A model-parameter invariant approach to HVAC fault detection and diagnosis

DARIO PAPALE

Master’s Degree Project
Stockholm, Sweden Sweden, September 2012

XR-EE-RT 2012:027
List of acronyms

**BA**  
*Building Automation*

**HVAC**  
*Heating, Ventilation and Air Conditioning*

**FDD**  
*Fault Detection and Diagnostics*

**ODE’s**  
*Ordinary Differential Equations*

**PDE’s**  
*Partial Differential Equations*

**GP**  
*Gaussian Process*

**SPRT**  
*Sequential Probability Ratio Test*

**WSN**  
*Wireless Sensor Network*

**SCADA**  
*Supervisory Control And Data Acquisition*
Abstract

This thesis develops and experimentally evaluates a model-based detector for detecting actuator failures in HVAC systems which dynamically estimates the model parameters while performing detection. Specifically, this thesis considers actuator failures which result in the actuator valve sticks in an unknown (but constant) position. A first order heat-equation model is assumed to model interactions between adjacent rooms, which is used to formulate a hypothesis testing problem assuming that the inter-room thermal parameters are constant. The detector is formulated to provide performance that asymptotically bounds both the probability of miss and the probability of false alarm. A simulation environment is developed that emulates the building thermal dynamics assuming parameters identified through experimental testing. Multiple testing scenarios are considered where the inter-room dynamics are physically altered between tests (by opening and closing windows and doors). Results indicate that the ability to accurately detect actuator failures is shown to be dependent on the inter-room thermal dynamics. Insights into other model-parameter-invariant based detectors are provided based on the results obtained from this thesis.
The term *Building Automation* (BA) indicates the application of automatic control techniques to the management of a building. Since the 1980’s BA has become an important issue in the modern conception of buildings [2]. In order to ensure comfort and to improve the quality of the time spent inside the buildings, different branches of engineering have cooperated to give birth to this new field.

The BA system is a computerized, intelligent network of electronic devices designed to monitor and control the mechanical devices, lighting systems and air conditioning systems in a building. These systems are designed to keep the building climate within a specified range, to provide lighting based on an occupancy schedule, to monitor system performance and device failures, and to provide feedback information to building engineering/maintenance staff, and in the more recent applications, to the people who live or work inside the building. Additionally, such systems reduce energy and maintenance costs of a building when compared to a non-controlled one. An example of BA is shown in figure 1.1.

A fundamental part of BA is *Heating, Ventilation and Air Conditioning* (HVAC): HVAC systems are designed to ensure a high quality of the air inside buildings [2]. Although air quality is determined by different factors, such as temperature, humidity, light and CO₂ levels, this work focuses just on temperature.

The main components of an HVAC system are specific to its functionality. In the HVAC system employed for this work, there are radiators for heating purpose, air conditioning for cooling and ceiling fans for ventilation issues. Heat is provided by a furnace and distributed through water pipes to the radiators. Ventilation includes both the exchange of air with the outside and the circulation of air within the building: the treated air, after a filtering, dehumidifying and mixing process, is forced to flow through the ducts of distribution. Air conditioning has mostly cooling purposes, but it can be used for heating, ventilation and humidity control. Cooling is provided by a refrigeration cycle and the fresh air is distributed in ceiling ducts parallel to ventilation ones.
1.1 Motivation

With increasing energy prices, energy saving has become a primary goal in the development of new technologies. Studies indicate that residential, office and commercial buildings account for nearly 40% and 47% of the U.S. and U.K. energy consumption, respectively [26, 25]. HVAC systems are known to be the largest contributor, accounting for 43% of U.S. residential energy consumption, therefore the design of energy-efficient HVAC systems has become a worldwide research priority. Recently, several researchers have studied the improvement of HVAC systems by deploying additional embedded sensors to monitor temperature, humidity, and CO$_2$ levels [16], using information about occupant behaviour citesmartthermostat,observe,agarwal11, and improved modelling and control approaches [19, 17, 23, 18, 22, 21].

Modern HVAC systems contain an increasing number of sensors and remotely controlled actuators. While the inclusion of these smart devices enable low-cost and environmentally friendly building energy management, undetected sensor and actuator failures can result in poor air quality management and result in societal rejection of these normally efficient systems. Moreover, HVAC Fault Detection and Diagnostics (FDD) schemes which result in unpredictable or sporadic false alarm rates can deter building managers from investigating potential failures.

In addition to these issues, an energetic matter is connected to actuator failures. Modern building energy management systems require accurate HVAC control to minimize energy usage while maintaining an acceptable level of comfort for the building occupants. Thus, actuator fault detection is necessary to ensure proper
building operation, since HVAC systems are subject to various aging and operation errors which can lead to hardware malfunction. A common failure in HVAC systems occurs when the actuator "sticks" and no longer changes its set point, despite controller requests. This type of actuator failure can occur in any position: for example, a valve can be stuck fully open, fully shut, or at any intermediate setting. If an actuator sticks in an open or closed position for a certain time, energy may be wasted and the environment may result uncomfortable.

For these reasons, technological development of FDD schemes tailored for HVAC systems is paramount and has received much research effort in the recent years [15, 12, 11, 13]. Identifying low-cost, timely, and accurate methods for detecting actuator faults is of particular interest [14]. While inclusion of additional hardware components strictly for the purposes of actuator fault detection can provide accurate detection capabilities, these solutions are far more expensive to both deploy and maintain than software-based approaches. Moreover, the inclusion of additional hardware has the added drawback of further increasing the complexity of the HVAC system itself.

1.2 Thesis outline

Below is an outline of the rest of this thesis:

- in the second chapter the modelling principles of the plant and the problem formulation, necessary for developing control and fault detection techniques, are reported;

- in the third chapter a simulator developed at the very beginning as a help instrument for understanding the system physics is described;

- in the fourth chapter the methodology and the theory thanks to which it is possible to realize the HVAC fault detection are explained; in the detail, the hypothesis testing problem and the reformulation of the model depending on the available information about the system, the use of a Kalman filter to estimate the state so to obtain a more precise measure of the state, neglecting the measurement noise, are shown;

- in the fifth chapter the results achieved during this work are described and discussed;

- in the sixth chapter the conclusions about this thesis and the future work starting from it are reported.
Among all the elements of building environment, as mentioned in chapter [1], this thesis is focused on temperature, which is the main parameter affecting people’s comfort inside a room. In this chapter we first show the reasoning underlying the choice of a simple, though non-trivial, model for describing temperature evolution in buildings, starting from the first principle of thermodynamics and the heat transfer law, and then the steps leading to the system representation that will be used in the fault detection problem.

### 2.1 Physics of the plant

Accurate modelling of building thermal dynamics allows to implement advanced control strategies which can improve the performance of building HVAC systems and ensure that such systems are operating free of faults, e.g., via model based fault detectors. However, detailed models are often hard to determine due to the inherently complex nature of building thermodynamics. For example, exact temperature distribution in an air mass is characterized by complex Partial Differential Equations (PDE’s) that are seldom easy to solve [8].

The difficulties in dealing with PDE’s arise mainly for two reasons. First, to solve them one requires a significant amount of information regarding the building construction and insulating materials, location, weather forecast, time of year, occupancy, usage, equipment contained in the room, etc. Second, as PDE’s are generally solved via numerical solvers, one faces the computational problems that inherently arises from using such solvers: there exist computer aided building modelling tools that result in complex models, requiring a huge amount of measurements and computation [10].

The difficulty in employing the building thermodynamical model is in the requirement that the building thermal coefficients be known. These parameters are notorious for changing with the time of year, the opening and closing of doors and windows, the humidity, etc. For example, the value of $\alpha_{ij}$ changes when a window or door is opened or closed: additionally, it changes with the airflow between adjacent air masses, since a breeze between two rooms causes the temperatures in the
respective rooms to converge faster, thus making it very difficult to obtain accurate
descriptions for their values. Moreover, the complexity involved in modelling the
building behaviour grows exponentially with the number of parameters which are
known to vary.

While the model does capture the effects that the actuator input has on the tem-
perature of the room, identifying the parameters necessary requires significant learn-
ing and may not be attractive or tractable for building environments with volatile
changes in occupancy and air movement (such as academic building environments).

2.2 Model of the system

Since the plant we are interested in monitoring and controlling is a thermal
system, it is governed by the heat equation, reported below:

$$\frac{\partial T}{\partial t} = k \nabla^2 T$$ (2.1)

where $T(x, y, z, t)$ is a function of the three spacial variables $x, y$ and $z$, and of the
time variable $t$, $k$ is a positive constant called thermal diffusivity, and $\nabla^2$ is the
Laplacian or Laplace operator, a differential operator equaling the divergence of the
gradient of a function and occurring in several PDE’s. An example of numerical
solution of the two-dimensional heat equation is reported in figure 2.1 for the initial
time instant $t_i$ and the final time instant $t_f$: you can notice that the temperature
distribution tends to uniform on the whole domain.

![Figure 2.1: Thermal distributions of the example: (a) at time $t = t_i$; (b) at time $t = t_f$](image)

By discretizing the domain of the heat equation, we can associate a temperature
distribution to each of the air masses resulting from the space division, so we can see
the system as a network of agents described by a thermal model, each temperature
distribution being a node state; therefore we downgrade the problem from solving
**Chapter 2. Problem formulation**

PDE’s to solving *Ordinary Differential Equations* (ODE’s), namely the state equations of the nodes. Let us consider a network of $N$ interconnected nodes whose interactions can be represented by the underlying network graph $G(V,E)$, where $V = \{j\}^N_1$ is the vertex set with $j \in V$ corresponding to node $j$, and $E$ is the edge set of the graph. The undirected edge $\{j,i\} \in E$ is incident on vertices $j$ and $i$, with $i,j \in V$, if and only if nodes $j$ and $i$ interact, i.e. if air masses $j$ and $i$ are thermally coupled. The dynamics of each node is described by a first order equation:

$$m_j \dot{x}_j(t) = \sum_{i \in N_j} \alpha_{ji}(t)(x_i(t) - x_j(t)) + \beta_j d_j(t) + \varepsilon_j(t)$$  \hspace{1cm} (2.2)

where $m_j, x_j, d_j, \beta_j, \varepsilon_j \in \mathbb{R}$ are the thermal capacitance of $j$-th node, the state of $j$-th node, the input signal applied to $j$-th node, the $j$-th actuator gain and the $j$-th thermal generation term respectively; the last one is due to the humans, computers, machines and all other sources of heat present in the room. The term $\alpha_{ji}(t) \in \mathbb{R}^+$ is an unknown time-varying differential gain between nodes $j$ and $i$, representing the thermal coupling coefficient between air masses $j$ and $i$, while $N_i = \{j \mid \{i,j\} \in E\}$ is the neighbourhood of node $j$. These coupling parameters $\alpha_{ji}(t)$, involved in the heat transmission, are contained in the *weighted adjacency matrix* of the network, through which it is possible to model the interactions between the nodes.

The network considered in this work is reported in figure 2.2: it is a 5 node network with a star topology, in which the central node is the one we are interested in. Every node corresponds to a room, except node 5 that represents the outside environment: the goal is to control the climate of the central room and to detect actuator faults in it, by using also the information from the other rooms.

![Figure 2.2: Network topology for the problem of this work](figure)

Remembering from the (2.2) that the terms at the right-hand side are to be divided by the $m_j$, the weighted adjacency matrix of this network isn’t symmetric, unlike the adjacency matrix, and has the following structure:

$$A_d = \begin{pmatrix} 0 & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} & \alpha_{1,5} \\ \alpha_{2,1} & 0 & 0 & 0 & 0 \\ \alpha_{3,1} & 0 & 0 & 0 & 0 \\ \alpha_{4,1} & 0 & 0 & 0 & 0 \\ \alpha_{5,1} & 0 & 0 & 0 & 0 \end{pmatrix}$$  \hspace{1cm} (2.3)
2.3. Discretization of the system

The Laplacian matrix of the network, indicated with $L$, equals $A_d - \Delta$, where $\Delta$ is the weighted degree matrix of the network, whose expression is:

$$\Delta = \begin{pmatrix}
\sum_{j \in \mathcal{N}_1} \alpha_{1,j} & 0 & \cdots & 0 \\
0 & \sum_{j \in \mathcal{N}_2} \alpha_{2,j} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sum_{j \in \mathcal{N}_2} \alpha_{12,j}
\end{pmatrix} \quad (2.4)$$

Also every actuator has a gain that is to be scaled with the node mass, so after incorporating it in the gain, the structure of the input matrix results:

$$B = \begin{pmatrix}
\beta_1 & 0 & \cdots & 0 \\
0 & \beta_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \beta_{12}
\end{pmatrix} \quad (2.5)$$

Thus the dynamics of the system can be represented in the following matrix form:

$$\begin{align*}
\dot{x} &= -Lx + Bu \\
y &= x
\end{align*} \quad (2.6)$$

which is the structure of a linear system, although time-variant, with $A = -L$, $C = I$ and $D = O$.

2.3 Discretization of the system

Incorporating the thermal capacitance in the parameters at the right-hand side of the (2.2), the state of each node evolves according to the continuous time equation:

$$\dot{x}_j(t) = \sum_{i \in \mathcal{N}_j} \alpha_{ji}(t)(x_i(t) - x_j(t)) + \beta_j d_j(t) + \varepsilon_j(t) \quad (2.7)$$

where we kept the same notation for the scaled parameters. Using the forward Euler method [20], we can approximate the system with the following time-difference equation:

$$x_j(k + 1) = x_j(k) + \sum_{i \in \mathcal{N}_j} a_{ij}(k) (x_i(k) - x_j(k)) + b_j(k) + e_j(k). \quad (2.8)$$

The state of each node is periodically measured according to

$$y_j(k) = x_j(kT_s) + v_j(k) \quad (2.9)$$

where $T_s$ is the sampling rate, $x_j(k)$ is the temperature of air mass $j$, $y_j(k)$ is the $k$-th temperature measurement of room $j$, $v_j(k) \in \mathbb{R}$ is a zero-mean i.i.d. Gaussian Process (GP) with variance $\sigma_j$, which models the measurement noise. The discrete-time parameters for all $i, j \in V$ are scaled versions of their continuous-time counterparts:

$$a_{ij}(k) = \frac{T_s}{m_j} \alpha_{ij}(kT_s); \quad b_j(k) = \frac{T_s}{m_j} \beta_j(kT_s); \quad e_j(k) = \frac{T_s}{m_j} \varepsilon_j(kT_s).$$
Replacing (2.9) in (2.8) we can isolate the actuator diagnostics to a single node, as follows:

\[ x_j(k + 1) = x_j(k) + \sum_{i \in N_j} a_{ij}(k) (y_i(k) - y_j(k)) + b_j(k)d_j(k) + e_j(k) + n_j(k) \] (2.10)

where \( d_j(k) \) is the applied actuation input such that \( d_j(k) = u_j(k) \), \( u_j(k) \) being the real control input, when the actuator is operating correctly and \( d_j(k) = d_j(k - 1) \) when it fails (i.e. the actuator sticks); \( n_j(k) \) is a zero-mean GP, modelling process noise, with covariance:

\[ \text{cov}(n_j(k)) = \left( \sum_{i \in N_j} \hat{a}_{ij}(k) \right)^2 \sigma_j + \sum_{i \in N_j} \hat{a}_{ij}^2(k) \sigma_i; \] (2.11)

when \( \hat{a}_{ij}(k) = a_{ij}(k) \) the discrete dynamics (2.10) is statistically equivalent to the discrete-time model before the replacing.

It is important to notice the difference between the models (2.8) and (2.10): the first is deterministic while the latter is stochastic because of the presence of the noise. This allows the use of probabilistic techniques for doing fault detection and diagnostics.
In order to better understand network dynamics and as a help instrument for validating the model through simulations, a simulator has been developed using MATLAB® environment. Since we decided to associate a temperature dynamics to each of the air masses monitored by the sensors of the testbed, which will be presented in chapter 5, the resulting model is a network in which each agent has a single state that represents air mass temperature. Thus the system has \( N \) states, \( N \) being the number of the sensors, and the generic node dynamics is modelled by equation (2.2), which is related to the following explicit form:

\[
\dot{x}_i = \frac{k_{i1}}{m_i c}(x_1 - x_i) + \frac{k_{i2}}{m_i c}(x_2 - x_i) + \ldots + \frac{k_{iN}}{m_i c}(x_N - x_i) + b_i u_i (3.1)
\]

where \( i \) is the index of the node, \( j = 1, 2, \ldots, N \) is the index of the other nodes in the network, \( m_i \) is the air mass associated to \( i \)-th node, \( c \) is the specific heat capacity of air (of course the same for each air mass), \( k_{ij} \) is the thermal conductivity of the material between \( i \)-th and \( j \)-th air masses, \( u_i \) is the input signal applied to \( i \)-th node and \( b_i \) is \( i \)-th actuator gain. Considering that each node can be linked just to some other nodes and not to all of them, the (3.1) can be written in compact form as:

\[
\dot{x}_i = \sum_{j \in N_i} a_{ij} (x_j - x_i) + b_i u_i (3.2)
\]

where \( N_i \) is the set of nodes linked to \( i \)-th node.

### 3.1 Modelling the network

At first, as the temperature sensor network was composed of 12 motes (plus central mote and intermediate mote which don’t acquire data but have only communication purposes), the system had 12 states; thus we started with a prototypal network whose generic node dynamics is modelled by the following:

\[
\dot{x}_i = \frac{k_{i1}}{m_i c}(x_1 - x_i) + \frac{k_{i2}}{m_i c}(x_2 - x_i) + \ldots + \frac{k_{iN}}{m_i c}(x_N - x_i) + b_i u_i = \sum_{j \in N_i} a_{ij} (x_j - x_i) + b_i u_i (3.3)
\]
The topology of this network is reported in figure 3.1, where the interconnections between the nodes are shown in detail. You can notice that the Water Tank Lab volume was split into four parts, so to have an air mass for each sensor. To evaluate air mass values in each node, we used the formula \( m = \rho \cdot V \), with \( \rho = 1.204 \) being air density at 20\(^\circ\)C and \( V \) being air volume, computed multiplying floor area of a room (or part of it) by its height; choosing a great mass for node 12, we modelled the stationarity of outside temperature. The computed values for air masses are reported in table 3.1. The specific heat capacity of air (at 0 - 20\(^\circ\)C and 1atm) is \( c = 1.007 \), while, depending on the material between two air masses, we chose the following indicative values for thermal conductivities:

- outside masonry \( k_o = 0.08 \);
- inside masonry \( k_i = 0.24 \);
- air convective coefficient (at 1m/s) \( h = 4.30 \).

<table>
<thead>
<tr>
<th>( m_1 )</th>
<th>81.526 kg</th>
<th>( m_2 )</th>
<th>81.526 kg</th>
<th>( m_3 )</th>
<th>81.526 kg</th>
<th>( m_4 )</th>
<th>81.526 kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_5 )</td>
<td>37.757 kg</td>
<td>( m_6 )</td>
<td>18.964 kg</td>
<td>( m_7 )</td>
<td>13.186 kg</td>
<td>( m_8 )</td>
<td>157.734 kg</td>
</tr>
<tr>
<td>( m_9 )</td>
<td>157.734 kg</td>
<td>( m_{10} )</td>
<td>162.299 kg</td>
<td>( m_{11} )</td>
<td>347.936 kg</td>
<td>( m_{12} )</td>
<td>( 10^8 ) kg</td>
</tr>
</tbody>
</table>

Table 3.1: Computed masses for the model implementation
3.1. Modelling the network

Weighted adjacency matrix, which models network topology, has the following structure:

\[
\text{Adj} = \begin{pmatrix}
0 & a_{1,2} & 0 & a_{1,4} & 0 & 0 & 0 & 0 & 0 & a_{1,10} & a_{1,11} & 0 \\
0 & a_{2,1} & 0 & a_{2,3} & 0 & 0 & 0 & 0 & 0 & a_{2,9} & 0 & 0 \\
0 & a_{3,2} & 0 & a_{3,4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{3,12} \\
a_{4,1} & 0 & a_{4,3} & 0 & a_{4,5} & a_{4,6} & a_{4,7} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a_{5,4} & 0 & a_{5,6} & 0 & a_{5,8} & 0 & 0 & 0 & a_{5,11} \\
0 & 0 & 0 & a_{6,4} & a_{6,5} & 0 & a_{6,7} & a_{6,8} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a_{7,4} & 0 & a_{7,6} & 0 & a_{7,8} & 0 & 0 & 0 & a_{7,12} \\
0 & 0 & 0 & a_{8,5} & a_{8,6} & a_{8,7} & 0 & 0 & 0 & a_{8,11} & a_{8,12} & 0 \\
0 & a_{9,2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{9,10} & 0 & a_{9,12} \\
a_{10,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{10,9} & 0 & a_{10,11} & 0 \\
a_{11,1} & 0 & 0 & 0 & a_{11,5} & 0 & 0 & a_{11,8} & 0 & a_{11,10} & 0 & 0 \\
0 & 0 & a_{12,3} & 0 & 0 & 0 & a_{12,7} & a_{12,8} & a_{12,9} & 0 & 0 & 0 \\
\end{pmatrix}
\]

(3.4)

Its generic coefficient is \(a_{ij} = k_{ij} / m_i c\), so it loses symmetry, unlike adjacency matrix.

The dynamics matrix of the system is given by the difference between weighted adjacency matrix and weighted degree matrix, whose expression is:

\[
\Delta = \begin{pmatrix}
\sum_{j \in \mathcal{N}_1} a_{1,j} & 0 & \cdots & 0 \\
0 & \sum_{j \in \mathcal{N}_2} a_{2,j} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sum_{j \in \mathcal{N}_{12}} a_{12,j}
\end{pmatrix}
\]

(3.5)

where \(\mathcal{N}_i\) is the set of nodes linked to \(i\)-th node. So the dynamics matrix is \(A = \text{Adj} - \Delta\), that is the opposite of the Laplacian matrix of the network, indicated with \(\mathcal{L}\); therefore the relation between them is \(A = -\mathcal{L}\).

Concerning the inputs, since air inlets, AC modules and radiators are installed in each of the monitored rooms, we modelled the actuators as separate signals, one for each room (except the outside environment that has no actuator), which makes the input matrix diagonal. Since the actuators are basically valves, we decided to express every input signal as a percentage of the full opening of the valve. Moreover, every actuator has a gain that is to be scaled with the node mass (see the model reported in chapter 2); this gain is the same for every actuator and its computed value is \(b = 2.4 \times 10^{-2}\). Dividing by the mass of each node, the input coefficients become \(b_j = b / m_j\), for \(j = 1, 2, \ldots, 11\), and \(b_{12} = 0\), as for the outside node there is no actuator. Hence the structure of the input matrix results:

\[
B = \begin{pmatrix}
b_1 & 0 & \cdots & 0 \\
0 & b_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & b_{12}
\end{pmatrix}
\]

(3.6)

and network model becomes then:

\[
\begin{align*}
\dot{x} &= -\mathcal{L}x + Bu \\
y &= x
\end{align*}
\]

(3.7)
Chapter 3. Simulator

that is the classical structure of a linear time-invariant system, with $A = -L$, $C = I$ and $D = O$.

### 3.1.1 Simulator implementation

The developed simulator is a software instrument that has the ability of changing the topology of the network: via interface or script, the user can choose the number of nodes and set the weight of the edges (if a link between two nodes doesn’t exist, you will set its weight to zero). The program, which finds the numerical solution of system equations, i.e. the ODE’s describing the behaviour of the system in continuous time, is entirely written in MATLAB® language; this means that no use of SIMULINK® has been made to simulate the network, which has led to a great time saving while running the code.

The numerical solver implements Crank-Nicolson method for ODE’s: it is an unconditionally stable method and it preserves stability when passing from continuous to discrete time. Since it is an implicit method, i.e. a system of algebraic equations must be solved at each time step, Newton-Raphson method has been implemented to solve it [4]. Recursion is the keystone of the program, as it is the way to let the two aforementioned methods work together.

### 3.2 Simulation results

Several simulations were performed with different initial conditions and using different input signals. Two typical behaviours of the network are shown in figures 3.2 and 3.3, starting from the same initial conditions, namely:

$$x = \begin{bmatrix} 22 & 22 & 22 & 22 & 19 & 18.5 & 18.5 & 18 & 20 & 20 & 14 \end{bmatrix}^T \degree C. \quad (3.8)$$

In the first, the natural response of the network is reported: you can notice that every node state converges to the same value, which is the average of the initial conditions; as this is dominated by the huge mass of the outside environment, the hole network reaches the outside temperature.

In the latter, the step response of the network is reported: the input was applied just on node 1 at 100% and you can notice that every node state converges to a certain value except the outside node state, that remains constant, which deals with the inability of controlling environmental conditions.

You can notice from the plots that the time constant is about 1 hour, which is confirmed both by hand computation and by experimental evaluation.
3.2. Simulation results

Figure 3.2: State evolution with no input

Figure 3.3: State evolution with step input applied at $t=0$
The model obtained in the problem formulation, reported in chapter 2, is necessary to carry out FDD, as the methodology for detecting actuator failures is to be applied to a mathematical representation of the plant. In this chapter we present the techniques and the theory thanks to which it is possible to perform FDD, starting from hypothesis testing up to the problem solution.

4.1 Fault detection

Fault detection and diagnostics is a subfield of control engineering which deals with monitoring a system, recognizing the occurrence of a fault, and identifying the type of fault and its location.

The main assumption in this work is that the parameters describing the model are constant. We concede that many practical problems arising in quality control, fault detection and monitoring, can be modelled with the aid of parametric models allowing abrupt changes at unknown time instants. By abrupt changes, we mean changes in plant characteristics that occur very fast with respect to the sampling period of the measurements, if not instantaneously. Because a large part of the information contained in the measurements lies in their nonstationarities, and because most of adaptive estimation algorithms basically can follow only slow changes, the detection of abrupt changes is an open research issue in many applications [3].

A way of dealing with detection problems is to use a statistical approach, formulating a hypothesis testing problem: given two models, the first describing the system in normal conditions, the latter describing it in case of fault, one should be able to understand in which situation the system is operating on the basis of the collected measurements. The following paragraphs introduce the theoretical arguments behind this method.

4.1.1 Hypothesis testing of actuator failures

We shall briefly review hypothesis testing, which is the base for the detection problem. Suppose to have two different candidate models for a physical system, each
model describing a different behaviour of the plant: system $S_0$ generates measurements which are distributed as $N[m_0, R_0]$, while system $S_1$ generates measurements which are distributed as $N[m_1, R_1]$, where $N[m_i, R_i]$ represents a multivariate normal distribution of mean $m_i$ and covariance $R_i$. The normal (also called Gaussian) distribution is characterized by a probability density function of the form:

$$f_{\theta_i}(x) = (2\pi)^{-N/2}|R_i|^{-1/2} \exp\left(-\frac{1}{2}(x - m_i)^T R_i^{-1}(x - m_i)\right)$$

$$\theta_i = (m_i, R_i)$$

(4.1)

where $N$ is the dimension of the measurement vector $x$. If system $S_0$ is the appropriate model, we say that hypothesis $H_0$ is in force, while if the model $S_1$ is the appropriate model, we say that the alternative $H_1$ is in force. The hypotheses are denoted as follows:

$$H_0 : X : N[m_0, R_0]$$
$$H_1 : X : N[m_1, R_1]$$

(4.2)

where $X$ is the observation vector, whose realizations are the measurements $x$ [5, 6].

Knowing about the above concepts, we can formulate a hypothesis testing problem for detecting actuator failures. Let the test employed to denote the decision whether the $j$-th actuator is working properly, depending on the vector of measurements at the current time step, be:

$$\phi_j(y_j(k)) = \begin{cases} H_0 & s_j(y_j(k)) \leq \eta_0 \\ H_1 & s_j(y_j(k)) \geq \eta_1 \\ H_{-1} & \text{otherwise} \end{cases}$$

(4.3)

which means that the hypothesis $H_0$ is in force when the $j$-th actuator is working properly, while the hypothesis $H_1$ is in force when the $j$-th actuator has become stuck; the hypothesis $H_{-1}$ is in force if there is not enough information to choose between the previous actuator conditions.

The decision is made using the Sequential Probability Ratio Test (SPRT), which is a sequential hypothesis test: this kind of tests have the sample size not fixed in advance. Instead data are evaluated as they are collected, and further sampling is stopped in accordance with a pre-defined stopping rule as soon as significant results are observed, letting a conclusion be sometimes reached at a much earlier stage than it would be possible with more classical hypothesis testing [24]. SPRT is strictly related to log-likelihood ratio, as its expression is:

$$s_j(y_j(k)) = s_j(y_j(k - 1)) + \ln \left( \frac{f_j(y_j(k)|d_j(k) = d_j(k - 1))}{f_j(y_j(k)|d_j(k) = u_j(k))} \right)$$

(4.4)

namely it is the cumulative sum of the log-likelihood ratio, updated every time that new data arrive. Notice that the distributions of the measurements under $H_1$ and $H_0$ are conditioned to the current measurement supposing respectively that the actuation signal at the current time step equal the one at the previous time
step (actuator stuck), and that the actuation signal equal the real applied control (actuator working).

The aforementioned $\eta_0$ and $\eta_1$ are the test thresholds, chosen using Wald’s approximation [24] as reported below:

$$
\eta_0 = \ln \left( \frac{P_M}{1 - P_{FA}} \right)
$$

$$
\eta_1 = \ln \left( \frac{1 - P_M}{P_{FA}} \right)
$$

where $P_M$ and $P_{FA}$ denote the maximum probability of false alarm and the maximum probability of miss, respectively.

### 4.2 Problem solution

Since we are interested in actuator failures, causing sudden changes of system parameters which we want to detect even under perturbations (like in the case of windows and doors open), we shall now apply the aforementioned techniques to the problem faced in this work. While it is known that the building model parameters are time-variant, over short periods of time or when the building is not being utilized, the building parameters tend to remain constant.

We want to design a detector capable of determining whether the actuator has failed (has become stuck) or is working properly (not stuck). Assuming that the parameters are constant, namely:

$$
a_{ij}(k+1) = a_{ij}(k), \quad b_j(k+1) = b_j(k), \quad e_j(k+1) = e_j(k)
$$

we remind that they should be estimated online, while performing detection. This is the motivation, besides the fact that the measurements are affected by noise, leading us to employ a Kalman filter which can estimate the state of an enlarged system, that is to say, a system including the central room temperature and the plant parameters as state variables. Generalizing this reasoning, we shall focus on the $j$-th room, monitored by the $j$-th sensor providing the measurement $y_j(k)$; hence the discrete-time dynamics for this new stochastic system can be written as:

$$
\begin{align*}
\left\{ 
\begin{array}{l}
z_j(k+1) = A(d_j(k))z_j(k) + w_j(k) \\
y_j(k) = Cz_j(k) + v_j(k)
\end{array}
\right.
\end{align*}
$$

where

$$
z_j(k) = \left[ x_j(k) \quad a_j^T(k) \quad b_j(k) \quad e_j(k) \right]^T
$$

$$
A(d_j(k)) = \begin{bmatrix}
1 & y(k) - 1y_j(k) & d_j(k) & 1 \\
0 & I & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

$$
C = \begin{bmatrix}
1 & 0^T & 0 & 0
\end{bmatrix}
$$
with \( a_j(k) = \{ a_{ij}(k) | \forall i \in N_j \} \) and \( y(k) = \{ y_i(k) | \forall i \in N_j \} \) being vectors including the thermal coefficients with respect to node \( j \) and the measurements of the other nodes, respectively.

The measurements have a normal distribution, written as:

\[
f_j(y_j(k)) = \frac{1}{\sqrt{2\pi (C P_j(k)C^T + V_j)}} \exp \left( -\frac{1}{2} \frac{(y_j(k) - C m_j(k))^2}{C P_j(k)C^T + V_j} \right)
\]

(4.7)

where the terms

\[
\begin{align*}
m_j(k + 1) &= (A(d_j(k)) - L_j(k)C) m_j(k) + L_j(k)y_j(k) \\
L_j(k) &= A(d_j(k))P_j(k)C^T (C P_j(k)C^T + V_j)^{-1} \\
P_j(k + 1) &= W_j + (A(d_j(k)) - L_j(k)C) P_j(k)A^T(d_j(k))
\end{align*}
\]

(4.8)

are the state estimate, the covariance of the estimate error, and the gain matrix, that is to say, the equations of the Kalman filter.

A test, \( \phi_j(y_j(k)) \in \{ H_0, H_1, H_{-1} \} \) is employed to make the decision whether the actuator \( j \) is working properly \( (\phi_j(y_j(k)) = H_0) \), has failed \( (\phi_j(y_j(k)) = H_1) \), or there is not enough information to choose between the two previous cases \( (\phi_j(y_j(k)) = H_{-1}) \). This decision is made using a [SPRT], introduced in the previous paragraph but reported below together with the other terms for clarity:

\[
\phi_j(y_j(k)) = \begin{cases} 
H_0 & s_j(y_j(k)) \geq \eta_1 \\
H_1 & s_j(y_j(k)) \leq \eta_0 \\
H_{-1} & \text{otherwise}
\end{cases}
\]

(4.9)

where \( s_j(y_j(k)) \), which is basically a cumulative sum of the log-likelihood ratio, has the following expression:

\[
s_j(y_j(k)) = s_j(y_j(k - 1)) + \ln \left( \frac{f_j(y_j(k)|d_j(k) = d_j(k - 1))}{f_j(y_j(k)|d_j(k) = u_j(k))} \right)
\]

(4.10)

The terms \( \eta_0 \) and \( \eta_1 \) are the test thresholds, written as:

\[
\eta_0 = \ln \frac{P_M}{1 - P_{FA}} \quad \text{and} \quad \eta_1 = \ln \frac{1 - P_M}{P_{FA}}
\]

(4.11)

with \( P_{FA} \) and \( P_M \) being the maximum probability of false alarm and the maximum probability of miss, respectively.
Experimental results

The theory underlying the possibility of detecting faults (see chapter 4) and its application to a thermal system (see chapter 2), which is the case of a building controlled by an HVAC system, were previously shown. In order to verify that the problem formulation be correct, namely, to check that the chosen model be a good representation of the reality, and to ensure that the aforementioned methodology do not bring just theoretical results, it is necessary to realize a testbed through which it is possible to obtain the acquisition of data. In this chapter we first describe the testbed realized for this work and used to establish the validity of the adopted fault detection strategy, and then we present and discuss the experimental results obtained from it.

5.1 Testbed realization

The testbed for this project is located on the 2nd floor of Q-building, which hosts the School of Electrical Engineering at KTH main Campus. The temperature measures are carried out by a Wireless Sensor Network (WSN), made up of several sensor nodes called motes. Such sensor networks has become popular in BA since they are cheap and there is no need of laying wires. The sensors used for this project are Tmote sky of Moteiv Corporation, one of which is shown in detail in figure 5.1: they are very small and optimized for low energy consumption. Their board can be powered either by USB supply or by battery pack; the data transmission is wireless in both cases [7].

The WSN covers an area of about 250 m\(^2\), distributed over 5 different rooms: Laboratory 2 (referred to as experimental room), which we want to monitor and where the data acquisition terminal is, PCB Lab, storage room, corridor and outside area. An overview of the network is shown in figure 5.2. In the place where the testbed is installed a real HVAC system is already working. This system occupies the entire area where the testbed is installed and it includes some radiators for heating, fan coil for the air conditioning and some vents for the fresh and clean air from the outside. The control input for the valves of the system is automatic or manual: if the control is automatic a Supervisory Control And Data Acquisition (SCADA) system
regulates the temperature in the rooms, depending on some different environmental factors; if the control is manual it is possible to regulate the valve opening of the actuators.

The testbed developed in order to acquire and store the temperature data is characterized by some parameters, among which the main is the sampling time $T_s$: this is the distance between two consecutive measurements and it is possible to change it as required when programming the motes. Problems concerning data transmission and loss, overcome by using oversampling and linear data interpolation, arose because of the thickness of walls, the transmission protocol, etc.
5.2 Experimental results

During the experiments, we wish to detect an actuator failure in the air conditioning system, namely whether the cooling value is stuck or not. To emulate an actuator failure, we simply do not apply the calculated control value to the cooling valve actuator and merely leave it in its current state (as if it were stuck). During all of the experiments, the door of the experimental room was shut such that the only changes in the air mass interactions were caused by the opening and closing of the windows. Under these testing conditions, the model-based detector was evaluated under four different scenarios, namely:

- windows closed and actuator working;
- windows open and actuator working;
- windows closed and actuator stuck;
- windows open and actuator stuck.

The experimental results for evaluating the model-based detector are shown in figure 5.3, where in all four subplots the horizontal axis represents the time in minutes since the test began, and the vertical axis indicates the log-likelihood statistic value for the tests performed under the respective testing scenario, as calculated by using (??).

The two dashed lines in each subplot indicate the decision regions of the test, $\phi_j(y_k)$, where it is decided that an actuator failure occurs ($\phi_j(y_k) = H_1$) when the
5.2. Experimental results

Figure 5.3: Comparison of the results under the four scenarios

statistic is greater than the upper dashed line, and conversely, it is decided that the actuator is working properly ($\phi_j(y_k) = H_0$) when the log-likelihood statistic is less than the lower dashed line. When the statistic is between the two dashed lines, no decision can be made regarding the state of the actuator ($\phi_j(y_k) = H_{-1}$). Several experiments were performed under each of the four scenarios and every subplot represents one of the testing cases: two experiments per subplot are reported to help illustrate trends.

5.2.1 Discussion of the results

As you can see in figure 5.3, the results suggest that the detector accurately identifies whether a fault occurs regardless of the state of the windows, if given enough time; each scenario is shown in detail in figures [5.4 - 5.7]. In the cases where no fault occurs, to decide that the actuator works good, it takes about 60 minutes when the windows are closed, while it takes about 150 minutes when the windows are open, as you can notice in figures 5.4 and 5.5, respectively. These results indicate that when nothing wrong happens, it takes more than two times longer to decide that everything is fine in the open windows case, compared to the closed windows case. To find an explanation to detector behaviour, we should remember that, when the windows are open, inside air mixes quickly with outside air, affecting system parameters which are estimated online; this situation leads to a slowing down of the diagnostics.
Figure 5.4: Detail of the experiments with no fault and windows closed

Figure 5.5: Detail of the experiments with no fault and windows open
In the cases where a fault occurs, to detect the failure it takes about 170 minutes when the windows are closed, while it takes about 220 minutes when the windows are open, as you can notice in figures 5.6 and 5.7 respectively. In the closed

![Figure 5.6: Detail of the experiments with fault and windows closed](image)

![Figure 5.7: Detail of the experiments with fault and windows open](image)
windows conditions, the reason underlying the longer time needed by the detector to identify a fault, compared to the time to decide that a fault did not occur, is that the actuation signal is basically constant. This implies that the system is under-stimulated, making parameter estimation, and thus detection, harder. Of course the open windows/occurring fault scenario combines the effects of the two aforementioned delays, stretching detection time even more.

You can see in figure 5.7 that the blue line don’t reach the upper threshold, which implies that no decision is taken about actuator condition. This means that, given a certain time, if the detector is not able to take a decision we have to modify the thresholds, reducing the region of uncertainty \( \phi_j(y_k) = H^{-1} \) by changing the probabilities of false alarm and miss. In summary the results show that FDD can require significant monitoring periods, ranging on the scale of hours. Since the test is based on the assumption that the room parameters do not change, it is likely that fault detection must be performed at night, when it is improbable that the state of windows and doors, as well as occupancy, will change.

Now we would like to point out an interesting characteristic of the SPRT: in all the plots, it displays a stepped trend. To better understand this remark, observe figure 5.8, reporting the statistic relating to one of the experiments without failure, and the actuation signal. As we chose to cool the room during the experiments, the actuator input is a 0-1 square wave, 1 being the full cooling effort. You can see a certain regularity in the flat zones of the statistic, corresponding to actuator turning on and off.

We can find the reason underlying this behaviour in figure 5.9, where central room measured temperature, actuation signal and temperature estimates under the two hypotheses are shown. Notice that by the points of turning on and off, the two estimates are very close: this implies that the statistic is almost constant in a neighbourhood of these points, as in this problem SPRT is just an index of the divergence between the estimates under the two hypotheses.
5.2. Experimental results

Figure 5.8: Statistic behaviour with respect to actuation signal

Figure 5.9: Measured temperature and estimates compared to actuation signal
Conclusions and future work

In this thesis a strategy for detecting actuator failures in HVAC systems has been developed and successfully tested on a real plant. Unlike other model-based approaches to actuator FDD, this method estimates dynamically the model parameters while performing detection and does not require occupancy data, a priori building parameter assumptions, or centralized computation since each room only requires measurements of the adjacent rooms (or outside air temperature) to test its actuator.

The experimental evaluation has revealed that the model-based detector requires significant monitoring periods with constant building dynamics to accurately determine whether an actuator has failed. Since the model-based strategy requires that the building dynamics do not change during its monitoring period, it is likely that the model-based detector can only be performed at night. Moreover, fault detection schemes that require long monitoring periods may not be necessary to identify a working actuator.

Therefore the idea is to use a steady state detector capable of quickly identifying a working actuator. As logic indicates, in the event that an actuator is working, applying a significant change in the actuation input will result in a change in the measured temperature. Using this reasoning, the following section introduces a steady-state detector which can confirm that an actuator is working based on a change in the measured temperature.

6.1 Future work

Starting from the aforementioned idea, future work may aim to the development of an actuator FDD strategy using a two-tier approach that includes a dynamic model-based detector and a fast-deciding steady-state detector: the steady-state detector can quickly confirm if an actuator is working, while the model-based detector can accurately decide whether an actuator fault has occurred regardless of the interaction between the surround air masses.

The premise of the steady-state detector is that if an actuator fails, the temperature in the room is not expected to change since no change in actuation will
occur. Thus, a goodness-of-fit test \( \bar{\phi}_j(y_k) \in \{H_0, \neg H_0\} \) can be employed to determine whether the measured temperatures are likely to be explained by a steady state model, which accurately describes several possible events including, but not limited to, an actuator failure. Another more common reason, besides an actuator failure, that the steady-state model may match the measured data can include that the window is open and the outside air mixes quickly with the room air such that no change in the actuator can cause a significant change in the room temperature.

The result of this test is a decision to either accept that the actuator is working properly, or to concede that accepting the actuator is working can not be possible at this time because the reason for not accepting that the actuator is working can be described not only by an actuator failure, but also by the existence of non-steady state building conditions or a significant influence from an adjacent air mass (such as opening the windows of a room). For this reason, the steady-state detector is only capable to infer that the actuator is working.

Since it is expected that faults are infrequent, it is preferred to have a test that can quickly identify whether the actuator is working, without having the added testing complexity required to determine if the actuator has failed. Thus an architecture requiring multiple steady-state detection experiments, to decide that the measurements could be explained by an actuator failure before performing model-based detection, can be introduced. The system starts with a preliminary detector (steady-state detector) and decides either that the detector is working \((H_0)\) or that no decision can be made about the state of the actuator \((\neg H_0)\) using the quick preliminary detector. If after \(N^*\) attempts of running the preliminary detector a decision can not be made about the state of the actuator, then a primary detector (model-based detector) is evaluated until deciding whether the actuator is working \((H_0)\) or that a failure has occurred \((H_1)\).

The selection of \(N^*\) can be made based upon how willing the building operator is to run the model-based detector, which requires significantly longer decision times and will result in reduced building efficiency. However, excessive testing with the steady-state detector can also result in reduced building energy management performance, coming to a trade-off.


