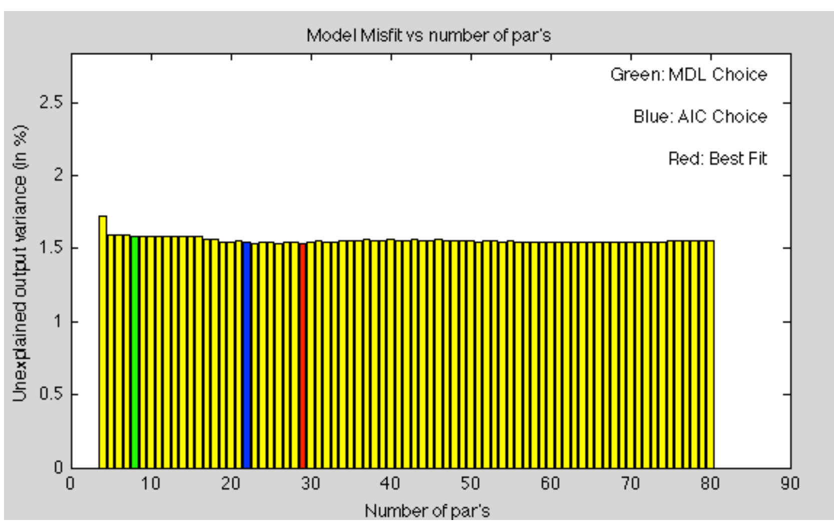


Figure 63: *24h ON Bi-Climatic* (with C1C2 profile as additional input) ARX models misfit vs number of parameters

Having an idea of the order of some of the coefficients that best fit the model we were searching we continued our analysis evaluating some ARMAX, OE and BJ models. Unfortunately there is no fast way, as for the ARX structure, to compute a large number of models all together. After having tried many other models we selected the best ones for each type and we collected them in the following chart:

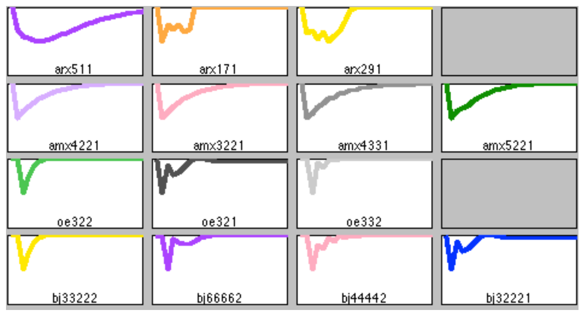


Figure 64: *24h ON Bi-Climatic* best estimation models with C1C2 profile as additional input

Representing the outlet temperature obtained by some of the models of the previous chart on a graph, compared with the real outlet temperature (validation data) we have:

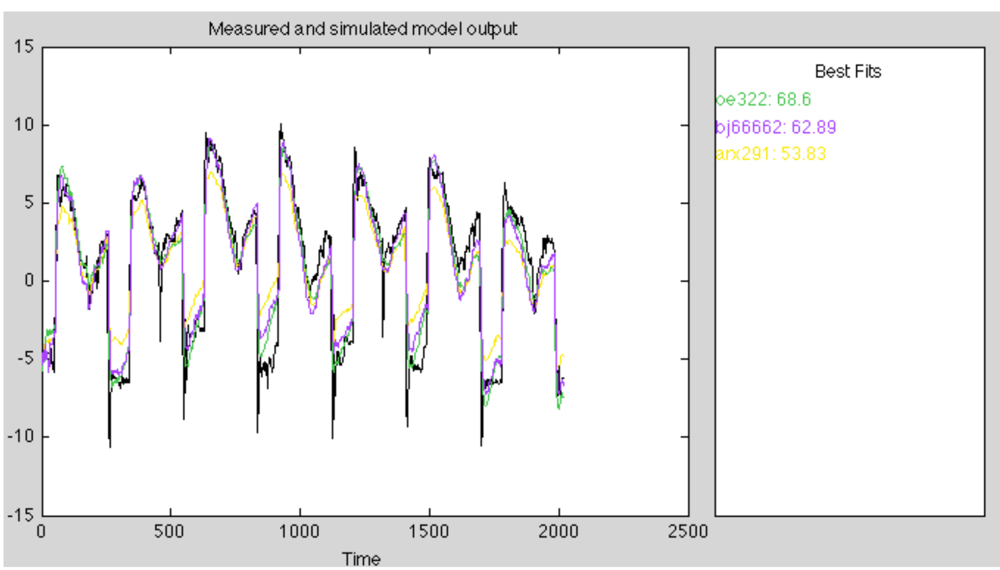


Figure 65: *24h ON Bi-Climatic* (with C1C2 profile as additional input) measured and simulated model outputs with their best fits

As can be seen all the selected models represent the real data very well. All of them are able to evaluate more than 50% of the values correctly but, more importantly, they are able to understand which climatic curve the boiler is using at the moment we are evaluating the outlet temperature. Therefore we can state that adding the C1C2 profile was a good strategy to obtain a correlation between the outdoor and the indoor temperatures with the outlet one. This was to be expected because giving the C1C2 profile as an input can confuse the program, which has to cope with different outlet temperatures for similar, if not equal, conditions. Considering only the Best-Fit numbers we can see that the OE322 model computes the outlet temperature with a small margin of error only. Furthermore this OE is the one with the lowest computational costs because of the low coefficient orders it presents. Therefore we can state that the green line represents the model that best evaluates the outlet temperature (the OE322) using the inputs given.

5.4 *Bi-Climatic WF* Control Strategy

The bi-climatic WF is a control strategy based on weather forecasts. It means that in addition to the two climatic curves the boiler uses to adjust the outlet temperature during daytime hours and during night hours, the system is also switched off during the warmest hours of the day (around 1pm) for a certain amount of time depending on the outdoor temperature. It means that besides the C1C2 profile, this control strategy also presents a variable ON-OFF profile. As demonstrated before we cannot expect to create a correlation model for this kind of system without giving it the ON-OFF and the C1C2 profiles as inputs. Therefore the inputs we used are the outdoor temperature (u1), the indoor temperature (u2) and the two profiles: C1C2 (u3) and ON-OFF (u4):

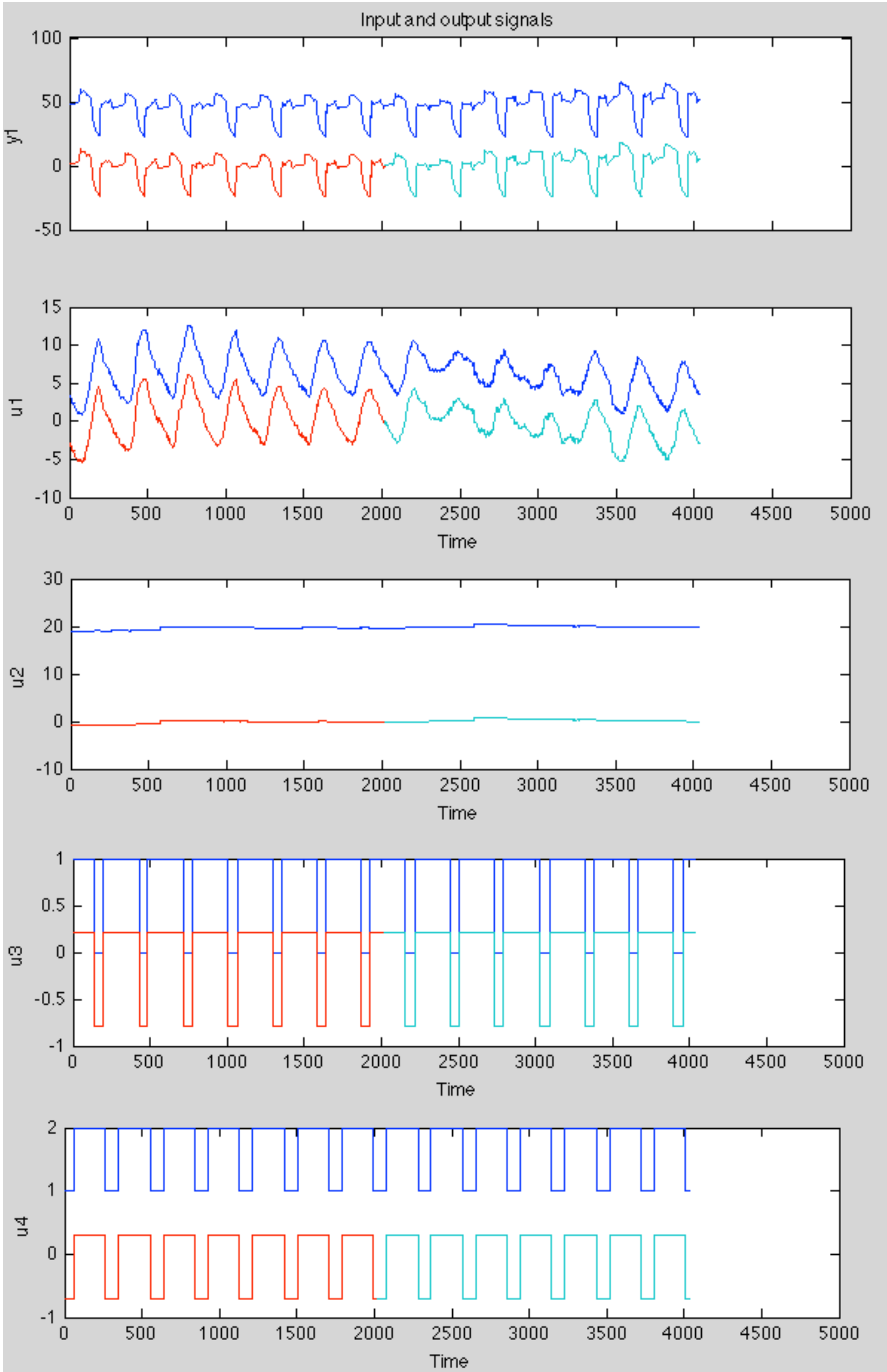


Figure 66: *Bi-Climatic WF* inputs (u1, u2, u3 and u4) and output (y1) signals

In this case the evaluating data set is represented with a red line while a cyan line represents the validation data set. Computing the ARX models as described before we obtained the following graph for the 'Order Selection' command:

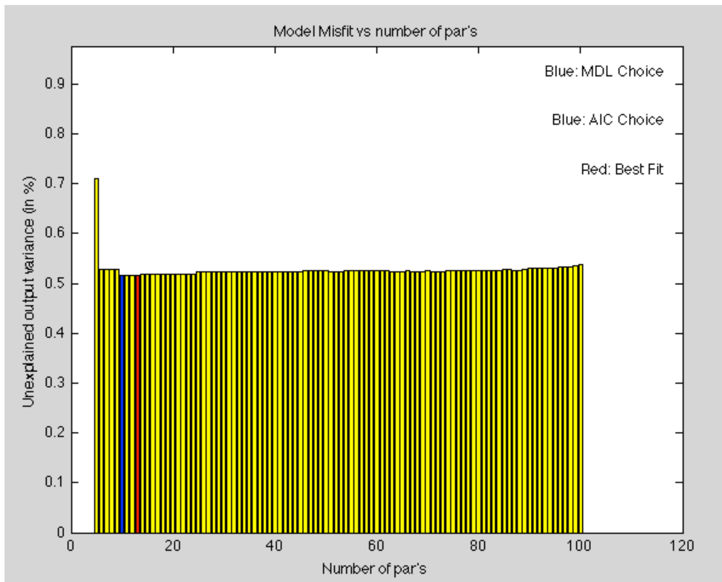


Figure 67: *Bi-Climatic WF* ARX models misfit vs number of parameters

Having an idea of the order of some of the coefficients that best fit the model we were searching for we continued our analysis evaluating some ARMAX, OE and BJ models. After having tried many other models we selected the best four for each type and we collected them in the following chart:

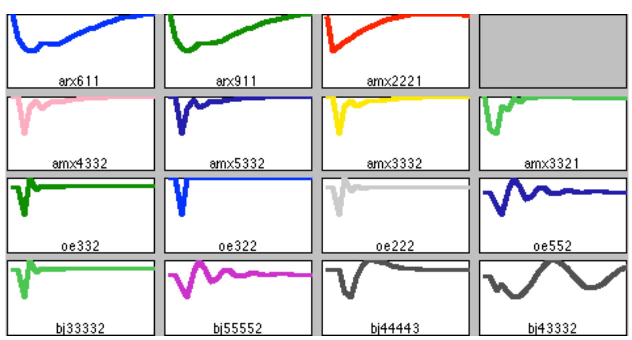


Figure 68: *Bi-Climatic WF* best estimation models

Representing the outlet temperature obtained by some of these models on a graph and comparing it with the real outlet temperature (validation data) with respect to time we have:

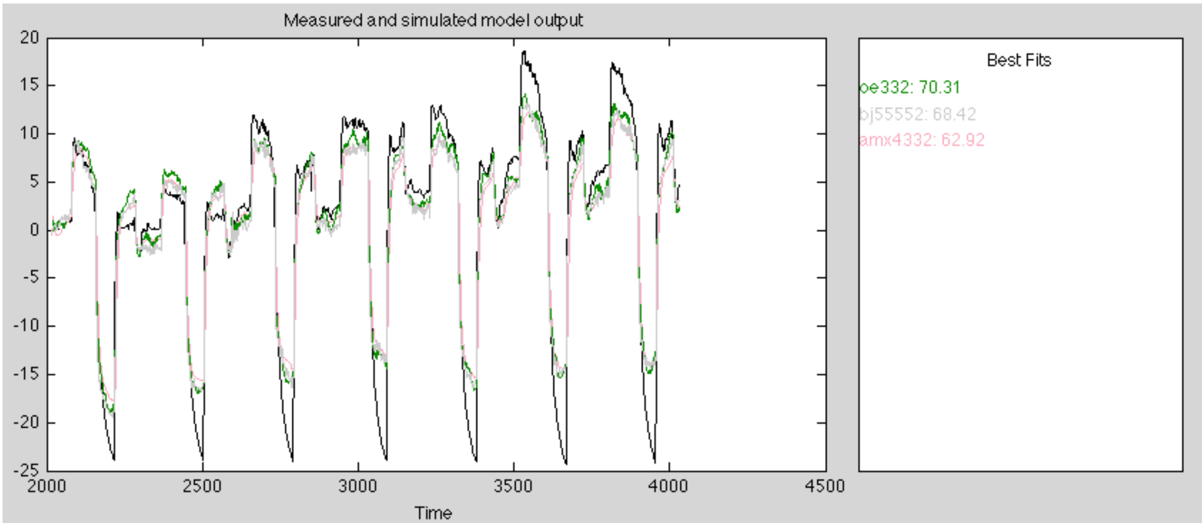


Figure 69: *Bi-Climatic* WF measured and simulated model outputs with their best fits

As can be see from the previous graph all the selected models represent the real data very well. All of them are able to evaluate more than the 60% of the values correctly but, more importantly, they are able to understand if the boiler is switched on or not and which climatic curve it is using an the moment we are evaluating the outlet temperature.

Considering only the Best Fit numbers we can see that the first OE332 model computes the outlet temperature with a small margin of error. Furthermore this OE is the one that has the lowest computational costs because of the low coefficient orders it presents. Therefore we can state that the green line, which represents the OE322 model, represents the model that best evaluates the outlet temperature.

CHAPTER 6

OTHER MODELS ESTIMATION

In this chapter we will describe all the other models we computed and the reasons why we created them. First we will concentrate our attention on the modeling of the indoor temperature from the outlet and the outdoor temperatures. Then we will describe how we correlated the outlet temperature with the gas consumption and the correlation we found between the outlet temperature and the temperature of the water coming back to the boiler.

6.1 Models to Estimate the Indoor Temperature From the Outlet Temperature

From the analysis in the previous chapter we can see that, in order to give the boiler an outlet temperature profile which can ensure a certain temperature inside the building and to minimize gas consumption, we need:

- Weather forecasts for the outdoor temperature;
- The comfort temperature we want to attain inside the building;
- The ON-OFF profile that best suits the outdoor weather conditions.

Unfortunately we do not have the third input we need to make this regulation possible. In order to overcome this problem we tried to find a model able to relate the outlet temperature and the ON-OFF profile with the indoor temperature – basically to find a new correlation between the two unknowns we had (the outlet temperature and the ON-OFF profile).

The first correlation we tried to obtain was the one linking the outlet temperature (u1) and the ON-OFF profile (u2) with the indoor temperature (y1). We did it for both the *24h ON* and the *18h ON* control strategies.

6.1.1 *24h ON* Control Strategy

As outlined in the previous chapter, the first thing to do, before starting a system identification procedure, is to clean the data and to save them on a .csv file. Once this had been done the data was then imported into the MatLab workspace, once more using the 'csvimport' function outlined in Appendix A as follows:

```
>> all
close all
%load the data file

path = 'I.csv';

[Text,ONOFF,Consumo,Tmand,Tint_all,Tint_uff,Trit] =
          csvimport(path,'delimiter',';','column',[3,4,5,6,7,8,9],
          'outputAsChar', false,'noHeader',true);
>>
```

We then opened the System Identification Toolbox for MatLab and we imported the input and output we needed for our model:

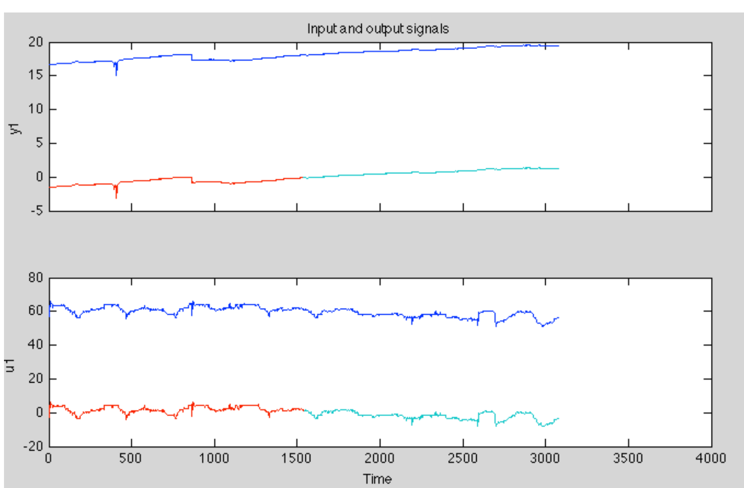


Figure 70: *24h ON* input (u1) and output (y1) signals vs time for the Tin estimation

In this graph we can see the real input and output represented by a blue line, and the input and output without their means represented by a different color. The red line represents the estimation data while the cyan one represents the validation data.

Estimating the ARX models using the order selection command we obtained:

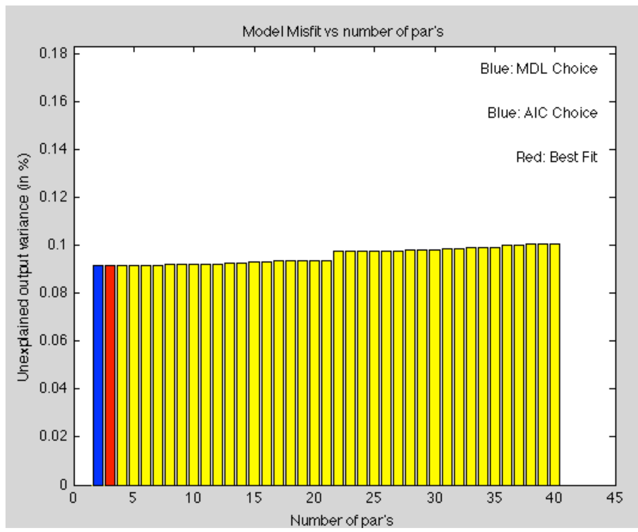


Figure 71: *24h ON* ARX models misfit vs number of parameters for the Tin estimation

Having an idea of the order of some of the coefficients that best fit the model we were searching for we continued our analysis evaluating ARMAX, OE and BJ models. After having tried more than other 500 models we selected the best four for each type (ARX, ARMAX…) They are:

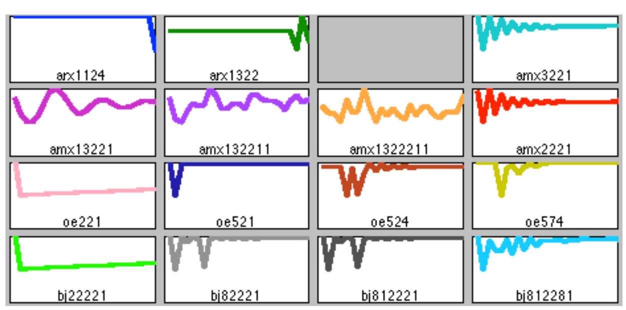


Figure 72: *24h ON* best estimation models for the Tin

Representing the indoor temperature obtained by these models on a graph together with the real indoor temperature (each of them without its mean) we have:

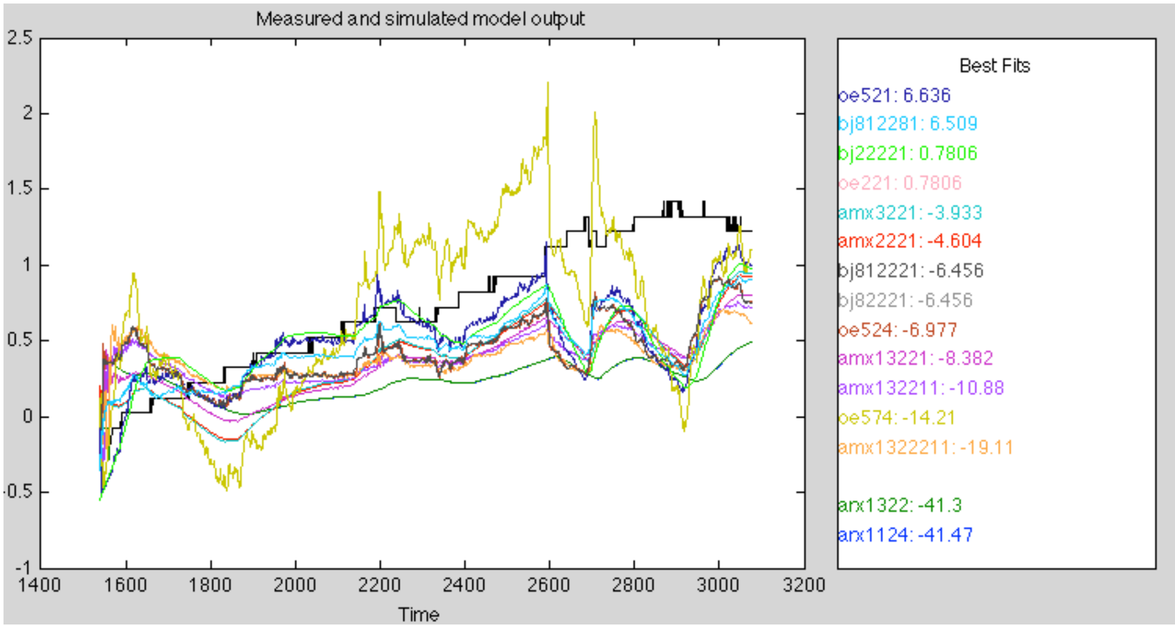


Figure 73: *24h ON* measured and simulated model (estimating Tin) outputs with their best fits

As we can see from the graph none of the evaluated models is able to describe the indoor temperature well enough for us to base a control strategy on it.

6.1.2 *18h ON* Regulation Strategy

Using the same approach as that for the *24h ON* control strategy, we imported the outlet temperature and the ON-OFF profile as inputs and the indoor temperature as output to the System Identification Toolbox. Representing both the real data set and the one without a mean on a graph we obtained:

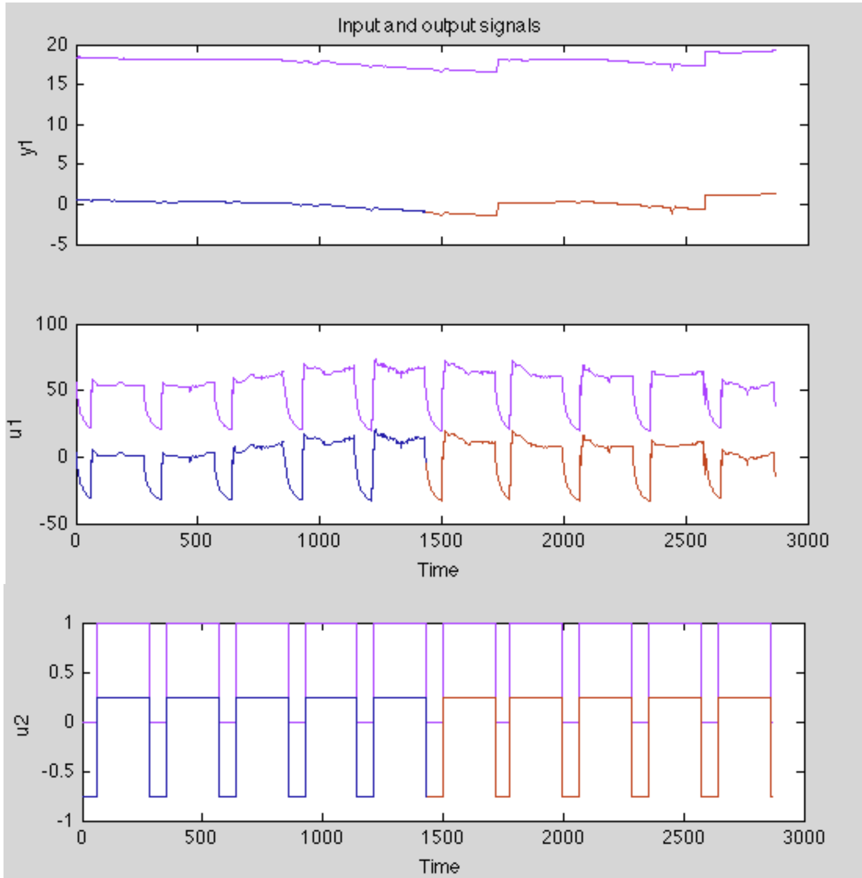


Figure 74: *18h ON* input (u1) and output (y1) signals vs time for the Tin estimation

In this graph we can see the real input and output represented by a violet line, and the inputs and outputs without their means represented by a different color. In particular the blue line represents the estimation data while the red one represents the validation data.

Estimating the ARX models using the order selection command we were able to get an idea of the order of some of the coefficients that best fit the model we were searching for. We continued our analysis by evaluating ARMAX, OE and BJ models. After having tried more than 800 other models we chose the following as the ones that are best able to evaluate the indoor temperature using the inputs given:

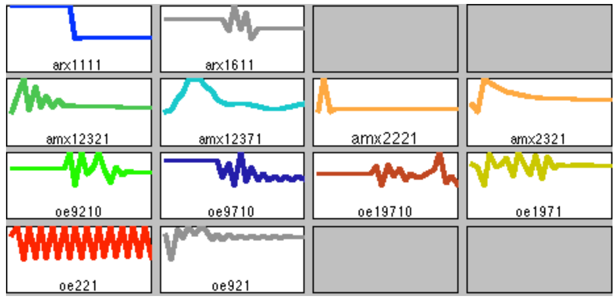


Figure 75: *18h ON* best estimation models for the Tin

Representing the indoor temperature obtained by these models on a graph together with the real indoor temperature (each of them without its mean) we have:

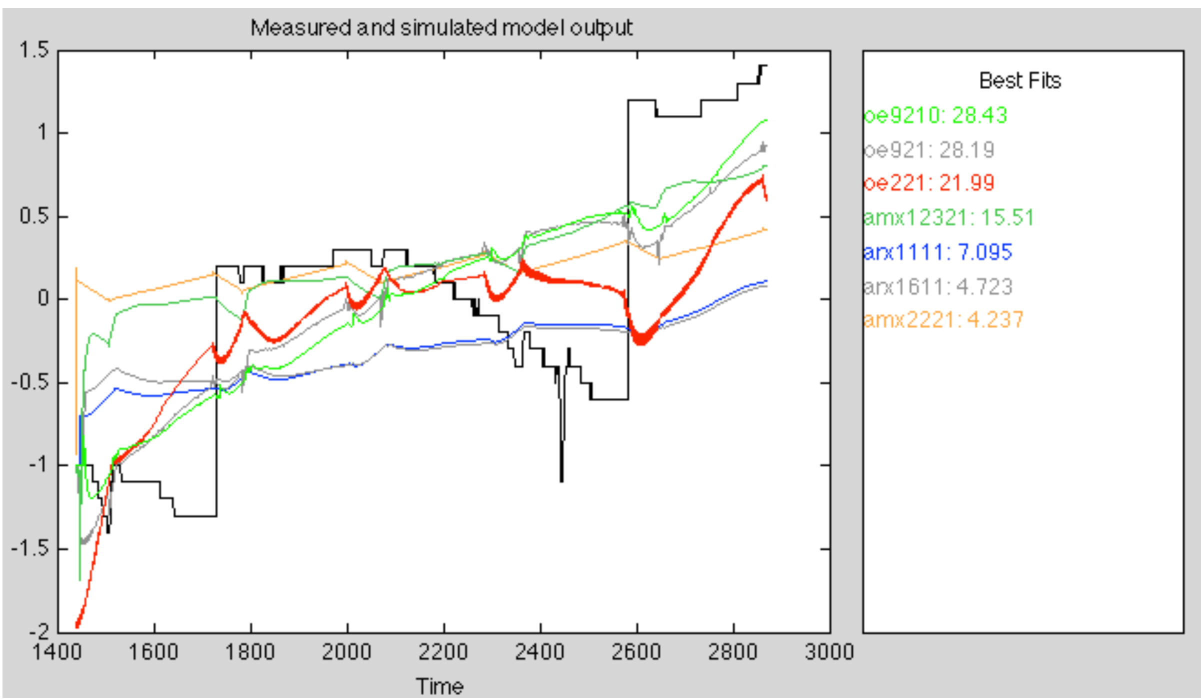


Figure 76: *18h ON* measured and simulated model (estimating Tin) outputs with their best fits

As we can see from the graph none of the evaluated models is able to describe the indoor temperature well enough for us to base a regulation strategy on it.

6.2 Model to Estimate the Indoor Temperature From the Outdoor Temperature and the Outlet Temperature

Since we weren't able to find a correlation model linking the outlet temperature with the indoor one (even using the ON-OFF profile), we decided to use another approach: to create a set of "Case Study Days" and of "Case Study Regulation Strategies" in order to evaluate the ON-OFF profile that best fits the outdoor conditions and the temperature we want to create inside the building, optimizing the choice according to gas consumption. Before doing this we needed to create a correlation model able to relate the outdoor temperature and the outlet one with the indoor temperature. In this case we decided not to use the ON-OFF profile as an input any more because we thought that this was provided implicitly by the outlet temperature. Using the *18h ON* data set, we imported all the inputs and outputs to the MatLab workspace and then we imported them to the System Identification Toolbox. Representing these data on a graph with the time on the x-axis we have:

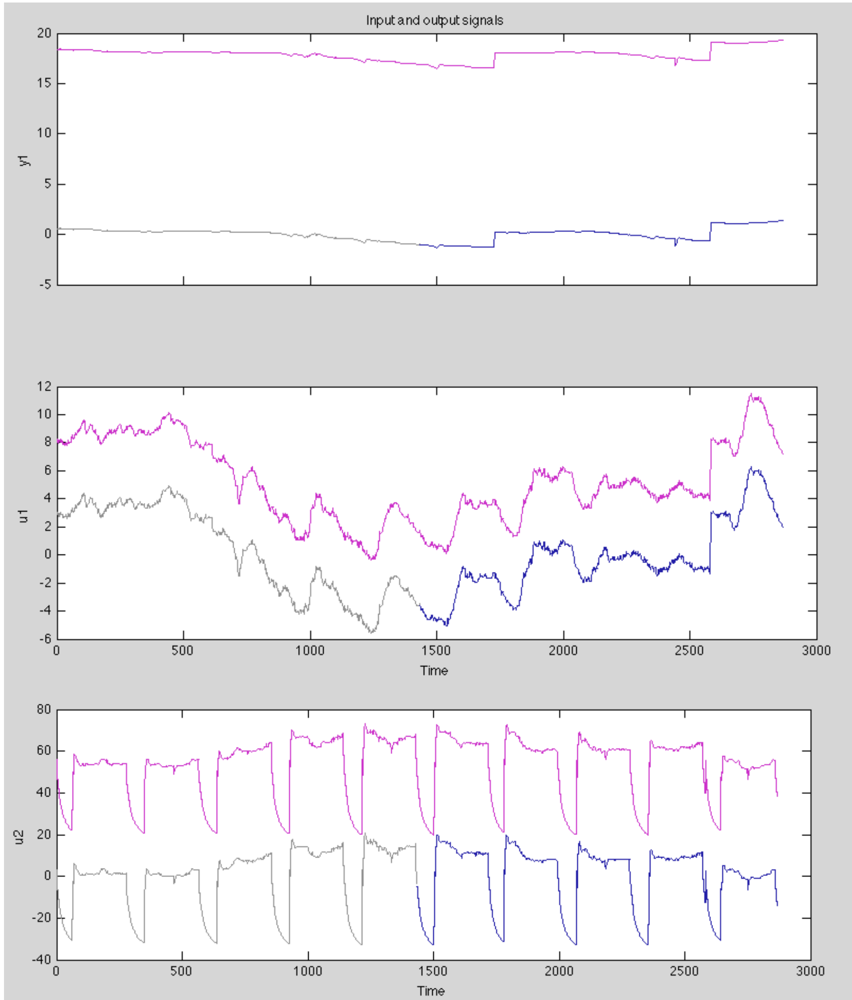


Figure 77: *18h ON* inputs (u1, u2) and output (y1) signals vs time for the Tin estimation

In this graph we can see the real inputs and outputs represented with a purple line, and the inputs and outputs without their means represented with a different color. In particular the grey line represents the estimation data while the red one represents the validation data.

Estimating the ARX models using the order selection command we were able to get an idea of the order of some of the coefficients that best fit the model we were searching for. We continued our analysis evaluating ARMAX, OE and BJ models. After having tried more than 200 other models we chose the following as the ones that were best able to evaluate the indoor temperature using the inputs given:

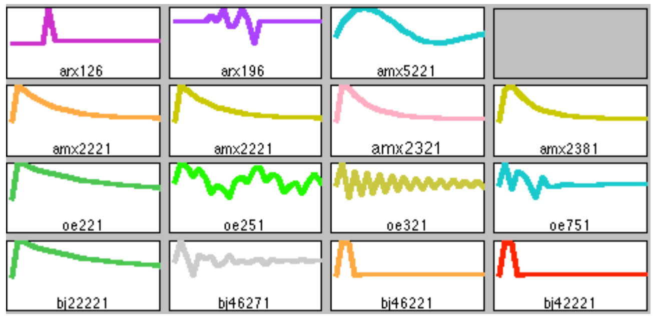


Figure 78: *18h ON* best estimation models for the Tin

Representing the indoor temperature obtained by these models on a graph together with the real indoor temperature (each of them without its mean) we have:

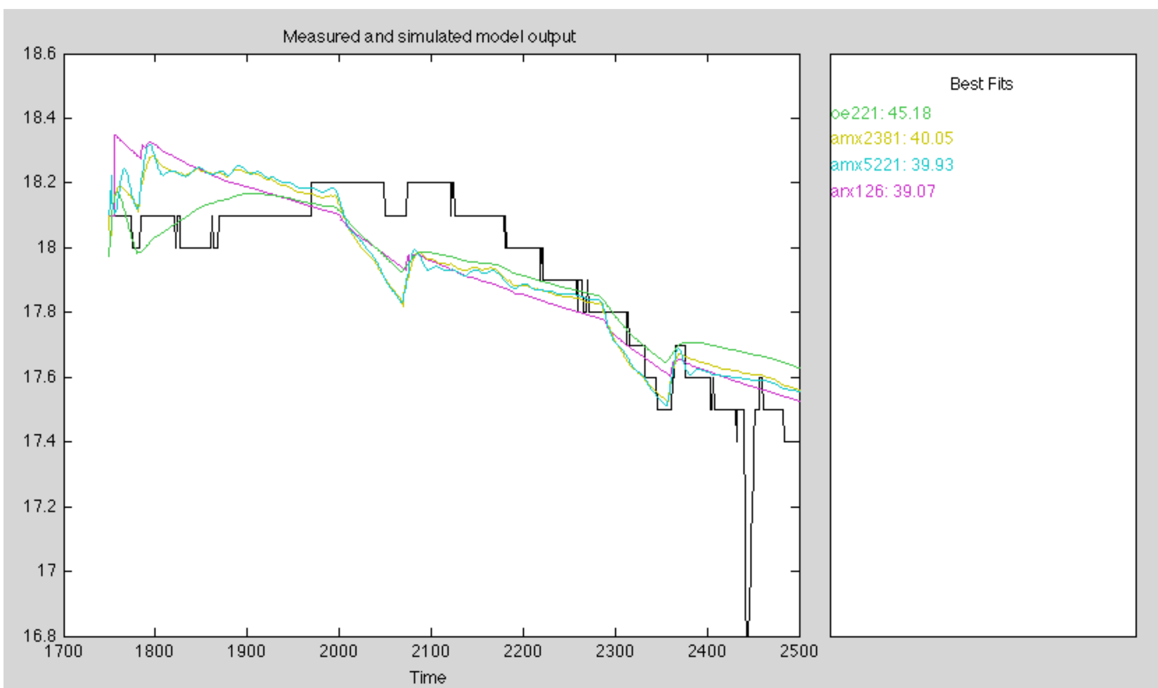


Figure 79: *18h ON* measured and simulated model (estimating Tin) outputs with their best fits

As we can see, the green line representing the OE221 model provides a precise estimate of the indoor temperature, never falling short by more than 0.3 degrees. Therefore we can state that the outlet temperature, the outdoor temperature and the indoor temperature are somehow interrelated.

6.3 Model to Estimate the Gas Consumption Knowing the Outlet Temperature

Before we stated that we wanted to create a control system able to choose the best ON-OFF strategy according to the outdoor temperature in order to optimize gas consumption. To do this we needed to find the relationship existing between the outlet temperature (u1) and the gas consumption (y1). To evaluate this model we decided to compare the 24h ON and 18h ON data sets. We did this in order to have a more complete data set and to avoid big drops in the data, caused by the interval between the days in which the two regulation strategies were applied.

The inputs we chose to compute the model were the outlet temperature and the ON-OFF profile while, obviously, we chose the gas consumption as the only output. After having imported these data to the MatLab workspace using the same script as before, we imported them to the System Identification Toolbox. Representing them on graph with the time on the x-axis we have:

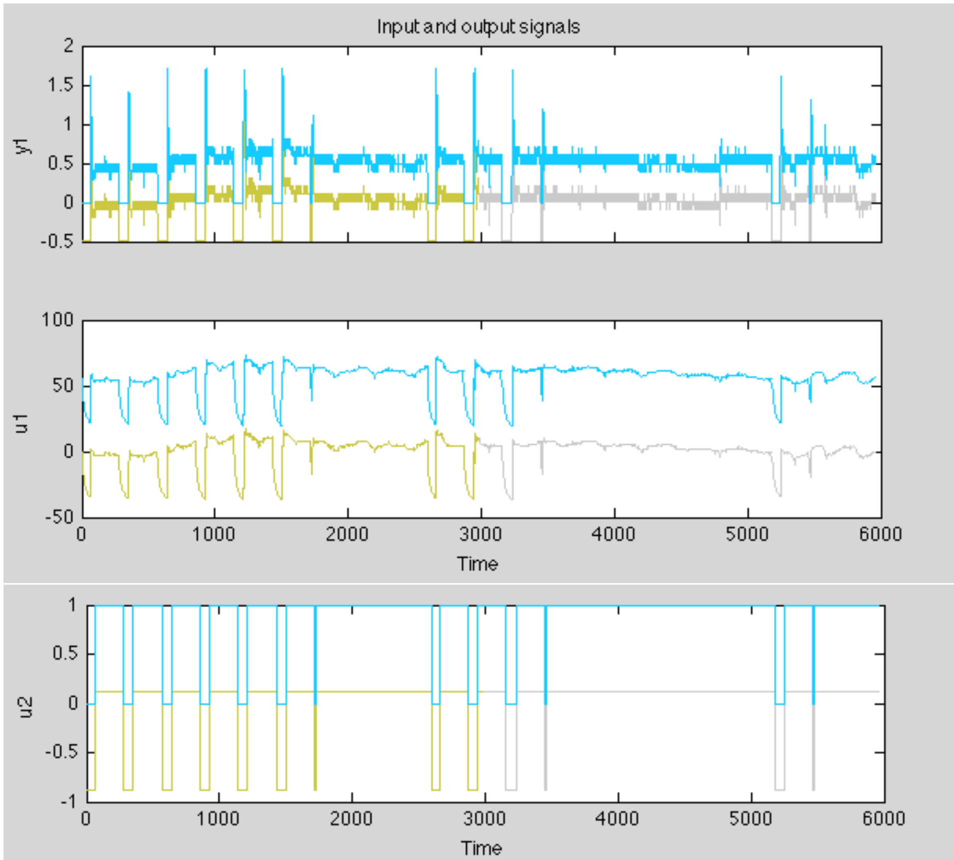


Figure 80: *18h ON* and *24h ON* input (u1) and output (y1) signals vs time for the GC estimation

In this graph we can see the real inputs and outputs represented by a cyan line, and the inputs and outputs without their means represented by a different color. In particular the yellow line represents the estimation data while the grey one represents the validation data. Estimating the ARX models using the order selection command we obtained:

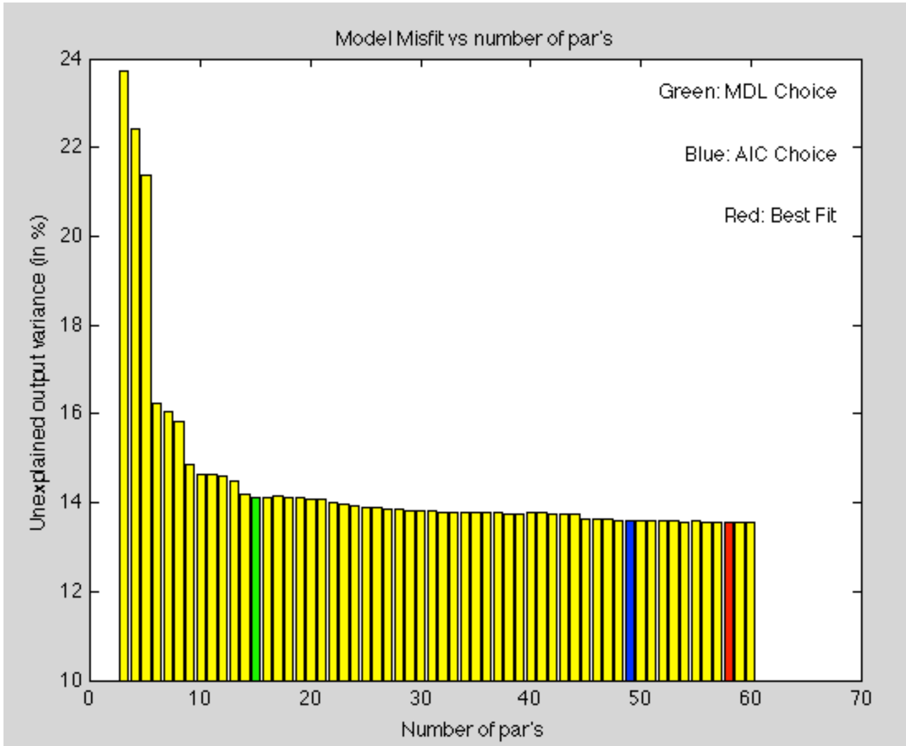


Figure 81: *18h ON* and *24h ON* ARX models misfit vs number of parameters for the Tin estimation

Having an idea of the order of some of the coefficients that best fit the model we were searching for we continued our analysis evaluating ARMAX, OE and BJ models. After having tried more than 800 other models we chose the following as the ones best able to evaluate the indoor temperature using the inputs given:

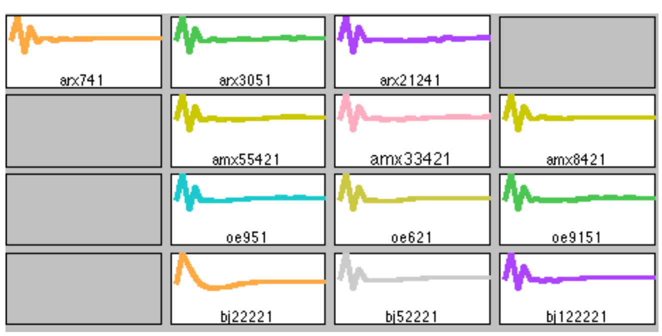


Figure 82: *18h ON* and *24h ON* best estimation models for the GC

Representing the gas consumption obtained for the only ARX741 (best fit 53.22) on a graph together with the real gas consumption (each of them without its mean) we have:

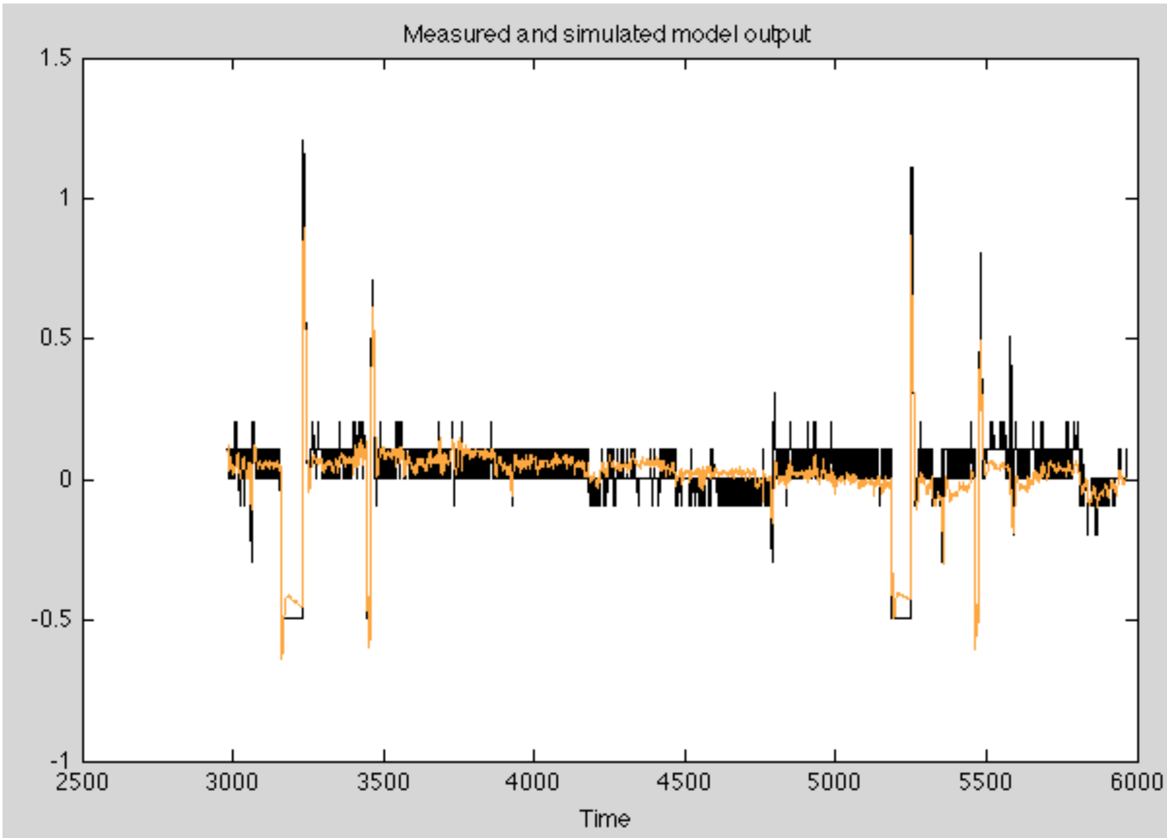


Figure 83: *18h ON* and *24h ON* measured and simulated model (estimating the GC) outputs with their best fits

The reason we decided to represent the values estimated by just one model is to make the graph easier to read. As can be seen, the orange line, representing the ARX741 model, provides a precise estimate of the gas consumption, or at least its average. Therefore we are now able to evaluate how much gas each regulation strategy will consume.

6.4 Model to Estimate the Return Water Temperature Knowing the Outlet One

The last model we evaluated is the one correlating the outlet temperature with the temperature of the water coming back to the boiler after having passed through the whole heating system. We will later show that we used this to calculate the power each radiator is giving the building for each outlet temperature the boiler will provide. In fact if this power is lower than a certain limit value we just want the boiler to turn off in order not to waste gas heating the water to a temperature not sufficient to heat the building. To compute this model we considered just one input: the outlet temperature (u1), and one output: the return water temperature (y1). Representing them on graph with the time on the x-axis we have:

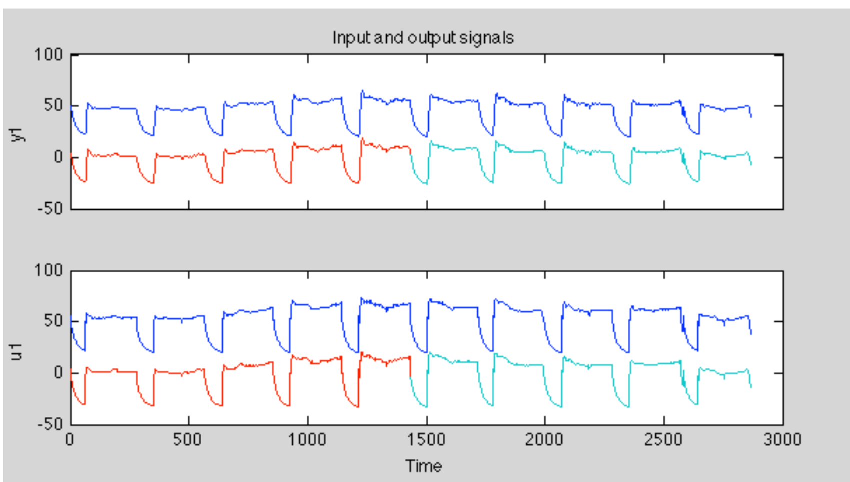


Figure 84: *24h ON* input (u1) and output (y1) signals vs time for the Tret estimation

In this graph we can see the real inputs and outputs represented by a blue line, and the inputs and outputs without their means represented by a different color. In particular the red line represents the estimation data while the blue one represents the validation data. Estimating the ARX models using the order selection command we obtained:

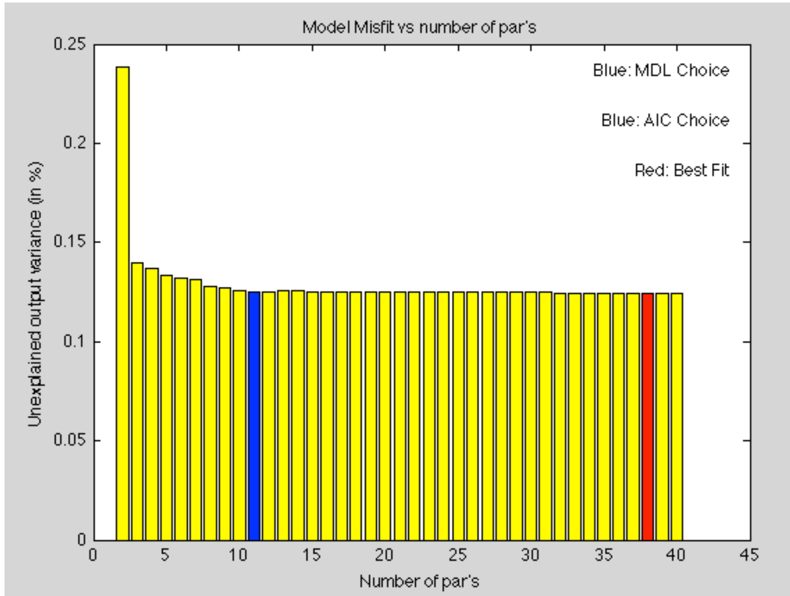


Figure 85: *24h ON* ARX models misfit vs number of parameters for the Tret estimation

Having an idea of the order of some of the coefficients that best fit the model we were searching we continued our analysis evaluating some more ARMAX models. In fact these were able to provide a very high reliability in the estimation of the return water temperature (best fit over 80). Therefore we chose to use them for their simplicity. After having tried more than 300 possible combinations of coefficient orders, we chose the following as the ones that best evaluate the return water temperature using the outlet temperature as an input:

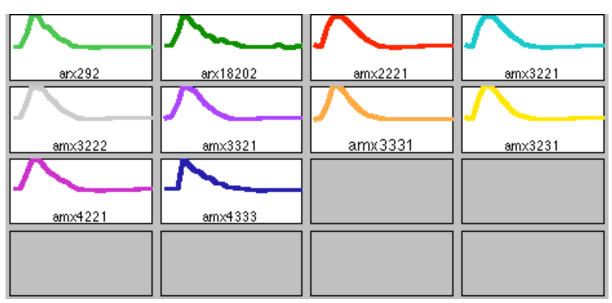


Figure 86: *24h ON* best estimation models for the Tret

Representing the gas consumption obtained for the only ARMX3222 (best fit 88.96) on a graph together with the real gas consumption (each of them without its mean) we have:

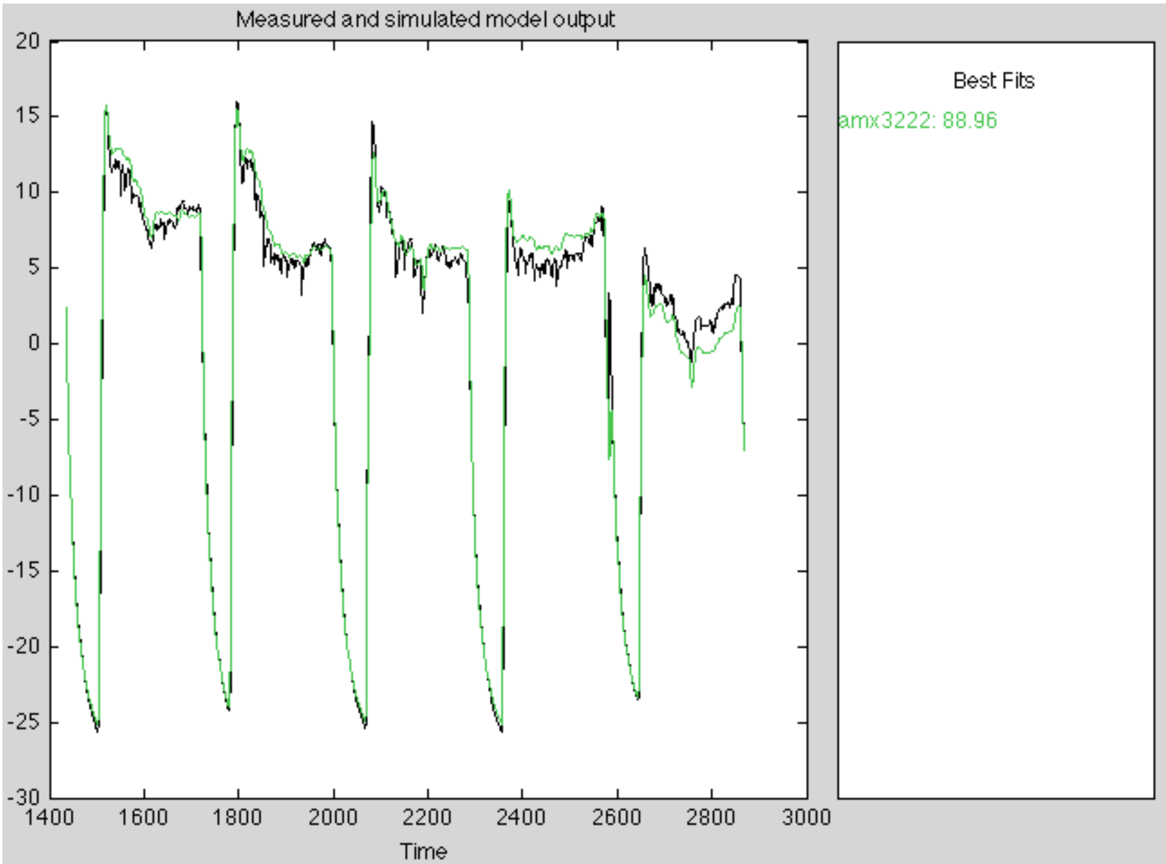


Figure 87: *24h ON* measured and simulated model (estimating the Tret) outputs with their best fits

The reason we decided to represent the values estimated by just one model is to make the graph easier to read. As can be seen, the green line, representing the ARMX3222 model, provides a precise estimate of the return water temperature. Therefore we are now able to evaluate the temperature difference of the water from the moment it leaves the boiler to the moment it comes back.

CHAPTER 7

THE ON-OFF PROFILE

As stated in the previous chapter in order to give the boiler an outlet temperature profile able to ensure a certain temperature inside the building and to minimize gas consumption we need:

- Weather forecasts for the outdoor temperature;
- The comfort temperature we want to ensure inside the building;
- The ON-OFF profile that best suits the outdoor weather conditions.

As we can't be sure of the third input in advance, in the previous chapter we tried to find a direct relationship with the other system variables – but it didn't give us the results we expected. However, the study presented in the previous chapter gave us three new relationships between these data:

A. Correlation between the outdoor and the outlet temperatures and the indoor temperature;
B. Correlation between the outlet temperature and gas consumption;
C. Correlation between the outlet temperature and the return water temperature.

The third will be used later to create a control loop inside the control system to prevent the boiler from working when the return water is exchanging almost no power with the building.

In this chapter we will explain how we used the first and the second relationship to obtain an ON-OFF profile with respect to the mean outdoor temperature and the outdoor temperature drop of the day in consideration. Since these relationships do not directly present the ON-OFF profile as an output we will proceed by trial and error, considering a certain number of sample days and sample control strategies.

The main idea of the procedure we are going to describe is to divide the possible temperatures of a winter day into four main categories:

1. Outdoor Temperature < 0°C;
2. 0°C < Outdoor Temperature < 3°C;
3. 3°C < Outdoor Temperature < 7°C;
4. Outdoor Temperature > 7°C.

Each of which can be further divided into two subcategories:

1. Small Outdoor Temperature Drop during the day;
2. Big Outdoor Temperature Drop during the day.

To take a sample day for each category and to apply to each of them a certain number of sample control strategies:

1. 24h ON at different temperatures;
2. 18h ON at different temperatures;
3. 18h ON (with two switch-offs) at different temperatures;
4. 13h ON at different temperatures;
5. 24h ON with Bi-Climatic curve at different temperatures;
6. 24h ON real Climatic curve equation.

The reason the Bi-Climatic WF wasn't considered as a possible structure is that it was only evaluated for very warm days and so the correlation model obtained isn't reliable for describing the outlet temperature for cold days.

Through plotting the indoor temperature for each sample day we were able to see which control strategy was able to ensure a certain comfort temperature inside the building. By computing the gas consumption for each control strategy we could then see that the control strategy that was already able to ensure a comfort temperature inside the building was also the one that minimized gas consumption.

7.1 Sample Days

Here are the days we chose as representative of each category outlined before, each day with a brief description.

7.1.1 Outdoor Temperature < 0°C

An outdoor temperature of lower than 0°C is quite unusual in Turin, in fact it occurs no more than 2 weeks a year. However we thought it was important to study it anyway in order to provide a comfort temperature inside the building on these rare occasions.

The sample days we chose for this category are:

7.1.1.1 12/17/2010 for low temperature drops

This day presented the following characteristics:

- Mean Temperature = -2.72°C;
- Temperature Drop = 2.83°C;

Temperature distribution:

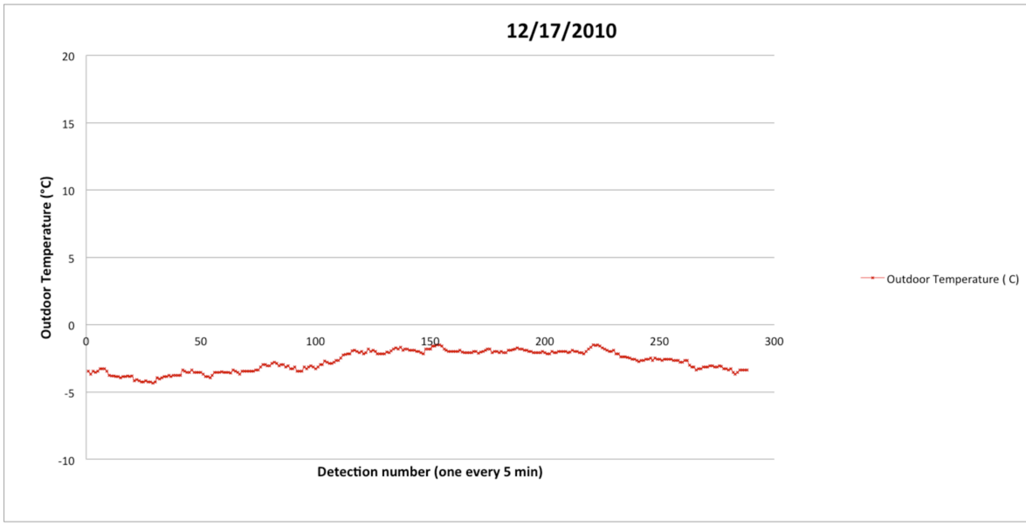


Figure 88: 12/17/2010 outdoor temperature distribution

7.1.1.2 12/18/2010 for high temperature drops

This day presented the following characteristics:

- Mean Temperature = -2.38°C;
- Temperature Drop = 7.03°C;

Temperature distribution:

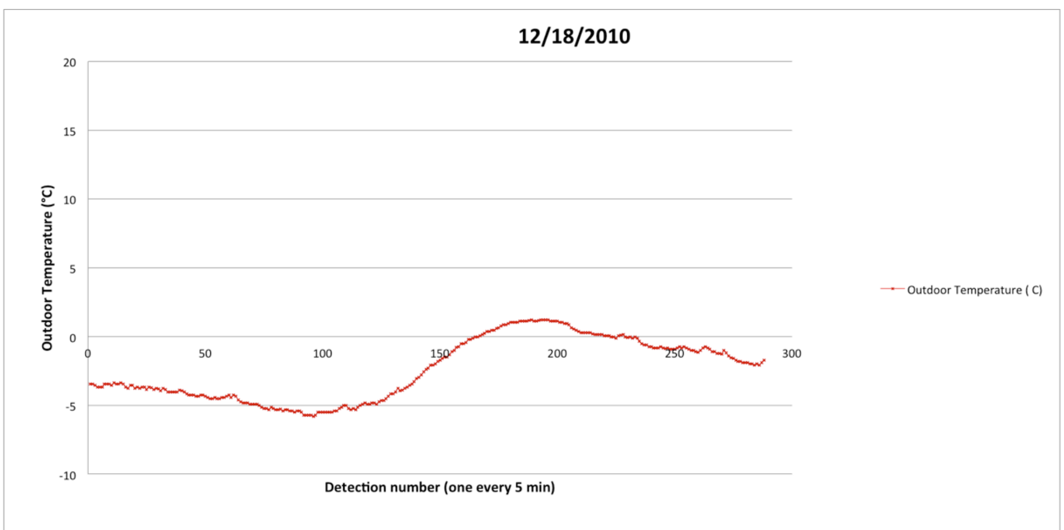


Figure 89: 12/18/2010 outdoor temperature distribution

7.1.2 Outdoor Temperature between 0°C and 3°C:

This situation is one of the most common during the winter in Turin, especially in December and January. The sample days we chose for this category are:

7.1.2.1 1/30/2010 for low temperature drops

This day presented the following characteristics:

- Mean Temperature = 1.19°C;
- Temperature Drop = 2.10°C;

Temperature distribution:

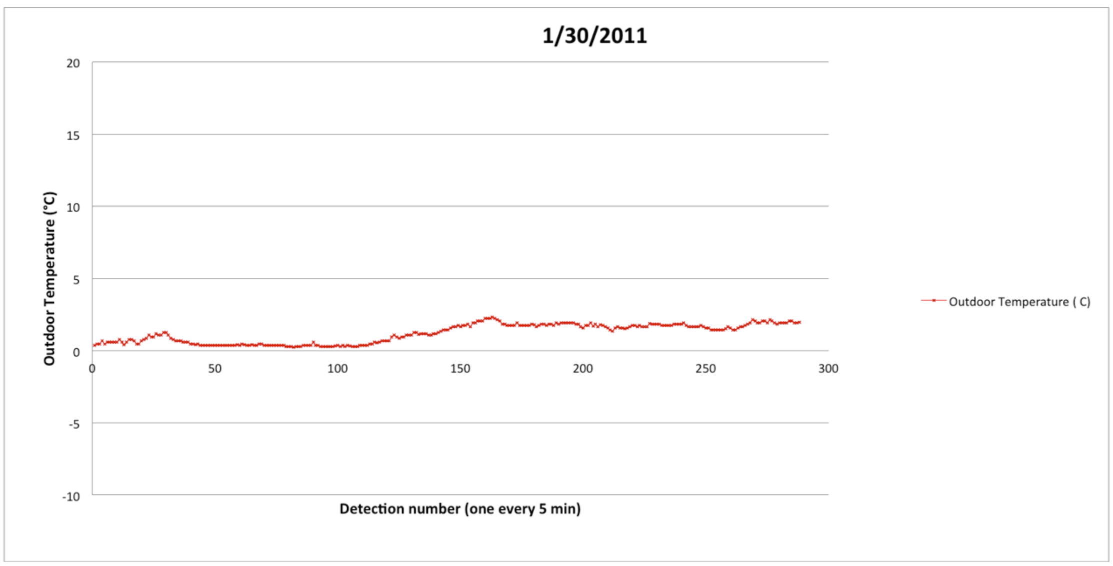


Figure 90: 1/30/2011 outdoor temperature distribution

7.1.2.2 1/24/2010 for high temperature drops

This day presented the following characteristics:

- Mean Temperature = 1.39°C;
- Temperature Drop = 7.64°C;

Temperature distribution:

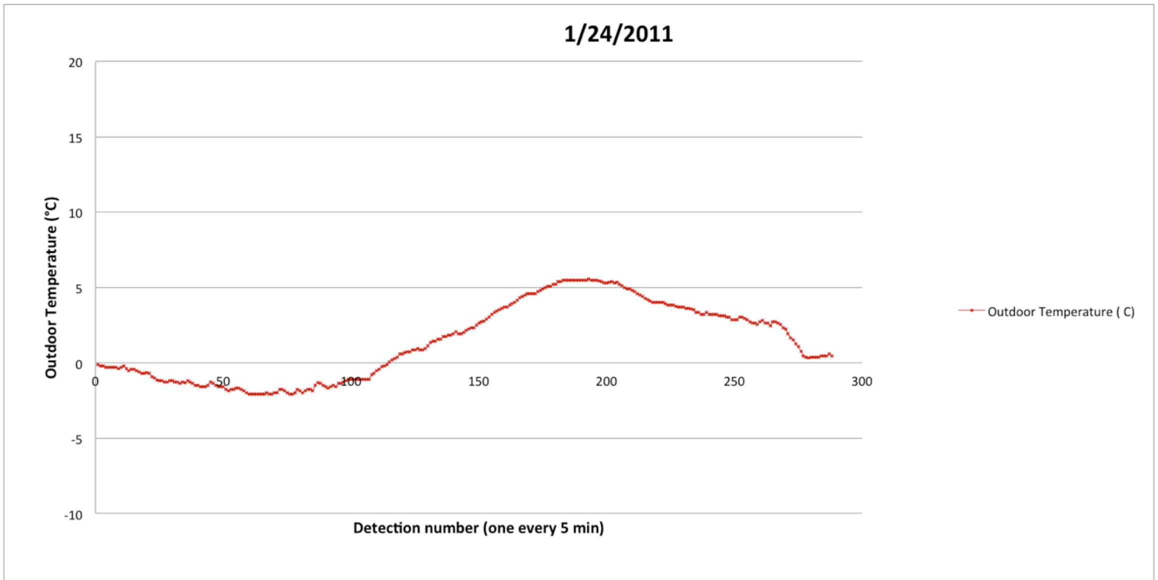


Figure 91: 1/24/2011 outdoor temperature distribution

7.1.3 Outdoor Temperature between 3°C and 7°C:

This situation is not very common during the winter in Turin, where the temperature is usually either under 3°C or above 7°C. We decided to model this category anyway in order to know what to tell the boiler when this transient temperature occurs. The sample days we chose for this category are:

7.1.3.1 1/10/2010 for low temperature drops

This day presented the following characteristics:

- Mean Temperature = 5.00°C;
- Temperature Drop = 2.25°C;

Temperature distribution:

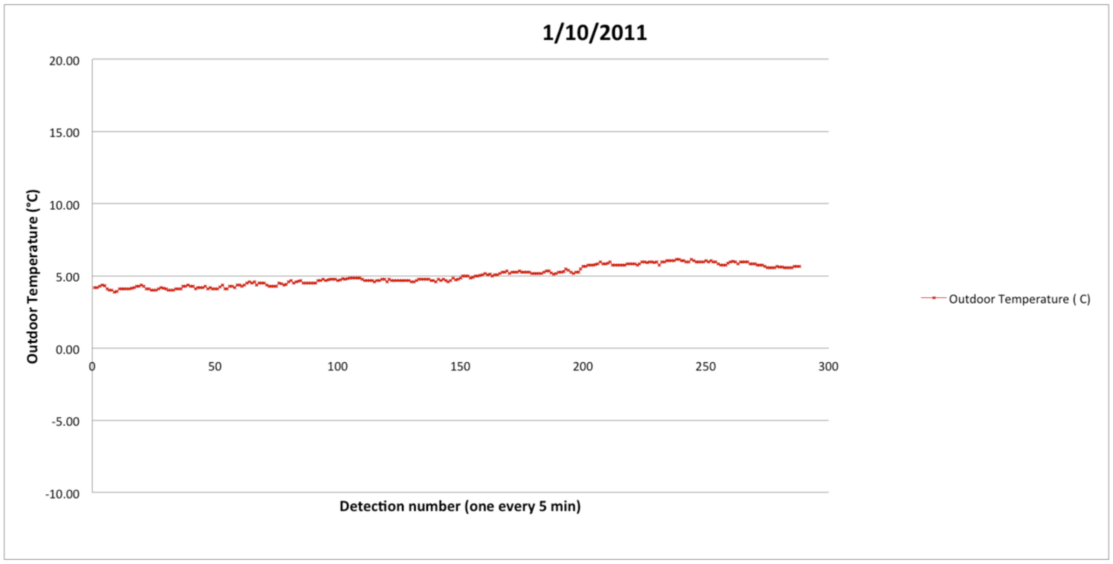


Figure 92: 1/10/2011 outdoor temperature distribution

7.1.3.2 11/29/2010 for high temperature drops

This day presented the following characteristics:

- Mean Temperature = 4.18°C;
- Temperature Drop = 10.38°C;

Temperature distribution:

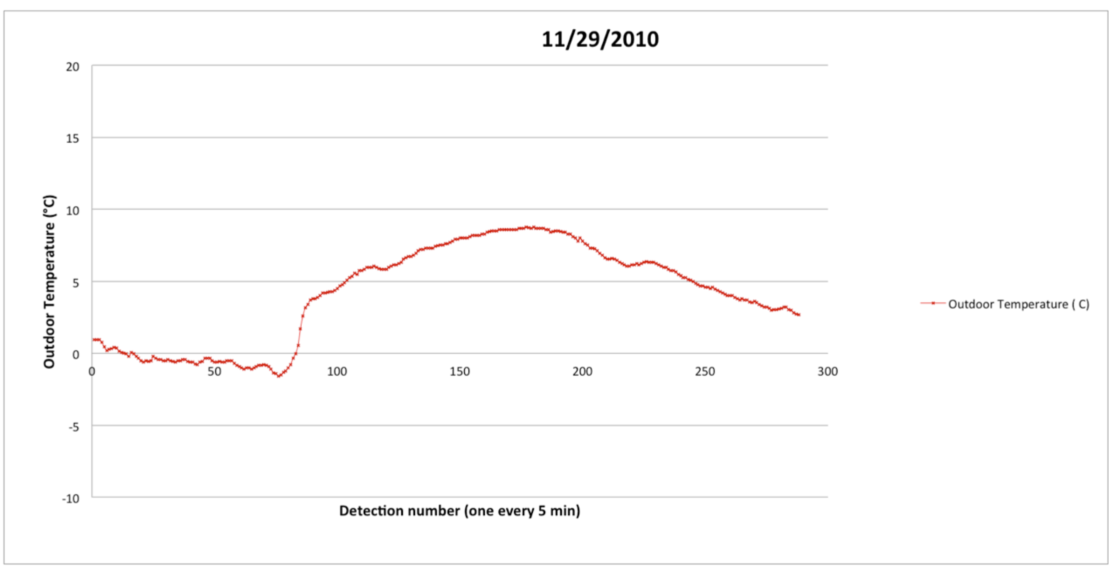


Figure 93: 11/29/2010 outdoor temperature distribution

7.1.4 Outdoor Temperature above 7°C:

This situation is very common at the end of winter and the beginning of spring in Turin, but it is very rare to find one of these days with a small temperature drop. In fact these days are usually sunny days, warm in the central hours of the day and cold at night. The sample days we chose for this category are:

7.1.4.1 3/22/2010 for low temperature drops

This day presented the following characteristics:

- Mean Temperature = 10.57°C;
- Temperature Drop = 7°C;

Temperature distribution:

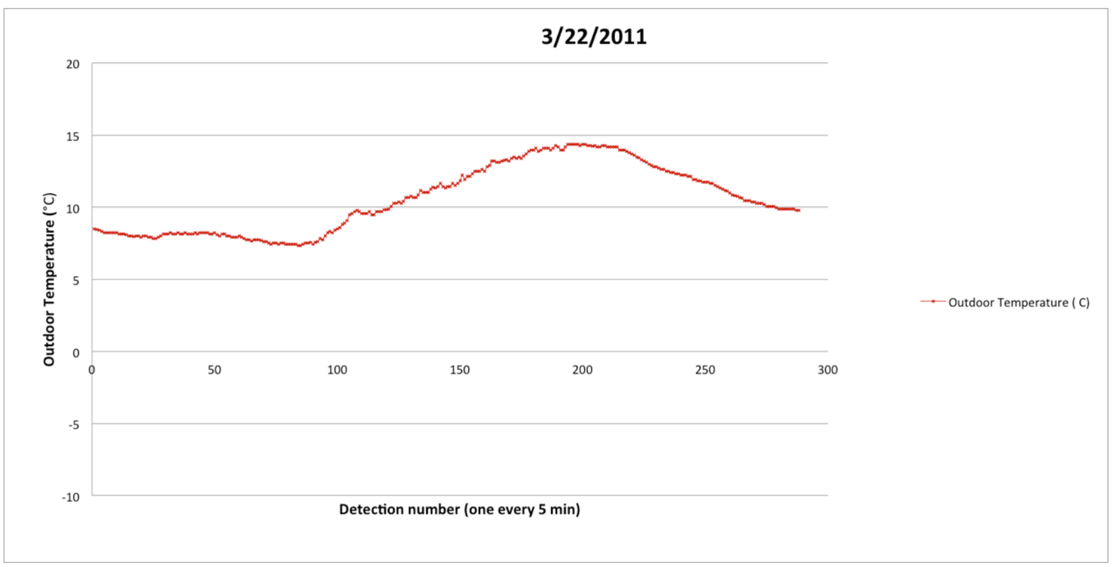


Figure 94: 3/22/2011 outdoor temperature distribution

7.1.4.2 3/18/2010 for high temperature drops

This day presented the following characteristics:

- Mean Temperature = 12.06°C;
- Temperature Drop = 10.84°C;

Temperature distribution:

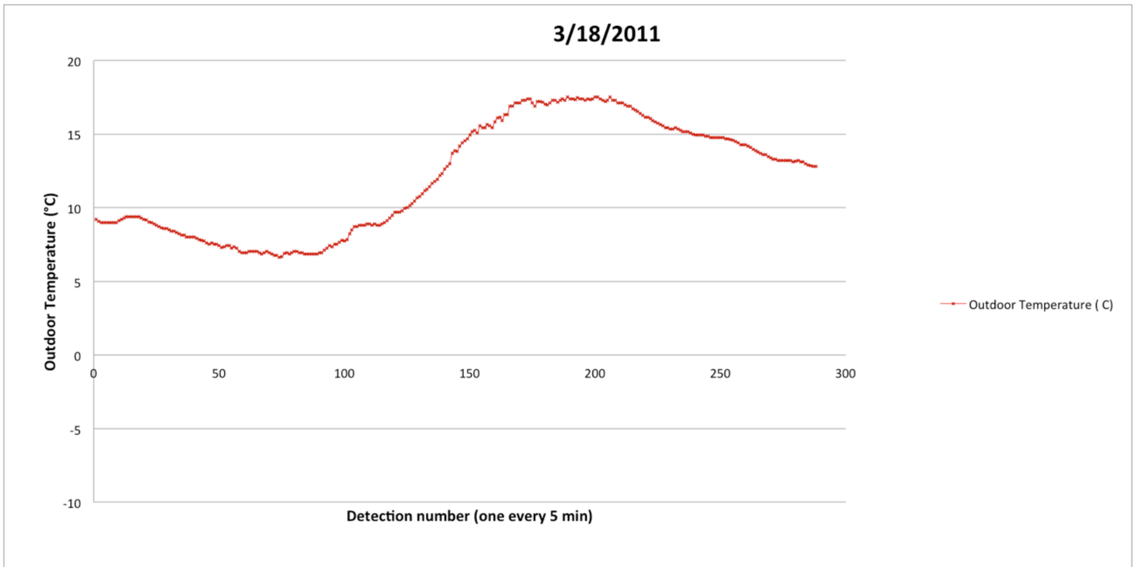


Figure 95: 3/18/2011 outdoor temperature distribution

7.2 Sample Control Strategies

In this paragraph we will list the control strategies we chose as representative of each category outlined before.

7.2.1 *24h ON* Control Strategy

In this first case the boiler is turned on 24 hours a day. The temperatures we considered as samples for this control strategy are:

7.2.1.1 Constant Temperature equal to 55°C

Outlet temperature distribution:

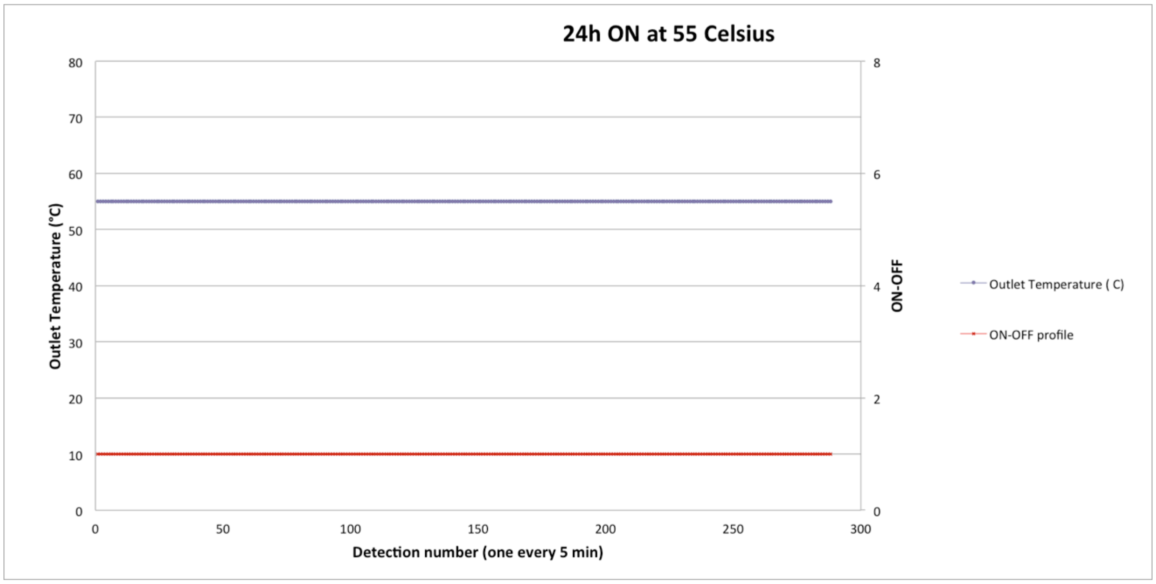


Figure 96: *24h ON* at 55°C daily outlet temperature distribution

7.2.1.2 Constant Temperature equal to 60°C

Outlet temperature distribution:

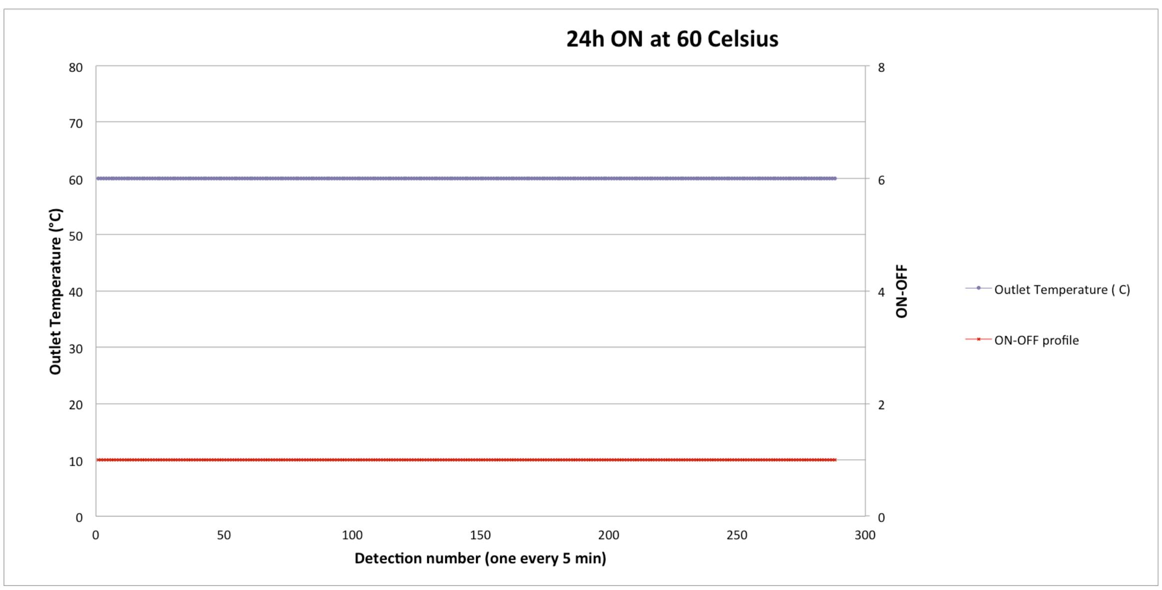


Figure 97: *24h ON* at 60°C daily outlet temperature distribution

7.2.1.3 Constant Temperature equal to 65°C

Outlet temperature distribution:

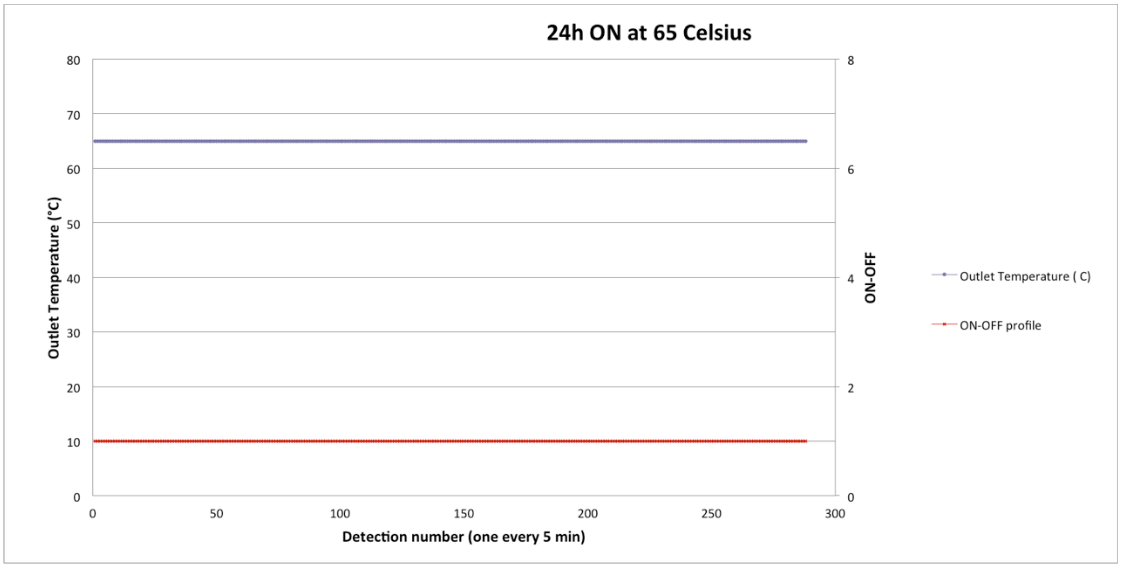


Figure 98: *24h ON* at 65°C daily outlet temperature distribution

7.2.1.4 Constant Temperature equal to 70°C

Outlet temperature distribution:

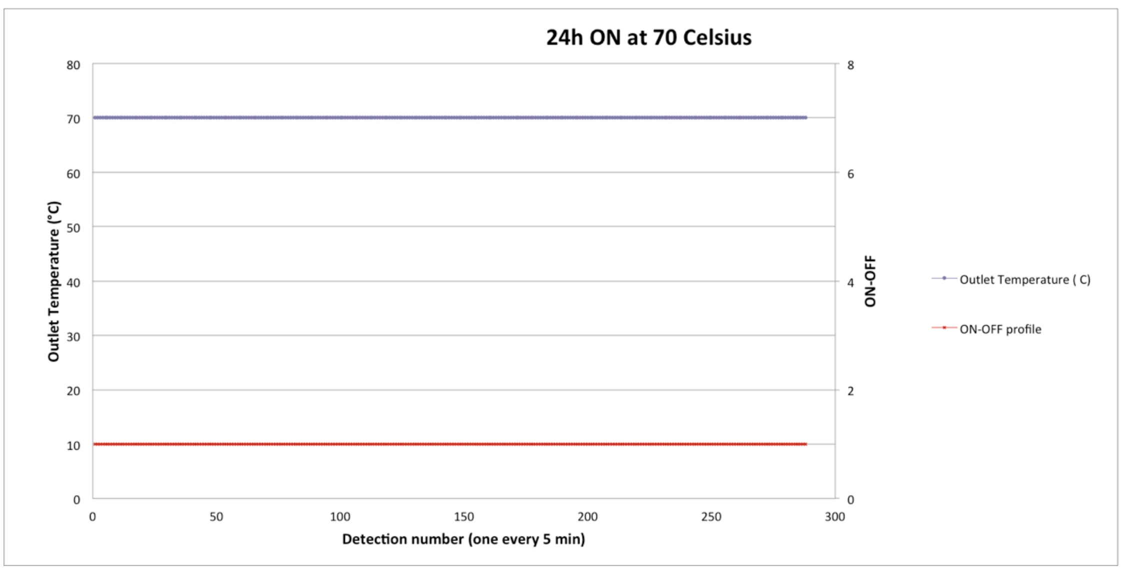


Figure 99: *24h ON* at 70°C daily outlet temperature distribution

7.2.2 *18h ON* Control Strategy

In this case the boiler is turned off at night between 11pm and 5am. The temperatures we considered as samples for this control strategy are:

7.2.2.1 Constant Temperature equal to 55°C when the boiler is turned on

Outlet temperature distribution:

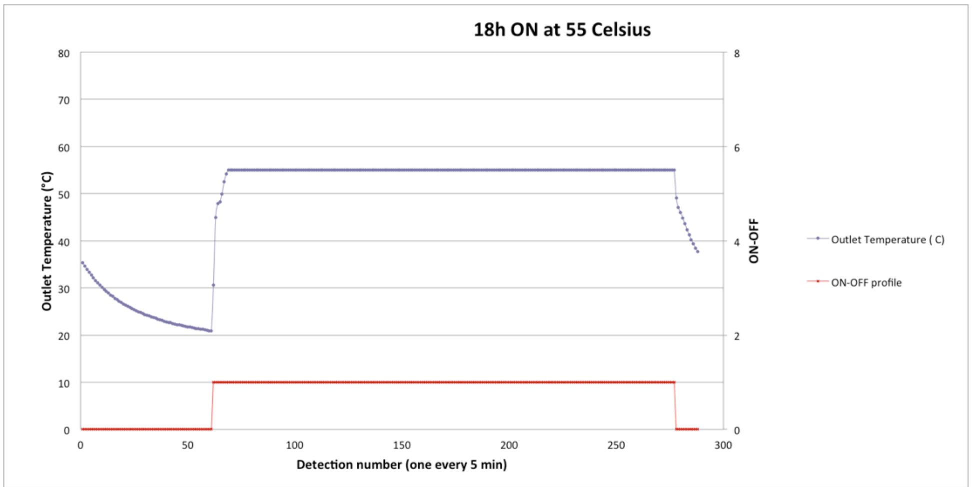


Figure 100: *18h ON* at 55°C daily outlet temperature distribution

7.2.2.2 Constant Temperature equal to 60°C when the boiler is turned on

Outlet temperature distribution:

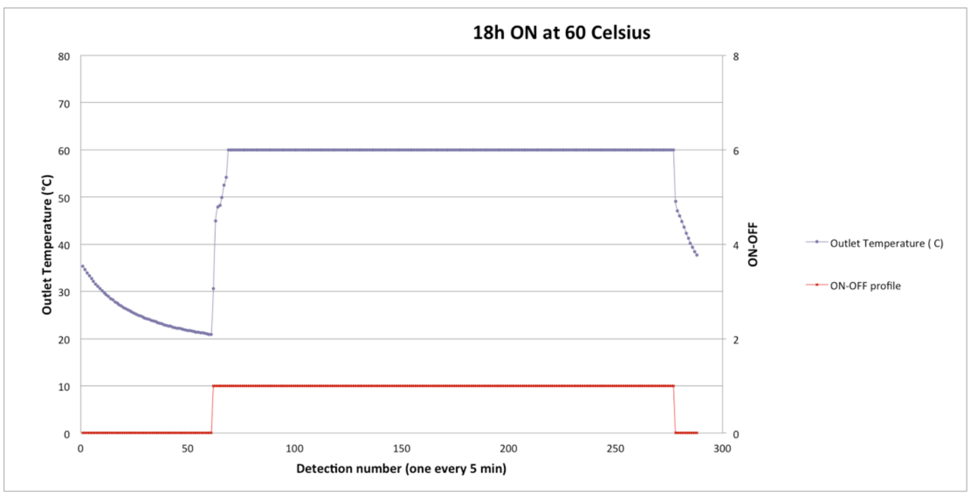


Figure 101: *18h ON* at 60°C daily outlet temperature distribution

7.2.2.3 Constant Temperature equal to 65°C when the boiler is turned on

Outlet temperature distribution:

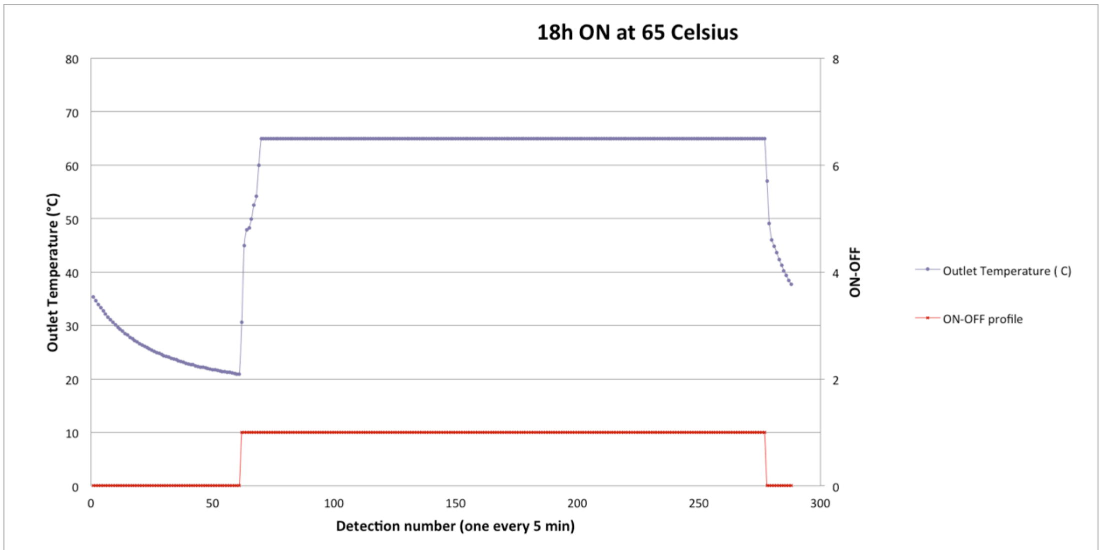


Figure 102: *18h ON* at 65°C daily outlet temperature distribution

7.2.2.4 Constant Temperature equal to 70°C when the boiler is turned on

Outlet temperature distribution:

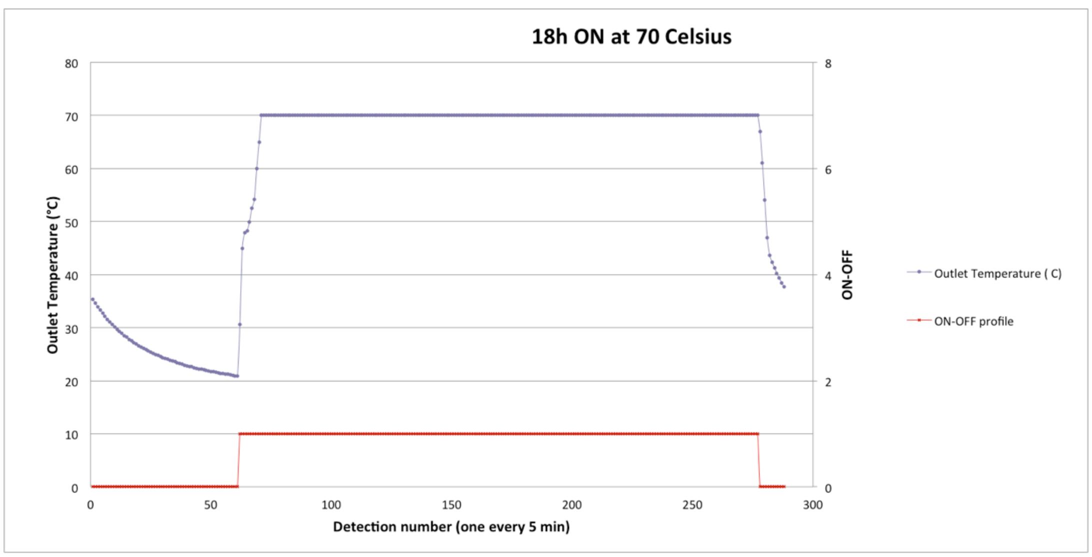


Figure 103: *18h ON* at 70°C daily outlet temperature distribution

7.2.3 *18h ON* (with double switch offs) Control Strategy

In this case the boiler is turned off at night between 12am and 4am and in the afternoon between 12pm and 2pm. The temperatures we considered as samples for this control strategy are:

7.2.3.1 Constant Temperature equal to 55°C when the boiler is turned on

Outlet temperature distribution:

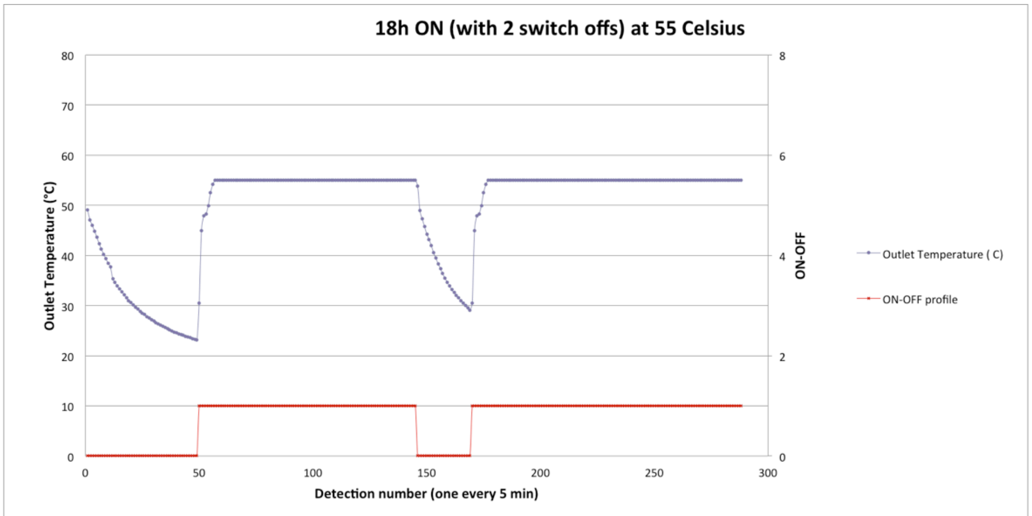


Figure 104: *18h ON 2SO* at 55°C daily outlet temperature distribution

7.2.3.2 Constant Temperature equal to 60°C when the boiler is turned on

Outlet temperature distribution:

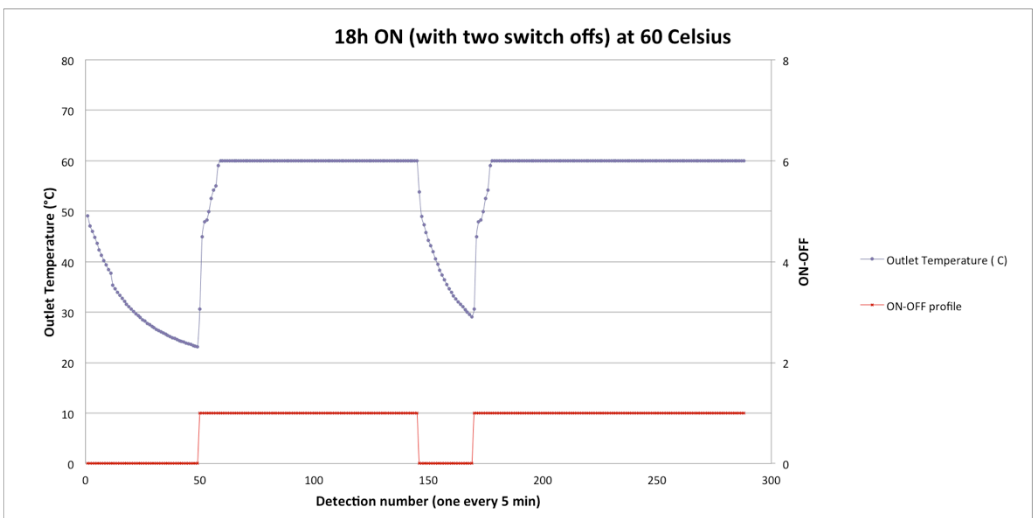


Figure 105: *18h ON 2SO* at 60°C daily outlet temperature distribution

7.2.3.3 Constant Temperature equal to 65°C when the boiler is turned on

Outlet temperature distribution:

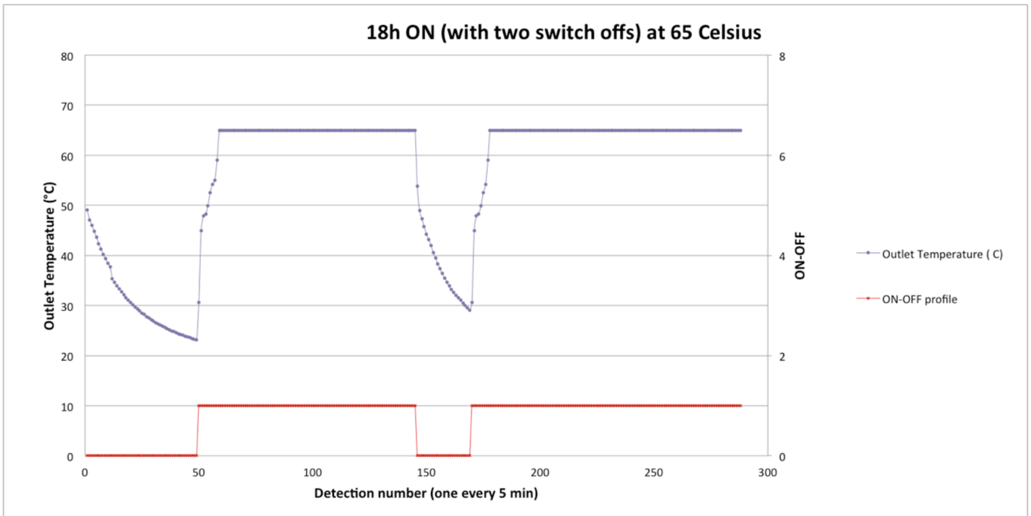


Figure 106: *18h ON 2SO* at 65°C daily outlet temperature distribution

<u>7.2.3.4 Constant Temperature equal to 70°C when the boiler is turned on</u>

Outlet temperature distribution:

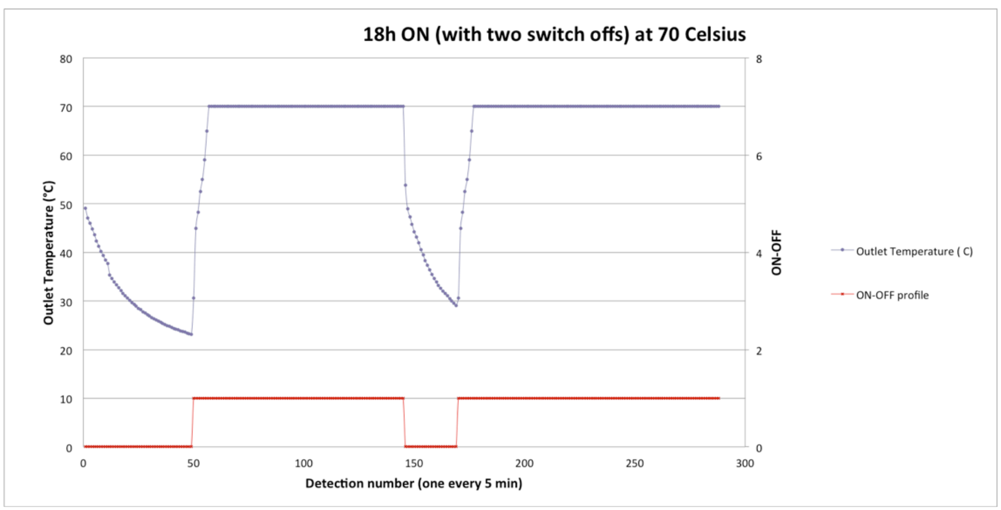


Figure 107: *18h ON 2SO* at 70°C daily outlet temperature distribution

7.2.4 *13h ON* (with double switch offs) Control Strategy

In this case the boiler is turned off at night between 10pm and 6am and in the afternoon between 2:30pm and 5:30pm. The temperatures we considered as samples for this control strategy are:

7.2.4.1 Constant Temperature equal to 60°C when the boiler is turned on

Outlet temperature distribution:

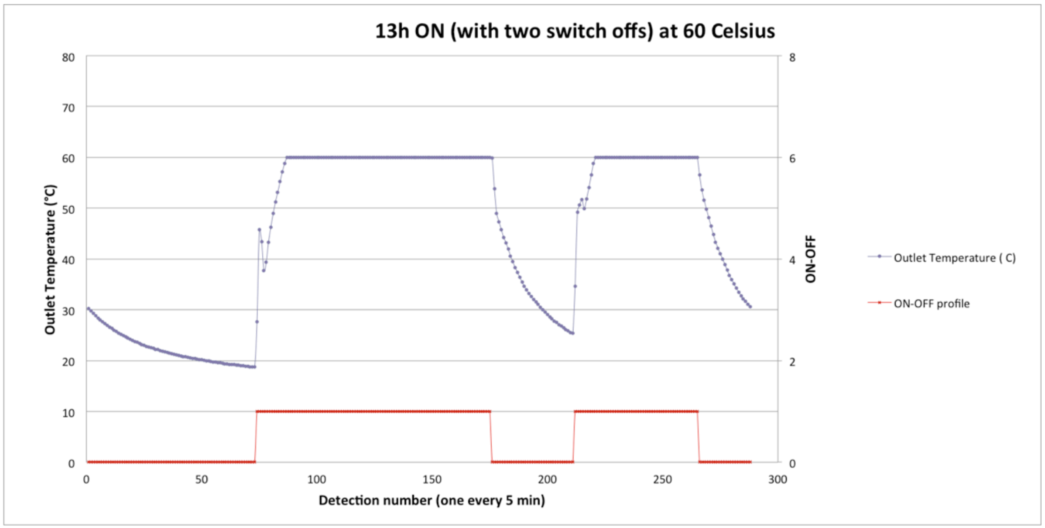


Figure 108: *13h ON* at 60°C daily outlet temperature distribution

7.2.4.2 Constant Temperature equal to 65°C when the boiler is turned on

Outlet temperature distribution:

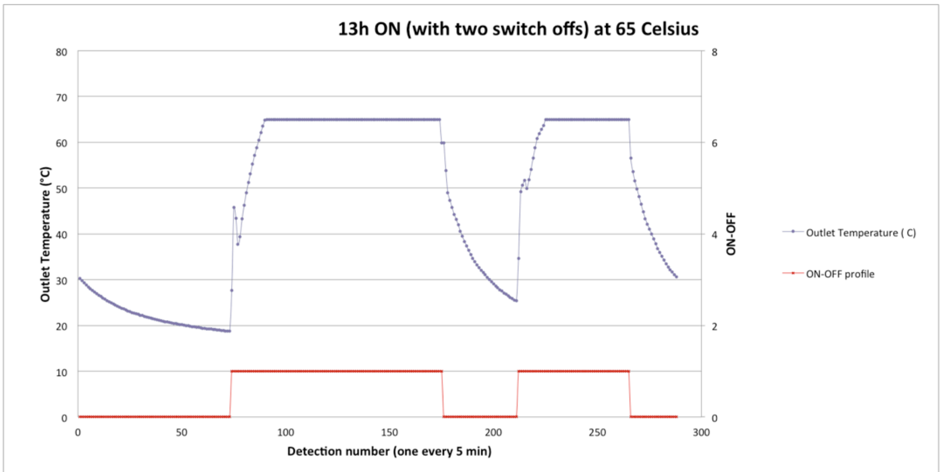


Figure 109: *13h ON* at 65°C daily outlet temperature distribution

7.2.4.3 Constant Temperature equal to 70°C when the boiler is turned on

Outlet temperature distribution:

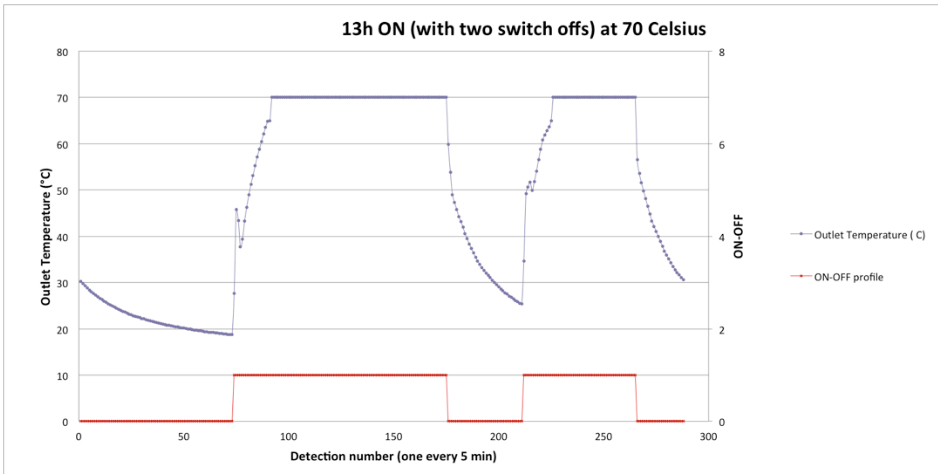


Figure 110: *13h ON* at 70°C daily outlet temperature distribution

7.2.4.4 Constant Temperature equal to 75°C when the boiler is turned on

Outlet temperature distribution:

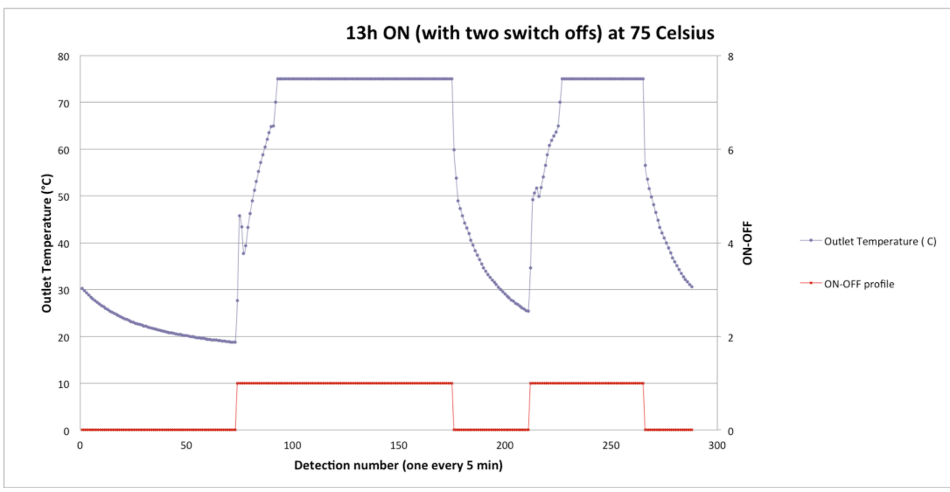


Figure 111: *13h ON* at 75°C daily outlet temperature distribution

7.2.5 *24h ON Bi-Climatic* Control Strategy

In this case the boiler is turned on 24h a day but the climatic curve changes: the outlet temperature is controlled with respect to the outdoor temperature in two different ways during the day – between 5am and 11pm- and at night - between 11pm and 5am. The temperatures we considered as samples for this control strategy are:

<u>7.2.5.1 Constant Temperature equal to 60°C during the day and equal to 50°C at night</u>

Outlet temperature distribution:

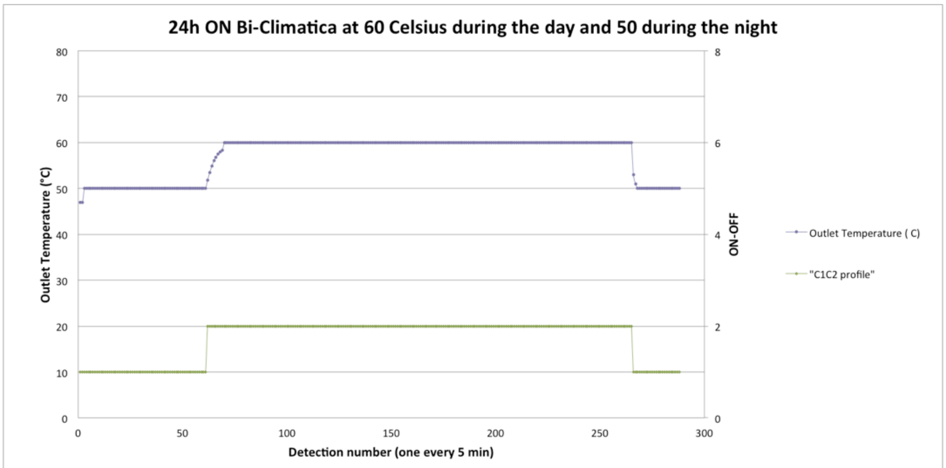


Figure 112: *24h ON Bi-Climatic* at 60-50°C daily outlet temperature distribution

<u>7.2.5.2 Constant Temperature equal to 65°C during the day and equal to 50°C at night</u>

Outlet temperature distribution:

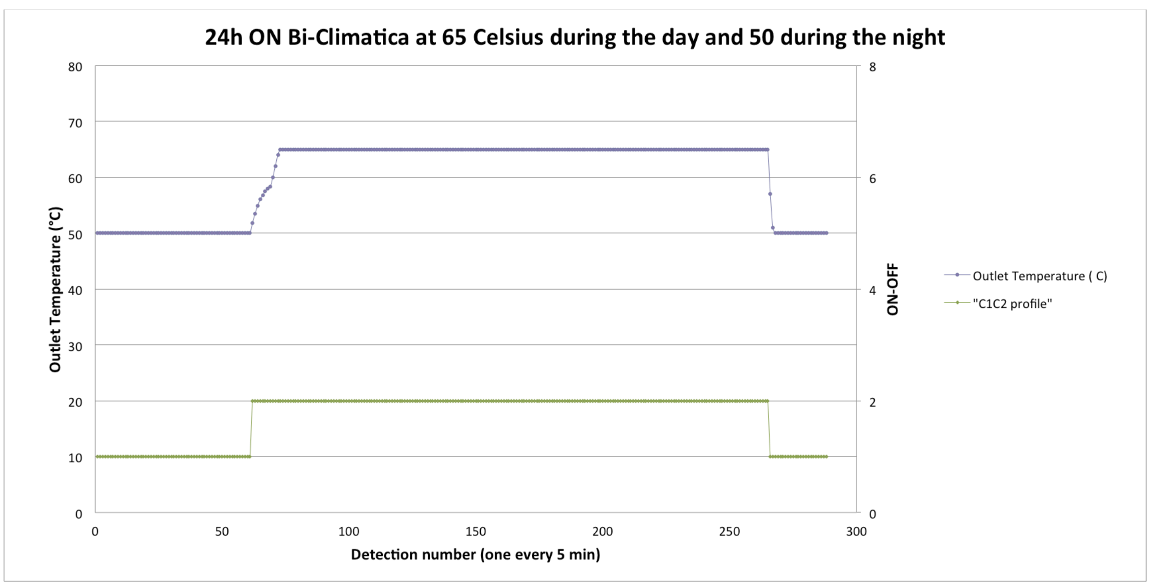


Figure 113: *24h ON Bi-Climatic* at 65-50°C daily outlet temperature distribution

7.2.5 *24h ON Real Mono-Climatic Curve* Control Strategy

For this last control strategy we didn't consider a pre-prepared Outlet Temperature Profile but we used the equation of one of the two climatic curves. Therefore the outlet temperature was directly correlated with the outdoor temperature. The equation relating the two temperatures is described by the equation below:

$$T_{outlet} = -1.5217 \cdot T_{outdoor} + 62.826$$

We obtained this relationship in Chapter 3 using the experimental data of the days when the boiler was regulating the outlet temperature with respect to the outdoor temperature.

7.3 The Program

In order to evaluate which control strategy is better for each sample day we wrote a script on MatLab (reported in Appendix B) that:

1. Imports the Outdoor Temperature profile of the sample day;
2. Sets the limits for the Indoor Temperature at 19°C and 21°C;
3. Loads the means of the data we used to evaluate the models;
4. Loads the Outlet Temperature Profile and the ON-OFF profile for each control strategy;
5. Iterates the daily data for an entire week to evaluate a possible relationship between the models used and the time. In this way we obtained a weekly profile where all the days had the same Outdoor temperatures and the same control strategy;
6. Computes the Indoor Temperatures and Gas consumption for each control strategy;

7. Plots the Indoor temperatures for each control strategy and displays the gas consumption values.

7.4 Results

Running the program for each sample day: As stated before, in order to analyze the possible influence of time, we created sample weeks (based on the sample days) with 7 days with the same outdoor temperature profile. Therefore all the graphs found in this paragraph relate to these sample weeks and not just to the sample days. For each of these sample weeks we obtained:

7.4.1 Outdoor Temperature < 0°C

7.4.1.1 12/17/2010 sample day for small temperature drops

Plotting the graphs for each control strategy we obtain:

1. *24h ON* control strategy:

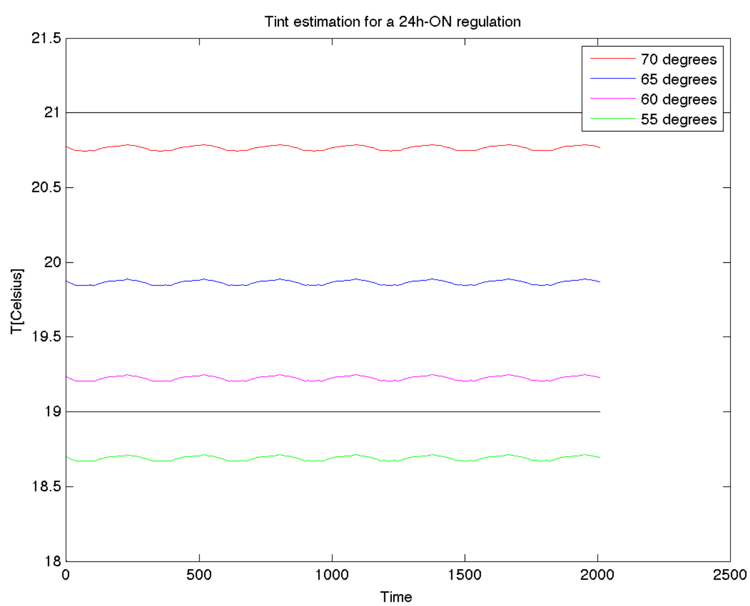


Figure 114: Indoor temperature we would have had during a week with a daily outdoor temperature profile equal to the one of 12/17/2010 if we had told the boiler to work 24h a day sending the water in outlet at various temperatures.

From this graph we can see that heating the building with a *24h ON at 55°C* control strategy is not sufficient to ensure a comfort temperature inside the apartments. On the other hand heating the building for 24 hours at 70°C creates an indoor temperature that is always over 20.5°C, which would cause wasteful gas consumption. Therefore when we evaluate the gas consumption we will consider only the *24h ON at 60 and 65°C* control strategies.

2. *18h ON* control strategy:

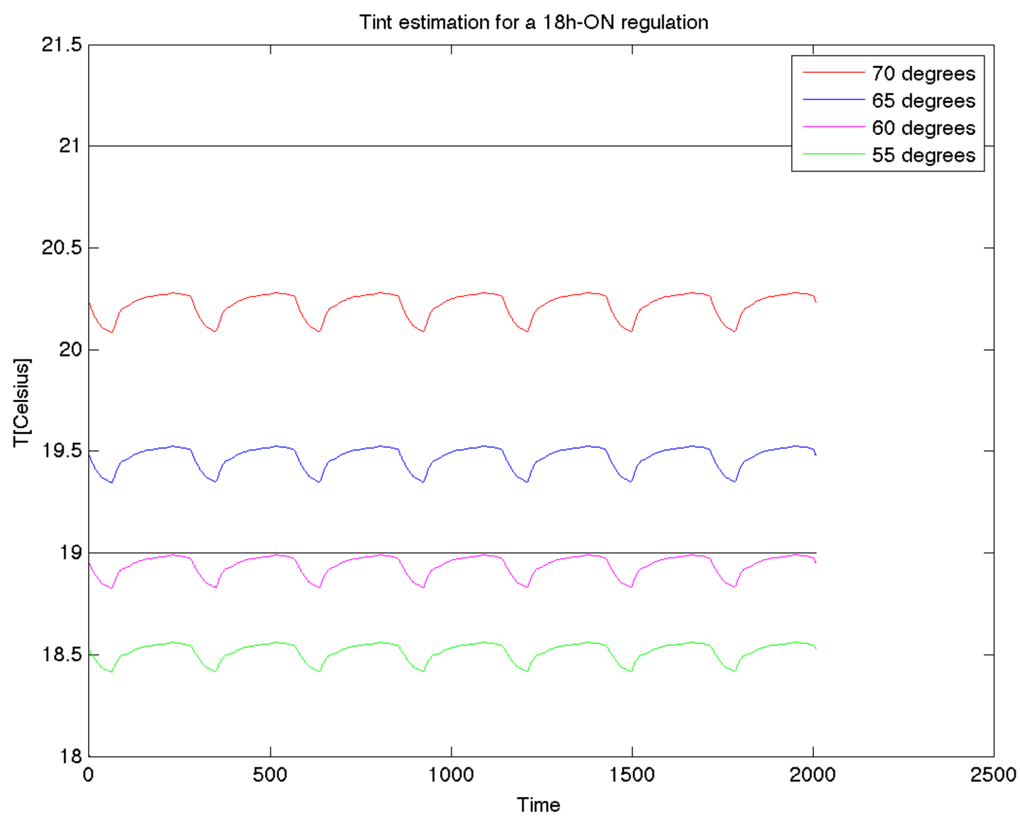


Figure 115: Indoor temperature we would have had during a week with a daily outdoor temperature profile equal to the one of 12/17/2010 if we had told the boiler to work 18h a day sending the water in outlet at various temperatures.

From this graph we can see that for equal outlet temperatures, heating the building with an *18h ON* control strategy causes lower indoor temperatures with respect to the temperatures attained using the *24h ON* control strategy. Furthermore regulating at 55 or 60°C isn't sufficient to ensure a comfort temperature inside the apartments. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 65 and 70°C* control strategies.

3. *18h ON with 2 switch-offs* control strategy:

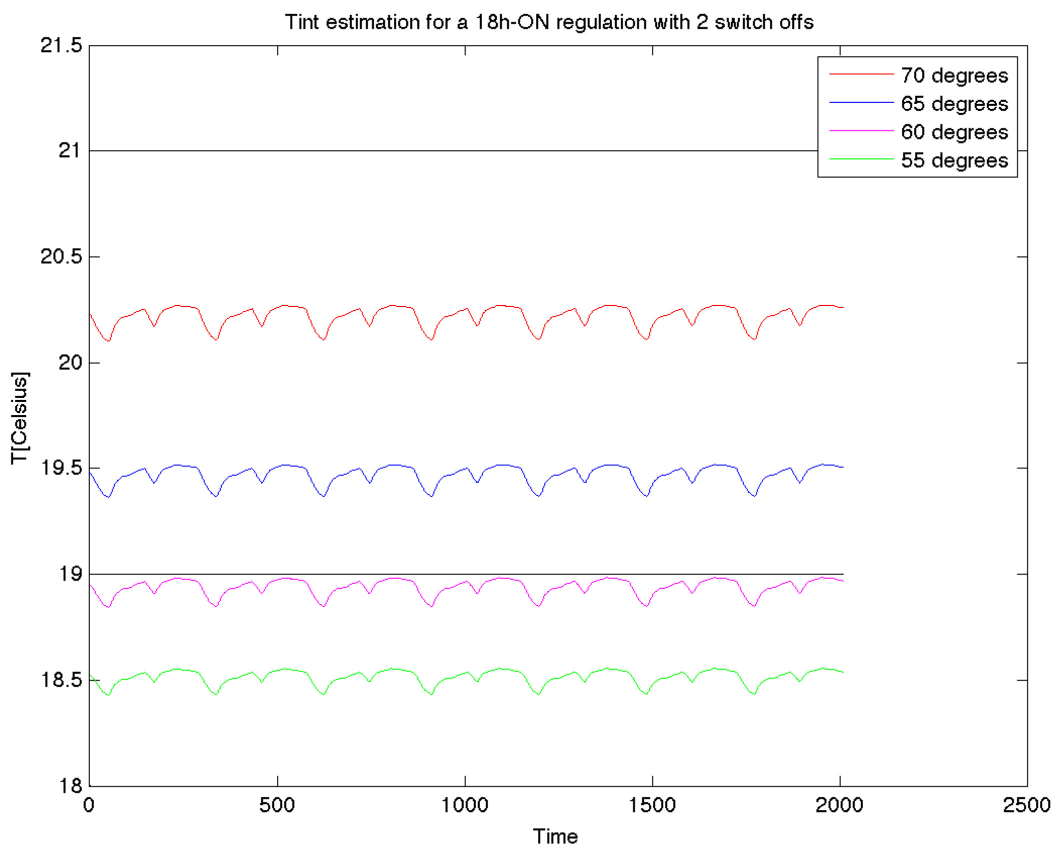


Figure 116: Indoor temperature we would have had during a week with a daily outdoor temperature profile equal to the one of 12/17/2010 if we had told the boiler to work 18h a day, with a double switch off, sending the water in outlet at various temperatures.

From this graph we can see that for equal outlet temperatures, heating the building with an *18h ON double switch-off* control strategy results in smaller indoor temperature drops during the day, with respect to the ones obtained using the *24h* ON control strategy. However, as before, regulating at 55 or 60°C isn't sufficient to ensure a comfort temperature inside the apartments. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 65 and 70°C* control strategies.

4. *13h ON* control strategy:

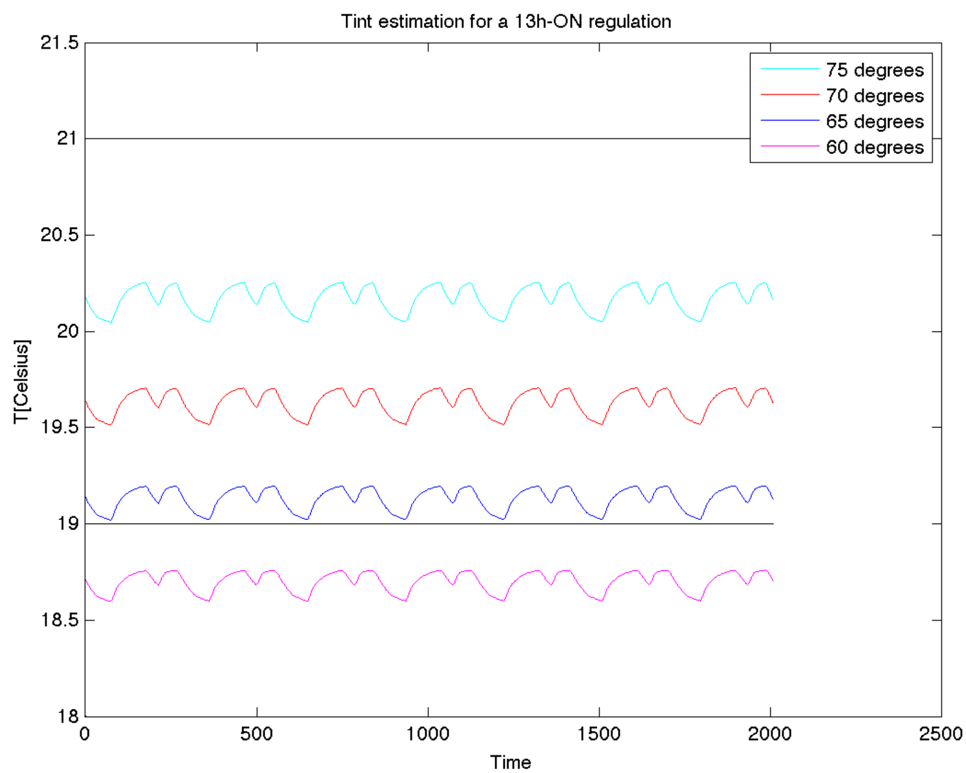


Figure 117: Indoor temperature we would have had during a week with a daily outdoor temperature profile equal to the one of 12/17/2010 if we had told the boiler to work 13h a day sending the water in outlet at various temperatures.

From this graph we can see that, for equal outlet temperatures, heating the building with a *13h ON* control strategy results in lower indoor temperature with respect to the ones attained using the *18h ON* control strategy. However this temperature drop is smaller than the one from the *24h ON* and *18h ON* control strategies, probably because the system is turned off during the warmest hours of the day (between 11am and 4pm). In this case we can see that regulating under 65°C isn't sufficient to ensure a comfort temperature inside the apartments. Therefore when we evaluate the gas consumption we will only consider the *13h ON* control strategies starting from 65°.

5. 24h ON Bi-Climatic control strategy:

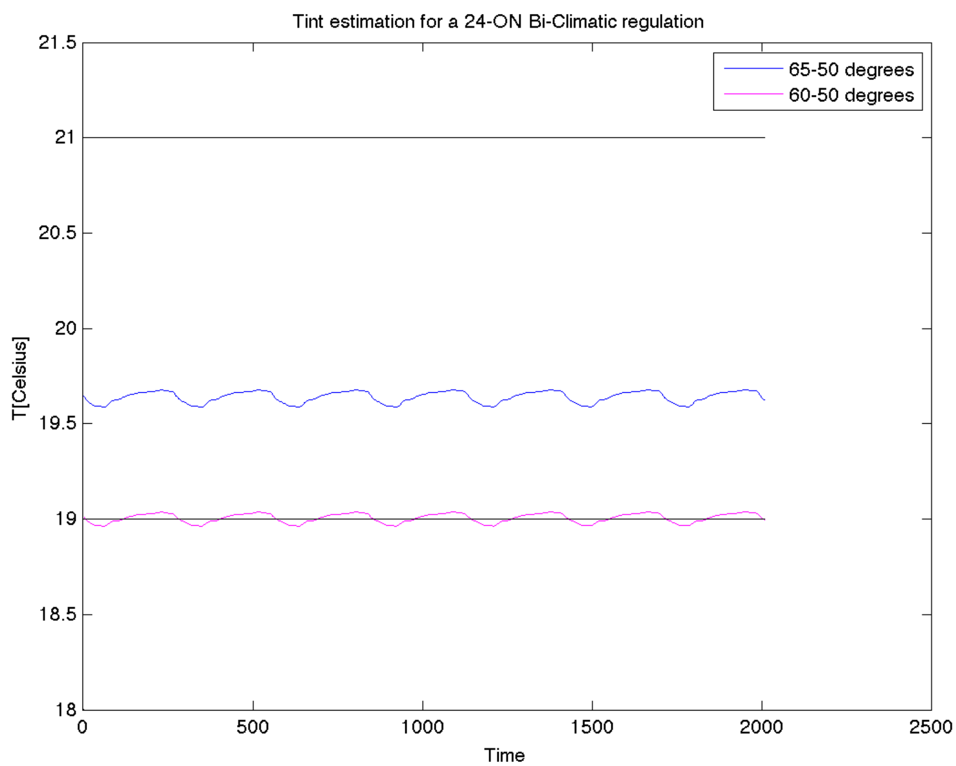


Figure 118: Indoor temperature we would have had during a week with a daily outdoor temperature profile equal to the one of 12/17/2010 if we had told the boiler to work 24h a day following a sort of Bi-Climatic line.

From this graph we can see that, for equal outlet temperatures, heating the building with a *24h ON Bi-climatic* control strategy results in lower indoor temperature with respect to the temperatures attained using the *24h ON* control strategy, but higher with respect to the *18h ON* control strategies. In this case both the sample strategies are able to ensure a minimum indoor temperature higher than 19°C (at least during the daytime). Therefore when we evaluate the gas consumption we will consider both the control strategies.

6. Indoor temperature as function of the Outdoor temperature control strategy:

   The function of the outdoor temperature for the 12/17/2010 was:

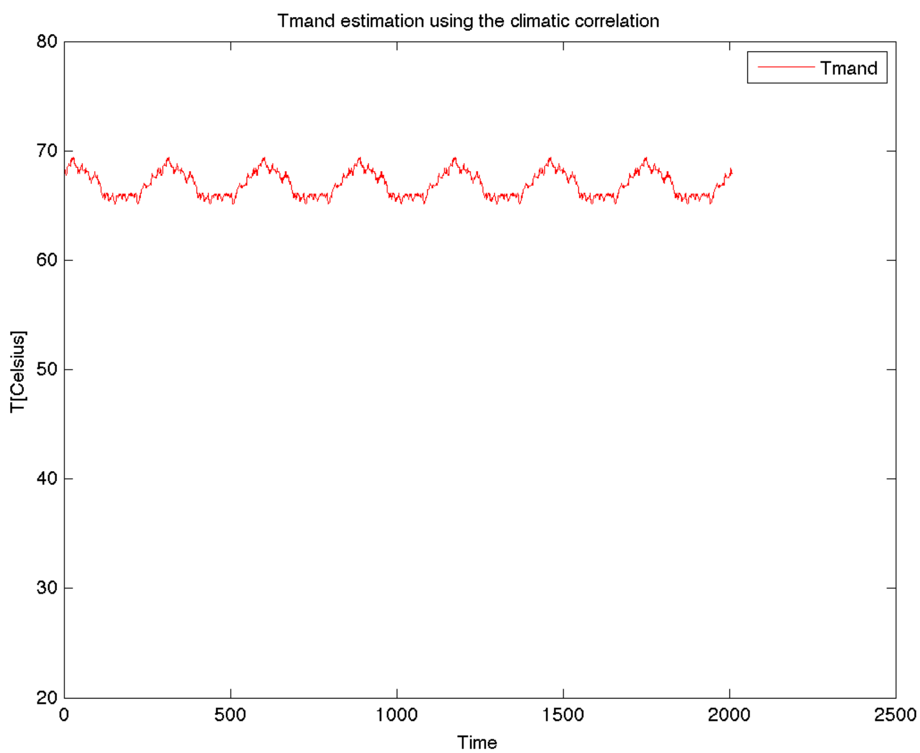


Figure 119: Outlet temperature for 12/17/2010 following the Climatic line

Using this outlet temperature as an input to compute the indoor temperature we obtained:

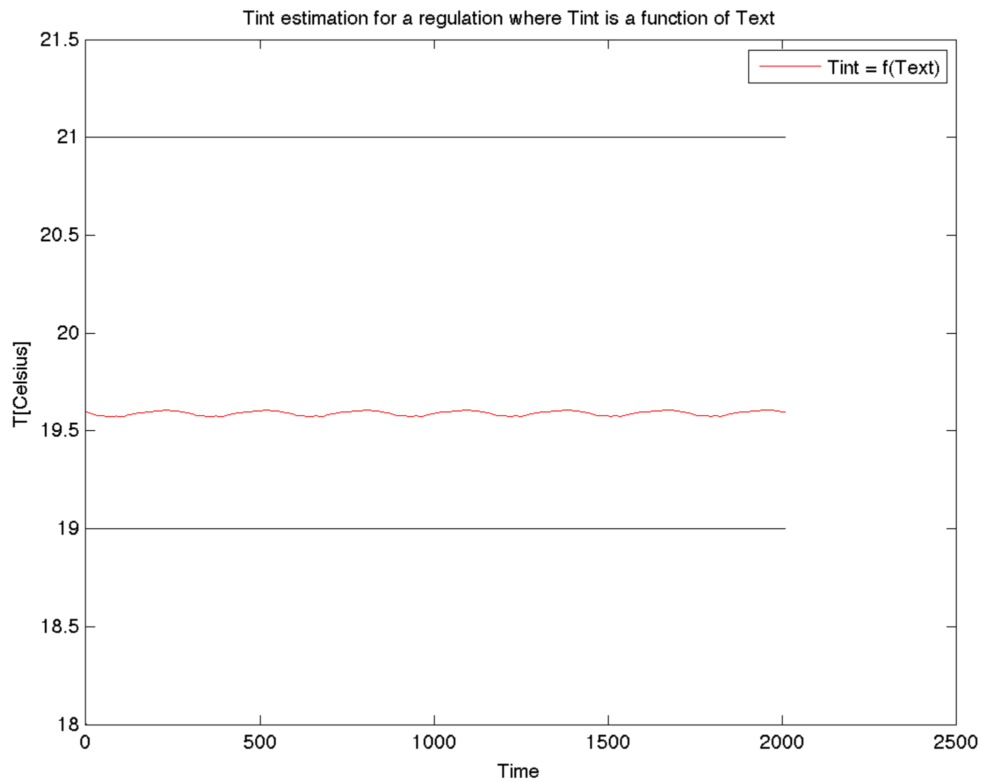


Figure 120: Indoor temperature we would have had during a week with a daily outdoor temperature profile equal to the one of 12/17/2010 if we had told the boiler to work 24h a day following the Mono-Climatic line just plotted.

From this graph we can see that the heating program relating the outlet temperature with the outdoor temperature is always able to ensure a minimum indoor temperature of higher than 19°C. Therefore when we evaluate the gas consumption we will consider this control strategy as one of the choices.

Gas consumption optimization

Once we were able to say which of the considered control strategies is able to ensure a minimum indoor temperature higher than 19°C we collected all the gas consumption estimations in order to see which of the selected strategies provided optimal results.

TABLE VIII: GAS CONSUMPTION EVALUATION FOR AN ENTIRE WEEK LIKE 12/17/10

Control Strategy	Temperature [°C]	T_{in}>19°C?	T_{in}<20.5°C?	Weekly Gas Consumption [m^3]
24h ON	55	NO		1123.650
24h ON	60			1253.742
24h ON	65			1368.571
24h ON	70		NO	1485.752
18h ON	55	NO		1218.879
18h ON	60	NO		1248.591
18h ON	65			1294.781
18h ON	70			1340.188
18h ON with 2 switch offs	55	NO		1204.437
18h ON with 2 switch offs	60	NO		1237.954
18h ON with 2 switch offs	65			1280.853
18h ON with 2 switch offs	70			1332.983
13h ON	60	NO		1208.754
13h ON	65			1249.963
13h ON	70			1292.971
13h ON	75			1364.383
24h ON Bi-Climatic	60d-50n			1187.947
24h ON Bi-Climatic	65d-50n			1248.631
24h ON Climatic Curve	= f (T_{out})			1391.191

This chart presents a lot of interesting information. As it can be seen, for equal outlet temperatures the gas consumption is not necessarily highest for the *24h ON* control strategy or lower for the *18h ON* heating program – or even lowest for the *13h ON* control strategy. In fact gas consumption for an outlet temperature of 55°C is lower for a *24h ON* control strategy than for a *18h ON* heating strategy. This is probably because when the outdoor temperatures are low, it costs more to heat the water when it is cold than to maintain a constant temperature.

Furthermore the chart shows that the *18h ON* controls with single and double switch-off consume more or less the same amount of gas. This is due to the small temperature drop during the day, which suggests a day with no sun. Finally we can see that the control that optimizes gas consumption (among the ones that are able to ensure a certain comfort indoor temperature) is the *Bi-Climatic control*. This result is not surprising as a *Bi-Climatic* control strategy heats the building 24 hours a day, and so doesn't entail any ignition consumption for heating up cold water. At the same time it requires lower outlet temperatures to heat the building, and we know that our boiler, being a condensation boiler, has a higher efficiency at lower temperatures.

7.4.1.2 12/18/2010 sample day for big temperature drops:

Plotting the graphs for each control strategy we obtain:

1. *24h ON* control strategy:

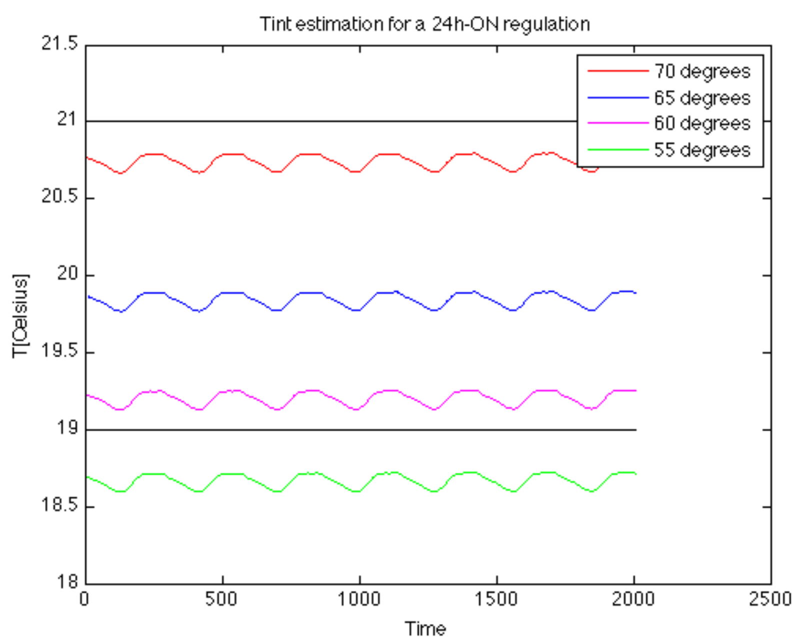


Figure 121: Indoor temperature on 12/18 for a 24h a day control

From this graph we can see similar results to those we noticed in the previous case: heating the building with a *24h ON at 55°C* control strategy is not sufficient, on the other hand heating the building for 24 hours at 70°C wastes gas. Therefore when we evaluate the gas consumption we will consider only the *24h ON at 60 and 65°C* control strategies.

2. *18h ON* control strategy:

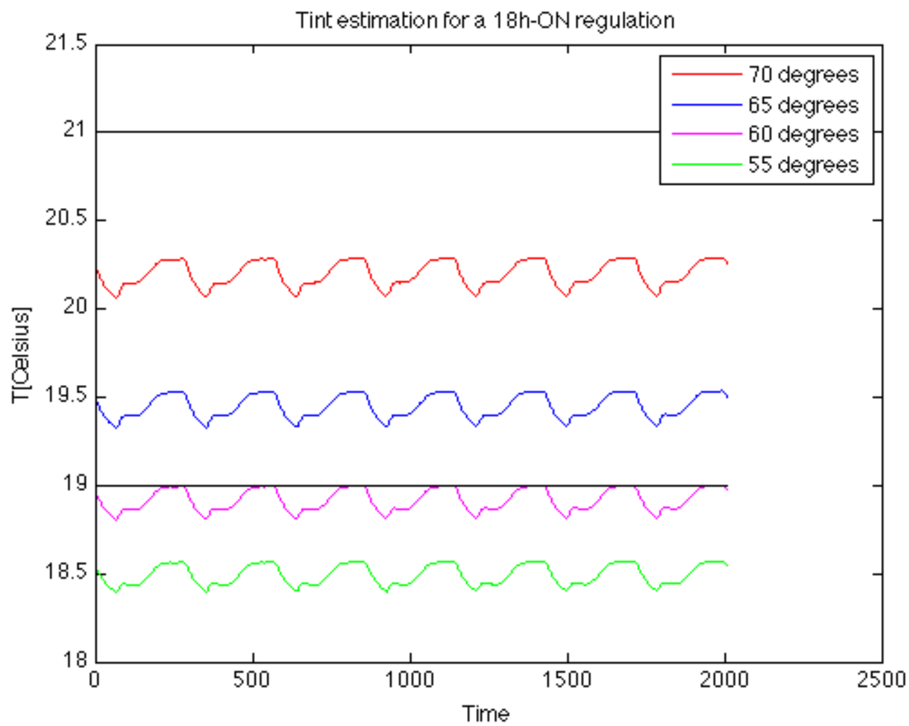


Figure 122: Indoor temperature on 12/18/2010 for an 18h ON control

As for the 12/17, regulating at 55 or 60°C isn't sufficient to ensure a comfort temperature inside the building. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 65 and 70°C* control strategies.

3. *18h ON (double switch-off)* control strategy:

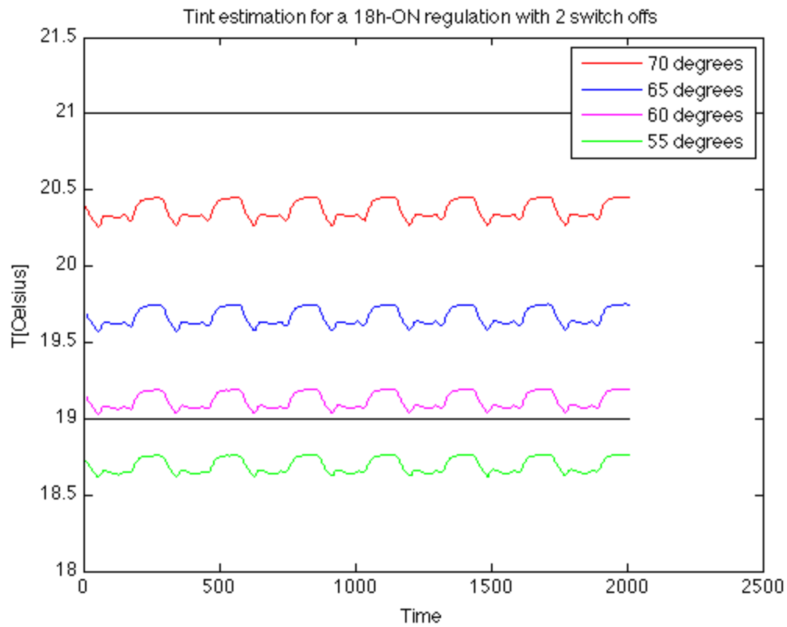


Figure 123: Indoor temperature on 12/18/2010 for an 18h a day, with a double switch off, control at various temperatures

Again we can notice how an *18h ON double switch-off* control strategy results in smaller indoor temperature drops during the day, with respect to the ones attained using the *24h ON* control strategy. What's new in this case is that the indoor temperatures are generally higher than those attained with the classic *18h ON* control strategy. This is caused by the more significant temperature spread between day and night for a similar daily mean of the outdoor temperatures. This spread also allows us to turn the system off when the weather is warmer (between 12pm and 2pm) and not just at night. Therefore in this case, regulating at 60°C is sufficient. Therefore when we evaluate the gas consumption we will consider all the *18h ON* control strategies with an outlet temperature equal to or higher than 60°C.

4. *13h ON* control strategy:

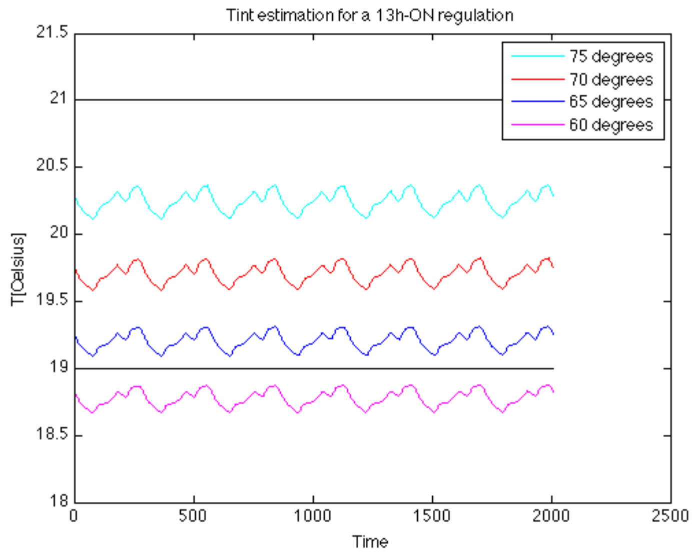


Figure 124: Indoor temperature on 12/18/2010 for a 13h ON control.

For this type of control we obtain more or less the same results as in the previous case study: regulating under 65°C isn't sufficient. Therefore when we evaluate the gas consumption we will only consider the *13h ON* control strategies starting from 65°C.

5. *24h ON Bi-Climatic* control strategy:

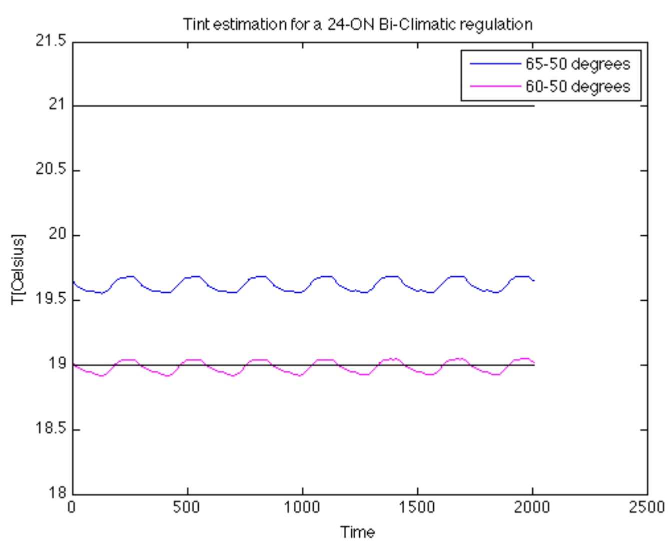


Figure 125: Indoor temperature on 12/18 for a 24h ON Bi-Climatic control

Again, in this case the results are more or less the same as in the previous case study. Therefore when we evaluate the gas consumption we will consider both the control strategies heating the water at 60°C during the day.

6. Indoor temperature as function of the outdoor temperature, control strategy:

The outlet temperature computed as a function of the outdoor temperature for the 12/18/2010 is:

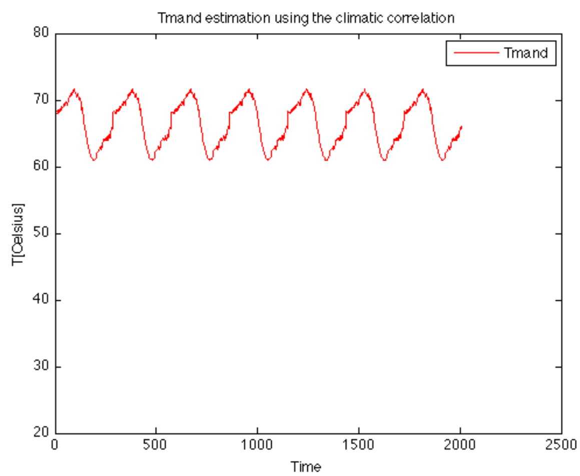


Figure 126: Outlet temperature for 12/18/2010 following the Climatic line

Using this outlet temperature as an input to compute the indoor temperature we obtained:

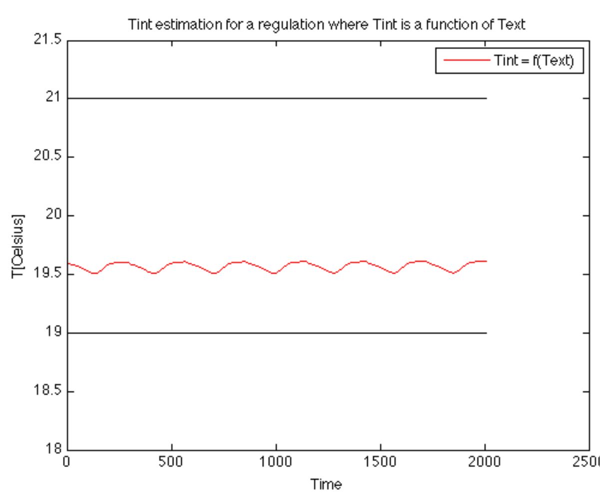


Figure 127: Indoor temperature on 12/18/2010 for a 24h a Mono-Climatic.

From this graph we can see that the heating strategy relating the outlet temperature with the outdoor temperature is always able to ensure a minimum indoor temperature higher than 19°C. Therefore when we evaluate the gas consumption we will consider this control strategy as one of the choices.

Gas consumption optimization

Once we were able to say which of the considered control strategies is able to ensure a minimum indoor temperature higher than 19°C, we collected all the gas consumption estimations in order to see which of the selected strategies optimizes consumption.

TABLE IX: GAS CONSUMPTION EVALUATION FOR AN ENTIRE WEEK LIKE 12/18/10

Control Strategy	Temperature [°C]	T_{in}>19°C?	T_{in}<20.5°C?	Weekly Gas Consumption [m^3]
24h ON	55	NO		1123.941
24h ON	60			1256.541
24h ON	65			1367.986
24h ON	70		NO	1484.183
18h ON	55	NO		1214.096
18h ON	60	NO		1247.214
18h ON	65			1295.523
18h ON	70			1342.486
18h ON with 2 switch offs	55	NO		1204.527
18h ON with 2 switch offs	60			1229.154
18h ON with 2 switch offs	65			1268.102
18h ON with 2 switch offs	70			1313.059
13h ON	60	NO		1206.120
13h ON	65			1249.431
13h ON	70			1292.057
13h ON	75			1363.886
24h ON Bi-Climatic	60d-50n			1184.819
24h ON Bi-Climatic	65d-50n			1248.943
24h ON Climatic Curve	= f (T_{out})			1390.112

This chart gives us more or less the same information we obtained from the previous case study. The Bi-Climatic control strategy again seems to be the one that best optimizes gas consumption. The only new information we can extract from this chart is that the gas consumption for the *18h ON double switch-off* control is now lower than the one obtained for the classic *18h ON* control.

7.4.2 Outdoor Temperature between 0°C and 3°C:

7.4.2.1 1/30/2011 sample day for small temperature drops:

Plotting the graphs for each control strategy we obtain:

1. *24h ON* control strategy:

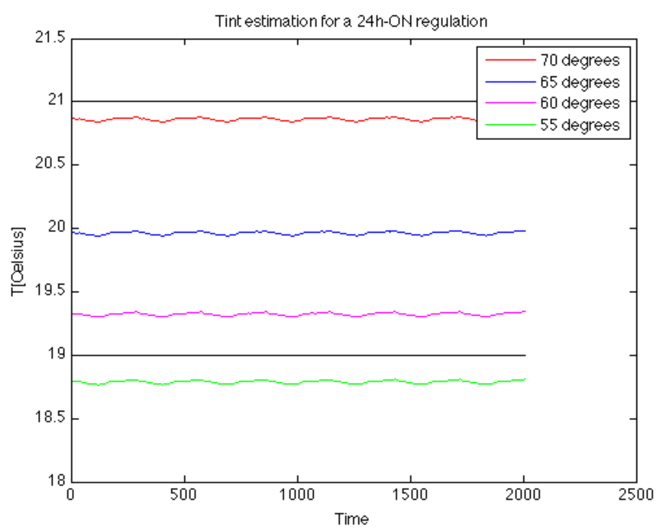


Figure 128: Indoor temperature on 1/30 for a 24h a day control

From this graph we can see that heating the building with a *24h ON at 55°C* control strategy is not sufficient. On the other hand heating the building for 24 hours at 70°C creates an indoor temperature always over 20.5°C, which could result in gas waste. Therefore when we evaluate the gas consumption we will only consider the *24h ON at 60 and 65°C* control strategies.

2. *18h ON* control strategy:

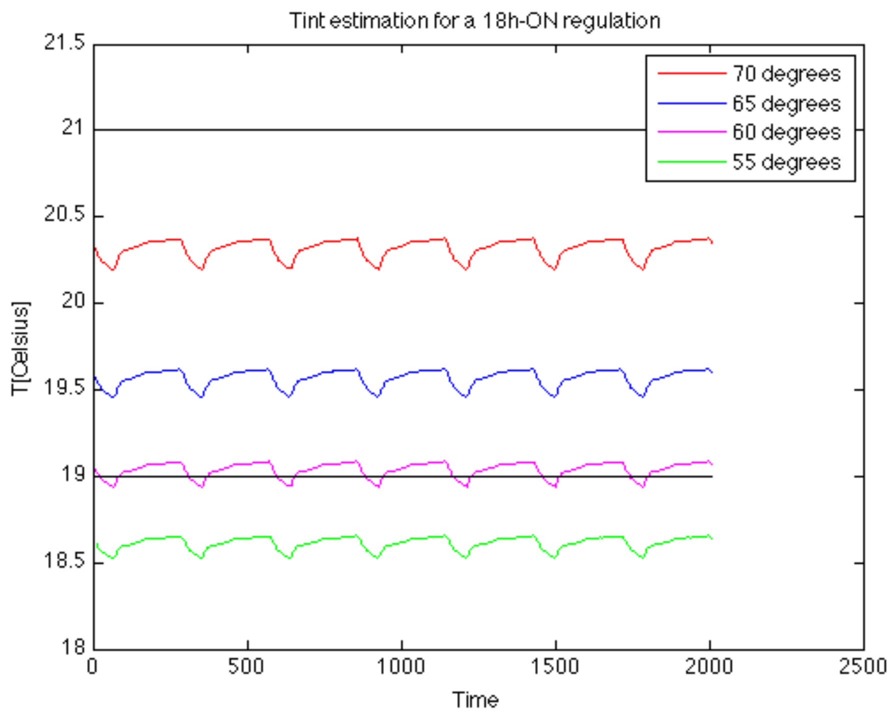


Figure 129: Indoor temperature on 1/30/2011 for an 18h ON control

From this graph we can see that, for equal outlet temperatures, heating the building with an *18h ON* control strategy results in lower indoor temperatureswith respect to the ones attained using the *24h ON* control strategy. Furthermore regulating at 55°C isn't sufficient to ensure a comfort temperature inside the apartments. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60, 65 and 70°C* control strategies.

3. *18h ON (double switch-off)* control strategy:

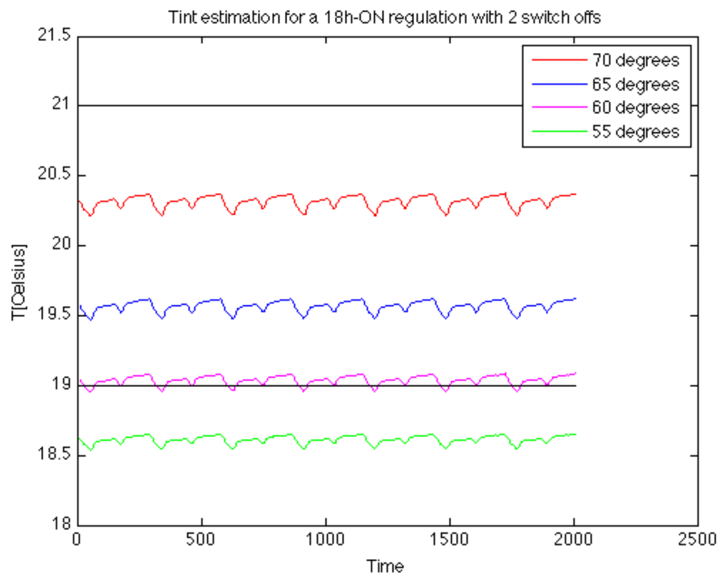


Figure 130: Indoor temperature on 1/30/2011 for an 18h a day, with a double switch off, control at various temperatures

From this graph we can see that, for equal outlet temperatures, heating the building with an *18h ON double switch-off* control strategy creates a more constant indoor temperature profile during the day. However, as before, regulating at 55 °C isn't sufficient.

4. *13h ON* control strategy:

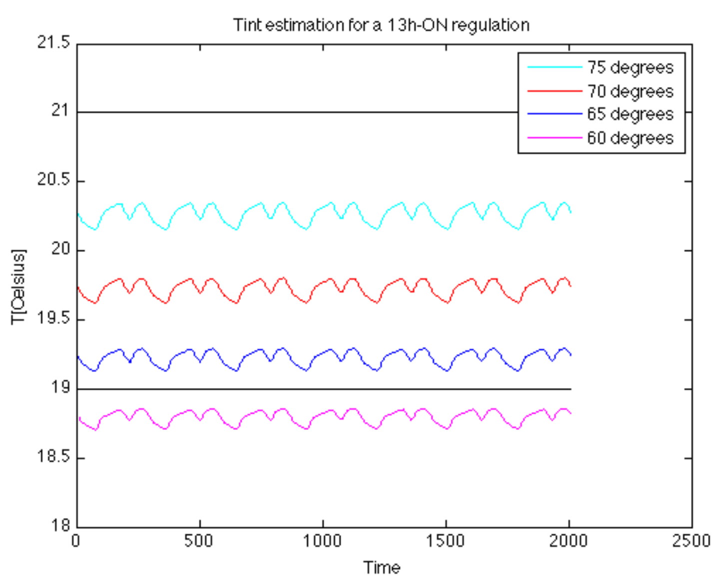


Figure 131: Indoor temperature on 1/30/2011 for a 13h ON control.

From this graph we can see that, for equal outlet temperatures, heating the building with a *13h ON* control strategy results in a lower indoor temperature with respect to those attained using the *18h ON* control strategy. However this temperature drop is smaller than the one that takes place passing from the *24h ON* and the *18h ON* control strategy. In this case we can see that regulating the outlet temperature at under 65°C isn't sufficient. Therefore when we evaluate the gas consumption we will only consider the *13h ON* control strategies starting from 65°C.

5. *24h ON Bi-Climatic* control strategy:

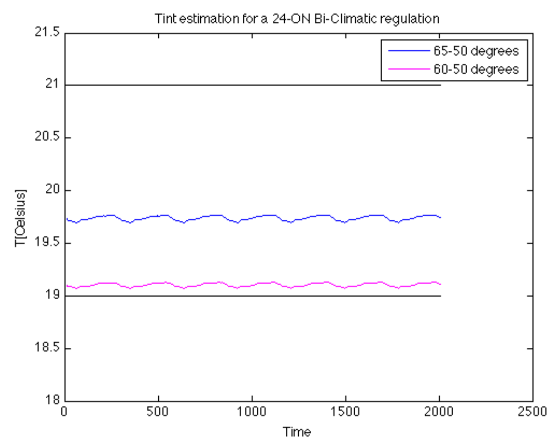


Figure 132: Indoor temperature on 1/30 for a 24h ON Bi-Climatic control

From this graph we can see that, for equal outlet temperatures, heating the building with a *24h ON Bi-Climatic* control strategy results in lower indoor temperatures than those attained using the *24h ON* control strategy, but higher ones with respect to the *18h ON* controls. In this case both the sample strategies are able to ensure a minimum indoor temperature of higher than 19°C (at least during the daytime). Therefore when we evaluate the gas consumption we will consider both of them.

6. *Indoor temperature as function of the Outdoor temperature* control strategy:

The outlet temperature computed as a function of the outdoor temperature for the 01/30/2011 is:

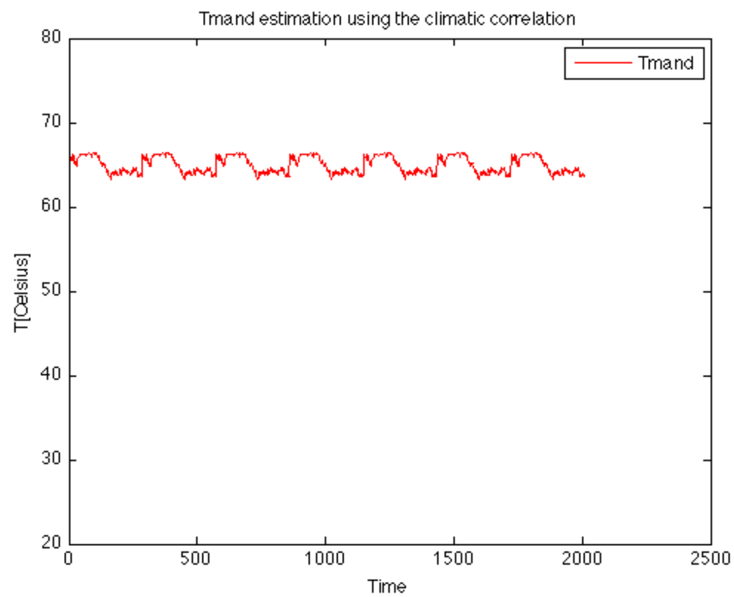


Figure 133: Outlet temperature for 12/18/2011 following the Climatic line

Using this outlet temperature as an input to compute the indoor temperature we obtained:

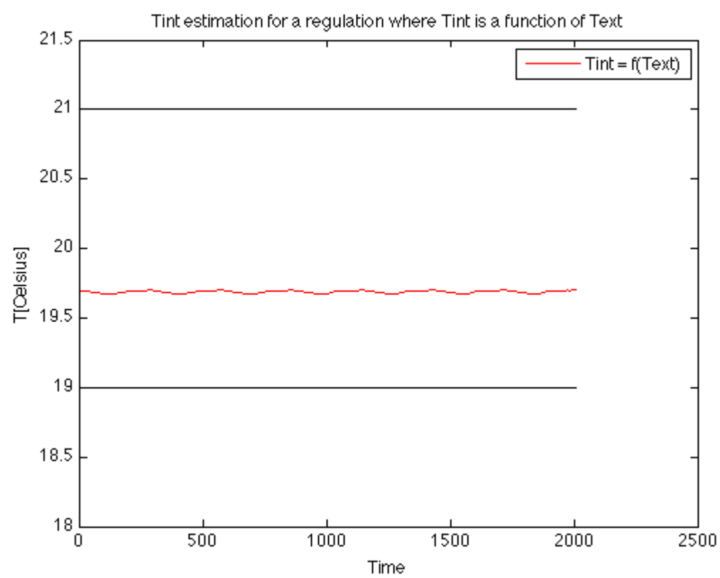


Figure 134: Indoor temperature on 1/30/2011 for a 24h a Mono-Climatic.

From this graph we can see that the control relating the outlet temperature with the outdoor temperature is always able to ensure a minimum indoor temperature of higher than 19°C. Therefore when we evaluate the gas consumption we will consider this control strategy as one of the possible choices.

Gas consumption optimization

Once we were able to say which of the considered control strategies was able to ensure a minimum indoor temperature of higher than 19°C, we collected all the gas consumption estimations in order to see which of the selected strategies optimizes consumption.

TABLE X: GAS CONSUMPTION EVALUATION FOR AN ENTIRE WEEK LIKE 1/30/11

Control Strategy	Temperature [°C]	T_{in}>19°C?	T_{in}<20.5°C?	Weekly Gas Consumption [m^3]
24h ON	55	NO		999.957
24h ON	60			1116.541
24h ON	65			1228.609
24h ON	70		NO	1344.183
18h ON	55	NO		1034.096
18h ON	60			1091.738
18h ON	65			1133.528
18h ON	70			1203.154
18h ON with 2 switch offs	55	NO		1074.587
18h ON with 2 switch offs	60			1091.931
18h ON with 2 switch offs	65			1134.298
18h ON with 2 switch offs	70			1191.236
13h ON	60	NO		991.128
13h ON	65			1109.431
13h ON	70			1153.376
13h ON	75			1223.266
24h ON Bi-Climatic	60d-50n			1094.675
24h ON Bi-Climatic	65d-50n			1138.943
24h ON Climatic Curve	= f (T_{out})			1250.412

As we can see in this case the control strategy that optimizes the gas consumption (among the ones that are able to ensure a certain comfort temperature inside the building) is the *18h ON* one (both with one or two switch-offs). This result was to be expected since in this case the temperatures are higher and so we don't need a *24h ON* control anymore to ensure a certain indoor temperature with an outlet temperature of 60°C. Furthermore, from this table we can see how gas consumption generally decreases for the same outlet temperature in respect to the previous cases. This happens because with a higher outdoor temperature the system needs less power to heat the water up to a certain temperature.

7.4.2.2 1/24/2011 sample day for big temperature drops:

Plotting the graphs for each control strategy we obtain:

1. *24h ON* control strategy:

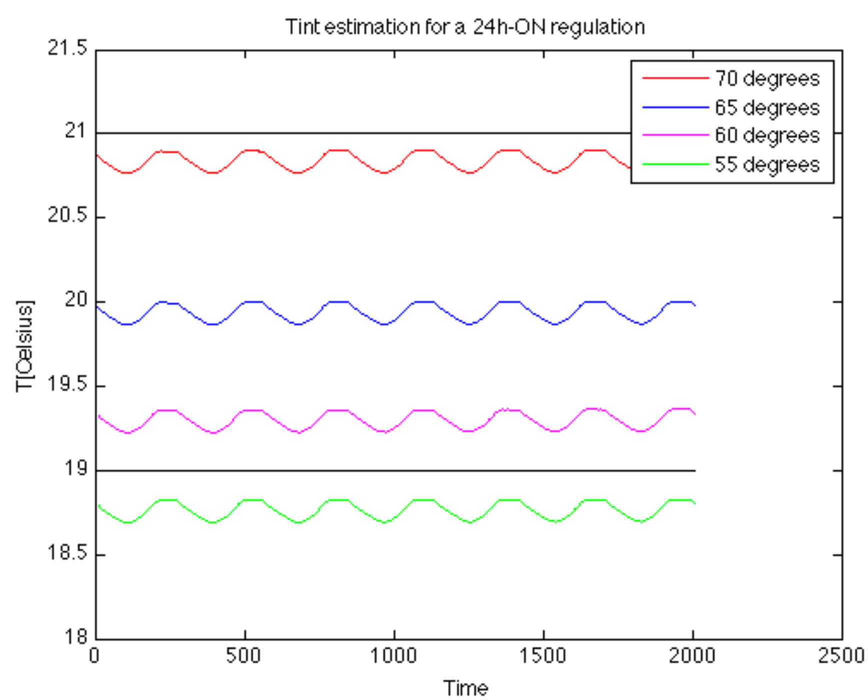


Figure 135: Indoor temperature on 1/24 for a 24h a day control

As we can see from the graph a bigger drop in the outdoor temperature causes the indoor temperature to fluctuate more around its mean value. However the results are the same as in the previous case: a *24h ON at 55°C* control strategy is not sufficient. On the other hand heating the building for 24 hours at 70°C can waste some gas. Therefore when we evaluate the gas consumption we will only consider the *24h ON at 60 and 65°C* control strategies.

2. *18h ON* control strategy:

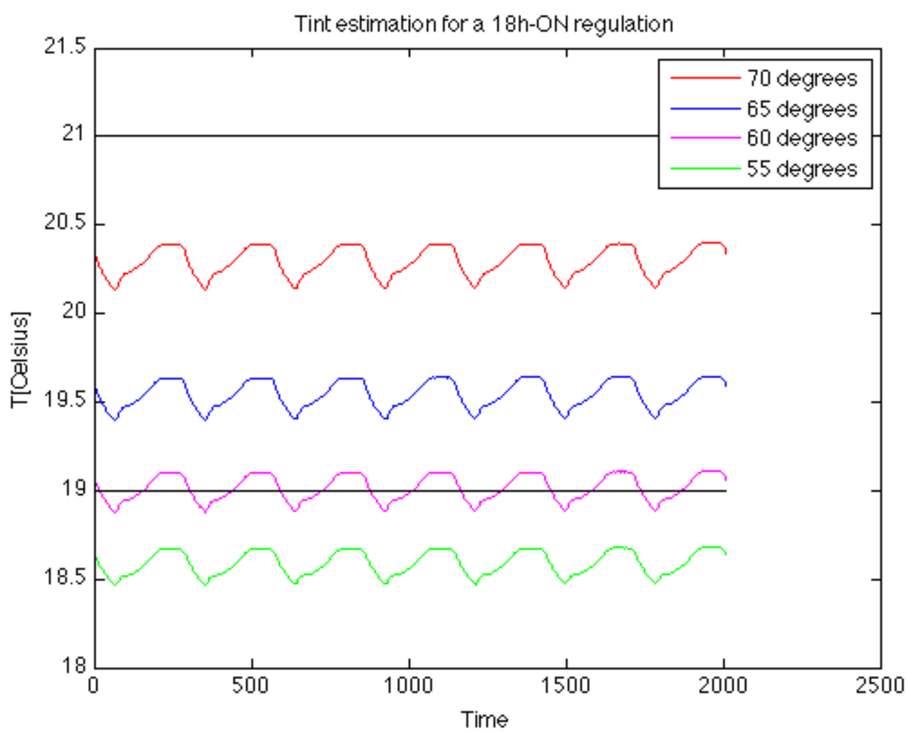


Figure 136: Indoor temperature on 1/24/2011 for an 18h ON control

As we can see the results are equal to the ones we obtained for the previous case: regulating at 55°C isn't sufficient. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60, 65 and 70°C* control strategies.

3. *18h ON (double switch-off)* control strategy:

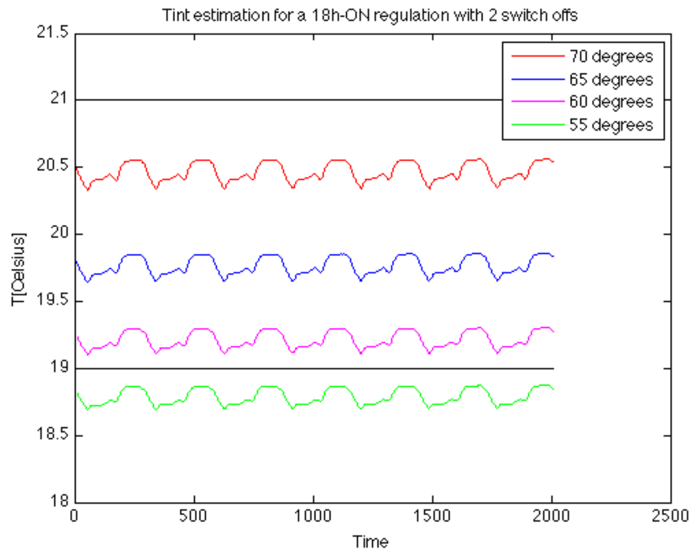


Figure 137 : Indoor temperature on 1/24/2011 for an 18h a day, with a double switch off, control at various temperatures

As before regulating at 55°C isn't sufficient to ensure a comfort temperature inside the building. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60, 65 and 70°C* control strategies.

4. *13h ON* control strategy:

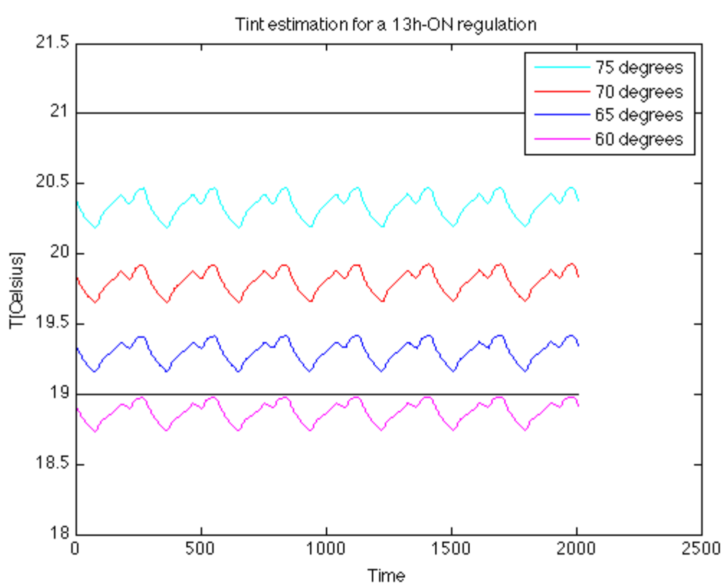


Figure 138: Indoor temperature on 1/24/2011 for a 13h ON control.

From this graph we can see that regulating under 60°C isn't sufficient. Therefore when we evaluate the gas consumption we will only consider the *13h ON* control strategies starting from 60°C.

5. *24h ON Bi-Climatic* control strategy:

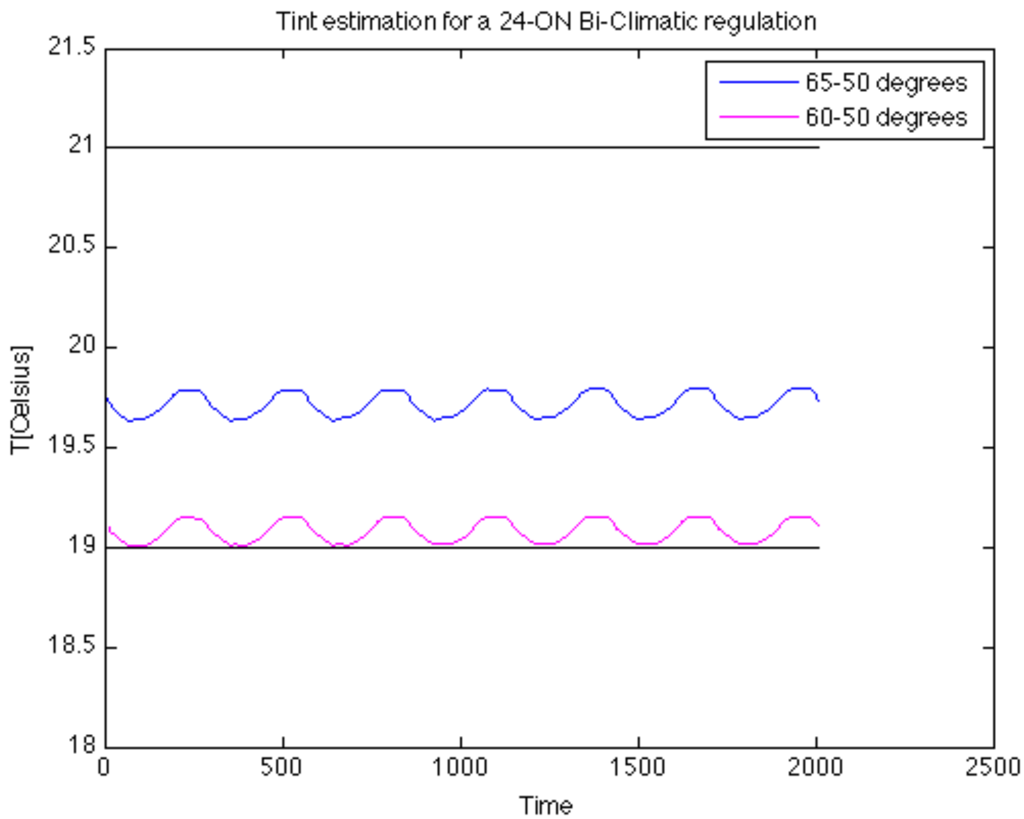


Figure 139: Indoor temperature on 1/24 for a 24h ON Bi-Climatic control

From this graph we can see that both the curves are completely inside the tolerance limits of the indoor temperature. Therefore when we evaluate the gas consumption we will consider both the control strategies.

6. *Indoor temperature as a function of the Outdoor temperature* control strategy:

The outlet temperature computed as a function of the outdoor temperature for the 01/24/2011 is:

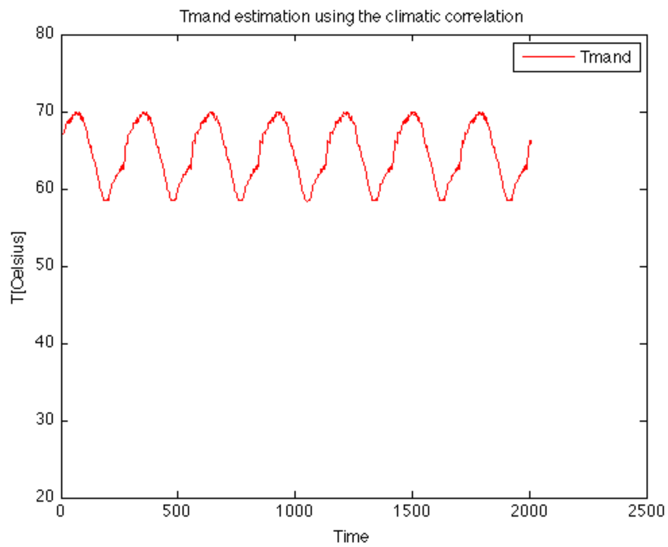


Figure 140: Outlet temperature for 1/24/2011 following the Climatic line

Using this outlet temperature as an input to compute the indoor temperature we obtained:

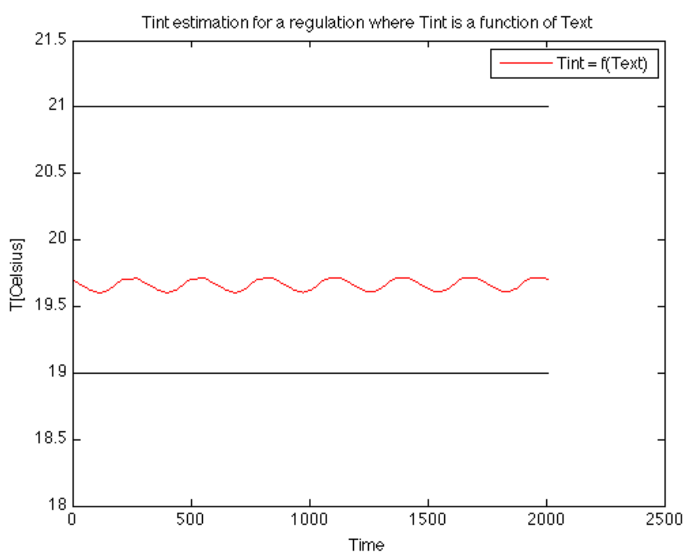


Figure 141: Indoor temperature on 1/24/2011 for a 24h a Mono-Climatic.

From this graph we can see that the control relating the outlet temperature with the outdoor one is always able to ensure a minimum indoor temperature of higher than 19°C.

Therefore when we evaluate the gas consumption we will consider this control strategy as one of the possible choices.

Gas consumption optimization

Once we were able to say which of the considered control strategies was able to ensure a minimum indoor temperature higher than 19°C, we collected all the gas consumption estimations in order to see which of the selected strategies optimized consumption.

TABLE XI: GAS CONSUMPTION EVALUATION FOR AN ENTIRE WEEK LIKE 1/24/11

Control Strategy	Temperature [°C]	T_{in}>19°C?	T_{in}<20.5°C?	Weekly Gas Consumption [m^3]
24h ON	55	NO		998.248
24h ON	60			1119.340
24h ON	65			1228.024
24h ON	70		NO	1342.614
18h ON	55	NO		1032.313
18h ON	60			1086.397
18h ON	65			1131.293
18h ON	70			1205.452
18h ON with 2 switch offs	55	NO		1024.677
18h ON with 2 switch offs	60			1032.386
18h ON with 2 switch offs	65			1103.915
18h ON with 2 switch offs	70			1149.312
13h ON	60	NO		993.494
13h ON	65			1089.923
13h ON	70			1153.278
13h ON	75			1222.769
24h ON Bi-Climatic	60d-50n			1091.547
24h ON Bi-Climatic	65d-50n			1139.255
24h ON Climatic Curve	= f (T_{out})			1249.333

This chart gives us more or less the same information we obtained from the previous case. The *18h ON* control is still the one that optimizes gas consumption but, because of the bigger temperature difference between day and night, the one with 2 switch-offs is the one that minimizes gas consumption. This result was to be expected as turning the heating strategy off while the temperature is higher results in smaller drops in the indoor temperature and so, it follows, in the outlet temperature.

7.4.3 Outdoor Temperature between 3°C and 7°C:

7.4.3.1 1/10/2011 sample day for low temperature drops:

Plotting the graphs for each control strategy we obtain:

1. *24h ON* control strategy:

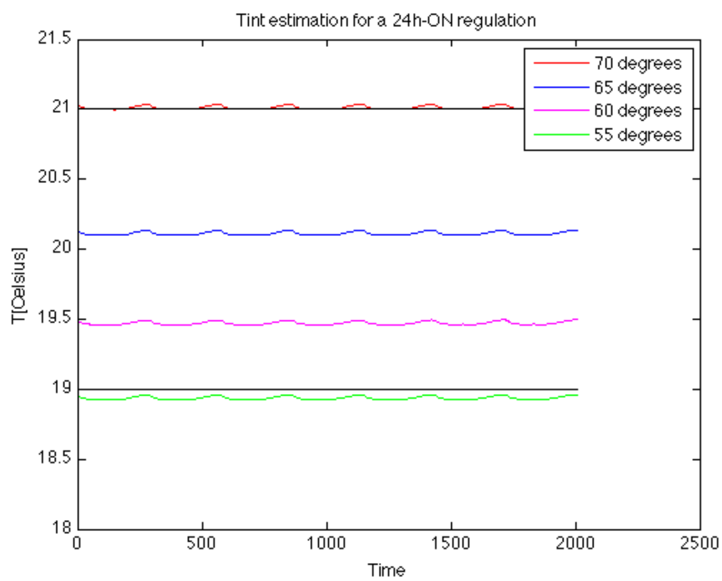


Figure 142: Indoor temperature on 1/10 for a 24h a day control

From this graph we can see how the indoor temperature still grew from the previous cases. In this case we can see that a *24h ON at 55°C* control strategy is not sufficient. On the other hand heating the building for 24 hours at 70°C creates an indoor temperature that's always around 21°C, which would cause gas waste. Therefore when we evaluate the gas consumption we will only consider the *24h ON at 60 and 65°C* control strategies.

2. *18h ON* control strategy:

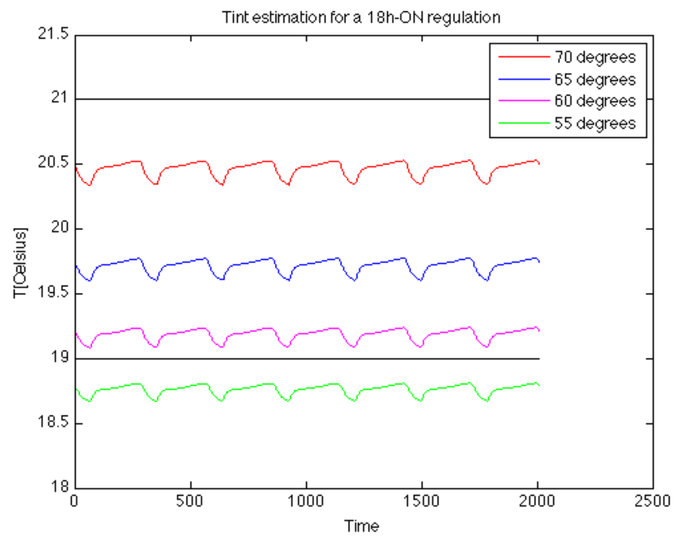


Figure 143: Indoor temperature on 1/10/2011 for an 18h ON control

As for the previous cases the only outlet temperature that doesn't ensure a certain comfort inside the building is 55°C. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 55, 65 and 70°C* control strategies.

3. *18h ON* (*double switch-off*) control strategy:

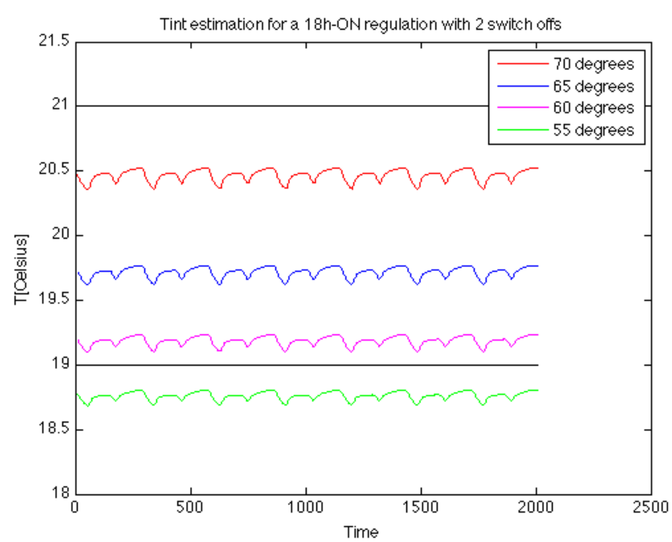


Figure 144: Indoor temperature on 1/10/2011 for an 18h a day, with a double switch off, control at various temperatures

As for the previous control strategy sending the water at 55°C isn't sufficient to ensure a comfort temperature inside the apartments. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60, 65 and 70°C* control strategies.

4. *13h ON* control strategy:

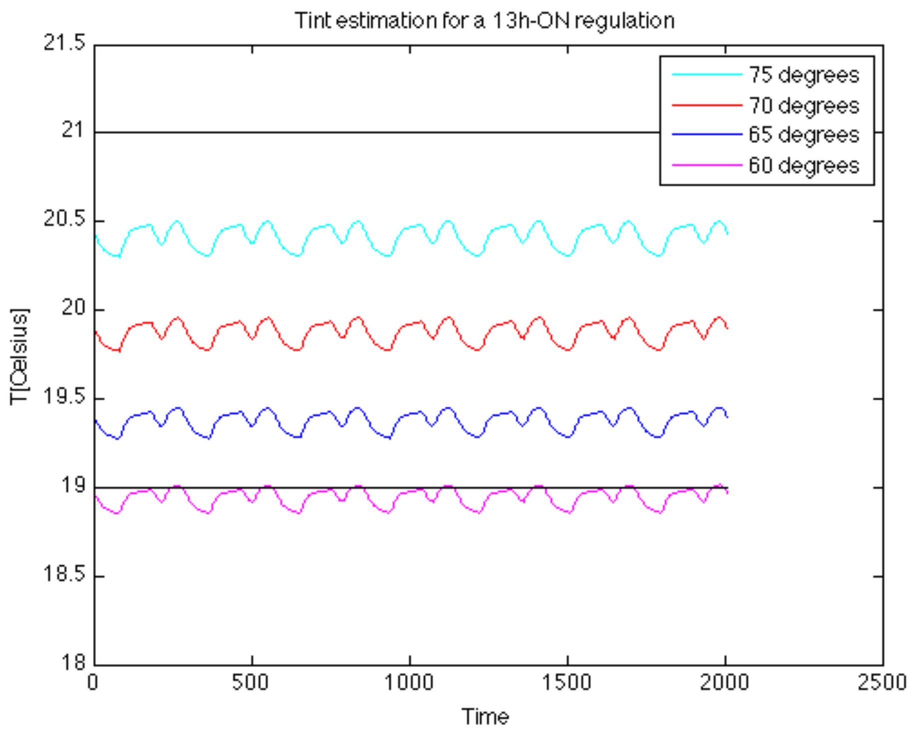


Figure 145: Indoor temperature on 1/10/2011 for a 13h ON control.

From this graph we can see that the *13h ON at 60°C* control is still not able to ensure a temperature inside the building of higher than 19°C. Therefore when we evaluate the gas consumption we will only consider the *13h ON* control strategies starting from 65°.

5. *24h ON Bi-Climatic* control strategy:

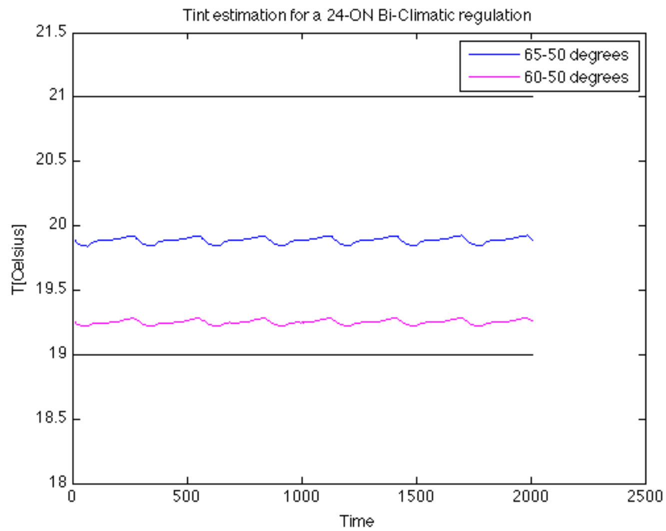


Figure 146: Indoor temperature on 1/10 for a 24h ON Bi-Climatic control

From this graph we can see that both the sample strategies are able to ensure a minimum indoor temperature of higher than 19°C. Therefore when we evaluate the gas consumption we will consider both of them.

6. *Indoor temperature as a function of the Outdoor temperature* control strategy:

The outlet temperature computed as a function of the outdoor temperature for 01/10/2011 is:

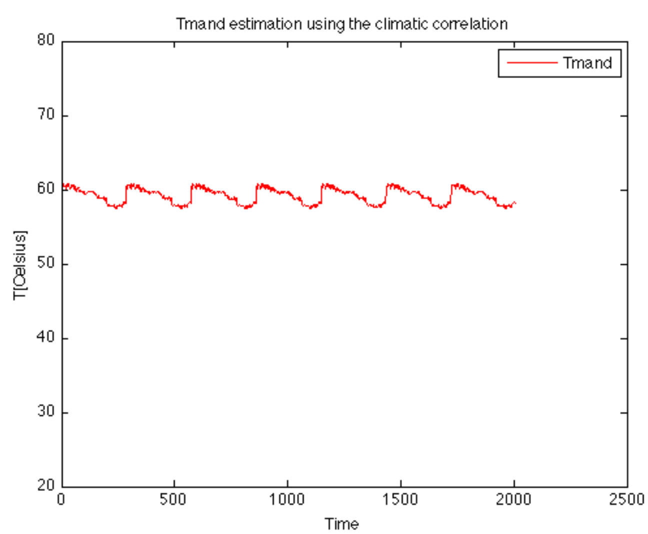


Figure 147: Outlet temperature for 1/10/2011 following the Climatic line

Using this outlet temperature as an input to compute the indoor temperature we obtained:

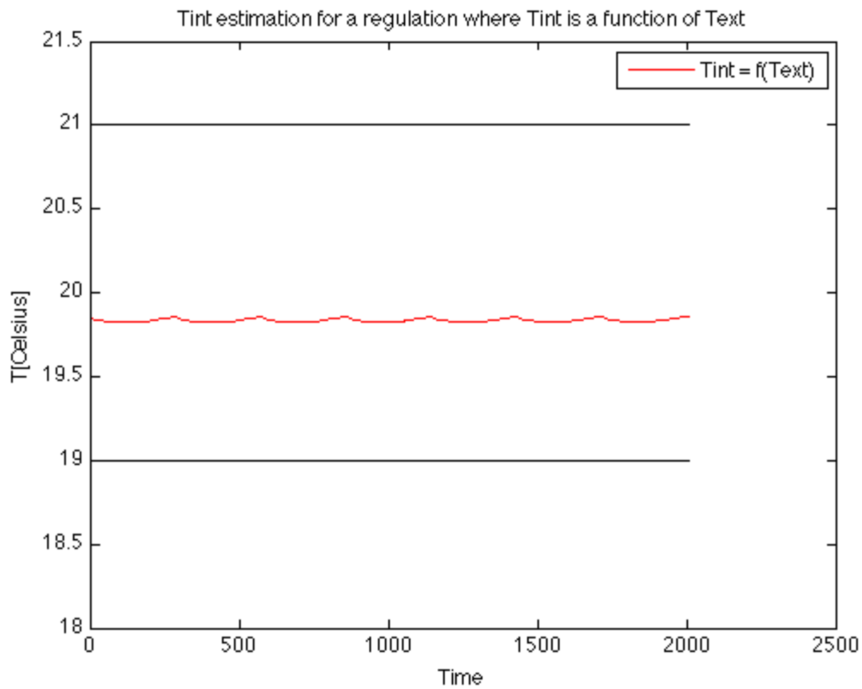


Figure 148: Indoor temperature on 1/10/2011 for a 24h a Mono-Climatic.

From this graph we can see that the control relating the Outlet temperature with the Outdoor one is always able to ensure a minimum indoor temperature of higher than 19°C.

Therefore when we evaluate the gas consumption we will consider this control strategy as one of the possible choices.

Gas consumption optimization

Once we were able to say which of the considered control strategies was able to ensure a minimum indoor temperature of higher than 19°C we collected all the gas consumption estimations in order to see which of the selected strategies optimized consumption.

TABLE XII: GAS CONSUMPTION EVALUATION FOR AN ENTIRE WEEK LIKE 1/10/11

Control Strategy	Temperature [°C]	T_{in}>19°C?	T_{in}<20.5°C?	Weekly Gas Consumption [m^3]
24h ON	55	NO		886.099
24h ON	60			1001.191
24h ON	65			1109.875
24h ON	70		NO	1224.465
18h ON	55	NO		914.164
18h ON	60			968.248
18h ON	65			1013.144
18h ON	70			1087.303
18h ON with 2 switch offs	55	NO		916.528
18h ON with 2 switch offs	60			966.237
18h ON with 2 switch offs	65			1015.766
18h ON with 2 switch offs	70			1081.163
13h ON	60	NO		878.345
13h ON	65			971.774
13h ON	70			1015.129
13h ON	75			1104.62
24h ON Bi-Climatic	60d-50n			973.398
24h ON Bi-Climatic	65d-50n			1021.106
24h ON Climatic Curve	= f (T_{out})			1091.184

As in the case of the outdoor temperature between 0 and 3°C the control strategy that best optimizes gas consumption is the *18h ON*, either with one or two switch-offs.

7.4.3.2 11/29/2010 sample day for low temperature drops:

Plotting the graphs for each control strategy we obtain:

1. *24h ON* control strategy:

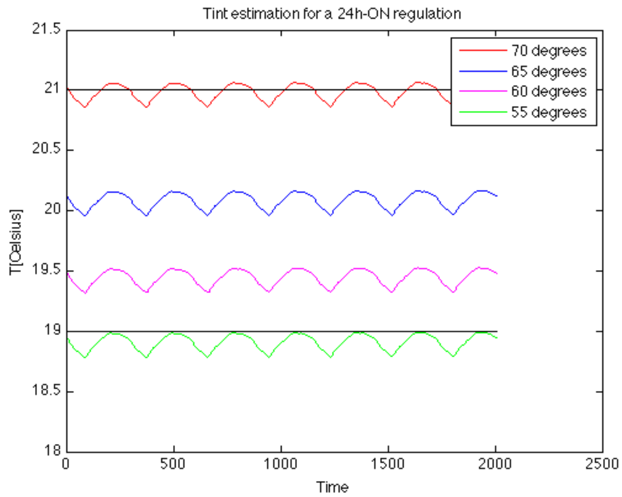


Figure 149: Indoor temperature on 11/29 for a 24h a day control

From this graph we can see that heating the building with a *24h ON at 55°C* control strategy is still not sufficient. On the other hand heating the building for 24 hours at 70°C creates an indoor temperature that's always around 21°C, which would cause gas waste. Therefore when we evaluate the gas consumption we will only consider the *24h ON at 60 and 65°C* control strategies.

2. *18h ON* control strategy:

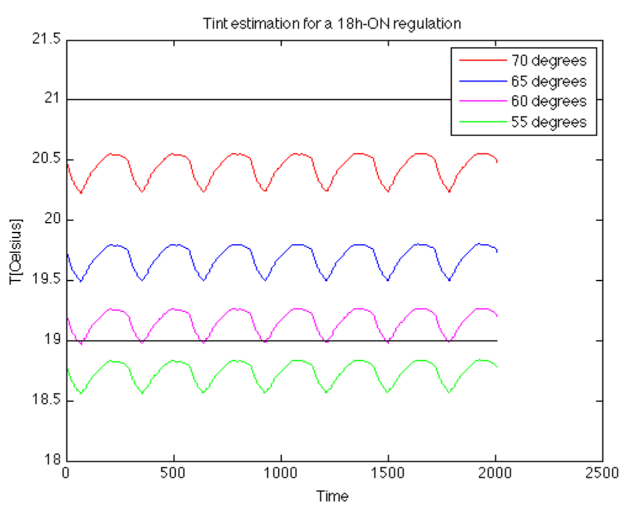


Figure 150: Indoor temperature on 11/29/2010 for an 18h ON control

In this case we can see that regulating at 55 °C isn't sufficient to ensure a comfort temperature inside the building. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60, 65 and 70°C* control strategies.

3. *18h ON (double switch-off)* control strategy:

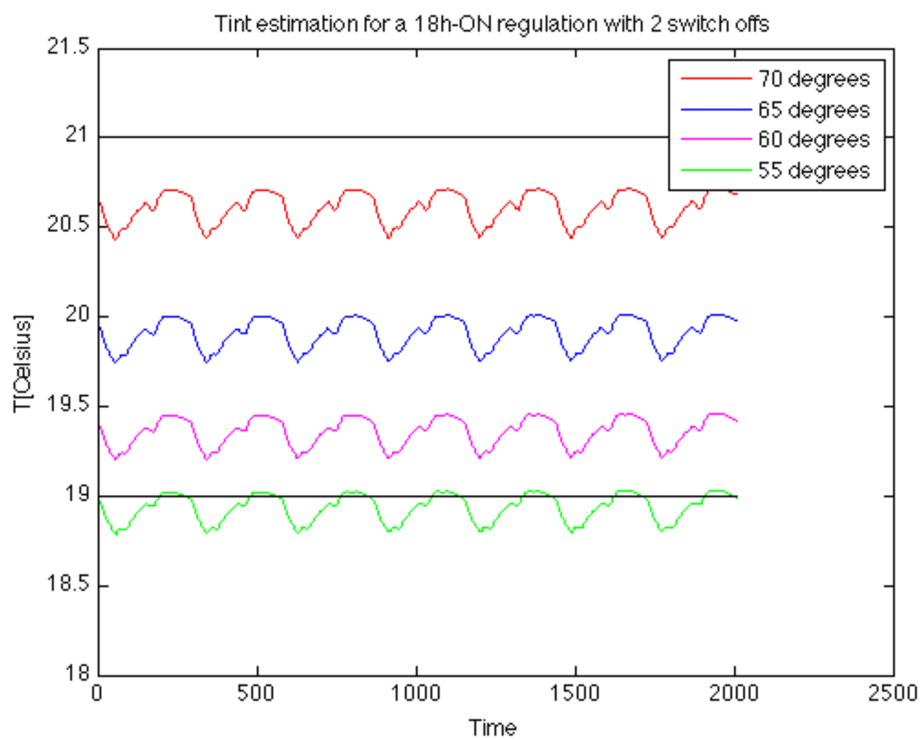


Figure 151: Indoor temperature on 11/29/2010 for an 18h a day, with a double switch off, control at various temperatures

As we can see regulating at 55°C isn't sufficient to ensure a comfort temperature inside the apartments. Furthermore, regulating with a 70°C constant outlet temperature makes the indoor temperature often higher than 20.5°C. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60 and 65°C* control strategies.

4. 13h ON control strategy:

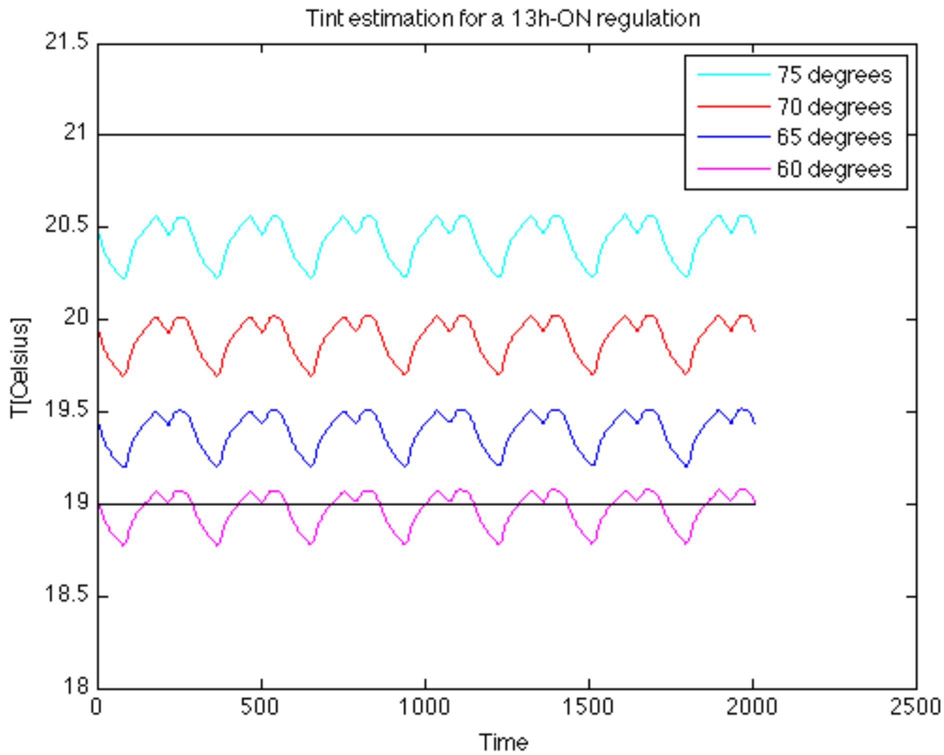


Figure 152: Indoor temperature on 11/29/2010 for a 13h ON control.

From this graph we can see that regulating under 65°C still isn't sufficient to ensure a comfort temperature inside the apartments. In fact even if for some hours a day the indoor temperature is higher than 19°C, it is not for most of the day. However we can notice that the mean temperature for this control strategy is higher than the previous case with a day presenting a small outdoor temperature drop. When we evaluate the gas consumption we will only consider the *13h ON* control strategies starting from 60°C.

5. *24h ON Bi-Climatic* control strategy:

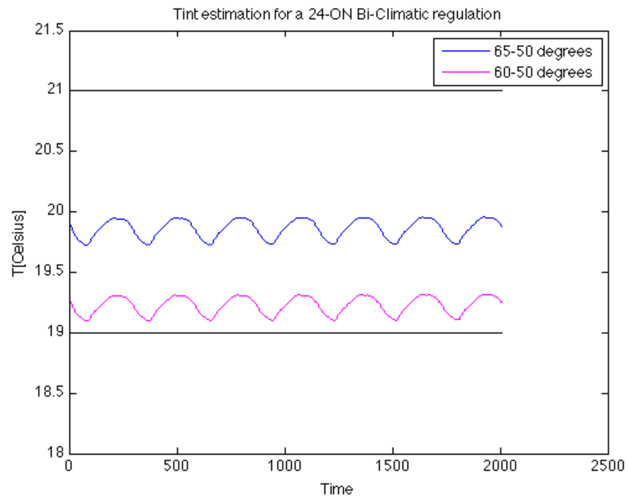


Figure 153: Indoor temperature on 11/29 for a 24h ON Bi-Climatic control

From this graph we can see that both the sample strategies are able to ensure a minimum indoor temperature of higher than 19°C (at least during the daytime). Therefore when we evaluate the gas consumption we will consider them both.

6. *Indoor temperature as a function of the Outdoor temperature* control strategy:

The outlet temperature computed as a function of the outdoor temperature for 12/29/2010:

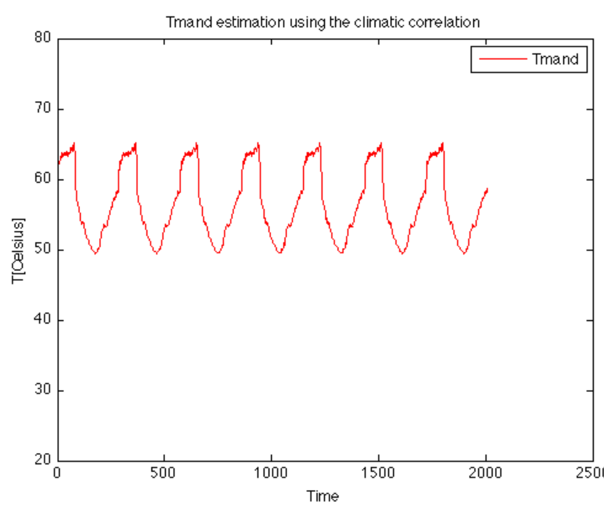


Figure 154: Outlet temperature for 11/29/2010 following the Climatic line

Using this outlet temperature as an input to compute the indoor temperature we obtained:

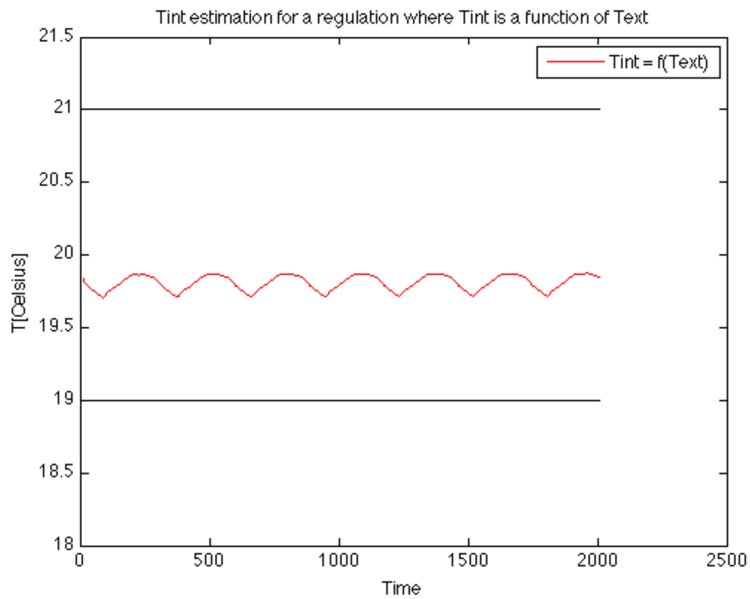


Figure 155: Indoor temperature on 11/29/2010 for a 24h a Mono-Climatic.

From this graph we can see that the control relating the Outlet temperature with the Outdoor one is always able to ensure a minimum indoor temperature of higher than 19°C.

Therefore when we evaluate the gas consumption we will consider this control strategy as one of the possible choices.

Gas consumption optimization

Once we were able to say which of the considered control strategies was able to ensure a minimum indoor temperature of higher than 19°C, we collected all the gas consumption estimations in order to see which of the selected strategies optimized gas consumption.

TABLE XIII: GAS CONSUMPTION EVALUATION FOR AN ENTIRE WEEK LIKE 11/29/10

Control Strategy	Temperature [°C]	T_{in}>19°C?	T_{in}<20.5°C?	Weekly Gas Consumption [m^3]
24h ON	55	NO		886.390
24h ON	60			1003.993
24h ON	65			1109.291
24h ON	70		NO	1222.896
18h ON	55	NO		912.381
18h ON	60			962.907
18h ON	65			1010.909
18h ON	70			1089.601
18h ON with 2 switch offs	55	NO		876.618
18h ON with 2 switch offs	60			906.692
18h ON with 2 switch offs	65			985.383
18h ON with 2 switch offs	70		NO	1039.239
13h ON	60	NO		878.711
13h ON	65			992.266
13h ON	70			1015.031
13h ON	75			1104.123
24h ON Bi-Climatic	60d-50n			970.275
24h ON Bi-Climatic	65d-50n			1021.418
24h ON Climatic Curve	= f (T_{out})			1090.105

The conclusions we can extrapolate from this chart are always the same as in the previous cases. What is important to notice again is that the control strategy that minimizes gas consumption is the *18h ON double switch-off*.

7.4.4 Outdoor Temperature >7°C:

7.4.4.1 3/22/2011 sample day for small temperature drops:

Plotting the graphs for each control strategy we obtain:

1. *24h ON* control strategy:

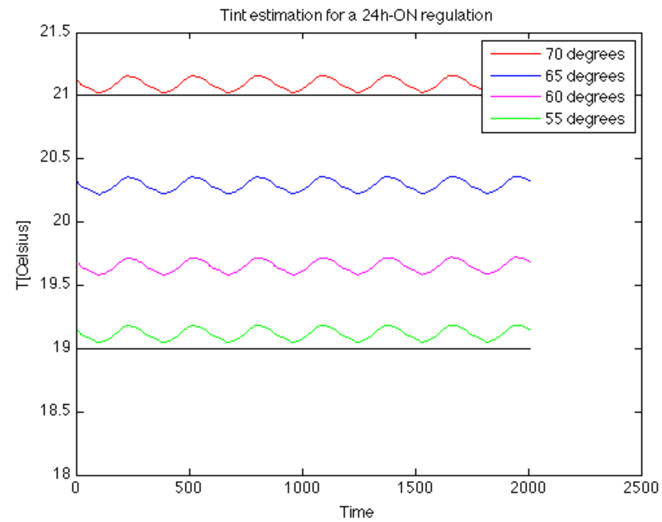


Figure 156: Indoor temperature on 3/22 for a 24h a day control

In this case we can see that we don't have any more restrictions for the 55°C control but, on the other hand, heating the building for 24 hours at 70°C creates an indoor temperature that's always over 21°C, which would cause gas waste. Therefore when we evaluate the gas consumption we will only consider the *24h ON at 55, 60 and 65°C* control strategies.

2. *18h ON* control strategy:

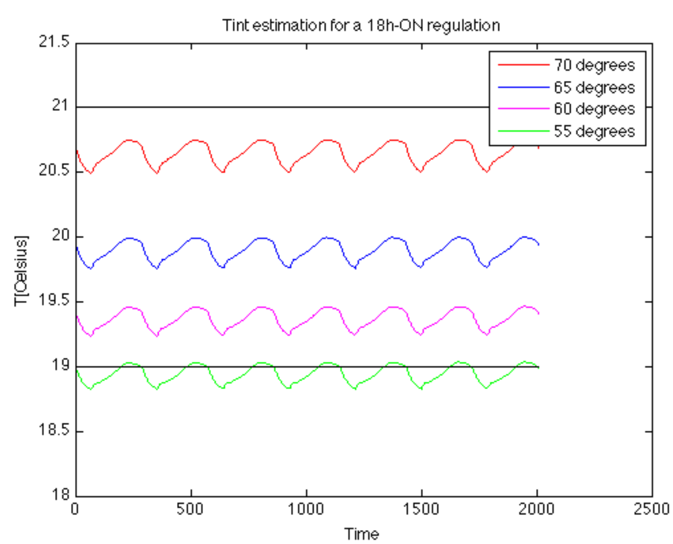


Figure 157: Indoor temperature on 3/22/2011 for an 18h ON control

As in the previous cases, regulating at 55°C isn't sufficient to ensure a comfort temperature inside the apartments. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60, 65 and 70°C* control strategies.

3. *18h ON (double switch-off)* control strategy:

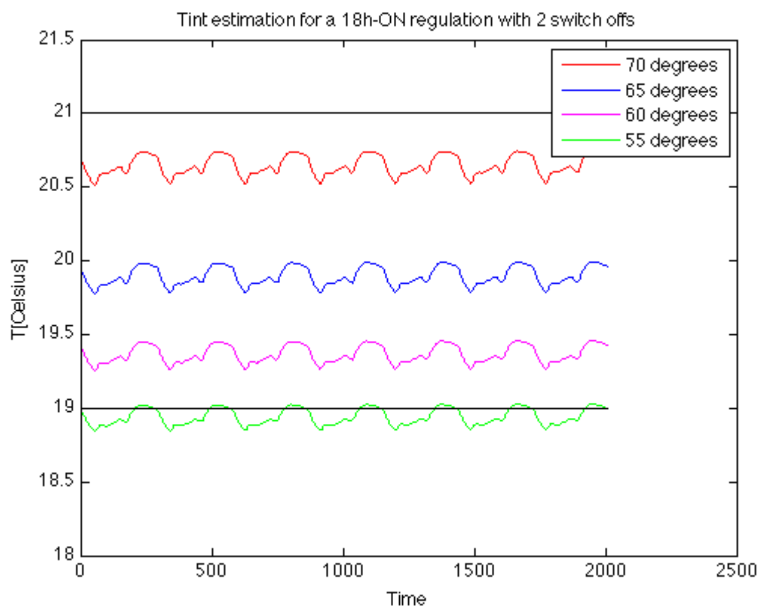


Figure 158: Indoor temperature on 3/22/2011 for an 18h a day, with a double switch off, control at various temperatures

From this graph we can see that the situation for this control strategy is almost the same as that of the previous one. Therefore regulating at 55°C is not sufficient. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60, 65 and 70°C* control strategies.

4. *13h ON* control strategy:

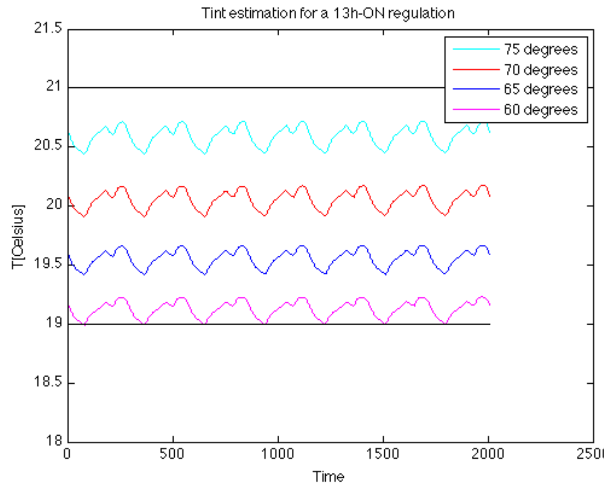


Figure 159: Indoor temperature on 3/22/2011 for a 13h ON control.

For the first time we can see that the *13h ON at 60°C* control is sufficient to ensure an indoor temperature that's always higher than 19°C. On the other hand for the first time the indoor temperature is almost always above 20.5°C when the outlet temperature is equal to 75°C. Therefore, in order not to waste gas, when we evaluate the gas consumption we will consider all the *13h ON* sample control strategies apart from the one at 75°C.

5. *24h ON Bi-Climatic* control strategy:

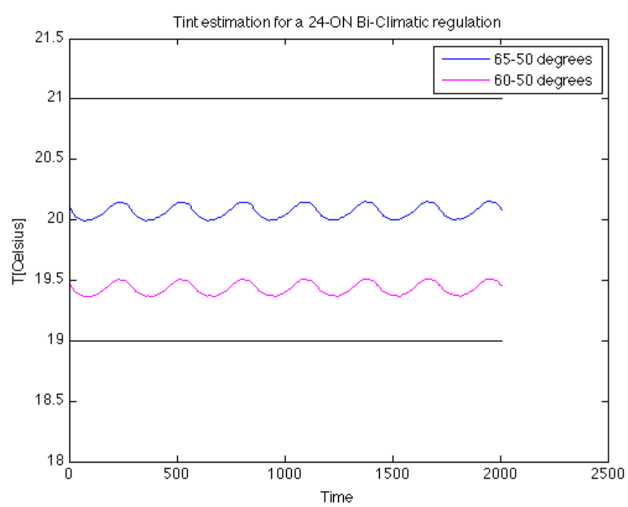


Figure 160: Indoor temperature on 3/22 for a 24h ON Bi-Climatic control

From this graph we can see that both the sample strategies are able to ensure a minimum indoor temperature of higher than 19°C (at least during the daytime). Therefore when we evaluate the gas consumption we will consider both of them.

6. *Indoor temperature as a function of the Outdoor temperature* control strategy:

The outlet temperature computed as a function of the outdoor temperature for the 03/22/2011:

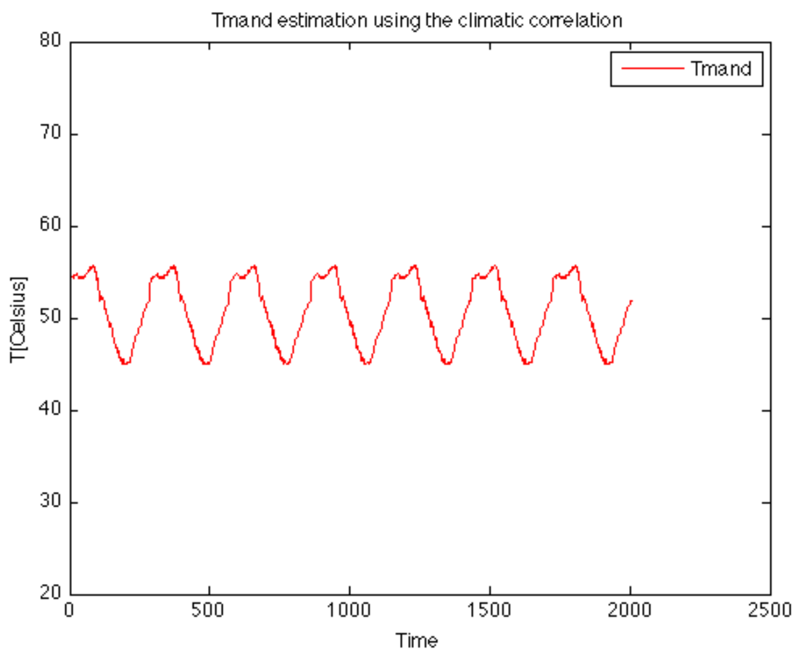


Figure 161: Outlet temperature for 3/22/2011 following the Climatic line

Using this outlet temperature as an input to compute the indoor temperature we obtained:

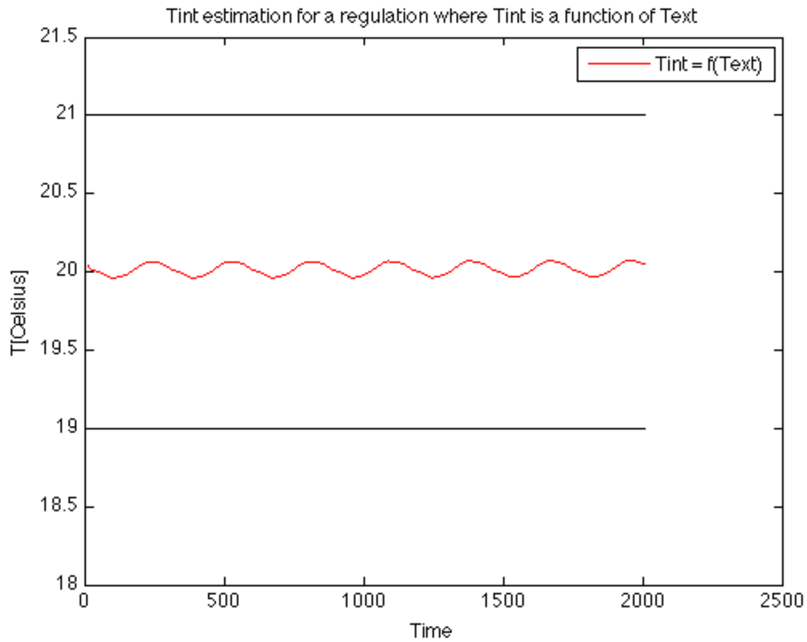


Figure 162: Indoor temperature on 3/22/2011 for a 24h a Mono-Climatic.

From this graph we can see that the control strategy adjusting the Outlet temperature to the Outdoor one is always able to ensure a minimum indoor temperature of higher than 19°C.

Therefore when we evaluate the gas consumption we will consider this control strategy as one of the possible choices.

Gas consumption optimization

Once we were able to say which of the considered control strategies was able to ensure a minimum indoor temperature higher than 19°C we collected all the gas consumption estimations in order to see which of the selected strategies optimized consumption.

TABLE XIV: GAS CONSUMPTION EVALUATION FOR AN ENTIRE WEEK LIKE 3/22/10

Control Strategy	Temperature [°C]	T_{in}>19°C?	T_{in}<20.5°C?	Weekly Gas Consumption [m^3]
24h ON	55			719.416
24h ON	60			837.347
24h ON	65			942.393
24h ON	70		NO	1055.906
18h ON	55	NO		745.391
18h ON	60			795.917
18h ON	65			843.919
18h ON	70		NO	922.611
18h ON with 2 switch offs	55	NO		699.628
18h ON with 2 switch offs	60			739.702
18h ON with 2 switch offs	65			818.393
18h ON with 2 switch offs	70		NO	872.249
13h ON	60			711.721
13h ON	65			825.276
13h ON	70			848.041
13h ON	75	NO		937.133
24h ON Bi-Climatic	60-50			803.28
24h ON Bi-Climatic	65-50			854.428
24h ON Climatic Curve	= f (T_{out})			923.115

In this case we found out that the *13h ON* control system is the one that minimizes gas consumption. When we did our first analysis this result didn't come out because the days in which we applied this control strategy were much colder than the ones we are considering now. Therefore it is safe to consider that the *13h ON* is a good strategy to use in the heating strategy in the last days of the heating season.

<u>7.4.4.2 3/18/2011 sample day for low temperature drops:</u>

Plotting the graphs for each control strategy we obtain:

1. *24h ON* control strategy:

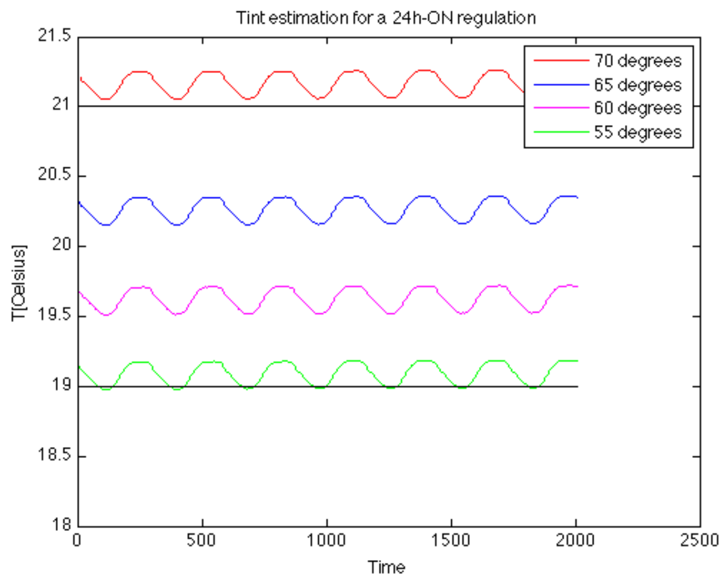


Figure 163: Indoor temperature on 3/18 for a 24h a day control

From this graph we can see that heating the building with a *24h ON at 70°C* control strategy we produce waste gas. Therefore when we evaluate the gas consumption we will only consider the *24h ON at 55, 60 and 65°C controls*.

2. *18h ON* control strategy:

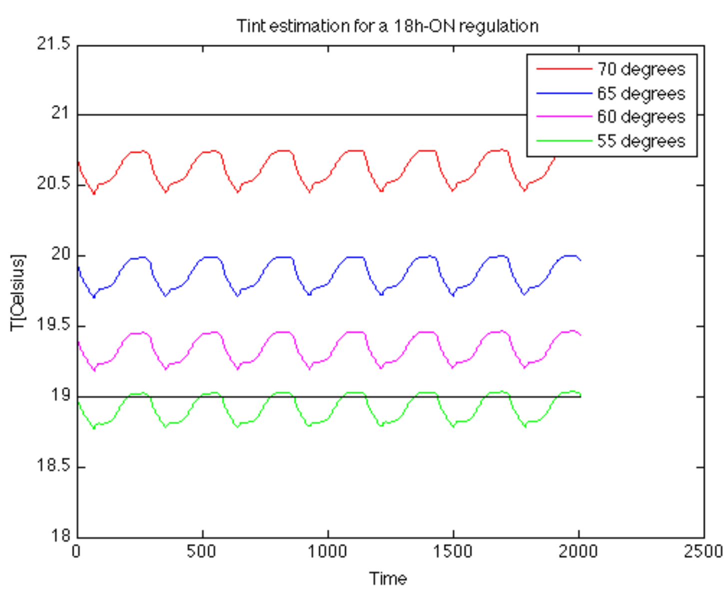


Figure 164: Indoor temperature on 3/18/2011 for an 18h ON control

From this graph we can see that, for equal outlet temperatures, heating the building with an *18h ON* control strategy results in lower indoor temperatures with respect to the ones attained using the *24h ON* control strategy. Furthermore regulating at 55°C isn't sufficient to ensure a comfort temperature inside the apartments and regulating at 70°C would cause gas waste. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60 and 65°C* control strategies.

3. *18h ON (double switch-off)* control strategy:

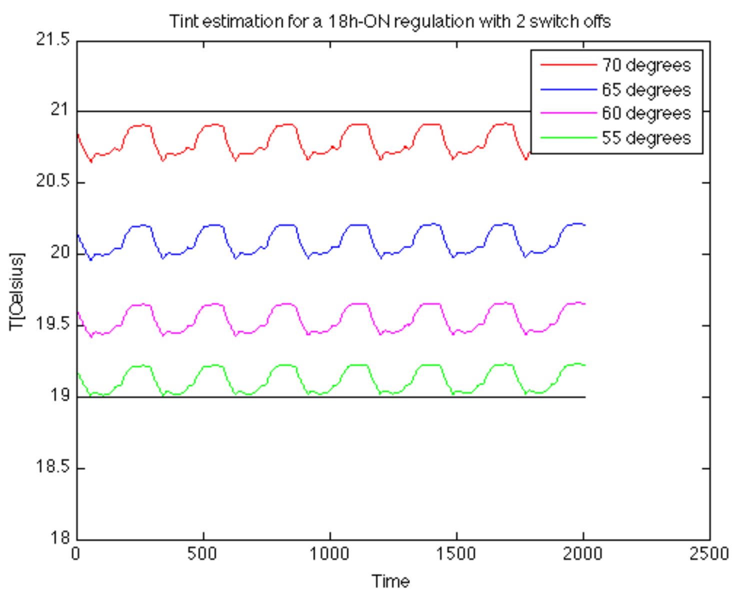


Figure 165: Indoor temperature on 3/18/2011 for an 18h a day, with a double switch off, control at various temperatures

As we can see, in this case regulating with an outlet temperature constantly equal to 70°C makes the indoor temperature always higher than 20.5°C causing gas waste. Therefore when we evaluate the gas consumption we will only consider the *18h ON at 60 and 65°C* control strategies.

4. *13h ON* control strategy:

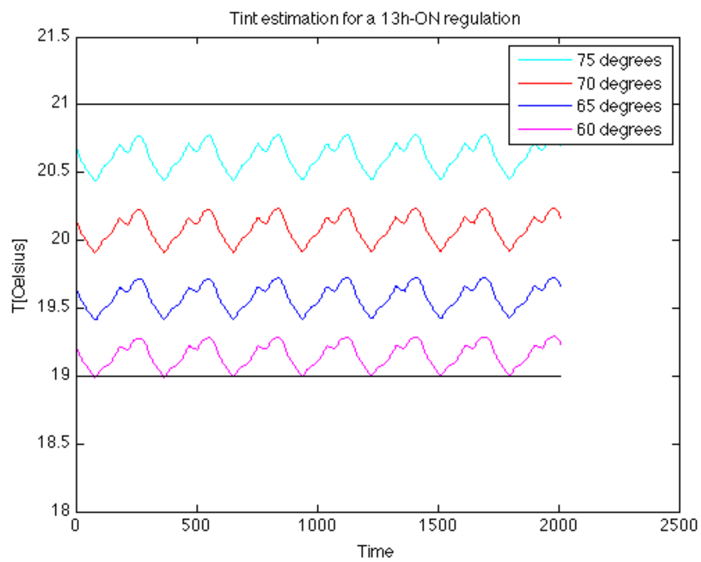


Figure 166: Indoor temperature on 3/18/2011 for a 13h ON control.

From this graph we can see that regulating above 70°C would result in some gas waste. Therefore when we evaluate the gas consumption we will only consider the *13h ON* control strategies up to 65°C.

5. *24h ON Bi-Climatic* control strategy:

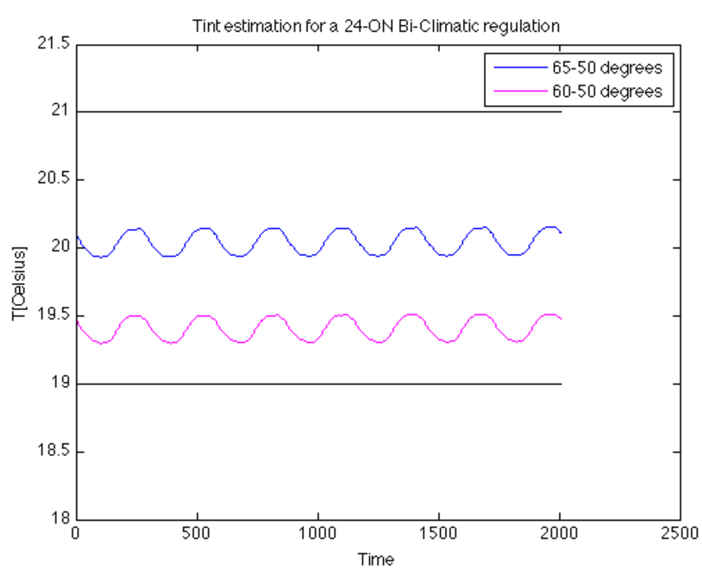


Figure 167: Indoor temperature on 3/18 for a 24h ON Bi-Climatic control

In this case both the sample strategies are able to ensure a minimum indoor temperature of higher than 19°C (at least during the daytime). Therefore when we evaluate the gas consumption we will consider both the control strategies.

6. *Indoor temperature as a function of the Outdoor temperature* control strategy:

   The outlet temperature computed as a function of the outdoor temperature for the 03/18/2011:

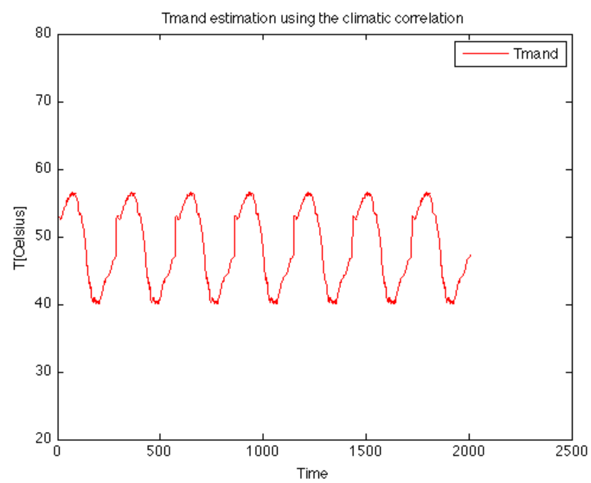


Figure 168: Outlet temperature for 3/18/2011 following the Climatic line

Using this outlet temperature as an input we obtained:

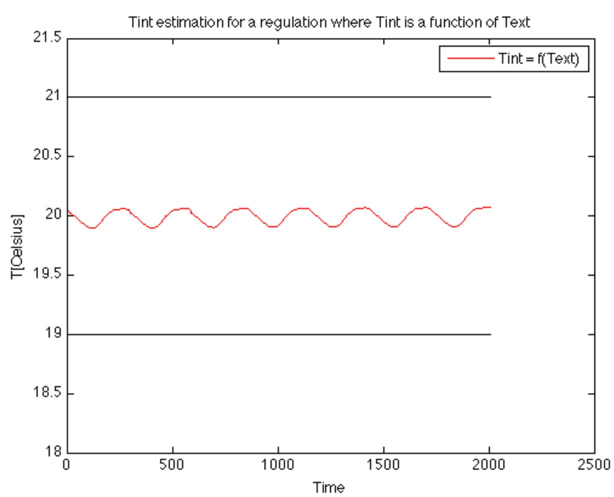


Figure 169: Indoor temperature on 3/18/2011 for a 24h a Mono-Climatic.

From this graph we can see that the control relating the outlet temperature with the outdoor temperature is always able to ensure a minimum indoor temperature of higher than 19°C. Therefore when we evaluate the gas consumption we will consider this control strategy as one of the choices.

Gas consumption optimization

Once we were able to say which of the considered control strategies was able to ensure a minimum indoor temperature higher than 19°C we collected all the gas consumption estimations in order to see which of the selected strategies optimized consumption.

TABLE XV: GAS CONSUMPTION EVALUATION FOR AN ENTIRE WEEK LIKE 3/18/11

Control Strategy	Temperature [°C]	T_{in}>19°C?	T_{in}<20.5°C?	Weekly Gas Consumption [m^3]
24h ON	55			719.707
24h ON	60			840.146
24h ON	65			941.808
24h ON	70		NO	1054.337
18h ON	55	NO		743.608
18h ON	60			790.576
18h ON	65			841.684
18h ON	70		NO	924.909
18h ON with 2 switch offs	55			735.718
18h ON with 2 switch offs	60			784.157
18h ON with 2 switch offs	65			838.015
18h ON with 2 switch offs	70		NO	919.325
13h ON	60			712.087
13h ON	65			825.768
13h ON	70			847.943
13h ON	75	NO		936.636
24h ON Bi-Climatic	60-50			800.152
24h ON Bi-Climatic	65-50			854.74
24h ON Climatic Curve	= f (T_{out})			922.036

As before, the control strategy that optimizes the gas consumption in this case is the *13h ON* one. What is interesting to notice is that the *24h ON* control consumes less gas – for low outlet temperatures – with respect to the *18h ON* control modes.

7.5 Conclusions

Summarizing the results just obtained we can see that:

TABLE XVI: CONTROL STRATEGIES THAT MINIMIZE THE GAS CONSUMPTION FOR EACH TEMPERATURE INTERVAL

Day Type	Tout < 0°C	0°C < Tout < 3°C	3°C < Tout < 7°C	Tout > 7°C
Small Temperature Drop	Bi-Climatic	18h – 1SO/2SO	18h – 1SO/2SO	13h
Big Temperature Drop	Bi-Climatic	18h – 2SO	18h – 2SO	13h

With this strategy we were able to find a way to extrapolate an ON-OFF for each temperature range and so, having all the inputs we needed, we are finally able to write our control program for the boiler.

CHAPTER 8

THE CONTROL PROGRAM

After having collected all the elements we needed for our control program, we wrote a script on MatLab to regulate the boiler with respect to the weather forecasts for the next day. In this chapter we will describe step-by-step what the control program we created does, reporting on some script parts and commenting on them.

8.1 The ARPA and the Data it Provides

The ARPA (Agenzia Regionale Protezione Ambiente) is a public body equipped with administrative, technical-juridical and financial autonomy. It is placed under the supervision of the President of the Regional Council to ensure the implementation of the Piedmont Region directives in the forecasting, prevention and environmental protection fields. The ARPA operates a regional meteorological service structured for both the monitoring of significant weather phenomena and predicting their evolution in the very short term ("Nowcasting"), and for weather forecasting in the short (up to 2-3 days) and medium (up to one week) term.

For our study we will consider the weather forecasts that ARPA Piedmont provides for the short term. After having downloaded the data we can see that they are supplied every hours, while our program uses data recorded at five minute intervals. Therefore the first thing we have to do is to interpolate them in order to obtain a "5 minute" profile. In order to do this we used the following MatLab function:

```
function[Var_scalare] =
Interpolation(Month,Day,Time,Simulation_Hours,Clock,Var_Year)

MonthVector=[31 28 31 30 31 30 31 31 30 31 30 31];

CurrentDay = 0;

if Month>1
    for M =1:(Month-1)
        CurrentDay=CurrentDay+MonthVector(M);
    end
else
end

CurrentDay = CurrentDay+(Day-1);
CurrentTime = CurrentDay*24+Time;

Var_SimHours = zeros(simulation_hours,1);

for idx = 1:simulation_hours
    Var_SimHours(idx) = Var_Year(CurrentTime+idx);
end
%% time vector for a day

for idx=1:simulation_hours
        V_s(idx)=3600 * (idx-1);
end

%% Interpolation
time = clock;

Var_scalar = interp1(V_s,Var_SimHours,time,'linear',Var_SimHours(end));
```

8.2 Indoor Temperature Choice

The second input we need to evaluate the outlet temperature is the desired indoor temperature. As an example we described the creation of a constant 20°C profile:

```
%% Creation of an indoor temperature profile.
%% Each day has 288 blocks of 5 minutes
for i=1:288*3
    Tin(i)=20;
end
```

As we can see we created a constant profile, with an indoor temperature equal at 20°C for 3 days, and not just for one. This choice was made because the data from our weather forecasts are for the next three days and not just the next day.

8.3 ON-OFF Profile Choice

The last input we need to compute the outlet temperature is the ON-OFF profile that best optimizes the gas consumption for the outdoor conditions. In order to do this we have to:

1. Compute the mean temperature and the temperature drop for each day:

```
%% Computation of the outdoor temperature average
Mean_TextD1 = mean (TextD1);
Mean_TextD2 = mean (TextD2);
Mean_TextD3 = mean (TextD3);

%% computation of the temperature drop during the day
Drop_TextD1 = max (TextD1) - min (TextD1);
Drop_TextD2 = max (TextD2) - min (TextD2);
Drop_TextD3 = max (TextD3) - min (TextD3);
```

2. Choose the best control strategy for each day:

   The following chart shows the best control strategy for each sample day we studied in the previous chapter:

Table XVII: CONTROL STRATEGIES THAT MINIMIZE THE GAS CONSUMPTION FOR EACH TEMPERATURE INTERVAL

Day Type	Tout < 0°C	0°C < Tout < 3°C	3°C < Tout < 7°C	Tout > 7°C
Small Temperature Drop	*24h Bi-Climatic*	*18h – 1SO/2SO*	*18h – 1SO/2SO*	*13h*
Big Temperature Drop	*24h Bi-Climatic*	*18h – 2SO*	*18h – 2SO*	*13h*

For simplicity we chose to apply an *18h ON double switch-off* control for days with small temperature drops and with a mean temperature between 0°C and 7°C. Therefore the previous table becomes:

Table XVIII: SIMPLIFIED CONTROL STRATEGIES CHART

Day Type	Tout < 0°C	0°C < Tout < 7°C	Tout > 7°C
Small & Big Temperature Drops	*24h Bi-Climatic*	*18h –2SO*	*13h*

Since these categories weren't obtained with a continuous model but with a discrete one, we chose to add an additional check to our control system in order to avoid continuous changes in the control strategy. In fact we don't want the system to change the control strategy if the mean temperature of the day we are considering enters a different category to the previous day for less than 0.5°C. This means that, for example, if one day the system is controlling the temperature according to a *24h Bi-Climatic* strategy and the next day the mean outdoor temperature is expected to be 0.4°C, the system maintains the *24h ON Bi-Climatic* strategy instead of changing an *18h ON double switch-off* strategy. Furthermore, we only want this check to be made on the first day when the boiler should change its control strategy – in order to avoid long periods without it controlling the temperature with the most efficient strategy. To do all this we used the following script:

```
%Choose the control strategy that minimizes the gas consumption

if Mean_TextD1 <= 0
    ONOFF1 = 1;
else if Mean_TextD1 > 0 && Mean_TextD1 < 7
        ONOFF1 = 2;
    else if Mean_TextD1 >= 7
            ONOFF1 = 3;
        end
    end
end

%having the ONOFF profile of the current day ONOFF0 and its mean
%temperature Mean_TextD0 we can write the extra check script:

if ONOFF0 == 1 && Mean_TextD1 <= 0.5
    ONOFF1 = 4;
end

if ONOFF0 == 2 && Mean_TextD1 >= -0.5 && Mean_TextD1 <= 7.5
```

```
    ONOFF1 = 5;
end

if ONOFF0 == 3 && Mean_TextD1 >= 6.5
    ONOFF1 = 6;
end

% The ONOFF profile 4,5 and 6 are respectively the same as the
number
% 1,2 and 3. We call them with a different name because we want to
% distinguish them. In fact we want the system to apply this
control
% analysis only to the first day in which the boiler would change
its
% control strategy.

% Doing the same also for the other 2 days we have the outdoor
% temperatures of, we have:

if Mean_TextD2 <= 0
    ONOFF2 = 1;
else if Mean_TextD2 > 0 && Mean_TextD2 < 7
        ONOFF2 = 2;
    else if Mean_TextD2 >= 7
            ONOFF2 = 3;
        end
    end
end

if ONOFF1 == 1 && Mean_TextD2 <= 0.5
    ONOFF2 = 4;
end

if ONOFF1 == 2 && Mean_TextD2 >= -0.5 && Mean_TextD2 <= 7.5
    ONOFF2 = 5;
end

if ONOFF1 == 3 && Mean_TextD2 >= 6.5
    ONOFF2 = 6;
end

if Mean_TextD3 <= 0
    ONOFF3 = 1;
else if Mean_TextD3 > 0 && Mean_TextD3 < 7
        ONOFF3 = 2;
    else if Mean_TextD3 >= 7
            ONOFF3 = 3;
        end
    end
end

if ONOFF2 == 1 && Mean_TextD3 <= 0.5
    ONOFF3 = 4;
end

if ONOFF2 == 2 && Mean_TextD3 >= -0.5 && Mean_TextD3 <= 7.5
    ONOFF3 = 5;
end

if ONOFF2 == 3 && Mean_TextD3 >= 6.5
    ONOFF3 = 6;
end
```

8.4 The Outlet Temperature Estimation

Once we had obtained all the inputs we needed to run our estimation model we wrote a different script for each control strategy (also because we had to apply different estimation models). Reporting these scripts for each control strategy we have:

8.4.1 *24h ON* control regulation

With this control strategy the boiler is on 24 hours a day and, as a result, its outlet temperature never falls below 50°C. This means we don't need to make any power control loop as we will have to do in the next cases when the boiler will turn off and the outlet temperature will be to be lower than a certain limit of usefulness.

```
% Load the data file
path = 'I.csv';

[Text,ONOFF,Consumo,Tmand,Tint_all,Tint_uff,Trit] =
csvimport(path,'delimiter',';','column',[3,4,5,6,7,8,9],
'outputAsChar',false,'noHeader',true);

% Estimation of the initial state
x0 = iddata (Tmand, [Text, Tint_uff], 1);

x0est = findstates (arx312, x0);

% Outlet temperature computation
sim ('simulink_I', 3074);
```

The Simulink block this script shows is:

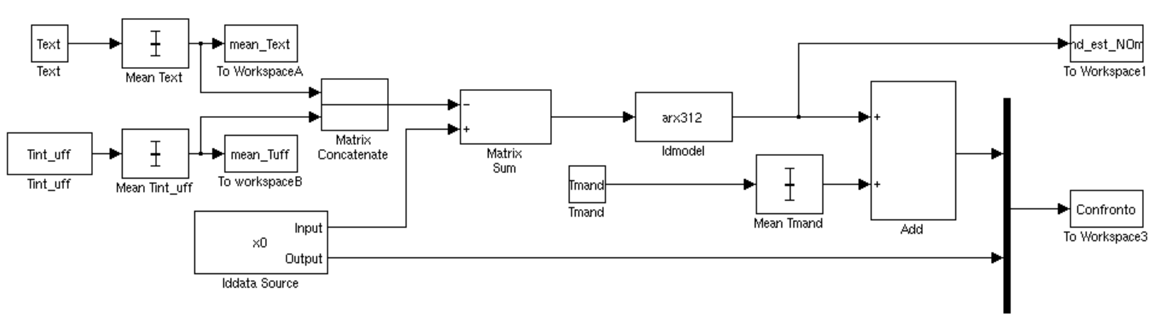


Figure 170: *24h ON* outlet temperature estimation Simulink block

8.4.2 18h ON and 13h ON

As stated before, the *18h ON* and *13h ON* control strategies can be treated in the same way. Therefore we wrote a script suitable for both of them. This script, after having loaded the data file, computes a first attempt outlet temperature. After this it ensures that the radiators' power never drops under a certain limit. As we can read in Appendix C, if a radiator exchanges heat with a power similar to 125W it is practically not exchanging any heat with the building. Therefore in these cases it is better to turn the boiler off instead of keeping it on and wasting gas. Considering the radiators' nominal power as equal to 900W we obtained:

```
% load the data file

path = 'Y.csv';

[TmandY,TritY] =
csvimport(path,'delimiter',';','column',[6,9],'outputAsChar',false,'noHea
der',true);

[Text,ONOFF,Consumo,Tint_all,Tint_uff] =
csvimport(path,'delimiter',';','column',[3,4,5,7,8],'outputAsChar',false,
'noHeader',true);

% compute the initial state for the model OE111

x0 = iddata (TmandY, [Text, Tint_uff, ONOFF], 1);

x0est = findstates (oe111, x0);

% compute the outlet temperature

sim ('simulink_Y_parte1', 2867)

for i=1:2868
    A(i,1)=i;
    A(i,2)=Tmand_est_NOmean.signals.values(i);
end

for i=1:2868
    AM(i)=Tmand_est_NOmean.signals.values(i);
end

% compute the water return temperature

sim ('simulink_Y_parte2', 2867)

for i=1:2868
    B(i,1)=i;
    B(i,2)=Trit_est_NOmean.signals.values(i);
end
```

```
for i=1:2868
    BM(i)=Trit_est_NOmean.signals.values(i);
end

for i=1:2868
    C(i,1)=i;
    C(i,2)=Tint_uff(i);
end

for i=1:2868
    CM(i)=Tint_uff(i);
end

for i=1:2868
    D(i,1)=i;
    D(i,2)=ONOFF(i);
end

for i=1:2868
    DM(i)=ONOFF(i);
end

for i=1:2868
    E(i,1)=i;
    E(i,2)=Text(i);
end

for i=1:2868
    EM(i)=Text(i);
end

% control that the boiler is always working over a certain minimum
% power and rewrite the ONOFF profile

sim ('simulink_Y_parte3', 2867)

for i=1:2868
    F(i,1)=i;
    F(i,2)=ONOFF2.signals.values(i);
end

for i=1:2868
    FM(i)=ONOFF2.signals.values(i);
end

for i=1:2868
    G(i,1)=i;
    G(i,2)=TmandY(i);
end

% calculate the final outlet temperature

sim ('simulink_Y_parte4', 2867)
```

The Simulink blocks this script shows are:

a. First Attempt Outlet Temperature Evaluation block

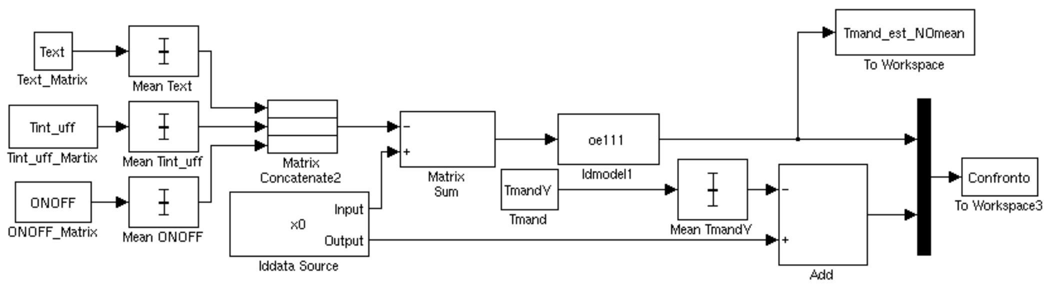


Figure 171: *18h ON* outlet temperature estimation Simulink block

b. Water Return Temperature block

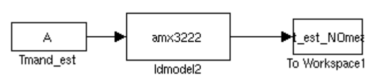


Figure 172: *18h ON* return water temperature estimation Simulink block

c. Power Check Block

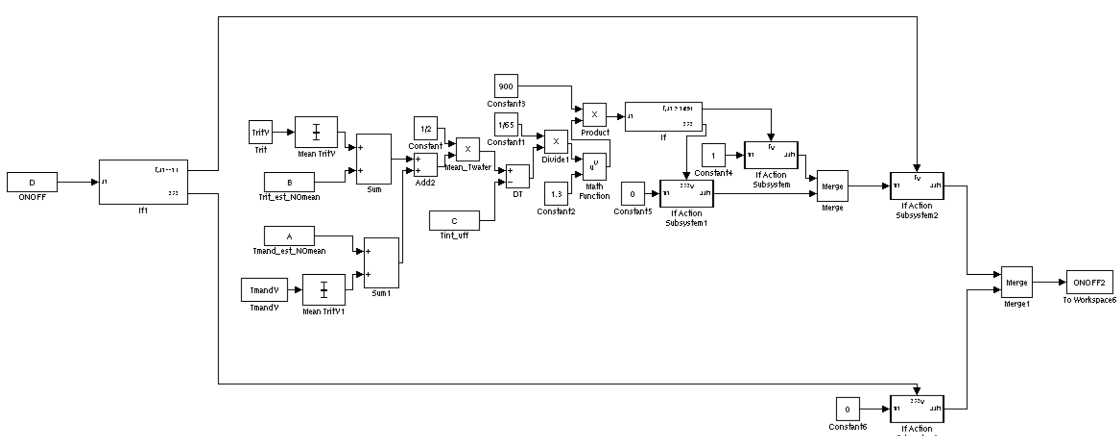


Figure 173: *18h ON* power check Simulink block

d. Final Outlet Temperature Block

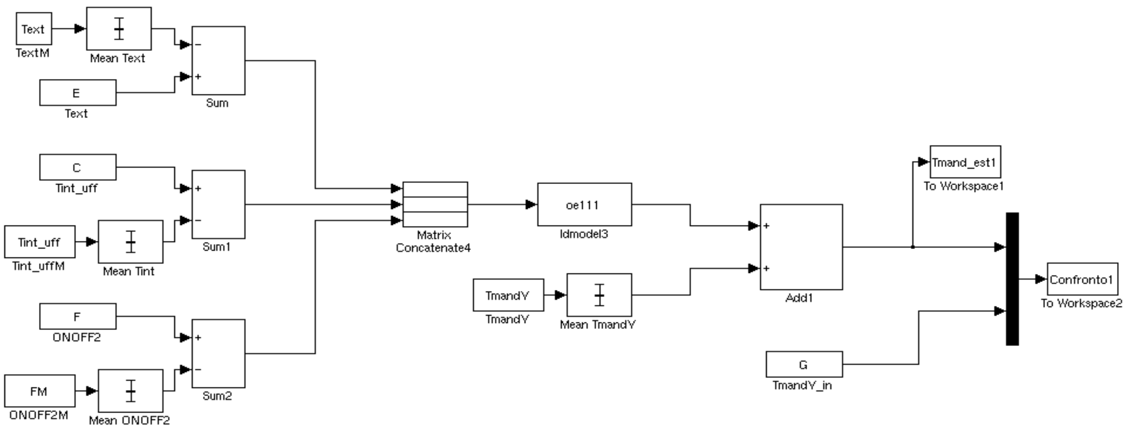


Figure 174: *18h ON final* outlet temperature estimation Simulink block

8.4.3 24h ON Bi-Climatic

As stated before, the Bi-Climatic WF doesn't isn't reliable enough in evaluating the outlet temperature (because this strategy was implemented when the temperature was already too high to have a complete set of data). The script for the *24h ON Bi-Climatic* control is practically the same as that for the *24h ON*, therefore it doesn't present any power check loop. The only difference is in the files it opens (model evaluated on a different data set) and the model it uses. The script is then:

```
%load the data file
path = 'Biclimatica.csv';

[TmandB,TritB] =
csvimport(path,'delimiter',';','column',[6,9],'outputAsChar',false,
'noHeader',true);

[Text,ONOFF,Consumo,Tint_all,Tint_uff,C1C2] =
csvimport(path,'delimiter',';','column',[3,4,5,7,8,10],'outputAsChar',fal
se,'noHeader',true);

% compute the initial state for the model OE322
x0 = iddata (TmandB, [Text, C1C2, Tint_uff], 1);

x0est = findstates (oe322, x0);

sim ('simulink_B_parte1', 5460)
```

The Simulink block this script shows is:

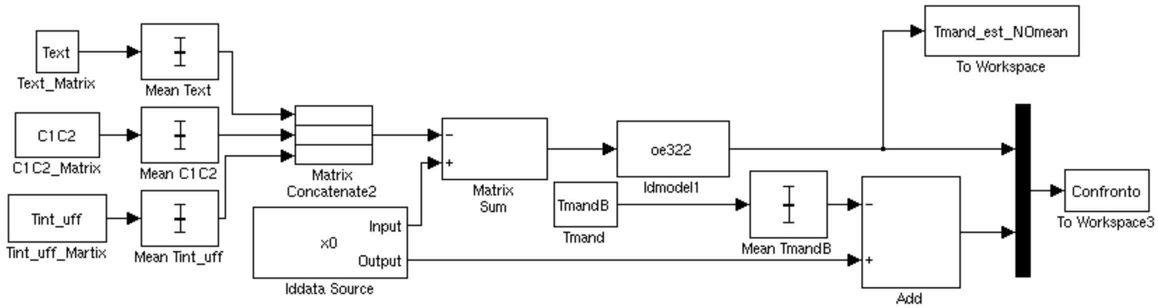


Figure 175: *24h ON Bi-Climatic* outlet temperature estimation Simluink block

8.4.4 Bi-Climatic WF

To be thorough we also looked at the control program we would have needed if we had considered the Bi-Climatic WF among the possible control strategies. As said before we didn't do this because we judged the models obtained for this control strategy as incomplete. The script is very similar to the one written for the *13h ON* and the *18h ON* control strategies. Therefore it also contains the power check loop. The script is as follows:

```
%load the data file

path = 'Biclimatica.csv';

[TmandB,TritB] =
csvimport(path,'delimiter',';','column',[6,9],'outputAsChar',false,'noHea
der',true);

[Text,ONOFF,Consumo,Tint_all,Tint_uff,C1C2] =
csvimport(path,'delimiter',';','column',[3,4,5,7,8,10],'outputAsChar',fal
se,'noHeader',true);

% compute the initial state for the model ARX611

x0 = iddata (Tmand, [Text, Tint_uff, ONOFF, C1C2], 1);

x0est = findstates (arx611, x0);

% compute the first attempt outlet temperature

sim ('simulink_B_parte1', 5460)

for i=1:5461
    A(i,1)=i;
    A(i,2)=Tmand_est_NOmean.signals.values(i);
end
```

```
for i=1:5461
    AM(i)=Tmand_est_NOmean.signals.values(i);
end

% compute the initial state for the model ARMAX3222

x2 = iddata (TritB, [TmandB], 1);

x2est = findstates (amx3222, x2);

sim ('simulink_B_parte2', 5460)

for i=1:5461
    B(i,1)=i;
    B(i,2)=Trit_est_NOmean.signals.values(i);
end

for i=1:5461
    BM(i)=Trit_est_NOmean.signals.values(i);
end

for i=1:5461
    C(i,1)=i;
    C(i,2)=Tint_uff(i);
end

for i=1:5461
    CM(i)=Tint_uff(i);
end

for i=1:5461
    D(i,1)=i;
    D(i,2)=ONOFF(i);
end

for i=1:5461
    DM(i)=ONOFF(i);
end

for i=1:5461
    E(i,1)=i;
    E(i,2)=Text(i);
end

for i=1:5461
    EM(i)=Text(i);
end

% compute the water return temperature

sim ('simulink_B_parte3', 5460)

for i=1:5461
    F(i,1)=i;
    F(i,2)=ONOFF2.signals.values(i);
end

for i=1:5461
    FM(i)=ONOFF2.signals.values(i);
end

for i=1:5461
    G(i,1)=i;
    G(i,2)=TmandB(i);
```

```
end

for i=1:5461
    H(i,1)=i;
    H(i,2)=C1C2(i);
end

for i=1:5461
    HM(i)=C1C2(i);
end

% compute the final outlet temperature

x1 = iddata (TmandB, [Text, C1C2, Tint_uff, ONOFF], 1);

x1est = findstates (arx611, x1);

sim ('simulink_B_parte4', 5460)
```

The Simulink blocks this script shows are:

a. First Attempt Outlet Temperature Evaluation block

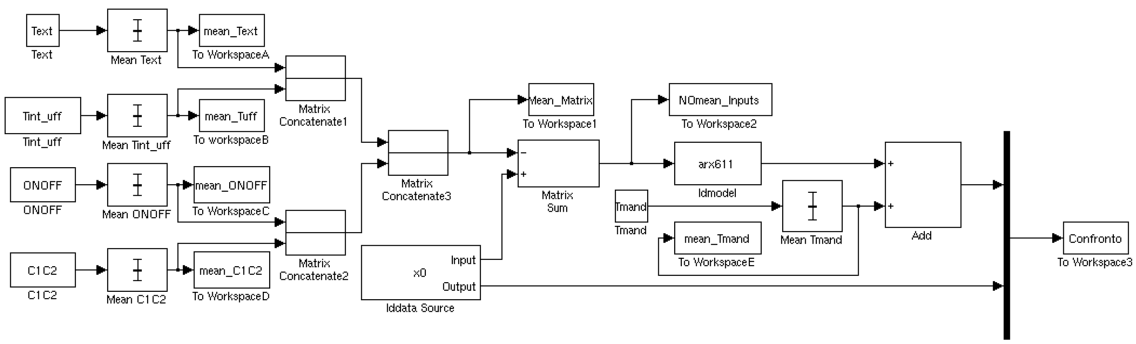


Figure 176: *Bi-Cimatic WF* outlet temperature estimation Simulink block

b. Water Return Temperature block

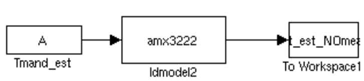


Figure 177: *Bi-Climatic WF* water return temperature estimation Simulink block

c. Power Check Block

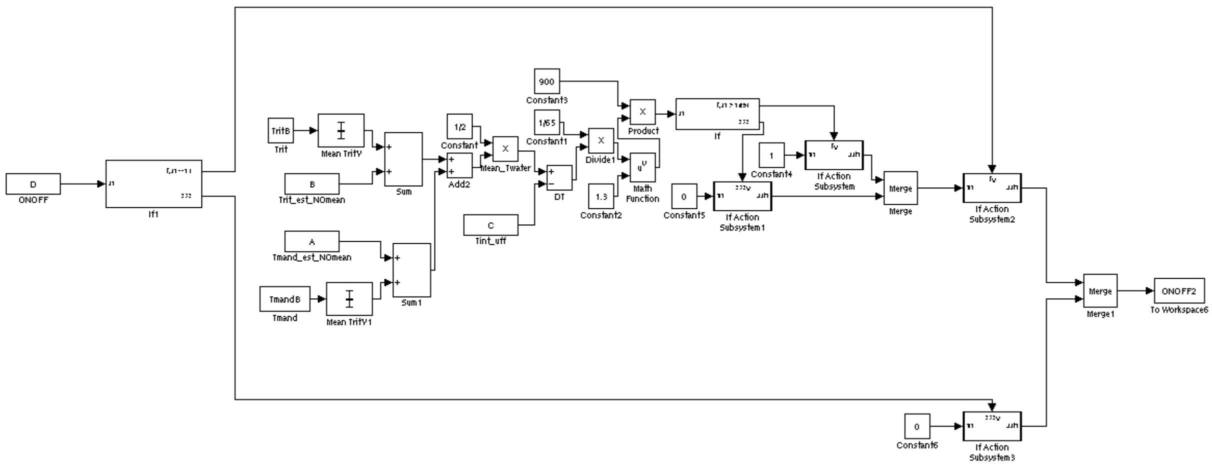


Figure 178: *Bi-Climatic WF* power check Simulink block

d. Final Outlet Temperature Block

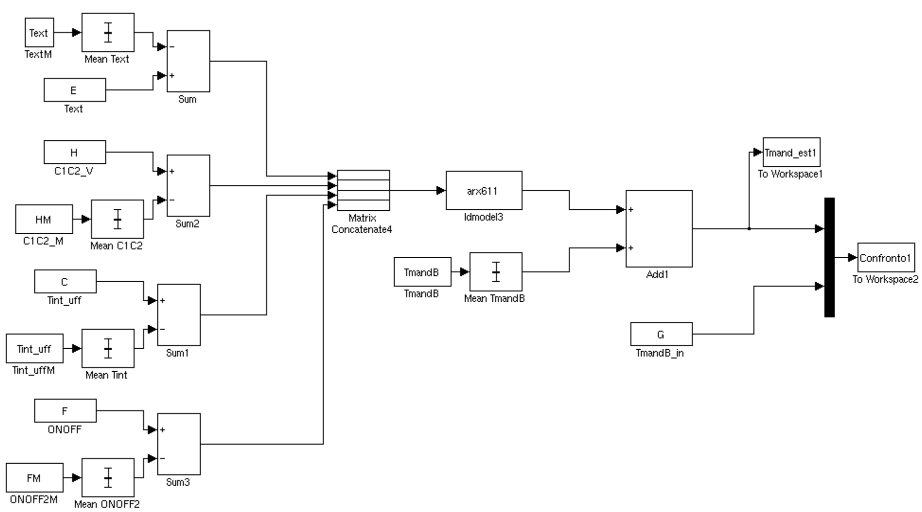


Figure 179: *Bi-Climatic WF final* outlet temperature estimation Simulink block

CHAPTER 9

CONTROL SYSTEM SIMULATIONS

Now that we have obtained our control program we want to see how it works and the results it provides. First we want to evaluate how the boiler would have worked last year during the heating season if the control system we have created had already been installed. The chart below represents the control strategy for the 2010 season:

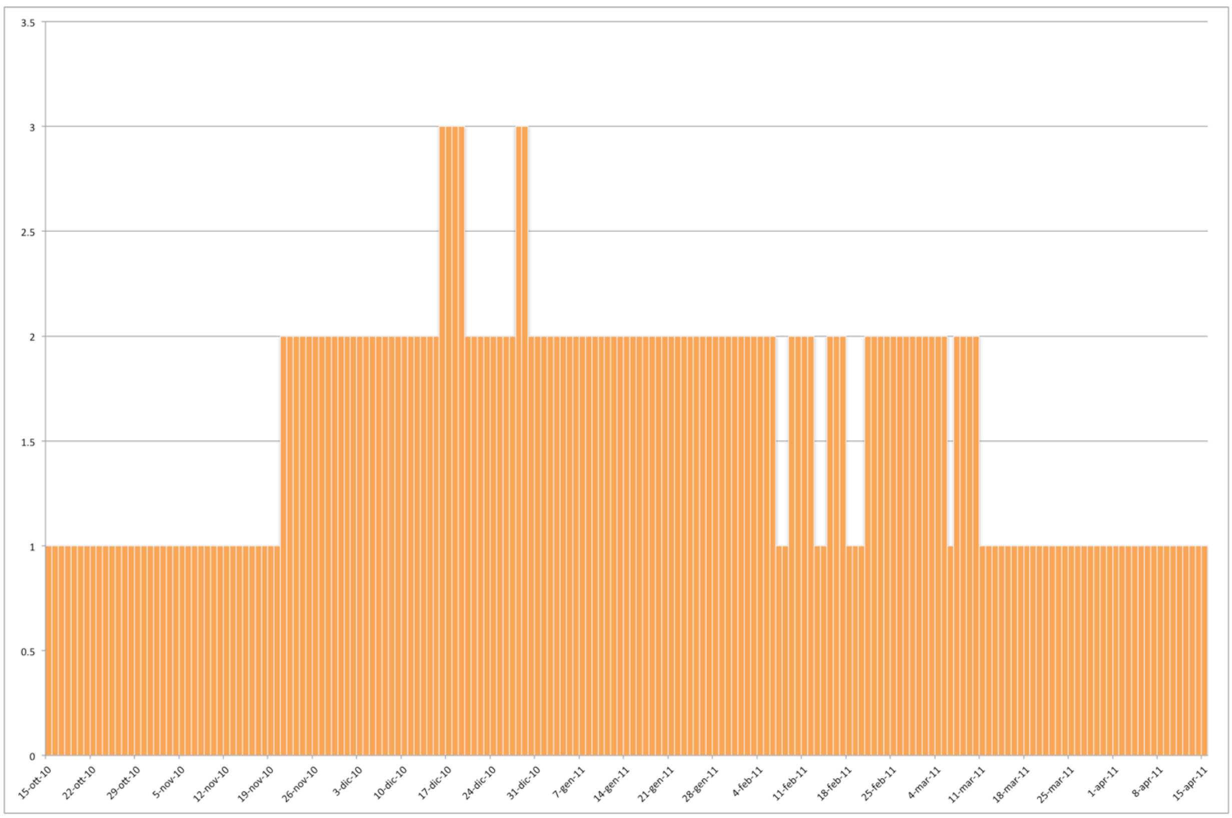


Figure 180: Control strategy for the 2010 winter season

Where the graph is equal to one the outdoor temperature was above 7°C and so the boiler would have worked for 13 hours with a double switch-off. Where it is equal to two the outdoor temperature was between 0 and 7°C, and the boiler would have worked for

18 hours with a double switch-off. Where it is equal to three the temperature was below 0°C and the boiler would have worked 24 hours a day following a Bi-Climatic control. As can be seen the number of hours the boiler works for each day is low at the beginning of the season, then increases in the middle of the winter to decrease again at the beginning of March. This pattern is to be expected but what is interesting to notice is that this program is able to save energy and gas on those days that are unexpectedly warm (in the coldest period of the year) always ensuring comfortable temperature indoors.

Having evaluated how our program would work during a real heating season we then applied it to three sample days (one for each temperature range) in winter 2010 in order to evaluate the differences between the *24h ON Mono-Climatic* control and the control that was actually applied that day.

9.1 T<0° C: 12/18/2010

The mean outdoor temperature on 12/18/2010 was -2.38°C and the outdoor temperature profile was:

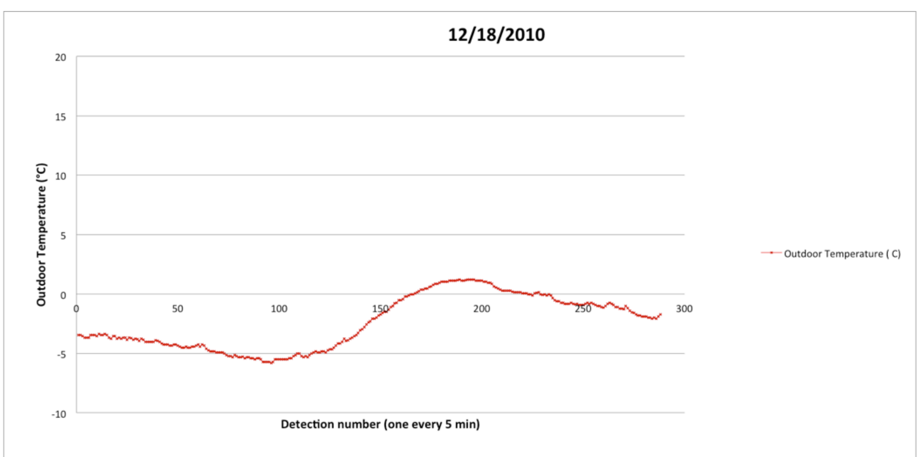


Figure 181: 12/18/2010 outdoor temperature distribution

We took this day and repeated it three times in order to have clearer graphs. Applying the control system we created to these three days and using the script summarized in Appendix D we obtained the following graph representing the estimated outlet temperature and the outlet temperature attained by programming the boiler to follow the climatic curve for 24 hours:

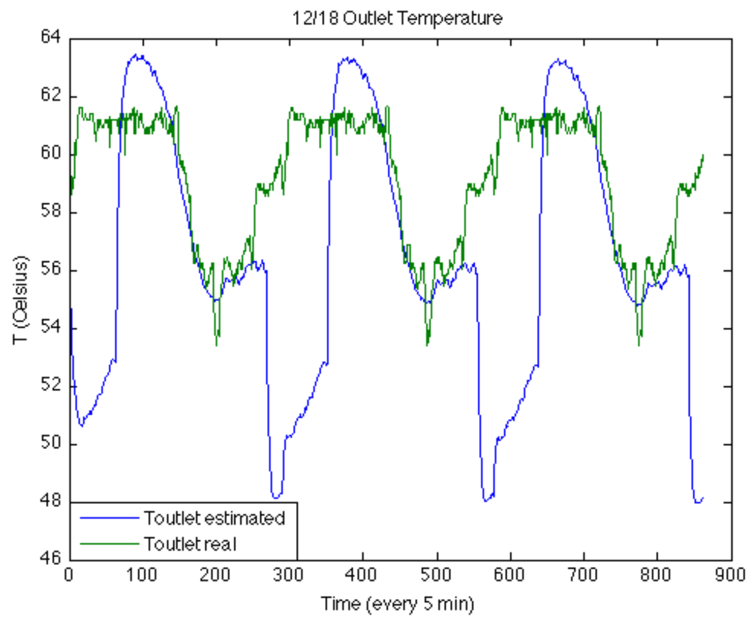


Figure 182: 12/18/2010 estimated and real outlet temperatures

As can be seen the outlet temperature estimated by our program is far less irregular and fairly similar to the real temperature – except for the early hours of the morning, when it is higher, and the nighttime hours, when it is much lower. In the first case these irregularities represent a good thing as the boiler is not continually changing its outlet power. However the difference in the nighttime outlet temperatures could lead one to think that the comfort temperature would not be ensured by the new control strategy. To understand further we then used our evaluation program to compare the indoor temperature with the two control strategies, obtaining the following graph:

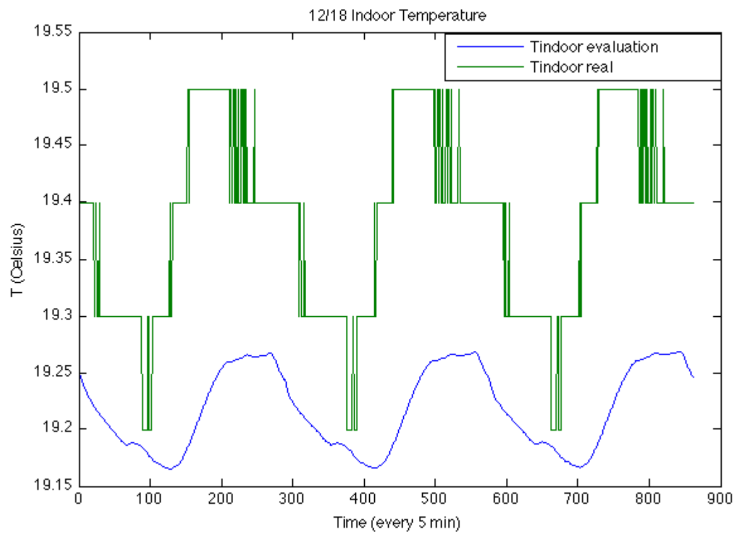


Figure 183: 12/18/2010 estimated and real indoor temperatures

As can be seen the temperature attained with our control program is always lower than the one attained with the *24h ON Mono-Climatic* control, but it is always above 19°C and so the comfort temperature is always guaranteed.

Lastly we evaluated gas consumption. Below is the graph comparing the estimated level of consumption with the actual level:

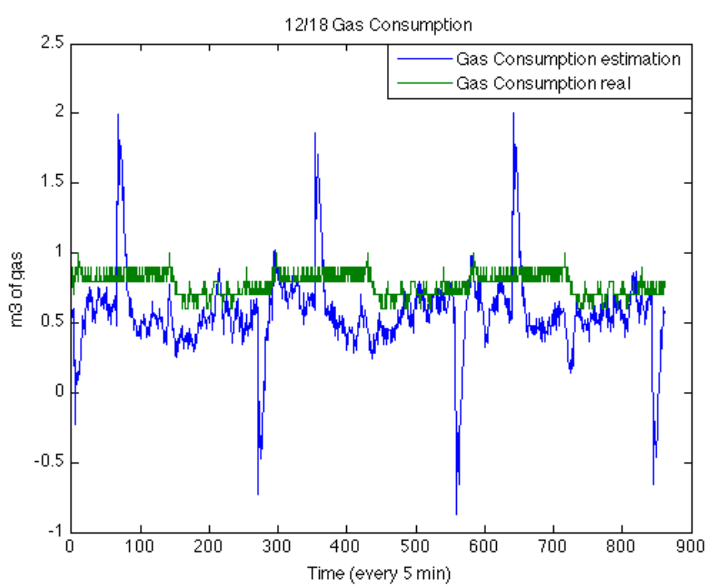


Figure 184: 12/18/2010 estimated and real gas consumptions

As can be seen the model shows that the estimated level of gas consumption is always lower than the actual level (the peaks must be discounted as they are clearly a model mistake: gas consumption lower than 0 is impossible; similarly it is impossible for gas consumption to peak at $2m^3$ when the boiler is ON 24 hours a day). As can be seen our control model (as far as concerns outdoor temperatures lower than 0°C) should be able to control the outlet temperature in order to ensure indoor comfort, at the same time minimizing gas consumption.

9.2 0°C<T<7°C: 11/29/2010

The mean outdoor temperature on the 11/29/2010 was 4.18°C and the outdoor temperature profile was:

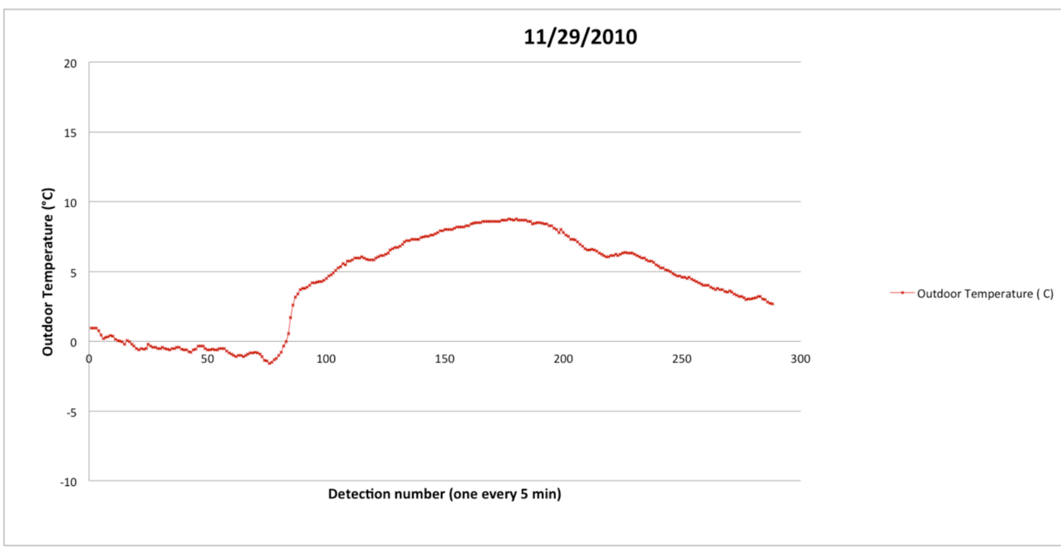


Figure 185: 11/29/2010 outdoor temperature distribution

We took this day and repeated it three times in order to have clearer graphs. Applying the control system we created to these three days and using the script summarized in Appendix D we obtained the following graph representing the estimated outlet temperature and the outlet temperature attained by programming the boiler to follow the climatic curve for 24 hours, as well as the temperature attained (in reality) by programming the boiler to work for 13 hours with a double switch-off (always following the climatic curve when on):

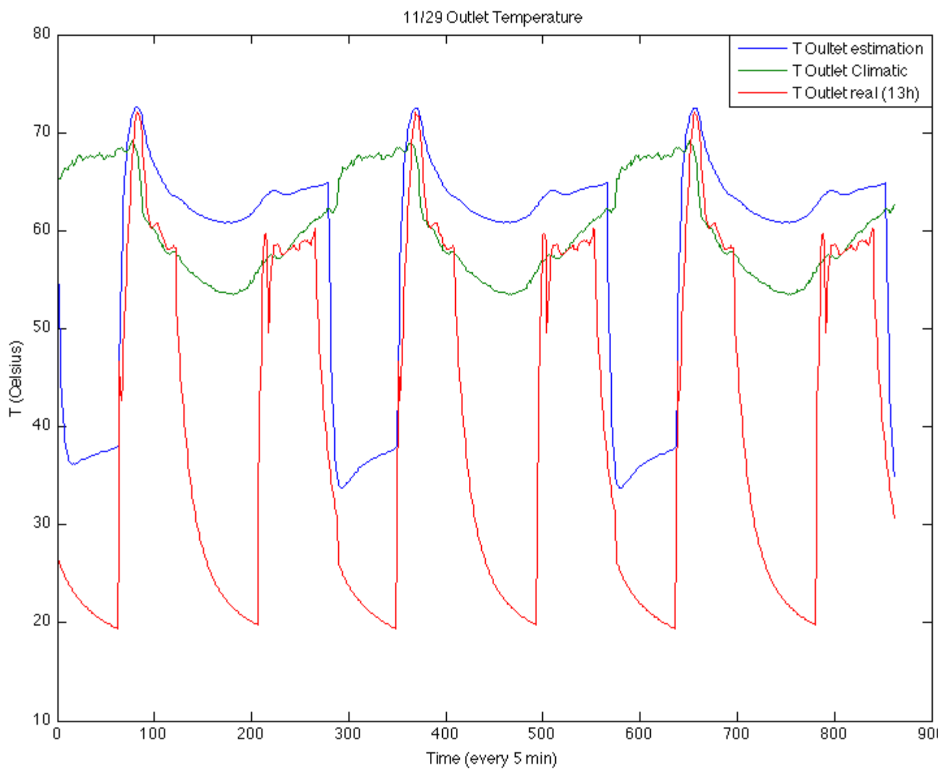


Figure 186: 11/29/2010 estimated and real outlet temperatures

As was to be expected the outlet temperature actually produced by the boiler (when on) follows the climatic curve. Furthermore we can see that the outlet temperature estimated by our program is quite often higher than the one attained with the climatic curve equation. This makes sense since we are heating for fewer hours to maintain a comfort temperature. To understand further we then used our evaluation program the indoor temperature attained with the two control strategies, obtaining the following graph:

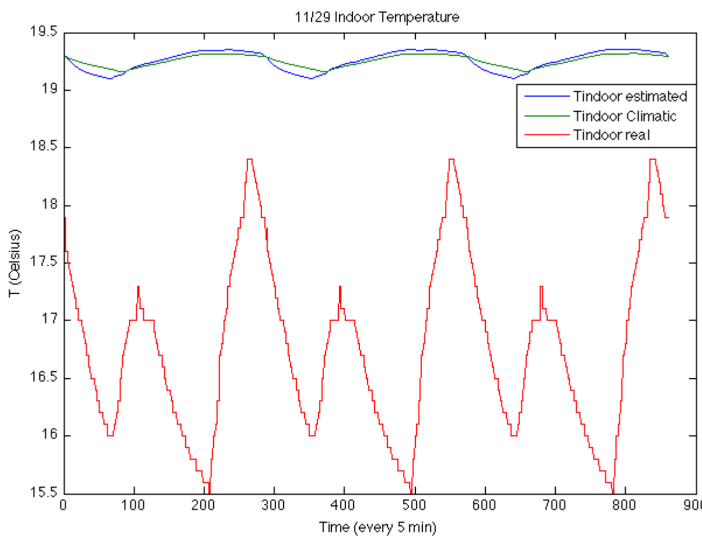


Figure 187: 11/29/2010 estimated and real indoor temperature

As can be seen the temperature we attain with our control program is almost the same as the one attained with the *24h ON Mono-Climatic* control, and it is always above 19°C, while the temperature we actually attained with a *13h ON* control strategy is always below 18.5°C. Lastly we evaluated gas consumption. Below is the graph comparing the estimated level of consumption with the actual level:

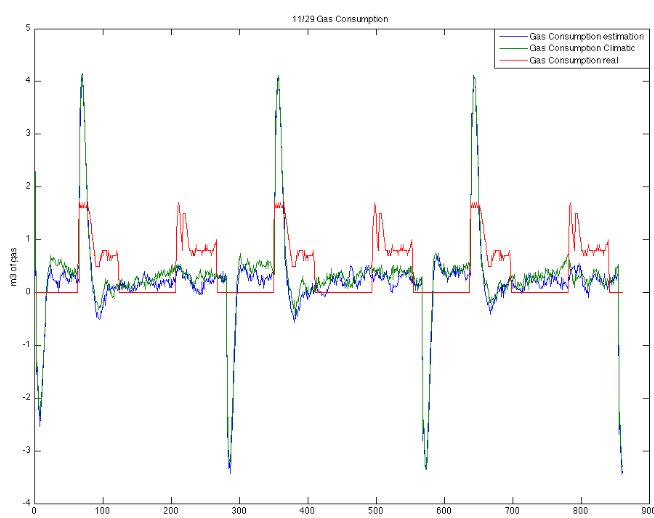


Figure 188: 11/29/2010 estimated and real gas consumptions

As can be seen the model shows that the estimated level of gas consumption is always lower than the climatic control and the *13h ON* controls (the peaks must be discounted as they are clearly a model mistake: gas consumption lower than 0 is impossible; similarly it is impossible for gas consumption to peak at $2m^3$ when the boiler is ON 24 hours a day). As can be seen our control model (as far as concerns outdoor temperatures between 0°C and 7°C) is able to control the outlet temperature in order to guarantee indoor comfort, at the same time minimizing gas consumption.

9.3 7°C<T: 11/29/2010

The mean outdoor temperature on the 3/18/2011 was 12.06°C and the outdoor temperature profile was:

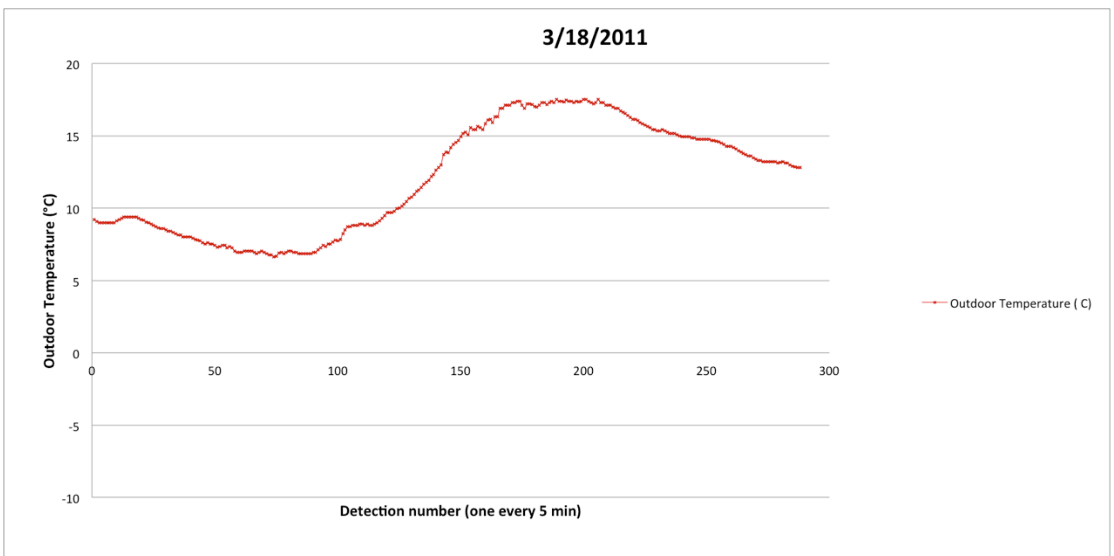


Figure 189: 3/18/2011 outdoor temperature distribution

We took this day and repeated it three times in order to have clearer graphs. Applying the control system we created to these three days and using the script summarized in Appendix D we obtained the following graph representing the estimated outlet temperature and the outlet temperature we would have attained by programming the boiler to follow the climatic curve for 24 hours and the temperature attained (in reality) by programming the boiler to work for 18 hours with a single switch-off during the day (always following the climatic curve when on):

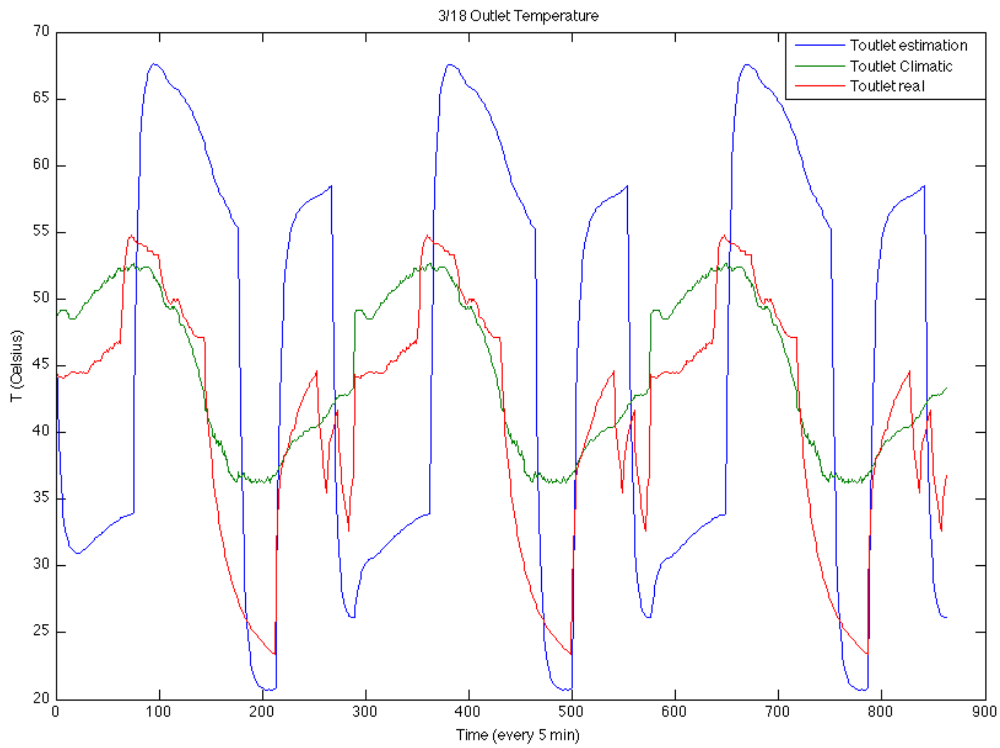


Figure 190: 3/18/2010 estimated and real outlet temperatures

As was to be expected the outlet temperature actually produced by the boiler (when on) follows the climatic curve. Furthermore we can see that the outlet temperature estimated by our program is quite often higher than the one attained with the climatic curve equation. This makes sense since we are heating for fewer hours to maintain a comfort temperature. To understand further we then used our evaluation program the indoor temperature attained with the two control strategies, obtaining the following graph:

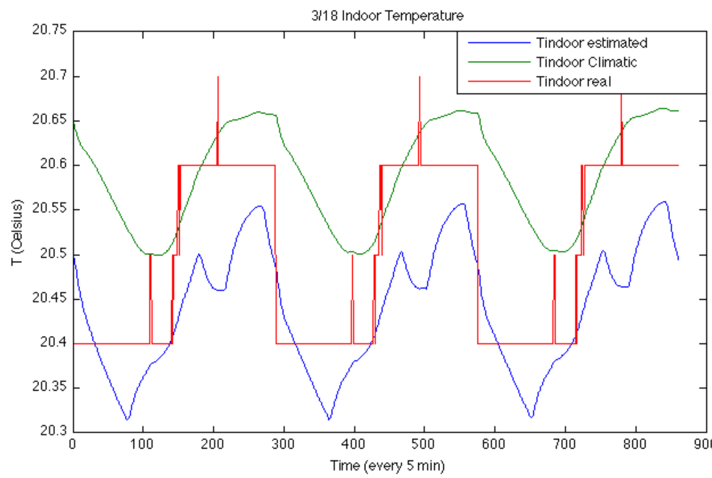


Figure 191: 3/18/2010 estimated and real outdoor temperatures

As can be seen the temperature we attain with our control program is almost the same as the one attained with the *24h ON Mono-Climatic* control, and it is always above 20°C.

Lastly we evaluated gas consumption. Below is the graph comparing the estimated level of consumption with the actual level:

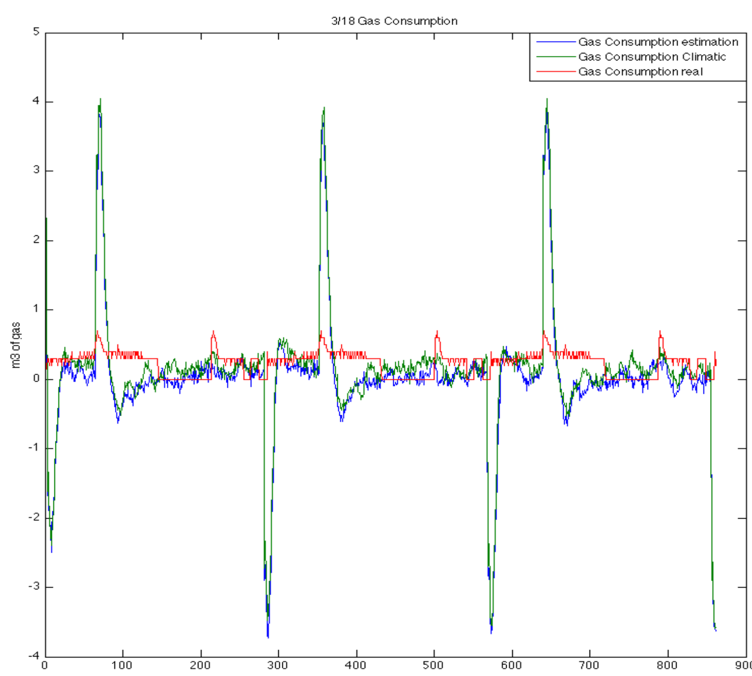


Figure 192: 3/18/2010 estimated and real gas consumptions

As can be seen the model shows that the estimated level of gas consumption is always lower than the climatic control and the *18h ON* controls (the peaks must be discounted as they are clearly a model mistake: gas consumption lower than 0 is impossible; similarly it is impossible for gas consumption to peak at $2m^3$ when the boiler is ON 24 hours a day). As can be seen our control model (as far as concerns outdoor temperatures between 0°C and 7°C) is able to control the outlet temperature in order to guarantee indoor comfort, at the same time minimizing gas consumption.

CHAPTER 10

CONCLUSIONS

The first aim of this project was to obtain a methodology for computing a regulation strategy. It can be summarized as follows:

1. Collect data for the outdoor, indoor, outlet and return temperatures and gas consumption for different regulation strategies changed by hand. The ideal way to do this is to clearly divide the collection period into blocks and to apply one regulation strategy for each of them. To compute a model we need five days at the same regulation strategy, three to evaluate and two to validate.
2. Compute the correlation models relating the indoor temperature, the outdoor temperature and the on-off profile with the outlet temperature for each regulation strategy applied during the collection period.
3. Compute the correlation models relating the outdoor temperature, the outlet temperature and the on-off profile with the indoor temperature (just for Mono-climatic regulations).
4. Compute the correlation model relating the outlet temperature and the on-off profile with gas consumption.
5. Compute the correlation model relating the outlet temperature with the return temperature.
6. Once we have computed all the models we have to choose eight sample days (or more if we want to consider more categories) and a certain number of sample regulations (in our case we had 19).

7. Apply all the sample regulations to each sample day with the MatLab script reported in Appendix A and for each day choose which regulation strategy ensures a comfort temperature inside the building, optimizing gas consumption at the same time.
8. Install the control program we wrote on MatLab in the boiler, making sure it is connected to the internet in order to download the ARPA Piedmont weather forecasts.

The disadvantage of this methodology is that it cannot be applied automatically to each building. In fact it will always need some human support in defining which regulation strategy optimizes the gas consumption for each outdoor temperature category (also manually chosen). The next step that needs to be taken to improve this project is to find a way to automate the entire process. Another reason we don't like the trial and error approach used to find the regulation categories is that we don't have a continuous equation capable of telling us where exactly it is more convenient to change regulation strategy, but we have discrete ranges. This is not ideal because some temperatures could be included in a category they don't belong in.

Apart from that, the regulation program computed exactly reflects our original purposes: the regulation models we computed consider the thermal capacity of the building – and so the indoor temperature – before evaluating, minimizing the gas consumption, the outlet temperature of the water. Furthermore the outlet temperature profile we obtain from each of the computed models cuts the highest peaks. This is important because these peaks are almost useless from the point of view of heating (the building thermal capacity is very slow) and they only waste gas. In the following figure we can see this cut at the beginning of each heating cycle:

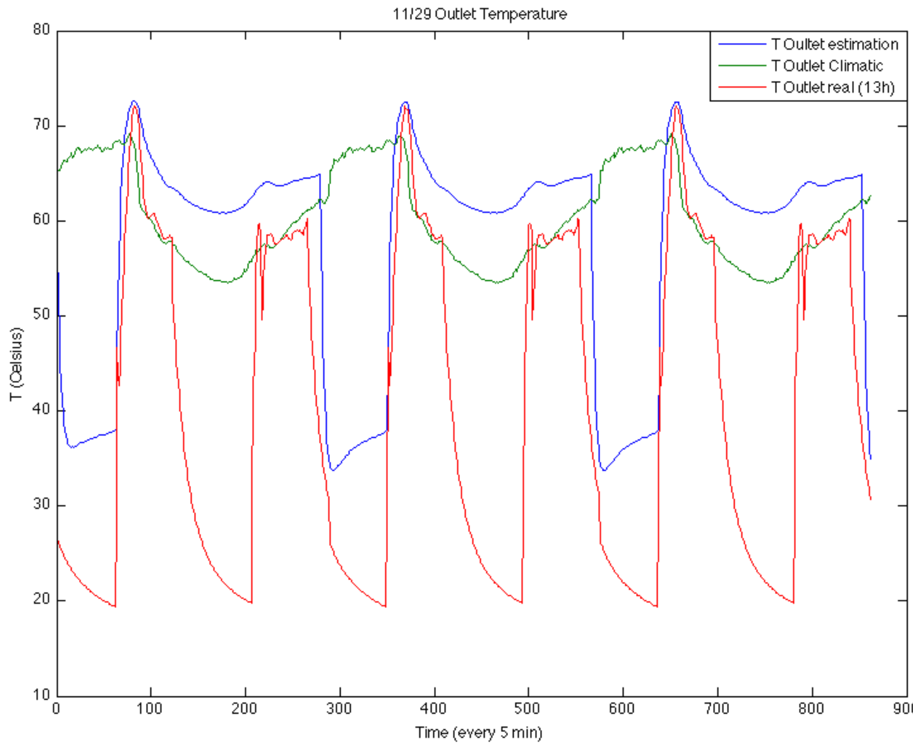


Figure 193: 11/29/2010 estimated and real outlet temperatures

Lastly, the regulation program is able to compute which regulation strategy is best to apply on each of the next three days in order to minimize gas consumption, and it is able to update it day by day.

The control system we obtained in this project is currently installed in the boiler of this building we considered to study it. We have been testing its efficacy since November 2011 and at the end of the heating season (April 2012) we will collect the data and see if the system studied was actually able to ensure lower gas consumption (with a difference large enough to justify the costs this control system would have, if put on the market) without affecting the comfort of the people living inside the building.

APPENDICES

APPENDIX A: CSVIMPORT FUNCION

Below is the MatLab function we used to import the .csv files we worked on for the whole project:

```
function varargout = csvimport( fileName, varargin )
% CSVIMPORT reads the specified CSV file and stores the contents in a cell array or matrix
%
% The file can contain any combination of text & numeric values. Output data format will vary
% depending on the exact composition of the file data.
%
% CSVIMPORT( fileName ):         fileName     -  String specifying the CSV file to be read. Set to
%                                                [] to interactively select the file.
%
% CSVIMPORT( fileName, ... ) : Specify a list of options to be applied when importing the .csv.
%                              The possible options are:
%                                delimiter     - String to be used as column delimiter. Default
%                                                value is , (comma)
%                                columns       - String or cell array of string listing the columns
%                                                from which data is to be extracted. If omitted data
%                                                from all columns in the file is imported.
%                                outputAsChar  - true / false value indicating whether the data
%                                                should be output as characters. If set to false the
%                                                function attempts to convert each column into a
%                                                numeric array, it outputs the column as characters
%                                                if conversion of any data element in the column
%                                                fails. Default value is false.
%                                uniformOutput - true / false value indicating whether output can be
%                                                returned without encapsulation in a cell array.
%                                                This parameter is ignored if the columns / table
%                                                cannot be converted into a matrix.
%                                noHeader      - true / false value indicating whether the CSV
%                                                file's first line contains column headings. Default
%                                                value is false.
%                                ignoreWSpace  - true / false value indicating whether to ignore
%                                                leading and trailing whitespace in the column
%                                                headers; ignored if noHeader is set to true.
%                                                Default value is false.
%
% The parameters must be specified in the form of param-value pairs, parameter names are not
% case-sensitive and partial matching is supported.
%
% [C1 C2 C3] = CSVIMPORT( fileName, 'columns', {'C1', 'C2', C3'}, ... )
%   This form returns the data from columns in output variables C1, C2 and C3 respectively, the
%   column names are case-sensitive and must match a column name in the file exactly. When fetching
%   data in column mode the number of output columns must match the number of columns to read or it
%   must be one. In the latter case the data from the columns is returned as a single cell matrix.
%
% [C1 C2 C3] = CSVIMPORT( fileName, 'columns', [2, 3, 4], ,'noHeader', true, ... )
%   This form returns the data from columns in output variables C1, C2 and C3 respectively, the
%   columns parameter must contain the column indices when the 'noHeader' option is set to true.

%
% Notes:  1. Function has not been tested on badly formatted CSV files.
%         2. Created using R2007b but has been tested on R2006b.
%
% Revisions:
%   04/28/2009: Corrected typo in an error message
%               Added ignoreWSpace option
%

if ( nargin == 0 ) || isempty( fileName )
  [fileName filePath] = uigetfile( '*.csv', 'Select CSV file' );
  if isequal( fileName, 0 )
    return;
  end
  fileName = fullfile( filePath, fileName );
else
  if ~ischar( fileName )
    error('csvimport:FileNameError','The first argument to %s must be .csv',
      mfilename );
  end
end
```

APPENDIX A (continued)

```
%Setup default values
p.delimiter       = ',';
p.columns         = [];
p.outputAsChar    = false;
p.uniformOutput   = true;
p.noHeader        = false;
p.ignoreWSpace    = false;

validParams     = {     ...
  'delimiter',          ...
  'columns',            ...
  'outputAsChar',       ...
  'uniformOutput',      ...
  'noHeader',           ...
  'ignoreWSpace'        ...
  };

%Parse input arguments
if nargin > 1
  if mod( numel( varargin ), 2 ) ~= 0
    error( 'csvimport:InvalidInput', ['All input parameters after the filename
      must be in the form of param-value pairs'] );
  end
  params  = lower( varargin(1:2:end) );
  values  = varargin(2:2:end);

  if ~all( cellfun( @ischar, params ) )
    error( 'csvimport:InvalidInput', ['All input parameters after the fileName
      must be in the form of param-value pairs'] );
  end

  lcValidParams   = lower( validParams );
  for ii =  1 : numel( params )
    result        = strmatch( params{ii}, lcValidParams );
    %If unknown param is entered ignore it
    if isempty( result )
      continue
    end
    %If we have multiple matches make sure we don't have a single unambiguous
    %match before throwing an error
    if numel( result ) > 1
      exresult    = strmatch( params{ii}, validParams, 'exact' );
      if ~isempty( exresult )
        result    = exresult;
      else
        %We have multiple possible matches, prompt user to provide an
unambiguous match
        error( 'csvimport:InvalidInput', 'Cannot find unambiguous match for
          parameter ''%s''', varargin{ii*2-1} );
      end
    end
    result      = validParams{result};
    p.(result)  = values{ii};
  end
end

%Check value attributes
if isempty( p.delimiter ) || ~ischar( p.delimiter )
  error( 'csvimport:InvalidParamType', ['The ''delimiter'' parameter must be a
    non-empty character array'] );
end
```

APPENDIX A (continued)

```
if isempty( p.noHeader ) || ~islogical( p.noHeader ) || ~isscalar( p.noHeader )
  error( 'csvimport:InvalidParamType', ['The ''noHeader'' parameter must be a
    non-empty logical scalar'] );
end
if ~p.noHeader
  if ~isempty( p.columns )
    if ~ischar( p.columns ) && ~iscellstr( p.columns )
      error( 'csvimport:InvalidParamType', ['The ''columns'' parameter must be
        a character array or a cell array of strings for CSV files containing
        column headers on the first line'] );
    end
    if p.ignoreWSpace
      p.columns = strtrim( p.columns );
    end
  end
else
  if ~isempty( p.columns ) && ~isnumeric( p.columns )
    error( 'csvimport:InvalidParamType', ['The ''columns'' parameter must be a
    numeric array for CSV files containing column headers on the first line']);
  end
end
if isempty( p.outputAsChar ) || ~islogical( p.outputAsChar ) || ~isscalar(
p.outputAsChar )
  error( 'csvimport:InvalidParamType', ['The ''outputAsChar'' parameter must be
    a non-empty logical scalar'] );
end
if isempty( p.uniformOutput ) || ~islogical( p.uniformOutput ) || ~isscalar(
p.uniformOutput )
  error( 'csvimport:InvalidParamType', ['The ''uniformOutput'' parameter must
    be a non-empty logical scalar'] );
end

%Open file
[fid msg] = fopen( fileName, 'rt' );
if fid == -1
  error( 'csvimport:FileReadError', 'Failed to open ''%s'' for reading.\nError
    Message: %s',fileName, msg );
end

colMode         = ~isempty( p.columns );
if ischar( p.columns )
  p.columns     = cellstr( p.columns );
end
nHeaders        = numel( p.columns );

if colMode
  if ( nargout > 1 ) && ( nargout ~= nHeaders )
    error( 'csvimport:NumOutputs', ['The number of output arguments must be 1
      or equal to the number of column names when fetching data for specific
      columns'] );
  end
end

%Read first line and determine number of columns in data
rowData         = fgetl( fid );
rowData         = regexp( rowData, p.delimiter, 'split' );
nCols           = numel( rowData );

%Check whether all specified columns are present if used in column mode and
%store their indices
if colMode
  if ~p.noHeader
```

```
    if p.ignoreWSpace
      rowData      = strtrim( rowData );
    end
    colIdx         = zeros( 1, nHeaders );
    for ii = 1 : nHeaders
      result       = strmatch( p.columns{ii}, rowData );
      if isempty( result )
        fclose( fid );
        error( 'csvimport:UnknownHeader', ['Cannot locate column header ''%s''
          in the file ''%s''. Column header names are case sensitive.'],
          p.columns{ii}, fileName );
      elseif numel( result ) > 1
        exresult   = strmatch( p.columns{ii}, rowData, 'exact' );
        if numel( exresult ) == 1
          result   = exresult;
        else
          warning( 'csvimport:MultipleHeaderMatches', ['Column header name
            ''%s'' matched multiple p.columns in the file, only the first match
            (%d) will be used.'], p.columns{ii}, result(1) );
        end
      end
      colIdx(ii)   = result(1);
    end
  else
    colIdx         = p.columns(:);
    if max( colIdx ) > nCols
      fclose( fid );
      error( 'csvimport:BadIndex', ['The specified column index ''%d'' exceeds
        the number of columns (%d) in the file'], max( colIdx ), nCols );
    end
  end
end

%Calculate number of lines
pos               = ftell( fid );
if pos == -1
  msg = ferror( fid );
  fclose( fid );
  error( 'csvimport:FileQueryError', 'FTELL on file ''%s'' failed.\nError
    Message: %s', fileName, msg );
end
data              = fread( fid );
nLines            = numel( find( data == sprintf( '\n' ) ) ) + 1;
%Reposition file position indicator to beginning of second line
if fseek( fid, pos, 'bof' ) ~= 0
  msg = ferror( fid );
  fclose( fid );
  error( 'csvimport:FileSeekError', 'FSEEK on file ''%s'' failed.\nError
    Message: %s', fileName, msg );
end

data              = cell( nLines, nCols );
data(1,:)         = rowData;
emptyRowsIdx      = [];
%Get data for remaining rows
for ii = 2 : nLines
  rowData         = fgetl( fid );
  if isempty( rowData )
    emptyRowsIdx = [emptyRowsIdx(:); ii];
    continue
  end
  rowData         = regexp( rowData, p.delimiter, 'split' );
```

APPENDIX A (continued)

```
  nDataElems    = numel( rowData );
  if nDataElems < nCols
    warning( 'csvimport:UnevenColumns', ['Number of data elements on line %d (
      (%d) differs from that on the first line (%d). Data in this line will be
      padded.'], ii, nDataElems, nCols );
    rowData(nDataElems+1:nCols) = {''};
  elseif nDataElems > nCols
    warning( 'csvimport:UnevenColumns', ['Number of data elements on line %d
      (%d) differs from that one the first line (%d). Data in this line will be
      truncated.'], ii, nDataElems, nCols );
    rowData     = rowData(1:nCols);
  end
  data(ii,:)    = rowData;
end
%Close file handle
fclose( fid );
data(emptyRowsIdx,:)   = [];

%Process data for final output
uniformOutputPossible  = ~p.outputAsChar;
if p.noHeader
  startRowIdx          = 1;
else
  startRowIdx          = 2;
end
if ~colMode
  if ~p.outputAsChar
    %If we're not outputting the data as characters then try to convert each
    %column to a number
    for ii = 1 : nCols
      colData     = cellfun( @str2num, data(startRowIdx:end,ii),
       'UniformOutput', false );
      %If any row contains an entry that cannot be converted to a number then
      %return the whole
      %column as a char array
      if ~any( cellfun( @isempty, colData ) )
        if ~p.noHeader
          data(:,ii)= cat( 1, data(1,ii), colData{:} );
        else
          data(:,ii)= colData;
        end
      end
    end
  end
  varargout{1}    = data;
else
  %In column mode get rid of the headers (if present)
  data            = data(startRowIdx:end,colIdx);
  if ~p.outputAsChar
    %If we're not outputting the data as characters then try to convert each
    %column to a number
    for ii = 1 : nHeaders
      colData     = cellfun( @str2num, data(:,ii), 'UniformOutput', false );
      %If any row contains an entry that cannot be converted to a number then
      %return the whole
      %column as a char array
      if ~any( cellfun( @isempty, colData ) )
        data(:,ii)= colData;
      else
        %If any column cannot be converted to a number then we cannot convert
        %the output to an array
        %or matrix i.e. uniform output is not possible
```

APPENDIX A (continued)

```
        uniformOutputPossible = false;
      end
    end
  end
  if nargout == nHeaders
    %Loop through each column and convert to matrix if possible
    for ii = 1 : nHeaders
      if p.uniformOutput && ~any( cellfun( @ischar, data(:,ii) ) )
        varargout{ii} = cell2mat( data(:,ii) );
      else
        varargout{ii} = data(:,ii);
      end
    end
  else
    %Convert entire table to matrix if possible
    if p.uniformOutput && uniformOutputPossible
      data          =  cell2mat( data );
    end
    varargout{1}  = data;
  end
end
```

APPENDIX B: CONTROL STRATEGY EVALUATION PROGRAM

Below is the MatLab script we used to evaluate the best Control strategy for each sample day and the Simulink blocks recalled when computing the Indoor Temperature and the Gas Consumption.

1. Simulink Block 1: For evaluating the Indoor Temperature

   This block scheme:

   - Takes the outdoor temperature and the outlet temperature vectors from the workspace;
   - Creates a matrix out of the 2 vectors;
   - Subtracts the Outdoor Temperature mean or the Outlet Temperature average (of the values used to create the OE221 model) from each value of the matrix just composed;
   - Takes the new matrix and uses it as an input for the OE221 model;
   - Adds the Indoor Temperature average computed to the data used to create the model, obtaining an estimation of the Indoor Temperature for the conditions given in the input;
   - Compares the Indoor temperature with the limit temperatures we set at the beginning.

APPENDIX B (continued)

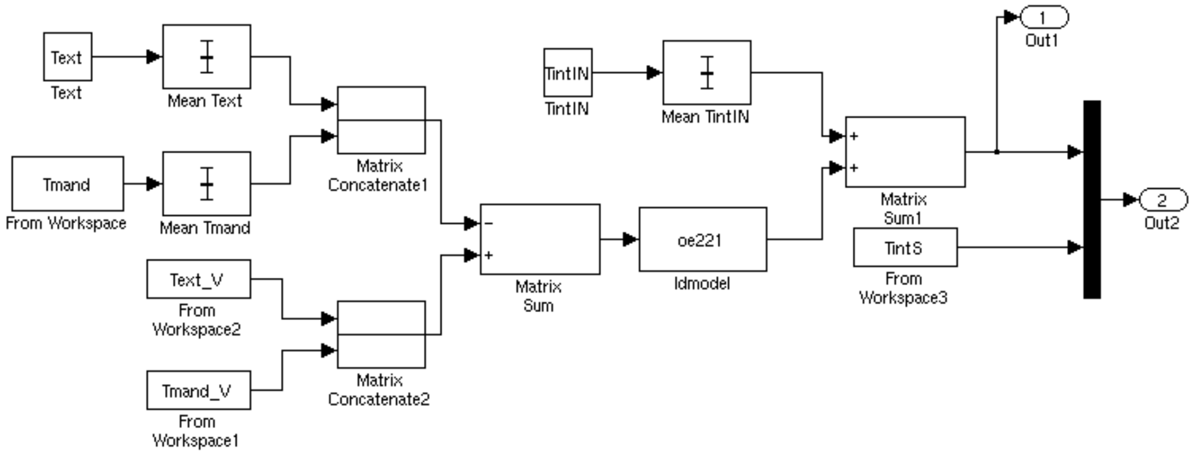


Figure 194: Outlet temperature estimation Simulink block

2. Simulink Block 2: For evaluating Gas Consumption

   This block scheme:

- Takes the outlet temperature and the ON-OFF profile vectors from the workspace;
- Creates a matrix out of the 2 vectors;
- Subtracts the Outlet Temperature mean or the ON-OFF profile average (of the values we used to create the ARX741 model) from each value of the matrix just composed;
- Takes the new matrix and uses it as an input for the ARX741 model;
- Adds the mean Gas Consumption computed to the data used to create the model, obtaining an estimation of the Gas Consumption for the conditions given in the input.

APPENDIX B (continued)

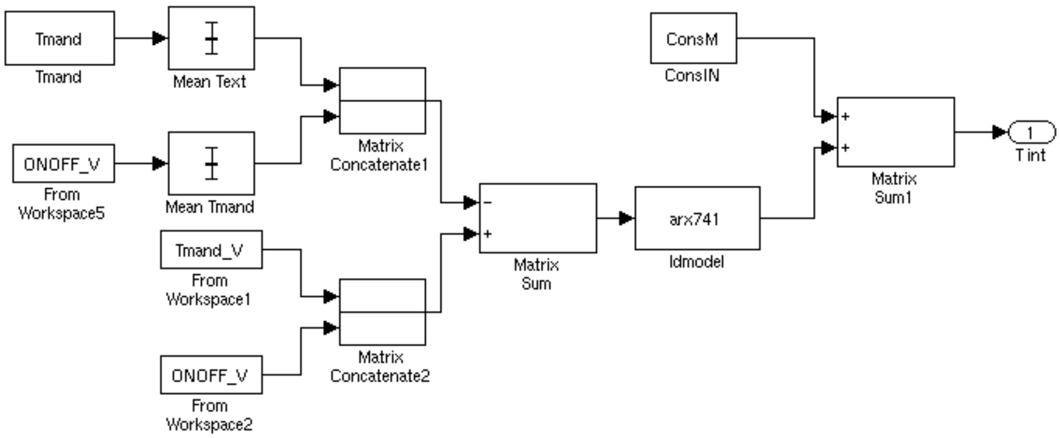


Figure 195: Gas Contumption estimation Simulink block

3. Control Strategy Program:

In order to evaluate which Control strategy is best for each sample day a script was written on MatLab that:

- Imports the Outdoor Temperature profile of the sample day;
- Fixes the limits for the Indoor Temperature at 19° Celsius and 21° Celsius;
- Loads the means of the data we used to evaluate the models;
- Loads the Outlet Temperature Profile and the ON-OFF profile for each Control strategy;
- Iterates the daily data for an entire week to evaluate a possible relationship between the models used and the time. In this way a weekly profile was obtained where all the days had the same Outdoor temperatures and the same Control strategy;
- Computes the Indoor Temperatures and the Gas consumption for each Control strategy;

APPENDIX B (continued)

- Plots the Indoor temperatures for each Control strategy and displays the gas consumption values.

```
%load the outdoor temperature profile

path = '22_3.csv';

[Text_solo] =
csvimport(path,'delimiter',';','column',[3],'outputAsChar',false,
'noHeader',true);


for i=1:287
    Text_V(i,1)=i;
    Text_V(i,2)=Text_solo(i);
    Text_V(i+287,1)=i+287;
    Text_V(i+287,2)=Text_solo(i);
    Text_V(i+574,1)=i+574;
    Text_V(i+574,2)=Text_solo(i);
    Text_V(i+861,1)=i+861;
    Text_V(i+861,2)=Text_solo(i);
    Text_V(i+1148,1)=i+1148;
    Text_V(i+1148,2)=Text_solo(i);
    Text_V(i+1435,1)=i+1435;
    Text_V(i+1435,2)=Text_solo(i);
    Text_V(i+1722,1)=i+1722;
    Text_V(i+1722,2)=Text_solo(i);
end

Text = zeros (2009,1);

for i=1:287
    Text(i)=Text_solo(i);
    Text(i+287)=Text_solo(i);
    Text(i+574)=Text_solo(i);
    Text(i+861)=Text_solo(i);
    Text(i+1148)=Text_solo(i);
    Text(i+1435)=Text_solo(i);
    Text(i+1722)=Text_solo(i);
end

%load the inside temperature extremes

for i=1:2009
    TintS(i,1)=i;
    TintS(i,2)=21;
end

for i=1:2009
    TintI(i,1)=i;
    TintI(i,2)=19;
end

%Since the models are evaluated with the data without their mean,
%they give as output the values of Inside Temperature and of Gas
%Consumption without their mean. To add the mean we have to load
%the data on which the model was evaluated, as follows:
```

APPENDIX B (continued)

```
path = 'I+Y.csv';

[ConsIN] =
csvimport(path,'delimiter',';','column',[5],'outputAsChar',false,'noHead
er',true);

ConsI = zeros (2009,1);

ConsM = mean (ConsIN);

for i=1:2009
    ConsI(i)=ConsIN(i);
end

path = 'Y.csv';

[TintIN1] =
csvimport(path,'delimiter',';','column',[8],'outputAsChar',false,
'noHeader',true);

%load the ONOFF and the output temperature profiles
%for the various Control strategies

path = '70_24h.csv';

[ONOFF_70_24h,Tmand_70_24h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

ONOFF_V = zeros (2009,2);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_70_24h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_70_24h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_70_24h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_70_24h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_70_24h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_70_24h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_70_24h(i);
end

Tmand_V = zeros (2009,2);

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_70_24h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_70_24h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_70_24h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_70_24h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_70_24h(i);
```

APPENDIX B (continued)

```
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_70_24h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_70_24h(i);
end

Tmand = zeros (2009,1);

for i=1:287
    Tmand(i)=Tmand_70_24h(i);
    Tmand(i+287)=Tmand_70_24h(i);
    Tmand(i+574)=Tmand_70_24h(i);
    Tmand(i+861)=Tmand_70_24h(i);
    Tmand(i+1148)=Tmand_70_24h(i);
    Tmand(i+1435)=Tmand_70_24h(i);
    Tmand(i+1722)=Tmand_70_24h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_70_24h,Confronto1] = sim ('Tint', 2008);

Mean_Tint_70_24h = mean (Tint_70_24h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_70_24h] = sim ('Consumo', 2008);

Tot_Cons_70_24h = 0;

for i = 1:2009
    Tot_Cons_70_24h = Tot_Cons_70_24h + Cons_70_24h (i);
end

path = '65_24h.csv';

[ONOFF_65_24h,Tmand_65_24h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_65_24h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_65_24h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_65_24h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_65_24h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_65_24h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_65_24h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_65_24h(i);
end

for i=1:287
```

APPENDIX B (continued)

```
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_65_24h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_65_24h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_65_24h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_65_24h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_65_24h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_65_24h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_65_24h(i);
end

for i=1:287
    Tmand(i)=Tmand_65_24h(i);
    Tmand(i+287)=Tmand_65_24h(i);
    Tmand(i+574)=Tmand_65_24h(i);
    Tmand(i+861)=Tmand_65_24h(i);
    Tmand(i+1148)=Tmand_65_24h(i);
    Tmand(i+1435)=Tmand_65_24h(i);
    Tmand(i+1722)=Tmand_65_24h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_65_24h,Confronto2] = sim ('Tint', 2008);

Mean_Tint_65_24h = mean (Tint_65_24h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_65_24h] = sim ('Consumo', 2008);

Tot_Cons_65_24h = 0;

for i = 1:2009
    Tot_Cons_65_24h = Tot_Cons_65_24h + Cons_65_24h (i);
end

path = '60_24h.csv';

[ONOFF_60_24h,Tmand_60_24h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_60_24h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_60_24h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_60_24h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_60_24h(i);
    ONOFF_V(i+1148,1)=i+1148;
```

APPENDIX B (continued)

```
    ONOFF_V(i+1148,2)=ONOFF_60_24h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_60_24h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_60_24h(i);;
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_60_24h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_60_24h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_60_24h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_60_24h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_60_24h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_60_24h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_60_24h(i);
end

for i=1:287
    Tmand(i)=Tmand_60_24h(i);
    Tmand(i+287)=Tmand_60_24h(i);
    Tmand(i+574)=Tmand_60_24h(i);
    Tmand(i+861)=Tmand_60_24h(i);
    Tmand(i+1148)=Tmand_60_24h(i);
    Tmand(i+1435)=Tmand_60_24h(i);
    Tmand(i+1722)=Tmand_60_24h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_60_24h,Confronto3] = sim ('Tint', 2008);

Mean_Tint_60_24h = mean (Tint_60_24h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_60_24h] = sim ('Consumo', 2008);

Tot_Cons_60_24h = 0;

for i = 1:2009
    Tot_Cons_60_24h = Tot_Cons_60_24h + Cons_60_24h (i);
end

path = '55_24h.csv';

[ONOFF_55_24h,Tmand_55_24h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,'
noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
```

APPENDIX B (continued)

```
    ONOFF_V(i,2)=ONOFF_55_24h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_55_24h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_55_24h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_55_24h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_55_24h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_55_24h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_55_24h(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_55_24h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_55_24h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_55_24h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_55_24h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_55_24h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_55_24h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_55_24h(i);
end

for i=1:287
    Tmand(i)=Tmand_55_24h(i);
    Tmand(i+287)=Tmand_55_24h(i);
    Tmand(i+574)=Tmand_55_24h(i);
    Tmand(i+861)=Tmand_55_24h(i);
    Tmand(i+1148)=Tmand_55_24h(i);
    Tmand(i+1435)=Tmand_55_24h(i);
    Tmand(i+1722)=Tmand_55_24h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_55_24h,Confronto4] = sim ('Tint', 2008);

Mean_Tint_55_24h = mean (Tint_55_24h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_55_24h] = sim ('Consumo', 2008);

Tot_Cons_55_24h = 0;

for i = 1:2009
    Tot_Cons_55_24h = Tot_Cons_55_24h + Cons_55_24h (i);
end
```

APPENDIX B (continued)

```
path = '70_18h.csv';

[ONOFF_70_18h,Tmand_70_18h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_70_18h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_70_18h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_70_18h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_70_18h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_70_18h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_70_18h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_70_18h(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_70_18h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_70_18h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_70_18h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_70_18h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_70_18h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_70_18h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_70_18h(i);
end

for i=1:287
    Tmand(i)=Tmand_70_18h(i);
    Tmand(i+287)=Tmand_70_18h(i);
    Tmand(i+574)=Tmand_70_18h(i);
    Tmand(i+861)=Tmand_70_18h(i);
    Tmand(i+1148)=Tmand_70_18h(i);
    Tmand(i+1435)=Tmand_70_18h(i);
    Tmand(i+1722)=Tmand_70_18h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_70_18h,Confronto5] = sim ('Tint', 2008);

Mean_Tint_70_18h = mean (Tint_70_18h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);
```

APPENDIX B (continued)

```
 [t,x,Cons_70_18h] = sim ('Consumo', 2008);

Tot_Cons_70_18h = 0;

for i = 1:2009
    Tot_Cons_70_18h = Tot_Cons_70_18h + Cons_70_18h (i);
end

path = '65_18h.csv';

[ONOFF_65_18h,Tmand_65_18h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_65_18h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_65_18h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_65_18h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_65_18h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_65_18h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_65_18h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_65_18h(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_65_18h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_65_18h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_65_18h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_65_18h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_65_18h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_65_18h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_65_18h(i);
end

for i=1:287
    Tmand(i)=Tmand_65_18h(i);
    Tmand(i+287)=Tmand_65_18h(i);
    Tmand(i+574)=Tmand_65_18h(i);
    Tmand(i+861)=Tmand_65_18h(i);
    Tmand(i+1148)=Tmand_65_18h(i);
    Tmand(i+1435)=Tmand_65_18h(i);
    Tmand(i+1722)=Tmand_65_18h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);
```

APPENDIX B (continued)

```
 [t,x,Tint_65_18h,Confronto6] = sim ('Tint', 2008);

Mean_Tint_65_18h = mean (Tint_65_18h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_65_18h] = sim ('Consumo', 2008);

Tot_Cons_65_18h = 0;

for i = 1:2009
    Tot_Cons_65_18h = Tot_Cons_65_18h + Cons_65_18h (i);
end

path = '60_18h.csv';

[ONOFF_60_18h,Tmand_60_18h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_60_18h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_60_18h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_60_18h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_60_18h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_60_18h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_60_18h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_60_18h(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_60_18h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_60_18h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_60_18h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_60_18h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_60_18h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_60_18h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_60_18h(i);
end

for i=1:287
    Tmand(i)=Tmand_60_18h(i);
    Tmand(i+287)=Tmand_60_18h(i);
    Tmand(i+574)=Tmand_60_18h(i);
    Tmand(i+861)=Tmand_60_18h(i);
    Tmand(i+1148)=Tmand_60_18h(i);
```

APPENDIX B (continued)

```
    Tmand(i+1435)=Tmand_60_18h(i);
    Tmand(i+1722)=Tmand_60_18h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_60_18h,Confronto7] = sim ('Tint', 2008);

Mean_Tint_60_18h = mean (Tint_60_18h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_60_18h] = sim ('Consumo', 2008);

Tot_Cons_60_18h = 0;

for i = 1:2009
    Tot_Cons_60_18h = Tot_Cons_60_18h + Cons_60_18h (i);
end

path = '55_18h.csv';

[ONOFF_55_18h,Tmand_55_18h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_55_18h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_55_18h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_55_18h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_55_18h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_55_18h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_55_18h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_55_18h(i);
end
for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_55_18h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_55_18h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_55_18h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_55_18h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_55_18h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_55_18h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_55_18h(i);
end
```

APPENDIX B (continued)

```
for i=1:287
    Tmand(i)=Tmand_55_18h(i);
    Tmand(i+287)=Tmand_55_18h(i);
    Tmand(i+574)=Tmand_55_18h(i);
    Tmand(i+861)=Tmand_55_18h(i);
    Tmand(i+1148)=Tmand_55_18h(i);
    Tmand(i+1435)=Tmand_55_18h(i);
    Tmand(i+1722)=Tmand_55_18h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_55_18h,Confronto8] = sim ('Tint', 2008);

Mean_Tint_55_18h = mean (Tint_55_18h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_55_18h] = sim ('Consumo', 2008);

Tot_Cons_55_18h = 0;

for i = 1:2009
    Tot_Cons_55_18h = Tot_Cons_55_18h + Cons_55_18h (i);
end

path = '75_13h(2spegn).csv';

[ONOFF_75_13h,Tmand_75_13h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_75_13h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_75_13h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_75_13h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_75_13h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_75_13h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_75_13h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_75_13h(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_75_13h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_75_13h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_75_13h(i);
```

APPENDIX B (continued)

```
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_75_13h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_75_13h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_75_13h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_75_13h(i);
end

for i=1:287
    Tmand(i)=Tmand_75_13h(i);
    Tmand(i+287)=Tmand_75_13h(i);
    Tmand(i+574)=Tmand_75_13h(i);
    Tmand(i+861)=Tmand_75_13h(i);
    Tmand(i+1148)=Tmand_75_13h(i);
    Tmand(i+1435)=Tmand_75_13h(i);
    Tmand(i+1722)=Tmand_75_13h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_75_13h,Confronto9] = sim ('Tint', 2008);

Mean_Tint_75_13h = mean (Tint_75_13h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_75_13h] = sim ('Consumo', 2008);

Tot_Cons_75_13h = 0;

for i = 1:2009
    Tot_Cons_75_13h = Tot_Cons_75_13h + Cons_75_13h (i);
end

path = '70_13h(2spegn).csv';

[ONOFF_70_13h,Tmand_70_13h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_70_13h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_70_13h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_70_13h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_70_13h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_70_13h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_70_13h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_70_13h(i);
end
```

APPENDIX B (continued)

```
for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_70_13h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_70_13h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_70_13h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_70_13h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_70_13h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_70_13h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_70_13h(i);
end

for i=1:287
    Tmand(i)=Tmand_70_13h(i);
    Tmand(i+287)=Tmand_70_13h(i);
    Tmand(i+574)=Tmand_70_13h(i);
    Tmand(i+861)=Tmand_70_13h(i);
    Tmand(i+1148)=Tmand_70_13h(i);
    Tmand(i+1435)=Tmand_70_13h(i);
    Tmand(i+1722)=Tmand_70_13h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_70_13h,Confronto10] = sim ('Tint', 2008);

Mean_Tint_70_13h = mean (Tint_70_13h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_70_13h] = sim ('Consumo', 2008);

Tot_Cons_70_13h = 0;

for i = 1:2009
    Tot_Cons_70_13h = Tot_Cons_70_13h + Cons_70_13h (i);
end

path = '65_13h(2spegn).csv';

[ONOFF_65_13h,Tmand_65_13h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_65_13h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_65_13h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_65_13h(i);
```

APPENDIX B (continued)

```
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_65_13h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_65_13h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_65_13h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_65_13h(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_65_13h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_65_13h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_65_13h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_65_13h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_65_13h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_65_13h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_65_13h(i);
end

for i=1:287
    Tmand(i)=Tmand_65_13h(i);
    Tmand(i+287)=Tmand_65_13h(i);
    Tmand(i+574)=Tmand_65_13h(i);
    Tmand(i+861)=Tmand_65_13h(i);
    Tmand(i+1148)=Tmand_65_13h(i);
    Tmand(i+1435)=Tmand_65_13h(i);
    Tmand(i+1722)=Tmand_65_13h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_65_13h,Confronto11] = sim ('Tint', 2008);

Mean_Tint_65_13h = mean (Tint_65_13h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_65_13h] = sim ('Consumo', 2008);

Tot_Cons_65_13h = 0;

for i = 1:2009
    Tot_Cons_65_13h = Tot_Cons_65_13h + Cons_65_13h (i);
end

path = '60_13h(2spegn).csv';

[ONOFF_60_13h,Tmand_60_13h] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);
```

APPENDIX B (continued)

```
for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_60_13h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_60_13h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_60_13h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_60_13h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_60_13h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_60_13h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_60_13h(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_60_13h(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_60_13h(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_60_13h(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_60_13h(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_60_13h(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_60_13h(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_60_13h(i);
end

for i=1:287
    Tmand(i)=Tmand_60_13h(i);
    Tmand(i+287)=Tmand_60_13h(i);
    Tmand(i+574)=Tmand_60_13h(i);
    Tmand(i+861)=Tmand_60_13h(i);
    Tmand(i+1148)=Tmand_60_13h(i);
    Tmand(i+1435)=Tmand_60_13h(i);
    Tmand(i+1722)=Tmand_60_13h(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_60_13h,Confronto12] = sim ('Tint', 2008);

Mean_Tint_60_13h = mean (Tint_60_13h);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_60_13h] = sim ('Consumo', 2008);

Tot_Cons_60_13h = 0;
```

APPENDIX B (continued)

```
for i = 1:2009
    Tot_Cons_60_13h = Tot_Cons_60_13h + Cons_60_13h (i);
end

path = 'D_Biclimatica (50-65).csv';

[ONOFF_65_Biclim,Tmand_65_Biclim] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_65_Biclim(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_65_Biclim(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_65_Biclim(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_65_Biclim(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_65_Biclim(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_65_Biclim(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_65_Biclim(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_65_Biclim(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_65_Biclim(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_65_Biclim(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_65_Biclim(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_65_Biclim(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_65_Biclim(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_65_Biclim(i);
end

for i=1:287
    Tmand(i)=Tmand_65_Biclim(i);
    Tmand(i+287)=Tmand_65_Biclim(i);
    Tmand(i+574)=Tmand_65_Biclim(i);
    Tmand(i+861)=Tmand_65_Biclim(i);
    Tmand(i+1148)=Tmand_65_Biclim(i);
    Tmand(i+1435)=Tmand_65_Biclim(i);
    Tmand(i+1722)=Tmand_65_Biclim(i);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_65_Biclim,Confronto13] = sim ('Tint', 2008);

Mean_Tint_65_Biclim = mean (Tint_65_Biclim);
```

APPENDIX B (continued)

```
x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_65_Biclim] = sim ('Consumo', 2008);

Tot_Cons_65_Biclim = 0;

for i = 1:2009
    Tot_Cons_65_Biclim = Tot_Cons_65_Biclim + Cons_65_Biclim (i);
end

path = 'D_Biclimatica (50-60).csv';

[ONOFF_60_Biclim,Tmand_60_Biclim] =
csvimport(path,'delimiter',';','column',[2,3],'outputAsChar',false,
'noHeader',true);

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_60_Biclim(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_60_Biclim(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_60_Biclim(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_60_Biclim(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_60_Biclim(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_60_Biclim(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_60_Biclim(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=Tmand_60_Biclim(i);
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=Tmand_60_Biclim(i);
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=Tmand_60_Biclim(i);
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=Tmand_60_Biclim(i);
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=Tmand_60_Biclim(i);
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=Tmand_60_Biclim(i);
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=Tmand_60_Biclim(i);
end

for i=1:287
    Tmand(i)=Tmand_60_Biclim(i);
    Tmand(i+287)=Tmand_60_Biclim(i);
    Tmand(i+574)=Tmand_60_Biclim(i);
    Tmand(i+861)=Tmand_60_Biclim(i);
    Tmand(i+1148)=Tmand_60_Biclim(i);
    Tmand(i+1435)=Tmand_60_Biclim(i);
    Tmand(i+1722)=Tmand_60_Biclim(i);
end
```

APPENDIX B (continued)

```
x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_60_Biclim,Confronto14] = sim ('Tint', 2008);

Mean_Tint_60_Biclim = mean (Tint_60_Biclim);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_60_Biclim] = sim ('Consumo', 2008);

Tot_Cons_60_Biclim = 0;

for i = 1:2009
    Tot_Cons_60_Biclim = Tot_Cons_60_Biclim + Cons_60_Biclim (i);
end

%let's evaluate the Control strategy based on the climatic graph
%(Tint=f(Text); the ONOFF is the same of one of the 24h Controls

for i=1:287
    ONOFF_V(i,1)=i;
    ONOFF_V(i,2)=ONOFF_70_24h(i);
    ONOFF_V(i+287,1)=i+287;
    ONOFF_V(i+287,2)=ONOFF_70_24h(i);
    ONOFF_V(i+574,1)=i+574;
    ONOFF_V(i+574,2)=ONOFF_70_24h(i);
    ONOFF_V(i+861,1)=i+861;
    ONOFF_V(i+861,2)=ONOFF_70_24h(i);
    ONOFF_V(i+1148,1)=i+1148;
    ONOFF_V(i+1148,2)=ONOFF_70_24h(i);
    ONOFF_V(i+1435,1)=i+1435;
    ONOFF_V(i+1435,2)=ONOFF_70_24h(i);
    ONOFF_V(i+1722,1)=i+1722;
    ONOFF_V(i+1722,2)=ONOFF_70_24h(i);
end

for i=1:287
    Tmand_V(i,1)=i;
    Tmand_V(i,2)=(Text(i)-2.6)*(-1.5217)+62.826;
    Tmand_V(i+287,1)=i+287;
    Tmand_V(i+287,2)=(Text(i)-2.6)*(-1.5217)+62.826;
    Tmand_V(i+574,1)=i+574;
    Tmand_V(i+574,2)=(Text(i)-2.6)*(-1.5217)+62.826;
    Tmand_V(i+861,1)=i+861;
    Tmand_V(i+861,2)=(Text(i)-2.6)*(-1.5217)+62.826;
    Tmand_V(i+1148,1)=i+1148;
    Tmand_V(i+1148,2)=(Text(i)-2.6)*(-1.5217)+62.826;
    Tmand_V(i+1435,1)=i+1435;
    Tmand_V(i+1435,2)=(Text(i)-2.6)*(-1.5217)+62.826;
    Tmand_V(i+1722,1)=i+1722;
    Tmand_V(i+1722,2)=(Text(i)-2.6)*(-1.5217)+62.826;
end

for i=1:287
    Tmand(i)=Tmand_V(i,2);
    Tmand(i+287)=Tmand_V(i,2);
    Tmand(i+574)=Tmand_V(i,2);
    Tmand(i+861)=Tmand_V(i,2);
```

APPENDIX B (continued)

```
    Tmand(i+1148)=Tmand_V(i,2);
    Tmand(i+1435)=Tmand_V(i,2);
    Tmand(i+1722)=Tmand_V(i,2);
end

x0 = iddata (TintS(:,2), [Text_V(:,2), Tmand_V(:,2)], 1);

x0est = findstates (oe221, x0);

[t,x,Tint_Clim,Confronto15] = sim ('Tint', 2008);

Mean_Tint_Clim = mean (Tint_Clim);

x1 = iddata (ConsI, [Tmand, ONOFF_V(:,2)], 1);

x1est = findstates (arx741, x1);

[t,x,Cons_Clim] = sim ('Consumo', 2008);

Tot_Cons_Clim = 0;

for i = 1:2009
    Tot_Cons_Clim = Tot_Cons_Clim + Cons_Clim (i);
end

%plot of the various Tint we have for each ONOFF strategy

figure
plot (ONOFF_V(:,1),Tint_70_24h, 'r',ONOFF_V(:,1), Tint_65_24h,
'b',ONOFF_V(:,1), Tint_60_24h, 'm',ONOFF_V(:,1), Tint_55_24h, 'g',
ONOFF_V(:,1), TintS(:,2), 'k', ONOFF_V(:,1), TintI(:,2), 'k')
legend ('70 degrees', '65 degrees', '60 degrees', '55 degrees')
title ('Tint estimation for a 24h-ON Control')
ylabel ('T[Celsius]')
xlabel ('Time')
hold on;
figure
plot (ONOFF_V(:,1),Tint_70_18h, 'r',ONOFF_V(:,1), Tint_65_18h,
'b',ONOFF_V(:,1), Tint_60_18h, 'm',ONOFF_V(:,1), Tint_55_18h, 'g',
ONOFF_V(:,1), TintS(:,2), 'k', ONOFF_V(:,1), TintI(:,2), 'k')
legend ('70 degrees', '65 degrees', '60 degrees', '55 degrees')
title ('Tint estimation for a 18h-ON Control')
ylabel ('T[Celsius]')
xlabel ('Time')
hold on;
figure
plot (ONOFF_V(:,1),Tint_75_13h, 'r',ONOFF_V(:,1), Tint_70_13h,
'b',ONOFF_V(:,1), Tint_65_13h, 'm',ONOFF_V(:,1), Tint_60_13h, 'g',
ONOFF_V(:,1), TintS(:,2), 'k', ONOFF_V(:,1), TintI(:,2), 'k')
legend ('75 degrees', '70 degrees', '65 degrees', '60 degrees')
title ('Tint estimation for a 13h-ON Control')
ylabel ('T[Celsius]')
xlabel ('Time')
hold on;
figure
plot (ONOFF_V(:,1),Tint_65_Biclim, 'r',ONOFF_V(:,1), Tint_60_Biclim,
'b', ONOFF_V(:,1), TintS(:,2), 'k', ONOFF_V(:,1), TintI(:,2), 'k')
legend ('65-50 degrees', '60-50 degrees')
title ('Tint estimation for a 24-ON Bi-Climatic Control')
ylabel ('T[Celsius]')
xlabel ('Time')
hold on;
```

APPENDIX B (continued)

```
figure
plot (ONOFF_V(:,1),Tint_Clim, 'r', ONOFF_V(:,1), TintS(:,2), 'k',
ONOFF_V(:,1), TintI(:,2), 'k')
legend ('Tint = f(Text)')
title ('Tint estimation for a Control where Tint is a function of Text')
ylabel ('T[Celsius]')
xlabel ('Time')
hold on;
figure
plot (ONOFF_V(:,1),Tmand, 'r')
legend ('Tmand')
title ('Tmand estimation using the climatic correlation')
ylabel ('T[Celsius]')
xlabel ('Time')

%display the gas consumption for each Control strategy

Tot_Cons_70_24h
Tot_Cons_65_24h
Tot_Cons_60_24h
Tot_Cons_55_24h
Tot_Cons_70_18h
Tot_Cons_65_18h
Tot_Cons_60_18h
Tot_Cons_55_18h
Tot_Cons_75_13h
Tot_Cons_70_13h
Tot_Cons_65_13h
Tot_Cons_60_13h
Tot_Cons_65_Biclim
Tot_Cons_60_Biclim
Tot_Cons_Clim
```

APPENDIX C: THE RADIATORS POWER

Radiators essentially work by heating the air that surrounds them via radiation and natural convection. The heat is carried around the radiator via a liquid medium, usually water, which itself is heated either by electricity (for self-contained heaters) or via the central heating and hot water system.

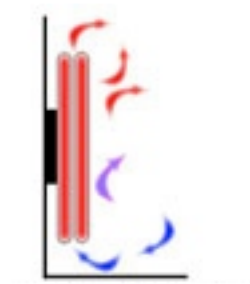


Figure 196: Panel Convector

Each radiator has a nominal power that it is able to deliver in ideal conditions. The radiators usually emit only a part of this power. This can be computed as follows:

$$P_{rad} = P_{nom} \cdot \left[\frac{\frac{T_{out} + T_{ret}}{2} - T_{in}}{60} \right]^{m}$$

Where:

- P_{rad} is the real power emitted by the radiator;
- P_{nom} is the radiator nominal power;
- T_{out} is the outlet temperature;
- T_{ret} is the return temperature;
- T_{in} is the temperature inside the building;
- m is a coefficient that depends on the radiator, usually $m = 1.3$;

APPENDIX C (continued)

- *60* is the reference temperature difference, which corresponds to the difference between the ideal outlet temperature (=80°C) and the typical indoor temperature (=20°C).

We know that a person produces energy of around 105Watt and that the presence of a person in a room doesn't really affect its temperature. Therefore we can state that if a radiator is emitting a power lower than 125W it is more or less like having a person inside a room. So when we see that the boiler is warming the water up to a temperature not sufficient to make the radiator power higher than 125W we just tell the system to turn the boiler off, so not to waste gas.

CITED LITERATURE

1. Aschemann, H., Kostin, G., Rauh, A., Saurin, V.: *Approaches to control design and optimization in heat transfer problems*. Moscow, ISSN1064-2307, Journal of Computer and System Sciences International, 2010.
2. Bae, C., Chun, C.: *Research on seasonal indoor thermal environment and residents' control behavior of cooling and heating systems in Korea*. Amsterdam, Elsevier, 2009.
3. Chen, J., Gu, G.: *Control-Oriented System Identification: An H_∞ approach*. New York: Jhon Wiley & Sons, 2000.
4. Chen, J., Wang, S.: *A Novel and simple load calculation model for building and system dynamic simulation*. Oxford, Pergamon Press, 2000.
5. Choi, S., Yoo, J., Lee, J., Shin, C., Park, M.: *Mathematical modeling and control of thermal plant in the district heating of Korea*. Amsterdam, Elsevier, 2010.
6. Freire, R., Oliveira, G., Mendes, N.: *Development of regression equations for predicting energy and hygrothermal performance of buildings*. Amsterdam, Elsevier, 2007.
7. Gambino, A.: *Diagnosi Energetica e Certificazione di Edifici: un approccio critico su due casi studio*. Turin, Politecnico di Torino: Energetic and Nuclear Engineering Thesis, 2009.
8. Ghiaus, C.: *Experimental estimation of Building Energy Performance by Robust Regression*. Amsterdam, Elsevier, 2005.
9. Giarré, L., Milanese, M., Taragna, M.: *H_∞ identification and model quality evaluation*. IEEE Transactions on Automatic Control, vol AC-42, no. 2, pp 118-197, 1997.

CITED LITERATURE (continued)

10. Givoni, B.: *Minimum climatic information needed to predict performance of passive buildings in hot climates*, pp. 197–202. Brisbane, Proc. of the Sixteenth International Passive and Low Energy Architecture Conference (PLEA'99), Australia, 1999.
11. Givoni, B., Krüger, E.: *An attempt to base prediction of indoor temperatures of occupied houses on their thermo-physical properties*. Santiago, Chile, Proc. of the Eighteenth International Passive and Low Energy Architecture Conference (PLEA'03), vol. 3, pp. 727–731, 2003.
12. Gustafsson, S., Rönnqvist, M.: *Optimal heating of large block of flats*. Amsterdam, Elsevier, 2008.
13. Haberl, J., Culp, C.: *Review of Methods for Measuring and Verifying Savings from Energy Conservation Retrofits to Existing Buildings*. Texas A&M University, Paper, 2005.
14. Jiménez, M., Madsen, H.: *Models for describing the thermal characteristics of building components*. Amsterdam, Elsevier, 2006.
15. Kaipio, J., Somersalo, E.: *Statistical Inverse Problems: Discretization, model reduction and inverse crimes*. Amsterdam, Elsevier, 2005.
16. Krisanee, M.: *Predictions of Monthly Energy Consumptions and annual plattens of Energy Usage for Convenience by using multiple and nonlinear regression models*. Texas A&M University, Architecture Thesis, 2004.
17. Krüger, E., Givoni, B.: *Predicting thermal performance in occupied dwellings*. London, Energy and Buildings 36 (3) pp. 301–307, Mendeley, 2004.

CITED LITERATURE (continued)

18. Lee, K., Braun, J.: *Evaluation of Methods for determining demand-limiting setpoint trajectories using short-term measurements.* Amsterdam, Elsevier, 2007.

19. Liao, Z., Dexter, A.: *An inferential model-based predictive control scheme for optimizing the operation of boilers in building space-heating systems*. New York, IEEE, 2009.

20. Ljung, L.: *System Identification: Theory for the user*, 2nd edition. Upper Saddle River, Prenctice Hall, 1999.

21. Ljung, L.: *System Identification Toolbox: For Use with MATLAB® user's guide*. Natick, The MathWorks, 1997.

22. Ma, J., Qin, J., Salsbury, T., Xu, P.: *Demand reduction in building energy systems based on economic model predictive control*. Amsterdam, Elsevier, 2011.

23. Mahdavi, A.: *Simulation-based control of building system operation*. Amsterdam, Pergamon, Elsevier, 2001.

24. Mustafaraj, G., Lowry, G., Chen, J.: *Prediction of room temperature and relative humidity by autoregressive linear and nonlinear neural network models for an open office*. Amsterdam, Elsevier, 2011.

25. Ogonowski, S.: *Modeling of the heating system in small building for control*. Amsterdam, Elsevier, 2010.

26. Olofsson, T., Andersson, S., Sjögren, J.: *Building Energy Parameter Investigations based on Multivariate Analysis*. Amsterdam, Elsevier, 2008.

27. Papst, L., Lamberts, R.: *Thermal performance evaluation of three houses in Floriano´polis, south of Brazil*. Proc. of the Eighteenth International Conference on Passive and Low Energy Architecture, pp. 293–297, PLEA'01, 2001.

CITED LITERATURE (continued)

28. Petersen, S., Svendsen, S.: *Method for simulating predictive control of building system operation in the early stages of building design*. Amsterdam, Elsevier, 2011.
29. Prívara, S., Široký, J., Ferkl, L., Cigler, J.: *Model predictive control of a building heating system: The first experience*. Amsterdam, Elsevier, 2010.
30. Široký, J., Frauke, O., Cigler, J., Prívara, S.: *Experimental analysis of model predictive control for an energy efficient building heating system*. Amsterdam, Elsevier, 2011.
31. Virk, G., Loveday, D.: *Model-based control for hvac applications,* pp. 1861–1866. Glasgow, Proc. of The Third IEEE Conference on Control Applications, 1994.
32. Yao, J., Zhu, N.: *Evaluation of indoor thermal environmental, energy and daylighting performance of thermotropic windows*. Amsterdam, Elsevier, 2011.
33. Yildirim, N., Macit, T., Gökçen, G.: *District heating system design for a university campus*. Amsterdam, Elsevier, 2005.
34. Zhang, J., Xu, Y., Li, J.: *Integrated system identification and reliability evaluation of stochastic building structures*. Amsterdam, Elsevier, 2011.
35. ISO 9004:2009. Geneva: 2009.
36. <http://consumatori.myblog.it/archive/2011/11/10/riscaldamento-autonomo-o-centralizzato.html>
37. <http://efficienzaenergetica.acs.enea.it/doc/dlgs_192-05.pdf>
38. <http://efficienzaenergetica.acs.enea.it/doc/dlgs_311-06.pdf>

.

VITA

NAME :

Federico Picca

EDUCATION :

Laurea di I livello in Mechanical Engineering, Politecnico di Torino, 2009

Laurea specialistica in Mechanical Engineering, Politecnico di Torino, 2011

Master of Science in Mechanical Engineering, University of Illinois at Chicago, 2012