UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Distributed Clustering-based Sensor Fault Diagnosis for HVAC Systems

Department of Information Engineering

Master's Degree in Automation Engineering

Academic Year 2015-2016

Student
Redona Reci

Advisor
Angelo Cenedese

Co-advisor
Thomas Parisini

To my parents.

Life is a play that does not allow testing. So sing, cry, dance, laugh and live intensely, before the curtain closes and the piece ends with no applause.

C. Chaplin

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

In this thesis we deal with a distributed Sensor Fault Diagnosis (SFD) architecture for Industrial Wireless Sensor Networks (IWSNs) monitoring heating, ventilation and air conditioning (HVAC) systems. We consider a wireless connection, which, differently from traditional wired systems, enables the possibility for nodes in a network to dynamically and autonomously group into clusters, allowing thus to leverage the flexibility and robustness of IWSNs in supervisory intelligent systems for high level tasks such as, for example, environmental sensing, condition monitoring and process automation. To enhance the robustness features characteristic of the IWSNs, we exploit a recently proposed distributed clustering method, where sensor node partitions are obtained according to communication network topology (*network decomposition*) and the data gathered from the environment, i.e. measurements distribution (*data clustering*), such that nodes that exhibit similar behaviors can be grouped together and share their information. Using this network partition strategy and replacing the measurements similarity criterion with a consistency one, we develop a fault diagnosis algorithm that allows the detection and isolation of multiple sensor faults and considers the possible presence of modeling uncertainties and disturbances. In order to prove the validity of our method, detectability and isolability conditions are provided. Finally, the proposed strategy is applied to an HVAC system, made of the electromechanical part and a single zone. Indeed, HVAC systems consist of all the equipment that control the conditions and distribution of indoor air and have a relevant impact on occupant's comfort and energy consumption. Faults in the system can provoke a loss in terms of efficiency with a consequent increased energy consumption, reduced life-time of its elements and discomfort. We show that a monitoring IWSN provided with a suitable sensor fault diagnosis architecture manages to cope with these issues. Simulation results show the effectiveness of the proposed method.

1 Introduction

In the last years, technological advancements in home automation have contributed to the design of the so-called smart buildings. Modern buildings can be viewed as cyber-physical systems, which beyond the physical-engineered component, i.e. the building itself, operate at the intersection of multiple sensors and control systems designed to monitor, coordinate and control the various interconnected physical subsystems constituting the building [1]. As the demand for comfort improvement, robustness and reliability of the systems together with higher productivity and safety increase, one of the issues that has been raising concerns in the context of building automation (BA) is energy efficiency management, for both monitoring and control. The global contribution from only buildings toward energy consumption, both residential and commercial, has increased to between 20% and 40% in developed countries and it is rapidly increasing, as the population grows and the demand for building services and comfort levels is rising, exceeding thus all the other major sectors [2]. Aiming to reduce wasteful energy consumption behavior, many different "Building Energy Management System (BEMS)" have been developed and applied to measure the energy consumption in buildings, as well as an increased efficiency of systems and materials together with the adoption of green policies, which have contributed to prevent a dramatic growth of the consumption in residential buildings.

The most energy consuming subsystem in a smart building is the HVAC system, which uses half of building consumption [3]. HVAC systems consist of components working together to introduce, distribute and condition air in a building such to guarantee human comfort. They play a major role in the control of Indoor Air Quality (IAQ) and thermal comfort. Indeed, poor ventilation, improper temperature and humidity cause a bad indoor environmental quality. People spend most of their working time indoors: they are less productive and more often get sick with a bad IAQ, since it can cause irritation of the eyes and nose, fatigue, headache and shortness of breath [4]. To cope with these problems, HVAC systems typically employ a control that maintains a fixed set point of fresh air ventilation based on the designed occupancy of the space; this technique though results inefficient in terms of energy consumption. Nevertheless, if designed properly, an intelligent control manages to reduce HVAC systems energy consumption by $20 - 30\%$. The energy saving would also have a positive effect on the CO_2 emissions as buildings account for about 33% of global CO_2 emissions, since a significant part of CO_2 emissions are in fact caused by the combustion of fossil fuels to provide heating, cooling, lighting and the power for home appliances and electronic devices [5].

Energy efficiency and occupant's comfort represent the most important goals that have to be achieved in the design and implementation of HVAC systems and they are strictly related together [6]. A new challenging scenario has recently appeared related to the thermal management of data centres where the interplay between a monitoring sensor network and the electromechanical components is fundamental to ensure energy efficiency and Quality of Service (QoS) [7], [8]. A punctual, accurate and robust monitoring of the environment thermal state can be performed through the employment of IWSNs. A standard wireless sensor network (WSN) consists of a group of sensor nodes which are spatially distributed in a measurement area. Each sensor node is equipped with a transducer which can provide an electric signal, changes of which depend on a physical variable like temperature, pressure and illumination intensity. Sensors in the WSN can communicate among each other and in case of a centralized approach, they forward their measurements to the base station. WSNs are cheap and flexible: a greater number of sensors can be used, offering more accurate measurements, allowing the network to be built and turned down quite easily without changes in the rest of the environment. Moreover, sensors can be reprogrammed in every moment, complex algorithm can be implemented and they can be placed everywhere, as they work on stand-alone energy supply. Additionally, wireless architecture enables the possibility for nodes in a network to dynamically and autonomously group into clusters according to some similar features. Such clustering approach results particularly suitable for enhancing the performances of IWSNs, which are a particular class of WSNs fitting high level tasks due to their additional attention posed to robustness, reliability and maintainability features with respect to standard WSNs.

A tool frequently used by IWSNs to accomplish these specifics is by adopting Fault Detection and Isolation (FDI) procedures which allow to determine anomalies and malfunctions in one or more sensors of the network, to isolate them and eventually to reconfigure the network accordingly, still preserving its robustness and reliability [9], [10]. In such sense, IWSNs are suitable for monitoring an HVAC system, since the latter is made of several sensors and actuators, which over time may be subject to failures and faults, making it necessary to adopt a fault detection and isolation algorithm in order to guarantee a safe behaviour of the system over time. Indeed, if a fault occurs in one of the sensors of the zone, it would transmit the wrong signal to the electromechanical part, which would continuously operate leading to undesired conditions in the room and both increased energy consumption. Furthermore, the use of sensors and information technology to monitor the health of components, diagnose problems, and recommend service allows automated fault detection and diagnostic systems for HVAC equipment which benefits are reduced downtime, service costs, and utility costs.

In addition to this, the capability of sensors to group into clusters leads to efficiency improvements and can attain fault resilience [11], [12]. Indeed, the advantages of partitioning the sensors into groups according to some similar features in order to perform fault detection are mainly two:

- *network decomposition* [13] allows to reduce communication, since, once clusters are created, sensors communicate only with neighbouring sensors belonging to the same cluster in order to detect faults;

- *data clustering* [14] allows to create groups of sensors having similar features and modeling uncertainty/disturbances. This implies that the conservativeness of the fault detection thresholds can be reduced and only measurement locality is exploited.

1.1 Literature Review

Before stating the novelties introduced in this thesis, we briefly review a list of works related to our framework. Recent fault tolerant clustering approaches proposed in the literature can be classified into centralized policies and distributed ones. In centralized approaches, the base station adopts active detection model to collect node information and recognizes the faulty nodes in the network by comparing current state with history information of nodes. Nevertheless, in case of nodes with heavy loads, failures may provoke the collapse of the entire network, with high communication delays, low efficiency and increased energy consumption [15]. Moreover, the Fault Detection (FD) problem for a sufficiently large-scale system cannot be solved in practice through a centralized approach, and real-time diagnosis is limited by the computation power needed for simulating the system model and communication resources needed to convey all the measurements.

Distributed policies are more suitable than centralized when dealing with large-scale interconnected dynamical systems, such as power networks and multi-agent systems due to the lower complexity and less use of network resources.

Distributed techniques generally divide the network into areas to which clusters are associated and the fault management tasks are distributed uniformly among the zones. Generally these techniques rely on special nodes, referred as cluster-heads (CH) which are high-energy sensors leading the other sensors of the network to form distinct clusters in the system, managing the network in the cluster, performing data fusion and organizing sensors according to the required missions and tasks. Beyond contributing in the clustering procedure, these special nodes support fault detection and anomaly recognition procedures, contributing in the recovery mechanisms which limit the performance impacts caused by failure [16]. Since the management nodes just monitor only a few nodes in the network, i.e. nodes belonging to their same clusters, they are responsible for local fault detection and hence they contribute to save communication energy consumption and reduce system response time when events occur [11], [17].

On the other hand, considerable benefits in terms of scalability, robustness and reconfigurability of the network are achieved when distributed techniques without relying on CHs are applied. In [18] a distributed fault detection and isolation methodology for nonlinear systems is presented, where a *divide et impera* partition of the network is applied in order to decompose the main problem into smaller and simpler subproblems. In this way, the scalability issues of the centralized architecture are overcome and the new resulting scheme can be applied to nonlinear and uncertain systems of large dimensions. In [19] a distributed fault detection and isolation method is proposed in the form of multi-agent network which provides high efficiency, scalability and robustness. Similarly, in [20], the increasing importance of distributed and networked systems is underlined. Indeed, as the complexity of the system grows, the risk of faulty operations in one or more subsystems increases, and detection and isolation of the faults may result as a difficult operation. This justifies the necessity to develop fault detection and isolation methods for these kinds of systems. In [21], a methodology is presented that detects and isolates multiple sensor faults in large-scale interconnected nonlinear systems. In particular, for each subsystem, a local sensor fault diagnosis (LSFD) agent is designed that is responsible for multiple sensor fault detection and isolation in the local sensor set. The design of the LSFD agent relies on the decomposition of the local sensor into smaller groups of sensors which may be disjoint or overlapping. The reason for decomposing the local set of sensors is that, in large-scale

systems, such set may contain a large number of sensors and the isolation process would be difficult to perform. The criterion used for decomposing the set of sensors relies only on the stability of the observer and does not consider any property related to topology or similarity of the data collected.
Finally, various methodologies have been developed for detecting and isolating faults in HVAC systems. In most cases, they have focused on faults and anomalies occurring in the actuators and in the plant, while sensor fault detection and isolation shows further difficulties when large-scale systems are adopted due to the high number of sensors which can be different in the structure and may measure different quantities.

1.2 Statement of Contribution

This thesis presents a design of a methodology for detecting and isolating single and multiple sensor faults in HVAC systems. The proposed methodology is developed in a distributed framework considering an HVAC systems made of two zones, the electromechanical part and a room, as a set of interconnected nonlinear subsystems. We deploy N sensors in the room which constitute a monitoring IWSN. Starting from the results presented in [12] and [22], distributed sensor fault detection is conducted using two consistency tests, one based on residuals which exploit the clustering of the network and the second one based on the similarity with past measurements. The distributed sensor fault isolation procedure is carried out by combining the two consistency tests together and applying a reasoning-based decision logic. The performance of the proposed methodology is analyzed with respect to sensor fault detectability and isolability. The main contribution proposed are:

- a modeling of the heat diffusion in a room belonging to the HVAC system, carried out considering the position of the sensors w.r.t. the heating source;
- a methodology that considers heterogeneous sensors measuring different quantities and a procedure to tune the measurement model and properly design the clustering threshold bounds for estimation and fault detection purposes;
- a FDI algorithm that takes advantage of such clustering procedure and provides a model-based clustering reconfiguration strategy, able to cope with both single and multiple sensor faults;
- a numerical validation within the scenario of temperature monitoring for the smart management of an HVAC system.

1.3 Outline

After this introductory chapter, the thesis is organized in 6 chapters. Chapter 2 is devoted to the introduction of HVAC systems, and in particular to the model used in this thesis. Nonlinear controllers are implemented and the heat diffusion model is derived. In Chapter 3, IWSNs for monitoring task are presented. We describe the advantages carried by WSNs for industrial applications and their enhanced capability to deal with sensor faults. After a brief review on graph theory useful to better understand the following section, we describe the recently clustering approach proposed by [22] which will be exploited to obtain the new distributed SFD architecture we propose. Chapter 4 deals with the problem formulation and it is dedicated to the complete description of the new SFDI method, followed by detectability and isolability conditions analysis. Chapter 5 illustrates the application of the distributed clustering-based sensor fault detection method to an IWSN monitoring the performance of the given HVAC system. Different kinds of faults are considered, for both single and multiple faults starting from two different scenarios of deployment of the sensors. Finally, Chapter 6 summarizes the main conclusions reached at the end of all the work done and proposes some future directions of research.

In conclusion, we underline the presence of two appendices: one dedicated to the main terminology used in model-based fault diagnosis, and one giving a brief overview of Voronoi tessellations as a tool to understand the optimal partition of a space for the deployment of the sensors.

1.4 Projects Related with the Thesis

Starting from this thesis project, the paper "Distributed Clustering-Based Sensor Fault Diagnosis for HVAC Systems" has been extracted and submitted to $20th$ IFAC World Congress.

2 HVAC Systems

In the first part of this chapter we introduce HVAC systems, initially giving a short and exhaustive general description of their main components and then presenting the model that we have considered throughout the thesis. In the second part we deal with the implementation of the controllers for achieving the desired temperature in the cooling coil and in the room belonging to the HVAC system. Hence, we introduce some basic concepts of two nonlinear control architectures, namely feedback linearization and backstepping controller and show then the main steps for their design using the given HVAC model. In the last part of the chapter, we deal with the problem of the heat diffusion in the room belonging to the system, introducing the heat equation as a tool for determining the temperature in a given position in the space as a function of time and the distance from the heating source.

2.1 Definition of HVAC Systems

HVAC is an acronym that stands for "heating, ventilation and air-conditioning" and generally includes a variety of active mechanical/electrical systems employed to provide thermal control in buildings [23]. Control of the thermal environment is a key objective for virtually all occupied buildings, such as single family homes, apartment buildings, hotels and senior living facilities, medium to large industrial and office buildings such as skyscrapers and hospitals, where safe and healthy building conditions are regulated with respect to temperature and humidity, such to guarantee occupant satisfaction and productivity.

A heating system ("H" in HVAC) is designed to add thermal energy to a space in order to maintain some selected air temperature that would otherwise not be achieved due to heat loss to the exterior environment and neighbouring ambients. Heating is usually provided by steam or hot water coils with remote boiler. Electric heating coils, heat pumps, and direct gas-fired duct heaters also are used. The heat can be transferred by convection, conduction, or radiation.

A ventilation system ("V") is intended to introduce air or remove any combination of moisture, odors, smoke, heat, dust such to provide high indoor air quality, without changing the temperature. Ventilation includes both the exchange of air to the outside as well as circulation of air within the building, providing improvements in the air quality and thermal comfort.

A cooling system ("C") is designed to remove thermal energy from a space in order to maintain some selected air temperature that would not be achieved due to heat gains from interior heat sources and the exterior environment. Cooling systems are generally considered as part of the air-conditioning systems ("AC"), which beyond cooling, provide also humidity control for all the parts of a building, air temperature control, air circulation control and air quality control. A typical air conditioning system has a gas, stoker, or oil-fired furnace, which has an add-on direct-expansion cooling coil installed in the discharge. Generally, a thermostat is provided, which not only starts and stops the supply fan, but controls also the heating and cooling.

HVAC systems are of great importance nowadays because the success or failure of thermal comforts is usually directly related to the success or failure of a building's HVAC system, and secondly because maintaining appropriate thermal conditions through HVAC system operation is a major driver of building energy consumption.

2.2 HVAC Components

The basic purpose of an HVAC system is to provide interior thermal conditions such to guarantee occupants comfort. Commonly, this is achieved by adding or removing heat to or from building spaces. Each building has a characteristic exterior air temperature, also called balance point temperature, at which the building in use offers thermal comfort without any need for a heating and cooling system. When the outside air temperature falls below the balance point temperature, the interior air temperature will drop unless heat is added to the building to compensate. Space heat may be added or removed by an electro-mechanical system, which is termed an active systems approach. An active system has the following general characteristics: it normally utilizes purchased energy for its operation, it requires special-purpose components that serve no other major building function, and it is generally relatively independent of the underlying architectural elements of the building. Alternatively, space heat may be added or removed by a system designed to make use of naturally occurring environmental forces. Such a system is termed a passive system. A passive system has the following general characteristics: it utilizes renewable site resources for energy inputs, it usually involves components that are integral parts of other building systems, and it is usually so tightly interwoven with the basic architectural fabric of a building that removal would be difficult. The system considered in this thesis belongs to the first category.

Among the main components of an HVAC system we recall:

- the boiler: it is a heating system component designed to heat water for distribution to many zones components of the system. Boilers are generally constituted by a heat-source element and some volume of water storage. According to the design, it may produce hot water or steam;
- the chiller: it is a refrigeration unit designed to produce cool water for cooling purposes. The chilled water circulates in one or more cooling coils belonging to the air handling units (AHU), fan-coils or induction units. Capacity control in a chilled water system is usually achieved through modulation of water flow through the coils;
- pipes: in water-based central systems, pipes are used to convey water from the source to the final delivery components. A minimum of two pipes is necessary, one for supplying and the other one for returning, such that a distribution loop is created;

- ducts: they are used in air-based central systems, to convey air from a primary or secondary source to the final delivery components. Also in this case, at least two ducts are necessary, one for supply air and one for return air; air distribution loops often recirculate as much indoor air as possible, since it is more economical to heat or cool return air than outdoor air;

- fans: they are used to provide energy input required to overcome friction losses due to the contact of water/air flowing with the duct walls and circulate air through a system. In a HVAC system, several fans are used, such as for supply air, return air and for exhaust air.

One of the most important equipment packages within HVAC systems is represented by the AHU, which comprises the main components for the operation of air-based central HVAC systems. Generally, it consists of a fan, a heating coil or heat source and a cooling coil, an air filter and control devices. The filter is used to remove indoor pollutants from the air stream, the heating or cooling coil receive heating or cooling from a boiler or a chiller and transfer the conditioning effect to the air stream. Control devices such as mixing dampers and valves are often part of the AHU. In addition to the AHU, a variable air volume (VAV) system can be encountered. The main purpose of the VAV system is to change the quantity of air supplied to a space in response to changes in loads. In particular, the supply air is maintained at a constant temperature, and the individual zone thermostat varies the air supply quantity to the zone to maintain the desired temperature condition. Generally, VAV systems do not include any heating function in the main air handling unit, and supplemental baseboard heating or reheat coils are used in exterior zones. The zone thermostat controls the VAV and decides the quantity of air flow necessary for the room. In such way, the total cooling coil load on the cooling coil diminishes almost linearly with diminishing room loads. At this point, in contrast to a reheat system, heating is required on only a fraction of the maximum air flow rate and thus, since only part of the air is heated, the energy requirements are substantially reduced. Moreover, the consequent reduced air flow rates result in substantial fan power savings which implies lower energy consumptions for the overall HVAC system.

2.3 HVAC Model

We consider a HVAC system made of a single zone and the electromechanical part. The basic components considered for the electromechanical part of the HVAC which are directly connected with the room are the cooling coil, the chiller and the chilled water tank, the fan, the supply and return ducts, and the VAV boxes. A schematic representation is shown in Figure 2.1.

The cooling coil is connected to the chiller through the chiller water tank, which regulates the water inserted to the cooling coil. The fan and the VAV box control the air flow rate to the zone, while a three-way valve controls the chilled water mass flow rate. By controlling the inputs, it is possible to reach the desired temperature in the zone and in the cooling coil, achieving thus comfort for the occupants and energy efficiency.

The design of controllers for HVAC systems, which in general are nonlinear and multivariable, requires the knowledge of accurate dynamic models of the system. In particular, the major focus in modeling regards the dynamics of the AHU and the zone with the

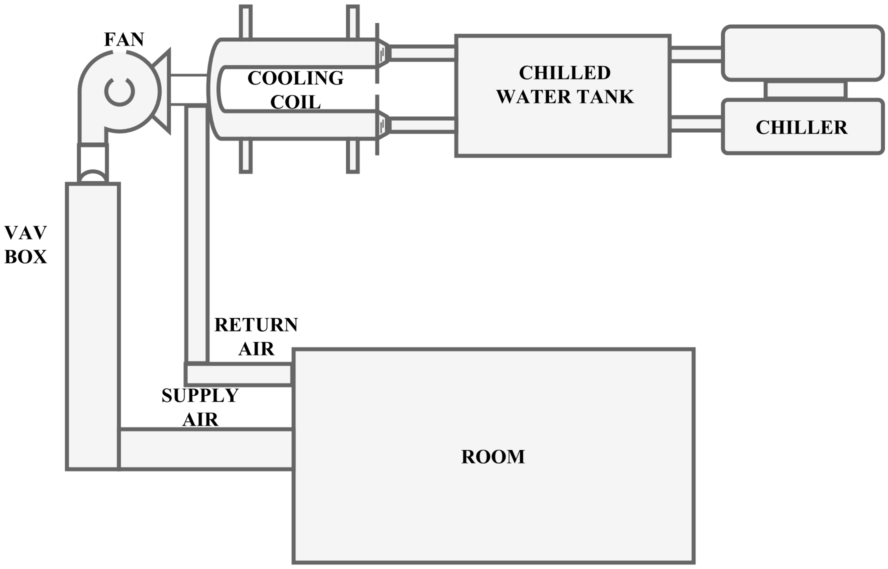


Figure 2.1: Schematic of the HVAC system.

AHU involving dynamics of the fan, heat and mass transfer in the cooling coil and the air distribution system.

Assumption 2.1 The temperature dynamics of the zone, cooling coil and chilled water tank can be modeled based on the fundamental mass and energy conservation equations under the following assumptions:

- the transient and spatial effects are neglected at the components that exchange air;
- at the exterior and interior surface of the zones, supply/return ducts, and so on, the heat transfer is modeled using constant heat transfer coefficients;
- the heat transfer at the chilled water tank with the ambient is modeled using a single constant heat transfer coefficient for all surfaces;
- the axial mixing of water is neglected and the water temperature is constant across the cross section of the tubes.

The temperature dynamic equations of the considered HVAC system are described by the following relations [24]:

$$\begin{aligned}
M_f C_v \frac{dT_f(t)}{dt} &= \rho_a C_{pa}(T_{ao}(t) - T_f(t))Q_a(t) + U_f A_f (T_{amb} - T_f(t)) \\
M_{cc} C_v \frac{dT_{ao}(t)}{dt} &= \rho_a C_{pa}(T_f(t) - T_{ao}(t))Q_a(t) + U_{cc} A_{cc}(T_{amb} - (T_{ao}(t) + T_f(t)))+ \\
&\quad + Q_w \rho_w C_{pw}(T_t(t) - T_{wo}) + \rho_a (h_{fg} - C_{pa})(w_f - w_{ao})Q_a(t) \\
M_t C_v \frac{dT_t(t)}{dt} &= Q_w \rho_w C_{pw}(T_{wo} - T_t(t)) + U_t A_t (T_{amb} - T_t(t)) + \frac{15000}{V_t \rho_w C_{pw}} \chi(t).
\end{aligned} \tag{2.1}$$

The variables used in 2.1 are collected in Table 2.1, together with their measurement units.

The control inputs to the HVAC system are the volumetric flow rate of air Q_a to the room and the chilled water mass flow rate to the storage tank χ, generated by distributed feedback controllers based on some reference signals.

PARAMETERS	MEANING	MEAS. UNIT
T_f	output air temperature from the fan	$^\circ C$
T_{ao}	air temperature of the cooling coil (CC)	$^\circ C$
T_t	temperature of the water in chiller tank (ST)	$^\circ C$
Q_a, Q_w	volumetric flow rate of air into room/ST	m^3/s
χ	chilled water mass flow rate	m^3/s
M_f, M_{cc}	heat mass capacitance of the room/CC	kg
C_v	specific heat at constant volume	$J/kg \cdot K$
U_f, U_{cc}, U_t	heat transfer coefficients of room/CC/ST	$W/m^2 \cdot K$
ρ_a, ρ_w	density of air and water	kg/m^3
A_f, A_{cc}, A_t	area of the room, CC, and ST	m^2
C_{pa}, C_{pw}	specific heat at constant pressure of air and water	$J/kg \cdot K$
h_{fg}	latent heat of water	J/kg
T_{wo}	temperature of output water	$^\circ C$
w_f, w_{ao}	humidity factors	$--$

Table 2.1: Variables used in the temperature dynamic equations

2.3.1 Architecture

The considered HVAC system can be regarded as a set of 2 interconnected nonlinear subsystems that correspond to the electromechanical part, comprising the cooling coil and chiller water tank, and the building zone, i.e. the room. Let us define $T_e = [T_{e,1}, T_{e,2}]^T = [T_{ao}, T_t]^T$, the subsystem that corresponds to the electromechanical part, denoted by Σ_e, can be expressed as:

$$\mathbf{\Sigma_e} : \frac{dT_e(t)}{dt} = A_e T_e(t) + \gamma_e(\chi(t)) + h_e(T_e(t), T_f(t), Q_a(t)) \tag{2.2}$$

where

$$A_e = \begin{bmatrix} -\frac{U_{cc}A_{cc}}{M_{cc}C_v} & \frac{Q_w\rho_w C_{pw}}{M_{cc}C_v} \\ 0 & -\frac{Q_w\rho_w C_{pw}+U_t A_t}{V_t\rho_w C_{pw}} \end{bmatrix}$$

$$\gamma_e(\chi) = \begin{bmatrix} \frac{U_{cc}A_{cc}}{M_{cc}C_v}T_{amb} - \frac{Q_w\rho_w C_w}{M_{cc}C_v}T_{wo} \\ \frac{U_t A_t}{V_t\rho_w C_{pw}}T_{amb} + \frac{Q_w\rho_w C_{pw}}{V_t\rho_w C_{pw}} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{15000}{V_t\rho_w C_{pw}} \end{bmatrix}\chi$$

$$h_e(T_{e,1}, T_f, Q_a) = \begin{bmatrix} h_{e,1}(T_{e,1}, T_f, Q_a) \\ 0 \end{bmatrix} \tag{2.3}$$

$$\begin{aligned} h_{e,1}(T_{e,1}, T_f, Q_a) = {} & \left(\frac{\rho_a C_{pa}}{M_{cc}C_v}Q_a - \frac{U_{cc}A_{cc}}{M_{cc}C_v}\right)T_f + \frac{\rho_a}{M_{cc}C_v}((h_{fg} - C_{pa})(w_f - w_{ao}) + \\ & - C_{pa}T_{e,1})Q_a. \end{aligned} \tag{2.4}$$

Similarly, the dynamics of the air temperature entering to the room can be expressed as

$$\mathbf{\Sigma_f} : \frac{dT_f(t)}{dt} = A_f T_f(t) + \gamma_f(T_f(t), Q_a(t)) + h_f(T_{e,1}(t), Q_a(t)) \tag{2.5}$$

where

$$\begin{aligned} A_f &= -(U_f A_f / M_f C_v) \\ \gamma_f(T_f, Q_a) &= -\frac{\rho_a C_{pa}}{M_f C_v} T_f Q_a + \frac{U_f A_f}{M_f C_v} T_{amb} \\ h_f(T_{e,1}, Q_a) &= \frac{\rho_a C_{pa}}{M_a C_v} T_{e,1} Q_a. \end{aligned} \tag{2.6}$$

The output air temperature from the fan entering into the room is monitored and controlled using a temperature sensor, denoted by S_f, characterized by the output:

$$S_f : y_f(t) = T_f(t) + d_f(t)$$

where d_f denotes the noise corrupting the measurements y_f of sensor S_f. The nonlinear subsystem Σ_e is monitored and controlled using a sensor set S_e that includes two temperature sensors $S_e\{1\}$ and $S_e\{2\}$, characterized by

$$\begin{aligned} S_e\{1\} &: y_{e,1}(t) = T_{e,1}(t) + d_{e,1}(t) \\ S_e\{2\} &: y_{e,2}(t) = T_{e,2}(t) + d_{e,2}(t) \end{aligned} \tag{2.7}$$

where $y_{e,j}$, $j = 1, 2$, is the sensor output and $d_{e,j}$ denotes the noise corrupting the measurements of sensor $S_e\{j\}$.

Assumption 2.2 The measurement noise at each sensor is assumed bounded, i.e. $|d_f(k)| \leq \bar{d}_f$, $|d_{e,j}(k)| \leq \bar{d}_{e,j}$, $j = 1, 2$, where $\bar{d}_f$ and $\bar{d}_{e,j}$ are known constant positive bounds.

We assume that all the sensors are characterised by the same noise bound $\bar{d}$, but this case can be trivially extended for heterogeneous sensors.

2.4 Nonlinear Control Architectures

In this section, we introduce two methods used in this thesis for nonlinear control design. In particular, we deal with feedback linearization, which is one of the most used techniques adopted for nonlinear systems and relies on the cancellation of the nonlinearities by the combined use of feedback and change of coordinates, and backstepping, which stabilizes the origin using state feedback, but has the drawback that, even with a simple second-order system, the control algorithm becomes complex ([25]).
We first give a brief description of these methods for those readers who are not familiar with nonlinear control techniques, underlying the most important steps necessary for their implementation, and then we show how we have designed them according to our system.

2.4.1 Feedback Linearization

Consider the single-input-single-output (SISO) nonlinear system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{2.8}$$

where $u \in \mathbf{R}^1$, $y \in \mathbf{R}^1$, $x \in \mathbf{R}^n$ and f, g and h are sufficiently smooth in a domain $D \subset \mathbf{R}^n$. The time derivative of $y = h(x)$ is given by

$$\dot{y} = \frac{\partial h}{\partial x}(x)f(x) + \frac{\partial h}{\partial x}(x)g(x)u. \tag{2.9}$$

If $\frac{\partial h}{\partial x}(x)g(x) \neq 0$ for any $x \in D_0 \subset D$, then the nonlinear system is said to have *relative degree one* in D_0. Hence, the control variable u appears explicitly in the differential equation for the first derivative of the output y. If $\frac{\partial h}{\partial x}(x)g(x) = 0$, i.e. u does not directly affect $\dot{y}$, then the output is differentiated until the input u appears explicitly. The derivative of h with respect to f is defined as

$$L_f h(x) = \frac{\partial h}{\partial x}(x)f(x) \tag{2.10}$$

and it is called *Lie derivative.* Accordingly, further derivatives can be written as shown below

$$\begin{aligned} L_f^k h(x) &= L_f L_f^{k-1} h(x) = \frac{\partial}{\partial x}(L_f^{k-1}h(x))f(x) \\ L_g L_f^k h(x) &= \frac{\partial}{\partial x}(L_f^k h(x))g(x). \end{aligned} \tag{2.11}$$

The definition of the Lie derivative implies that if

$$L_g h(x) = \frac{\partial h}{\partial x}(x)g(x) = 0 \tag{2.12}$$

only the derivatives until $L_g L_f^{r-1} h(x) \neq 0$ are taken, since u first appears explicitly in the equation for $y^{(r)}$, being r the degree of derivative of the output.
The nonlinear system 2.8 is said to have relative degree r if the following conditions are satisfied for any $x \in D_0$:

$$\begin{aligned} L_g L_f^i h(x) &= 0, \qquad i = 0, 1, 2, \ldots, r-2 \\ L_g L_f^{r-1} h(x) &\neq 0. \end{aligned} \tag{2.13}$$

If the system has relative degree r, then

$$y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x) u. \tag{2.14}$$

Finally, the system is input-output linearizable, since the state feedback control

$$u = \frac{1}{L_g L_f^{r-1} h(x)}[-L_f^r h(x) + v] \tag{2.15}$$

gives the linear input-output mapping

$$y^{(r)} = v. \tag{2.16}$$

Using differential geometric tools, it can be shown that for a system with relative degree r there exists a diffeomorphism $z = T(x)$, with $T(0) = 0$ that transforms the nonlinear system 2.8 into the input-output linearizable canonical form

$$\begin{aligned} \dot{\eta} &= \phi(\eta, \xi) \\ \dot{\xi} &= A_0 \xi + B_0 \beta_0^{-1}(\eta, \xi)[u - \alpha_0(\eta, \xi)] \\ y &= C_0 \xi \end{aligned} \tag{2.17}$$

where $z = [\eta^T, \xi^T]^T$, with $\eta \in \mathbf{R}^{n-r}$, $\xi \in \mathbf{R}^r$, and (A_0, B_0, C_0) is a canonical form representation of a system consisting of r integrators in series. The transformed system described by 2.17 is said to be in *normal form.* The change of variables decomposes the original nonlinear system in two parts, the η variables which characterize the internal dynamics of the system, and the ξ-dynamics, which can be linearized and controlled by using a feedback controller of the form

$$u = \alpha_0(\eta, \xi) + \beta_0(\eta, \xi)v \quad = -\frac{L_f^r h(x)}{L_g L_f^{r-1} h(x)} + \frac{1}{L_g L_f^{r-1} h(x)} v \tag{2.18}$$

where v is chosen according to the desired convergence rate of the ξ-dynamics or to achieve reference input tracking. The feedback linearizing control functions α_0 and β_0 are computed based on the Lie derivatives of the output.

In the tracking problem, we assume that the control objective is for $y(t)$ to track a desired signal $y_d(t)$. Let $e(t) = y(t) - y_d(t)$ be the tracking error, in the feedback control law 2.18, v is chosen to be

$$v = y_d^{(r)} - k_{r-1}e^{(r-1)} - \ldots - k_1\dot{e} - k_0 e. \tag{2.19}$$

Thus, the control law 2.18 results in the linear error dynamics

$$e^{(r)} + k_{r-1}e^{(r-1)} + \ldots + k_1\dot{e} + k_0 e = 0. \tag{2.20}$$

Selecting appropriately the coefficients $\{k_0, k_1, \ldots, k_{r-2}, k_{r-1}\}$, the roots of the characteristic equation

$$s^r + k_{r-1}s^{r-1} + \ldots + k_1 s + k_0 = 0 \tag{2.21}$$

can be arbitrarily assigned and thus the tracking error can be made to converge to zero asymptotically.

2.4.2 Backstepping

Consider a second-order system

$$\begin{aligned} \dot{x}_1 &= f(x_1) + g(x_1)x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{2.22}$$

where $(x_1, x_2) \in \mathbf{R}^2$ is the state, $g(x_1) \neq 0$ for x_1 and u is the control input. The goal is to define a feedback control algorithm to cause x_1 to converge to y_d. The idea behind backstepping procedure is that the tracking problem can be solved if the control input u is able to force x_2 such that it satisfies

$$x_2 = \frac{1}{g(x_1)}[-f(x_1) - k_1(x_1 - y_d) + \dot{y}_d], \tag{2.23}$$

with $k_1 > 0$. In this case, x_1 satisfies $\dot{x}_1 - \dot{y}_d = -k_1(x_1 - y_d)$, which implies that x_1 converges to y_d. Doing so, x_2 becomes a *virtual control* input for the x_1 subsystem. Let us call such virtual variable $\alpha(x_1, y_d, \dot{y}_d)$, defined as 2.23

$$\alpha(x_1, y_d, \dot{y}_d) = \frac{1}{g(x_1)}[-f(x_1) - k_1(x_1 - y_d) + \dot{y}_d]. \tag{2.24}$$

Let $z_1 = x_1 - y_d$, then

$$\dot{z}_1 = -k_1 z_1 + g(x_1)(x_2 - \alpha) \tag{2.25}$$

Moreover

$$z_2 = x_2 - \alpha(x_1, y_d, \dot{y}_d) \tag{2.26}$$

which derivative is

$$\begin{aligned} \dot{z}_2 &= u - \frac{\partial \alpha}{\partial x_1}\dot{x}_1 - \frac{\partial \alpha}{\partial y_d}\dot{y}_d - \frac{\partial \alpha}{\partial \dot{y}_d}\ddot{y}_d \\ &= v \end{aligned} \tag{2.27}$$

where v is referred to as *modified control input.* With the new change of variables, system 2.22 becomes

$$\begin{aligned} \dot{z}_1 &= -k_1 z_1 + g(x_1) z_2 \\ \dot{z}_2 &= v. \end{aligned} \tag{2.28}$$

From the definition of v, the feedback control law u is given by

$$\begin{aligned} u &= \frac{\partial \alpha}{\partial x_1}\dot{x}_1 + \frac{\partial \alpha}{\partial y_d}\dot{y}_d + \frac{\partial \alpha}{\partial \dot{y}_d}\ddot{y}_d - z_1 g(x_1) - k_2 z_2 \\ &= \frac{\partial \alpha}{\partial x_1}[f(x_1) + g(x_1)x_2] + \frac{k_1 \dot{y}_d}{g(x_1)} + \frac{\ddot{y}_d}{g(x_1)} - z_1 g(x_1) + \\ &- k_2[x_2 + \frac{1}{g(x_1)}(f(x_1) + k_1(x_1 - y_d) - \dot{y}_d)] \end{aligned} \tag{2.29}$$

which ensure perfect tracking of y_d by x_1.
The drawbacks of backstepping control is that, even with a simple second-order systems, the control algorithm becomes complex. For higher order systems of the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + g_1(x_1)x_2 \\ \dot{x}_2 &= f_2(x) + g_2(x)u \end{aligned} \tag{2.30}$$

the algorithm can be still applied, assuming that V_1 is unbounded and it is such that

$$\frac{\partial V_1}{\partial z_1}[f_1 + g_1\alpha - \dot{y}_d] \leq W_1(z_1), \tag{2.31}$$

where $W_1(z_1)$ is a positive definite function.

2.4.3 Controller Design for the HVAC System

In order to obtain the desired temperature for the room, we apply the feedback linearization controller described in Section 2.4.1 to the I/O system describing the dynamics of the temperature at the fan:

$$\begin{aligned} \dot{T}_f(t) &= -\frac{U_f A_f}{M_f C_v}T_f(t) - \frac{\rho_a C_{pa}}{M_f C_v}T_f(t)Q_a(t) + \frac{U_f A_f}{M_f C_v}T_{amb} + \frac{\rho_a C_{pa}}{M_f C_v}T_{e,1}(t)Q_a(t) \\ y_d(t) &= T_d \end{aligned} \tag{2.32}$$

where $y_d(t) = T_d = 22°C$ represents the desired temperature of the room.
Let $x(t) = T_f(t)$, we apply the feedback linearization algorithm described in Section 2.4.1 and we obtain

$$\begin{aligned} e(t) &= x(t) - y_d(t) \\ \dot{e}(t) &= \dot{x}(t) \end{aligned} \tag{2.33}$$

The state feedback control $u(t) = Q_a(t)$ is given by

$$u(t) = \frac{M_f C_v}{\rho_a C_{pa}(-x(t) + T_{e,1}(t))}[(\frac{U_f A_f}{M_f C_v})x(t) - \frac{U_f A_f}{M_f C_v}T_{amb} - ke(t)] \tag{2.34}$$

where k is the gain of the controller which determines the rate of convergence of the tracking error.

For what concerns the electromechanical part, it consists of two sensors measuring the temperature of the air existing the cooling coil $T_{e,1}$ and the temperature of the chilled water in the tank $T_{e,2}$, while the air entering in the room is controlled by a sensor at the fan. The control input to the zones are the volumetric flow rate of air Q_a and the chilled water mass flow rate to the storage tank χ. A feedback linearization controller is adopted for the room, which is responsible to keep the temperature at $22°C$. The backstepping controller is applied for maintaining the temperature of the output air of the cooling coil at $10°C$. Moreover, the room temperature controller uses the measurements of the temperature of the cooling coil, while the controller of the electromechanical part uses *a priori* known set points of the temperature of the room, as well as the air flow rate, i.e. the control input, of the room.

Given the electromechanical system obtained by 2.3

$$\begin{aligned}\dot{T}_{e,1}(t) &= -\frac{U_{cc}A_{cc}}{M_{cc}C_v}T_{e,1}(t) + \frac{Q_w p_w C_{pw}}{M_{cc}C_v}T_{e,2}(t) + \frac{U_{cc}A_{cc}}{M_{cc}C_v}T_{amb} - \frac{Q_w p_w C_{pw}}{M_{cc}C_v}T_{wo}+\\ &+\left(\frac{\rho_a C_{pa}}{M_{cc}C_v}Q_a(t) - \frac{U_{cc}A_{cc}}{M_{cc}C_v}\right)T_f(t) + \frac{\rho_a}{M_{cc}C_v}((h_{fg}-C_{pa})(w_f - w_{ao})+\\ &- C_{pa}T_{e,1})Q_a(t)\\ \dot{T}_{e,2}(t) &= -\frac{Q_w\rho_w C_{pw} + U_t A_t}{M_{cc}C_v}T_{e,2}(t) + \frac{U_t A_t}{V_t\rho_w C_{pw}}T_{amb} + \frac{Q_w\rho_w C_{pw}}{V_t\rho_w C_{pw}}T_{wo} + \frac{15000}{V_t\rho_w C_{pw}}\chi(t)\end{aligned} \tag{2.35}$$

let $x_1(t) = T_{e,1}(t)$ and $x_2(t) = T_{e,2}(t)$, system 2.35 can be rewritten in terms of the new variables as

$$\begin{aligned}\dot{x}_1(t) &= (-\frac{U_{cc}A_{cc}}{M_{cc}C_v} - \frac{\rho_a C_{pa}}{M_{cc}C_v}Q_a(t))x_1(t) + \frac{Q_w\rho_w C_{pw}}{M_{cc}C_v}x_2(t) + \frac{U_{cc}A_{cc}}{M_{cc}C_v}T_{amb} - \frac{Q_w p_w C_{pw}}{M_{cc}C_v}T_{wo}+\\ &+\frac{\rho_a C_{pa}Q_a}{M_{cc}C_v} - \frac{U_{cc}A_{cc}}{M_{cc}C_v}T_f(t) + \frac{\rho_a}{M_{cc}C_v}((h_{fg}-C_{pa})(w_z - w_{ao}))Q_a(t)\\ \dot{x}_2(t) &= -\frac{Q_w\rho_w C_{pw} + U_t A_t}{M_{cc}C_v}x_2(t) + \frac{U_t A_t}{V_t\rho_w C_{pw}}T_{amb} + \frac{Q_w\rho_w C_{pw}}{V_t\rho_w C_{pw}}T_{wo} + \frac{15000}{V_t\rho_w C_{pw}}\chi(t)\end{aligned} \tag{2.36}$$

For easier computation, we define the following variables:

$$\begin{aligned}f(x_1) &= (-\frac{U_{cc}A_{cc}}{M_{cc}C_v} - \frac{\rho_a C_{pa}}{M_{cc}C_v}Q_a)\\ g(x_1) &= \frac{Q_w\rho_w C_{pw}}{M_{cc}C_v}\\ c &= \frac{U_{cc}A_{cc}}{M_{cc}C_v}T_{amb} - \frac{Q_w p_w C_{pw}}{M_{cc}C_v}T_{wo} + \frac{\rho_a C_{pa}Q_a}{M_{cc}C_v}(1 + (h_{fg}-C_{pa})(w_f - w_{ao})) - \frac{U_{cc}A_{cc}}{M_{cc}C_v}T_f\\ u_\chi &= -\frac{Q_w\rho_w C_{pw} + U_t A_t}{M_{cc}C_v}x_2 + \frac{U_t A_t}{V_t\rho_w C_{pw}}T_{amb} + \frac{Q_w\rho_w C_{pw}}{V_t\rho_w C_{pw}}T_{wo} + \frac{15000}{V_t\rho_w C_{pw}}\chi\\ \alpha &= \frac{M_{cc}C_v}{Q_w\rho_w C_{pw}}[(\frac{U_{cc}A_{cc}}{M_{cc}C_v} + \frac{\rho_a C_{pa}}{M_{cc}C_v}Q_a)x_1 - k_1(x_1 - y_d)]\\ \frac{\partial\alpha}{\partial x_1} &= \frac{M_{cc}C_v}{Q_w\rho_w C_{pw}}[\frac{U_{cc}A_{cc}}{M_{cc}C_v} + \frac{\rho_a C_{pa}}{M_{cc}C_v}Q_a - k_1]\end{aligned} \tag{2.37}$$

Note that u_χ is related to the control input signal $u = \chi$ through the following

$$u = \frac{u_\chi + \frac{Q_w \rho_w C_{pw} + U_t A_t}{M_{cc} C_v} x_2 - \frac{U_t A_t}{V_t \rho_w C_{pw}} T_{amb} - \frac{Q_w \rho_w C_{pw}}{V_t \rho_w C_{pw}} T_{wo}}{\frac{15000}{V_t \rho_w C_{pw}}}. \tag{2.38}$$

System 2.36 can be rewritten as follows:

$$\begin{aligned} \dot{x}_1 &= f(x_1)x_1 + g(x_1)x_2 + c \\ \dot{x}_2 &= u_\chi \end{aligned} \tag{2.39}$$

We apply the algorithm of Section 2.4.2 and since y_d is constant, its derivatives are equal to zero; hence:

$$u_\chi = \frac{\partial \alpha}{\partial x_1} \dot{x}_1 - z_1 g(x_1) - k_2 z_2. \tag{2.40}$$

Once u_χ is computed, it is possible to obtain u trough 2.38.

2.5 Discretization

The HVAC system considered is described by continuous-time equations which have to be converted into discrete-time ones. We give an overview of the discretization method used, namely the Euler's explicit method, for a linear system, but the same results apply to non-linear systems.

In general, given a continuous time-invariant plant of the type

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.41}$$

with $x(t) \in \mathbf{R}^n$, measured output $y(t) \in \mathbf{R}^p$ and control input $u(t) \in \mathbf{R}^m$.

For digital control purposes it is desired to define a discrete time index k such that

$$t = k\Delta \tag{2.42}$$

with Δ sampling period. Then, the discrete control input u_k is to be switched at times $k\Delta$, $k = 0, 1, \ldots$. The usual procedure for controlling the plant is to hold the control input $u(t)$ constant between control switchings. The continuous plant input $u(t)$ is given in terms of the discrete control $u(k)$ by

$$u(t) = u(k), \qquad kT \leq t \leq (k+1)T. \tag{2.43}$$

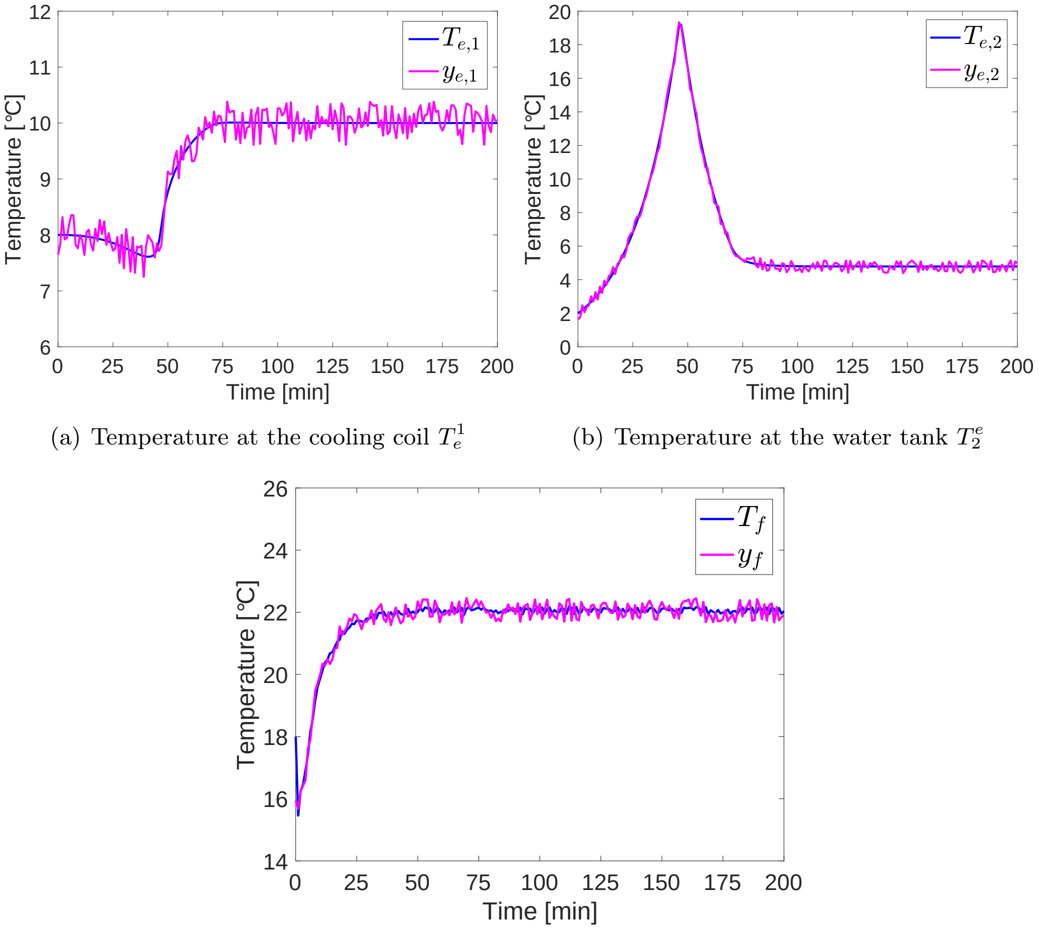
Given 2.41, the unknown variable is $x(t)$ and the solution for the problem will be given by a set of points sampled over time $x(k)$, $k \in \mathbf{N}$, $k = 0, 1, \ldots$. Adopting Euler's explicit method, we have that the state derivative may be approximated through the following:

$$\dot{x}(t) = \frac{x(k+1) - x(k)}{\Delta} \tag{2.44}$$

with Δ sampling period. Therefore, the continuous-time state equation is approximately equal to

$$\dot{x}(t) = \frac{x(k+1) - x(k)}{\Delta} = Ax(k) + Bu(k) \tag{2.45}$$

The main advantage of the explicit method is that it directly gives $x(k+1)$. The pseudocode of the procedure is shown in Algorithm 1, where t_0 and x_0 represent respectively



(a) Temperature at the cooling coil T_e^1

(b) Temperature at the water tank T_2^e

(c) Temperature at the fan T_f

Figure 2.2: Temperature dynamics of the cooling coil, water tank and the fan obtained applying respectively a backstepping controller and a feedback controller.

Algorithm 1 `ExplicitEuler`$(f, \Delta, t_0, t_f, x_0)$

```
*Initialization*
$t = t_0$
$x = x_0$
while $t < t_f$ do
    for $t = 1, 2, \ldots$ do
        for $k = 1, 2, \ldots$ do
            if $t = k\Delta$ then
                $x(k+1) = x(k) + \Delta f(k, x(k))$
            end
        end
    end
end
```

the initial time and state, while t_f is the final time.

In our case, given the slow dynamics of the system, we have sampled the continuous-time system every $\Delta = 60s$. Applying the aforementioned control inputs χ and Q_a, we obtain respectively the temperature at the cooling coil $T_{e,1}$ and at the water tank $T_{e,2}$, shown in Figure 2.2(a) and 2.2(b) (backstepping controller), and the temperature at the fan T_f (feedback linearization), shown in Figure 2.2(c). Note that when the temperature of the cooling coil decreases, the temperature of the water in the tank increases, aiming to increase the temperature in the cooling coil.

2.6 Heat Diffusion Model

Once we have dealt with the general model of the HVAC system, we focus our attention to the room. Our purpose is to model the diffusion of the heat in the space as a function of the distance from the heating source, represented by the fan, which introduces in the room air at temperature T_f. The tool exploited to achieve such purpose is represented by the heat equation, which we briefly recall below before obtaining the analytical expressions related to our problem.

2.6.1 Heat Equation

The heat equation is a parabolic partial differential equation that describes the distribution of heat, i.e. the variation in temperature, in a given region over time [26]. Given a function $T(t)$ representing the temperature and defined in a coordinate system, given the time variable t, the heat equation is expressed by:

$$\frac{\partial T(t)}{\partial t} = k\nabla^2 T(t) \tag{2.46}$$

where k is a positive constant defining the thermal diffusivity and ∇ is the Laplace operator. The equation describes how the function changes over time as heat spreads throughout space.

The heat equation is derived from Fourier's law and conservation of energy and if the medium considered is not the whole space, in order to uniquely solve it, it is necessary to specify some boundary conditions on the function $T(t)$. According to the boundary conditions, different solutions can be encountered. We first present some basic cases, which will help understanding the solution obtained for our system.

2.6.2 Homogeneous Boundary Conditions

Let us consider the heat equation for one space variable, which could be the case of modelling heat conduction in a rod [27]. The equation is given by 2.6.3, where $T(t)$ in this case is a function of two variables P and t. In particular, P is the space variable and it is such that $P \in [0, L]$, with L being the length of the rod, while t is the time variable such that $t \geq 0$.

Assume the initial condition

$$T(t)(P,0) = f(P), \quad \forall P \in [0, L] \tag{2.47}$$

with f given, and boundary conditions

$$T(0,t) = 0 = T(L,t), \quad \forall t > 0. \tag{2.48}$$

Applying the separation of variables principle, the solution will have the following form:

$$T(P,t) = (B\sin(\sqrt{\lambda}P) + C\cos(\sqrt{\lambda}P))Ae^{-\lambda kt} \tag{2.49}$$

The initial conditions imply that $C = 0$ and for some positive integer n, $\sqrt{\lambda} = n\frac{\pi}{L}$. Finally the solution is given by:

$$T(P,t) = \sum_{n=1}^{\infty} B_n \sin(\frac{n\pi P}{L})e^{-\frac{n^2\pi^2kt}{L^2}} \tag{2.50}$$

where

$$B_n = \frac{2}{L}\int_0^L f(P)\sin(\frac{n\pi x}{L})dP. \tag{2.51}$$

2.6.3 Stationary Heat Equation

The stationary heat equation does not depend on time. The condition that has to be satisfied is that

$$\frac{\partial T(t)}{\partial t} = 0 \tag{2.52}$$

and it represents the equilibrium temperature distribution. It is a function of the space only, and it has to satisfy the heat equation. Thus

$$\frac{\partial^2 T(t)}{\partial P^2} = 0$$

Integrating twice with respect to P, we find that $T(P)$ must be in the form of a polynomial of first degree:

$$T(P) = AP + B.$$

2.6.4 Insulated and Nonhomogeneous Boundary Conditions

The general solution of a partial differential equation is dependant of boundary conditions and it will assume different forms under different sets of them. Once having shown how to solve the heat equation in the simplest cases, we deal with a more realistic case, i.e when one of the boundary conditions is no longer equal to zero and it takes a finite value, while the second condition implies perfect insulation at the other end of the medium at position $P = L$, so that no heat can escape to the outside environment, i.e. $\frac{\partial T(t)}{\partial P} = 0$. This is the case that we consider for the room model in the HVAC system.

The air is introduced to the room from a fan connected to the VAV box with constant velocity at a temperature $T_f(t)$ which depends on Q_a, representing the control signal computed by the feedback linearization controller. T_f is thus one of the boundaries conditions, i.e. the nonhomogeneous one. The initial temperature in a given position of the room corresponds to the initial condition $T(P,0)$ (assumed to be constant over the space), while the boundaries conditions are:

$$\begin{aligned} &T(0,t) = T_f(t) \\ &\frac{\partial T}{\partial P}(L,t) = 0. \end{aligned} \tag{2.53}$$

Given the heat equation, the initial condition and the boundary conditions, assuming $k = 1$, it follows that:

$$T(P,t) = T_{ss}(P,t) + T_{tr}(P,t) \tag{2.54}$$

where $T_{ss}(P,t)$ is called *steady-state solution* and $T_{tr}(P,t)$ is the *transient solution.* The steady state solution is given by:

$$T_{ss}(P,t) = AP + B \tag{2.55}$$

where A and B are two positive constants, which values are obtained by imposing the boundary conditions. It follows that:

$$T_{ss}(P,t) = T_f(t) \tag{2.56}$$

which can be interpreted as the equilibrium temperature reached for $t \to \infty$.
In order to solve the second part of (2.54), it is necessary to impose the boundary conditions and the initial condition with respect to $T_{tr}(P,t) = T(P,t) - T_{ss}(P,t)$:

$$\begin{aligned} &T_{tr}(0,t) = T_f(t) - T_f(t) = 0 \\ &\frac{\partial T_{tr}}{\partial P}(L,t) = 0 - 0 = 0 \\ &T_{tr}(P,0) = T(P,0) - T_f(0) \end{aligned} \tag{2.57}$$

The boundary conditions for the transient component are now homogeneous and the solution can be written as

$$T_{tr}(P,t) = A\cos pP + B\sin pP \tag{2.58}$$

Imposing the initial conditions, we obtain:

$$\begin{aligned} &T_{tr}(0,t) = A = 0 \\ &\frac{\partial T_{tr}}{\partial P}(L,t) = 0 \Leftrightarrow \\ &Bp\cos(pP)_{|(L,t)} = 0 \Leftrightarrow \\ &Bp\cos(pL) = 0 \Leftrightarrow \\ &\cos(pL) = 0 \Leftrightarrow p = \frac{(2n-1)\pi}{2L}, \qquad n = 1,2,... \end{aligned} \tag{2.59}$$

The solution is then given by

$$T_{tr}(P,t) = \sum_{n=1}^{\infty} B_n(t)\sin(\frac{(2n-1)}{L}\frac{\pi}{2}P)e^{-(\frac{2n-1}{L})^2\frac{\pi^2}{4}t} \tag{2.60}$$

where

$$B_n(t) = \frac{2}{L}\int_0^L (T(P,t-1) - T_f(t))\sin(\frac{(2n-1)}{L}\frac{\pi}{2}P)dP \tag{2.61}$$

and $T(t-1)$ is the temperature reached by each sensor at the end of the previous instant thanks to the diffusion. Finally, the complete solution including also the stationary part is:

$$T(P,t) = T_f(t) + \sum_{n=1}^{\infty} B_n(t)\sin(\frac{(2n-1)}{L}\frac{\pi}{2}P_i)e^{-(\frac{2n-1}{L})^2\frac{\pi^2}{4}t} \tag{2.62}$$

with $B_n(t)$ defined as in (5.2).
Note that the solution has been computed in the continuous-time, but it needs to be discretized according to what already done for the rest of the system. We obtain:

$$T(P,k) = T_f(k) + \sum_{n=1}^{\infty} B_n(k) \sin(\frac{(2n-1)}{L}\frac{\pi}{2}P) e^{-(\frac{2n-1}{L})^2 \frac{\pi^2}{4} k} \tag{2.63}$$

where

$$B_n(k) = \frac{2}{L}\int_0^L (T(P,k-1) - T_f(k)) \sin(\frac{(2n-1)}{L}\frac{\pi}{2}P) dP. \tag{2.64}$$

3

Industrial Wireless Sensor Networks

The room of the HVAC system described in the previous chapter can be accurately monitored through a set of N sensors. Sensors can be deployed in a random way, or such to satisfy some optimality criteria (as Voronoi's optimal partition), or simply considering the topology of the environment and existence of possible obstacles. Monitoring can be fulfilled through an IWSN, which properties and advantages over standard WSN will be explained in the first part of the chapter.

When considering sensor networks, one must take into account the possibility of faults, anomalies and failures for the sensors, which can be caused by communication issues, degradation, high background noise levels, radio frequency interference, de-synchronization, battery exhaustion, or dislocation [9]. For this reason, fault diagnosis procedures need to be included in the network in order to guarantee the correct operation of the system and avoid the loss of performance of the entire network. This topic will be treated in the second part of the chapter, when some basic concepts related to Fault Tolerance are presented. To this aim, it is shown that fault diagnosis capabilities of the network can be enhanced exploiting the possibility for the nodes to cluster according to some common features. In particular, the clustering approach recently proposed by [22] is recalled in the last part of the chapter, preceeded by a brief overview of graphy theory resuming the main definitions and properties necessary for the explanation of the clustering algorithm.

3.1 Definition and Applications

A WSN can generally be described as a network of nodes that cooperatively sense and control the environment, enabling interaction between persons or computers and the surrounding space. WSNs nowadays usually include a large number of sensor nodes deployed into a monitoring area, forming networks through their self-organization capabilities. Moreover, thanks to the technologic advances, the cost of WSN equipment has dropped dramatically, and their applications spread from military target tracking and surveillance to biomedical monitoring and hazardous environment exploration and sensing. This has been enabled by the availability, particularly in recent years, of sensors that are smaller, cheaper, and intelligent. Sensors are equipped with wireless interfaces with which they can communicate with one another to form a network and in some cases, organize into groups according to some common features. Among WSNs, we introduce the emerging class of IWSNs which, besides having all the properties of WSNs, show an enhanced capacity to face specific constraints related to applications in the industrial production, improving the efficiency

of production processes while further limiting implementation costs. In such a contest, IWSNs have to deal with several challenges such as the reliability and robustness in harsh environments, as well as the ability to properly execute and achieve their specific goal in parallel with all the other industrial processes.
Based on the requirements of the industrial field, IWSN applications can be classified into three groups [9]:

- *Environmental sensing.* In these scenarios, the general purpose is the efficient information gathering, used both for prevention and analysis. The migration from the wired sensor networks to their wireless counterpart brings numerous advantages by facilitating the deployment and the information gathering process. Moreover, due to their small size, sensors can be easily and quickly deployed over large scales at low cost. The wireless features that make them independent from any costly and fixed infrastructure also contribute to their success. Hence, environmental sensing is a very broad area that include, above all, monitoring for disaster prevention, like volcano monitoring and healing operations when sensors indicate a critic area.

- *Condition monitoring.* This group generally covers the problems of structural condition monitoring, providing both the structure health information and machine condition monitoring including possible automatic maintenance. Furthermore, structural monitoring systems manage to detect system damages before possible failures, limiting the time that the affected component is out of service and thus increasing the efficiency. Without automated monitoring system, it would be necessary to schedule regular system checks and preventively replace production equipments. All these issues are avoided when WSNs intended for condition monitoring are used, since, in comparison with wired sensing systems, they are easily deployable and reconfigurable, reduce the system installation and condition monitoring cost in general.

- *Process automation.* The last group of applications provides the users with the information regarding the resources for the production and service provision. One of the most important issues from the user perspective is the production performance monitoring, evaluation and improvement, that are achieved through IWSNs.

Figure 3.3 provides a schematic representation of the applications mentioned above.
Finally, among the challenges the IWSN has to deal with, we recall the deployment and the set up of the network. Indeed, the environment where IWSNs are deployed in order to monitor environmental or production processes is extremely dynamic, it can depend on the specific product, the phase of life of the product and the kind of service provision considered and, according to the application, different constraints are imposed on the monitoring system.
We summarize below the major advantages of IWSNs:

- *Cost.* One of the main reason for deploying WSN in process automation is that they are easier and less costly to install than traditional wired systems.

- *Flexibility.* WSN can be used for measuring different process and quantities, without any need to install further components.

- *Emerging Applications.* The spread and heterogeneity of WSNs allows their application in different fields and for various purposes.

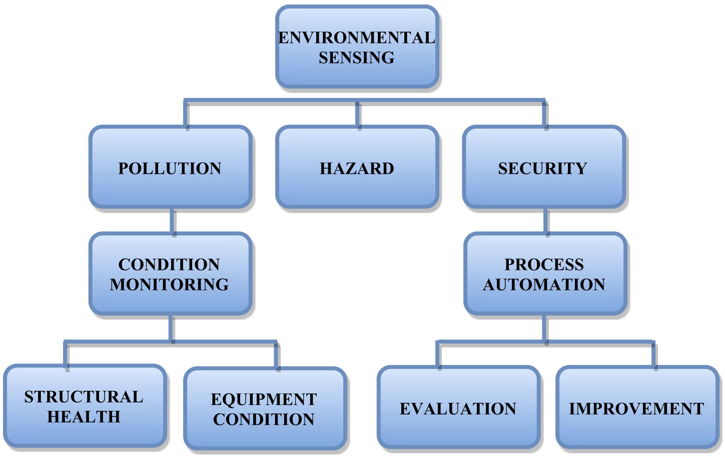


Figure 3.1: Taxonomy of IWSN applications.

- *Reliability.* It refers to the monitoring system that has to provide accurate and real-time information regarding the monitored process even in harsh industrial environments under noise, disturbances and system uncertainties. The information gathered during the monitoring process is fundamental for the correct system operations, given that errors in such sense may result in fatal consequences for the system. Therefore, system *security* represents another important feature that has to be accomplished.

In the industrial environment, the networks should be able to autonomously guarantee robust operations and ensure the safety of personnel, machinery and propriety as well as fast detection and recovery from anomalies and failures. To this purpose, fault detection algorithms have to be developed capable of identifying sensors and actuators which operating conditions are different from those expected.

3.2 Fault Tolerant IWSNs

Sensors are installed in order to monitor and measure critical parameters such as temperature, pressure, ecc. and can be deployed in harsh environments and hazardous spaces. Consequently, they are vulnerable to faults, anomalies and failures which may compromise the operation of the whole network. To cope with these issues, *sensor fault detection* and *sensor fault diagnosis* procedures need to be carried on in order to allow the plant to learn about any potential problems associated to the sensors. Indeed, indications concerning which faults are developing can help avoid system breakdown, mission abortion and catastrophes. In particular, there is an increasing need for the systems to continue operating acceptably to fulfil specified functions following faults in the system. A control system with this kind of fault-tolerance capability is defined as a *fault-tolerant* system. There may be some graceful performance degradation for a fault-tolerant system to operate under a faulty condition, however the primary objective is to maintain system operation, to lately repair the system or to use alternative measures to avoid catastrophes.
When faults are relevant in terms of effects they can produce on the rest of the network and when they can compromise the correct operations, an isolation procedure has to be carried on in order to identify the source of the faults in the network, and take appropriate steps to prevent or recover from the detected fault according to the severity of each malfunction. Therefore, we distinguish between fault detection, which refers to the process of recogni-

tion that a fault has occurred in a node of the system, i.e. it implies a binary decision, "either something has gone wrong or everything is fine", and fault diagnosis which aims to detect and determine the source of the fault (*fault isolation*) and assess their significance and severity.
There are several methods that deal with fault detection and isolation and they can be classified into *physical redundancy*, *model-based* approaches and *model-free* ones. While the first method is in many cases not practical because it involves high cost of installations and maintenance, in addition to problems of limited space, the second one is used in most of the cases. Model-based techniques imply a priori knowledge of the system and they use the relationships between several measured variables to extract information on possible changes caused by faults. In Figure 3.2 it is possible to observe a general scheme representing a process model-based fault detection. In particular, when model-based techniques are used, the input signals u and the output signals y are represented by a mathematical process model. The comparison between the nominal values obtained by the model and some features allows to perform detection of anomalies and changes in the system behaviour.

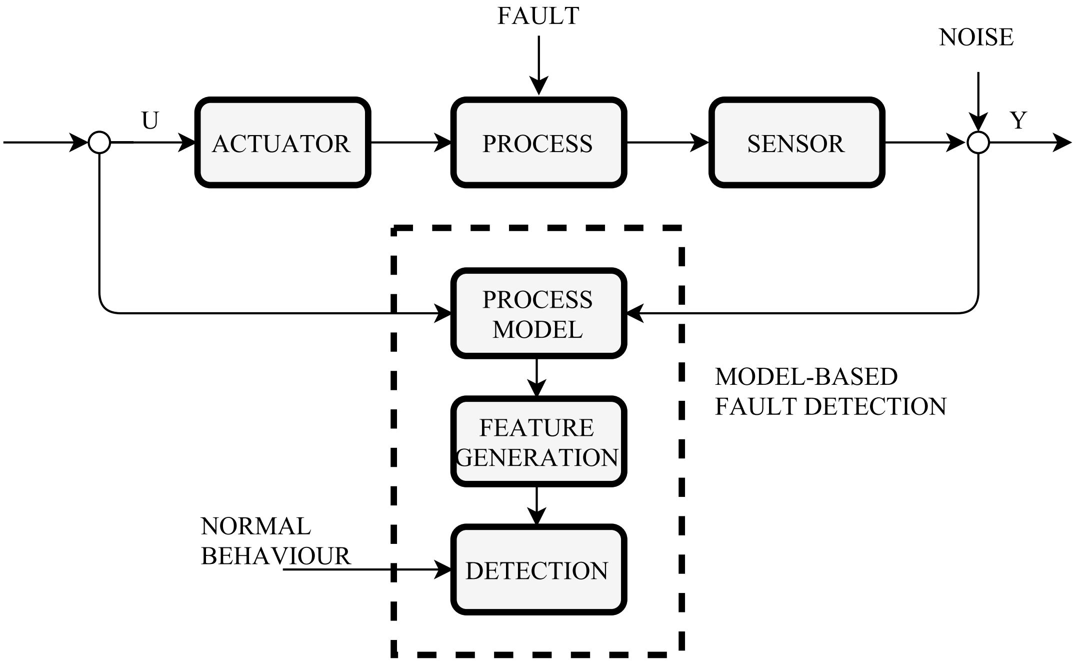


Figure 3.2: Scheme of process model-based fault detection.

Model-based techniques can be further classified as quantitative or qualitative ([28]); the first category relies on nominal mathematical signal and process models that can generate faulty features, while the second one uses symbolic and/or qualitative system representations. A list containing the main definitions concerning model-based fault diagnosis are collected in Appendix A.
Among the quantitative model-based approaches used in FDI we recall *parity equations* and *observers*. The first ones are based on the idea that a straightforward way to detect process faults is to compare the process behaviour with a process model describing the nominal behaviour, i.e. the behaviour of the system in healthy conditions. The difference of signals between the process and the model is referred to as the *residual*, which aims to describe the lack of consistency between the two outputs, contributing in this way to

detect the existence of anomalies. The residual should be zero-valued when the system is normal, and should diverge from zero when a fault occurs in the system. This zero and non-zero property of the residual is used to determine whether or not faults have occurred. The other alternative for model-based fault detection is to use state observers, widely adopted for nonlinear systems, which consider the output error between a measured process output and an adjustable model output.

Qualitative model-based techniques are generally used by the artificial intelligence community, and they rely on causal models, signed digraphs, bond graphs, fault trees, etc. In all these cases, no analytical mathematical expressions are considered and they are tuned on long term experimental data. For this reason, they are widely used to deal with large-scale systems.

Model-based approaches rely on analytical redundancies, need mathematical models necessary for the analytical redundancies, state estimation and parameter identification, and must take into account model uncertainties. On the contrary, model-free methods are signal-based and thus do not necessitate for a mathematical model, but historical data under healthy and faulty modes of behaviour are necessary in order to use them.

Considering a model-based approach, one possible solution for IWSNs to improve network level fault diagnosis relies on the idea that sensors belonging to the same areas will generate similar values. For this reason, a network partitioning strategy able to group nodes that exhibit similar behaviors can serve as a useful tool.

3.3 Clustering in an IWSN

Clustering problem in a wireless sensor network can be tackled by considering the topology (*network decomposition*) or the data gathered from the environment (*data clustering*). The first approach is generally considered in the design of communication algorithms and protocols, where the formation of clusters can lead to higher energy efficiency and reduced communication delays. On the other hand, data based network partitioning allows to reduce the computational load by taking into account similarities among nodes measurements in applications that deal with large amounts of information.

We consider the clustering strategy proposed in [22], where differently from other approaches proposed in the literature, network decomposition and data clustering are considered together, ensuring thus the performance standards of industrial communications. Before presenting the algorithm, we provide some basic definitions and operations from the theory of graphs.

3.3.1 Basics of Graph Theory

Graphs provide natural abstractions for how information is shared among agents in a network [29]. Such abstractions do not contain any information regarding what is shared by nodes or what protocol is used for the communication; instead, graphs represent a useful tool for an high-level description of the network topology in terms of basic objects.

Given a multi-agents system, it can be represented with a network made up of N nodes organized into an undirected graph $\mathcal{G}$. A *graph* $\mathcal{G}$ is an ordered pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Another way to refer to edges and vertices of $\mathcal{G}$ is by using the notation $\mathcal{V}(\mathcal{G})$ and $\mathcal{E}(\mathcal{G})$ or by using the indices only (Figure 3.3). The vertices of the graph correspond to the nodes of the

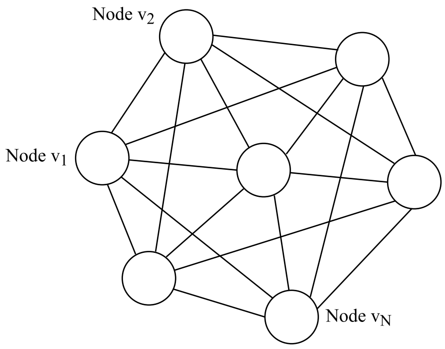


Figure 3.3: Graph based network example.

network, while the pair (i, j) belongs to the set $\mathcal{E}$ if and only if the node i can communicate with the node j and viceversa. In this case, the two vertices are called *adjacent* and the edge (i, j) is called *incident* with vertices i and j. In this case, the graph is referred as *undirected* or *non-oriented*, namely the pairs of vertices representing the edges are not ordered (otherwise, the graph is called *directed* or *oriented*). Finally, a graph in said to be *complete* if each pair of vertices has an edge that connects them. The *neighbourhood* $\mathcal{N}_i \subseteq \mathcal{V}$ of the vertex v_i is the set $\mathcal{N}_i$ of vertices which are connected with i in the graph:

$$\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}. \tag{3.1}$$

In particular, $\mathcal{N}_i$ is a subset of $\mathcal{V}$ and its cardinality corresponds to the *degree* of the node i, *deg(i)* or $d(v_i)$, which represents the number of nodes adjacent to node i. If $v_j \in \mathcal{N}_i$, it follows that $v_j \in \mathcal{N}_j$, since the edge set in a undirected graph consist of unordered vertex pairs.
Other meaningful elements of graph theory are the following ones:

- A *path* of length l from node i to node j in $\mathcal{G}$ is a sequence of distinct vertices starting with i and ending in j such that the consecutive *inner* vertices are connected by an edge in $\mathcal{E}$. More specifically, $l = \{v_1, v_2, \ldots, v_N\}$, $v_m \in \mathcal{V}$, $v_1 = i$, $v_N = j$, $(v_m, v_{m+1}) \in \mathcal{E}$, $m = \{1, \ldots, N-1\}$ (Figure 3.4).
  A *path* is referred as *simple* if all the vertices of the path are distinct.
  A graph is said *connected* if, for every pair of vertices in $\mathcal{V}$, there is a path that has them as its end vertices. If this property is not satisfied, the graph is called *disconnected*.

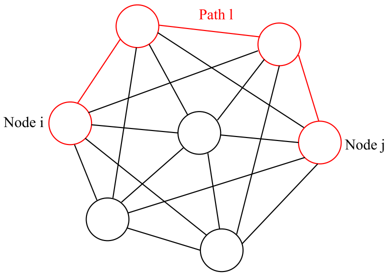


Figure 3.4: Path example.

- A *cycle* is a path from node i to itself without repeated nodes, i.e. the vertices of the path are distinct except for its end vertices (Figure 3.5).

A cycle is referred as *simple* if all the nodes composing it are distinct, except for the the initial and final one.
A graph without cycles is called *acyclic*.

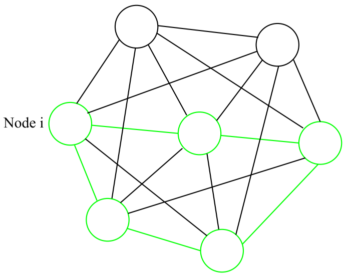


Figure 3.5: Cycle example.

- A *subgraph* of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a graph whose vertex set is a subset of that of $\mathcal{G}$, and whose adjacency relation is a subset of that of $\mathcal{G}$ restricted to the subset (Figure 3.6). It is a graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ such that $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$.

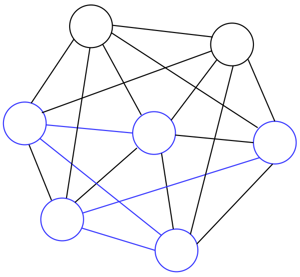


Figure 3.6: Subgraph example.

Graphs and Matrices. Graphs are constructed to represent relations between a finite number of objects, having the advantage of admitting a straightforward graphical representation in terms of vertices and edges. Another way to represent a graph is by using matrices.
Given an undirected graph $\mathcal{G}$, and the degree of each vertex $d(v_i)$, it is possible to associate a matrix to the graph, called *degree matrix* of $\mathcal{G}$, which is a diagonal matrix containing the vertex-degree of $\mathcal{G}$ on the diagonal:

$$\Delta(\mathcal{G}) = \begin{pmatrix} d(v_1) & 0 & \dots & 0 \\ 0 & d(v_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d(v_n) \end{pmatrix}. \tag{3.2}$$

The *adjacency matrix* $A(\mathcal{G})$ is the $n \times n$ matrix describing the adjacency relationships in the graph $\mathcal{G}$, such that

$$[A(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

One particular property of the adjacency matrix is that it is symmetric in those cases where the corresponding graph is indirect. On the contrary, the matrix does not have this property when the graph is directed. In the case when the graph has *self-loops*, i.e. there are edges connecting vertices to themselves, the elements on the main diagonal of A are equal to one (or to two in some cases).

Another important matrix representation of a graph $\mathcal{G}$ is given by the *graph Laplacian*, $\mathcal{L}(\mathcal{G})$. One of its most straightforward definitions is

$$\mathcal{L}(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G}), \tag{3.4}$$

where $\Delta(\mathcal{G})$ is the degree matrix of $\mathcal{G}$ and $A(\mathcal{G})$ is its adjacency matrix. Elements of matrix $\mathcal{L}(\mathcal{G})$ are such that

$$l_{ij} = \begin{cases} deg(\mathcal{N}_i) & i = j \\ -1 & i \neq j \end{cases} \tag{3.5}$$

This implies that $l_{ij} \neq 0 \leftrightarrow (v_i, v_j) \in \mathcal{E}$.

One of the properties of the graph Laplacian is that, for all graphs, the rows of the matrix sum to zero. As a consequence $\lambda = 0$ is an eigenvalue for $\mathcal{L}(\mathcal{G})$ and it is associated to the unitary eigenvalue $u = [1, 1, \ldots, 1]^T \in \mathbf{R}^n$, which can be also written as $\mathbf{1}$.

$\mathcal{L}(\mathcal{G})$ is symmetric and positive semidefinite, hence its real eigenvalues can be ordered as by *Geršgorin theorem* they are such that:

$$0 = \lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq \ldots \leq \lambda_n(\mathcal{G}) \leq 2d_{max}, \tag{3.6}$$

with $\lambda_1(\mathcal{G}) = 0$. As a consequence the following theorem derives:

Theorem 3.1 A graph $\mathcal{G}$ is connected if and only if $\lambda_2(\mathcal{G}) > 0$.

3.3.2 Clustering Strategy

Consider an IWSN composed by N nodes, whose topology is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of communication links among them. Each node v_i is associated with a measurement m_i, gathered from the environment. Given the graph and the measurements, the clustering task consists in identifying the node clusters $\{\mathcal{C}_l\}$ that constitute the unique partition of $\mathcal{V}$, defined as $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ satisfying the following criteria:

(i) *Connectivity*: $\forall \mathcal{C}_l \in \mathcal{C}$, $\forall v_i, v_j \in \mathcal{C}_l$, $\exists$ path $p = \{v_1, \ldots, v_h, \ldots, v_n\}$ such that $v_1 = v_i$, $v_n = v_j$, $(v_h, v_{h+1}) \in \mathcal{E}$ and $v_h \in \mathcal{C}_l$, $\forall h \in [1, n-1]$;

(ii) *Measurement similarity*: $\forall \mathcal{C}_l \in \mathcal{C}$, $\forall v_i \in \mathcal{C}_l, \exists v_j \in \mathcal{C}_l$ such that $||m_i - m_j|| \leq b$, according to some norm: $b \in \mathbb{R}$ is called *clustering bound*;

(iii) *Maximality*: let $\mathcal{C}^{(i)} = \{\mathcal{C}_1^{(i)}, ..., \mathcal{C}_{k_i}^{(i)}\}$, $i \in \mathbb{N}$ be a generic partition of the network satisfying the first two criteria, the maximal partition is $\mathcal{C}^* = \{\mathcal{C}_1 .... \mathcal{C}_{k^*}\}$, where

$$k^* = \underset{i \in \mathbb{N}}{\operatorname{argmin}}[k_i] \tag{3.7}$$

The elements of the obtained partition $\mathcal{C}_1, \ldots, \mathcal{C}_{k^*}$ are denoted as *optimal clusters* and the function $\mathcal{F} : \mathcal{V} \rightarrow \mathcal{C}^*$ is introduced that maps each node to the optimal cluster it belongs to in the optimal partition. Optimal clusters are non-overlapping and cover the entire network, i.e. any node in $\mathcal{V}$ belongs to one and only one cluster following the three criteria.

Looking at the three criteria, criterion 1 requires that each cluster forms a connected subgraph, which implies that the measurements and in general data obtained from nodes within the same cluster are shared. Criterion 2 instead states that some similarity exists among the measurements of nodes in a cluster: the *clustering bound* b is a setup parameter related to the expected variance in the measurements range. Finally, the third criterion is introduced to select the partition composed by the minimum number of clusters, in order to ensure that the partition is unique.

One of the novelties of this method is the introduction of the *clustering bound* b used for the accomplishment of the measurements similarity. Nevertheless, in **??** no analytical expression is given for it and its value has to be chosen according to some prior knowledge on the system.

Considering the distributed clustering algorithm, each node v_i of the network executes the same instructions until the unique partition $\mathcal{C}^*$ that fulfills the three criteria is determined. To this aim, a label c_i is associated to each mode v_i to specify the cluster to which it belongs to. This variable is updated during the algorithm execution, so that the output of the whole procedure is a set of labels, one for each node, describing the partition of the network, such that $c_i = c_j \Leftrightarrow \mathcal{F}(v_i) = \mathcal{F}(v_j)$. Each node v_i is associated to a lower and upper bound B_i^l and B_i^u defined as

$$\begin{aligned} B_i^l &= m_i - b, \quad \forall i \\ B_i^u &= m_i + b, \quad \forall i \end{aligned} \tag{3.8}$$

The algorithm is based on an iterative exploration of the neighbours, performed by each node in a distributed manner. When two neighbour nodes have similar measurements according to criterion 2, then both labels c_i and c_j are set to the $\min(c_i, c_j)$. Bounds are update at each iteration and nodes are included into the correspondent clusters.

The presented procedure is shown in Algorithm 2 and it will represent the starting point for the development of the fault detection and isolation method that we propose.

Algorithm 2 `Distributed Clustering Algorithm (DCA)`

```
procedure DCA($m_i, b, \mathcal{N}(i)$)
    $active(v_i) \leftarrow false$
    $B^l_{min} \leftarrow \min_{v_h : c_h = c_i} B^l_h$
    $B^u_{max} \leftarrow \max_{v_h : c_h = c_i} B^u_h$
    for each $v_j \in \mathcal{N}_i, c_j \neq c_i$ do
        if $B^l_i \leq m_j \leq B^u_i$ then
            $c_j \leftarrow min(c_i, c_j)$
            $c_i \leftarrow c_j$
            if $B^l_{min} > B^l_j$ then
                $B^l_{min} \leftarrow B^l_j$
            end if
            if $B^u_{max} < B^u_j$ then
                $B^u_{max} \leftarrow B^u_j$
            end if
            $active(v_i) \leftarrow true$
            $active(v_j) \leftarrow true$
        end if
    end for
    if $B^l_{min} < B^l_i$ or $B^u_{max} > B^u_i$ then
        $B^l_i \leftarrow B^l_{min}$
        $B^u_i \leftarrow B^u_{max}$
        $active(v_i) \leftarrow true$
    end if
    for each $j \in \mathcal{N}_i, c_i = c_j$ do
        if $B^l_{min} < B^l_j$ then
            $B^l_j \leftarrow B^l_{min}$
            $active(v_j) \leftarrow true$
        end if
        if $B^u_{max} > B^u_j$ then
            $B^u_j \leftarrow B^u_{max}$
            $active(v_j) \leftarrow true$
        end if
    end for
end procedure
```

4

Clustering-based Sensor Fault Detection and Isolation

In this chapter we introduce the problem we want to tackle in the case of an HVAC system made of a single room monitored by an IWSN of N sensors. Initially, we show how we deal with the deployment problem, then we formally define the problem we aim to solve, introducing also some theory related to different kind of faults, we propose our clustering algorithm starting from the clustering method of [22], followed by the new sensor fault diagnosis architecture developed in this thesis, and we conclude the chapter with some detectability and isolability analysis in order to prove the validity of the proposed solution.

4.1 Problem Formulation

We consider an IWSN with N sensors. The communication among sensors is allowed according to an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We consider the HVAC system introduced in Section 2.3, made of the electromechanical part and a single room, but such analysis can be trivially extended to multi-zones HVAC systems. Usually, deployment of the sensors can be performed in the following ways:

- randomly;
- according to Voronoi's optimal partition;
- according to obstacles and presence of critical items in the space.

In the first case, we exploit the results of *circle packing* together with the theorems deriving from the theory of graphs such to obtain a network which would be connected.

Assumption 4.1 We assume that every sensor has a finite sensing range, that we consider to be circular, centred in the sensor, of radius r. The communication radius is defined as $d := 2r$.

If the distance between two sensors is less or equal than d, we say that they communicate and an edge can be drawn between them. Communication happens along the virtual link that is d (Figure 4.1). In order to define the value of r, given the number of nodes in the

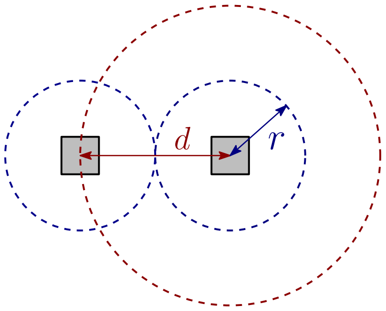


Figure 4.1: Radius of sensing and communication.

network, we exploit the results of circle packing, which is the study of the arrangement of circles on a surface. The theoretical sensing and communication radius are thus given by

$$\begin{aligned} \pi r^2 = \frac{A_{square}}{N} \Rightarrow r_{th} &= \sqrt{\frac{A}{\pi N}} \\ d_{th} &= 2\sqrt{\frac{A}{\pi N}} \end{aligned} \tag{4.1}$$

where A represents the area, N is the number of the agents, r is the sensing radius and d is the communication radius.

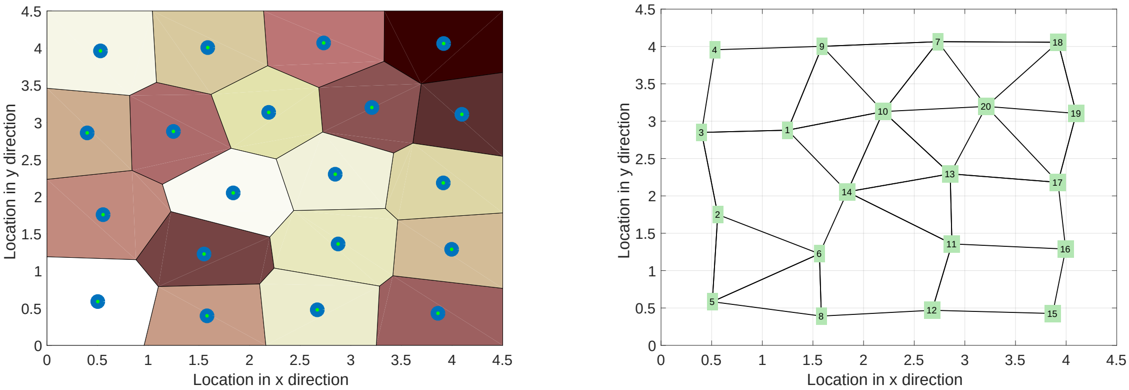
Note that r_{th} and d_{th} represent respectively a lower and upper bound for the radius. In practical, an intermediate value is chosen, which ensures that the network is connected without falling in special cases i.e. $r_{th} < r < d_{th}$. Applying the properties on the matrices associated to a graph (Section 3.3.1), we develop the procedure shown in Algorithm 3, which allows to obtain a connected network. Initially, the N sensors are deployed randomly with an uniform distribution (row 4); then for each couple of sensors i and j, the

Algorithm 3 `WSN Generation`

```
 1: procedure WSN(N, A, $r$)
 2:     $flag = 0$;
 3:     while $flag = 0$ do
 4:         $net.loc = rand\_distr(N, 2)$;
 5:         for i = 1, . . . , N do
 6:             for j = 1, . . . , N do
 7:                 if $euclid\_dist(net.loc(i), net.loc(j)) <= r$ then
 8:                     $nei\_tmp = [nei\_tmp; j]$;
 9:                     $net\_l(i, j) = -1$;
10:                 end if
11:             end for
12:             $net.nei(i) = nei\_tmp$;
13:             $net.l(i, i) = size(nei\_tmp, 1) - 1$;
14:         end for
15:         if $length(find(eig(net.l) < 0.00001)) < 2$ then
16:             $flag = 1$;
17:         end if
18:     end while
19: end procedure
```

euclidian distance is computed and if it satisfies the given radius, then $j \in \mathcal{N}\{i\}$, while element l_{ij} of the Laplacian matrix is set to -1 according to 3.5 (rows 38-46).
After all the neighbours of each node are computed, it is possible to define the diagonal elements of the Laplacian matrix according to 3.5 (rows 48-49). Finally, it is necessary to verify that the network obtained is connected. To this aim, we apply the results of *Geršgorin theorem* and Theorem 3.1: if the second least eigenvalue is larger than 0, the graph is connected (rows 53-55). The procedure terminates when a connected graph is found.

When we deal with a wireless sensor networks having the task of monitoring a certain area, the distribution of the sensors could be achieved considering also the additional aim of having an optimal partition of the space. This can be obtained applying Voronoi tessellations. In particular, when the sensors are located in positions that are both generators and centroids (see Appendix B), the particular Voronoi diagramm is called Centroidal Voronoi Tessellation (CVT) and it combines both optimal partitioning of the region with optimal placement of the generators (Figure 4.2).



(a) Voronoi tessellations

(b) Wireless Sensor Network correspondent to Voronoi's tessellations

Figure 4.2: Example of deployment according to Voronoi's optimal partition and respective network.

From the implementation point of view, this partitioning of the space is obtained computing the Voronoi diagram of a closed area and an approximating the integral on the polygons, generated through the tessellations to calculate the centroids. Moreover, we use the distance between the sensors and the centroids of their correspondent Voronoi's regions as condition of termination for the deployment. The distance between the agents and their respective centroid is stacked into a matrix, the $2l$ norm is computed and compared to a tolerance constant. Using Lloyd's algorithm (which procedure is shown in Algorithm 4), the space can be organized in a CVT and agents can converge to their respective centroids. We denote with p_i the position of the sensors and with C_{V_i} the centroids of the correspondent regions. The wireless sensor network corresponding to the optimal partitioning of Figure 4.2(a) is shown in Figure 4.2(b).

We do not analyze the the third scenario since it can depend on the specific environment considered and may be tackled through one of the aforementioned cases.
Once sensors are deployed such to form a wireless sensor network, we have that the diffusion of the heating from the source represented by the fan of the HVAC system at each

Algorithm 4 `Lloyd's algorithm`

```
N = # of agents
toll = $10^{-2}$
while $termination = false$ do
    compute the Voronoi tessellations,
    compute $C_{V_i}$,
    compute $||p_i - C_{Vi}||$

    if $||p_i - C_{Vi}|| > toll$ then
        continue updating Voronoi
    else
        $termination = true$
    end if
end while
```

point is modelled as

$$T_i(k+1) = h(T_i(k), u(k), P_i) + \eta_i(T_i(k), u(k), P_i, u_e(k)), \tag{4.2}$$

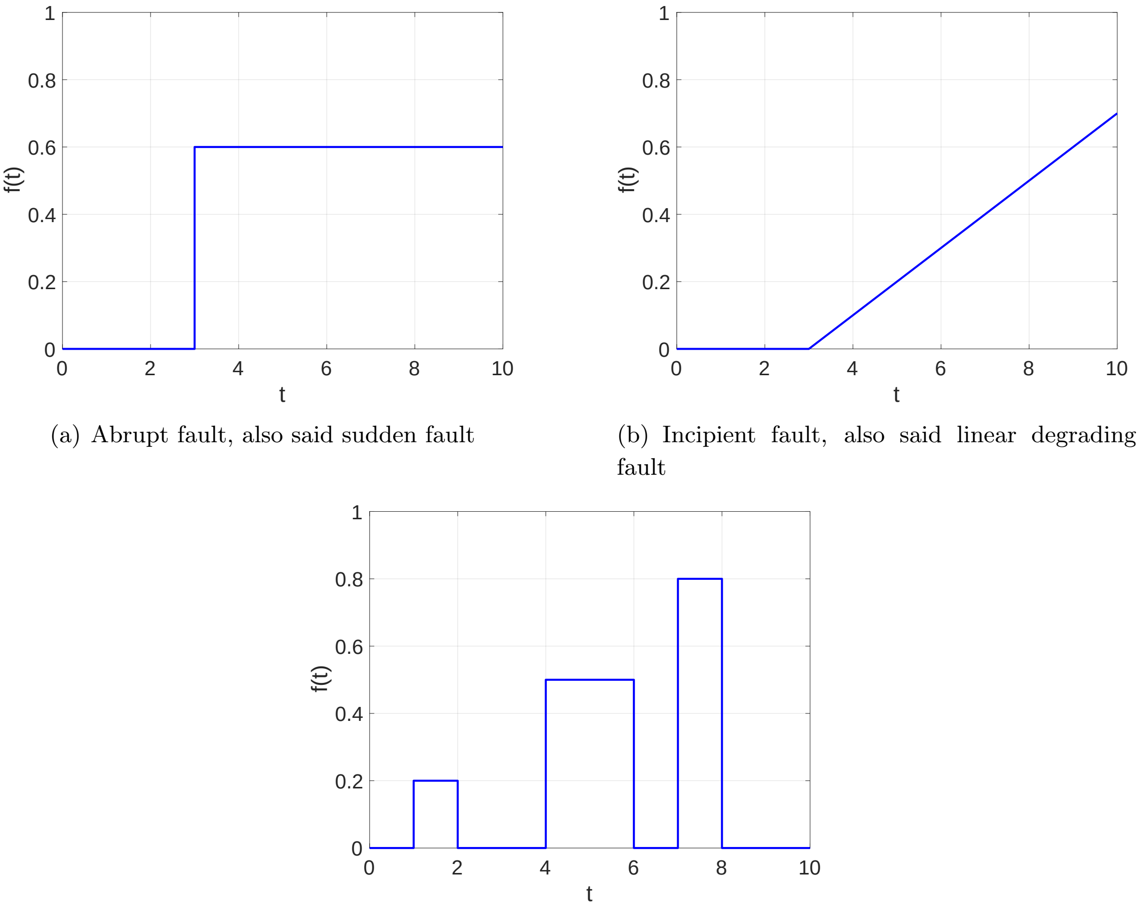
where $T_i(k)$ is the temperature in a point i at time k; h is a field modelling the nominal heat diffusion depending on the past temperature, the local inputs u (including the HVAC input and possibly the external temperature if known) and the position P_i; η_i considers modeling uncertainty and external disturbances u_e, including the unknown/unmodelled influence of neighboring rooms' temperature or unknown/unmodelled inputs or phenomena, such as the presence of windows, lights or electrical appliances.
Each node i provides a noisy measurement y_i of the temperature T_i:

$$y_i(k) = T_i(k) + d_i(k) + f_i(k), \tag{4.3}$$

where $d_i(k)$ is the measurement noise at time k and $f_i(k)$ is the effect of possible faults affecting sensor i at time k ($f_i(k) = 0$ in healthy conditions).

Assumption 4.2 The measurement noise at each node is assumed bounded, i.e. $|d_i(k)| \leq \bar{d}_i$, $i = 1, \dots, N$, where $\bar{d}_i$ is a known constant positive bound.

We assume that all the sensors are characterised by the same noise bound $\bar{d}$, but this case can be extended also for heterogeneous sensors.
For what concerns the faults, they usually show a characteristic behaviour for the various components of a system and thus, they can be distinguished by their form, time behaviour and extent. The *form* can be either systematic or random. The *time behaviour* may be described by permanent, transient, intermittent, noise or drift, while the extent of the faults is either local or global and includes the size. Faults can be seen as changes of signals or parameters and they can be distinguished in *additive* faults, when the output $y(t)$ is changed by the addition of a function $f(t)$and *multiplicative faults*, when another variable η multiples $f(t)$. In this thesis we consider only the first category of faults and we take into account three kind of faults, which behaviour is shown in Figure 4.3:



(a) Abrupt fault, also said sudden fault

(b) Incipient fault, also said linear degrading fault

(c) Intermittent fault, also said temporary fault

Figure 4.3: Different kind of faults.

- *constant sudden faults*: $f(k) = c \cdot u(k - k_f)$ which are modelled as steps or drifts;
- *linear degrading fault*: $f(k) = c \cdot k \cdot u(k - k_f)$, which behaviour is smoother than in the previous case and they are modelled as ramps;
- *temporary fault*: $f(k) = c \cdot [u(k - k_{f_1}) - u(k - k_{f_2})]$, $k_{f_2} > k_{f_1}$, which extension is limited in time;

where c is a positive random constant representing the amplitude of the fault and $u(\cdot)$ is the unit step.

With the aim of performing fault detection on the sensor network, we build on the clustering method proposed in [22] and recalled in Section 3.3.2 in order to group sensors in the room into clusters having similar uncertainty conditions. Clustering results and a novel model-based sensor fault diagnosis strategy are used to detect the presence of faults and isolate faulty sensors in the clusters.

4.2 Distributed Clustering

Given the distributed clustering technique introduced in [22] and recalled in Section 3.3.2, we replace the measurements similarity criterion with a consistency criterion, allowing sensors to measure different quantities, and designing the bound b based on the model of the system. This allows to obtain clusters which have similar modeling uncertainty or disturbances. In particular, sensors can compute the bound b in a distributed way using only local communication with neighbours.

4.2.1 The Consistency Criterion

Each sensor i communicates its own measurement y_i to the neighbouring sensors and compares it with the measurements y_j taken by neighbouring nodes $j \in \mathcal{N}_i$. For each couple (i, j) in healthy conditions:

$$|y_i - y_j| = |T_i - T_j + d_i - d_j| .$$

Since the actual temperature at each point is unknown, and temperatures at different spatial points may be quite different simply due to the physics of the problem and not due to the presence of anomalies, each node can compute a theoretical estimate of the temperature based on the nominal model, past measurements, the known inputs and the position:

$$\hat{T}_i(k) = h(y_i(k-1), u(k-1), P_i). \tag{4.4}$$

Similarly, each node can compute the estimate also for the neighbouring sensors, assuming that the positions P_j, $j \in \mathcal{N}_i$ are known to node i. Otherwise, the estimates $\hat{T}_j$ can be communicated together with the measurements y_j reducing the computation cost at each node, but increasing the communication cost. We have that

$$\begin{aligned} |y_i - y_j|| &= |T_i - T_j + d_i - d_j| && (4.5)\\ &\leq \left|\hat{T}_i - \hat{T}_j\right| + |-\Delta h_i + \Delta h_j| + |\eta_j - \eta_i| + 2\bar{d}, && (4.6) \end{aligned}$$

where $\Delta h_i(k) = h(y_i(k), u(k), P_i) - h(x_i(k), u(k), P_i)$. Given Assumption 4.2, it is possible to compute a bound $\bar{\Delta} h_i(k)$ for $\Delta h_i(k)$:

$$\bar{\Delta} h_i := \max_{|d_i| \leq \bar{d}} |h(y_i, u, P_i) - h(y_i - d_i, u, P_i)| .$$

Furthermore, since the goal of the clustering is to partition the sensors into groups with similar uncertainty, we neglect the term $\eta_j - \eta_i$ for the definition of the clustering bound b_i:

$$b_i = \min_j \left[\left|\hat{T}_i - \hat{T}_j\right| + \bar{\Delta} h_i + \bar{\Delta} h_j + 2\bar{d} \right]$$

We then apply the distributed clustering algorithm of Section 3.3.2 to partition the sensor network in clusters.

4.3 Sensor Fault Diagnosis

Once the IWSN is partitioned into clusters, the distributed fault detection and isolation method is implemented at each sensor i. We assume that the initial clustering is performed in healthy conditions.

At each time step, each node communicates its measurements (and estimates) only to neighbouring nodes belonging to the same cluster: $\mathcal{N}_i^* = \mathcal{N}_i \cap \mathcal{C}(i)$. Sensor i computes two different residual signals r_1^{ij} and r_2^i for sensor fault detection:

$$\begin{aligned} r_1^{ij}(k) &= y_i(k) - y_j(k), \\ r_2^i(k) &= y_i(k) - \hat{y}_i(k) \end{aligned} \tag{4.7}$$

where

$$\hat{y}_i(k) = h(y_i(k-1), u(k-1), P_i) + \lambda(y_i(k-1) - \hat{y}_i(k-1)), \tag{4.8}$$

is the model-based observer estimate, with $0 < \lambda < 1$ to guarantee the convergence. Then, it firstly checks the coherence with neighbouring sensors, similarly as what done for the clustering. Indeed, in healthy conditions, $\forall j \in \mathcal{N}_i^*$

$$\left| r_1^{ij}(k) \right| \leq \left| \hat{T}_i(k) - \hat{T}_j(k) \right| + \bar{\Delta} h_i + \bar{\Delta} h_j + 2\bar{d} := \bar{r}_1^{ij}(k). \tag{4.9}$$

Secondly, it checks the coherence with its own past measurements and the model. In healthy conditions, it holds that

$$\begin{aligned} \left| r_2^i(k) \right| &= \left| \lambda r_2^i(k-1) + \Delta h_i(k-1) + \eta_i(k-1) + d_i(k) \right| \\ &\leq \left| \lambda r_2^i(k-1) \right| + \bar{\Delta} h_i(k) + \bar{\eta}_i(k-1) + \bar{d} := \bar{r}_2^i(k), \end{aligned} \tag{4.10}$$

where $\bar{\eta}_i$ satisfies the following

Assumption 4.3 The modeling uncertainty at each node is assumed to be bounded, i.e. $|\eta(k)| \leq \hat{\eta}_i(k)$, $\forall k$, $i = 1, \ldots, N$, where $\bar{\eta}_i(k)$ is a known positive value.

In healthy conditions the residual signals (r_1^{ij}, r_2^i) are lower than the corresponding thresholds $(\bar{r}_1^{ij}, \bar{r}_2^i)$. When at least one of the two residual signals crosses the corresponding threshold in at least one of the sensors of the cluster, then it is possible to say that a fault has occurred in one or more sensors of the cluster. It is important to note that the proposed thresholds guarantee the absence of false-alarms thanks to the way they are defined in 4.9-4.10. It is also worth noting that residual r_1^{ij} is sensitive both to faults in sensor i and sensor j. On the other hand, residual r_2^i is sensitive only on faults on sensor i. This double redundancy allows the isolation of the faulty sensor(s) in each cluster. After fault isolation, the faulty sensors are removed. The clustering algorithm is performed to reconfigure the clusters only with the healthy sensors.

4.4 Detectability and Isolability Analysis

In this section, we analyse some sufficient conditions on the faults with respect to the noises and uncertainties to allow the detection by the proposed distributed method.

4.4.1 Detectability of a Single Fault

Let us consider that a general fault f_i is occurring on sensor i, that is, for $k \geq k_f$, the i-th output equation is

$$y_i(k) = T_i(k) + d_i(k) + f_i(k), \tag{4.11}$$

where $f_i(k)$ could even be zero at some time after k_f in the case of intermittent sensor faults. We are not assuming a specific type of sensor fault (persistent, intermittent, bias, drift,..).

Proposition 4.4 Let us consider that sensor i is affected by a fault f_i for $k \geq k_f$. It is sufficient that the fault effect satisfies the following condition to guarantee fault detection by means of residual r_2^i:

$$\left| \sum_{h=k_f}^{k-1} \lambda^{k-h-1} f_i(h+1) \right| > 2\bar{r}_2^i(k).$$

Proof In this scenario the residual r_2^i becomes:

$$r_2^i(k) = \lambda r_2^i(k-1) + \Delta h_i(k-1) + \eta_i(k-1) + d_i(k) + f_i(k), \tag{4.12}$$

that can be rewritten as

$$\begin{aligned} r_2^i(k) &= \lambda^k r_2^i(0) + \sum_{h=0}^{k-1} \lambda^{k-h-1}(\Delta h_i(h) + \eta_i(h) + d_i(h+1) + f_i(h+1)) \\ &= \lambda^k r_2^i(0) + \sum_{h=0}^{k-1} \lambda^{k-h-1}(\Delta h_i(h) + \eta_i(h) + d_i(h+1)) + \sum_{h=k_f}^{k-1} \lambda^{k-h-1} f_i(h+1). \end{aligned} \tag{4.13}$$

From (4.13), we can write that

$$\left| r_2^i(k) \right| \geq \left| \sum_{h=k_f}^{k-1} \lambda^{k-h-1} f_i(h+1) \right| - \left| \lambda^k r_2^i(0) + \sum_{h=0}^{k-1} \lambda^{k-h-1}(\Delta h_i(h) + \eta_i(h) + d_i(h+1)) \right| \tag{4.14}$$

In order to have detection, the following condition has to be satisfied:

$$\left| r_2^i(k) \right| \geq \bar{r}_2^i(k).$$

In the worst case, in order to satisfy the detection condition we require

$$\left| \sum_{h=k_f}^{k-1} \lambda^{k-h-1} f_i(h+1) \right| > \bar{r}_2^i(k) + \left| \lambda^k r_2^i(0) + \sum_{h=0}^{k-1} \lambda^{k-h-1}(\Delta h_i(h) + \eta_i(h) + d_i(h+1)) \right|, \tag{4.15}$$

thus the statement of the proposition, remembering that, thanks to the way the threshold is defined for r_2^i in (4.10), we have in healthy conditions that

$$\left| \lambda^k r_2^i(0) + \sum_{h=0}^{k-1} \lambda^{k-h-1}(\Delta h_i(h) + \eta_i(h) + d_i(h+1)) \right| \leq \bar{r}_2^i(k).$$

$\diamondsuit$

The condition in the previous proposition gives a characterization of the cumulative fault effect sufficient to be detected by the proposed architecture by means of residual r_2^i despite the presence of uncertainties and disturbances that may hide its effect.

We now provide a sufficient condition regarding the instantaneous effect of the fault.

Proposition 4.5 Let us consider that sensor i is affected by a fault f_i at time $\bar{k}$. It is sufficient that the fault satisfies the following condition to guarantee fault detection at time $\bar{k}$:

$$\left|f_i(\bar{k})\right| > 2\bar{r}_2^i(\bar{k}).$$

Proof In order to have fault detection at time k_f by the residual r_2^i, it is necessary that

$$\left|r_2^i(\bar{k})\right| > \bar{r}_2^i(\bar{k}).$$

since in faulty condition the residual is modeled as in (4.12), in order to have detection in the worst case it is sufficient that

$$\left|f_i(\bar{k})\right| > \bar{r}_2^i(\bar{k}) + \left|\lambda r_2^i(\bar{k}-1) + \Delta h_i(\bar{k}-1) + \eta_i(\bar{k}-1) + d_i(\bar{k})\right|, \tag{4.16}$$

implying the statement of the proposition. ♢

Furthermore, we provide the detectability condition sufficient for a fault f_i to be detected at time $k_d \geq k_f$ by means of the residual r_1^{ij}.

Proposition 4.6 Let us consider the case that sensor i is affected by a fault f_i. It is sufficient that at time k_d the fault satisfies the following condition to guarantee fault detection:

$$|f_i(k_d)| > 2\bar{r}_1^{ij}(k_d).$$

Proof In order to have fault detection at time k_d by means of the residual r_1^{ij}, it is necessary that

$$\left|r_1^{ij}(k_d)\right| > \bar{r}_1^{ij}(k_d).$$

In the considered faulty case, residual r_1^{ij} can be expressed as

$$r_1^{ij}(k) = T_i(k) + d_i(k) + f_i(k) - T_j(k) + d_j(k).$$

In order to have detection in the worst case the following condition has to be satisfied

$$|f_i(k_d)| > \bar{r}_1^{ij}(k_d) + |T_i(k) + d_i(k) - T_j(k) + d_j(k)|,$$

and since by the definition of the threshold in (4.9) we know that

$$|T_i(k) + d_i(k) - T_j(k) + d_j(k)| \leq \bar{r}_1^{ij},$$

the statement of the proposition is proved. ♢

In a similar way, it is possible to prove the following.

Proposition 4.7 If a fault f_j is occurring in sensor j, $j \in \mathcal{N}_i^*$,

- it will not be detected by residual r_2^i in sensor i;
- the following condition is sufficient for sensor i to detect the fault by means of residual r_1^{ij} at time k_d:

$$|f_j(k_d)| > 2\bar{r}_1^{ij}(k_d).$$

4.4.2 Detectability of Multiple Faults

We have the following theoretical result.

Proposition 4.8 Let us consider the case that sensor i is affected by a fault f_i and simultaneously a fault f_j is occurring in sensor j, $j \in \mathcal{N}_i^*$. The following condition is sufficient to guarantee the fault detection at time k_d by the proposed distributed sensor fault detection scheme:

$$|f_i(k_d) - f_j(k_d)| > 2\bar{r}_1^{ij}(k_d).$$

Proof In order to have fault detection at time k_f by means of the residual r_1^{ij}, it is necessary that

$$\left| r_1^{ij}(k_d) \right| > \bar{r}_1^{ij}(k_d).$$

In the considered faulty case, residual r_1^{ij} can be expressed as

$$r_1^{ij}(k) = T_i(k) + d_i(k) + f_i(k) - T_j(k) + d_j(k) - f_j(k).$$

In order to have detection in the worst case the following condition has to be satisfied

$$|f_i(k_d) - f_j(k_d)| > \bar{r}_1^{ij}(k_d) + |T_i(k) + d_i(k) - T_j(k) + d_j(k)|,$$

and since by the definition of the threshold in (4.9) we know that

$$|T_i(k) + d_i(k) - T_j(k) + d_j(k)| \leq \bar{r}_1^{ij},$$

the statement of the proposition is proved. $\diamondsuit$

It is worth noting that, depending on the sign of the faults, the presence of multiple faults may either improve or compromise the fault detection.

By analyzing the results in the previous propositions, it is important to note that the use of two different residual signals may possibly increase the detectability performance of the proposed distributed architecture and, as we will see in the following section, allows to isolate faults distinguishing between local and neighbouring faults.

4.4.3 Isolability Analysis

After fault detection, node i can communicate the alarm to neighbouring sensors j, $j \in \mathcal{N}_i^*$, according to the following rules:

$$d_1^{ij}(k) = \begin{cases} 0 \text{ if } \left|r_1^{ij}(k)\right| \leq \bar{r}_1^{ij}(k) \\ 1 \text{ if } \left|r_1^{ij}(k)\right| > \bar{r}_1^{ij}(k) \end{cases}$$

$$d_2^i = \begin{cases} 0 \text{ if } \left|r_2^i\right| \leq \bar{r}_2^i \\ 1 \text{ if } \left|r_2^i\right| > \bar{r}_2^i \end{cases}$$

and can receive analogous information from the neighbours.

In order to reduce the communication cost, the communication is required only after fault detection. If not received, the quantities d_1^j are assumed to be null. By exploiting the fact that, as shown in the previous sections, residual r_2^i is sensitive only on local faults f_i, while residual r_1^{ij} is sensitive both to local faults affecting sensor i, and faults occurring in the neighbouring sensor j, it is possible to develop a fault isolation logic. In Table 4.1, we provide the fault isolation decisions for each couple (i, j) depending on the values of the signals $d_1^{ij} = d_1^{ji}$, d_2^i, d_2^j.

Table 4.1: Fault isolation logic

d_1^{ij}	d_2^i	d_2^j	Decision
1	0	0	f_i OR f_j
1	1	0	f_i
1	0	1	f_j
1	1	1	f_i AND f_j
0	1	0	f_i
0	0	1	f_j
0	1	1	f_i AND f_j

It is worth noting that the isolation decisions in Table 4.1 guarantee the absence of false isolation. Due to the way the detection thresolds are designed, the fact that the residual crosses the corresponding threshold guarantees the presence of a fault. Viceversa, as long as the residual is lower than the corresponding threshold, the absence of the fault cannot be guaranteed, since it could be "‘hidden"’ by the noise, disturbances or other faults presence. The communication with other neighbouring sensors in $\mathcal{N}_i^*$ can support the choice when it is not possible to distinguish between the presence of f_i or f_j (see the first row of Table 4.1) only considering the signals $d_1^{ij} = d_1^{ji}$, d_2^i, d_2^j.

It is interesting to note the scenario described by the last row of Table 4.1. The simultaneous presence of f_i and f_j may be not detected by d_1^{ij} for two different reasons: in the specific application threshold $\bar{r}_1^{ij}$ could be slightly more conservative than $\bar{r}_2^i$ or the two faults have same sign an similar magnitude (see Proposition 4.8) and therefore there is not detection by means of the residual r_1^{ij}.

4.5 Fault Detection and Isolation Algorithm

After the distributed clustering procedure, the distributed fault detection and isolation method is implemented at each sensor i according to Algorithm 5.

Algorithm 5 `FD algorithm`

```
for $i = 1, \ldots, N$ do
    for $k = 2, 3, \ldots$ do
        Compute $\hat{y}_i(k)$, $\hat{T}_i(k)$, $\hat{T}_j(k)$
        Compute $\bar{r}_2^i(k)$
        Acquire $y_i(k)$
        for $j \in \mathcal{N}_i^*$, $j \neq i$ do
            Acquire $y_j(k)$
            Compute $|r_1^{ij}(k)| = |y_i(k) - y_j(k)|$
            Compute $\bar{r}_1^{ij}(k)$
            if $|r_1^{ij}(k)| > \bar{r}_1^{ij}(k)$ then
                Fault in $i$ or $j$
                Communicate to $j$
            end if
            Compute $|r_2^i(k)| = |y_i(k) - \hat{y}_i(k)|$
            if $|r_2^i(k)| > \bar{r}_2^i(k)$ then
                Fault in $i$
                Communicate to $j$
            end if
        end for
    end for
end for
```

5

Simulation Results

In this chapter, we illustrate the application of the distributed clustering-based sensor fault detection method to an HVAC system, composed of the electromechanical part and a single zone, i.e. a room. In particular, we simulate the HVAC system presented in Section 2.3, using the values collected in Table 5.1.

PARAMETERS	VALUE
$\frac{U_{cc}A_{cc}}{M_{cc}C_v}$	0.02815
$\frac{Q_w\rho_w C_{pw}}{M_{cc}C_v}$	1.2084
$\frac{Q_w\rho_w C_{pw}+U_t A_t}{V_t\rho_w C_{pw}}$	0.0007
T_{wo}	5
$\frac{U_t A_t}{V_t\rho_w C_{pw}}$	5.4566×10^{-4}
$\frac{Q_w\rho_w C_{pw}}{V_t\rho_w C_{pw}}$	1.544×10^{-5}
$\frac{15000}{V_t\rho_w C_{pw}}$	0.006
$\frac{\rho_a C_{pa}}{M_{cc}C_v}$	3.932
$\frac{\rho_a}{M_{cc}C_v}((h_{fg}-C_{pa})(w_f-w_{ao}))$	0.0005
A_f	-0.0006
$\frac{\rho_a C_{pa}}{M_f C_v}$	0.1144
$\frac{U_f A_f}{M_f C_v}$	0.0006
T_{amb}	16

Table 5.1: Values of the variables considered for the HVAC system.

The dimension of the room is chosen to be 4.5 m × 1.75 m × 4.5 m. The gain of the feedback controller is chosen to be $k = 1$, while the backstepping controller is designed considering $k_1 = 4$ and $k_2 = 6$. The desired values of the temperature of the cooling coil and the room are set up as follows: $T_{e,1} = 10°C$ and $T = 22°C$. The diffusion of the heating in the room at time t is modelled according to the heat equation solution for a room of length L as shown in Section 2.6.3:

$$T_i(P_i,k) = T_f(k) + \sum_{n=1}^{\infty} B_n(k)\sin(\frac{(2n-1)}{L}\frac{\pi}{2}P_i)e^{-(\frac{2n-1}{L})^2\frac{\pi^2}{4}k} \tag{5.1}$$

where T_f is the temperature of the air introduced in the room by the fan and

$$B_n(k) = \frac{2}{L}\int_0^L (T_i(P_i,k-1) - T_f(k))\sin(\frac{(2n-1)}{L}\frac{\pi}{2}P_i)dP_i. \tag{5.2}$$

The room temperature is characterized by some uncertainty modelled everywhere as the sum of a noise $\eta^{(1)} \sim \mathcal{N}(0, 00001)$ and an additive term computed at each position P_i as $\eta^{(2)} = -2.3e^{-l}$, being $l = |P_i - P_s|$ the distance between P_i and the source of an unmodelled phenomenon, such as the presence of a window or a door, located in $P_s = (4 \text{ m}, 4.5 \text{ m})$. This phenomenon, not considered in the nominal model for estimation, causes a reduction of the temperature in the top-right corner of the room that decays from $-2.3°C$ in P_s to $0°C$ going towards the centre of the room. The bound on the uncertainty is set to $\bar{\eta}_i = 2.006$ at each point. Other parameters used for the simulations are the filter gain $\lambda = 0.7$ and the measurement noise bound $\bar{d} = 0.4$. In the simulation experiments, we consider both single and multiple sensor faults scenarios for three different kind of faults explained in the previous chapter.

5.1 Random Deployment

In the first scenario we consider 20 sensors randomly deployed in the room such that the resulting graph is connected (see Figure 5.1 for a network example). At the first time step, the proposed distributed clustering algorithm is performed, assuming that no faults are affecting the sensors. The result of the clustering (see Figure 5.1) is the set of clusters $\mathcal{C}^* = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$, with $\mathcal{C}_1 = \{1, 2, 3, 4, 5, 6, 7\}$, $\mathcal{C}_2 = \{8, 9, 10, 11, 12, 13, 14, 15\}$ and $\mathcal{C}_3 = \{16, 17, 18, 19, 20\}$, which satisfies both the conditions on connectivity and on measurement similarity. Moreover, as expected the sensors in the top-right corner, the area with higher uncertainty, are grouped together.

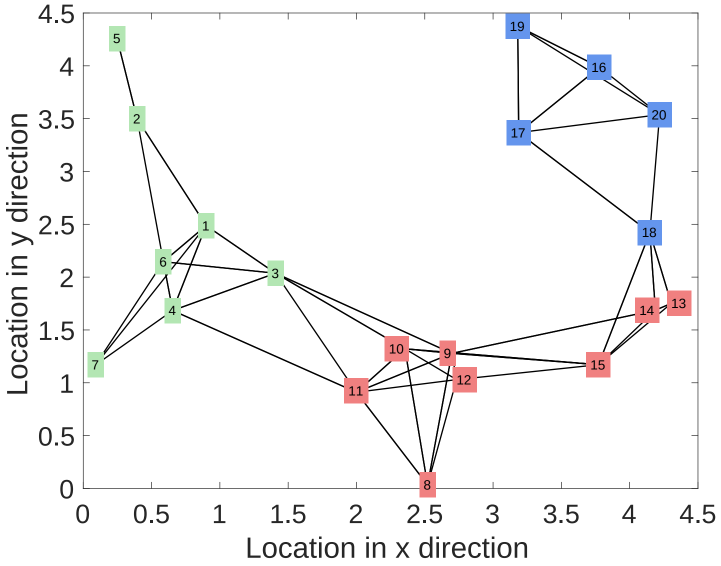


Figure 5.1: Wireless Sensor Network with 20 sensors randomly deployed. The green nodes belong to $\mathcal{C}_1$, the red nodes belong to $\mathcal{C}_2$ and the blue ones belong to $\mathcal{C}_3$.

5.1.1 Single Faults

5.1.1.1 Abrupt Fault

Let us consider a first scenario where a single abrupt fault $f_{18}(k) = 0.8 \cdot u(k - k_{f_{18}})$ occurs in the 18-th sensor at time $k_{f_{18}} = 60$ min. The results of the distributed sensor fault detection and isolation method are shown in Figure 5.2, where the performances of the faulty sensor 18 are compared with those of the healthy sensor 20 belonging to the same cluster $\mathcal{C}_3$. We show the results only for a couple of sensors, but similar and coherent behaviours are obtained for the other sensors. In particular, in 5.2(a) and 5.2(b) it is possible to compare the measurements taken by the two sensors with the actual temperatures T_i, $i = \{18, 20\}$; in 5.2(c) and 5.2(d) the first residual $|r_1^{ij}|$ is plotted together with the threshold $\bar{r}_1^{ij}$, $i = \{18, 20\}$, $j = \{20, 18\}$ for both sensors with respect to each other, while in 5.2(e) and 5.2(f) the residual $|r_2^i|$, $i = \{18, 20\}$ and its corresponding threshold $\bar{r}_2^i$ are shown.

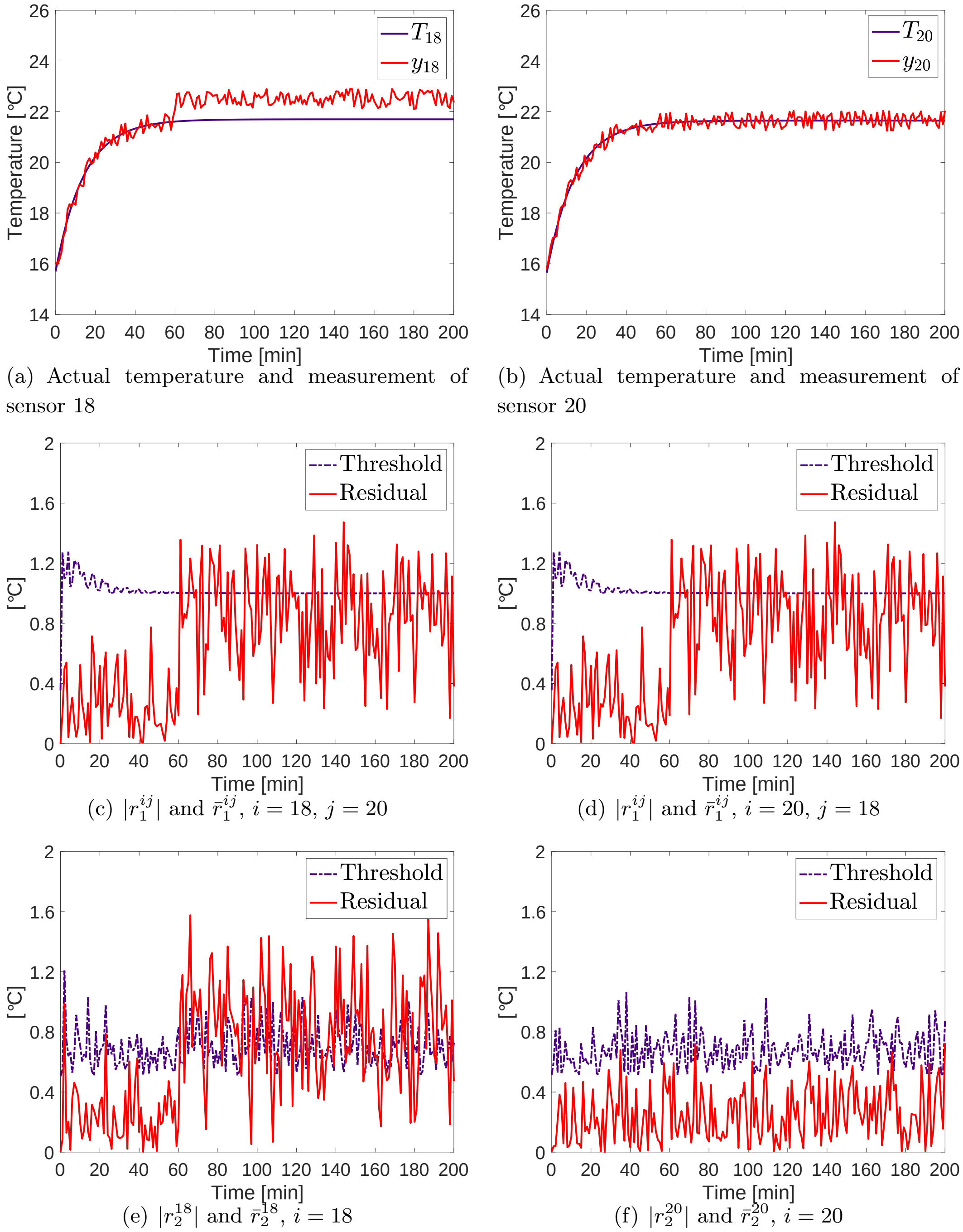
By analyzing $\bar{r}_1^{ij}$ both sensors can detect the presence of a fault at $k = 61min$. At the same time, by observing the residuals $|r_2^i|$ (detection for $|r_2^{18}|$ and no detection for $|r_2^{20}|$), the actual fault is isolated. Even though the fault is small in amplitude, the proposed algorithm manages to detect and isolate the fault and gives optimal results in terms of detection time. For this reason, no further simulations with faults having higher amplitude are necessary to prove the effectiveness of the method.

5.1.1.2 Temporary Fault

We can observe similar results in Figure 5.3 for the case of a temporary fault $f_{(18)}(k) = 1.5 \cdot [u(k - k_{f_{18,1}}) - u(k - k_{f_{18,2}})]$, $k_{f_{18,1}} = 60$ min, $k_{f_{18,2}} = 70$ min. The detection and isolation method is successful from the very beginning of the fault where the residuals cross the related thresholds in correspondence of the faulty values.

5.1.1.3 Linear Degrading Fault

We are now interested in the case when faults develop slowly in time and their amplitude is small. The fault detection and isolation method must be able to diagnose the fault before they are manifested and generate problems to the entire system. To this purpose, in Figure 5.4 we illustrate the relationship between the detection time and the amplitude of the fault, represented by the rate c in the case of linear degrading faults $f(k) = c \cdot k \cdot u(k - k_f)$ occurring at $k_f = 60$ min. The parameter c is chosen in the interval $[0.02, 1]$. As expected, as the amplitude of the fault increases, the detection delay $(k_d - k_f)$ becomes smaller.



(a) Actual temperature and measurement of sensor 18

(b) Actual temperature and measurement of sensor 20

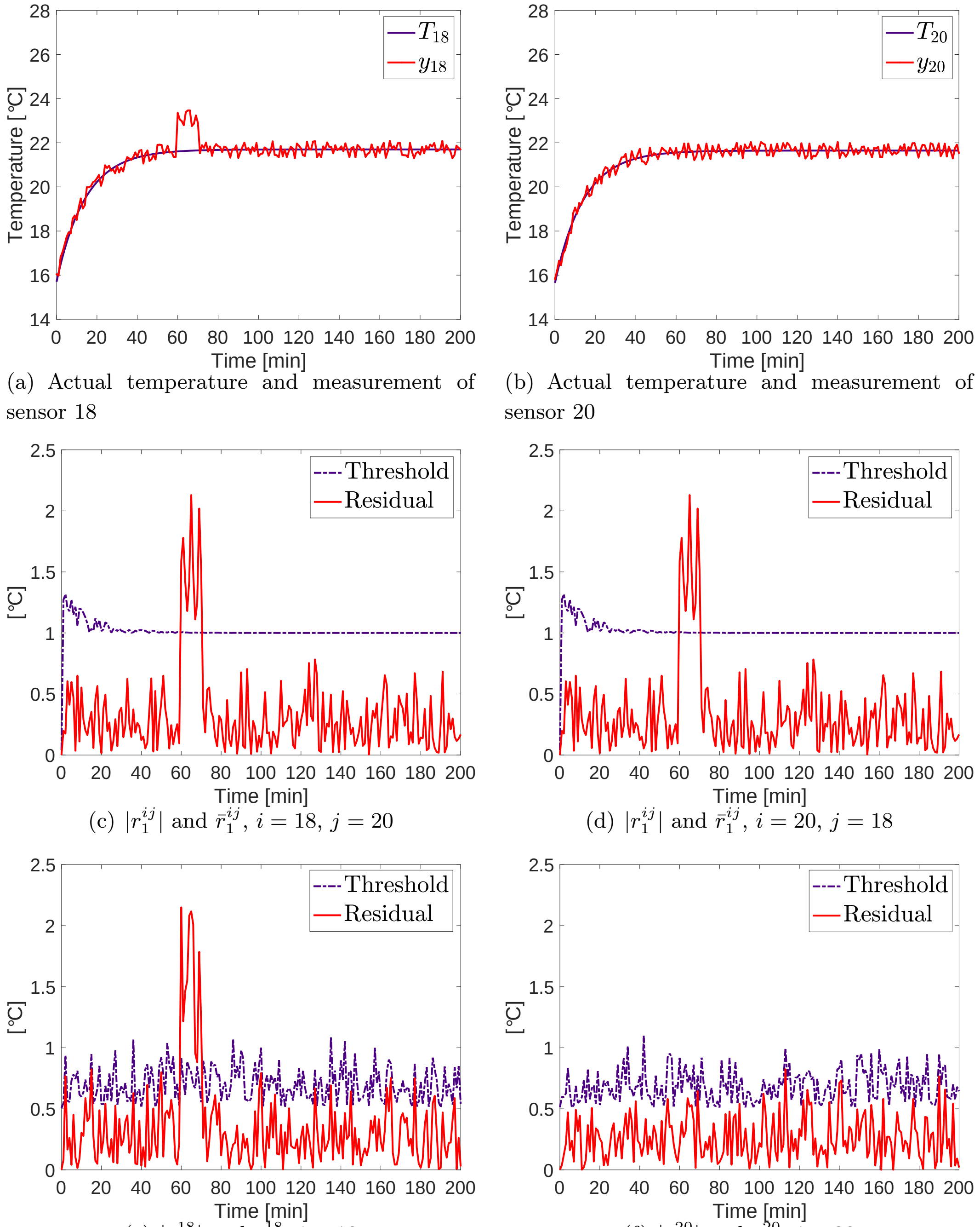
(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$, $i = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$

Figure 5.2: Comparison between sensor 18, affected by a single sudden fault $f_{(18)}(k) = 0.8 \cdot u(k - k_{f_{18}})$ at $k_{f_{18}} = 60$ min and the healthy sensor 20.



(a) Actual temperature and measurement of sensor 18

(b) Actual temperature and measurement of sensor 20

(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$, $i = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$

Figure 5.3: Comparison between sensor 18, affected by an intermittent fault $f_{(18)}(t) = 1.5 \cdot [u(k - k_{f_{18,1}}) - u(k - k_{f_{18,2}})]$, $k_{f_{18,1}} = 60$ min, $k_{f_{18,2}} = 70$ min, and the healthy sensor 20.

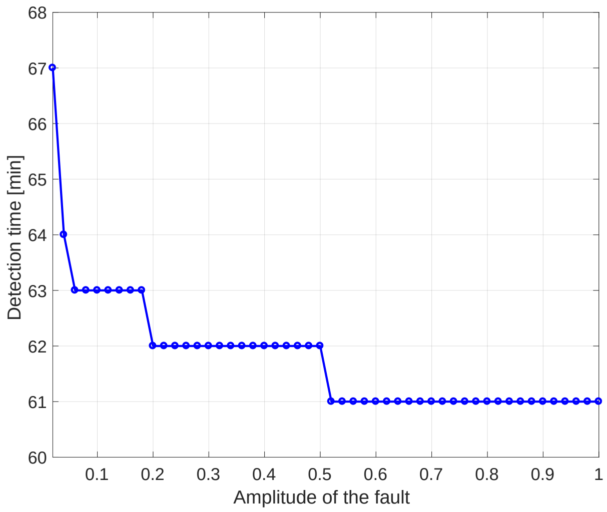


Figure 5.4: Detection time vs. fault amplitude (c parameter) for linear degrading faults.

5.1.2 Multiple Faults

5.1.2.1 Abrupt Faults

Let us now consider the case that multiple faults may simultaneously affect the sensors in the sensor network. For example, let consider the case that sensors 18 and 20 belonging to the same cluster are affected by faults $f_{(18)}(k) = 2 \cdot u(k - k_{f_{18}})$, $f_{(20)}(k) = 0.5 \cdot u(k - k_{f_{20}})$ and $k_{f_{18}} = 60$ min, $k_{f_{20}} = 61$ min. As it can be seen from Figure 5.5(c) and 5.5(d), a fault is detected at $k = 60$ min by means of the first residual. The detection and isolation of both faults are guaranteed for both sensors by means of the second residual. It is possible to see in Figure 5.5(f) that the detection is not persistent over time due to the small intensity of the fault affecting sensor 20.

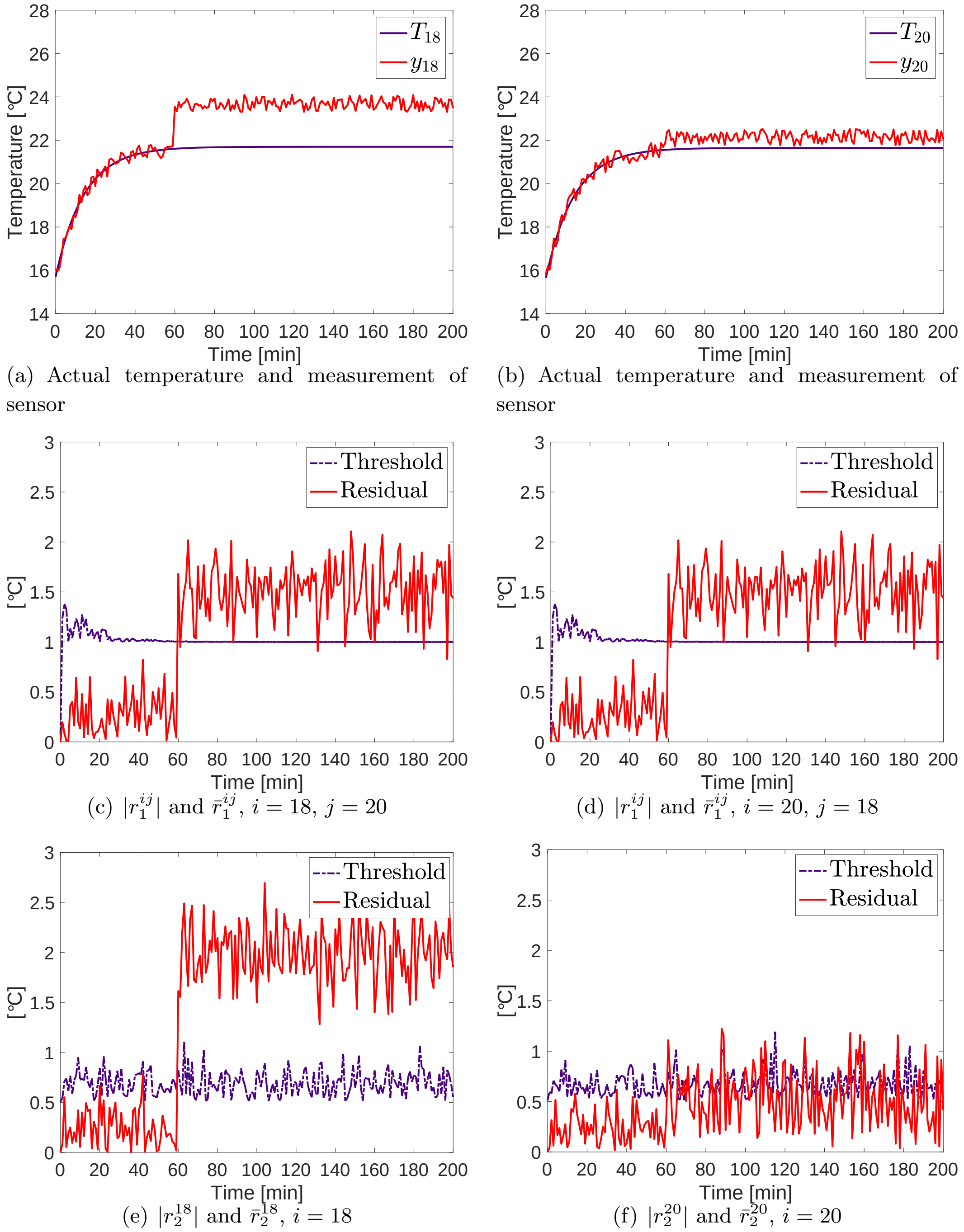
5.1.2.2 Linear Degrading Faults

We then consider in Figure 5.6 the case of multiple linear degrading faults occurring in sensors 18 and 20 belonging to the same cluster. The considered faults are $f_{(18)}(k) = 0.06 \cdot u(k - k_{f_{18}})$ occurring at $k_{f_{18}} = 60$ min and $f_{(20)}(k) = 0.02 \cdot u(k - k_{f_{20}})$ occurring at time $k_{f_{20}} = 61$ min. In Figure 5.6(d) we can see that the detection of the faults through the first residual is achieved by the two faulty sensors at $k = 76$ min. However, the second residual r_1^i detects the fault in sensor 18 at $k = 67$ min (Figure 5.6(e)), and at $k = 75$ min (Figure 5.6(f)) in sensor 20.

In Fig. 5.7 we consider a similar scenario with faults with different sign. We can see that in this case the simultaneous occurrence of the faults helps the detection by $\bar{r}_1^{ij}$, which is achieved at $k = 73min$, even if the fault magnitude is smaller on sensor 20.

We finally consider the case that the two faults are simultaneously affected by similar linear degrading fault charaterized by the same sign. It is possible to see in Fig. 5.8, as predicted by the isolability analysis in Section 4.4.2, that in this case the first residual r_1^{ij} is not able to detect the simultaneous multiple faults. It is necessary to analyse the second residual r_2^i both in 18 and 20 to correctly identify them.

We do not show the results for multiple temporary faults as they such case is analogue to the scenario of multiple sensor faults and does not give add any further contribution to what already said.



(a) Actual temperature and measurement of sensor

(b) Actual temperature and measurement of sensor

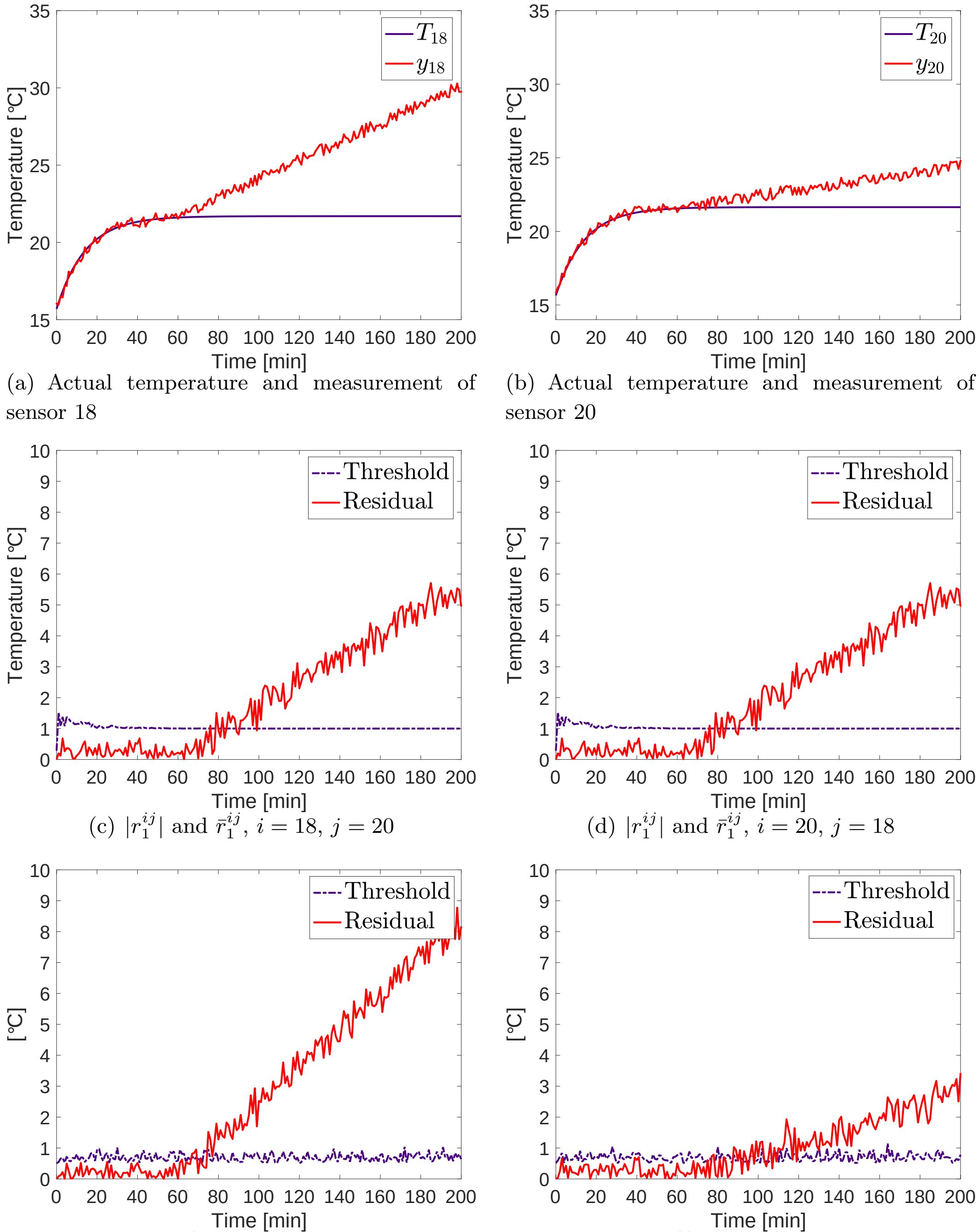
(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$, $i = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$

Figure 5.5: Comparison between sensor 18, affected by a sudden fault $f_{(18)}(k) = 2 \cdot u(k - k_{f_{18}})$ at $k_{f_{18}} = 60$ min and sensor 20 affected by $f_{(20)}(k) = 0.5 \cdot u(k - k_{f_{20}})$ at $k_{f_{20}} = 61$ min.



(a) Actual temperature and measurement of sensor 18

(b) Actual temperature and measurement of sensor 20

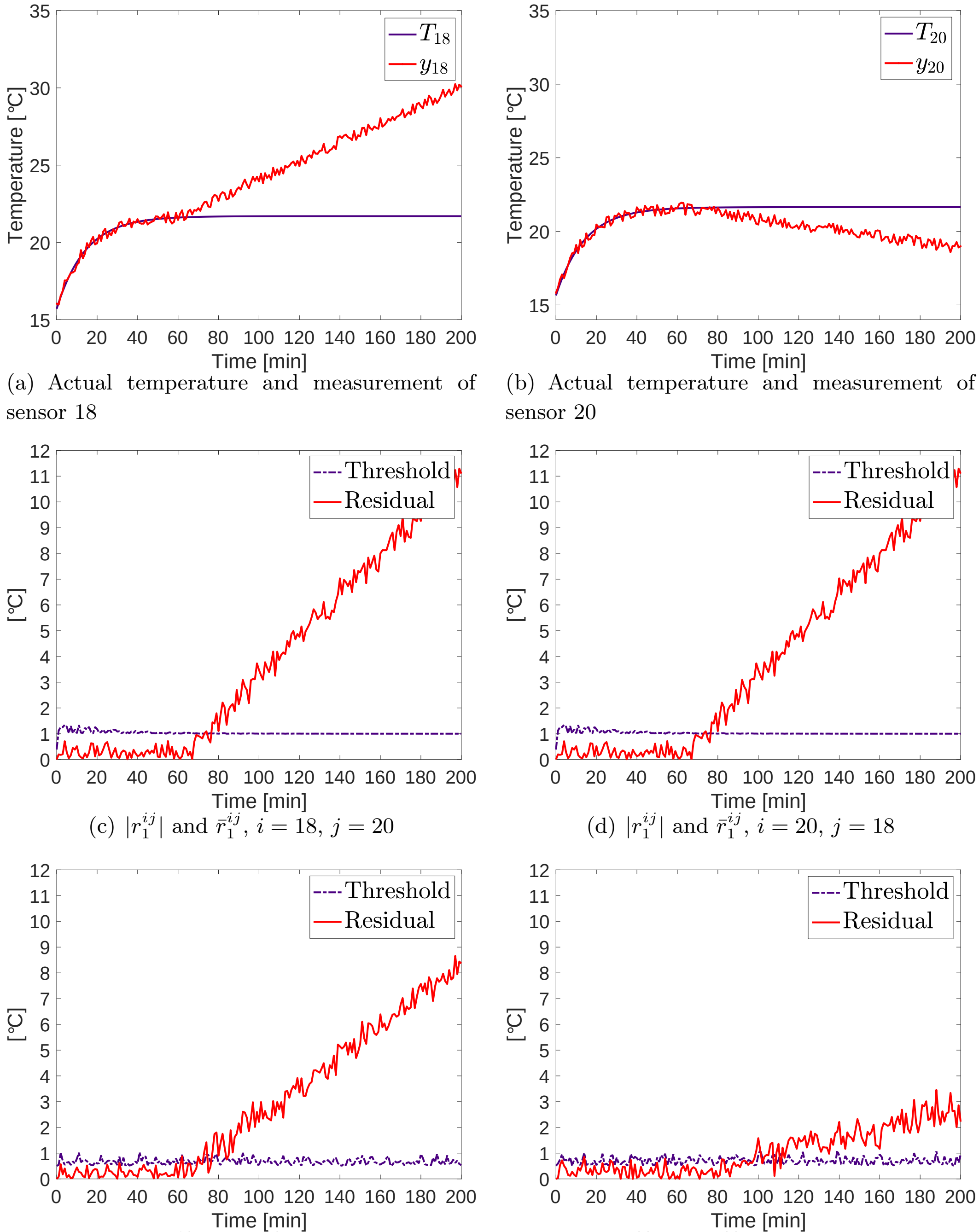
(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$,$i = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$

Figure 5.6: Comparison between sensor 18, affected by a linear degrading fault $f_{(18)}(k) = 0.06 \cdot u(k - k_{f_{18}})$ at $k_{f_{18}} = 60$ min and sensor 20 affected by $f_{(20)}(k) = 0.02 \cdot u(k - k_{f_{20}})$ at $k_{f_{20}} = 61$ min.



(a) Actual temperature and measurement of sensor 18

(b) Actual temperature and measurement of sensor 20

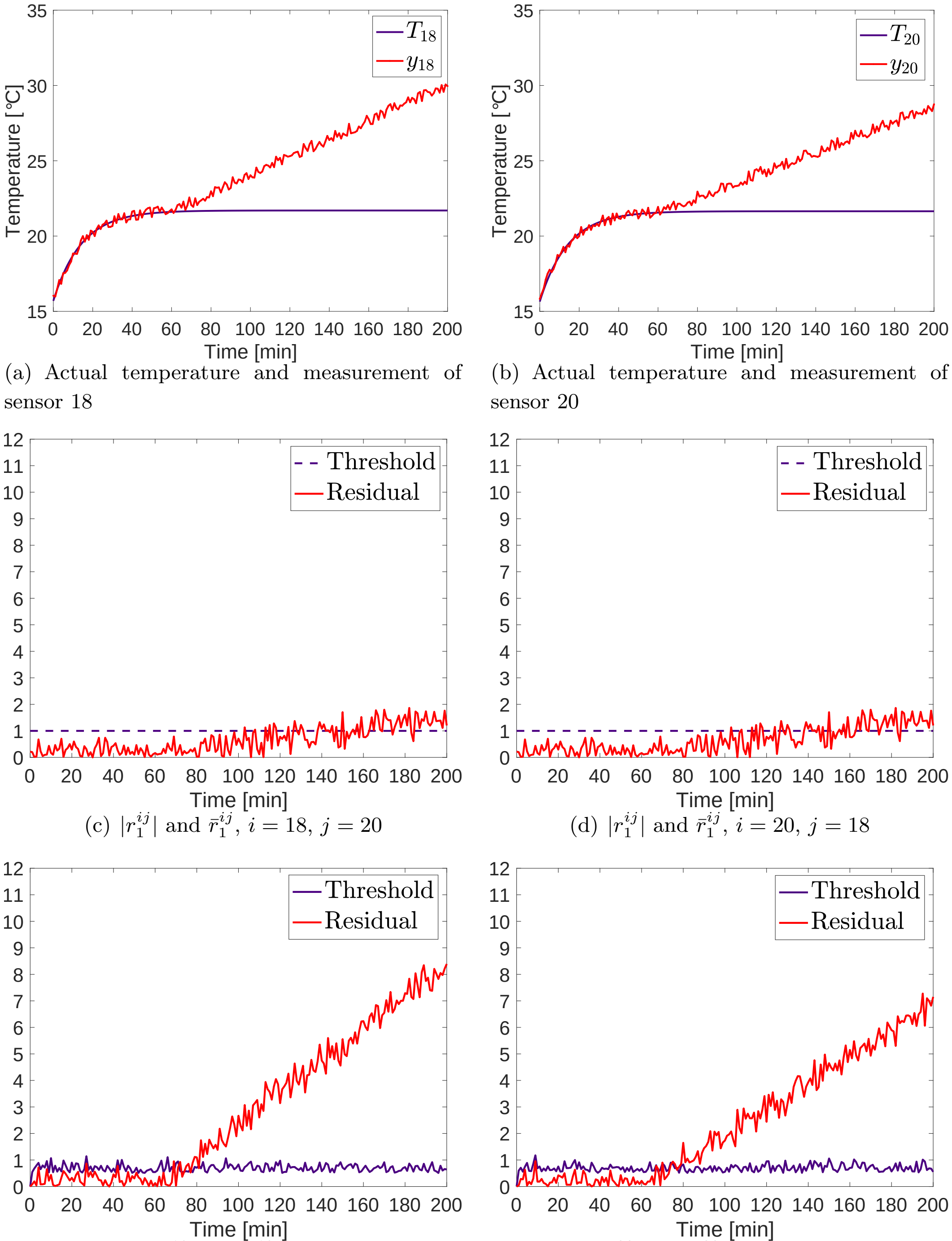
(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$, $j = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$ (faulty)

Figure 5.7: Comparison between sensor 18, affected by a linear degrading fault $f_{(18)}(k) = 0.06 \cdot u(k - k_{f_{18}})$ at $k_{f_{18}} = 60$ min and sensor 20 affected by $f_{(20)}(k) = -0.02 \cdot u(k - k_{f_{20}})$ at the same time.



(a) Actual temperature and measurement of sensor 18

(b) Actual temperature and measurement of sensor 20

(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$, $i = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$ (faulty)

Figure 5.8: Comparison between sensor 18, affected by a linear degrading fault $\phi^{(18)}(k) = 0.06 \cdot u(k - k_{f_{18}})$ at $k_{f_{18}} = 60$ min and sensor 20 affected by $\phi^{(20)}(k) = 0.05 \cdot u(k - k_{f_{20}})$ at $k_{f_{20}} = 61$ min.

5.2 Voronoi's Optimal Partition Deployment

In the second scenario, sensors are deployed according to Voronoi's optimal partition. An example of the network is shown in Figure 5.9. As it can be seen, as the sensors are distributed all over the area to monitor, the network is more connected than in the previous case. The clusters obtained are $\mathcal{C}^* = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}$, with $\mathcal{C}_1 = \{1, 2, 3, 4, 5, 6, 7\}$, $\mathcal{C}_2 = \{8, 9, 10, 11, 12, 13, 14, 15, 16\}$ and $\mathcal{C}_3 = \{17, 18, 19, 20\}$. Also in this case the conditions on distance and measurements similarity are satisfied, and the sensors closer to the source of uncertainty are grouped together (cluster $\mathcal{C}_3$). In particular, the aim of the simulations following below is to prove the effectiveness of our method independently from the topology of the network. In particular, we consider single and multiple faults for abrupt and linear degrading faults and neglect the temporary case since it does not carry any additional information.

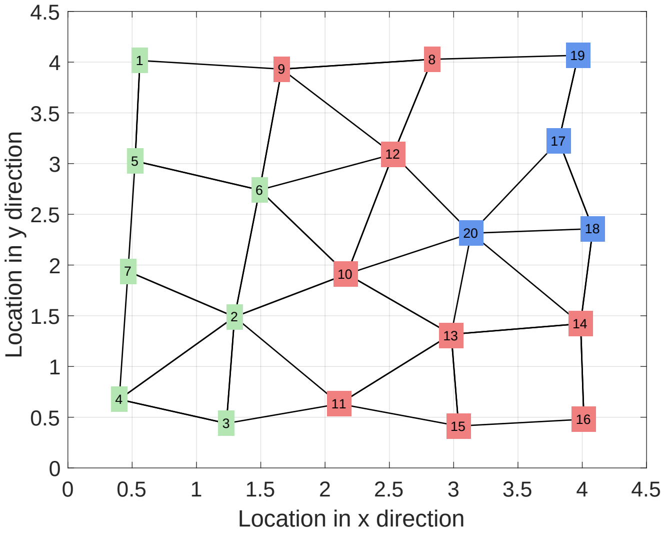


Figure 5.9: Wireless Sensor Network with 20 sensors deployed according to Voronoi's optimal partition. The green nodes belong to $\mathcal{C}_1$, the red nodes belong to $\mathcal{C}_2$ and the blue ones belong to $\mathcal{C}_3$.

5.2.1 Single fault

5.2.1.1 Abrupt fault

In this simulation, the same scenario proposed in Section 5.1.1.1 is presented and shown in Figure 5.10. The performances of the faulty sensor 18 are compared with those of the healthy sensor 20 belonging to the same cluster $\mathcal{C}_3$.

The fault in the first sensor 18 is detected through the first residual at time $k = 60$ min (Figure 5.10(c)) by both sensors $\{18, 20\}$, but is is not immediately detected by the second one (Figure 5.10(e)). However, both detection and isolation are guaranteed at $k = 61$ min through the second residual r_2^{18}. With respect to the case of single abrupt fault with initial random deployment presented in Section 5.1.1.1, in this case the detection is immediate, while the performances concerning the isolation procedure are the same in both cases.

5.2.1.2 Linear degrading fault

We show the results for a fault $f_{(18)}(k) = 0.08 \cdot u(k - k_{f_{18}})$ occurring in 18-th sensor at time $k_{f_{20}} = 60$ min (Figure 5.11). As expected, fault detection and isolation are non immediate because the fault is too small in the first time instants. By considering the first residual r_1^{ij}, $i = \{18, 20\}$ (Figure 5.11(c) and 5.11(d)), it is possible to detect the presence of a fault at time $k = 70$ min, while the second residual allows to detect and isolate the fault earlier, i.e. at time $k = 68$ min (Figure 5.11(e)).

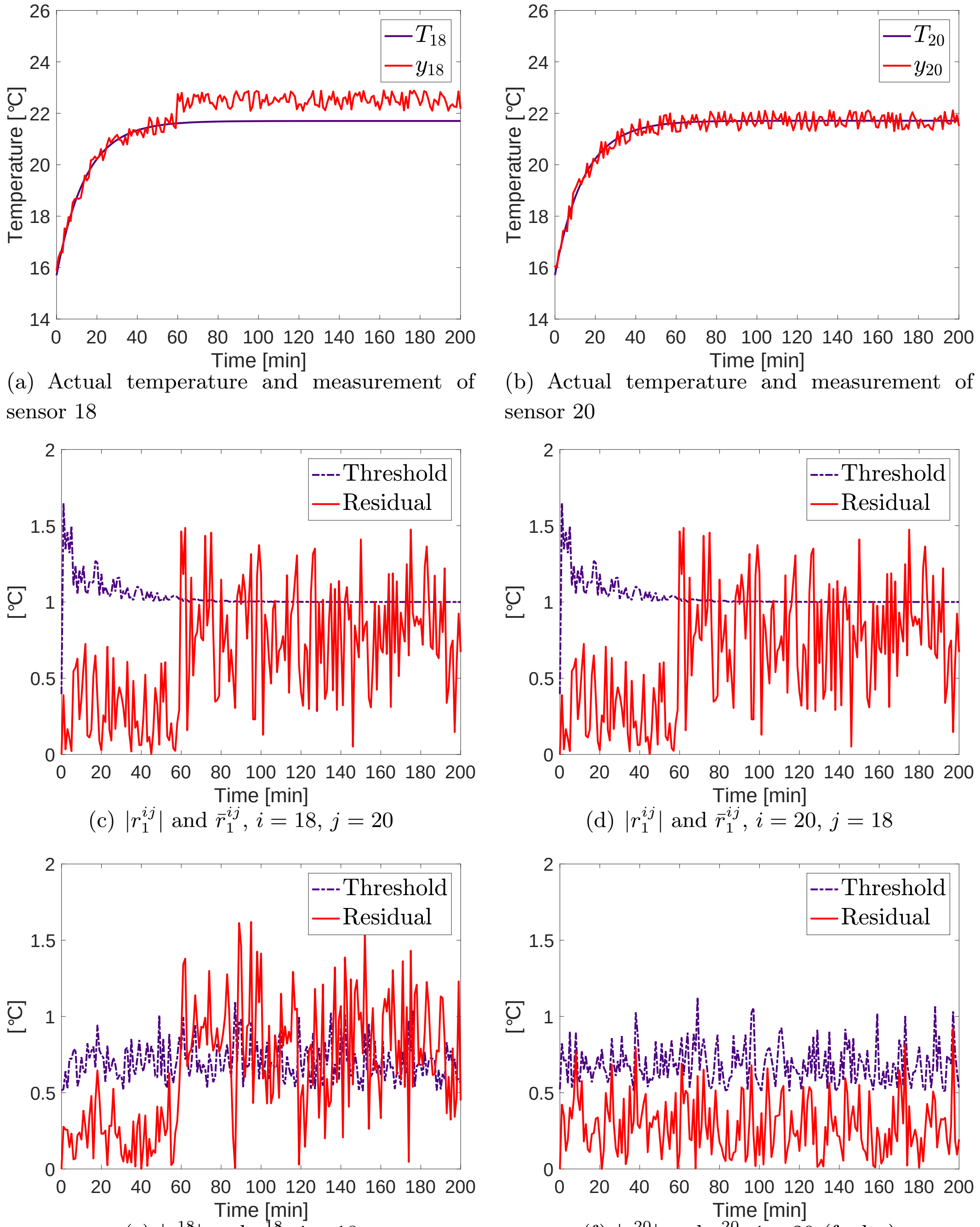
5.2.2 Multiple faults

5.2.2.1 Abrupt faults

We consider now the case when multiple faults simultaneously affect for example, sensors 18 and 20 belonging to the same cluster, affected by faults $f_{(18)}(k) = 2 \cdot u(k - k_{f_{18}})$, $f_{(20)}(k) = 0.5 \cdot u(k - k_{f_{20}})$ and $k_{f_{18}} = 60$ min, $k_{f_{20}} = 61$ min, as in the case of random deployment shown in Section 5.1.2.1. From Figure 5.12(c) and 5.12(d) it can be observed that a fault is detected at $k = 60$ min by means of the first residual r_1^{ij}, and the detection and isolation of both faults are guaranteed for both sensors by means of the second residual. Also in this case, since the second fault has a small intensity, its detection is not guaranteed at all time (Figure 5.12(f)).

5.2.2.2 Multiple faults

Finally we consider the case of multiple linear degrading faults of quite different amplitude occurring in sensors 18 and 20 at the same time $k_{f_{18}} = k_{f_{20}} = 60$min, respectively $\phi^{(18)}(k) = 0.06 \cdot u(k - k_{f_{18}})$ and $\phi^{(20)}(k) = 0.02 \cdot u(k - k_{f_{20}})$. The first residual detects the presence of a fault many time after the faults have occurred, more precisely at $k = 77$min (Figure 5.13(c) and 5.13(d)). The second residual $r_2^{(18)}$ detects and isolates the fault first at $k = 72$ min (Figure 5.13(e)), but full detection and isolation are guaranteed only for $k \geq 74$ min. For what concerns the second sensor, the fault is detected and isolated by the second residual at $k = 80$ min and perfect detection and isolation are guaranteed for $k \geq 131$ min (Figure 5.13(f)). Once again, even if the presence of a fault is detected, by means of only the first residual it is not possible to distinguish if the fault is single or multiple. Moreover, the fault of the second sensor results being more difficult to isolate since its amplitude is small and it can be compensated by noise and external disturbance, or by the same fault occurring in the first sensor.



(a) Actual temperature and measurement of sensor 18

(b) Actual temperature and measurement of sensor 20

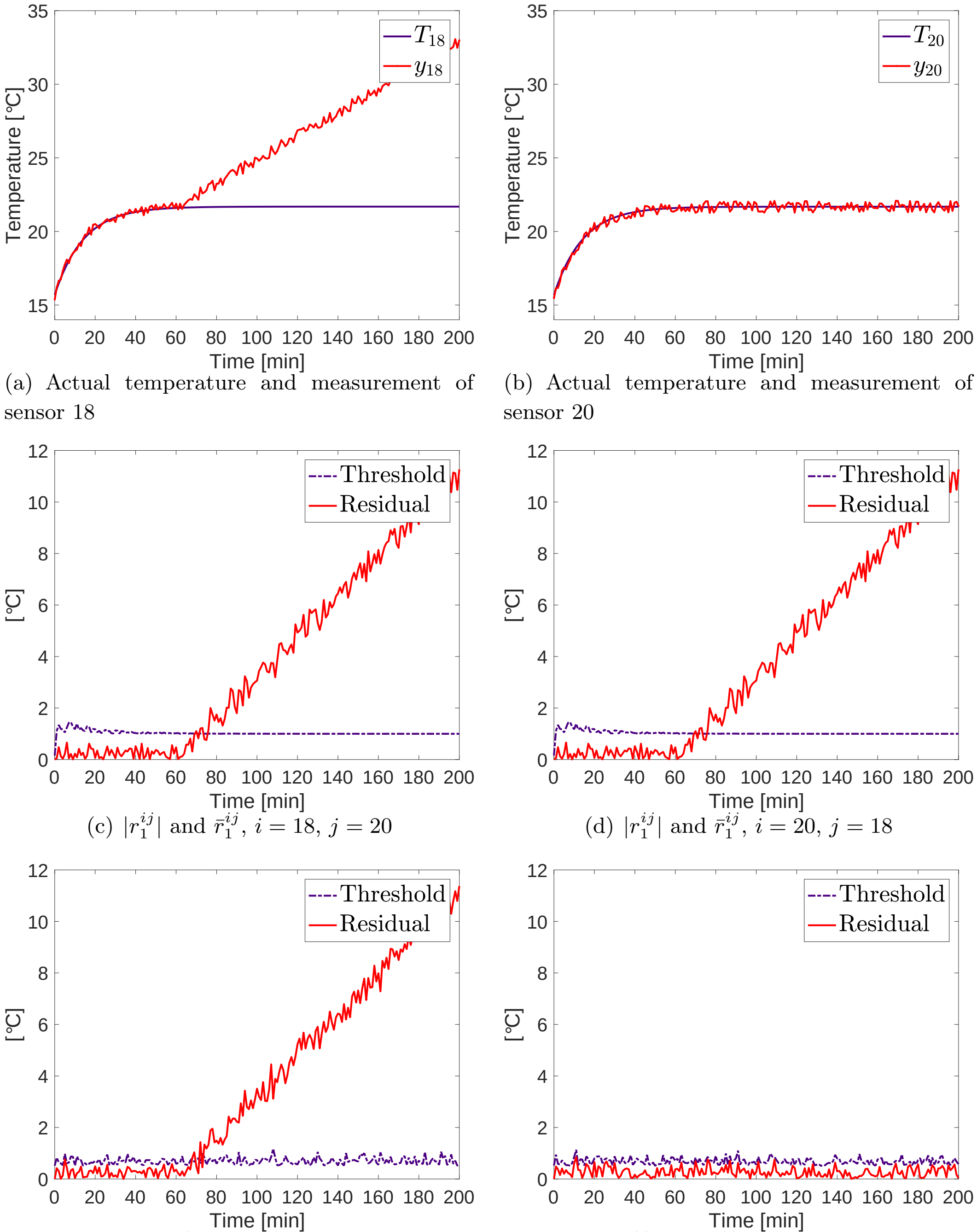
(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$, $i = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$ (faulty)

Figure 5.10: Comparison between sensor 18, affected by a single sudden fault $\phi^{(18)}(k) = 0.08 \cdot u(k - k_{f_{18}})$ at $k_{f_{18}} = 60$ min and sensor 20.



(a) Actual temperature and measurement of sensor 18

(b) Actual temperature and measurement of sensor 20

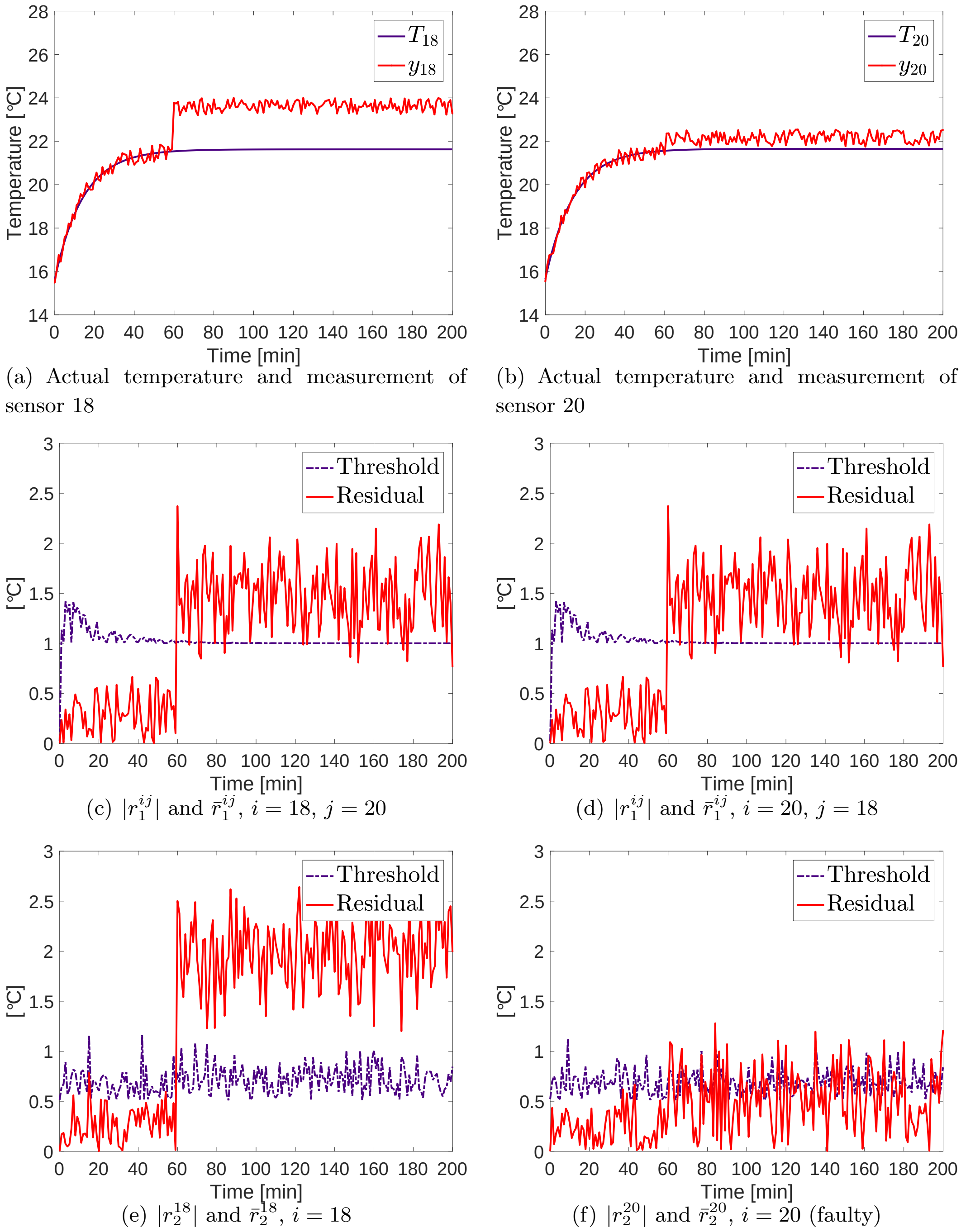
(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$, $i = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$ (faulty)

Figure 5.11: Comparison between sensor 18, affected by a linear degrading fault $\phi^{(18)}(k) = 0.08 \cdot u(k - k_{f_{18}})$ at $k_{f_{18}} = 60\text{min}$ and the healthy sensor 20.



(a) Actual temperature and measurement of sensor 18

(b) Actual temperature and measurement of sensor 20

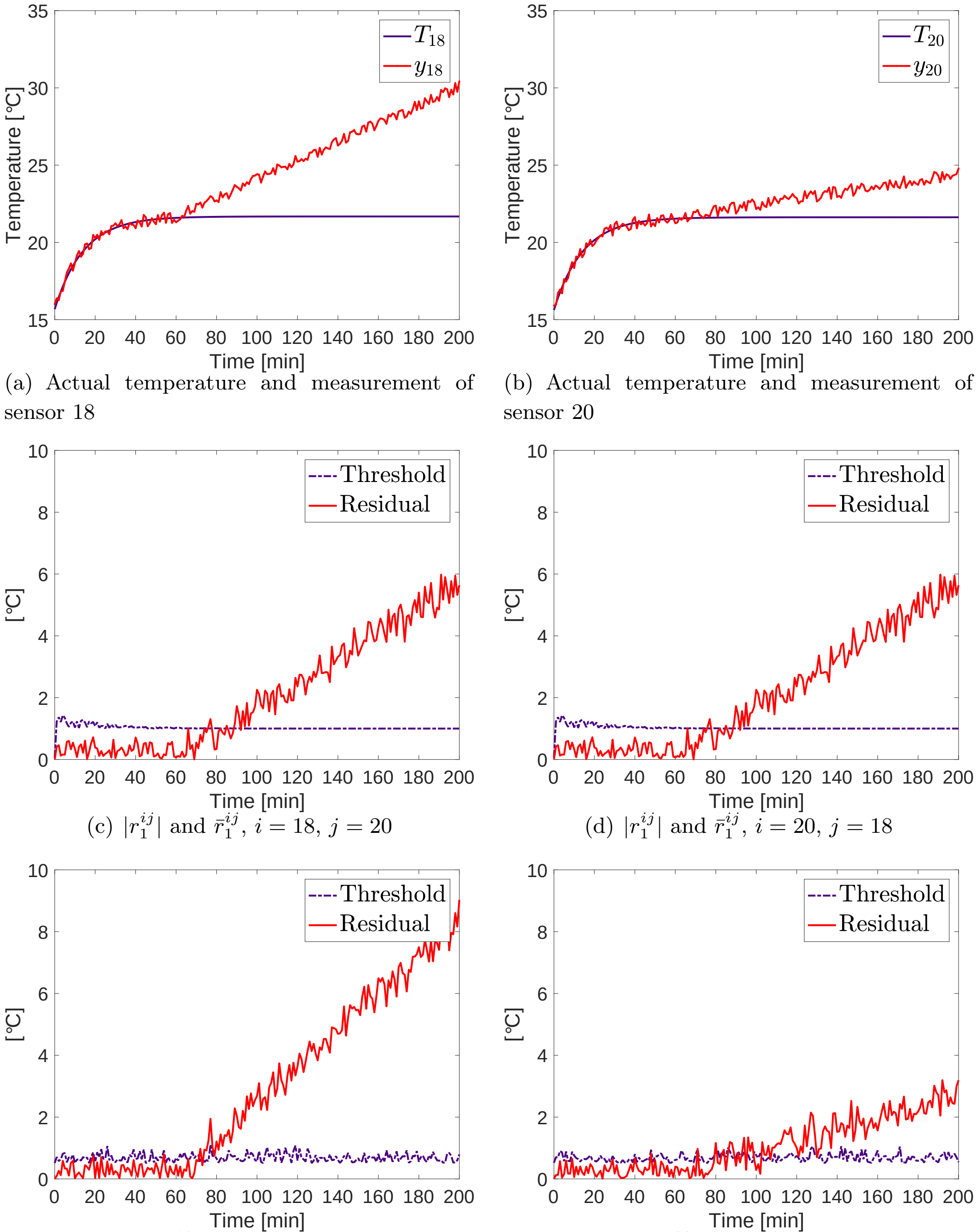
(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$, $i = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$ (faulty)

Figure 5.12: Comparison between sensor 18, affected by a single sudden fault $\phi^{(18)}(k) = 2 \cdot u(k - k_{f_{18}})$ at $k_{f_{18}} = 60\text{min}$ and sensor 20 affected by $\phi^{(20)}(k) = 0.5 \cdot u(k - k_{f_{20}})$ at $k_{f_{20}} = 61\text{min}$.



(a) Actual temperature and measurement of sensor 18

(b) Actual temperature and measurement of sensor 20

(c) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 18$, $j = 20$

(d) $|r_1^{ij}|$ and $\bar{r}_1^{ij}$, $i = 20$, $j = 18$

(e) $|r_2^{18}|$ and $\bar{r}_2^{18}$, $i = 18$

(f) $|r_2^{20}|$ and $\bar{r}_2^{20}$, $i = 20$

Figure 5.13: Comparison between sensor 18, affected by a linear degrading fault $\phi^{(18)}(k) = 0.06 \cdot u(k - k_{f_{18}})$ and sensor 20 affected by $\phi^{(20)}(k) = 0.02 \cdot u(k - k_{f_{20}})$ at the same time $k = 60$ min.

6 Conclusions and Future Works

In this thesis, a new distributed sensor fault diagnosis architecture based on a clustering approach is proposed for industrial wireless sensor networks monitoring HVAC systems. The HVAC system is modeled as a set of interconnected subsystems, made of the electromechanical part and a room. For the monitoring task of the room, we deploy N sensors which constitute an IWSN.

The sensor fault diagnosis method is tackled exploiting both network decomposition and data clustering, which allows to group sensors belonging to the same regions of a zone and sharing similar features. Partitioning the network has two important advantages for performing sensor fault detection:

- communication is reduced, because each sensor can share information with sensors belonging to the same cluster and do not need to communicate with all the nodes of the network;
- nodes can be grouped according to similar uncertainties and disturbances and detection can be carried on exploiting only measurement locality.

In particular, the second property is obtained by substituting the measurement criterion with a consistency one, which allows the sensors to measure different quantities and compute the clustering bound b in a distributed way, communicating with only their neighbours.

The performance of the proposed methodology is analyzed with respect to sensor fault detectability and isolability. It is proven that the novelty of using two different residual signals may possibly increase the detectability performance of the proposed distributed architecture and allows to isolate faults distinguishing between local and neighbouring faults. Simulation results are used for illustrating the effectiveness of the proposed methodology applied to the aforementioned HVAC system. In particular, we consider single and multiple faults, and different models of faults are taken into account, such as abrupt, intermittent and linear degrading ones. In the first and second case, faults are immediately diagnosed, while for the last case, detection and isolation strongly depend on the intensity of the fault. We show the relation between the fault amplitude and the detection time in the case of a single fault. When multiple faults occur, diagnosis depends also on the sign of the faults and accordingly it can improve or compromise the fault detection. Moreover, we show the results obtained are valid independently from the initial deployment of the sensors, i.e. from the given topology.

The clustering-based distributed sensor fault diagnosis architecture defined in this thesis may have different practical applications in smart buildings: it could contribute to the reduction of energy consumption in large-scale buildings, it can allow the reduction of maintenance work and finally improve occupants' comfort. Moreover, the reliability ensured by the IWSN makes it possible to deploy the sensors in harsh environments, enhancing their spread and application. Future research work will involve the application of the proposed method to the case of possible simultaneous presence of process and sensor faults. The challenge in this case it to distinguish the compensated effects of sensor and process faults on the residuals obtained. Another open topic refers to the diagnosis of both sensor and actuator faults, in particular for what concerns the isolation process, in case of multiple faults and/or propagated faults. Finally, the design of the clustering-based sensor fault diagnosis could consider communication problems and limitations such as delays, packet dropouts, ecc.

A

Terminology in Model-based Fault Diagnosis

In this appendix, definitions extracted from the *IFAC PROCESS* Technical Committee's initiative of defining common terminology in the field of model-based fault diagnosis are presented.

A.1 State and Signals

Fault: An unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition.
Failure: A permanent interruption of a system's ability to perform a required function under specified operating conditions.
Malfunction: An intermittent irregularity in the fulfilment of a system's desired function.
Error: A deviation between a measured or computed value (of an output variable) and the true, specified or theoretically correct value.
Disturbance: An unknown (and uncontrolled) input acting on a system.
Perturbation: An input acting on a system, which results in a temporary departure from the current state.
Residual: A fault indicator, based on a deviation between measurements and model-equation-based computations.
Symptom: A fault indicator, based on a deviation between measurements and model-equation-based computations.

A.2 Functions

Fault detection: Determination of the faults present in a system and the time of detection.
Fault isolation: Determination of the kind, location and time of detection of a fault. Follows fault detection.
Fault identification: Determination of the size and time-variant behaviour of a fault. Follows fault isolation.
Fault diagnosis: Determination of the kind, size, location and time of detection of a fault. Includes fault detection, isolation and identification.
Monitoring: A continuous real-time task of determining the conditions of a physical system, by recording information, recognising and indicating anomalies in the behaviour.
Supervision: Monitoring a physical system and taking appropriate actions to maintain the

operation in the case of faults.

A.3 Models

Quantitative model: Use of static and dynamic relations among system variables and parameters in order to describe a system's behaviour in quantitative mathematical terms.
Qualitative model: Use of static and dynamic relations among system variables and parameters in order to describe a system's behaviour in qualitative terms such as causalities or if-then rules.

A.4 System Properties

Reliability: Ability of a system to perform a required function under stated conditions, within a given scope, during a given period of time.
Safety: Ability of a system not to cause danger to persons or equipment or the environment.
Availability: Probability that a system or equipment will operate satisfactorily and effectively at any point of time.

$\mathcal{B}$

Voronoi Tessellations

Let the region of deployment be modeled as a convex polygon $Q \in \mathbf{R}^2$ and let the distribution density function $\phi : Q \rightarrow \mathbf{R}^+$ be bounded and continously differentiable; $\phi(q_i)$ represents the probability that a certain event takes place in a given point $q_i \in Q$. Moreover, let $P = (p_1, ..., p_n)$ be the positions of the N sensors in the region. It is reasonable to assume that the sensors have a finite sensing range, therefore their performance decreases with the distance from the event. In particular, the degradation rate is modelled as proportional to the square of the Euclidean distance among the sensors and the points $q_i \in Q$. Considering a distributed approach, sensors are capable of making observations independently, without any need of synchronizing with a base station.

Optimal deployment is achieved when the locational optimization function

$$\mathcal{H}(P, W) = \sum_{i=1}^{N} \int_{W_i} \|q_i - p_i\|^2 \phi(q)\, dq \tag{B.1}$$

is minimized, where $W = \{W_1, ..., W_N\}$ is a partition of Q such that $p_i \in W_i$, i.e. each sensor is responsible for measurements over its dominance region W_i. This optimization problem is twofold, since the cost function (B.1) needs to be minimized both with respect to W (the choice of how the region is partitioned) and with respect to P (the positions of the sensors in said partitions).

It can be proven that, at fixed sensors locations, the optimal partition of Q is given by the Voronoi tessellation $V(P) = \{V_1, ..., V_N\}$, where each V_i has as generator the position p_i of the i-th sensor:

$$V_i = \{q \in Q : \|q - p_i\| \leq \|q - p_j\|, \forall i \neq j\}. \tag{B.2}$$

Therefore, in our attempt to minimize the cost function (B.1), we may replace W with $V(P)$ in order to obtain $\mathcal{H}_V(P) = \mathcal{H}(P, V(P))$.

However, we still need to minimize $\mathcal{H}_V(P)$ with regards to P. It can be proven that the stationary point found by setting to zero the first order derivative of $\mathcal{H}_V(P)$ with respect to P is at least a local minimum. Moreover, the derivation sign can be brought inside the integral; therefore we obtain

$$\frac{\partial}{\partial p_i}\mathcal{H}_V(P) = \int_{Vi} \frac{\partial}{\partial p_i} \|q_i - p_i\|^2 \phi(q)\, dq \tag{B.3}$$

$$= \int_{Vi} -2(q - p_i)^T \phi(q)\, dq. \tag{B.4}$$

At this point, it is useful to define some quantities associated to each partition V_i, namely its (generalized) mass M_{V_i}, centroid C_{V_i} (also called center of mass), and polar moment of inertia $J_{V_i,p}$. They can be determined through these formulas:

$$M_{V_i} = \int_{V_i} \phi\left(q\right) dq \tag{B.5}$$

$$C_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} q\phi\left(q\right) dq \tag{B.6}$$

$$J_{V_i,p} = \int_{V_i} \left\|q - p\right\|^2 \phi\left(q\right) dq. \tag{B.7}$$

Using these quantities we just defined, the cost function $\mathcal{H}_V\left(P\right)$ can be rewritten as

$$\mathcal{H}_V\left(P\right) = \sum_{i=1}^{N} J_{Vi,C_{V_i}} + \sum_{i=1}^{N} M_{V_i} \left\|p - C_{V_i}\right\|^2 \tag{B.8}$$

and, more importantly, (B.4) becomes

$$\frac{\partial}{\partial p_i} \mathcal{H}_V\left(P\right) = 2M_{V_i}\left(p_i - C_{V_i}\right)^T. \tag{B.9}$$

Looking at (B.9), it is clear that the partial derivative is equal to zero when $p_i = C_{V_i}$. Therefore, we can conclude that the objective function is minimized when the sensors are located in positions p_i that are both generators and centroids of the corresponding partition. This particular type of Voronoi diagram is called Centroidal Voronoi Tessellation (CVT), and it combines optimal partitioning of the region with optimal placement of the generators. On the other hand, it is worth noting that, since the critical points given by the derivative are generally only local minima and no convergence to a global minimum of (B.1) is implied, there can be multiple solutions to the optimization problem. This means that there can be multiple different CVTs with different coverage costs with respect to the same region $Q \in \mathbf{R}^2$, number of sensors and density ϕ.

Acknowledgments

This journey would have not be possible without the support of several people who, in one way or another, contributed to its completion. I would like to express my gratitude to all those who believed in me and constantly gave me their support.

First of all I would like to thank my advisors Angelo Cenedese, for his patience and for offering me the chance to go to London, and Thomas Parisini, who accepted me and welcomed me at Imperial College, making this experience possible.

I am deeply indebted to Francesca Boem, for her help, her advices, her presence, her unbelievable patience and her guidance during all my period in London.

Thanks to my best friend Valentina. You are a sister to me and I know that you will always be there for me wherever we will be.

Thanks to Sabrina, Giacomo and Riccardo for their patience and for listening to me, giving me the strength to carry on and never give up.

A special thank to Andrea, for his patience, for his advices, for his help, and in particular for his capacity to always brighten my grey days. There are no sufficient words to describe how much I owe you.

I would like to thank Carlos, my very first friend in my Erasmus in Barcelona. You helped me when I was down and taught me to enjoy life as it is, be positive and always smile. We shared awesome moments together and I really hope to meet you soon.

Thanks to the best flatmates ever, Rachele and Valeria. All the time spent together was awesome and I am glad that our friendship went beyond the experience of living together.

Thanks to my colleagues at university and all my friends spread around the world. We shared stories, experiences, travels, laughs, tears and each of you taught me something.

Last but not least, many thanks to my parents Alim and Flora. Thank you for encouraging me in all my pursuits and inspiring me to follow my dreams. Thank you for accepting my decisions and supporting me emotionally and financially. I know the sacrifices you make everyday and I hope to repay it all back one day.

Bibliography

[1] P. Antsaklis *et al.*, “Control of cyberphysical systems using passivity and dissipassivity based methods,” in *Eur. J. Control*, vol. 19, no. 5, 2013, pp. 379–388. 17

[2] J. Yuan, C. Farnham, and K. Emura, “Development and application of a simple BEMS to measure energy consumption of buildings,” in *Energy and buildings*, vol. 109, 2015, pp. 1–11. 17

[3] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information,” in *Energy and buildings*, vol. 40, no. 3, 2008, pp. 394–398. 17

[4] U. D. Energy Information Administration. (2003) Commercial buildings energy consumption survey. 17

[5] M. Gul and S. Patidar, “Understanding the energy consumption and occupancy of a multi-purpose academic building,” in *Energy and buildings*, vol. 87, 2015, pp. 155–165. 17

[6] S. Gupta, M. S. Kar, K., and J. Wen, “Collaborative energy and thermal comfort management through distributed consensus algorithms,” in *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, 2015, pp. 1285–1296. 18

[7] L. Parolini, B. Sinopoli, B. Krogh, and Z. Wang, “A cyber-physical systems approach to data center modeling and control for energy efficiency,” in *Proceedings of the IEEE*, vol. 100, no. 1, 2012, pp. 254–268. 18

[8] Q. Fang, J. Wang, and Q. Gong, “QoS-driven power management of data centers via model predictive control,” in *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 4, 2016, pp. 1557–1566. 18

[9] V. Gungor and G. Hancke, *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*, 1st Editions, Ed. CRC Press, Inc., Boca Raton, FL, USA, 2013. 18, 39, 40

[10] V. Reppa, M. M. Polycarpou, and C. Panayiotou, “Sensor fault diagnosis,” *Foundations and Trends*, vol. 3, no. 1-2, pp. 1–248, 2016. 18

[11] W. Zhang, G. Han, Y. Feng, L. Cheng, D. Zhang, X. Tan, and L. Fu, "A novel method for node fault detection based on clustering in industrial wireless sensor networks," in *International Journal of Distributed sensor networks*, 2015. 18, 19

[12] G. Bianchin, A. Cenedese, A. Luvisotto, and G. Michieletto, "Distributed fault detection in sensor networks via clustering and consensus," in *54th IEEE International Conference on Decision and Control (CDC)*, 2015, pp. 3828–3833. 18, 20

[13] A. Abbassi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," in *Computer Communications*, vol. 30, no. 1415, 2007, pp. 2826–2841. 18

[14] Y. Ma, Y. Guo, and M. Tian, X. abd Ghanem, "Distributed clustering-based aggregation algorithm for spatial correlated sensor networks," in *IEEE Sensors Journal*, vol. 11, no. 3, 2011, pp. 641–648. 18

[15] H.-C. Shih, J.-H. Ho, B.-Y. Liao, and J. Pan, "Fault node recovery algorithm for a wireless sensors network," in *IEEE Sensors Journal*, vol. 13, no. 7, 2013, pp. 2683–2689. 19

[16] G. Gupta and M. Younis, "Fault-tolerant clustering of wireless sensor networks," in *Wireless Communications and Networking (WCNC), IEEE Conf. on*, vol. 3, 2003, pp. 1579–1584. 19

[17] P. Chen, S. Yang, and J. McCann, "Distributed real-time anomaly detection in networked industrial sensing systems." in *IEEE Transactions on Industrial Elec- tronics*, vol. 62, no. 6, 2015, pp. 3832–3842. 19

[18] F. Boem, R. Ferrari, and T. Parisini, "Distributed fault detection and isolation of continuous-time non-linear systems," in *European Journal of Control*, vol. 17, no. 5, 2011, pp. 603–620. 19

[19] S. Stankovic, N. Illic, Z. Djurovic, M. Stankovic, and J. K., "Consensus based overlapping decentralized fault detection and isolation," in *Control and Fault-Tolerant Systems Conference,*, 2010, pp. 570–575. 19

[20] F. Boem, R. Ferrari, C. Keliris, T. Parisini, and M. M. Polycarpou, "A distributed networked approach for fault detection of large-scale systems," in *IEEE Transactions on Automatic Control (To appear)*, doi: 10.1109/TAC.2016.2539326, 2016. 19

[21] V. Reppa, M. Polycarpou, and C. Panayiotou, "Multiple sensor fault detection and isolation for large-scale interconnected nonlinear systems," in *European Control Conference*, Zürich, Switzerland, 17-19 July 2013. 19

[22] A. Cenedese, A. Luvisotto, and G. Michieletto, "Distributed clustering strategies in industrial wireless sensor networks," in *Transactions on Industrial Informatics (To appear)*, doi:10.1109/TII.2016.2628409, 2016. 20, 39, 43, 49, 53

[23] W. Grondzik and R. Furst, "HVAC components and systems," vital Signs Curriculum Material Project. 23

[24] A. Thosar, A. Patra, and S. Bhattacharyya, "Feedback linearization based control of a variable air volume air conditioning system for cooling applications," in *ISA Trans.*, vol. 47, no. 3, 2008, pp. 339–349. 26

[25] J. Farrell and M. Polycarpou, *Adaptive Approximation Based Control*, P. John Wiley & Sons, Inc., Ed. Wiley-Interscience, 2006. 28

[26] R. B. Guenther and J. W. Lee, *Partial Differential Equations of Mathematical Physics and Integral Equations*, new ed. Dover Publications Inc., June 1996. 35

[27] T. Myint-U and L. Debnath, *Linear Partial Differential Equations for Scientists and Engineers*, Fourth, Ed. Birkhäuser, 2007. 35

[28] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. Kavuri, "A review of process fault detection and diagnosis, part i: Quantitative model-based methods," *Computer & Chemical Engineering*, vol. 27, no. 3, pp. 293–311, 2003. 42

[29] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks.* Princeton University Press, 2010. 43

[30] I. Shames, A. Teixeira, H. Sanberg, and K. Johansson, "Distributed fault detection for interconnected second order systems," in *Automatica*, vol. 47, no. 12, 2011, pp. 2757–2764.

[31] V. Reppa, P. Papadopoulos, M. Polycarpou, and C. Panayiotou, "A distributed architecture for HVAC sensor fault detection and isolation," in *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, July 2015.

[32] R. Isermann, *Fault-Diagnosis Systems.* Springer, 2006.

Acknowledgments

This journey would have not be possible without the support of several people who, in one way or another, contributed to its completion. I would like to express my gratitude to all those who believed in me and constantly gave me their support.

First of all I would like to thank my advisors Angelo Cenedese, for his patience and for offering me the chance to go to London, and Thomas Parisini, who accepted me and welcomed me at Imperial College, making this experience possible.

I am deeply indebted to Francesca Boem, for her help, her advices, her presence, her unbelievable patience and her guidance during all my period in London.

Thanks to my best friend Valentina. You are a sister to me and I know that you will always be there for me wherever we will be.

Thanks to Sabrina, Giacomo and Riccardo for their patience and for listening to me, giving me the strength to carry on and never give up.

A special thank to Andrea, for his patience, for his advices, for his help, and in particular for his capacity to always brighten my grey days. There are no sufficient words to describe how much I owe you.

I would like to thank Carlos, my very first friend in my Erasmus in Barcelona. You helped me when I was down and taught me to enjoy life as it is, be positive and always smile. We shared awesome moments together and I really hope to meet you soon.

Thanks to the best flatmates ever, Rachele and Valeria. All the time spent together was awesome and I am glad that our friendship went beyond the experience of living together.

Thanks to my colleagues at university and all my friends spread around the world. We shared stories, experiences, travels, laughs, tears and each of you taught me something.

Last but not least, many thanks to my parents Alim and Flora. Thank you for encouraging me in all my pursuits and inspiring me to follow my dreams. Thank you for

accepting my decisions and supporting me emotionally and financially. I know the sacrifices you make everyday and I hope to repay it all back one day.