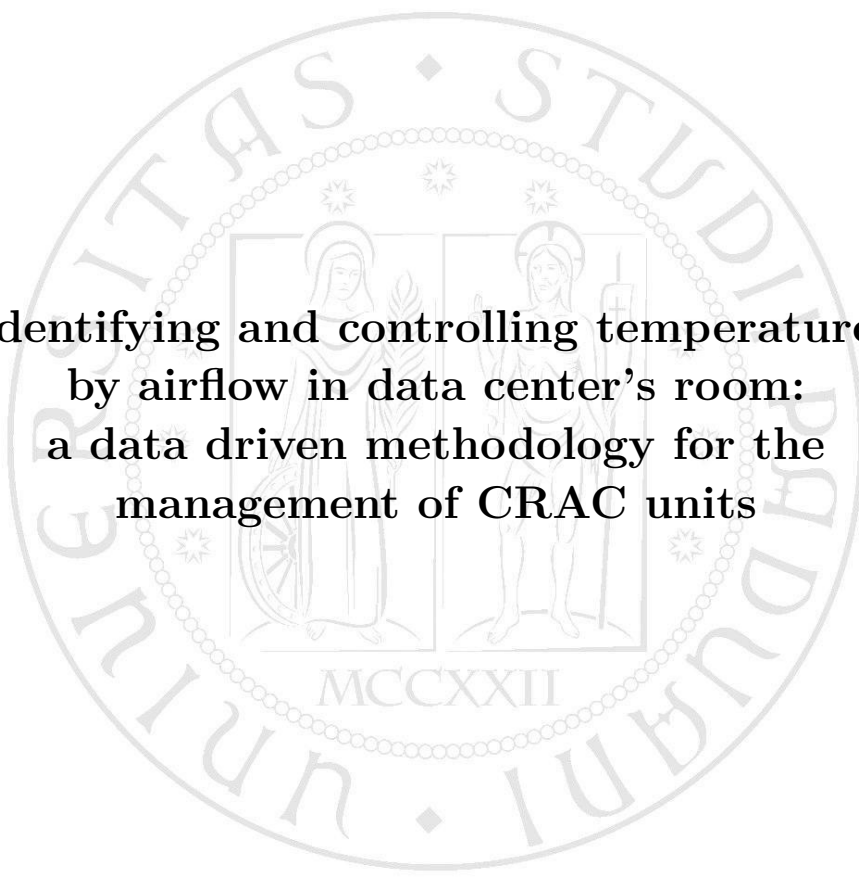


University of Padova

---

DEPARTMENT OF INFORMATION ENGINEERING

*Master's degree in Automation Engineering*

The background features a large, faint watermark of the University of Padova seal. The seal is circular and contains the Latin text 'UNIVERSITAS STUDII PADOVAE' around the perimeter and 'MCCXXII' at the bottom. In the center, there are two figures, likely saints, standing under a gothic arch.

**Identifying and controlling temperatures  
by airflow in data center's room:  
a data driven methodology for the  
management of CRAC units**

*Author*

**Emanuele Simonazzi**

*Supervisor*

**Prof. Ruggero Carli**

---

ACADEMIC YEAR 2017/2018



Identifying and controlling temperatures by  
airflow in data center's room:  
a data driven methodology for the  
management of CRAC units

Emanuele Simonazzi

September 11, 2018



# Acknowledgements

Vorrei anzitutto ringraziare chi mi ha accompagnato lungo questo progetto: Prof. Ruggero Carli, Damiano Varagnolo, Winston Garcia Gabin e gli impiegati del Research Institutes of Sweden (RISE) Swedish Institute of Computer Science (SICS). Speciale menzione va riconosciuta ai miei colleghi/amici Riccardo e Michele, che mi hanno supportato sia fuori che dentro all'ufficio.

Impossibile dimenticare chi invece mi ha accompagnato e sostenuto per tutto il mio percorso universitario aiutandomi a non fermarmi mai. Ringrazio quindi tutta la mia famiglia, per essermi stata vicina con affetto e consigli. I miei amici di Poggio, che mi supportano (e sopportano) sin da quando eravamo piccoli. I coinquilini di casa Maroncelli e Flacco con i quali ho condiviso i migliori momenti della vita universitaria a Padova. I miei compagni di corso i quali da colleghi sono diventati fidati amici. Ed infine Carmen con la quale ho condiviso i dolori e soprattutto le gioie di questa lunga esperienza (diventando di volta in volta l'una la spalla dell'altro).

Ancora grazie quindi a tutti voi perché se questo giorno è qui arrivato non è solo merito mio, ma anche vostro.



# Abstract

Data centers are fundamental infrastructures for the digital world that we know today. To work, however, they use a high amount of energy and this consumption is expected to grow in the future. The purpose of this thesis is studying and finding solutions that would allow to decrease their energy impact. Specifically it attempts to address a critical issue that occurs in air cooled data centers when both cooling and exhaust air flows mix within the computer rooms, this issue is due to poor design or to non-efficient management and operation of the cooling infrastructure. When these mixes occur, indeed, the energetic efficiency of the cooling operations drops.

So, in order to improve the air cooling efficiency in this thesis we focused on the minimization of cooling and exhaust air flows mixing, using control on fans speed of the cooling infrastructure, avoiding in that way to install an opportune hardware which is the most expensive solution to these kind of problems.

The following work is so developed: after a brief introduction, methods to detect where and when flow mixings happen are discussed and proposed; then strategies are built in order to identify, from field data, models that can help detecting whether these mixing events occur or not, on top of classical Prediction Error Methods (PEM) approaches; in the end, Model Predictive Controllers are designed for the operation of the cooling infrastructure of a data center, exploiting the identified models, mentioned above, and compare them against model free Proportional Integrative Derivative (PID) control strategies. The results are then based on field tests performed in an industrial-scale air-cooled datacenter with an installed capacity of 240 kW.





# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	The structure of a data center, in general terms . . . . .	3
1.2	Temperature control and the flow mixing phenomenon problem	4
1.3	Thesis collocation . . . . .	5
1.4	Literature review . . . . .	7
<b>2</b>	<b>Motivation</b>	<b>9</b>
2.1	Working mechanisms of air cooled data centers . . . . .	9
2.2	Flow mixing phenomena . . . . .	11
<b>3</b>	<b>Structure and study</b>	<b>17</b>
3.1	SICS' data center . . . . .	17
3.1.1	Sensors and actuators . . . . .	18
3.2	Identification of areas of interest . . . . .	20
3.2.1	Identify in which zones it is relevant to detect flows mixing phenomena . . . . .	21
3.2.2	Listing and/or clustering important sensors and actu- ators available in the plant . . . . .	21
3.2.3	Executing air flow experiments and finding the most relevant signals . . . . .	23
3.3	Determining regions of different provisioning . . . . .	28
<b>4</b>	<b>Modelling and identification</b>	<b>31</b>
4.1	Introduction . . . . .	31
4.2	SISO models . . . . .	35

4.3	MISO models . . . . .	39
4.4	Results . . . . .	44
<b>5</b>	<b>Air flow control</b>	<b>45</b>
5.1	Introduction . . . . .	45
5.2	PID . . . . .	46
5.2.1	PID theory . . . . .	46
5.3	MPC . . . . .	47
5.3.1	MPC theory . . . . .	49
5.3.2	The objective function for our Single Input Single Output (SISO) models . . . . .	50
5.4	Implementation . . . . .	51
5.5	Simulation . . . . .	52
5.6	Results . . . . .	54
<b>6</b>	<b>Conclusions and future works</b>	<b>57</b>
	References . . . . .	58

# Abbreviations

ARX	Auto-regressive with eXogenous input
BJ	Box-Jenkins
CFD	Computational Fluid Dynamics
CT	Cooling Technology
HVAC	Heating, Ventilation of Air Conditioning
IT	Information Technology
CRAH	Computer Room Air Handling
CRAC	Computer Room Air Cooling
SCADA	Supervisory Control And Data Acquisition
RISE	Research Institutes of Sweden
SICS	Swedish Institute of Computer Science
LS	Least Square
LTI	Linear Time Invariant
LQR	Linear Quadratic Regulator
MIMO	Multi Input Multi Output
MISO	Multi Input Single Output
MPC	Model predictive control
P&ID	Piping and Instrumentation Diagram
PEM	Prediction Error Methods
PID	Proportional Integrative Derivative
SISO	Single Input Single Output



# Chapter 1

## Introduction

In nowadays' connected world, data centers have become essential. They are facilities composed of networked computers and storage that businesses or other organizations use to organize, process, store and disseminate large amounts of data. With so much information exchange and with developing countries increasingly eager to exploit the possibilities of an information-based society, the number and dimension of data centers is constantly growing - as much as their electricity demand.

This increasing energy consumption due to the exchange and storage of information has been a driving force behind a progressive improvement in the cooling systems of data centers. Since their efficiency is yet not at the physical limit, for this reason it has been decided to develop this project, dedicated to improving data center's cooling system operations through dedicated data-driven modelling and development of model-based control strategies.

### **1.1 The structure of a data center, in general terms**

Data centers are generally composed by an Information Technology (IT) and a Cooling Technology (CT) infrastructure. The first one comprises mainly the computer systems, i.e., servers, storage devices, and network hardware. This group is the core of the data center and it provides services to the users.

While working to meet all the users' needs, the large energy consumption of this infrastructure generates a huge amount of heat. The CT system (that is also associated to a big energy consumption for its operation) solves the heat rejection problem by taking this heat out of the facility. This infrastructure includes server fans, Computer Room Air Cooling (CRAC), chillers, and cooling towers. By using these tools the temperature inside the room is kept constant, as discussed in more details below.



Figure 1.1: Picture of the interior of a data center room in Dallas.

## 1.2 Temperature control and the flow mixing phenomenon problem

In 2016 the data centers around the world used roughly 416 Terawatts (or about 3% of the total electricity), nearly 40% more than the entire United Kingdom, and this consumption is expected to double every four years. Up to 40% of that electrical power were consumed by CT system. It is obvious how developing room cooling control systems that keep the room temperature steady while minimizing the energetic inefficiencies is fundamental to save money and produce less environmental pollution.

The most commonly employed cooling systems is air cooling: in this case the CT works by transforming the warm air of the room into cold air, trying to keep the temperature within the computer room at a predetermined temperature. During this operation, hot air and cold air may mix up. This phenomenon, that will be studied in deep in the following sections, can lead to uncertainties in the temperature measured inside the data center, a non-ideal response of the cooling infrastructure, and to an overall non-linear behaviour of the thermal dynamics within the data center. As consequence of this, the energy consumption will increase. Therefore, before continuing, it is important for us to study in detail these mixing phenomena and make educated decisions about how to control the cooling infrastructure based on the results.



Figure 1.2: Aerial view of the Facebook data center in Luleå, Sweden.

### 1.3 Thesis collocation

The purpose of this work is to develop data-driven strategies to understand how air flow mixing phenomena happen inside a generic computer room.

These will then constitute the basis for the design of cooling room temperature control algorithms. The section below this one shows how different working regions exist within a data center room. They depend on the amount of flow mixing generated by different value of the cooling system's fans speed. Considering these different working regimes, chapter 4 contributes by designing data-driven linear modeling strategies that can be used to simulate how the computer room temperature behaves in each of these regions depending on the operating conditions of the cooling infrastructure. In Chapter 5 these models are used to design Model predictive control (MPC) strategies for the management of CRAC systems. These strategies are then compared against a model free controller PID, so that the thesis focuses also in comparing different operation strategies, trying to understand which one leads to better energy consumption.

We notice that, as for the air control, usually it consists in keeping the cooling system's airflow constant, and changing only its temperature. Driven by the intuitions developed in the last two years by the thesis' supervisors, in this thesis we instead control the room temperature through changing the airflow of the cooling system (i.e., of the CRACs) while keeping the temperature of the output flow constant. This method is innovative because in literature there are few evidence of temperature and flow mixing control only through CRAC fans speed. Therefore our duty will be to give a greater formalism to this new approach that tries controlling the room temperature through taking into account the flow mixing phenomena.

To introduce also the conclusions that we drawn through working on this specific problem, we can say that in our opinion flow mixing phenomena should always been taken into account when studying how to both design and operate a data center room. Identifying these phenomena, moreover, can be useful for better defining the room temperature model, which may in its turn benefit the performance for the controller. However, we leave more details on what we just said in this paragraph to Section 6.



## 1.4 Literature review

In literature we can find many examples of data center models. Fluid dynamics studies had given a great contribution, for example: [1] proposes a model that predicts transient air temperatures in an air-cooled data center, [2] presents a heat flow model which uses temperature and ambient sensors to characterize the hot air recirculation, based on these informations, and accelerates the thermal evaluation process for high performance datacenters. What is missing in these papers and mostly in all fluid dynamics study is the control part that we decided to introduce in this thesis. An example of model designed for control it is given by [3] [4]. The data center room studied was the same of our project. In their study, linear and non-linear models of the room temperature were designed relying on the servers temperature; after that, was developed a direct control (Linear Quadratic Regulator (LQR)) of the CPU temperatures, but with drawback big non-linear phenomena caused by the servers' fans. In our case, instead, we have decided to use a smaller and simpler model, that excludes the servers temperature, in order to avoid non-linearity phenomena.

Concerning the control, [5], for example, introduces a system based on CRAC control. A distributed sensor network is used to increase the awareness/information on data center state; the strategy is based on simple PID controllers. An example on how to tune them can be seen in [6] where an optimal self tuning controller is presented, it combines, for a server fan cooling system, a PID neural network (PIDNN) and a fan-power-based optimization in the transient-state temperature response in time domain. PIDNN with a time domain criterion is used to tune all online and optimized PID gains.

As above mentioned, the intention of this study is to use PID only as starting point for the development of more complex controller, as in [7]. Indeed, LQR and MPC were designed to control the data center room temperature. Other ideas on how to design a MPC can be found in [8] and [9] which implement a stochastic MPC to control the servers' temperatures. All of these examples were taken into account when the controllers were designed for this project. For completeness of information, not only the technique proposed is

used to cool down the data center temperature, e.g. [10] defines 13 different basic heat removal methods, classifying them as direct air (our case) and as indirect air methods relied on the outdoor conditions.

# Chapter 2

## Motivation

### 2.1 Working mechanisms of air cooled data centers

As mentioned in Section 1.1, different data centers are cooled using different types of cooling systems. The most commonly technologies used are CRACs (see also Figure 2.1), that use refrigerants to cool the air flowing through them, or Computer Room Air Handling (CRAH) units, that use instead chilled water to achieve the same result. These two types of units can work in both direct output or raised floor system configurations. A typical example of a data center with a direct output configuration of the cooling infrastructure is shown in Figure 2.2. With this configuration there are CRACs on both sides of the computer room, and racks in the middle, dividing in this way the room into separate hot and cold aisles. Racks (as the one in Figure 2.3) are used to contain several servers by stacking them in ad-hoc metallic structures. The rack shown in Figure 2.2 is for example 2.1 meters high and may contain up to 30/40 servers. To create homogeneity in the air flows, all the servers have the same air flow direction. Thus, servers' fans push hot air into the hot aisle, that then rises to the ceiling and goes inside the CRAC unit as the sketch of Figure 2.5. Usually, CRACs have fans on their top, and these fans push the air down through a chiller tube where the refrigerant is flowing. The cooled air exits then the CRACs and travels up to the front of the servers,



Figure 2.1: Example of a typical CRAC unit.



Figure 2.2: Example of a data center with hot aisle (central panel) and cold aisles (right and left panels) configuration.

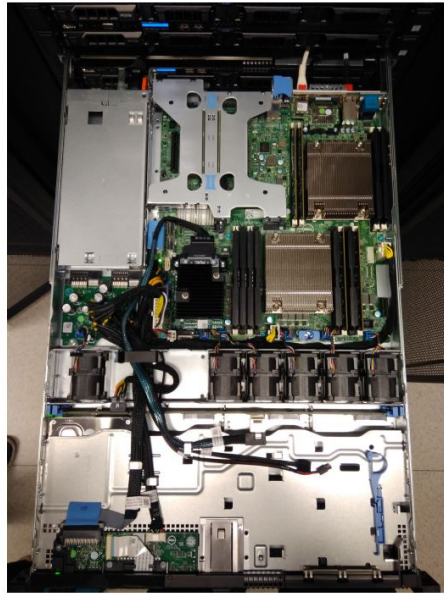


Figure 2.3: A typical Dell R430 server.

enters them, becomes warmer, and in this way closes the cycle that continues indefinitely.

In the raised floor case the CRAC units push the air directly into a cavity in the floor that arrives up to where the servers inlets are mounted in the room. With raised floors the phenomena will be the same as with the direct output configuration; however, in this case the flow direction will be different since coming from the bottom.

Both cooling configurations can be affected by flow mixing phenomena. This phenomena arises when hot air and cold air meet and create turbulences. The latter as we will see create several problems, especially regarding the temperature control of the room. Is purpose of this thesis try to explain how that phenomena happen and how to solve it.

## 2.2 Flow mixing phenomena

In the past, HVAC mixing applications did not demand strict performance requirements. As long as the air-handling unit did not shut down, or a water

coil did not freeze due to stratification, the mixing was assumed to be satisfactory. However, with the advent of more precise control systems and the higher needs for energy efficiency, the problems of an improper management of the air flows has become more pronounced. Regarding this problem, we notice that air temperature and flow sensor readings may be greatly affected by stratification problems in the mixed air plenum [11]. So, for example we can have problems with air, gas or liquid mixing phenomena.

In this project the attention focused on air flow mixing phenomena within a data center room endowed with classic configurations of direct cold flows from the CRACs towards the servers inlets. In this configuration, flow mixing phenomena in the cold aisle is likely to occur, specially if improper air containment mechanisms are implemented – something that happens often in mid sized data centers. Indeed in these cases there is often no strategy for mitigating air leakage from the cold to the hot aisle and viceversa. For this reason the hot exhaust air going towards the CRAC could mix with the cold stream already within the cold aisle. For new generation's data centers the solution adopted is to use new hardware, building for example a hot aisle enclosure as in Figure 2.4. For old or cheaper installations, instead, the only solution is to study this phenomenon and try to mitigate it using data driven identification and control methods, as done below.

The main problem when the flow mixing occurs are room's temperature measures. Indeed, if the room temperature is measured, for example, on the top of a rack in the cold aisle side and we have a flow mixing phenomena in that region, it happens that the temperature in that point is different from the one that would be without mixing. Evidence, that will be presented later on in the thesis, shows that there may exist differences in the measurements up to  $+5^{\circ}C$  between the two cases. For example, in a standard working case with cold air equal to  $+21^{\circ}C$ , and hot air coming from the hot aisle equal to  $+35^{\circ}C$ , we can understand how flow mixing can change the measure of a the sensor on the top, mentioned before, that instead of detect only the cold air temperature it measures a mix between both of them. As result, it will present an higher value than the one expected.

From a control point of view this is not acceptable, since this difference

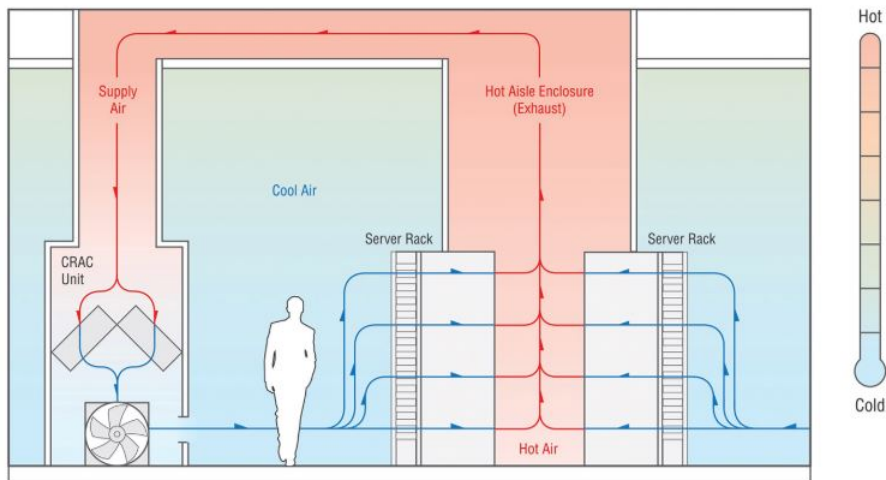


Figure 2.4: Classic example of a typical CRAC unit with hot aisle enclosure. From <https://journal.uptimeinstitute.com>

caused by the flow mixing phenomenon changes the temperature measurements that are sometimes used as reference signals. This offset, thus, causes an extra job for the actuators and disturb the functioning of the system. The consequence is, at the end, an higher and useless energy consumption.

For the reasons above it is important to study flow mixing phenomena: how they manifest, what are their root causes, and how we can mitigate their influence on the performance of the cooling system. In our studying real-world systems, we detected several types of flow mixings, that we cluster in 3 main groups:

- *underprovisioning phenomena*: here hot air is mixed with the cold one coming out from the CRAC (i.e., more air is needed to reach the reference temperature);
- *overprovisioning phenomena*: here cold air bounces against the racks and goes back into the CRAC (i.e., there is a waste of cooling energy);
- *standard provisioning situations*: here there is no air mixing, so that cold air enters directly into the servers and hot air goes directly to the top of the CRAC. This is the best situation possible, corresponding to

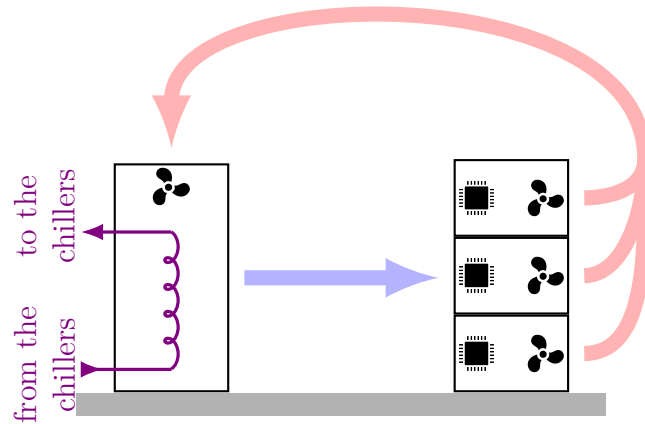


Figure 2.5: Schematic representation of the computer room while in a standard provisioning situation. Here hot air goes directly into the CRAC, so that the temperatures sensors in the cold aisle are not affected by flow mixing phenomena.

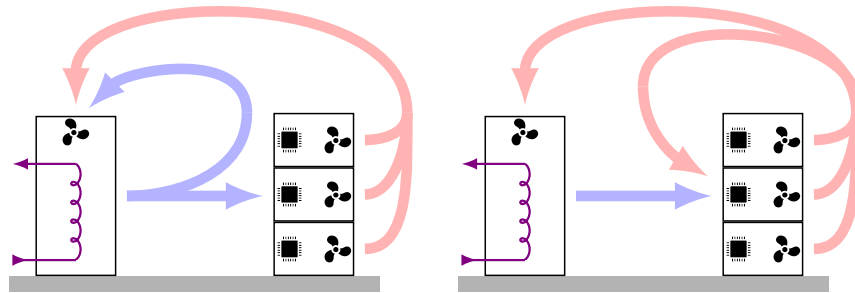


Figure 2.6: Schematic representation of the situations corresponding to CRAC units overprovisioning cooling flows (in the left panel) or underprovisioning them (right panel).

the nominal behavior that the cooling system should have during its design phase.

In the next sections we will explain all the procedures that we developed for detecting these phenomena, and how to mitigate their influence through opportune control strategies.

To conclude this section, we also notice that it is clear how sensors in different positions can lead to different detection capabilities of the provisioning phenomena described above. In other words, it is fundamental to



choose the correct placement of the sensing infrastructure. We thus continue the thesis with a concise description of the data center room used for our field experiments. This will allow us to describe in more details the sensors positioning problem and the provisioning detection issues.



# Chapter 3

## Structure and study of a real data center

In this chapter will be explained the real system used for the experiments, and the study done to define how and where the flow mixing phenomena affects the data center room.

### 3.1 SICS' data center

Tests and studies have been performed at the Swedish Institute of Computer Science (SICS) ICE facility. The access has been provided by Research Institute of Sweden (RISE) SICS North, a research center in Luleå(SE). The institute owns and operates two data centers; the one employed in this project is module 2, i.e., the one that was already shown in Figure 2.2.

This infrastructure is composed by 4 SEE Cooler HDZ CRACs, and 10 racks with approximately 250 servers in them in total. Within racks 1 to 3 the computer room presents *HPE BladeSystem c7000 Enclosures*, containing 32 servers each. Incidentally, these servers are the most powerful servers within module 2, and this means that from an airflow point of view the computer room cannot be considered having geometrical symmetries. The other servers mounted in the other racks are *Dell poweredge r430* units (see also Figure 2.3) and *Dell poweredge r530* units. These servers are mounted

in racks 4 to 10, for a total of 164 units.

To complete the picture we notice that each server has 4 to 5 fans for aiding the air flow within the servers enclosure. Importantly, these fans operate commanded by servers' native control algorithms that work independently from the room's system. In other words, for each server the fans are controlled by their own controller; since there is no coordination among the various cooling systems, these local fans can lead (as we will see later on) to non-linear dynamics plus practical issues on the whole system identification procedure. Notice that modelling, identification and control of local fans are problems not related with the scope of this thesis and thus will not be taken into account in this study [3]. For a comprehensive study on the issues and experiments on the field we send the reader back to [8] and [9] theses.

We won't focus on servers fans because we want to work with simpler system. Indeed, it has been decided to focus our attention on the cold aisle area, and more precisely modelling the phenomena happening between CRACs and racks. To link our models with a servers fans' model will be a future topic.

We now proceed with considering that the status of the computer rooms is typically sensed through many sensors; before proceeding we thus describe the Supervisory Control And Data Acquisition (SCADA) system employed to sense and control our testbed.

### 3.1.1 Sensors and actuators

The knowledge of what is measured, where sensors are positioned, and of what actuators can actually do to steer the system is fundamental, since this is the starting point for any detection and solution of our control problems. This is especially true for understanding and controlling the flow mixing phenomena described before – for example, it is crucial to understand at which height mixings are happening and what they affect. Therefore, knowing the sensors location is essential (see also Figure 3.1).

In our testbed the sensing infrastructure can be summarized with:

- *Temperature sensors:*

- there are 3 for each side of each Rack: on the top, in the middle and at the bottom;
  - one sensor is placed on the ceiling of each aisle – in this case it is even more important to know the exact location of the cold aisle sensor because, as hinted before, it can change the sensibility of the flow mixing detection.
  - an additional output temperature sensor is present for each CRAC in the middle of its outlet air area;
  - we can also sense the temperature of the inlet and outlet refrigerant (that, incidentally, is composed by 70% of water and 30% of Ethylene Glycol);
- *Fans speed*: for each CRAC we can collect its fans speed (in % or *rpm*);
  - *Electrical power*: almost all the power produced by the room's equipment can be collected. In this thesis the focus will be only on the CRACs power consumption.

The last screening process that must be done at this stage is the description of the actuation infrastructure. Since CRACs will be the only actuators that we can use to solve or at least manage the problem. So, also in this case, it is fundamental to map their location. In this thesis will be used:

- CRAC fans speed =  $u_1$ ;
- IT load =  $u_2$ .
- Refrigerant temperature =  $u_3$ .

Where  $u_2$  and  $u_3$  will be kept constant for all the experiments and only  $u_1$  will be used to control the temperature inside the room.

Once this preliminary work has been done, the next step is to make tests, identify and visualize the problem.

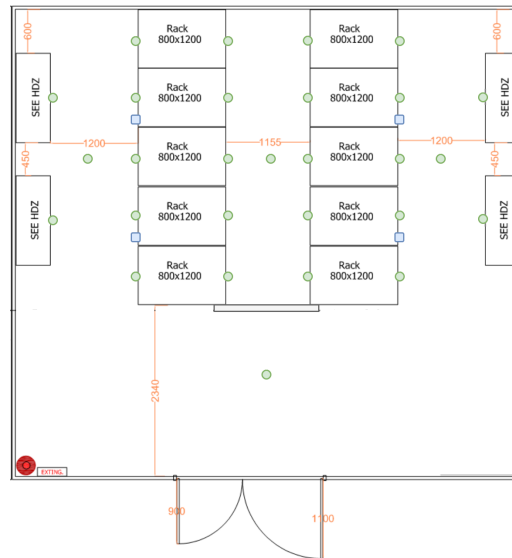


Figure 3.1: Temperature sensors position in Module 2. Each green dot attached to the Racks corresponds to 3 sensors respectively on top, middle and bottom of them. Blue squares are sensors not taken into account for this project.

### 3.2 Identification of areas of interest

To detect the areas of interest it has been decided to design an a-doc methodology that could be used for other experiments with different systems structure. For an exhaustive explanation we send the reader back to [12]. The methodology used is:

- Identify in which zones it is relevant to detect flows mixing phenomena;
- Listing and/or clustering the most important sensors and actuators available in the plant;
- Executing air flow experiments and find the most relevant signals;
- Determining the regions where the flow models are approximately linear.

A further explanation of what done at each step of the list is shown in subsections below.

### 3.2.1 Identify in which zones it is relevant to detect flows mixing phenomena

All the computer rooms have different designs, that could leads to different flow mixing regions. In this specific computer room, after its inspection it has been clear that the flow mixing was in the cold aisles. In that areas cold flow and hot flow meet generating air turbulence. They could be schematized as in Figure 2.5 and Figure 2.6. Where as already mentioned in Section 2.2, three different groups have been detected: *underprovisioning*, *overprovisioning*, *standardprovisioning*.

After this inspection it is fundamental to study the system through sensors and actuator.

### 3.2.2 Listing and/or clustering important sensors and actuators available in the plant

Usually people think that more sensors are in the system and better it is. The statement is correct, but it brings an issue. Indeed, too many sensors are difficult to study individually. For example, for each cold aisle there are almost 20 temperature sensors. Being this study oriented to a model and control implementation, this number is to high – it is impossible try to control or model so many sensors. Therefore, after sensors and actuators identification of Section 3.1.1 is advisable to understand how to use them. So, on the purpose of studying them a good solution is to cluster sensors in different groups, in order to find the one that better explains the phenomena and that can be used as controller's reference signal.. In this case, sensors in cold aisle are more important because, as said after the inspection, they are in the region where air recirculation happen.

Clustering sensors is also important to have a bigger view of the system and mitigate measures errors that could arise for several reasons e.g. sensor fault, flow mixing etc.. Since the two sides of the room are almost specular, was decided to consider just one of them in order to simplify the job according to server availability, the only difference is represented by the servers' power.



Figure 3.2: Clustered sensors within cold aisle. The image refers to the right side, but nothing change with the left side used.

It was chosen the left side between rack 1-5 which, as already mentioned, it has more powerful servers and they can give us a wider choice on  $u_2$ . The clusters are:

- $T_1$ : mean of all the rack input temperature sensors between rack 1-5. Where input temperature means the flow temperature that is going inside the rack – from the cold aisle side;
- $T_2$ : mean of all top rack input temperature sensors between rack 1-5, blue line in Figure 3.2;
- $T_3$ : mean of middle rack input temperature sensors between rack 1-5, orange line in Figure 3.2;
- $T_4$ : mean of bottom rack input temperature sensors of one side, green line in Figure 3.2;



- $T_5$ : Crac output temperatures, they can be approximated all equals, red line in Figure 3.2;
- $T_6$ : cold aisle temperature, purple dot in Figure 3.2;
- $T_7$ : hot aisle temperature.

After signals clustering it is time to make the first experiment and understand something more about the system.

### 3.2.3 Executing air flow experiments and finding the most relevant signals

Choosing the correct experiments to do is important. Indeed, sometimes a bad one can lead to a waste of time or, worse, we could not understand that is a bad experiment and arrive to false conclusions. The final result will be the project failure. So, to do not incur in this types of problems, and to find different type of provisioning linked to the air flow, it has been decided to use an inverse *steps-test* of the CRACs fans speed, between  $u_1 = 90 - 30\%$  (for data center's safety the fans never go under 30% speed) as in Figure 3.3. Each step is 10 *min* long, usually enough for the system to reach or almost to reach the thermal equilibrium.

It seemed to be the best test choice because it could explain how the temperature grows related to CRACs fans decreasing. Also, It can help to visualize different provisioning regions finding their thresholds. We will see it later in Section 3.3.

Being the goal to stress the system only by CRAC fans the rest of the actuators have been left constant:

- $u_2 = 50\%$ ;
- $u_3 = 18^\circ C$ .

Actually, several  $u_2$  have been tested as 0, 25, 50, 75, 100, one for each test. But, only 50% will be shown for many reasons: first, step responses of the temperatures have always the same behaviour, only the gain changes.

Second, being not enough time to study all of them, it has been decided to focus only on one IT load. From all of them it has been chosen 50% because it is a good trade off between stress of the system (heat production) and the needed of a certain  $\Delta T = T_7 - T_5$  used to visualize better the flow mixing. For instance, if  $\Delta T = 1^\circ C$ ,  $u_1$ , used to keep the reference temperature constant, will be small. In this way it will be difficult to study flow mixing with high CRAC fans speed. On the contrary, it has been decided to do not use values greater than 50 % to do not stress too much the system too much.

Two experiments for each  $u_2$  have been done to compare the data and check repeatability. It is also a good solution design smaller steps where there are feelings or evidences of a region's provisioning change. This solution has been applied as can be seen in Figure 3.3.

In Figure 3.2.3 can be appreciated the temperatures behaviour of the *steps-test*

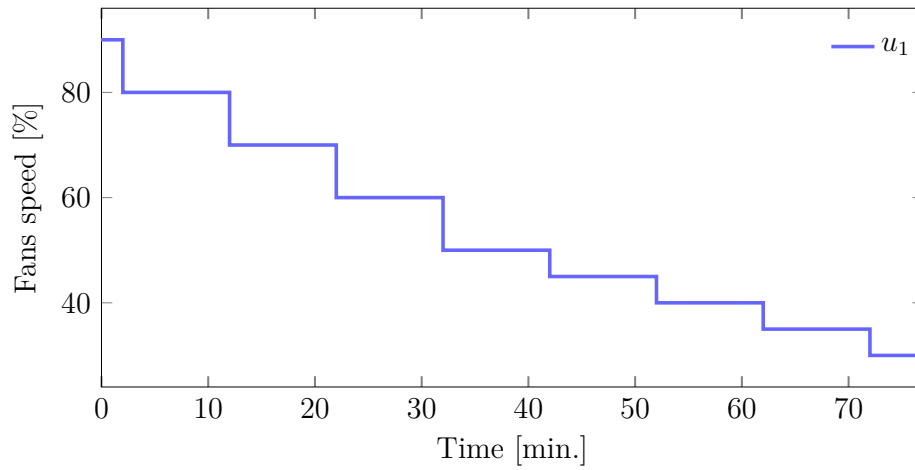
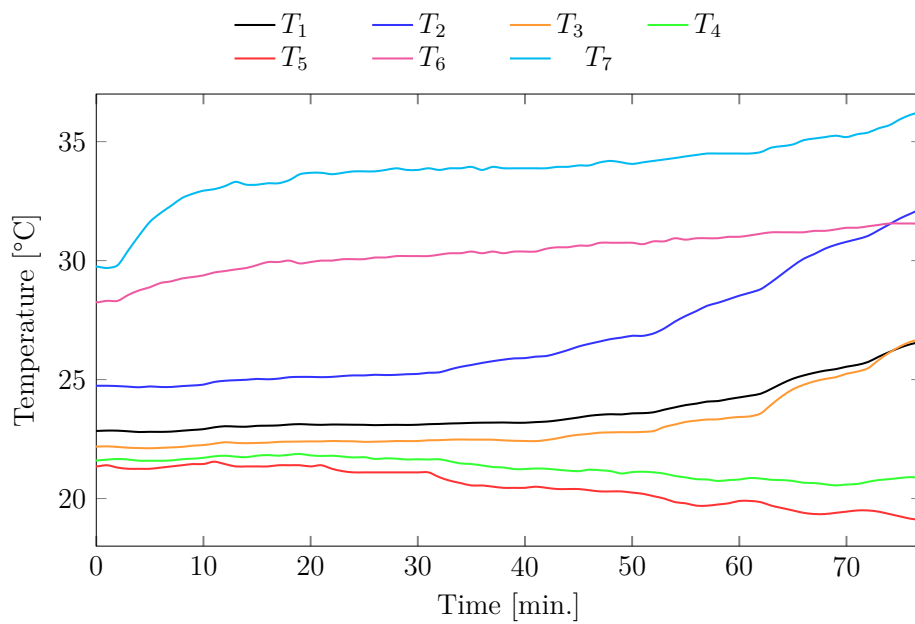
### 3.2.3.1 Identifying the most relevant signals through correlation analysis

Once all the signals have been collected (in  $\mathcal{D} = \{u_1, \dots, u_3, T_1, \dots, T_7\}$  dataset) and visualized, can be beneficial a step of pruning this set of potential  $u$ 's and  $T$ 's. In this way only that components having an actual information content can be detected. Although more advanced approaches may be performed, it has been decided to start this pruning step by performing a correlation analysis, more precisely, given the dataset  $\mathcal{D}$  of synchronized time-series with every component in this dataset being a specific signal. Then what can happen is that: two distinct inputs  $u_i$  ( $T_i$ ) and  $u_j$  ( $T_j$ ) are highly correlated, say, e.g.,

$$\left| \frac{\mathbb{E}[(u_i - \mathbb{E}[u_i])(u_j - \mathbb{E}[u_j])]}{\text{st.dev}(u_i) \text{st.dev}(u_j)} \right| \geq 0.95. \quad (3.1)$$

but this is not the case because three actuators are independent between each other.

A similar concept can be applied to couples outputs  $T_i$  and  $T_j$ : if there

Figure 3.3: *Stairs-test*,  $u_1$  between 90% – 30%.Figure 3.4: *Stairs-test*, temperatures comparison. See Section 3.2.2

exist highly correlated outputs then it may be meaningful to ignore one of the two. In this case, the most correlated signals are  $T_1$  and  $T_3$ , but since  $T_1$  is the mean of all the rack input temperatures and it could be approximated as the input temperature located almost in the middle of the racks where  $T_3$  is located. For this reason only  $T_3$  has been taken into account.

It can also happen that a certain  $T_i$  is simultaneously uncorrelated to all the various inputs  $u_j$  and others  $T_s$ . In this case this may be seen as an indication that there *may* not be extractable information from the sensor (something that in any case can be double-checked during the data-driven modelling step). This is the case of  $T_6$  which can be seen as almost constant. This sensor will not be useful as flow mixing detector, but, as it will be explained below, it will be useful as model variable because it can catch the IT load variation (not the case of this test) by increasing or decreasing the hot flow temperature coming from the hot aisle.

### 3.2.3.2 Most relevant signal choice

The last step before determining the different provisioning regions is to choose the signal that better explain the phenomena. The last signals to be studied after the correlation step are in Figure 3.5.

As it can be seen, two different groups can be detected: *decreasing signals* and *increasing signals*. The first one is composed by  $T_4$  and  $T_5$ .  $T_4$  does not give significant information on provisioning phenomena because it does not have appreciable changes during the test so it will be discarded.  $T_5$ , instead, looking at Figure 3.5 seems to be a perfect candidate because it follow the steps in input. But, it has three big problems: first, the sensor is inside the CRAC so it can not detect flow mixing; second, it decreases when  $u_1$  decreases and this in a room temperature control is counter-intuitive; third, the sensor is too close to the actuator  $u_1$  and this makes the control too sensible. Also  $T_5$  will be discarded.

Then, we study *increasing signals*  $T_2$ ,  $T_3$ ,  $T_7$ . The last one has a non-linear behaviour and also it is in a position where there is no flow mixing so it will not be used.  $T_2$  and  $T_3$  are the most interesting signals, they have similar

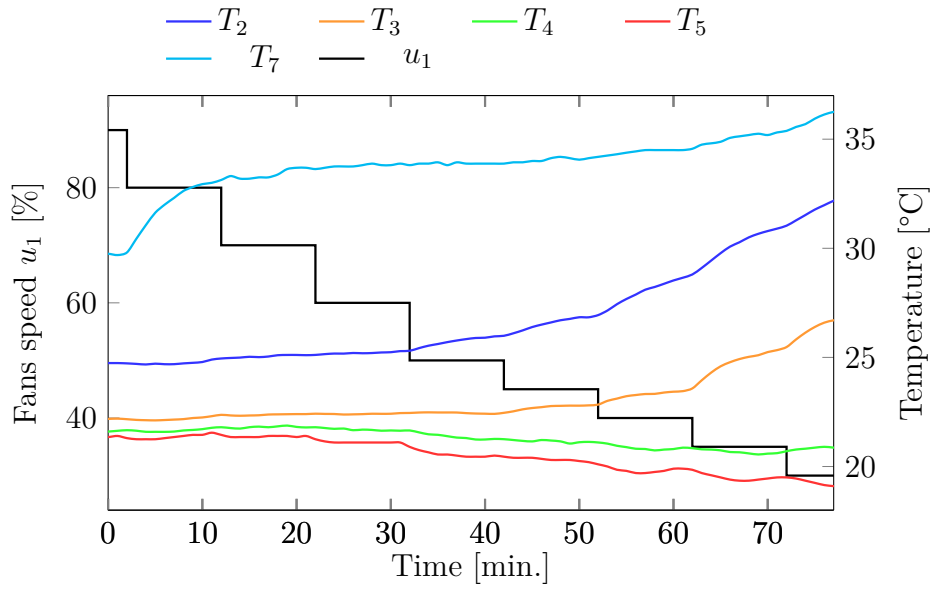


Figure 3.5: Last temperatures comparison. *Stairs-test*

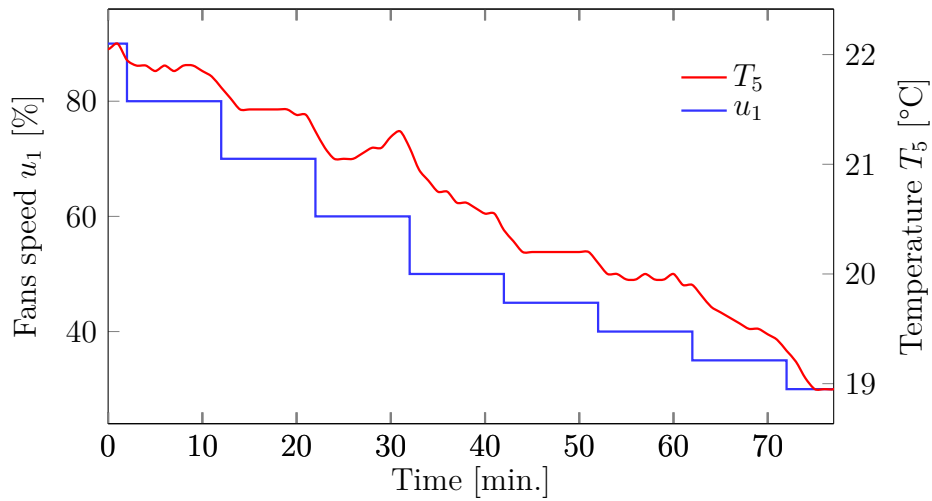


Figure 3.6: Comparison of  $T_5$  and  $u_1$  during *stairs-test*.

dynamics (Figure 3.7), they are in a perfect position for our modelling and control purpose, and as it can be seen they have regions of linear behaviour. For this reason the last choice between  $T_2$  and  $T_3$  will be taken into account only after a better study of them as done below.

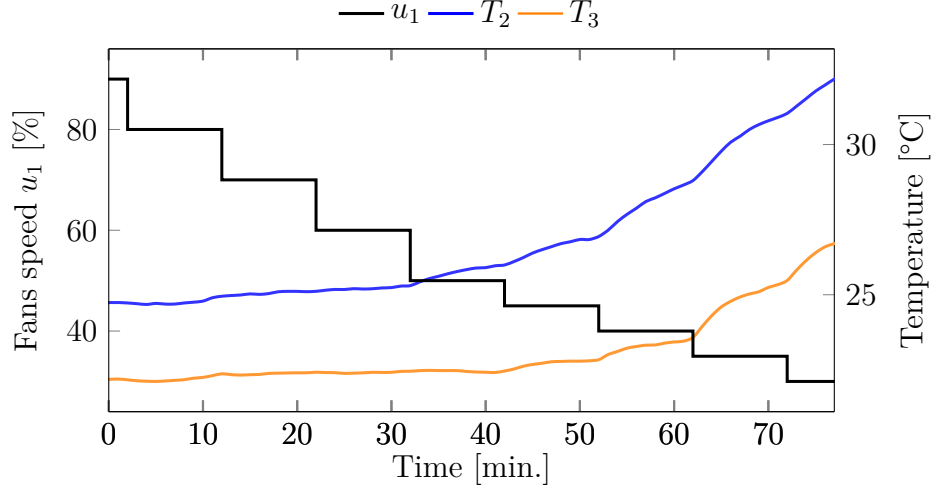


Figure 3.7: Comparison of  $T_2$ ,  $T_3$  and  $u_1$  during *stairs-test*.

### 3.3 Determining regions of different provisioning

The key point of this thesis is to identify when different provisioning of the cooling flows happen and obtain data-driven models that can forecast the thermal dynamics of the room when these phenomena occur.

Referring to Figure 2.5 and Figure 2.6 for an intuitive explanation, these events happen depending on the values of the air pressure field within the computer room (something that is in its turn affected by the speeds of the various fans rotating within the computer room).

As explained in Section 2.2 three different areas have been detected. The intuition suggests that the gains will typically be much smaller when the CRAC are overprovisioning the cooling flows. This implies that these input-output gains are expected to be clearly different, depending on the flows

	$T_2$	$T_3$
<b>Overprovisioning</b>	$u_1 = 90 - 60\%$	$u_1 = 90 - 50\%$
<b>Standardprovisioning</b>	$u_1 = 60 - 45\%$	$u_1 = 50 - 40\%$
<b>Underprovisioning</b>	$u_1 = 45 - 30\%$	$u_1 = 40 - 30\%$

Table 3.1: Provisioning regions detected for  $T_2$  and  $T_3$ 

region. This eventually implies that from the *stairs-test* it should be immediately possible to not only compute the input-output gains of the system for the various operating conditions, but also verify for which operating conditions the system experiences a shift from various types of provisioning.

Therefore, looking at Figure 3.7 and checking each gain of each step response the provisioning regions detected are as in Table 3.1. As it can be seen in the table, usually with  $T_2$  the new provisioning region start a step before  $T_3$ . Obviously, it happens because  $T_2$ 's sensors are in a higher position than  $T_3$  and they are more affected by the flow mixing caused by hot aisle's air flow.

So, two useful temperatures have been detected, but with only this elements there are no evidences for preferring one to the other. For this reason the final choice will be done only after their model identification, where will be chosen only the one that has better linear behaviour.





# Chapter 4

## Modeling and identifying air provisioning phenomena in air-cooled data centers

### 4.1 Introduction

As we saw in the previous chapter, it is possible to detect that the air-conditioning system induces different flow conditions. The purpose of this section is to design a strategy to identify a series of Linear Time Invariant (LTI) data-driven models of these different flow conditions, using classical PEM based system identification strategies.

For the sake of completeness, PEM is a framework used in system identification to estimate the parameters of dynamic systems having either standard parametric model structures (like ARX, BJ, OE, or ARMAX) or also non-standard and non-linear models. The estimate is done through maximizing the predictive performance of the parameters of these models. More precisely, the parameters estimation step is performed using data collected after one or more field test on the real system.

To be more mathematically rigorous, we consider in this thesis that a

typical model structure is

$$\mathbf{y}(k) = \mathcal{F}(z)\mathbf{u}(k) + \mathcal{G}(z)\mathbf{e}(k), \quad (4.1)$$

with the generic parameters vector indicated with  $\theta = [f_1, \dots, f_{n_F}, g_1, \dots, g_{n_G}]$ . Due to its simplicity and typical effectiveness, in our case we will estimate  $\theta$  using the Least Square (LS) principle [13]. Given a regression model  $\mathbf{y}(k) = g_k(u(k), \theta) + \mathbf{e}(k)$  with  $\mathbf{e}$  a random innovation vector with zero mean and covariance matrix  $\sigma^2 R$ ,  $k = 1, \dots, N$ , and saying also that  $N$  is the number of samples available for the estimation step, the estimated parameters will be

$$\hat{\theta}_{LS}(y) = \arg \min_{\theta \in \Theta} \frac{1}{\sigma^2} \|y - g(u, \theta)\|_{R^{-1}}^2. \quad (4.2)$$

The system identification procedure that will be adopted in this thesis is thus shown in Figure 4.1 and it is composed by the following logical steps:

1. *Data preprocessing*: when performing estimation tasks, data usually have to be preprocessed for many reasons. For example, they can be filtered, specially if we are interested to model the system only in certain frequency (e.g., when the measurements are subject to high-frequency noise we could then use a low-pass filter to "clean" the signal from this noise). An other option is that we can de-trend the data if we notice that there exists a trend causing some kind of distortion (e.g., a non-zero mean or some slow ramp due to seasonal effects). The most important and fundamental preprocessing when dealing with modeling linear systems is to re-center the data around the desired equilibrium; in other words, given an equilibrium point with input  $u_0$  and output  $y_0$  we shall de-offset the data around that point using the transformations  $\tilde{u}(k) = u(k) - u_0$  and  $\tilde{y}(k) = y(k) - y_0$ ;
2. *Model structure design*: when possible, a priori information (typically under the form of some more or less detailed physics laws) may be used to guess  $n$  potential alternative model structures  $\mathcal{M}_1, \dots, \mathcal{M}_n$ , that shall all be checked to see which one gives the best predictive performance in relevant test sets. In our case there was no a priori information giving

clear indication to which alternative we shall use. For this in this thesis we took all the 4 model structures listed above into account;

3. *Training step*: using a dataset or a part of it (usually its 70%) and applying it to  $\mathcal{M}_1, \dots, \mathcal{M}_n$  we compute the estimates  $\hat{\theta}_1, \dots, \hat{\theta}_n$ . In this way we obtain the "candidate" models  $\mathcal{M}_1(\hat{\theta}_1), \dots, \mathcal{M}_n(\hat{\theta}_n)$ ;
4. *Validation step*: using a second dataset or the second part of the one used during the training step, we can choose the best model among  $\mathcal{M}_1(\hat{\theta}_1), \dots, \mathcal{M}_n(\hat{\theta}_n)$  which better describe the system. Latter choice is done using bias variance tradeoff. It says that a model must both accurately captures the regularities in its training data, but also generalizes well to unseen data. Considering both ways it is almost impossible, then the model chosen will be the one with a better tradeoff between them.

Below we will present two different type of models: SISO and Multi Input Single Output (MISO). It has been decided to consider these alternative models to provide comparisons that may work as a baseline for future derivations and to have the possibility of checking different control configuration strategies.

Models have been trained and tested with a dataset divided as explained in point 3 and 4 above, during the test only  $u_1$  (CRAC fans speed) changes with  $u_2 = 50\%$  (IT load) and  $u_3 = 18^\circ C$  (refrigerant temperature) kept constant. Each test has been repeated for each provisioning region changing  $u_1$  upper and lower limit. Unfortunately, for time issues we could not test both  $T_2$  and  $T_3$  regions. So in the end will be presented tests only on  $T_2$  regions (see Table 3.1).  $T_3$  will be still tested and it will be useful to understand whether this regions hypothesis is confirmed or not. A test example for the overprovisioning region can be seen in Figure 4.2.

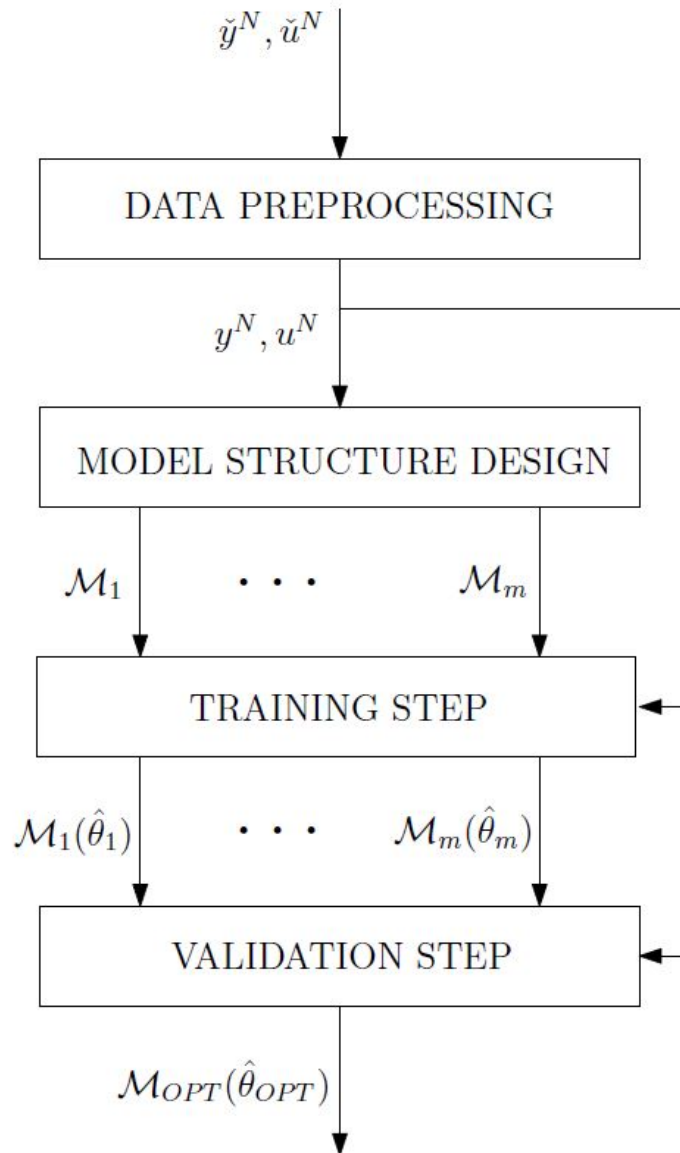


Figure 4.1: Graphical representation of the logical steps used in this thesis to identify the best model.

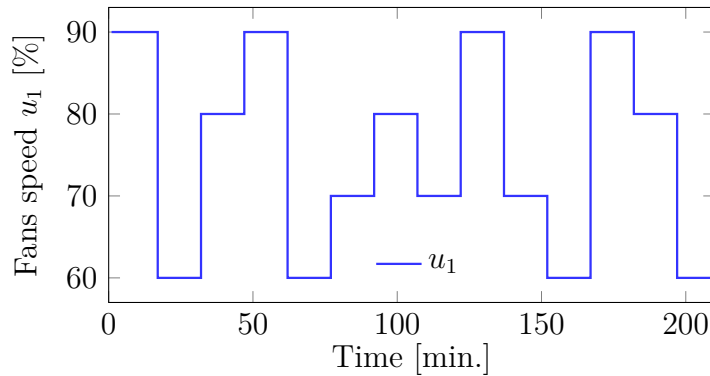


Figure 4.2: Overprovisioning test with  $u_1$  between 90 % – 60 %.

## 4.2 SISO models for air provisioning phenomena

With only one input and one output, SISO is the simplest model. Keeping  $u_2$  and  $u_3$  ideally constant and being the output  $(T_2, T_3)$  only affected by input  $u_1$ , SISO model can still be a good solution even if too simple. This models will be also used as starting point for future models.

We can start giving new notations to outputs that we are going to test:

$$\begin{aligned} y_1 &= T_2, \\ y_2 &= T_3, \end{aligned}$$

and to our input  $u = u_1$ , too. As said 3 different tests have been done, one for each region of Table 3.1. They produced three models one for each provisioning region.

Before the *training step* is good to notice as in Figure 4.3  $y_1$  has a growing trend so, as explained before a good choice is to detrend the data and after that start with the parameters estimation. Data detrended can be seen in Figure 4.4 .

As second step we decided to test all the model structures, and after *training* and *validation step* we found for both 3 models a discrete-time Box-

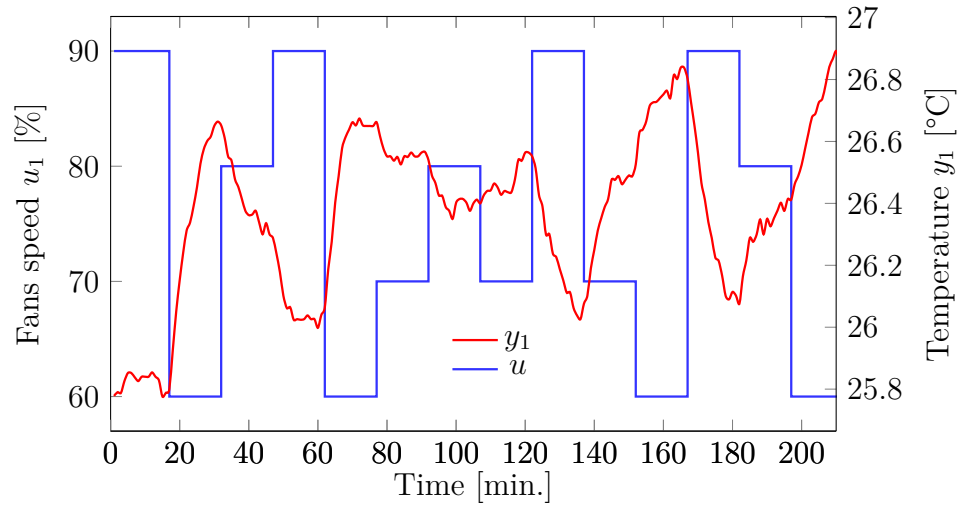


Figure 4.3: Evidence of a growing trend of  $y_1$ .

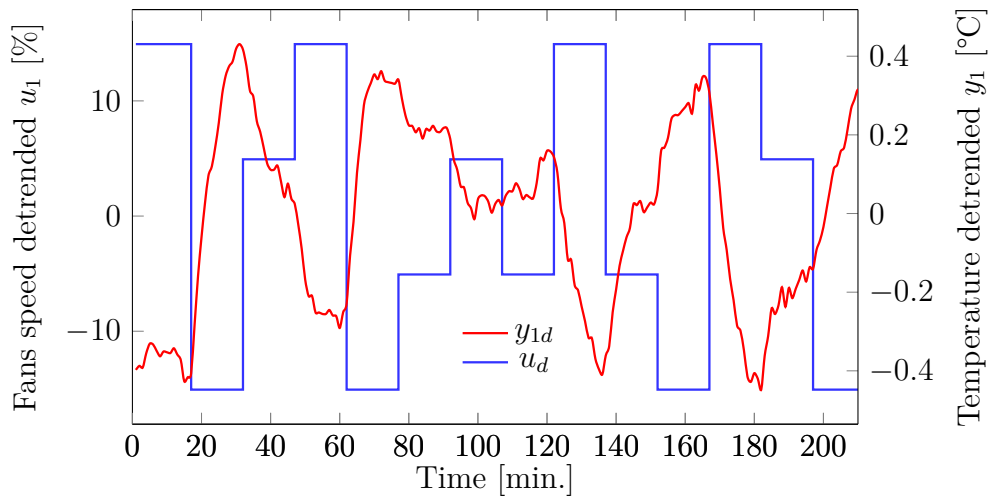


Figure 4.4: Data detrended with MATLAB's function *detrend*.

Jenkins (BJ) structure:

$$\mathbf{y}(t) = \frac{B(z)}{F(z)}\mathbf{u}(t) + \frac{C(z)}{D(z)}\mathbf{e}(t) \quad (4.3)$$

with sampling time equal to 1 *min.*, and  $A$ ,  $B$ ,  $C$ ,  $D$  composed by the estimated parameters, shown below.

$$\Omega_o \ni \begin{cases} B(z) = -0.001 - 0.002z^{-1} + 0.004z^{-2} \\ C(z) = 1 - 0.139z^{-1} + 0.103z^{-2} - 0.047z^{-3} \\ D(z) = 1 - 1.139z^{-1} + 0.139z^{-2} \\ F(z) = 1 - 1.75z^{-1} + 0.758z^{-2} \end{cases} \quad (4.4)$$

$$\Omega_s \ni \begin{cases} B(z) = -0.004 + 0.003z^{-1} \\ C(z) = 1 + 0.836z^{-1} - 0.164z^{-2} \\ D(z) = 1 - 0.159z^{-1} - 1.267z^{-2} + 0.08z^{-3} + 0.258z^{-4} + 0.088z^{-5} \\ F(z) = 1 - 1.936z^{-1} + 1.366z^{-2} - 0.632z^{-3} + 0.424z^{-4} - 0.181z^{-5} \end{cases} \quad (4.5)$$

$$\Omega_u \ni \begin{cases} B(z) = -0.003 - 0.012z^{-1} - 0.003z^{-2} + 0.01z^{-3} + 0.01z^{-4} \\ C(z) = 1 - 1.35z^{-1} + 0.5z^{-2} - 0.057z^{-3} - 0.0278z^{-4} - 0.011z^{-5} \\ D(z) = 1 - 2.507z^{-1} + 1.671z^{-2} + 0.51z^{-3} - 0.933z^{-4} + 0.259z^{-5} \\ F(z) = 1 - 1.027z^{-1} - 0.29z^{-2} - 0.052z^{-3} + 0.636z^{-4} - 0.239z^{-5} \end{cases} \quad (4.6)$$

with  $\Omega_o$  = overprovisioning region,  $\Omega_s$  = standardprovisioning region, and  $\Omega_u$  = underprovisioning region.

Validation plots can be seen in Figure 4.5. With this second part of the tests almost all the models follow the real behaviour of their  $y_1$ . The less representative model is in  $\Omega_s$  region, where between  $t = 20 - 40$  min  $y_1$  has a non-linear behaviour that let fails the model input response. This non linearity occurs mostly with  $u = 50\%$ , we can suppose it happens because we are close to the transition point with  $\Omega_u$ . Therefore, we can expect that some different turbulences are happening and that the too simple model can not detect them. Further results will be shown below in Section 4.4.

To show that with  $y_2$  we do not have the correct dataset, as said before, is enough to show Figure 4.6 with  $\Omega_o$  region. In this case we can see how

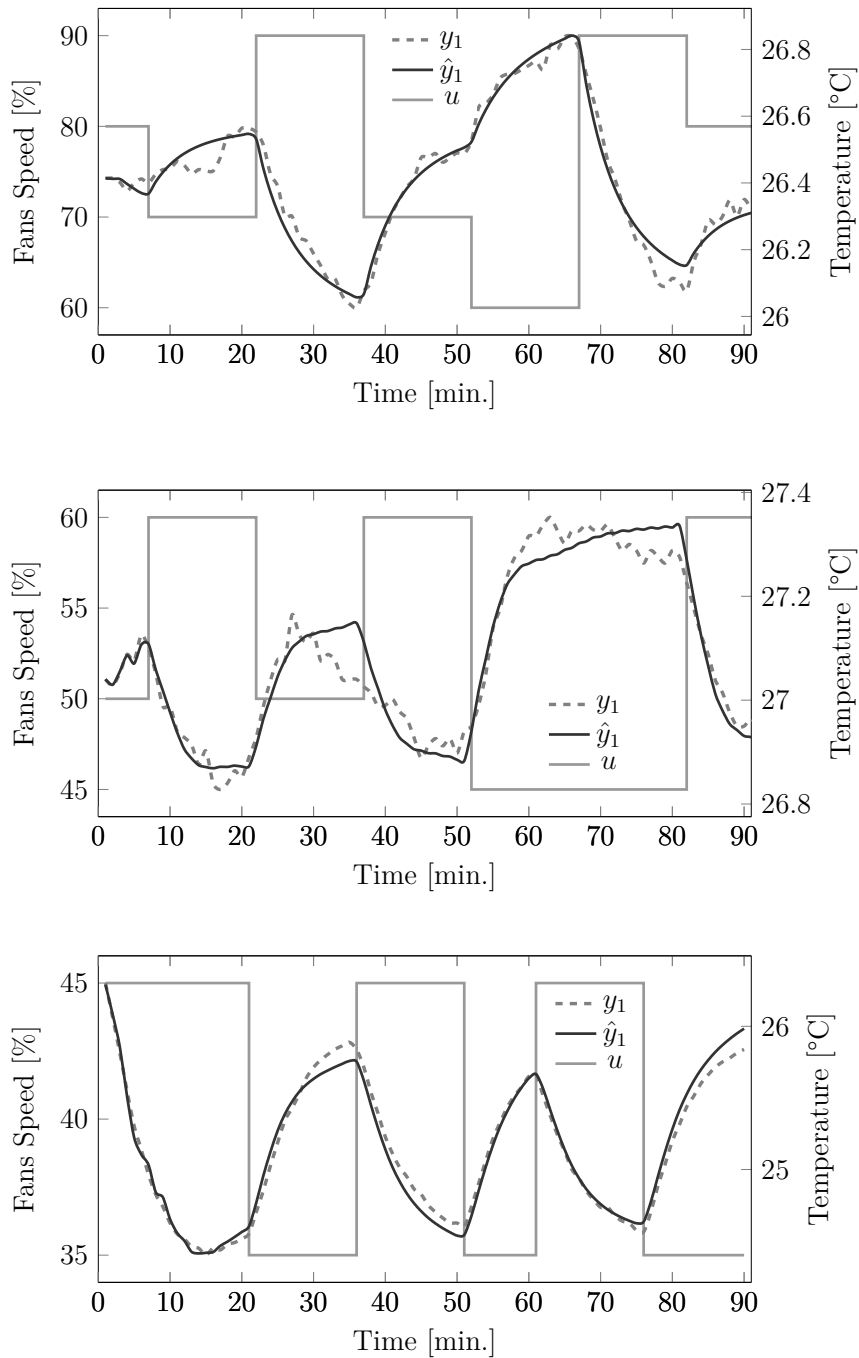


Figure 4.5: *Validation test* plot with  $y_1$  real signal and  $y_{1e}$  model output signal (estimated). Starting from the top we'll have respectively: overprovisioning, standard provisioning, underprovisioning.



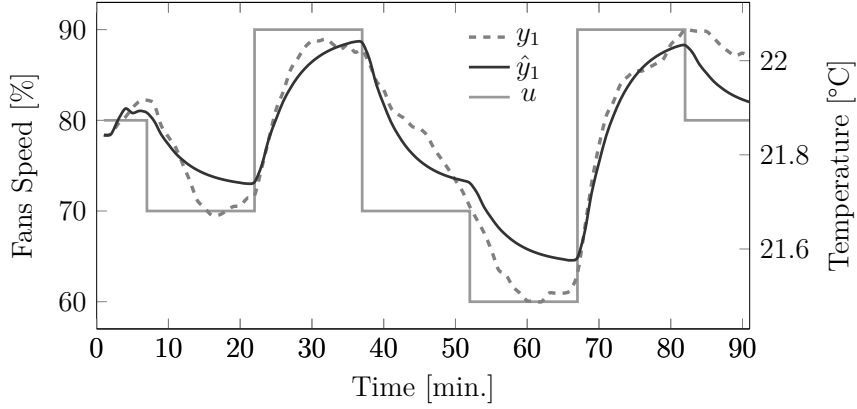


Figure 4.6: Validation test with  $u \in \Omega_o$  and  $y_2$  as output.  $y_{1e}$  performance not acceptable.

the model completely fails the gain in many points e.g. with  $u = 60\%$ . This reason pushed us to choose models with output  $y_1$ .

More elaborated models will be shown in section below where new variables have been added.

### 4.3 MISO models for air provisioning phenomena

We want to try to design a more complex model that can give better forecast. We decided then to add more than one input to the model and what we got is a MISO model. Using the same test of Section 4.2 we decided to use 3 model's inputs:  $u_1$ ,  $T_6$ ,  $T_r$  (refrigerant temperature inside the CRACs pipes). If we define:

$$T_{rin} = \text{CRAC inlet refrigerant temperature,}$$

$$T_{rout} = \text{CRAC outlet refrigerant temperature,}$$

we get  $T_r = \frac{T_{rin} + T_{rout}}{2}$ .

When  $u_1$  changes it has been detected as also  $T_r$  changes see Figure 4.7 as example. This variation means that also the flow outlet temperature of

the CRAC changes and this is fundamental to forecast the output, that is directly affected by it.

It has been decided to use also  $T_6$  (cold aisle temperature) because when  $u_2$  is constant, as in this case, it does not have big variations, but if we think to a standard working day: we will have continuous  $u_2$  (IT load) changes with a connected heat production. Therefore,  $T_6$  will be useful as feedback to detect the different heat produced by the servers. Obviously, warmer or colder air coming from the hot aisle can produce another type of flow mixing. Then it is fundamental to study it in the future.

As in Section 4.2 the same steps have been done and three models will be presented. This time both three are discrete-time Auto-regressive with eXogenous input (ARX) model, and their structure is:

$$A(z)\mathbf{y}(t) = B(z)\mathbf{u}(t - n_k) + \mathbf{e}(t) \quad (4.7)$$

with sampling time equal to 1 min.,  $n_k$  respectively equal to [1 5 1], [1 7 1], [1 5 1], and  $A, B, C, D$  with estimated parameters:

$$\Omega_o \ni \begin{cases} A(z) = 1 - 0.9723z^{-1} + 0.08614z^{-2} \\ B_1(z) = -0.003873z^{-1} + 0.000683z^{-2} \\ B_2(z) = -0.0005198z^{-5} + 0.0005479z^{-6} \\ B_3(z) = 0.09698z^{-1} + 0.05518z^{-2} \end{cases} \quad (4.8)$$

$$\Omega_s \ni \begin{cases} A(z) = 1 - 0.7676z^{-1} - 0.3205z^{-2} + 0.2027z^{-3} \\ B_1(z) = -0.005818z^{-1} + 0.00249z^{-2} + 0.0003114z^{-3} \\ B_2(z) = -0.0007365z^{-7} + 0.001416z^{-8} - 0.0006097z^{-9} \\ B_3(z) = 0.09947z^{-1} - 0.1501z^{-2} + 0.203z^{-3} \end{cases} \quad (4.9)$$

$$\Omega_u \ni \begin{cases} A(z) = 1 - 1.169z^{-1} - 0.2505z^{-2} + 0.4593z^{-3} \\ B_1(z) = -0.0137z^{-1} - 0.001871z^{-2} + 0.01031z^{-3} \\ B_2(z) = -0.001216z^{-5} + 0.002803z^{-6} - 0.001944z^{-7} \\ B_3(z) = -0.002418z^{-1} + 0.1158z^{-2} - 0.05215z^{-3} \end{cases} \quad (4.10)$$

We can now analyse the validation plots below where for each test two plots are shown. In each plot group are visualized all the inputs and output used,

comparing the real ones with the estimated output  $\hat{y}_1$ .

Also in this case we can see from Figure 4.8 as  $y_{1e}$  still has a gain problem compared to  $y_1$ . The augmented inputs did not solve the problem found in Section 4.2 even if between 20 – 40 min we have a better estimation, in other part of the plot the forecast is worse than before. For the other two regions instead we can detect an also better behaviour than before. Further details will be shown in next section.

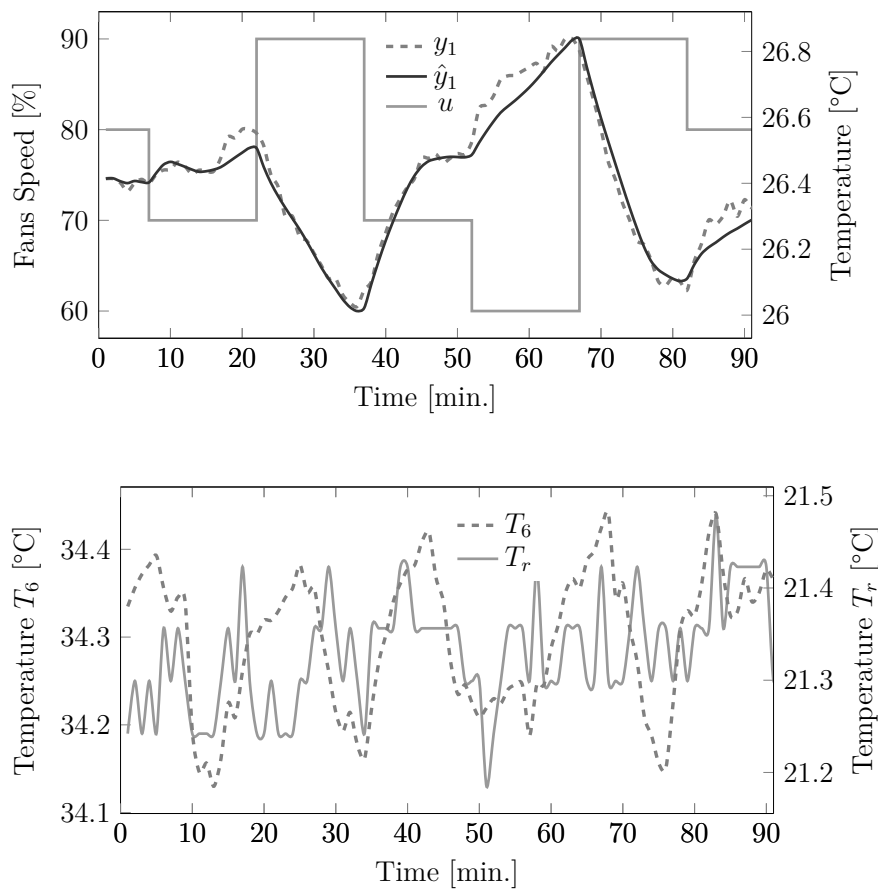


Figure 4.7: *Validation test* plot with  $y_1$  real signal and  $y_{1e}$  model output signal (estimated).  $u_1 \in \Omega_o$ .

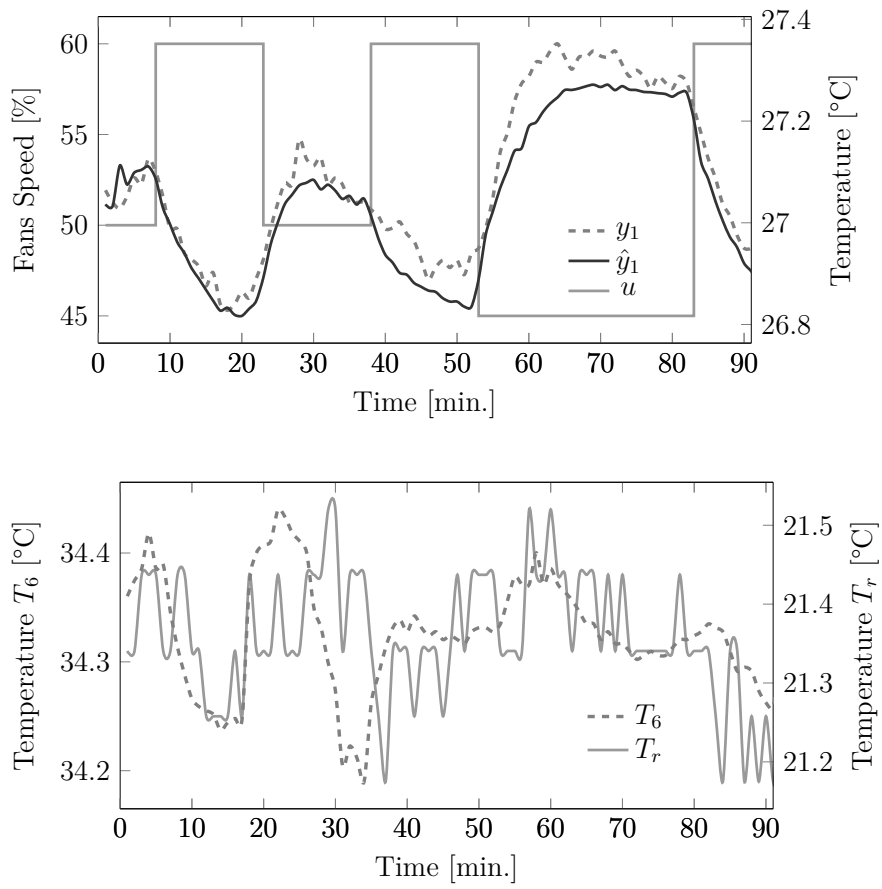


Figure 4.8: *Validation test* plot with  $y_1$  real signal and  $\hat{y}_1$  model output signal (estimated).  $u_1 \in \Omega_s$ .

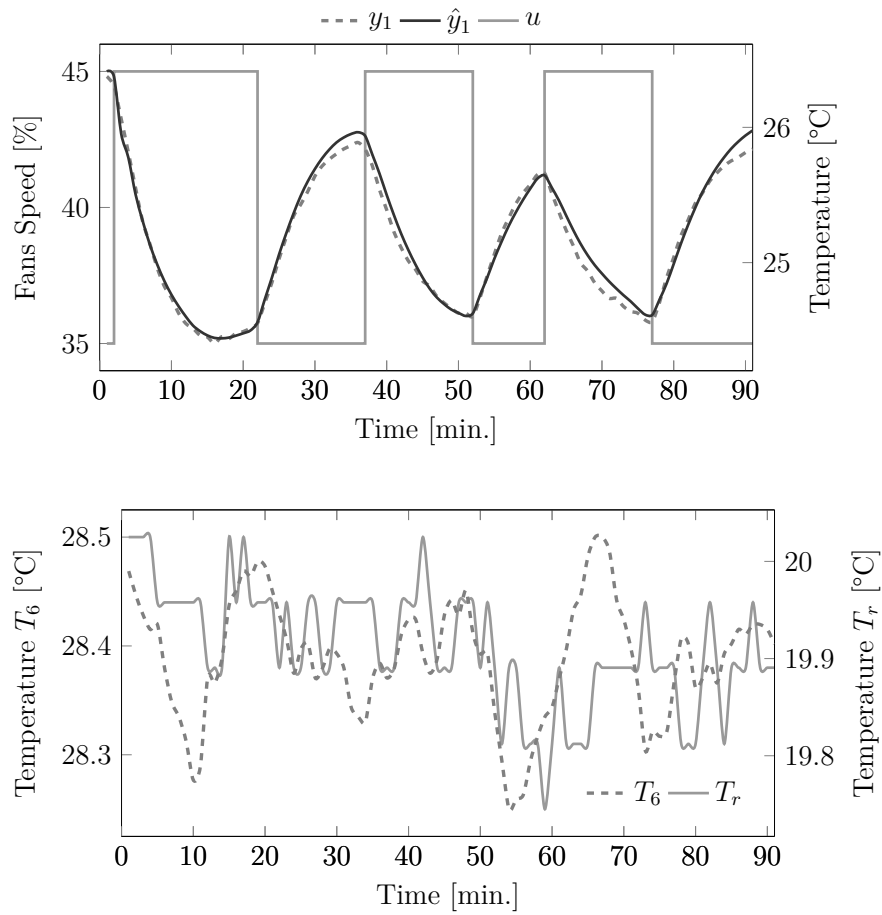


Figure 4.9: *Validation test* plot with  $y_1$  real signal and  $y_{1e}$  model output signal (estimated).  $u_1 \in \Omega_u$ .

Provisioning region	Model type	Type	Order	Fit
$\Omega_o$	SISO	BJ	[3 3 2 2]	81 %
	MISO	ARX	[2 2 2 2]	83 %
$\Omega_s$	SISO	BJ	[2 2 5 5]	75 %
	MISO	ARX	[3 3 3 3]	69 %
$\Omega_u$	SISO	BJ	[2 2 5 5]	85 %
	MISO	ARX	[3 3 3 3]	88 %

Table 4.1: Models *Fit* comparison generated by Validation test

## 4.4 Results

To better understand how many improvements we have with MISO models instead of SISO, we can make not only a visual comparison as done before, but also a numerical one. The easiest way to have it is comparing for each region the *Fit* of the two models obtained.

In this case, for validation dataset, *Fit* means the goodness of estimated output, so in other word, how close  $\hat{y}_1$  is to  $y_1$  in percentage. Obviously, we would like to find a *Fit* as close as possible to 100 % without overfitting.

As it can be valued in Table 4.1 for  $\Omega_o$  and  $\Omega_u$  we have a small improvement with MISO model using smaller orders. This means that in this case the augmented input model works better than single input one. But, we can not say the same of  $\Omega_s$  case, where an improvement in the behaviour of the signal has not been followed by an improvement of the gain.

We can conclude that LTI models proposed are not good enough to describe the all fans range. Mostly for models  $\in \Omega_s$  a deep study even on non-linear model should be done. Studying the non-linearity effects on the system (e.g. servers fans) could help not only to design better model, but also to have an advanced knowledge of the behaviours within the data center room.

To give a starting point on the control field of this system it has been decided to design a controller based on that models. Its study can be found in following chapter.

# Chapter 5

## Air flow control in data center room

### 5.1 Introduction

A lot of research efforts have already been put on minimizing the energy consumption of IT units. What is perceived as the research direction for the next few years it is to improve the data center room temperature control algorithms. As already mentioned, the room-wide cooling systems are energy-greedy and, if no new hardware technology will arrive, the only solution for improving the overall efficiency of the system is to design better coolant control strategies.

About control, in this study two approaches are considered: a model free controller based on PID regulation concepts and a model based approach using a MPC framework. In order to show two different approaches both models were implemented.

This chapter will focus on the implementation of the controllers based on SISO models - indeed, due to time issue, this project did not arrive far enough to implement a PID controller that could work on-line with *MATLAB* and the system.

It has been chosen to use SISO instead of MISO models because latter needs to work on-line. Indeed, PID can control only one input signal (in

our case  $u_1$ ), the other two inputs ( $T_6$  and  $T_r$ ), also affected by  $u_1$  should be estimated or used live with an on-line implementation. Not being able to do either it has been decided to use only the single input models. This choice is still supported by the fact that this chapter has been done to explain how two different ideas of control systems work and not to give an exact solution of them.

The forthcoming text will present three different results:

1. the design and test of a dedicated PID controller on our SISO models;
2. the design and test of a dedicated MPC algorithm on our SISO models;
3. the design and test of a dedicate PID controller on the actual system, developed on top of an ABB's controller that was already installed on the system.

The section will thus show also some simulations with Simulink on both controllers and the field results. As hinted before, for every controller we will use as a reference signal the temperature  $y_1$  and the induced air flow  $u$  as a control variable.

## 5.2 PID

PID controllers are widely used in every industrial settings for their simplicity in understanding, implementing and tuning them. Their big diffusion (they constitute almost 80% of the various industrial controllers installed worldwide) is due to the fact that the implementation of the controller does not require any knowledge of the model of the system to be controlled. Indeed, PID control action is computed starting only from the error given by the output value measured and the reference signal that the system should follow. A theoretical explanation of this controller is shown below.

### 5.2.1 PID theory

A PID is a feedback controller that can work either in continuous or discrete time. Restricting our attention on the last one, we can think of the controller



as a system that at every step check if the error  $\mathbf{e}(t)$  given by subtraction of the reference signal  $\mathbf{r}(t)$  (setted by the user) and the process variable  $\mathbf{y}(t)$  is equal to zero, if is not PID changes its output trying to solve the problem.

This specific type of controller is composed by three terms: Proportional, Integrative and Derivative. Applying them we will have an output (input of the system to be controlled) equal to:

$$\mathbf{u}_{PID}(t) = \left[ K_p + \frac{K_p}{T_i} \frac{T_s}{1 - z^{-1}} + \frac{K_p}{T_s} T_d (1 - z^{-1}) \right] \mathbf{e}(t) = PID(z)\mathbf{e}(t) \quad (5.1)$$

where we define  $T_s$  as the sampling time,  $K_p$  as the proportional gain,  $T_i = K_p/K_i$ ,  $T_d = K_d/K_p$  with  $K_i$  defined as the integral gain and with  $K_d$  said to be the derivative gain.

As shown in (5.1), the formula defining PID controllers is composed by three terms that can be added or neglected according to our need. In this way can be adopted many strategies using different types of configuration as: P, PI, PD and PID.

P controllers, augmenting  $K_p$ , makes the system response faster, but it is subject to disturbances that leads to steady state errors. A possible solution is to use the integral action which reduce the steady state errors with drawback of bigger overshoot, less stability and saturation problems. If this issues are too big, to solve them the derivative term can be added. This action can reduce the overshoot and stabilize the system, see Figure 5.1. Instead, to solve the integral saturation we can adopt an anti-windup scheme that is widely explained in literature, e.g., see [14].

Sections 5.4 and 5.5 will describe a practical usage of the Proportional, Integrative, Derivative controller that is ad-hoc for our system.

## 5.3 MPC

In this chapter we briefly explain the working principles behind MPCs. These are considered to be advanced control techniques to be implemented usually in difficult control problems. For more theoretical details beyond the mechanic description provided here we send the reader back to [15] [16]. The

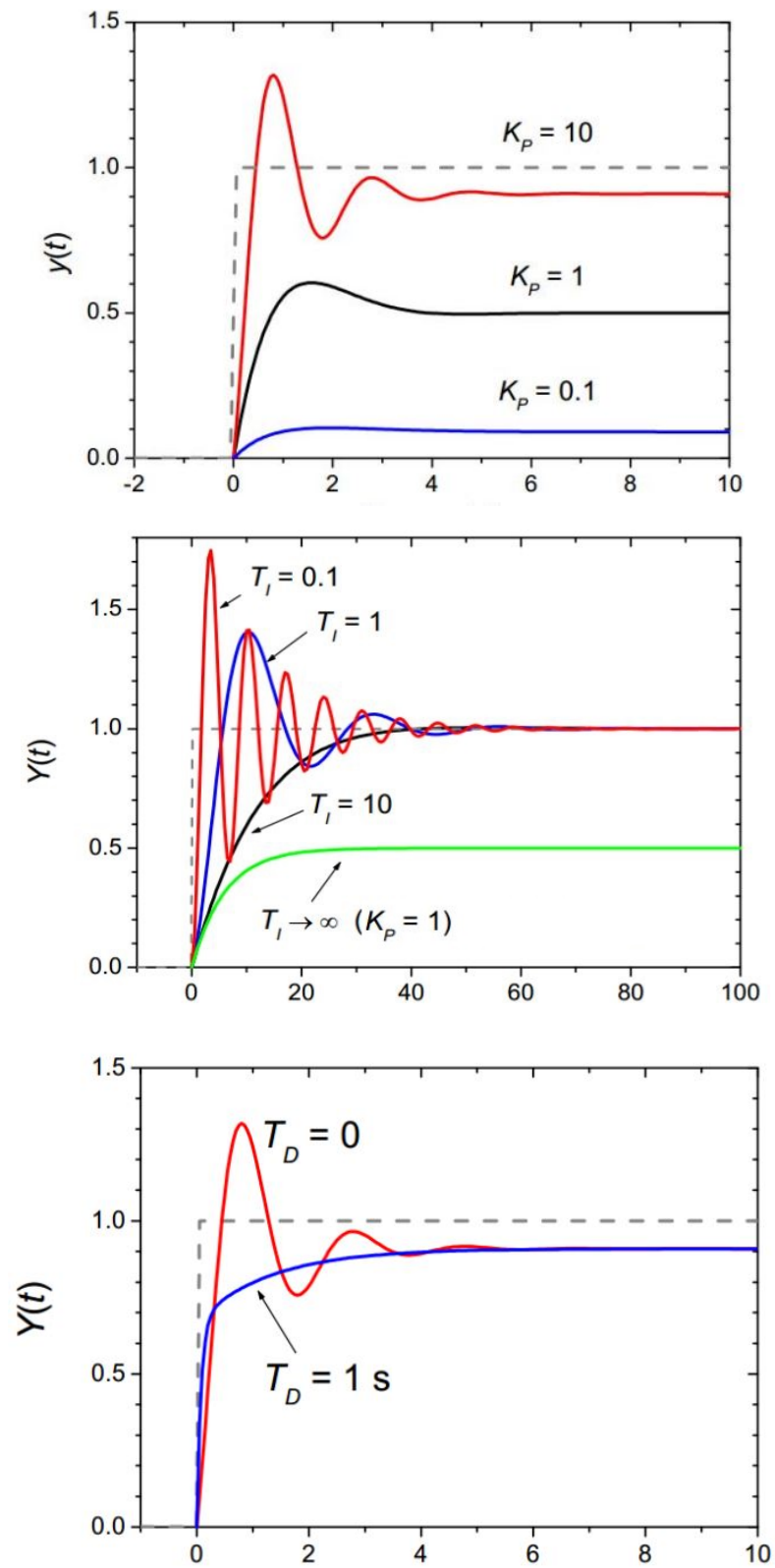


Figure 5.1: Example of a PID step response. From first to third panel we can see respectively the contributions of Proportional, Integrative and Derivative part.

basic MPC concept can be summarized as follows:

Suppose that we want to control a generic I/O process while satisfying inequality constraints on the input and output variables. Also suppose that we have a precise model of aforementioned system that we want to control. Using that model and current measurements we can predict future values of the outputs, and the appropriate changes in the input variables can be calculated based on both predictions and measurements.

### 5.3.1 MPC theory

To give a fast theory explanation we can use a simple example and think to design a controller that takes the state of a deterministic, linear system to the origin. If the setpoint is not the origin (as will be in our case), we can make modification of the setpoint problem to account for that. The system model taken as example is:

$$x(t+1) = Ax(t) + Bu(t) \quad (5.2)$$

$$y(t) = Cx(t). \quad (5.3)$$

For a simpler discussion, the state is assumed to be measured, i.e., that  $C = I$ . For simplicity we ignore the case where there is the need to perform state estimation tasks.

Now we can use the model 5.2 to predict the state evolution given  $N$  inputs collected in a vector  $\mathbf{u}$ . Importantly, the choice of input and output constraints is a fundamental aspect that distinguishes MPC for linear systems from LQR. Indeed, both use an objective function  $V(\cdot)$  to measure the deviation of the trajectory of  $x(k)$ ,  $u(k)$  from zero. Many functions can be used, but in general the control problem is to minimize the following cost function:

$$V(x(0), \mathbf{u}) = \frac{1}{2} \sum_{k=0}^{N-1} [x(k)^T Q x(k) + u(k)^T R u(k)] + \frac{1}{2} x(N)^T P_f x(N), \quad (5.4)$$

subject to

$$x(t+1) = Ax(t) + Bu(t). \quad (5.5)$$

So, the objective function explicitly depends on the input sequence and initial state. Note that, tuning the control parameters means varying the matrices  $Q$  and  $R$ , i.e., the relative weight of the state being non-zero versus the input being non-zero. For generality we allow the final state penalty to have a different weighting matrix,  $P_f$ . As hinted before, we can drive the state to the origin quickly at the expenses of large control action giving larger values to  $Q$  instead of  $R$ . Instead, with  $R$  large and  $Q$  smaller we reduce the control action and slow down the speed at which the state try to reach the setpoint.

To guarantee that the solution exists and is unique a classic and general requirement is to have  $Q$ ,  $P_F$  and  $R$  real and symmetric,  $Q$  and  $P_F$  positive semidefinite, and  $R$  positive definite.

### 5.3.2 The objective function for our SISO models

In this section we propose our MPC controller applied to the models developed in Section 4.2. Then, it will be a simple simulation of the real system.

As explained before, MPC to work needs the state of the system, for this reason models designed in Section 4.2 have been transformed in State Space models using *ss MATLAB* function. Starting from an initial temperature of  $y_1(0) = 28^\circ C$  and with a reference signal set to  $r = 26.5^\circ C$  the MPC constraints used are in Table 5.1. The number of inputs  $u(t)$  collected to predict the state evolution were  $N = 10$ . It has been decided to use a simple objective function  $V(\cdot)$  to be minimized, and it is shown below

$$V(x_0, \mathbf{u}) = \sum_{k=0}^{N-1} e(k)^T R e(k), \quad (5.6)$$

subject to

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t). \quad (5.7)$$

I	$u_{min}$	35 %
	$u_{max}$	90 %
O	$y_{1min}$	22 °C
	$y_{1max}$	30 °C

Table 5.1: MPC constraints

With  $e(t)$  equal to  $\hat{y}_1$  estimated using the model minus  $r$ .

The test explanation and results are listed below in Section 5.5.

## 5.4 Implementation

This section will show some field tests that we performed leveraging a physical PID that was already installed on the system. This controller has been supplied by ABB, and is delegated to control the data center room temperature through the CRAC's fans speed (i.e., through respectively the variables  $y_1$  and  $u$  defined in our problem formulation). To be more precise, physically there are two different controllers, one for each side of the room, and one can decide if he wants to control each CRAC independently (i.e., the two sides of the room separately), or to control the whole system with only one controller, so that the second one will work as a slave device.

For technical issues given by the still not complete installation of the control system, we had to use the configuration where there is only one controller for all the CRACs installed in our computer room. Specifically, it means that there was one master CRAC with a specific reference signal, and all the others three CRACs working as slaves. Being our control signal  $y_1$  strongly connected to only one side of the room, it has been decided to study only that part even if the PID could control the whole of it.

To complete the technical description of our hardware, the ABB system employed in our field experiments was such that its output voltage signal (i.e., the signal  $u_{PID}$ ) was between a range of 0-10 Volts and was controlling the CRAC fans speed  $u$  with a ratio of 1:10 (i.e., example  $u = 35\%$  if  $u_{PID} \simeq 3.5V$  or  $u = 72\%$  if  $u_{PID} \simeq 7.2V$ , and so on).

As for tuning the parameters of this PID, we applied a classical procedure and tried several different PID's coefficients in similar tests. These were made to try the various configurations and they had 3 different parts: first 30 *min* open loop with  $u$  fixed, second 30 *min* closed loop with reference signal  $r = y_1(30)$ , third a step response is applied to the system changing the servers IT load ( $u_2$ ) for 90 *min*. Sometimes even with 90 minutes it has been difficult to reach the steady state and unfortunately due to time issues we could not extend the tests longer. A message that we can extrapolate from these partial results is that if after 90 minutes stability can not be achieved, then the system is probably too slow and the PID coefficients should be changed.

Said so, many coefficient have been tested starting from the PI configuration; sadly, when we thought to have the perfect ones the test could not be performed for system's issues. So, below will be shown only some of the coefficients among the many tested, and our table can give an idea of the path taken to get the best results that we achieved. We will start from low  $K_p$  with  $K_d = 0$  until a higher proportional and derivative gain, for more detail about the theory behind it look Section 5.2.1.

In Figure 5.2 we can see that using a PI with  $K_p = 2$  and  $K_i = 0.05 * 60$  we have a controller response too slow when the error is different to 0. For example, we can see that it needs at least 10 *min*. to bring an error from 0.1 to 0. Also the integral actions seems to be not strong enough so: in Figure 5.3 one can hopefully appreciate how we tried to solve the problem with a PID with  $K_p = 9$ ,  $K_i = 0.07 * 60$  and  $K_d = 0.05/60$ . We can see how the slope is higher but still not steep enough. To try to solve this problem, our last choice was to set  $K_p = 11$ ,  $K_i = 0.11$  and  $K_d = 0.05$ , unfortunately as said before, due to some technical problem on the system we could not try these last values that seemed promising.

## 5.5 Simulation

This section will present some tests performed using our PID and MPC controllers described in Section 5.2 and 5.3. As for these technologies, sev-

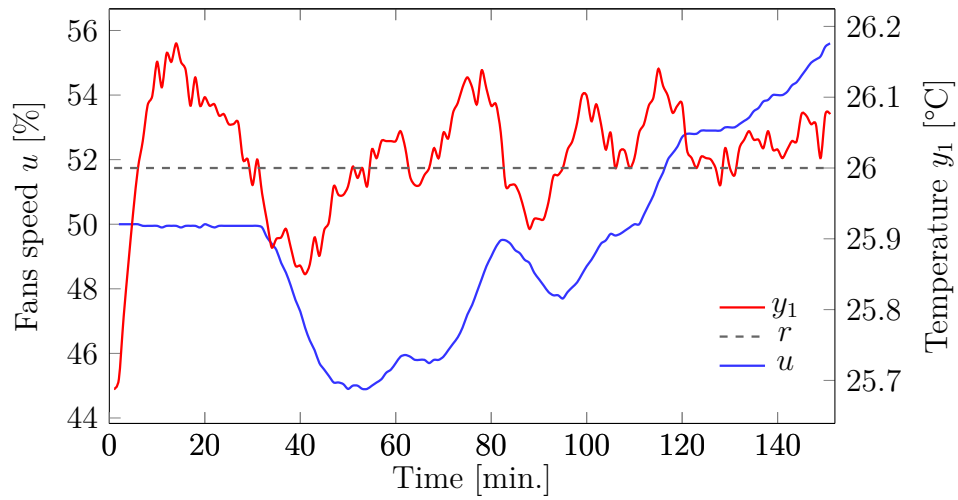


Figure 5.2: PI test with coefficients  $K_p = 2$ ,  $K_i = 0.05$ .

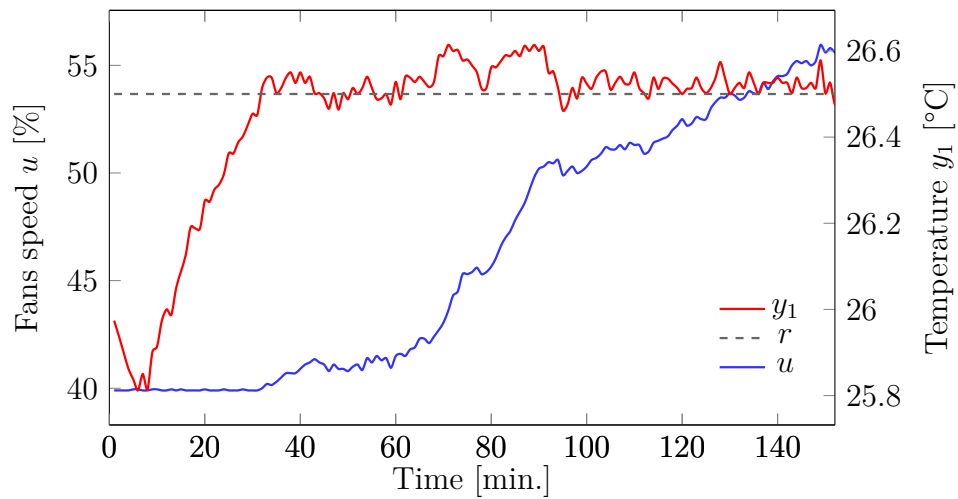


Figure 5.3: PID test with coefficients  $K_p = 9$ ,  $K_i = 0.07$  and  $K_d = 0.05$ .

eral coefficients have been tested, but only the most promising ones will be presented. As said in Section 5.1, the simulator used in our experiments corresponds to the SISO models that we trained from the field data. However, using that models we could not make the same step test as in Section 5.4 because there was no input that would allow us to add an IT load to the system. It was therefore decided to create a step by changing the value of the reference signal.

In the following we will present two examples of such kind of tests. During the first one we will compare the same inputs corresponding to field and simulated realizations of the system dynamics under the action of the PID controller; this to understand how much the simulation deviates from reality. The results obtained here show the existence of two steps, the first one happening at  $90 \text{ min}$  in open loop with  $u$  fixed, and then the system will enter into a closed loop with a specific reference signal with  $r < y_1(90)$ . As one can see in Figure 5.5, the main difference with Figure 5.4 is that without noises that can be given by many different components inside the room we will have almost the same rising time, but a completely different settling time and fans equilibrium point. The solution as already said is to obtain a better model that can detect all that noises.

The second test focuses instead on the performance of the MPC. In this case an initial value and a reference value (to which the system will have to follow) will be set, as specified in Section 5.3.2. As one can see, in this case after  $10 \text{ min}$ . the controller has minimized the objective function 5.6 ( $e(10) = 0$ ) making sure that  $y_1$  is aligned with  $r$ .

More details about the results obtained using the PID and MPC schemes will be given in the next section.

## 5.6 Results

Differences between the performance of PID and MPC schemes can be summarized as follows: if we take as an example a test as the one explained in section above, we can see how the MPC strategy does not produce overshoots, and how its settling time is extremely short.



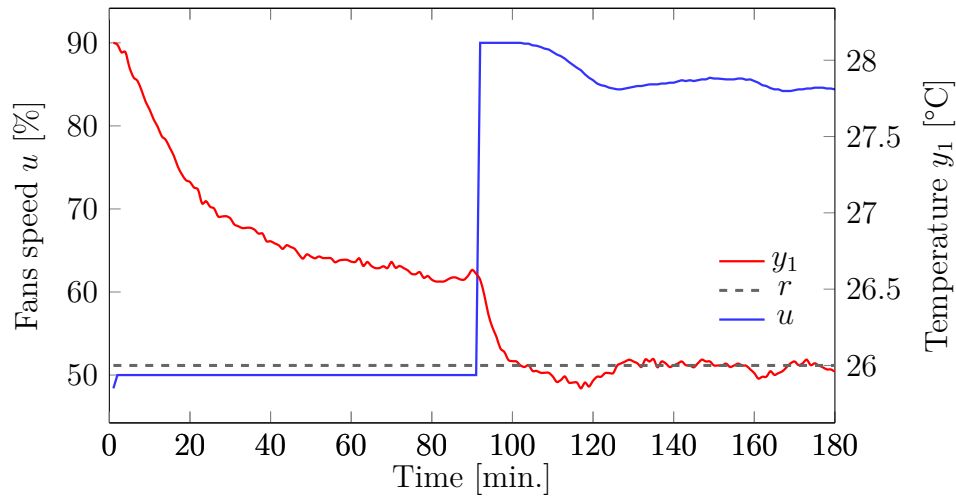


Figure 5.4: ABB's PI field test with coefficients  $K_p = 2$ ,  $K_i = 0.05 * 60$ .  $y_1(0) = 28.1^\circ\text{C}$ ,  $r = 26^\circ\text{C}$

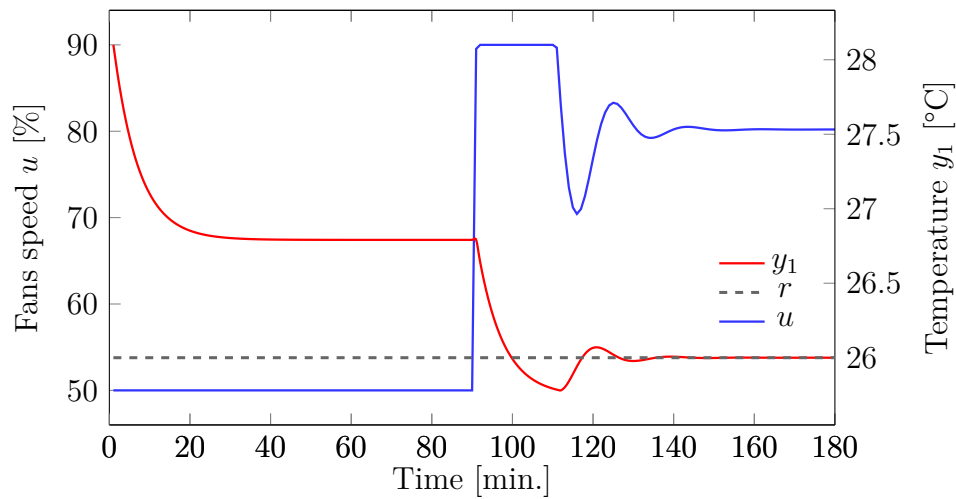


Figure 5.5: PI simulation with coefficients  $K_p = 2$ ,  $K_i = 0.05 * 60$ .  $y_1(0) = 28.1^\circ\text{C}$ ,  $r = 26^\circ\text{C}$

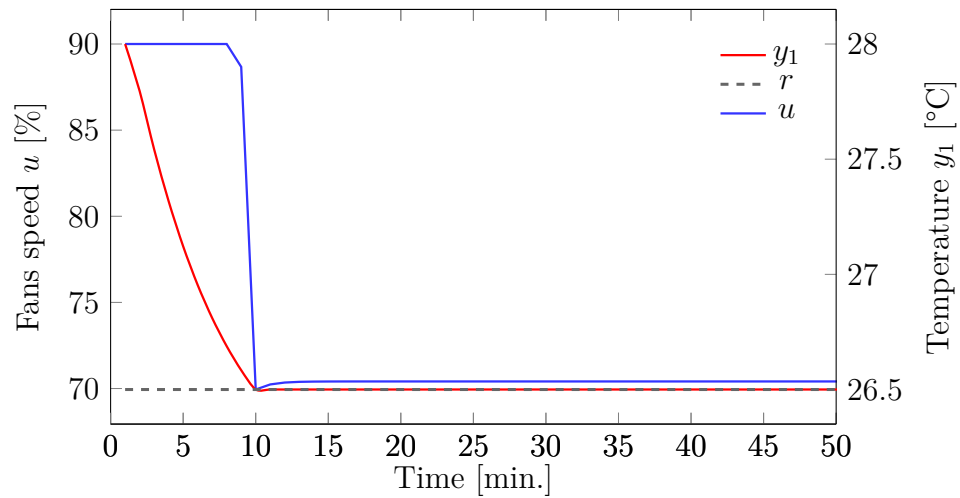


Figure 5.6: MPC simulation with  $y_1(0) = 28^\circ\text{C}$ ,  $r = 26.5^\circ\text{C}$

Having a controller based on the model thus makes the whole system more precise and less prone to steady state error. Having a fast but at the same time precise system can also improve energy consumption, one of the fundamental reasons for the implementation of these systems. This said, however, we have to notice that our tests are based on simulations performed through the very same model that was used to construct the controller. In a sense, it is thus expected that the MPC scheme has excellent performance. One should however test it on the real system to draw more meaningful conclusion.

In any case, we can claim that using PID technologies is a good solution when one does not have reliable models at hand, some developmental time, or special needs to implement advanced control scheme. That said, even if we cannot make claims based on evidence from the field, for this specific problem of managing CRAC systems we firmly believe that the use of an MPC scheme can be a better choice from an energetic point of view.

# Chapter 6

## Conclusions and future works

This study on air behaviour within a data center led us conclude that, first of all, choosing which physical signal is most meaningful to be analyzed to obtain models that can be useful from a control perspective requires a careful and data-driven study of the air flows phenomena. What we learned is indeed that non-expected phenomena may occur, and that blueprints of the computer rooms are insufficient to forecast what actually may happen from an air flow distributions phenomena.

We also discovered that, at least in our setup, it may be meaningful to try to find combinations of different linear models each characterizing a specific provisioning region. During our efforts we compared several black box strategies, for both SISO and MISO formulations of our models. Interestingly, we found that in our specific field case even if the last type of models seems to lead to better forecasting results, there is the need for some improvements in some specific air provisioning region before being able to implement effective MPC schemes.

As for this topic, we also confirmed that, through using an identified SISO model to describe our physical setup, MPC strategies can, when compared with a classic PID approaches, lead to completely different ways to bring the system to a chosen target value of the temperature within the computer room. At last, even if not supported by quantitative evidence, the intuitions that we developed leads us to conclude that if it is possible, the use of predictive

control is preferable in air cooled data centers management systems.

In conclusions, we also identified a list of possible future works, whose foremost points could be:

- with the same data collected during this project, design more complicated models taking into account the provisioning regions detected and other non-linear phenomena as servers fans;
- implement a controller focusing on maintaining a specific room temperature while minimizing the energy consumption;
- explore the possibility of using Computational Fluid Dynamics (CFD) programs as an aid to identify the various air provisioning regions and to train the models describing these phenomena.

# Bibliography

- [1] RajatGhoshYogendraJoshi, “Error estimation in pod-based dynamic reduced-order thermal modeling of data centers.”
- [2] S. K. S. G. Qinghui Tang, Tridib Mukherjee, “Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters.”
- [3] J. Aasa, “Linear-quadratic regulation of computer room air conditioners,” Master’s thesis, LuleåTekniska Universitet, 2018.
- [4] H. Z. K. M. X. Z. Winston Garcia-Gabin, Erik Berglund, “A data center model for testing control and optimization algorithms,” *IECON conference*, 2017.
- [5] C. P. C.E. Bash and R. Sharma., “Dynamic thermal management of air cooled data centers.”
- [6] C. Lee and R. Chen, “Optimal self-tuning pid controller based on low power consumption for a server fan cooling system.”
- [7] E. Berglund, “Lqr and mpc control of a simulated data center,” Master’s thesis, KTH Stockholm, 2017.
- [8] J. Olsson, “Stochastic model predictive control for data centers,” Master’s thesis, LuleåTekniska Universitet, 2016.
- [9] M. Eriksson, “Monitoring, modelling and identification of data center servers,” Master’s thesis, LuleåTekniska Universitet, 2017.
- [10] “The different technologies for cooling data centers.”

- [11] “Air mixing handbook,” Blender Products, Inc.
- [12] D. V. J. G. W. G.-G. Emanuele Simonazzi, Miguel Ramos Galrinho, “Detecting and modelling air flow overprovisioning / underprovisioning in air-cooled datacenters,” *IECON conference*, 2018.
- [13] M. Zorzi, “Lecture notes system identification,” Department in Information Engineering, Padua(IT), Tech. Rep., 2018.
- [14] P. F. Ticozzi, “Appunti del corso di controllo digitale,” Department in Information Engineering, Padua(IT), Tech. Rep., 2012.
- [15] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, Wisconsin: Nob Hill Publishing, 2012.
- [16] A. B. F. Borelli and M. Morari, *Predictive Control for linear and hybrid systems*.